

Assignment 09

Your name and student ID

Today's date

```
BEGIN ASSIGNMENT
requirements: requirements.R
generate: true
files:
- data
- src
```

Instructions

- Solutions will be released on Tuesday, August 10
- This semester, homework assignments are for practice only and will not be turned in for marks.

Helpful hints:

- Every function you need to use was taught during lecture! So you may need to revisit the lecture code to help you along by opening the relevant files on Datahub. Alternatively, you may wish to view the code in the condensed PDFs posted on the course website. Good luck!
- Knit your file early and often to minimize knitting errors! If you copy and paste code for the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting! We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of knitting errors more easily. This will save you and the GSIs from frustration!

Section 1: Abstract interpretation [5 points]

Read the following abstract and answer the questions that follow. J Asthma. 2018 Oct 11:1-12. doi: 10.1080/02770903.2018.1508471. [Epub ahead of print] Impact of scenario based training on asthma first aid knowledge and skills in school staff: an open label, three-arm, parallel-group repeated measures study. Luckie K1, Saini B1, Soo YYB1,2, Kritikos V1,3, Charles Collins JB1, Jane Moles R1.

OBJECTIVE: To test the hypothesis that scenario-based skills training is more effective than knowledge training alone in improving the asthma first aid (AFA) skills of school personnel. Education developed specifically for non-primary caregivers such as school staff is vital to minimize the risk of mortality associated with asthma.

METHODS: Schools were allocated to one of three arms to compare AFA knowledge and AFA skills. Arm 1 underwent conventional asthma training, arm 2 underwent scenario-based training and arm 3 had a combination of the two. Conventional asthma training involved a didactic oral presentation. The scenario-based skills training required the participant to describe and demonstrate how they would manage a child having a severe exacerbation of asthma using equipment provided. Follow-up occurred at 3 weeks post baseline and again between 3-7 months after the first training/education visit.

RESULTS: Nineteen primary schools (204 participants) were recruited. One-way ANOVA and Bonferroni Post-Hoc Tests showed there was a significant difference in AFA skills scores between the study arms who underwent scenario-based training; arms 2 and 3 (91.5% and 91.1%) and arm 1 who underwent conventional asthma training (77.3%) ($p < 0.001$). AFA knowledge improved significantly in all study arms with no differences between study arms. Improvements seen in both AFA knowledge and AFA skills were maintained over time.

CONCLUSIONS: Scenario-based training was superior to conventional didactic asthma training for AFA skills acquisition and overall competency in the administration of AFA and should be included in future asthma training programs.

1. [1 point] Two methods of hypothesis testing (types of tests) are mentioned in the abstract. What is the null hypothesis for each of these tests (please list in the order they are mentioned in the abstract?)

BEGIN QUESTION

name: p1

manual: true

H_0 : There is no difference between scenario-based skills training and knowledge training alone in improving the asthma first aid (AFA) skills of school personnel.

H_0 : There is no difference in the improvement of asthma first aid skills of school personnel after receiving training scenario-based skills training and a combination of didactic and skills training.

2. [1 point] There are two outcomes of interest in this study. For which **outcome** would you conclude that there is a significant difference between the training groups.

BEGIN QUESTION

name: p2

manual: true

There is a significant difference in outcome between arm1 and arms2 and 3 of the training groups.

3. [1 point] If you were a school administrator why might you choose the arm 3 training?

BEGIN QUESTION

name: p3

manual: true

Arm 3 showed a significant improvement over just receiving didactic training but it includes some components that are more traditional to other teaching scenarios. The teaching program would not need to be completely overhauled but rather just include skills-based training.

4. [1 point] List one question you might want to ask about the methods, sample or results that would help you interpret the findings of this study?

BEGIN QUESTION

name: p4

manual: true

Possible questions could be: 1. What, if any, training had these recruits already experienced? 2. Were the participants randomized to a study arm? 3. What was the difference between arm 2 and arm 3? 4. How was improvement defined in the context of this study?

5. [1 point] What is another test that could have been considered for these study data?

BEGIN QUESTION

name: p5

manual: true

Two sample t test could have been considered to see if there was a difference in average improvement between arm 2 and arm 3.

Section 2: ANOVA and Tukey's HSD [6 points]

For this question we will use the data from the NHANES survey ‘

```
##
## -- Column specification -----
## cols(
##   .default = col_character(),
##   ridageyr = col_double(),
##   drinks = col_double(),
##   bmxwt = col_double(),
##   bmxht = col_double(),
##   bmxbmi = col_double(),
##   bpxpls = col_double(),
##   bpxsy1 = col_double(),
##   bpxsy2 = col_double(),
##   bpxdi1 = col_double(),
##   bpxdi2 = col_double(),
##   lbdhdd = col_double(),
##   sleep = col_double(),
##   lbdldl = col_double()
## )
## i Use 'spec()' for the full column specifications.

## # A tibble: 6 x 40
##   ridageyr agegroup gender military  born  citizen drinks drinkscat bmxwt bmxht
##   <dbl> <chr>    <chr> <chr>    <chr> <chr>    <dbl> <chr>    <dbl> <dbl>
## 1      72 65+      Male  History o~ Born~ US cit~    0 0      88.9 175.
## 2      73 65+      Female No      Born~ US cit~    0 0      52 162.
## 3      61 50-64      Female No      Born~ US cit~    2 11-Jan 93.4 162.
## 4      26 20-34      Female No      Born~ US cit~   209 96-364 47.1 152.
## 5      33 20-34      Female No      No    US cit~    NA NA     56.8 158
## 6      32 20-34      Male  No      No    No      300 96-364 79.7 166.
## # ... with 30 more variables: bmxbmi <dbl>, bmicat <chr>, bpxpls <dbl>,
## #   bpxsy1 <dbl>, bpxsy2 <dbl>, sys1d <chr>, sys2d <chr>, bpxdi1 <dbl>,
## #   bpxdi2 <dbl>, dias1d <chr>, dias2d <chr>, bpcat <chr>, chest <chr>,
## #   fs1 <chr>, fs2 <chr>, fs3 <chr>, lbdhdd <dbl>, hdlcat <chr>, highhdl <chr>,
## #   hi <chr>, asthma <chr>, vwa <chr>, vra <chr>, va <chr>, aspirin <chr>,
## #   sleep <dbl>, is <chr>, hs <chr>, lbdldl <dbl>, highldl <chr>
```


6. [1 point] Generate the mean and standard deviations in a dataframe for blood lipid level “lbddl” by Blood pressure group “bpcat”. Use dplyr functions.

BEGIN QUESTION

name: p6

manual: false

points: 1

```
. = " # BEGIN PROMPT
p6 <- 'Your code here'
p6
" # END PROMPT

# BEGIN SOLUTION
p6 <- nhanes %>% group_by(bpcat) %>% summarize(mean = mean(lbddl), sd = sd(lbddl))
p6
```

```
## # A tibble: 3 x 3
##   bpcat      mean    sd
##   <chr>    <dbl> <dbl>
## 1 Elevated    111.  35.0
## 2 Hypertension 113.  35.1
## 3 Normal     113.  35.1
```

```
# END SOLUTION
```

```
## Test ##
test_that("p6a", {
  expect_true("data.frame" %in% class(p6))
  print("Checking: p6 is a dataframe")
})
```

```
## [1] "Checking: p6 is a dataframe"
## Test passed
```

```
## Test ##
test_that("p6b", {
  expect_true(all.equal(p6[[2]], c(111.0524, 113.1113, 113.3995), tolerance = .001))
  print("Checking: correct mean values")
})
```

```
## [1] "Checking: correct mean values"
## Test passed
```

```
## Test ##
test_that("p6c", {
  expect_true(all.equal(p6[[3]], c(35.00980, 35.09351, 35.11905), tolerance = .001))
  print("Checking: correct sd values")
})
```

```
## [1] "Checking: correct sd values"
## Test passed
```

7. [1 point] Create a boxplot that helps you to visualize these data.

BEGIN QUESTION

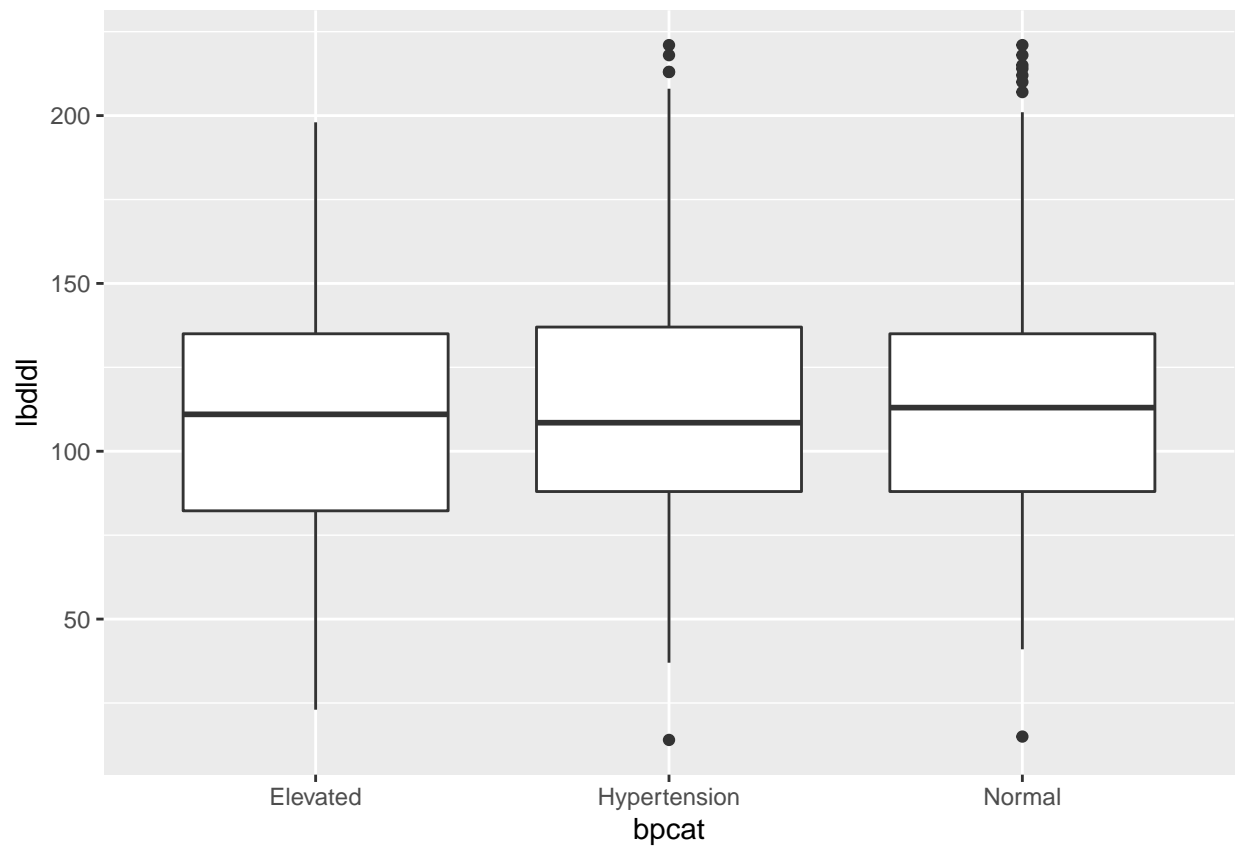
name: p7

manual: false

points: 1

```
. = " # BEGIN PROMPT
p7 <- 'Your plot here'
p7
" # END PROMPT

# BEGIN SOLUTION
p7 <- ggplot(nhanes, aes(x = bpcat, y = lbdldl)) +
  geom_boxplot()
p7
```



```
# END SOLUTION
```

```
## Test ##
test_that("p7a", {
  expect_true("ggplot" %in% class(p7))
  print("Checking: ggplot defined")
})
```

```
## [1] "Checking: ggplot defined"
## Test passed
```

```
## Test ##
test_that("p7b", {
  expect_true(identical(p7$data, nhanes))
  print("Checking: nhanes data used")
})
```

```
## [1] "Checking: nhanes data used"
## Test passed
```

```
## Test ##
test_that("p7c", {
  expect_true(rlang::quo_get_expr(p7$mapping$x) == "bpcat")
  print("Checking: blood pressure categories on x axis")
})
```

```
## [1] "Checking: blood pressure categories on x axis"
## Test passed
```

```
## Test ##
test_that("p7d", {
  expect_true(rlang::quo_get_expr(p7$mapping$y) == "lbdldl")
  print("Checking: blood lipid levels on y axis")
})
```

```
## [1] "Checking: blood lipid levels on y axis"
## Test passed
```

```
## Test ##
test_that("p7e", {
  expect_true("GeomBoxplot" %in% class(p7$layers[[1]]$geom))
  print("Checking: boxplot created")
})
```

```
## [1] "Checking: boxplot created"
## Test passed
```

8. [2 points] Conduct an ANOVA with Tukey's HSD for these data. Assign your model to the variable `tukey`.

BEGIN QUESTION

name: p8

manual: false

points: 2

```
. = " # BEGIN PROMPT
tukey <- 'Your code here'
# p8 <- tidy(tukey) #uncomment this line
" # END PROMPT

# BEGIN SOLUTION
model <- aov(lbdl-dl~bpcat, data = nhanes)
tukey <- TukeyHSD(model, conf.level = 0.95)
p8 <- tidy(tukey)
p8
```

```
## # A tibble: 3 x 7
##   term contrast          null.value estimate conf.low conf.high adj.p.value
##   <chr> <chr>              <dbl>     <dbl>    <dbl>    <dbl>    <dbl>
## 1 bpcat Hypertension-Elevated      0      2.06    -4.66     8.77    0.752
## 2 bpcat Normal-Elevated            0      2.35    -4.56     9.26    0.705
## 3 bpcat Normal-Hypertension        0      0.288   -5.03     5.61    0.991
```

```
# END SOLUTION
```

```
## Test ##
test_that("p8a", {
  expect_true("TukeyHSD" %in% class(tukey))
  print("Checking: TukeyHSD function used")
})
```

```
## [1] "Checking: TukeyHSD function used"
## Test passed
```

```
## Test ##
test_that("p8b", {
  expect_true(all.equal(tukey$bpcat[1,4], 0.751907, tol = 0.001))
  print("Checking: adjusted p-values")
})
```

```
## [1] "Checking: adjusted p-values"
## Test passed
```

9. [1 point] What are the null and alternative hypotheses for this test?

BEGIN QUESTION

name: p9

manual: true

H_0 = The average blood lipid level is equal across blood pressure groups.

H_A = The average blood lipid level is different for at least one of the three blood pressure groups.

10. [1 point] What do you conclude from your analysis?

BEGIN QUESTION

name: p10

manual: true

Since all of the confidence intervals contain 0 and none of the p-values are significant at an $\alpha = 0.05$ level, we fail to reject the null hypothesis that the average blood lipid level is equal across blood pressure groups.

Section 3: Non-parametric [3 points]

You are testing the change in test scores following an intensive tutoring session.

You have the following data from a small group of students each student is tested before and after the tutoring session.

Each row represents one student.

Time 1	Time 2
65	77
87	100
77	75
90	89
70	80
84	81
92	91
83	96
85	84
91	89
68	88
72	100
81	81

```
#this code makes a dataframe of the table you see above
test_scores <- tribble(
  ~time1, ~time2,
  65, 77,
  87, 100,
  77, 75,
  90, 89,
  70, 80,
  84, 81,
  92, 91,
  83, 96,
  85, 84,
  91, 89,
  68, 88,
  72, 100,
  81, 81)
```

11. [2 point] Calculate the appropriate non-parametric test for these data by hand. Attach an image to show your work if you would like. Make sure to place the image in the `src` directory. Uncomment the line by deleting the pound sign. Report the p-value> Keep it as a decimal and round to 4 places.

BEGIN QUESTION

name: p11

manual: false

points: 2

```
. = " # BEGIN PROMPT
#knitr::include_graphics('src/path-to-file')
p11 <- 'YOUR P-VALUE HERE'
```

```

p11
" # END PROMPT

# BEGIN SOLUTION
test_scores <- test_scores %>% mutate(diff = time1 - time2)
n <- nrow(test_scores) - 1 #one obs was dropped as the difference was 0
t <- 21
mu <- (n*(n + 1))/4
mu

```

```
## [1] 39
```

```

sigma <- sqrt((n*(n+1)*(2*n+1))/24)
sigma

```

```
## [1] 12.74755
```

```

z.stat <- (t - mu)/sigma
z.stat

```

```
## [1] -1.412036
```

```

p11 <- round(2*pnorm(z.stat),4)
p11

```

```
## [1] 0.1579
```

```
# END SOLUTION
```

```

## Test ##
test_that("p11", {
  expect_true(all.equal(p11, 0.1579393, tolerance = .001))
  print("Checking: p-value to at least 3 decimals")
})

```

```

## [1] "Checking: p-value to at least 3 decimals"
## Test passed

```


12. [1 point] Check your work using [insert your test].test() function in R. Keep your answer as a decimal rounded to 4 decimals. Report your p-value and save it to the variable p12.

BEGIN QUESTION

name: p12
manual: false
points: 1

```
. = " # BEGIN PROMPT
p12 <- 'Your p-value here'
p12
" # END PROMPT

# BEGIN SOLUTION
wilcox.test(test_scores %>% pull(time1), test_scores %>% pull(time2), paired=T, correct=FALSE)
```

```
##
## Wilcoxon signed rank test
##
## data: test_scores %>% pull(time1) and test_scores %>% pull(time2)
## V = 21, p-value = 0.157
## alternative hypothesis: true location shift is not equal to 0
```

```
p12 <- 0.1579
p12
```

```
## [1] 0.1579
```

```
# END SOLUTION
```

```
## Test ##
test_that("p12", {
  expect_true(all.equal(p12, 0.1579393, tol = 0.001))
  print("Checking: p-value to 3 decimals")
})
```

```
## [1] "Checking: p-value to 3 decimals"
## Test passed
```

Section 4: Voting during the 1992 election [21 points]

This code loads in the data frame `counties`:

```
load("data/A10_counties.sav")
```

These data are from the 1992 election and looks at the percent of votes cast (for each county) for the **democrat** (Bill Clinton), **republican** (George Bush), and independent presidential nominees (Ross Perot).

Ideally, if you were interested in voting patterns, you might look at the relationship between individual characteristics and whether each individual voted Democrat or Republican. However, data like that is often hard to come by. The `counties` data provide data on 3141 counties. Use `View()` to examine these data briefly and read the labels corresponding to the variables. Note that Alaska is not included and that two other counties with populations = 0 have also been excluded.

As discussed in class we have the entire population (not just a sample), so strictly speaking we don't need to perform statistical inference. However, we might pretend this is a sample so that we can apply the techniques of inference and gain competence creating and interpreting a linear model.

13. [2 points] Looking only at California, plot the relationship between the % of votes cast for the Democratic candidate (`democrat`) and the population density of the county (`pop.density`). Since we will only be using the counties from California, go ahead and subset the full `counties` dataset to only include observations from the state of CA.

BEGIN QUESTION

name: p13

manual: false

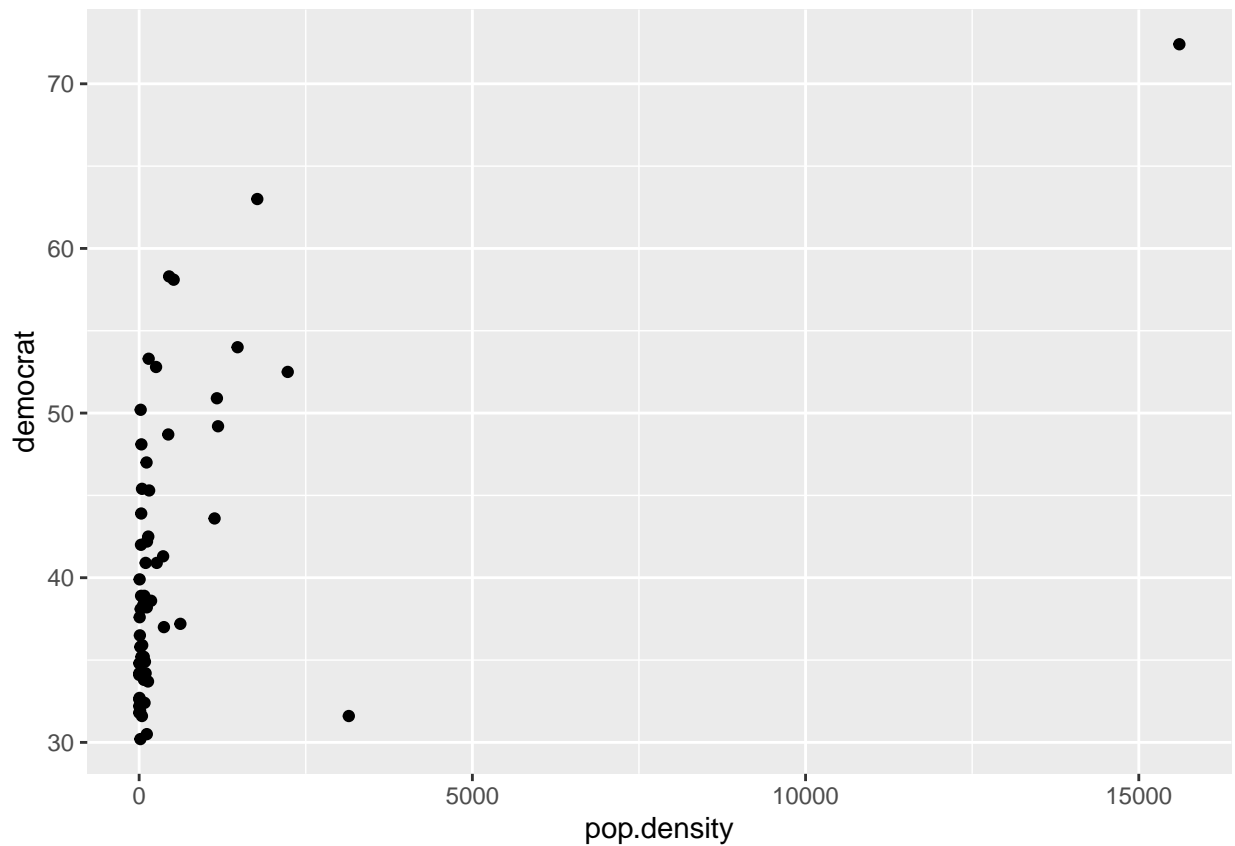
points: 2

```
. = " # BEGIN PROMPT
counties_CA <- 'SUBSET DATA HERE'

p13 <- 'YOUR PLOT HERE'
p13
" # END PROMPT

# BEGIN SOLUTION
counties_CA <- counties %>% filter(state == "CA")

p13 <- ggplot(counties_CA, aes(x = pop.density, y = democrat)) +
  geom_point()
p13
```



```
# END SOLUTION
```

```
## Test ##
```

```
test_that("p13a", {  
  expect_true("ggplot" %in% class(p13))  
  print("Checking: ggplot created")  
})
```

```
## [1] "Checking: ggplot created"  
## Test passed
```

```
## Test ##
```

```
test_that("p13b", {  
  expect_true(rlang::quo_get_expr(p13$mapping$x) == "pop.density")  
  print("Checking: pop.density on x axis")  
})
```

```
## [1] "Checking: pop.density on x axis"  
## Test passed
```

```
## Test ##
```

```
test_that("p13c", {  
  expect_true(rlang::quo_get_expr(p13$mapping$y) == "democrat")  
  print("Checking: democrat on y axis")  
})
```

```
## [1] "Checking: democrat on y axis"  
## Test passed
```

```
## Test ##
```

```
test_that("p13d", {  
  expect_true("GeomPoint" %in% class(p13$layers[[1]]$geom))  
  print("Checking: scatterplot added")  
})
```

```
## [1] "Checking: scatterplot added"  
## Test passed
```

14. [1 point] The above plot you made does not look very good. The distribution of population density is skewed right, with a few counties having much higher densities than the majority of counties. To see which counties these are, we will use `geom_text_repel` from the library `ggrepel` (loaded at the top of this assignment).

The template for using this function is: `geom_text_repel(aes(label = your_labelling_var))`. You will want to set the labeling variable to be the variable in the dataset containing the county names.

BEGIN QUESTION

name: p14

manual: false

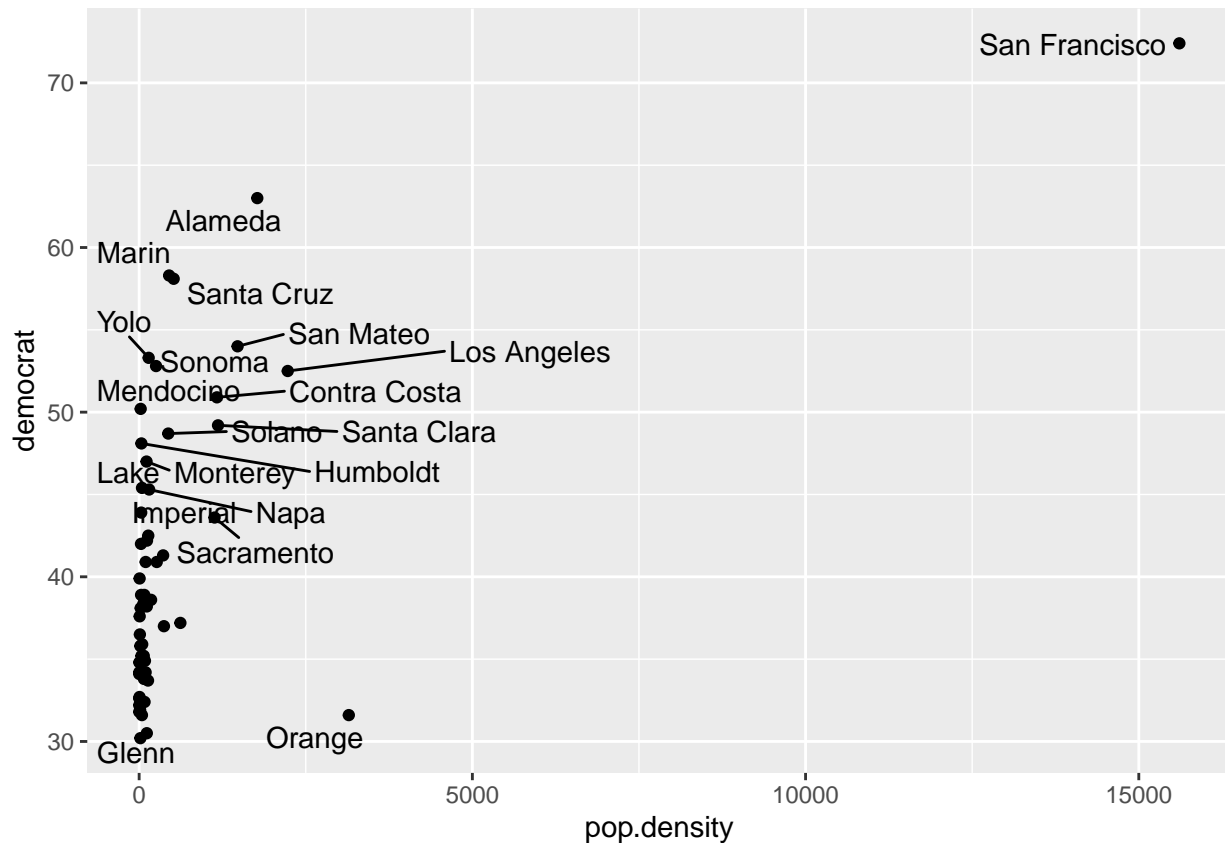
points: 1

```
. = " # BEGIN PROMPT
p14 <- 'YOUR CODE HERE'
p14
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
p14 <- ggplot(counties_CA, aes(x = pop.density, y = democrat)) +
  geom_point() +
  geom_text_repel(aes(label = county))

p14
```

```
## Warning: ggrepel: 38 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
# END SOLUTION
```

```
## Test ##
test_that("p14a", {
  expect_true("ggplot" %in% class(p14))
  print("Checking: ggplot created")
})
```

```
## [1] "Checking: ggplot created"
## Test passed
```

```
## Test ##
test_that("p14b", {
  expect_true(rlang::quo_get_expr(p14$mapping$x) == "pop.density")
  print("Checking: pop.density on x axis")
})
```

```
## [1] "Checking: pop.density on x axis"
## Test passed
```

```
## Test ##
test_that("p14c", {
  expect_true(rlang::quo_get_expr(p14$mapping$y) == "democrat")
  print("Checking: democrat on y axis")
})
```

```
## [1] "Checking: democrat on y axis"
## Test passed
```

```
## Test ##
test_that("p14d", {
  expect_true("GeomPoint" %in% class(p14$layers[[1]]$geom))
  print("Checking: scatterplot added")
})
```

```
## [1] "Checking: scatterplot added"
## Test passed
```

```
## Test ##
test_that("p14e", {
  expect_true("GeomTextRepel" %in% class(p14$layers[[2]]$geom))
  print("Checking: labels added")
})
```

```
## [1] "Checking: labels added"
## Test passed
```

The current issue with these data is that San Francisco (as you can now hopefully point out) has a much higher population density than other counties, and that generally there is a large right skew in the distribution of the population density variable.

If we tried and fit a linear model to these data, it would not fit well— because the relationship between population density and the response variable is not linear. However, this is the perfect situation to try transforming the x variable.

15. [2 points] Try adding a log-transformed version of population density to the data frame and remake your plot using this new variable. Call this new variable `log_pop_density`. Keep the population labels. Also add a smoothed fitted line:

BEGIN QUESTION

name: p15

manual: false

points: 2

```
. = " # BEGIN PROMPT
# uncomment the line below by deleting the pound sign
# counties_CA <- 'Add new variable here'

p15 <- 'YOUR PLOT HERE'
p15
" # END PROMPT

# BEGIN SOLUTION NO PROMPT

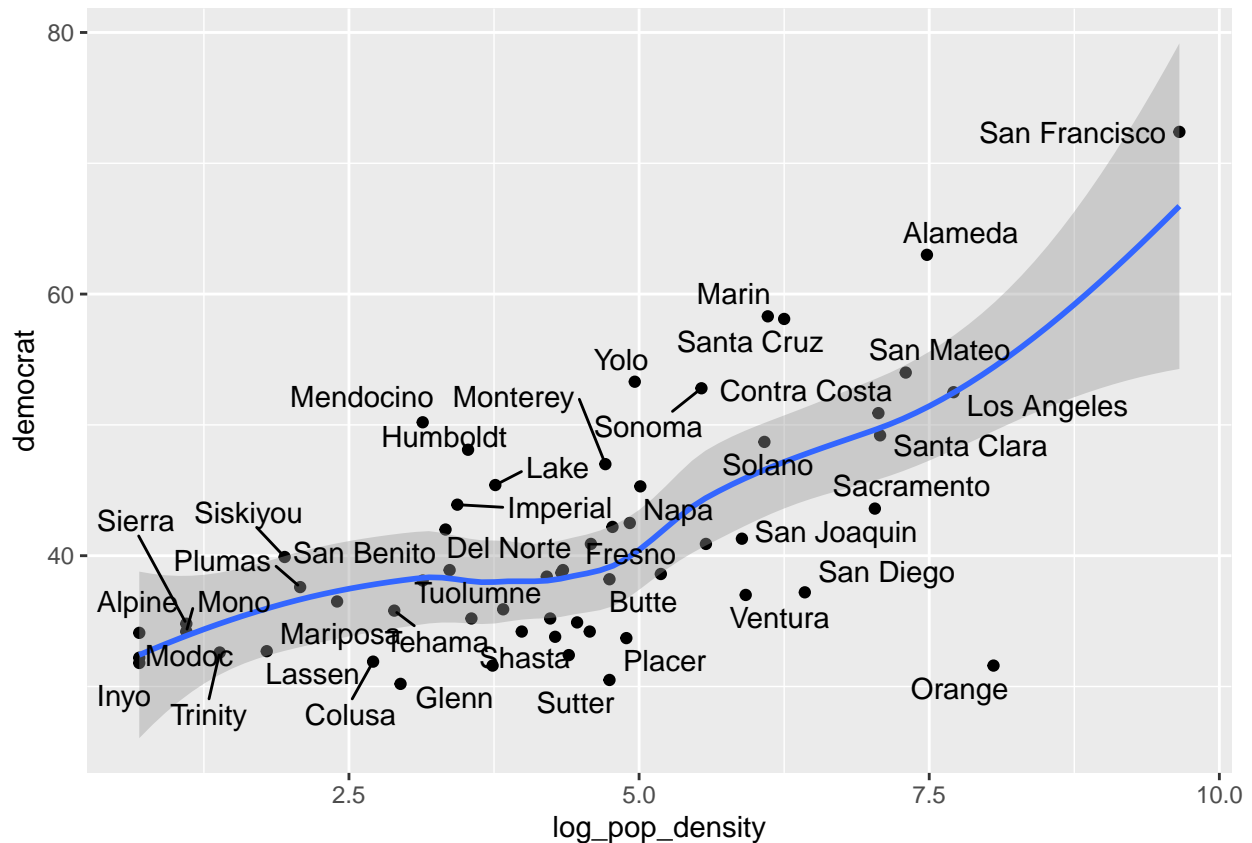
counties_CA <- counties_CA %>%
  mutate(log_pop_density = log(pop.density))

p15 <- ggplot(counties_CA, aes(x = log_pop_density, y = democrat)) +
  geom_point() +
  geom_smooth() +
  geom_text_repel(aes(label = county))

p15
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
# END SOLUTION
```

```
## Test ##
test_that("p15a", {
  expect_true("ggplot" %in% class(p15))
  print("Checking: ggplot created")
})
```

```
## [1] "Checking: ggplot created"
## Test passed
```

```
## Test ##
test_that("p15b", {
  expect_true(rlang::quo_get_expr(p15$mapping$x) == "log_pop_density")
  print("Checking: logged pop density on x axis")
})
```

```
## [1] "Checking: logged pop density on x axis"
## Test passed
```

```
## Test ##
test_that("p15c", {
  expect_true(rlang::quo_get_expr(p15$mapping$y) == "democrat")
  print("Checking: democrat on y axis")
})
```

```
## [1] "Checking: democrat on y axis"
## Test passed
```

```
## Test ##
test_that("p15d", {
  expect_true("GeomPoint" %in% class(p15$layers[[1]]$geom))
  print("Checking: scatterplot added")
})
```

```
## [1] "Checking: scatterplot added"
## Test passed
```

```
## Test ##
test_that("p15e", {
  expect_true("GeomTextRepel" %in% c(class(p15$layers[[3]]$geom), class(p15$layers[[2]]$geom)))
  print("Checking: labels added")
})
```

```
## [1] "Checking: labels added"
## Test passed
```

16. [4 points] Describe the relationship between the (logged) population density and the response variable in terms of the shape, direction, strength, and outliers. These are concepts from Chapter 3. Calculate the correlation (round to 4 decimals) to comment on one of these aspects.

BEGIN QUESTION

name: p16
manual: false
points: 4

```
. = " # BEGIN PROMPT
p16 <- 'CALCULATE THE CORRELATION HERE'
p16
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
p16 <- counties_CA %>% summarise(cor = cor(democrat, log_pop_density))
p16 <- round(p16[[1]],4)
p16
```

```
## [1] 0.6381
```

```
# END SOLUTION
```

```
## Test ##
test_that("p16", {
  expect_true(all.equal(p16, 0.6381187, tolerance = .001))
  print("Checking: value")
})
```

```
## [1] "Checking: value"
## Test passed
```

Direction: There is a positive association between logged population density and the % of votes cast for the democratic candidate is

Shape: Roughly linear, or slightly curved. Outliers: No real large outliers, though SF and Orange County are a bit further out from the rest of the points.

Strength: 0.638

The correlation between the variables is 64%, indicating a moderate positive association.

17. [4 points] Run a linear model regression of the % votes cast for the democratic candidate as a function of the population density. Make sure you get the order of variables right in the `lm()` function! Use the `tidy()` function to show the slope and intercept estimates. Interpret the relationship between the logged population density and the response variable. (You can `View()` the data frame to make sure you are getting your units right by checking the descriptions in the labels for each variable). Use another function from `broom` show the r-squared. Report and interpret this value for the model.

BEGIN QUESTION

name: p17
manual: false
points: 4

```
. = " # BEGIN PROMPT
lm_CA <- 'YOUR MODEL HERE'

r.squared <- 'Report r-squared here. Leave as decimal and round to 2 places'
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
lm_CA <- lm(formula = democrat ~ log_pop_density, data = counties_CA)
tidy(lm_CA)
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>   <dbl>
## 1 (Intercept)    28.4      2.22     12.8 3.10e-18
## 2 log_pop_density  2.88     0.464     6.20 7.12e- 8
```

```
model<- glance(lm_CA)
r.squared <- round(model$r.squared, 3)
r.squared
```

```
## [1] 0.407
```

```
# END SOLUTION
```

```
## Test ##
test_that("p17a", {
  expect_true("lm" %in% class(lm_CA))
  print("Checking: linear model created")
})
```

```
## [1] "Checking: linear model created"
## Test passed
```

```
## Test ##
test_that("p17b", {
  expect_true(all.equal(r.squared, 0.4071955, tolerance = .001))
  print("Checking: r-squared value to at least 3 decimals")
})
```

```
## [1] "Checking: r-squared value to at least 3 decimals"
## Test passed
```

A one unit change in the logged population density is associated with a 2.88 (where population density was the 1992 population per square-mile) percentage point increase in the percent of votes cast for the democratic candidate.

The r-squared is 0.41, implying that 41% of the variation in percentage votes casts is explained by the logged population density.

18. [4 points] Using the code learned in class, that was also shown in Lab 9, make the four plots to examine the assumptions.

BEGIN QUESTION

name: p18

manual: false

points: 4

```
. = " # BEGIN PROMPT
plot1 <- 'Code for scatterplot here'
plot1

plot2 <- 'Code for QQ plot here'
plot2

plot3 <- 'Code for Fitted vs. Residuals plot here'
plot3

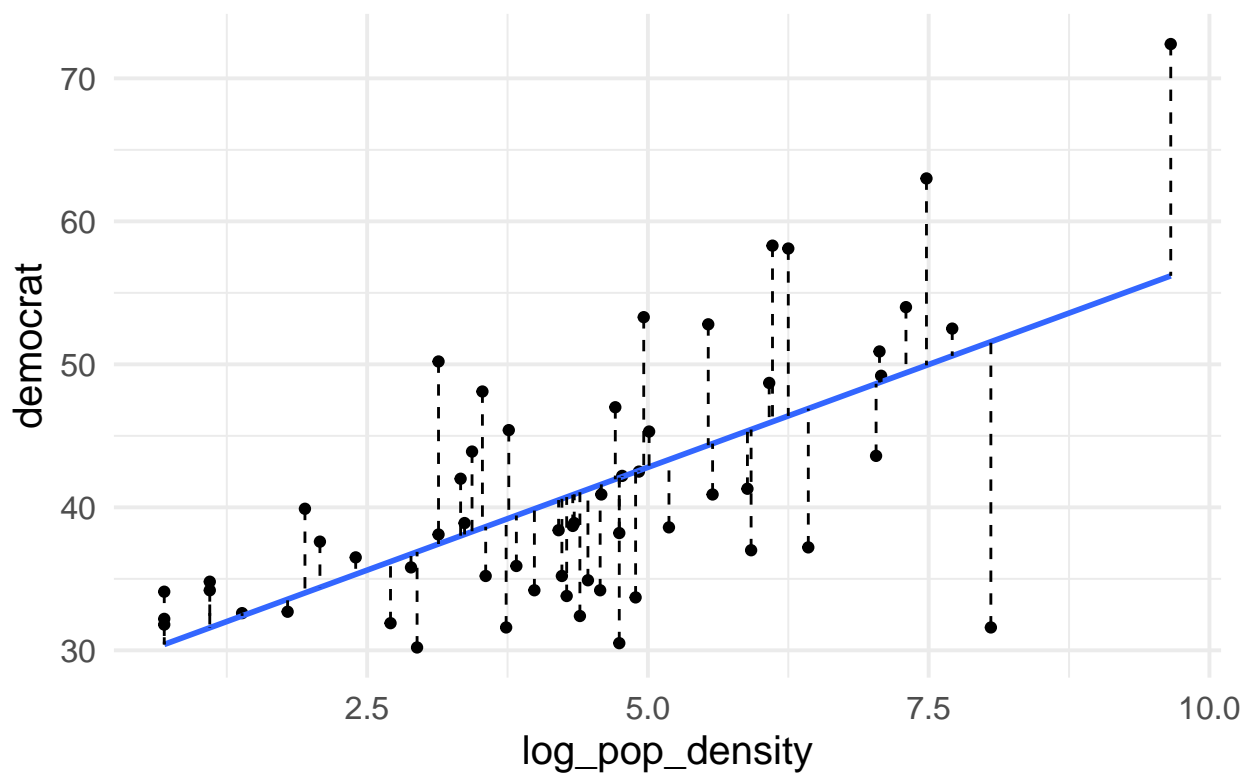
plot4 <- 'Code for Amount explained plot here'
plot4
" # END PROMPT

# BEGIN SOLUTION NO PROMPT
CA_augment <- augment(lm_CA)

# scatter plot
plot1 <- ggplot(CA_augment, aes(y = democrat, x = log_pop_density)) +
  geom_point() +
  geom_smooth(method = "lm", se = F) +
  geom_segment(aes(xend = log_pop_density, yend = .fitted), lty = 2) +
  theme_minimal(base_size = 15) +
  labs(title = "(a) Scatter plot")
plot1
```

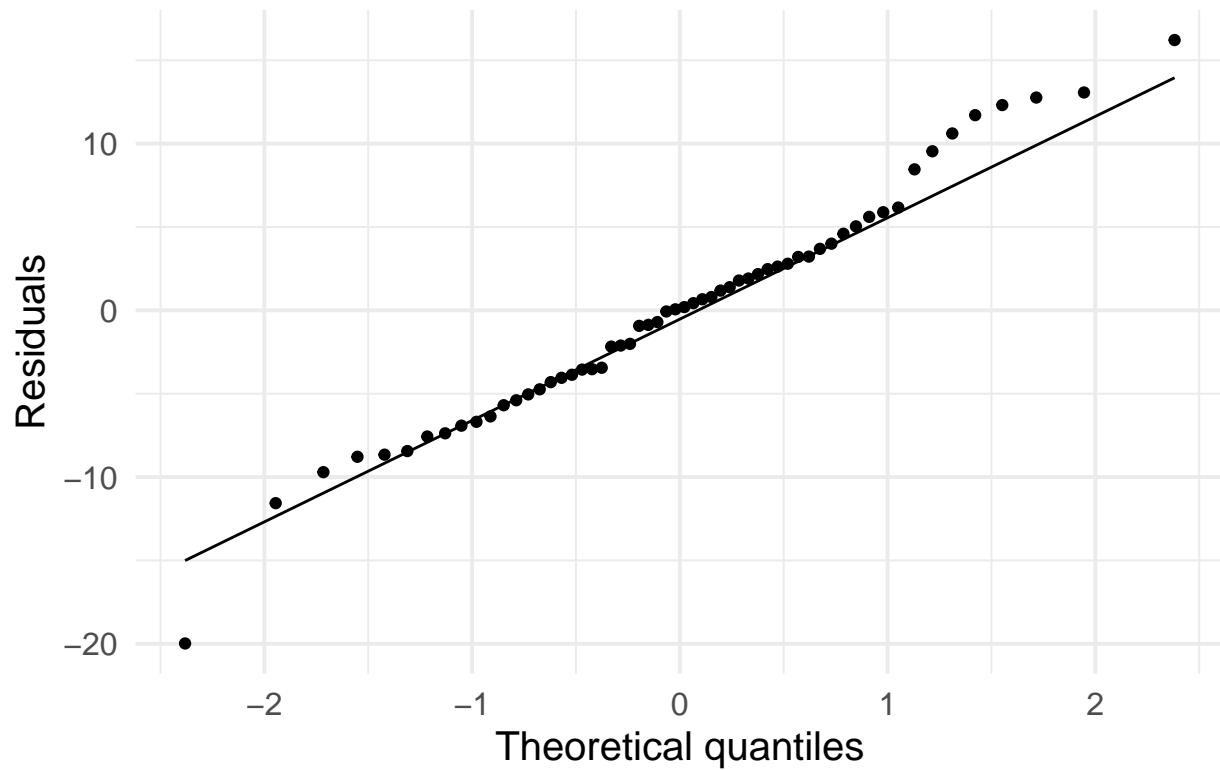
```
## 'geom_smooth()' using formula 'y ~ x'
```

(a) Scatter plot



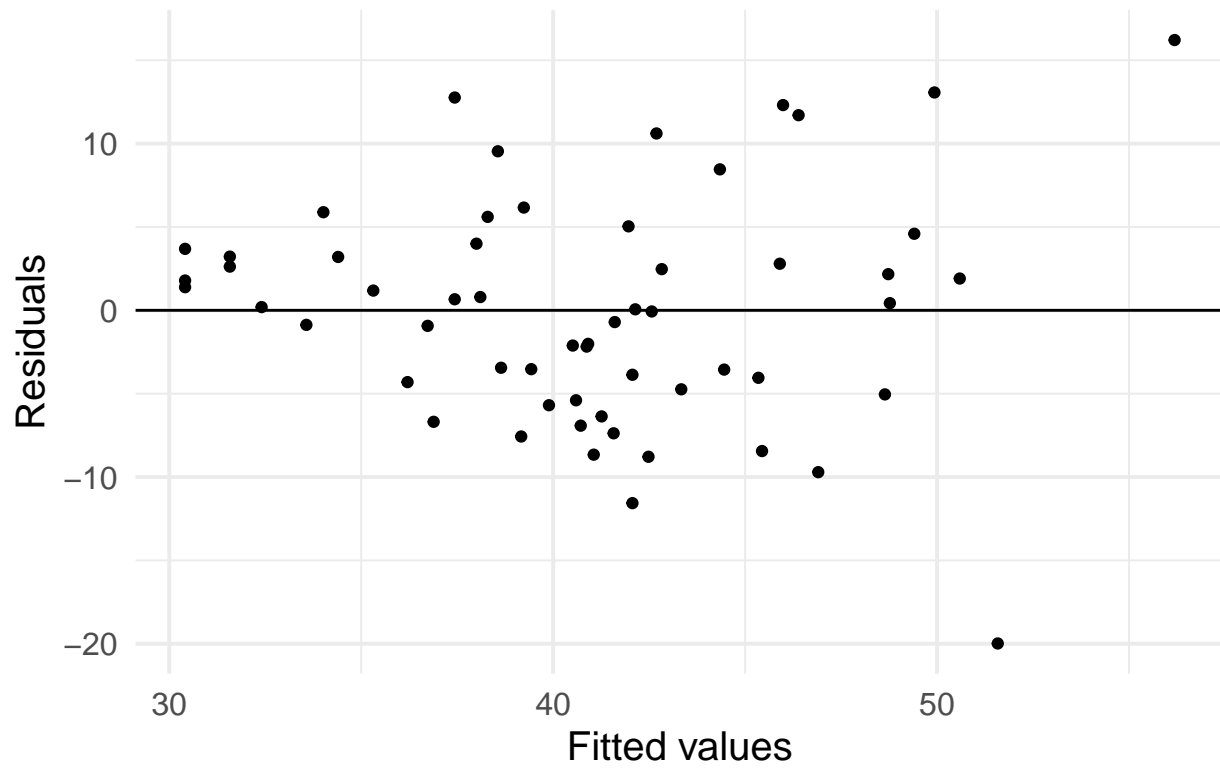
```
# QQ plot
plot2 <- ggplot(CA_augment, aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line() +
  theme_minimal(base_size = 15) +
  labs(y = "Residuals", x = "Theoretical quantiles", title = "(b) QQplot")
plot2
```

(b) QQplot



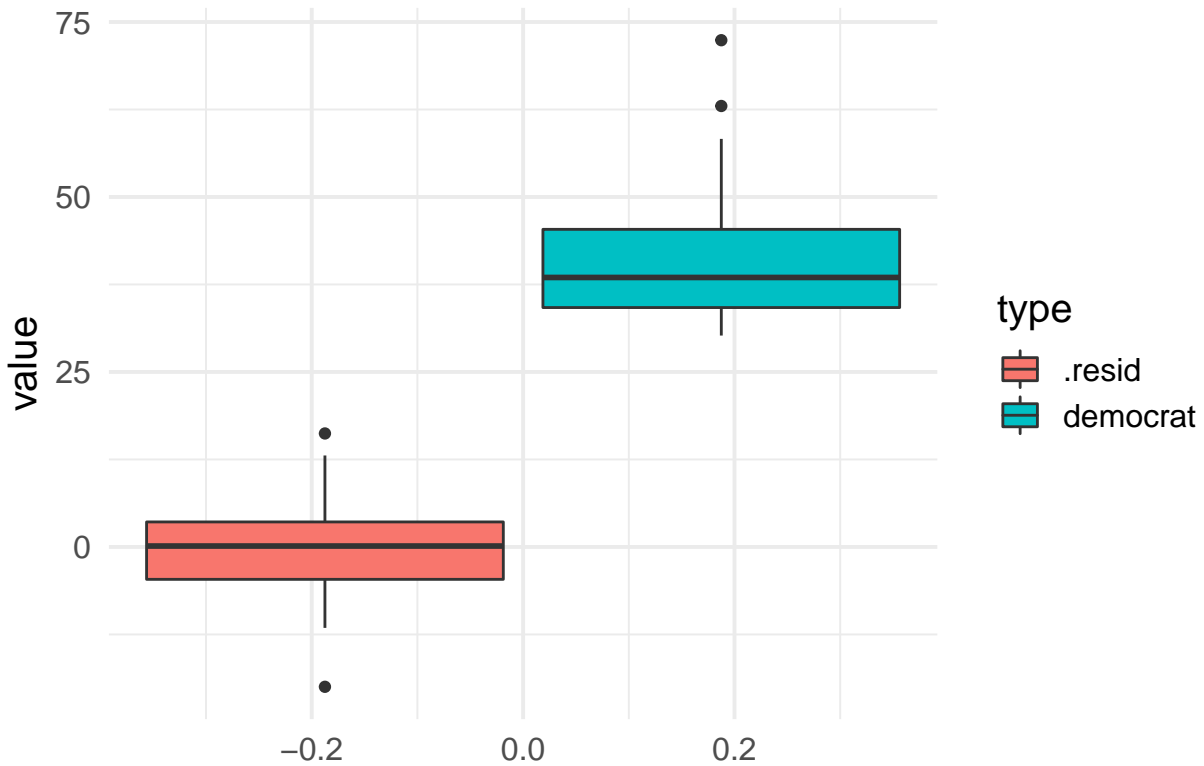
```
## Fitted vs. residuals
plot3 <- ggplot(CA_augment, aes(y = .resid, x = .fitted)) +
  geom_point() +
  theme_minimal(base_size = 15) +
  geom_hline(aes(yintercept = 0)) +
  labs(y = "Residuals", x = "Fitted values", title = "(c) Fitted vs. residuals")
plot3
```


(c) Fitted vs. residuals



```
## Amount explained
bolt_gather <- CA_augment %>% select(democrat, .resid) %>%
  gather(key = "type", value = "value", democrat, .resid)
plot4 <- ggplot(bolt_gather, aes(y = value)) +
  geom_boxplot(aes(fill = type)) +
  theme_minimal(base_size = 15) +
  labs(title = "(d) Amount explained")
plot4
```

(d) Amount explained



END SOLUTION

```
## Test ##
test_that("p18a", {
  expect_true("ggplot" %in% class(plot1))
  print("Checking: ggplot created in plot1")
})
```

```
## [1] "Checking: ggplot created in plot1"
## Test passed
```

```
## Test ##
test_that("p18b", {
  expect_true(rlang::quo_get_expr(plot1$mapping$x) == "log_pop_density")
  print("Checking: logged pop density on x axis of plot1")
})
```

```
## [1] "Checking: logged pop density on x axis of plot1"
## Test passed
```

```
## Test ##
test_that("p18c", {
  expect_true(rlang::quo_get_expr(plot1$mapping$y) == "democrat")
  print("Checking: democrat on y axis of plot1")
})
```

```
## [1] "Checking: democrat on y axis of plot1"
## Test passed
```

```
## Test ##
test_that("p18d", {
  expect_true("GeomPoint" %in% class(plot1$layers[[1]]$geom))
  print("Checking: scatterplot added to plot1")
})
```

```
## [1] "Checking: scatterplot added to plot1"
## Test passed
```

```
## Test ##
test_that("p18e", {
  expect_true("ggplot" %in% class(plot2))
  print("Checking: ggplot created in plot2")
})
```

```
## [1] "Checking: ggplot created in plot2"
## Test passed
```

```
## Test ##
test_that("p18f", {
  expect_true(rlang::quo_get_expr(plot2$mapping$sample) == ".resid")
  print("Checking: residuals plotted as 'sample' in plot2")
})
```

```
## [1] "Checking: residuals plotted as 'sample' in plot2"
## Test passed
```

```
## Test ##
test_that("p18g", {
  expect_true("StatQqLine" %in% class(plot2$layers[[2]]$stat))
  print("Checking: qqplot defined for plot2")
})
```

```
## [1] "Checking: qqplot defined for plot2"
## Test passed
```

```
## Test ##
test_that("p18h", {
  expect_true("GeomPoint" %in% class(plot2$layers[[1]]$geom))
  print("Checking: scatterplot added to plot1")
})
```

```
## [1] "Checking: scatterplot added to plot1"
## Test passed
```

```
## Test ##
test_that("p18i", {
  expect_true("ggplot" %in% class(plot3))
  print("Checking: ggplot created in plot3")
})
```

```
## [1] "Checking: ggplot created in plot3"
## Test passed
```

```
## Test ##
test_that("p18j", {
  expect_true(rlang::quo_get_expr(plot3$mapping$x) == ".fitted")
  print("Checking: fitted values on x axis of plot3")
})
```

```
## [1] "Checking: fitted values on x axis of plot3"
## Test passed
```

```
## Test ##
test_that("p18k", {
  expect_true(rlang::quo_get_expr(plot3$mapping$y) == ".resid")
  print("Checking: residuals on y axis of plot3")
})
```

```
## [1] "Checking: residuals on y axis of plot3"
## Test passed
```

```
## Test ##
test_that("p18l", {
  expect_true("GeomPoint" %in% class(plot3$layers[[1]]$geom))
  print("Checking: scatterplot added to plot3")
})
```

```
## [1] "Checking: scatterplot added to plot3"
## Test passed
```

```
## Test ##
test_that("p18m", {
  expect_true("ggplot" %in% class(plot4))
  print("Checking: ggplot created in plot4")
})
```

```
## [1] "Checking: ggplot created in plot4"
## Test passed
```

```
## Test ##
test_that("p18n", {
  expect_true("GeomBoxplot" %in% class(plot4$layers[[1]]$geom))
  print("Checking: boxplot created for plot4")
})
```

```
## [1] "Checking: boxplot created for plot4"
## Test passed
```

19. [4 points] Comment on each of the plots and conclude about which assumptions appear violated vs. not violated. Don't forget to comment on the one assumption that cannot be investigated using plots.

```
BEGIN QUESTION
name: p19
manual: true
```

- There is a violation of the assumption that the standard deviation of the response variable are identical for all values of the explanatory variable. We see here that as the log population density increases, the residuals become larger.
- The relationship between X and Y is approximately linear, though there is a lot of variation around the line of best fit. (This points to the fact that though population density is predictive of the response variable, there are other factors that are not included in our model that have further predictive power.)
- The QQ plot looks ok. (May be interpreted as a problem with the largest residuals.
- We can't check the assumption that the points are independent using this plot.) Here that corresponds to the counties being independent of one another. This model treats them as independent units. [More sophisticated models can take the spatial relationships between the counties into account (to account for the fact that counties closer to each other may be more similar.)]