Introduction to linear regression

Kayleah Griffen

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.

1. What are the dimensions of the dataset?

```
dim(hfi)
```

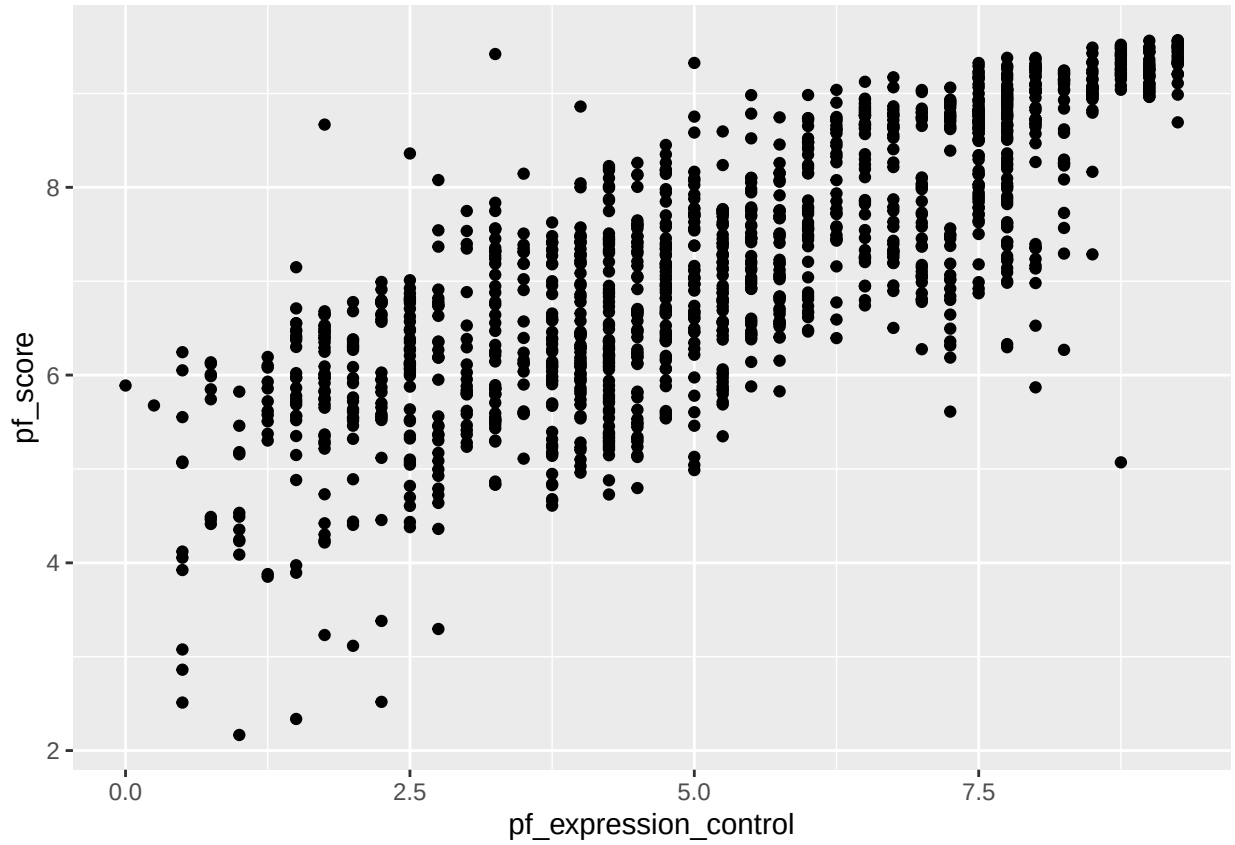
```
## [1] 1458  123
```

There are 1458 cases of 123 variables - so the dimension is 1458 rows by 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?

To display the relationship between the personal freedom score, `pf_score` and one of the other numerical variables, such as the `pf_expression_control` I would use a scatterplot to get a basic idea.

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



The relationship looks linear to me. Knowing a country's `pf_expression_control` (or its score out of 10 with 0 being the most of political pressures and control on media content) 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.

```
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.

Looking at my plot from the previous exercise, the relationship between the two variables is linear with the "strength" of the relationship being represented by the correlation number - which is 0.796. There is a positive relationship where as `pf_expression_control` increases the `pf_score` also increases. There are some points that appear to be outliers in the data - such as a point that lies below the general trend (around a `pf_expression_control` of 8.25 and a `pf_score` of 5)

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)
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`.

```
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?

Using `plot_ss` I played around with multiple lines and the smallest sum of squares I obtained was 1033.8. This is comparable to my classmates.

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. * I think the professor mean to write "For this model, 63.42% of the variability in pf_score is explained by pf_expression_control."

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?

```
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 equation of the regression line is:

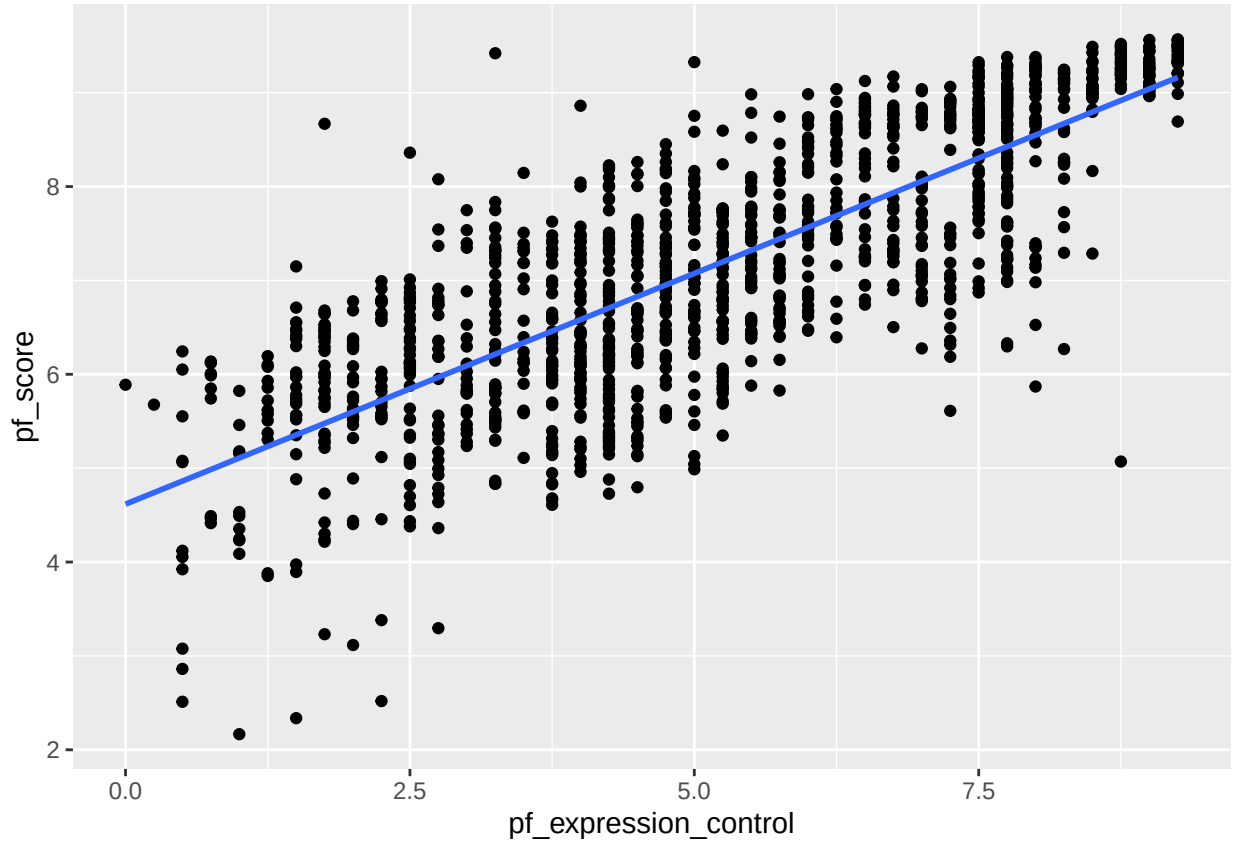
$$\hat{y} = 5.153687 + 0.349862 \times pf\_expression\_control$$

In the context of the relationship between the human freedom and the amount of political pressure on media content is that when there is less political pressure on the media content there is a greater amount of human freedom. An increase of 1 in the `pf_expression_control` variable results in an increase of 0.349862 in the human freedom score. For this model, 57.7% of the variability in hf_score is explained by pf_expression_control.

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?

If someone saw the least squares regression line and not the actual data, for a `pf_expression_control` of 6.7 they would approximately predict that the `pf_score` is about 7.8. In reality, at a `pf_expression_control` of 6.7 more points lie above the line than below so the `pf_score` of 7.8 is likely an underestimate. I would estimate the residual for this prediction is about 0.5.

I can check to see if my estimate for the `pf_score` based on looking at the line is correct using the formula for the line.

```
4.61707+0.49143*6.7
```

```
## [1] 7.909651
```

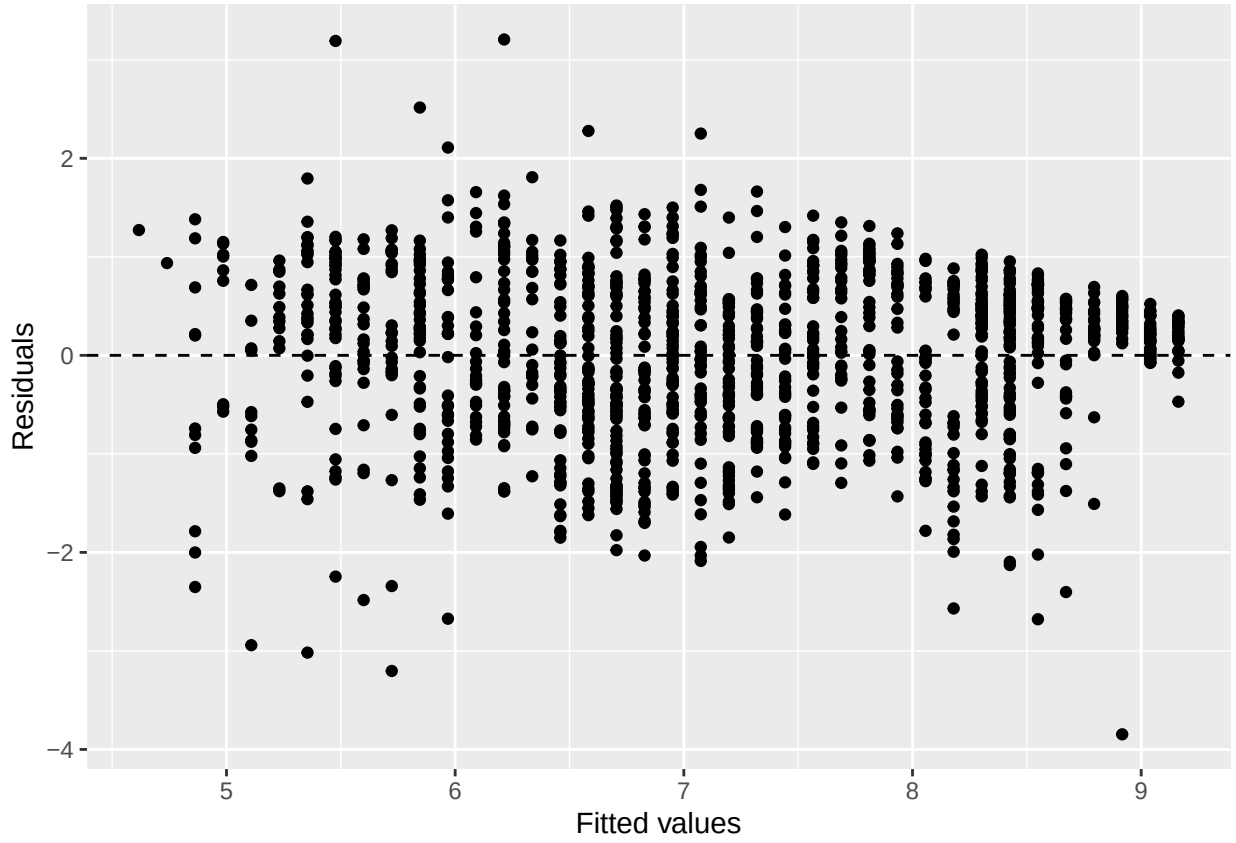
My estimate of 7.8 is close to the actual value predicted by the line which is 7.9.

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")
```



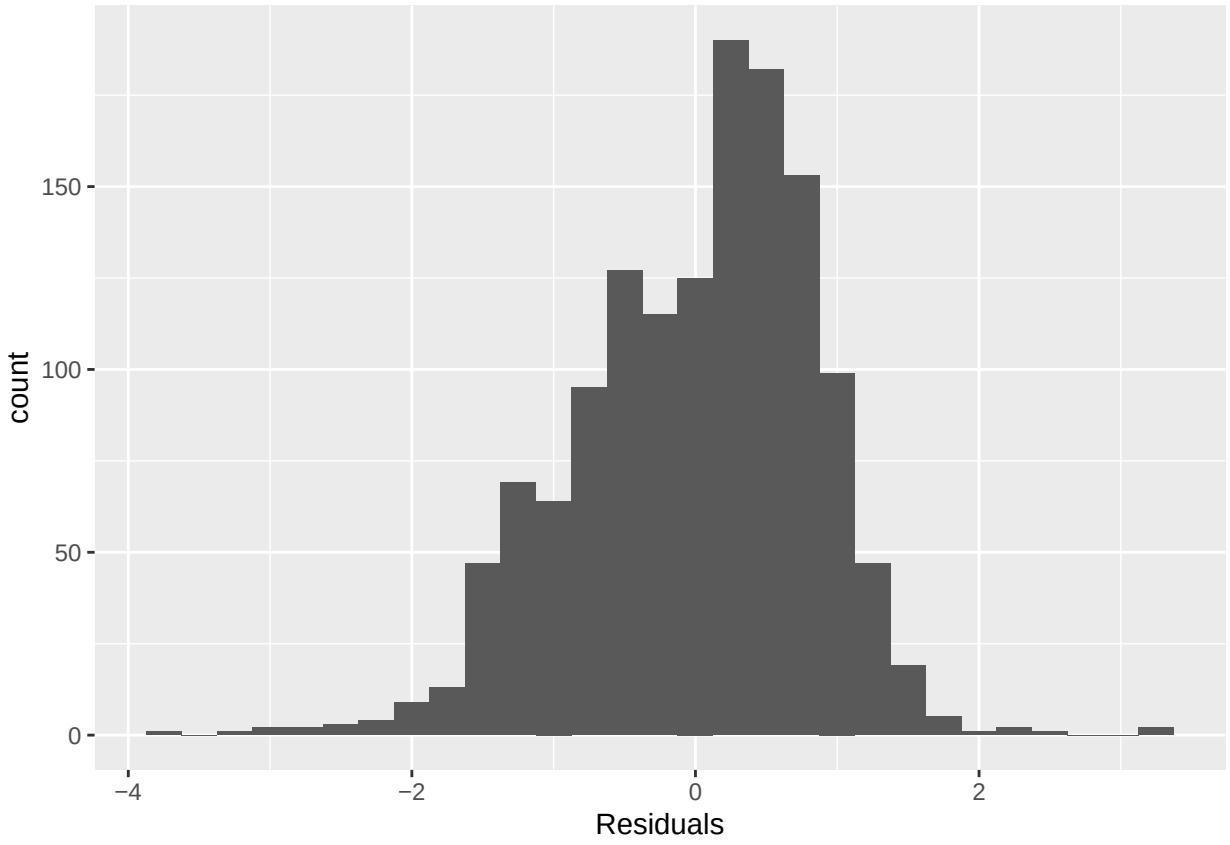
Notice here that `m1` can also serve as a data set because stored within it are the fitted values ($\hat{y}$) and the residuals. Also note that we're getting fancy with the code here. After creating the scatterplot on the first layer (first line of code), we overlay a horizontal dashed line at $y = 0$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

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 plot is centered around a residual of 0 and appears to have no pattern besides being centered at 0 with a slope of 0. This indicates that the linear relationship between the two variables is a good representation for the relationship.

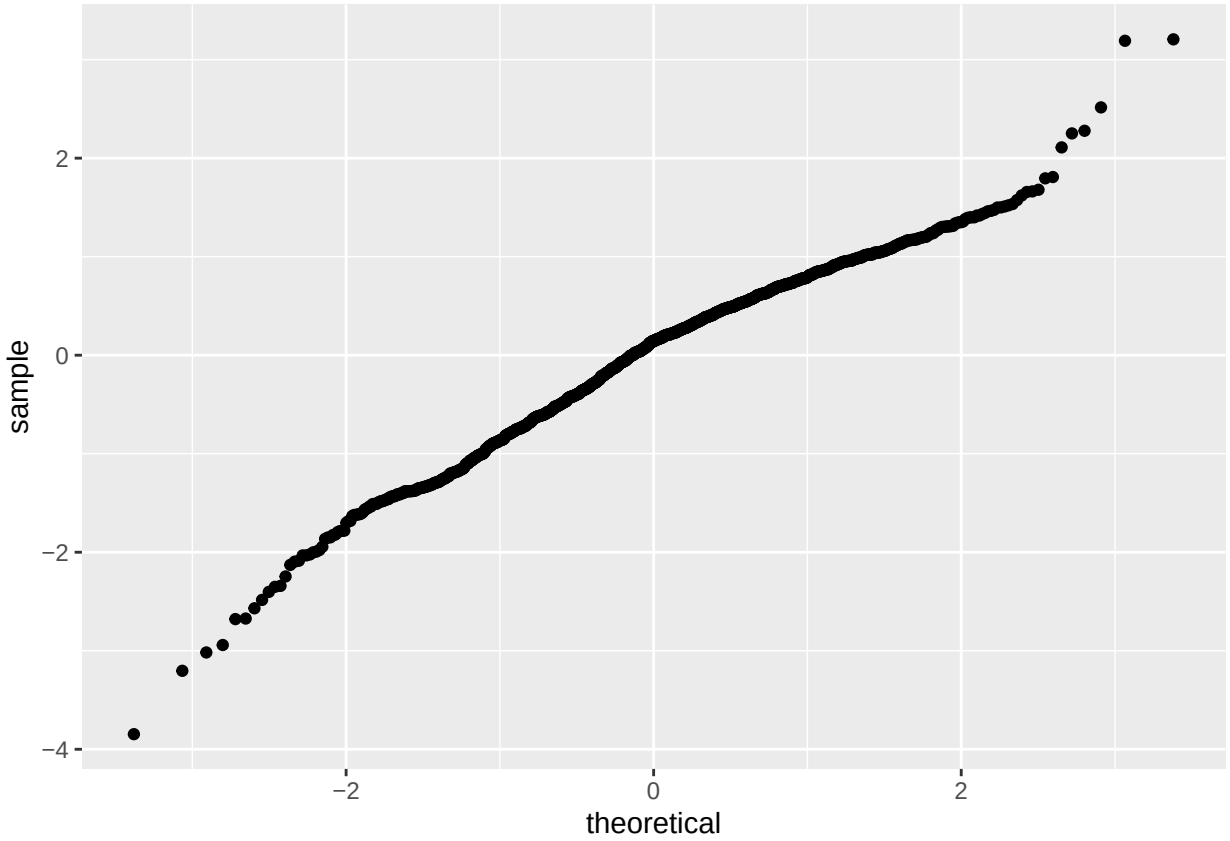
Nearly normal residuals: To check this condition, we can look at a histogram

```
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.

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

Based on the histogram and normal probability plot, the nearly normal condition of the residuals does appear to be met because in the normal probability plot has approximately a slope of 1 with the exceptions being at the lower and upper limits of the data - which is typical as there is less data in those regions and they have the most standard deviation. The histogram appears normally distributed, just with a slight left skew.

Constant variability:

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

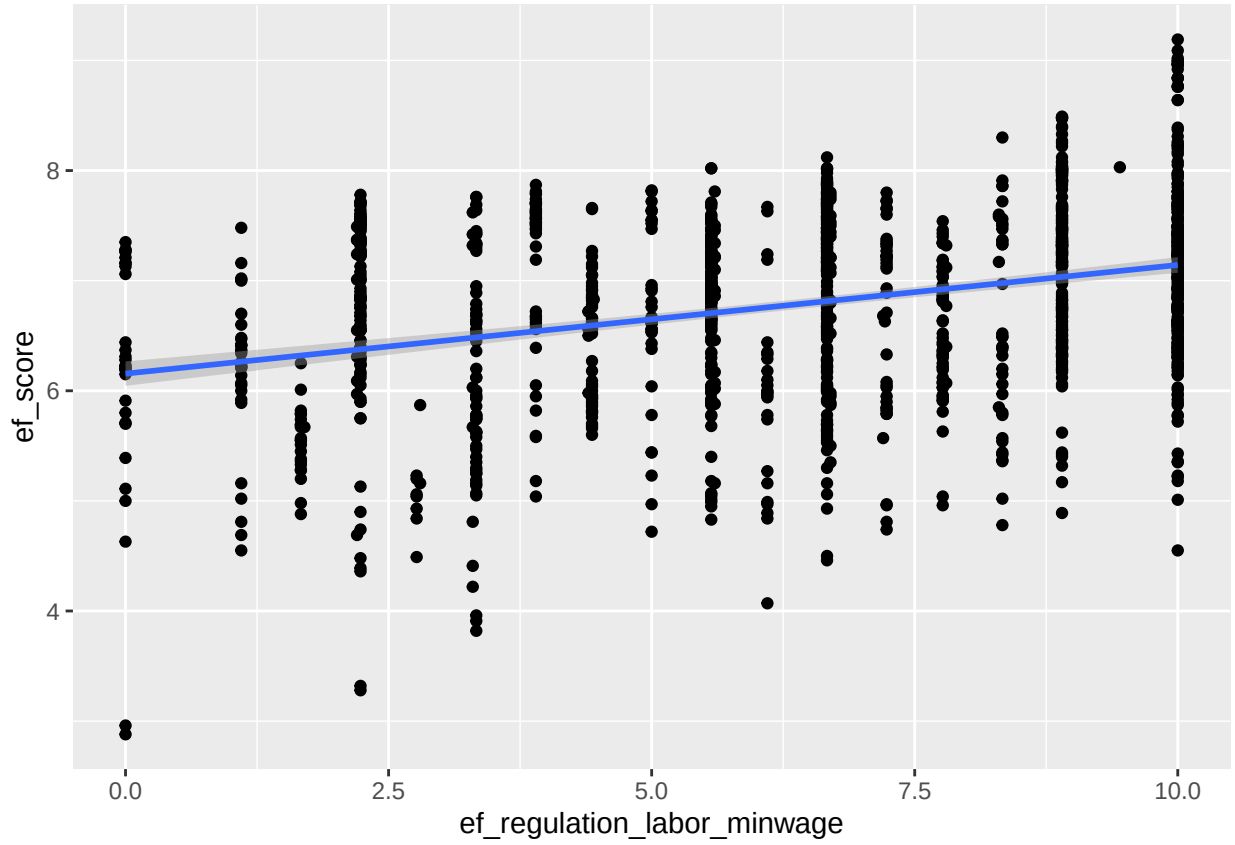
Based on the residuals vs. fitted plot, the constant variability condition appears to be met because in the histogram the count of the different residuals is approximately the same for each residual.

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?

First, I wanted to find the descriptions of the variables, which I found on the open intro website (https://www.openintro.org/data/index.php?data=hfi). Based on reading through the variables, my guess is that the `ef_regulation_labor_minwage` (Labor market regulations - Hiring regulations and minimum wage) is correlated with the `ef_score` (Economic freedom score). I am predicting this because I would think that higher minimum wage would give more people more economic freedom.

```
hfi |> ggplot(aes(x = ef_regulation_labor_minwage, y = ef_score)) +
  geom_point() +
  geom_smooth(method = lm)
```




```
hfi %>%
  summarise(cor(ef_regulation_labor_minwage, ef_score, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(ef_regulation_labor_minwage, ef_score, use = "complete.obs")`
##                                                                <dbl>
## 1                                                              0.316
```

```
m3 <- lm(ef_score ~ ef_regulation_labor_minwage, data = hfi)
summary(m3)
```

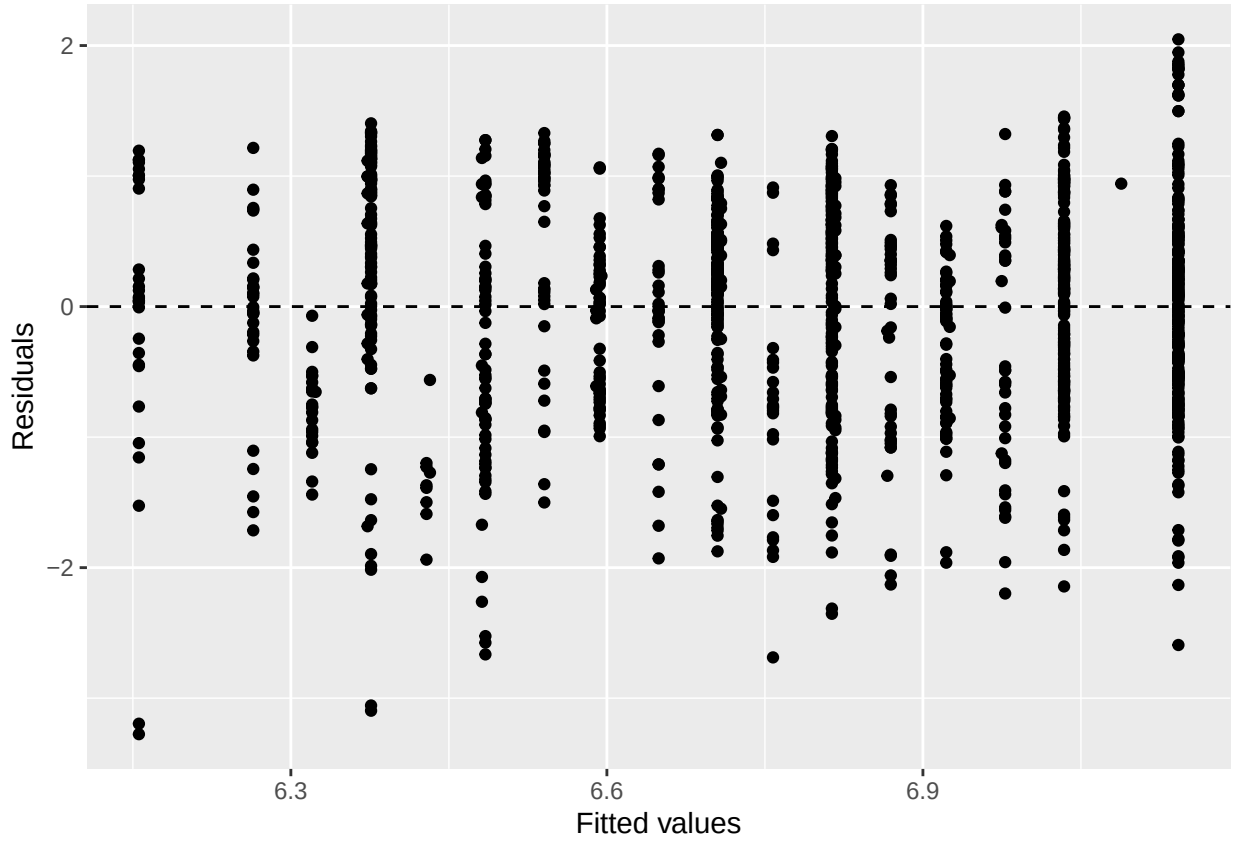
```
##
## Call:
## lm(formula = ef_score ~ ef_regulation_labor_minwage, data = hfi)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2757 -0.5683  0.1175  0.5792  2.0475
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.155742   0.056412  109.12   <2e-16 ***
## ef_regulation_labor_minwage 0.098680   0.008028   12.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8299 on 1365 degrees of freedom
##   (91 observations deleted due to missingness)
## Multiple R-squared:  0.09965,    Adjusted R-squared:  0.09899
## F-statistic: 151.1 on 1 and 1365 DF,  p-value: < 2.2e-16
```

Based on the plot there does seem to be a direct relationship between `ef_regulation_labor_minwage` and `ef_score` whereby as one increases the other increases, the correlation reflects that there is some, 0.316, correlation. This is not that strong of a correlation, I was expecting the relationship to be stronger. The equation for the line is: ef_score = 6.155742 + 0.098680* ef_regulation_labor_minwage.

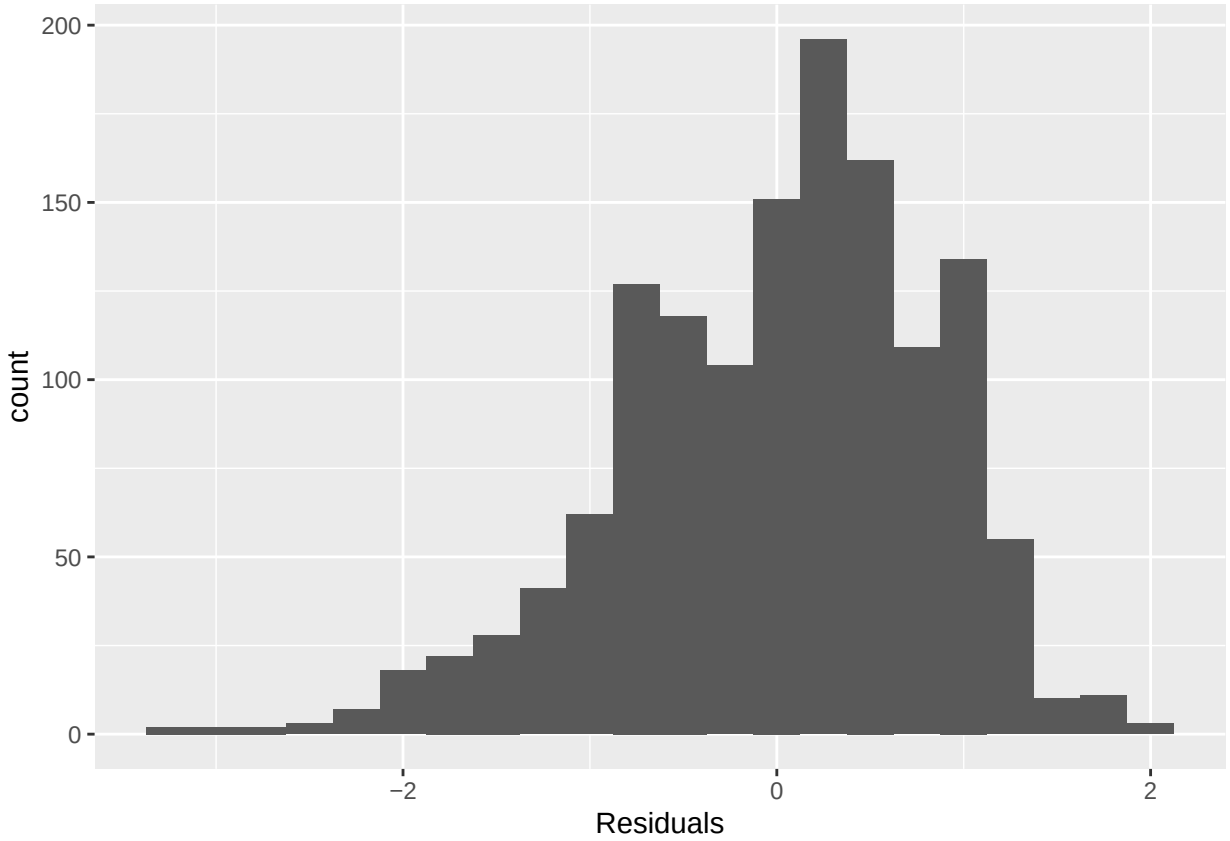
I wanted to double check the normality and linarity assumptions.

```
ggplot(data = m3, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



It appears the linear assumption holds with the residuals centered around 0 and evenly distributed above and below with no apparent pattern in the data.

```
ggplot(data = m3, aes(x = .resid)) +
  geom_histogram(binwidth = .25) +
  xlab("Residuals")
```



The histogram seems to show a normal distribution, just with a slight left skew.

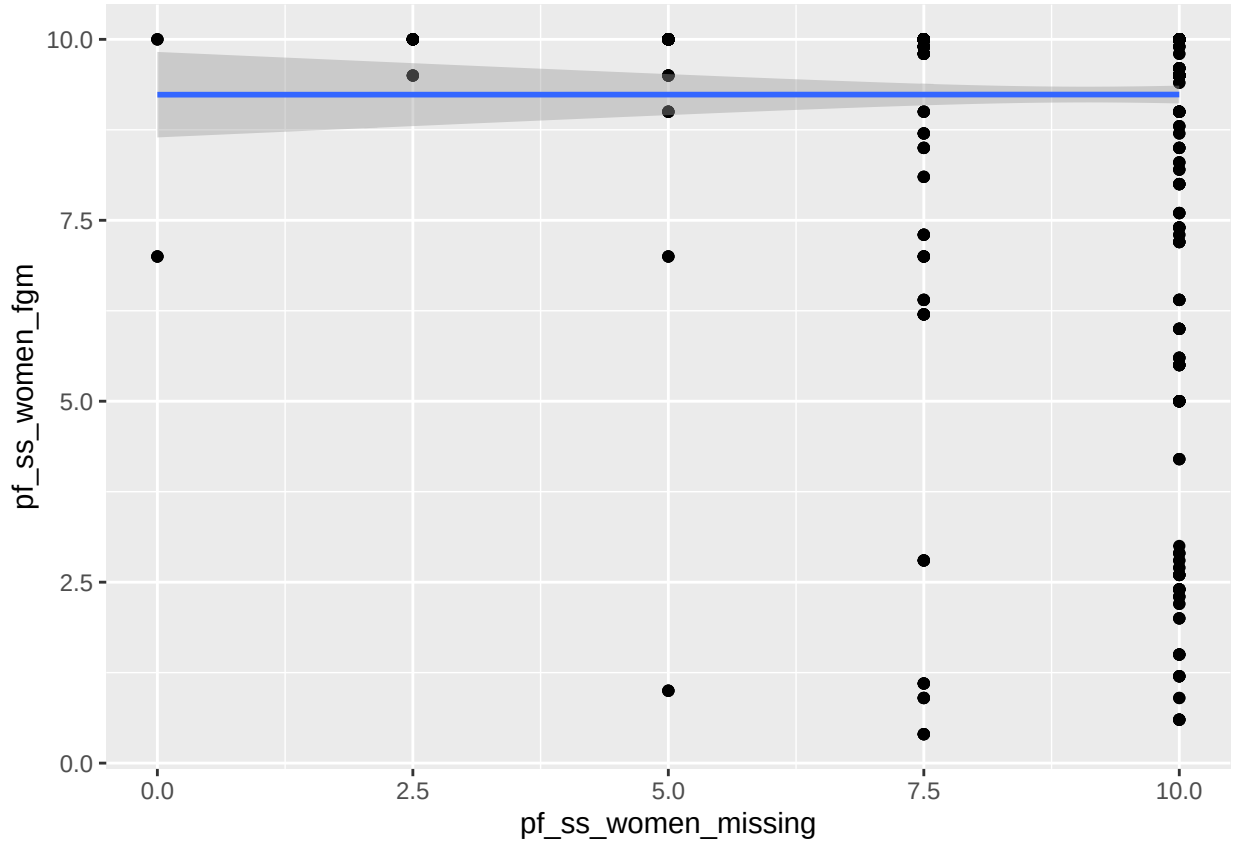
- How does this relationship compare to the relationship between `pf_expression_control` and `pf_score`? Use the R^2 values from the two model summaries to compare. Does your independent variable seem to predict your dependent one better? Why or why not?

The R squared value for the relationship between `pf_expression_control` and `pf_score` is 0.6342. The R squared value for the relationship between `ef_regulation_labor_minwage` and `ef_score` is 0.09965. The relationship is much stronger for the `pf_expression_control` and `pf_score` because it is closer to 1. My independent variable does NOT seem to predict my dependent one better then the `pf_expression_control` and `pf_score`. Apparently `ef_regulation_labor_minwage` and `ef_score` are not as linked as I thought.

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

I found my previous results surprising with there not being a strong relationship between `ef_regulation_labor_minwage` and `ef_score`. In looking for another relationship

```
hfi |> ggplot(aes(x = pf_ss_women_missing, y = pf_ss_women_fgm)) +
  geom_point() +
  geom_smooth(method = lm)
```




```
hfi %>%
  summarise(cor(pf_ss_women_missing, pf_ss_women_fgm, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(pf_ss_women_missing, pf_ss_women_fgm, use = "complete.obs")`
##                                                              <dbl>
## 1                                                         0.000162
```

```
m4 <- lm(pf_ss_women_fgm ~ pf_ss_women_missing,  data= hfi)
summary(m3)
```

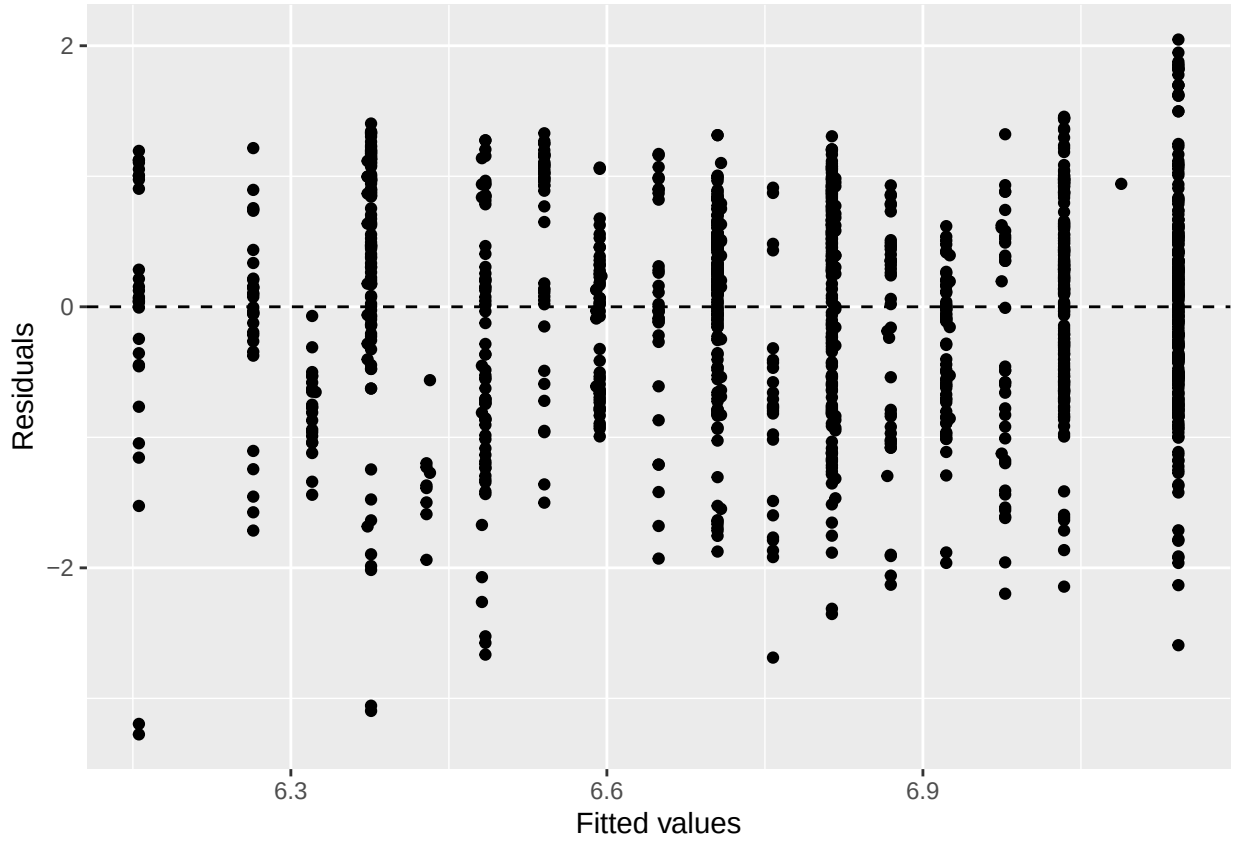
```
##
## Call:
## lm(formula = ef_score ~ ef_regulation_labor_minwage, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2757 -0.5683  0.1175  0.5792  2.0475
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.155742   0.056412  109.12   <2e-16 ***
## ef_regulation_labor_minwage 0.098680   0.008028   12.29   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8299 on 1365 degrees of freedom
##   (91 observations deleted due to missingness)
## Multiple R-squared:  0.09965,    Adjusted R-squared:  0.09899
## F-statistic: 151.1 on 1 and 1365 DF,  p-value: < 2.2e-16
```

The relationship between `pf_ss_women_missing` (missing women) and `pf_ss_women_fgm` (female genital mutilization) is weak - with an R squared of only 0.172. The equation for the line is pf_ss_women_fgm = 4.0937 + 0.3398*pf_ss_women_missing. I was expecting there to be a stronger relationship because both are pertaining to women - one the rights they have and the other the violence inflicted on them.

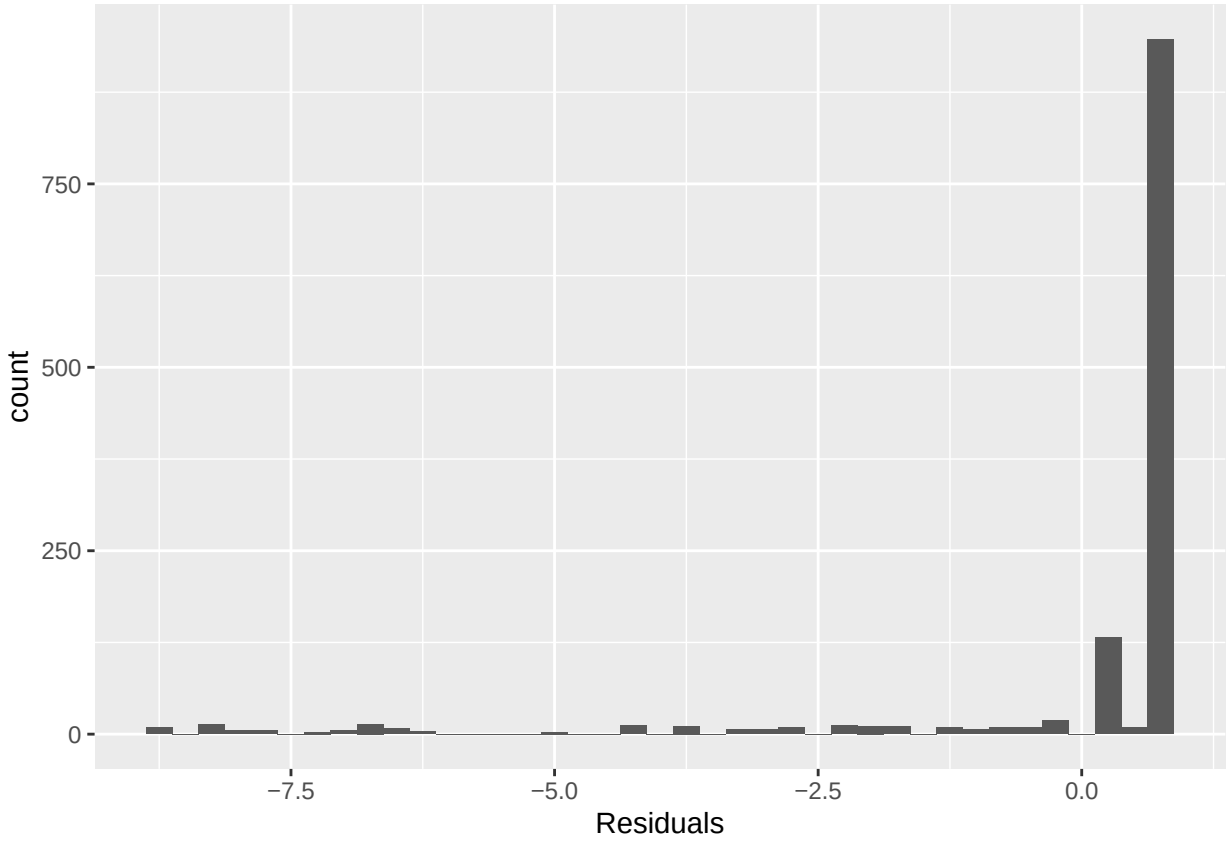
I wanted to double check the normality and linarity assumptions.

```
ggplot(data = m3, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



It appears the linear assumption holds with the residuals centered around 0 and evenly distributed above and below with no apparent pattern in the data.

```
ggplot(data = m4, aes(x = .resid)) +
  geom_histogram(binwidth = .25) +
  xlab("Residuals")
```



The histogram does not show a normal distribution at all, this is reason for discrediting the linear regression model.
