Statistics and Data Analysis Assignment

Domen Mohorčič, Larsen Cundrič

09/04/2021

INTRODUCTION

Our goal was to…

ROLLER COASTERS DATASET

The dataset Coaster2015 presents data from various roller coasters across the globe. It has 16 attributes: name, park city, state, country, type, construction, height, speed, length, inversions, numinversions, duration, geforce, opened and region.

Summary Statistics

```
roller_coasters_raw <- readr::read_csv('datasets/roller_coasters.csv')
```

Coaster2015 dataset has 408 instances. As we can see, some values are missing. Attributes name, park city, state, country, type, construction and regian are categorical, while others are numerical. Where type and construction are basically the same attributes as seen later on…

```
summary(roller_coasters_raw)
```

```
##      Name               Park               City              State          
##  Length:408         Length:408         Length:408         Length:408        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    Country              Type           Construction           Height       
##  Length:408         Length:408         Length:408         Min.   :  2.438  
##  Class :character   Class :character   Class :character   1st Qu.:  8.651  
##  Mode  :character   Mode  :character   Mode  :character   Median : 18.288  
##                                                           Mean   : 23.125  
##                                                           3rd Qu.: 33.167  
##                                                           Max.   :128.016  
##                                                           NA's   :82
```

```
##      Speed           Length         Inversions        Numinversions
##  Min.   :  9.72   Min.   :  12.19   Length:408         Min.   : 0.0000
##  1st Qu.: 45.00   1st Qu.: 291.00   Class :character   1st Qu.: 0.0000
##  Median : 68.85   Median : 415.75   Mode  :character   Median : 0.0000
##  Mean   : 69.36   Mean   : 597.04                      Mean   : 0.7843
##  3rd Qu.: 88.95   3rd Qu.: 833.12                      3rd Qu.: 0.0000
##  Max.   :194.40   Max.   :2243.02                      Max.   :10.0000
##  NA's   :138      NA's   :90
##     Duration         GForce          Opened        Region
##  Min.   :  0.3   Min.   :2.100   Min.   :1924   Length:408
##  1st Qu.: 75.0   1st Qu.:3.175   1st Qu.:1991   Class :character
##  Median :108.0   Median :4.500   Median :1999   Mode  :character
##  Mean   :112.5   Mean   :4.115   Mean   :1995
##  3rd Qu.:140.8   3rd Qu.:5.000   3rd Qu.:2004
##  Max.   :300.0   Max.   :6.200   Max.   :2014
##  NA's   :216     NA's   :348     NA's   :28
```

Lets have a look at the categorical variables first. We will skip the Name, Park, and City since they have 339, 168, and 150 unique values respectively. As for the rest, look at the summary below:

```r
table(roller_coasters_raw$Country)
```

```
##
##  AR  BR  CL  CO  CR   D  EQ   F  GT  MX  PE  US  VE
##  10  19   3   7   5  82   2  44   3  17   2 213   1
```

```r
table(roller_coasters_raw$State)
```

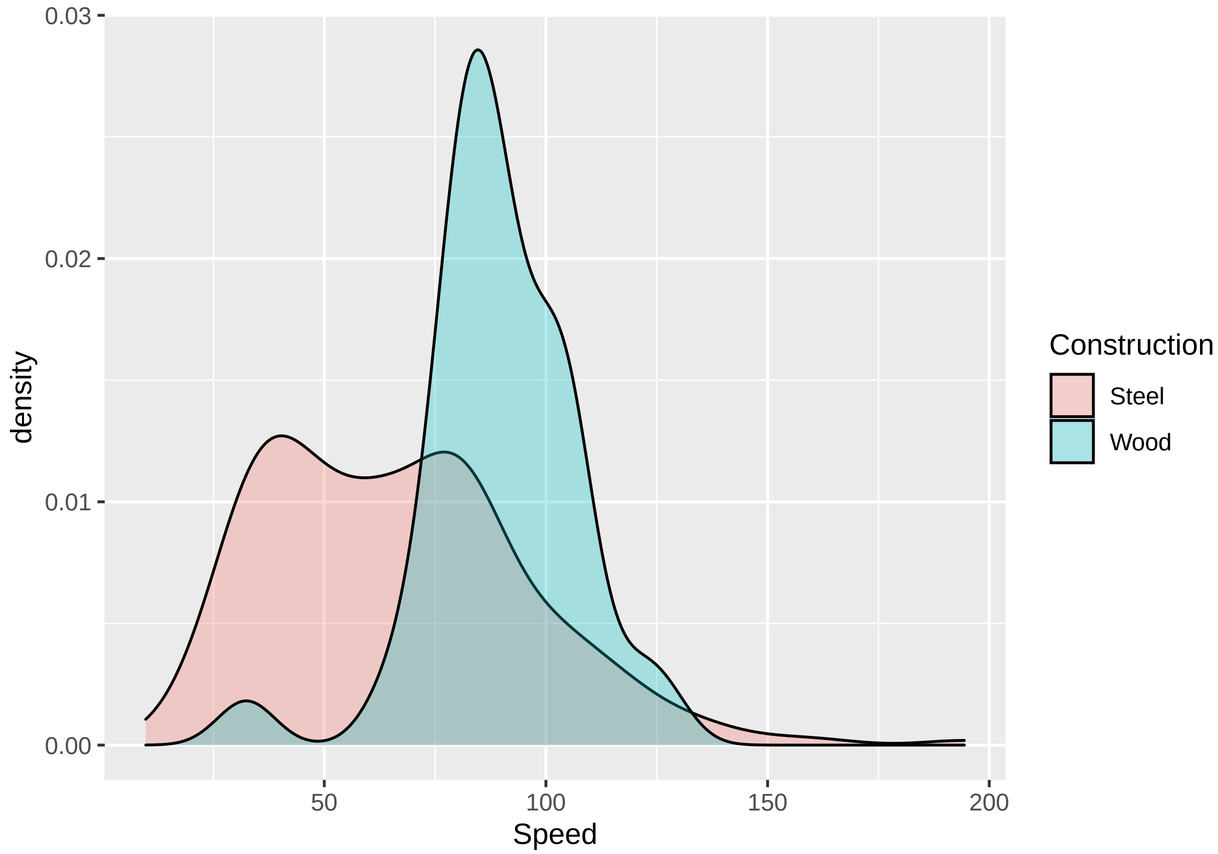
```
##
## AR BR CA CL CO CR  D EQ  F GT IL IN MX OH OR PE TX VE WA
## 10 19 77  3 21  5 82  2 44  3 18 13 17 37  4  2 38  1 12
```

```r
table(roller_coasters_raw$Construction) # Same as Type
```

```
##
## Steel  Wood
##   366    42
```

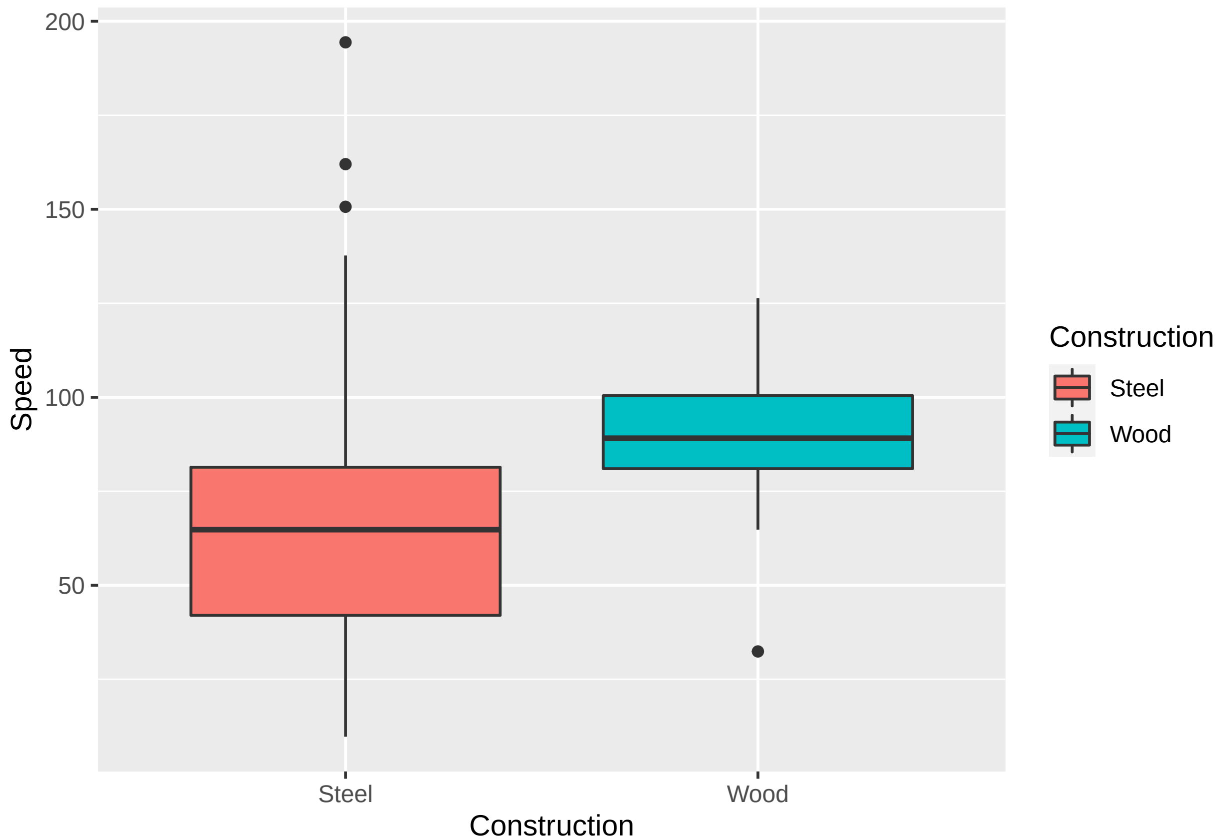
As for the numerical ones, we are generally most interested in speed. So we present most data relative to the speed of coasters. The speed is measured in milles per hour (mph), and its distribution relative to Construction can be seen here:

```r
roller_coasters_raw %>% ggplot()+
  geom_density(aes(x = Speed, fill = Construction), alpha = 0.3)
```



We present the same data on a boxplot:

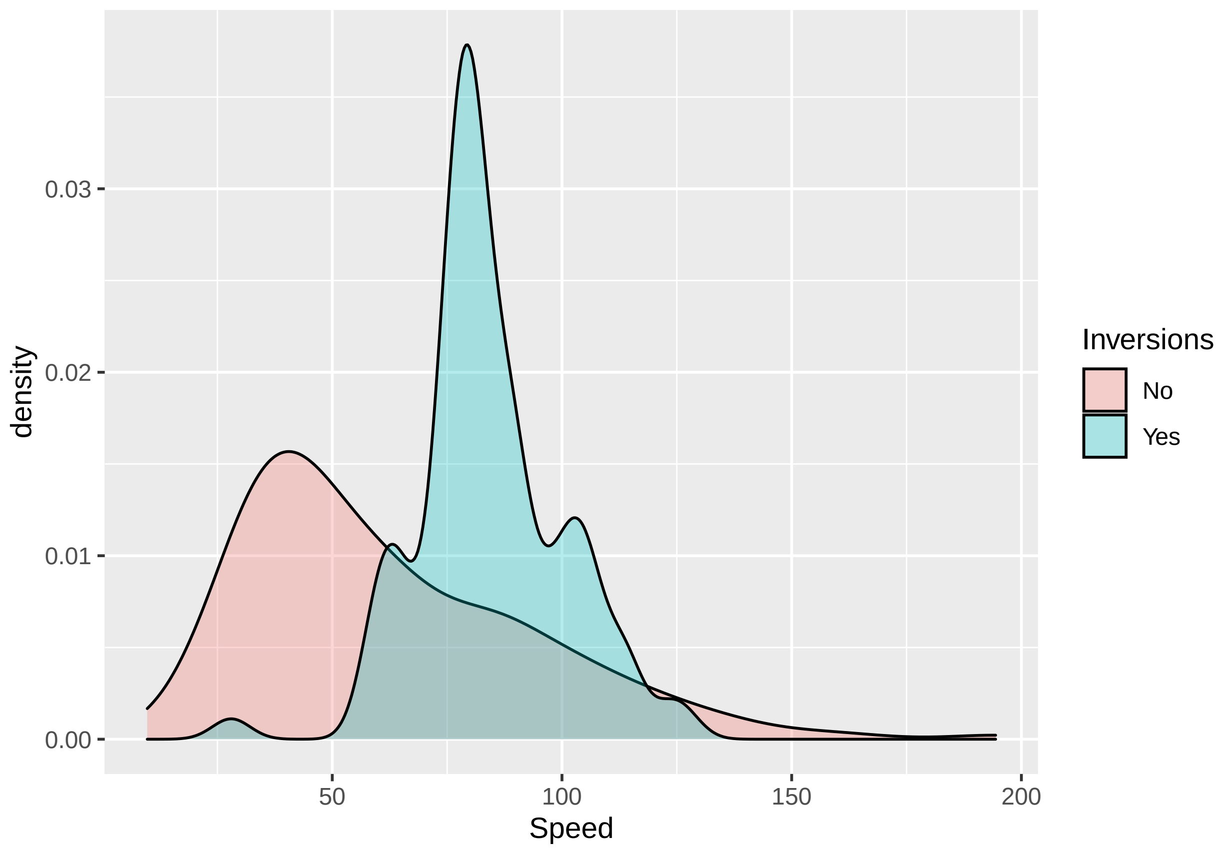
```
ggplot(data = roller_coasters_raw) +
  geom_boxplot(mapping = aes(x = Construction, y = Speed, fill = Construction))
```



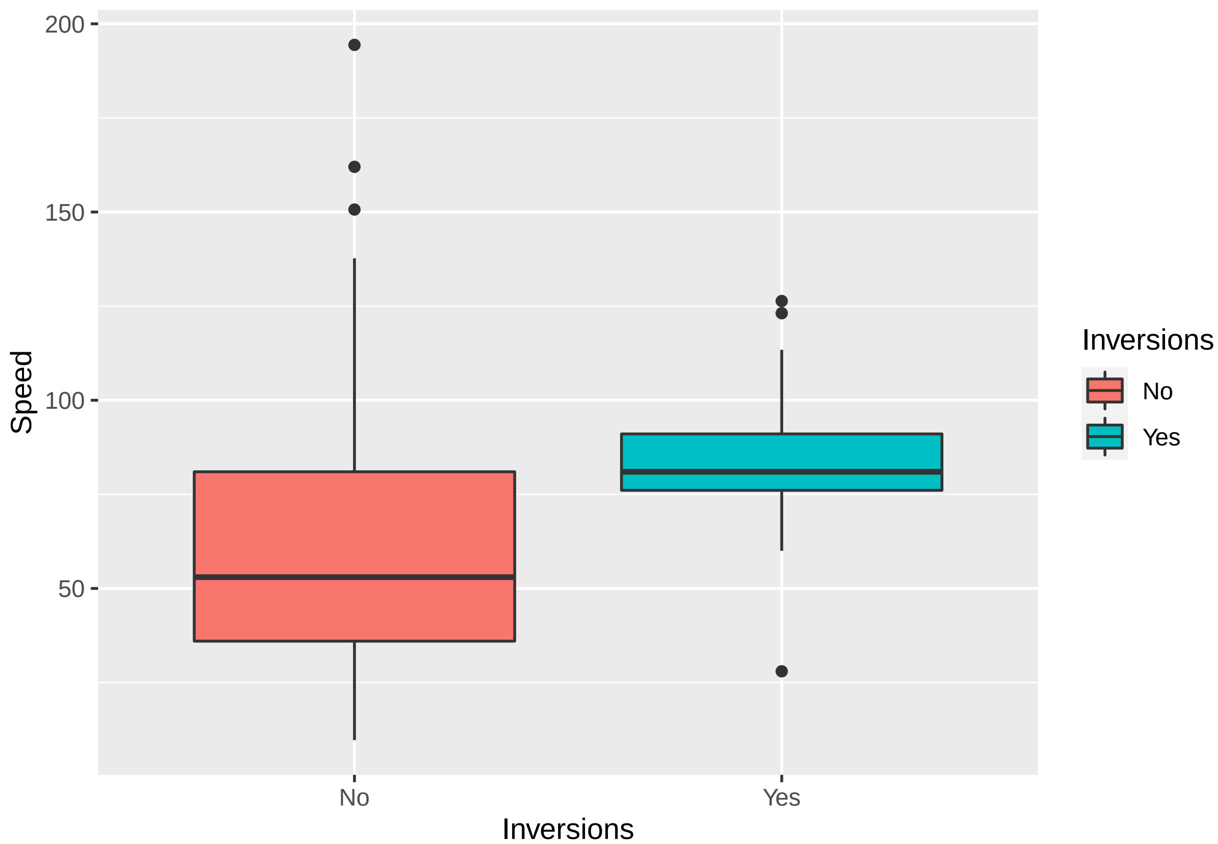
From the density plot and boxplot we can observe that wooden coaster are on average faster than the steel ones. This will also be one of the hypothesis tests later on to confirm our observations.

Inversions also present some interesting data. When we have inversions we tend to have higher speeds as shown below on a density plot and box plot:

```
roller_coasters_raw %>% ggplot()+
  geom_density(aes(x = Speed, fill = Inversions), alpha = 0.3)
```

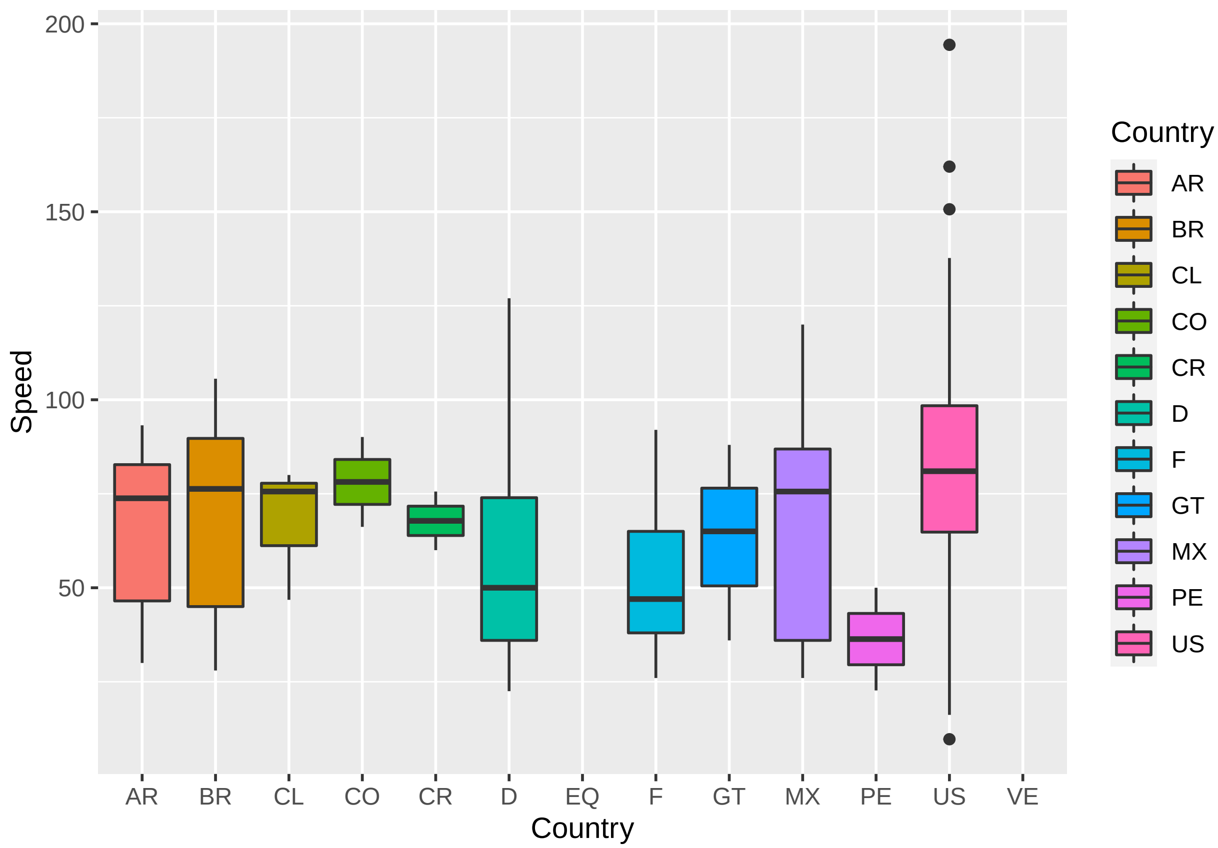



```r
ggplot(data = roller_coasters_raw) +
  geom_boxplot(mapping = aes(x = Inversions, y = Speed, fill = Inversions))
```



Last but not least, we compared the Countries and saw averages move from 50 to 75 mph, where US has the highest average:

```
ggplot(data = roller_coasters_raw) +
  geom_boxplot(mapping = aes(x = Country, y = Speed, fill = Country))
```



We also tested symmetry of Speed distributions for Steel and wood constructions:

```
symmetry.test(roller_coasters_raw$Speed)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw$Speed
## Test statistic = 0.37053, p-value = 0.806
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  58
```

```
symmetry.test(roller_coasters_raw[roller_coasters_raw$Construction == "Steel",]$Speed)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw[roller_coasters_raw$Construction == "Steel",     ]$Speed
## Test statistic = 1.1388, p-value = 0.312
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  32
```

```r
symmetry.test(roller_coasters_raw[roller_coasters_raw$Construction == "Wood",]$Speed)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw[roller_coasters_raw$Construction == "Wood",     ]$Speed
## Test statistic = 0.18255, p-value = 0.92
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  17
```

We see that Speed is a symmetric distribution and also both Steel and Wood have symmetric distributions which will help us later in the hypothesis testing.

Inference and Hypothesis testing

The usual procedure for hypothesis testing is such:

0) Check CLT conditions:

- Samples are independent,
- Sample size is bigger or equal to 30,
- Population distribution is not strongly skewed.

1) Set-up the hypothesis
2) Assume threshold values

- α - typically 0.05

3) Calculate the Results:

- point est.
- number of cases
- sd - standard deviation
- se - standard error
- df - degrees of freedom $df = n - 1$
- t-statistics
- p-value

4) Draw conclusions - Accept or reject hypothesis

If we meet those criteria, we can infer about the population based on the analysis we do on the sample. We firstly assume that all the instances are independent. We can also see that there are more than enough instances:

```r
roller_coasters_raw %>% filter(!is.na(Speed)) %>% nrow()
```

```
## [1] 270
```

```
roller_coasters_raw %>% filter(!is.na(Height)) %>% nrow()
```

```
## [1] 326
```

```
roller_coasters_raw %>% filter(!is.na(Length)) %>% nrow()
```

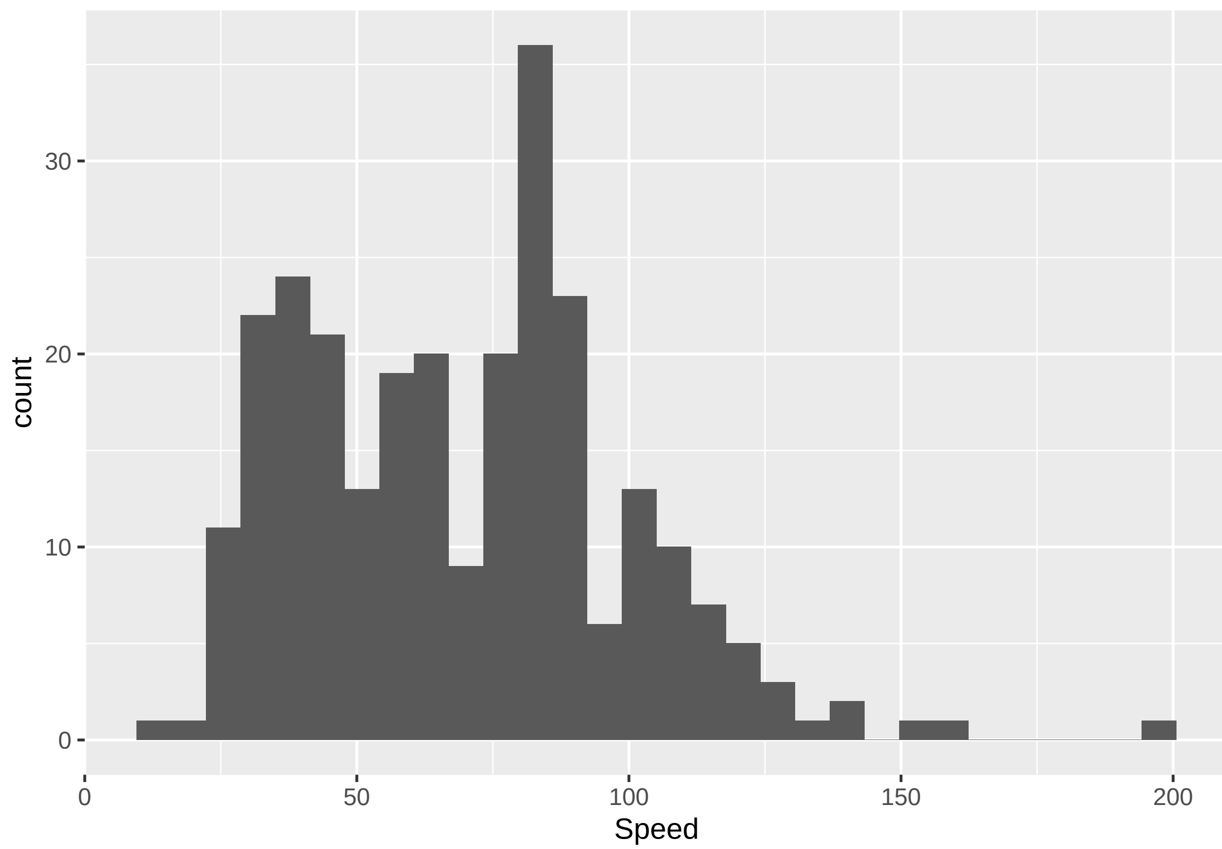
```
## [1] 318
```

```
roller_coasters_raw %>% filter(!is.na(Numinversions)) %>% nrow()
```

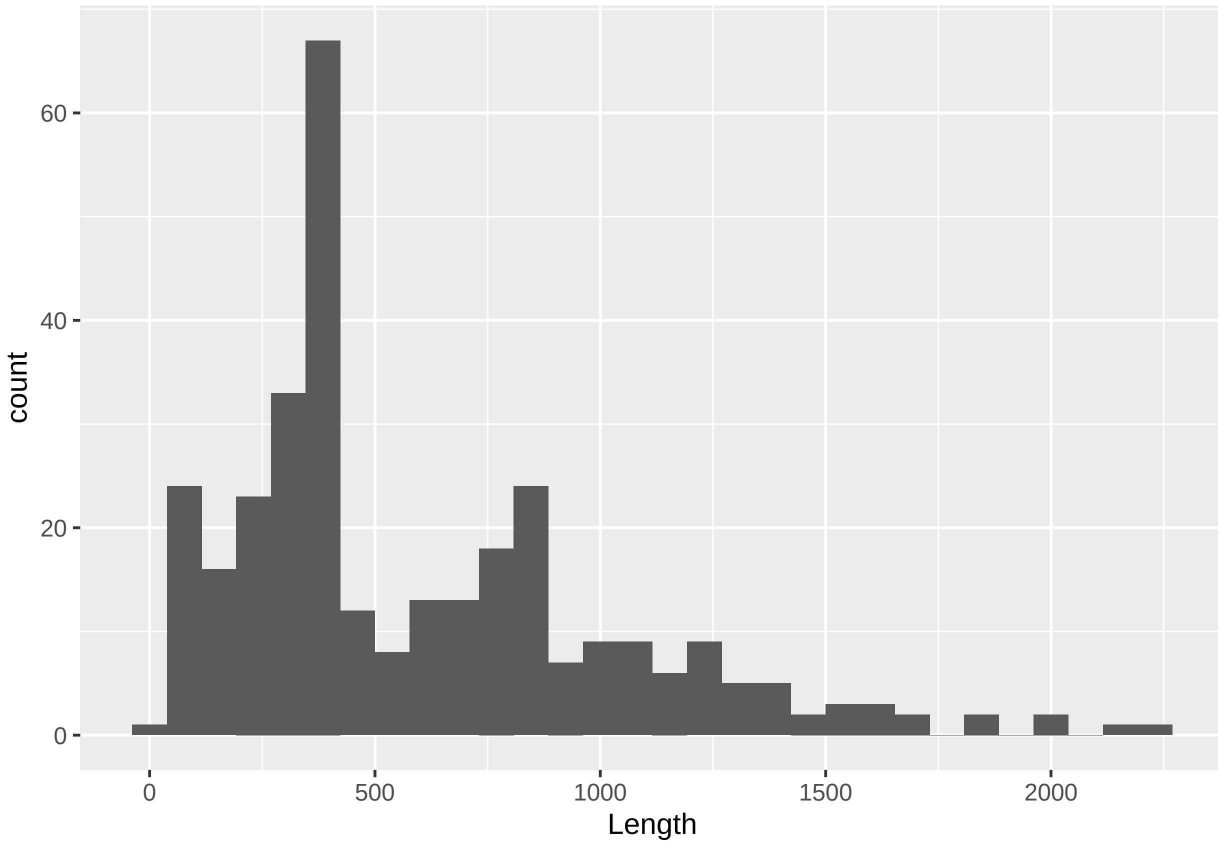
```
## [1] 408
```

Lastly, we want to see if the data is not heavily skewed:

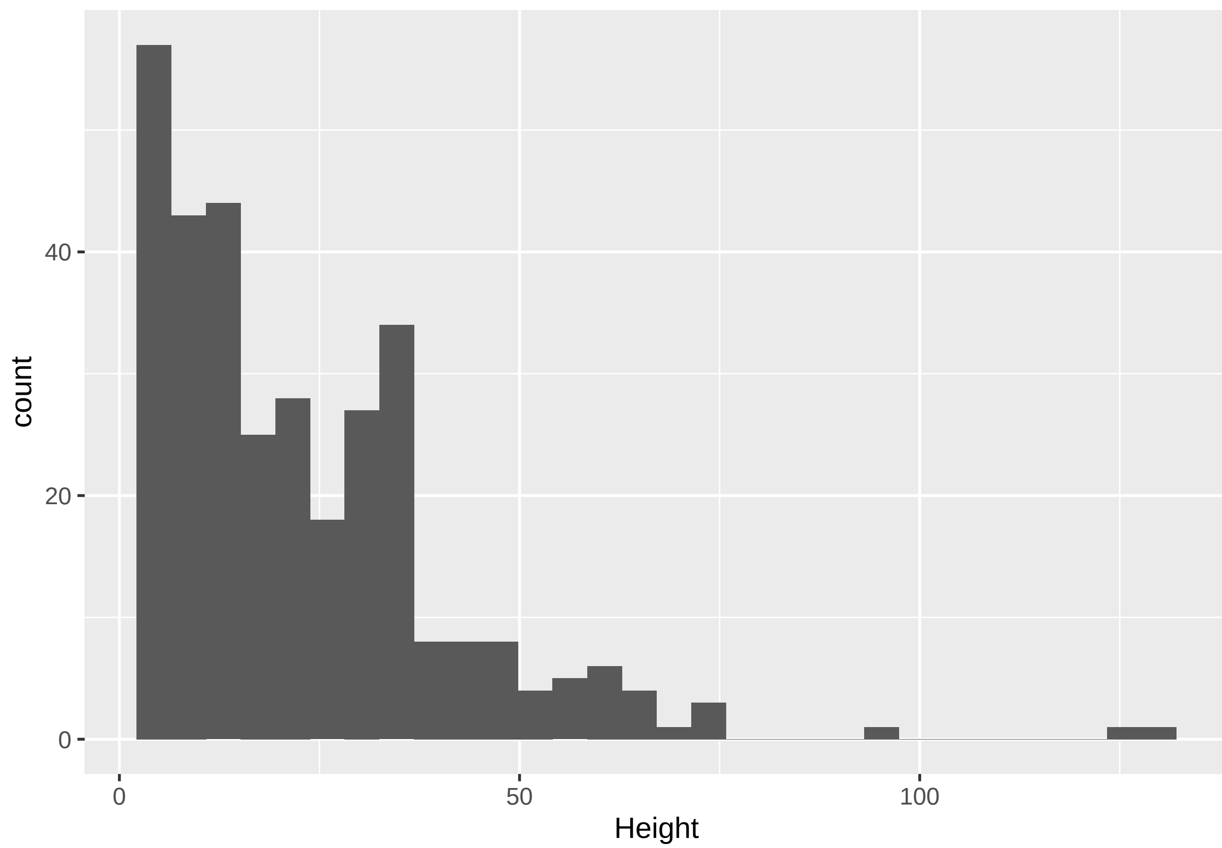
```
ggplot(roller_coasters_raw) +
  geom_histogram(aes(x = Speed))
```



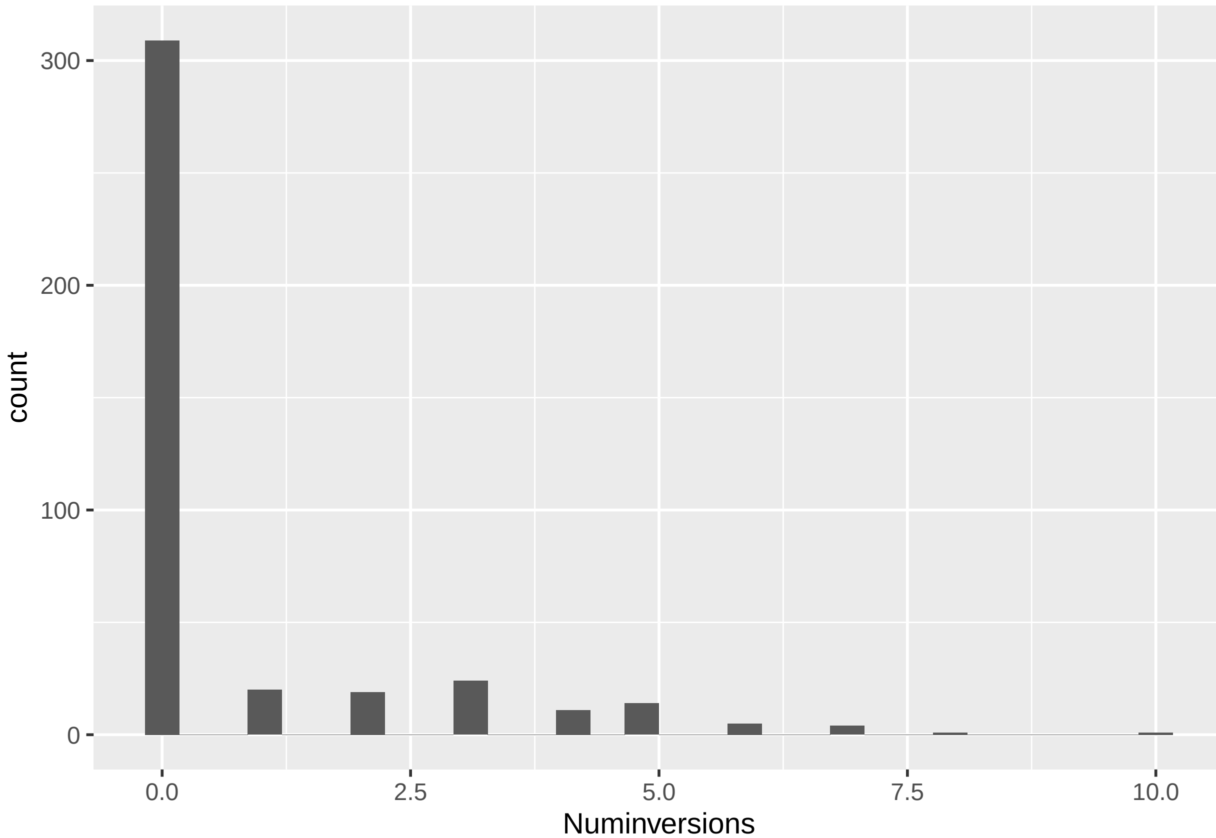

```
ggplot(roller_coasters_raw) +
  geom_histogram(aes(x = Length))
```




```
ggplot(roller_coasters_raw) +
  geom_histogram(aes(x = Height))
```




```r
ggplot(roller_coasters_raw) +
  geom_histogram(aes(x = Numinversions))
```



From the above distributions we can observe that the most suitable distribution to make hypothesis testing on is Speed. And its symmetry is already proven in the summary statistics section.

We can also prove that Height, Length and Numinversions are not normally distributed nor are they symmetric using the symmetry and shapiro test below:

```
symmetry.test(roller_coasters_raw$Height)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw$Height
## Test statistic = 6.772, p-value < 2.2e-16
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  44
```

```
shapiro.test(roller_coasters_raw$Height)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  roller_coasters_raw$Height
## W = 0.84671, p-value < 2.2e-16
```

```
symmetry.test(roller_coasters_raw$Length)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw$Length
## Test statistic = 10.298, p-value < 2.2e-16
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  77
```

```
shapiro.test(roller_coasters_raw$Length)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  roller_coasters_raw$Length
## W = 0.90217, p-value = 1.789e-13
```

```
symmetry.test(roller_coasters_raw$Numinversions)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  roller_coasters_raw$Numinversions
## Test statistic = 21.332, p-value < 2.2e-16
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                  44
```

```
shapiro.test(roller_coasters_raw$Numinversions)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  roller_coasters_raw$Numinversions
## W = 0.54578, p-value < 2.2e-16
```

As such, we are allowed to infere and do hypothesis testing on Speed, since only Speed meets the Limit Theorem requirements.

```
roller_coasters_speeds <- roller_coasters_raw %>%
  select(Speed) %>%
  filter(!is.na(Speed))
```

Hypothesis 1 - One sample t-test

Is the mean speed of roller coasters equal to 70mph?

$$H_0 : \mu = 70$$
$$H_A : \mu \neq 70$$
$$\alpha = 0.05$$

We calculate the necessary variables:

```
(point_est_speed <- 70)
```

```
## [1] 70
```

```
(mean_speed <- mean(roller_coasters_speeds$Speed))
```

```
## [1] 69.36267
```

```
(sd_speed <- sd(roller_coasters_speeds$Speed)) # standard deviation
```

```
## [1] 29.32774
```

```
(sem_speed <- sd_speed / nrow(roller_coasters_speeds)) # standard error
```

```
## [1] 0.1086213
```

```
(df_speed <- nrow(roller_coasters_speeds) - 1)
```

```
## [1] 269
```

```
(t_speed <- (point_est_speed-mean_speed) / sem_speed)
```

```
## [1] 5.867482
```

```
(p_val <- 2*(1- pt(t_speed, df = df_speed)))
```

```
## [1] 1.296661e-08
```

We can alsocalculate 95% confidence intervals:

```
#lower limit
# mean - 1.96 * SE
mean_speed + qt(0.025, df = df_speed) * sem_speed
```

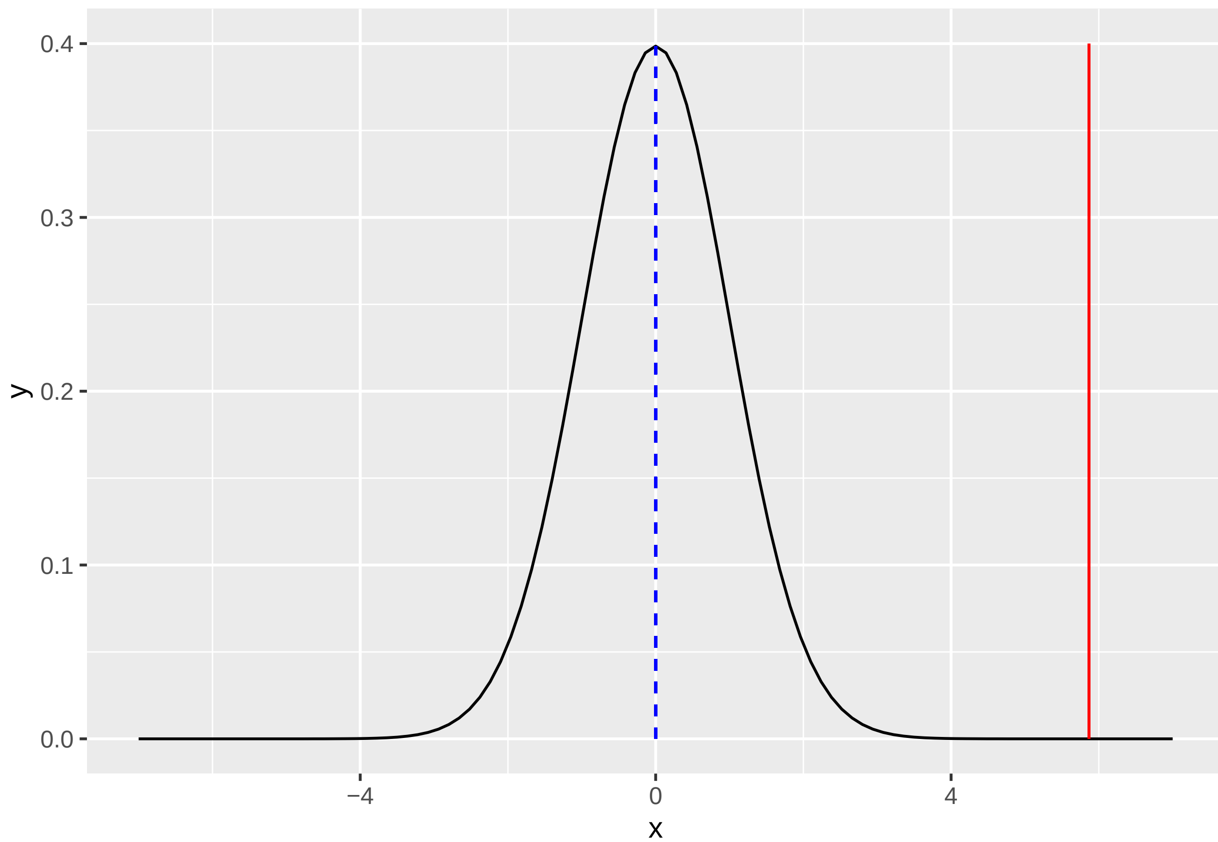
```
## [1] 69.14881
```

```r
#upper limit
# mean + 1.96 * SE
mean_speed + qt(0.975, df = df_speed) * sem_speed
```

```
## [1] 69.57652
```

Finnaly we can plot our discovery:

```r
xframe <- seq(-7, 7, length = 100)
ggplot(data.frame(x = xframe), aes(x = x)) +
  stat_function(fun = dt, args = list(df = df_speed)) +
  geom_segment(aes(x = 0, y = 0, xend = 0, yend = dt(0, df = df_speed)),
               color = 'blue',
               linetype = 'dashed') +
  geom_segment(aes(x = t_speed, y = 0, xend = t_speed, yend = 0.4),
               color = 'red')
```



We reject the null hypothesis in favor of the alternative. Mean roller coaster speed is not 70mph!

Hypothesis 2 - Difference of two means t-test

We want to check if the Wooden roller coasters are on average faster that the Steel ones.

```
roller_coasters_steel <- roller_coasters_raw %>%
  filter(Construction == "Steel" & !is.na(Speed))
roller_coasters_wood <- roller_coasters_raw %>%
  filter(Construction == "Wood" & !is.na(Speed))
```

Check number of instances:

```
nrow(roller_coasters_steel)
```

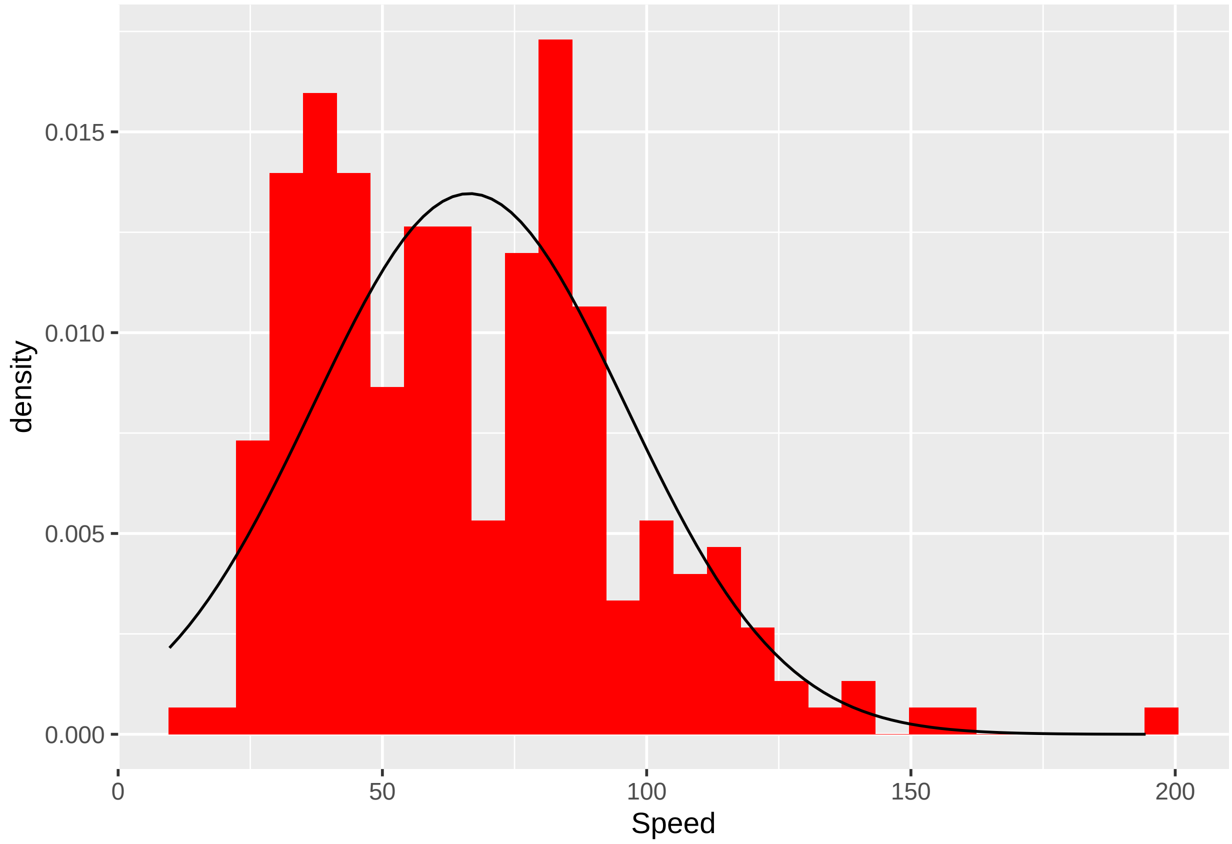
```
## [1] 236
```

```
nrow(roller_coasters_wood)
```

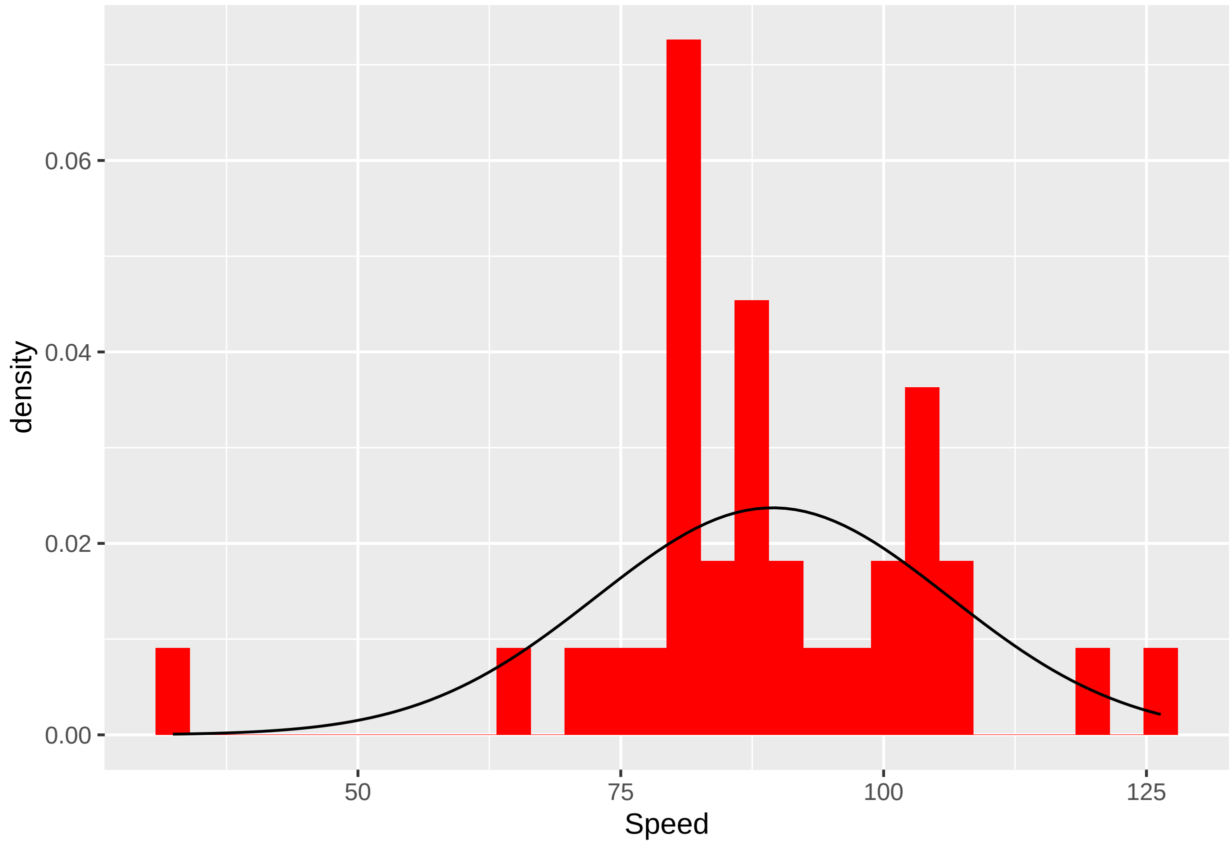
```
## [1] 34
```

Although already proven with symmetry test in the summary statistics, let's have a look at our distribution plots and their skewness:

```
ggplot(roller_coasters_steel) +
  geom_histogram(aes(x = Speed, y = ..density..), fill ='red') +
  stat_function(fun = dnorm, args = list(mean = mean(roller_coasters_steel$Speed), sd = sd(roller_coast
```



```
ggplot(roller_coasters_wood) +
  geom_histogram(aes(x = Speed, y = ..density..), fill ='red') +
  stat_function(fun = dnorm, args = list(mean = mean(roller_coasters_wood$Speed), sd = sd(roller_coaste
```



This is enough to assume we can proceede with our hypothesis testing.

Our hypothesis 2:

$$H_O : mean_{Wood} - mean_{Steel} = 0$$
$$H_A : mean_{Wood} - mean_{Steel} \neq 0$$

$$\alpha = 0.05$$

Calculate necessary variables:

```
(point_est_const <- mean(roller_coasters_wood$Speed) - mean(roller_coasters_steel$Speed))
```

```
## [1] 22.98329
```

```
# (sample_sd <- sd(kiwi_gs_m$height_cm))
(SE <- sqrt((sd(roller_coasters_wood$Speed)^2/nrow(roller_coasters_wood)) + sd(roller_coasters_steel$Spe
```

```
## [1] 3.470155
```

```r
(df <- nrow(roller_coasters_wood) - 1)
```

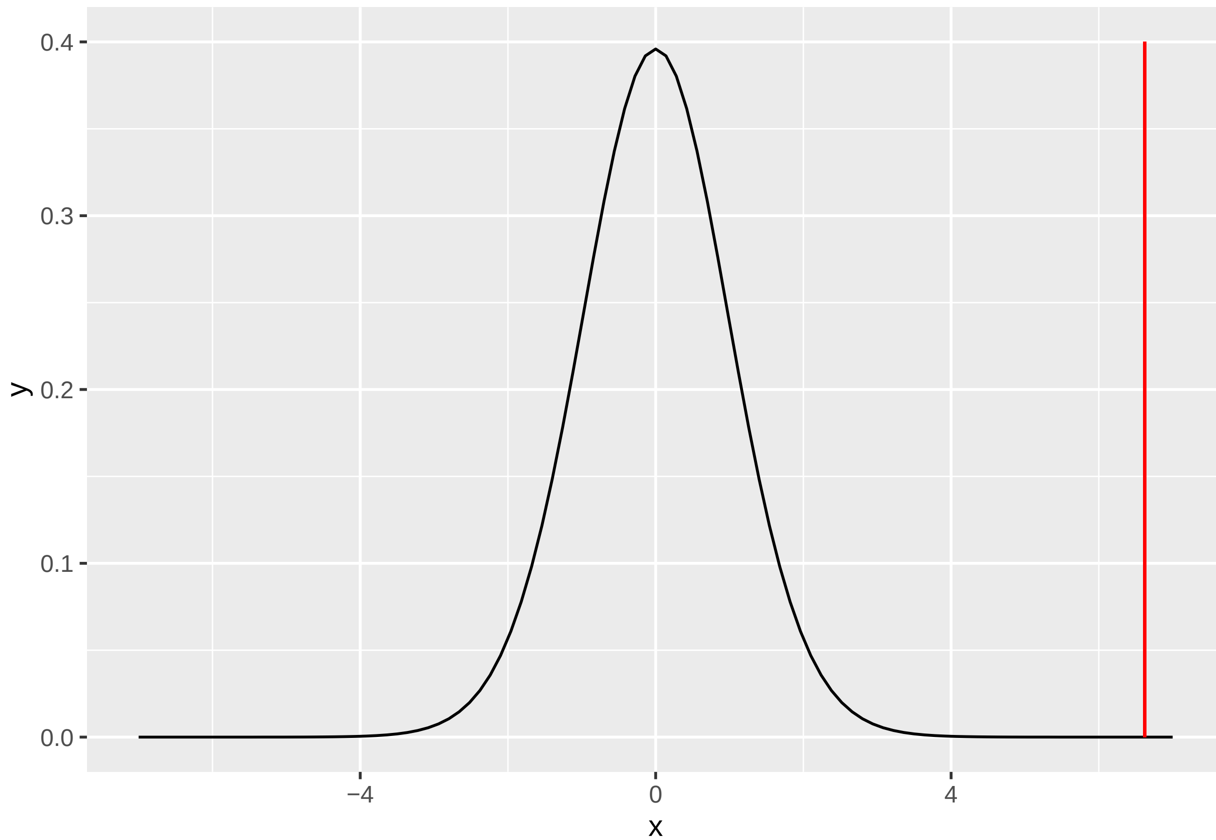
```
## [1] 33
```

```r
(t_stat_const <- (point_est_const - 0) / SE)
```

```
## [1] 6.62313
```

Plot our findings:

```r
ggplot(data.frame(x = seq(-7, 7, length = 100)), aes(x = x)) +
  stat_function(fun = dt, args = list(df = df)) +
  geom_segment(aes(x = t_stat_const, y = 0, xend = t_stat_const, yend = 0.4), color = 'red')
```



p-value:

```r
(p_val <- 2 * (1 - pt(t_stat_const, df)))
```

```
## [1] 1.560164e-07
```

We reject the NULL hypothesis in favour of the alternative. The difference in means is significant and wooden roller coasters go faster on average.

Regression Analysis

Our goal is to make a linear regression model for prediction of coasters Speed attribute.

Correlation Analysis

Let's have a look at the correlations (Pearson) and see which are the best candidates.

```
(cor.test(roller_coasters_raw$Height, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$Height and roller_coasters_raw$Speed
## t = 38.222, df = 256, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9019179 0.9388051
## sample estimates:
##       cor
## 0.9224392
```

```
(cor.test(roller_coasters_raw$Length, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$Length and roller_coasters_raw$Speed
## t = 15.582, df = 258, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6278199 0.7540719
## sample estimates:
##       cor
## 0.6962931
```

```
(cor.test(roller_coasters_raw$Numinversions, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$Numinversions and roller_coasters_raw$Speed
## t = 5.5742, df = 268, p-value = 6.061e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2110692 0.4253337
## sample estimates:
##       cor
## 0.3223236
```

```
(cor.test(roller_coasters_raw$Duration, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$Duration and roller_coasters_raw$Speed
## t = 3.9954, df = 162, p-value = 9.781e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1532868 0.4328823
## sample estimates:
##       cor
## 0.2995011
```

```
(cor.test(roller_coasters_raw$GForce, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$GForce and roller_coasters_raw$Speed
## t = 3.3676, df = 56, p-value = 0.001377
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1701111 0.6045861
## sample estimates:
##       cor
## 0.4103754
```

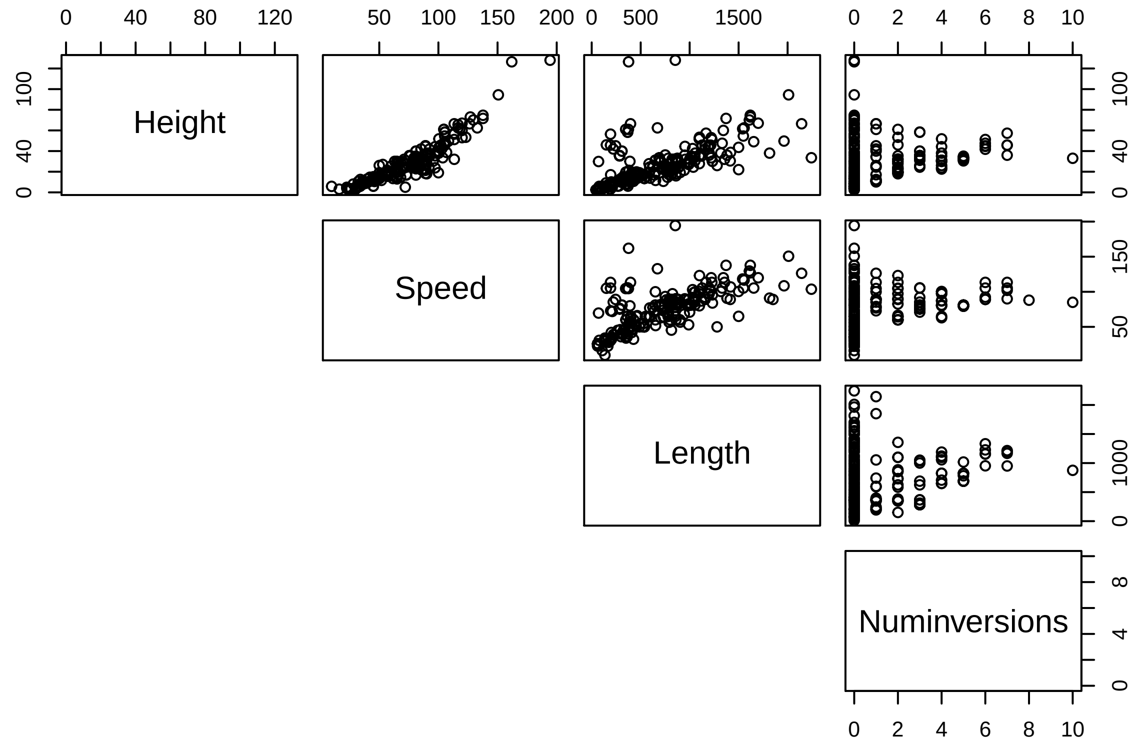
```
(cor.test(roller_coasters_raw$Opened, roller_coasters_raw$Speed))
```

```
##
##  Pearson's product-moment correlation
##
## data:  roller_coasters_raw$Opened and roller_coasters_raw$Speed
## t = 0.26238, df = 260, p-value = 0.7932
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1051251  0.1371870
## sample estimates:
##        cor
## 0.01626982
```

We see that highest correlations to Speed have Height, Length, and Numinversions. We discard GForce because of many missing values.

From the pairplot below, we can observe that there are some linear or non linear relationships between length, height, speed, and numinversions:

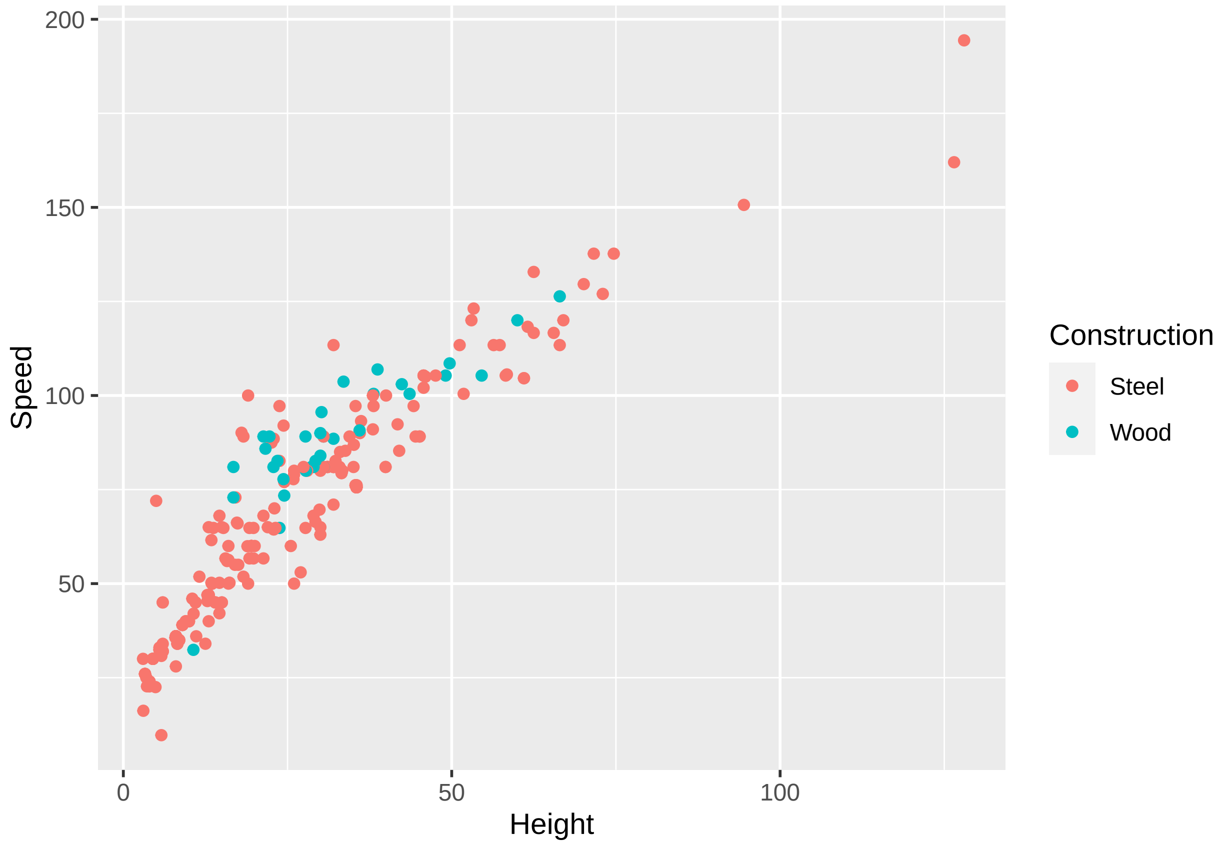
```
pairs(select(roller_coasters_raw, 8:10, 12), lower.panel = NULL)
```



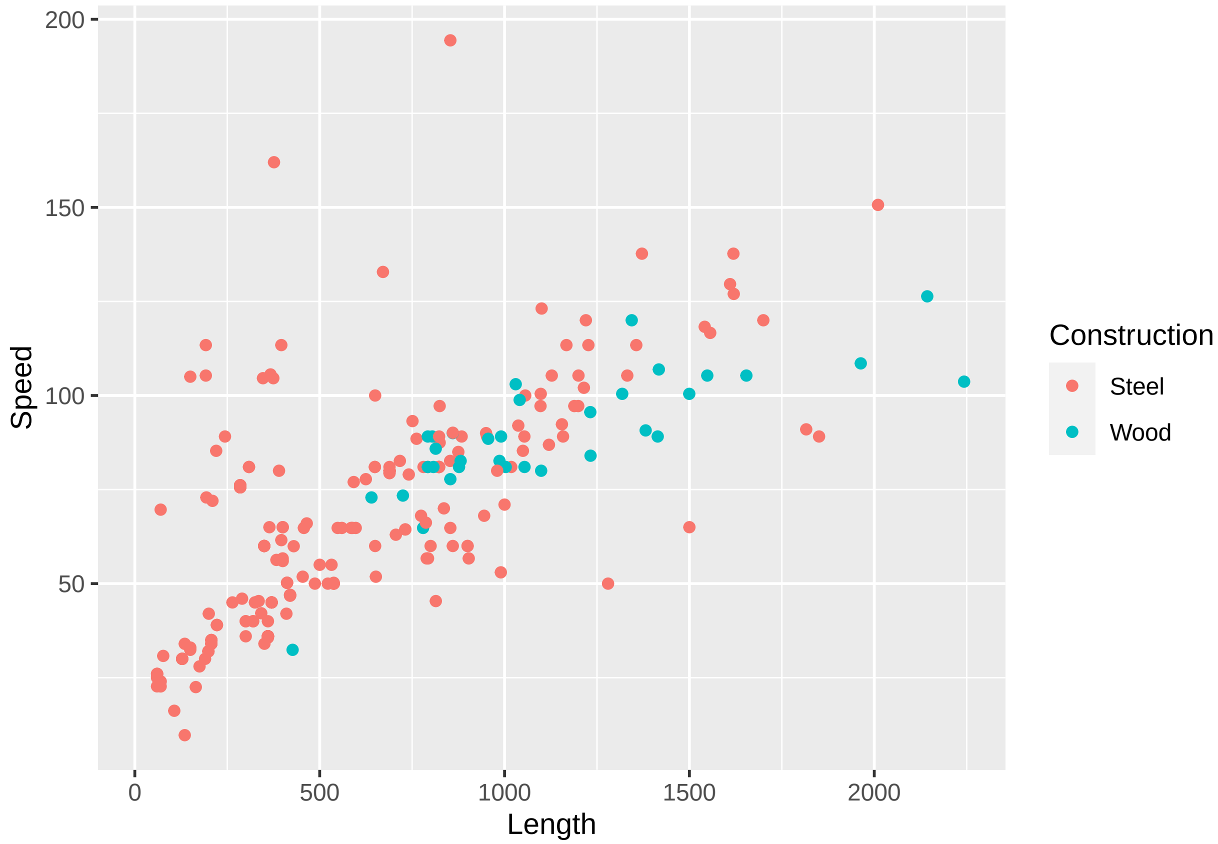
Regressoin Plots

To make sure we get the right attributes for our regression prediction of speed we wanted to take a look at the regression plots:

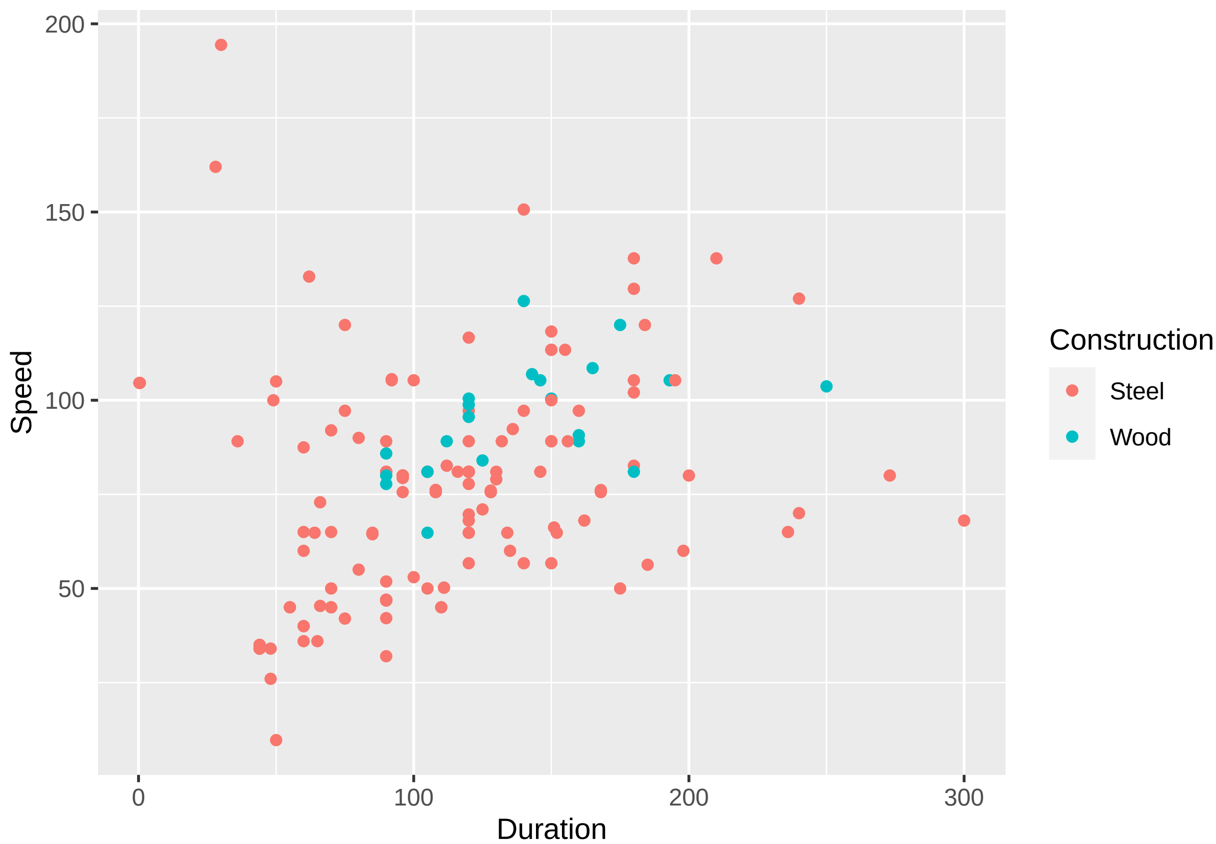
```
roller_coasters_raw %>%
  ggplot() +
    geom_point(aes(x = Height, y = Speed, color = Construction))
```




```
roller_coasters_raw %>%
  ggplot() +
    geom_point(aes(x = Length, y = Speed, color = Construction))
```



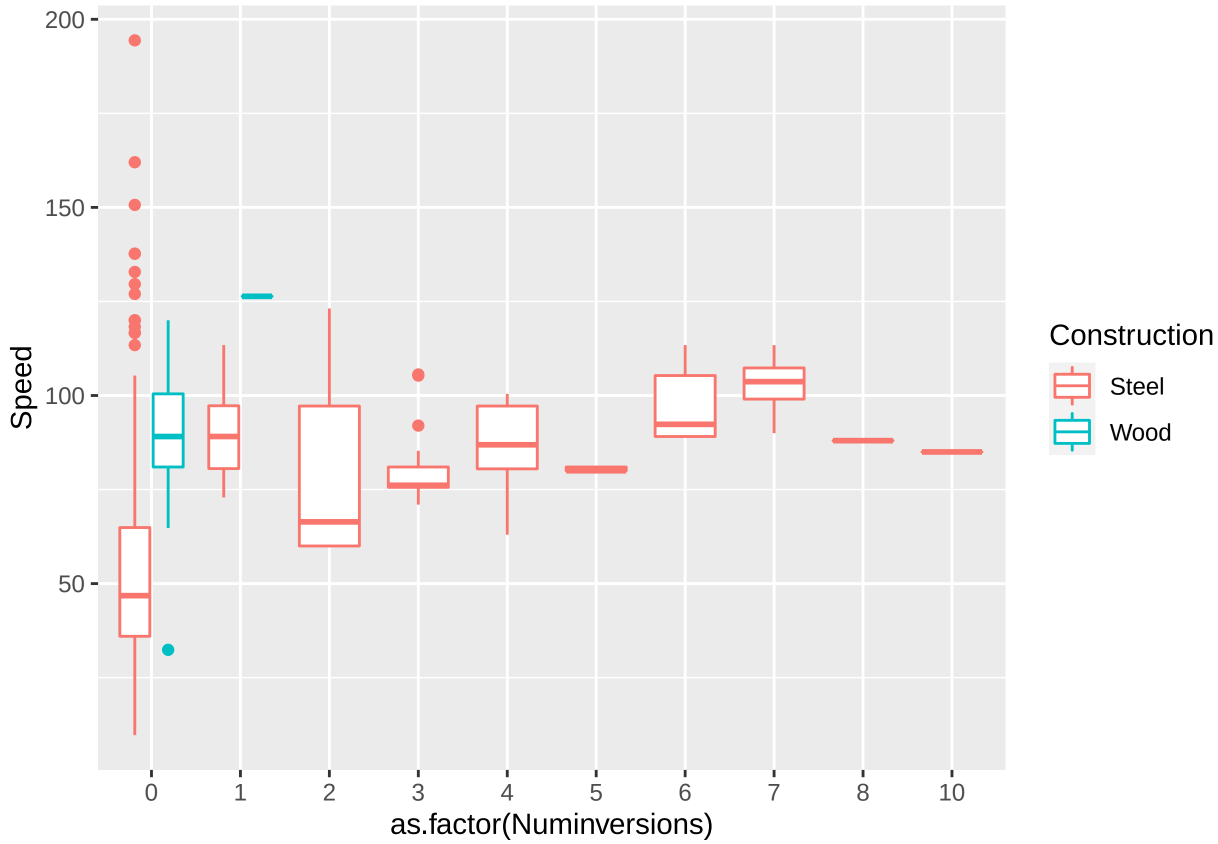

```
roller_coasters_raw %>%
  ggplot() +
    geom_point(aes(x = Duration, y = Speed, color = Construction))
```



We treated Numinversions as a categorical variable since it has too few values to make a proper regression plot.

```
roller_coasters_raw %>%
  ggplot() +
    geom_boxplot(aes(x = as.factor(Numinversions), y = Speed, color = Construction))
```

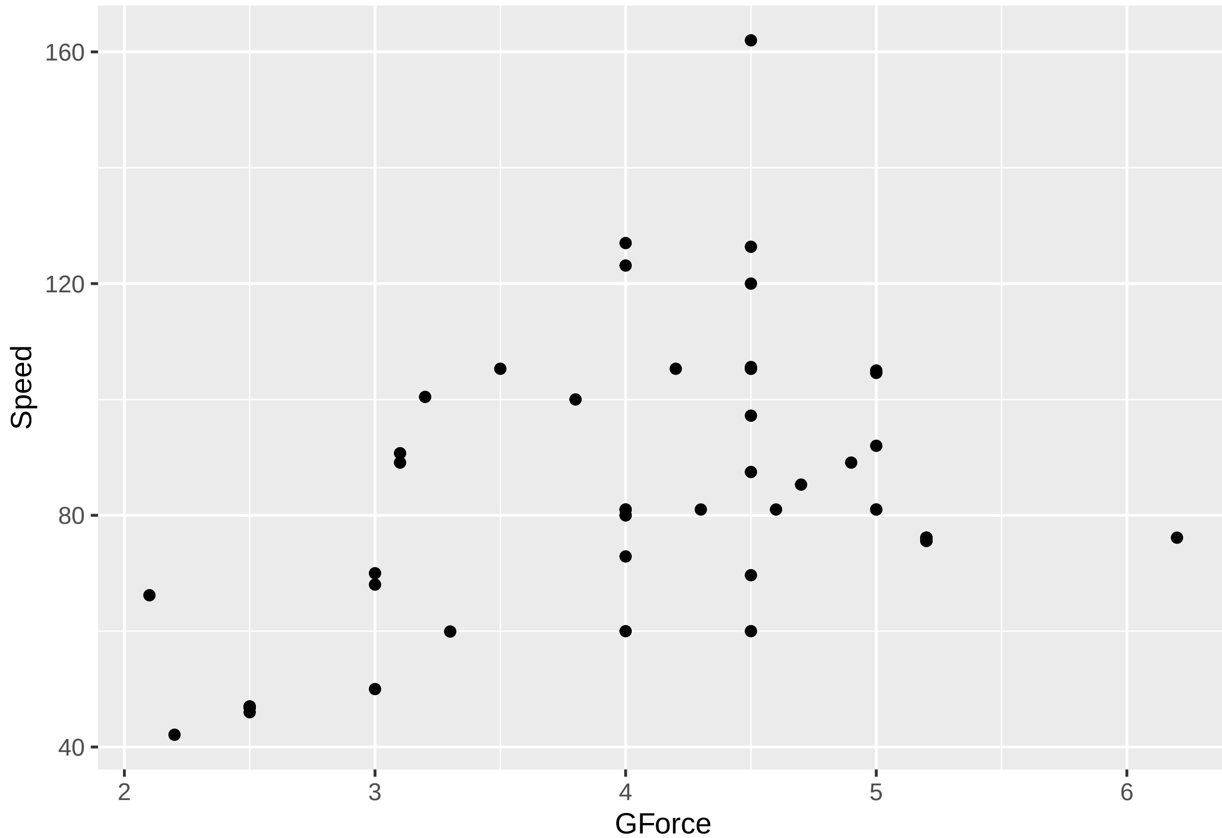
```
## Warning: Removed 138 rows containing non-finite values (stat_boxplot).
```



With every plot above we can see some linearity going on. But the best are definitely height, length, and categorical variable Construction, since the boxplot and hypothesis test clearly showed there is a significant difference between the average speeds.

We also wanted to show that GForce has to few values and not a good linear relationship, so that is why we won't include it into our prediction model:

```
roller_coasters_raw %>%
  filter(!is.na(GForce)) %>%
  ggplot() +
    geom_point(aes(x = GForce, y = Speed))
```



Regression

We prepared a cleaned dataset with only the variables that are going to predict speed.

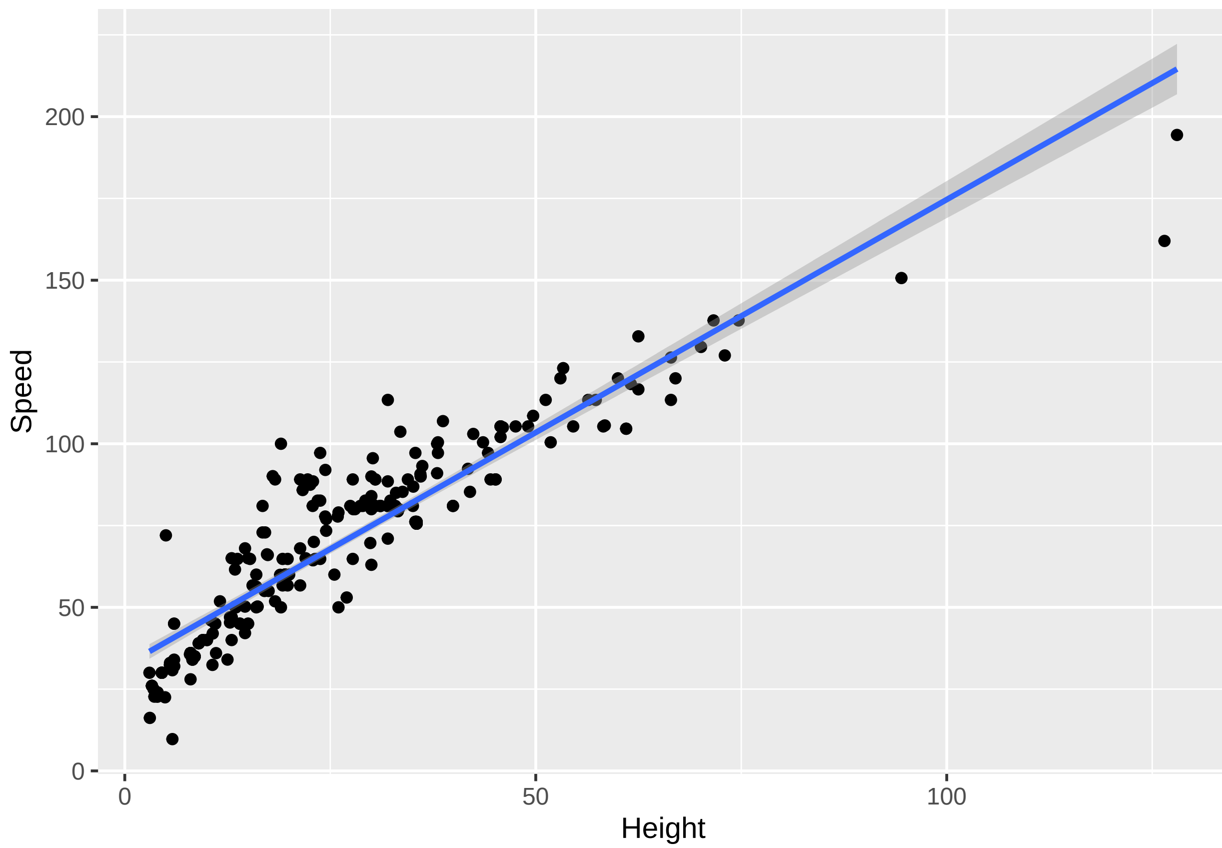
```
roller_coasters <- roller_coasters_raw %>%
  select(Construction, Length, Height, Speed) %>%
  filter(!is.na(Speed) & !is.na(Height) & !is.na(Length)) %>%
  mutate("Steel" = as.numeric(Construction == 'Steel')) %>%
  select(-Construction)
knitr::kable(head(roller_coasters))
```

Length	Height	Speed	Steel
853.440	128.016	194.40	1
376.428	126.492	162.00	1
2010.160	94.488	150.66	1
1619.100	74.676	137.70	1
1620.000	73.000	127.00	1
1371.600	71.628	137.70	1

For linear models we have to take care that the following hold: 1) Linearity of the data 2) Nearly normal residuals (also check for outliers, mostly influential outliers) 3) Constant variability and 4) Independent observations.

We will assume that all the observations are independent.

```r
roller_coasters %>% ggplot()+
  geom_point(aes(x = Height, y = Speed))+
  geom_smooth(aes(x = Height, y = Speed), method = lm)
```




```r
lin_model <- lm(Speed ~ Height, data = roller_coasters)
summary(lin_model)
```

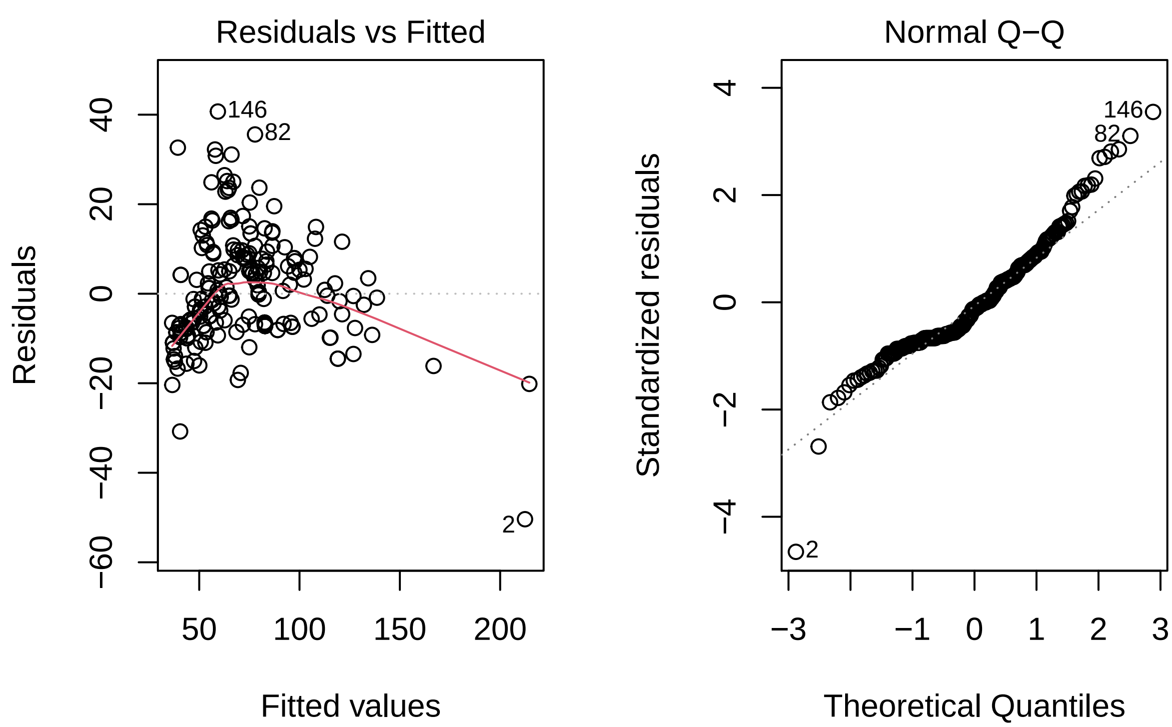
```
##
## Call:
## lm(formula = Speed ~ Height, data = roller_coasters)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.375  -7.634  -1.459   6.163  40.701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.2415     1.2231   26.36   <2e-16 ***
## Height        1.4241     0.0377   37.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.49 on 250 degrees of freedom
```

```
## Multiple R-squared:  0.8509, Adjusted R-squared:  0.8503
## F-statistic:  1427 on 1 and 250 DF,  p-value: < 2.2e-16
```

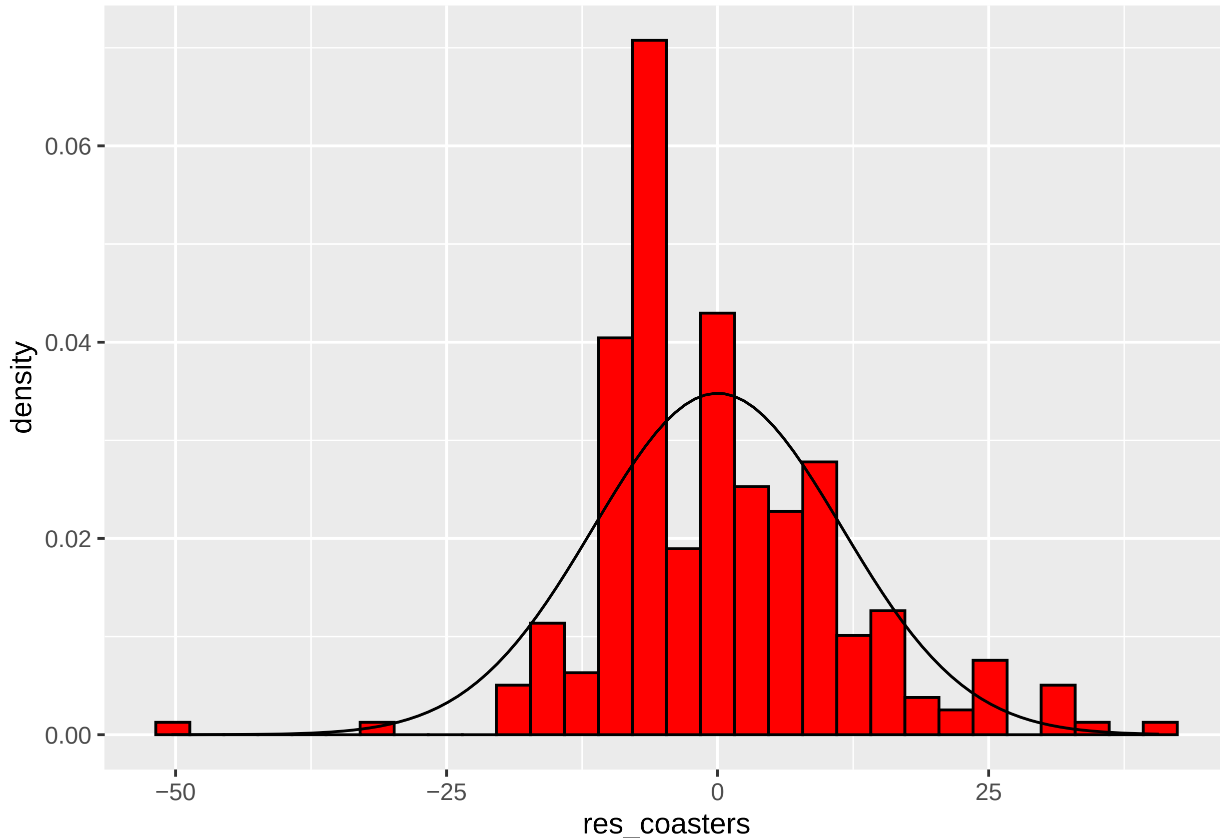
```
coef(lin_model)
```

```
## (Intercept)      Height
##   32.241543    1.424073
```

```
par(mfrow=c(1,2))
plot(lin_model, which = 1:2)
```




```
res_coasters <- residuals(lin_model)
roller_coasters %>%
  ggplot() +
  geom_histogram(aes(x = res_coasters, y = ..density..), fill = "red", color = "black") +
  stat_function(fun = dnorm, args = list(mean = mean(res_coasters), sd = sd(res_coasters)))
```



From the above linear regression analysis and plots we are able to see that there is indeed a linearity (as p values show). But there are some influential outliers! Not to forget, the residuals also don't have a quite constant varibility. Although the model has a high R^2 we need to be careful when using this model since it does not completely meet the requirements of linear regression analysis.

We also made a multiple regression model using the most important features: height, steel and length.

```
rc_all <- lm(Speed ~ ., data = roller_coasters)
summary(rc_all)
```

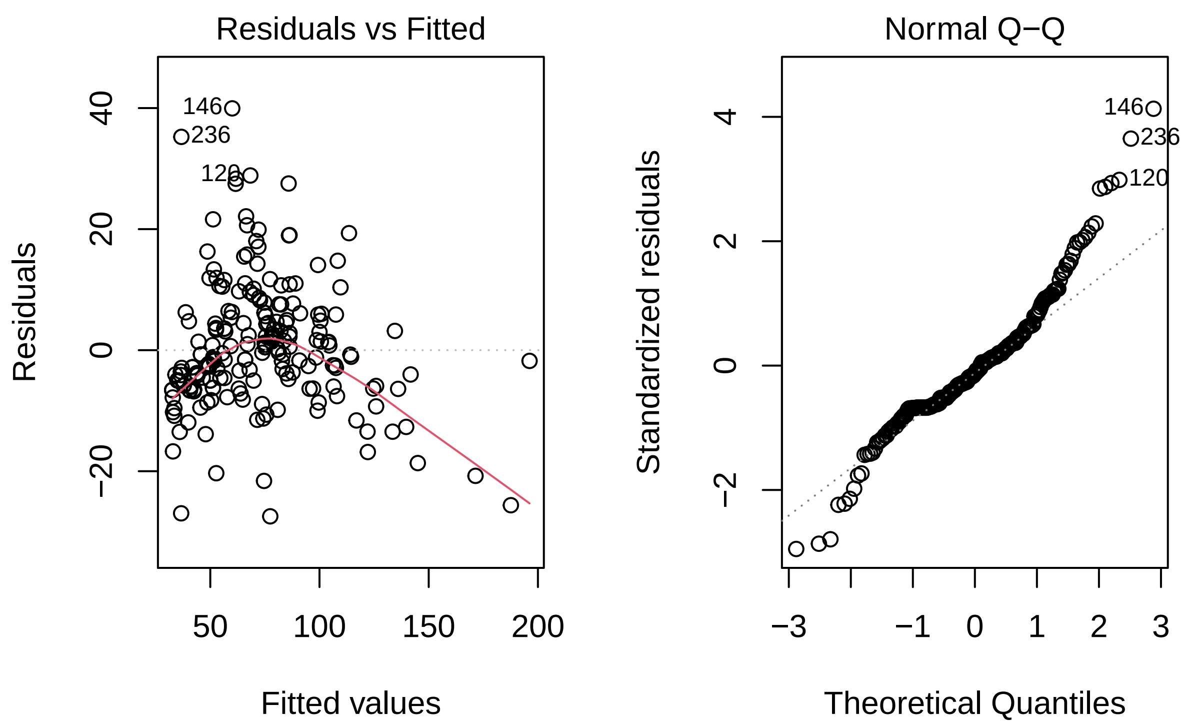
```
##
## Call:
## lm(formula = Speed ~ ., data = roller_coasters)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.452  -6.131  -1.180   3.826  39.948
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.699899   2.434493  13.843  < 2e-16 ***
## Length       0.014037   0.001915   7.329 3.26e-12 ***
## Height       1.222438   0.039889  30.646  < 2e-16 ***
## Steel       -5.998684   2.046036  -2.932  0.00368 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.701 on 248 degrees of freedom
## Multiple R-squared:  0.8946, Adjusted R-squared:  0.8933
## F-statistic: 701.3 on 3 and 248 DF,  p-value: < 2.2e-16
```

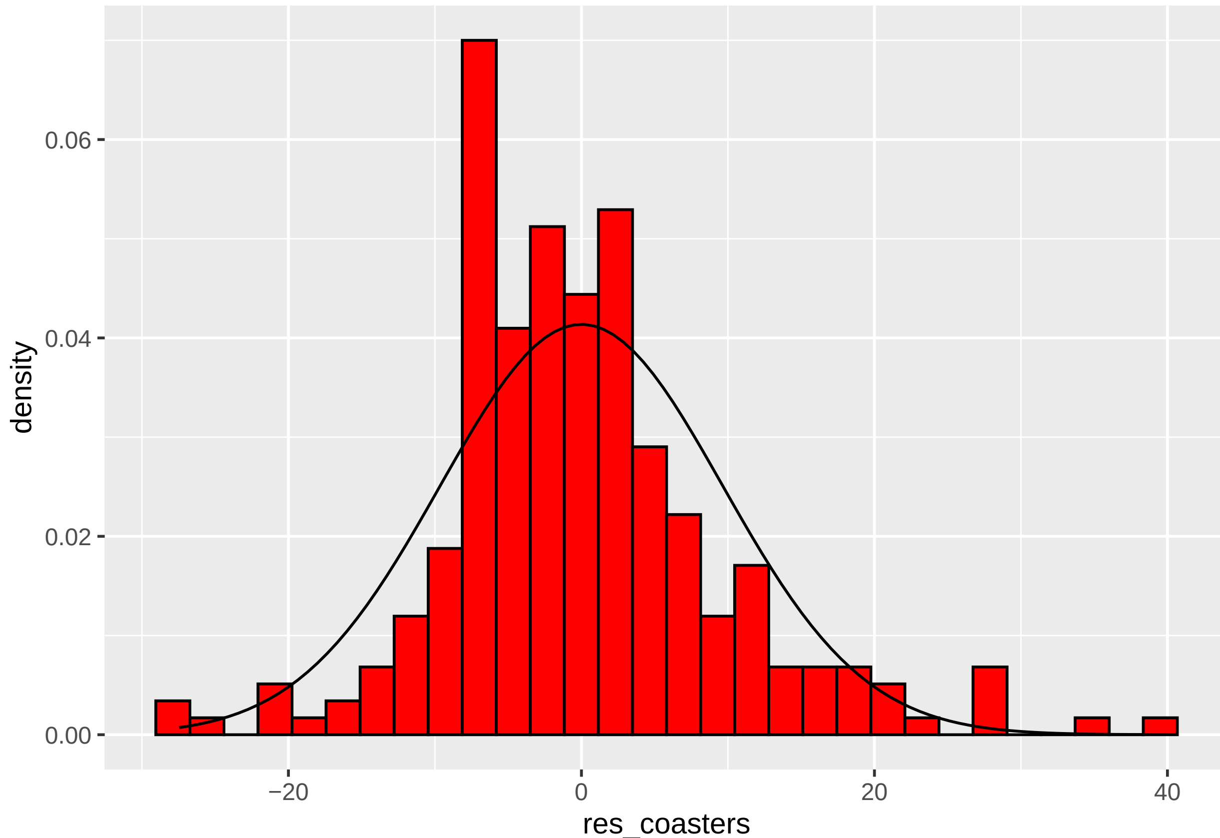
```
coef(rc_all)
```

```
## (Intercept)      Length      Height       Steel
## 33.69989868  0.01403739  1.22243835 -5.99868362
```

```
par(mfrow=c(1,2))
plot(rc_all, which = 1:2)
```




```
res_coasters <- residuals(rc_all)
roller_coasters %>%
  ggplot() +
  geom_histogram(aes(x = res_coasters, y = ..density..), fill = "red", color = "black") +
  stat_function(fun = dnorm, args = list(mean = mean(res_coasters), sd = sd(res_coasters)))
```



Similarly as in the one regression analysis we assumed the data is independent. The R^2 is even higher with all significant attributes (as the p values show), so there is definitely a linearity. The problem again is that we have some influential outliers and the variability is not constant! So again, as in the first regression we are allowed to use this but we need to be careful, as the model does not meet all the requirements for linear regression analysis.

Red billed seagulls

The dataset seagulls.csv represents the data collected about seagulls in Auckland, New Zeland. Dataset can be found

$$[here]$$

(https://grapher.jake4maths.com/?folder=sneddon&dataset=GULLS.csv).

Data was collected on two seperate occasions (summer and winter) and on four different locations: Muriwai (a), Piha (b), Mareatai (c), and Waitawa (d).

They collected seagull's weight, length, and sex, as well as its location and season. Authors of the dataset also point out that none of the locations is a major breeding site.

We also cleaned dataset a bit. Some cases have misspelled "MURIWAI" as "MURWAI". Variables location, coast, season, and sex have been converted from strings to factors, and length was renamed to height, since that is more accurate varable description.

```
seagulls <- read.csv("datasets/seagulls.csv")
seagulls[seagulls$LOCATION == "MURWAI",]$LOCATION <- "MURIWAI"
colnames(seagulls)[2] <- "HEIGHT"
```

Figure 1: Auckland region

```
seagulls$LOCATION <- as.factor(seagulls$LOCATION)
seagulls$COAST <- as.factor(seagulls$COAST)
seagulls$SEASON <- as.factor(seagulls$SEASON)
seagulls$SEX <- as.factor(seagulls$SEX)
```

WEIGHT	HEIGHT	LOCATION	COAST	SEASON	SEX
262	38.9	MARAETAI	EAST	WINTER	MALE
300	41.3	MURIWAI	WEST	SUMMER	MALE
250	36.6	MURIWAI	WEST	WINTER	MALE
242	36.0	MARAETAI	EAST	WINTER	FEMALE
261	37.1	MURIWAI	WEST	WINTER	MALE
262	38.2	MURIWAI	WEST	WINTER	MALE

Summary statistics

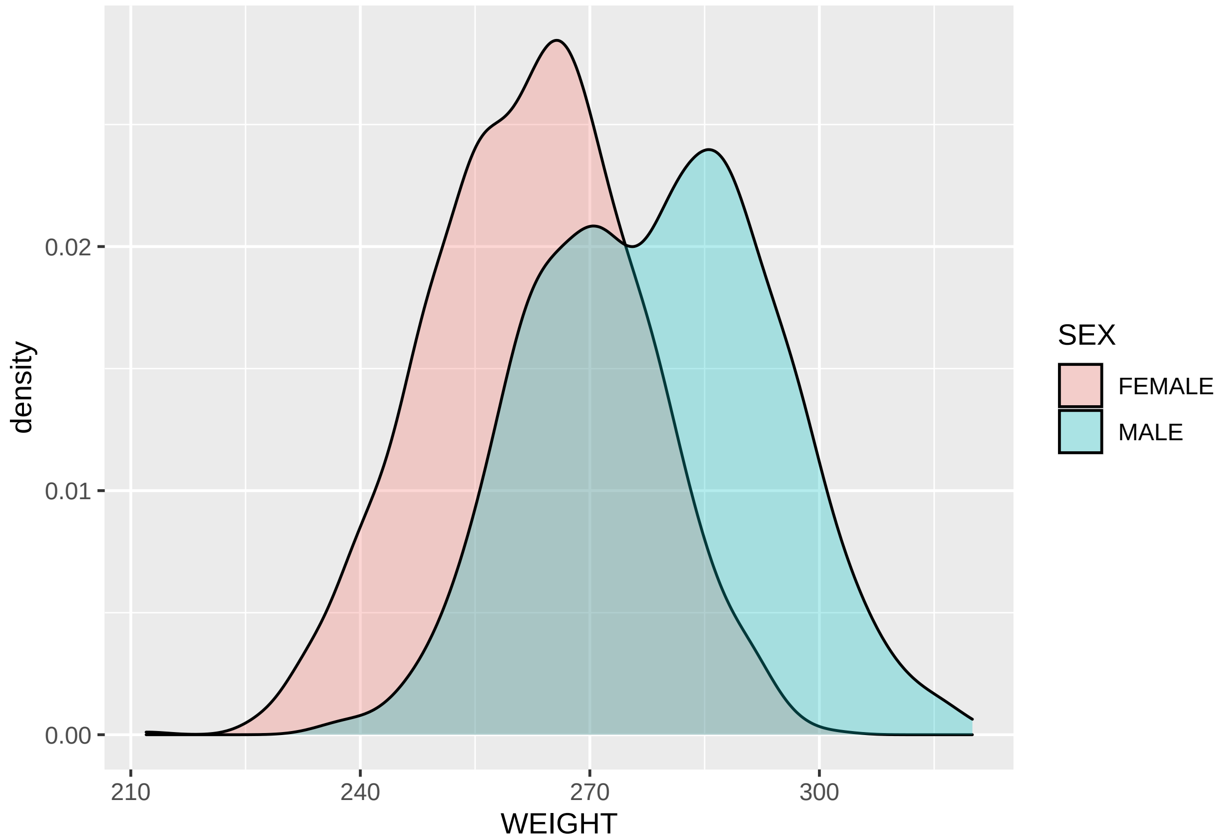
Seagulls dataset has 2487 cases and 6 variables: weight, height, location, coast, season, and sex. Weight and length are numerical, while location, coast, season, and sex are categorical.

```
knitr::kable(summary(seagulls))
```

WEIGHT	HEIGHT	LOCATION	COAST	SEASON	SEX
Min. :212.0	Min. :28.5	MARAETAI:673	EAST:1251	SUMMER:1313	FEMALE:1280
1st Qu.:259.0	1st Qu.:35.5	MURIWAI :589	WEST:1236	WINTER:1174	MALE :1207
Median :269.0	Median :37.1	PIHA :647	NA	NA	NA
Mean :270.4	Mean :37.1	WAITAWA :578	NA	NA	NA
3rd Qu.:282.0	3rd Qu.:38.8	NA	NA	NA	NA
Max. :320.0	Max. :44.8	NA	NA	NA	NA

Weight of seagulls is in grams (g), and its distribution can be seen here:

```
seagulls %>% ggplot()+
  geom_density(aes(x = WEIGHT, fill = SEX), alpha = 0.3)
```



Average weight of males is 278.73g with minimum of 235g and maximum of 320g. Average weight of females is 262.49g with minimum of 212g and maximum of 302g. We can see that weights of males are not normally distributed, while weights of females could be. We can check this with Shapiro test:

```
shapiro.test(seagulls[seagulls$SEX == "MALE",]$WEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  seagulls[seagulls$SEX == "MALE", ]$WEIGHT
## W = 0.994, p-value = 8.841e-05
```

```
shapiro.test(seagulls[seagulls$SEX == "FEMALE",]$WEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  seagulls[seagulls$SEX == "FEMALE", ]$WEIGHT
## W = 0.99724, p-value = 0.02575
```

We can see that weight is not normally distributed neither for males nor females, but latter are very close to passing the Shapiro test. We can also check if the distributions are at least symmetric:

```r
symmetry.test(seagulls[seagulls$SEX == "MALE",]$WEIGHT)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  seagulls[seagulls$SEX == "MALE", ]$WEIGHT
## Test statistic = -0.80072, p-value = 0.478
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                 954
```

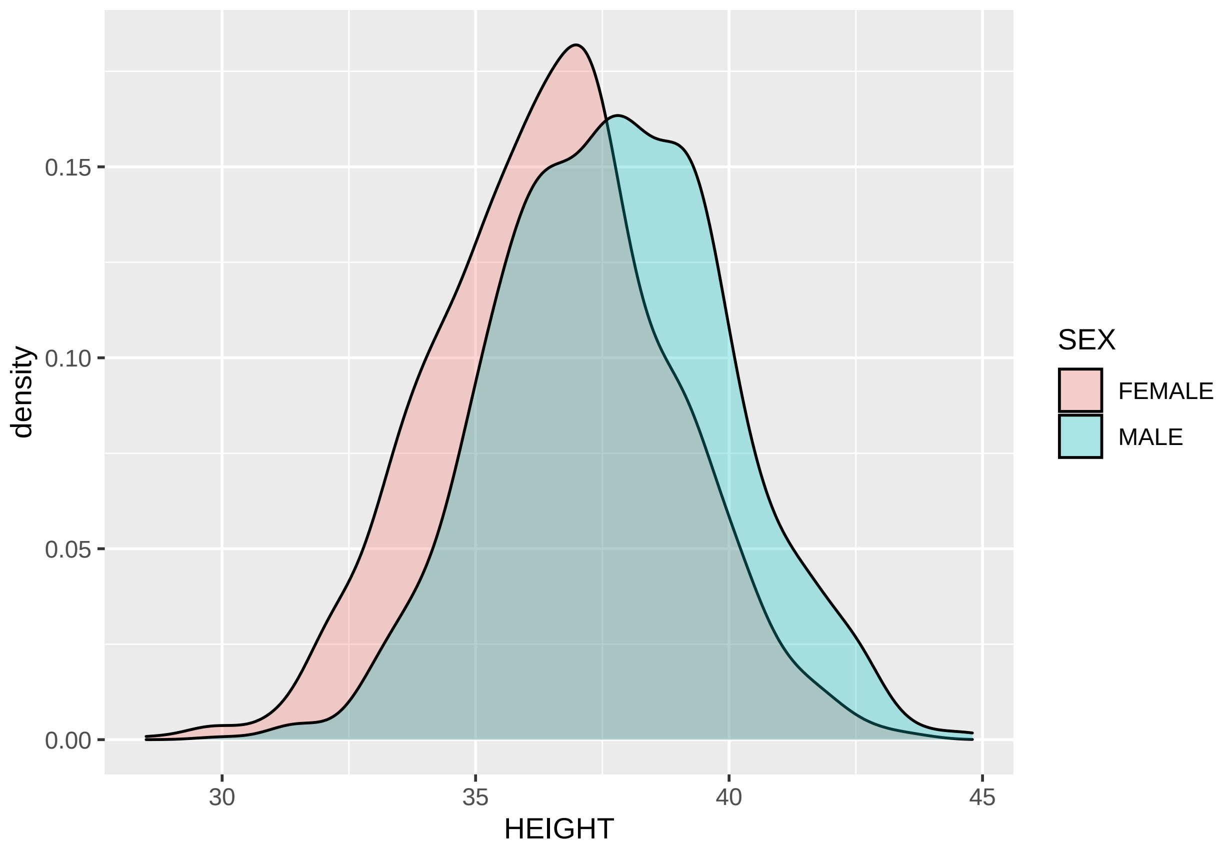
```r
symmetry.test(seagulls[seagulls$SEX == "FEMALE",]$WEIGHT)
```

```
##
##  m-out-of-n bootstrap symmetry test by Miao, Gel, and Gastwirth (2006)
##
## data:  seagulls[seagulls$SEX == "FEMALE", ]$WEIGHT
## Test statistic = -1.773, p-value = 0.09
## alternative hypothesis: the distribution is asymmetric.
## sample estimates:
## bootstrap optimal m
##                 246
```

Both pass symmetry test, meaning they are not strongly skewed and can be used later for inference.

Height of seagulls is in centimeters (cm):

```r
seagulls %>% ggplot()+
  geom_density(aes(x = HEIGHT, fill = SEX), alpha = 0.3)
```



Average height of males is 37.74cm. Smallest male's height is 30cm, while largest is 44.8cm. Female's average height is 36.5cm with minimum of 28.5cm and maximum of 43.7cm. Seagulls height seems more normally distributed than weight, but we can check:

```
shapiro.test(seagulls[seagulls$SEX == "MALE",]$HEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  seagulls[seagulls$SEX == "MALE", ]$HEIGHT
## W = 0.9983, p-value = 0.2733
```

```
shapiro.test(seagulls[seagulls$SEX == "FEMALE",]$HEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  seagulls[seagulls$SEX == "FEMALE", ]$HEIGHT
## W = 0.9989, p-value = 0.6345
```

We can see that height for both sexes passes as normally distributed.

We have four locations in our dataset: Maraetai, Waitawa, Muriwai, and Piha. Coast is either east or west and is a more broad description of location (Maraetai and Waitawa are under east coast and Muriwai and Piha are under west coast). Locations are almost equally represented in our dataset:

```r
table(seagulls$LOCATION) / nrow(seagulls)
```

```
##
##  MARAETAI   MURIWAI      PIHA   WAITAWA
## 0.2706072 0.2368315 0.2601528 0.2324085
```

Coast variable is also equaly distributed:

```r
table(seagulls$COAST) / nrow(seagulls)
```

```
##
##      EAST      WEST
## 0.5030157 0.4969843
```

Season is either winter or summer. There are a little more entries for summer than for winter, but the difference is miniscule:

```r
table(seagulls$SEASON) / nrow(seagulls)
```

```
##
##    SUMMER    WINTER
## 0.5279453 0.4720547
```

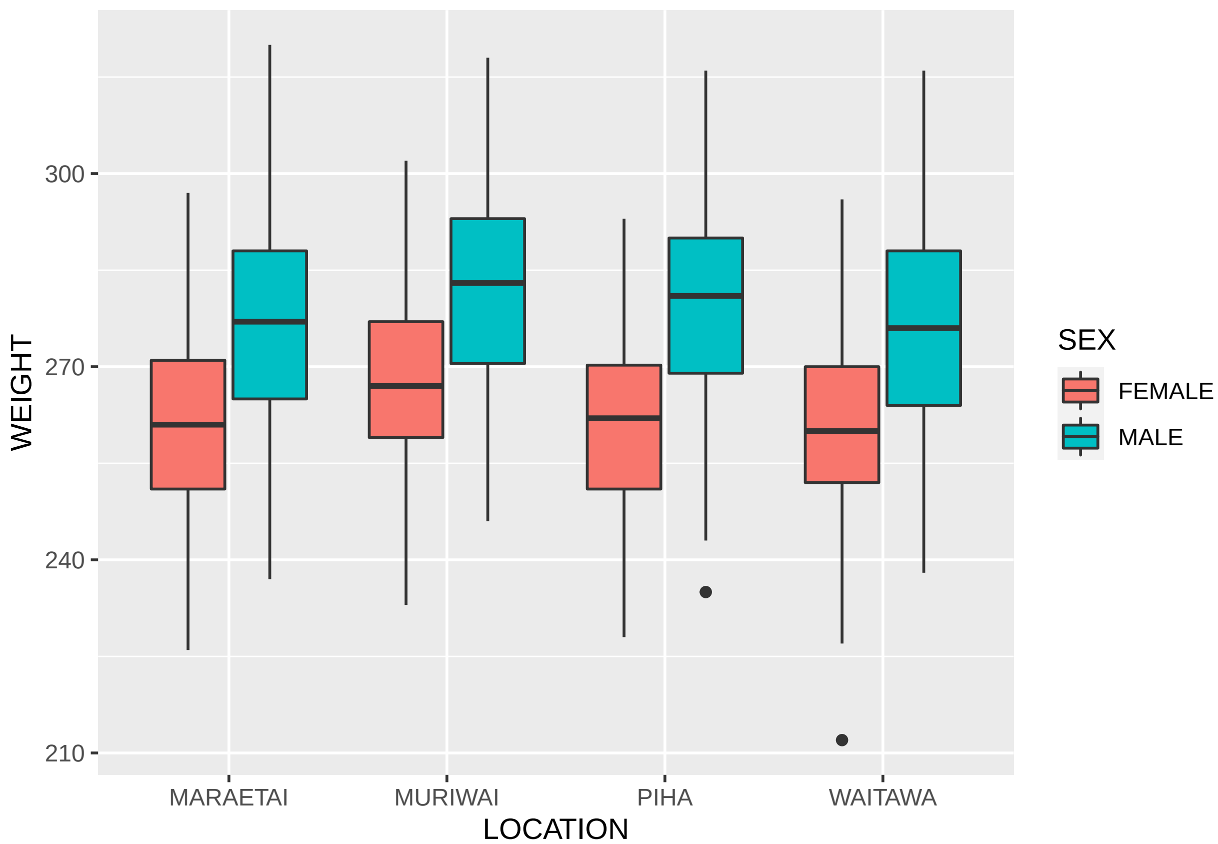
There are more females presented in our dataset but the difference can be ignored:

```r
table(seagulls$SEX) / nrow(seagulls)
```

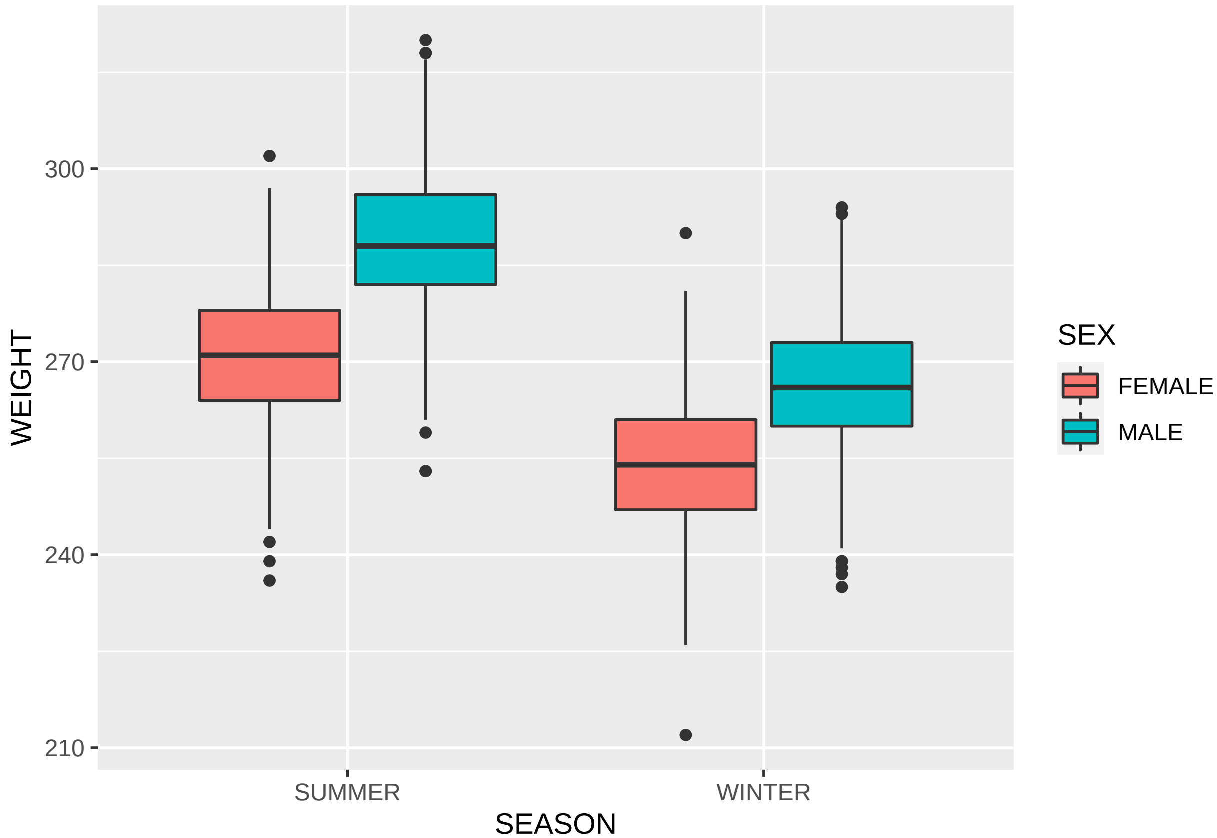
```
##
##    FEMALE      MALE
## 0.5146763 0.4853237
```

We also drew some other plots representing how different variables are connected:

```r
seagulls %>% ggplot()+
  geom_boxplot(aes(x = LOCATION, y = WEIGHT, fill = SEX))
```




```
seagulls %>% ggplot()+
  geom_boxplot(aes(x = SEASON, y = WEIGHT, fill = SEX))
```



Inference

Since we can divide our datasets in many ways, we can also check many different hypothesis.

Is weight of males same on east and west coast?

We want to know if there is a difference between the males in east and west coast.

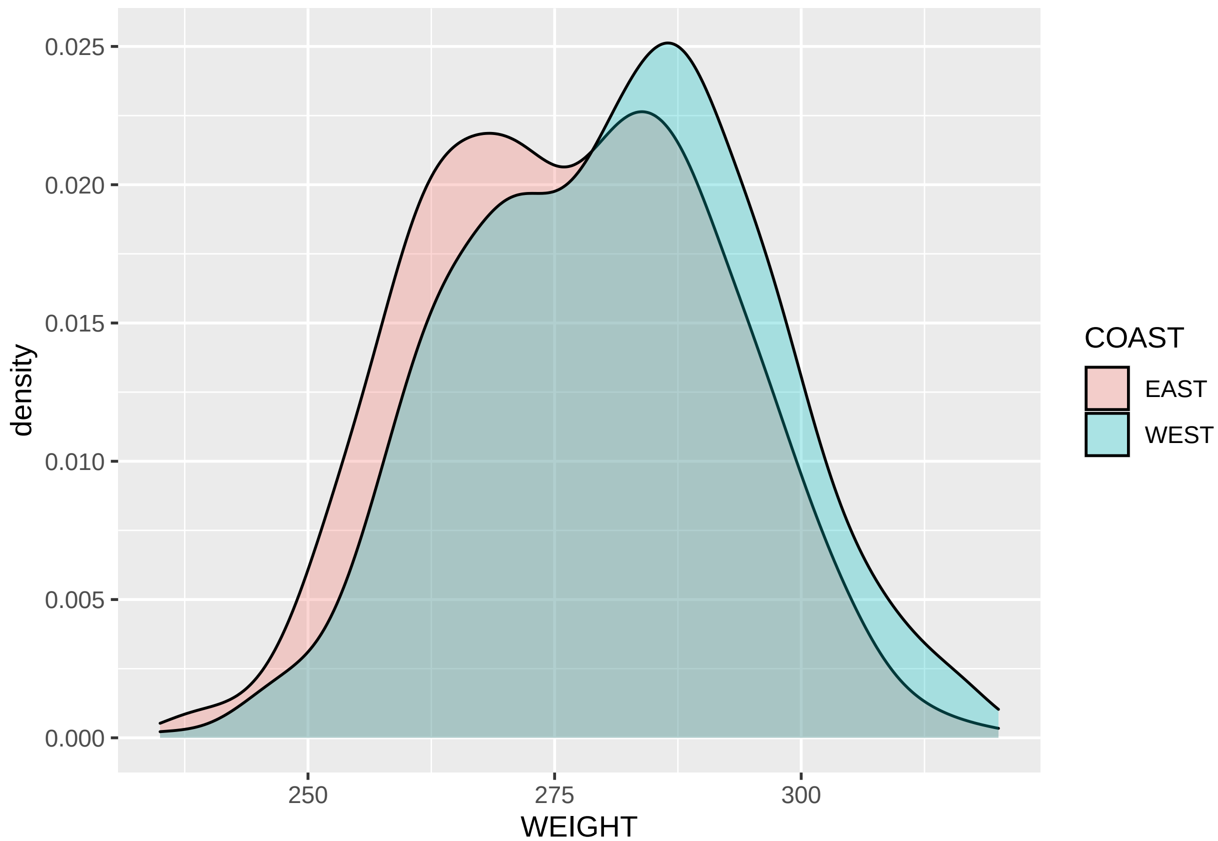
$$H_0 : mean_{east} - mean_{west} = 0$$
$$H_A : mean_{east} - mean_{west} \neq 0$$

We first divide our dataset into two smaller ones, which represent males from different coasts.

```
sg_east <- seagulls %>% filter(COAST == "EAST", SEX == "MALE")
sg_west <- seagulls %>% filter(COAST == "WEST", SEX == "MALE")
```

Next we need to check CLT conditions. Since samples were collected independently from one another, first condition is true. Next we need to check if both samples have sufficient size. There are 629 males from east and 578 males from west. Both samples are larger than 30, so second condition is also true. Then we need to check if any of samples is skewed. We can draw their distributions and see that they both are somewhat symmetrical.

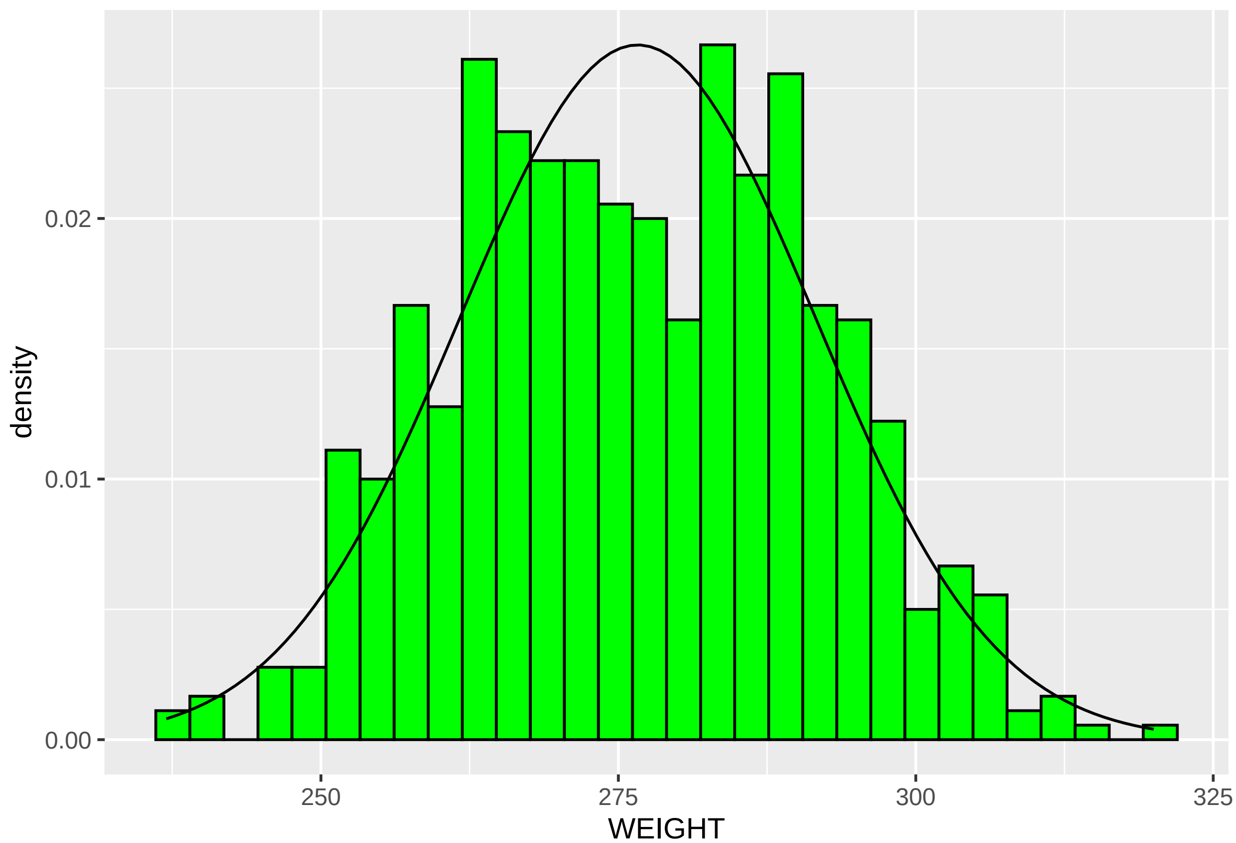
```r
seagulls %>% filter(SEX == "MALE") %>% ggplot()+
  geom_density(aes(x = WEIGHT, fill = COAST), alpha = 0.3)
```



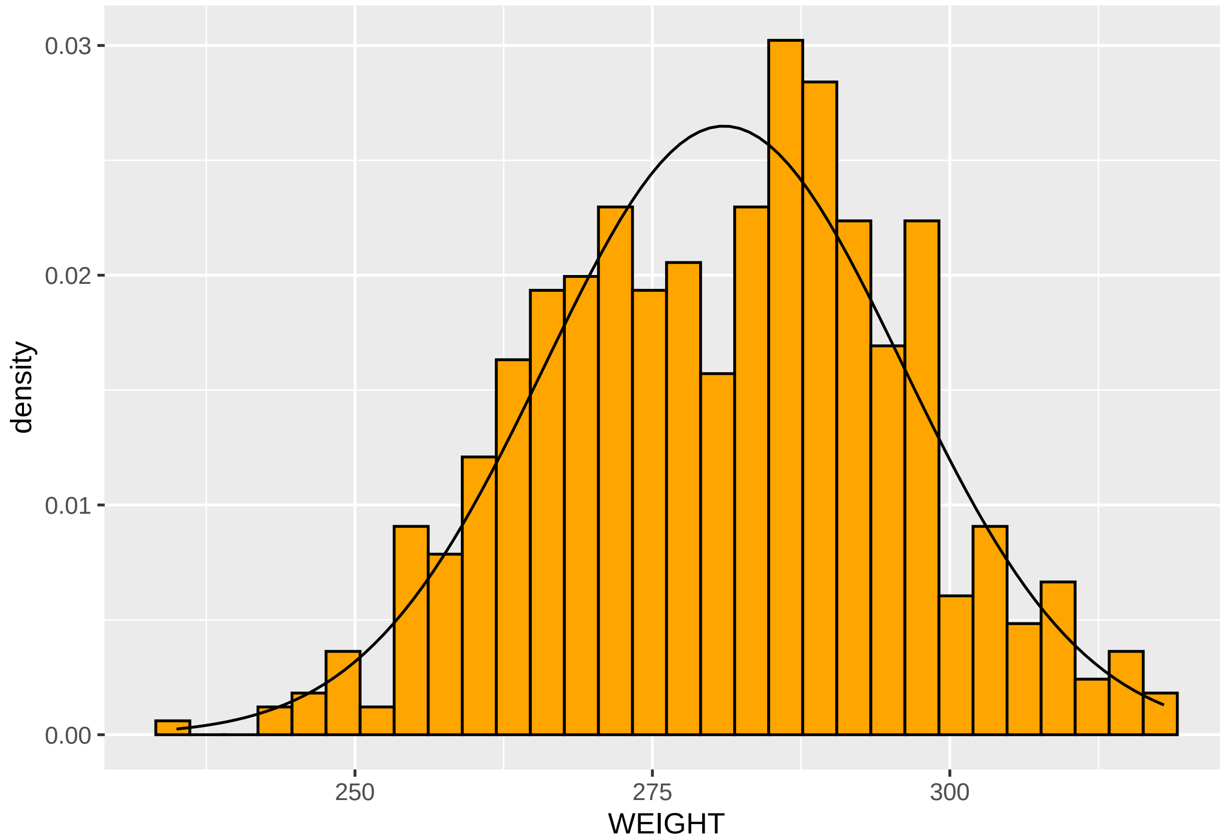
We can also calculate skewness of both distributions. Weight of males from east coast have skewness of 0.0370527 and males from west have skewness of -0.0483849. Both values are small, so we can safely say that neither distribution is strongly skewed.

We also need to check wether cases from groups are independant from each other. Since they were collected on different locations, they are independant. We can check if both groups are normally distributed. For that we can draw histogram of weights and overlay it with normal distribution with same average and standard deviation:

```r
east.mean <- mean(sg_east$WEIGHT)
east.sd <- sd(sg_east$WEIGHT)
sg_east %>% ggplot()+
  geom_histogram(aes(x = WEIGHT, y = ..density..), fill = "green", color = "black")+
  stat_function(fun = dnorm, args = list(mean = east.mean, sd = east.sd))
```




```r
west.mean <- mean(sg_west$WEIGHT)
west.sd <- sd(sg_west$WEIGHT)
sg_west %>% ggplot()+
  geom_histogram(aes(x = WEIGHT, y = ..density..), fill = "orange", color = "black")+
  stat_function(fun = dnorm, args = list(mean = west.mean, sd = west.sd))
```



Neither distribution seem normally distributed. We can further test that hypothesis with normallity test:

```
shapiro.test(sg_east$WEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sg_east$WEIGHT
## W = 0.99247, p-value = 0.002899
```

```
shapiro.test(sg_west$WEIGHT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sg_west$WEIGHT
## W = 0.99417, p-value = 0.0257
```

Neither group has normal distribution, but they are symmetrical, so we will continue with our hypothesis testing.

We set a threshold value $\alpha = 0.05$.

We calculate our point estimate, standard error, and t-score and plot it:

```r
(point_estimate <- east.mean - west.mean)
```

```
## [1] -4.38067
```

```r
(SE <- sqrt(east.sd ^ 2 / nrow(sg_east) + west.sd ^ 2 / nrow(sg_west)))
```

```
## [1] 0.8650424
```

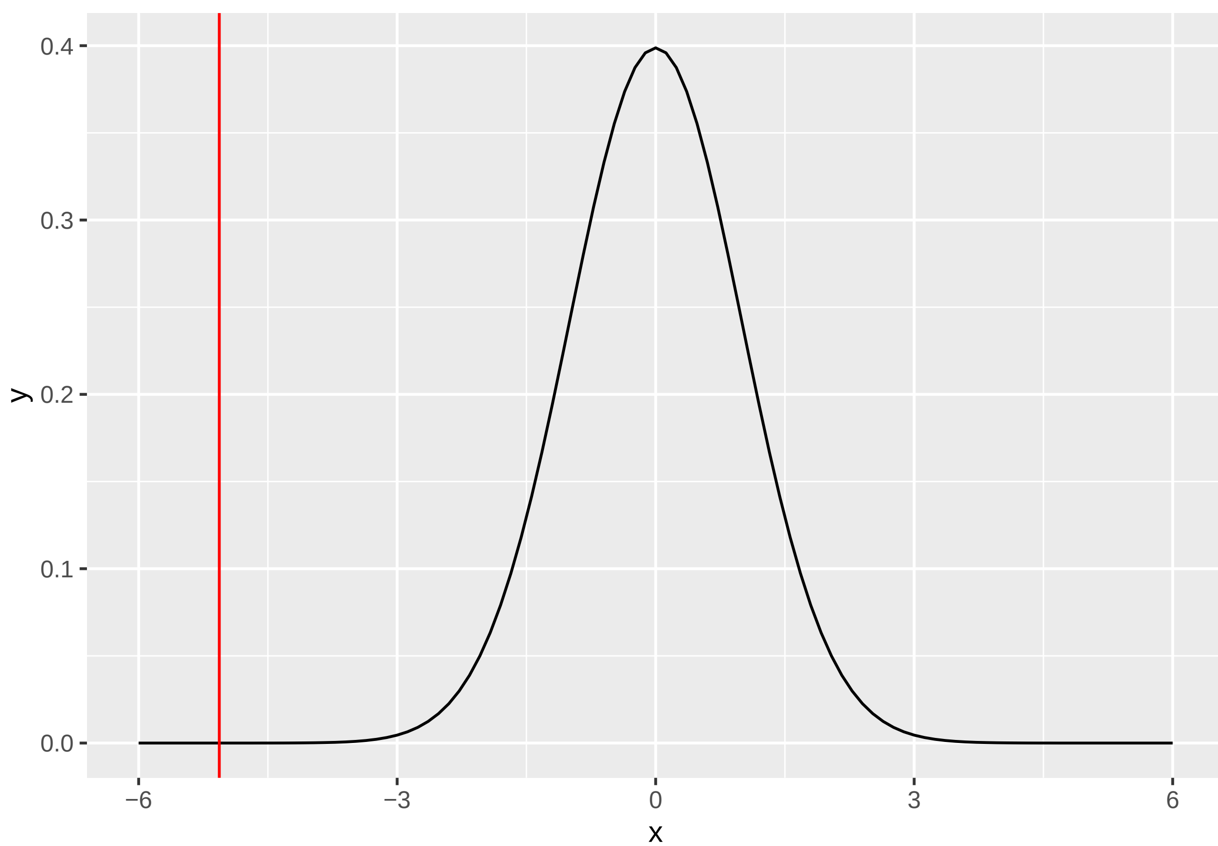
```r
(df <- min(nrow(sg_east) - 1, nrow(sg_west) - 1))
```

```
## [1] 577
```

```r
(t_score <- point_estimate / SE)
```

```
## [1] -5.06411
```

```r
ggplot(data.frame(x = seq(-6, 6, length = 200)), aes(x = x))+
  stat_function(fun = dt, args = list(df = df))+
  geom_vline(xintercept = t_score, color = "red")
```



We can see that our t-score (red line) falls to the left of student's t-distribution, so our null hypothesis is very likely false. We can further confirm that with our p-value calculation:

```r
(p_value <- 2 * pt(t_score, df))
```

```
## [1] 5.528673e-07
```

Since p-value is smaller than α ($5.5286726 \times 10^{-7} < 0.05$), we reject H_0 in favor of H_A. Seagulls on east and west coast do not weight the same. Because our point estimate is negative, we can say that seagulls on west coast weight more than seagulls on east coast.

Are males and females equaly represented?

We want to know if males and females are equaly represented, that is, if ratio of males to entire population is 50%.

$$H_0 : p_{males} = 0.50$$
$$H_A : p_{males} \neq 0.50$$

Since samples in our dataset are independant observations, first CLT condition is satisfied. We also have 1207 males and 1280 females. Both numbers are greater than 10, so we can proceed with categorical inference on proportion testing.

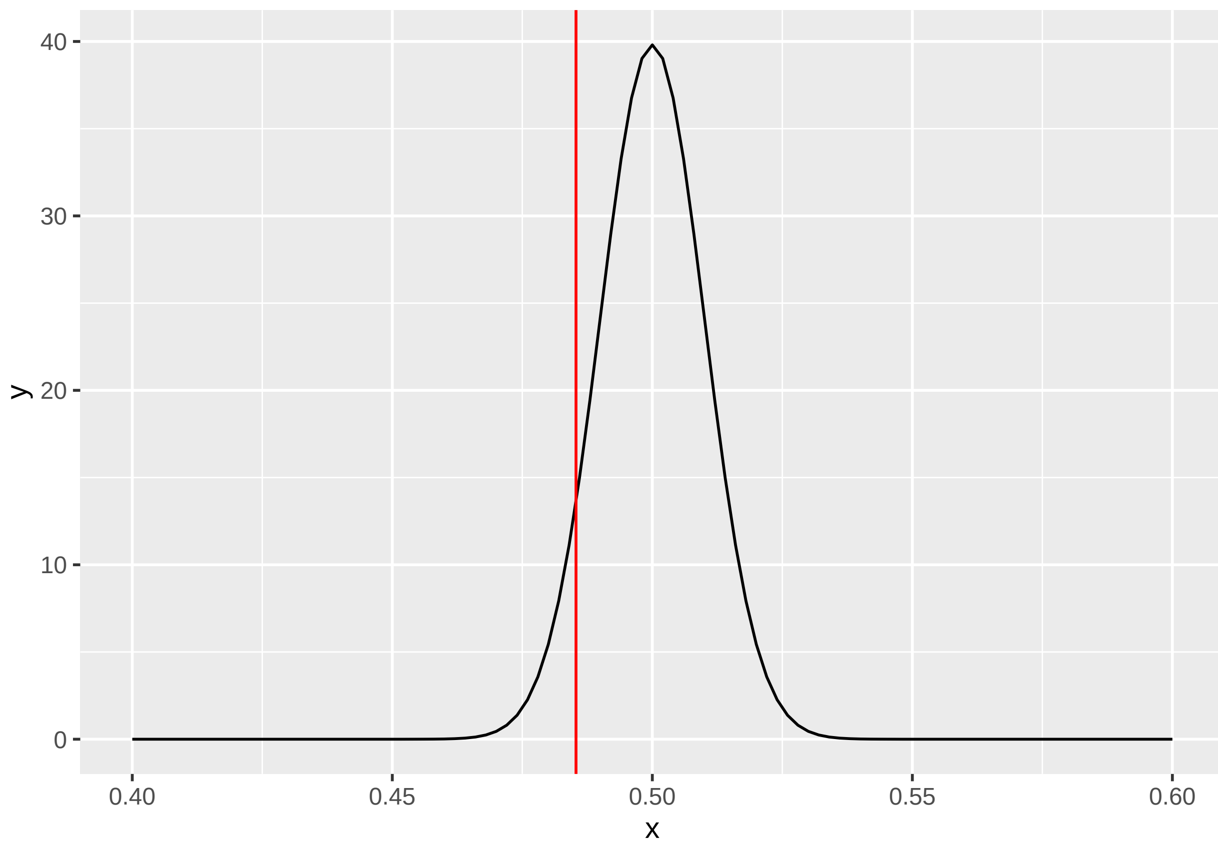
```r
(ratio <- seagulls %>% filter(SEX == "MALE") %>% nrow() / nrow(seagulls))
```

```
## [1] 0.4853237
```

```r
(SE <- sqrt(ratio * (1 - ratio) / nrow(seagulls)))
```

```
## [1] 0.01002178
```

```r
ggplot(data.frame(x = seq(0.4, 0.6, length = 100)), aes(x = x))+
  stat_function(fun = dnorm, args = list(mean = 0.5, sd = SE))+
  geom_vline(xintercept = ratio, color = "red")
```




```r
(p_value <- pnorm(ratio, mean = 0.5, sd = SE))
```

```
## [1] 0.07153663
```

Since p_value of 0.0715366 is greater than our threshold value of 0.05 we accept the null hypothesis. There is the same number of males and females in seagull population.

Are the locations in our dataset representative?

We are interested if the locations in our dataset are represented equally.

$$H_0$$

: Equal proportions of all locaitons

$$H_A$$

: Unequal proportions of all locations

We are goind to do a xhi-squared test for goodness of fit.

Collected data about locations is independant. All 4 categories also have at least 5 cases to them, so both chi-square test conditions are met.

```r
table(seagulls$LOCATION)
```

```
##
## MARAETAI  MURIWAI     PIHA  WAITAWA
##      673      589      647      578
```

We need to calculate expected count for each category and for every category we calculate its Z score and then sum squares of Z scores together. Finally we check where on this chi-squared distribution lies our score.

```
(num_classes <- length(unique(seagulls$LOCATION)))
```

```
## [1] 4
```

```
(expected_location <- nrow(seagulls) / num_classes)
```

```
## [1] 621.75
```

```
(z <- (table(seagulls$LOCATION) - expected_location) / sqrt(expected_location))
```

```
##
##  MARAETAI   MURIWAI      PIHA   WAITAWA
##  2.055351 -1.313419  1.012636 -1.754568
```

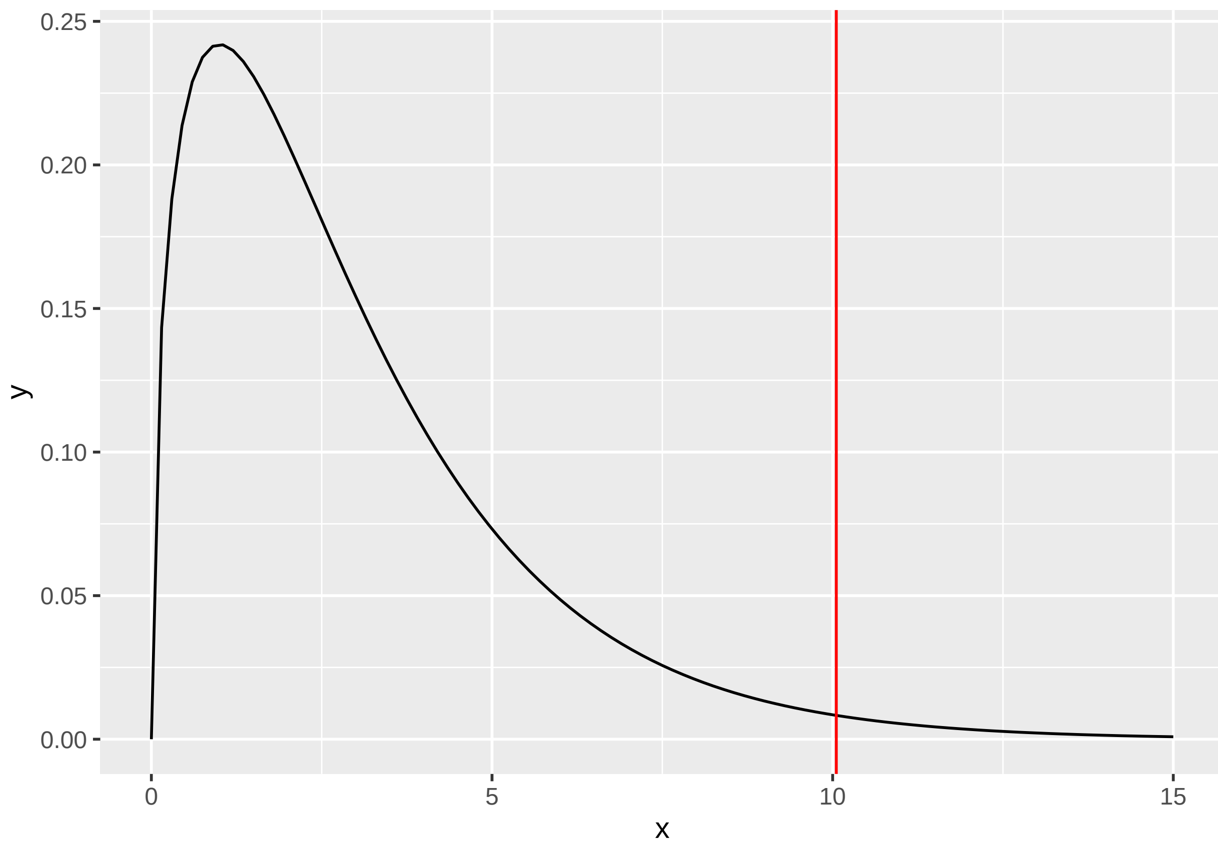
```
(chi <- sum(z ^ 2))
```

```
## [1] 10.05348
```

```
(df <- num_classes - 1)
```

```
## [1] 3
```

```
ggplot(data.frame(x = seq(0, 15, length = 100)), aes(x = x))+
  stat_function(fun = dchisq, args = list(df = df))+
  geom_vline(xintercept = chi, color = "red")
```




```
(p_value <- 1 - pchisq(chi, df))
```

```
## [1] 0.01811698
```

Since our p_value is smaller then α ($0.018117 < 0.05$), we can reject out null hypothesis in favor of alternative. Locations in our dataset are not equaly represented.

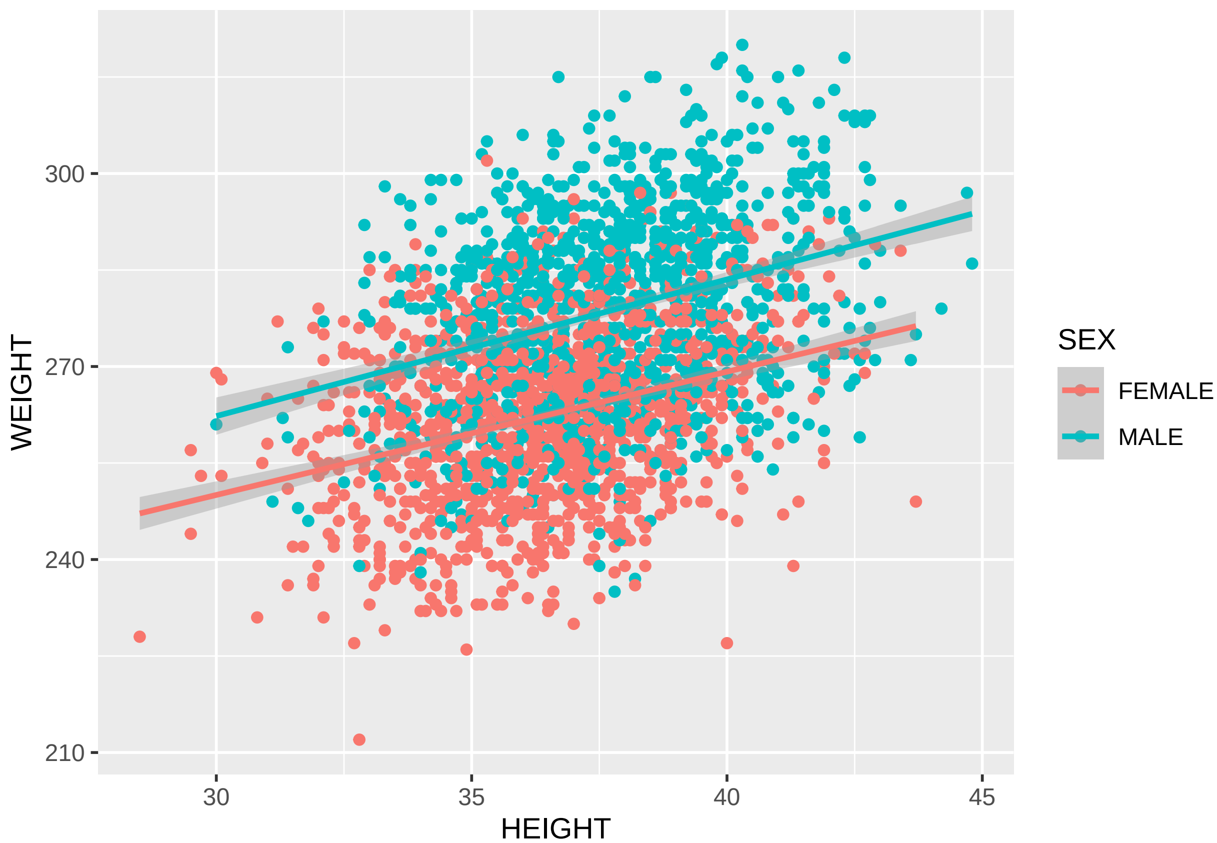
Linear regression

There is one more thing we want to know about seagulls: is the weight of a seagull dependant of its height.

First we can take a look at height-weight graph:

```
seagulls %>% ggplot(aes(x = HEIGHT, y = WEIGHT, color = SEX))+
  geom_point()+
  geom_smooth(method = lm)
```

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



It seems like there is really no connection between height and weight. We further calculate its correlation coefficient:

```
(c <- cor(seagulls$HEIGHT, seagulls$WEIGHT))
```

```
## [1] 0.3972763
```

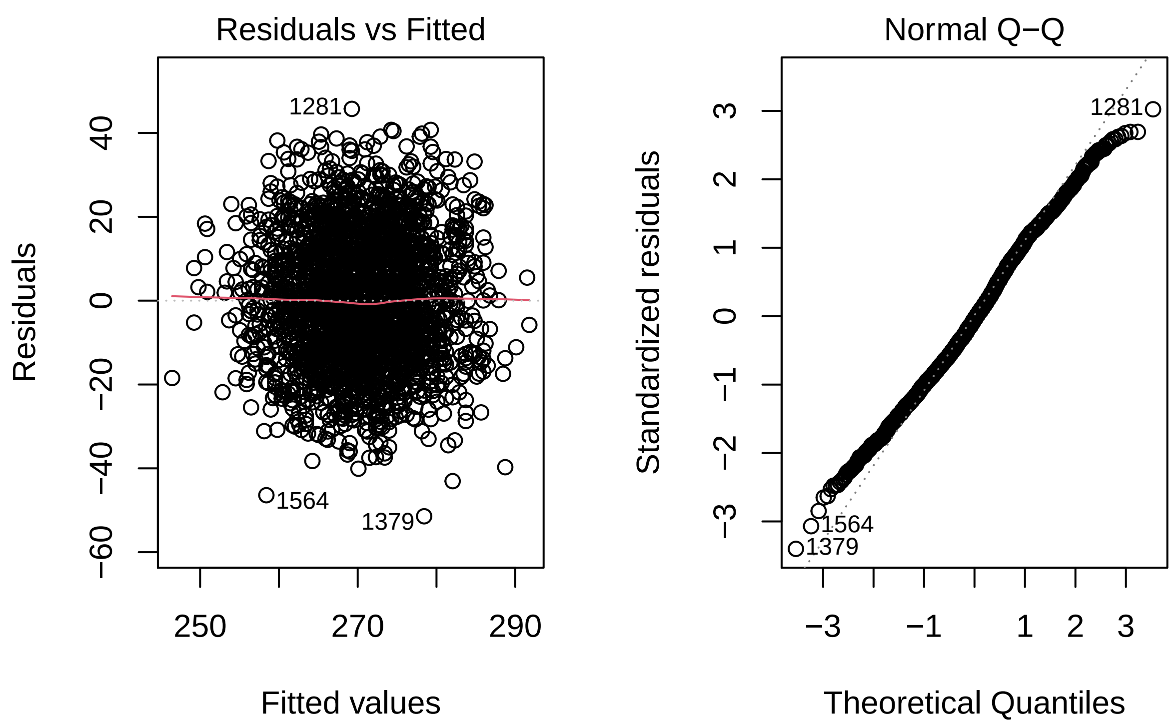
Correlation coefficient of 0.3972763 hints that there is a weak correlation between height and weight of seagulls. But we can still try and create a linear model between those two variables:

```
fit_wh <- lm(WEIGHT ~ HEIGHT, data = seagulls)
summary(fit_wh)
```

```
##
## Call:
## lm(formula = WEIGHT ~ HEIGHT, data = seagulls)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.435 -11.008  -0.694  11.406  45.744
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 167.1826     4.7911   34.90   <2e-16 ***
## HEIGHT        2.7813     0.1289   21.58   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.13 on 2485 degrees of freedom
## Multiple R-squared:  0.1578, Adjusted R-squared:  0.1575
## F-statistic: 465.7 on 1 and 2485 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(fit_wh, which=1:2)
```

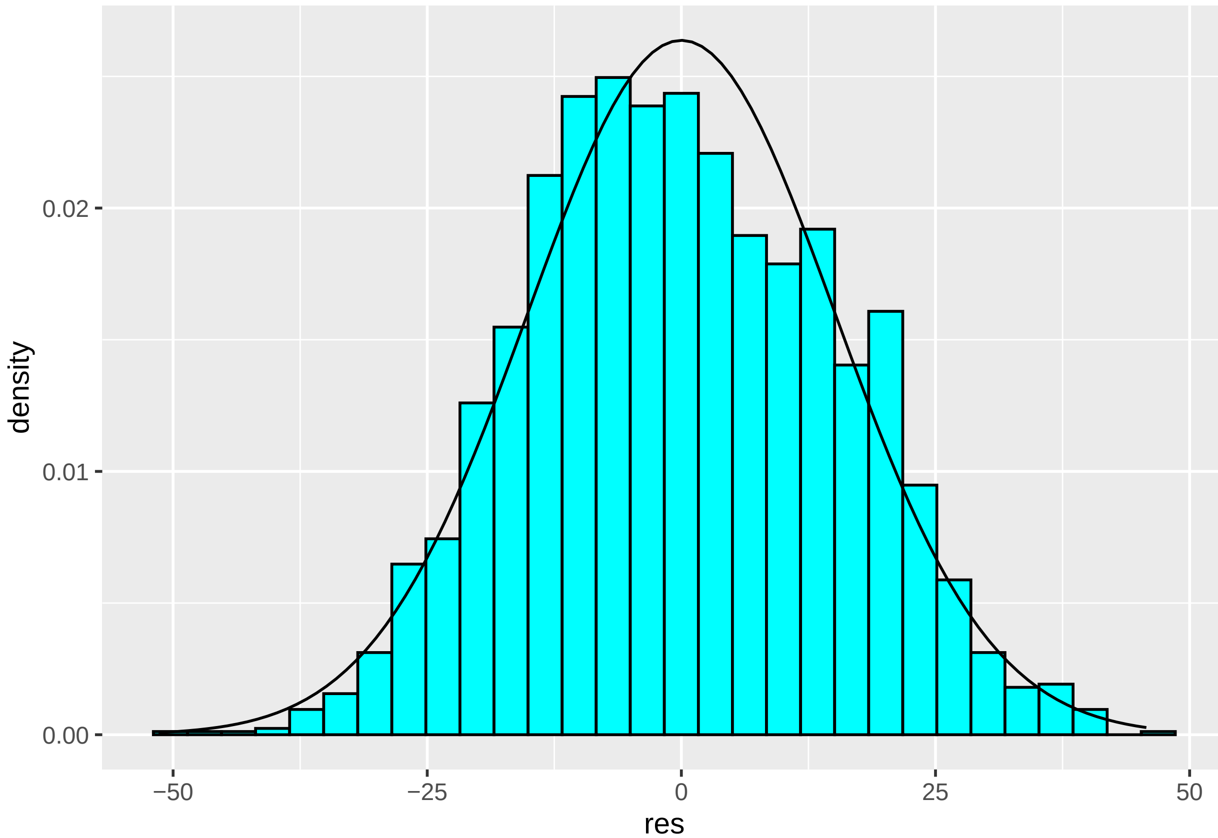


We can se that residuals have no visible pattern and on the q-q plot the residuals follow a nice line, so our model is valid. The most concerning thing is the R^2 value, which is only 0.1578285.

We can also plot the density of residuals and see that they follow a normal distribution.

```
res <- residuals(fit_wh)
seagulls %>% ggplot()+
  geom_histogram(aes(x = res, y = ..density..), fill = "cyan", color = "black")+
  stat_function(fun = dnorm, args = list(mean = mean(res), sd = sd(res)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


0.02
0.01
0.00
density
−50
−25
0
25
50
res