Introduction to Linear Regression

Zachary Palmore

2020-11-01

```
library(tidyverse)
library(openintro)
library(statsr)
```

Pre-Lab

```
data(hfi)
```

Exercise 1

What are the dimensions of the dataset?

There are 1,458 rows and 123 columns.

```
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,...
```

Exercise 2

What type of plot would you use to display the relationship between the personal freedom score, pf_score, and one of the other numerical variables? Plot this relationship using the variable pf_expression_control as the predictor. Does the relationship look linear? If you knew a country’s pf_expression_control, or its score out of 10, with 0 being the most, of political pressures and controls on media content, would you be comfortable using a linear model to predict the personal freedom score?

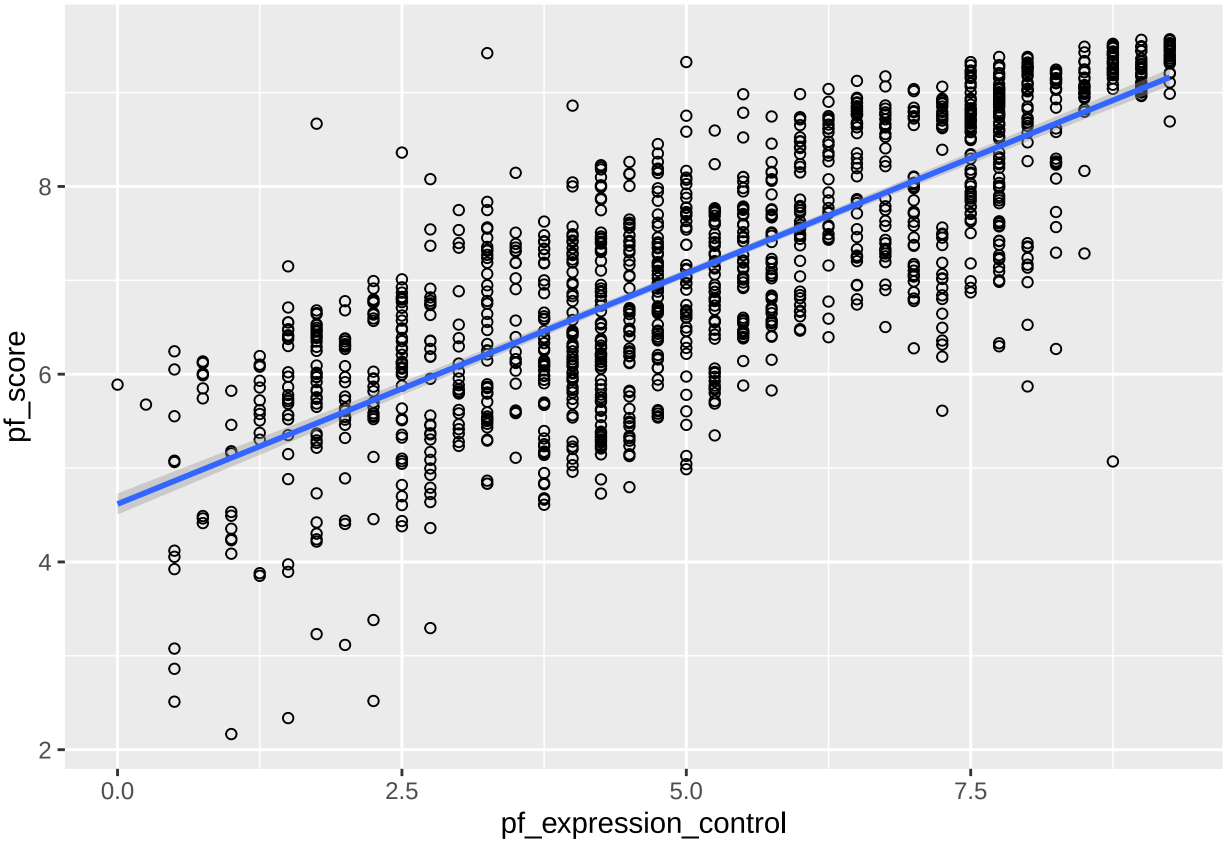
To plot two numerical variables you would use a scatter plot. This relationship looks linear. It exhibits a positive trend in pf_score as pf_expression_control increases. I would be reasonably comfortable using a linear model to predict the personal freedom score of a country from the pf_expression_control score because the data are well-correlated and the sample is large enough that we have a representative distribution.

```
ggplot(hfi, aes(x = pf_expression_control, y = pf_score)) + geom_point(shape=1) + geom_smooth(method =
```

```
## ‘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).
```



```
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
```

```
cor.test(hfi$pf_expression_control, hfi$pf_score)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hfi$pf_expression_control and hfi$pf_score
## t = 48.847, df = 1376, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7762265 0.8149248
## sample estimates:
##       cor
## 0.7963894
```

Exercise 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.

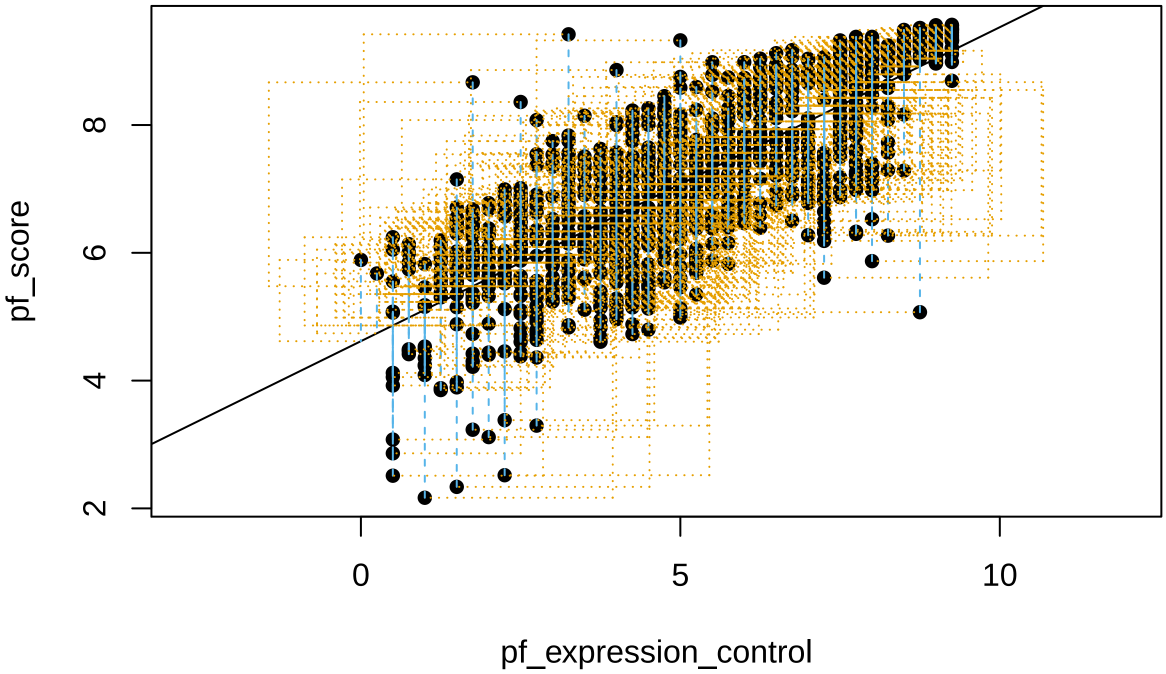
The relationship is strong, positive, and linear. Its form exhibits a positive upward trend. The pf_score increases as pf_expression_control increases. There are a small number of unusual observations but it is a very small proportion of the data overall.

Exercise 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?

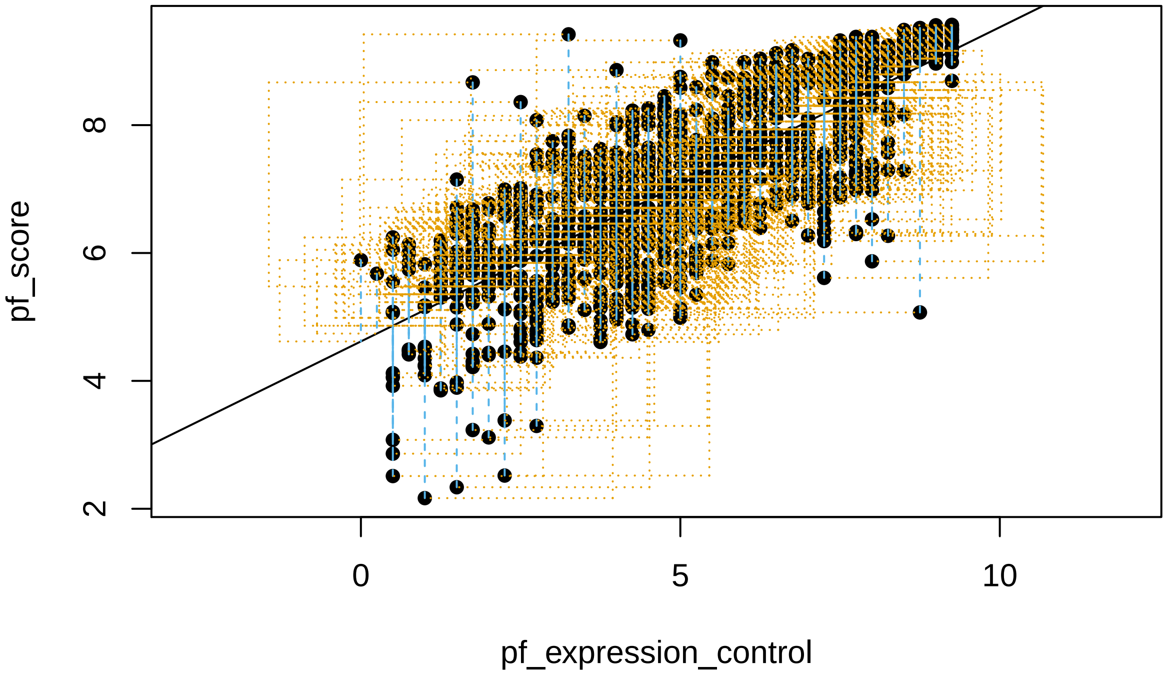
After running the function several times, the results were identical each time. The sum of squares returned was 952.153. It is about the same as my "neighbors."

```
m1 <- lm(pf_score ~ pf_expression_control, data = hfi)
pf_expression_control <- (hfi$pf_expression_control)
pf_score <-(hfi$pf_score)
pf_data <- cbind(pf_expression_control, pf_score)
pf_data <- as.data.frame(pf_data)
pf_data <- na.omit(pf_data)
# First Run
plot_ss(x = pf_expression_control, y = pf_score, data = pf_data, showSquares = TRUE)
```



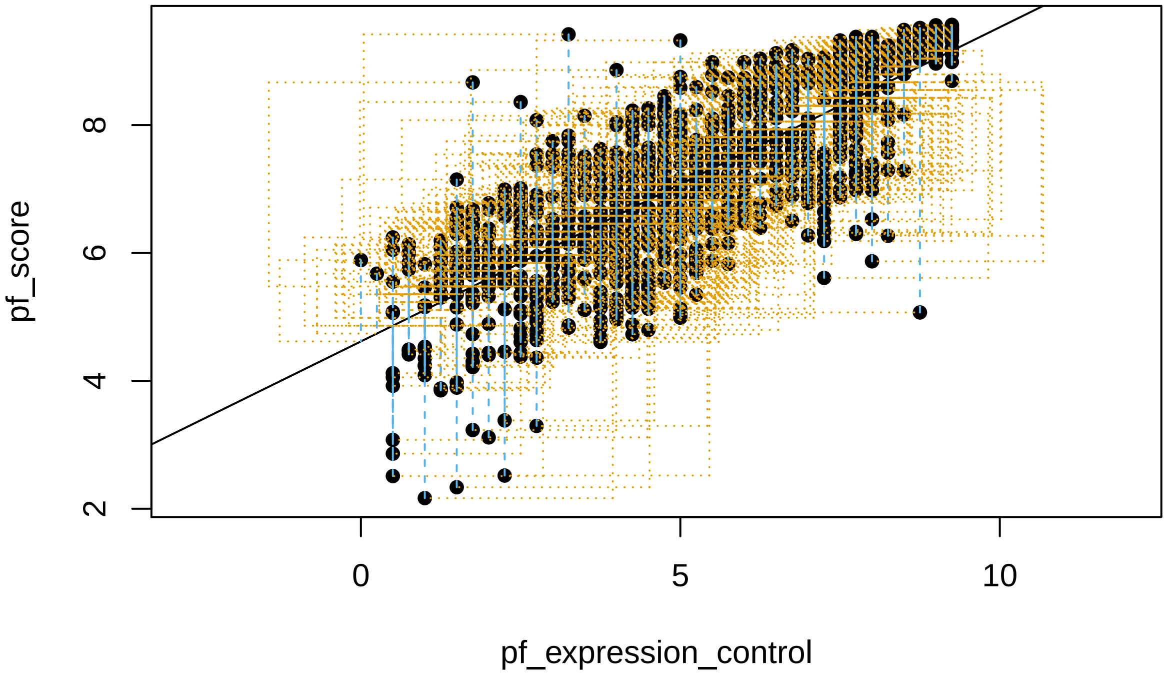
```
## Click two points to make a line.
## Call:
## lm(formula = y ~ x, data = pts)
##
## Coefficients:
## (Intercept)            x
##      4.6171       0.4914
##
## Sum of Squares:  952.153
```

```
# Second Run
plot_ss(x = pf_expression_control, y = pf_score, data = pf_data, 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
```

```
# Third Run
plot_ss(x = pf_expression_control, y = pf_score, data = pf_data, 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
```

The results of each run are identical.

Exercises 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?

The slope tells us the direction and rate of change of the relationship between human freedom and the amount of political pressure on media content. The more pressure on media content, the lower the human freedom score. Since the score for pf_expression_control starts at 0 and goes to 10, with 0 being the most pressure, the relationship shows a positive trend on the graph.

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

```
##
## Call:
## lm(formula = hf_score ~ pf_expression_control, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6198 -0.4908  0.1031  0.4703  2.2933
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.153687   0.046070  111.87   <2e-16 ***
## pf_expression_control 0.349862   0.008067   43.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.667 on 1376 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.5775, Adjusted R-squared:  0.5772
## F-statistic:  1881 on 1 and 1376 DF,  p-value: < 2.2e-16
```

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

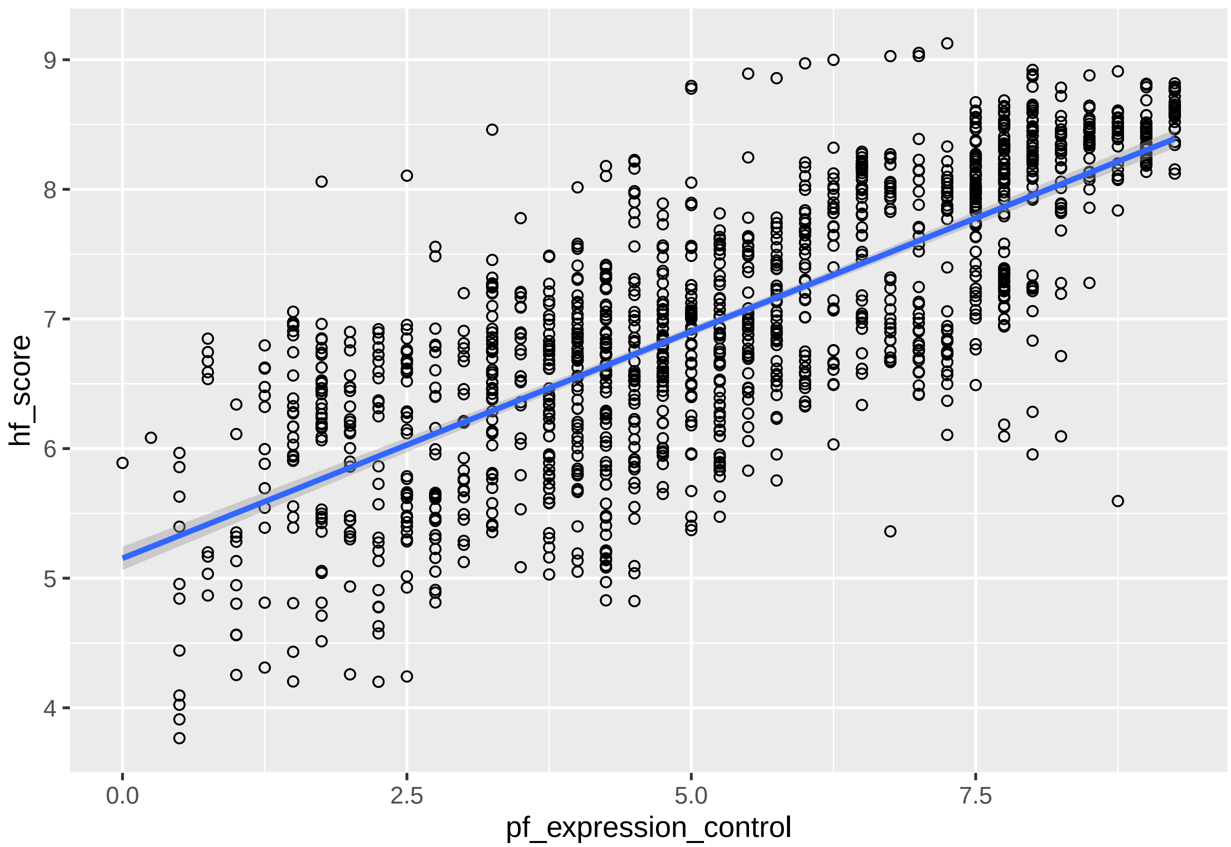
```
## # A tibble: 1 x 1
##   `cor(pf_expression_control, hf_score, use = "complete.obs")`
##                                                          <dbl>
## 1                                                        0.760
```

```
ggplot(hfi, aes(x = pf_expression_control, y = hf_score)) + geom_point(shape=1) + geom_smooth(method =
```

```
## `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).
```



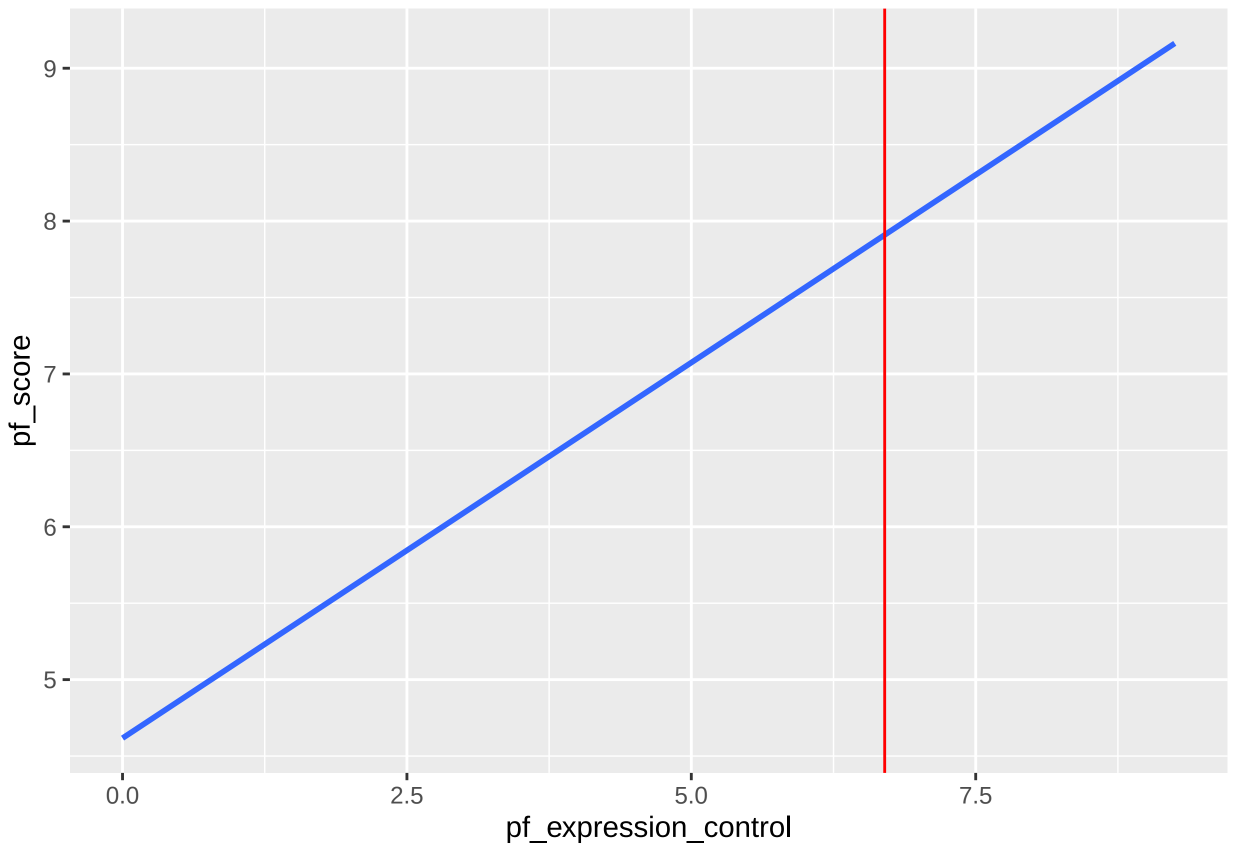
Exercise 6

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?

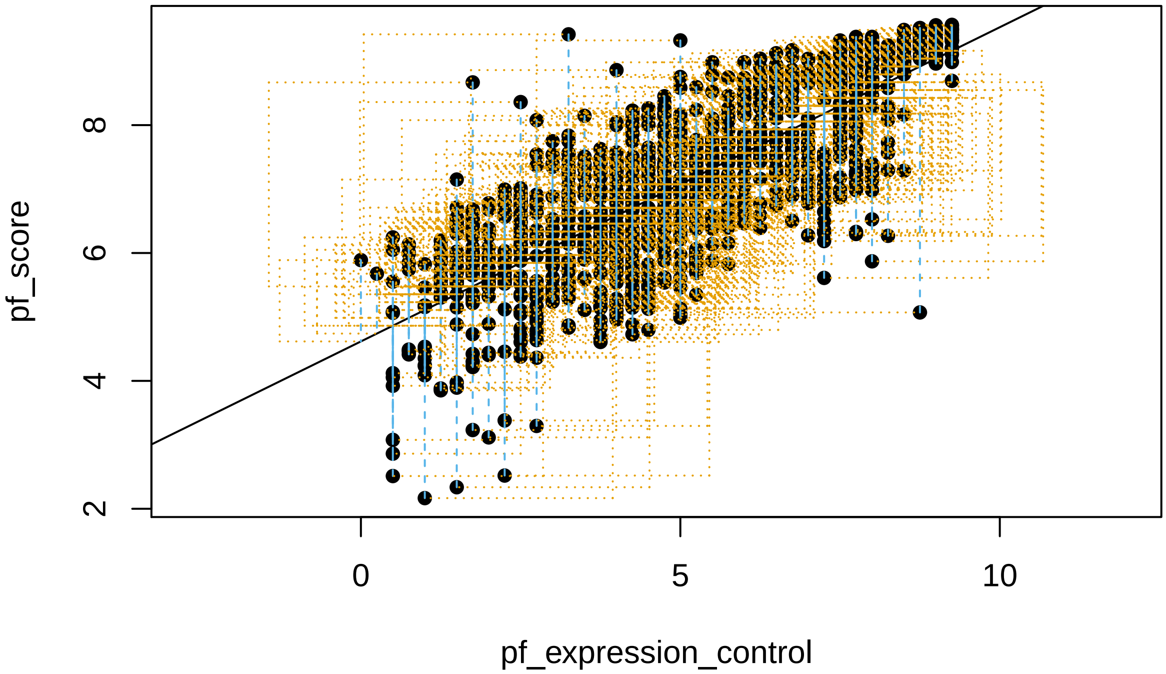
```
ggplot(data = hfi, aes(x = pf_expression_control, y = pf_score)) +
  stat_smooth(method = "lm", se = FALSE) + geom_vline(xintercept = 6.7, linetype="solid",
                color = "red", size=0.5)
```

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

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



```
plot_ss(x = pf_expression_control, y = pf_score, data = pf_data, 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
```

Based on the graph, if they saw the least squares regression line and not the actual data, they should predict a pf_score around 7.8 for a country's personal freedom school for a 6.7 pf_expression_control. Thus, we have the estimated point of (6.7, 7.8) and the equation of the line as $y = 4.6171 + 0.4914(x)$. We can compute the predicted value and compare.

```
y<-7.8
x<-6.7
int<-4.6171
slp<-0.4914
y_hat <-int+slp*(x)
y_hat
```

```
## [1] 7.90948
```

```
y-y_hat
```

```
## [1] -0.10948
```

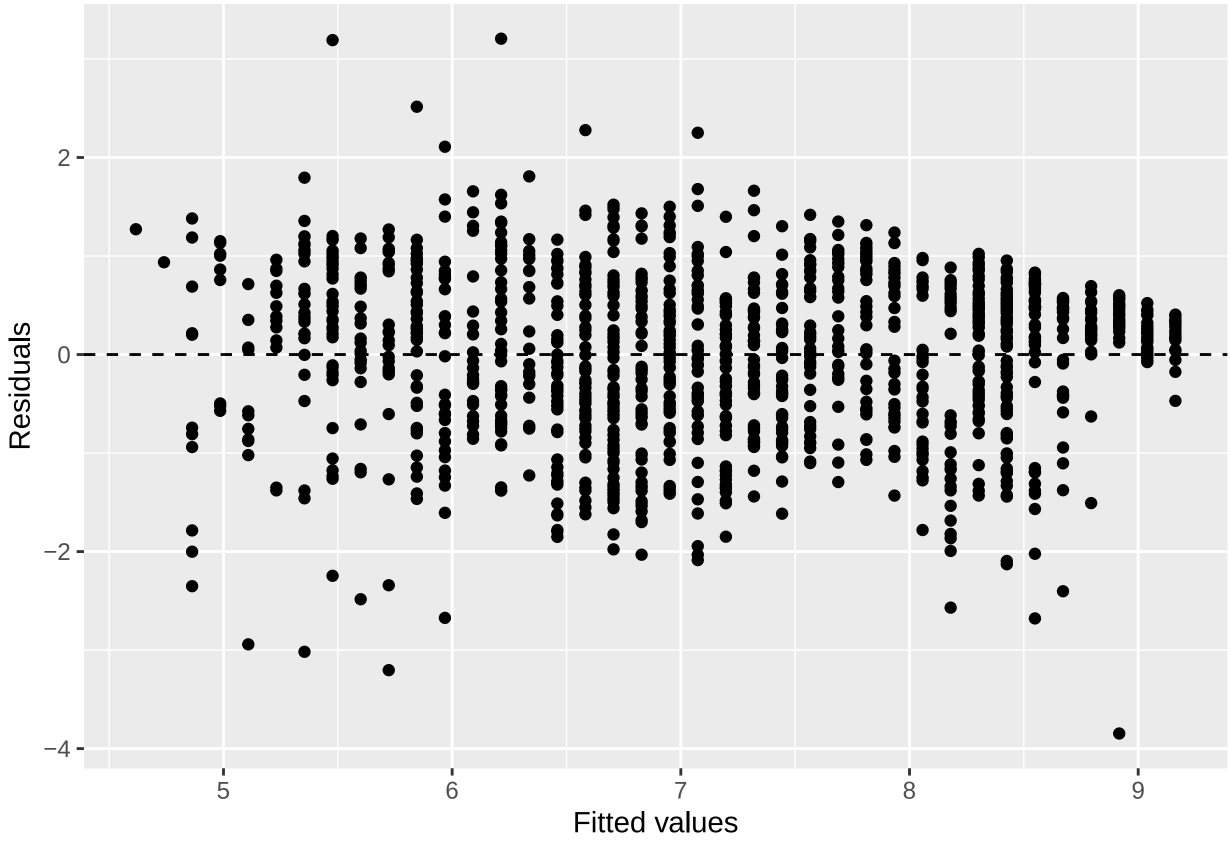
The error is -0.10948 which is close to the predicted. It is an overestimate by 0.10948 because the model would have predicted a higher value than the actual observation.

Exercise 7

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

These residuals show no obvious patterns. The residuals appear to be randomly scattered around zero and are roughly constant in variability. The linear trend in the data was strong and positive. All of this indicates the linearity of the relationship between the two variables is well-correlated and likely not due to chance.

```
ggplot(data = m1, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```

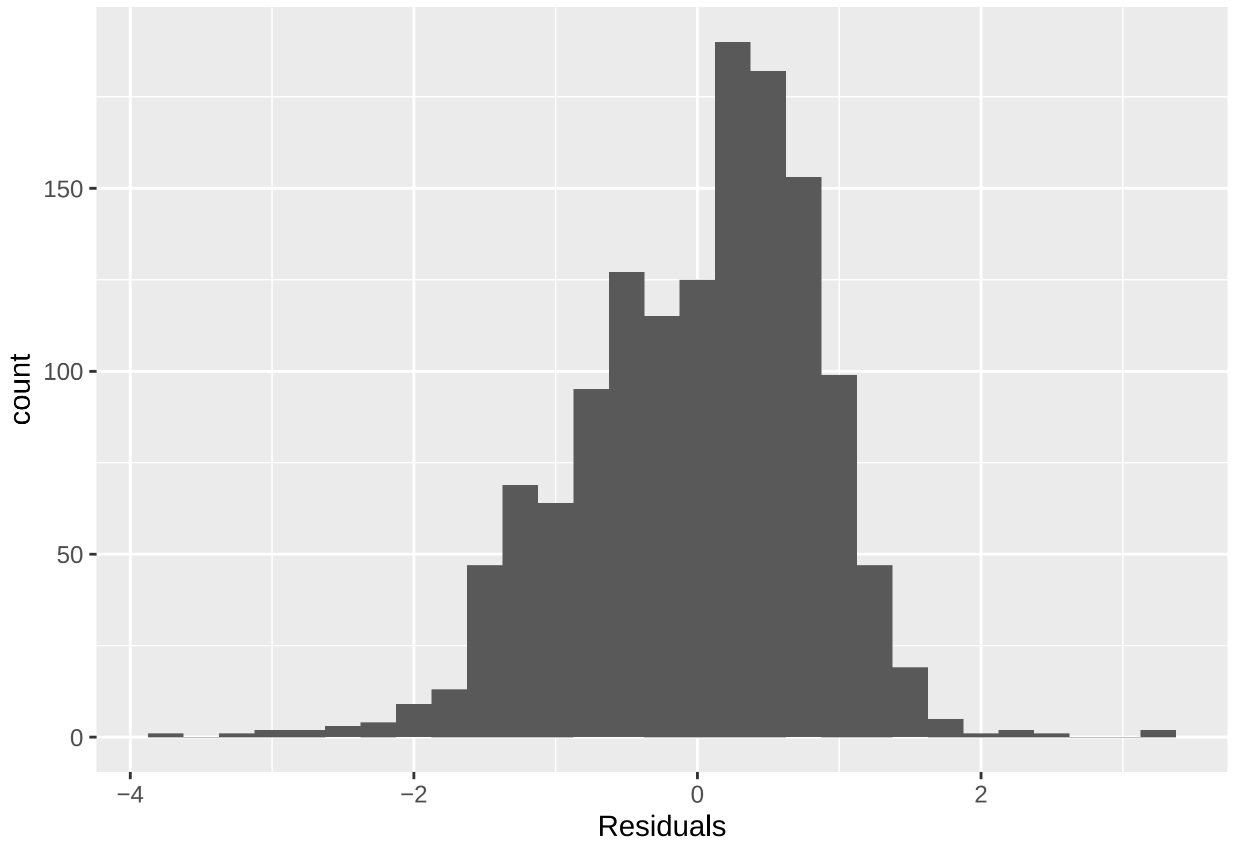


Exercise 8

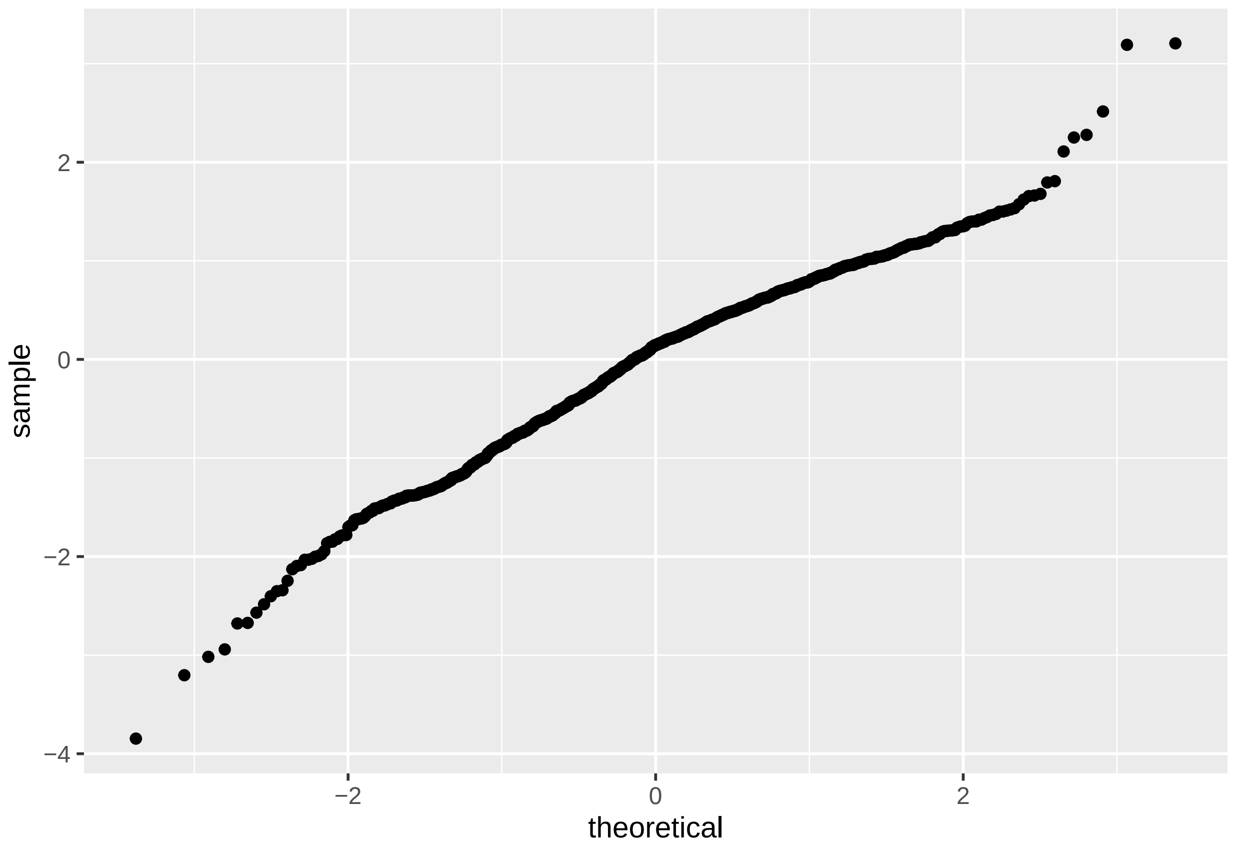
Based on the histogram and the normal probability plot, does the nearly normal residuals condition appear to be met?

Yes, the histogram of the residuals looks roughly unimodal and symmetric without many extreme values and therefore is nearly normal.

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



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



Exercise 9

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

Yes, it appears the variability of residuals around the zero line is roughly constant.

More Practice

1. 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?

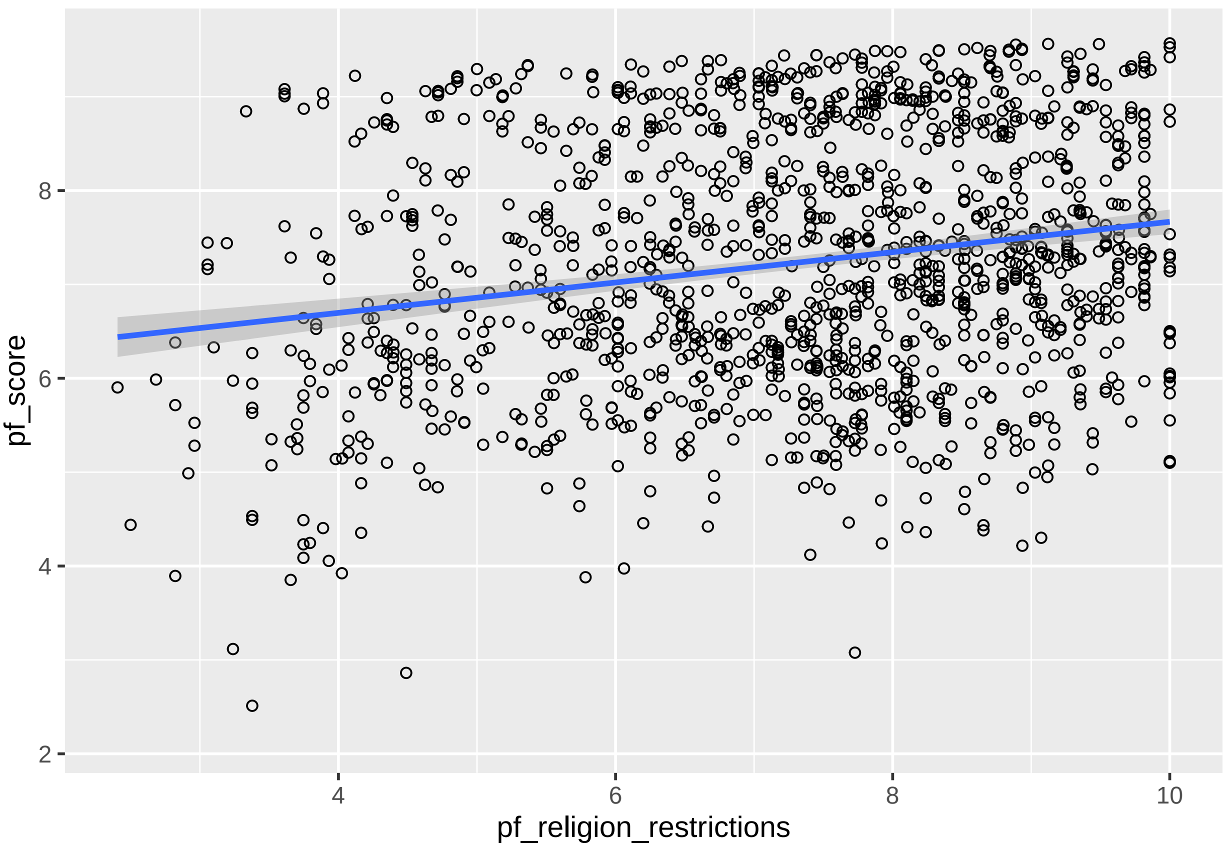
Yes, at a glance, there appears to be a slightly positive, weak, linear relationship between pf_religion_restrictions and pf_score.

```
ggplot(hfi, aes(x = pf_religion_restrictions, y = pf_score)) + geom_point(shape=1) + geom_smooth(method
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
## Warning: Removed 94 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 94 rows containing missing values (geom_point).
```



```
hfi %>%
  summarise(cor(pf_religion_restrictions, pf_score, use = "complete.obs"))
```

```
## # A tibble: 1 x 1
##   `cor(pf_religion_restrictions, pf_score, use = "complete.obs")`
##                                                              <dbl>
## 1                                                            0.206
```

```
cor.test(hfi$pf_religion_restrictions, hfi$pf_score)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hfi$pf_religion_restrictions and hfi$pf_score
## t = 7.7725, df = 1362, p-value = 1.507e-14
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1547003 0.2563589
## sample estimates:
##       cor
## 0.2060856
```

2. 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?

This relationship is much weaker than that of pf_expression_control and pf_score with a correlation coefficient of 0.2061. The independent variable does not seem to accurately predict the dependent variable better than pf_expression_control and pf_score. The R2 values from the two model summaries are 0.6342 for pf_expression_control and pf_score and 0.0425 for pf_religion_restrictions and pf_score. The R2 for pf_religion_restrictions and pf_score is much lower than pf_expression_control and pf_score which means the strength of the relationship between pf_expression_control and pf_score is greater and more correlated than pf_religion_restrictions and pf_score. This is why the pf_religion_restriction variable is worse at predicting the dependent pf_score.

```
# Rsquared of pf religion restrictions and pf score
pfsrr_lm <- lm(pf_religion_restrictions ~ pf_score, data = hfi)
summary(pfsrr_lm)$r.squared
```

```
## [1] 0.04247127
```

```
# Rsquared of pf expression control and pf score
pfsec_lm <- lm(pf_expression_control ~ pf_score, data = hfi)
summary(pfsec_lm)$r.squared
```

```
## [1] 0.6342361
```

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

I would consider the relationship between the rule of law and freedom score the most surprising of the relationships reviewed. The relationship satisfies conditions for the least square regression line but it looks like another type of best fit line may be a better choice. For example, a logarithmic relationship. The correlation between the variables pf_rol and pf_score are strong, positive, and linear but the residuals display another “u-shaped” trend. The R2 is 0.5997.

```
# Collect
m4 <- lm(pf_rol ~ pf_score, data = hfi)
summary(m4)
```

```
##
## Call:
## lm(formula = pf_rol ~ pf_score, data = hfi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9396 -0.6811 -0.0765  0.6228  3.4529
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.89301    0.13909  -6.421 1.86e-10 ***
## pf_score     0.86133    0.01897  45.401  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.968 on 1376 degrees of freedom
##   (80 observations deleted due to missingness)
## Multiple R-squared:  0.5997, Adjusted R-squared:  0.5994
## F-statistic:  2061 on 1 and 1376 DF,  p-value: < 2.2e-16
```

```
hfi %>%
  summarise(cor(pf_rol, pf_score, use = "complete.obs"))
```

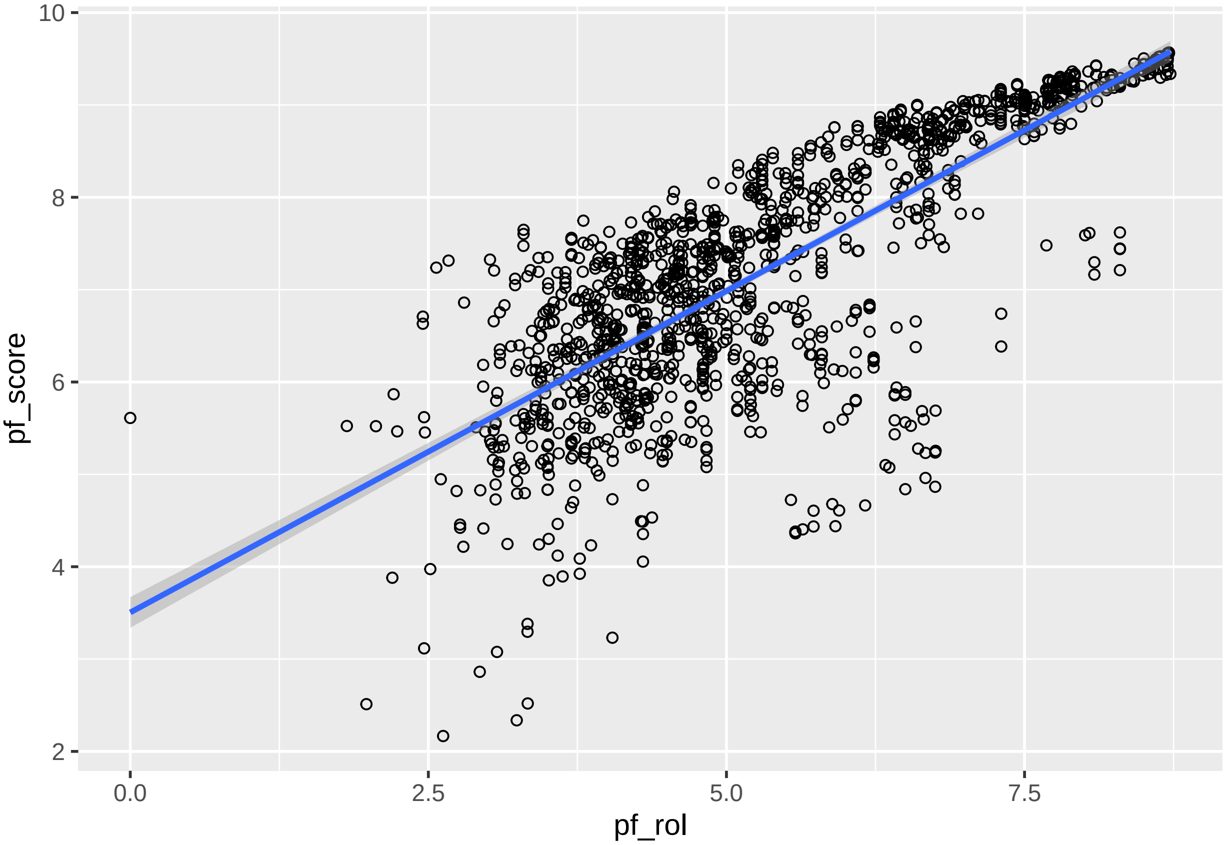
```
## # A tibble: 1 x 1
##   `cor(pf_rol, pf_score, use = "complete.obs")`
##                                           <dbl>
## 1                                         0.774
```

```
ggplot(hfi, aes(x = pf_rol, y = pf_score)) + geom_point(shape=1) + geom_smooth(method = "lm")
```

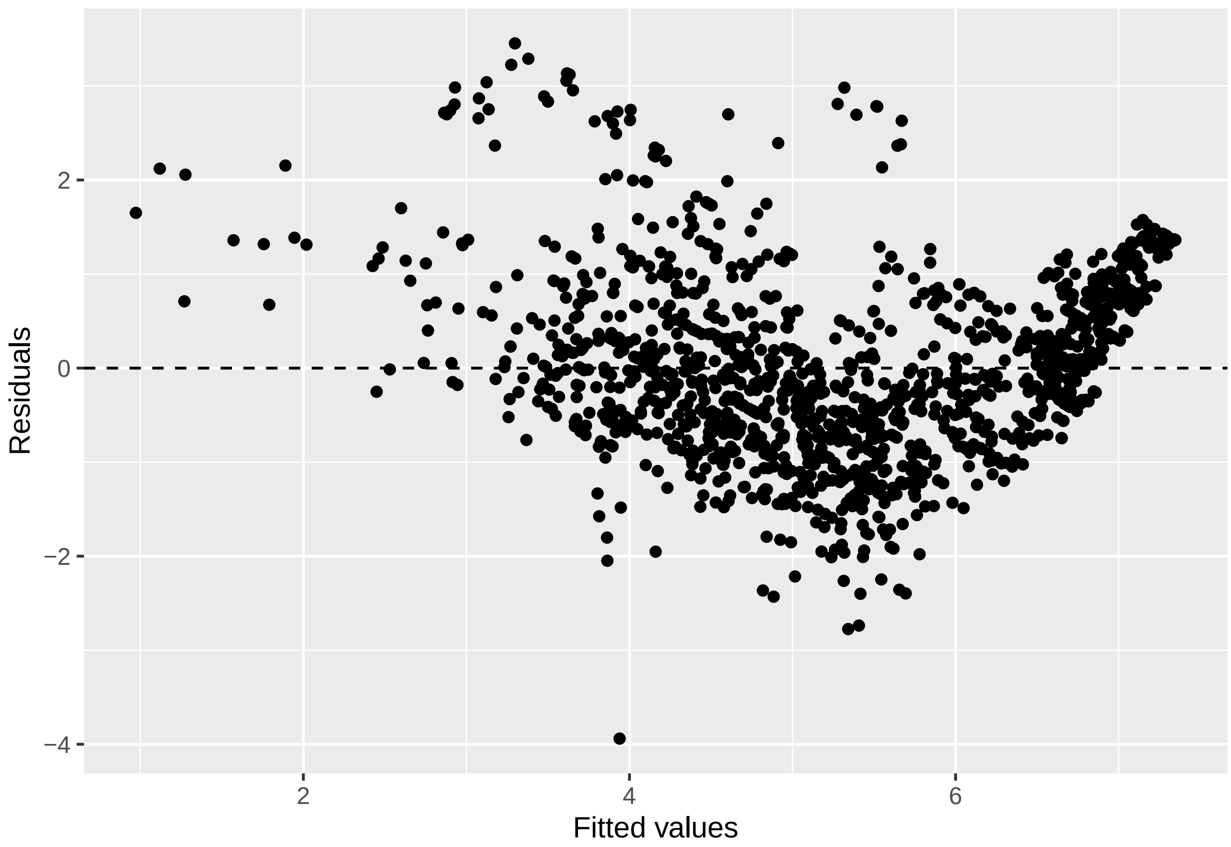
```
## `geom_smooth()` using formula 'y ~ x'
```

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

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



```
# Visualize
ggplot(data = m4, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



```
pfsrl_lm <- lm(pf_rol ~ pf_score, data = hfi)
summary(pfsrl_lm)$r.squared
```

```
## [1] 0.5996762
```

...