Homework 11 Solutions

11/28/2020

Problem 1

```
library(tidyverse)
library(arm)
library(lme4)
library(latex2exp) # for nice plot labeling (using latex notation)

cdi <- read.table(file = "../cdi.dat")

cdi$pci <- rescale(2*cdi$per.cap.income)
cdi$phg <- rescale(2*cdi$pct.hs.grad)
```

(a)

See Figure 1.

```
# pooled regression
lm_all <- lm(pci ~ phg, data = cdi)

lm_all_rep <- data.frame(intercept = lm_all[["coefficients"]][1],
                         slope = lm_all[["coefficients"]][2],
                         state = unique(cdi$state))

# unpooled regression
state_lm_func <- function(df){
    mod <- lm(pci ~ phg, data = df)
    return(data.frame(intercept = mod[["coefficients"]][1],
                      slope = mod[["coefficients"]][2]))
}

state_lm_df <- cdi %>%
  group_by(state) %>% do(., state_lm_func(.))

## unpooled regression version II
lm_unpooled <- lm(pci ~ state + phg:state - 1, data = cdi) # single model

coef_unpooled <- coef(lm_unpooled)
state_loc <- substring(names(coef_unpooled), 6,7)
slope_loc <- substring(names(coef_unpooled), 9,11)

intercepts <- coef_unpooled[slope_loc == ""]
intercepts_df <- data.frame(state = substring(names(intercepts), 6,7),
                            intercept = intercepts)
```

```r
slopes <- coef_unpooled[slope_loc == "phg"]
slopes_df <- data.frame(state = substring(names(slopes), 6,7),
                        slope = slopes)

state_lm_df2 <- intercepts_df %>% left_join(slopes_df, by = 'state')

testthat::expect_equivalent(state_lm_df, state_lm_df2) # no error - so true

## unpooled regression version III
lm_unpooled2 <- lm(pci ~ state*phg - 1 - phg, data = cdi) # single model

coef_unpooled2 <- coef(lm_unpooled2)
state_loc2 <- substring(names(coef_unpooled2), 6,7)
slope_loc2 <- substring(names(coef_unpooled2), 9,11)

intercepts2 <- coef_unpooled2[slope_loc2 == ""]
intercepts_df2 <- data.frame(state = substring(names(intercepts2), 6,7),
                             intercept = intercepts2)
slopes2 <- coef_unpooled2[slope_loc2 == "phg"]
slopes_df2 <- data.frame(state = substring(names(slopes2), 6,7),
                         slope = slopes2)

state_lm_df3 <- intercepts_df2 %>% left_join(slopes_df2, by = 'state')

testthat::expect_equivalent(state_lm_df, state_lm_df3) # no error - so true


# multilevel model
multilevel_lmer <- lmer(formula = pci ~ 1 + phg + (1 | state) + (0 + phg | state),
                        data = cdi)

multilevel_lmer_df <- coef(multilevel_lmer)$state %>%
  tibble::rownames_to_column() %>%
  rename(state = "rowname",
         intercept = "(Intercept)",
         slope = "phg")

# multilevel model version II
betas <- fixef(multilevel_lmer)
etas <- ranef(multilevel_lmer)$state

combo_multilevel_lmer_df <- etas +
  matrix(rep(betas, each = nrow(etas)), nrow = nrow(etas))

testthat::expect_equivalent(combo_multilevel_lmer_df,
                            coef(multilevel_lmer)$state) # doesn't error if true

# combined
combined <- rbind(mutate(lm_all_rep[, c("state", "intercept", "slope")],
                         id = "pooled regression"),
                  mutate(state_lm_df[, c("state", "intercept", "slope")],
                         id = "unpooled regression"),
                  mutate(multilevel_lmer_df[, c("state", "intercept", "slope")],
```

```
                          id = "multilevel model")) %>%
  mutate(id = factor(id, levels = c("pooled regression",
                                    "unpooled regression",
                                    "multilevel model")))
```

```
ggplot(cdi) +
  geom_point(aes(x = phg, y= pci)) +
  geom_abline(data = combined,
              aes(slope = slope, intercept= intercept, group = state,
                  color = id))+
  facet_wrap(~ state) +
  theme_minimal() +
  labs(color = "model type")
```

(b)

See figure 2.

```
# multilevel model
multilevel_res_fit <- data.frame(fitted = fitted(multilevel_lmer),
                                 residual = resid(multilevel_lmer),
                                 state = cdi$state)

# multilevel model, part II
source("../residual-functions.r")
multilevel_res_fit2 <- data.frame(fitted = yhat.cond(multilevel_lmer),
                                  residual = r.cond(multilevel_lmer),
                                  state = cdi$state)

testthat::expect_equivalent(multilevel_res_fit, multilevel_res_fit2) # no error if true
```

```
multilevel_res_fit %>%
ggplot() +
  geom_point(aes(x = fitted, y = residual)) +
  facet_wrap(~ state) +
  geom_hline(yintercept = 0) +
  theme_minimal() +
  labs(y = "conditional residual",
       x = TeX("conditional $\\hat{y}$"))
```

Problem 2

Intentionally missing.

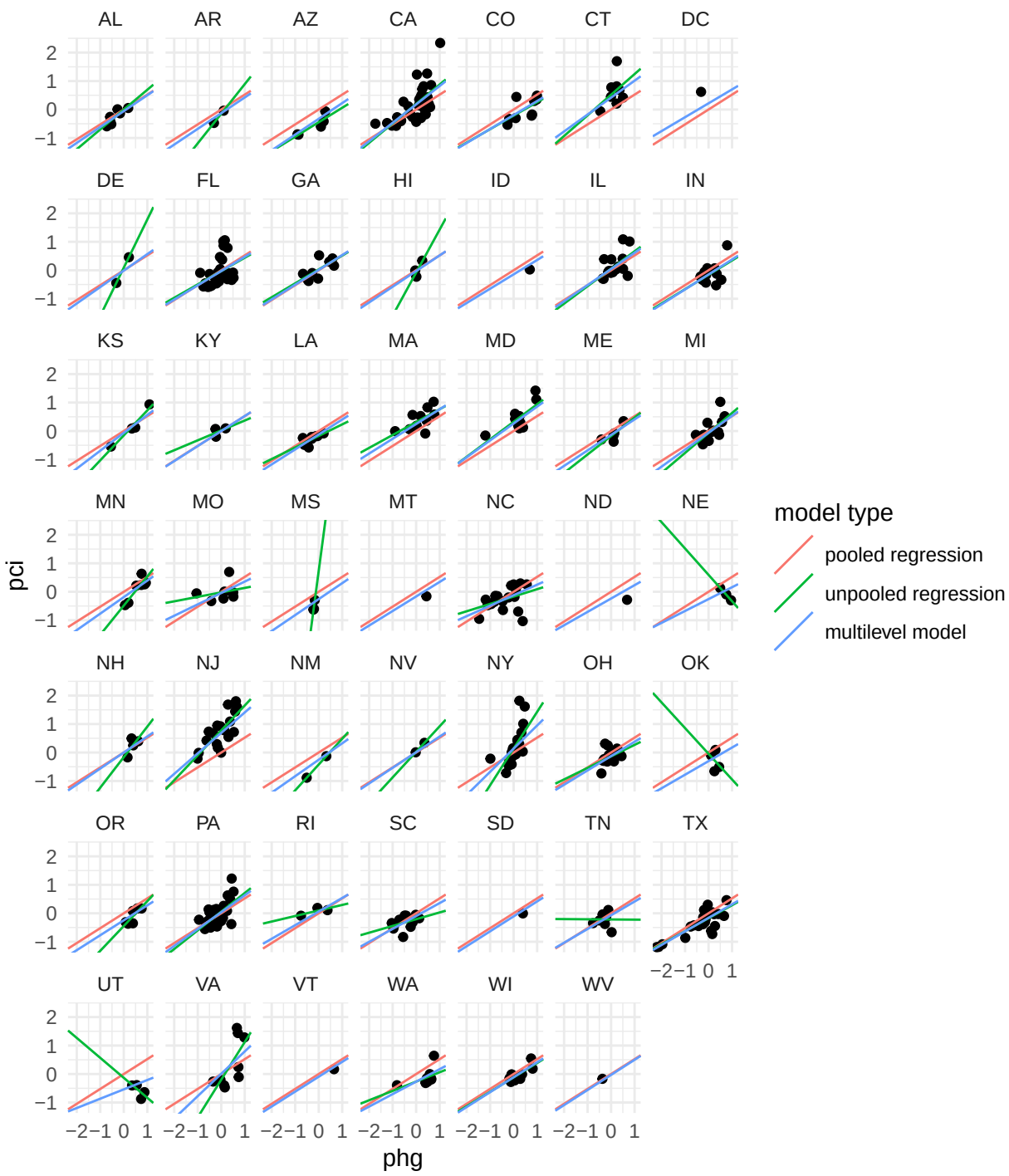

Figure 1: Problem 1a: Different linear models to compare per capital income to percent high school graduates
.

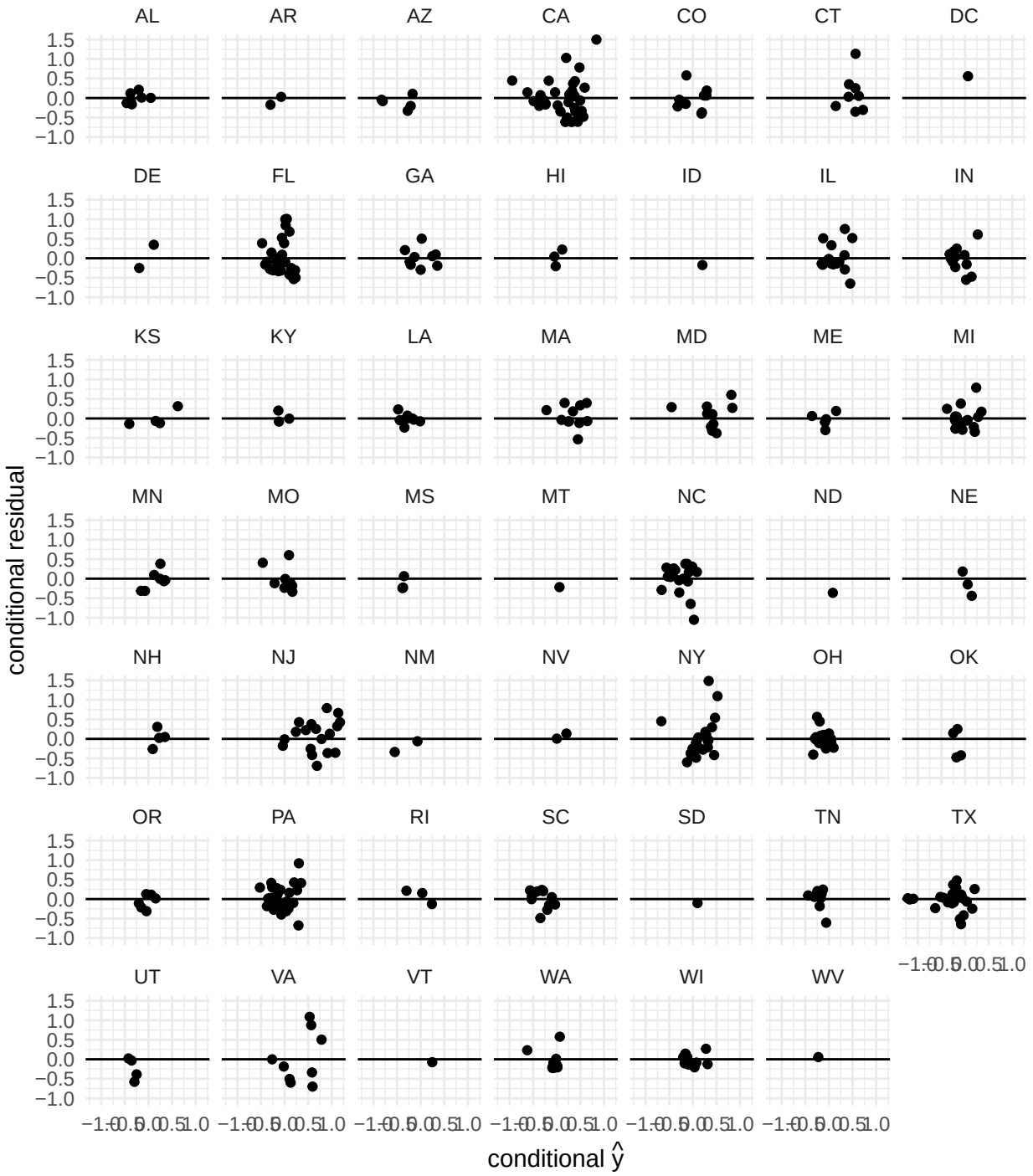

Figure 2: Problem 1b: Examinig the conditional residual vs conditional $\hat{y}$ for multilevel model .