Chapter 8: Introduction to linear regression

Alexis Mekueko

10/24/2020

R Packages

```
library(tidyverse) #loading all library needed for this assignment
library(openintro)
#head(fastfood)
#library(readxl)
library(data.table)
#library(DT)
library(knitLatex)
library(knitr)
library(markdown)
library(rmarkdown)

#library(readr)
#library(plyr)
#library(dplyr)
library(stringr)
#library(XML)
#library(RCurl)
#library(jsonlite)
#library(httr)

#library(maps)
#library(dice)
# #library(VennDiagram)
# #library(help = "dice")
#ibrary(DBI)
#library(dbplyr)

# library(rstudioapi)
# library(RJDBC)
# library(odbc)
# library(RSQLite)
# #library(rvest)

#library(readtext)
#library(ggpubr)
#library(fitdistrplus)

library(plyr)
```

```
library(pdftools)
library(plotrix)
library(gplots)
library(tibble)
#library(moments)
#library(qualityTools)
#library(normalp)
#library(utils)
#library(MASS)
#library(qqplotr)
#library(stats)
library(statsr)
```

```
## Warning: package 'statsr' was built under R version 4.0.3
```

```
#(DATA606)
```

Github Link: https://github.com/asmozo24/DATA606_Lab8

Web link: https://rpubs.com/amekueko/681421

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.

##The data The data we're working with is in the openintro package and it's called hfi, short for Human Freedom Index.

#Exercice 1

What are the dimensions of the dataset? Answer: 1458 rows, 123 columns, data collected from 2008 to 2016,

```
# loading the dataset from the openIntro/data606 library
data(hfi)

#view hfi dataset details
glimpse(hfi)
```

```
## Rows: 1,458
## Columns: 123
## $ year                          <dbl> 2016, 2016, 2016, 2016, 2016, 20...
## $ ISO_code                      <chr> "ALB", "DZA", "AGO", "ARG", "ARM...
## $ countries                     <chr> "Albania", "Algeria", "Angola", ...
## $ region                        <chr> "Eastern Europe", "Middle East &...
## $ pf_rol_procedural             <dbl> 6.661503, NA, NA, 7.098483, NA, ...
```

```
## $ pf_rol_civil                       <dbl> 4.547244, NA, NA, 5.791960, NA, ...
## $ pf_rol_criminal                    <dbl> 4.666508, NA, NA, 4.343930, NA, ...
## $ pf_rol                             <dbl> 5.291752, 3.819566, 3.451814, 5....
## $ pf_ss_homicide                     <dbl> 8.920429, 9.456254, 8.060260, 7....
## $ pf_ss_disappearances_disap         <dbl> 10, 10, 5, 10, 10, 10, 10, 10, 1...
## $ pf_ss_disappearances_violent       <dbl> 10.000000, 9.294030, 10.000000, ...
## $ pf_ss_disappearances_organized     <dbl> 10.0, 5.0, 7.5, 7.5, 7.5, 10.0, ...
## $ pf_ss_disappearances_fatalities    <dbl> 10.000000, 9.926119, 10.000000, ...
## $ pf_ss_disappearances_injuries      <dbl> 10.000000, 9.990149, 10.000000, ...
## $ pf_ss_disappearances               <dbl> 10.000000, 8.842060, 8.500000, 9...
## $ pf_ss_women_fgm                    <dbl> 10.0, 10.0, 10.0, 10.0, 10.0, 10...
## $ pf_ss_women_missing                <dbl> 7.5, 7.5, 10.0, 10.0, 5.0, 10.0,...
## $ pf_ss_women_inheritance_widows     <dbl> 5, 0, 5, 10, 10, 10, 10, 5, NA, ...
## $ pf_ss_women_inheritance_daughters  <dbl> 5, 0, 5, 10, 10, 10, 10, 10, NA,...
## $ pf_ss_women_inheritance            <dbl> 5.0, 0.0, 5.0, 10.0, 10.0, 10.0,...
## $ pf_ss_women                        <dbl> 7.500000, 5.833333, 8.333333, 10...
## $ pf_ss                              <dbl> 8.806810, 8.043882, 8.297865, 9....
## $ pf_movement_domestic               <dbl> 5, 5, 0, 10, 5, 10, 10, 5, 10, 1...
## $ pf_movement_foreign                <dbl> 10, 5, 5, 10, 5, 10, 10, 5, 10, ...
## $ pf_movement_women                  <dbl> 5, 5, 10, 10, 10, 10, 10, 5, NA,...
## $ pf_movement                        <dbl> 6.666667, 5.000000, 5.000000, 10...
## $ pf_religion_estop_establish        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_religion_estop_operate          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_religion_estop                  <dbl> 10.0, 5.0, 10.0, 7.5, 5.0, 10.0,...
## $ pf_religion_harassment             <dbl> 9.566667, 6.873333, 8.904444, 9....
## $ pf_religion_restrictions           <dbl> 8.011111, 2.961111, 7.455556, 6....
## $ pf_religion                        <dbl> 9.192593, 4.944815, 8.786667, 7....
## $ pf_association_association         <dbl> 10.0, 5.0, 2.5, 7.5, 7.5, 10.0, ...
## $ pf_association_assembly            <dbl> 10.0, 5.0, 2.5, 10.0, 7.5, 10.0,...
## $ pf_association_political_establish <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_political_operate   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_political           <dbl> 10.0, 5.0, 2.5, 5.0, 5.0, 10.0, ...
## $ pf_association_prof_establish      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_prof_operate        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_prof                <dbl> 10.0, 5.0, 5.0, 7.5, 5.0, 10.0, ...
## $ pf_association_sport_establish     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_sport_operate       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ pf_association_sport               <dbl> 10.0, 5.0, 7.5, 7.5, 7.5, 10.0, ...
## $ pf_association                     <dbl> 10.0, 5.0, 4.0, 7.5, 6.5, 10.0, ...
## $ pf_expression_killed               <dbl> 10.000000, 10.000000, 10.000000,...
## $ pf_expression_jailed               <dbl> 10.000000, 10.000000, 10.000000,...
## $ pf_expression_influence            <dbl> 5.0000000, 2.6666667, 2.6666667,...
## $ pf_expression_control              <dbl> 5.25, 4.00, 2.50, 5.50, 4.25, 7....
## $ pf_expression_cable                <dbl> 10.0, 10.0, 7.5, 10.0, 7.5, 10.0...
## $ pf_expression_newspapers           <dbl> 10.0, 7.5, 5.0, 10.0, 7.5, 10.0,...
## $ pf_expression_internet             <dbl> 10.0, 7.5, 7.5, 10.0, 7.5, 10.0,...
## $ pf_expression                      <dbl> 8.607143, 7.380952, 6.452381, 8....
## $ pf_identity_legal                  <dbl> 0, NA, 10, 10, 7, 7, 10, 0, NA, ...
## $ pf_identity_parental_marriage      <dbl> 10, 0, 10, 10, 10, 10, 10, 10, 1...
## $ pf_identity_parental_divorce       <dbl> 10, 5, 10, 10, 10, 10, 10, 10, 1...
## $ pf_identity_parental               <dbl> 10.0, 2.5, 10.0, 10.0, 10.0, 10....
## $ pf_identity_sex_male               <dbl> 10, 0, 0, 10, 10, 10, 10, 10, 10...
## $ pf_identity_sex_female             <dbl> 10, 0, 0, 10, 10, 10, 10, 10, 10...
## $ pf_identity_sex                    <dbl> 10, 0, 0, 10, 10, 10, 10, 10, 10...
```

```
## $ pf_identity_divorce                <dbl> 5, 0, 10, 10, 5, 10, 10, 5, NA, ...
## $ pf_identity                        <dbl> 6.2500000, 0.8333333, 7.5000000,...
## $ pf_score                           <dbl> 7.596281, 5.281772, 6.111324, 8....
## $ pf_rank                            <dbl> 57, 147, 117, 42, 84, 11, 8, 131...
## $ ef_government_consumption          <dbl> 8.232353, 2.150000, 7.600000, 5....
## $ ef_government_transfers            <dbl> 7.509902, 7.817129, 8.886739, 6....
## $ ef_government_enterprises          <dbl> 8, 0, 0, 6, 8, 10, 10, 0, 7, 10,...
## $ ef_government_tax_income           <dbl> 9, 7, 10, 7, 5, 5, 4, 9, 10, 10,...
## $ ef_government_tax_payroll          <dbl> 7, 2, 9, 1, 5, 5, 3, 4, 10, 10, ...
## $ ef_government_tax                  <dbl> 8.0, 4.5, 9.5, 4.0, 5.0, 5.0, 3....
## $ ef_government                      <dbl> 7.935564, 3.616782, 6.496685, 5....
## $ ef_legal_judicial                  <dbl> 2.6682218, 4.1867042, 1.8431292,...
## $ ef_legal_courts                    <dbl> 3.145462, 4.327113, 1.974566, 2....
## $ ef_legal_protection                <dbl> 4.512228, 4.689952, 2.512364, 4....
## $ ef_legal_military                  <dbl> 8.333333, 4.166667, 3.333333, 7....
## $ ef_legal_integrity                 <dbl> 4.166667, 5.000000, 4.166667, 3....
## $ ef_legal_enforcement               <dbl> 4.3874441, 4.5075380, 2.3022004,...
## $ ef_legal_restrictions              <dbl> 6.485287, 6.626692, 5.455882, 6....
## $ ef_legal_police                    <dbl> 6.933500, 6.136845, 3.016104, 3....
## $ ef_legal_crime                     <dbl> 6.215401, 6.737383, 4.291197, 4....
## $ ef_legal_gender                    <dbl> 0.9487179, 0.8205128, 0.8461538,...
## $ ef_legal                           <dbl> 5.071814, 4.690743, 2.963635, 3....
## $ ef_money_growth                    <dbl> 8.986454, 6.955962, 9.385679, 5....
## $ ef_money_sd                        <dbl> 9.484575, 8.339152, 4.986742, 5....
## $ ef_money_inflation                 <dbl> 9.743600, 8.720460, 3.054000, 2....
## $ ef_money_currency                  <dbl> 10, 5, 5, 10, 10, 10, 10, 5, 0, ...
## $ ef_money                           <dbl> 9.553657, 7.253894, 5.606605, 5....
## $ ef_trade_tariffs_revenue           <dbl> 9.626667, 8.480000, 8.993333, 6....
## $ ef_trade_tariffs_mean              <dbl> 9.24, 6.22, 7.72, 7.26, 8.76, 9....
## $ ef_trade_tariffs_sd                <dbl> 8.0240, 5.9176, 4.2544, 5.9448, ...
## $ ef_trade_tariffs                   <dbl> 8.963556, 6.872533, 6.989244, 6....
## $ ef_trade_regulatory_nontariff      <dbl> 5.574481, 4.962589, 3.132738, 4....
## $ ef_trade_regulatory_compliance     <dbl> 9.4053278, 0.0000000, 0.9171598,...
## $ ef_trade_regulatory                <dbl> 7.489905, 2.481294, 2.024949, 4....
## $ ef_trade_black                     <dbl> 10.00000, 5.56391, 10.00000, 0.0...
## $ ef_trade_movement_foreign          <dbl> 6.306106, 3.664829, 2.946919, 5....
## $ ef_trade_movement_capital          <dbl> 4.6153846, 0.0000000, 3.0769231,...
## $ ef_trade_movement_visit            <dbl> 8.2969231, 1.1062564, 0.1106256,...
## $ ef_trade_movement                  <dbl> 6.406138, 1.590362, 2.044823, 4....
## $ ef_trade                           <dbl> 8.214900, 4.127025, 5.264754, 3....
## $ ef_regulation_credit_ownership     <dbl> 5, 0, 8, 5, 10, 10, 8, 5, 10, 10...
## $ ef_regulation_credit_private       <dbl> 7.295687, 5.301526, 9.194715, 4....
## $ ef_regulation_credit_interest      <dbl> 9, 10, 4, 7, 10, 10, 10, 9, 10, ...
## $ ef_regulation_credit               <dbl> 7.098562, 5.100509, 7.064905, 5....
## $ ef_regulation_labor_minwage        <dbl> 5.566667, 5.566667, 8.900000, 2....
## $ ef_regulation_labor_firing         <dbl> 5.396399, 3.896912, 2.656198, 2....
## $ ef_regulation_labor_bargain        <dbl> 6.234861, 5.958321, 5.172987, 3....
## $ ef_regulation_labor_hours          <dbl> 8, 6, 4, 10, 10, 10, 6, 6, 8, 8,...
## $ ef_regulation_labor_dismissal      <dbl> 6.299741, 7.755176, 6.632764, 2....
## $ ef_regulation_labor_conscription   <dbl> 10, 1, 0, 10, 0, 10, 3, 1, 10, 1...
## $ ef_regulation_labor                <dbl> 6.916278, 5.029513, 4.560325, 5....
## $ ef_regulation_business_adm         <dbl> 6.072172, 3.722341, 2.758428, 2....
## $ ef_regulation_business_bureaucracy <dbl> 6.000000, 1.777778, 1.333333, 6....
## $ ef_regulation_business_start       <dbl> 9.713864, 9.243070, 8.664627, 9....
```

```
## $ ef_regulation_business_bribes      <dbl> 4.050196, 3.765515, 1.945540, 3....
## $ ef_regulation_business_licensing   <dbl> 7.324582, 8.523503, 8.096776, 5....
## $ ef_regulation_business_compliance  <dbl> 7.074366, 7.029528, 6.782923, 6....
## $ ef_regulation_business             <dbl> 6.705863, 5.676956, 4.930271, 5....
## $ ef_regulation                      <dbl> 6.906901, 5.268992, 5.518500, 5....
## $ ef_score                           <dbl> 7.54, 4.99, 5.17, 4.84, 7.57, 7....
## $ ef_rank                            <dbl> 34, 159, 155, 160, 29, 10, 27, 1...
## $ hf_score                           <dbl> 7.568140, 5.135886, 5.640662, 6....
## $ hf_rank                            <dbl> 48, 155, 142, 107, 57, 4, 16, 13...
## $ hf_quartile                        <dbl> 2, 4, 4, 3, 2, 1, 1, 4, 2, 2, 4,...
```

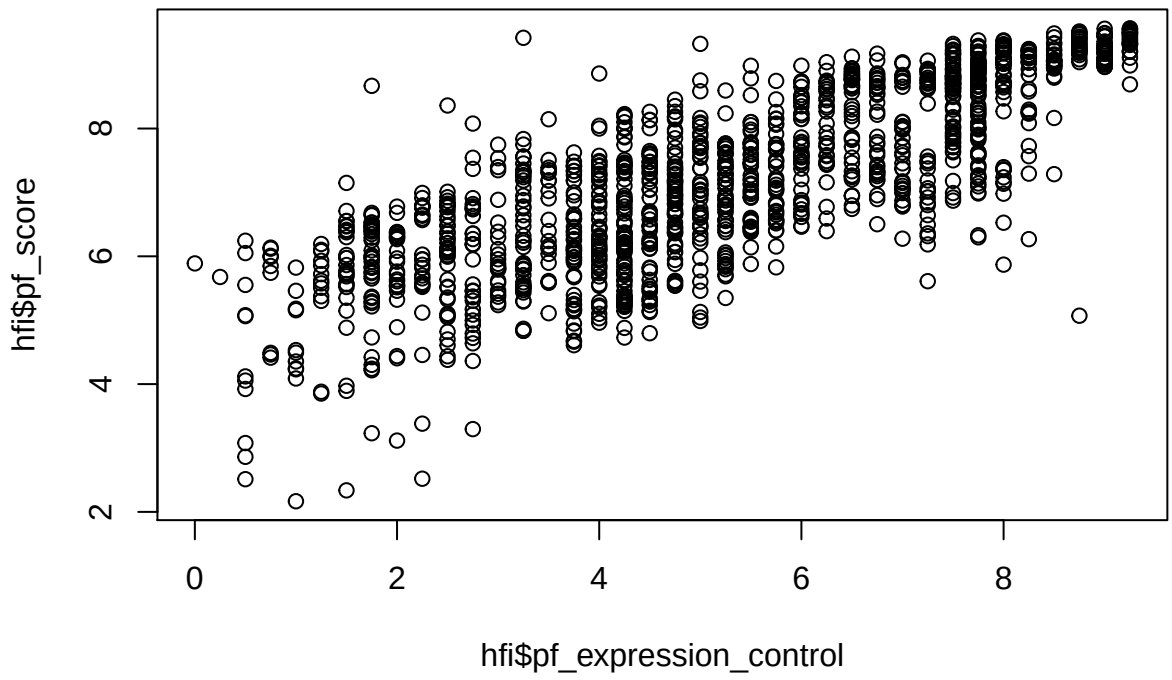
```r
#view table in Rstudio
view(hfi)
```

#Exercice 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?

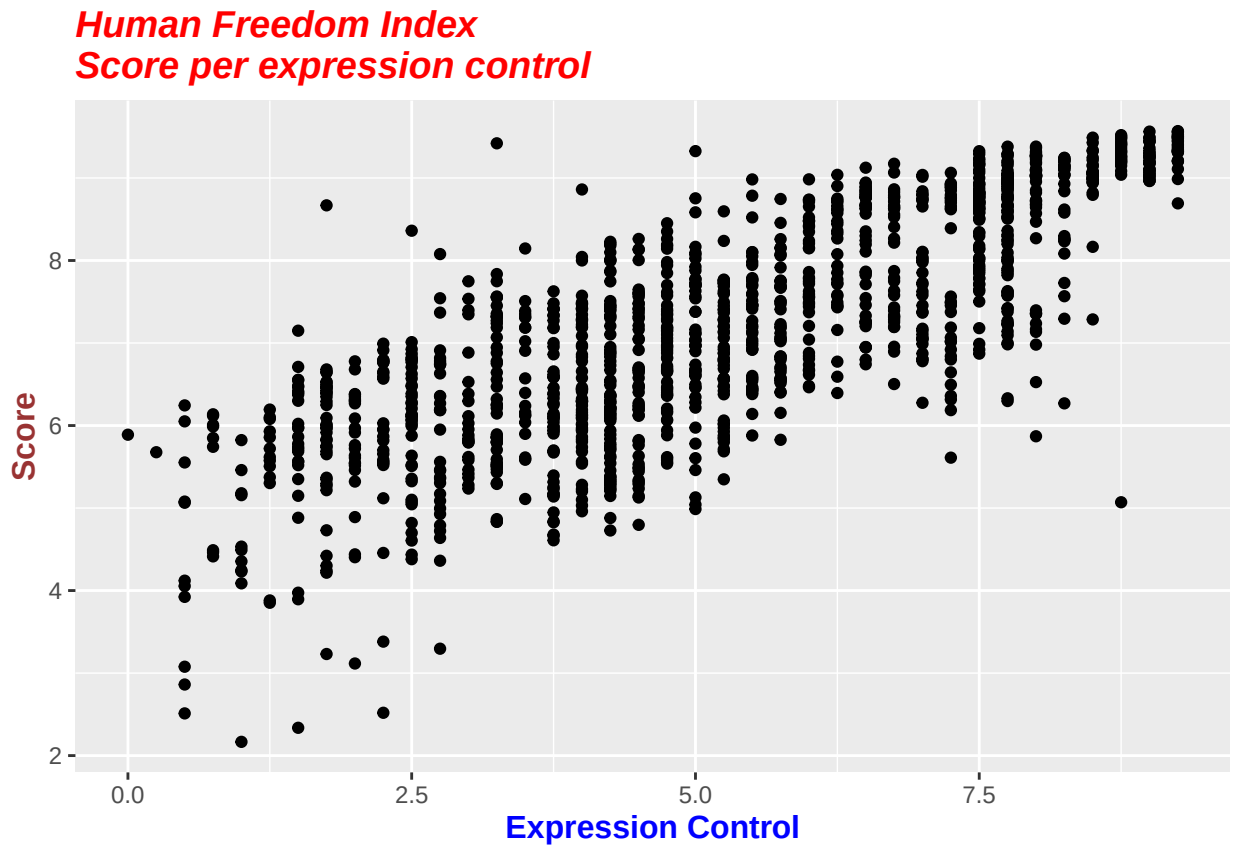
Answer: pf_score is numerical and one of the other numerical variables will make a 02 variables plot: I can use scatterplot of y against x [plot(x,y)-cartesian or plot(x~y)-formula] or box-and-whisker plot of y at levels of factor [plot(factor, y)] or heights from a vector of y values ([barplot(y)]). The relationship does show some linearity trend, but the data looks scatters. No, I would not be comfortable because on a large sample data, it is difficult to determine the type of plot. It is after carefully examining the variable with their observation that we can determine the type of plot which can fit.

```r
#hfi <- na.omit(hfi) # delete/remove the missings data because it is an imcomplete observation
# somehow, the removing NA delete the entire dataset.

#plot to display the relationship between the personal freedom score, pf_score, and one of the other nu
plot(hfi$pf_score ~ hfi$pf_expression_control)
```




```
#scattered plot x = pf_expression_control, y = pf_score
ggplot(hfi, aes(x=pf_expression_control, y=pf_score)) + geom_point() +
  ggtitle("Human Freedom Index \nScore per expression control") +
  xlab("Expression Control") + ylab("Score") +

# Change the color, the size and the face of
# the main title, x and y axis labels
 theme(
plot.title = element_text(color="red", size=14, face="bold.italic"),
axis.title.x = element_text(color="blue", size=12, face="bold"),
axis.title.y = element_text(color="#993333", size=12, face="bold"))
```

```
## Warning: Removed 80 rows containing missing values (geom_point).
```




```
#ggplot(aes(x=pf_expression_control,y=pf_score,  main =" Human Freedom Index \nScore per expression con
#  geom_point()

#If the relationship looks linear, we can quantify the strength of the relationship with the correlatio
hfi %>%
  summarise(cor(pf_expression_control, pf_score, use = "complete.obs"))
```

```
##   cor(pf_expression_control, pf_score, use = "complete.obs")
## 1                                                  0.7963894
```

```
#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.

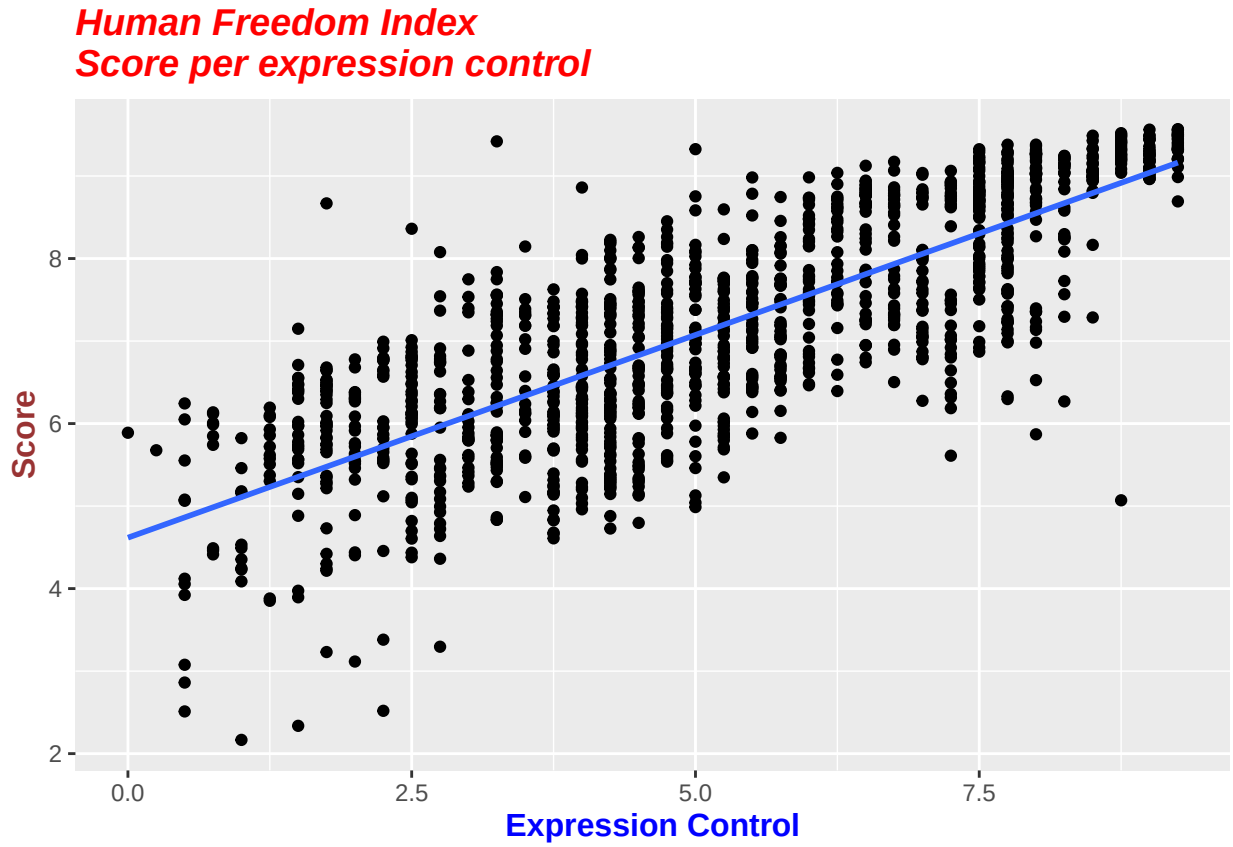
#Exercice 3 Looking at your plot from the previous exercise, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations. Answer: Based on the corelation coefficient ($0.7963894 \sim 0.8$) , there is a strong corelation for a linear relation between pf_expression_control and pf_score. there are many points off which affect the linearity of the plot...

```
#actually let's see a fit line
ggplot(hfi, aes(x=pf_expression_control, y=pf_score)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Human Freedom Index \nScore per expression control") +
  xlab("Expression Control") + ylab("Score") +

# Change the color, the size and the face of
# the main title, x and y axis labels
 theme(
plot.title = element_text(color="red", size=14, face="bold.italic"),
axis.title.x = element_text(color="blue", size=12, face="bold"),
axis.title.y = element_text(color="#993333", size=12, face="bold"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 80 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 80 rows containing missing values (geom_point).
```




```
#ggplot(hfi, aes(pf_expression_control, pf_score)) + geom_point()+
#  geom_smooth(method = "lm", se = FALSE) # + facet_wrap(~cyl)

theme <- theme(
plot.title = element_text(color="red", size=14, face="bold.italic"),
```

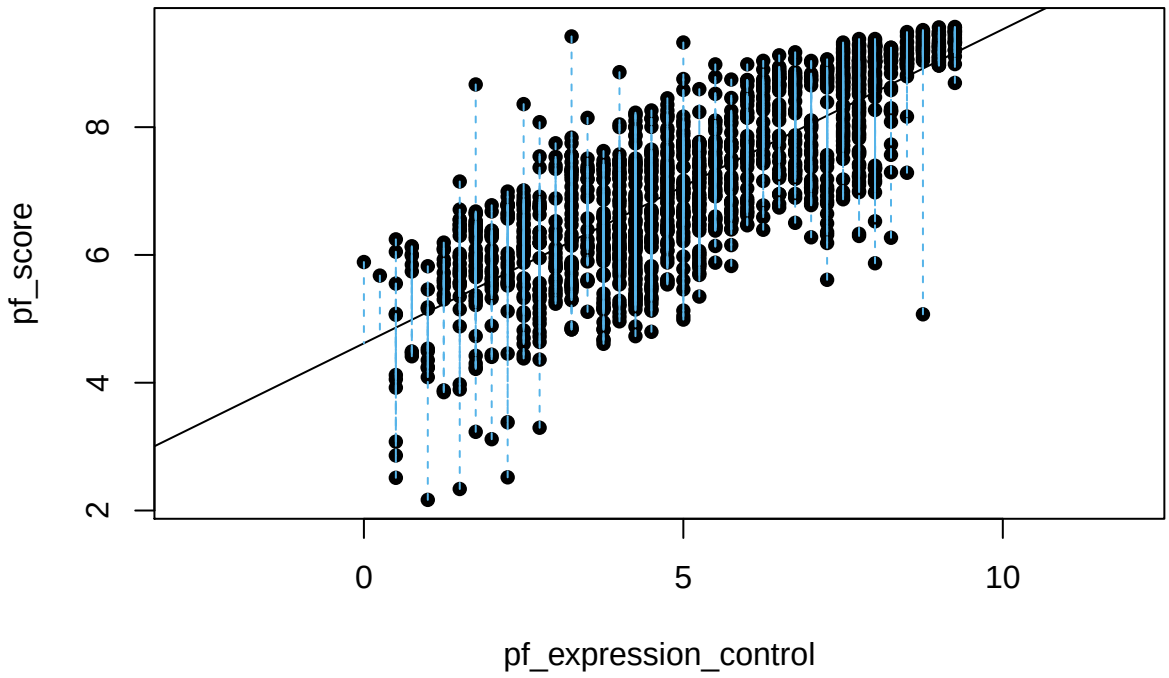
```
axis.title.x = element_text(color="blue", size=12, face="bold"),
axis.title.y = element_text(color="#993333", size=12, face="bold"))
```

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 isn't working, wondering if we should just remove rows with NA
# ok let/s remove the NA, I tried #hfi <- na.omit(hfi) # delete/remove the missings data because it is
# somehow, the removing NA delete the entire dataset.
# this time , I will try removing NA in selected variable

hfi1 <- select(hfi, pf_expression_control, pf_score )
#checking how many values are missing in hfi
sum(is.na(hfi1))
```

```
## [1] 160
```

```
hfi2 <- na.omit(hfi1)

# trying again to plot_ss
plot_ss(x = pf_expression_control, y = pf_score, data = hfi2)
```



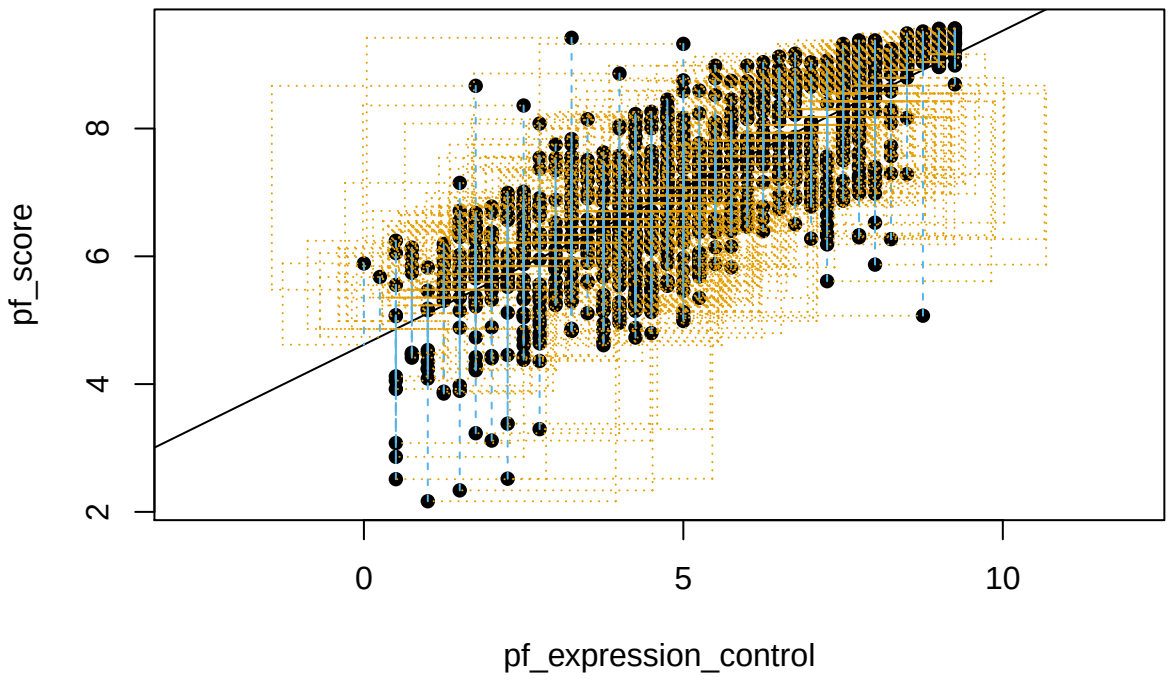
```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##      4.6171       0.4914
##
## Sum of Squares:  952.153
```

```
# # trying to create a title function but somehow I cannot use it on plot_ss
# title1 <- title(main = "Human Freedom Index \nScore per expression control",
#       xlab = "Expression Control", ylab = "Score",
#       cex.main = 2,   font.main= 4, col.main= "red",
#       cex.sub = 0.75, font.sub = 3, col.sub = "green",
#       col.lab ="darkblue"
#       )

#plot(hfi$pf_score ~ hfi$pf_expression_control)
```

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: e1 = y_i - 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 = hfi$pf_expression_control, y = hfi$pf_score, showSquares = TRUE)
plot_ss(x = pf_expression_control, y = pf_score, data = hfi2, showSquares = TRUE)
```




```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##      4.6171       0.4914
##
## Sum of Squares:  952.153
```

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.

#Exercice 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? Answer: Sum of Squares: 952.153, I don't think it changes as I rerun.

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).

```r
m1 <- lm(pf_score ~ pf_expression_control, data = hfi2)
```

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.

```r
# display statistical details in m1
summary(m1)
```

```
##
## Call:
## lm(formula = pf_score ~ pf_expression_control, data = hfi2)
##
## 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
## 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 at_bats. 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, b or y-intercept = 4.61707, slope or a-coeficient = 0.461707

One last piece of information we will discuss from the summary output is the Multiple R-squared, or more simply, R2 . The R2 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.

Exercice 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?

Answer: $\hat{y} = 5.153687 + 0.349862 \times$ pf_expression_control slope: the human freedom score increases by 0.35 amount of political pressure on media content intercept: when they started this study on political pressure on media content, the human freedom score has a reference of 5.153

```r
# Just replicate few lines from above, this time with new variable hf_score
hfi3 <- select(hfi, pf_expression_control, hf_score )
#checking how many values are missing in hfi
sum(is.na(hfi3))
```

```
## [1] 160
```

```r
hfi3 <- na.omit(hfi3)

m2 <- lm(hf_score ~ pf_expression_control, data = hfi3)
summary(m2)
```

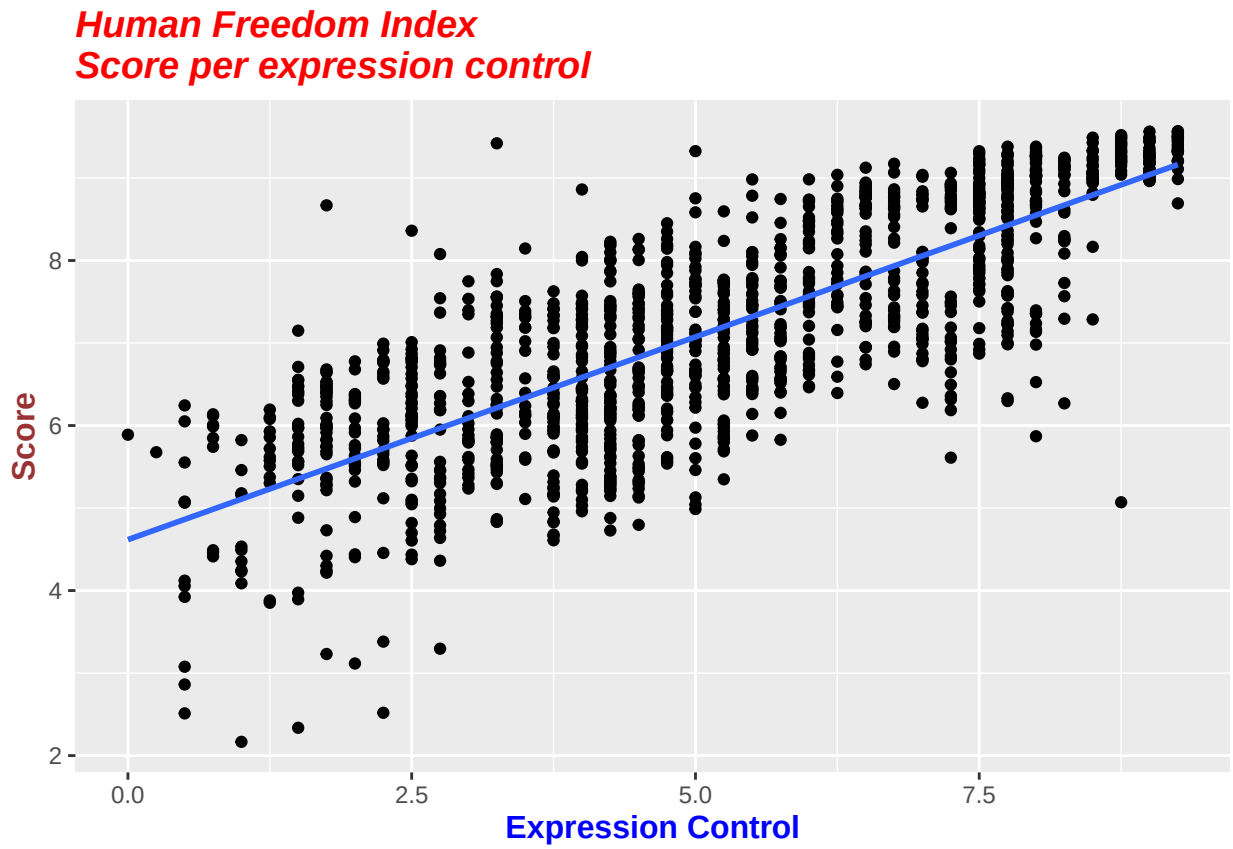
```
##
## Call:
## lm(formula = hf_score ~ pf_expression_control, data = hfi3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6198 -0.4908  0.1031  0.4703  2.2933
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.153687   0.046070  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
## Multiple R-squared:  0.5775, Adjusted R-squared:  0.5772
## F-statistic:  1881 on 1 and 1376 DF,  p-value: < 2.2e-16
```

Prediction and prediction errors

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

```r
# I did this already...was just looking at the fitting line
ggplot(data = hfi2, aes(x = pf_expression_control, y = pf_score))   +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE) +
   ggtitle("Human Freedom Index \nScore per expression control") +
  xlab("Expression Control") + ylab("Score") +
  theme
```

```
## `geom_smooth()` using formula 'y ~ x'
```



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.

Exercice 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?

Answer: If someone saw the least squares regression line and not the actual data, to predict a country's personal freedom school for one with a 6.7 rating for pf_expression_control I would plot 6.7 and find the its correspondant on the least squares regression line which gives about ~7.25. This can be more acurated with redefiined scale on axis or axis increment value. However, I am actually wrong, when I filtered the value pf_expression_control = 6.7 from view(hfi2), there is no corresponding value for pf_score. we could use filter as showed in the code section below... observed (I approximated by using mean value) pf_score = 8.006315....Now what the equation would have given.

Residual = Observed value - Fitted value. Linear residual = 8.006315-7.909651 = 0.09666408... very close or undersestimate

```
# filter value pf_score = 6.7
hfi6.7 <- hfi2 %>%
  filter(pf_expression_control > 6.7 & pf_expression_control <6.8)
# there is more than one corresponding values for pf_score for the same pf_expression_control. let's us
mean(hfi6.7$pf_score)
```

```
## [1] 8.006315
```

```
y = 4.61707 + 0.49143*6.7
y
```

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

```
residual = mean(hfi6.7$pf_score) - y
residual
```

```
## [1] 0.09666408
```

```
# ggplot(data = hfi2, aes(x = pf_expression_control, y = pf_score), xaxt="n")+
#   geom_point() +
#   stat_smooth(method = "lm", se = FALSE)
#   # axis(side=1, at=seq(0, 10, by =1), labels = seq(0, 23, 1)) #changing scale on the axis...does not
#   # axis(side=2, at=seq(0, 10, by= 0.1))
# box() +
#    ggtitle("Human Freedom Index \nScore per expression control") +
#   xlab("Expression Control") + ylab("Score") +
#   theme
```

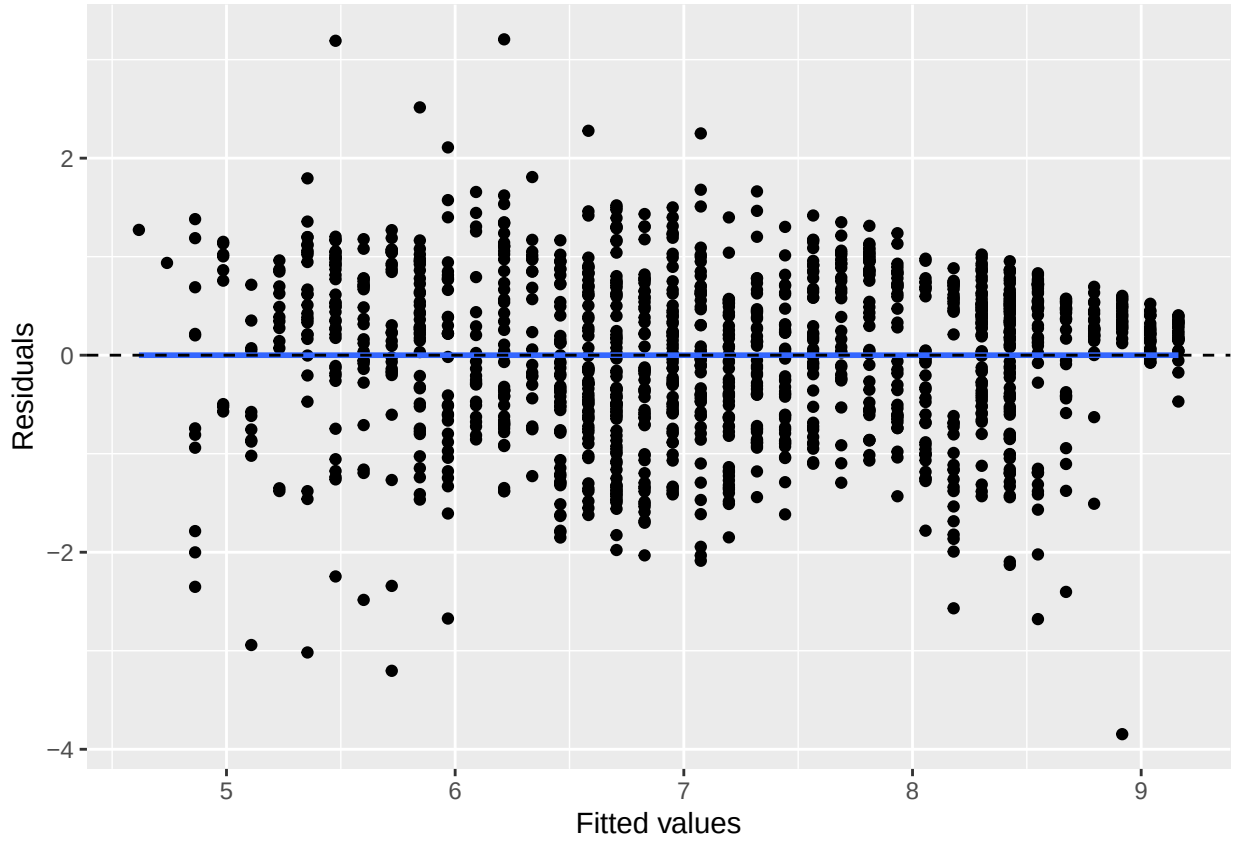
Model diagnostics

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_smooth(method = "lm", se = FALSE)+ # added for the fitted line
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Notice here that m1 can also serve as a data set because stored within it are the fitted values (yˆ) 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.

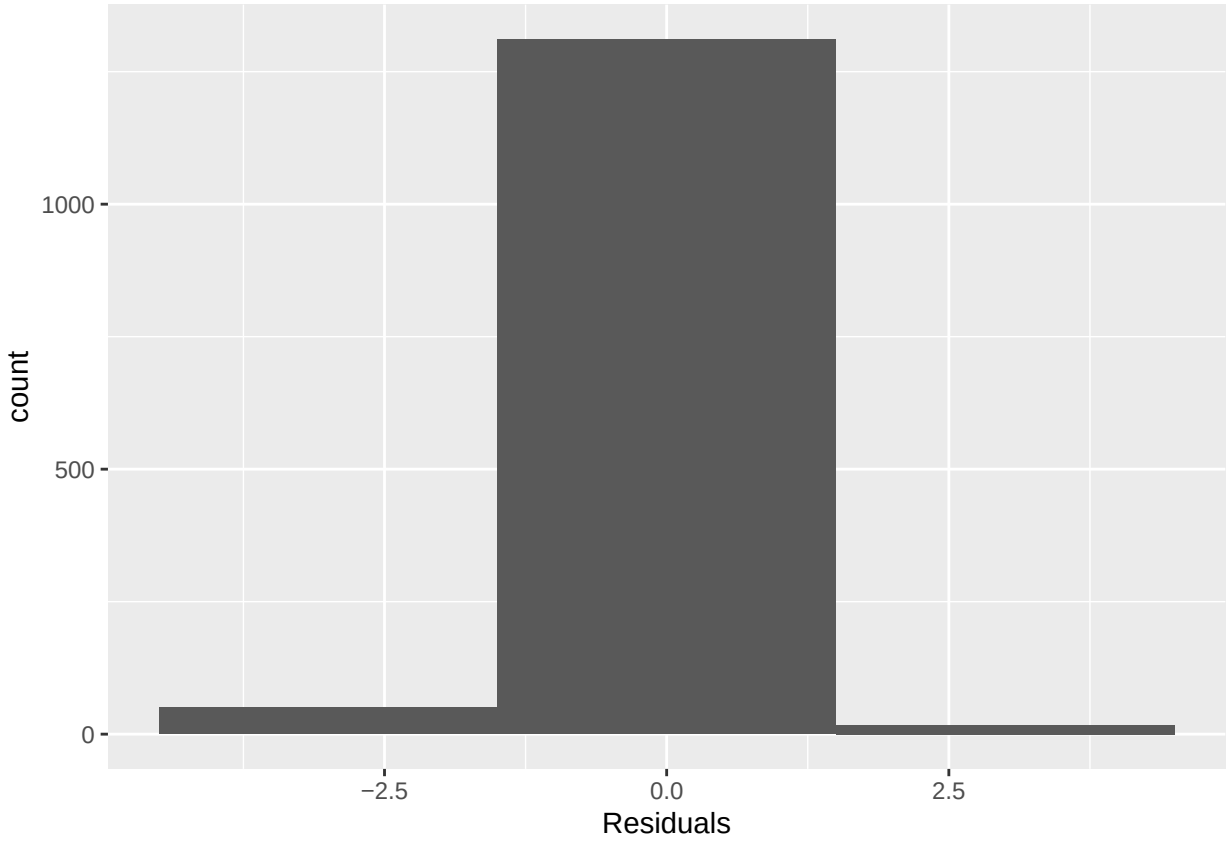
Exercice 7

Is there any apparent pattern in the residuals plot? What does this indicate about the linearity of the relationship between the two variables? Answer: based on the residuals plot , there is not pattern, and the variance of the error predicted value is constant across the x-axis which indicates there is likely a linear relationship between the two variables.

Nearly normal residuals:

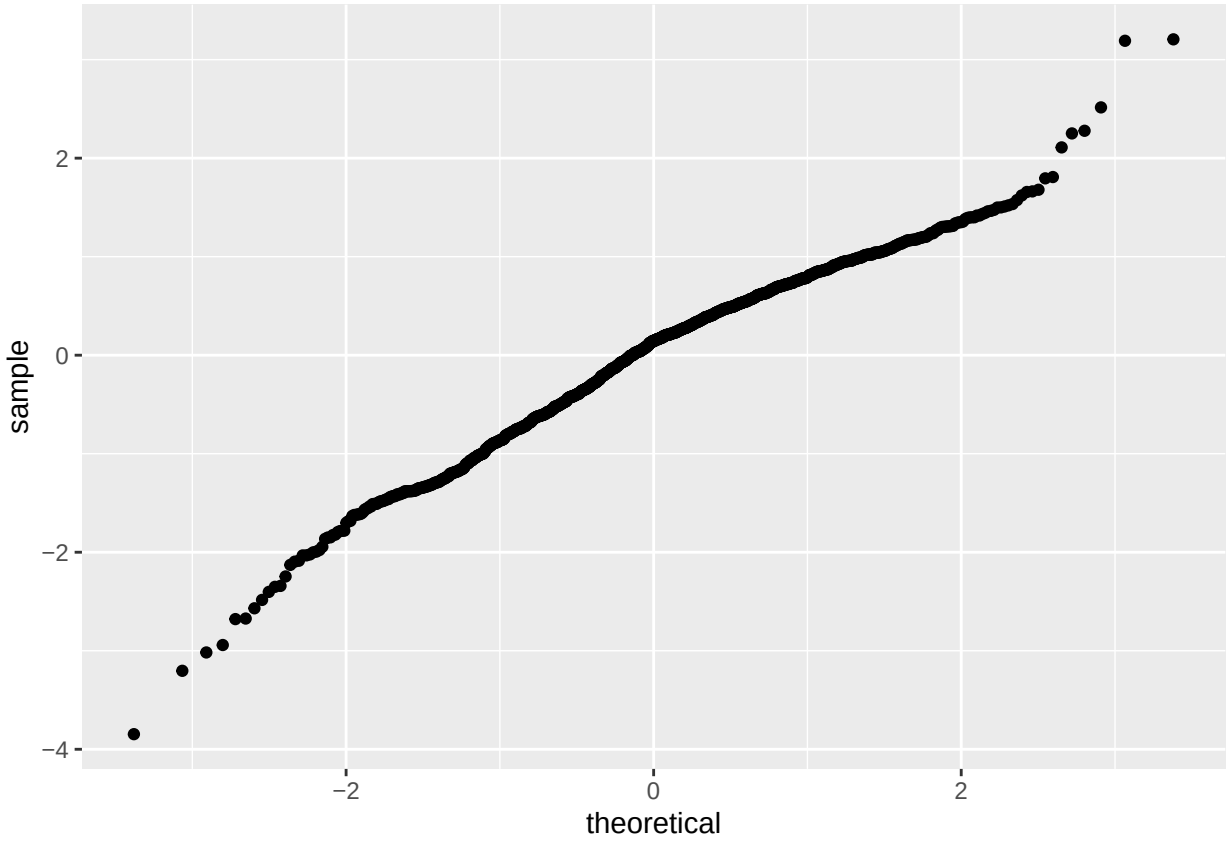
To check this condition, we can look at a histogram

```
ggplot(data = m1, aes(x = .resid)) +
  geom_histogram(binwidth = 3) +  # residual of 25 is too high...max residual is around 2.5
  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.

Exercice 8

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

Constant variability

Exercice 9

Based on the residuals vs. fitted plot, does the constant variability condition appear to be met? based on the residuals vs. fitted plot, there no pattern in the plot , which mean our assumption of constant variance (constant variability) condition appear to be met.

More Practice

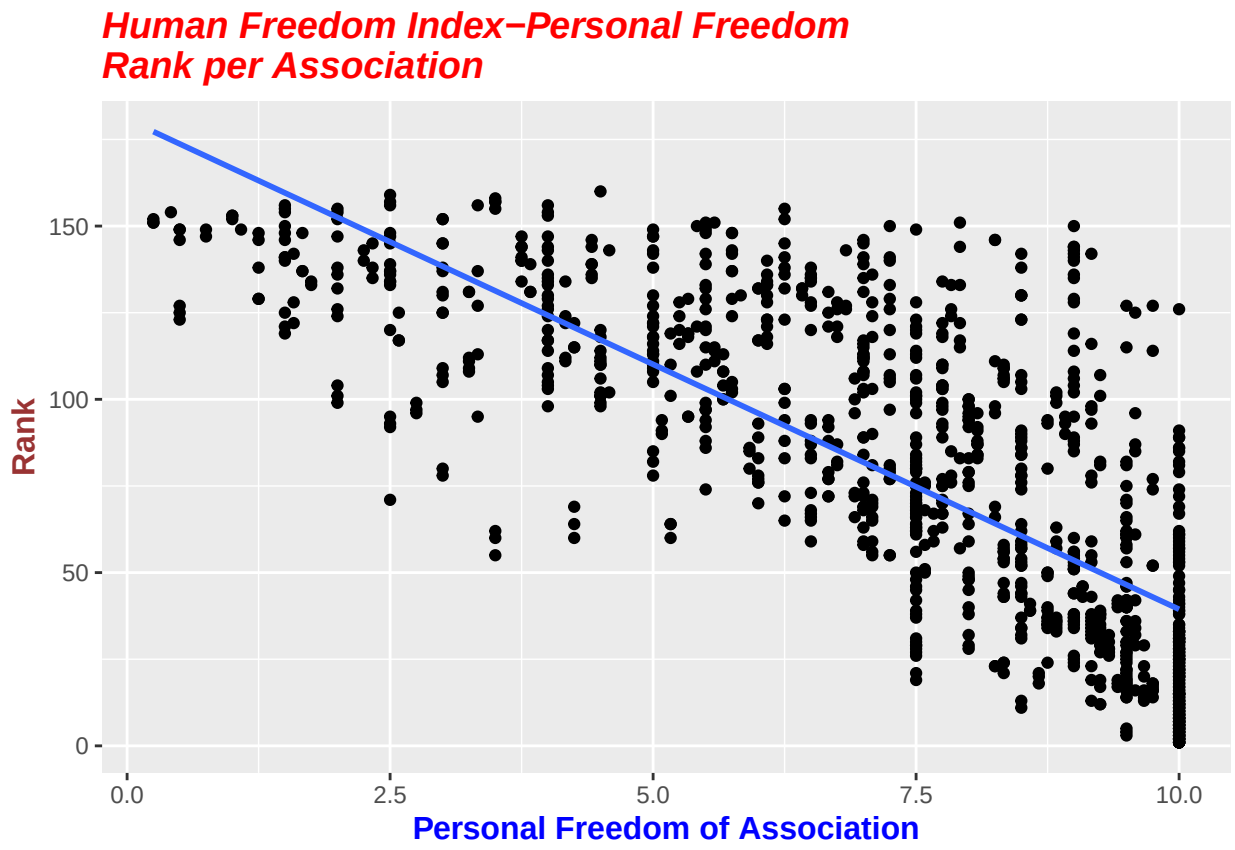
Choose another freedom variable and a variable you think would strongly correlate with it.. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship? Display the relationship between the personal freedom rank, pf_rank, and pf_association as the predictor.

```
view(hfi)
# Just replicate few lines from above, this time with new variable hf_score  pf_association, pf_rank
hfi4 <- select(hfi, pf_association, pf_rank )
#checking how many values are missing in hfi
sum(is.na(hfi4))
```

```
## [1] 409
```

```
hfi4 <- na.omit(hfi4)

#actually let's see a fit line
ggplot(hfi4, aes(x=pf_association, y=pf_rank)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Human Freedom Index-Personal Freedom \nRank per Association") +
  xlab("Personal Freedom of Association") + ylab("Rank") + theme
```

```
## `geom_smooth()` using formula 'y ~ x'
```



How does this relationship compare to the relationship between pf_expression_control and pf_score? Use the R2 values from the two model summaries to compare. Does your independent variable seem to predict your dependent one better? Why or why not? Answer: coefficient of determination/coefficient of multiple determination/R-squared is a statistical measure of how close the data are to the fitted regression line. if R-squared = 100%, then the model explain all the variability of the response data around its mean. In other word, there is no variance in the dependent variable that can be explained by the predictors

pf_expression_control and pf_score: multiple R-squared = 0.6342 ~ 63.4% pf_association and pf_rank: multiple R-Squared = 0.5804 ~58%

Based on the R-squared of the two relationships , there is about 5.4$ of a difference. The independent variable or predictor or explanatory (pf_association) does not seem to predict better the dependent variable or response variable or outcome variable. And that is the coefficient of correlation of the relationship between pf_association and pf_rank downhill(negative) -0.7618509 while the relationship between pf_expression_control and pf_score has a better coefficient of correlation uphill(positive) 0.7963894.

```
m3 <- lm(pf_rank ~ pf_association, data = hfi4)
summary(m1)
```

```
##
## Call:
## lm(formula = pf_score ~ pf_expression_control, data = hfi2)
##
## 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
## Multiple R-squared:  0.6342, Adjusted R-squared:  0.634
## F-statistic:  2386 on 1 and 1376 DF,  p-value: < 2.2e-16
```

```
summary(m3)
```

```
##
## Call:
## lm(formula = pf_rank ~ pf_association, data = hfi4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.308 -21.902  -6.476  20.781  96.455
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    180.7927     2.8036   64.49   <2e-16 ***
## pf_association -14.1386     0.3581  -39.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.91 on 1127 degrees of freedom
## Multiple R-squared:  0.5804, Adjusted R-squared:   0.58
## F-statistic:  1559 on 1 and 1127 DF,  p-value: < 2.2e-16
```

```
cor(hfi4$pf_association, hfi4$pf_rank)
```

```
## [1] -0.7618509
```

```
cor(hfi2$pf_expression_control, hfi2$pf_score)
```

```
## [1] 0.7963894
```

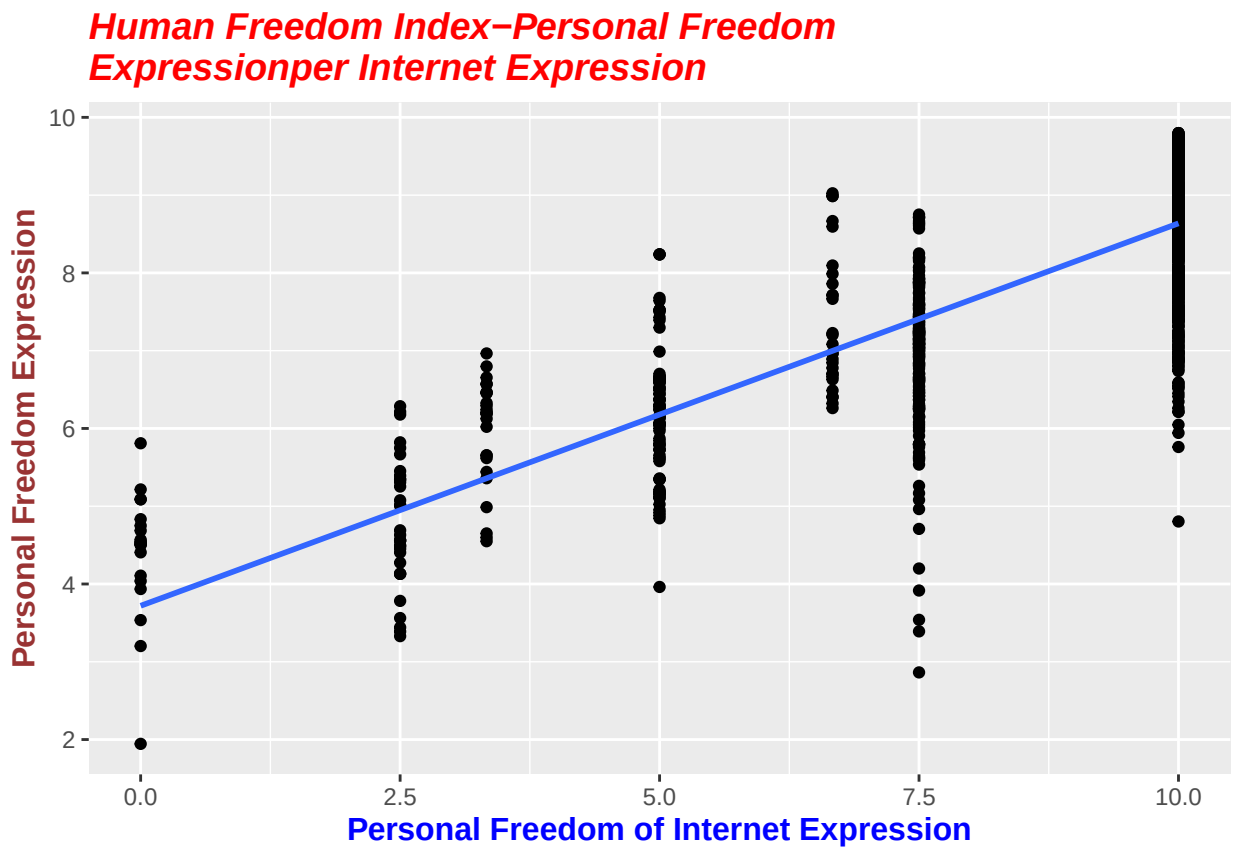
What's one freedom relationship you were most surprised about and why? Display the model diagnostics for the regression model analyzing this relationship. Answer: pf_expression_internet, pf_expression...considering the penetration of internet and human rights world wide , we expect this relationship to be close to best fit. Howerver , it is not the case

```
hfi5 <- select(hfi, pf_expression_internet, pf_expression)
#checking how many values are missing in hfi
sum(is.na(hfi5))
```

```
## [1] 409
```

```
hfi5 <- na.omit(hfi5)

ggplot(hfi5, aes(x=pf_expression_internet, y=pf_expression)) + geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Human Freedom Index-Personal Freedom \nExpressionper Internet Expression ") +
  xlab("Personal Freedom of Internet Expression") + ylab("Personal Freedom Expression") + theme
```

```
## `geom_smooth()` using formula 'y ~ x'
```



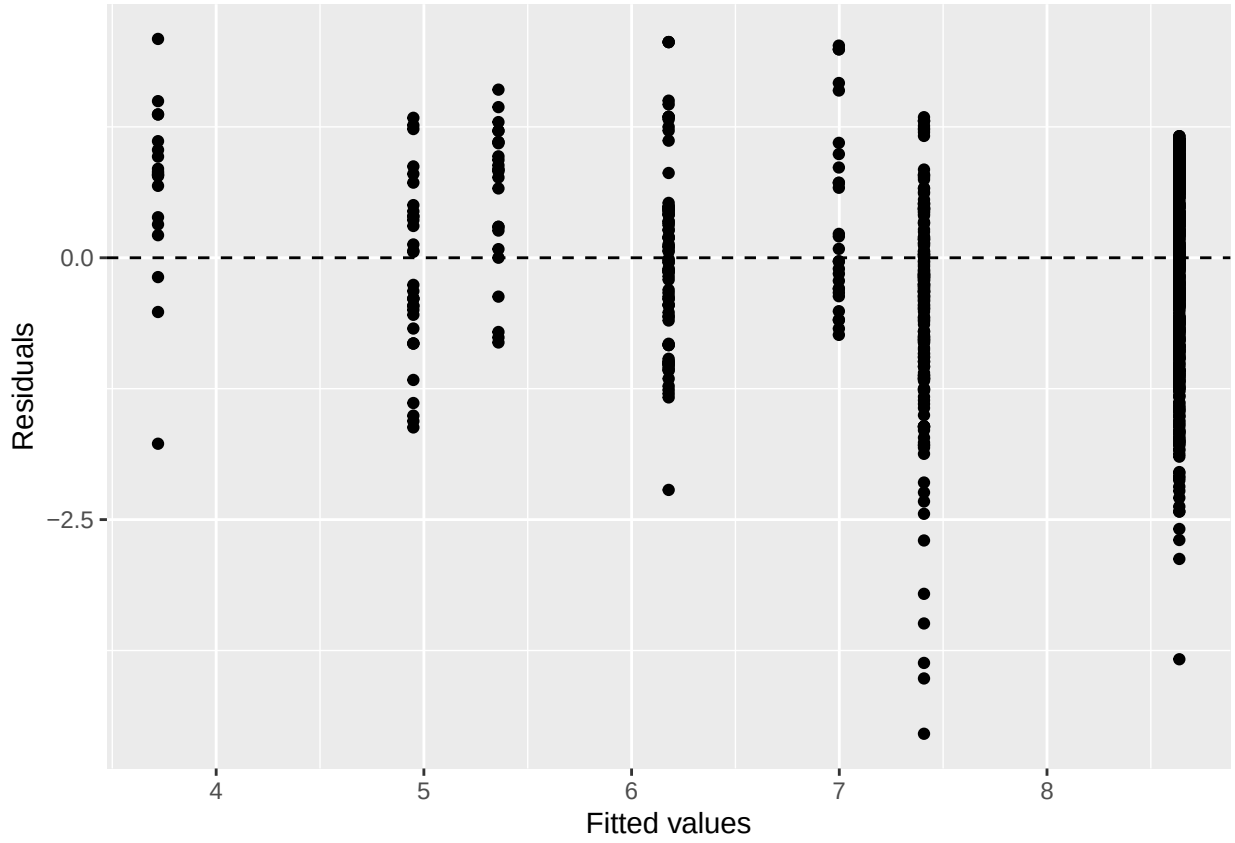

```
m5 <- lm(pf_expression ~ pf_expression_internet, data = hfi5)

summary(m5)
```

```
##
## Call:
## lm(formula = pf_expression ~ pf_expression_internet, data = hfi5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5446 -0.5575  0.0862  0.7155  2.0895
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.72006    0.10012   37.15   <2e-16 ***
## pf_expression_internet   0.49170    0.01128   43.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8949 on 1127 degrees of freedom
## Multiple R-squared:  0.6277, Adjusted R-squared:  0.6274
## F-statistic:  1900 on 1 and 1127 DF,  p-value: < 2.2e-16
```

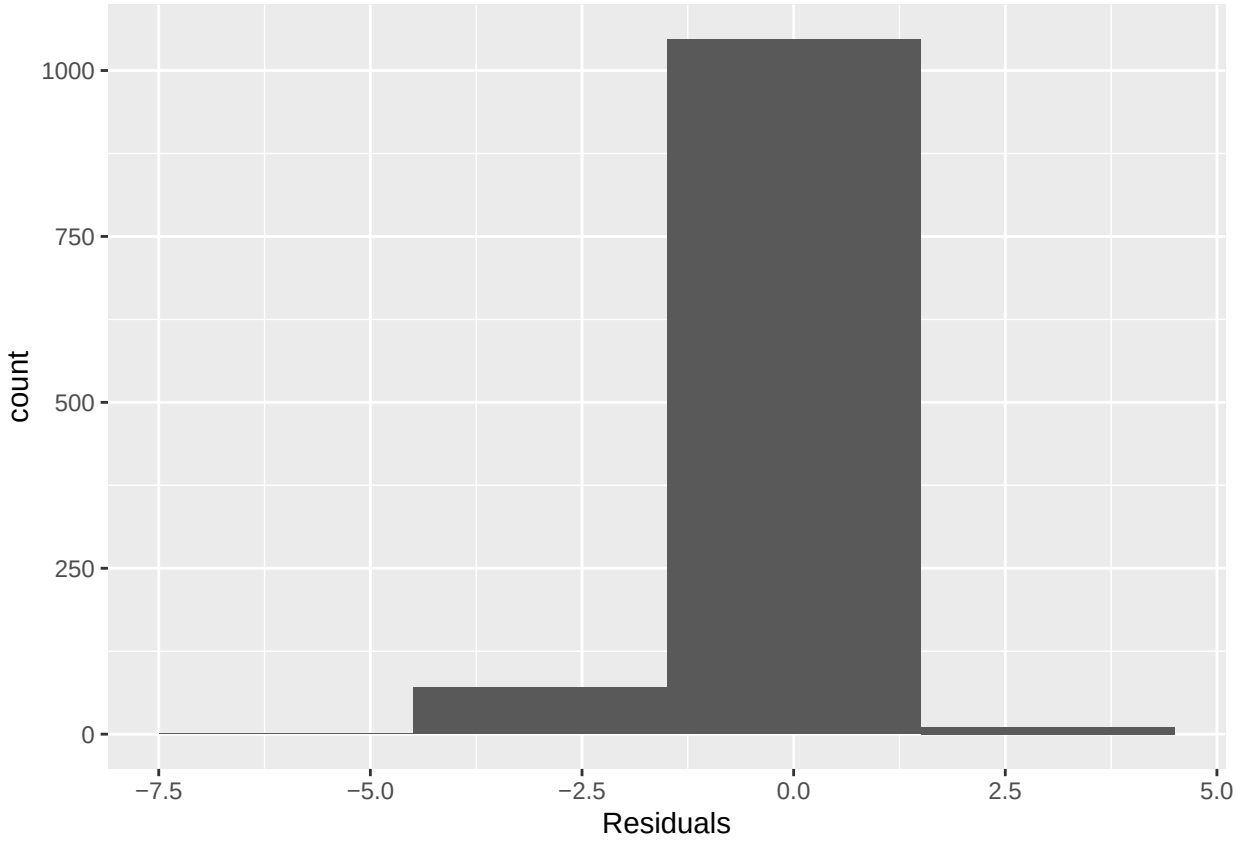
```
cor(hfi5$pf_expression_internet, hfi5$pf_expression)
```

```
## [1] 0.7922678
```

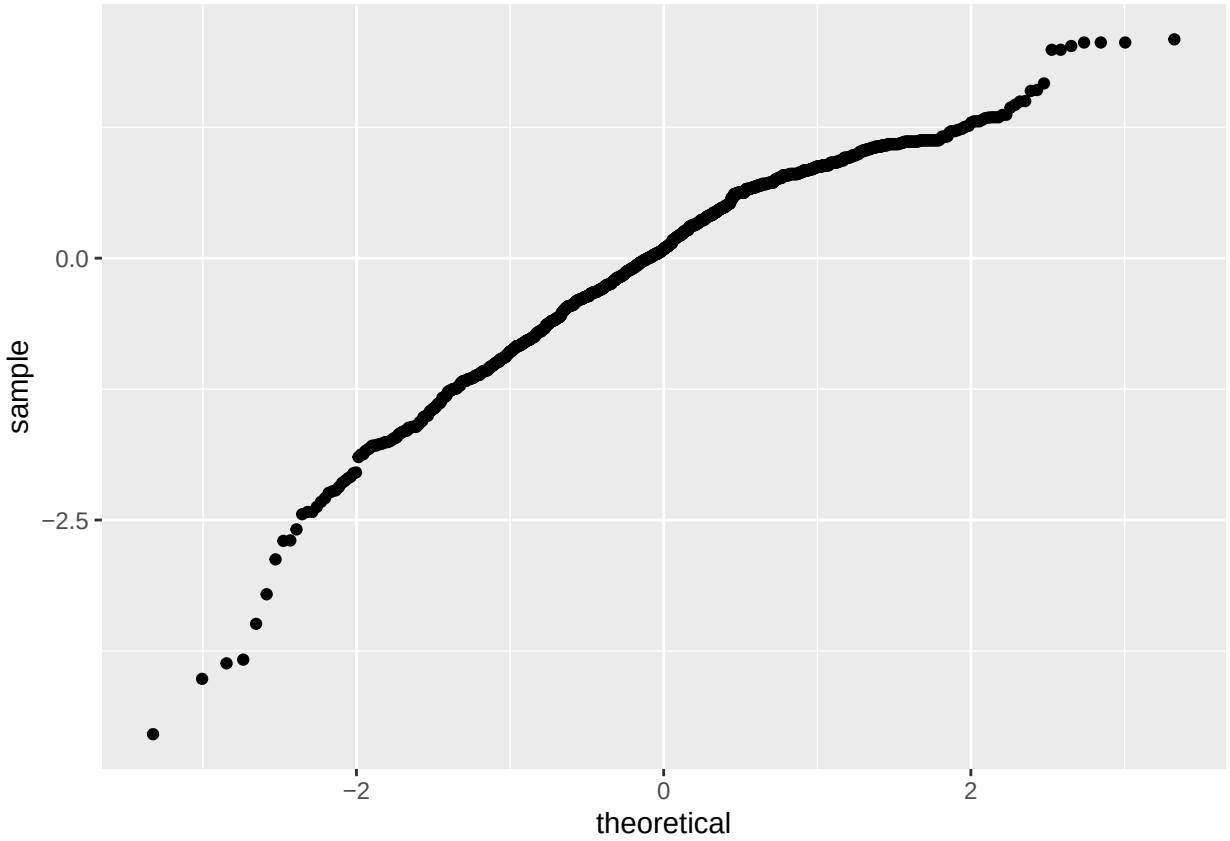
```
ggplot(data = m5, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```




```
ggplot(data = m5, aes(x = .resid)) +
  geom_histogram(binwidth = 3) +
  xlab("Residuals")
```




```
ggplot(data = m5, aes(sample = .resid)) +
  stat_qq()
```


0.0
−2.5
sample
−2
0
2
theoretical