Introduction to linear regression

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 and Data

```
library(tidyverse)
library(openintro)
library(gt)
library(magrittr)
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: Dimensions

What are the dimensions of the dataset?

There are a lot. Seems like an odd question that will throw out a lot of output to the screen but here we go. Hfi has the following 123 dimensions.

```
colnames(hfi)
```

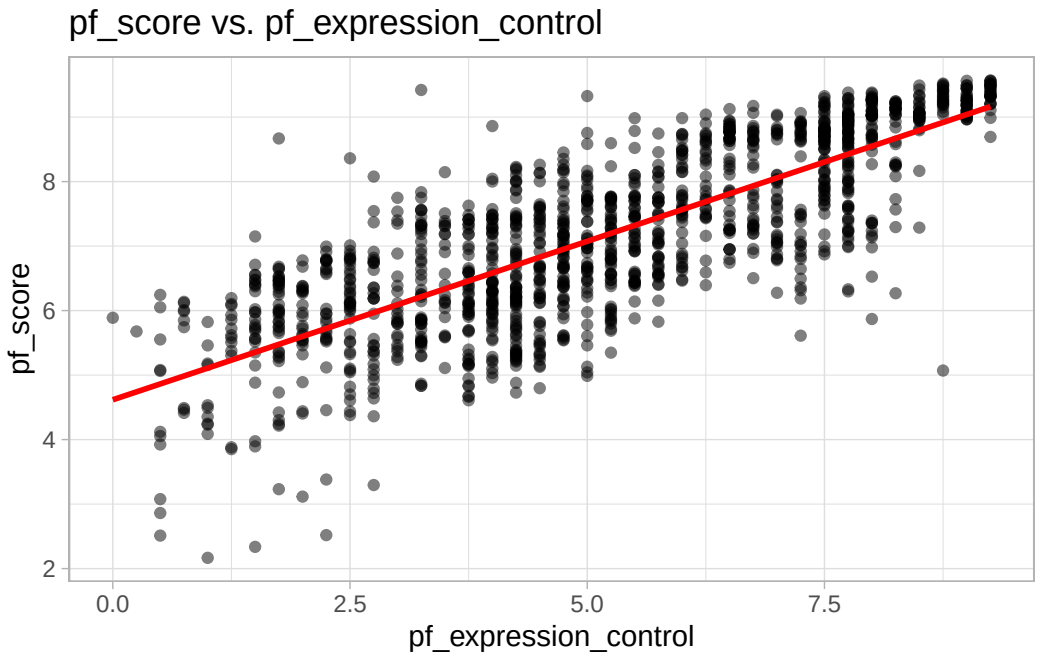
```
  [1] "year"                                 "ISO_code"
  [3] "countries"                            "region"
  [5] "pf_rol_procedural"                    "pf_rol_civil"
  [7] "pf_rol_criminal"                      "pf_rol"
  [9] "pf_ss_homicide"                       "pf_ss_disappearances_disap"
 [11] "pf_ss_disappearances_violent"         "pf_ss_disappearances_organized"
 [13] "pf_ss_disappearances_fatalities"      "pf_ss_disappearances_injuries"
 [15] "pf_ss_disappearances"                 "pf_ss_women_fgm"
 [17] "pf_ss_women_missing"                  "pf_ss_women_inheritance_widows"
 [19] "pf_ss_women_inheritance_daughters"    "pf_ss_women_inheritance"
 [21] "pf_ss_women"                          "pf_ss"
 [23] "pf_movement_domestic"                 "pf_movement_foreign"
 [25] "pf_movement_women"                    "pf_movement"
 [27] "pf_religion_estop_establish"          "pf_religion_estop_operate"
 [29] "pf_religion_estop"                    "pf_religion_harassment"
 [31] "pf_religion_restrictions"             "pf_religion"
 [33] "pf_association_association"           "pf_association_assembly"
 [35] "pf_association_political_establish"   "pf_association_political_operate"
 [37] "pf_association_political"             "pf_association_prof_establish"
 [39] "pf_association_prof_operate"          "pf_association_prof"
 [41] "pf_association_sport_establish"       "pf_association_sport_operate"
 [43] "pf_association_sport"                 "pf_association"
 [45] "pf_expression_killed"                 "pf_expression_jailed"
 [47] "pf_expression_influence"              "pf_expression_control"
 [49] "pf_expression_cable"                  "pf_expression_newspapers"
 [51] "pf_expression_internet"               "pf_expression"
```

```
 [53] "pf_identity_legal"                  "pf_identity_parental_marriage"
 [55] "pf_identity_parental_divorce"       "pf_identity_parental"
 [57] "pf_identity_sex_male"               "pf_identity_sex_female"
 [59] "pf_identity_sex"                    "pf_identity_divorce"
 [61] "pf_identity"                        "pf_score"
 [63] "pf_rank"                            "ef_government_consumption"
 [65] "ef_government_transfers"            "ef_government_enterprises"
 [67] "ef_government_tax_income"           "ef_government_tax_payroll"
 [69] "ef_government_tax"                  "ef_government"
 [71] "ef_legal_judicial"                  "ef_legal_courts"
 [73] "ef_legal_protection"                "ef_legal_military"
 [75] "ef_legal_integrity"                 "ef_legal_enforcement"
 [77] "ef_legal_restrictions"              "ef_legal_police"
 [79] "ef_legal_crime"                     "ef_legal_gender"
 [81] "ef_legal"                           "ef_money_growth"
 [83] "ef_money_sd"                        "ef_money_inflation"
 [85] "ef_money_currency"                  "ef_money"
 [87] "ef_trade_tariffs_revenue"           "ef_trade_tariffs_mean"
 [89] "ef_trade_tariffs_sd"                "ef_trade_tariffs"
 [91] "ef_trade_regulatory_nontariff"      "ef_trade_regulatory_compliance"
 [93] "ef_trade_regulatory"                "ef_trade_black"
 [95] "ef_trade_movement_foreign"          "ef_trade_movement_capital"
 [97] "ef_trade_movement_visit"            "ef_trade_movement"
 [99] "ef_trade"                           "ef_regulation_credit_ownership"
[101] "ef_regulation_credit_private"       "ef_regulation_credit_interest"
[103] "ef_regulation_credit"               "ef_regulation_labor_minwage"
[105] "ef_regulation_labor_firing"         "ef_regulation_labor_bargain"
[107] "ef_regulation_labor_hours"          "ef_regulation_labor_dismissal"
[109] "ef_regulation_labor_conscription"   "ef_regulation_labor"
[111] "ef_regulation_business_adm"         "ef_regulation_business_bureaucracy"
[113] "ef_regulation_business_start"       "ef_regulation_business_bribes"
[115] "ef_regulation_business_licensing"   "ef_regulation_business_compliance"
[117] "ef_regulation_business"             "ef_regulation"
[119] "ef_score"                           "ef_rank"
[121] "hf_score"                           "hf_rank"
[123] "hf_quartile"
```

Exercise 2: pf_score to pf_expression_control

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?

```
ggplot(hfi, aes(x = pf_expression_control, y = pf_score)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  theme_minimal() +
  labs(title = "pf_score vs. pf_expression_control",
       x = "pf_expression_control",
       y = "pf_score") +
  theme_light()
```



It looks pretty linear to me, so yes I'd say we can predict. The problem is the variance. With a scale of 10 and a standard deviation of 2, that's not very accurate.

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 mark-down 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: Describe the relationship

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.

Those two variables have an high positive linear correlation to each other. And yes you can see all that in the chart, but the more important indicator is the correlation of .79. That's high, really high.

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.

Exercise 4: Choose a line and describe

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?

954 was my lowest guess after trying it a dozens times, but I didn't ask any of my neighbors. And WOW did I get close with my eyeball guesses!!

```
model <- lm(pf_score ~ pf_expression_control, data=hfi)
sum(residuals(model)^2)  # [1] 952.1532
```

```
[1] 952.1532
```

```
# summary(model)
```

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)
```

well how about that?!

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.

Exercise 5: pf_expression_control and hf_score

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?

There is a positive correlation. Looking at the correlation, summary, lm output, and scope and intercept, I bet the plot looks similar to pf_expresison_control and pf_score.

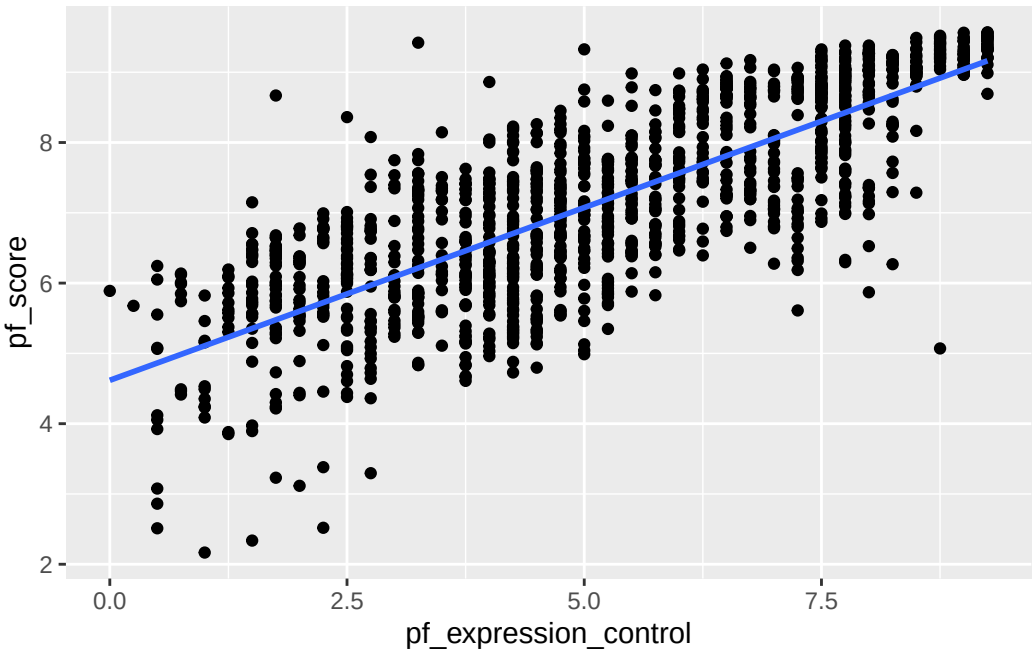
```
# hfi %>%  summarise(cor(pf_expression_control, hf_score, use = "complete.obs"))
# 0.760

# lm(hf_score ~ pf_expression_control, data = hfi)
# Coefficients:
#           (Intercept)  pf_expression_control
#                5.1537                 0.3499
```

$$\hat{y} = 5.1537 + 0.3499 \times pf\_expression\_control$$

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.

Exercise 6: What are regression lines are for?

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?

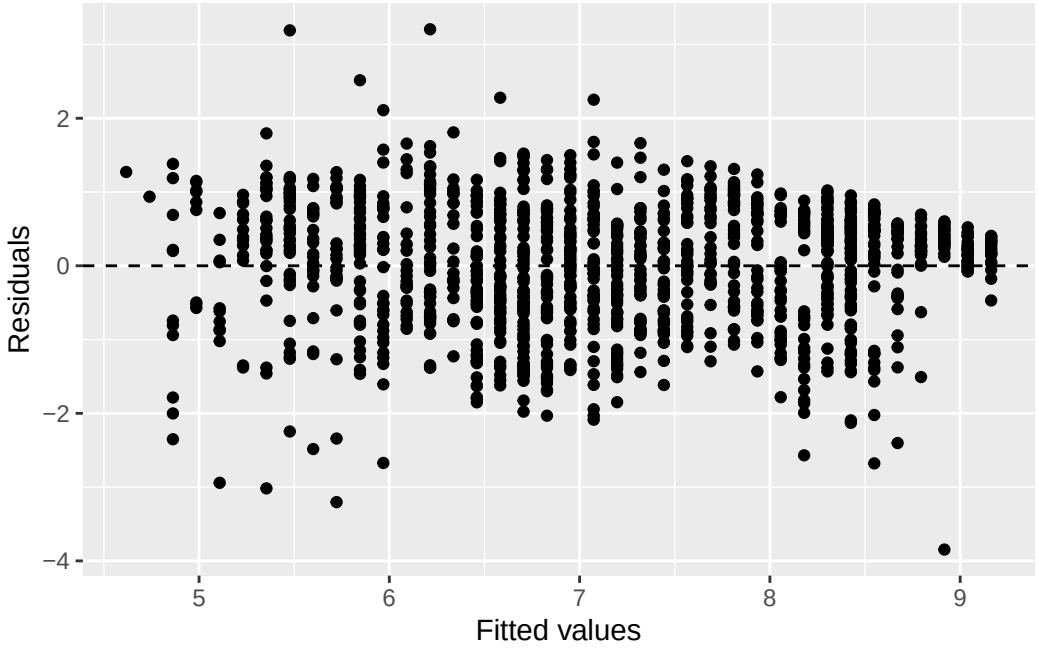
If they only saw the line of y = 0.49143 ×pf_expression_control + 4.61707, they would solve for y and there would be no residual. That's the point of regression line – you use it to predict.

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")

# It is breaking my brain that that above code even works so I rewrote it.
# It's definitely not a method according to the str() function.
# If the m1 object is going to have a .resid variable, it should show up when
# you run str(m1)


ggplot(data = m1, aes(x = m1$fitted, y = m1$residuals)) +
  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: Residuals versus fits

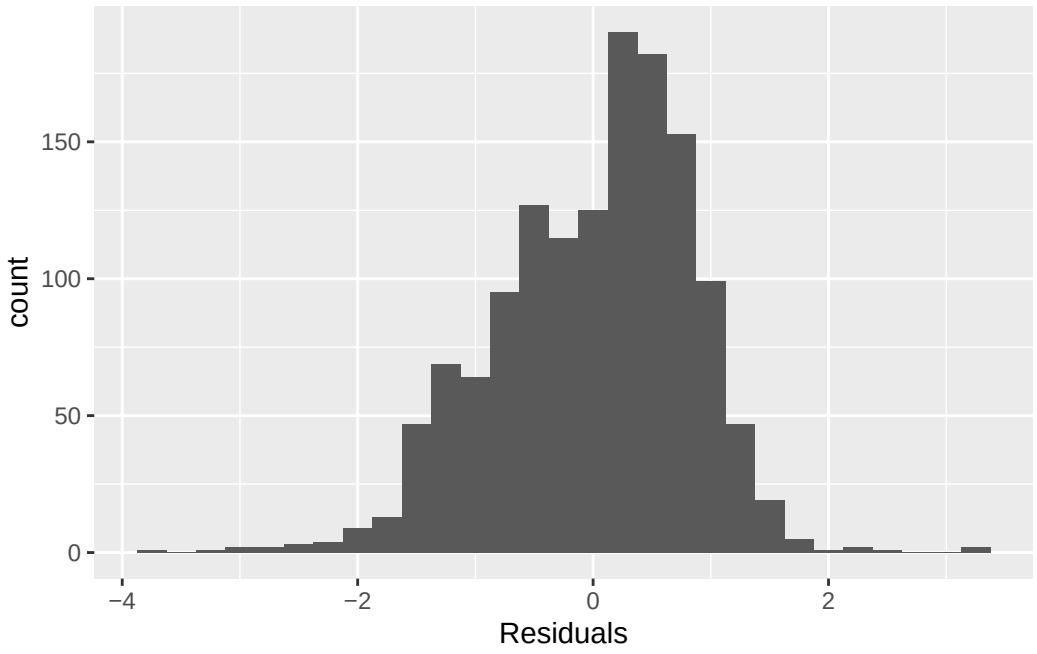
Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between the two variables?

It looks like all those dots are sort of evenly distributed above and below the zero line. That's what I would expect, to minimize or make that line as close to zero residuals (error) that's what we are looking for.

PS I made a comment in the code for making this chart as well.

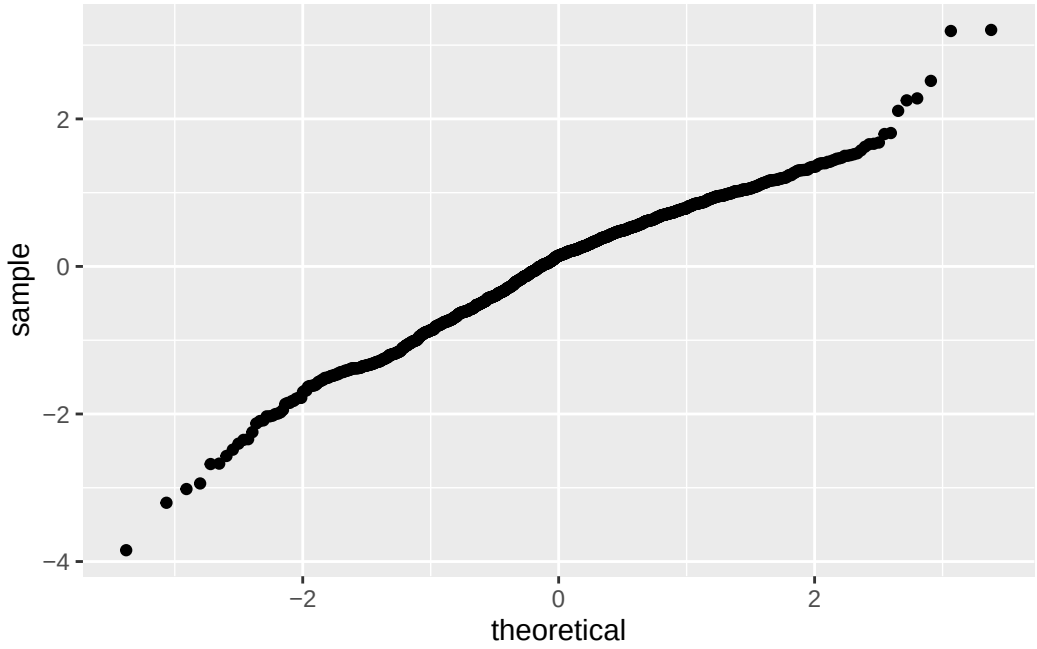
Nearly normal residuals: To check this condition, we can look at a histogram

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



or a normal probability plot of the residuals.

```
ggplot(data = m1, aes(sample = m1$residuals)) +
  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: Nearly normal residuals

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

Both conditions are met. The histogram centers around zero with not so much outlier data. This a fairly normal probability plot. And in the QQ plot, the line is pretty straight. Deviations from the line would indicate non-normality, but there really aren't any.

More Practice

Exercise 9: Pick another pair

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?

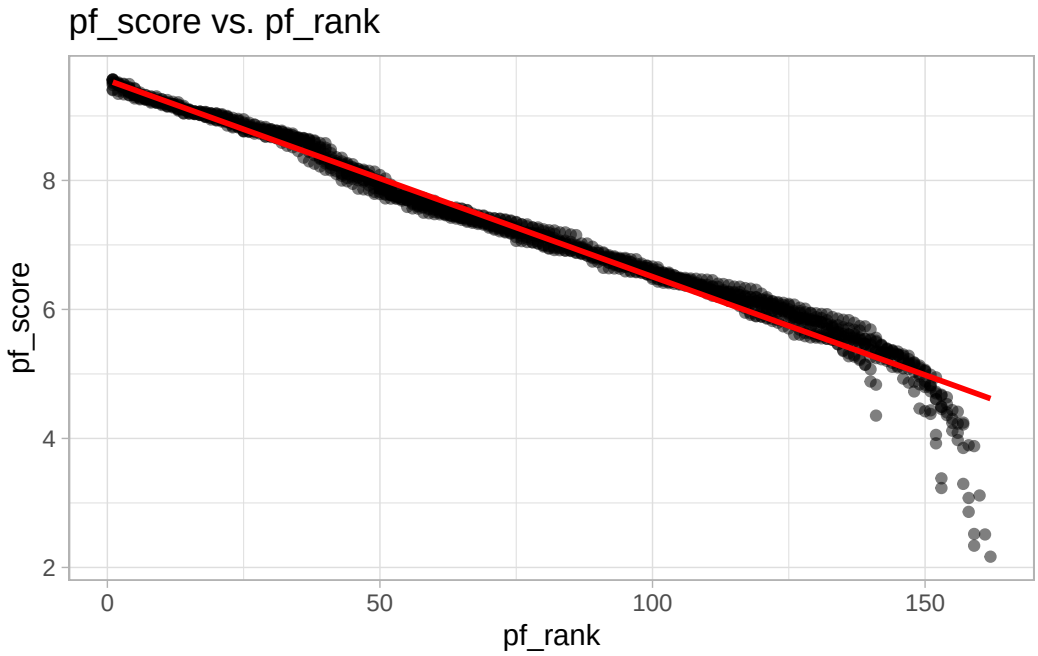
Hmm, my vote is let's find the most correlated of them all and see what almost perfect correlation looks like. Objective, diag the list of correlation variables, pick the top one, and then look at that.

```
library(dplyr)

hfi_numeric <- hfi %>% select(-c("ISO_code", "countries", "region"))
## If null, replace with mean.
hfi_numeric_imputed <- hfi_numeric %>% mutate(across(everything(), ~ifelse(is.na(.), mean(
cor_matrix <- cor(hfi_numeric_imputed, use = "complete.obs")

# Exclude the diagonal (set it to 0)
```

```r
diag(cor_matrix) <- 0

# Flatten the matrix and sort by absolute value
cor_values <- as.data.frame(as.table(cor_matrix))
cor_values <- cor_values %>% arrange(desc(abs(Freq))) %>% head(10)

# Print the pairs and their correlation values using correct column names
for (i in 1:nrow(cor_values)) {
  cat(colnames(cor_matrix)[as.numeric(cor_values$Var1[i])], "&",
      colnames(cor_matrix)[as.numeric(cor_values$Var2[i])], ":",
      cor_values$Freq[i], "\n")
}
```

```
pf_rank & pf_score : -0.9862123
pf_score & pf_rank : -0.9862123
hf_rank & hf_score : -0.9843632
hf_score & hf_rank : -0.9843632
hf_quartile & hf_rank : 0.9641413
hf_rank & hf_quartile : 0.9641413
ef_rank & ef_score : -0.9494376
ef_score & ef_rank : -0.9494376
ef_trade_regulatory & ef_trade_regulatory_compliance : 0.9447433
ef_trade_regulatory_compliance & ef_trade_regulatory : 0.9447433
```

```r
# -> pf_rank & pf_score : -0.9862123

ggplot(hfi, aes(x = pf_rank, y = pf_score)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  theme_minimal() +
  labs(title = "pf_score vs. pf_rank",
       x = "pf_rank",
       y = "pf_score") +
  theme_light()
```




```
lm(pf_score ~ pf_rank, data = hfi)
```

```
Call:
lm(formula = pf_score ~ pf_rank, data = hfi)

Coefficients:
(Intercept)      pf_rank
    9.55466     -0.03048
```

Wow. Of course this makes sense, but there is a very clear relationship here. If you have a higher rank, you have a lower pf_score. I wish there was more of an explanation of what these scores represented.

$$\hat{y} = (-0.03048 \times pf\_rank) + 9.55466$$

Exercise 10: Compare

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

```
print(sprintf("f_score ~ pf_expression_control  %s | pf_score ~ pf_rank %s",
    round(sum(residuals(lm(pf_score ~ pf_expression_control, data=hfi))^2),1),
    round(sum(residuals(lm(pf_score ~ pf_rank, data=hfi))^2),1)))
```

```
[1] "f_score ~ pf_expression_control  952.2 | pf_score ~ pf_rank 71.3"
```

It's as expected. Of course one predicts the other, and the sum of square is is way lower. There's a much better fit line.

Exercise 11: What surpised you?

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

I think the only to have an educated answer for this is to start with the stuff that does not correlate well, and then think about it. That said, let's find the least correlated things, go look up the columns and then pick and talk about the one that makes the least sense.

```
hfi_numeric <- hfi %>% select(-c("ISO_code", "countries", "region"))
## If null, replace with mean.
hfi_numeric_imputed <- hfi_numeric %>% mutate(across(everything(), ~ifelse(is.na(.), mean(
cor_matrix <- cor(hfi_numeric_imputed, use = "complete.obs")

# Exclude the diagonal (set it to 0)
diag(cor_matrix) <- 0

# Flatten the matrix and sort by absolute value
cor_values <- as.data.frame(as.table(cor_matrix))
cor_values <- cor_values %>% arrange(desc(abs(Freq))) %>% head(500)
```

```r
cor_pairs_df <- cor_values %>%
  transmute(
    Pair = paste(colnames(cor_matrix)[as.numeric(Var1)], "&",
                 colnames(cor_matrix)[as.numeric(Var2)]),
    Correlation = Freq
  )

# > glimpse(cor_pairs_df)
# Rows: 500
# Columns: 2
# $ Pair        <chr> "pf_rank & pf_score", "pf_score & pf_rank", "hf_rank & hf_…
# $ Correlation <dbl> -0.9862123, -0.9862123, -0.9843632, -0.9843632, 0.9641413,…

# It was trial and error to find the lowest correlation numbers, but here they are, about
cor_pairs_df %>% filter(Correlation < 0.675 & Correlation > -0.675)
```

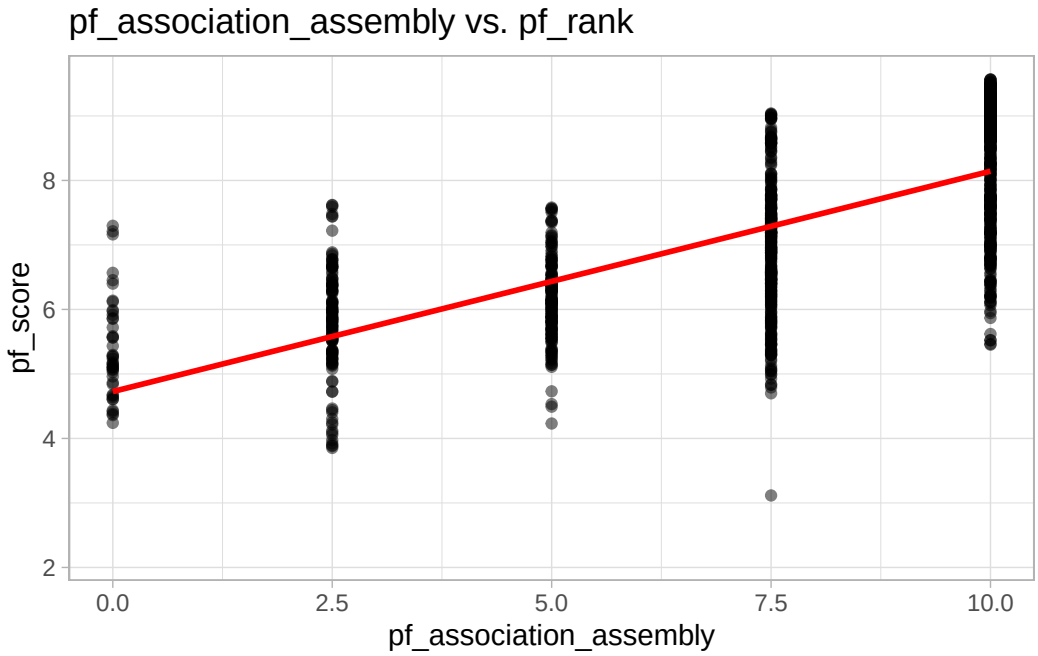
```
                                                    Pair Correlation
1                  pf_religion_estop & pf_religion_estop_operate   0.6746493
2                  pf_religion_estop_operate & pf_religion_estop   0.6746493
3                              pf_score & pf_identity_parental   0.6740800
4                              pf_identity_parental & pf_score   0.6740800
5                                        hf_quartile & pf_ss  -0.6732745
6                                        pf_ss & hf_quartile  -0.6732745
7                           hf_rank & pf_ss_women_inheritance  -0.6725140
8                           pf_ss_women_inheritance & hf_rank  -0.6725140
9                           hf_quartile & ef_trade_regulatory  -0.6723793
10                          ef_trade_regulatory & hf_quartile  -0.6723793
11                                       ef_rank & pf_rank   0.6721506
12                                       pf_rank & ef_rank   0.6721506
13                             pf_rank & pf_identity_parental  -0.6720351
14                             pf_identity_parental & pf_rank  -0.6720351
15                          pf_rank & pf_association_assembly  -0.6710553
16                          pf_association_assembly & pf_rank  -0.6710553
17 pf_association_prof_operate & pf_association_political_operate   0.6708482
18 pf_association_political_operate & pf_association_prof_operate   0.6708482
19                                     pf_rol & pf_rol_civil   0.6704210
20                                     pf_rol_civil & pf_rol   0.6704210
21                     pf_expression & pf_association_assembly   0.6703739
22                     pf_association_assembly & pf_expression   0.6703739
23            ef_regulation_business_bribes & ef_legal_integrity   0.6698419
```

```
24            ef_legal_integrity & ef_regulation_business_bribes   0.6698419
25   pf_ss_disappearances_injuries & pf_ss_disappearances_violent   0.6695305
26   pf_ss_disappearances_violent & pf_ss_disappearances_injuries   0.6695305
27                     hf_score & pf_ss_disappearances_organized   0.6694248
28                     pf_ss_disappearances_organized & hf_score   0.6694248
29                                     hf_score & ef_regulation   0.6692874
30                                     ef_regulation & hf_score   0.6692874
31                ef_regulation_business_bribes & ef_legal_crime   0.6672237
32                ef_legal_crime & ef_regulation_business_bribes   0.6672237
33         ef_regulation_labor & ef_regulation_labor_conscription   0.6668604
34         ef_regulation_labor_conscription & ef_regulation_labor   0.6668604
35                                   hf_rank & ef_trade_movement  -0.6667438
36                                   ef_trade_movement & hf_rank  -0.6667438
37              ef_regulation_labor & ef_regulation_labor_minwage   0.6665219
38              ef_regulation_labor_minwage & ef_regulation_labor   0.6665219
```

```
# Now let's go look them up.
```

pf_rank ~ pf_association_assembly -0.6710553 stands out. Freedom of assembly and personal freedoms are not more highly correlated? let's do a quick scatterplot and see, but we can see from being oen of the lowest correlated numbers that it's going to be all over the place.

```
ggplot(hfi, aes(x = pf_association_assembly, y = pf_score)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  theme_minimal() +
  labs(title = "pf_association_assembly vs. pf_rank",
       x = "pf_association_assembly",
       y = "pf_score") +
  theme_light()
```


pf_association_assembly vs. pf_rank
pf_score
8
6
4
2
0.0
2.5
5.0
7.5
10.0
pf_association_assembly
