Introduction to linear regression

Alyssa Gurkas

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.

```r
library(tidyverse)
library(openintro)
library(DATA606)
```

```
##
## Welcome to CUNY DATA606 Statistics and Probability for Data Analytics
## This package is designed to support this course. The text book used
## is OpenIntro Statistics, 4th Edition. You can read this by typing
## vignette(’os4’) or visit www.OpenIntro.org.
##
## The getLabs() function will return a list of the labs available.
##
## The demo(package=’DATA606’) will list the demos that are available.
```

```r
library(car)
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.

1. What are the dimensions of the dataset?

Exercise 1 Response

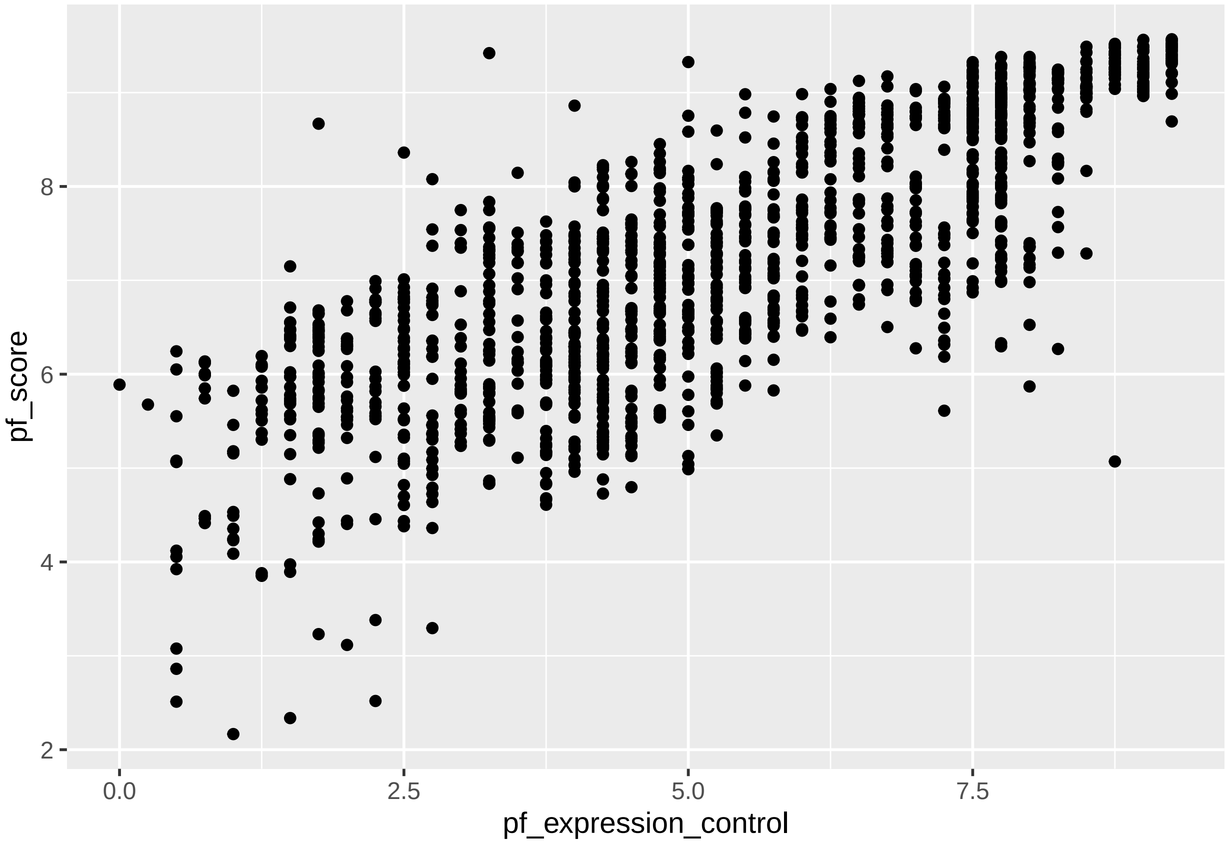
The dataset 'hfi' has 1458 observations. Each observation represents the annual data for the respective nation included in the Human Freedom Index reports from 2008-2016. There are 123 different variables in this dataset that describe the idea of "freedom", such as political, religious, economic, or personal freedom.

The dataset has 1458 rows and 123 columns.

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?

Exercise 2 Response

```r
ggplot(hfi, aes(x=pf_expression_control , y=pf_score)) +
  geom_point()
```



To determine if there is a relationship between the personal freedom score and the personal freedom expression control score a scatter plot can be used to visualize the linear relationship. Based on the plot produced, there is evidence that there is a relationship. Therefore, I would be comfortable using a linear model to predict the personal freedom score.

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

```r
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.

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.

Exercise 3 Response

There is a somewhat strong, positive linear relationship between the personal freedom expression control and personal freedom score.

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.

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

```r
DATA606::plot_ss(x = hfi$pf_expression_control, y = hfi$pf_score, 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.

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?

Exercise 4 Response

```
# plot_ss(hfi$pf_expression_control,
#         hfi$pf_score,
#         showSquares=FALSE,
#         leastSquares=FALSE)
```

The lowest sum of squares produced in this exercise was 1068.176. Other people in the class had lower sums of squares, such as 953.

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 `pf_expression_control`. With this table, we can write down the least squares regression line for the linear model:

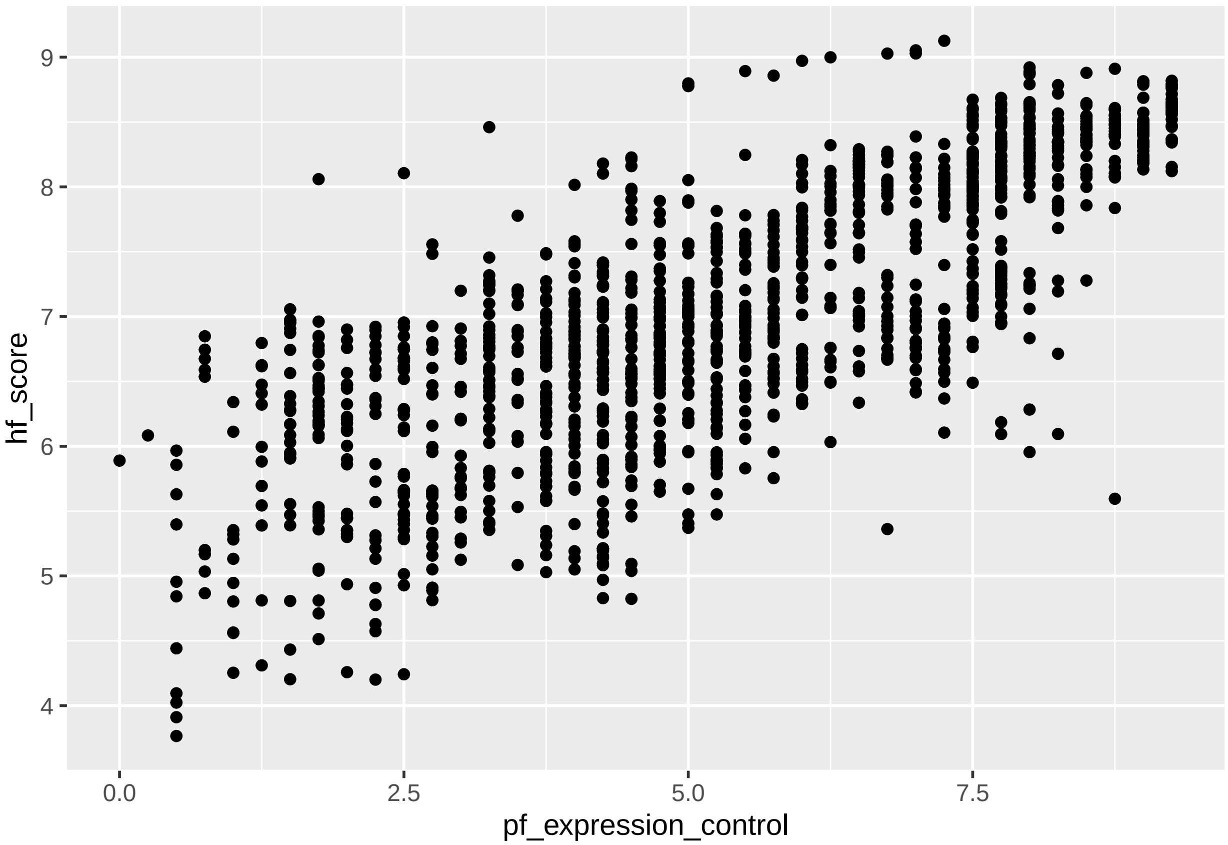
$$\hat{y} = 4.61707 + 0.49143 \times pf\_expression\_control$$

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.

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?

Exercise 5 Response

```
ggplot(hfi, aes(x=pf_expression_control, y=hf_score)) +
  geom_point()
```



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

```
## # A tibble: 1 x 1
##   ‘cor(pf_expression_control, hf_score, use = "complete.obs")‘
##                                                          <dbl>
## 1                                                        0.760
```

```
m2 <- lm(hf_score ~ pf_expression_control, data = hfi)
summary(m2)
```

```
##
## 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
```

The slope provided in the R output from the lm() function is 0.349862. Using the values provided in the output from ‘lm’ we can produce the following equation of the regression line:

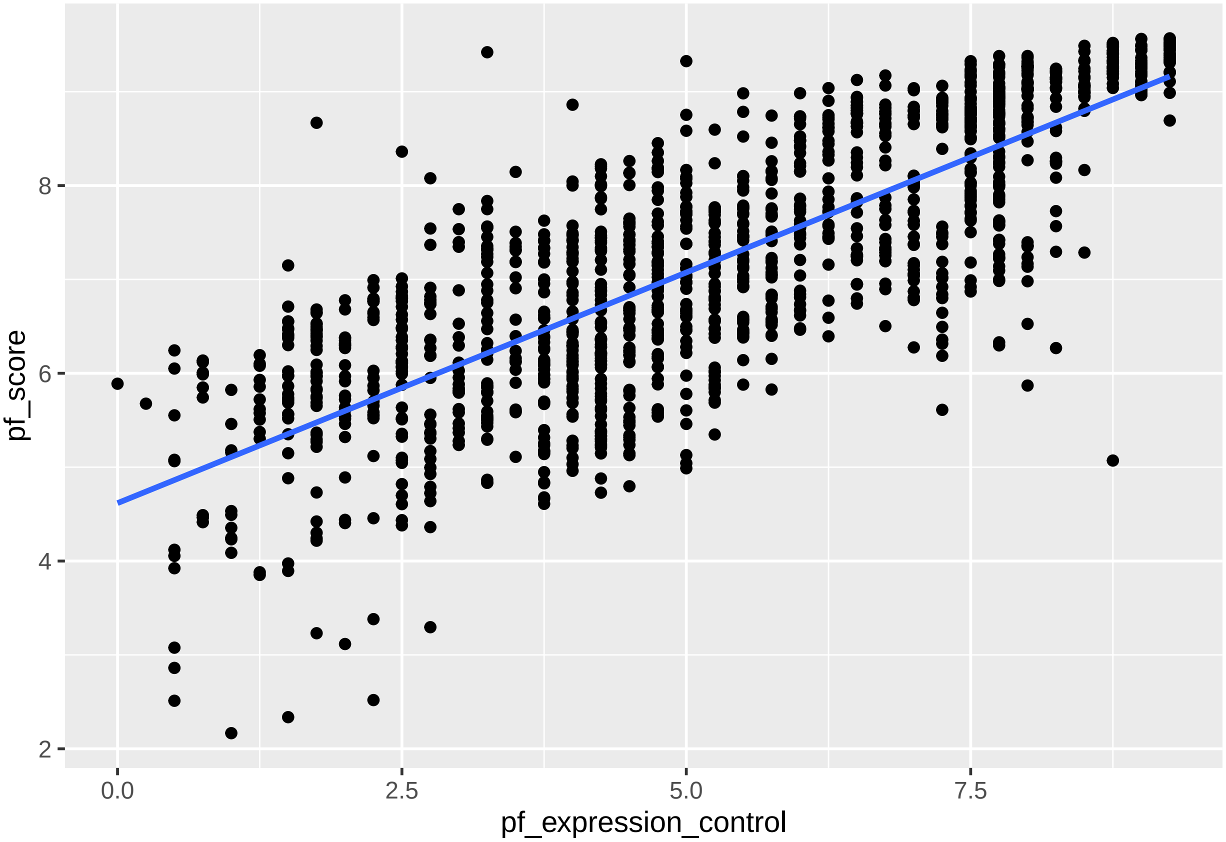
$y = 5.153687 + 0.349862 * pfexpressioncontrol$

The slope tells us that for each additional point in the political expression indicator we can expect roughly 0.349862 additional points in the human freedom score. This means that if a nation were to have a higher personal freedom expression score, you can expect that it has a relatively high human freedom score.

Prediction and prediction errors

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

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



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.

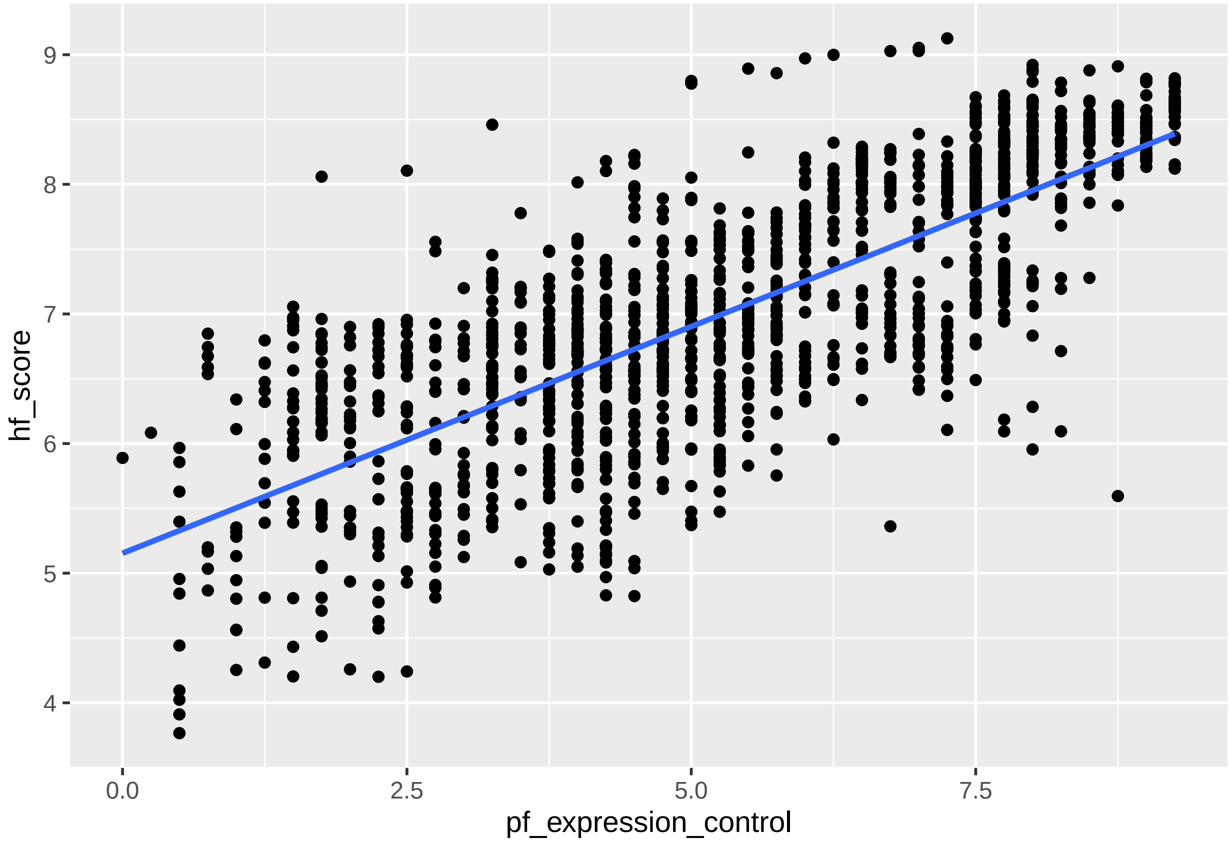
6. If someone saw the least squares regression line and not the actual data, how would they predict a country's personal freedom school for one with a 6.7 rating for `pf_expression_control`? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction?

Exercise 6 Response

```
result <- 5.153687 + 0.349862 * 6.7
result
```

```
## [1] 7.497762
```

```
ggplot(data = hfi, aes(x = pf_expression_control, y = hf_score)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)
```



If someone saw the least squares regression line and not the actual data, they would predict a human freedom score of ~7.5.

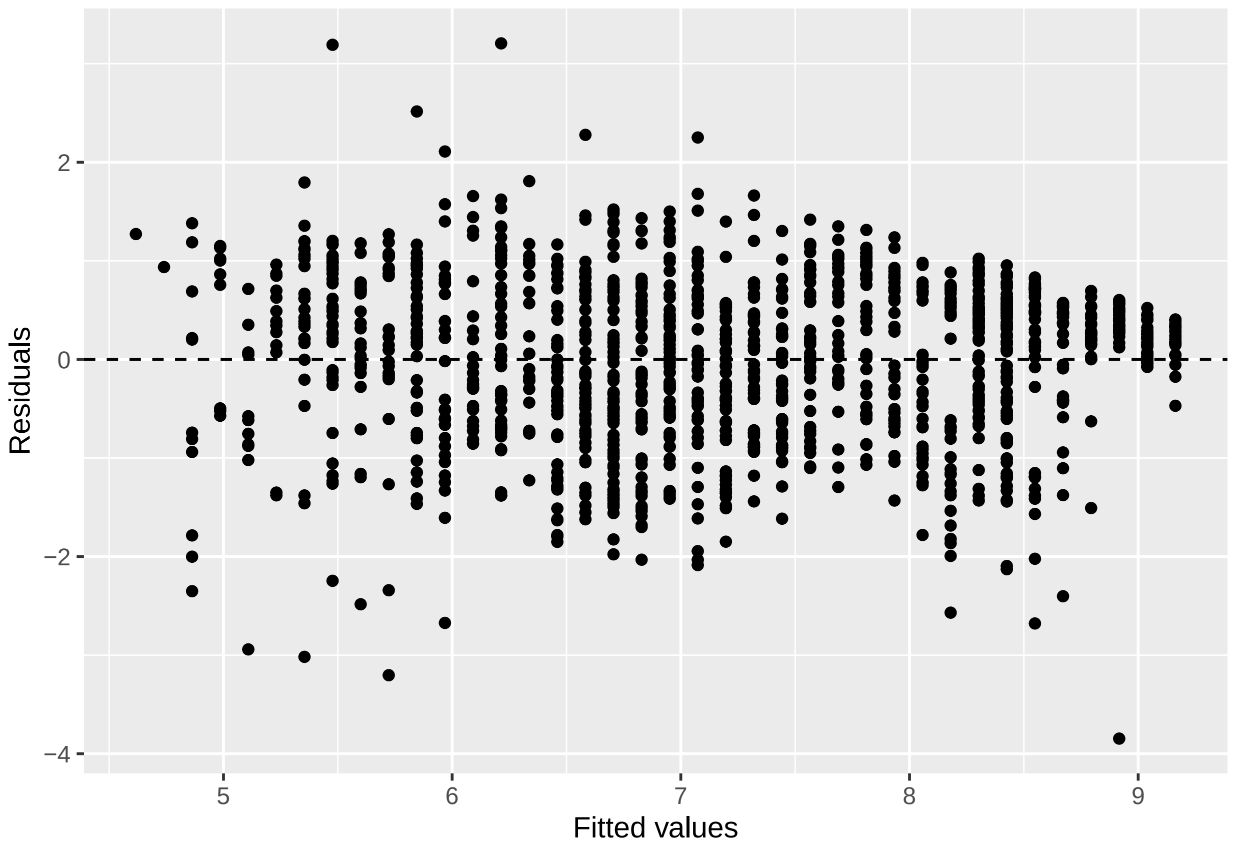
Based on the graph shown above, the model is an overestimate. We can determine how much the model is overestimating by reviewing the output of the m2 summary. In the summary table, it shows that there is a residual standard error of 0.667 on 1376 degrees of freedom and notes that 80 observations were removed due to missing data. We can conclude that this model overestimates by 0.667 for the human freedom score, on average.

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")
```



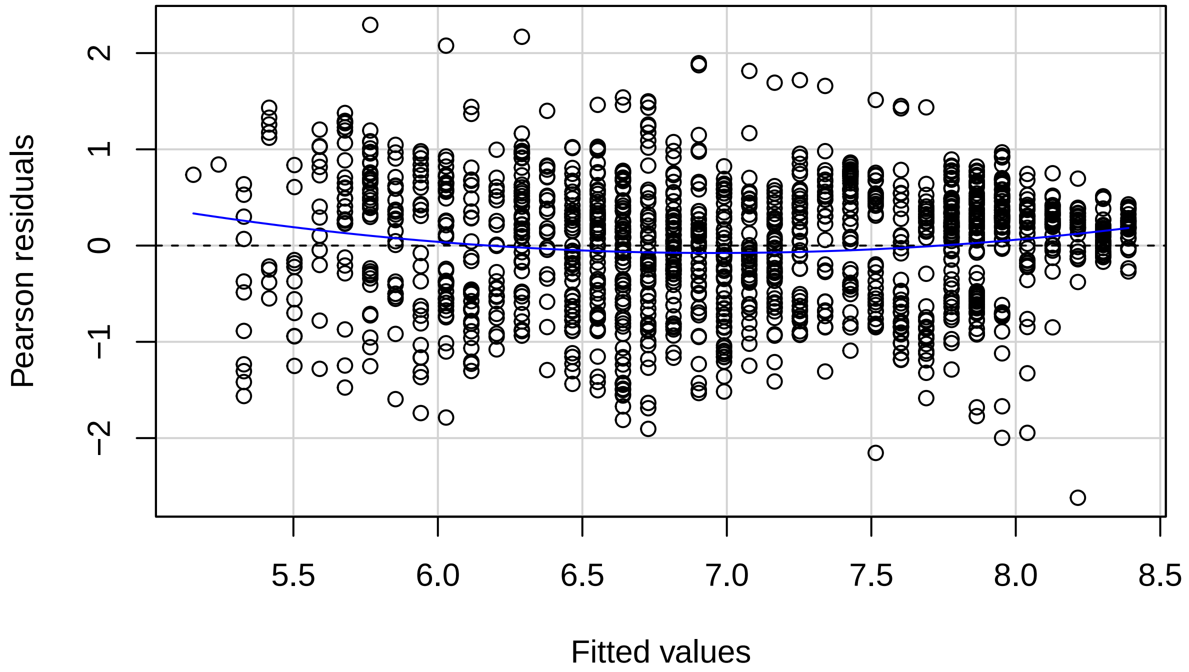
Notice here that `m1` can also serve as a data set because stored within it are the fitted values ($\hat{y}$) and the residuals. Also note that we're getting fancy with the code here. After creating the scatterplot on the first layer (first line of code), we overlay a horizontal dashed line at $y = 0$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

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

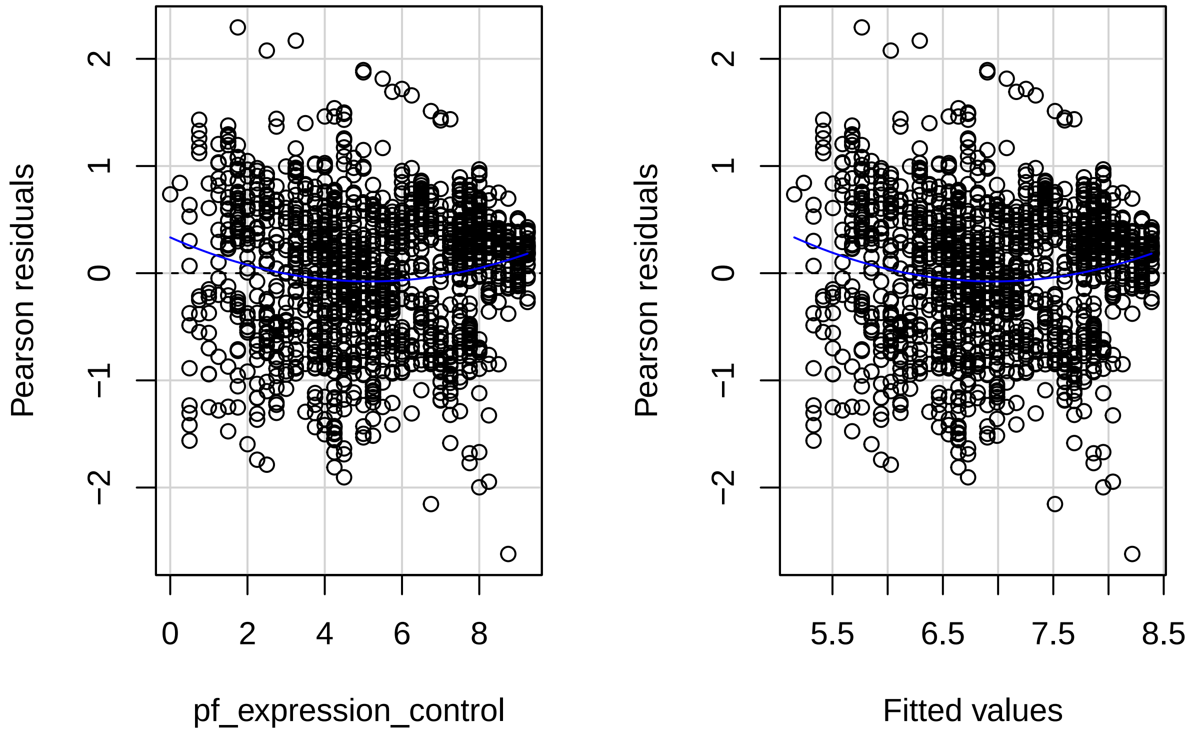
Exercise 7 Response

There is one pattern detected in the residuals plot. At the upper end of the fitted values distribution, the residuals are almost entirely positive. The trend appears to be linear and the data falls around the line with no obvious outliers. If we were to notice a curvature, we should not use a straight line to model the data. In the event that we wanted to complete additional assessments to ensure the relationship is linear we can use the residualPlot() function from the car package, adding a curvature test:

```
car::residualPlot(model=m2)
```



```
car::residualPlots(model=m2)
```

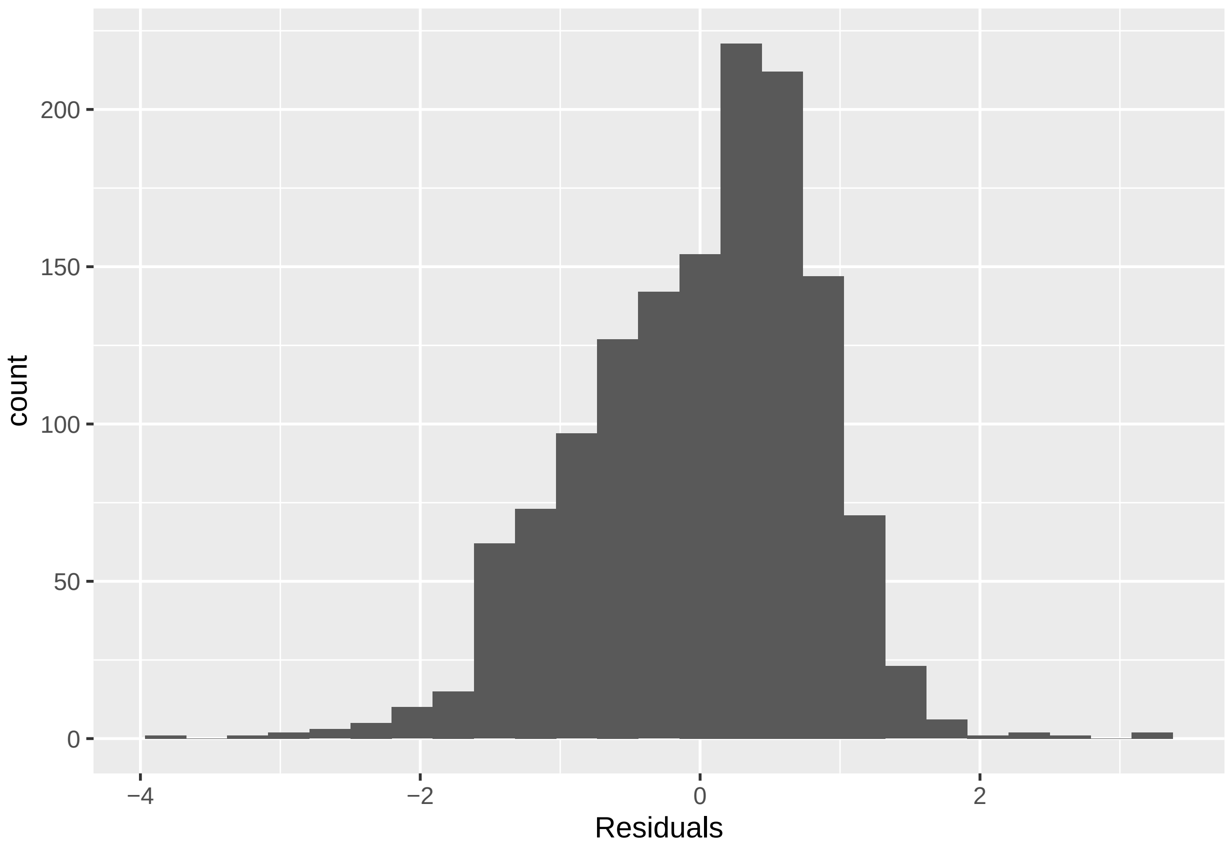


```
##                         Test stat Pr(>|Test stat|)
## pf_expression_control     4.4253        1.039e-05 ***
## Tukey test                4.4253        9.630e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From these plots, we can see that there is a slight curvature although it seems that the deviations from linearity are decently small, and may not be worth any concern. The third line on these plots are the Tukey tests which square the fitted-value.

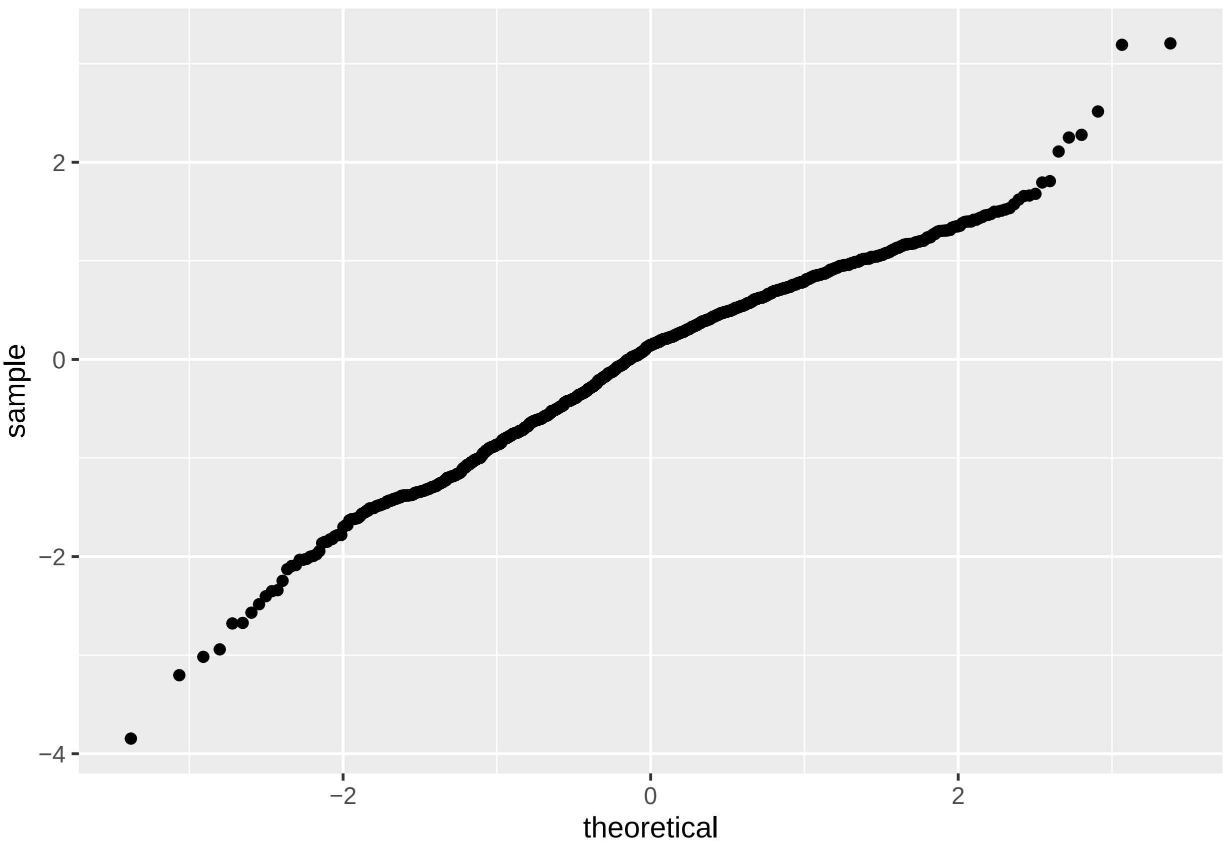
Nearly normal residuals: To check this condition, we can look at a histogram

```
ggplot(data = m1, aes(x = .resid)) +
  geom_histogram(bins = 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.

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

Exercise 8 Response

Based on the histogram and the normal probability plot, the residuals are nearly normal but are slightly left skewed. There are not any residuals that are much further from the regression line than others. However, it should be noted that there are a few residuals that may be considered outliers, although the spread is still evenly distributed, and the data seems nearly normal enough.

Constant variability:

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

Exercise 9 Response

Based on the residuals vs. fitted plot, the constant variability condition does appear to be met. If we wanted to conduct further analysis to confirm, we could complete a non-constant variance test using the car package:

```
car::ncvTest(m2)
```

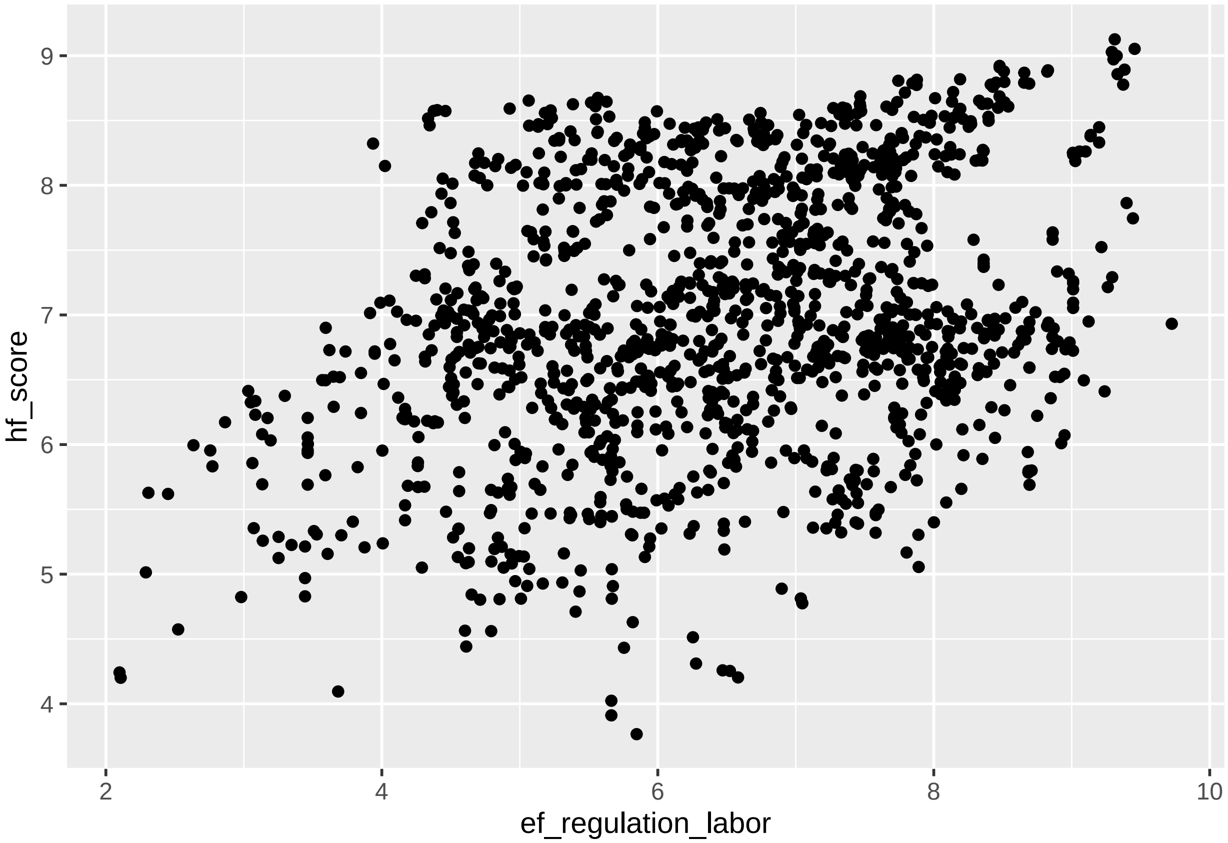
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 40.60409, Df = 1, p = 1.8642e-10
```

When conducting a Non-Constant Variance Test using the ncvTest() function from the car package, it seems that there is not homoscedasticity. Based on this finding, I would be inclined to use a "heteroscedasticity corrected covariance matrix" when estimating standard errors.

More Practice

- 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(hfi, aes(x=ef_regulation_labor, y=hf_score)) +
  geom_point()
```



At a glance, it seems that there may be a positive, somewhat linear relationship, but it does not seem very strong. We can further investigate by calculating the correlation:

```
hfi %>%
  summarise(cor(ef_regulation_labor, hf_score, use = "complete.obs"))
```

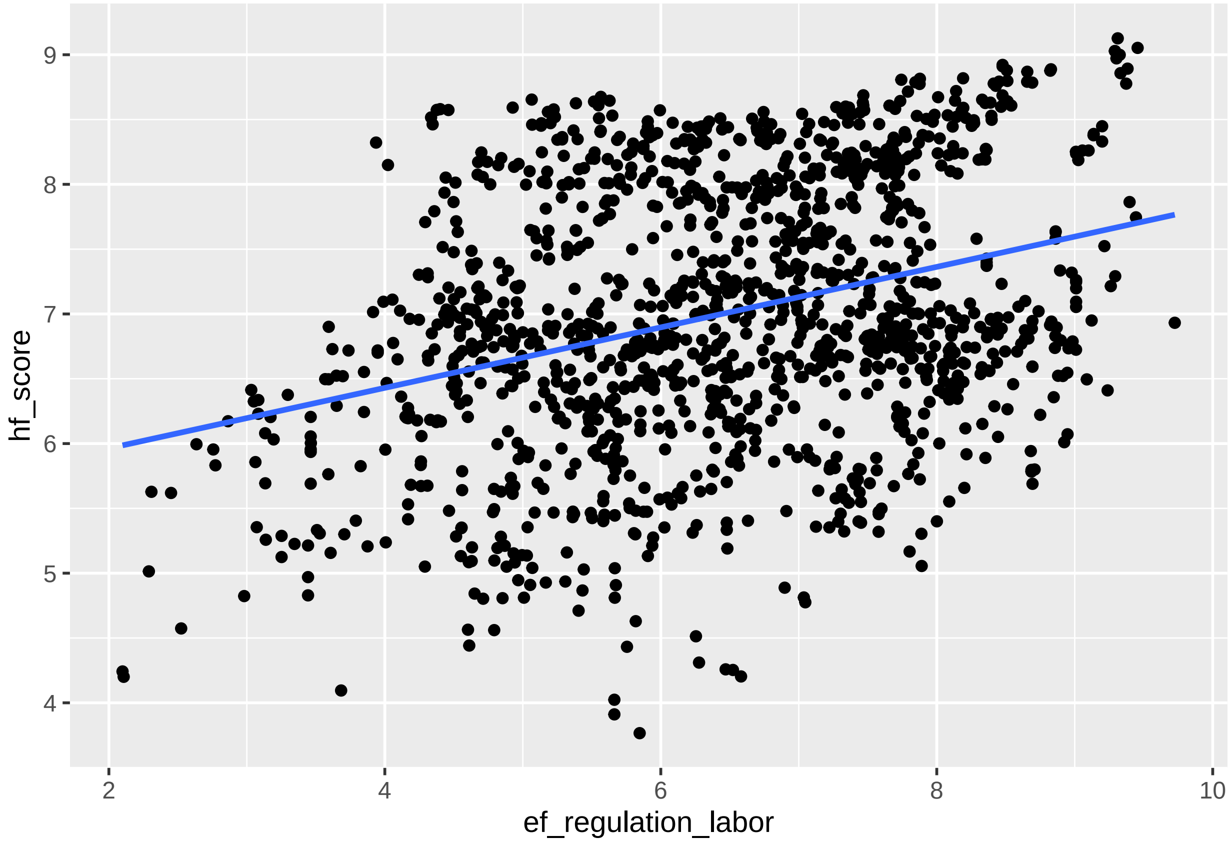
```
## # A tibble: 1 x 1
##   ‘cor(ef_regulation_labor, hf_score, use = "complete.obs")‘
##                                                        <dbl>
## 1                                                      0.325
```

Based on this result, we can confirm the correlation is positive, but is in fact weak, with R=0.325.

```
m3 <- lm(hf_score ~ ef_regulation_labor, data = hfi)
summary(m3)
```

```
##
## Call:
## lm(formula = hf_score ~ ef_regulation_labor, data = hfi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.09499 -0.66777 -0.01863  0.82932  2.05665
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.49644    0.12078   45.51   <2e-16 ***
## ef_regulation_labor  0.23335    0.01831   12.74   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9665 on 1372 degrees of freedom
##   (84 observations deleted due to missingness)
## Multiple R-squared:  0.1058, Adjusted R-squared:  0.1052
## F-statistic: 162.4 on 1 and 1372 DF,  p-value: < 2.2e-16
```

```
ggplot(data = hfi, aes(x = ef_regulation_labor, y = hf_score)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)
```



- 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?

Exercise 11 Response

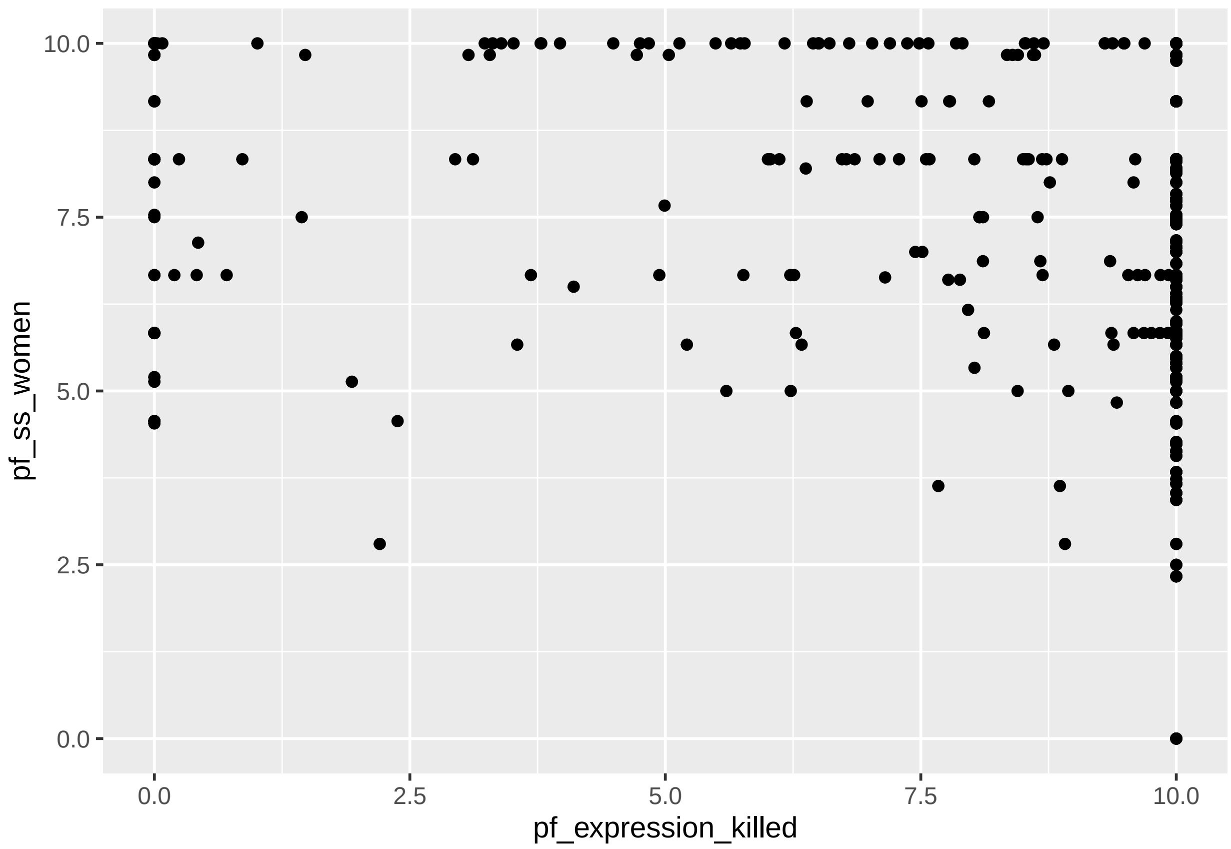
The R^2 value for the third model is 0.1058. This means that a 10.58% of the variation in the human freedom score can be explained by the economic freedom of regulation of labor score. When comparing this result to model 2, where we compared the personal freedom score and the control of expression's (`pf_expression_control`) ability to predict the human freedom score (`hf_score`), we may be able to conclude that model 2 is stronger. Model 2's R^2 value is 0.5775.

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

Exercise 12 Response

I was very surprised to see that there was not a relationship between the score for press killed and the safety and security of women. In the code below, I attempted to fit a model that uses `pf_expression_killed` to predict the safety and security of women, however, there is not a relationship between the two indicators. This model proves to be a very bad model that likely cannot be used as a prediction tool.

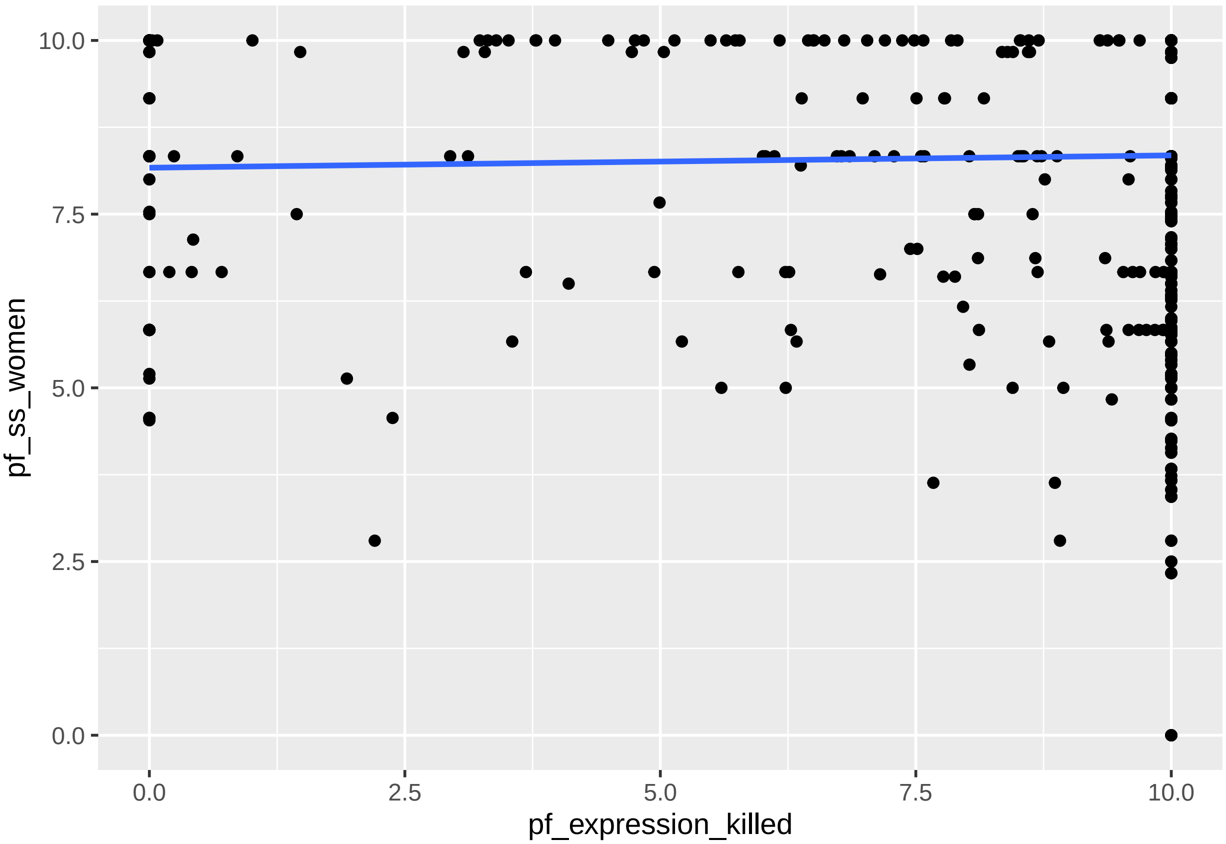
```
ggplot(hfi, aes(x=pf_expression_killed, y=pf_ss_women)) +
  geom_point()
```



```
m4 <- lm(pf_ss_women ~ pf_expression_killed, data = hfi)
summary(m4)
```

```
##
## Call:
## lm(formula = pf_ss_women ~ pf_expression_killed, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3456 -1.1448  0.0141  1.6544  1.8319
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.16812    0.20031  40.776   <2e-16 ***
## pf_expression_killed  0.01774    0.02104   0.843    0.399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.838 on 1356 degrees of freedom
##   (100 observations deleted due to missingness)
## Multiple R-squared:  0.000524,   Adjusted R-squared:  -0.0002131
## F-statistic: 0.7109 on 1 and 1356 DF,  p-value: 0.3993
```

```
ggplot(data = hfi, aes(x = pf_expression_killed, y = pf_ss_women)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE)
```


pf_ss_women
pf_expression_killed