

Homework 1

Due Tuesday, September 1, 2026 at 11:59 p.m.

[Your Name Here]

Table of contents

Problem 1: Graph Critique (20 points)	1
Problem 2: Course Access and Policies (5 points)	2
Problem 3: Resisting the First Order for Categories (15 points)	3
Problem 4: UFO Sightings in Pittsburgh (60 points)	4
Problem 5: 2D visualization and inference of videogames (55 points)	6

General instructions for all assignments:

- Use this file as the template for your submission. Be sure to write your name at the top of this page in the author section.
- Open the Homework 1 Gradescope assignment from Canvas and submit the rendered PDF there. Keep this .qmd source file with your work.
- Your file should contain the code to answer each question in its own code block. Your code should produce plots/output that will be automatically embedded in the rendered PDF. Your lab and homework files will include template code chunks like the following:

```
# PUT YOUR CODE AND PLOT HERE
```

- Although it's okay to discuss homework problems with other students, all of your homework (code, written answers, etc.) should be only your own. Instances of identical, nearly identical, or copied homework will be considered cheating and plagiarism. In other words, you must follow rules of academic integrity (as detailed in the syllabus).

Problem 1: Graph Critique (20 points)

In this question, you will practice describing and critiquing a visualization from a recent source. The goal is to make the expectations for a concise, evidence-based graph critique clear.

- (5 points) Find a graph from the Internet from the past 30 days (e.g., from a news article, blog post, online forum, etc.). For this part, all you have to do is include the graph. You have a few choices here:
 - Option 1 (HTML only): embed the graph/image directly in RMarkdown (see below for instructions - **but note this approach does NOT work if knitting to PDF!**), or

```
# OPTION 1 INSTRUCTIONS: Replace image address in the code below with your graph/  
image address (just use copy image address and paste it in here).  
# Then set echo = FALSE so the code is NOT displayed in your submission.  
#knitr::include_graphics("https://flowingdata.com/wp-content/uploads/2022/08/  
serena-williams-ranking.png")
```

- Option 2 (HTML or PDF): include a link to the graph in your answer to this question (**do this if knitting to PDF!**), or

OPTION 2 INSTRUCTIONS: Here is a link to the graph that will be in the solutions. This is from a New York Times article but I accessed it from the FlowingData website managed by Nathan Yau.

- Option 3 (HTML or PDF): save the image locally and refer to the file-path in a code chunk (but provide a link to source of the image in your file).

```
# OPTION 3 INSTRUCTIONS: Save the image on your computer in the same directory  
as this Rmd file, then just update the file name (make sure to make the file  
type correct, e.g., change jpg to png etc) in the code below and uncomment to  
knit it.  
#knitr::include_graphics("insert_graph_name_here.jpg")
```

- b. (5 points) **Describe the graph** in 2-5 sentences. Be sure that your description touches on the following points:

- What does the graph show?
- What variables are plotted, whether it's via symbols, color, or other features of the graph?
- What is the main result of the graph?

[PUT YOUR ANSWER HERE]

- c. (10 points) **Critique the graph** in 4-7 sentences. Be sure that your critique touches on the following points:

- What are the main goal(s) of the graph?
- Does the graph do a good job of achieving its goals?
- What are the strengths and weaknesses (if any) of the graphic?
- What would you change (if anything) about this graphic?

[PUT YOUR ANSWER HERE]

Problem 2: Course Access and Policies (5 points)

- a. (1 point) Open the course Canvas site and confirm that you can find the syllabus, course calendar, and Homework 1 Gradescope assignment link.
- b. (2 points) Write the Homework 1 due date and time shown on the course calendar.

- c. (2 points) Review the collaboration and academic-integrity rules in the syllabus. Then write: “I have read and understand the course rules for discussing assignments and submitting my own work.” Type your name after the statement.

Problem 3: Resisting the First Order for Categories (15 points)

For this problem, we’ll work with the `starwars` dataset that is included in the `dplyr` package. The following code chunk loads the data, then uses the `unnest` function to make a new dataset `starwars_character_films` where each row corresponds to a character appearance in a film. Type `help(starwars)` in the console to view more information about the `starwars` dataset.

```
library(tidyverse)
data("starwars")
starwars_character_films <- starwars %>%
  dplyr::select(name, films) %>%
  unnest(films)
```

We will focus on the `films` variable, which indicates the film a particular character appeared in. Specifically, we will demonstrate how to reorder categories on graphs in R. This is particularly useful when visualizing categorical variables, because the plotting order that R chooses by default may not be the best choice for your graphs.

- a. (3pts): First, make a bar plot of the `films` variable. It’s okay if your plot isn’t properly titled/ labeled, but be sure to choose a non-default color. After making your plot, answer the following: What is the default plotting order for categorical (“character” or “factor”) variables in R? (You should be able to determine this from glancing at the order of the categories in your graph.) You will notice that you can’t read the labels on the x-axis. A simple fix to this is to add `+ coord_flip()` to your plot to flip the axes. (Note the `coord_flip` order starts from the bottom.)

```
# PUT YOUR CODE AND PLOT HERE
```

- b. (3pts): Read this introduction to the `forcats` package, which was designed to work with categorical data in R. The cheatsheet on that page will be particularly useful. Install and load the `forcats` package to your computer. Answer the following:
- What function in the `forcats` package can be used to reorder the levels of a factor in any order you want?

[PUT YOUR ANSWER HERE]

- What function in the `forcats` package can be used to reorder the categories from most frequent to least frequent?

[PUT YOUR ANSWER HERE]

- What combination of functions in the `forcats` package can be used to reorder the categories from least frequent to most frequent?

[PUT YOUR ANSWER HERE]

- c. (3pts) Recreate the plot in Part A but this time, order the categories from most frequent to least frequent. This time, be sure that your graph is appropriately labeled.

```
# PUT YOUR CODE AND PLOT HERE
```

- d. (3pts) Recreate the plot in Part A, but this time, order the categories in a logical order (such as episode order: The Phantom Menace, Attack of the Clones, ..., The Force Awakens; or release date order: A New Hope, The Empire Strikes Back, ..., The Force Awakens). Again, be sure that your graph is appropriately labeled. See the Star Wars wikipedia page for the episode and release orders.

```
# PUT YOUR CODE AND PLOT HERE
```

- e. (3pts) Recreate the plot from the previous part (Part D), but this time, rename the categories so that they use the following film abbreviations: I, II, III, IV, V, VI, VII. **Hint:** Check the help documentation for `fct_recode()`. See the Star Wars wikipedia page above for the episode order. Note that you no longer need to use `+ coord_flip()` to see the x-axis labels.

```
# PUT YOUR CODE AND PLOT HERE
```

Problem 4: UFO Sightings in Pittsburgh (60 points)

For this problem with a dataset via the #TidyTuesday project containing information about UFO sightings. The following code reads in the dataset using the `read_csv()` function:

```
library(tidyverse)
ufo_sightings <- read_csv("https://raw.githubusercontent.com/rfordatascience/tidyuesday/master/data/2019/2019-06-25/ufo_sightings.csv")
```

- a. (3pts) What are the number of rows and columns in this dataset? What are the names of the variables in this dataset? For this part, be sure to include any code you used to answer these questions. **You should always include all code you used to answer a question, unless stated otherwise. However, you should also be sure to answer questions outside of code blocks (unless stated otherwise), even if you think your code is self-explanatory.**

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

- b. (2pts) Read this dataset description to better understand the UFO sightings dataset. In particular, look at the Data Dictionary section. For this question, please do the following:
- Name at least three categorical variables.

[PUT YOUR ANSWER HERE]

- Name at least two quantitative variables.

[PUT YOUR ANSWER HERE]

- c. (10pts) Create a **single graph** that displays the **marginal distribution** of `country` (make sure your graph is labeled appropriately).

```
# PUT YOUR CODE AND PLOT HERE
```

- d. (3pts) Based on your graph, which country has the most UFO sightings? Which has the least?

[PUT YOUR ANSWER HERE]

- e. (6pts) Create a summarized dataset that contains the following information about `ufo_shape` for UFO sightings in Pittsburgh (note: use lower-case `pittsburgh`):

- `ufo_shape`: the UFO shape (this dataset should have one row for each shape)
- `prop`: proportion of Pittsburgh UFO sightings with the particular UFO shape
- `se`: standard error for UFO shape proportion estimate
- `lower`: lower bound for 95% CI
- `upper`: upper bound for 95% CI

Make sure to only look at UFO sightings in Pittsburgh and remove any sightings where the `ufo_shape` is missing (i.e., `is.na(ufo_shape)`).

```
# PUT YOUR CODE HERE
```

- f. (20pts) Create a **single graph** that displays the **marginal distribution** of `ufo_shape` for UFO sightings in Pittsburgh, using the dataset you constructed in part (e). Additionally, make sure your graph has the following:

- add 95% CI to your display (make sure the intervals are a different color than the geometric object displaying the proportions),
- order the `ufo_shape` category levels by frequency in your display,
- make sure the `ufo_shape` labels are easy to read and that your graph is labeled appropriately.

```
# PUT YOUR CODE AND PLOT HERE
```

- g. (2pts) Is there anything non-intuitive with any of the 95% CI you've displayed on your graph in part (f)? State Yes or No, and give a 1-2 sentence explanation.

[PUT YOUR ANSWER HERE]

- h. (2pts) Do you observe any pair of UFO shapes with intervals that do NOT overlap with each other? State Yes or No, and if Yes provide an example.

[PUT YOUR ANSWER HERE]

- i. (10pts) In the previous part, we ignored the issue of multiple testing. You can find that there are 120 different pairwise comparisons ($120 = \binom{16}{2} = \text{ncol}(\text{combn}(16, 2))$) you could make from your graph in part (f). As a result, if we attempt to make multiple pairwise comparisons from a single graph with 95% CIs, our chance of making a Type 1 error is greater than 5%.

In general, α -level CIs are constructed using the Normal quantile $z_{1-\alpha/2}$. When we construct 95% CIs, $\alpha = 0.05$, and thus we use $z_{1-\alpha/2} = z_{0.975}$ (computed with `qnorm(0.975)`). This is exactly what we used in the concept material by constructing 95% CIs using ± 2 standard errors.

Recreate your figure in part (f) but update your CIs with new CIs following the Bonferroni correction. For the Bonferroni correction, $\alpha = 0.05$ is instead set to $0.05/k$, where k is equal to the number of comparisons. Thus, for this part, all you need to do create new `upper` and `lower` values for your CIs based on this correction (i.e., use `qnorm` to replace your use of 2 in part (e)).

Additionally, use `pmin` and `pmax` to cap and floor your intervals respectively so they are within the $[0,1]$ interval (see `help(pmin)` and `help(pmax)`).

```
# PUT YOUR CODE AND PLOT HERE
```

- j. (2pts) Do you observe any pair of UFO shapes with intervals that do NOT overlap with each other in your updated figure in part (i)? State Yes or No, and if Yes provide an example.

[PUT YOUR ANSWER HERE]

Problem 5: 2D visualization and inference of videogames (55 points)

For this problem, we will work with a dataset from Kaggle containing information about videogame sales. The following code reads in the dataset using the `read_csv()` function from the course GitHub repository:

```
games_data <-  
  read_csv("https://raw.githubusercontent.com/ryurko/DataViz-Class-Data/main/  
  vgsales.csv")
```

- a. (5pts) What are the number of rows and columns in this dataset? What are the names of the variables in this dataset? For this part, be sure to include any code you used to answer these questions. **You should always include all code you used to answer a question, unless stated otherwise. However, you should also be sure to answer questions outside of code blocks (unless stated otherwise), even if you think your code is self-explanatory.**

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

- b. (5pts) Create a smaller dataset that only contains rows for games where the `Platform` is either `PS3`, `X360`, or `Wii`. (HINT: use the `%in%` operator as explained in the top answer on stackoverflow here.)

```
# PUT YOUR CODE HERE
```

- c. (20pts) Create a **single graph** using the dataset you made in part (b) that has the following qualities:
- can easily view the **marginal** distribution of `Genre`,
 - can see the *conditional* distribution of `Platform` given `Genre`,
 - make sure your plot labels are easy to read and that is your graph is labeled appropriately (i.e., include a plot title and titles for your axes).

```
# PUT YOUR CODE AND PLOT HERE
```

- d. (10) Next, conduct a **statistical test** to assess if `Genre` and `Platform` are independent. Report your *p*-value and formal conclusion from this test.

```
# PUT YOUR CODE HERE
```

[PUT YOUR ANSWER HERE]

- e. (15pts) Finally, create a mosaic plot with Pearson residuals to determine which, if any, combination of `Genre-Platform` display an unusually high (or low) count of games. Do you observe any combination displaying an unusually high (or low) count, answer **Yes** or **No** and if **Yes** then given an example from your plot. Make sure your plot labels are easy to read (HINT: change the `las` and `cex.axis` arguments in `mosaicplot`).

```
# PUT YOUR CODE AND PLOT HERE
```

[PUT YOUR ANSWER HERE]