Week 5 Exercises

Katz

2/13/2021

There are two exercises for practicing regression in this rmd file. The first involves using two data sets provided to you. The second exercise involves generating your own data as part of the process of gaining some intuition about what kind of underlying data generating process (there's that term from week 2!) might be creating the data you actually observe in your sample.

In the first exercise, there are instructions outlining each step to follow. The basic building blocks of the code are provided (commented out). There are also three hashtags (###) in certain places, which you should replace with your own values.

Exercise 1 - Teacher salary linear regression demo

First, read in the data. In this case, the two CSV files are stored in a folder called "data". Be sure to adjust this path however you need.

```
file_path_prin <- "./data/principalSalaries.csv"
file_path_tea <- "./data/teacherSalaries.csv"

principal_salaries <- read_csv(file_path_prin)
```

```
## Rows: 208 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): div_num, div_name, FY2007P, FY2010P
## dbl (14): FY2005P, FY2006P, FY2008P, FY2009P, FY2011P, FY2012P, FY2013P, FY2...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
teacher_salaries <- read_csv(file_path_tea)
```

```
## Rows: 206 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): div_num, div_name
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Your code here to filter and select the two columns
```

Try to filter out the public schools and just pick out the average salaries in the 2014-2016 year range (hint: this is stored as "Av14_16P" for principals.) and div_num column.

```
## Your code here
```

Convert the div_num column to numeric

```
## Your code here for filtering, selecting, and converting data type
```

Repeat the same two steps above for teacher salaries (filter public schools, pick out the 2014-2016 salaries (using select()), and convert the div_num column to a numeric data type (using something like as.numeric())

```
#combined_salaries <- inner_join(###, ###, by=###)
```

Join the teacher salaries and principal salaries into one dataframe by using an inner_join. You should think about which column to join on (i.e., which to pass to the by = " " argument within inner_join()) This section helps make tables for formatting in r or printing to csv file to load into excel etc. You should be able to run this by uncommenting the code in the following block.

```
#making_a_table <- describe(combined_salaries$Av14_16P) %>%
#  select(mean,sd,skew,kurtosis)
```

This is an example of making a table in R markdown. You should be able to run this by uncommenting the code in the following block.

```
#kable(making_a_table) %>%
#  kable_styling("striped", full_width = F)
```

Let's transition to running a linear model (simple regression) with the principal salaries as the outcome variable and the teacher salaries as the predictor.

```
#fit1 <- lm(### ~ ###, data= ###)

#summary(###)
```

Create a linear model using the lm() function that models principal salary as a function of teacher salary (i.e., principal salary is the outcome variable). Replace the hashtags below. Plot the principal salaries against the teacher salaries. Think about what kind of plot makes the most sense.

```
### Your code here
```

Example Solution

Teacher salary linear regression demo

```
file_path_prin <- "./data/principalSalaries.csv"
file_path_tea <- "./data/teacherSalaries.csv"

principal_salaries <- read_csv(file_path_prin)
```

```
## Rows: 208 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): div_num, div_name, FY2007P, FY2010P
## dbl (14): FY2005P, FY2006P, FY2008P, FY2009P, FY2011P, FY2012P, FY2013P, FY2...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
teacher_salaries <- read_csv(file_path_tea)
```

```
## Rows: 206 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): div_num, div_name
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
public_schools_principal_salaries <- principal_salaries %>%
  filter(str_detect(div_name,"Public")) %>%
  select(div_num,Av14_16P)
```

convert the div_num column to numeric

```
public_schools_principal_salaries$div_num <- as.numeric(public_schools_principal_salaries$div_num)
```

```
public_schools_teacher_salaries <- teacher_salaries %>%
  filter(str_detect(div_name,"Public")) %>%
  select(div_num,Av14_16T)
```

	mean	sd	skew	kurtosis
X1	85937.95	14598.03	1.29433	2.926321

```
public_schools_teacher_salaries$div_num <- as.numeric(public_schools_teacher_salaries$div_num)
```

```
combined_salaries <- inner_join(public_schools_teacher_salaries,
                                public_schools_principal_salaries,
                                by="div_num")
```

This section helps make tables for formatting in r or printing to csv file to load into excel etc

```
making_a_table <- describe(combined_salaries$Av14_16P) %>%
  select(mean,sd,skew,kurtosis)
```

```
kable(making_a_table) %>%
  kable_styling("striped", full_width = F)
```

let's talk about running a linear model (simple regression) with the teacher

salary data

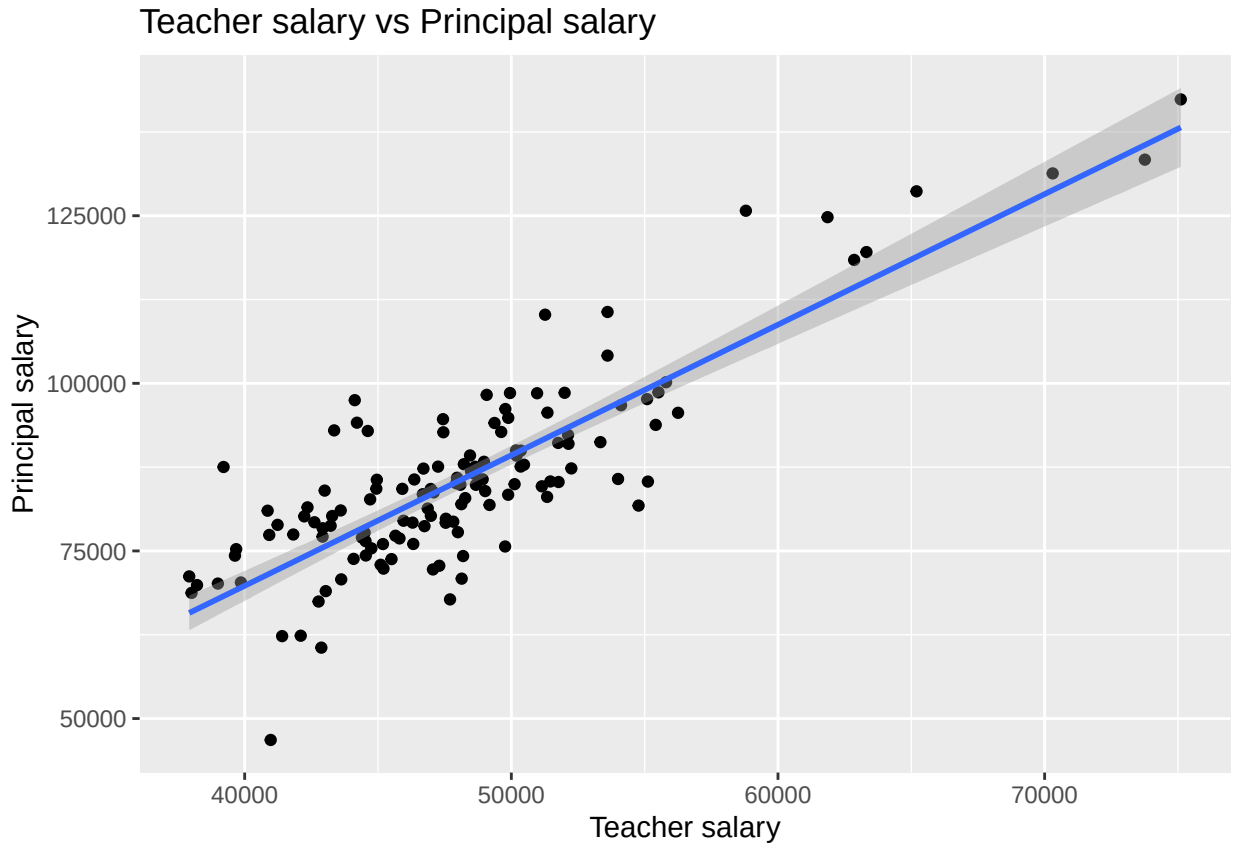
```
fit1 <- lm(Av14_16P ~ Av14_16T, data=combined_salaries)
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = Av14_16P ~ Av14_16T, data = combined_salaries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24895.7  -4590.5   -286.6   4370.2  19646.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8100.0583  5255.5665  -1.541    0.126
## Av14_16T        1.9475     0.1079  18.045   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7827 on 130 degrees of freedom
## Multiple R-squared:  0.7147, Adjusted R-squared:  0.7125
## F-statistic: 325.6 on 1 and 130 DF,  p-value: < 2.2e-16
```

```
combined_salaries %>%
  ggplot(aes(x = Av14_16T,y = Av14_16P)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(x = "Teacher salary",
       y = "Principal salary",
       title = "Teacher salary vs Principal salary")
```

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



Exercise 2 - Create Your Own Adventure

For this exercise, first generate your own data by creating a series of outcome variable values and a series of predictor variable values. For each outcome you should have a predictor (and vice versa). That's just a long way of saying if you want to simulate having 50 observations then you should have 50 outcomes and 50 predictors.

You can generate your data in R using some of the functions we have used for generating data in demos (e.g., rep(), sample(), seq(), rnorm()) or manually in Excel, saving that CSV files, and importing it to R. A particularly useful way to do this might be to imagine your own research topic of interest and thinking of a setting where you might collect data yourself. Then imagine what those data might look like and generate an example data set based on that. These kinds of data simulation exercises are helpful for thinking about what you might expect to see in practice when you are actually collecting real data for your future projects.

Once you have generated this data set, try running a simple regression. This means that you should probably have a continuous outcome (no probably about that part) and a continuous predictor (this is a little more flexible, but for now it might be easiest to stick with this).

Step 1: Generating data

```
# Your code here
```

Steps 2: Importing and tidying data (if needed)

```
# Your code here
```

Step 3: Linear Model

```
# Your code here
```

Step 4: Plot the data

```
# Your code here
```