Introduction to linear regression

Kenan Sooklall

The Human Freedom Index is a report that attempts to summarize the idea of “freedom” through a bunch of different variables for many countries around the globe. It serves as a rough objective measure for the relationships between the different types of freedom - whether it’s political, religious, economical or personal freedom - and other social and economic circumstances. The Human Freedom Index is an annually co-published report by the Cato Institute, the Fraser Institute, and the Liberales Institut at the Friedrich Naumann Foundation for Freedom.

In this lab, you’ll be analyzing data from Human Freedom Index reports from 2008-2016. Your aim will be to summarize a few of the relationships within the data both graphically and numerically in order to find which variables can help tell a story about freedom.

Getting Started

Load packages

In this lab, you will explore and visualize the data using the **tidyverse** suite of packages. The data can be found in the companion package for OpenIntro resources, **openintro**.

Let’s load the packages.

```
library(tidyverse)
library(openintro)
data(’hfi’, package=’openintro’)
```

The data

The data we’re working with is in the openintro package and it’s called `hfi`, short for Human Freedom Index.

Exercise 1

1. What are the dimensions of the dataset?

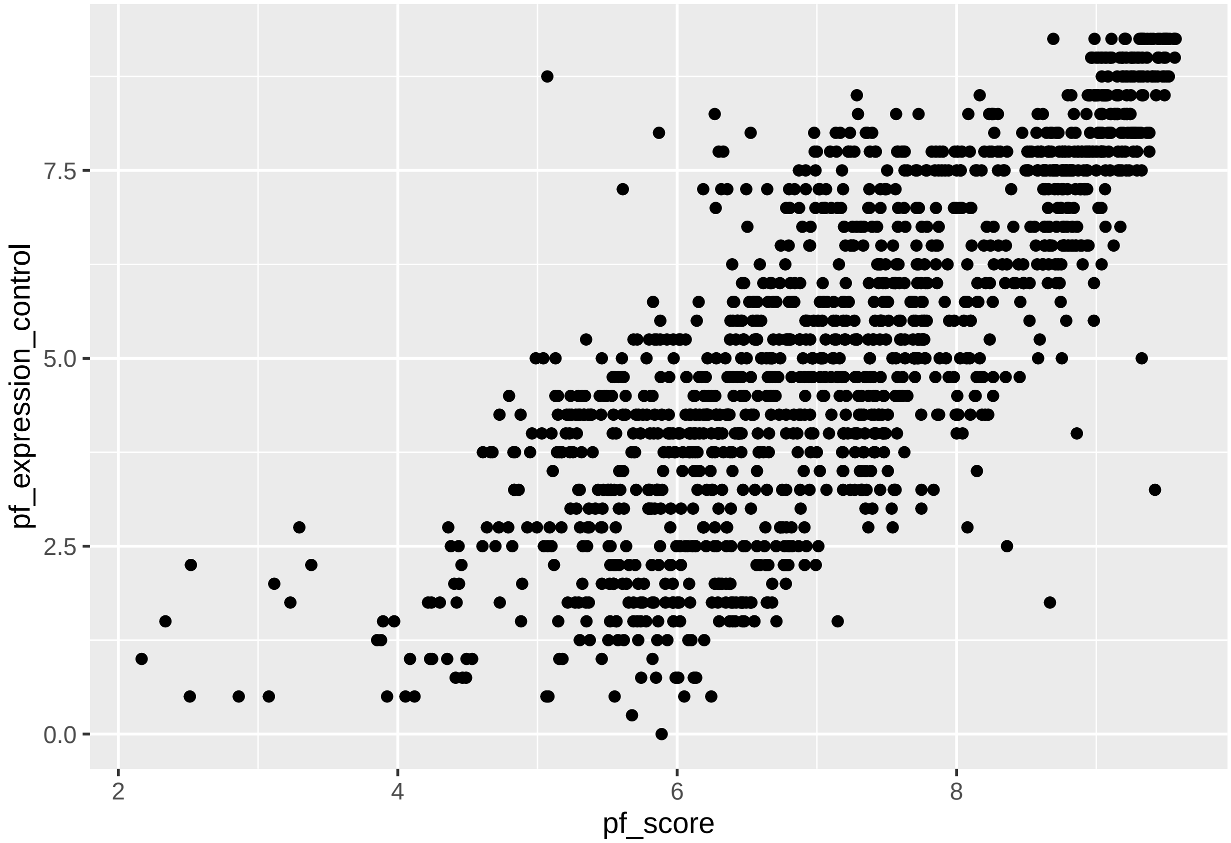
The Human Freedom Index (hfi) has 1458 rows and 123 columns.

Exercise 2

2. What type of plot would you use to display the relationship between the personal freedom score, `pf_score`, and one of the other numerical variables? Plot this relationship using the variable `pf_expression_control` as the predictor. Does the relationship look linear? If you knew a country’s `pf_expression_control`, or its score out of 10, with 0 being the most, of political pressures and controls on media content, would you be comfortable using a linear model to predict the personal freedom score?

A scatter plot shows the pf_score increases with the pf_expression_control.

```
hfi %>% select(c(pf_score, pf_expression_control)) %>% ggplot(aes(x=pf_score, pf_expression_control)) +
```



I would be comfortable giving a range for the pf_score not a specifc number.

If the relationship looks linear, we can quantify the strength of the relationship with the correlation coefficient.

```
hfi %>%
  summarise(cor(pf_expression_control, pf_score, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(pf_expression_control, pf_score, use = "complete.obs")`
##                                                           <dbl>
## 1                                                         0.796
```

Here, we set the use argument to "complete.obs" since there are some observations of NA.

Sum of squared residuals

In this section, you will use an interactive function to investigate what we mean by "sum of squared residuals". You will need to run this function in your console, not in your markdown document. Running the function also requires that the hfi dataset is loaded in your environment.

Think back to the way that we described the distribution of a single variable. Recall that we discussed characteristics such as center, spread, and shape. It's also useful to be able to describe the relationship of two numerical variables, such as pf_expression_control and pf_score above.

Exercise 3

3. Looking at your plot from the previous exercise, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.

The scatter plot show a linear relationship between pf_score and pf_expression_control. The direction is to the right, both pf_score and pf_expression_control increase together. Some unusual observations are high pf_expression_control with low pf_scores

```
hfi %>% select(c(pf_score, pf_expression_control)) %>% filter(pf_score > 5 & pf_score < 6) %>% filter(p
```

```
## # A tibble: 1 x 2
##   pf_score pf_expression_control
##      <dbl>                 <dbl>
## 1     5.07                  8.75
```

Just as you've used the mean and standard deviation to summarize a single variable, you can summarize the relationship between these two variables by finding the line that best follows their association. Use the following interactive function to select the line that you think does the best job of going through the cloud of points.

```
# This will only work interactively (i.e. will not show in the knitted document)
df <- hfi %>% select(c(pf_score, pf_expression_control)) %>% drop_na()
DATA606::plot_ss(x = df$pf_expression_control, y = df$pf_score)
```

After running this command, you'll be prompted to click two points on the plot to define a line. Once you've done that, the line you specified will be shown in black and the residuals in blue. Note that there are 30 residuals, one for each of the 30 observations. Recall that the residuals are the difference between the observed values and the values predicted by the line:

$$e_i = y_i - \hat{y}_i$$

The most common way to do linear regression is to select the line that minimizes the sum of squared residuals. To visualize the squared residuals, you can rerun the plot command and add the argument `showSquares = TRUE`.

```
DATA606::plot_ss(x = pf_expression_control, y = pf_score, data = df, showSquares = TRUE)
```

Note that the output from the `plot_ss` function provides you with the slope and intercept of your line as well as the sum of squares.

Exercise 4

4. Using `plot_ss`, choose a line that does a good job of minimizing the sum of squares. Run the function several times. What was the smallest sum of squares that you got? How does it compare to your neighbors?

The smallest sum of squares i got is 1108.238

The linear model

It is rather cumbersome to try to get the correct least squares line, i.e. the line that minimizes the sum of squared residuals, through trial and error. Instead, you can use the `lm` function in R to fit the linear model (a.k.a. regression line).

```
m1 <- lm(pf_score ~ pf_expression_control, data = hfi)
```

The first argument in the function `lm` is a formula that takes the form `y ~ x`. Here it can be read that we want to make a linear model of `pf_score` as a function of `pf_expression_control`. The second argument specifies that R should look in the `hfi` data frame to find the two variables.

The output of `lm` is an object that contains all of the information we need about the linear model that was just fit. We can access this information using the summary function.

```
summary(m1)
```

```
##
## Call:
## lm(formula = pf_score ~ pf_expression_control, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8467 -0.5704  0.1452  0.6066  3.2060
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.61707    0.05745   80.36   <2e-16 ***
## pf_expression_control  0.49143    0.01006   48.85   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8318 on 1376 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.6342, Adjusted R-squared:  0.634
## F-statistic:  2386 on 1 and 1376 DF,  p-value: < 2.2e-16
```

Let’s consider this output piece by piece. First, the formula used to describe the model is shown at the top. After the formula you find the five-number summary of the residuals. The “Coefficients” table shown next is key; its first column displays the linear model’s y-intercept and the coefficient of `at_bats`. With this table, we can write down the least squares regression line for the linear model:

$$\hat{y} = 4.61707 + 0.49143 \times pf\_expression\_control$$

One last piece of information we will discuss from the summary output is the Multiple R-squared, or more simply, R^2. The R^2 value represents the proportion of variability in the response variable that is explained by the explanatory variable. For this model, 63.42% of the variability in runs is explained by at-bats.

Exercise 5

5. Fit a new model that uses `pf_expression_control` to predict `hf_score`, or the total human freedom score. Using the estimates from the R output, write the equation of the regression line. What does the slope tell us in the context of the relationship between human freedom and the amount of political pressure on media content?

```r
summary(lm(hf_score ~ pf_expression_control, data=hfi))
```

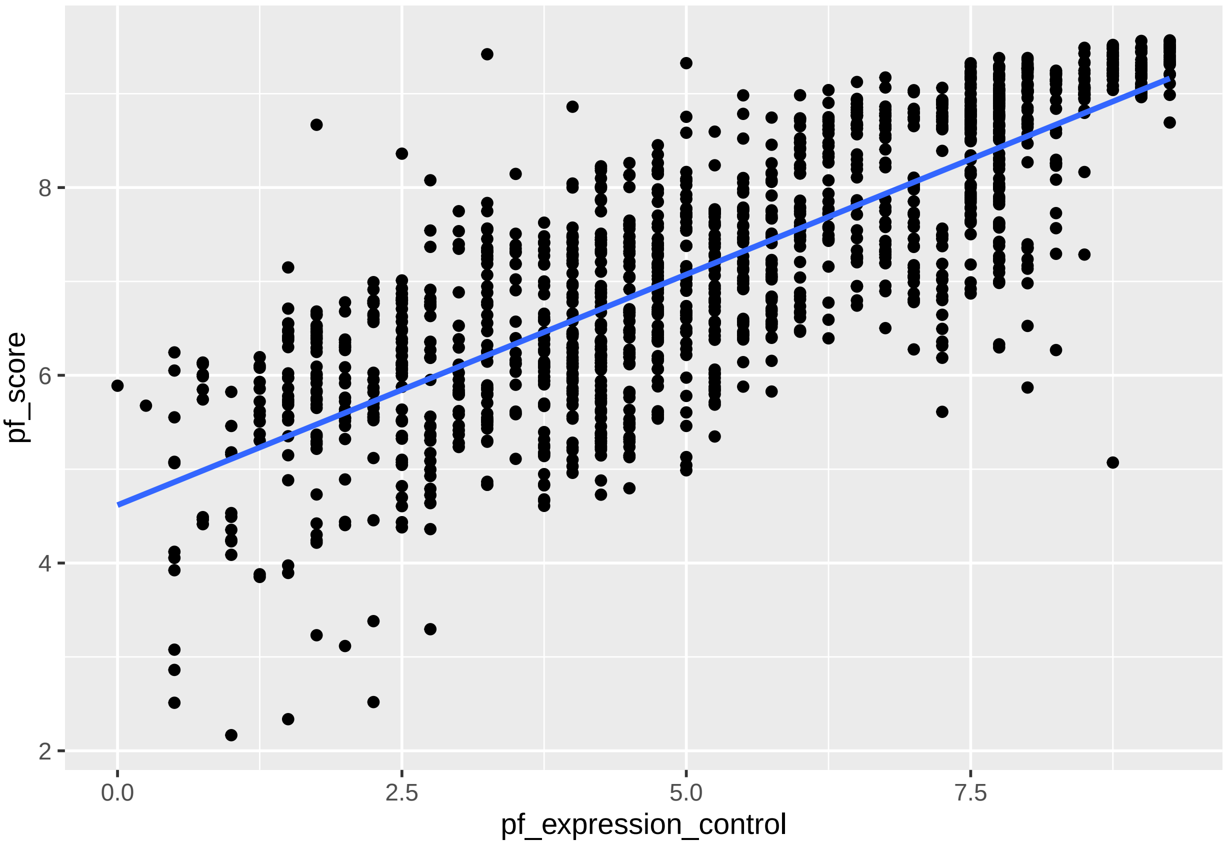
```
##
## Call:
## lm(formula = hf_score ~ pf_expression_control, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6198 -0.4908  0.1031  0.4703  2.2933
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.153687   0.046070  111.87   <2e-16 ***
## pf_expression_control 0.349862   0.008067   43.37   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.667 on 1376 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.5775, Adjusted R-squared:  0.5772
## F-statistic:  1881 on 1 and 1376 DF,  p-value: < 2.2e-16
```

$ hat{y} = 0.35 * pf_expression_control + 5.15 The slope of 0.35 means we expect as pf_expression_control increase we expected hr_score to increase by 0.35.

Prediction and prediction errors

Let’s create a scatterplot with the least squares line for `m1` laid on top.

```r
ggplot(data = hfi, aes(x = pf_expression_control, y = pf_score)) +
  geom_point(na.rm = T) +
  stat_smooth(method = "lm", se = FALSE, na.rm = T)
```



Here, we are literally adding a layer on top of our plot. `geom_smooth` creates the line by fitting a linear model. It can also show us the standard error `se` associated with our line, but we'll suppress that for now.

This line can be used to predict y at any value of x. When predictions are made for values of x that are beyond the range of the observed data, it is referred to as *extrapolation* and is not usually recommended. However, predictions made within the range of the data are more reliable. They're also used to compute the residuals.

Exercise 6

6. If someone saw the least squares regression line and not the actual data, how would they predict a country's personal freedom school for one with a 6.7 rating for `pf_expression_control`? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction? geom_rug(sides='rt')

```
lmodel = lm(pf_score ~ pf_expression_control, data=hfi)
data = summary(lmodel)
hfi$yhat = data$coefficients[2] * hfi$pf_expression_control + data$coefficients[1]
hfi$residual <- hfi$yhat - hfi$pf_score
hfi %>% select(pf_expression_control,pf_score, yhat, residual) %>% filter(pf_expression_control > 6.6 &
```

```
## # A tibble: 34 x 4
##    pf_expression_control pf_score  yhat residual
##                    <dbl>    <dbl> <dbl>    <dbl>
##  1                  6.75     7.43  7.93    0.503
##  2                    6.75     8.22  7.93  -0.282
##  3                    6.75     8.77  7.93  -0.833
##  4                    6.75     7.87  7.93   0.0621
##  5                    6.75     7.39  7.93   0.539
##  6                    6.75     7.25  7.93   0.679
##  7                    6.75     7.79  7.93   0.146
##  8                    6.75     8.27  7.93  -0.331
##  9                    6.75     7.19  7.93   0.740
## 10                    6.75     7.75  7.93   0.184
## # ... with 24 more rows
```

```
data$coefficients[2] * 6.7 + data$coefficients[1]
```

```
## [1] 7.909663
```

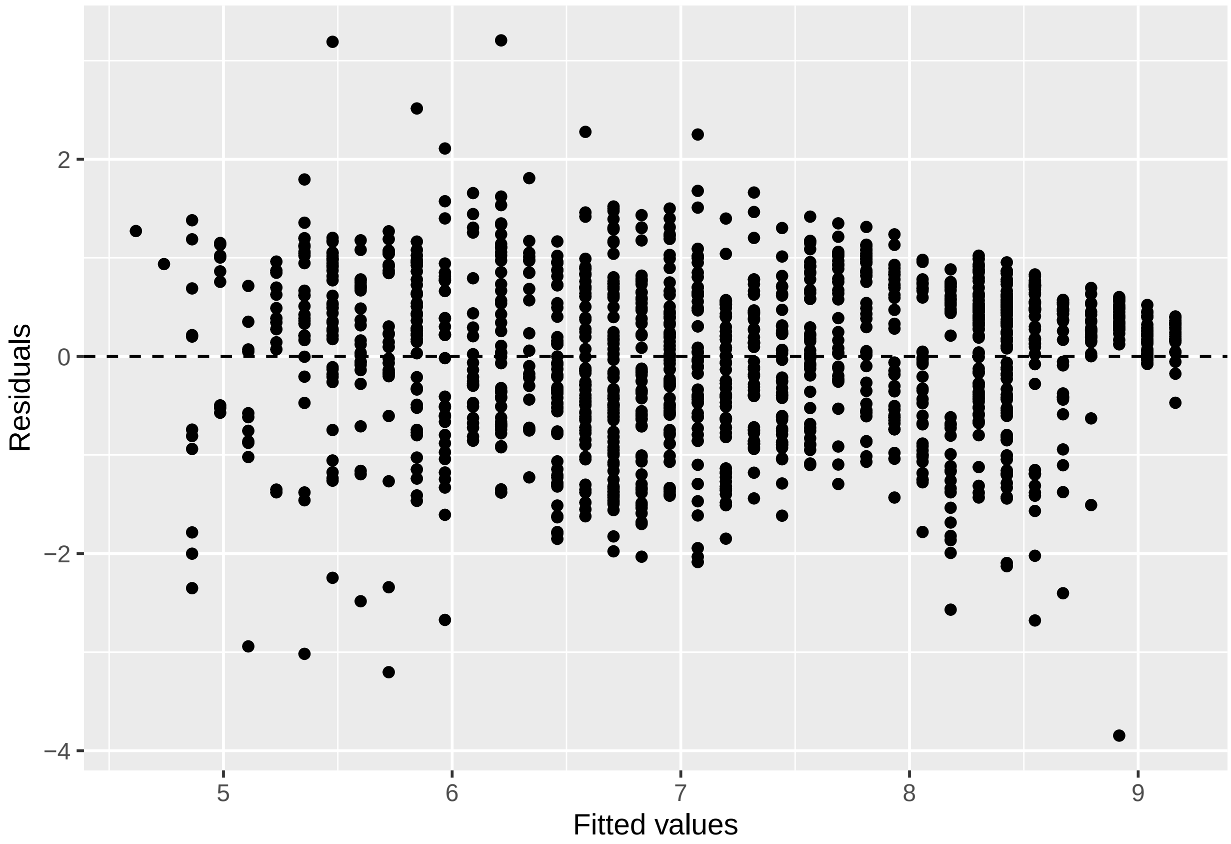
A `pf_expression_control` of 6.7 corresponds to a pf_score of between 7.19 and 8.77 in the data but the model would predict 7.91, which is within those values.

Model diagnostics

To assess whether the linear model is reliable, we need to check for (1) linearity, (2) nearly normal residuals, and (3) constant variability.

Linearity: You already checked if the relationship between `pf_score` and 'pf_expression_control' is linear using a scatterplot. We should also verify this condition with a plot of the residuals vs. fitted (predicted) values.

```
ggplot(data = m1, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



Notice here that `m1` can also serve as a data set because stored within it are the fitted values ($\hat{y}$) and the residuals. Also note that we're getting fancy with the code here. After creating the scatterplot on the first layer (first line of code), we overlay a horizontal dashed line at $y = 0$ (to help us check whether residuals are distributed around 0), and we also reanme the axis labels to be more informative.

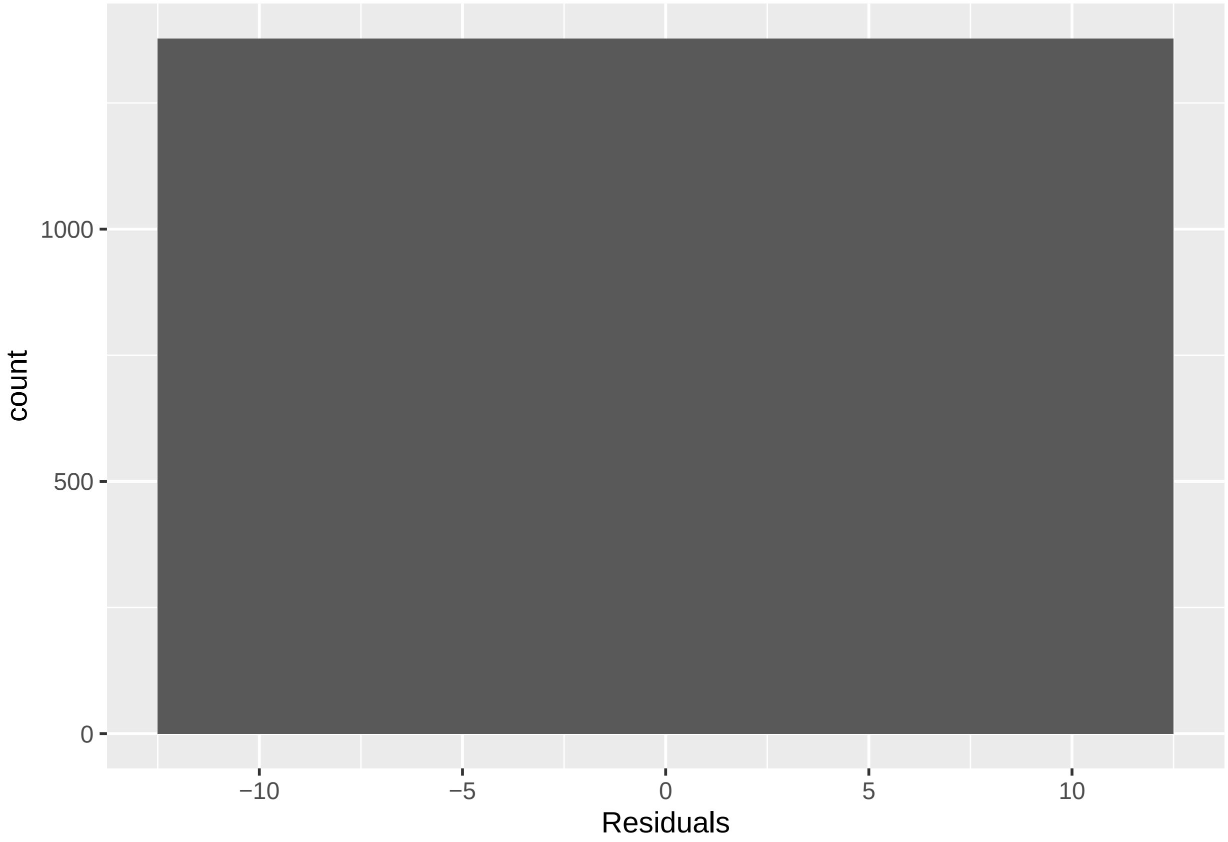
Exercise 7

7. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between the two variables?

The residuals are very close the 0 line with fewer and you move away. That might indicate the residual plot is normally distributed.

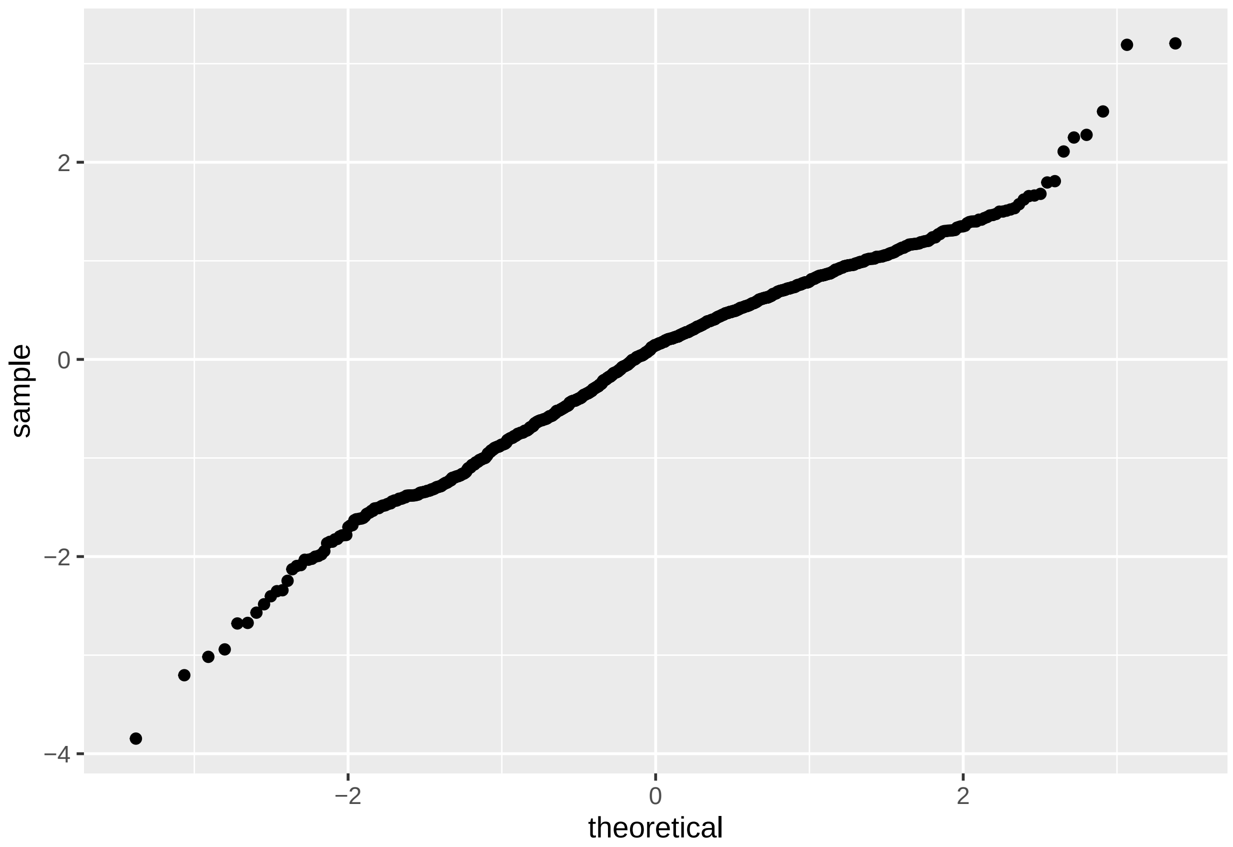
Nearly normal residuals: To check this condition, we can look at a histogram

```
ggplot(data = m1, aes(x = .resid)) +
  geom_histogram(binwidth = 25) +
  xlab("Residuals")
```



or a normal probability plot of the residuals.

```
ggplot(data = m1, aes(sample = .resid)) +
  stat_qq()
```



Note that the syntax for making a normal probability plot is a bit different than what you're used to seeing: we set `sample` equal to the residuals instead of `x`, and we set a statistical method `qq`, which stands for "quantile-quantile", another name commonly used for normal probability plots.

Exercise 8

8. Based on the histogram and the normal probability plot, does the nearly normal residuals condition appear to be met?

The histogram is one rectangle, with smaller bins it does look normal. The qq plot is more convincing as it move to the top right of the graph.

Constant variability:

Exercise 9

9. Based on the residuals vs. fitted plot, does the constant variability condition appear to be met?

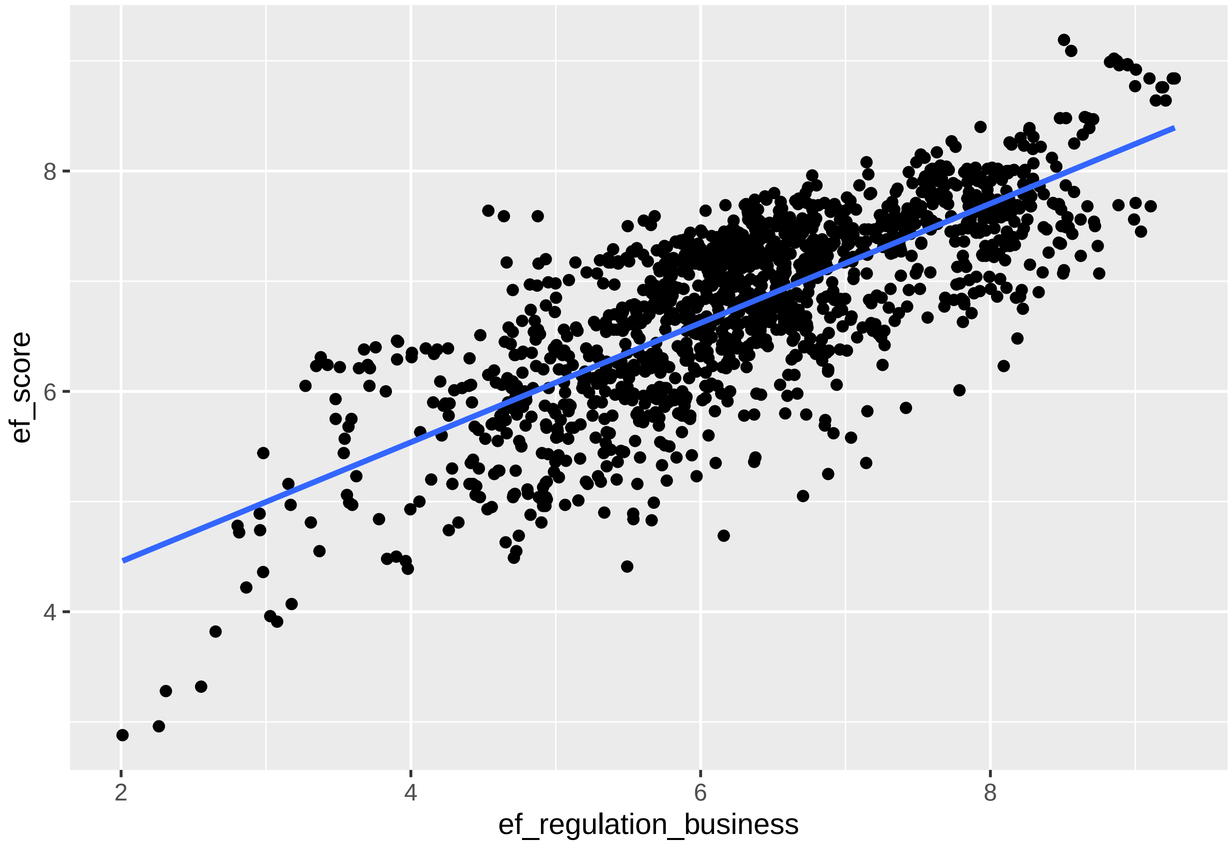
Yes it does * * *

More Practice

Exercise 10

- Choose another freedom variable and a variable you think would strongly correlate with it.. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship?

```
ggplot(data = hfi, aes(x = ef_regulation_business, y = ef_score)) +
  geom_point(na.rm = T) +
  stat_smooth(method = "lm", se = FALSE, na.rm = T)
```



It definitely seem linear with a positive correlation

Exercise 11

- How does this relationship compare to the relationship between `pf_expression_control` and `pf_score`? Use the R^2 values from the two model summaries to compare. Does your independent variable seem to predict your dependent one better? Why or why not?

```
efm1 <- lm(ef_score ~ ef_regulation_business, data = hfi)
summary(efm1)
```

```
##
## Call:
## lm(formula = ef_score ~ ef_regulation_business, data = hfi)
```

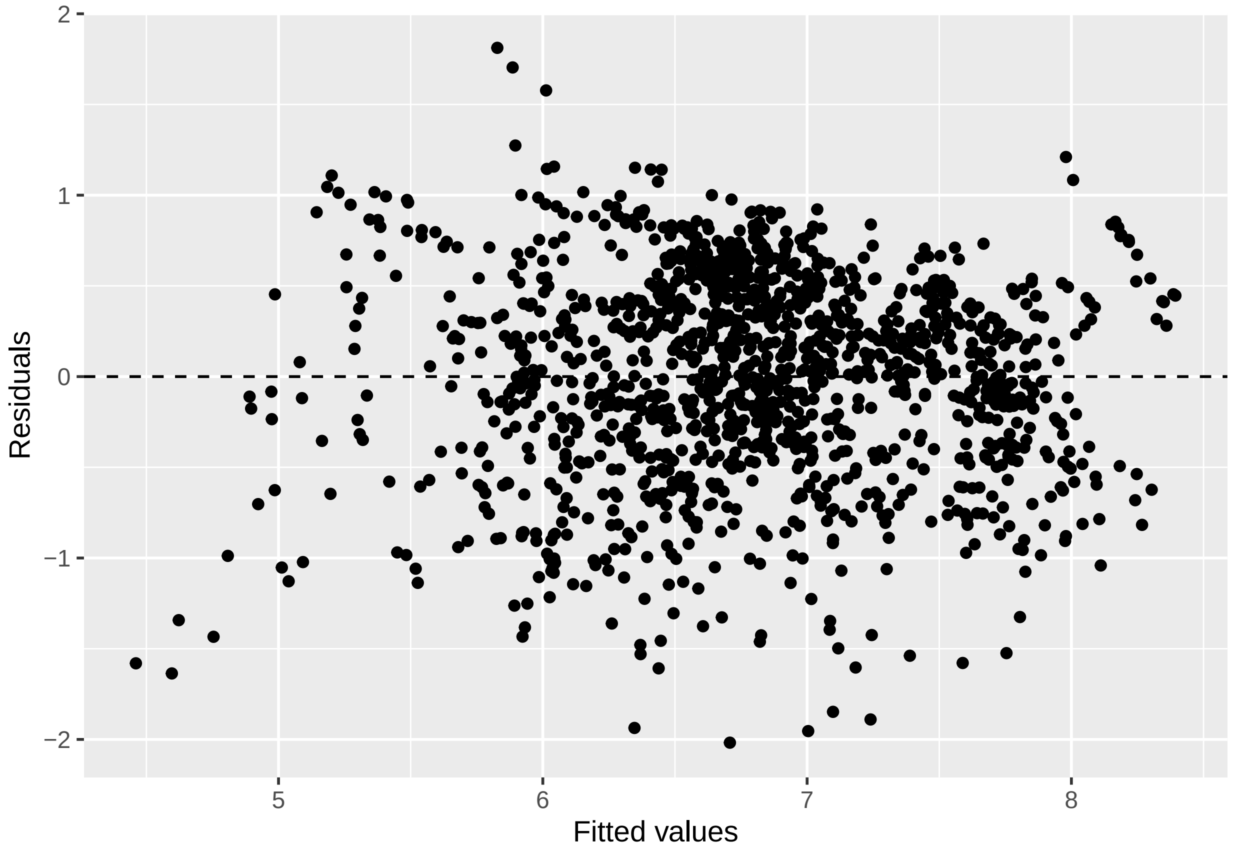
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01787 -0.37134  0.03646  0.42194  1.81218
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.37200    0.07962   42.35   <2e-16 ***
## ef_regulation_business  0.54154    0.01237   43.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5642 on 1372 degrees of freedom
##   (84 observations deleted due to missingness)
## Multiple R-squared:  0.5828, Adjusted R-squared:  0.5825
## F-statistic:  1917 on 1 and 1372 DF,  p-value: < 2.2e-16
```

The relationship between `pf_expression_control` and `pf_score` is stronger (0.63) than `ef_regulation_business` and `ef_score` (0.58), The independent variable ef_regulation_business is not a good predictor of ef_score

Exercise 12

- What's one freedom relationship you were most surprised about and why? Display the model diagnostics for the regression model analyzing this relationship.

```
ggplot(data = efm1, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



When looking at the fitted vs residuals the ef_score model is surprisingly bad when compared to the pf_score. There might be a better variable to use beside ef_regulation_business. * * *