Problem Set #5

EH6105 - Quantitative Methods

Steven V. Miller

This homework makes use of data available in `{stevedata}` and implies the use of `{tidyverse}` to answer the questions. It will also require some functions available in `{stevemisc}`, `{lmtest}` and one of `{fixest}` or `{modelr}`. The final question in this lab script will be a "choose your adventure" for functionality included in either `{fixest}` or `{modelr}`. `{tidyverse}` is not necessary to answer these questions though it will assuredly make the process easier (especially if you want to use the bootstrap approach for the last question). Load these libraries to get started answering these questions.

```
library(tidyverse)
library(stevedata)
library(stevemisc)
library(lmtest)
# library(fixest)
# library(modelr)
```

Attitudes Toward National Spending in the General Social Survey (2018)

This homework will refer to the `gss_spending` data set that is available in `{stevedata}`. I created this data set a few years ago largely for some illustration of ordinal modeling, but it has a decent application here for diagnostics of the ordinary least squares model. The data come from a 2018 wave of the General Social Survey in the United States and probe a few thousand Americans about what they think about national spending on various public goods. You can find out more information about the data by visiting this part of the package's website, or with the following command.

```
?gss_spending
```

Here's a little preview of these data. Admittedly, there are a lot of columns here and we're going to focus on just a handful of them.

```
gss_spending
```

```
## # A tibble: 2,348 x 33
##     year    id   age   sex  educ degree  race rincom16 partyid polviews xnorcsiz
##    <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
##  1  2018     1    43     0    14      2     1       NA       5        6        6
##  2  2018     2    74     1    10      1     1       NA       2       NA        6
##  3  2018     3    42     0    16      3     1       22       4        5        6
```

```
##  4  2018     4    63     1    16      3     1       23       2        4        6
##  5  2018     5    71     0    18      4     2       NA       6        7        6
##  6  2018     6    67     1    16      3     1       NA       2        3        6
##  7  2018     7    59     1    13      1     2       12       0        4        2
##  8  2018     8    43     0    12      1     1       17       5        5        2
##  9  2018     9    62     1     8      0     1        2       3        4        2
## 10  2018    10    55     0    12      1     1       22       1       NA        4
## # ... with 2,338 more rows, and 22 more variables: news <dbl>, wrkstat <dbl>,
## #   natspac <dbl>, natenvir <dbl>, natheal <dbl>, natcity <dbl>,
## #   natcrime <dbl>, natdrug <dbl>, nateduc <dbl>, natrace <dbl>, natarms <dbl>,
## #   nataid <dbl>, natfare <dbl>, natroad <dbl>, natsoc <dbl>, natmass <dbl>,
## #   natpark <dbl>, natchld <dbl>, natsci <dbl>, natenrgy <dbl>, sumnat <dbl>,
## #   sumnatsoc <dbl>
```

Answer these questions. A successful answer of these question must include the R code you used to help you answer the question.

1. This is less of a question and more of a command. I want you to create two new variables based on the data I present here, along with the codebook that describes them. One, create a new variable—call it `black`—that is a dummy variable if the respondent says their race is black (and not white or "other"). Next: I want you to address a coding problem in the `partyid` variable. Create another variable—call it `pid`—that removes the other party supporters (by making them `NA`s) or find some other way to remove—or "filter (out)", if you will—those observations from the data. As always, read the codebook.
2. Run an OLS model that regresses the `sumnatsoc` variable on the respondent's age, sex, years of education, their ideology ("political views"), the dummy variable for black respondents you created, and the adjusted partisanship variable you also created (or otherwise modified from the original data). Explain the results to me.[1]
3. Let's check for the linearity of the model. Run the `linloess_plot()` function from `{stevemisc}` on the regression model you ran and tell me if you think there is evident of non-linearity in the model.[2] Explain your answer. Tell me what you see.
4. Run a Durbin-Watson test and Breusch-Godfrey test for autocorrelation from this model and tell me what you find.
5. (2 POINTS) Let's check for potential heteroskedasticity in the regression model. Run two different diagnostics for detecting potential heteroskedasticity in the regression model and tell me what you find. Is there evidence of heteroskedasticity in the regression model?
6. (2 POINTS) Run a weighted least squares version of this regression model and compare the results to regular OLS. What differences do you see?
7. (2 POINTS) Run one other heteroskedasticity correction and report the results back to me. You can either use the functionality included in `{fixest}` (the easier option, perhaps), or you can go hardcore with the bootstrapping approach. You'll want the `{modelr}` approach for that.

[1] Here's a regression modeling pro-tip in R. add `na.action = na.exclude` as an additional argument to the `lm()` command after the formula. This is going to be useful for when you have missing observations in the data that you want to ignore. It's not necessary, but it's useful.

[2] `{car}` has a wonderful `residualPlots()` function that includes a Tukey test for non-additivity. You may want to check that out too.