Introduction to linear regression

William Jasmine

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?

WJ Response:

```
dim(hfi)
```

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

Based on the output above, we see that the `hfi` dataset has 123 columns of 1,458 observations.

---

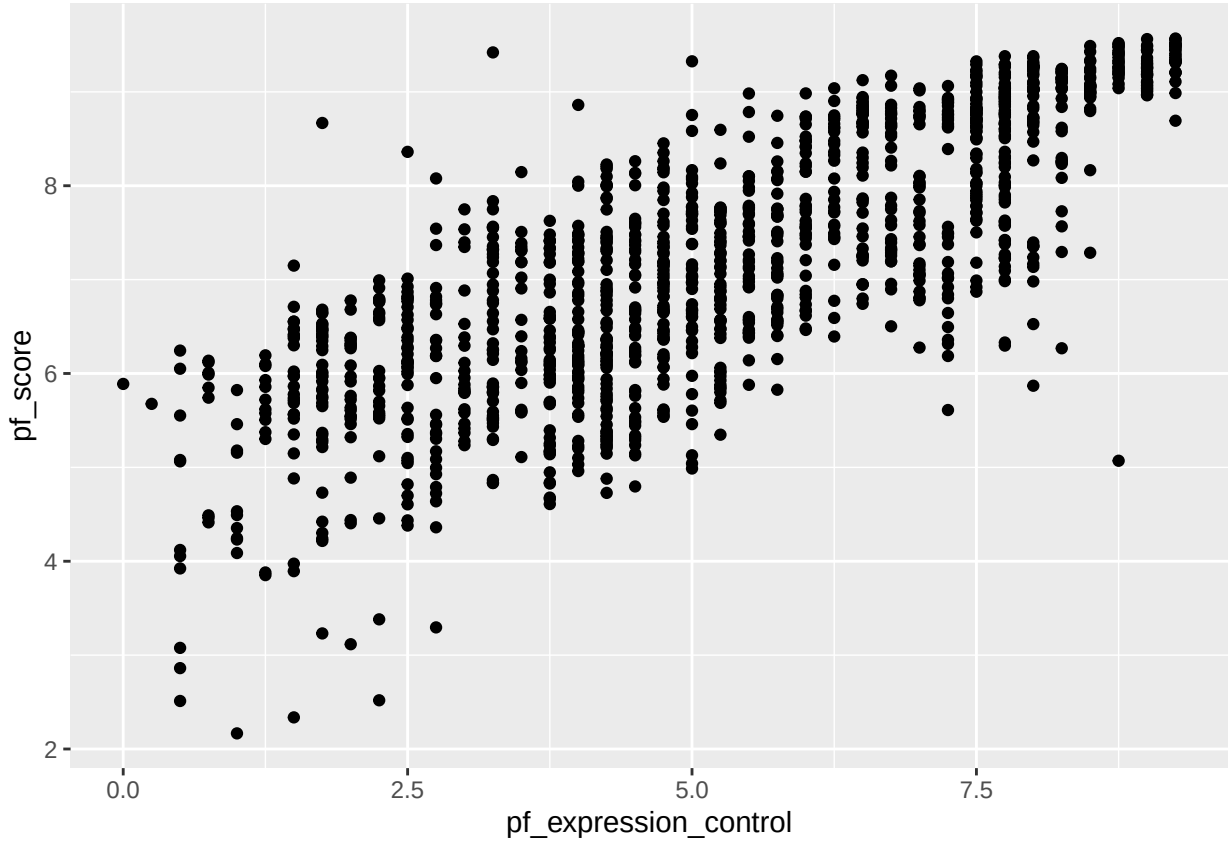
2. What type of plot would you use to display the relationship between the personal freedom score, `pf_score`, and one of the other numerical variables? Plot this relationship using the variable `pf_expression_control` as the predictor. Does the relationship look linear? If you knew a country's `pf_expression_control`, 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?

WJ Response:

Since both `pf_score` and `pf_expression_control` are numerical variables, the best visualization to see if the two are related is a scatter plot, such as the one created below:

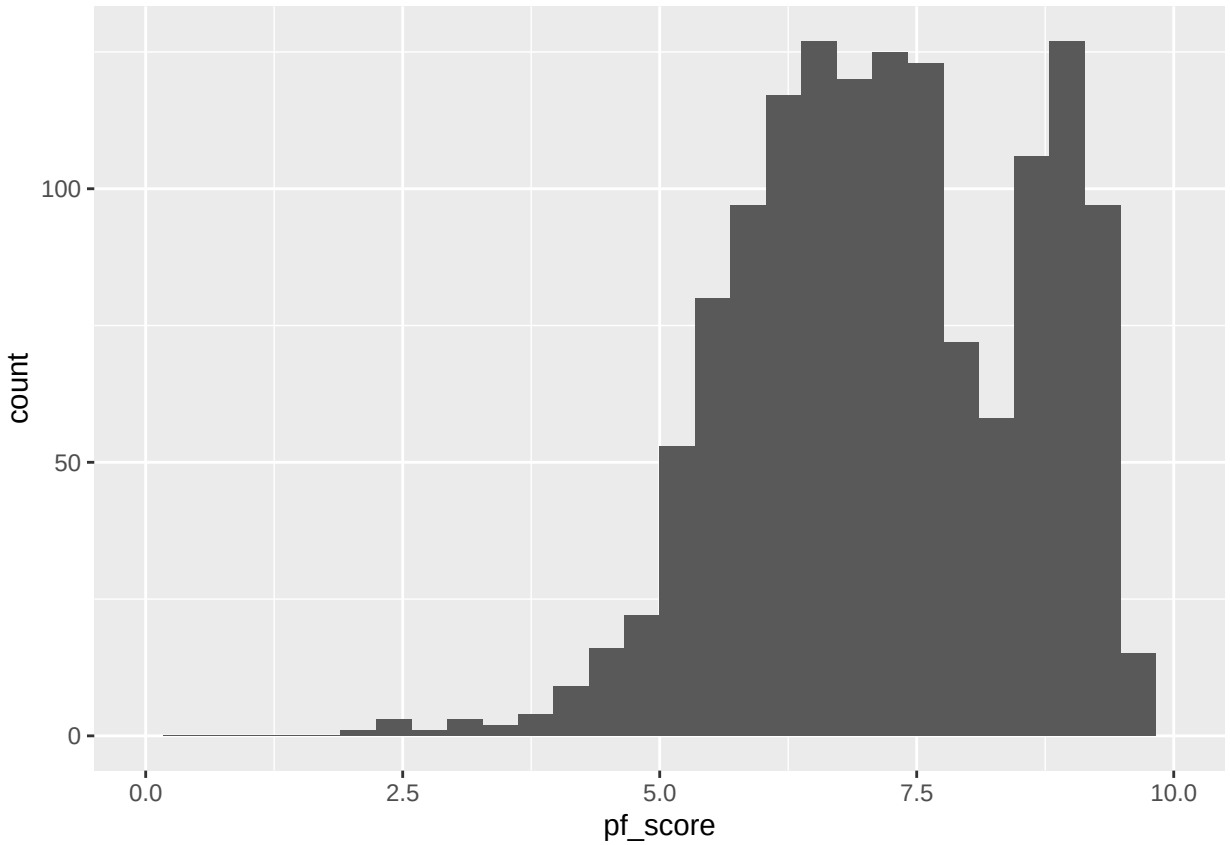
```
ggplot(data = hfi, aes(y = pf_score, x = pf_expression_control)) +
  geom_point()
```



The data does seem to have a slight positive correlation, meaning we can hypothesize that higher `pf_expression_control` values tend to correlate with higher `pf_score` values.

While an ideal linear regression requires a dependent variable that is continuous (in this case `pf_score` is bounded between 0 and 10), it appears as though the majority of values do not collect at these bounds. This is shown in the histogram below:

```
ggplot(data = hfi, aes(x = pf_score)) +
  geom_histogram() + xlim(0, 10)
```



As such, using a linear regression model to predict values of `pf_scores` should be okay.

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` data set 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.

WJ Response:

The relationship shown in the scatter plot above is the result of a positive correlation given the upward trend in `pf_score` as `pf_expression_score` increases. Visual analysis of this positive correlation indicates that it is likely quite strong, given that it is easy to visualize a positively sloped line that captures the trend of the scatter plot. This is confirmed by the ~0.8 correlation coefficient value shown above.

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?

WJ Response:

The smallest sum of squared residuals I was able to achieve was 968.692.

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?

WJ Response:

The code below provides the output of running a linear regression model in which we use `pf_expression_control` to predict `hf_score`:

```
lin_model <- lm(hf_score ~ pf_expression_control, data = hfi)
m <- unname(lin_model$coefficients[2])
b <- unname(lin_model$coefficients[1])
eqn <- sprintf("$\\hat{y} = %.2f + (%.2f \\beta_1$)", b, m)
summary(lin_model)
```

```
##
## 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
```

Based on our results above, if $\hat{y}$ = `hf_score` and β_1 = `pf_expression_control`, then the equation that relates the two is as follows:

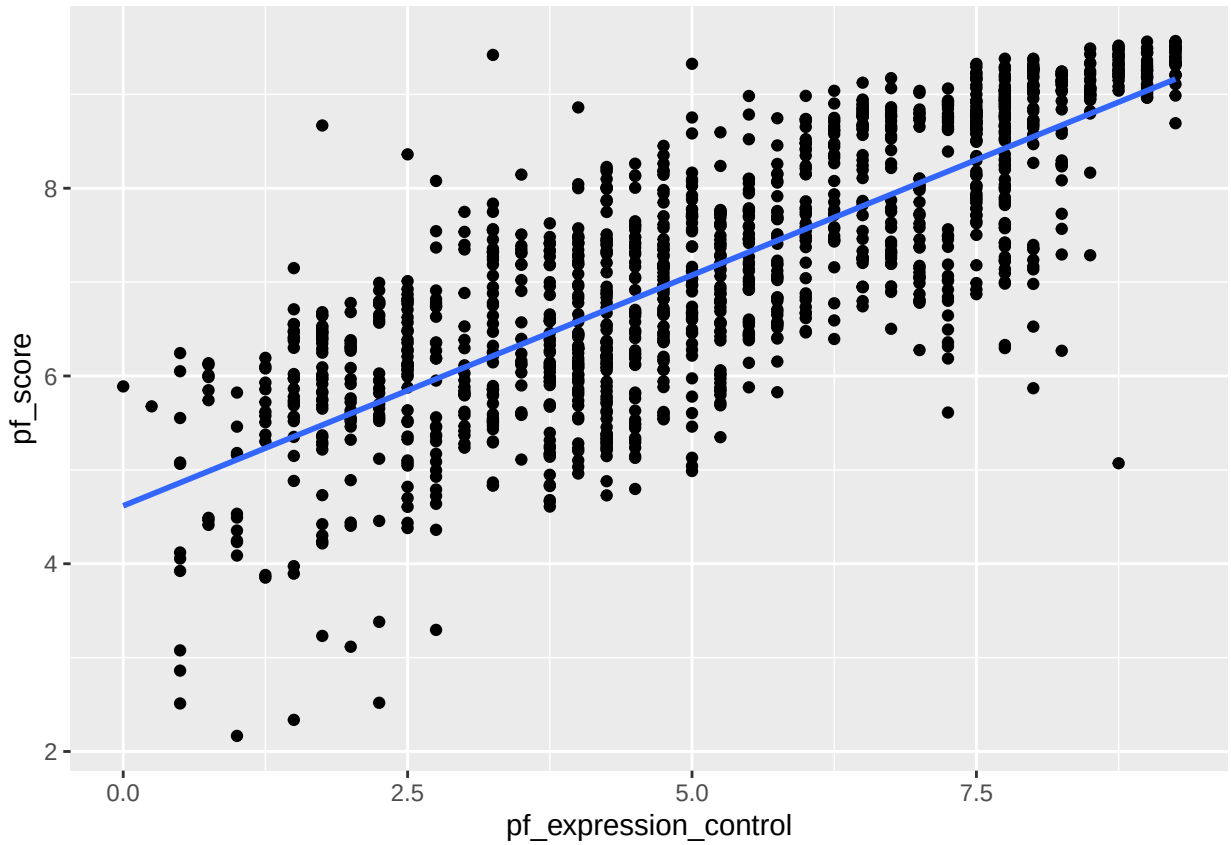
$\hat{y} = 5.15 + (0.35\beta_1)$

The positive slope tells us that these two variables have a positive correlation: an increase in `pf_expression_control` leads in to an increase in `hf_score`. This makes sense seeing as more political pressure on media content is likely to lead to a decline in human freedom (inversed here since a `pf_expression_control` score of 0 means that there is intense pressure).

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 score 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?

WJ Response:

Since we now have a model to estimate `pf_score` from `pf_expression_control`, we can use it to predict the `pf_score` when `pf_expression_control` = 6.7:

```
m = unname(m1$coefficients[2])
b = unname(m1$coefficients[1])

pred <- b + m * 6.7
pred
```

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

The prediction in this case is ~7.9. To get a residual for this value, the below code searches for the data point that is closest to having a `pf_expression_control` value of 6.7 and then extracts the `pf_score`. It then compares this actual `pf_score` to our prediction shown above:

```r
actual <- hfi %>%
  mutate(res = abs(pf_expression_control - 6.7)) %>%
    arrange(desc(res)) %>%
      select(pf_score) %>%
        filter(row_number()==1) %>% pull()

residual <- actual - pred
residual
```

```
## [1] -2.020804
```

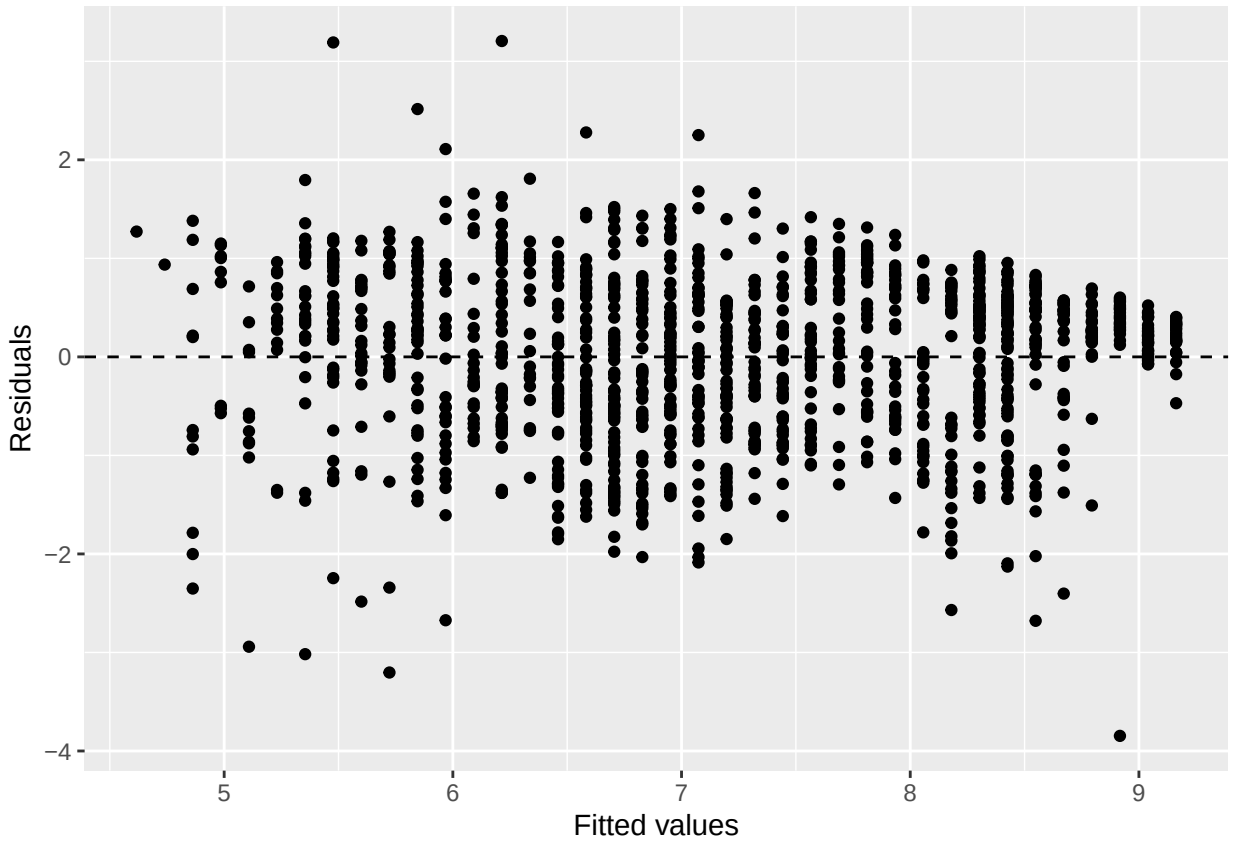
According the the output above, using the data point having the value of `pf_expression_control` closest to 6.7 results in a residual of ~2.02 when comparing it to our prediction of ~7.9.

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.

```r
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.

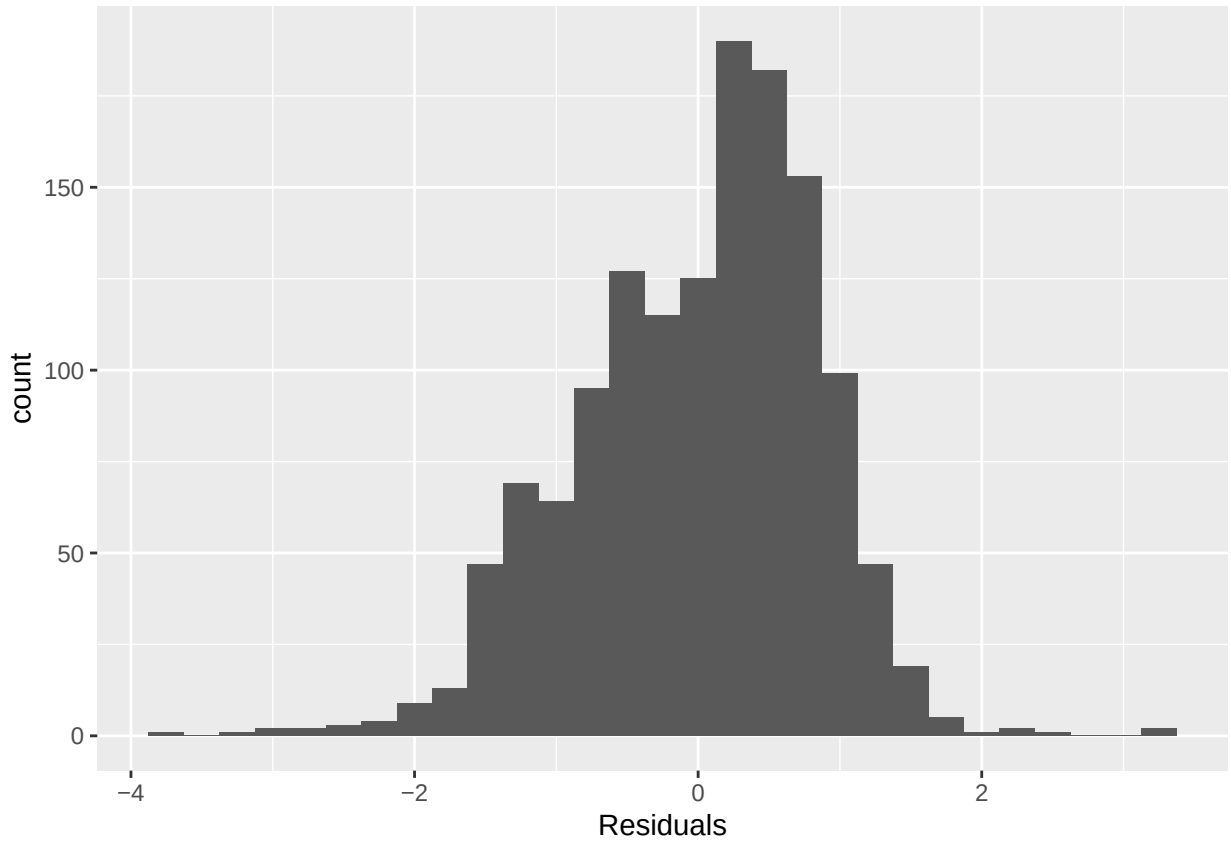
7. Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between the two variables?

WJ Response:

No, there doesn't appear to be any correlation between the residuals and the predicted values. This gives us confidence in the quality of our fit, seeing as if the fit is correctly modeling the data the distribution of residual lengths should look about the same for different sub-ranges of the predicted values. If there were any type of distinct shape in the above plot it would indicate that the relationship between our two variables is not actually linear.

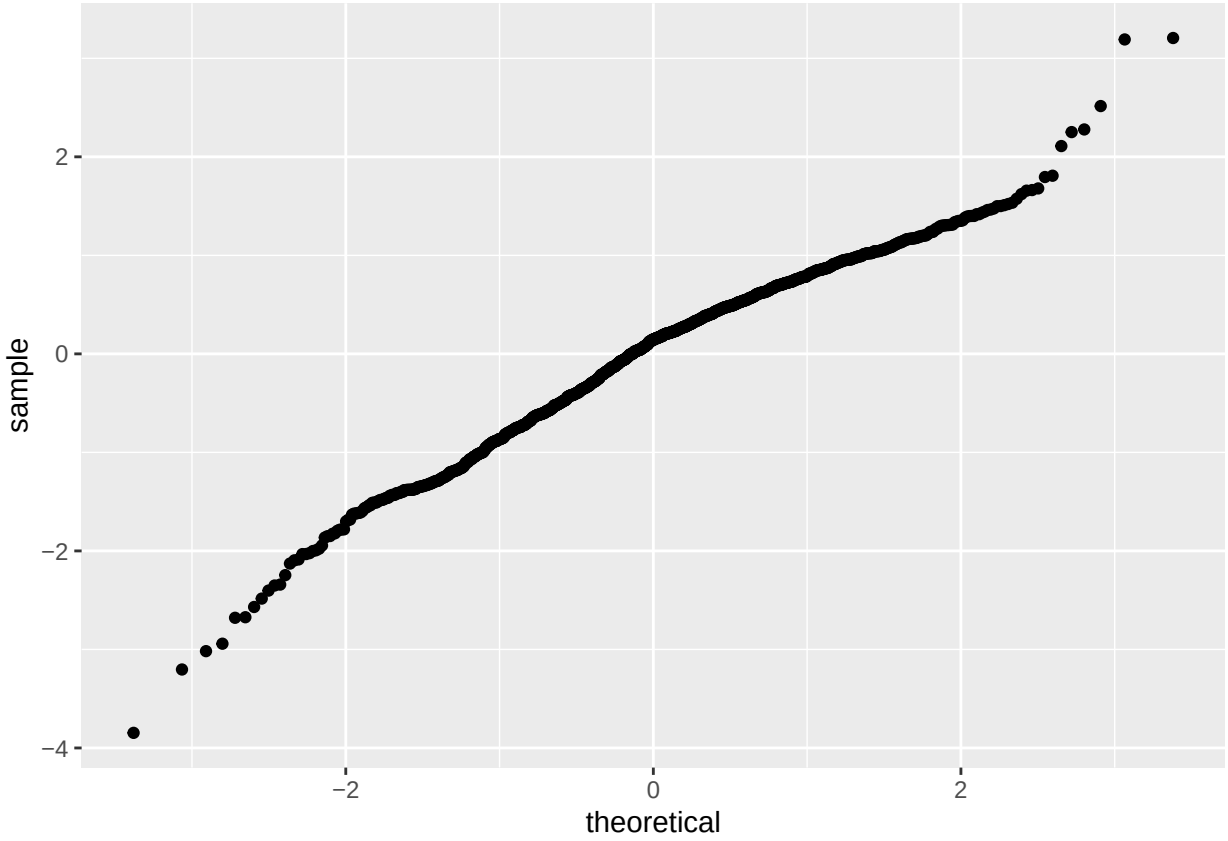
Nearly normal residuals: To check this condition, we can look at a histogram

```
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?

WJ Response:

Yes, the histogram of the residuals appears normal upon visual inspection, a fact that is further evidenced by the apparent linearity of its analogous QQ plot.

Constant variability:

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

WJ Response:

Yes: looking at the plot, it appears that the spread of residuals at each of the predicted values is relatively similar. There is a slight downward trend in residual spread towards the higher predicted values, but not enough to warrant that the assumption of homoscedasticity is not met.

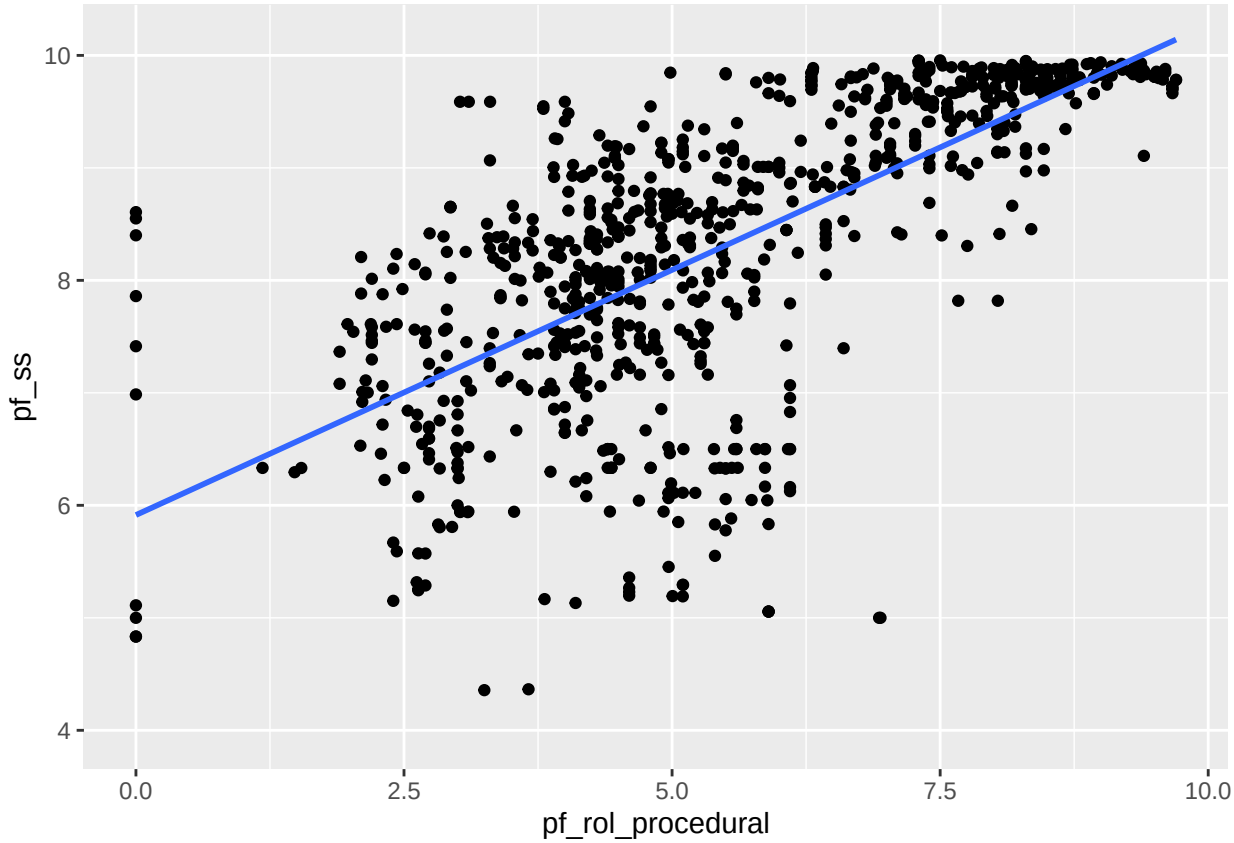
More Practice

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?

WJ Response:

I chose to test how the personal sense of security and safety (`pf_ss`) correlates with the role the procedural justice a given country (`pf_rol_procedural`). My hypothesis is that people will feel a higher sense of safety and security if the country they live in has systems in place that allow for procedural justice to take place. The following shows a scatterplot of the two variables along with their linear fit:

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



There does appear to be a linear correlation between these two variables that is consistent with my aforementioned hypothesis.

---

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?

WJ Response:

The code below prints out the details of the linear model shown in the graph above:

```
lin_model <- lm(data = hfi, formula = pf_ss ~ pf_rol_procedural)
summary(lin_model)
```

```
##
## Call:
## lm(formula = pf_ss ~ pf_rol_procedural, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9398 -0.3569  0.1601  0.5722  2.6927
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.91391    0.09164   64.53   <2e-16 ***
## pf_rol_procedural  0.43577    0.01537   28.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.948 on 878 degrees of freedom
##   (578 observations deleted due to missingness)
## Multiple R-squared:  0.4781, Adjusted R-squared:  0.4775
## F-statistic: 804.3 on 1 and 878 DF,  p-value: < 2.2e-16
```

According to the output, the R^2 value in this case is 0.4781, meaning that `pf_rol_procedural` accounts for approximately 47.81% of the variance seen in `pf_ss`. This is slightly worse than the model we saw that correlated `pf_score` and `pf_expression_control`, but still pretty good!

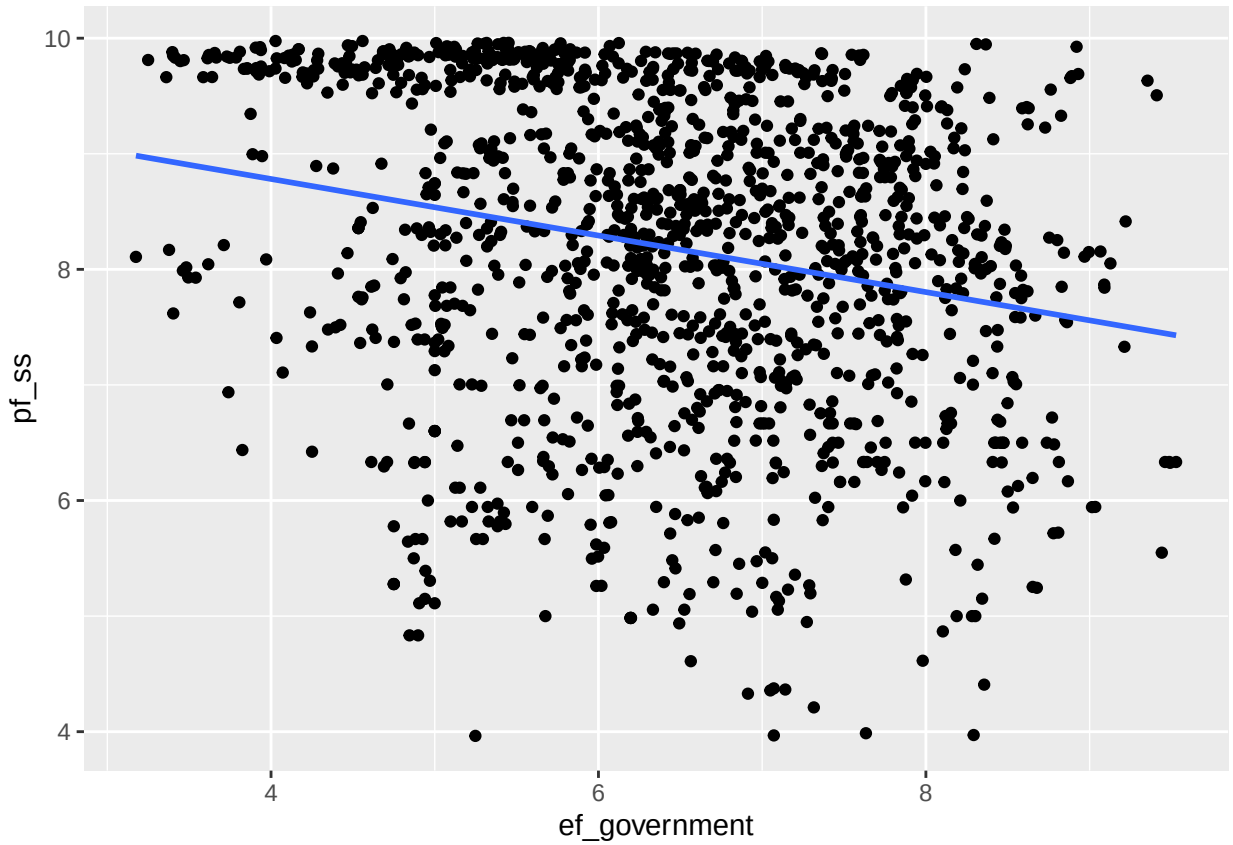
---

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

WJ Response:

The below plot models the relationship between one's feeling of safety and security (`pf_ss`) as a function of the size of the government they live in (`ef_government`):

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



As is clear in the plot above, there appears to be a negative correlation between these two variables. In other words, it appears that countries with larger governments appear to have citizens that feel a lesser sense of safety and security. This correlation is confirmed by analyzing the results of the linear fit shown below:

```
lin_model <- lm(data = hfi, formula = pf_ss ~ ef_government)
summary(lin_model)
```

```
##
## Call:
## lm(formula = pf_ss ~ ef_government, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5127 -0.9362  0.2548  1.1053  2.3476
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.76033    0.18660  52.305   <2e-16 ***
## ef_government -0.24462    0.02836  -8.624   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.334 on 1376 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.05128,    Adjusted R-squared:  0.05059
## F-statistic: 74.38 on 1 and 1376 DF,  p-value: < 2.2e-16
```

This correlation is certainly interesting, though one can only speculate as to the reasoning behind it. A couple

ideas do come to mind: is there higher incidence of corruption in larger governments? Do larger governments breed political discord? Further analysis would be needed to identify the cause, but I definitely was surprised by the result.