Problem set 8: Regression

Due November 20, 2023, at 9pm

(Your name here)

NOTE: Start with the file ***ps8_2023_regression.qmd*** *(available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F23/tree/main/assignments). Modify that file to include your answers. Make sure you can "render" the file (e.g. in RStudio by clicking on the* ***Render*** *button). Submit both the qmd file and the PDF via Canvas.*

The dataset `brexit_data_gb_subset.csv` (available on the github under data) contains results of the 2016 UK Brexit referendum by local authority (cities, counties, etc), collected from the Electoral Commission website and 2011 census data by Claire Peacock.

```
brexit <- read.csv("./../../data/brexit_data_gb_subset.csv")
library(tidyverse)
```

Question 1

(1a) Having loaded the data, use `lm()` to regress `Percent_Leave` (the support for Brexit in the local authority) on `Region` (the region in which the local authority is located). Present a regression table using `modelsummary::modelsummary()` or a similar approach.

Answer:

```
oc_reg <- lm(Percent_Leave ~ Region, data = brexit)
modelsummary::modelsummary(oc_reg, gof_map = c("nobs", "r.squared"))
```

(1b) In the regression you just presented, what does the `(Intercept)` coefficient mean, i.e. what does that number signify?

Answer: It indicates the average support for Brexit across local authorities in the omitted region, which is "East".

(1c) Using the regression output above, report and interpret a 95% confidence interval for the coefficient on `RegionLondon`.

	(1)
(Intercept)	56.963
	(1.207)
RegionEast Midlands	2.612
	(1.780)
RegionLondon	−17.872
	(1.880)
RegionNorth East	2.515
	(2.677)
RegionNorth West	−1.048
	(1.793)
RegionScotland	−17.826
	(1.897)
RegionSouth East	−4.792
	(1.575)
RegionSouth West	−4.584
	(1.805)
RegionWales	−3.615
	(2.138)
RegionWest Midlands	3.352
	(1.934)
RegionYorkshire and The Humber	1.686
	(2.172)
Num.Obs.	381
R2	0.418

Answer: This coefficient measures the difference in average support for Brexit between local authorities in London and those in the East region. A 95% confidence interval for that difference is

```
coef(oc_reg)["RegionLondon"] + c(-1,1)*1.96*
  summary(oc_reg)$coefficients["RegionLondon", "Std. Error"]
```

```
[1] -21.55590 -14.18806
```

(1d) Using the regression above, what is the estimated average support for Brexit in London local authorities?

Answer: The estimated support for Brexit in London local authorities is the sum of the `(Intercept)` term and the `RegionLondon` term, i.e.

```
sum(coef(oc_reg)[c("(Intercept)", "RegionLondon")])
```

```
[1] 39.0907
```

(1e) For this question you will produce several different estimates for the standard error of the estimated average support for Brexit in London local authorities.

(1e.1) Compute the standard error using the regression above (hint: use variance rule and `vcov()`)

Answer

```
(se_london_lm <- sqrt(sum(vcov(oc_reg)[c("(Intercept)", "RegionLondon"),
                                       c("(Intercept)", "RegionLondon")])))
```

```
[1] 1.440649
```

(1e.2) Obtain the standard error from the output of the regression of `Percent_Leave` on `Region` with no intercept (hint: add `-1` to the `formula` argument)

Answer

```r
summary(lm(Percent_Leave ~ Region -1,
           data = brexit))$coefficients["RegionLondon", "Std. Error"]
```

```
[1] 1.440649
```

(1e.3) Compute the standard error of the sample mean for London local authorities, i.e. with no regression

Answer

```r
sqrt(var(brexit$Percent_Leave[brexit$Region == "London"])/
       sum(brexit$Region == "London"))
```

```
[1] 2.222091
```

(1e.4) Compute the standard error as you did in (1e.1), except this time use `estimatr::lm_robust()` for the regression, which uses the "sandwich estimator" (Huber-White standard errors) and thus does not assume homoskedasticity

Answer

```r
oc_reg_robust <- estimatr::lm_robust(Percent_Leave ~ Region, data = brexit)
(se_london_lm_robust <- sqrt(sum(vcov(oc_reg_robust)[
  c("(Intercept)", "RegionLondon"), c("(Intercept)", "RegionLondon")])))
```

```
[1] 2.222091
```

(1e.5) What does homoskedasticity mean in this case? Check whether it appears to be valid using the regions of `London`, `East`, and `Scotland` as examples.

Answer

In general, homoskedasticity means that the variance of Y does not vary with X. In this case it would mean that the variance of `Percent_Leave` is the same across regions.

These three regions show that homoskedasticity is not a valid assumption in this case:

```
  var(brexit$Percent_Leave[brexit$Region == "London"])
```

```
[1] 162.9437
```

```
  var(brexit$Percent_Leave[brexit$Region == "East"])
```

```
[1] 91.35564
```

```
  var(brexit$Percent_Leave[brexit$Region == "Scotland"])
```

```
[1] 32.07229
```

We can show this more comprehensively with this code:

```
  brexit |>
    group_by(Region) |>
    summarize(region_var = var(Percent_Leave))
```

```
# A tibble: 11 x 2
   Region                   region_var
   <chr>                         <dbl>
 1 East                           91.4
 2 East Midlands                  51.2
 3 London                        163.
 4 North East                     35.4
 5 North West                     46.2
 6 Scotland                       32.1
 7 South East                     59.0
 8 South West                    101.
 9 Wales                          30.2
10 West Midlands                  44.2
11 Yorkshire and The Humber       62.6
```

Essentially, the classical standard errors ignore the fact that the variance of the outcome varies across regions, so the estimated standard error for the predicted outcome in London reflects the average variance in outcomes across regions, not the variance specific to London. In this

simple case, the standard error we want the regression to produce is really the standard error of the sample mean, and the robust standard errors give us that.

Question 2

(2a) Regress `Percent_Leave` on `Bachelors_deg_percent` (the percent of local authority inhabitants who have at least a bachelors degree) and `Birth_UK_percent` (the percent of local authority inhabitants who were born in the UK). Interpret all of the coefficients.

Answer

```
reg2 <- lm(Percent_Leave ~ Bachelors_deg_percent +
             Birth_UK_percent, data = brexit)
coef(reg2)
```

```
        (Intercept) Bachelors_deg_percent      Birth_UK_percent
         65.7146247            -0.9875180             0.1742402
```

An increase in 1 percentage point in the share of inhabitants with a bachelors degree is associated with a decrease of .988 percentage points in support for Leave, holding fixed the proportion of inhabitants born in the UK.

An increase in 1 percentage point in the proportion of inhabitants born in the UK is associated with an increase of .174 percentage points in support for Leave, holding fixed the share of inhabitants with a bachelors degree.

The intercept is not really meaningful on its own – it indicates that a local authority where none of the inhabitants have a bachelors degree or are born in the UK is predicted to have support for Leave of 65.7%, but this isn't very useful because there is no such place.

(2b) Use the bootstrap (with $m = 1000$ resamples) to compute standard errors for your coefficients. You should store the m sets of coefficients in a matrix with m rows and 3 columns.

Answer

```
# getting subset of dataset
n <- nrow(brexit)
m <- 1000
sto <- matrix(NA, nrow = m, ncol = 3)
colnames(sto) <- c("Intercept", "Bachelors", "Born_in_UK")
for(i in 1:m){
```

```
  this_dat <- brexit[sample(1:n, size = n, replace = T), ]
  this_lm <- lm(Percent_Leave ~ Bachelors_deg_percent +
                  Birth_UK_percent, data = this_dat)
  sto[i, ] <- coef(this_lm)
}
```

Here are three ways to get standard errors from the bootstrap coefficients:

```
# standard deviation, column by column
c(sd(sto[,1]), sd(sto[,2]), sd(sto[,3]))
```

```
[1] 3.37336466 0.04413387 0.03124496
```

```
# applying the function `sd` to the 2nd dimension (the columns) of `sto`
apply(sto, 2, sd)
```

```
 Intercept  Bachelors Born_in_UK
3.37336466 0.04413387 0.03124496
```

```
# the square root of the diagonal of the variance covariance matrix
(bootstrap_ses <- sqrt(diag(cov(sto))))
```

```
 Intercept  Bachelors Born_in_UK
3.37336466 0.04413387 0.03124496
```

(2c) Compare the standard errors you compute above to the classical standard errors generated by `lm()` and the robust (Huber-White) standard errors generated by `estimatr::lm_robust()`.

Answer

The table below shows the three sets of standard errors:

```
reg2_robust <- estimatr::lm_robust(Percent_Leave ~ Bachelors_deg_percent +
                                     Birth_UK_percent, data = brexit)
se_mat <- rbind(bootstrap_ses, summary(reg2)$coefficients[,"Std. Error"],
                summary(reg2_robust)$coefficients[,"Std. Error"])
rownames(se_mat) <- c("Bootstrap", "Classical", "Robust")
```

```
se_mat
```

```
           Intercept  Bachelors Born_in_UK
Bootstrap  3.373365 0.04413387 0.03124496
Classical  2.990596 0.03498557 0.02654691
Robust     3.416593 0.04752250 0.03087709
```

The robust standard errors and the bootstrap standard errors are very similar; the classical standard errors are lower. This suggests there is heteroskedasticity in the data.

(2d) Report the variance-covariance matrix of your bootstrap coefficient estimates. Interpret the bottom left entry of the matrix; what explains its sign? Confirm that it is similar to the variance covariance matrix you obtain by running the regression with `estimatr::lm_robust()`.

Answer

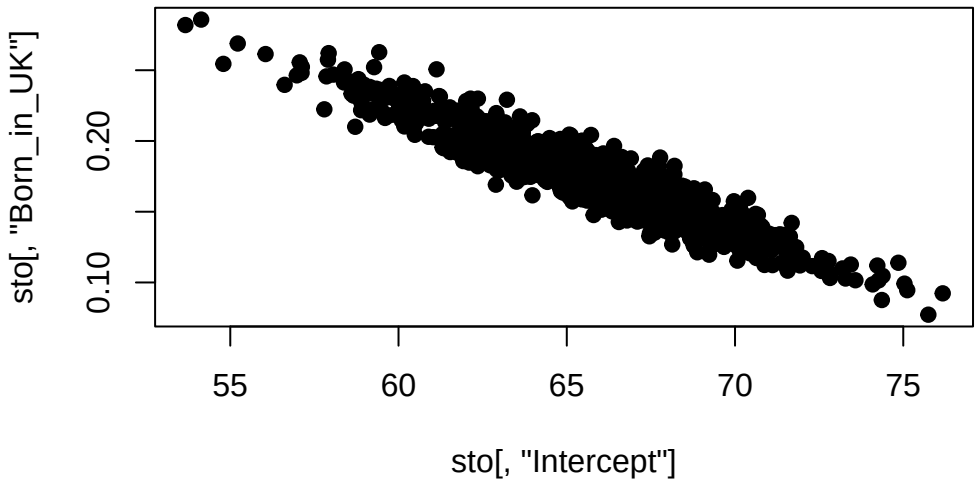
This is the variance-covariance matrix:

```
cov(sto)
```

```
              Intercept     Bachelors    Born_in_UK
Intercept  11.37958914 -0.0871247319 -0.0991081640
Bachelors  -0.08712473  0.0019477988  0.0003879149
Born_in_UK -0.09910816  0.0003879149  0.0009762477
```

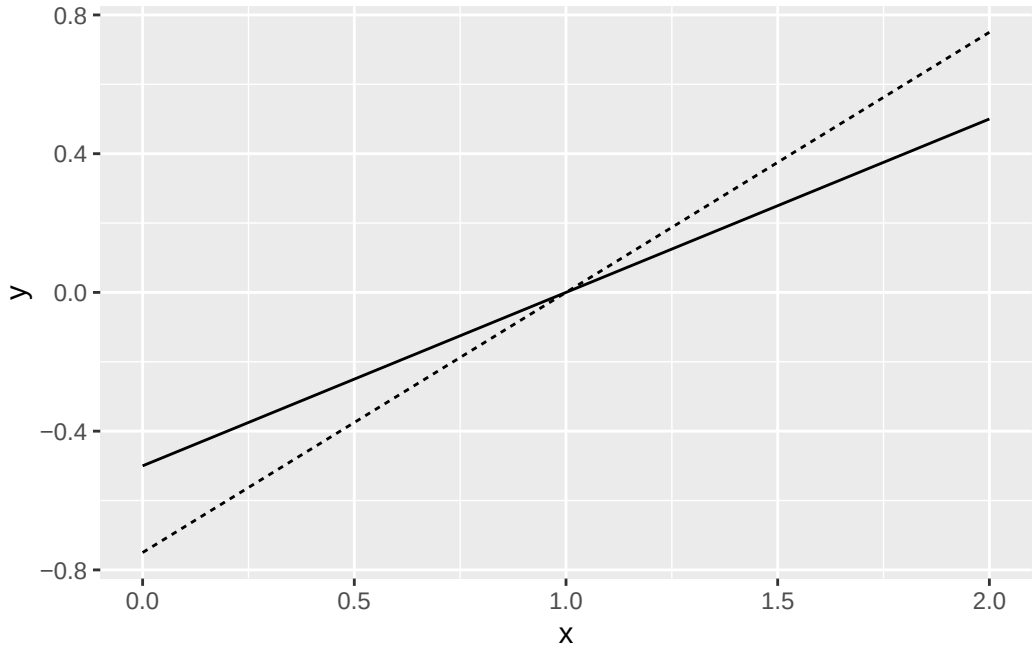
The bottom-left entry indicates that the covariance between the coefficient estimates for the proportion born in the UK and the coefficient estimates for the intercept is about -.096. You can plot this to see the negative covariance:

```
plot(sto[,"Intercept"], sto[, "Born_in_UK"], type = "p", pch = 19)
```



The reason for the negative covariance is that in samples where the slope on `Birth_UK_percent` is larger, the intercept tends to be smaller (and vice versa). Similarly, in the illustration below, the line with the lower slope (the solid one) has a higher intercept.

```
expand_grid(x = seq(-1, 1, length = 100),
            case = c("More steep", "Less steep")) |>
  mutate(y = ifelse(case == "More steep", x*.75, x*.5),
         x = x + 1) |>
  ggplot(aes(x, y, lty = case)) +
  geom_line(show.legend = F)
```


0.8
0.4
0.0
−0.4
−0.8
y
0.0
0.5
1.0
1.5
2.0
x