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.

```
library(tidyverse)
library(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?

```
sprintf("The hfi table has %d columns and %d rows", ncol(hfi), nrow(hfi))
```

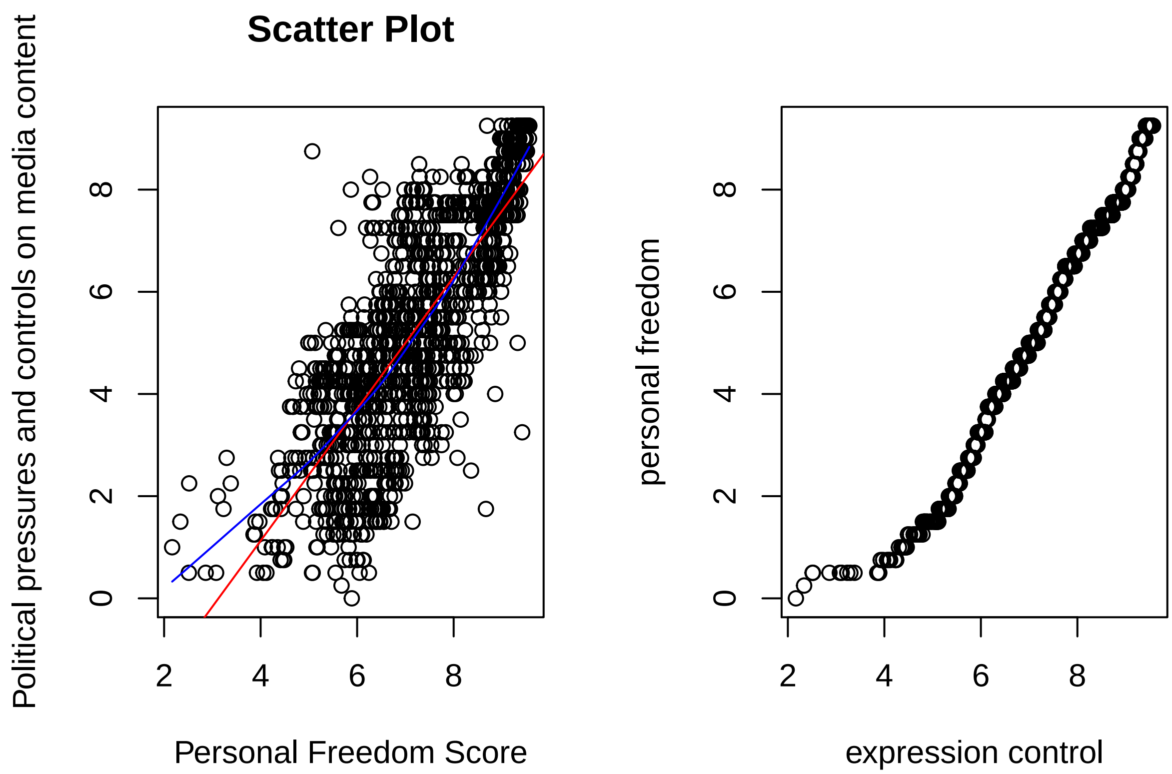
```
## [1] "The hfi table has 123 columns and 1458 rows"
```

1. 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 (x). 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 (y) ?

It looks somewhat linear although Im not sure if the wide range of the response variables is an issue for a linear model.

```
# note: expression control is somehat discrete and
#    one expression control value may map to many personal freedome scores
# View(hfi[c("year", "region", "countries", "pf_score", "pf_expression_control")])

hfi_new<-hfi %>%
  drop_na(pf_score) %>%
  drop_na(pf_expression_control)
```

```r
x<-hfi_new$pf_score
y<-hfi_new$pf_expression_control


par(mfrow=c(1,2))
plot(x, y, main = "Scatter Plot",
     xlab ="Personal Freedom Score",
     ylab="Political pressures and controls on media content" )
abline(lm(y~x), col="red") # regression line (y~x)
lines(lowess(x,y), col="blue")   # non-linear best fit


# qqplot plots one against the other
qqplot(hfi$pf_score, hfi$pf_expression_control, xlab="expression control", ylab="personal freedom")
```



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.

1. 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.

While I observe that they generally increase or decrease together, there are quite a few exceptions.

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.

```
 # x = hfi_new$pf_score

plot_ss(x = x, y = y, showSquares = FALSE, leastSquares = FALSE)

# View(plot_ss)
```

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`.

```
plot_ss(x = x, y = y, 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.

1. 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?

After running it 3 times, I have the following linear equations. The last one is a bit high. The intercept seems to have more effect than I expected.

$$y = 1.33x - 3.283 \qquad (SSE\ or\ RSS = 4054.05)$$

$$y = 1.271x - 2.96 \qquad (SSE\ or\ RSS = 3697.986)$$

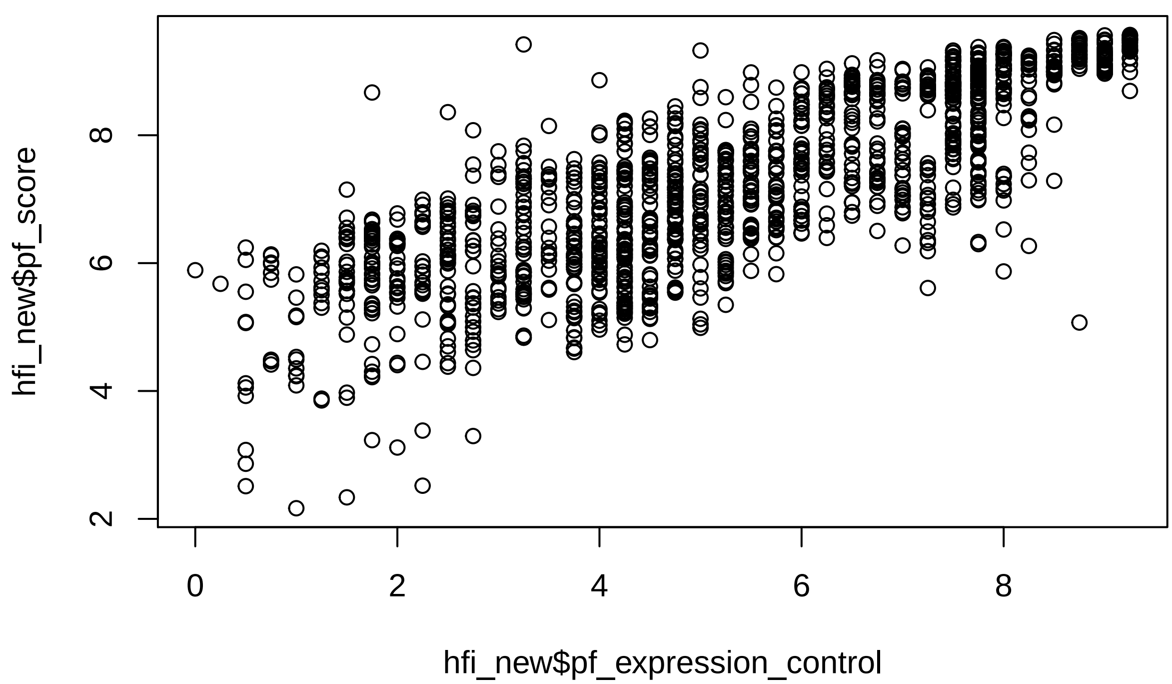
$$y = 1.279x - 2.598 \qquad (SSE\ or\ RSS = 5020.977)$$

$$y = 1.335x - 2.462 \qquad (SSE\ or\ RSS = 7448.953)$$

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).

```
# NOTE: y ~ x
m1 <- lm(pf_score ~ pf_expression_control, data = hfi)

plot(hfi_new$pf_expression_control,hfi_new$pf_score)
```



.

```
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:

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

```
s1<-summary(m1)

# View(s1$residuals)
# View(resid(s1))

# the second row is the responnse variable pf_expression control
coef(s1)
```

```
##                        Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)           4.6170738 0.05745470 80.36025  0.000000e+00
## pf_expression_control 0.4914312 0.01006071 48.84659 8.193309e-303
```

```
s1$coefficients[2,]
```

```
##      Estimate    Std. Error       t value      Pr(>|t|)
##  4.914312e-01  1.006071e-02  4.884659e+01 8.193309e-303
```

```
s1_b<-s1$coefficients[1,1]
s1_m<-s1$coefficients[2,1]
RSS<-deviance(m1)    # RSS

sprintf("y = %.3f * x + %.3f        RSS=%.3f",s1_b, s1_m, RSS )
```

```
## [1] "y = 4.617 * x + 0.491        RSS=952.153"
```

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 pf_expression_control (y) is explained by pf_score(x)

```
R<-cor(hfi_new$pf_score, hfi_new$pf_expression_control)
sprintf('R is %.2f. Since its positive, above .5, it tells me as x increases, y increases',R)
```

```
## [1] "R is 0.80. Since its positive, above .5, it tells me as x increases, y increases"
```

```
sprintf('R Squared is %.2f. R Squared is reffered to as the variablity of y explained by x. ',R^2)
```

```
## [1] "R Squared is 0.63. R Squared is reffered to as the variablity of y explained by x. "
```

R squared obviously is the same metric as correlation R, but its the absolute magnitude of the correlation.

1. Fit a new model that uses `pf_expression_control` to predict `pf_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(pf_expression_control ~ pf_score, data = hfi)
s2<-summary(m2)

s2_b<-s1$coefficients[1,1]
s2_m<-s1$coefficients[2,1]
RSS2<-deviance(m2)    # RSS

sprintf("y = %.3f * x + %.3f        RSS=%.3f",s2_b, s2_m, RSS2 )
```

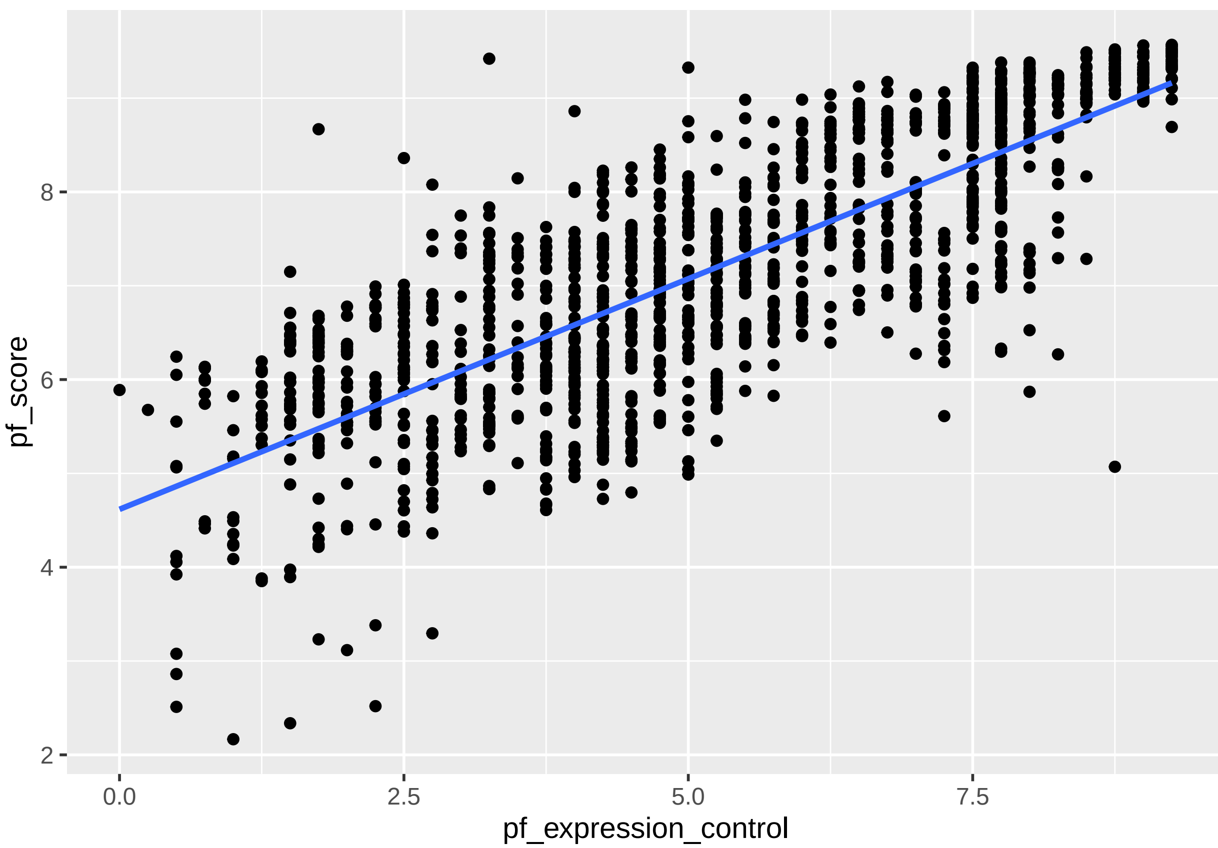
```
## [1] "y = 4.617 * x + 0.491        RSS=2500.531"
```

```
# plot(m2)
```

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.

1. 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?

There are 2 ways to do this

```
new_x<-6.7           # where x=pf_expression_control
new_y<-(new_x*s2_m) + s2_b

sprintf("Using are known slope and intercept, and given x=%.2f then y= %.2f", new_x, new_y)
```

```
## [1] "Using are known slope and intercept, and given x=6.70 then y= 7.91"
```

```
# there is no actual 6.7 in the data but there are many 6.5 and 6.75
#  subset(hfi_new, pf_expression_control<6.77 & pf_expression_control>6.4)[c("pf_expression_control","p
```

```
six.seven <- data.frame( pf_expression_control = 6.7 )
predicted.fit<-predict(m1, newdata = six.seven, interval = "prediction")

sprintf("The predict function used our linear model and predicted pf_control= %.2f", predicted.fit[1,1])
```

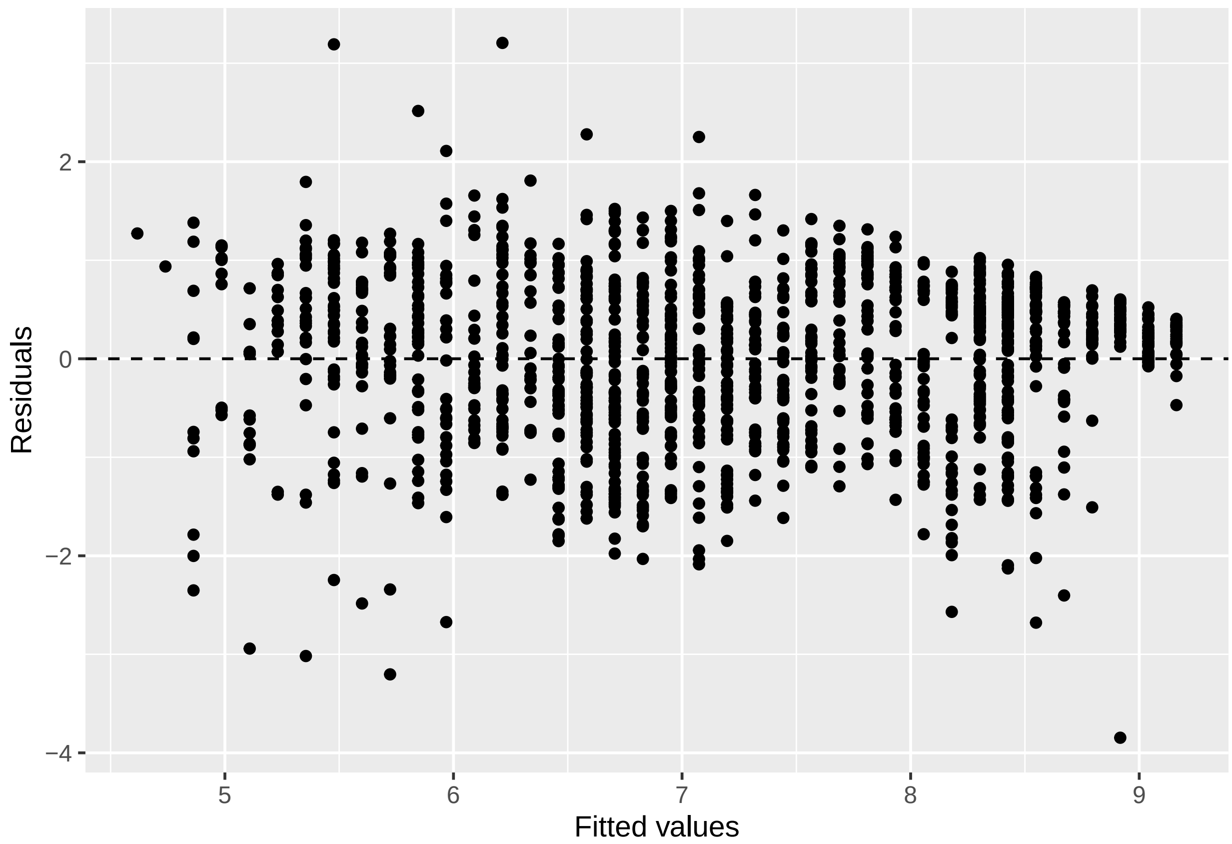
```
## [1] "The predict function used our linear model and predicted pf_control= 7.91"
```

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.

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

The data is somewhat linear.

x and y tend to increase together.

There are some outliers.

\textcolor{blue}{My concern is for a given value of pf_expression_control the pf_score can vary quite a bit.}

```
lm_xy<-data.frame(y = m1$residuals + m1$fitted.values, residuals = m1$residuals, fitted = m1$fitted.valu
hfi_xy<-data.frame(y = hfi_new$pf_score, x = hfi_new$pf_expression_control )
merged_xy<-merge(lm_xy, hfi_xy, by="y")

# View(merged_xy)


show_everyone<-merged_xy %>%
  filter(x>4 & x<5)  %>%
  group_by(x) %>%
  summarize(min_y=min(y), max_y=max(y), mean_y= mean(y))


knitr::kable(show_everyone, caption='Range of pf_score for every pf_expression_control ')
```

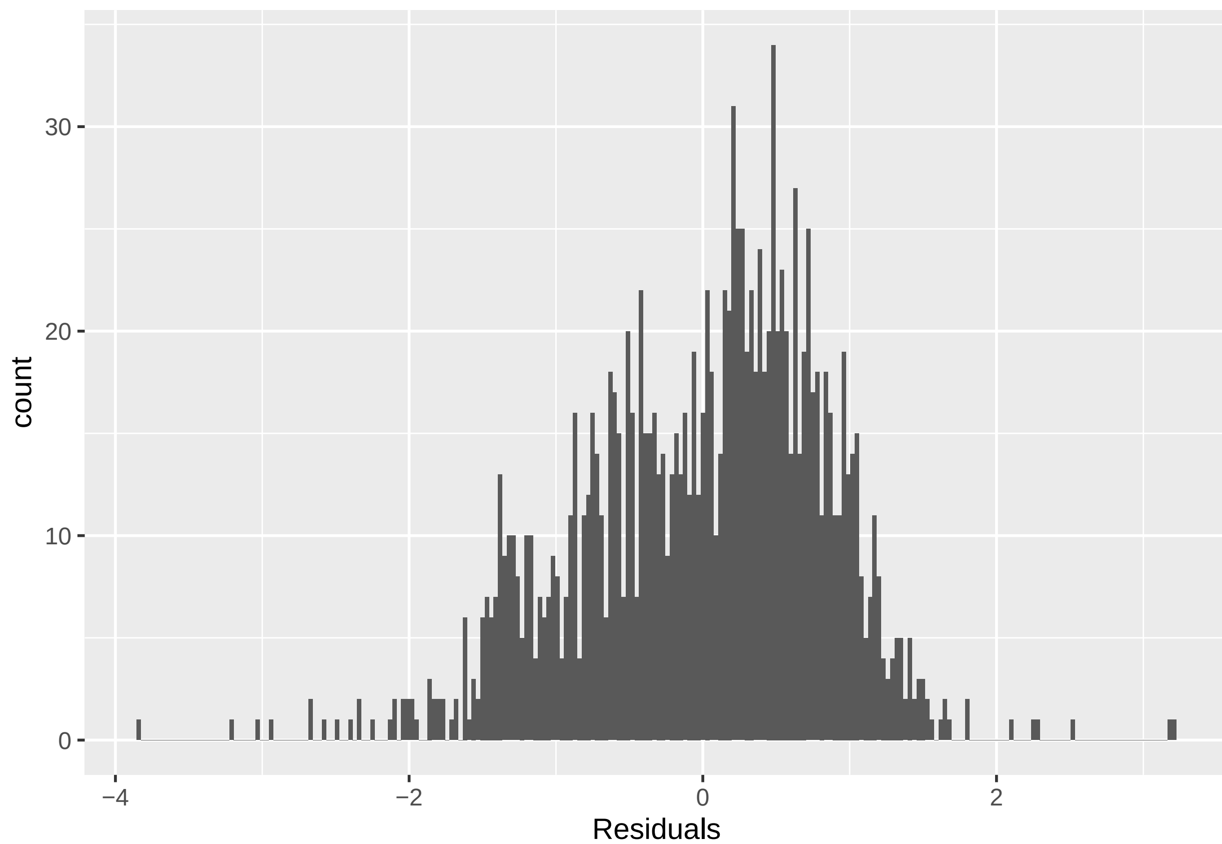
Table 1: Range of pf_score for every pf_expression_control

x	min_y	max_y	mean_y
4.25	4.728113	8.226588	6.444446
4.50	4.796654	8.261328	6.567624
4.75	5.538410	8.451635	6.915519

```
# View(cbind(m1$residuals, m1$fitted.values, m1$residuals + m1$fitted.values))
# View(cbind(hfi_new$pf_score, hfi_new$pf_expression_control))
```

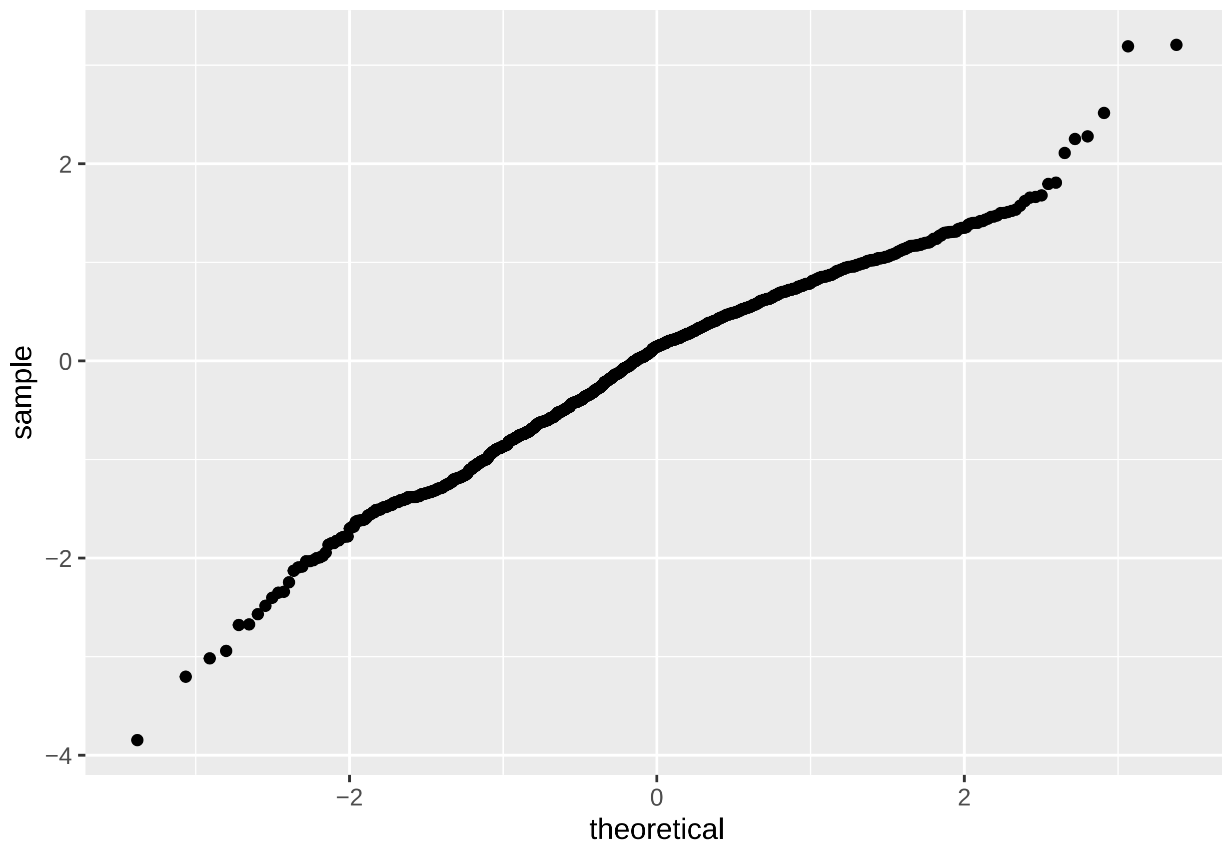
Nearly normal residuals: To check this condition, we can look at a histogram

```
# .resid seems to be the same as m1$residuals
ggplot(data = m1, aes(x = m1$residuals)) +
  geom_histogram(binwidth = .03) +
  xlab("Residuals")
```



or a normal probability plot of the residuals.

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




```
# shapiro.test(m1$residuals)
# qqnorm(m1$residuals)
```

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.

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

Its somewhat normal.

Constant variability:

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

It seems somewhat constant yes.

More Practice

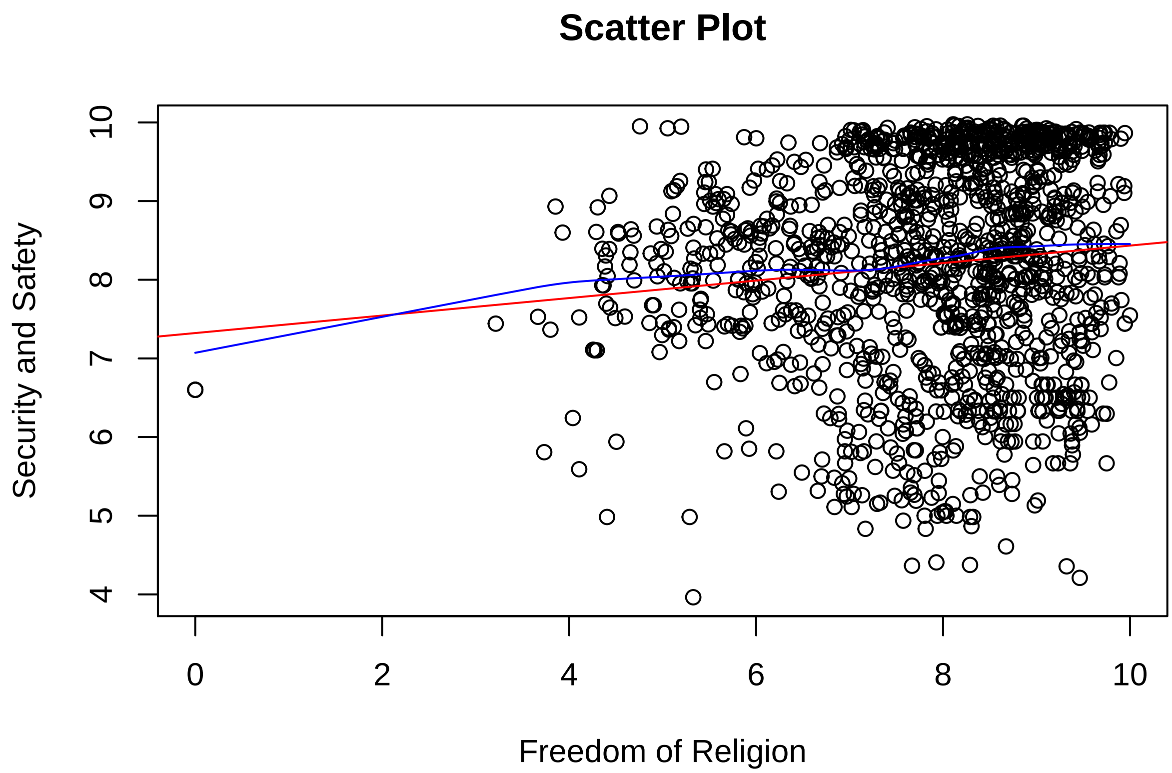
- Choose another freedom variable and a variable you think would strongly correlate with it.. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship?

Lets look at Security and Safety vs Freedome of Religion

```
# pf_ss =Security and Safety and pf_religion = Religious freedom

# View(cbind(hfi$pf_association_assembly, hfi$pf_religion))

 hfi_new<-hfi %>%
  drop_na(pf_ss) %>%
  drop_na(pf_religion)

 x<-hfi_new$pf_religion
 y<-hfi_new$pf_ss




 # y = pf_association_assembly
 # m3 <- lm(hfi_new$pf_association_assembly ~ hfi_new$pf_religion, data = hfi_new)


 R<-cor(x,y)
sprintf('R Squared is %.2f. This is the variablity of y explained by x',R^2)
```

```
## [1] "R Squared is 0.01. This is the variablity of y explained by x"
```

```
# plot_ss(x = hfi_new$pf_religion, y =hfi_new$pf_association_assembly, showSquares = FALSE, leastSquare

 plot(x, y, main = "Scatter Plot",
     xlab ="Freedom of Religion",
     ylab="Security and Safety" )
abline(lm(y~x), col="red") # regression line (y~x)
lines(lowess(x,y), col="blue")
```



Not a strong relationship at all.

- 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?

Just by looking, it seems like there is no correlation at all. The R squared value before was .64 and this one is only .1 Im not sure I expected a correlation although I think of some countries with little freedom have a lot of civil strife.

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

I find it difficult to assume that any of the metrics are clearly explanatory or response. It seems to me any 2 attributes may correlate without implying causality. Anyway originally I thought the pf score correlation was weak, but as I continued to plot I changed my mind. The box plot was a pretty useful visualation to establish the xy relationship. So was qqline.

```
par(mfrow=c(1,2))

boxplot(hfi$pf_score ~ hfi$pf_expression_control, data=hfi, main = "Box Plot",
        xlab="Political pressures and controls on media content",
        ylab="Personal Freedom Score")
```

```r
# qqnorm plots one vector agains its theoretical normalized quantiles
qqnorm(hfi$pf_score)
qqline(hfi$pf_score, col="red")
qqline(hfi$pf_expression_control, col="blue")
```