The Effect of COVID-19 Pandemic on Use Patterns on Online Learning Platform and Student Performances

Jinghao Huang
jinghaoh@andrew.cmu.edu

Jenny Mao
jiayim@andrew.cmu.edu

Shihua Pei
spei@andrew.cmu.edu

Client: Norman Bier (Open Learning Initiative)

Faculty Advisor: Prof. Valerie Ventura

Abstract

The Open Learning Initiative is trying to evaluate how the COVID pandemic in 2020 has been impacted on user behaviors and students' performances. Researches have shown conflicting results on the changes in students' performances during the pandemic. Using data from Georgia State University from Spring 2019 - Fall 2020, we use weekly logins to determine user behavior changes and average assessment correctness for students performance changes. The result shows a statistically significant decrease in logins for both instructors and students after pandemic, as well as higher assessment correctness in Spring 2020. Other factors not listed in the dataset may also contribute to this effect, including the implementation of a new exam platform to better monitor remote students, and potential course policy changes due to switching to remote learning mode.

1 Introduction

Open Learning Initiative (OLI)[1] is a scientifically-based online learning environment that supports learning and instruction with high-quality, classroom-tested online courses and materials. It allows educational institutions to share courses and materials openly and freely and develop a community of use, research, and development. Researchers at Brookings Institution sampled 8,000 public schools in the US in both fall 2019 and fall 2020 because of online learning[2]. Their results show that for students in Grade 3 to Grade 9, the math scores dropped in fall 2020 significantly. This project is interested in exploring whether similar patterns can be found in college classrooms.

The goal of this project is to define any use pattern changes on the OLI platform for instructors and students at Georgia State University's statistics course from Spring 2019 - Fall 2020 due to COVID19. This project focuses on comparing OLI use patterns for both instructors and students before and during the pandemic, and investigating how students' performances are changed during the same time period.

2 Data

The data set of this project consists of information about students' performance, students' behaviors and instructors' behaviors from the spring semester of 2019 to the fall semester of 2020. The data is provided by the Open Learning Initiative in .csv format. For each of the four semesters, we have 34 .csv files.

[1] https://oli.cmu.edu/

[2] https://www.brookings.edu/blog/brown-center-chalkboard/2020/12/03/how-is-covid-19-affecting-student-learning/

	2019 Spring	2019 Fall	2020 Spring	2020 Fall
Number of students	1,524	1,672	1,686	2,078
Number of instructors identified by name	Prof. H Prof. P	Prof. H Prof. P Prof. C	Prof. H Prof. P Prof. C	Prof. H Prof. P

<Table 1: Number of students and instructors identified for each semester.>

3 Methods

The goal of this project is to see whether there were any behavior changes in a GSU Statistics course because of the COVID pandemic starting in the Spring 2020 semester. Primarily, we focus on whether there are any use pattern changes in the instructors' and students' behaviors, as well as whether there are any changes in students' performances. To determine the impact of switching to remote teaching mode in GSU at the end of March 2020, we compare user behaviors in the Fall and Spring semesters separately, due to different behavior patterns in the Fall and Spring semesters in 2019, before the pandemic.

Besides separating user behaviors by spring and fall semesters, we also focus on four instructors who taught in multiple semesters between Spring 2019 – Fall 2020. Prof. H and Prof. L taught all four semesters where Prof. P and Prof. M taught two or more semesters. The assumption is that each instructor has very different teaching styles and pace, and students in the same instructor's sections would have use patterns more similar to each other than to students in other instructor's sections. In addition, there are lots of low logins and some double student registration in Spring 2019, as well as several missing tables, we still incorporate Spring 2019 data but proceed with caution.

The main analysis for the use pattern changes is to perform 95% confidence interval and hypothesis testing, including two sample t-tests to determine whether the differences in different semesters are statistically significant. We also use various plots to visually assess the use pattern changes and performance changes. Bar plots and line plots are used to compare the average logins and average performances, and violin plots are used to compare the difference in distribution over time.

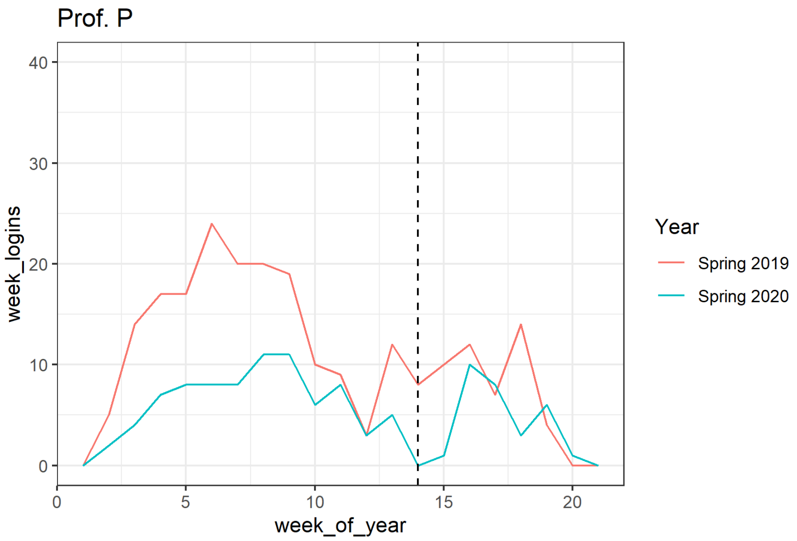
All analyses are done in Rstudio (Rstudio, 2020).

4 Results

4.1 Instructor Login Changes

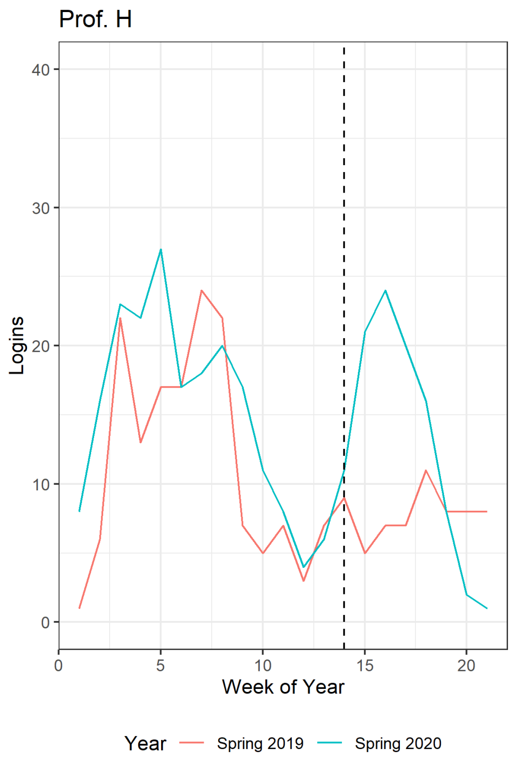
Even though there are four instructors who taught in multiple semesters from Spring 2019 to Fall 2020, there are only 2 instructors who are identified by name in the instructor .csv files. Records for all other instructors are aggregated and cannot be analyzed separately.

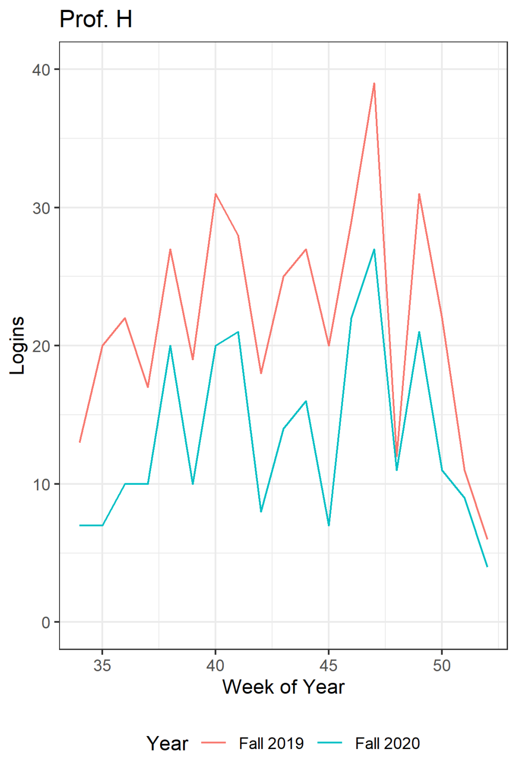
Because Prof. P did not teach in Fall 2020, we can only compare his login patterns in Spring 2019 and Spring 2020. The dash line at week 14 in Figure 1 indicates the week when GSU switched to online teaching mode. Visually, there are no significant differences in the logins patterns, before and after week 14.



<Figure 1: Logins per week before and after week 14 in Spring 2019 and Spring 2020.>

Prof. H taught in all four semesters, and from Figure 2 we observe that login patterns in the spring semesters are very similar before week 14, while there are some differences after week 14. Login patterns for Prof. H in the fall semesters have the same patterns but different magnitudes (Figure 3). One of the reasons for the decrease in number of logins in Fall 2020 is that instead of using the OLI platform to administer exams as before, GSU used a different platform to better administer exams in the remote setting.





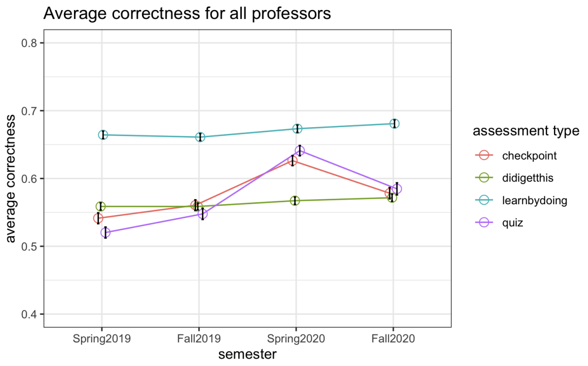
<Figure 2: Logins per week before and after week 14 in Spring 2019 and Spring 2020.>

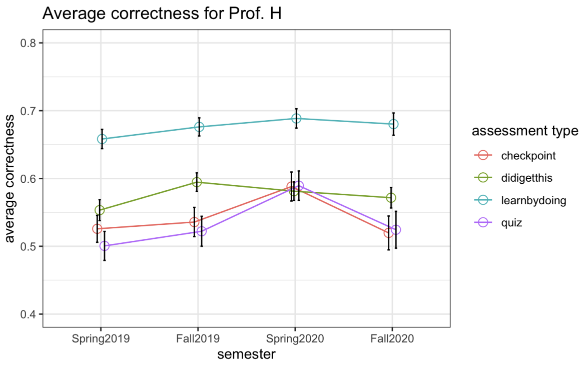
<Figure 3: Logins per week in Fall 2019 and Fall 2020.>

Through hypothesis testing we confirm there is a statistically significant decrease in instructor logins in Fall 2020 (p-value = 0.0012). Difference in the spring semesters after week 14 is not statistically significant for neither of the instructors at 5% level. Detailed R codes and plots can be found in Appendix 1.

4.2 Student Performance Changes

The plot shows the changes of average correctness for all the sections during the four semesters, grouped by type of assessments. High stake assessments include quiz and checkpoint, while low stake assessments include learn-by-doing and did-i-get-this. From Figure 4 and 5, we observe a large increase in the average correctness for high stake assessments from Fall 2019 to Spring 2020, followed by a drop in Fall 2020, returning to the average correctness level in Fall 2019. Performance on low stake assessments varies slightly before and after the pandemic.



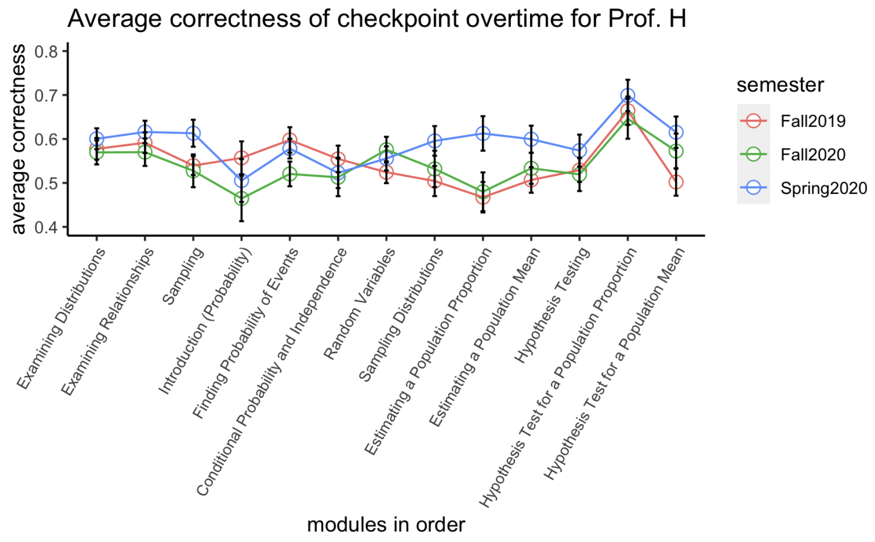


<Figure 4: Average correctness for all professors over time.>

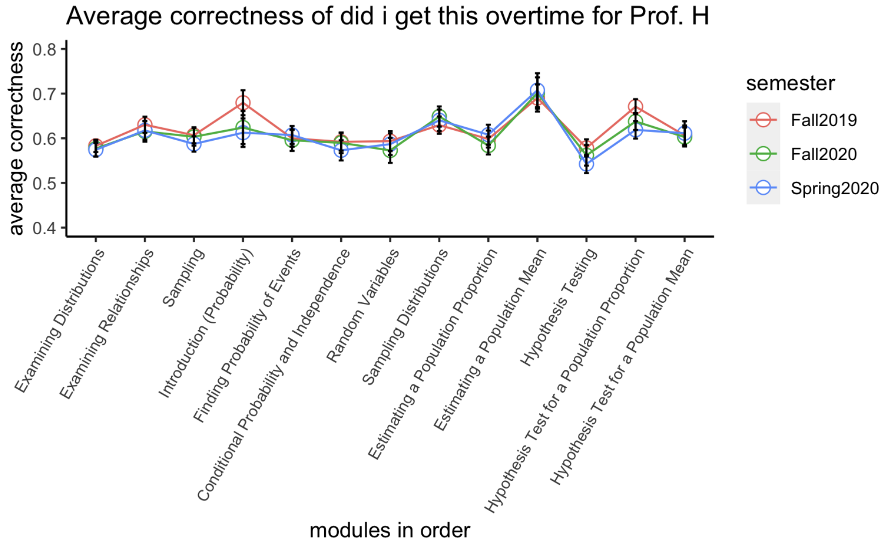
<Figure 5: Average correctness for Prof.H.>

We then dig more into the average correctness for every section of the three assessments (checkpoint/did i get this/ learn by doing) sequentially in Fall 2019, Spring 2020 and Fall 2020. Figure 6, 7 and 8 align with our previous findings.

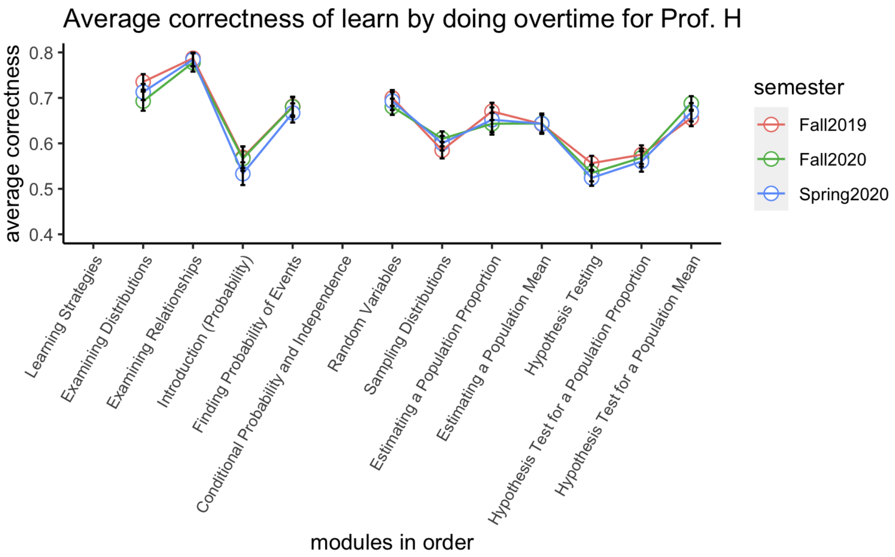
Detailed R codes and plots can be found in Appendix 2.


Average correctness of checkpoint overtime for Prof. H
average correctness
0.8
0.7
0.6
0.5
0.4
Examining Distributions
Examining Relationships
Sampling
Introduction (Probability)
Finding Probability of Events
Conditional Probability and Independence
Random Variables
Sampling Distributions
Estimating a Population Proportion
Estimating a Population Mean
Hypothesis Testing
Hypothesis Test for a Population Proportion
Hypothesis Test for a Population Mean
modules in order
semester
Fall2019
Fall2020
Spring2020

<Figure 6: Average correctness of checkpoint overtime for Prof. H.>


Average correctness of did i get this overtime for Prof. H
average correctness
0.8
0.7
0.6
0.5
0.4
Examining Distributions
Examining Relationships
Sampling
Introduction (Probability)
Finding Probability of Events
Conditional Probability and Independence
Random Variables
Sampling Distributions
Estimating a Population Proportion
Estimating a Population Mean
Hypothesis Testing
Hypothesis Test for a Population Proportion
Hypothesis Test for a Population Mean
modules in order
semester
Fall2019
Fall2020
Spring2020

<Figure 7: Average correctness of did i get this overtime for Prof. H.>

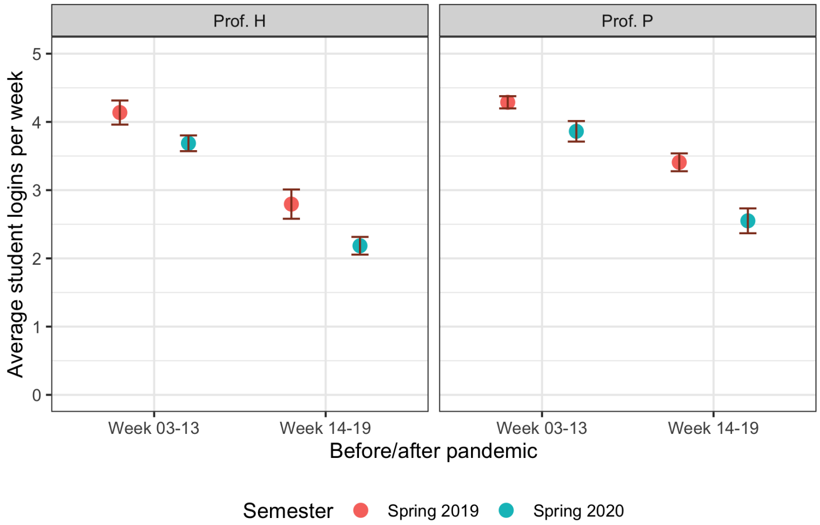


<Figure 8: Average correctness of learn by doing overtime for Prof. H.>

4.3.1 Student Login Changes

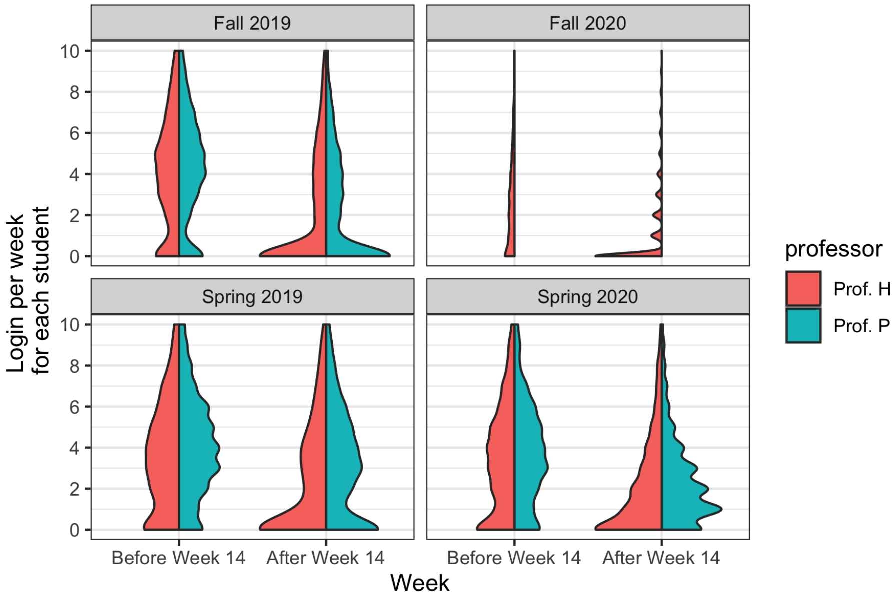
We observe that the average student logins in Spring 2020 is significantly lower than that of Spring 2019 (Figure 9). Through hypothesis testing we confirm there is a statistically significant decrease in average student logins in week 03-13 in Spring 2020 for Prof. H (p-value = 2.94e-05) and Prof. P (p-value = 2.04e-06). Similarly, there is a statistically significant decrease in average student logins in week 14-19 in Spring 2020 for Prof. H (p-value = 2.02e-06) and Prof. P (p-value = 1.12e-13).

We also observe that the average student logins after COVID-19 are lower than before COVID-19 (Figure 9). Through hypothesis testing we confirm there is a statistically significant decrease in average student logins from week 03-13 to week 14-19 in Spring 2020 for Prof. H (p-value < 2.2e-16) and Prof. P (p-value < 2.2e-16). Detailed R codes and plots can be found in Appendix 3.1.



*<Figure 9: Average student Logins per week before and after week 14 in Spring 2019 and Spring 2020 , by instructor. Red error bar calculated as 1.96*standard deviation divided by square root of number of students.>*

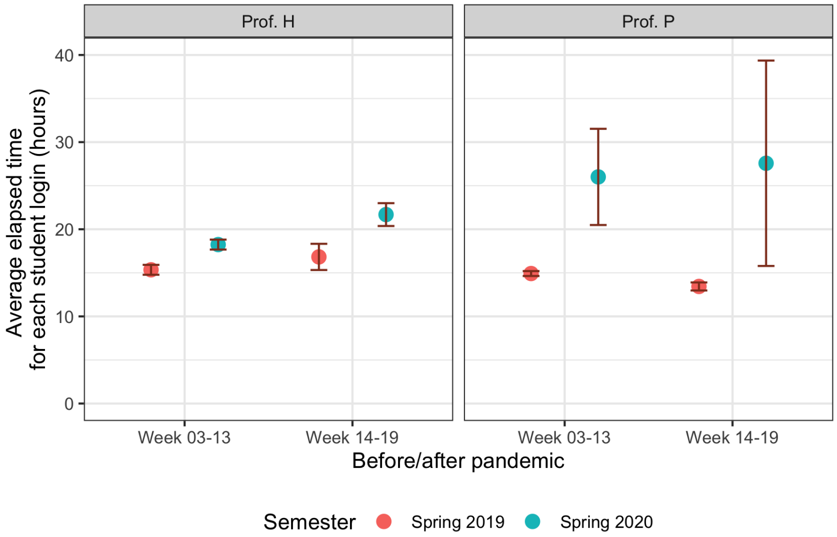
In Figure 10, we also compare the distribution of student logins per week before and after week 14 by instructors from Spring 2019 to Fall 2020. We observe that the distributions are very similar in semesters before the pandemic. The distribution of logins before week 14 in Spring 2020 is similar to the distribution of logins before week 14 in Spring 2019. However, the distribution of logins after week 14 in Spring 2020 varies from the distribution of logins after week 14 in Spring 2019.



<Figure 10: Distributions of student logins per week before and after week 14 by instructor for all four semesters.>

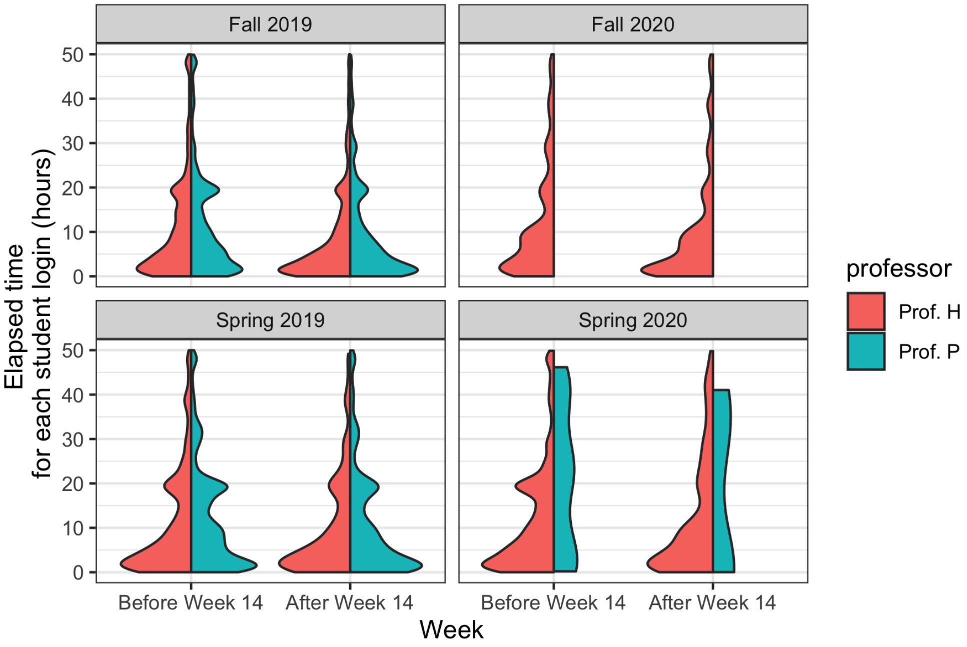
One possible reason for the low logins after COVID-19 is that students log in to the system fewer times but have longer study sessions because of remote learning. To find whether there is evidence for this conjecture, we calculate the elapsed time between the session start time and last action taken time for each student login. Notice that the elapsed time does not account for long periods of inactivity but it is calculated consistently. Besides, students will be automatically logged out of the system if they have an inactive period of two hours.

We observe that the average elapsed time for each login in Spring 2020 is significantly higher than that of Spring 2019 (Figure 11). Through hypothesis testing, we confirm there is a statistically significant increase in the average elapsed time for each login in week 03-13 in Spring 2020 for Prof. H (p-value = 1.96e-12) and Prof. P (p-value = 2.17e-04). Similarly, there is a statistically significant increase in average elapsed time in week 14-19 in Spring 2020 for Prof. H (p-value = 1.87e-06) and Prof. P (p-value = 3.29e-02).



*<Figure 11: Average elapsed time for each login before and after week 14 in Spring 2019 and Spring 2020, by instructor. Red error bar calculated as 1.96*standard deviation divided by square root of number of students>*

In Figure 12, we also compare the distribution of the average elapsed time for each login before and after week 14 by instructors from Spring 2019 to Fall 2020. We observe that the distributions are very similar in semesters before the pandemic. The distribution of the average elapsed time for students who enrolled in Prof. P's class in Spring 2020 differs from the other semesters. For students who enrolled in Prof. H's class, the distribution of their average login elapsed time before week 14 seems to be quite similar across all four semesters. However, their average login elapsed time after week 14 in Spring 2020 seems to be different from the other semesters.



<Figure 12: Distributions of elapsed time for each login before and after week 14 by instructor for all semesters.>

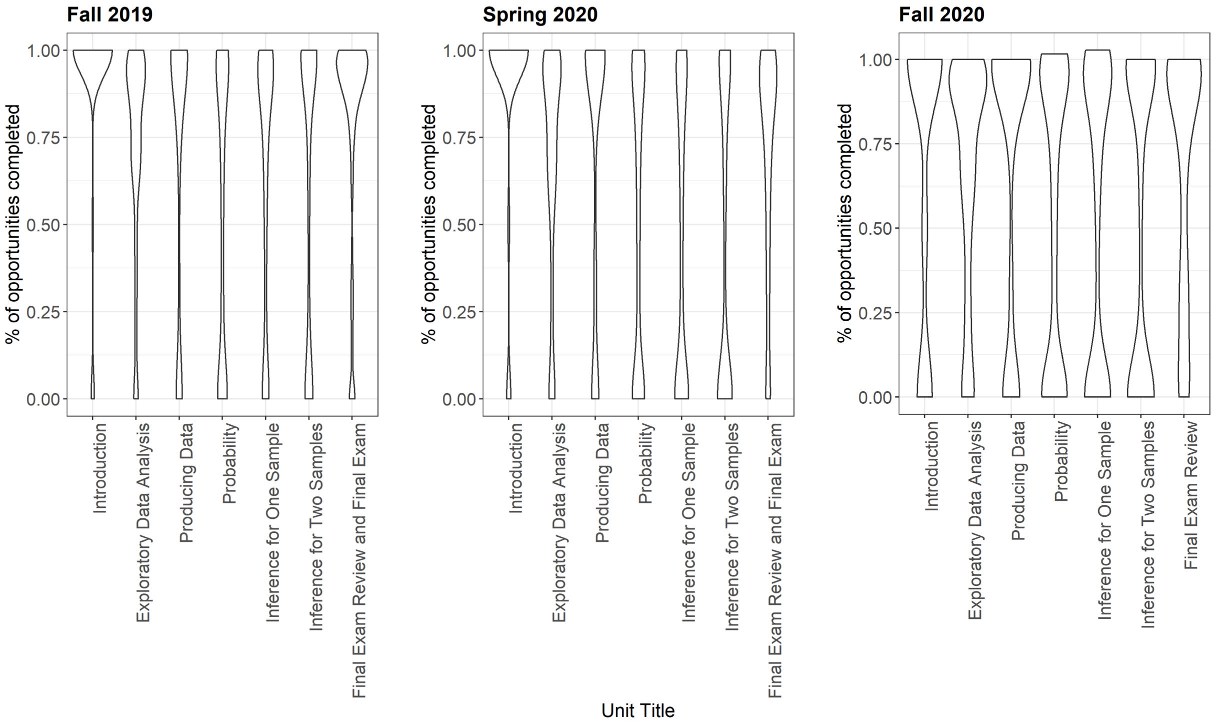
4.3.2 Student Use Pattern Change by Unit

From Figure 13, we can visually observe that the percentage of practice questions completed by unit is similar between Fall 2019 and Spring 2020, for sections taught by Prof. H, with slight increase in lower percentage completion on the two inference units. However in Fall 2020, it appears that more students are completing fewer practice questions per unit, which may indicate that students are less engaging in practicing unit questions when they are remote. In order to determine whether there is indeed a different type of student population in Fall 2020, which consistently completed low percentages of unit questions, we count the number of students who did not complete any of the unit questions in these semesters (Table 2). Table 2 shows comparisons for all four semesters except Spring 2019, as no such data is recorded in this semester.

Based on the four instructors we select, the percentages of students who did not complete any unit questions increase slightly from Fall 2019 to Spring 2020, with the exception in Prof. H's sections where the percentage doubles. The percentage increases much more in Fall 2020, especially with sections taught by Prof. M, as the percentage of students completed zero unit questions is 22 times higher than the percentage in Fall 2019. Prof. M did not teach in Spring 2020, and going directly from full in-person mode to full remote mode may increase the difficulty for students to complete unit questions. One thing to note is that Prof. M has a smaller student population in Fall 2020, compared to the student population in previous semesters.

	Percentage of Students Completed 0% of All Unit Questions		
Instructors	**Fall 2019**	**Spring 2020**	**Fall 2020**
Prof. H	2 / 238 = 0.84%	4 / 235 = 1.70%	9 / 244 = 3.69%
Prof. L	4 / 119 = 3.36%	5 / 121 = 4.13%	13 / 228 = 5.70%
Prof. P	5 / 240 = 2.08%	3 / 122 = 2.46%	-
Prof. M	1 / 234 = 0.43%	-	6 / 63 = 9.52%

<Table 2: Percentage of students complete no practice questions in all units in Fall 2019 - Fall 2020.>



<Figure 13: Percentage of practice questions completed by unit in Fall 2019 - Fall 2020, taught by Prof. H>

In addition to the lower percentage of questions completed, students also request fewer hints, and get higher overall correctness in Fall 2020 compared to Fall 2019, across sections taught by the four selected instructors. Detailed R codes and plots can be found in Appendix 3.2.

5 Discussions

Georgia State University changed to online teaching mode starting at the end of March 2020 due to the COVID-19 pandemic, and they want to explore the impact of this switch to instructors' and students' use patterns of the platform, as well as the impact to students' performance.

Based on the number of logins from the selected instructors and students in their sections, we observe that the number of logins after switching to the remote setting decreases significantly for both instructors and students compared to the number before the pandemic. One of the reasons for the change in the instructors' behavior is due to the fact that GSU changed their exam-administration platform from the OLI to better assist with the online mode. Similarly, one reason for the students' login decrease is that they have longer study sessions per login. Through hypothesis testing, we confirm that the increase in the average elapsed time is statistically significant in sections taught by Prof. P and Prof. H.

Students also behave differently towards unit questions between Fall 2019 and Fall 2020. They tend to complete less unit practice questions in Fall 2020, indicating they may be less engaging. On the other hand, they request fewer hints out of the questions completed, and also get higher overall correctness on the unit questions.

The remote learning environment also impacts the students' performances. There is a large increase in the average correctness on high stake assessments from Fall 2019 to Spring 2020, followed by a drop in Fall 2020. However, there is only slight performance variation on low stake assessments. Additionally, the result for average correctness for every section of the three assessments (checkpoint/did i get this/ learn by doing) sequentially in Fall 2019, Spring 2020 and Fall 2020 aligns with our previous findings.

One of the limitations in this analysis is that there are much fewer variables recorded for instructors. Most of the instructors' activities are aggregated in each semester, especially in Spring and Fall 2019. As a result, the significant difference in number of logins in Fall 2019 and Fall 2020 may likely be biased, as it is observed from a single instructor.

In this project we mainly use student login data and assessment scores to determine student behavior changes; however, there are lots of other effects that are not captured by our dataset. For instance, the change of course policy due to COVID will likely impact student performances. If instructors allow students to drop more high/low stake assessments than before, students may perform differently than before. With the implementation of a new exam platform in Fall 2020, which better monitor remote students, there may be changes not captured in the dataset, but could have an impact on student performances.

Reference

Bier, N. (2020), 36-726 Capstone Description. Unpublished assignment sheet, Open Learning Initiative, Carnegie Mellon University, Pittsburgh PA.

Kuhfeld, M., Soland, J., Tarasawa, B., Johnson, A., Ruzek, E., & Lewis, K. (2020, December 3). *How is COVID-19 affecting student learning?* Brookings. https://www.brookings.edu/blog/brown-center-chalkboard/2020/12/03/how-is-covid-19-affecting-student-learning/.

RStudio Team (2020). R Studio: Integrated Development Environment for R. RStudio, PBC, Boston MA. URL http://www.rstudio.com/.

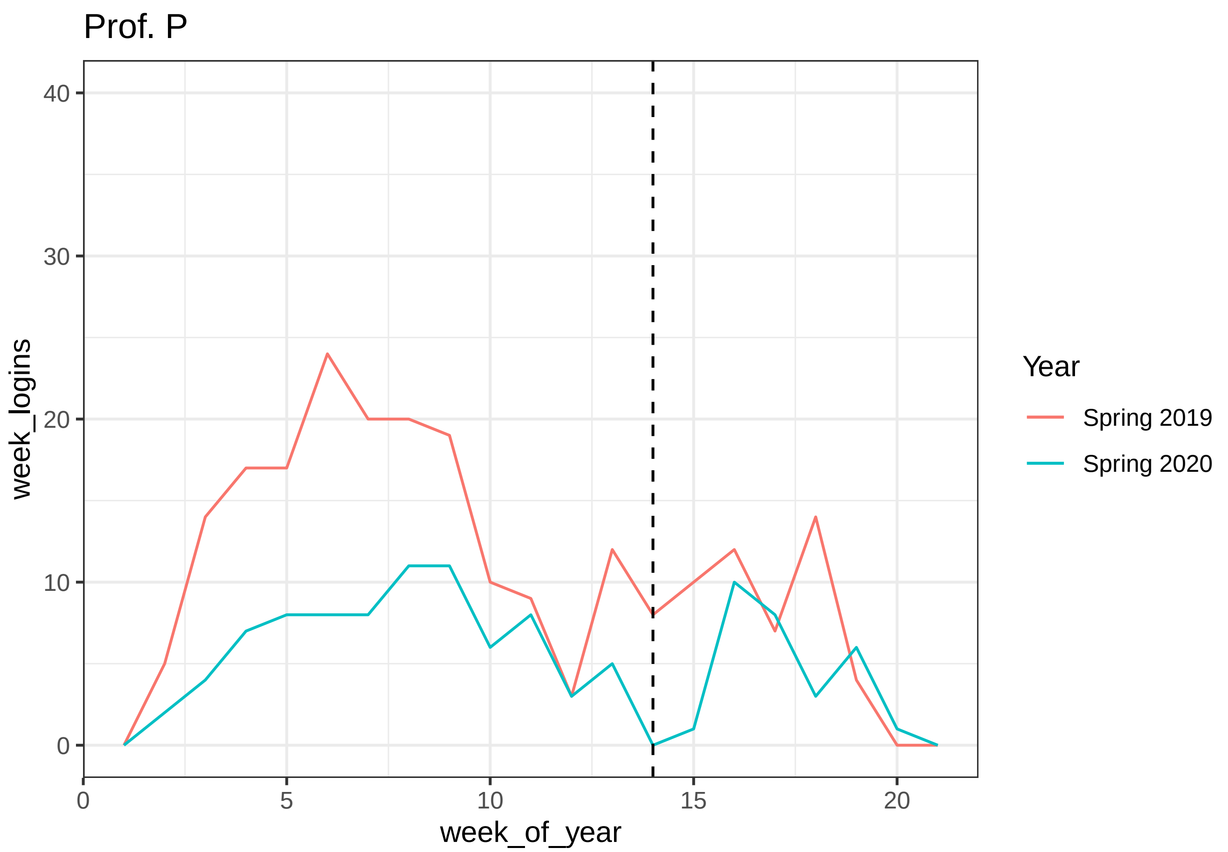
Appendix 1

```
library(tidyverse)
library(lubridate)
```

```
# Spring 2019
setwd("~/CMU Coursework/Spring/726 Capstone/OLI/data/Export/s19")
instructor_action_s19 <- read.csv("instructor_action.csv", header = T)
instructor_action_s19_new <- read.csv("instructor_action_new.csv", header = T)

# Fall 2019
setwd("~/CMU Coursework/Spring/726 Capstone/OLI/data/Export/f19")
instructor_action_f19 <- read.csv("instructor_action.csv", header = T)

# 2020
setwd("~/CMU Coursework/Spring/726 Capstone/OLI/data/Export/f20")
instructor_action_exam <- read.csv("instructor_action.csv", header = T)
instructor_action_2020 <- read.csv("instructor_action_all.csv", header = T)
courses <- read.csv("course.csv", header = T)
courses$start_date <- mdy_hm(courses$start_date)
courses$end_date <- mdy_hm(courses$end_date)
courses <- courses %>%
  mutate(semester = ifelse((month(start_date) == 8 & day(start_date) == 24),
                           "Fall",
                           ifelse(month(start_date) == 1, "Spring",
                                  "Summer")))
course_instructor_2020 <- full_join(instructor_action_2020, courses,
                                    by = "course_key")
setwd("~/CMU Coursework/Spring/726 Capstone/OLI/report")
```

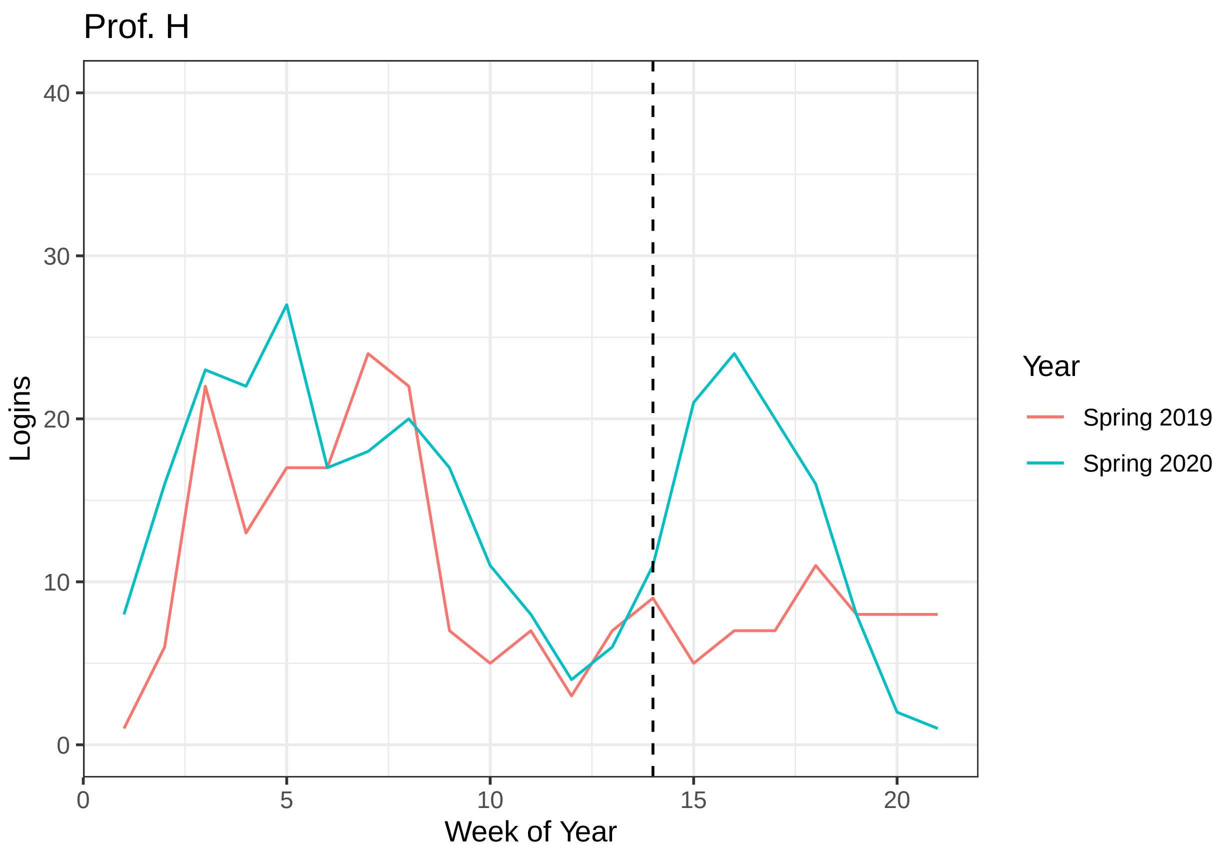
Instructor Weekly Login

```
BP_login <- instructor_action_s19 %>%
  filter(user_id == "BPidgeon7033", course_key == 10405)

BP_login <- BP_login %>%
  mutate(week_logins_2020 = unlist(course_instructor_2020 %>%
           filter(semester == "Spring", user_id == "BPidgeon7033",
                  course_key == 21313) %>%
           select(week_logins)))

BP_login %>%
  ggplot(aes(x = week_of_year)) +
  geom_line(aes(y = week_logins, color = "Spring 2019")) +
  geom_line(aes(y = week_logins_2020, color = "Spring 2020")) +
  scale_y_continuous(breaks = c(0, 10, 20, 30, 40), limits = c(0, 40)) +
  xlim(1, 21) +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  labs(title = "Prof. P",
       color = "Year")
```

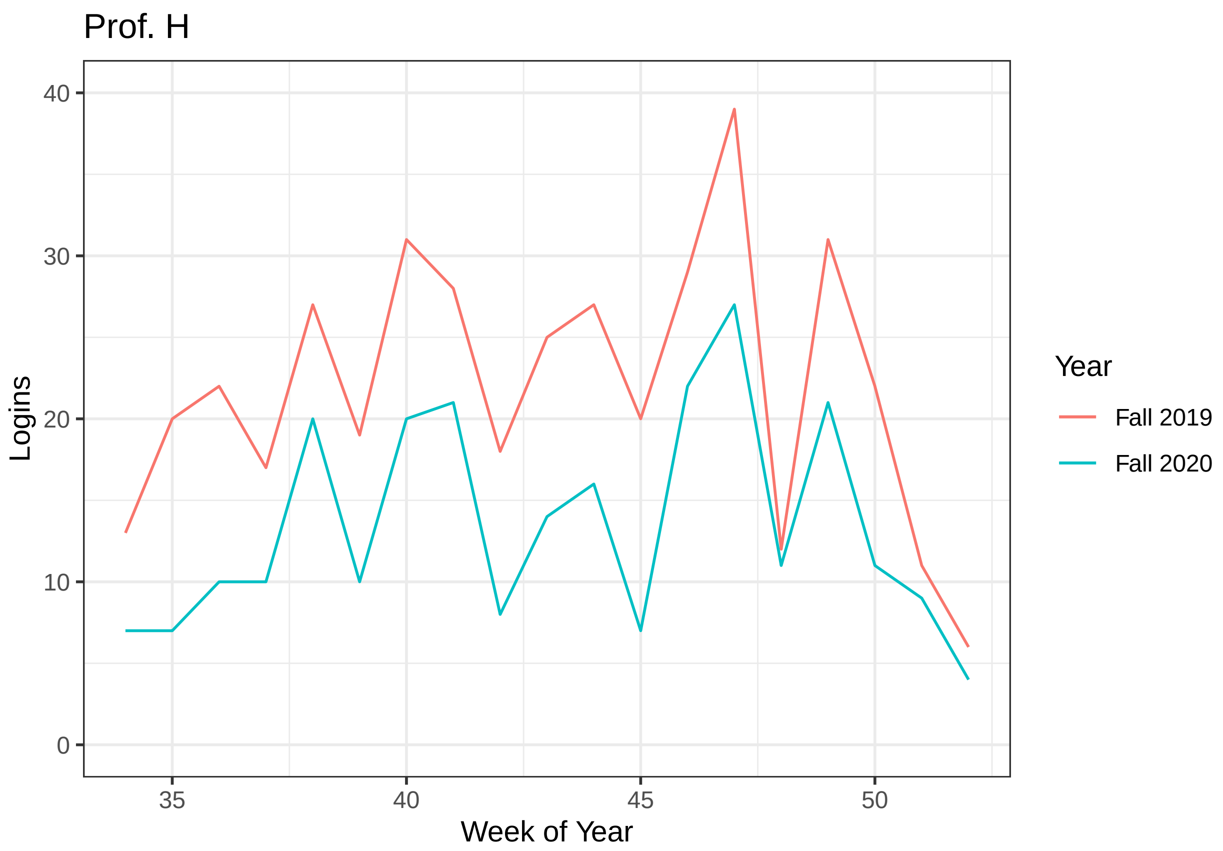


This shows Figure 1 in the report. Week 14 is when GSU changed to remote learning mode. It does not look like the login patterns changes much for Prof. P.

```
DH_login <- instructor_action_s19 %>%
  filter(user_id == "dharden@gsu.edu", course_key == 10401)
DH_login <- DH_login %>%
  mutate(week_logins_2020 = unlist(course_instructor_2020 %>%
                                    filter(semester == "Spring",
                                           user_id == "dharden@gsu.edu",
                                           course_key == 21308) %>%
                                    select(week_logins)))
DH_login %>%
  ggplot(aes(x = week_of_year)) +
  geom_line(aes(y = week_logins, col = "Spring 2019")) +
  geom_line(aes(y = week_logins_2020, col = "Spring 2020")) +
  scale_y_continuous(breaks = c(0, 10, 20, 30, 40), limits = c(0, 40)) +
  xlim(1, 21) +
  geom_vline(xintercept = 14, linetype = "dashed") +
  theme_bw() +
  labs(title = "Prof. H",
       color = "Year",
       y = "Logins",
       x = "Week of Year")
```



```
DH_login_Fall <- instructor_action_f19 %>%
  filter(user_id == "dharden@gsu.edu", course_key == 93383)
DH_login_Fall <- DH_login_Fall %>%
  mutate(week_logins_2020 = unlist(course_instructor_2020 %>%
                                     filter(semester == "Fall",
                                            user_id == "dharden@gsu.edu",
                                            course_key == 90984) %>%
                                     select(week_logins)))
DH_login_Fall %>%
  ggplot(aes(x = week_of_year)) +
  geom_line(aes(y = week_logins, col = "Fall 2019")) +
  geom_line(aes(y = week_logins_2020, col = "Fall 2020")) +
  scale_y_continuous(breaks = c(0, 10, 20, 30, 40), limits = c(0, 40)) +
  xlim(34, 52) +
  theme_bw() +
  labs(title = "Prof. H",
       color = "Year",
       y = "Logins",
       x = "Week of Year")
```



The plots is shown in the report as Figure 2 and Figure 3.

Prof. H's login patterns do look different after week 14 in Spring 2020, and the login patterns in Fall 2020 have the same pattern as Fall 2019, but smaller in magnitude. The login in Fall 2020 does not include logins in the new exam platform.

```
## T TEST
# Fall
t.test(DH_login_Fall$week_logins[35:53], DH_login_Fall$week_logins_2020[35:53])
```

```
##
##  Welch Two Sample t-test
##
## data:  DH_login_Fall$week_logins[35:53] and DH_login_Fall$week_logins_2020[35:53]
## t = 3.5386, df = 34.342, p-value = 0.001177
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.631376 13.421255
## sample estimates:
## mean of x mean of y
##  21.94737  13.42105
```

```
# Spring after week 14
t.test(DH_login$week_logins[14:21], DH_login$week_logins_2020[14:21])
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  DH_login$week_logins[14:21] and DH_login$week_logins_2020[14:21]
## t = -2.003, df = 7.6825, p-value = 0.08162
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.4178386   0.9178386
## sample estimates:
## mean of x mean of y
##      7.75     13.50
```

```
t.test(BP_login$week_logins[14:21], BP_login$week_logins_2020[14:21])
```

```
##
##  Welch Two Sample t-test
##
## data:  BP_login$week_logins[14:21] and BP_login$week_logins_2020[14:21]
## t = 1.9787, df = 13.188, p-value = 0.06912
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3721941  8.6221941
## sample estimates:
## mean of x mean of y
##     8.375     4.250
```

From the t-test we can see that the difference in Spring 2020 after week 14 is not statistically significant at 5% level for either of the instructors.

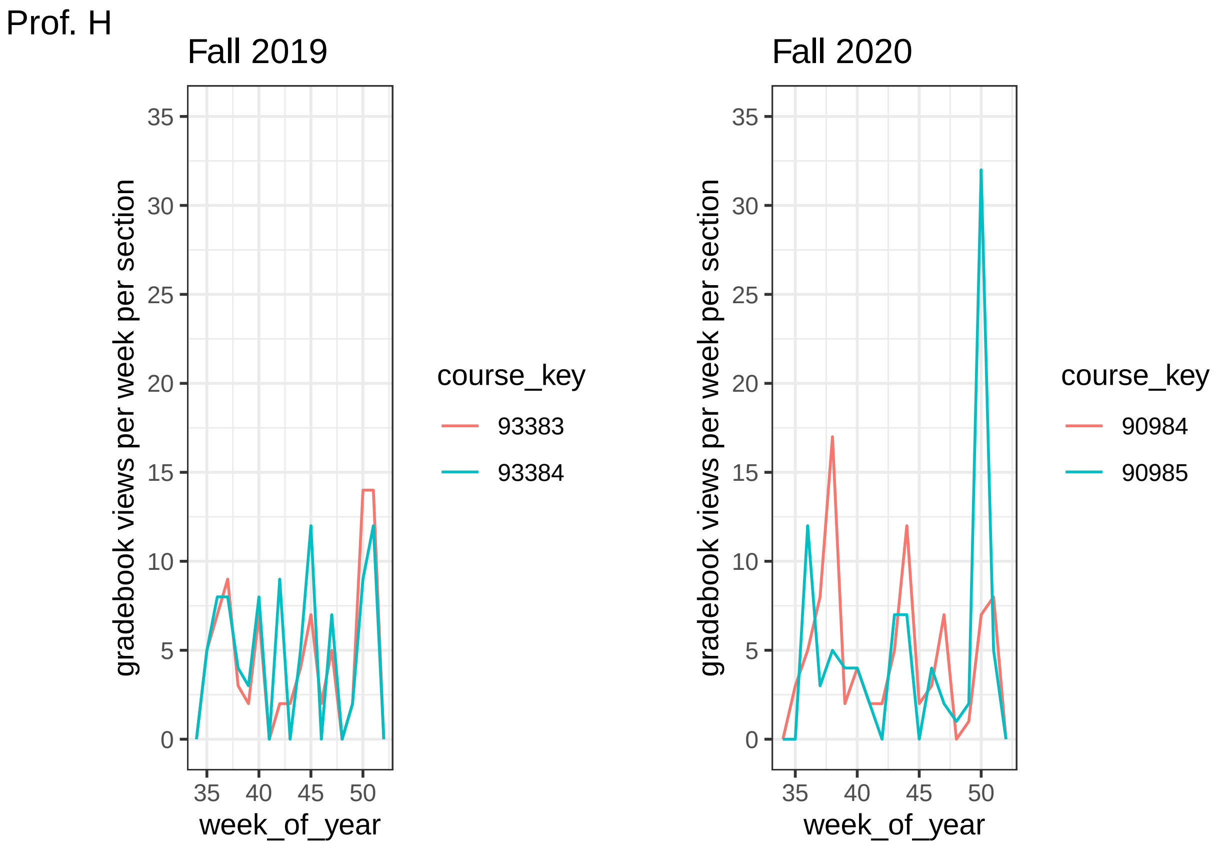
The login difference in Fall 2020 is statistically significant for Prof. H at 5% level.

Instructor Activities - Gradebook Views

```
# Fall
plot_DH.f19_gbView <- instructor_action_f19 %>%
  filter(user_id == "dharden@gsu.edu", (course_key == 93383| course_key == 93384)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25, 30, 35), limits = c(0, 35)) +
  xlim(34, 52) +
  labs(title = "Fall 2019",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "Prof. H")


plot_DH.f20_gbView <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu", semester == "Fall",
         (course_key == 90984 |course_key == 90985)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25, 30, 35), limits = c(0, 35)) +
  xlim(34, 52) +
  labs(title = "Fall 2020",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "       ")
require(gridExtra)
```

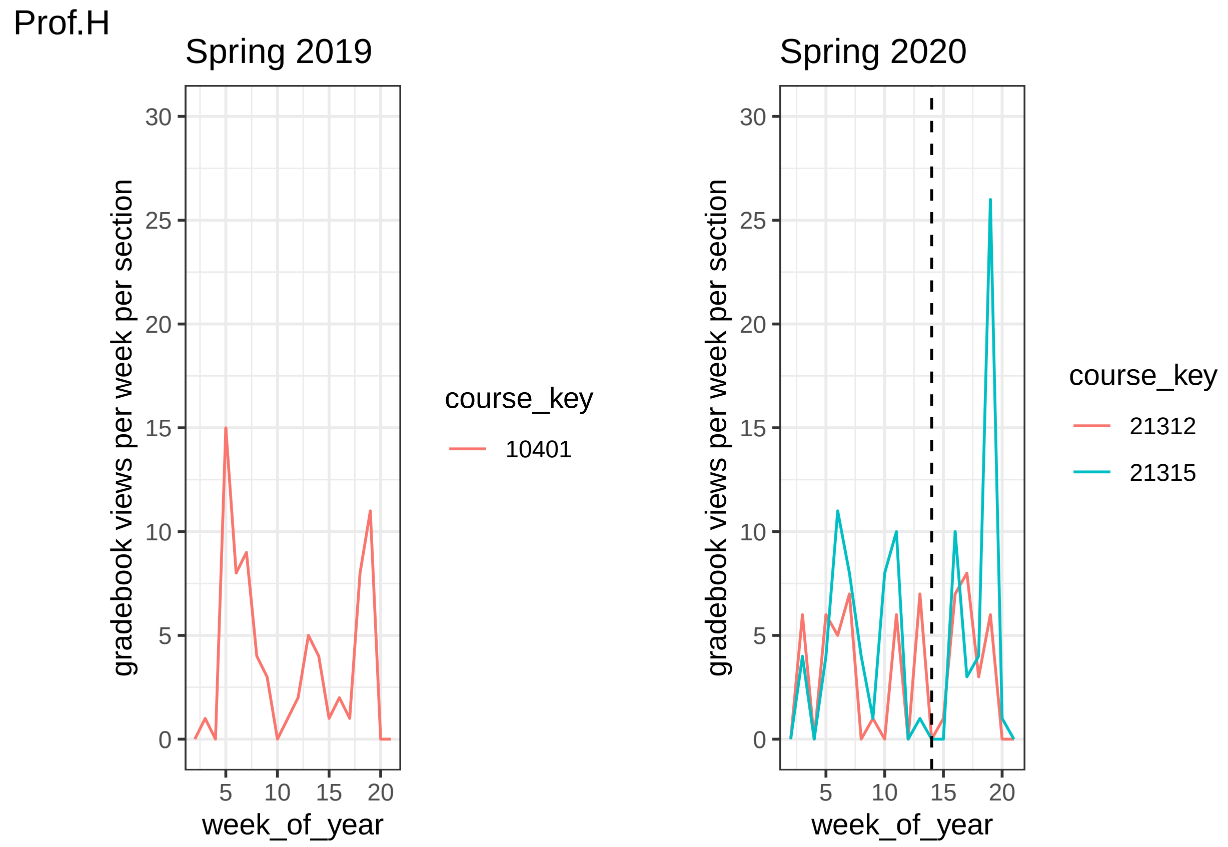
```
## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(plot_DH.f19_gbView, plot_DH.f20_gbView, ncol=2)
```



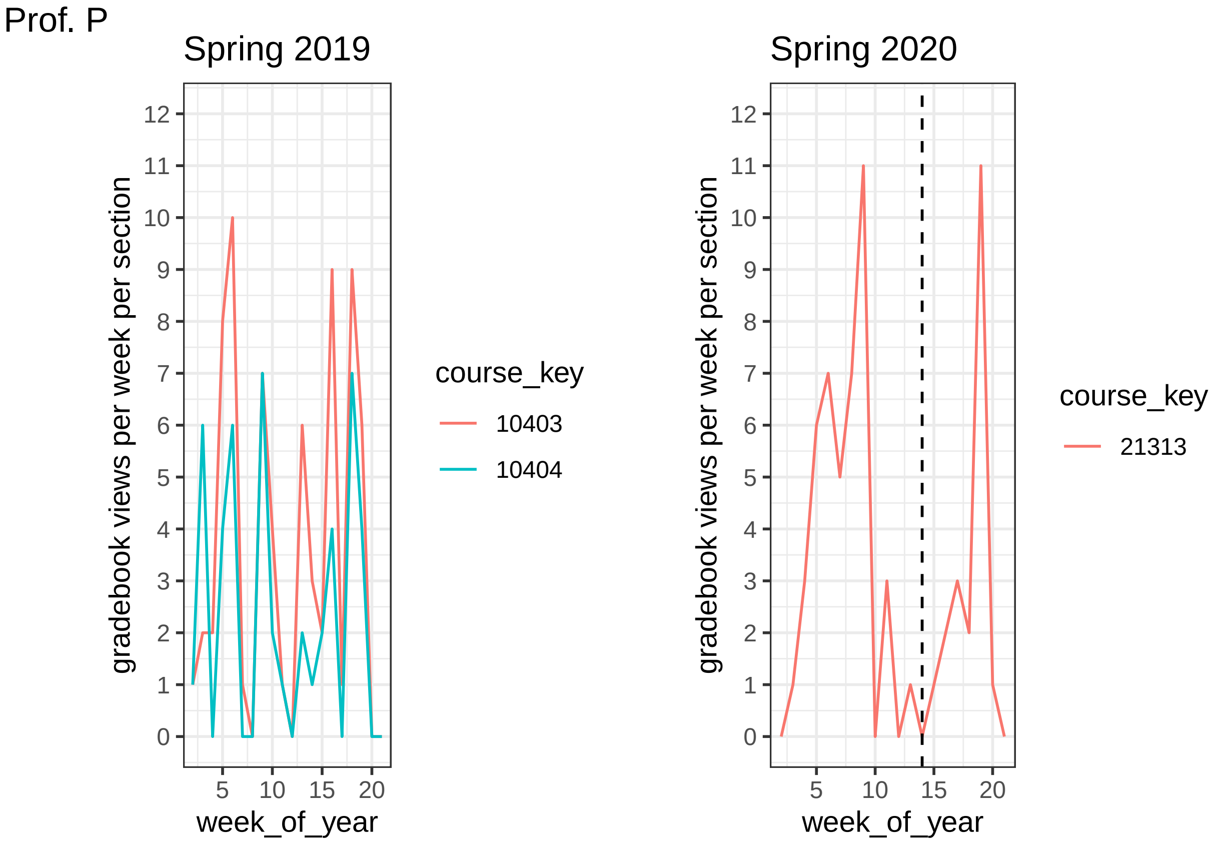
The side by side plot shows the gradebook views by Prof. H in Fall 2019 and Fall 2020. It looks like tere is a spike for one of the sections at the end of Fall 2020.

```
# Spring
plot_DH.s19_gbView <- instructor_action_s19_new %>%
  filter(user_id == "dharden@gsu.edu", course_key == 10401) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25, 30), limits = c(0, 30)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "Prof.H")


plot_DH.s20_gbView <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Spring",
         (course_key == 21312 | course_key == 21315)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25, 30), limits = c(0, 30)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "       ")
grid.arrange(plot_DH.s19_gbView, plot_DH.s20_gbView, ncol=2)
```



Similarly for the Spring semesters, there is a spike at the end of Spring 2020 that is much larger in magnitude than the increase at the end of Spring 2019.

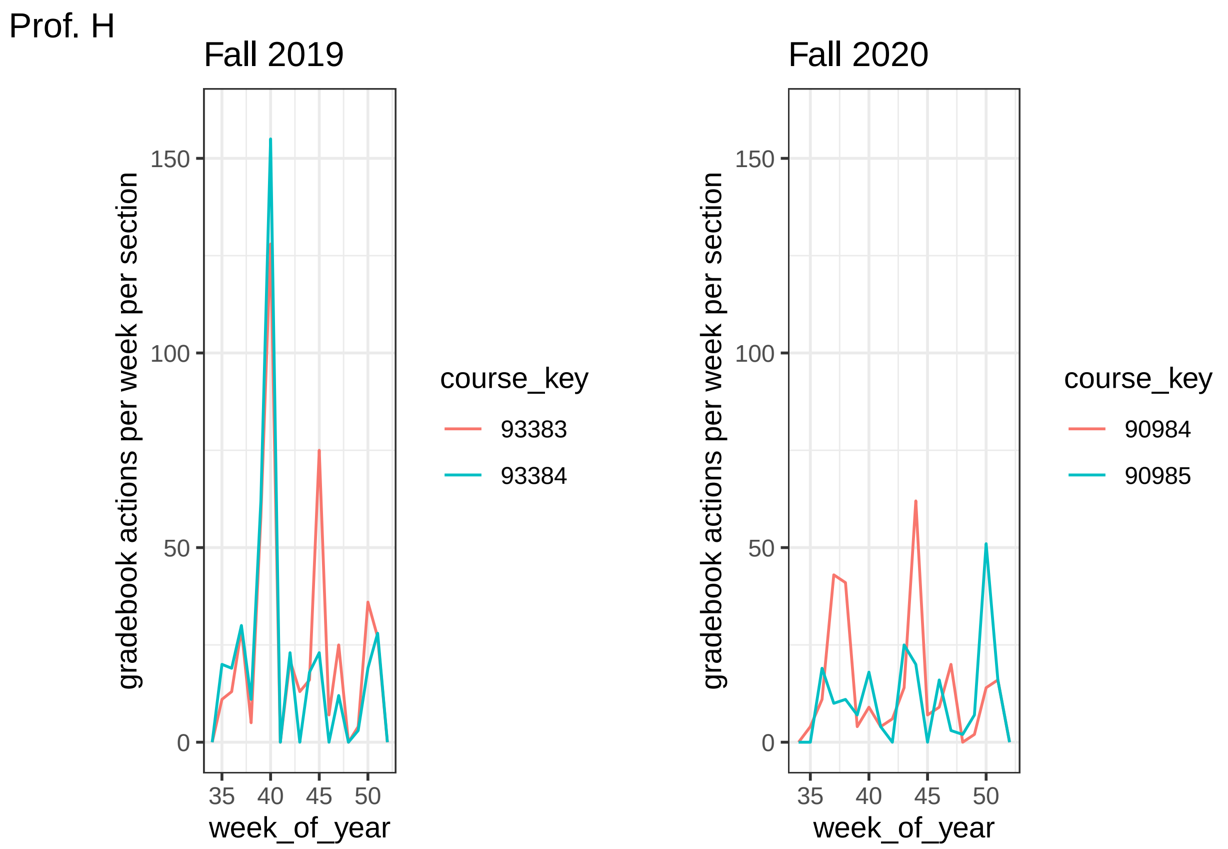
```
# BP
plot_BP.s19_gbView <- instructor_action_s19_new %>%
  filter(user_id == "BPidgeon7033",
         (course_key == 10403 | course_key == 10404)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0:12), limits = c(0, 12)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "Prof. P")

plot_BP.s20_gbView <- course_instructor_2020 %>%
  filter(user_id == "BPidgeon7033" & semester == "Spring", course_key == 21313) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0:12), limits = c(0, 12)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "gradebook views per week per section",
       color = "course_key",
       tag = "       ")
grid.arrange(plot_BP.s19_gbView, plot_BP.s20_gbView, ncol=2)
```



The patterns for gradebook actions for Prof. P in the spring semesters are very similar.

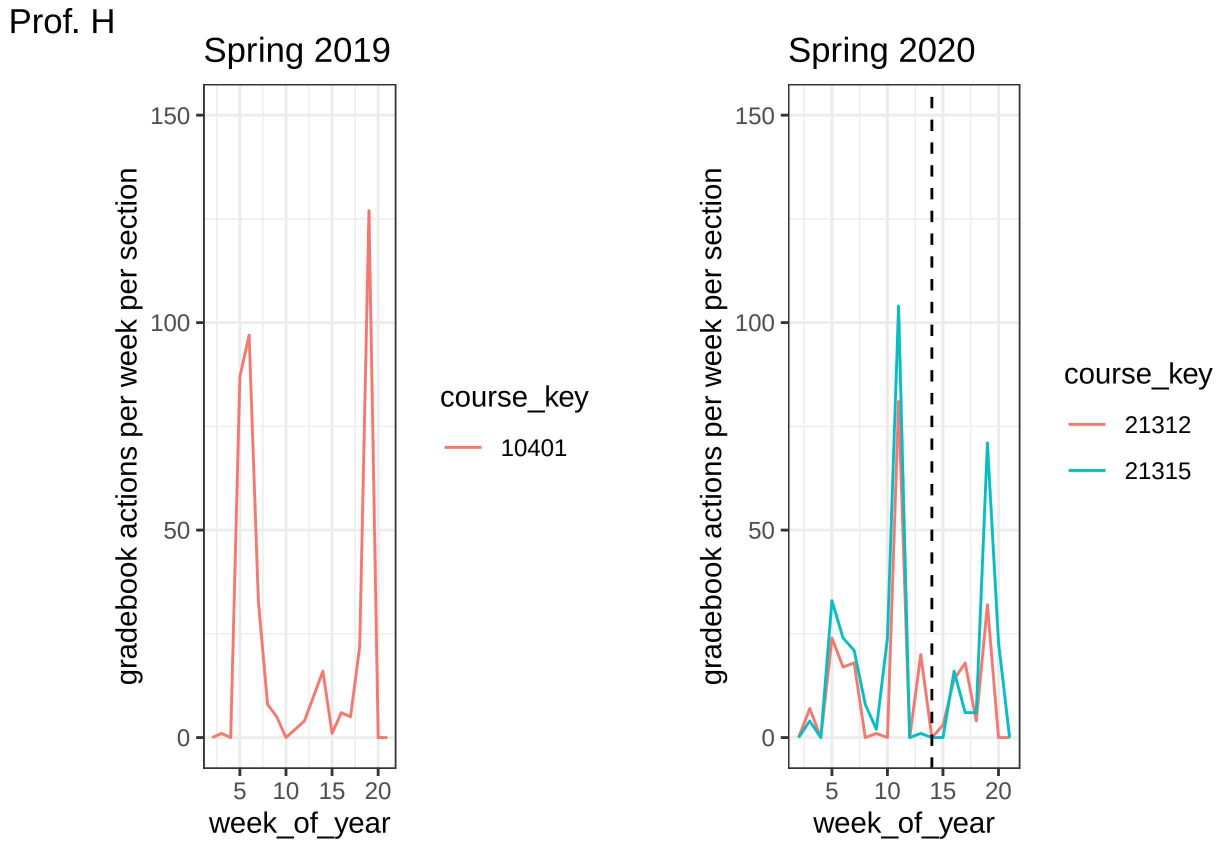
Instructor Activities - Gradebook Actions

```r
plot_DH.f19_gbAction <- instructor_action_f19 %>%
  filter(user_id == "dharden@gsu.edu", (course_key == 93383|course_key == 93384)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0,50,100,150), limits = c(0, 160)) +
  xlim(34, 52) +
  labs(title = "Fall 2019",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "Prof. H")

plot_DH.f20_gbAction <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Fall",
         (course_key == 90984| course_key == 90985)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0,50,100,150), limits = c(0, 160)) +
  xlim(34, 52) +
  labs(title = "Fall 2020",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "       ")
grid.arrange(plot_DH.f19_gbAction, plot_DH.f20_gbAction, ncol=2)
```



The side by side plot shows the gradebook actions done by Prof. H. The general patterns are the same for Fall 2019 and Fall 2020, with 1 spike at the beginning of Fall 2019, and the magnitude of this increase in Fall 2020 is much smaller.

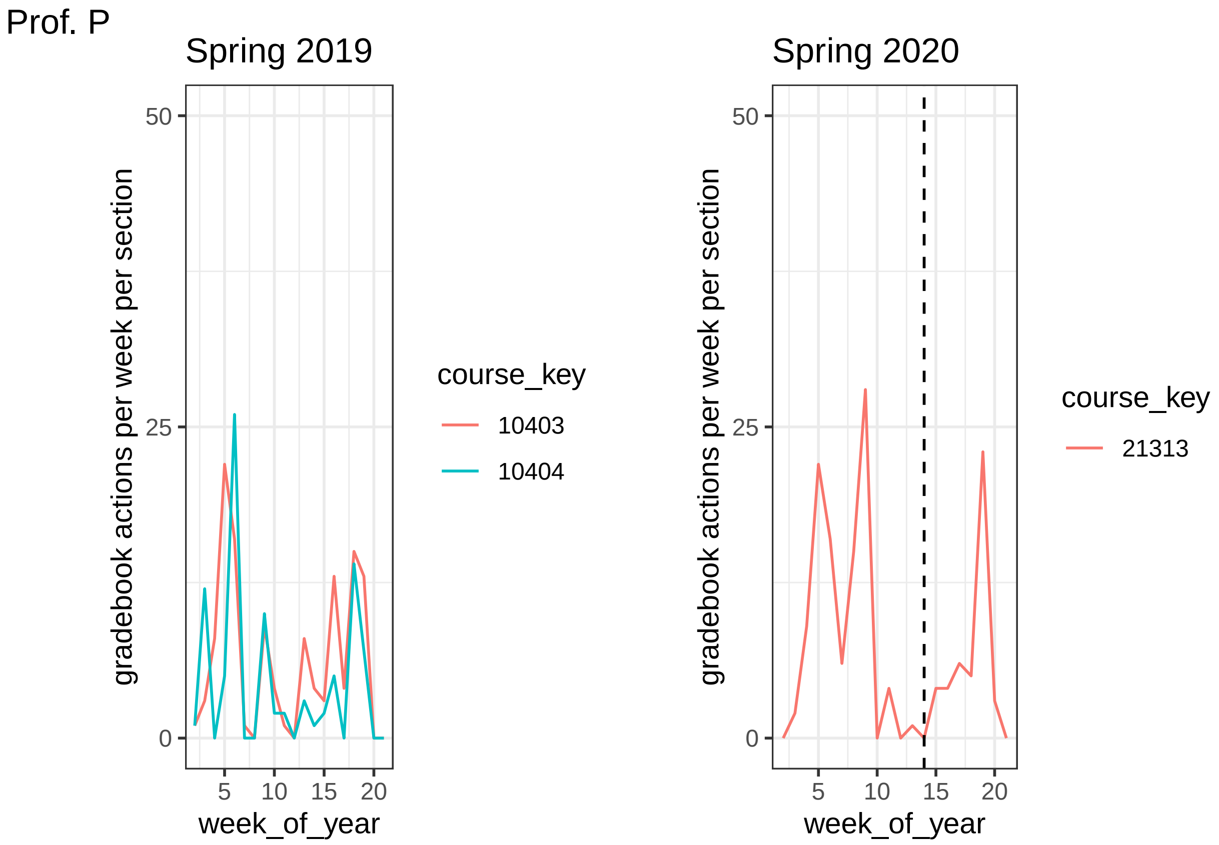
```
plot_DH.s19_gbAction <- instructor_action_s19_new %>%
  filter(user_id == "dharden@gsu.edu", course_key == 10401) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  scale_y_continuous(breaks = c(0, 50, 100, 150), limits = c(0, 150)) +
  xlim(2, 21) +
  theme_bw() +
  labs(title = "Spring 2019",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "Prof. H")

plot_DH.s20_gbAction <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Spring",
         (course_key == 21312 | course_key == 21315)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0,50,100,150), limits = c(0, 150)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "        ")
grid.arrange(plot_DH.s19_gbAction, plot_DH.s20_gbAction, ncol=2)
```



Similarly for the spring semesters, the spike at the end of Spring 2019 is larger than the spike at the end of Spring 2020.

There is also a spike in the middle of Spring 2020, which does not occur in Spring 2019.

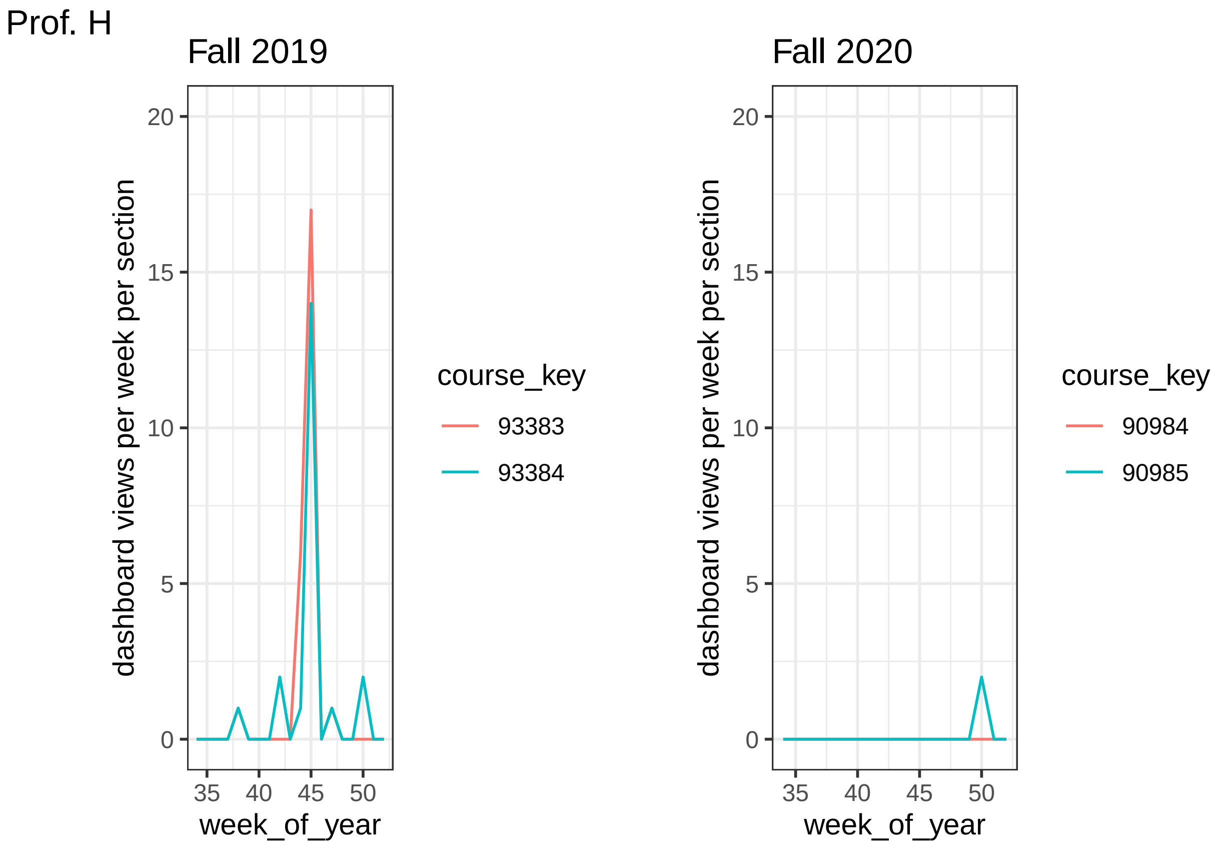
```
# BP
plot_BP.s19_gbAction <- instructor_action_s19_new %>%
  filter(user_id == "BPidgeon7033",
         (course_key == 10403 | course_key == 10404)) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 25, 50), limits = c(0, 50)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "Prof. P")

plot_BP.s20_gbAction <- course_instructor_2020 %>%
  filter(user_id == "BPidgeon7033" & semester == "Spring", course_key == 21313) %>%
  ggplot(aes(x = week_of_year,  y = gradebook_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0,25, 50), limits = c(0, 50)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "gradebook actions per week per section",
       color = "course_key",
       tag = "        ")
grid.arrange(plot_BP.s19_gbAction, plot_BP.s20_gbAction, ncol=2)
```



The patterns for gradebook actions for Prof. P in the spring semesters are very similar.

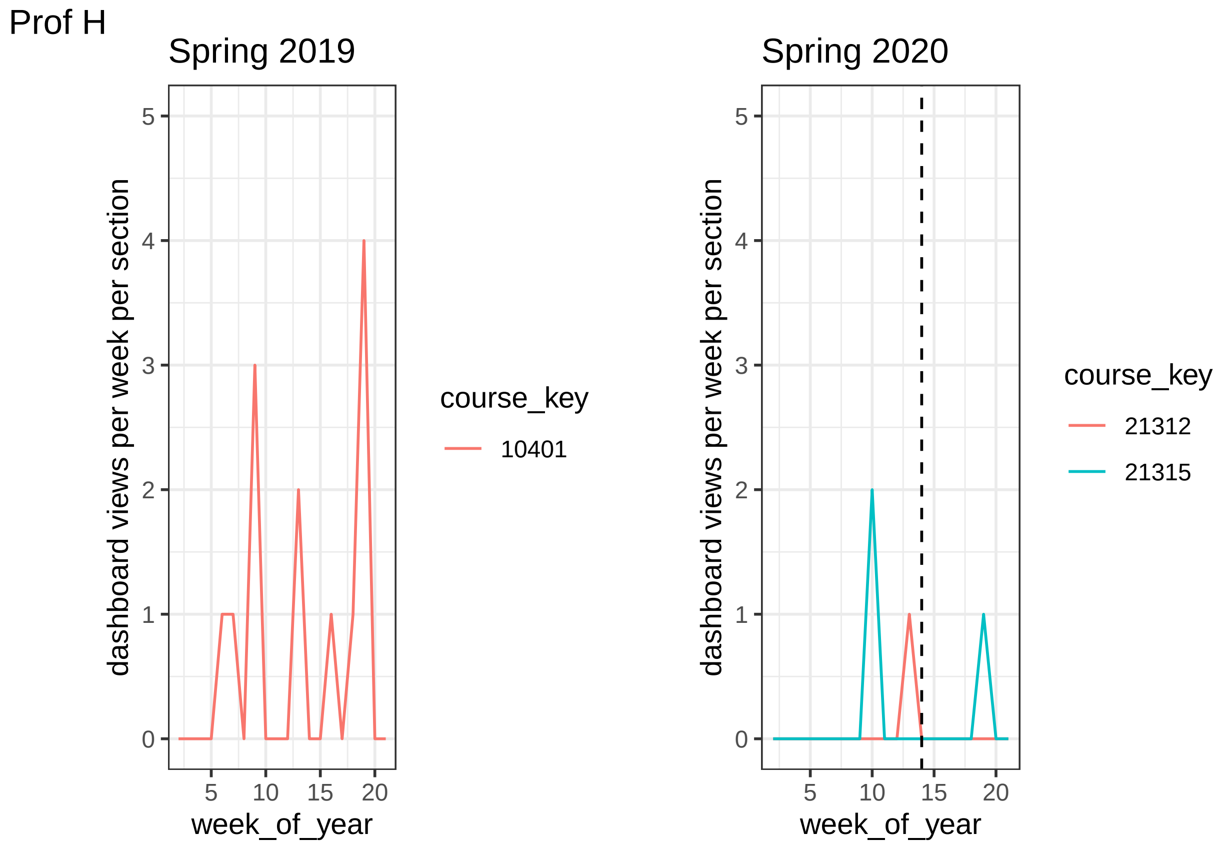
Instructor Activities - Dashboard Views

```r
plot_DH.f19_dbView <- instructor_action_f19 %>%
  filter(user_id == "dharden@gsu.edu",
         (course_key == 93383| course_key == 93384)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20), limits = c(0, 20)) +
  xlim(34, 52) +
  labs(title = "Fall 2019",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "Prof. H")


plot_DH.f20_dbView <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu", semester == "Fall",
         (course_key == 90984| course_key == 90985)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20), limits = c(0, 20)) +
  xlim(34, 52) +
  labs(title = "Fall 2020",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "       ")
grid.arrange(plot_DH.f19_dbView, plot_DH.f20_dbView, ncol=2)
```



The patterns for dashboard views in Fall 2020 is not the same as the patterns in Fall 2019 - there is no spike in the middle of Fall 2020. There is also 1 section in Fall 2020 that does not have any dashboard views recorded.

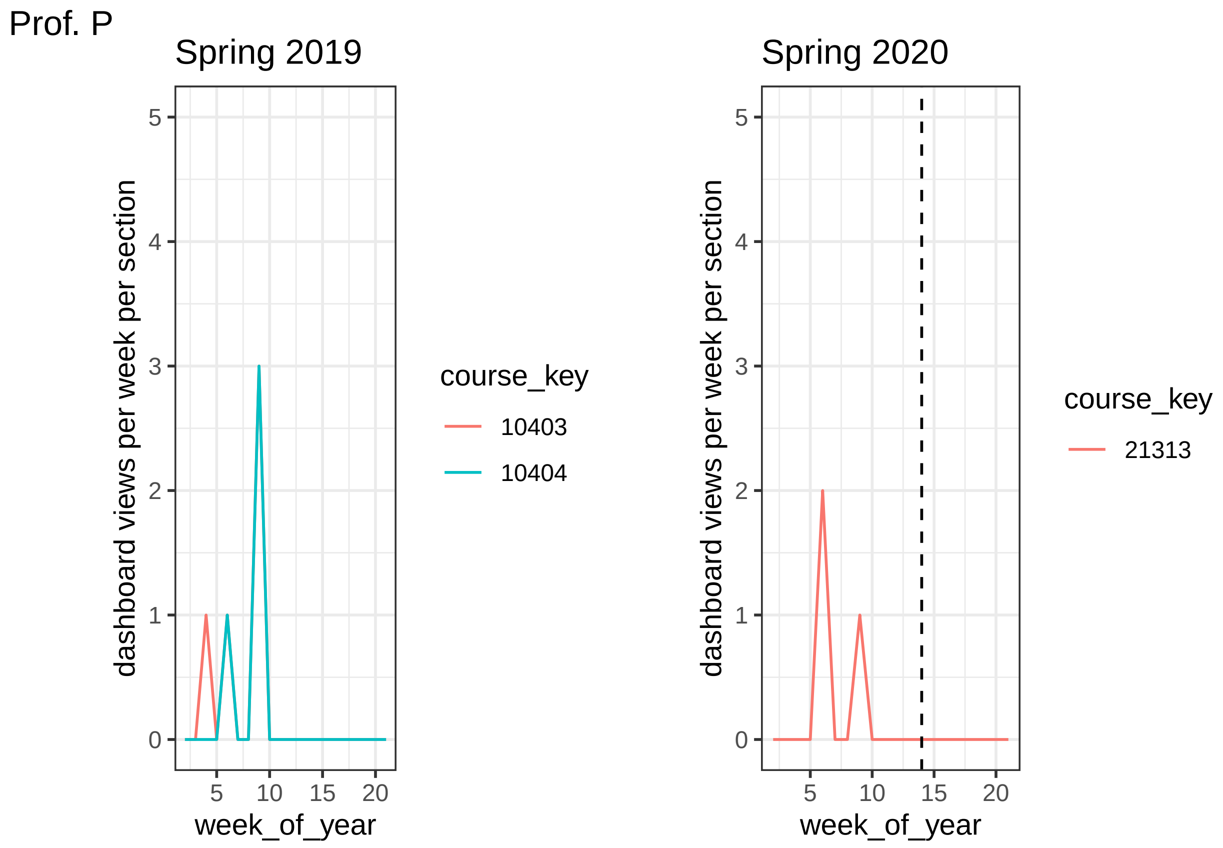
```
# Spring
plot_DH.s19_dbView <- instructor_action_s19_new %>%
  filter(user_id == "dharden@gsu.edu", course_key == 10401) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0:5), limits = c(0, 5)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "Prof H")


plot_DH.s20_dbView <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Spring",
         (course_key == 21312 | course_key == 21315)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0:5), limits = c(0, 5)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "         ")
grid.arrange(plot_DH.s19_dbView, plot_DH.s20_dbView, ncol=2)
```



For the patterns of dashboard views in the spring semesters, there are much fewer records of dashboad views in Spring 2020 compared to Spring 2019.

```
# BP
plot_BP.s19_dbView <- instructor_action_s19_new %>%
  filter(user_id == "BPidgeon7033",
         (course_key == 10403 | course_key == 10404)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0:5), limits = c(0, 5)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "Prof. P")
```

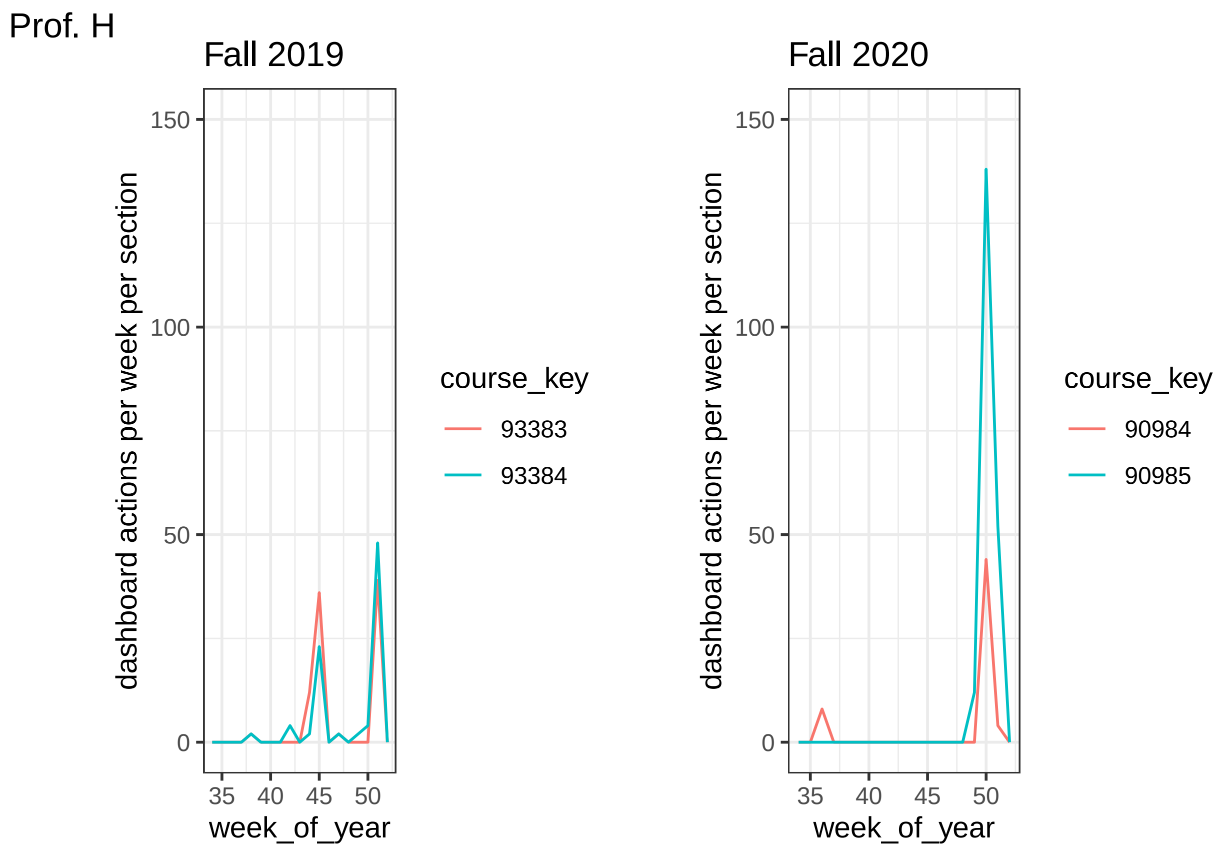
```
plot_BP.s20_dbView <- course_instructor_2020 %>%
  filter(user_id == "BPidgeon7033" & semester == "Spring",
         course_key == 21313) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_views,
              group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0:5), limits = c(0, 5)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "dashboard views per week per section",
       color = "course_key",
       tag = "       ")

grid.arrange(plot_BP.s19_dbView, plot_BP.s20_dbView, ncol=2)
```



The dashboard view patterns for Prof. P does not change much from Spring 2019 to Spring 2020.

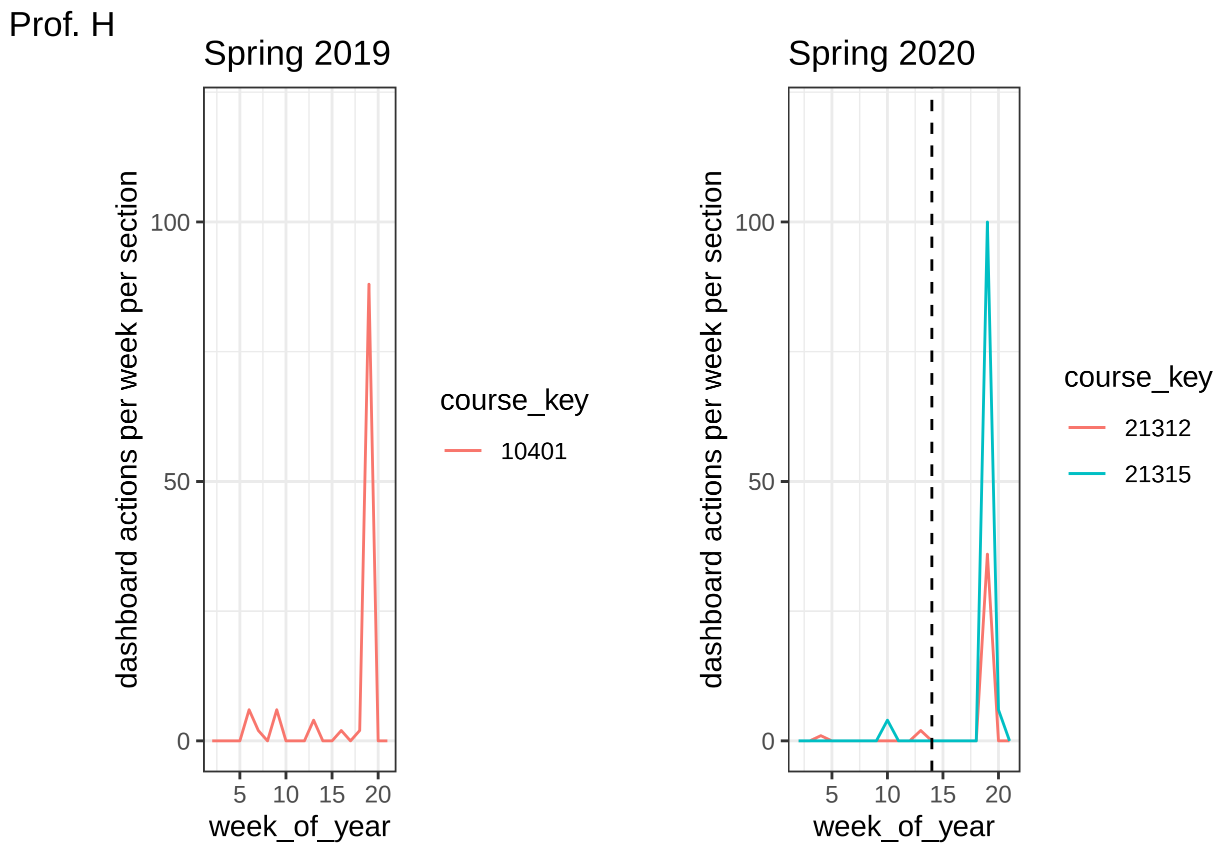
Instructor Activities - Dashboard Actions

```
plot_DH.f19_dbAction <- instructor_action_f19 %>%
  filter(user_id == "dharden@gsu.edu",
         (course_key == 93383| course_key == 93384)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0,50,100,150), limits = c(0, 150)) +
  xlim(34, 52) +
  labs(title = "Fall 2019",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "Prof. H")

plot_DH.f20_dbAction <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Fall",
         (course_key == 90984| course_key == 90985)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0,50,100,150), limits = c(0, 150)) +
  xlim(34, 52) +
  labs(title = "Fall 2020",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "       ")

grid.arrange(plot_DH.f19_dbAction, plot_DH.f20_dbAction, ncol=2)
```



The dashboard action pattern in Fall 2020 does not have an increase in the middle of the semester, similar to what happened in Fall 2019. They both have an increase at the end of the semester, with 1 section having a much larger number of actions in Fall 2020.

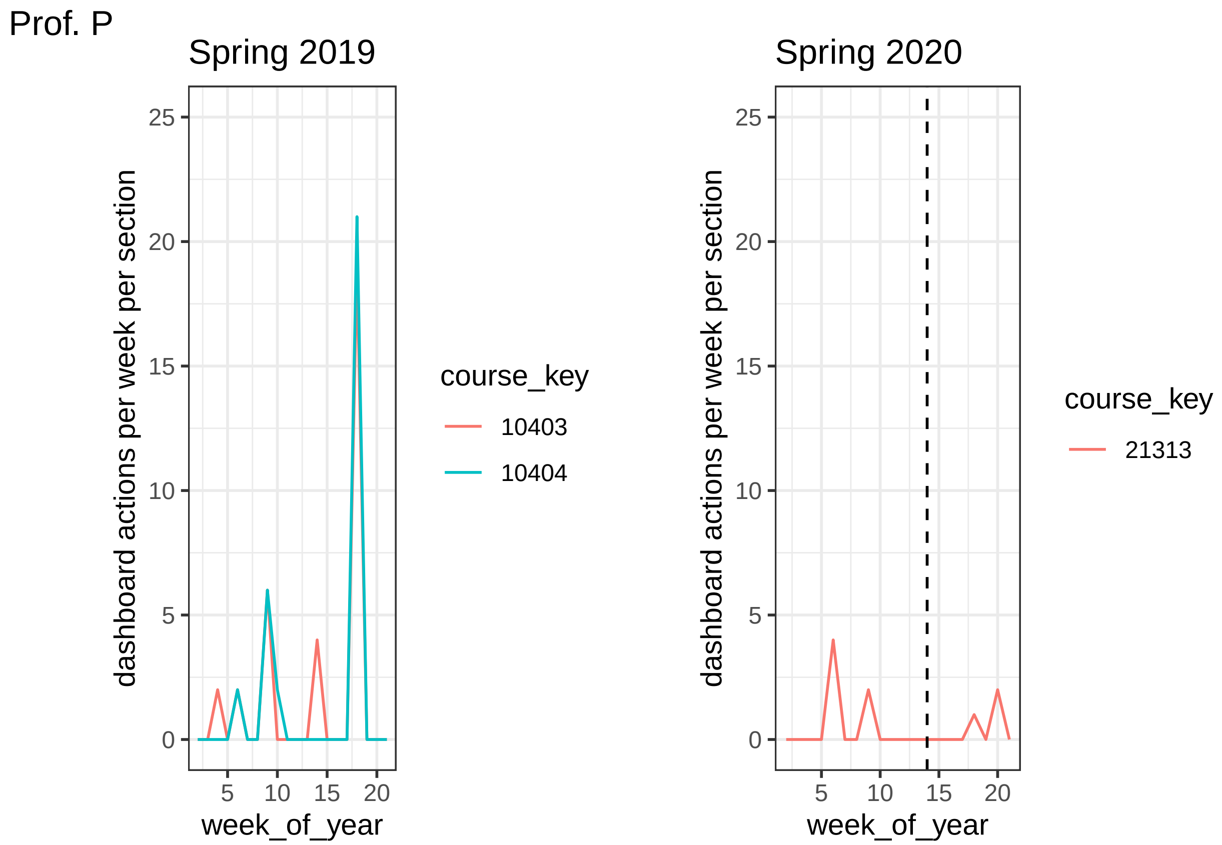
```
# Spring
plot_DH.s19_dbAction <- instructor_action_s19_new %>%
  filter(user_id == "dharden@gsu.edu", course_key == 10401) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 50, 100), limits = c(0, 120)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "Prof. H")

plot_DH.s20_dbAction <- course_instructor_2020 %>%
  filter(user_id == "dharden@gsu.edu" & semester == "Spring",
         (course_key == 21312 | course_key == 21315)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0,50,100), limits = c(0, 120)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "        ")
grid.arrange(plot_DH.s19_dbAction, plot_DH.s20_dbAction, ncol=2)
```



The patterns for dashboard actions in Spring 2019 and Spring 2020 are very similar. There is 1 section in Fall 2020 that has a smaller dashboard actions at the end of Spring 2020, comparing to the other Fall 2020 section taught by Prof. H.

```
# BP
plot_BP.s19_dbAction <- instructor_action_s19_new %>%
  filter(user_id == "BPidgeon7033",
         (course_key == 10403 | course_key == 10404)) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25), limits = c(0, 25)) +
  xlim(2, 21) +
  labs(title = "Spring 2019",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "Prof. P")
```

```r
plot_BP.s20_dbAction <- course_instructor_2020 %>%
  filter(user_id == "BPidgeon7033" & semester == "Spring",
         course_key == 21313) %>%
  ggplot(aes(x = week_of_year,  y = dashboard_actions,
             group = as.factor(course_key), color = as.factor(course_key))) +
  geom_line() +
  theme_bw() +
  geom_vline(xintercept = 14, linetype = "dashed") +
  scale_y_continuous(breaks = c(0, 5, 10, 15, 20, 25), limits = c(0, 25)) +
  xlim(2, 21) +
  labs(title = "Spring 2020",
       y = "dashboard actions per week per section",
       color = "course_key",
       tag = "       ")

grid.arrange(plot_BP.s19_dbAction, plot_BP.s20_dbAction, ncol=2)
```



The dashboard actions for Prof. P in Spring 2020 is different from the patterns in Spring 2019. The spike at the end of Spring 2020 is much smaller comparing to the spike at the end of Spring 2019.

Appendix 2

Read all the files and clean the data

```r
#7890 rows(different courses)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.6.2
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```r
question_summary_by_course_19s<-
  read.csv("csv19s/question_summary_by_course.csv",sep = ";")
question_summary_by_course_19sa <-
  question_summary_by_course_19s[question_summary_by_course_19s$purpose!="none",]






#8560 rows
question_summary_by_course_19f<-
  read.csv("csv19f/question_summary_by_course.csv",sep = ";")
question_summary_by_course_19f$course_key <-
  as.factor(question_summary_by_course_19f$course_key)
question_summary_by_course_19fa <-
  question_summary_by_course_19f[question_summary_by_course_19f$purpose!="none",]



#21805 rows
question_summary_by_course_20a<-
  read.csv("csv20/question_summary_by_course.csv",sep = ";")
```

```
question_summary_by_course_20sa <-
  question_summary_by_course_20a[question_summary_by_course_20a$course_key
                                 %in%  c(21313,21318,21309,21321,21310,21316,21315,21312,21320,
                                         "Sp2020-069",21311,21308,21319,21322,21314,"Sp2020-063"),]
question_summary_by_course_20fa <-
  question_summary_by_course_20a[question_summary_by_course_20a$course_key
                                 %in% c(90988,90984,90372,90995,93110,90487,90989,
                                        92275,90370,92055,90368,91561,90996,90993,
                                        95340,90992,90985,90987,90990,90986,93062,
                                        94468,90999,90997,90367,90366,93112),]


sa19<-question_summary_by_course_19sa %>%
  group_by(course_key,purpose) %>%
  summarise(total = n(),
            correct_total = sum(correct),
            error_total = sum(errors),
            avg_correct = correct_total/(correct_total+error_total))
```

```
## `summarise()` regrouping output by 'course_key' (override with `.groups` argument)
```

```
fa19<-question_summary_by_course_19fa %>%
  group_by(course_key,purpose) %>%
  summarise(total = n(),
            correct_total = sum(correct),
            error_total = sum(errors),
            avg_correct = correct_total/(correct_total+error_total))
```

```
## `summarise()` regrouping output by 'course_key' (override with `.groups` argument)
```

```
sa20<-question_summary_by_course_20sa %>%
  group_by(course_key,purpose) %>%
  summarise(total = n(),
            correct_total = sum(correct),
            error_total = sum(errors),
            avg_correct = correct_total/(correct_total+error_total))
```

```
## `summarise()` regrouping output by 'course_key' (override with `.groups` argument)
```

```
fa20<-question_summary_by_course_20fa %>%
  group_by(course_key,purpose) %>%
  summarise(total = n(),
            correct_total = sum(correct),
            error_total = sum(errors),
            avg_correct = correct_total/(correct_total+error_total))
```

```
## `summarise()` regrouping output by 'course_key' (override with `.groups` argument)
```

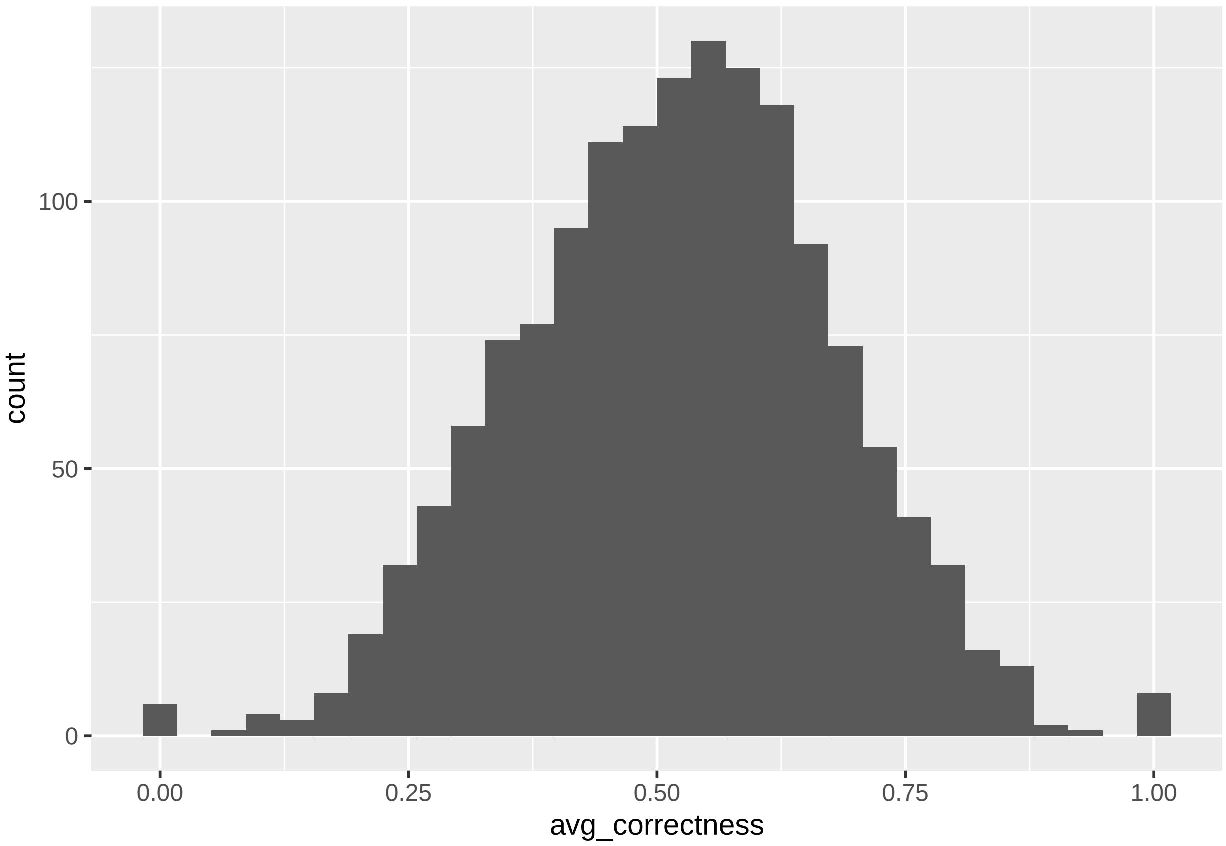
```r
sa19$semester<-"Spring2019"
fa19$semester<-"Fall2019"
sa20$semester<-"Spring2020"
fa20$semester<-"Fall2020"
question_all<- rbind(sa19,fa19,sa20,fa20)
question_all$semester<-as.factor(question_all$semester)
```

Histograms for every purpose

```r
question_summary_by_course_19sa %>%
  filter(purpose=="quiz")%>%
  ggplot(aes( x = correct/(correct+errors)))+
  geom_histogram()+
  xlab("avg_correctness")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

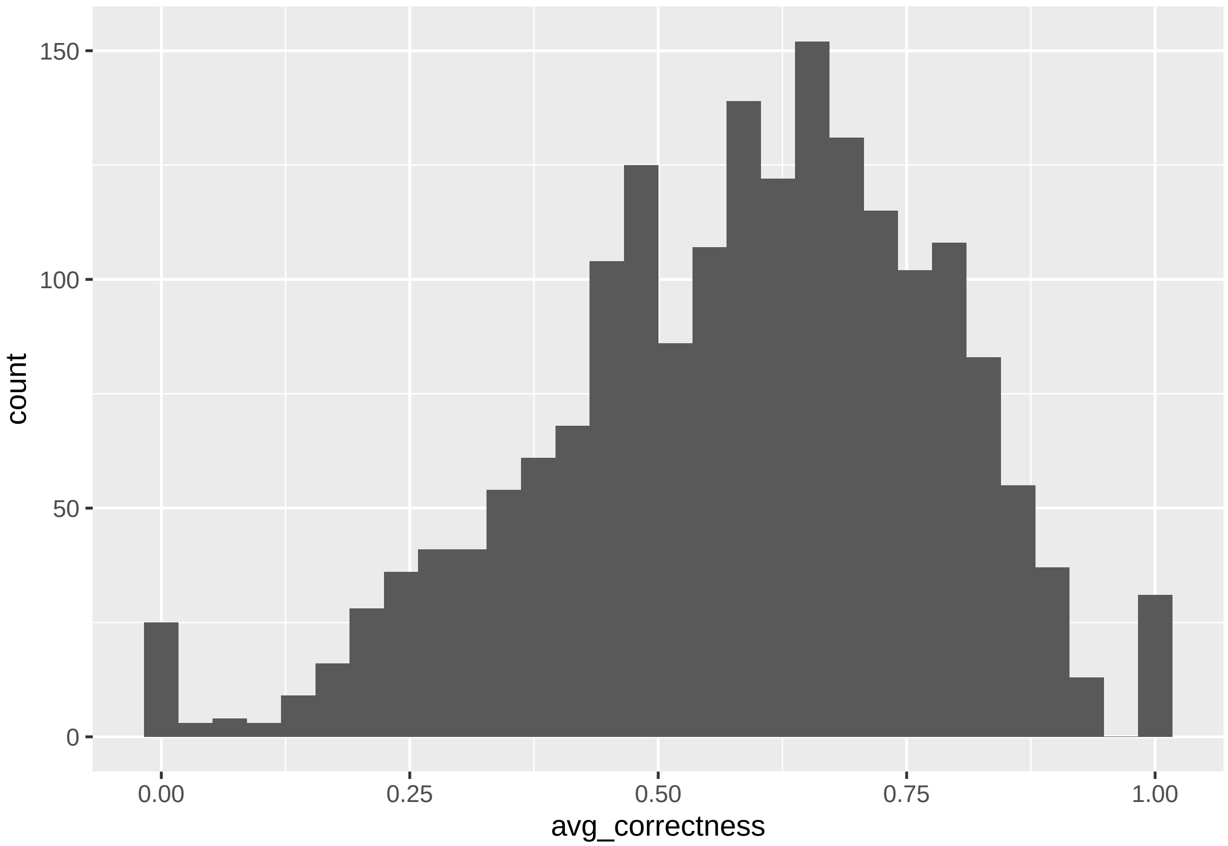
```
## Warning: Removed 105 rows containing non-finite values (stat_bin).
```



```r
question_summary_by_course_20fa %>%
  filter(purpose=="quiz")%>%
  ggplot(aes( x = correct/(correct+errors)))+
  geom_histogram()+
  xlab("avg_correctness")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

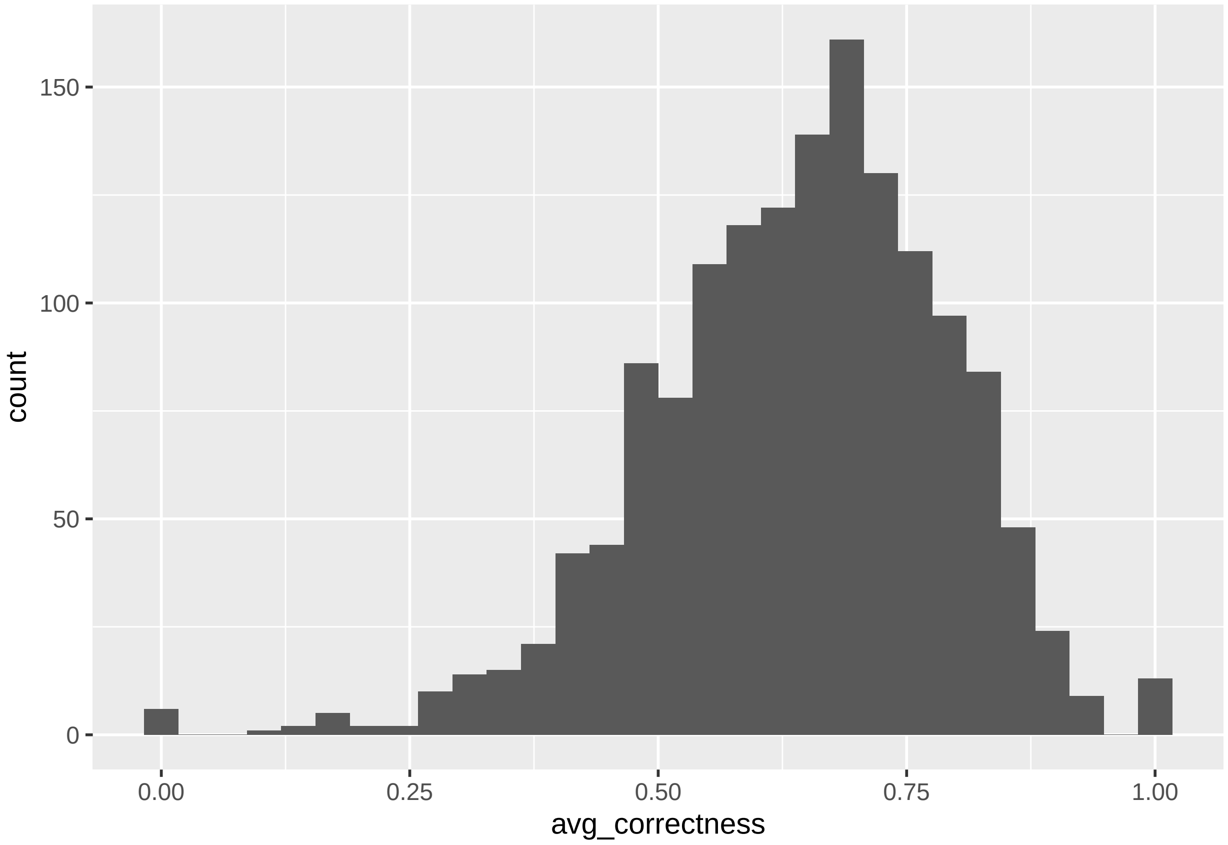
```
## Warning: Removed 158 rows containing non-finite values (stat_bin).
```



```
question_summary_by_course_20sa %>%
  filter(purpose=="quiz")%>%
  ggplot(aes( x = correct/(correct+errors)))+
  geom_histogram()+
  xlab("avg_correctness")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

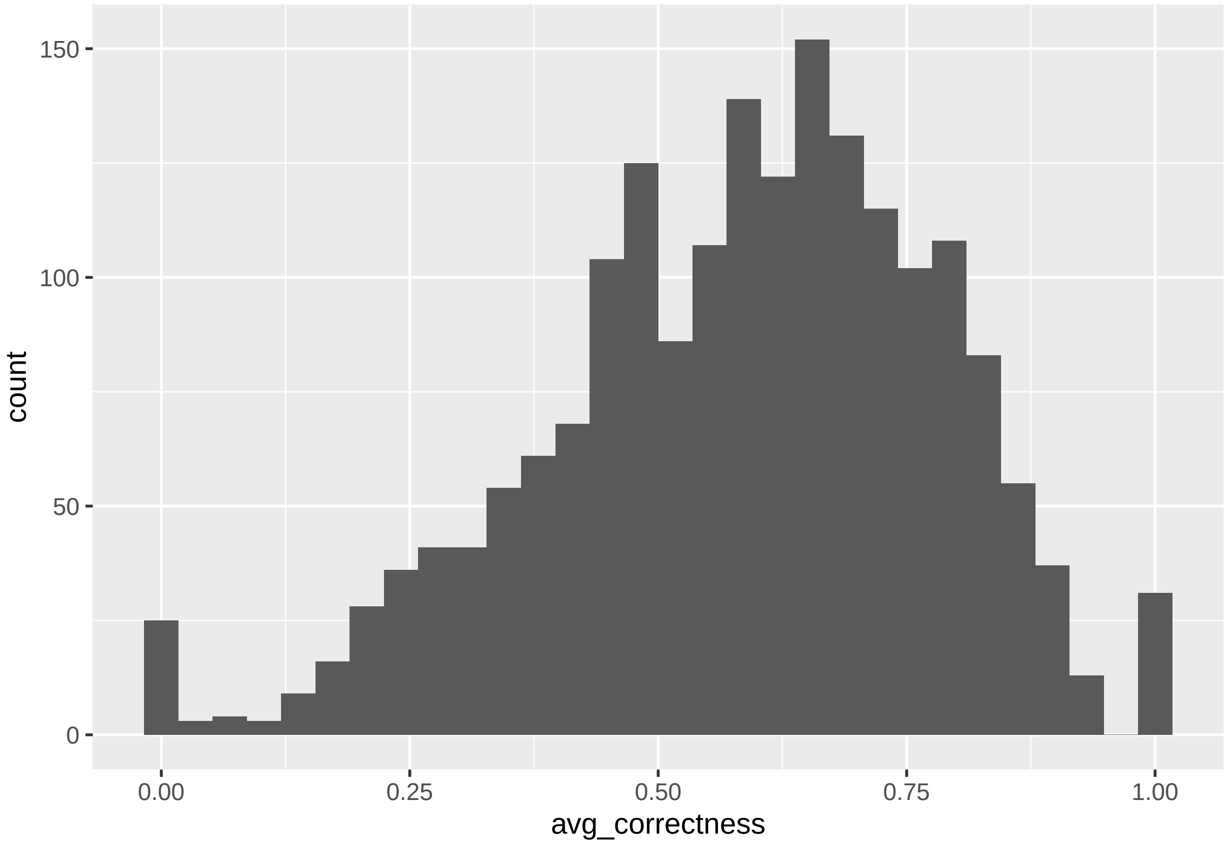
```
## Warning: Removed 110 rows containing non-finite values (stat_bin).
```



```
question_summary_by_course_20fa %>%
  filter(purpose=="quiz")%>%
  ggplot(aes( x = correct/(correct+errors)))+
  geom_histogram()+
  xlab("avg_correctness")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 158 rows containing non-finite values (stat_bin).
```

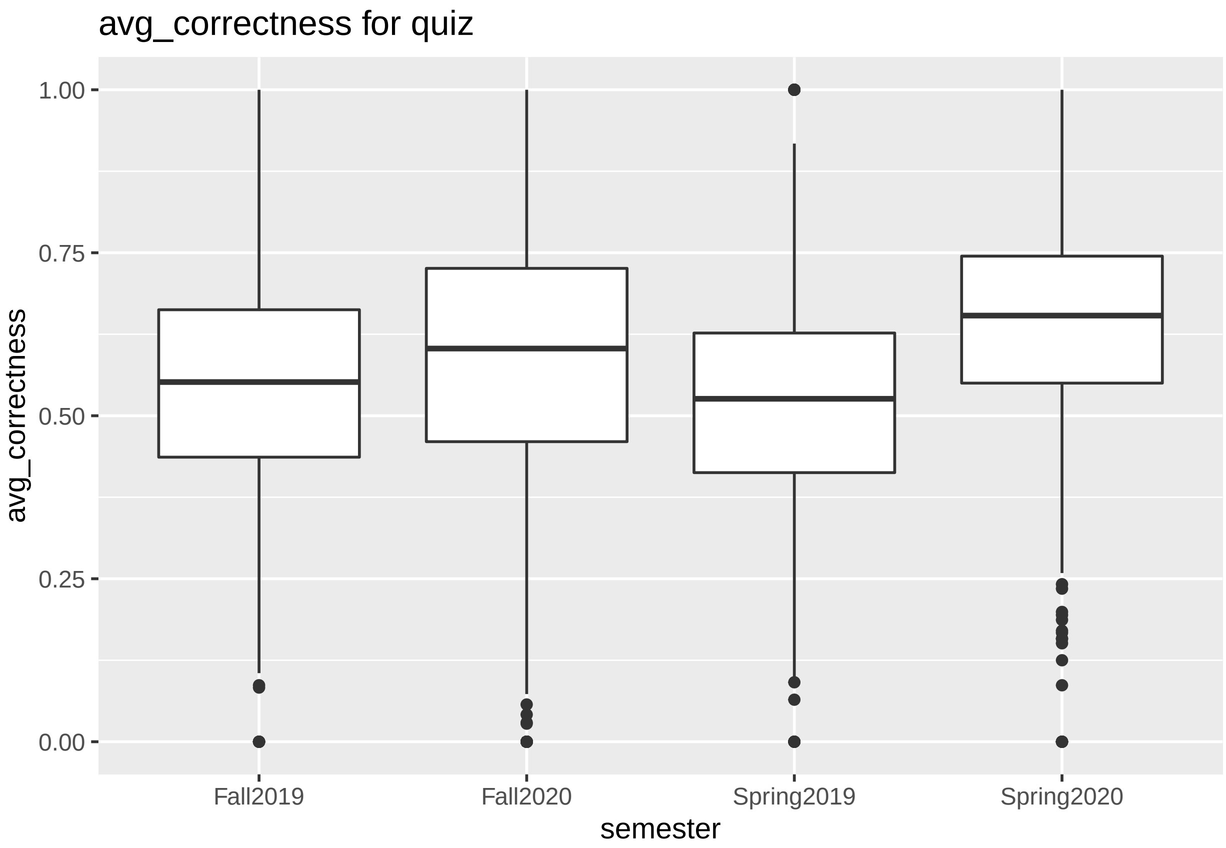


From the histograms, we can see that the distributions are very close to normal distribution which is reasonable and we can dig more using other methods

Boxplots for every purpose

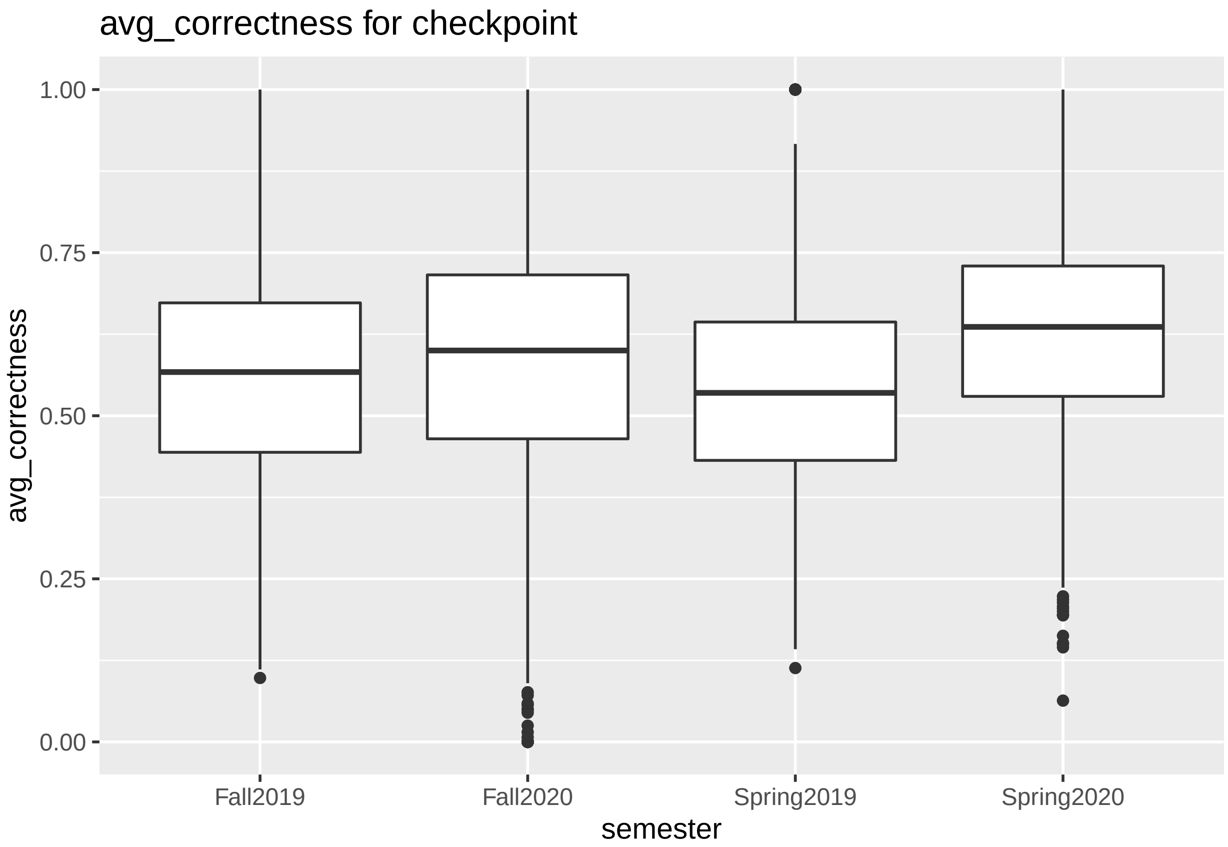
```
sa19$semester<-"Spring2019"
fa19$semester<-"Fall2019"
sa20$semester<-"Spring2020"
fa20$semester<-"Fall2020"

question_summary_by_course_19sa$semester<-"Spring2019"
question_summary_by_course_19fa$semester<-"Fall2019"
question_summary_by_course_20sa$semester<-"Spring2020"
question_summary_by_course_20fa$semester<-"Fall2020"
question_all1 <-
  rbind(question_summary_by_course_19fa,question_summary_by_course_19sa,
        question_summary_by_course_20sa,question_summary_by_course_20fa)
question_all1$semester<-
  as.factor(question_all1$semester)




par(mfrow=c(2,2))
question_all1 %>%
  filter(purpose=="quiz")%>%
  ggplot(aes( x=semester, y = correct/(correct+errors)))+
  geom_boxplot()+
  ylab("avg_correctness")+
  labs(title = "avg_correctness for quiz")
```

```
## Warning: Removed 470 rows containing non-finite values (stat_boxplot).
```



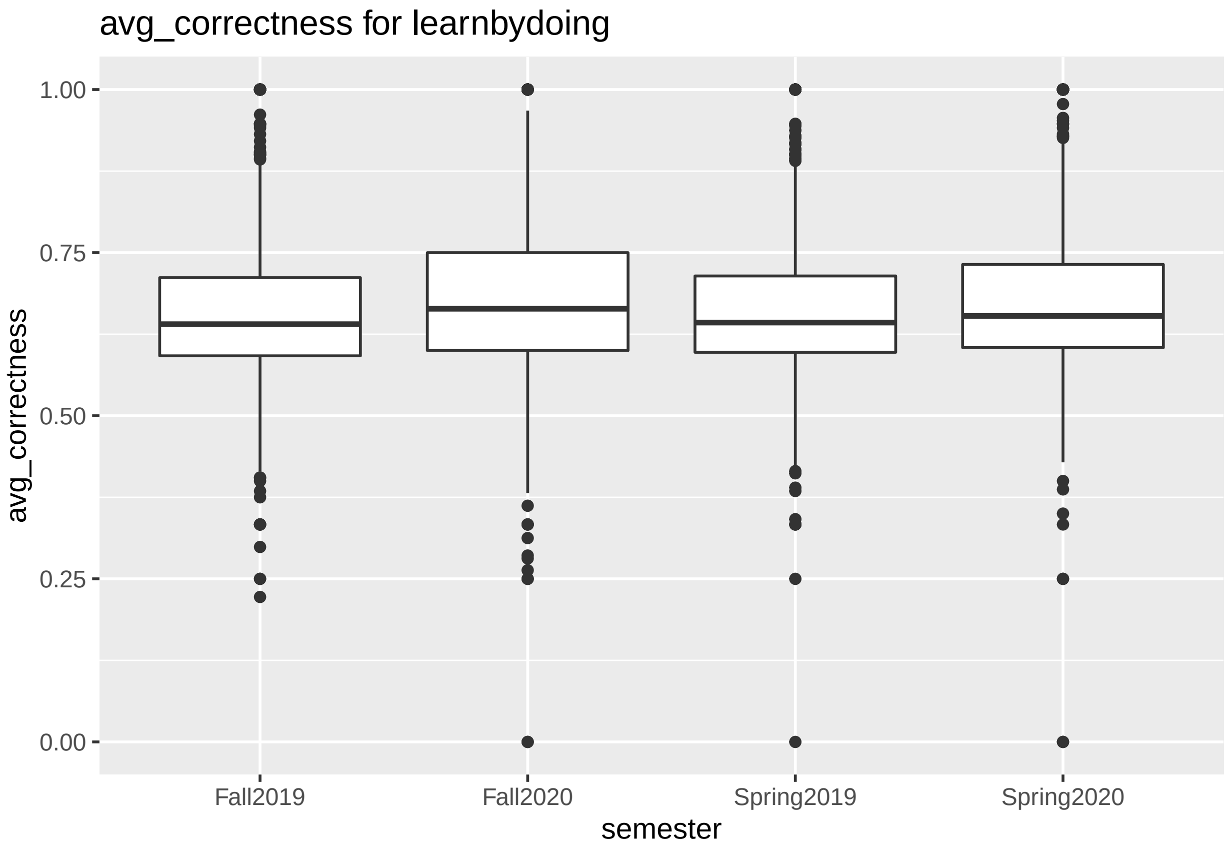
```
question_all1 %>%
  filter(purpose=="checkpoint")%>%
  ggplot(aes( x=semester, y = correct/(correct+errors)))+
  geom_boxplot()+
  ylab("avg_correctness")+
  labs(title = "avg_correctness for checkpoint")
```

```
## Warning: Removed 284 rows containing non-finite values (stat_boxplot).
```



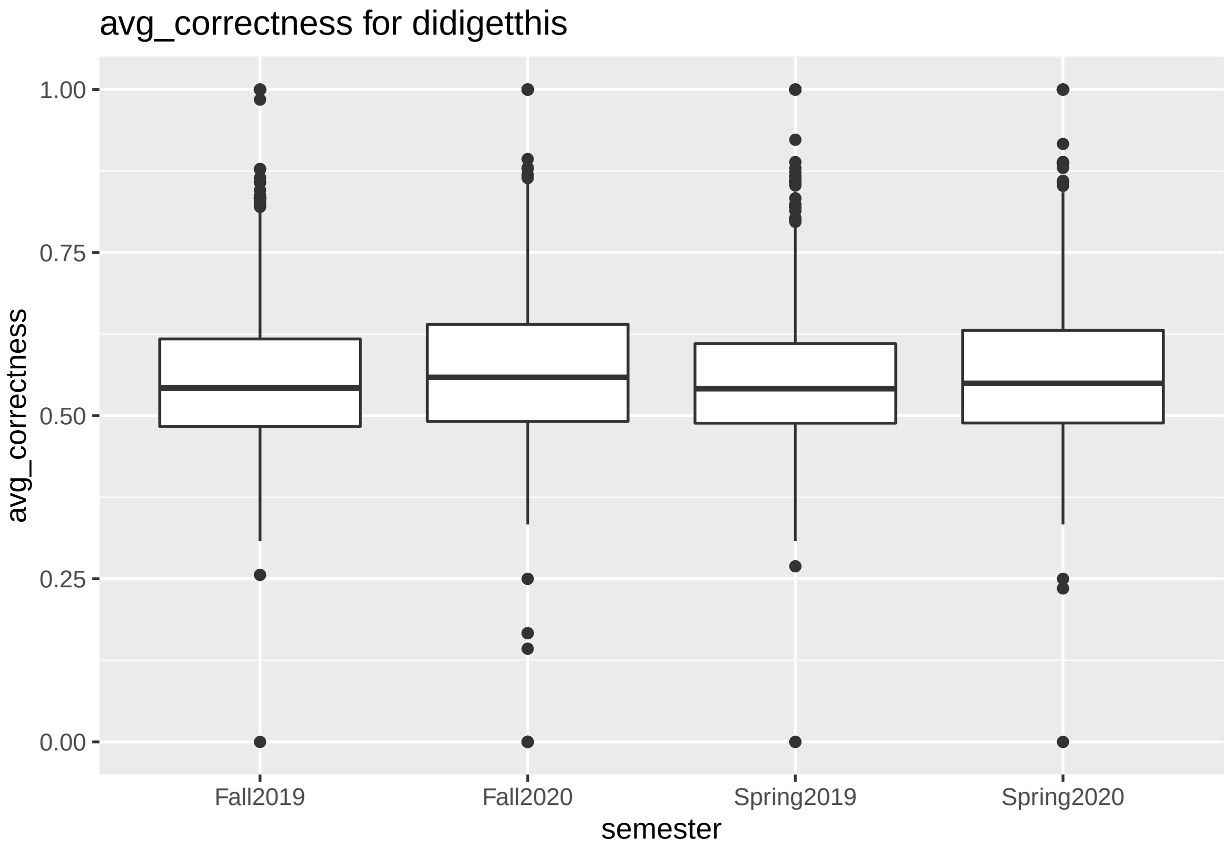
```r
question_all1 %>%
  filter(purpose=="learnbydoing")%>%
  ggplot(aes( x=semester, y = correct/(correct+errors)))+
  geom_boxplot()+
  ylab("avg_correctness")+
  labs(title = "avg_correctness for learnbydoing")
```

```
## Warning: Removed 427 rows containing non-finite values (stat_boxplot).
```



```
question_all1 %>%
  filter(purpose=="didigetthis")%>%
  ggplot(aes( x=semester, y = correct/(correct+errors)))+
  geom_boxplot()+
  ylab("avg_correctness")+
  labs(title = "avg_correctness for didigetthis")
```

```
## Warning: Removed 721 rows containing non-finite values (stat_boxplot).
```



From these boxplots, we can have a simple look of all those distributions and find that there are some outliners.

```
library(ggplot2)
library(dplyr)
question_summary_by_activity_20<-read.csv("csv20/question_summary_by_activity.csv",sep = ";")
question_summary_by_activity_20s<-
  question_summary_by_activity_20[question_summary_by_activity_20$course_key
                                  %in% c(21313,21318,21309,21321,21310,21316,21315,
                                         21312,21320,21311,21308,21319,21322,21314),]
```

Figure 4 and Figure 5

```
question_all1$instructor <-"gsuinstructor"

question_all1[question_all1$course_key %in%
                c("10404","10405","10406"),]$instructor<-"BPidgeon7033"

question_all1[question_all1$course_key %in%
                c("10403","10401","10757"),]$instructor<-"dharden@gsu.edu"

question_all1[question_all1$course_key %in%
                c("93385","93387","93388"),]$instructor<-"BPidgeon7033"

question_all1[question_all1$course_key %in%
                c("93383","93384","81685"),]$instructor<-"dharden@gsu.edu"
```

```
question_all1[question_all1$course_key %in%
                c("21313","21315"),]$instructor<-"BPidgeon7033"


question_all1[question_all1$course_key %in%
                c("21308","21312","10711"),]$instructor<-"dharden@gsu.edu"




question_all1[question_all1$course_key %in%
                c("81605","90984","90985"),]$instructor<-"dharden@gsu.edu"




question_all1$avg_correctness <-
  question_all1$correct/(question_all1$correct+question_all1$errors)


question_all2<-question_all1
question_all2$semester<-
  as.character(question_all2$semester)


question_all2$instructor <-"gsuinstructor"

question_all2[question_all2$course_key
              %in% c("10404","10405",
                     "10406"),]$instructor<-"BPidgeon7033"

question_all2[question_all2$course_key
              %in% c("10403","10401",
                     "10757"),]$instructor<-"dharden@gsu.edu"

question_all2[question_all2$course_key
              %in% c("93385","93387",
                     "93388"),]$instructor<-"BPidgeon7033"

question_all2[question_all2$course_key
              %in% c("93383","93384",
                     "81685"),]$instructor<-"dharden@gsu.edu"



question_all2[question_all2$course_key
              %in% c("21313","21315"),
              ]$instructor<-"BPidgeon7033"


question_all2[question_all2$course_key
              %in% c("21308","21312",
                     "10711"),]$instructor<-"dharden@gsu.edu"



question_all2[question_all2$course_key
              %in% c("81605","90984",
                     "90985"),]$instructor<-"dharden@gsu.edu"

summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {
    library(plyr)

    # New version of length which can handle NA's: if na.rm==T, don't count them
    length2 <- function (x, na.rm=FALSE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }

    # This does the summary. For each group's data frame, return a vector with
    # N, mean, and sd
    datac <- ddply(data, groupvars, .drop=.drop,
      .fun = function(xx, col) {
        c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
        )
      },
      measurevar
    )

    # Rename the "mean" column
    datac <- rename(datac, c("mean" = measurevar))

    datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult

    return(datac)
}





tgc1 <- summarySE(na.omit( question_all2),
                  measurevar="avg_correctness",
                  groupvars=c("purpose","semester"))
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
tgc1$assess<-tgc1$purpose
tgc1[tgc1$purpose%in% c("didigetthis","learnbydoing"),]$assess<-"low"
```

```
## Warning in `[<-.factor`(`*tmp*`, iseq, value = c("low", "low", "low", "low", :
## invalid factor level, NA generated
```

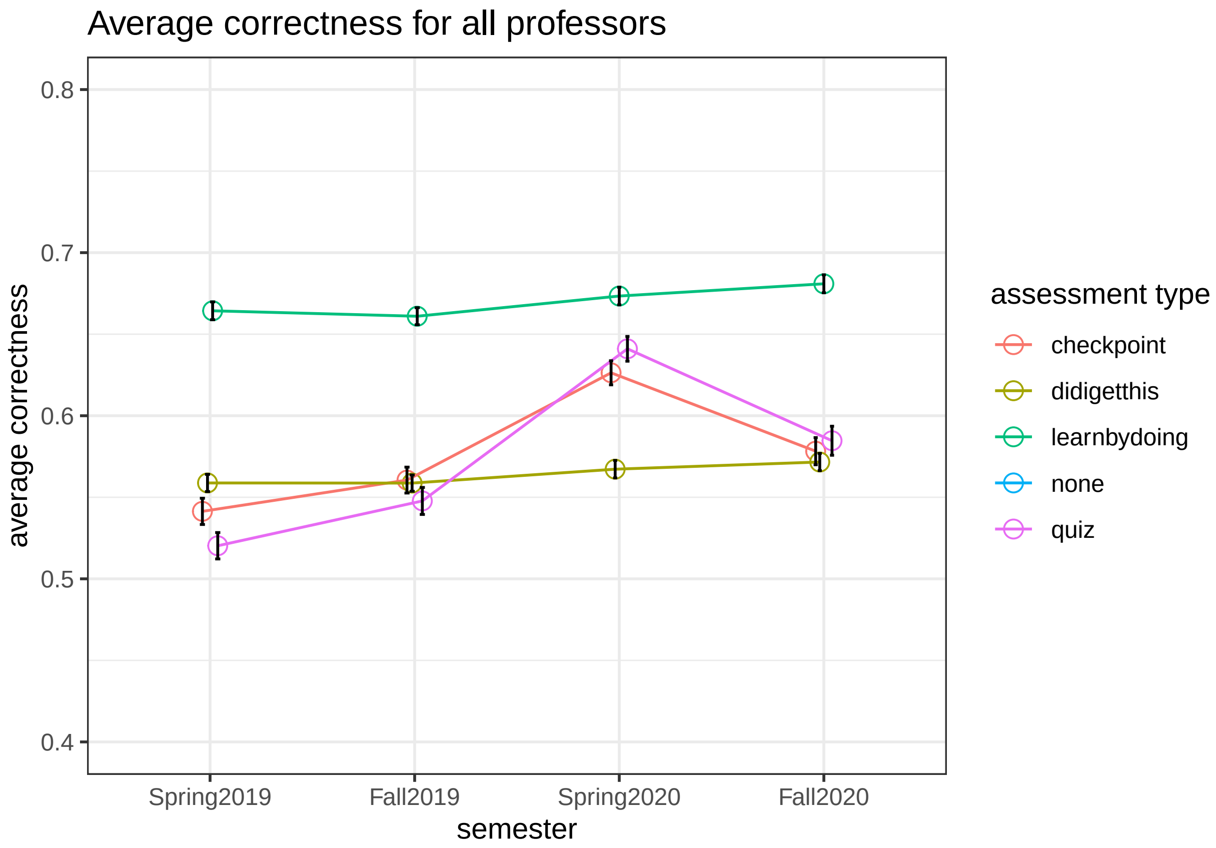
```
tgc1[tgc1$purpose%in% c("checkpoint","quiz"),]$assess<-"high"
```

```
## Warning in `[<-.factor`(`*tmp*`, iseq, value = c("high", "high", "high", :
## invalid factor level, NA generated
```

```
pd <- position_dodge(0.1)
ggplot(tgc1,
       aes(x=factor(semester,
                    level=c("Spring2019","Fall2019","Spring2020","Fall2020")),
           y=avg_correctness,
           group=purpose,colour=purpose)) +
    geom_point(position=pd,shape=21,
               size=3, fill="white")+
   geom_line(position=pd)+
    geom_errorbar(aes(ymin=avg_correctness-ci,
                      ymax=avg_correctness+ci),colour="black",
                  width=.1, position=pd) +
  scale_y_continuous(breaks = c(.4,.5,.6,.7,.8),limits = c(.4,.8))+
  ylab("average correctness")+
  xlab("semester")+
  labs(title="Average correctness for all professors",col="assessment type")+
  theme_bw()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

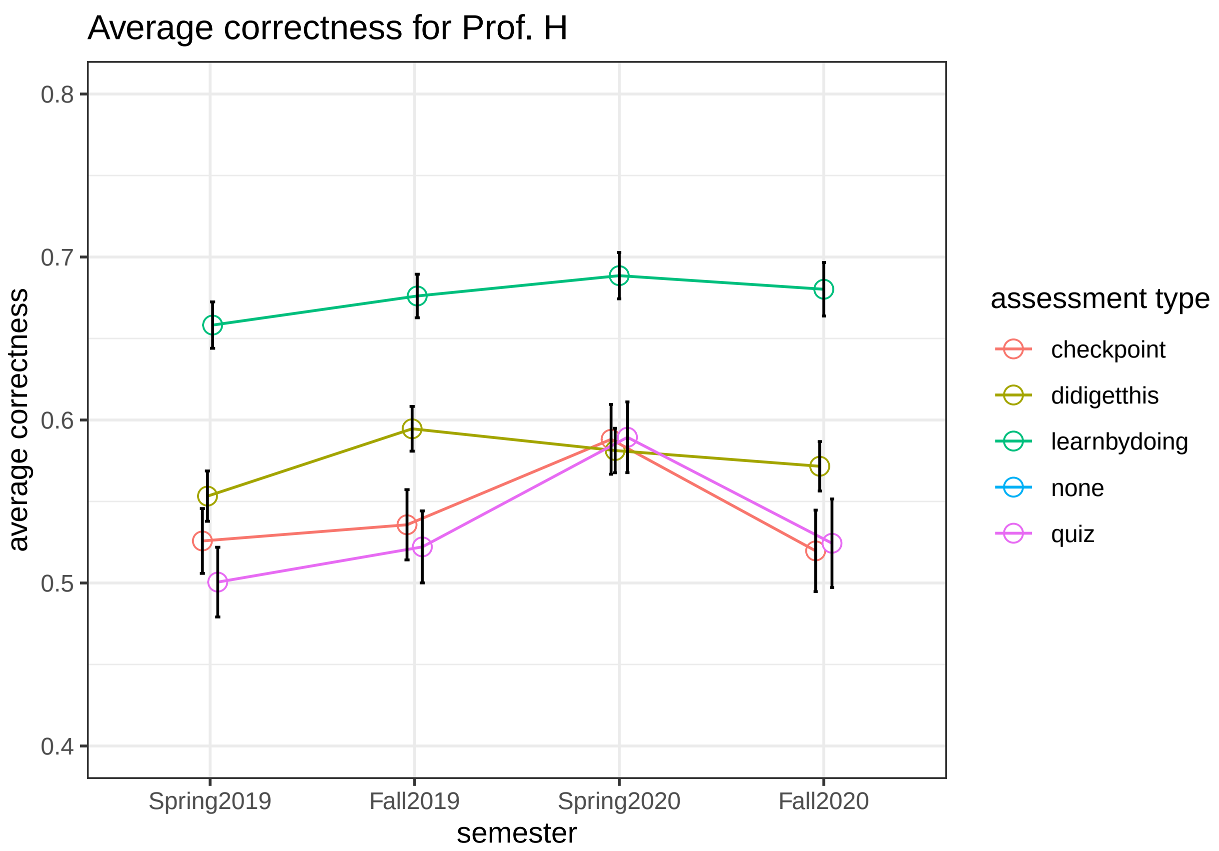
```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



```
question_all3 <- question_all2 %>%
  filter(purpose=="quiz")

#instructor-prof. H
tgc <- summarySE(na.omit(  filter(question_all2,instructor=="dharden@gsu.edu")),
                 measurevar="avg_correctness",
                 groupvars=c("purpose","semester"))
pd <- position_dodge(0.1)
ggplot(tgc, aes(x=factor(semester,
                         level=c("Spring2019","Fall2019","Spring2020","Fall2020")),
                y=avg_correctness, group=purpose,colour=purpose)) +
    geom_point(position=pd, size=3, shape=21, fill="white")+
   geom_line(position=pd)+
    geom_errorbar(aes(ymin=avg_correctness-ci,
                      ymax=avg_correctness+ci),
                  colour="black", width=.1, position=pd) +
  scale_y_continuous(breaks = c(.4,.5,.6,.7,.8),
                     limits = c(.4,.8))+
  ylab("average correctness")+
  xlab("semester")+
  labs(title="Average correctness for Prof. H",
       col="assessment type")+
  theme_bw()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



```
question_all4<-question_all3 %>%
filter(instructor=="dharden@gsu.edu")
```

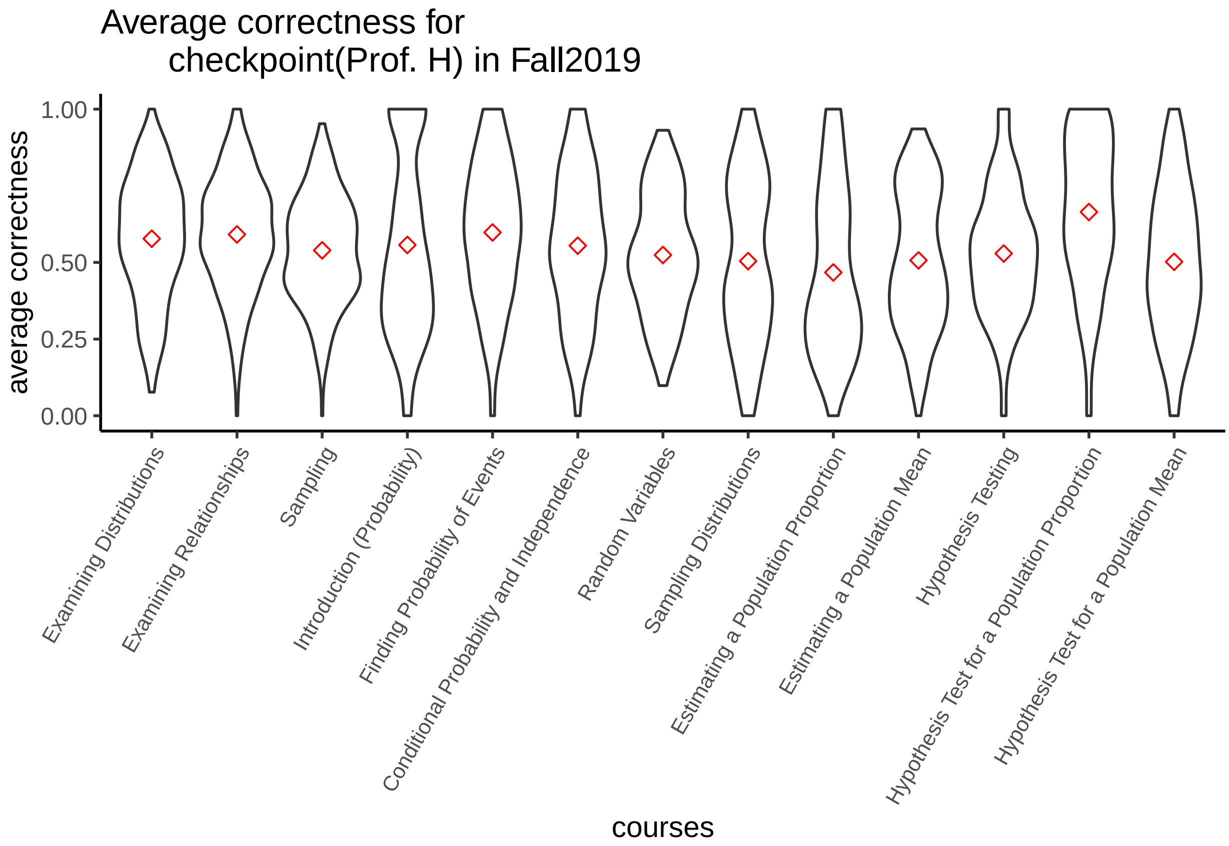
Violin plots for Fall2019

```
modules_19f<-read.csv("csv19f/modules.csv",sep = ";")


question_summary_by_module_19f<-
  read.csv("csv19f/question_summary_by_module.csv",sep = ";")

unique(question_summary_by_module_19f$purpose)
```

```
## [1] checkpoint   learnbydoing didigetthis  none
## Levels: checkpoint didigetthis learnbydoing none
```

```
question_overtime_19f<-
  left_join(question_summary_by_module_19f,modules_19f[,c(5,6)], by="module_id")


question_overtime1_19f<-
  question_overtime_19f[question_overtime_19f$module_title
                      %in% c("Learning Strategies","The Big Picture",
                             "Introduction to Microsoft Excel",
                             "Examining Distributions","Examining Relationships",
                             "Sampling","Introduction (Probability)",
                             "Finding Probability of Events",
                             "Conditional Probability and Independence",
                             "Random Variables","Sampling Distributions",
                             "Estimating a Population Proportion",
                             "Estimating a Population Mean","Hypothesis Testing",
                             "Hypothesis Test for a Population Proportion",
                             "Hypothesis Test for a Population Mean"),]

question_overtime1_19f$instructor <-"gsuinstructor"






question_overtime1_19f[question_overtime1_19f$course_key %in%
                         c("93383","93384","81685"),]$instructor<-"dharden@gsu.edu"

question_overtime1_19f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="checkpoint")%>%
  ggplot(aes( x=factor(module_title,
                       level=c("Learning Strategies","The Big Picture",
                             "Introduction to Microsoft Excel",
                             "Examining Distributions",
                             "Examining Relationships","Sampling",
                             "Introduction (Probability)",
                             "Finding Probability of Events",
                             "Conditional Probability and Independence",
                             "Random Variables","Sampling Distributions",
                             "Estimating a Population Proportion",
                             "Estimating a Population Mean",
                             "Hypothesis Testing",
                             "Hypothesis Test for a Population Proportion",
                             "Hypothesis Test for a Population Mean")),
            y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for
       checkpoint(Prof. H) in Fall2019")+
  theme(axis.text.x =
          element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean,
               geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 425 rows containing non-finite values (stat_ydensity).

## Warning: Removed 425 rows containing non-finite values (stat_summary).
```

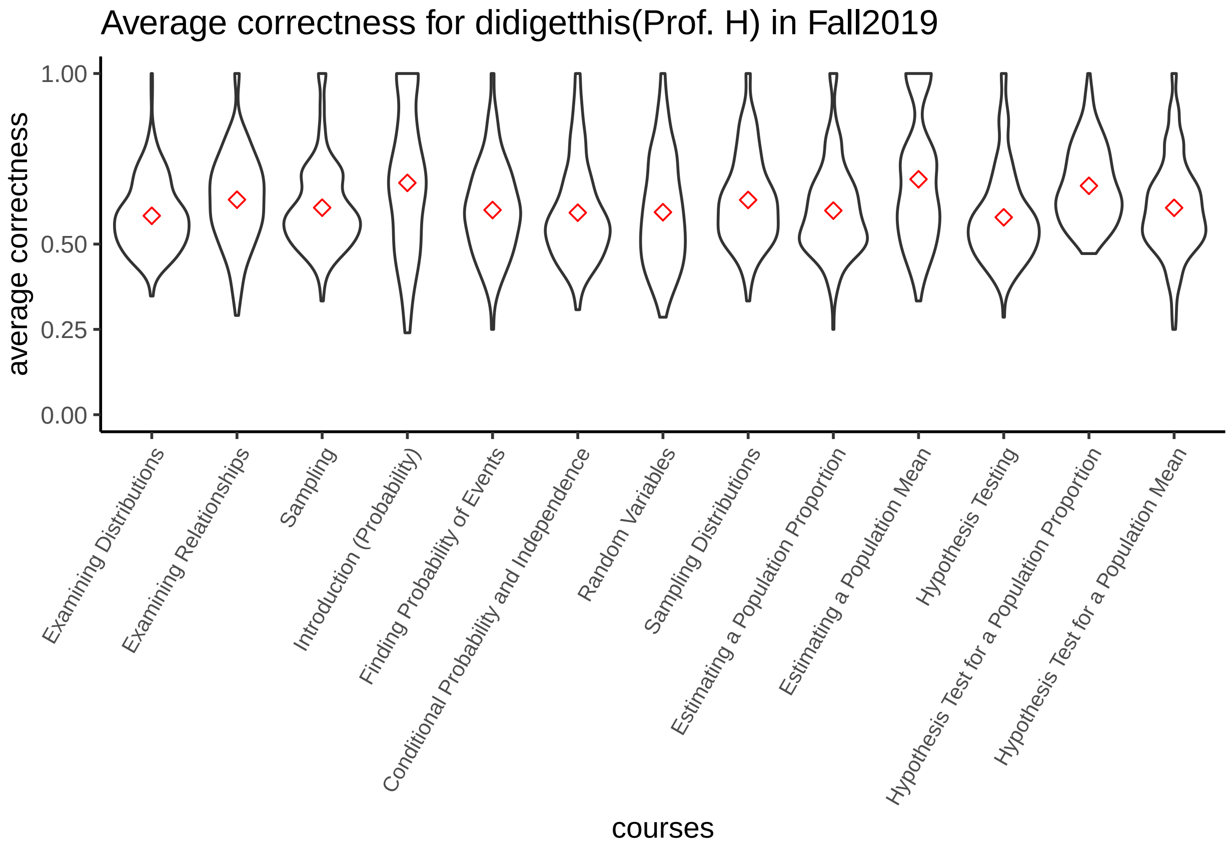


```
question_overtime1_19f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="didigetthis")%>%
  ggplot(aes( x=factor(module_title,level=c("Learning Strategies","The Big Picture",
                              "Introduction to Microsoft Excel",
                              "Examining Distributions",
                              "Examining Relationships","Sampling",
                              "Introduction (Probability)",
                              "Finding Probability of Events",
                              "Conditional Probability and Independence",
                              "Random Variables","Sampling Distributions",
                              "Estimating a Population Proportion",
                              "Estimating a Population Mean",
                              "Hypothesis Testing",
                              "Hypothesis Test for a Population Proportion",
                              "Hypothesis Test for a Population Mean")),
             y = correct/(correct+errors)))+
 geom_violin()+
 ylab("average correctness")+
 xlab("courses")+
 labs(title = "Average correctness for didigetthis(Prof. H) in Fall2019")+
 theme(axis.text.x =
         element_text(angle = 60, hjust = 1,size = 8))+
 stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
 scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
 theme( # Hide panel borders and remove grid lines
 panel.border = element_blank(),
 panel.background = element_blank(),
 panel.grid.major = element_blank(),
 panel.grid.minor = element_blank(),
 # Change axis line
 axis.line = element_line(colour = "black")
 )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

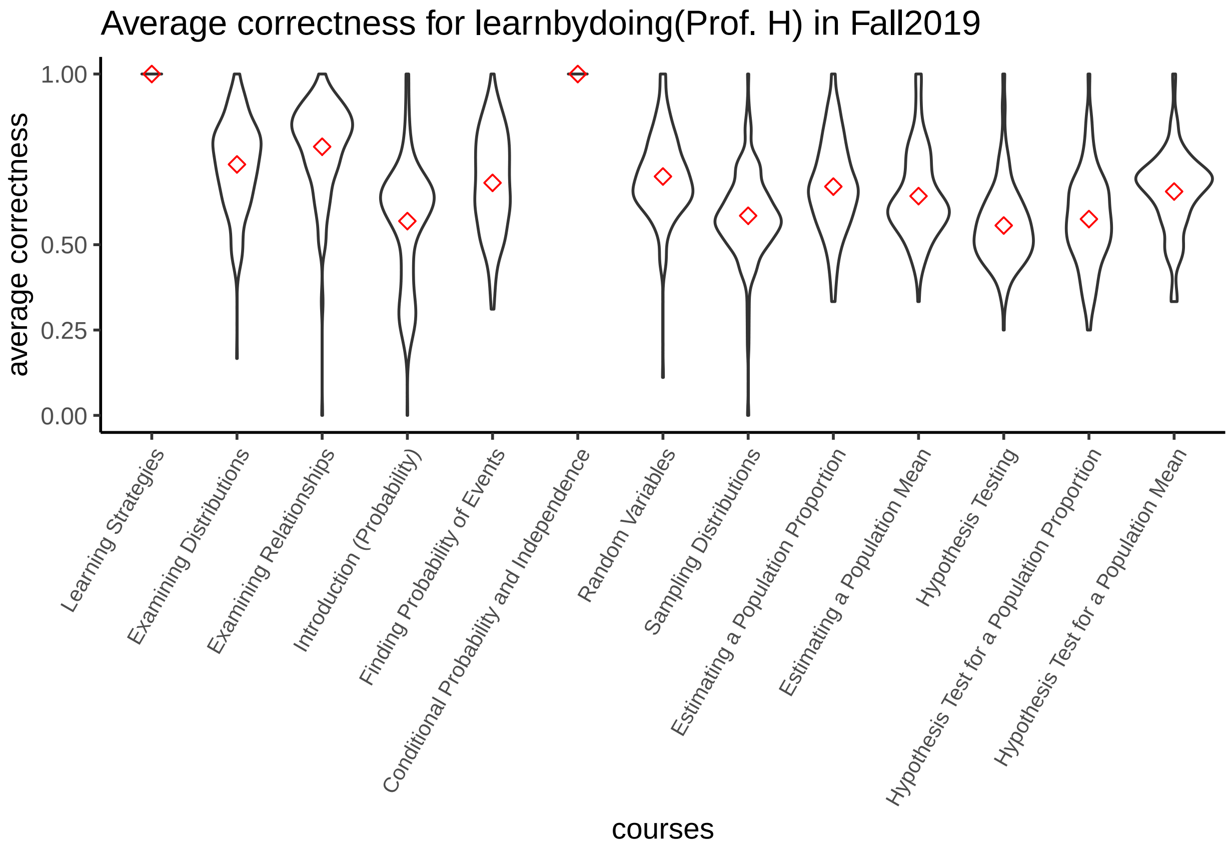
```
## Warning: Removed 1036 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 1036 rows containing non-finite values (stat_summary).
```



```
question_overtime1_19f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="learnbydoing")%>%
  ggplot(aes( x=factor(module_title,level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
              y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for learnbydoing(Prof. H) in Fall2019")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 1026 rows containing non-finite values (stat_ydensity).

## Warning: Removed 1026 rows containing non-finite values (stat_summary).
```



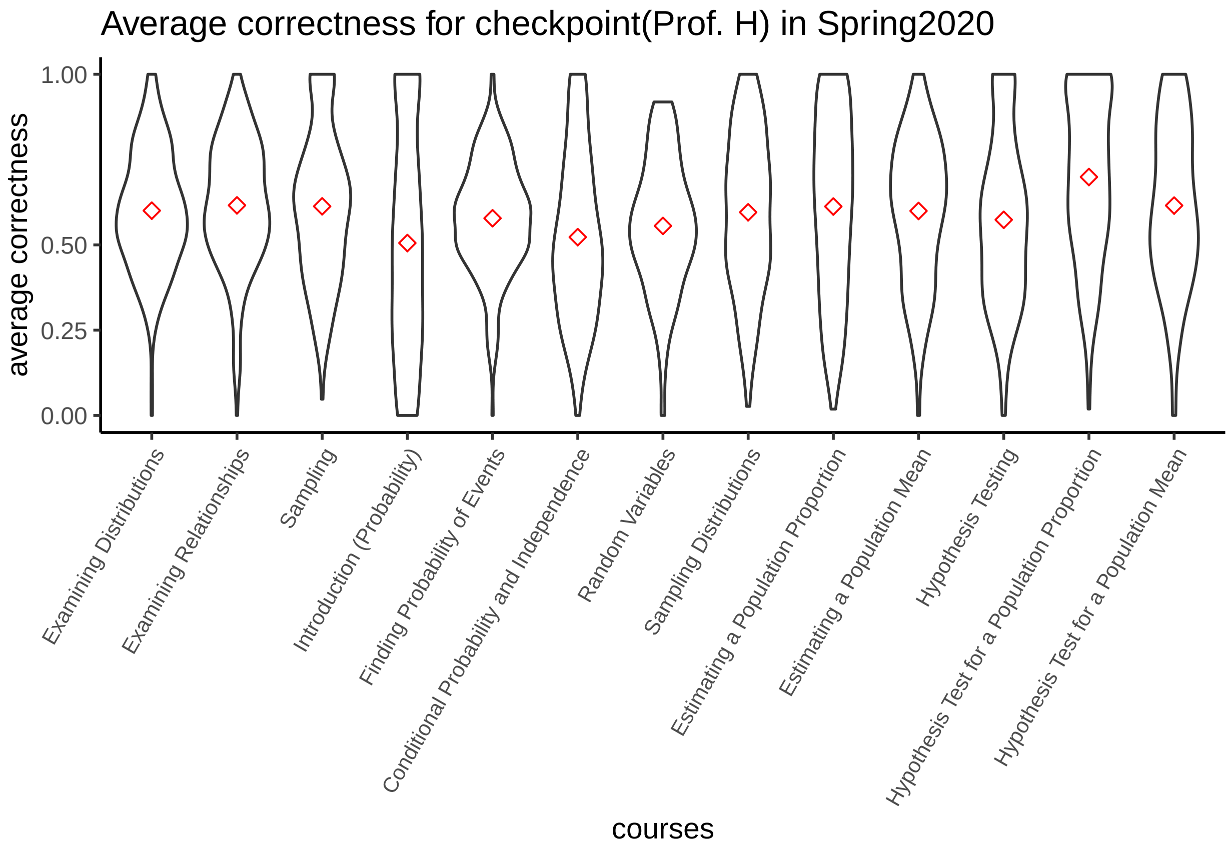
Violin plots for Spring2020

```
library(ggplot2)
library(dplyr)
modules_20<-read.csv("csv20/modules.csv",sep = ";")


question_summary_by_module_20<-
  read.csv("csv20/question_summary_by_module.csv",sep = ";")

unique(question_summary_by_module_20$purpose)
```

```
## [1] checkpoint   learnbydoing didigetthis  none
## Levels: checkpoint didigetthis learnbydoing none
```

```
question_overtime_20<-left_join(question_summary_by_module_20,
                                modules_20[,c(5,6)], by="module_id")
question_overtime_20s<-question_overtime_20[question_overtime_20$course_key
                                            %in% c(21313,21318,21309,21321,21310,
                                              21316,21315,21312,21320,21311,21308,
                                              21319,21322,21314),]

question_overtime1_20s<-
  question_overtime_20s[question_overtime_20s$module_title
                        %in% c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean"),]




question_overtime1_20s$instructor <-"gsuinstructor"







question_overtime1_20s[question_overtime1_20s$course_key
                       %in% c("21308","21312",
                              "10711"),]$instructor<-"dharden@gsu.edu"




question_overtime_20s_first3<-
  question_overtime_20s[question_overtime_20s$module_title
                        %in% c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel"),]




question_overtime1_20s%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="checkpoint")%>%
  ggplot(aes( x=factor(module_title,level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
              y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for checkpoint(Prof. H) in Spring2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 559 rows containing non-finite values (stat_ydensity).

## Warning: Removed 559 rows containing non-finite values (stat_summary).
```



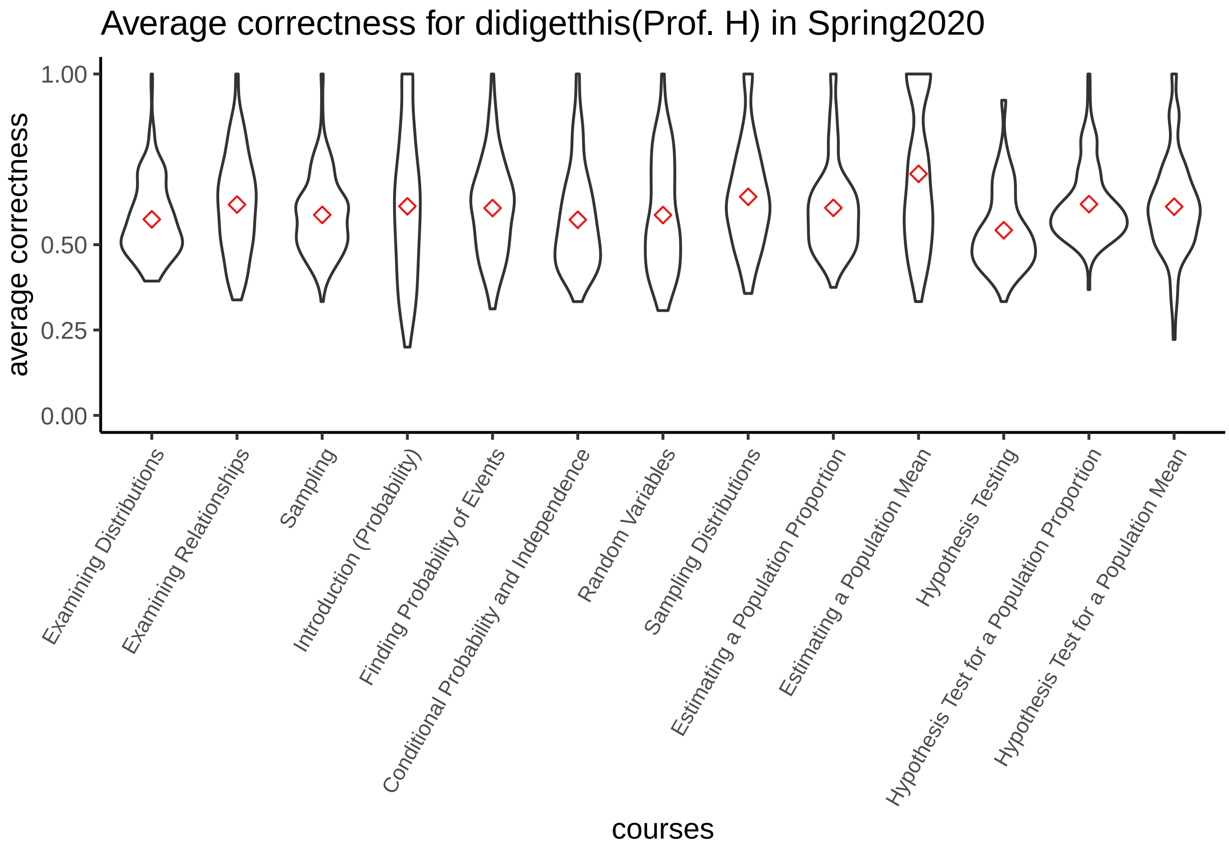
```
question_overtime1_20s%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="didigetthis")%>%
```

```r
  ggplot(aes( x=factor(module_title,
                       level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
              y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for didigetthis(Prof. H) in Spring2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
## Warning: Removed 984 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 984 rows containing non-finite values (stat_summary).
```

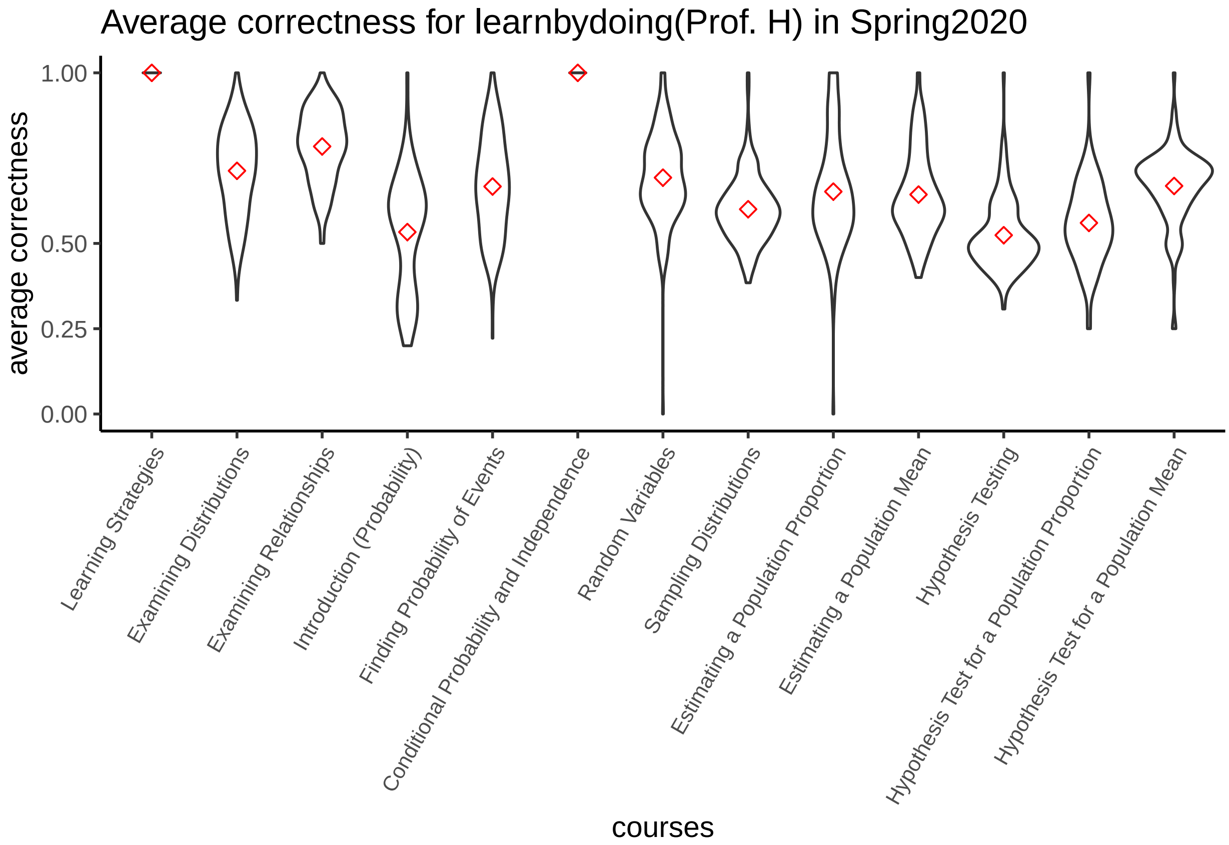


```
question_overtime1_20s%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="learnbydoing")%>%
  ggplot(aes( x=factor(module_title,
                      level=c("Learning Strategies","The Big Picture",
                              "Introduction to Microsoft Excel",
                              "Examining Distributions",
                              "Examining Relationships","Sampling",
                              "Introduction (Probability)",
                              "Finding Probability of Events",
                              "Conditional Probability and Independence",
                              "Random Variables","Sampling Distributions",
                              "Estimating a Population Proportion",
                              "Estimating a Population Mean",
                              "Hypothesis Testing",
                              "Hypothesis Test for a Population Proportion",
                              "Hypothesis Test for a Population Mean")),
             y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for learnbydoing(Prof. H) in Spring2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
## Warning: Removed 977 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 977 rows containing non-finite values (stat_summary).
```



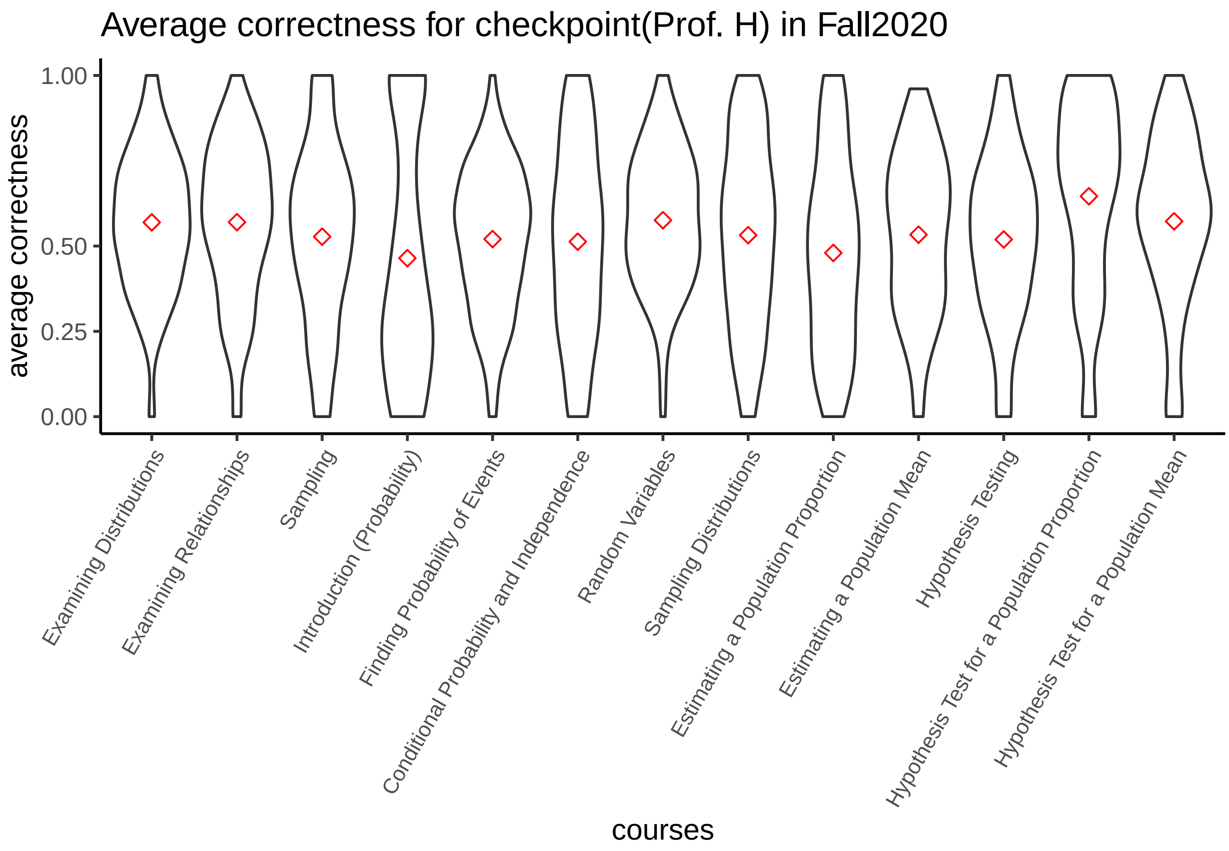
Violin plots for Fall2020

```
unique(question_summary_by_module_20$purpose)
```

```
## [1] checkpoint   learnbydoing didigetthis  none
## Levels: checkpoint didigetthis learnbydoing none
```

```
question_overtime_20<-left_join(question_summary_by_module_20,modules_20[,c(5,6)], by="module_id")
question_overtime_20f<-
  question_overtime_20[question_overtime_20$course_key
                       %in% c(90988,90984,90372,90995,93110,90487,
                              90989,92275,90370,92055,90368,91561,90996,
                              90993,95340,90992,90985,90987,90990,
                              90986,93062,94468,90999,90997,
                              90367,90366,93112),]

question_overtime1_20f<-
  question_overtime_20f[question_overtime_20f$module_title
                                              %in%
                          c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean"),]



question_overtime1_20f$instructor <-"gsuinstructor"







question_overtime1_20f[question_overtime1_20f$course_key
                       %in% c("81605","90984","90985"),
                       ]$instructor<-"dharden@gsu.edu"





question_overtime1_20f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="checkpoint")%>%
  ggplot(aes( x=factor(module_title,
                       level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
               y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for checkpoint(Prof. H) in Fall2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 718 rows containing non-finite values (stat_ydensity).

## Warning: Removed 718 rows containing non-finite values (stat_summary).
```

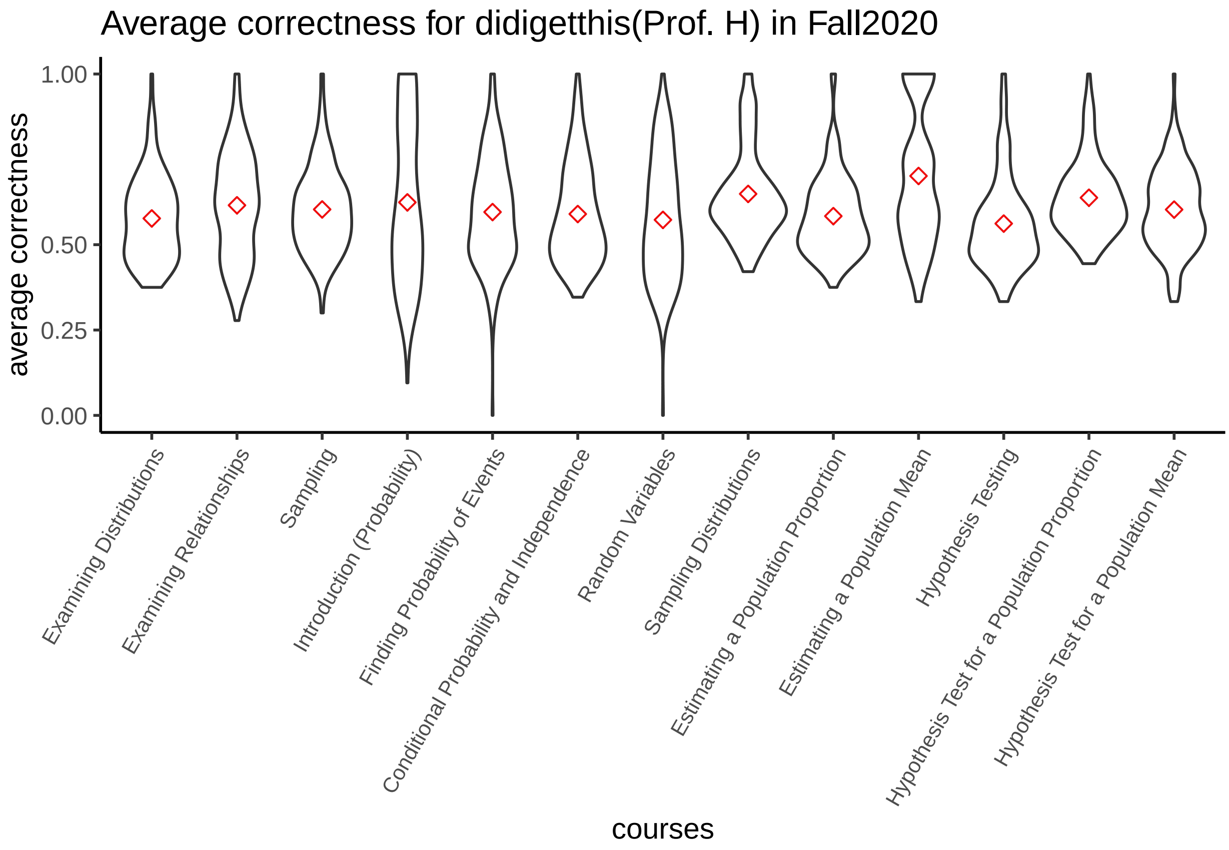


```
question_overtime1_20f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="didigetthis")%>%
  ggplot(aes( x=factor(module_title,
                       level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
              y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for didigetthis(Prof. H) in Fall2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

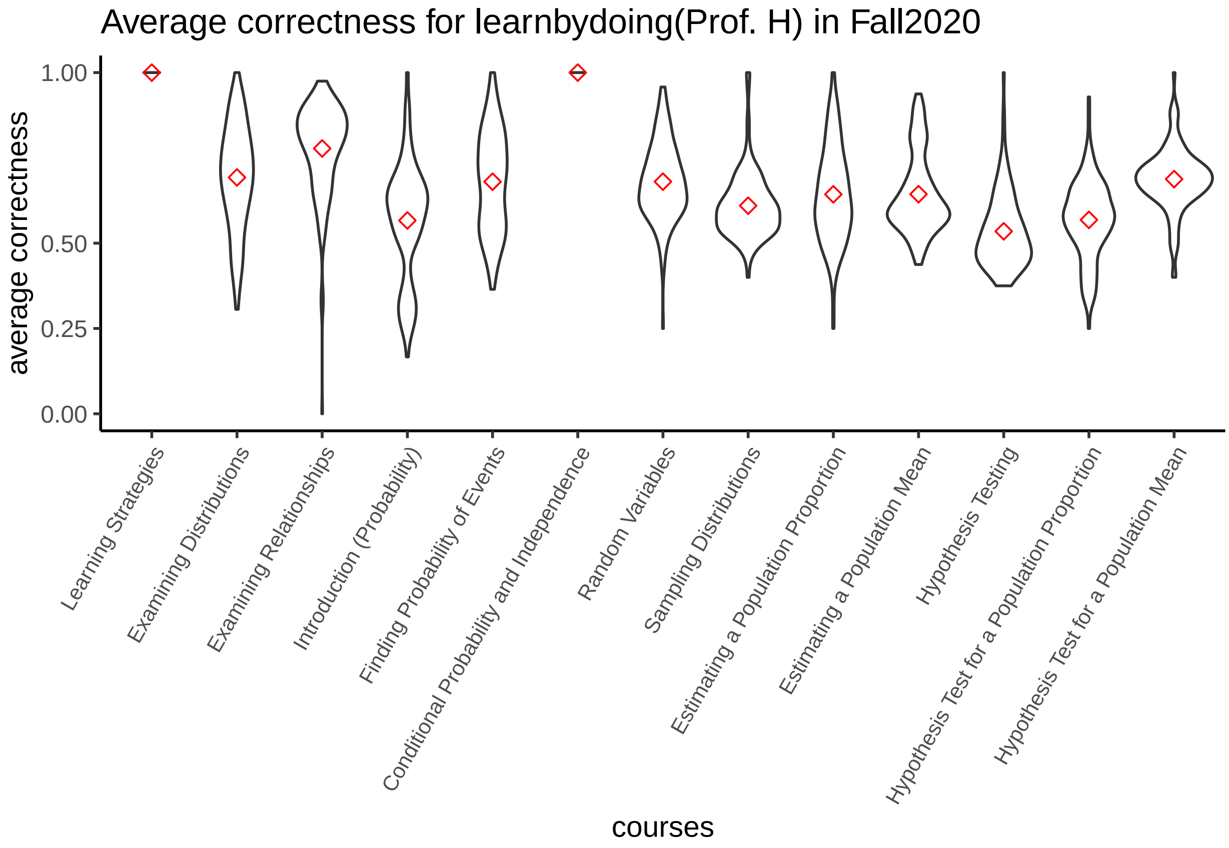
```
## Warning: Removed 1093 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 1093 rows containing non-finite values (stat_summary).
```



```
question_overtime1_20f%>%
  filter(instructor=="dharden@gsu.edu")%>%
  filter(purpose=="learnbydoing")%>%
  ggplot(aes( x=factor(module_title,
                       level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
              y = correct/(correct+errors)))+
  geom_violin()+
  ylab("average correctness")+
  xlab("courses")+
  labs(title = "Average correctness for learnbydoing(Prof. H) in Fall2020")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=2, color = "red")+
  scale_y_continuous(breaks = c(0,.25,0.5,1.0),limits = c(0,1))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Removed 1100 rows containing non-finite values (stat_ydensity).

## Warning: Removed 1100 rows containing non-finite values (stat_summary).
```

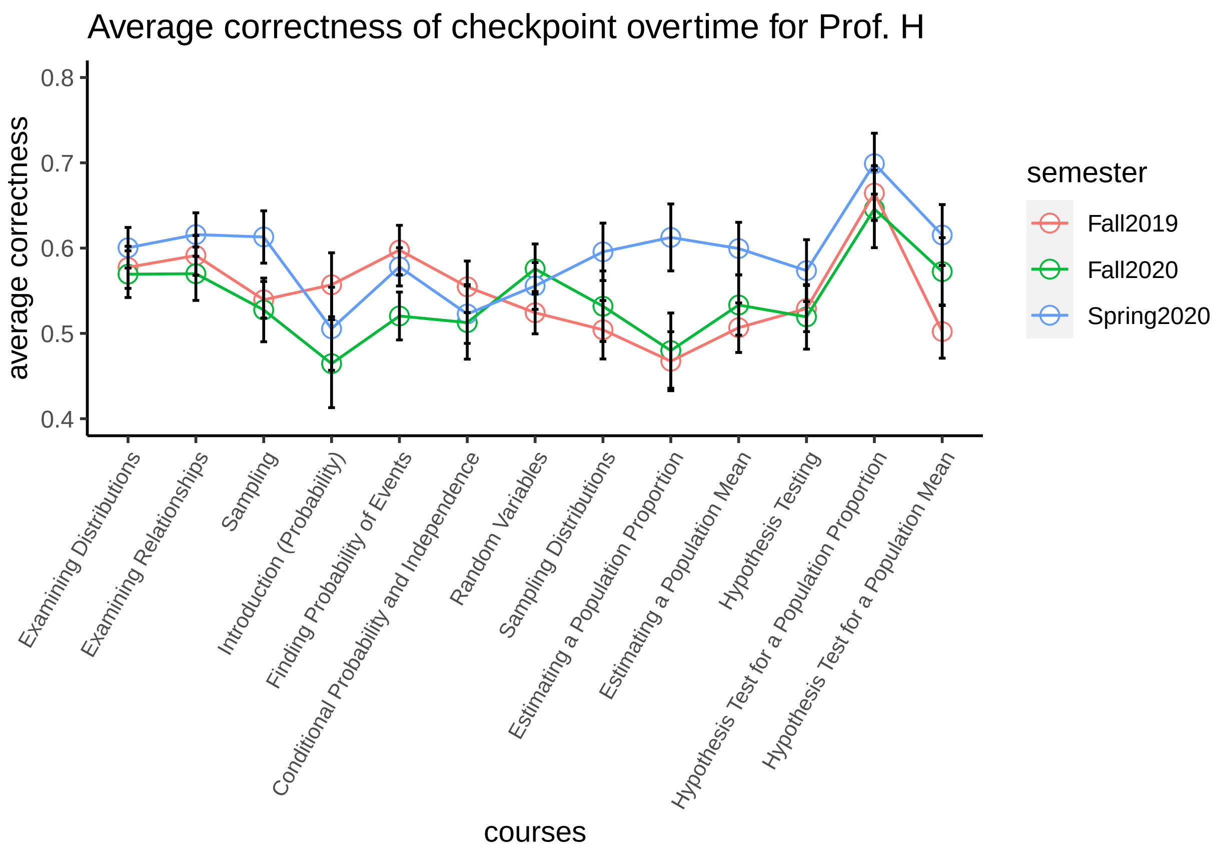


From all these violin plots, we can't see much difference of the mean change but we can clearly see that some distributions change significantly for the same courses

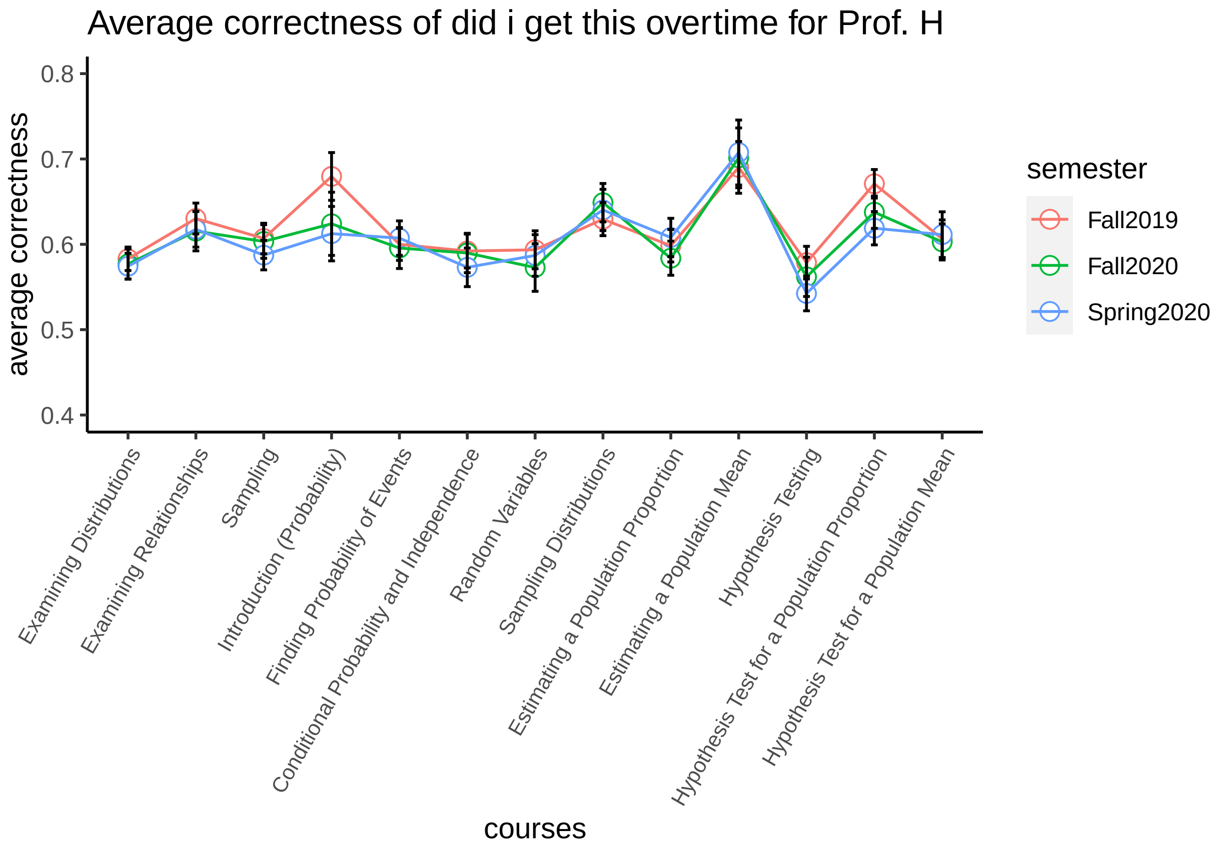
Figure 6,7,8

```
question_overtime1_19f$semester<-"Fall2019"
question_overtime1_20s$semester<-"Spring2020"
question_overtime1_20f$semester<-"Fall2020"

question_overtime1 <- rbind(question_overtime1_19f,
                            question_overtime1_20f,
                            question_overtime1_20s)
question_overtime1$semester<-
```

```
  as.factor(question_overtime1$semester)


question_overtime1$avg_correctness<-
  question_overtime1$correct/(question_overtime1$correct
                              +question_overtime1$errors)

tgc2 <- summarySE(na.omit(filter(filter(question_overtime1,
                                        instructor=="dharden@gsu.edu"),
                                 purpose=="checkpoint")),
                  measurevar="avg_correctness",
                  groupvars=c("semester","module_title"))
pd <- position_dodge(0.1)

tgc3 <- summarySE(na.omit(filter(filter(question_overtime1,
                                        instructor=="dharden@gsu.edu"),
                                 purpose=="didigetthis")),
                  measurevar="avg_correctness",
                  groupvars=c("semester","module_title"))

tgc4 <- summarySE(na.omit(filter(filter(question_overtime1,
                                        instructor=="dharden@gsu.edu"),
                                 purpose=="learnbydoing")),
                  measurevar="avg_correctness",
                  groupvars=c("semester","module_title"))


ggplot(tgc2, aes( x=factor(module_title,
                           level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
                  y=avg_correctness, group=semester,colour=semester)) +
    geom_point( size=3, shape=21, fill="white")+
   geom_line()+
    geom_errorbar(aes(ymin=avg_correctness-ci, ymax=avg_correctness+ci),colour="black", width=.1) +
  scale_y_continuous(breaks = c(.4,.5,.6,.7,.8),limits = c(.4,.8))+
  ylab("average correctness")+
  xlab("courses")+
  labs(title="Average correctness of checkpoint overtime for Prof. H",
       col="semester")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```



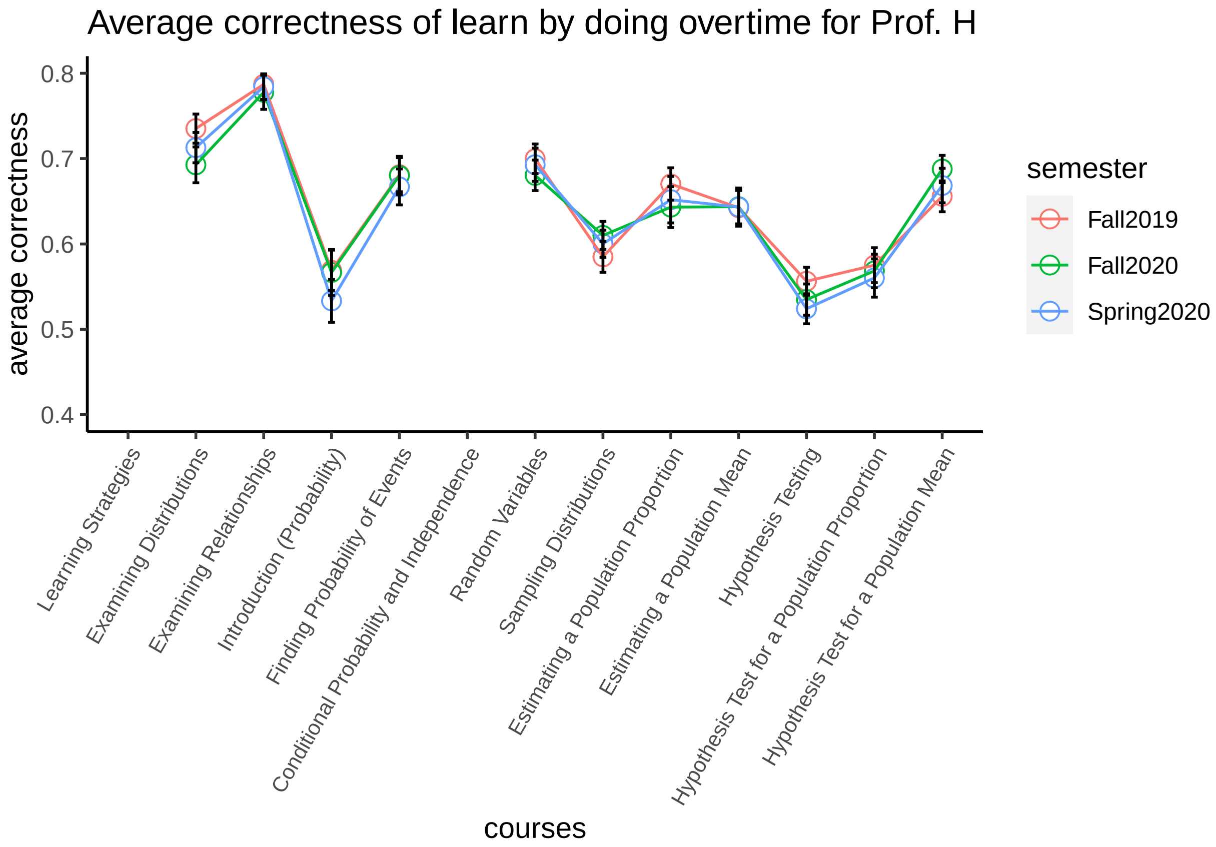
```
ggplot(tgc3, aes( x=factor(module_title,
                           level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
                  y=avg_correctness, group=semester,colour=semester)) +
    geom_point( size=3, shape=21, fill="white")+
   geom_line()+
    geom_errorbar(aes(ymin=avg_correctness-ci, ymax=avg_correctness+ci),colour="black", width=.1) +
  scale_y_continuous(breaks = c(.4,.5,.6,.7,.8),limits = c(.4,.8))+
  ylab("average correctness")+
  xlab("courses")+
  labs(title="Average correctness of did i get this overtime for Prof. H",
       col="semester")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```



```
ggplot(tgc4, aes( x=factor(module_title,
                           level=c("Learning Strategies","The Big Picture",
                               "Introduction to Microsoft Excel",
                               "Examining Distributions",
                               "Examining Relationships","Sampling",
                               "Introduction (Probability)",
                               "Finding Probability of Events",
                               "Conditional Probability and Independence",
                               "Random Variables","Sampling Distributions",
                               "Estimating a Population Proportion",
                               "Estimating a Population Mean",
                               "Hypothesis Testing",
                               "Hypothesis Test for a Population Proportion",
                               "Hypothesis Test for a Population Mean")),
                  y=avg_correctness, group=semester,colour=semester)) +
    geom_point( size=3, shape=21, fill="white")+
   geom_line()+
    geom_errorbar(aes(ymin=avg_correctness-ci, ymax=avg_correctness+ci),colour="black", width=.1) +
  scale_y_continuous(breaks = c(.4,.5,.6,.7,.8),limits = c(.4,.8))+
  ylab("average correctness")+
  xlab("courses")+
  labs(title="Average correctness of learn by doing overtime for Prof. H",
       col="semester")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1,size = 8))+
  theme( # Hide panel borders and remove grid lines
  panel.border = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Change axis line
  axis.line = element_line(colour = "black")
  )
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```



Appendix 3.1

```
instructor_action_s19 <- read.csv("s19_instructor_action_per_week.csv", header = T)
instructor_action_f19 <- read.csv("f19_instructor_action_per_week.csv", header = T)
#instructor_action_exam <- read.csv("instructor_action.csv", header = T)
instructor_action_2020 <- read.csv("20_instructor_action_per_week.csv", header = T)
courses <- read.csv("20_courses.csv", header = T)
courses$start_date <- mdy_hm(courses$start_date)
courses$end_date <- mdy_hm(courses$end_date)
courses <- courses %>%
  mutate(semester = ifelse(month(start_date) == 8, "Fall",
                           ifelse(month(start_date) == 1, "Spring", "Summer")))
course_instructor_2020 <- full_join(instructor_action_2020, courses, by = "course_key")

student_20 <- read.csv("20_student.csv")
student_action_20 <- read.csv("20_student_action_per_week.csv")
student_action_19 <- read.csv("19_student_action.csv")
#head(student_action_20)


s20_course_key <- course_instructor_2020 %>%
  filter(semester == "Spring") %>%
  distinct(course_key, user_id)

f20_course_key <- course_instructor_2020 %>%
  filter(semester == "Fall") %>%
  distinct(course_key, user_id)

student_s20 <- student_20 %>%
  filter(course_key %in% s20_course_key$course_key)

student_f20 <- student_20 %>%
  filter(course_key %in% f20_course_key$course_key)

student_action_s20 <- student_action_20 %>%
  filter(course_key %in% s20_course_key$course_key) %>%
  left_join(s20_course_key, by = c("course_key" = "course_key")) %>%
  filter(week_of_year %in% 3:19) %>%
  rename(instructor_id = user_id.y) %>%
  rename(student_id = user_id.x)

student_action_f20 <- student_action_20 %>%
  filter(course_key %in% f20_course_key$course_key) %>%
  left_join(f20_course_key, by = c("course_key" = "course_key")) %>%
  filter(week_of_year %in% 34:53) %>%
  rename(instructor_id = user_id.y) %>%
  rename(student_id = user_id.x)
```

```
compare <- student_action_s20 %>%
  filter(course_key %in% c("21313", "21308",
                           "21312", "10711")) %>%
  filter(instructor_id %in% c("BPidgeon7033",
                              "dharden@gsu.edu"))
```

```
comparef20 <- student_action_f20 %>%
  filter(course_key %in% c("90984", "90985",
                           "81605")) %>%
  filter(instructor_id %in% c("dharden@gsu.edu"))

compare20_plot <- ggplot(compare, aes(x = as.factor(week_of_year),
                              y = week_logins)) +
  facet_wrap(~instructor_id) +
  geom_boxplot() +
  labs(x = "week",
       y = "students weekly logins",
       title = "Spring 2020 Students Weekly Logins, Pidgeon & Barkson") +
  theme_bw() +
  geom_vline(xintercept = 11.5, color = "red",
             linetype = "dashed") +
  ylim(c(0, 20))

ggsave(plot = compare20_plot, filename = "img1.png",
       width = 10, height = 2.5)
```

```
stu19 <- read.csv("19_student_action.csv")
stuf19 <- read.csv("19f_student_action.csv")
compare19 <- stu19 %>%
  filter(course_key %in% c("10757",
                           "10401",
                           "10404",
                           "10406",
                           "10405")) %>%
  mutate(instructor_id = ifelse(course_key %in% c("10401", "10757"),
                                "dharden@gsu.edu",
                                "BPidgeon7033")) %>%
  filter(week_of_year %in% 3:19)
#
# !!!!!
#
compare19_fall <- stuf19 %>%
  filter(course_key %in% c("93385",
                           "93387",
                           "93388",
                           "93383",
                           "93384",
                           "81685")) %>%
  mutate(instructor_id = ifelse(course_key %in% c("93383", "93384", "81685"),
                                "dharden@gsu.edu",
                                "BPidgeon7033")) %>%
  filter(week_of_year %in% 34:53)

compare19_plot <- ggplot(compare19, aes(x = as.factor(week_of_year),
                              y = week_logins)) +
  facet_wrap(~instructor_id) +
  geom_boxplot() +
  labs(x = "week",
       y = "students weekly logins",
       title = "Spring 2019 Students Weekly Logins, Pidgeon & Barkson") +
  theme_bw() +
  geom_vline(xintercept = 11.5, color = "red",
             linetype = "dashed") +
  ylim(c(0, 20))

ggsave(plot = compare19_plot, filename = "img2.png",
       width = 10, height = 2.5)
```

```
dharde20 <- compare %>% filter(instructor_id == "dharden@gsu.edu")
bpidgeonn20 <- compare %>% filter(instructor_id == "BPidgeon7033")

# t.test(dharde20$week_logins, bpidgeonn20$week_logins)

dharde19 <- compare19 %>% filter(instructor_id == "dharden@gsu.edu")
bpidgeonn19 <- compare19 %>% filter(instructor_id == "BPidgeon7033")

dharde19f <- compare19_fall %>% filter(instructor_id == "dharden@gsu.edu")
bpidgeonn19f <- compare19_fall %>% filter(instructor_id == "BPidgeon7033")

# t.test(dharde19$week_logins, bpidgeonn19$week_logins)

dharde20_before <- dharde20 %>% filter(week_of_year < 14)
dharde20_after <- dharde20 %>% filter(week_of_year >= 14)

dharde19_before <- dharde19 %>% filter(week_of_year < 14)
dharde19_after <- dharde19 %>% filter(week_of_year >= 14)

dharde19f_before <- dharde19f %>% filter(week_of_year < 44)
dharde19f_after <- dharde19f %>% filter(week_of_year >= 44)

# t.test(dharde19_before$week_logins, dharde20_before$week_logins)
# t.test(dharde20_after$week_logins, dharde19_after$week_logins)

bpidgeonn20_before <- bpidgeonn20 %>% filter(week_of_year < 14)
bpidgeonn20_after <- bpidgeonn20 %>% filter(week_of_year >= 14)

bpidgeonn19_before <- bpidgeonn19 %>% filter(week_of_year < 14)
bpidgeonn19_after <- bpidgeonn19 %>% filter(week_of_year >= 14)

bpidgeonn19f_before <- bpidgeonn19f %>% filter(week_of_year < 44)
bpidgeonn19f_after <- bpidgeonn19f %>% filter(week_of_year >= 44)

# t.test(bpidgeonn19_before$week_logins, bpidgeonn20_before$week_logins)
# t.test(bpidgeonn20_after$week_logins, bpidgeonn19_after$week_logins)
# t.test(dharde20_before$week_logins, dharde20_after$week_logins)
# t.test(bpidgeonn20_before$week_logins, bpidgeonn20_after$week_logins)
```

```
dharde20fall_before <- comparef20 %>% filter(week_of_year <= 44)
dharde20fall_after <- comparef20 %>% filter(week_of_year > 44)
```

```r
p <- cbind(c(mean(bpidgeonn19_before$week_logins),
             mean(bpidgeonn19_after$week_logins),
             mean(bpidgeonn20_before$week_logins),
             mean(bpidgeonn20_after$week_logins)))
p <- as.data.frame(p)
p_1 <- c("Week 03-13", "Week 14-19", "Week 03-13", "Week 14-19")
p_2 <- c("2019", "2019", "2020", "2020")
p_3 <- c("Prof. P", "Prof. P", "Prof. P", "Prof. P")
p_4 <- cbind(c(sd(bpidgeonn19_before$week_logins),
             sd(bpidgeonn19_after$week_logins),
             sd(bpidgeonn20_before$week_logins),
             sd(bpidgeonn20_after$week_logins)))
p_5 <- cbind(c(length(bpidgeonn19_before$week_logins),
             length(bpidgeonn19_after$week_logins),
             length(bpidgeonn20_before$week_logins),
             length(bpidgeonn20_after$week_logins)))
p <- as.data.frame(p)
p_bind <- cbind(p, p_1, p_2, p_3, p_4, p_5)
colnames(p_bind) <- c("mean", "pandemic", "year", "instructor", "sd", "n")
p_df <- p_bind %>%
  mutate(error = 1.96*sd/sqrt(n))

h <- cbind(c(mean(dharde19_before$week_logins),
             mean(dharde19_after$week_logins),
             mean(dharde20_before$week_logins),
             mean(dharde20_after$week_logins)))
h <- as.data.frame(h)
h_3 <- c("Prof. H", "Prof. H", "Prof. H", "Prof. H")
h_4 <- cbind(c(sd(dharde19_before$week_logins),
             sd(dharde19_after$week_logins),
             sd(dharde20_before$week_logins),
             sd(dharde20_after$week_logins)))
h_5 <- cbind(c(length(dharde19_before$week_logins),
             length(dharde19_after$week_logins),
             length(dharde20_before$week_logins),
             length(dharde20_after$week_logins)))
h_bind <- cbind(h, p_1, p_2, h_3, h_4, h_5)
colnames(h_bind) <- c("mean", "pandemic", "year", "instructor", "sd", "n")
h_df <- h_bind %>%
  mutate(error = 1.96*sd/sqrt(n))
df_compare <- rbind(p_df, h_df)

avg_login_plot <- ggplot(df_compare, aes(x = pandemic, y=mean, fill=as.factor(year),
                       color=as.factor(year))) +
  geom_point(stat="identity", position=position_dodge(.8), size = 3) +
  geom_errorbar(aes(ymin=mean-error, ymax=mean+error),
                color = "#903C23", width=.2,
                 position=position_dodge(.8)) +
  facet_wrap(~instructor) +
  labs(x = "Before/after pandemic",
       y = "Average student logins per week",
       fill = "Semester",
       color = "Semester") +
  theme_bw() +
  #scale_fill_manual(values=c("#DBCFAA", "#ECECEC")) +
  theme(legend.position = "bottom") +
  ylim(c(0, 5))
ggsave(plot = avg_login_plot, filename = "img3.png",
       width = 6, height = 4)
```

Login Time

```
# load and mutate data
# 2019
login_s19 <-read_csv("s19_login.csv")
login_f19 <-read_csv("f19_login.csv")

# get user id
h_s19_user <- unique(dharde19_before$user_id)
p_s19_user <- unique(bpidgeonn19_before$user_id)
login_s19_h <- login_s19 %>%
  filter(user_id %in% h_s19_user)
login_s19_p <- login_s19 %>%
  filter(user_id %in% p_s19_user)

h_f19_user <- unique(dharde19f_before$user_id)
p_f19_user <- unique(bpidgeonn19f_before$user_id)
login_f19_h <- login_f19 %>%
  filter(user_id %in% h_f19_user)
login_f19_p <- login_f19 %>%
  filter(user_id %in% p_f19_user)


# check if contain all students
# length(unique(login_s19_h$user_id)) == length(h_s19_user)
# length(unique(login_s19_p$user_id)) == length(p_s19_user)

login_s19_h <- login_s19_h %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2019-01-14") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2018/09/Spring-2019-Semester-Calendar.pdf
  filter(date <= "2019-05-07") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)

login_s19_p <- login_s19_p %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2019-01-14") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2018/09/Spring-2019-Semester-Calendar.pdf
  filter(date <= "2019-05-07") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)



login_f19_h <- login_f19_h %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2019-08-26") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2019/02/Fall-2019-Semester-Calendar.pdf
  filter(date <= "2019-12-17") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)

login_f19_p <- login_f19_p %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2019-08-26") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2019/02/Fall-2019-Semester-Calendar.pdf
  filter(date <= "2019-12-17") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)



login_s19_h_before <- login_s19_h %>%
  filter(date <= min(login_s19_h$date) + 7*11)
login_s19_h_after <- login_s19_h %>%
  filter(date > min(login_s19_h$date) + 7*11)

login_s19_p_before <- login_s19_p %>%
  filter(date <= min(login_s19_p$date) + 7*11)
login_s19_p_after <- login_s19_p %>%
  filter(date > min(login_s19_p$date) + 7*11)


login_f19_h_before <- login_f19_h %>%
  filter(date <= min(login_f19_h$date) + 7*11)
login_f19_h_after <- login_f19_h %>%
  filter(date > min(login_f19_h$date) + 7*11)

login_f19_p_before <- login_f19_p %>%
  filter(date <= min(login_f19_p$date) + 7*11)
```

```
login_f19_p_after <- login_f19_p %>%
  filter(date > min(login_f19_p$date) + 7*11)

# 2020
login_s20 <-read_csv("20_login.csv")

# get user id
h_s20_user <- unique(dharde20_before$student_id)
p_s20_user <- unique(bpidgeonn20_before$student_id)
login_s20_h <- login_s20 %>%
  filter(user_id %in% h_s20_user)
login_s20_p <- login_s19 %>%
  filter(user_id %in% p_s20_user)
h_f20_user <- unique(dharde20fall_before$student_id)
login_f20_h <- login_s20 %>%
  filter(user_id %in% h_f20_user)

login_s20_h <- login_s20_h %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2020-01-13") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2020/05/Spring-2020-5.1.20.pdf
  filter(date <= "2020-05-05") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)

login_s20_p <- login_s20_p %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2020-01-13") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2020/05/Spring-2020-5.1.20.pdf
  filter(date <= "2020-05-05") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)

login_f20_h <- login_f20_h %>%
  mutate(date = sapply(strsplit(login, split = " "), "[", 1)) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  filter(date >= "2020-08-24") %>%
  # base on school calendar
  # https://registrar.gsu.edu/files/2020/08/2020-2021-At-A-Glance-Semester-Calendars.pdf
  filter(date <= "2020-12-15") %>%
  mutate(hour = sapply(strsplit(diff, split = ":"), "[", 1)) %>%
  mutate(minute = sapply(strsplit(diff, split = ":"), "[", 2)) %>%
  mutate(elapsed_min = as.numeric(hour)*24 + as.numeric(minute)) %>%
  select(user_id, date, elapsed_min)
```

```
login_s20_h_before <- login_s20_h %>%
  filter(date <= min(login_s20_h$date) + 7*11)
login_s20_h_after <- login_s20_h %>%
  filter(date > min(login_s20_h$date) + 7*11)
login_f20_h_before <- login_f20_h %>%
  filter(date <= min(login_f20_h$date) + 7*11)
login_f20_h_after <- login_f20_h %>%
  filter(date > min(login_f20_h$date) + 7*11)

login_s20_p_before <- login_s20_p %>%
  filter(date <= min(login_s20_p$date) + 7*11)
login_s20_p_after <- login_s20_p %>%
  filter(date > min(login_s20_p$date) + 7*11)
```

```
# create plot for login times
p <- cbind(c(mean(login_s19_p_before$elapsed_min),
             mean(login_s19_p_after$elapsed_min),
             mean(login_s20_p_before$elapsed_min),
             mean(login_s20_p_after$elapsed_min)))
p <- as.data.frame(p)
p_1 <- c("Week 03-13", "Week 14-19", "Week 03-13", "Week 14-19")
p_2 <- c("2019", "2019", "2020", "2020")
p_3 <- c("Prof. P", "Prof. P", "Prof. P", "Prof. P")
p_4 <- cbind(c(sd(login_s19_p_before$elapsed_min),
             sd(login_s19_p_after$elapsed_min),
             sd(login_s20_p_before$elapsed_min),
             sd(login_s20_p_after$elapsed_min)))
p_5 <- cbind(c(length(login_s19_p_before$elapsed_min),
             length(login_s19_p_after$elapsed_min),
             length(login_s20_p_before$elapsed_min),
             length(login_s20_p_after$elapsed_min)))
p_bind <- cbind(p, p_1, p_2, p_3, p_4, p_5)
colnames(p_bind) <- c("mean", "pandemic", "year", "instructor", "sd", "n")
p_df <- p_bind %>%
  mutate(error = 1.96*sd/sqrt(n))
p <- cbind(c(mean(login_s19_p_before$elapsed_min),
             mean(login_s19_p_after$elapsed_min),
             mean(login_s20_p_before$elapsed_min),
             mean(login_s20_p_after$elapsed_min)))
p <- as.data.frame(p)
p_1 <- c("Week 03-13", "Week 14-19", "Week 03-13", "Week 14-19")
p_2 <- c("2019", "2019", "2020", "2020")
p_3 <- c("Prof. P", "Prof. P", "Prof. P", "Prof. P")
p_4 <- cbind(c(sd(login_s19_p_before$elapsed_min),
             sd(login_s19_p_after$elapsed_min),
             sd(login_s20_p_before$elapsed_min),
             sd(login_s20_p_after$elapsed_min)))
p_5 <- cbind(c(length(login_s19_p_before$elapsed_min),
             length(login_s19_p_after$elapsed_min),
             length(login_s20_p_before$elapsed_min),
             length(login_s20_p_after$elapsed_min)))
p_bind <- cbind(p, p_1, p_2, p_3, p_4, p_5)
colnames(p_bind) <- c("mean", "pandemic", "year", "instructor", "sd", "n")
```

```
p_df <- p_bind %>%
  mutate(error = 1.96*sd/sqrt(n))


h <- cbind(c(mean(login_s19_h_before$elapsed_min),
             mean(login_s19_h_after$elapsed_min),
             mean(login_s20_h_before$elapsed_min),
             mean(login_s20_h_after$elapsed_min)))
h <- as.data.frame(h)
h_1 <- c("Week 03-13", "Week 14-19", "Week 03-13", "Week 14-19")
h_2 <- c("2019", "2019", "2020", "2020")
h_3 <- c("Prof. H", "Prof. H", "Prof. H", "Prof. H")
h_4 <- cbind(c(sd(login_s19_h_before$elapsed_min),
             sd(login_s19_h_after$elapsed_min),
             sd(login_s20_h_before$elapsed_min),
             sd(login_s20_h_after$elapsed_min)))
h_5 <- cbind(c(length(login_s19_h_before$elapsed_min),
             length(login_s19_h_after$elapsed_min),
             length(login_s20_h_before$elapsed_min),
             length(login_s20_h_after$elapsed_min)))
h_bind <- cbind(h, h_1, h_2, h_3, h_4, h_5)
colnames(h_bind) <- c("mean", "pandemic", "year", "instructor", "sd", "n")
h_df <- h_bind %>%
  mutate(error = 1.96*sd/sqrt(n))

df_logintime_compare <- rbind(p_df, h_df)

avg_time_plot <- ggplot(df_logintime_compare, aes(x = pandemic, y=mean/60, fill=as.factor(year),
                        color=as.factor(year))) +
  geom_point(stat="identity", position=position_dodge(.8), size = 3) +
  geom_errorbar(aes(ymin=(mean-error)/60, ymax=(mean+error)/60),
              color = "#903C23", width=.2,
               position=position_dodge(.8)) +
  facet_wrap(~instructor) +
  labs(x = "Before/after pandemic",
       y = "Average elapsed time \nfor each student login (minutes)",
       fill = "Semester",
       color = "Semester") +
  theme_bw() +
  #scale_fill_manual(values=c("#DBCFAA", "#ECECEC")) +
  theme(legend.position = "bottom") +
  ylim(c(0, 40))

ggsave(plot = avg_time_plot, filename = "img4.png",
       width = 6, height = 4)
```

```
t.test(login_s19_h_before$elapsed_min, login_s20_h_before$elapsed_min)
t.test(login_s19_p_before$elapsed_min, login_s20_p_before$elapsed_min)

t.test(login_s19_h_after$elapsed_min, login_s20_h_after$elapsed_min)
t.test(login_s19_p_after$elapsed_min, login_s20_p_after$elapsed_min)

t.test(login_s20_h_before$elapsed_min, login_s20_h_after$elapsed_min)
```

```
t.test(login_s20_p_before$elapsed_min, login_s20_p_after$elapsed_min)
```

```
###  violin plot 2020 spring
login_s20_p_violin <- login_s20_p %>%
  mutate(professor = "Prof. P",
         elapsed_hour = elapsed_min/60)
login_s20_h_violin <- login_s20_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60)

login_f19_p_violin <- login_f19_p %>%
  mutate(professor = "Prof. P",
         elapsed_hour = elapsed_min/60)
login_f19_h_violin <- login_f19_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60)

login_s19_p_violin <- login_s19_p %>%
  mutate(professor = "Prof. P",
         elapsed_hour = elapsed_min/60)
login_s19_h_violin <- login_s19_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60)

login_f20_violin <- login_f20_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60) %>%
  mutate(week = ifelse(date <= min(login_f20_h$date) + 7*11,
                       "Before Week 14",
                       "After Week 14"),
         semester = "Fall 2020") %>%
  filter(elapsed_hour <= 50)

login_s20_violin <- rbind(login_s20_p_violin,
                          login_s20_h_violin) %>%
  mutate(week = ifelse(date <= min(login_s20_p$date) + 7*11,
                       "Before Week 14",
                       "After Week 14"),
         semester = "Spring 2020") %>%
  filter(elapsed_hour <= 50)
login_s19_violin <- rbind(login_s19_p_violin,
                          login_s19_h_violin) %>%
  mutate(week = ifelse(date <= min(login_s19_p$date) + 7*11,
                       "Before Week 14",
                       "After Week 14"),
         semester = "Spring 2019") %>%
  filter(elapsed_hour <= 50)

login_f19_violin <- rbind(login_f19_p_violin,
                          login_f19_h_violin) %>%
  mutate(week = ifelse(date <= min(login_f19_p$date) + 7*11,
                       "Before Week 14",
                       "After Week 14"),
         semester = "Fall 2019") %>%
  filter(elapsed_hour <= 50)


logins_violin <- rbind(login_s20_violin,
                       login_f20_violin,
                       login_s19_violin,
                       login_f19_violin)

logins_violin$week <- factor(logins_violin$week,
                             levels = c("Before Week 14",
                                        "After Week 14"))

plot_8 <- ggplot(logins_violin, aes(x = week, y=elapsed_hour,
                             fill = professor)) +
  geom_split_violin() +
  facet_wrap(~semester) +
  labs(x = "Week",
       y = "Average elapsed time \nfor each student login (hours)") +
  theme_bw()

ggsave(plot = plot_8, filename = "img8.png",
       width = 6, height = 4)
```

```
###  violin plot 2019 spring
login_s19_p_violin <- login_s19_p %>%
  mutate(professor = "Prof. P",
         elapsed_hour = elapsed_min/60)
login_s19_h_violin <- login_s19_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60)
login_s19_violin <- rbind(login_s19_p_violin,
                          login_s19_h_violin) %>%
  mutate(week = ifelse(date <= min(login_s19_p$date) + 7*11,
                       "Week 03-13",
                       "Week 14-19")) %>%
  filter(elapsed_hour <= 100)

ggplot(login_s19_violin, aes(x = week, y=elapsed_hour)) +
  geom_violin() +
  facet_wrap(~professor) +
  labs(x = "Week",
       y = "Average elapsed time \nfor each student login (hours)") +
  theme_bw()
```

```
###  violin plot 2019 spring
login_s19_p_violin <- login_s19_p %>%
  mutate(professor = "Prof. P",
         elapsed_hour = elapsed_min/60)
login_s19_h_violin <- login_s19_h %>%
  mutate(professor = "Prof. H",
         elapsed_hour = elapsed_min/60)
login_s19_violin <- rbind(login_s19_p_violin,
                          login_s19_h_violin) %>%
  mutate(week = ifelse(date <= min(login_s19_p$date) + 7*11,
                       "Week 03-13",
                       "Week 14-19")) %>%
  filter(elapsed_hour <= 100)

ggplot(login_s19_violin, aes(x = week, y=elapsed_hour)) +
  geom_violin() +
  facet_wrap(~professor) +
  labs(x = "Before/after pandesmic",
       y = "Average elapsed time \nfor each student login (hours)") +
  theme_bw()
```

```
dharde19_before_plot <- dharde19_before %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Spring 2019") %>%
  mutate(week = "Before Week 14")

dharde19_after_plot <- dharde19_after %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Spring 2019") %>%
  mutate(week = "After Week 14")

dharde20_before_plot <- dharde20_before %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Spring 2020") %>%
  mutate(week = "Before Week 14")

dharde20_after_plot <- dharde20_after %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Spring 2020") %>%
  mutate(week = "After Week 14")

dharde19f_before_plot <- dharde19f_before %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Fall 2019") %>%
  mutate(week = "Before Week 14")

dharde19f_after_plot <- dharde19f_after %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Fall 2019") %>%
  mutate(week = "After Week 14")

dharde20f_before_plot <- dharde20fall_before %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Fall 2020") %>%
  mutate(week = "Before Week 14")

dharde20f_after_plot <- dharde20fall_after %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. H") %>%
  mutate(semester = "Fall 2020") %>%
  mutate(week = "After Week 14")

login_plot_h <- rbind(dharde19_before_plot,
                      dharde19_after_plot,
                      dharde19f_before_plot,
                      dharde19f_after_plot,
                      dharde20_before_plot,
                      dharde20_after_plot,
                      dharde20f_before_plot,
                      dharde20f_after_plot)

bpidgeonn19_before_plot <- bpidgeonn19_before %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Spring 2019") %>%
  mutate(week = "Before Week 14")

bpidgeonn19_after_plot <- bpidgeonn19_after %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Spring 2019") %>%
  mutate(week = "After Week 14")

bpidgeonn19f_before_plot <- bpidgeonn19f_before %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Fall 2019") %>%
  mutate(week = "Before Week 14")

bpidgeonn19f_after_plot <- bpidgeonn19f_after %>%
  select (user_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Fall 2019") %>%
  mutate(week = "After Week 14")

bpidgeonn20_before_plot <- bpidgeonn20_before %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Spring 2020") %>%
  mutate(week = "Before Week 14")

bpidgeonn20_after_plot <- bpidgeonn20_after %>%
  select (user_id = student_id, week_logins) %>%
  mutate(professor = "Prof. P") %>%
  mutate(semester = "Spring 2020") %>%
  mutate(week = "After Week 14")
```

```
login_plot_p <- rbind(bpidgeonn19_after_plot,
                      bpidgeonn19_before_plot,
                      bpidgeonn19f_after_plot,
                      bpidgeonn19f_before_plot,
                      bpidgeonn20_before_plot,
                      bpidgeonn20_after_plot)

activity_violin <- rbind(login_plot_p, login_plot_h)

activity_violin$week <- factor(activity_violin$week,
                             levels = c("Before Week 14",
                                        "After Week 14"))

activity_violin <- activity_violin %>%
  filter(week_logins <= 10)
plot_9 <- ggplot(activity_violin, aes(x = week, y=week_logins,
                             fill = professor)) +
  geom_split_violin() +
  facet_wrap(~semester) +
  labs(x = "Week",
       y = "Average login per week \nfor each student") +
  theme_bw() +
  scale_y_continuous(breaks = seq(0, 10, by = 2),
                     labels = seq(0, 10, by = 2))

ggsave(plot = plot_9, filename = "img9.png",
       width = 6, height = 4)
```

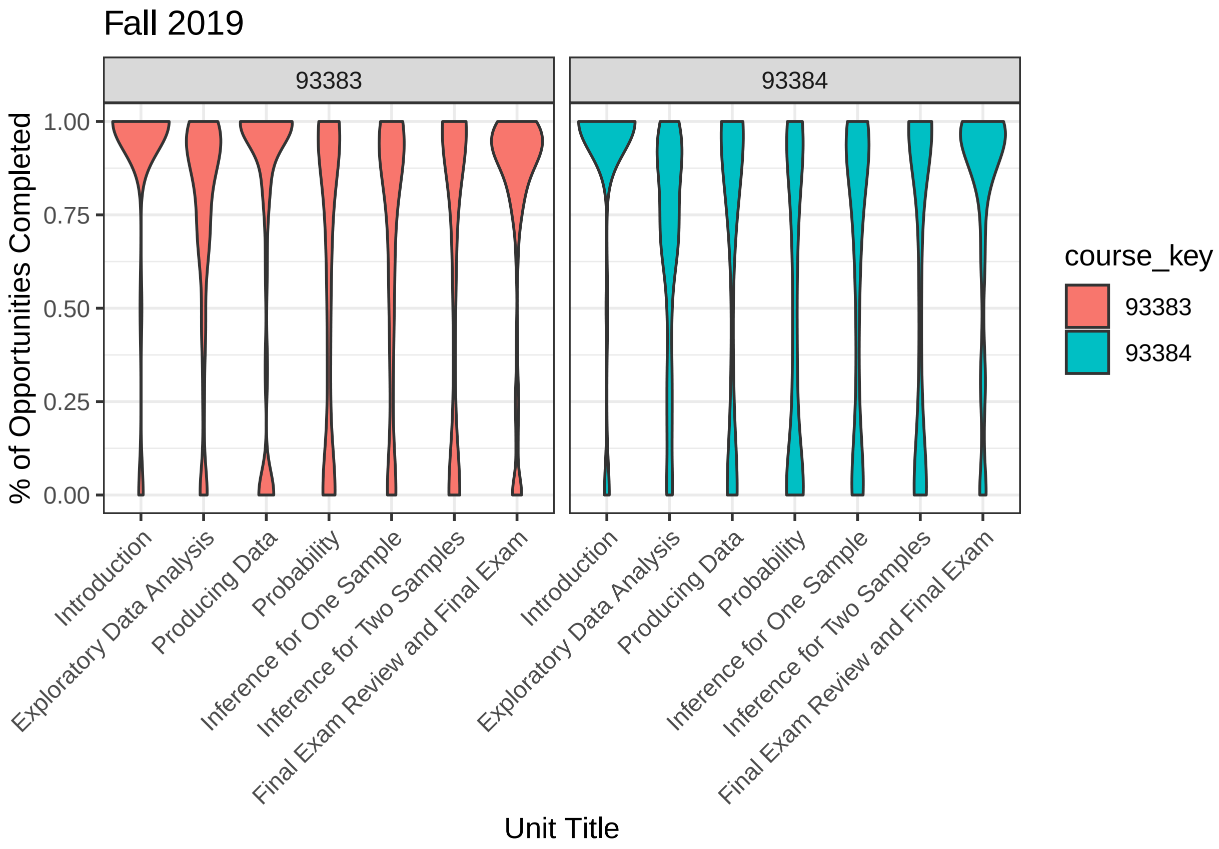
```
ggplot(activity_violin, aes(x = week, y=week_logins,
                             fill = professor)) +
  geom_split_violin() +
  facet_wrap(~semester) +
  labs(x = "Week",
       y = "Average login per week \nfor each student") +
  theme_bw() +
  scale_y_continuous(breaks = seq(0, 10, by = 2),
                     labels = seq(0, 10, by = 2))
```

Appendix 3.2

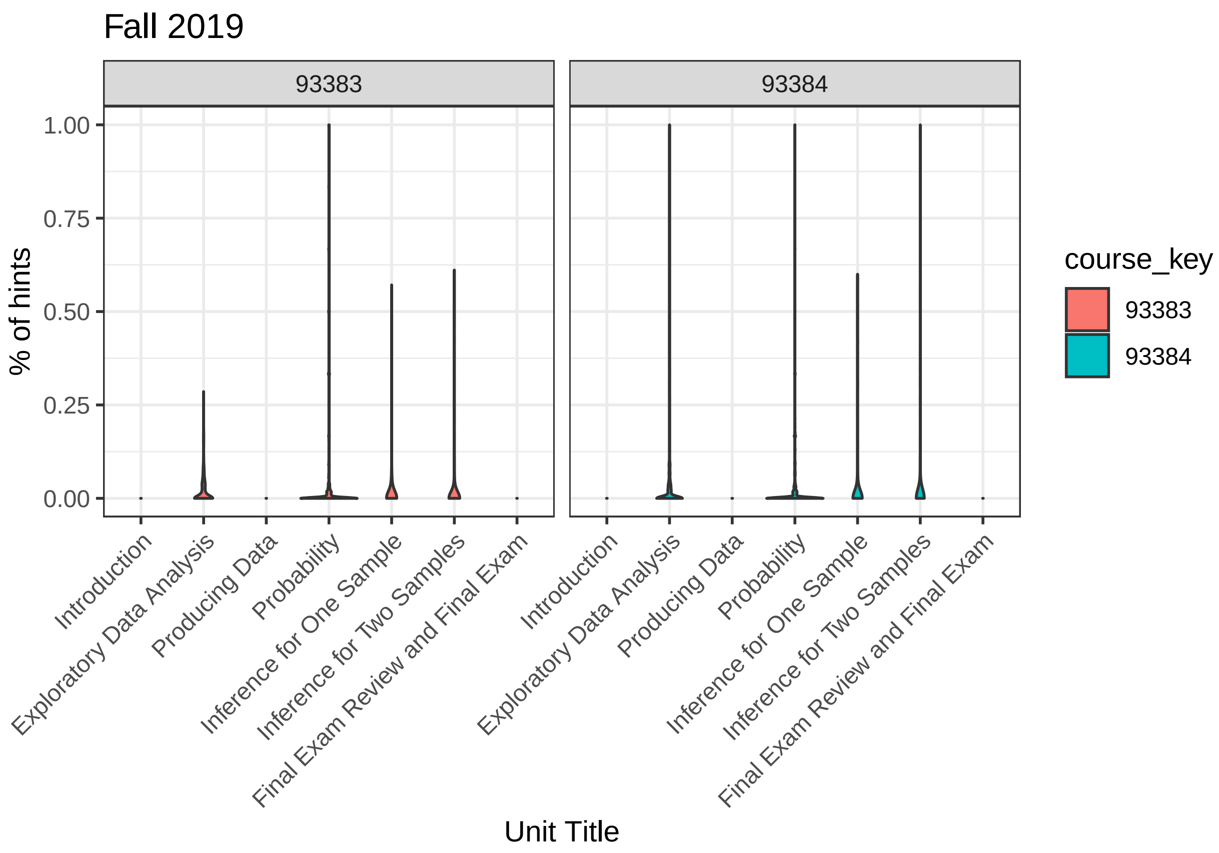
```r
library(lubridate)
library(tidyverse)
```

```r
setwd("~/CMU Coursework/Spring/726 Capstone/OLI/data/Export/f19")
summary_module_f19 <- read.csv("summary_module.csv", header = T)
summary_unit_f19 <- read.csv("summary_unit.csv", header = T)
instructor_f19 <- read.csv("instructor_action.csv", header = T)
unit_f19 <- read.csv("units.csv", header = T)
summary_unit_f19_joined <- left_join(summary_unit_f19, unit_f19, by = "unit_id")

setwd("~/CMU Coursework/Spring/726 Capstone/OLI/data/Export/f20")
summary_module_f20 <- read.csv("summary_module.csv", header = T)
summary_unit_f20 <- read.csv("summary_unit.csv", header = T)
course_f20 <- read.csv("course.csv", header = T)
instructor_2020 <- read.csv("instructor_action_all.csv", header = T)

course_f20$start_date <- mdy_hm(course_f20$start_date)
course_f20$end_date <- mdy_hm(course_f20$end_date)
course_f20 <- course_f20 %>%
  mutate(semester = ifelse(month(start_date) == 8, "Fall",
                          ifelse((month(start_date) == 1 & day(start_date) == 8),
                                 "Spring", NA)))
course_instructor_2020 <- full_join(instructor_2020, course_f20, by = "course_key")
unit_f20 <- read.csv("unit.csv", header = T)
summary_unit_f20_joined <- full_join(summary_unit_f20, course_f20, by = "course_key")


unit_id_order <- unit_f19$unit_title
summary_unit_f19_joined$unit_title <- factor(summary_unit_f19_joined$unit_title,
                                             levels = unit_id_order)

summary_unit_f20_joined <- full_join(summary_unit_f20_joined, unit_f20, by = "unit_id")
unit_id_order_20 <- unit_f20 %>%
  distinct(unit_title)
summary_unit_f20_joined$unit_title <- factor(summary_unit_f20_joined$unit_title,
                                             levels = unlist(unit_id_order_20))
```

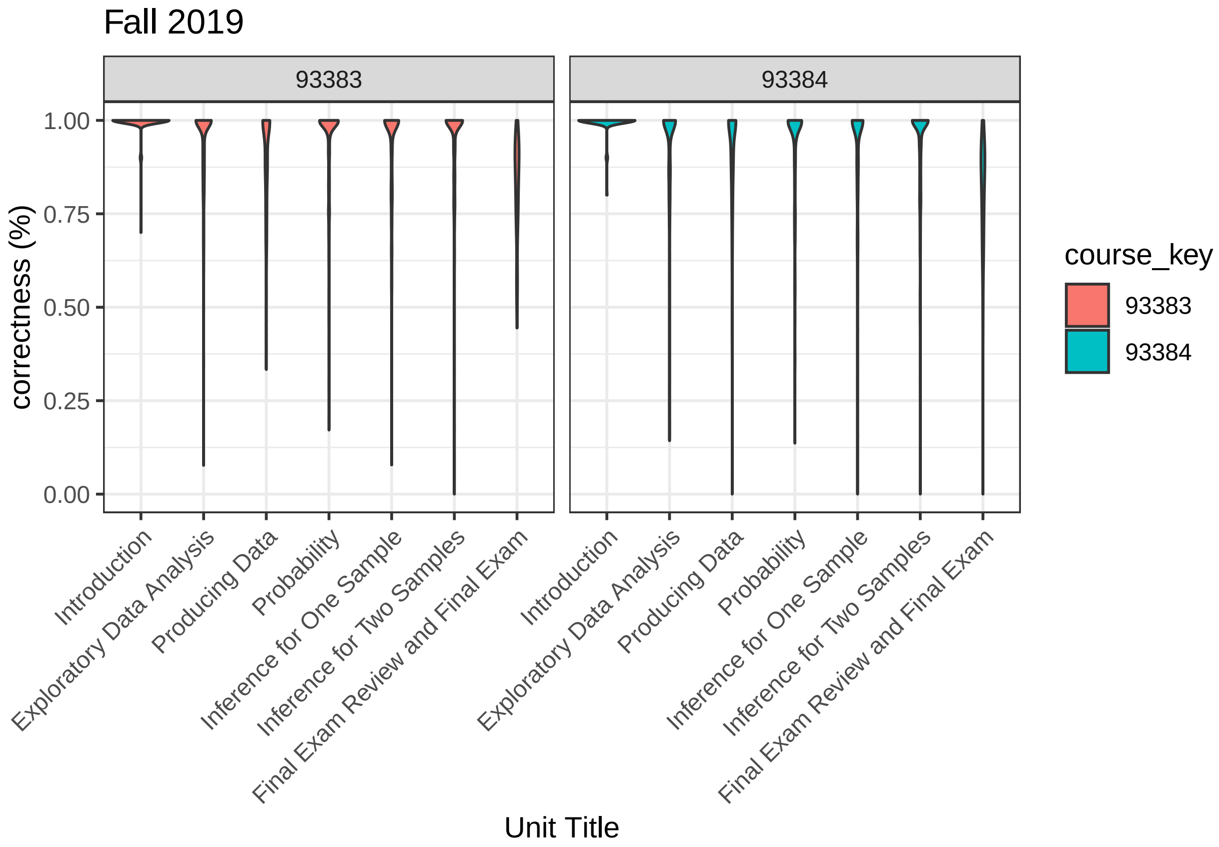
```r
# Fall 2019
# prof. H
course_key_harden_f19 <- c(93383, 93384)

# % opportunities completed
summary_unit_f19_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct, fill = as.factor(course_key))) +
  geom_violin() +
  theme_bw() +
  facet_wrap(~course_key) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(limits = c(0:1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       fill = "course_key")
```



```
# % hints
summary_unit_f19_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct, fill = as.factor(course_key))) +
  geom_violin() +
  theme_bw() +
  facet_wrap(~course_key) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of hints",
       fill = "course_key")
```



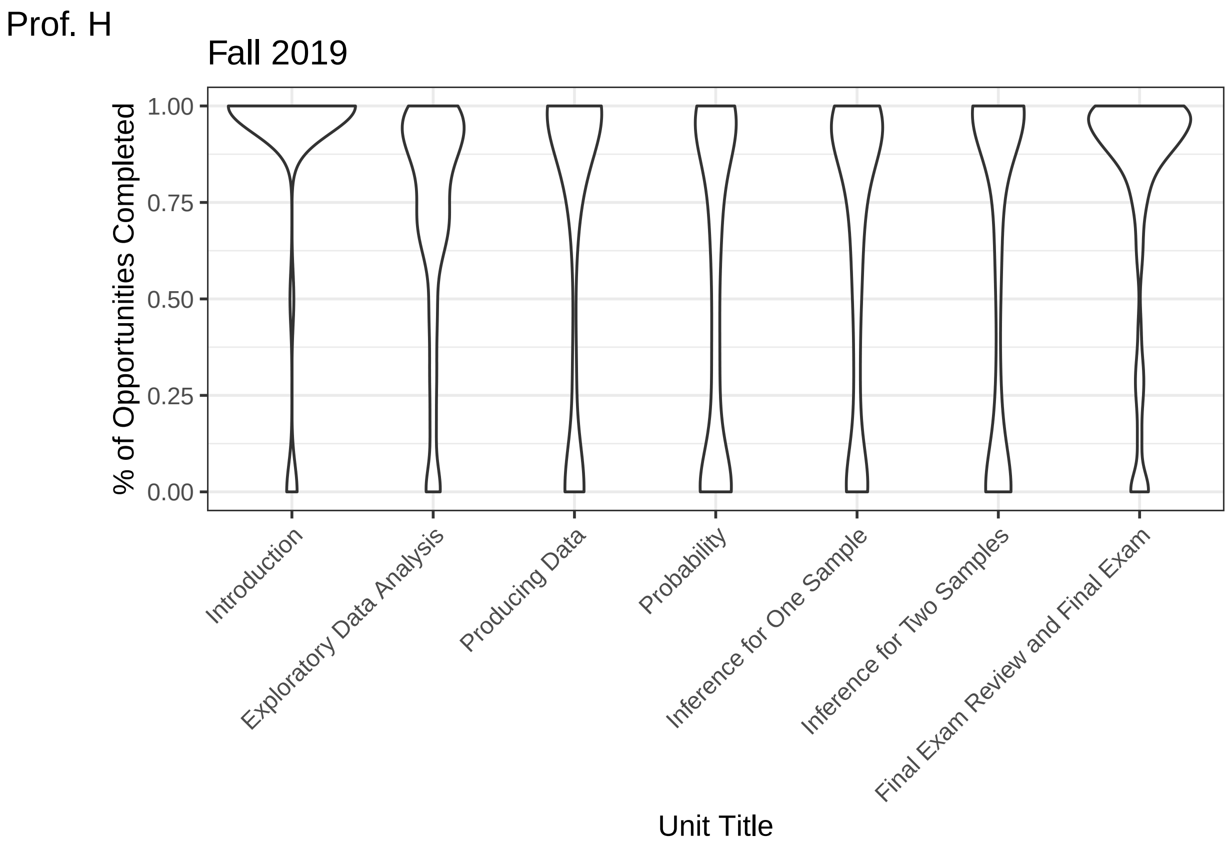
```r
# % correct
summary_unit_f19_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct, fill = as.factor(course_key))) +
  geom_violin() +
  theme_bw() +
  facet_wrap(~course_key) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "correctness (%)",
       fill = "course_key")
```



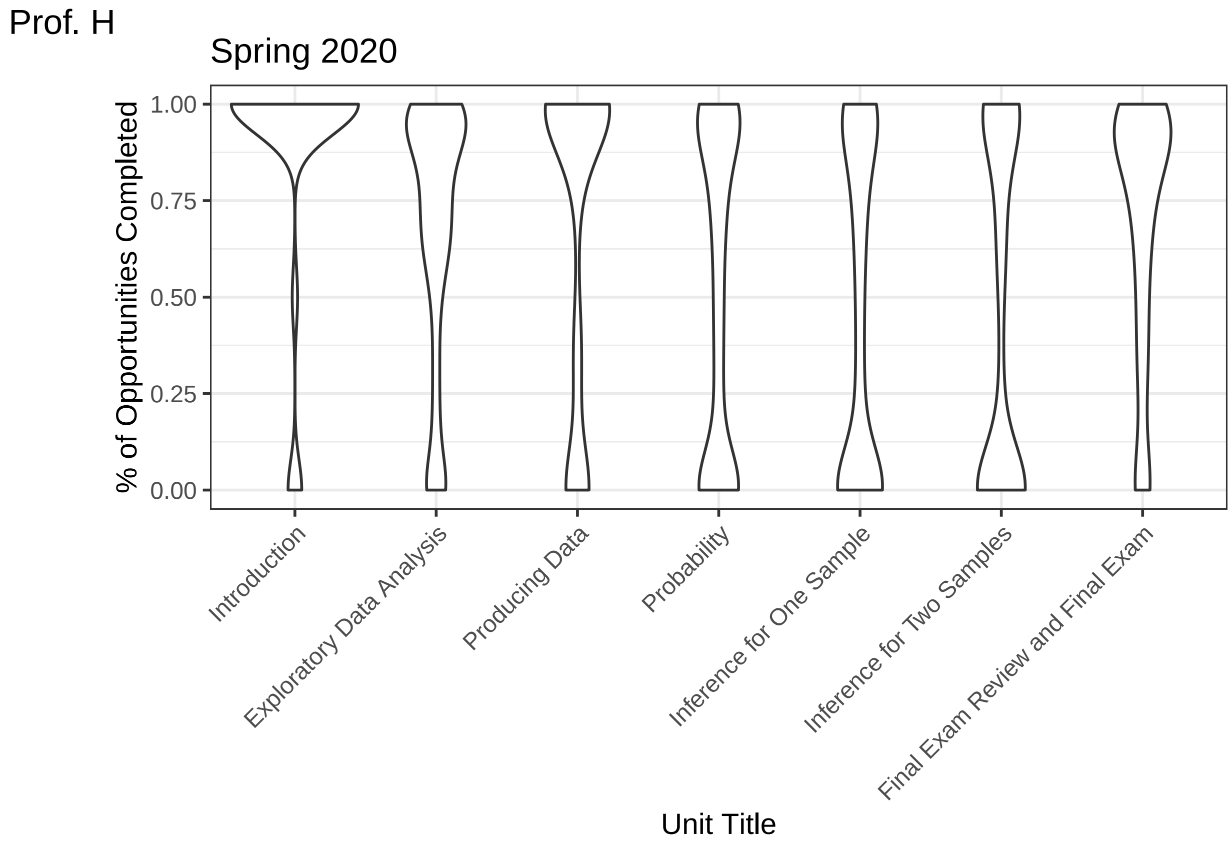
From the violin distributions for sections taught by Prof. H, the distributions of the 2 sections are similar enough by the naked eye to combine together into one distribution. The combined plot for the percentage of questions completed is shown in the report as the Fall 2019 plot in Figure 13.

Percentage of Unit Questions Completed

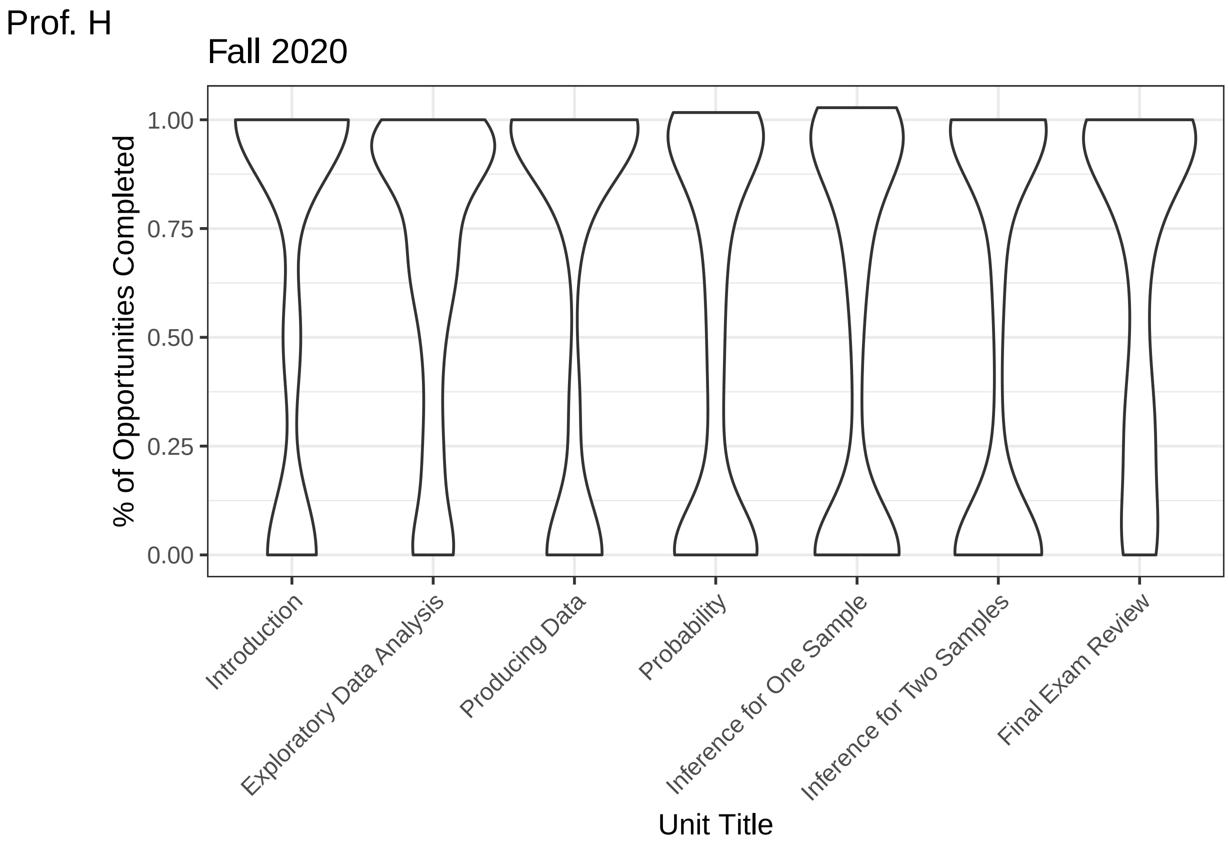
```
# Prof. H
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. H")
```



```
# Spring 2020
course_key_harden_s20 <- c(21312, 21315)
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_harden_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. H")
```



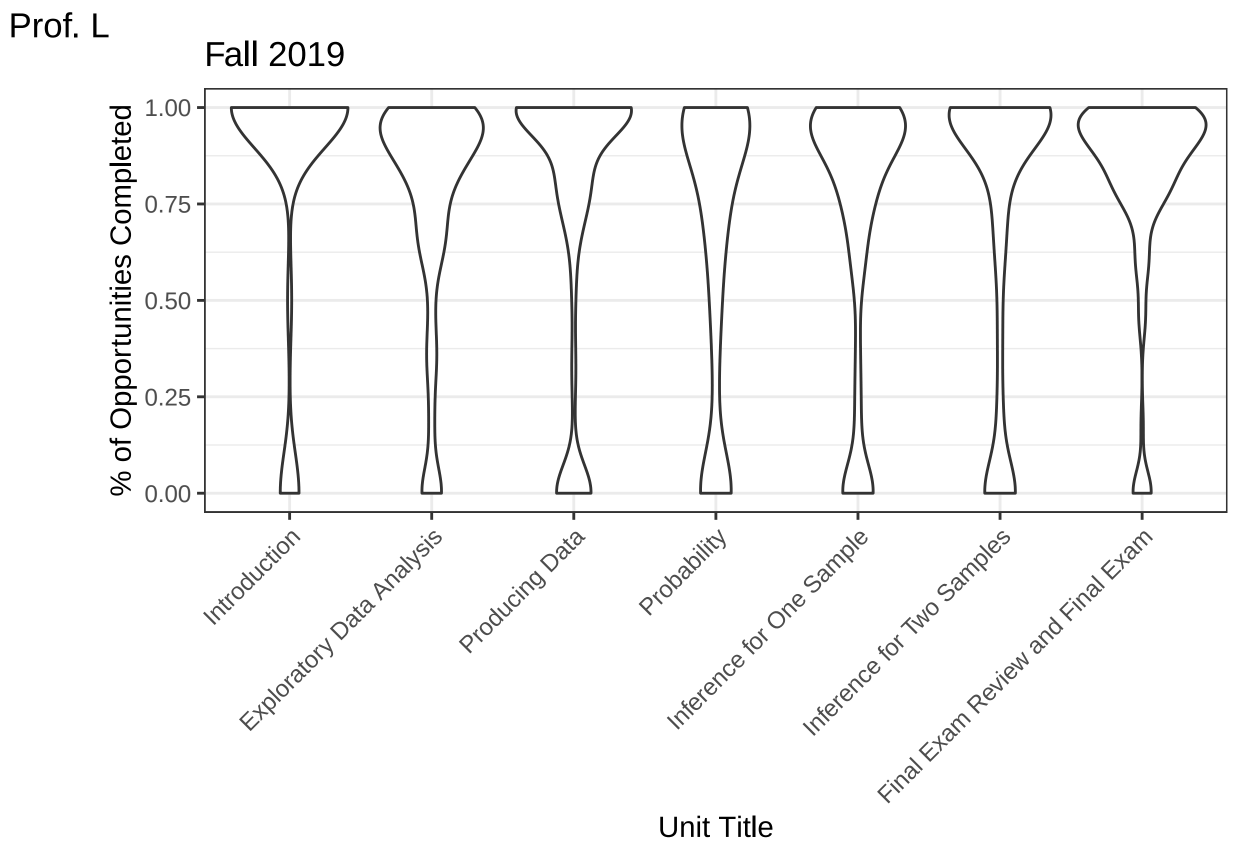
```
# Fall 2020
course_key_harden_f20 <- c(90984, 90985)
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_harden_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. H")
```



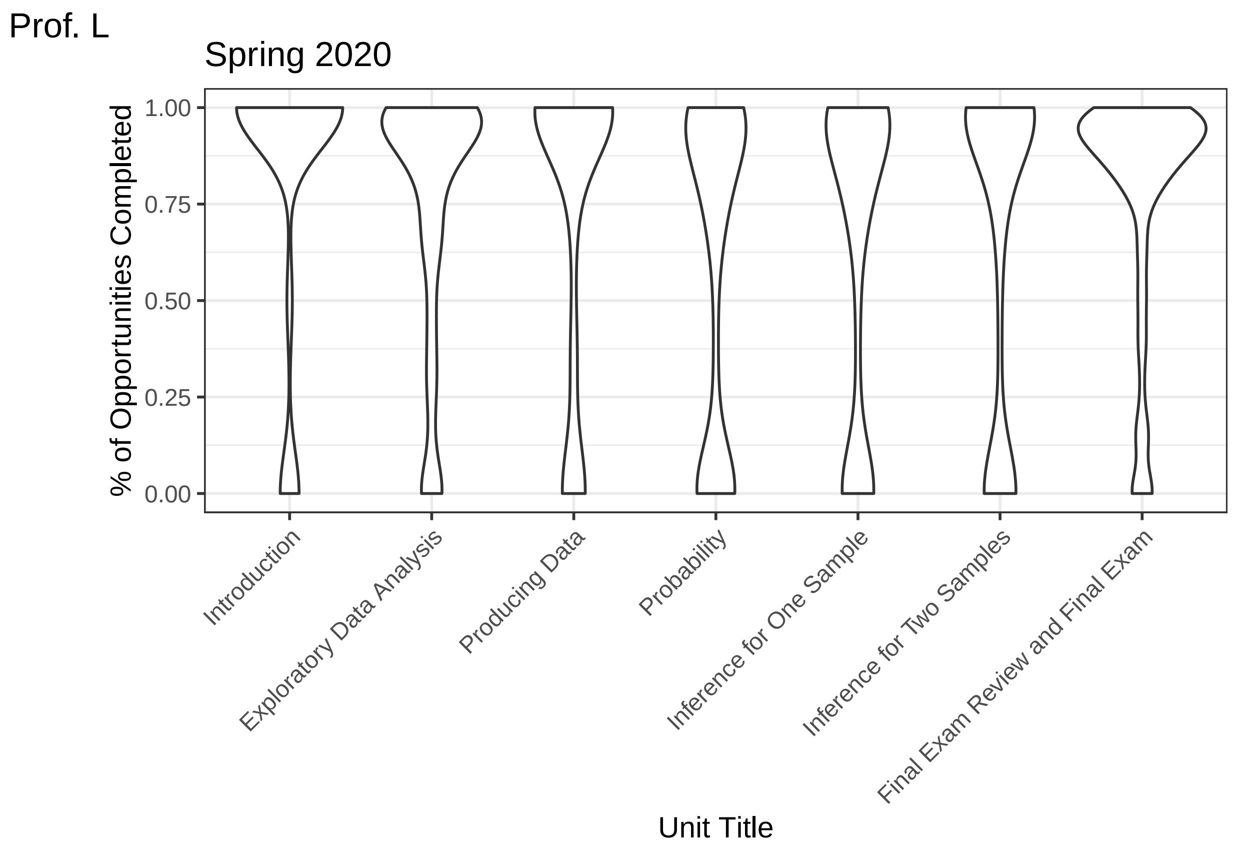
The plots are shown in the report as Figure 13. From the plot we can see that there are more students completing fewer unit questions in the Fall 2020 than the previous two semesters.

We will then plot the percentage of unit questions completed for sections taught by other instructors.

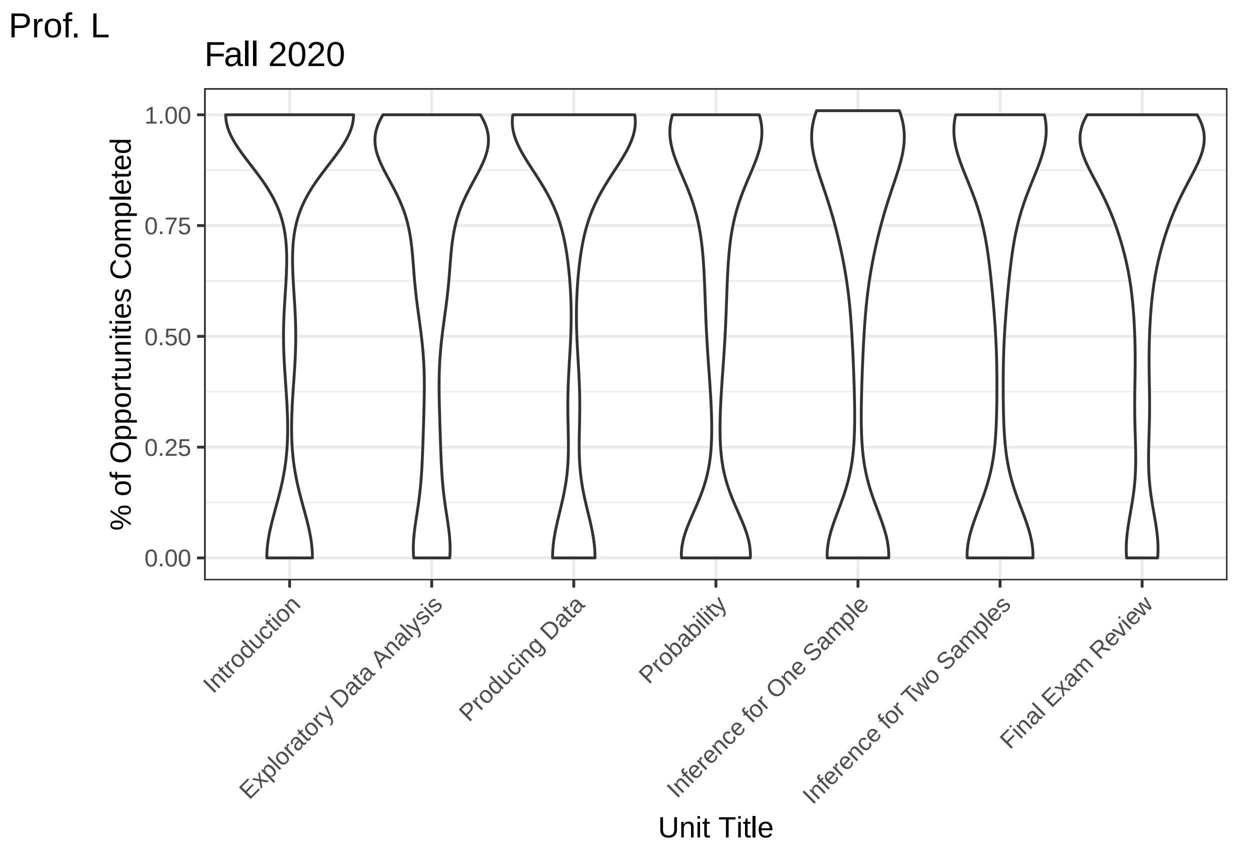
```
# Prof. L
course_key_liu_f19 <- c(93389)
course_key_liu_s20 <- c(21309)
course_key_liu_f20 <- c(90990, 90995)

# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_liu_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. L")
```



```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_liu_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. L")
```

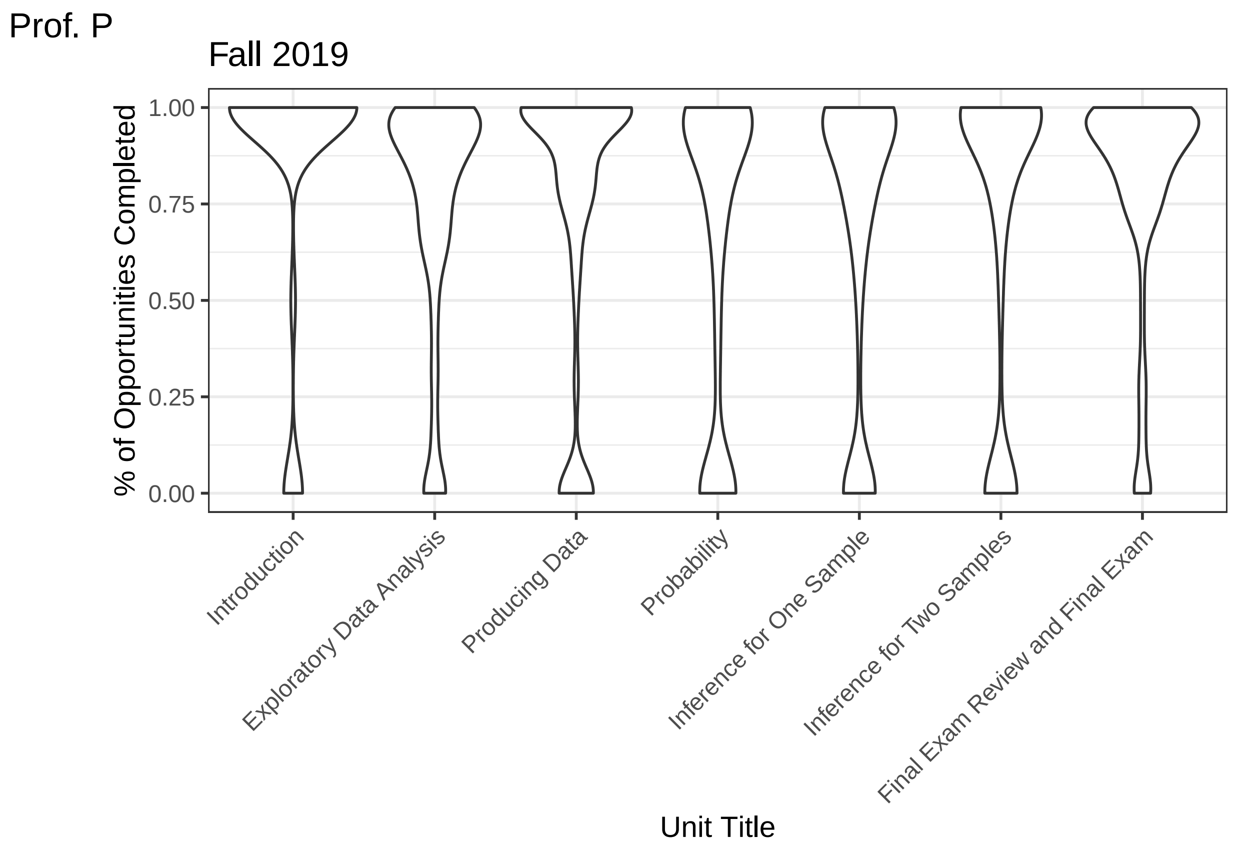


```r
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_liu_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. L")
```

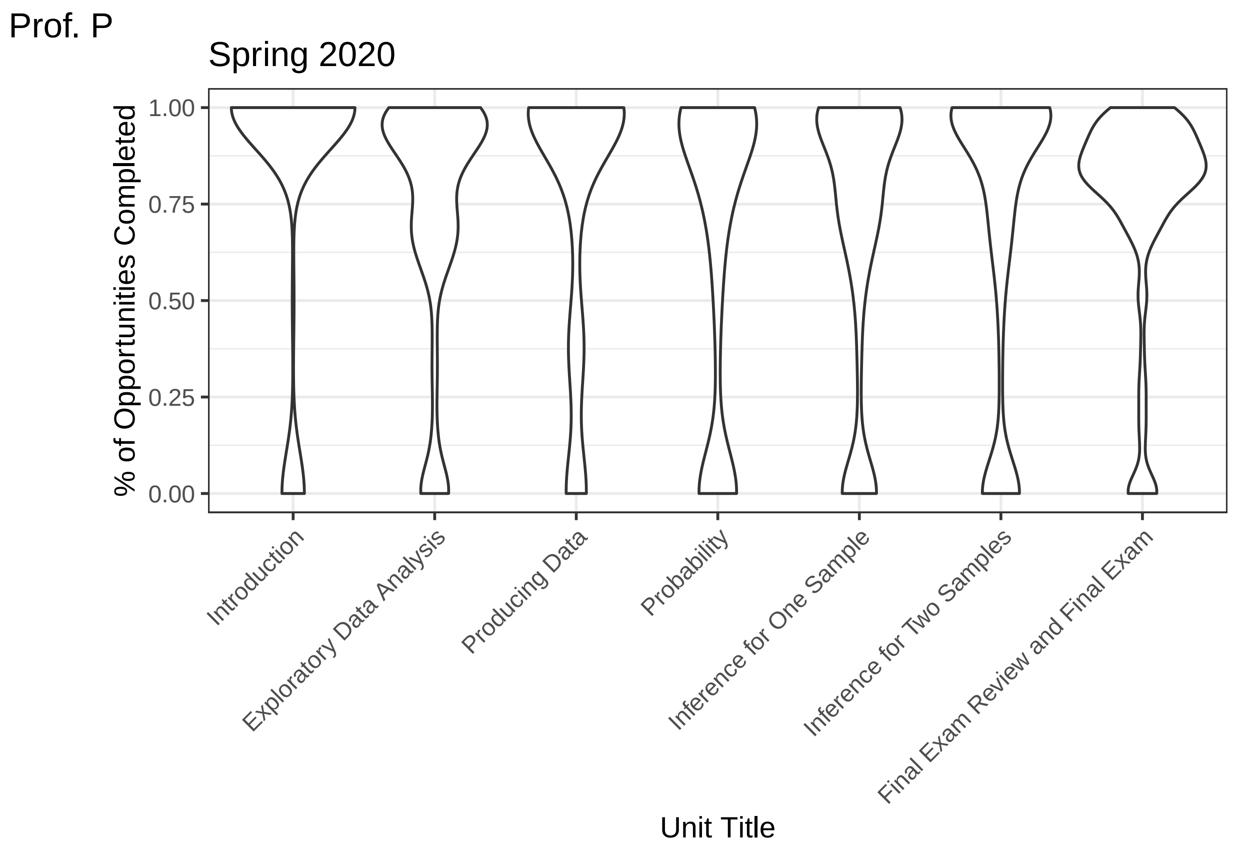


We can observe a similar pattern in the sections taught by Prof. L.

```
# Prof. P
course_key_bp_f19 <- c(93385, 93387)
course_key_bp_s20 <- c(21313)

# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_bp_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. P")
```

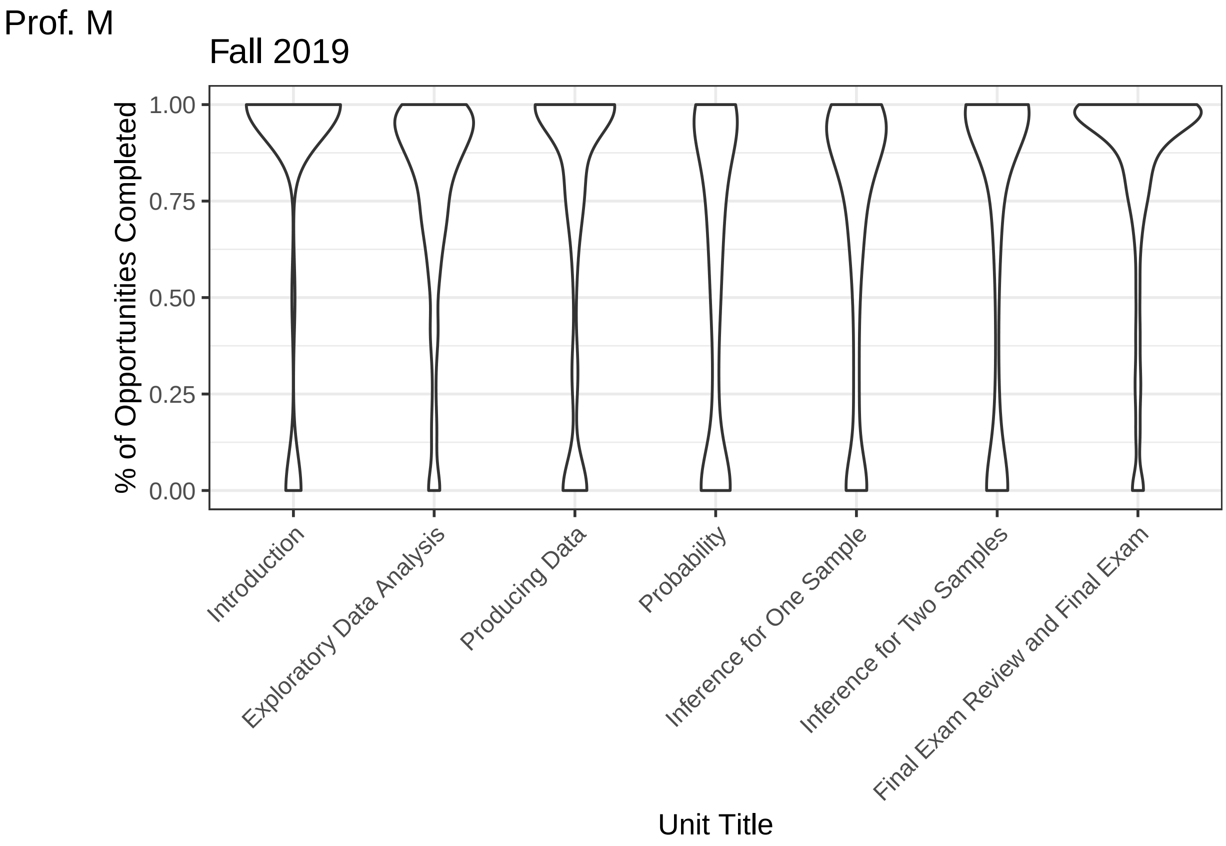


```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_bp_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. P")
```

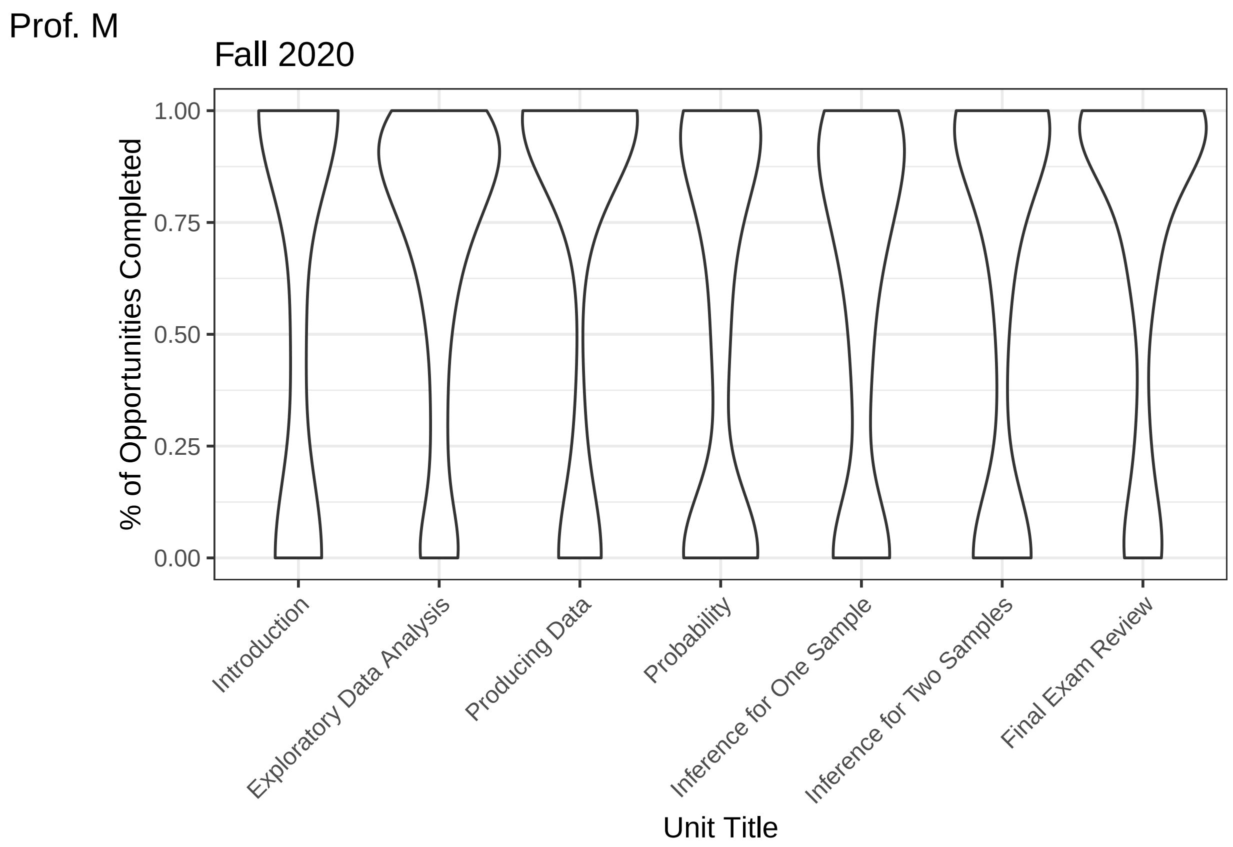


Prof. P did not teach in Fall 2020, and the student behavior for unit questions completion are very similar in Fall 2019 and Spring 2020, with slight increase of students completing fewer unit questions.

```
# Prof. M
course_key_meadows_f19 <- c(93391, 93394)
course_key_meadows_f20 <- c(90999)

# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_meadows_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. M")
```



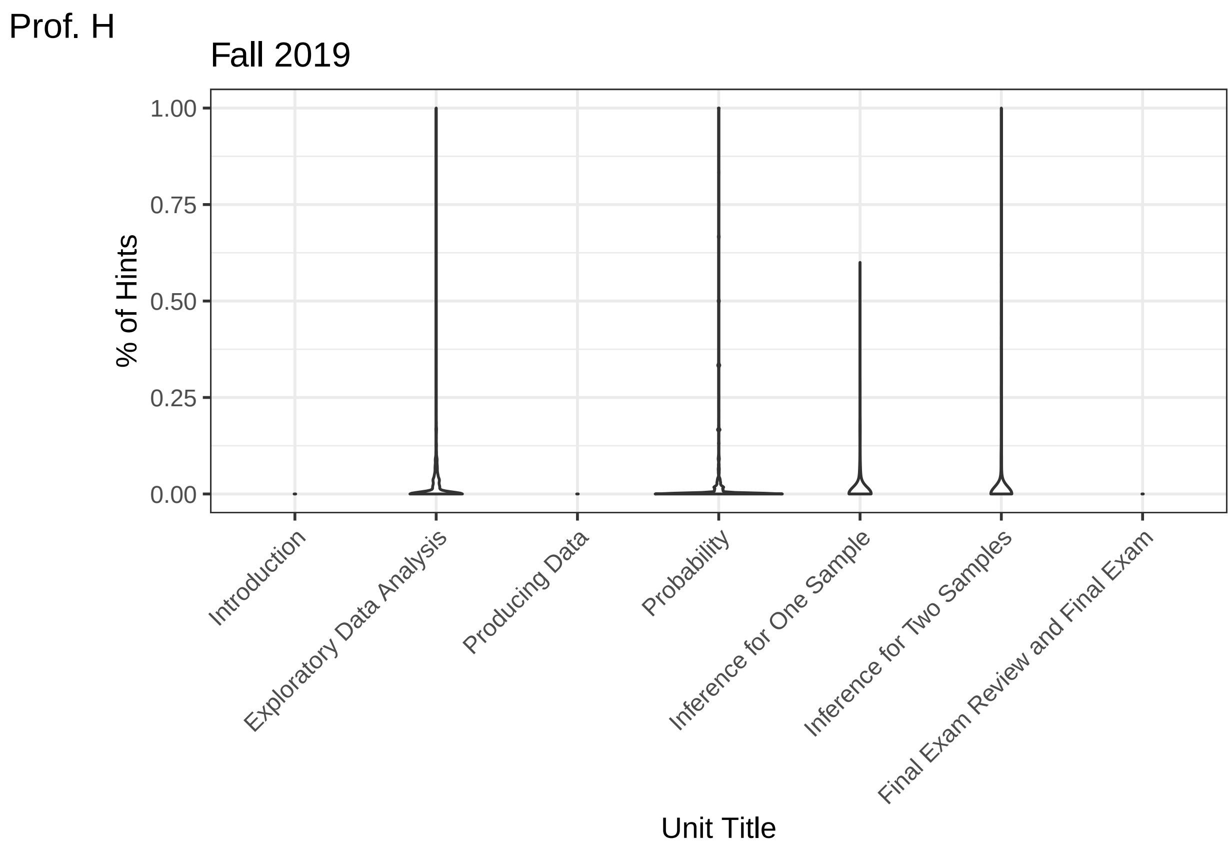
```r
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = practice/opportunities) %>%
  filter(course_key %in% course_key_meadows_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Opportunities Completed",
       tag = "Prof. M")
```



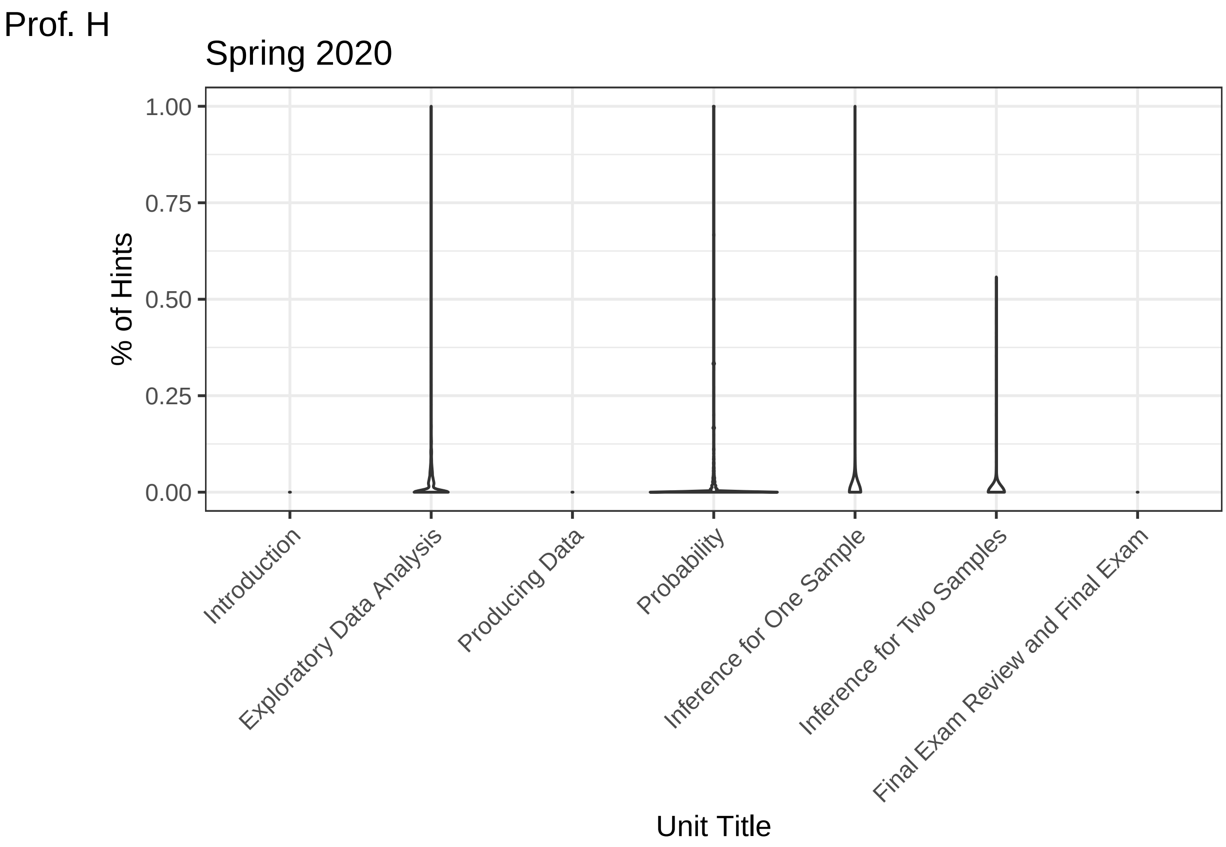
Prof. M taught in Fall 2019 and Fall 2020, not in Spring 2020. We can observe that more students are completing fewer unit questions in Fall 2020 than in Fall 2019, which may indicate students are becoming less engage in completing unit questions.

Percentage of Hints

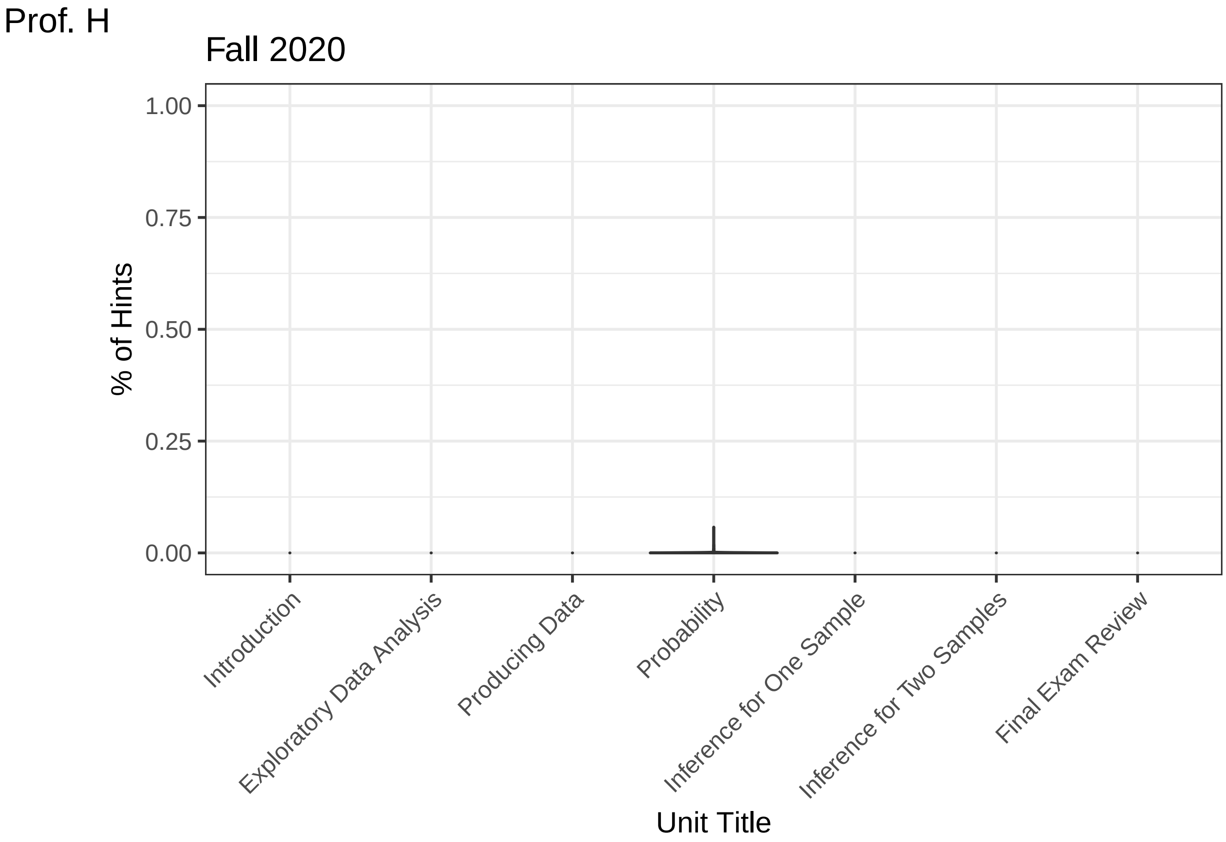
```r
# Prof. H
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. H")
```



```r
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_harden_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. H")
```



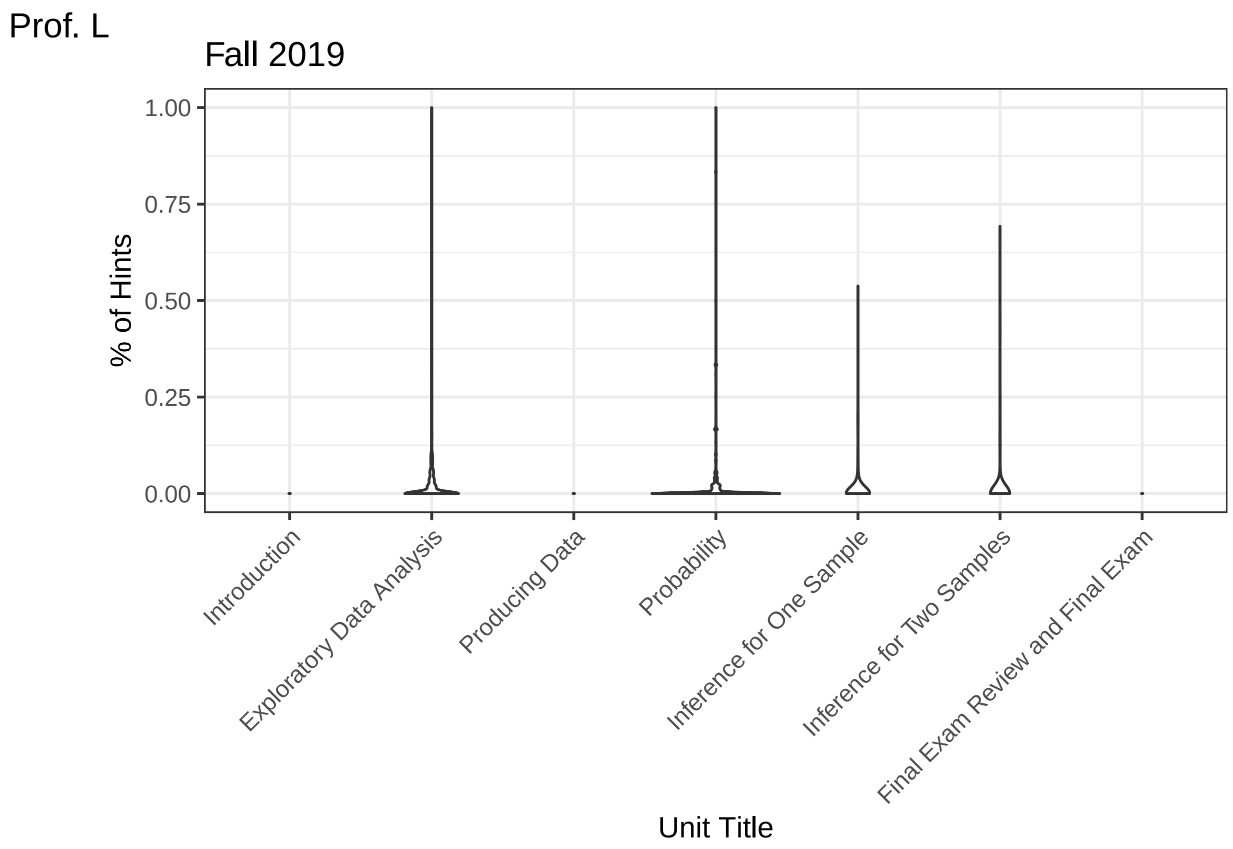
```
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_harden_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. H")
```



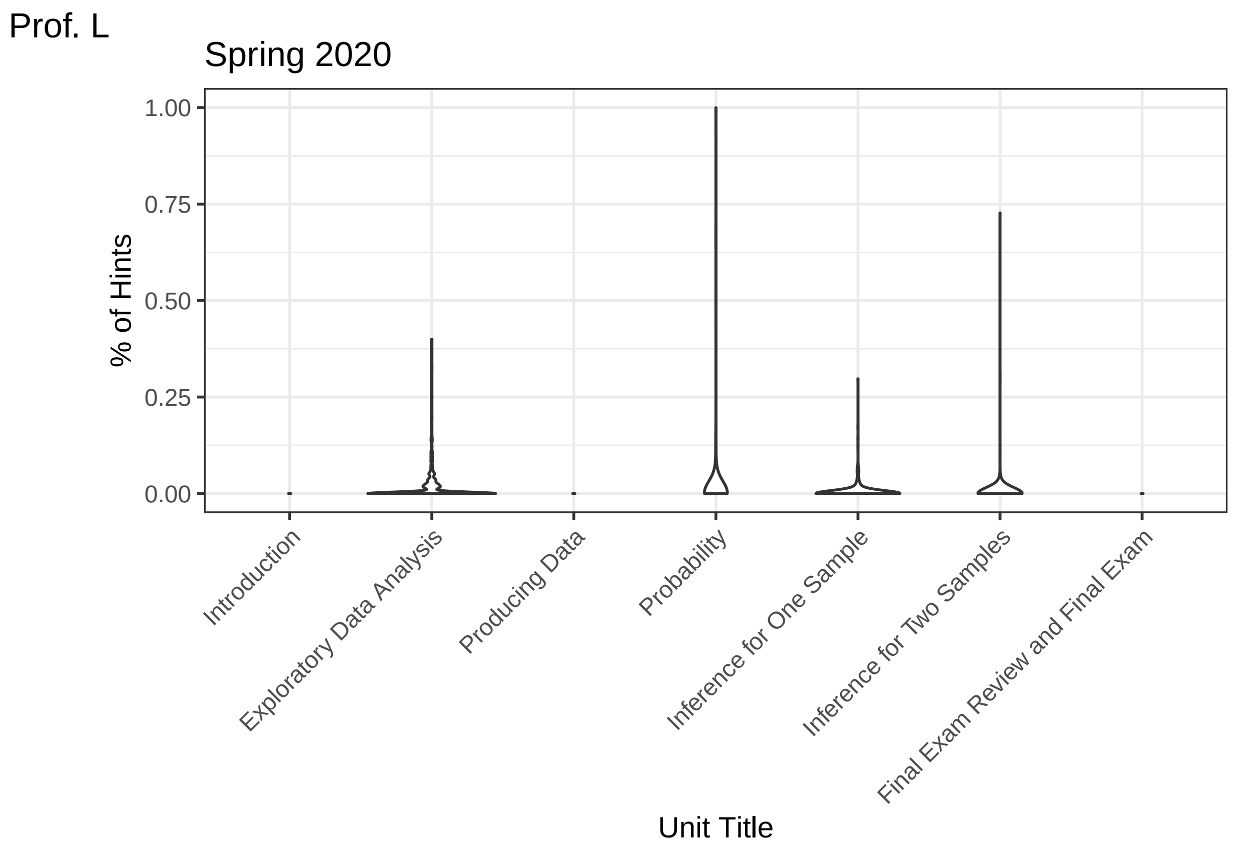
We can observe that students did not request any hints for any other units questions other than "Probability" in Fall 2020 in sections taught by Prof. H.

With the "Probability" unit, students are requesting fewer hints in Fall 2020 than the previous semesters.

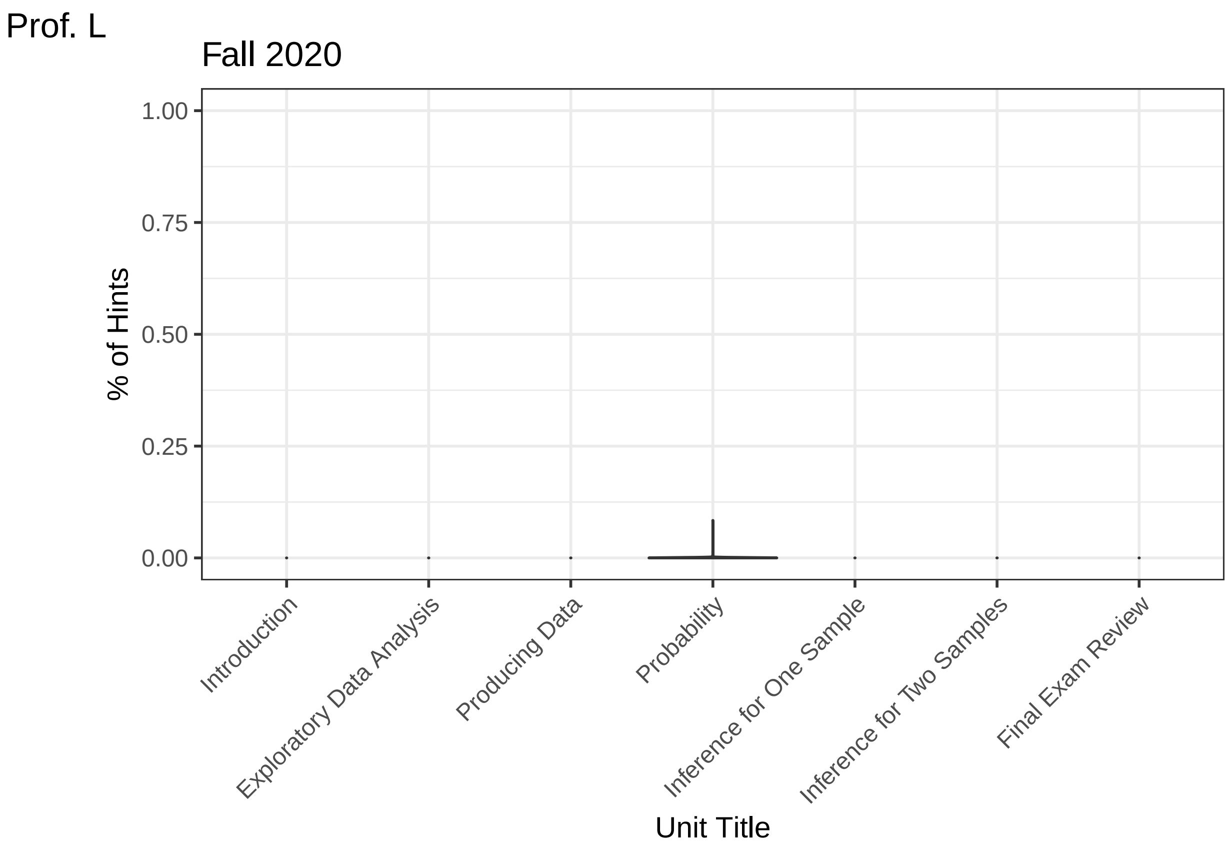
```r
# Prof. L
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_liu_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. L")
```



```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_liu_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. L")
```

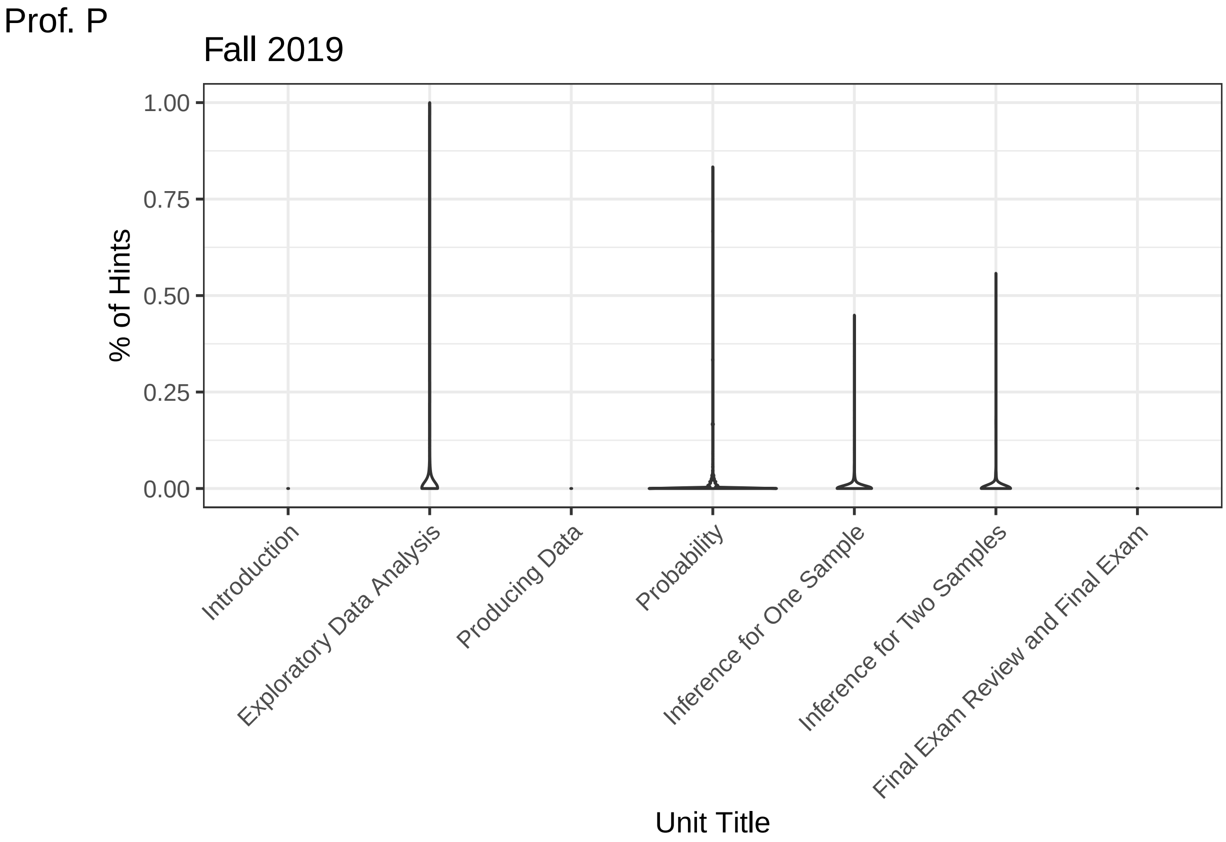


```r
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_liu_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. L")
```

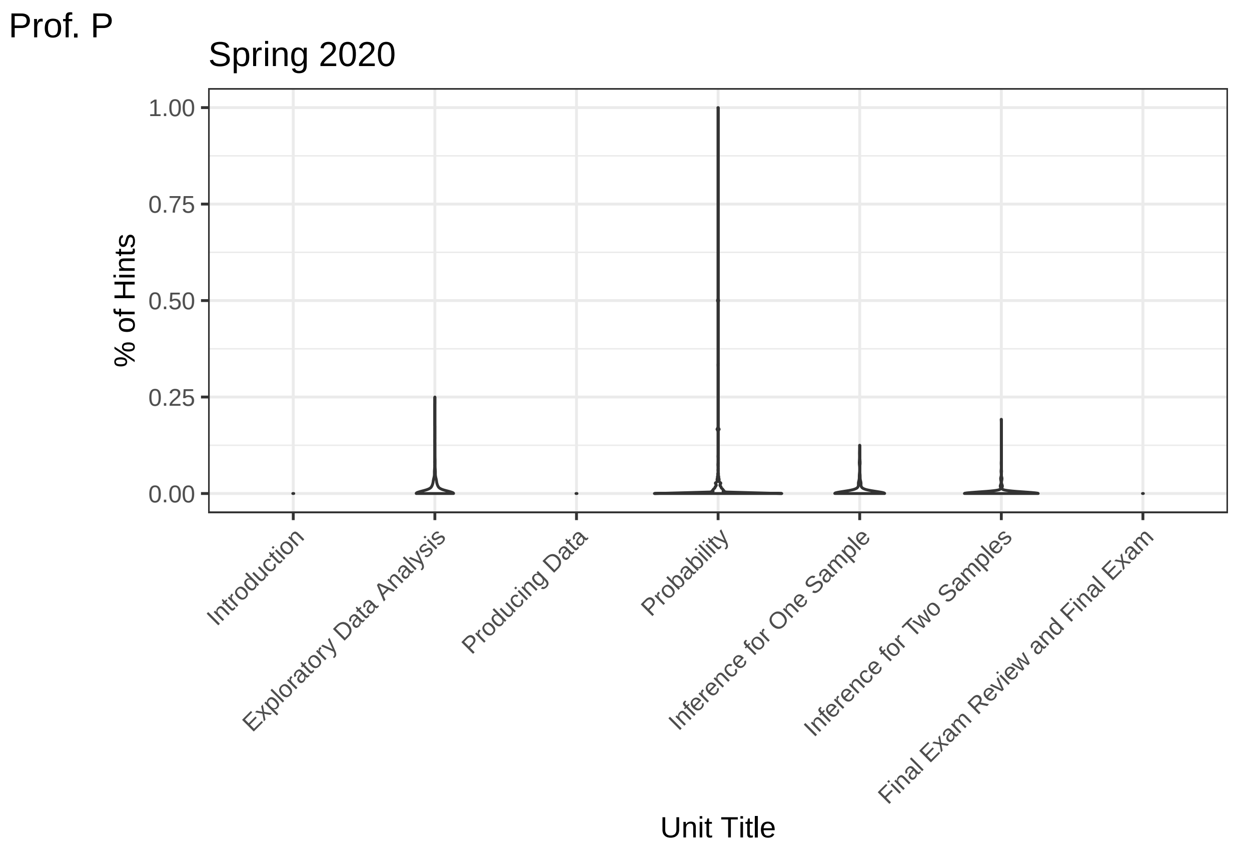


Similar patterns are observed in sections taught by Prof. L. There is no record of students requesting hints for any unit questions other than the unit questions from "Probability" in Fall 2020.

```r
# Prof. P
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_bp_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. P")
```

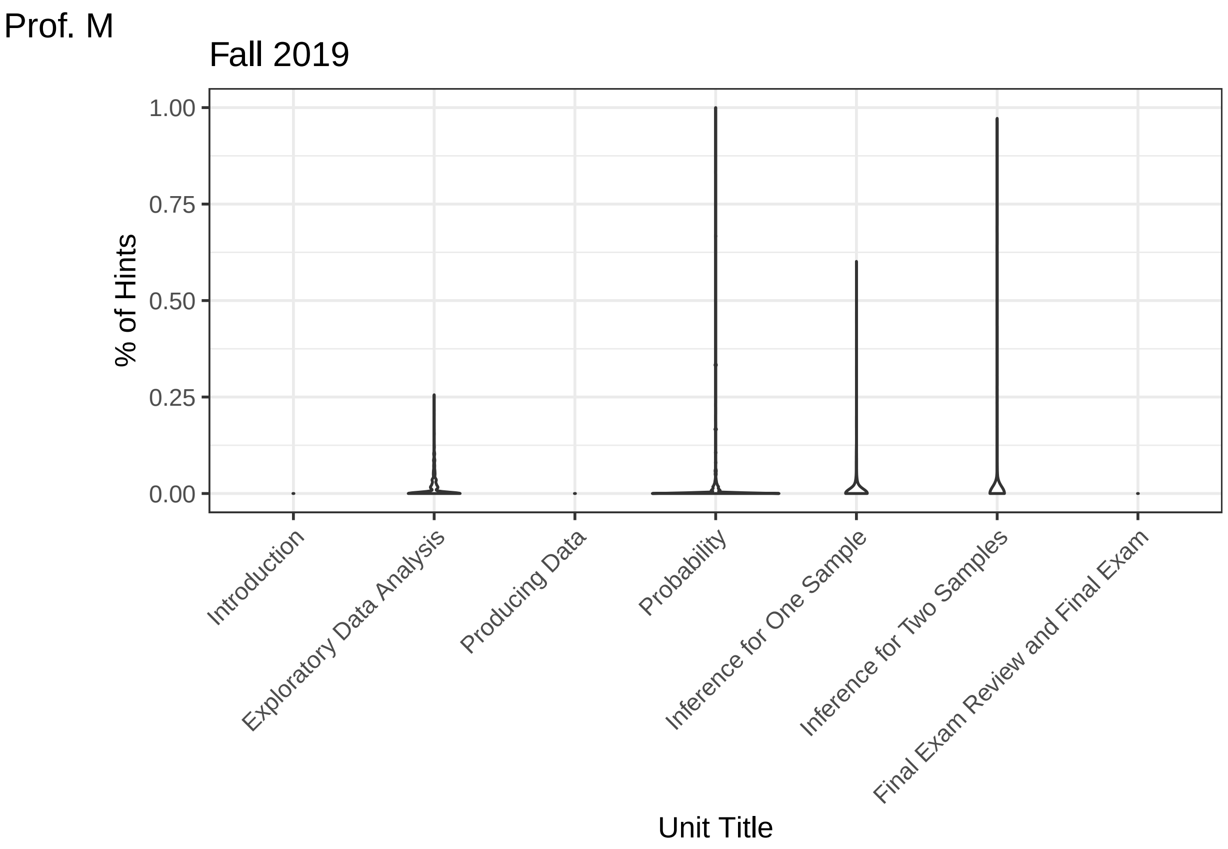


```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_bp_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. P")
```



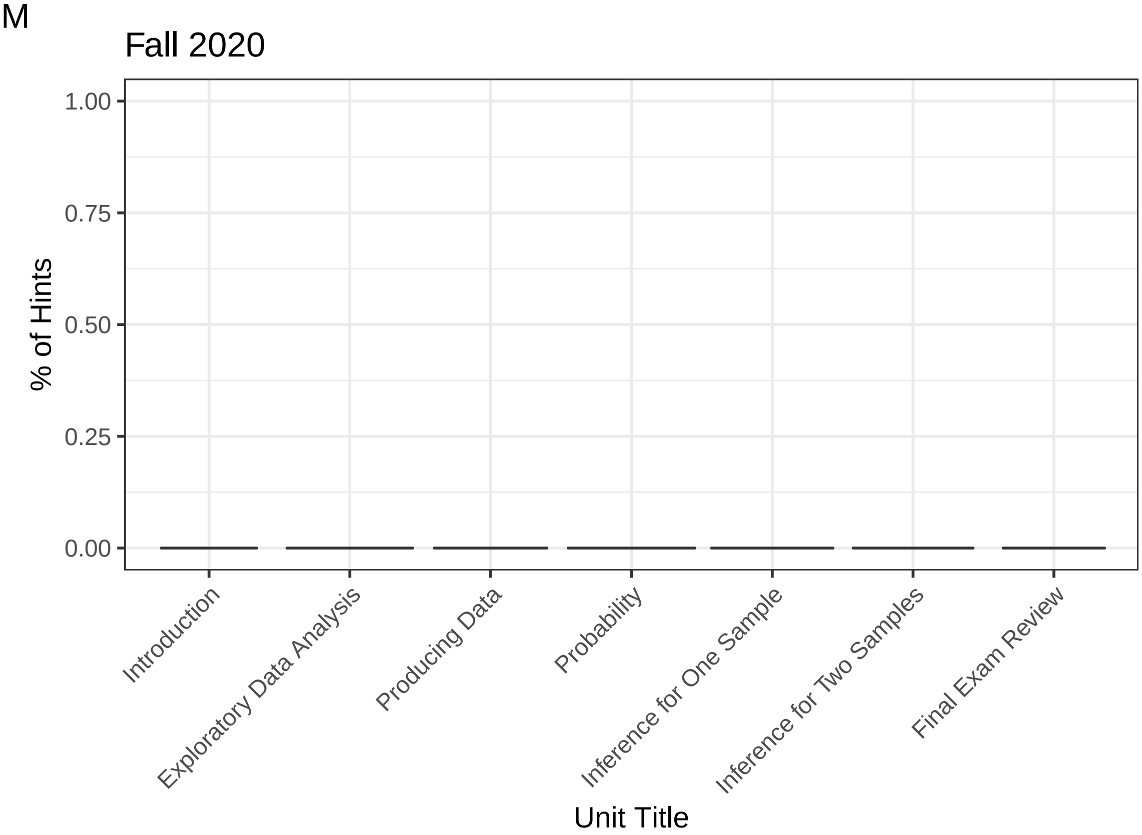
There are more students requesting more hints out of the unit questions they complete in Fall 2019 than in Spring 2020 in units other than “Probability” for sections taught by Prof. P.

```r
# Prof. M
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_meadows_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. M")
```



```
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = hints/practice) %>%
  filter(course_key %in% course_key_meadows_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "% of Hints",
       tag = "Prof. M")
```

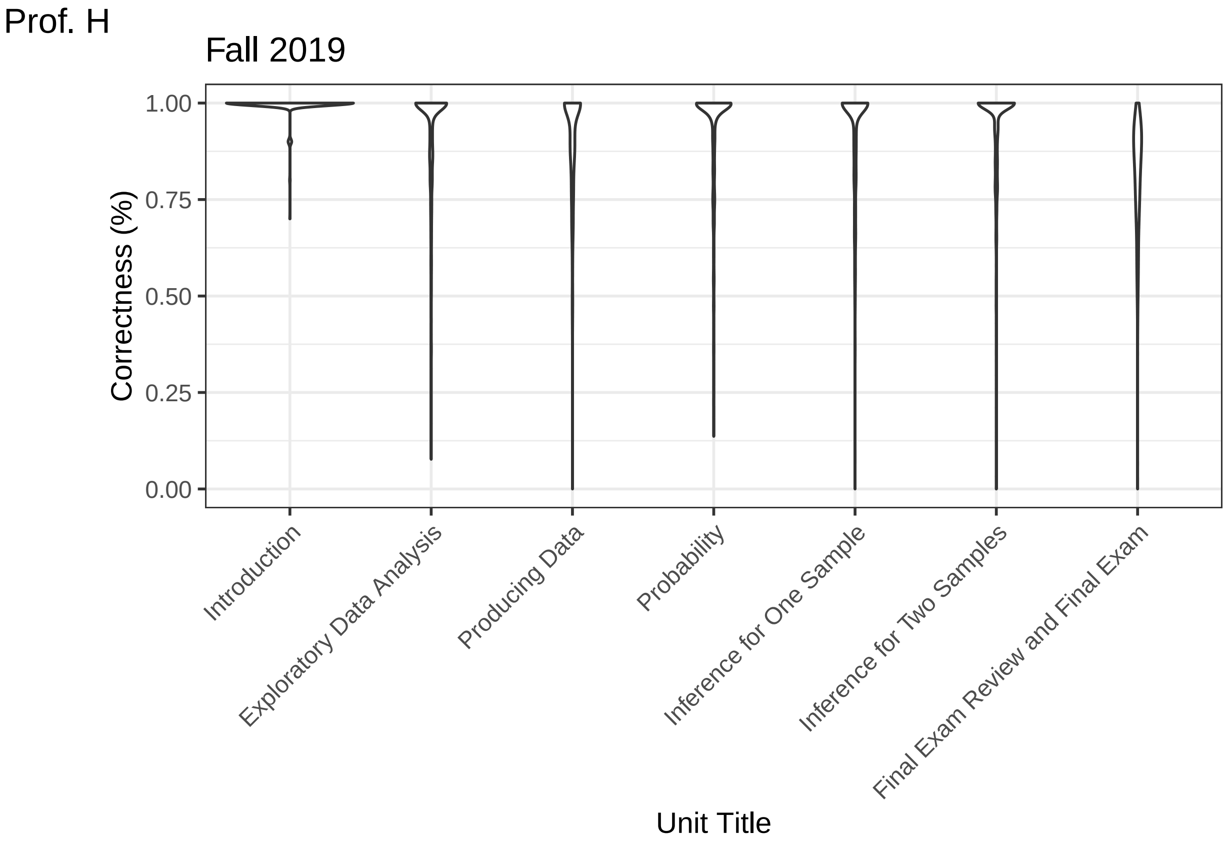
Prof. M



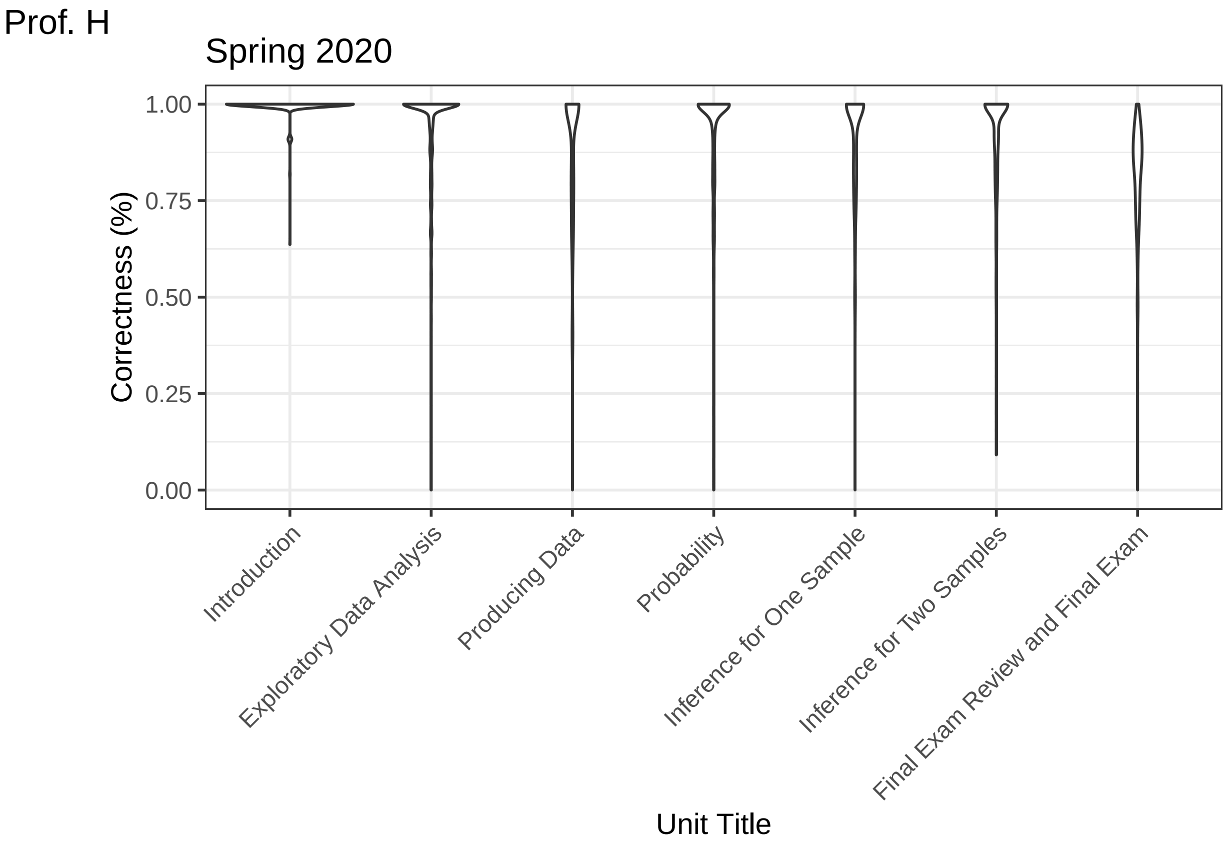
There is no records of students requesting hints in Fall 2020.

Percentage of Correctness

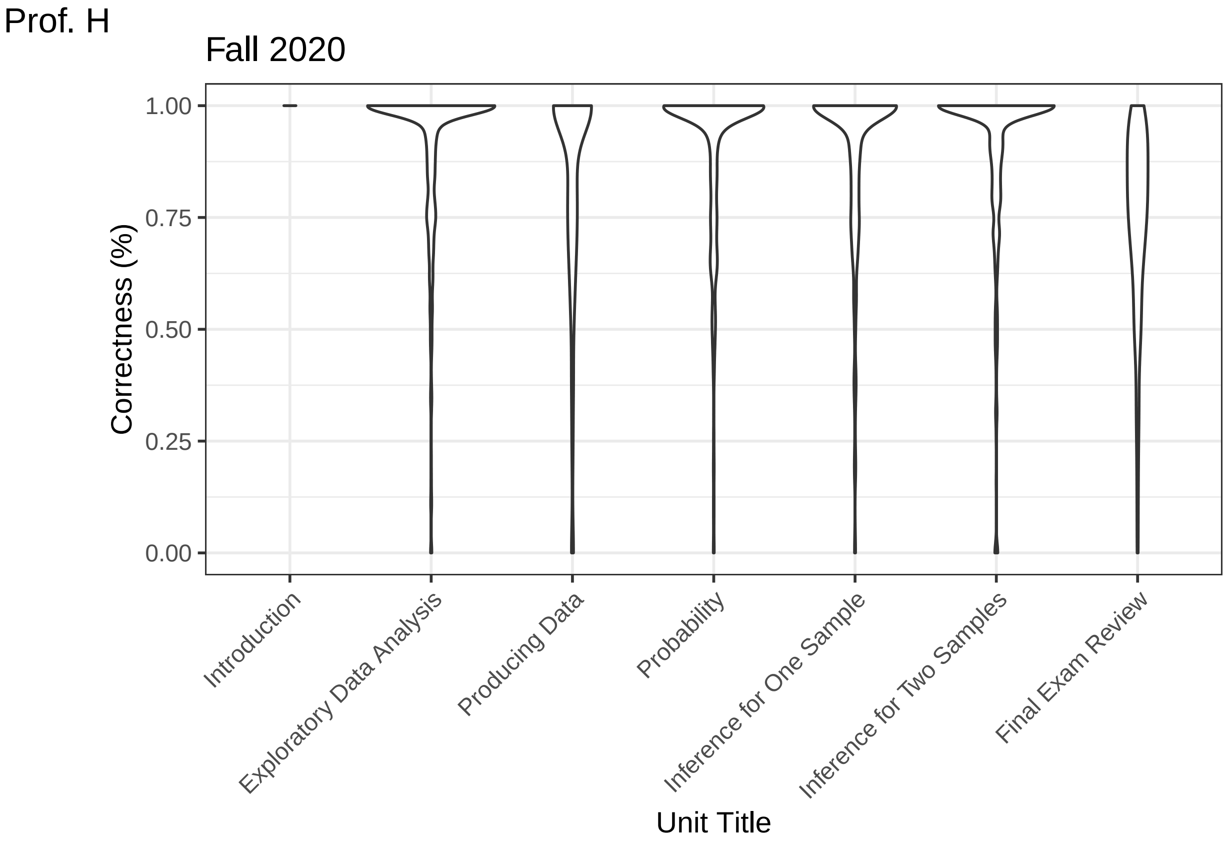
```r
# Prof. H
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_harden_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. H")
```



```r
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_harden_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. H")
```

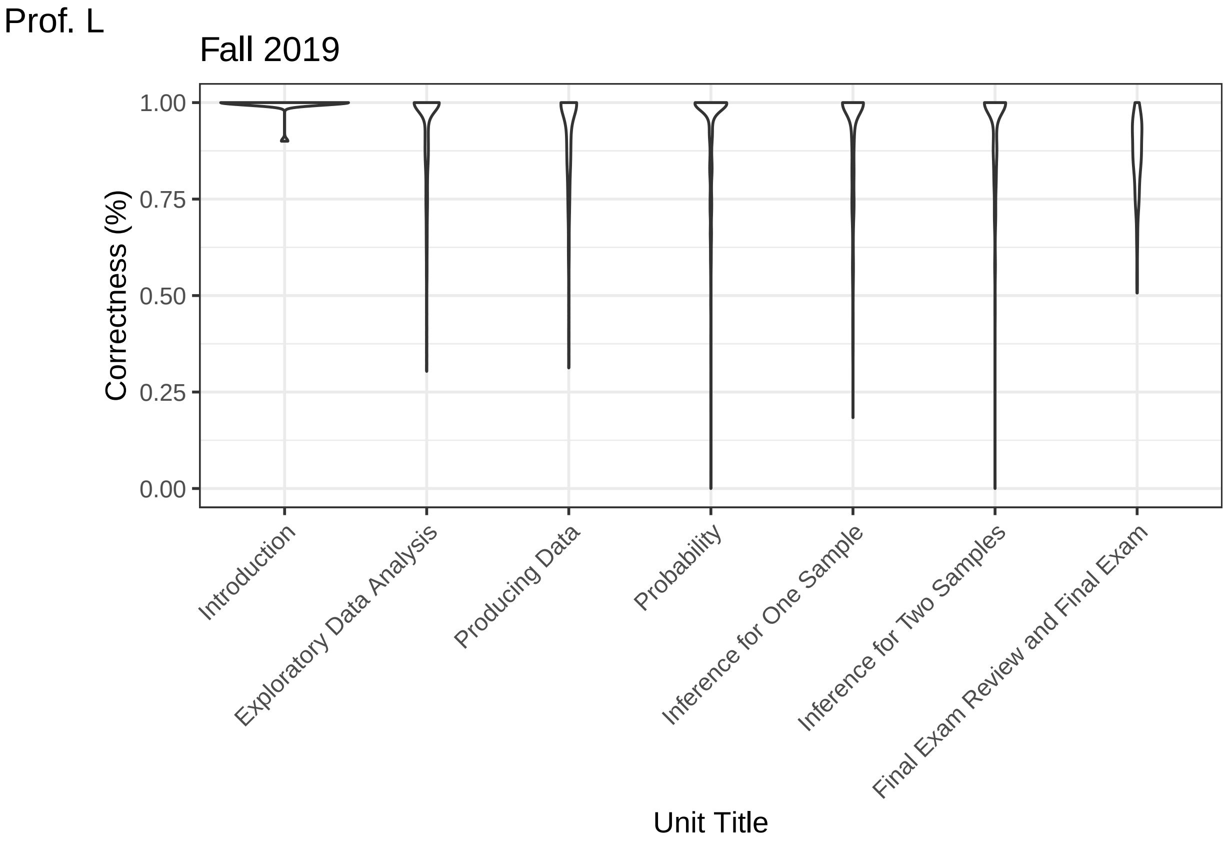


```
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_harden_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. H")
```

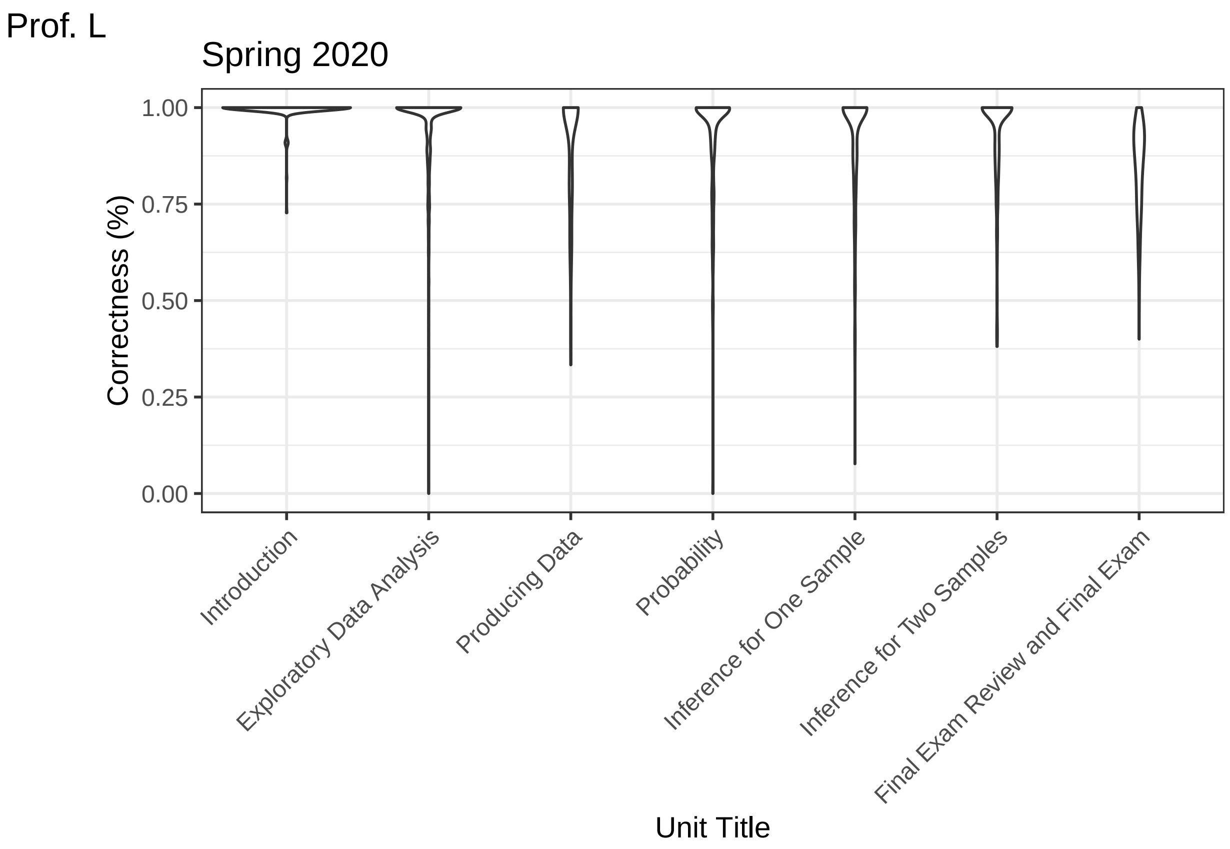


The distribution from the violin plots show that there are more students getting higher overall correctness in the unit questions in Fall 2020 than the previous 2 semesters, in sections taught by Prof. H.

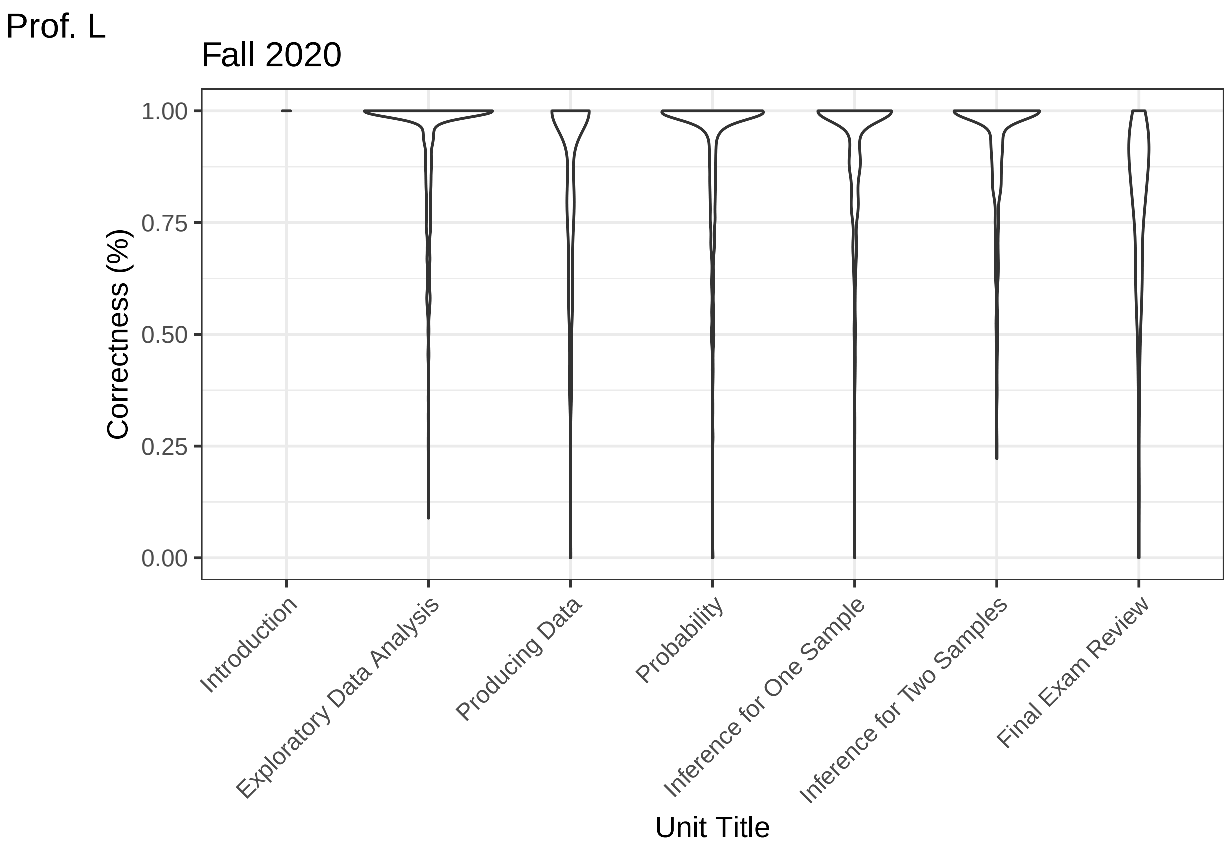
```
# Prof. L
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_liu_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. L")
```



```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_liu_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. L")
```

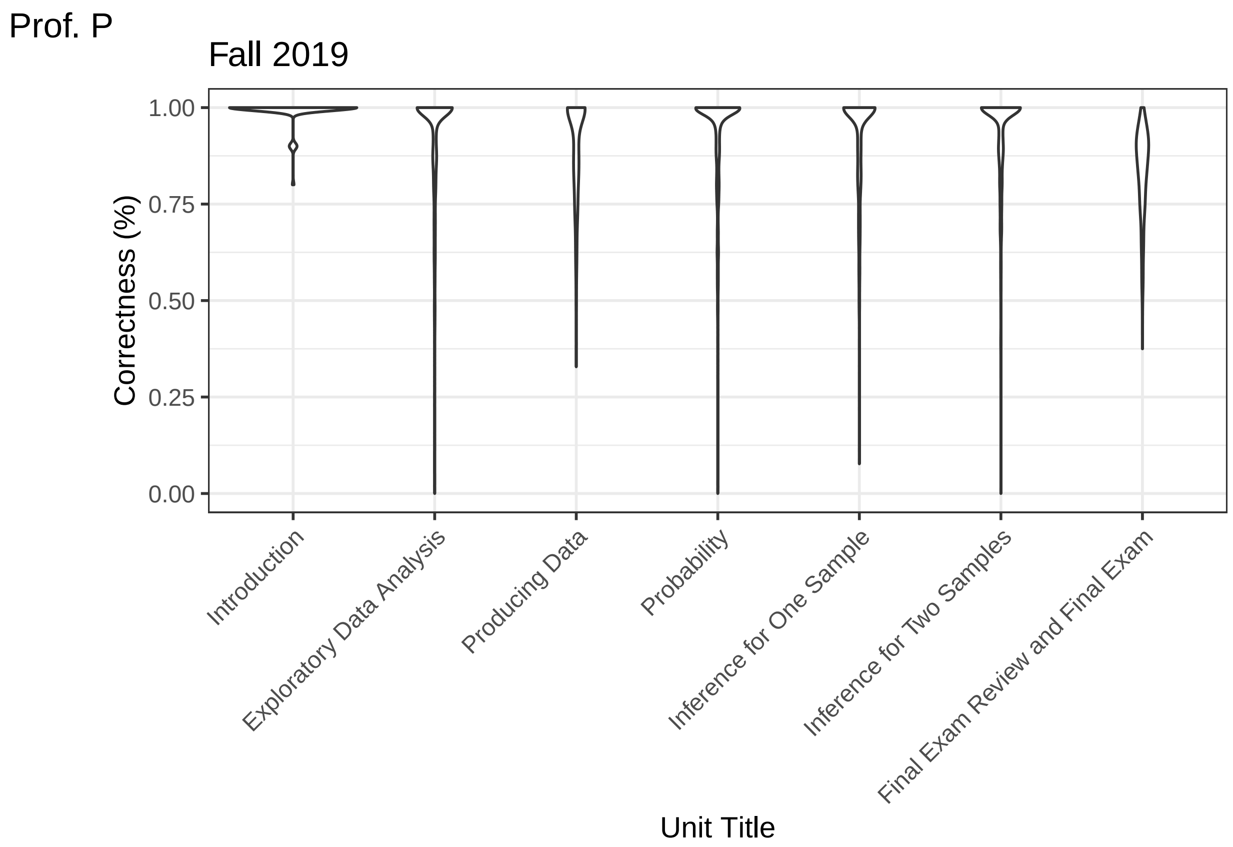


```
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_liu_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. L")
```

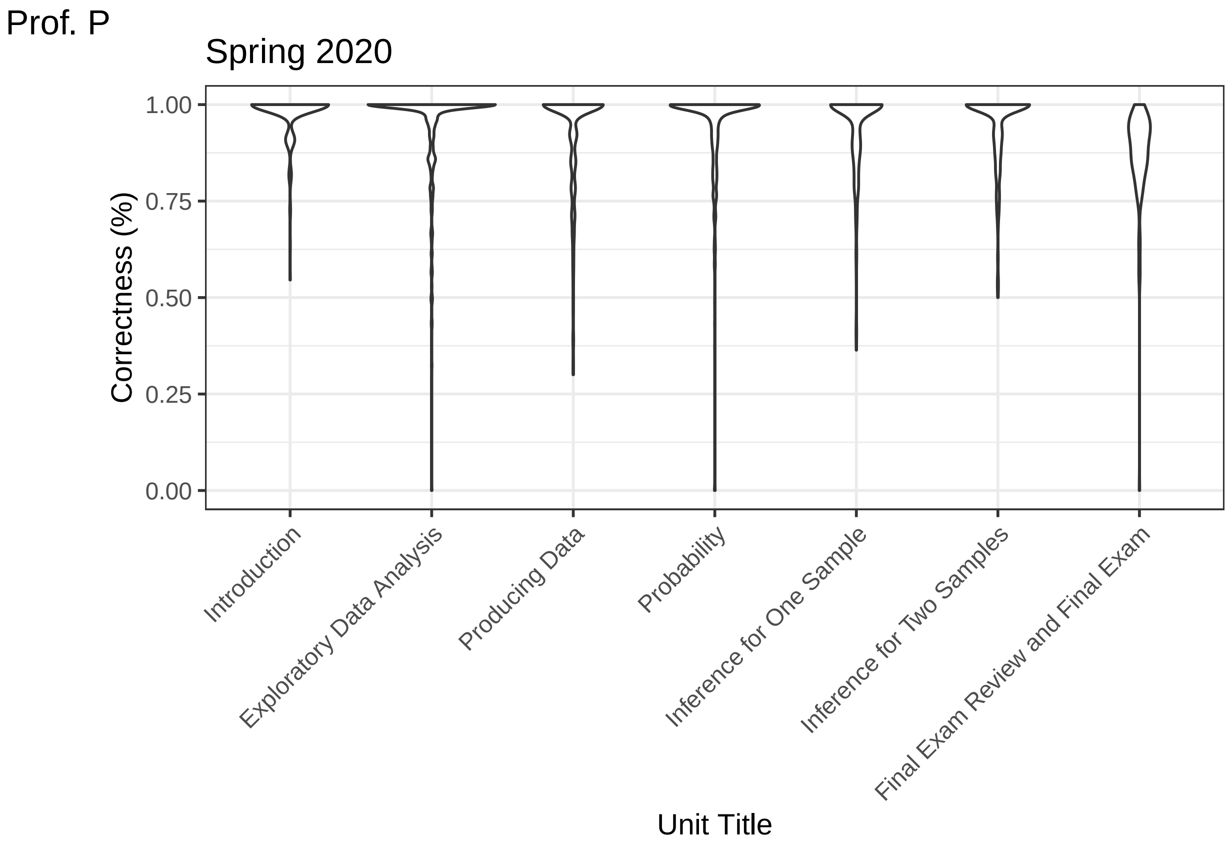


Similar patterns are found in the sections taught by Prof. L.

```
# Prof. P
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_bp_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. P")
```

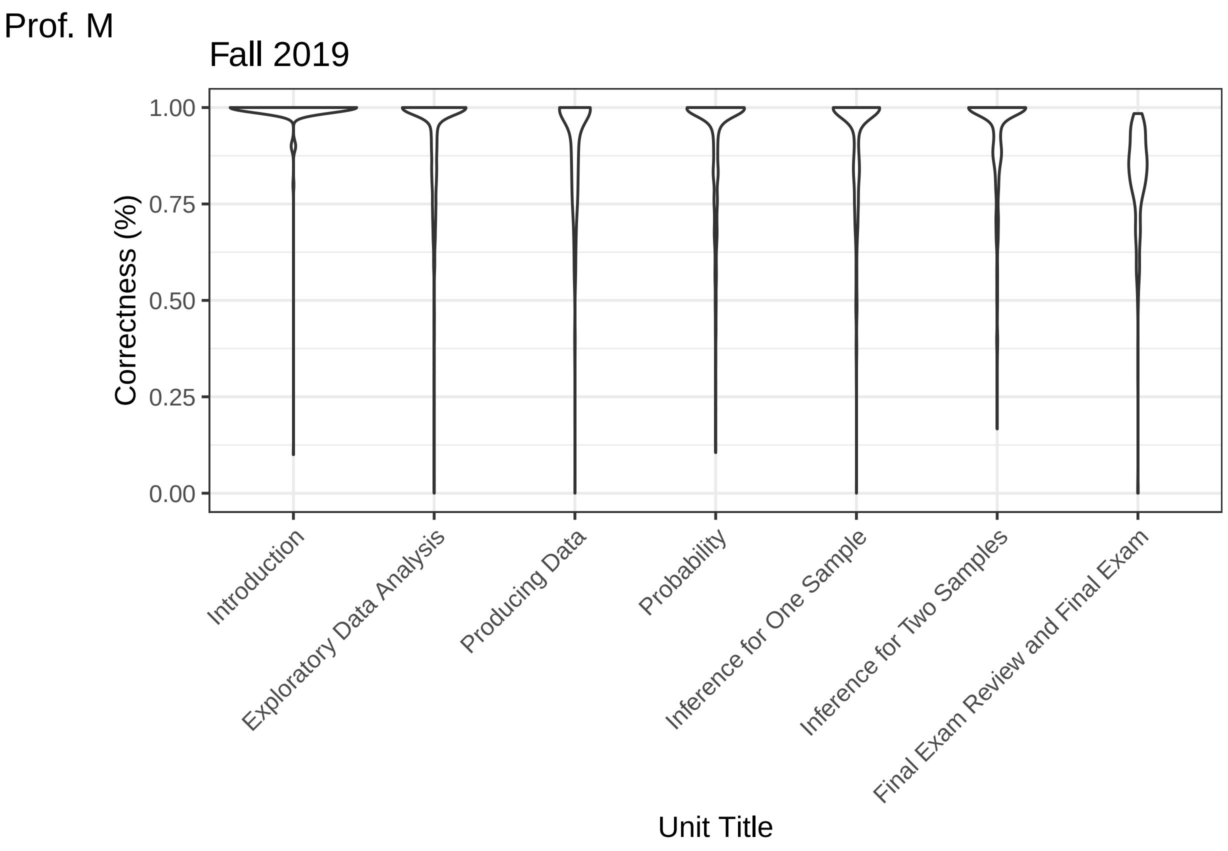


```
# Spring 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_bp_s20) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Spring 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. P")
```

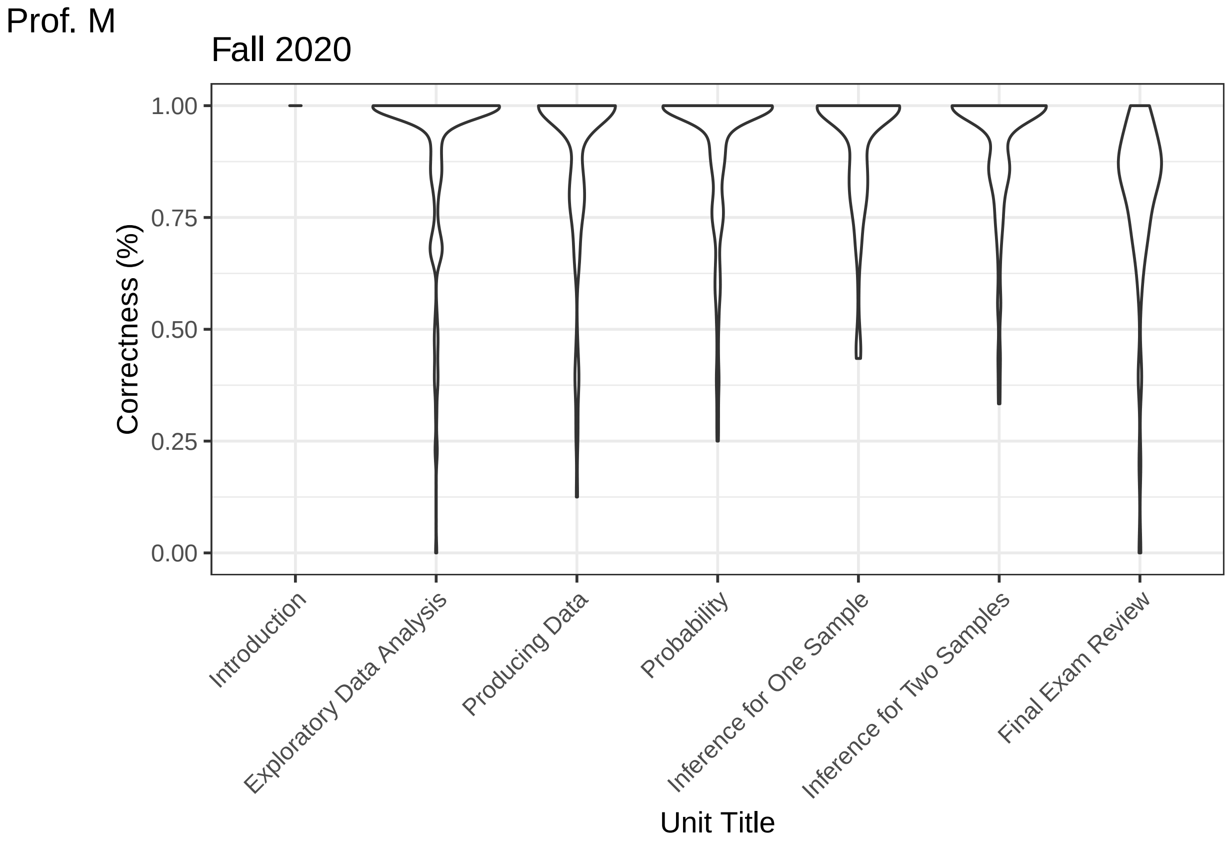


More students get higher overall correctness in Spring 2020 than in Fall 2019 in the sections taught by Prof. P.

```
# Prof. M
# Fall 2019
summary_unit_f19_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_meadows_f19) %>%
  filter(unit_title != "Tests 1 â€" 4") %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2019",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. M")
```



```
# Fall 2020
summary_unit_f20_joined %>%
  mutate(perct = correct/practice) %>%
  filter(course_key %in% course_key_meadows_f20) %>%
  ggplot(aes(x = unit_title, y = perct)) +
  geom_violin() +
  theme_bw() +
  scale_y_continuous(limits = c(0:1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fall 2020",
       x = "Unit Title",
       y = "Correctness (%)",
       tag = "Prof. M")
```



Overall, we observe the overall correctness in Fall 2020 has improved for students across all units, among sections taught by all the selected instructors.