MATH 423/533: ASSIGNMENT 3

Hair Parra

November 20, 2019

Hair ALbeiro Parra Barrera

260738619

MATH 423/533: ASSG1

Assignment 1

Question 1

```
# Load the datasett
file1 <- "http://www.math.mcgill.ca/yyang/regression/data/abalone.csv"
abalone <- read.csv(file1, header = TRUE)
head(abalone)
```

```
##   Height Rings
## 1  0.095    15
## 2  0.090     7
## 3  0.135     9
## 4  0.125    10
## 5  0.080     7
## 6  0.095     8
```

```
nrow(abalone) # number of observations = 4177
```

```
## [1] 4177
```

Research Problem: Abalones, also called ear-shells or sea ears, are one type of reef-dwelling marine snails. The flesh of abalones is widely considered to be a desirable food, and is consumed raw or cooked in a variety of cultures. It is difficult to tell the ages of abalones because their shell sizes not only depend on how old they are, but also depend on the availability of food.The study of age is usually by obtaining a stained sample of the shell and looking at the number of rings through a microscope. A research group are interested in using some of abalones' physical measurements, especially the height measurement to predict their ages. The research group believe that a simple linear regression model with normal error assumption is appropriate to describe the relationship between the height of abalones and theirages, and particularly, that a larger height is associated with an older age.

Research Problem

We wish to investigate the relationship between between the height of abalones and their ages, assuming the simple linear regression model with normal error assumption is appropriate. As this is is related to the number of rings thet have, we hypothesize there exists a linear relationship between the variables "Height" and "Rings". (Reseach problem formulation). If there is such a relationship, we could especulate the simulatenous change of these.

Variables basic statistics

We examine the variables independly, providing summary statistics as a part of basic data exploration.

Height variable

```
# Obtain minium and maximum values, 1st and 3rd quantiles, mean and median of both vairables
print("Summary statistics: ")
```

```
## [1] "Summary statistics: "
```

```
height <- abalone$Height
summary(height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1150  0.1400  0.1395  0.1650  1.1300
```

```
height.mean <- mean(height)
height.variance <- var(height)
height.std <- sqrt(height.variance)

sprintf("Mean of the Height variable: %.3f", height.mean)
```

```
## [1] "Mean of the Height variable: 0.140"
```

```
sprintf("Variance of the Height variable: %.3f", height.variance)
```

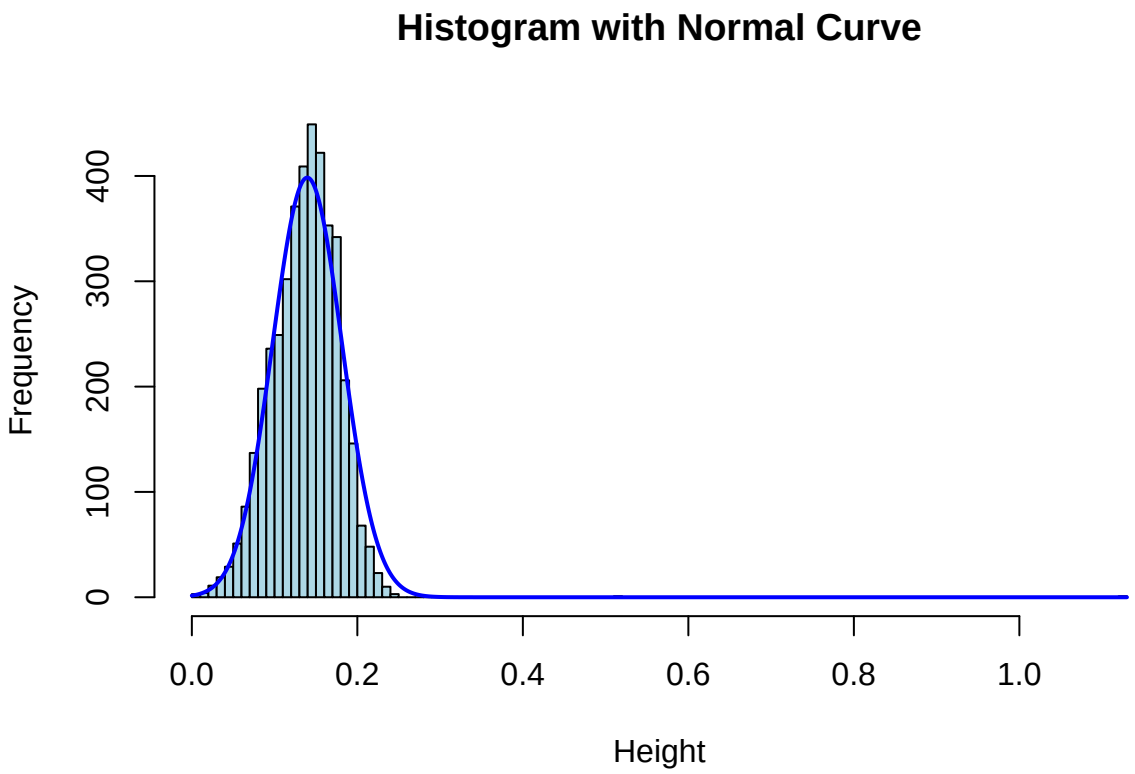
```
## [1] "Variance of the Height variable: 0.002"
```

```
sprintf("Standard deviation of the Height variable: %.3f", height.std)
```

```
## [1] "Standard deviation of the Height variable: 0.042"
```

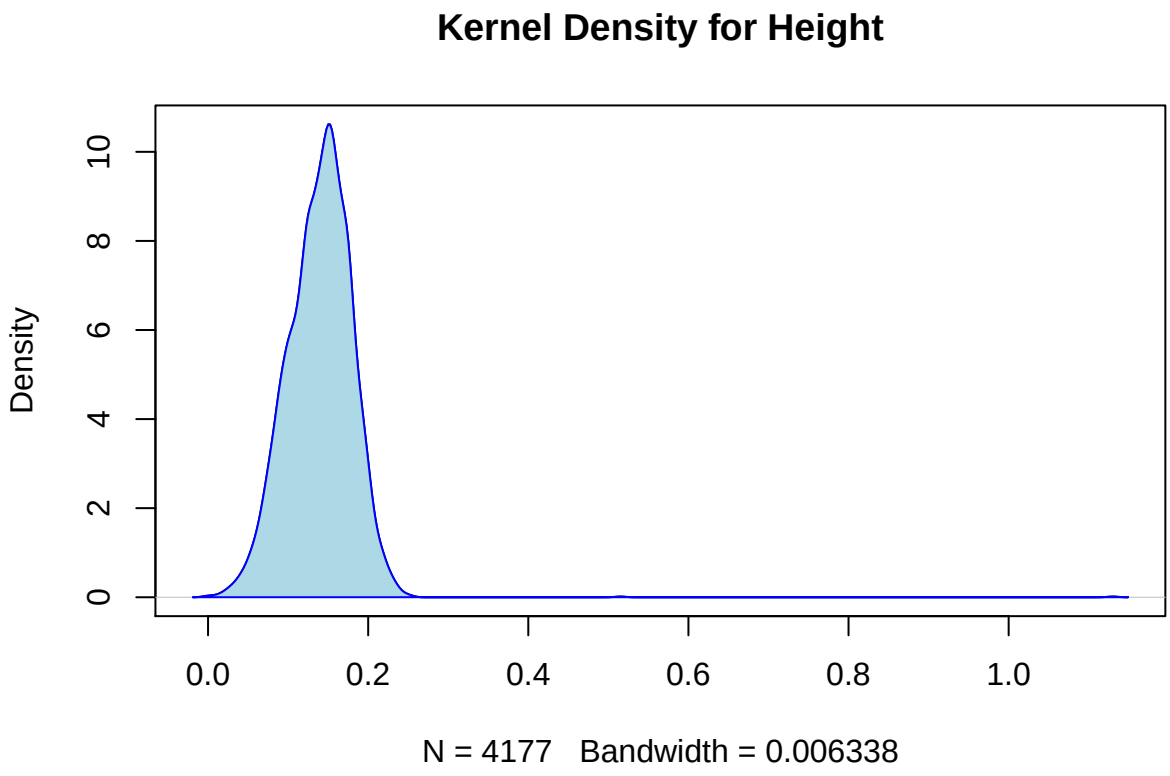
First note, that these measurements seem to be in a small scale overall, which could be due to the fact that they were taken with a microscope. We observe a mean close to zero and a meadian around 0.1395, but the maximum observation with a value of 1.1300 is indicative of the presence of outliers.

```
# plot a histogram along with a normal curve
h <- hist(abalone$Height, breaks = 100 , col="light blue", xlab="Height", main="Histogram with Normal Cu
xfit <- seq(min(height), max(height),length = 4177)
yfit <- dnorm(xfit, mean=height.mean, sd=height.std)
yfit <- yfit*diff(h$mids[1:2])*length(height)
lines(xfit, yfit, col="blue", lwd=2)
```



We observe that although the distribution seems to be approximatedly normal, the huge range with concentration towards the left reflects the presence of outliers, so we should be aware of this. The following Kernel Density plot confirms this.

```
# Filled Denisty plot
d <- density(height)
plot(d, main="Kernel Density for Height")
polygon(d, col="light blue", border = "blue")
```



Rings variable

```
# Obtain minium and maximum values, 1st and 3rd quantiles, mean and median of both vairables
print("Summary statistics: ")
```

```
## [1] "Summary statistics: "
```

```
rings <- abalone$Rings
summary(rings)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   8.000   9.000   9.934  11.000  29.000
```

```
rings.mean <- mean(rings)
rings.variance <- var(rings)
rings.std <- sqrt(rings.variance)

sprintf("Mean of the Rings variable: %.3f", rings.mean)
```

```
## [1] "Mean of the Rings variable: 9.934"
```

```
sprintf("Variance of the Rings variable: %.3f", rings.variance)
```

```
## [1] "Variance of the Rings variable: 10.395"
```

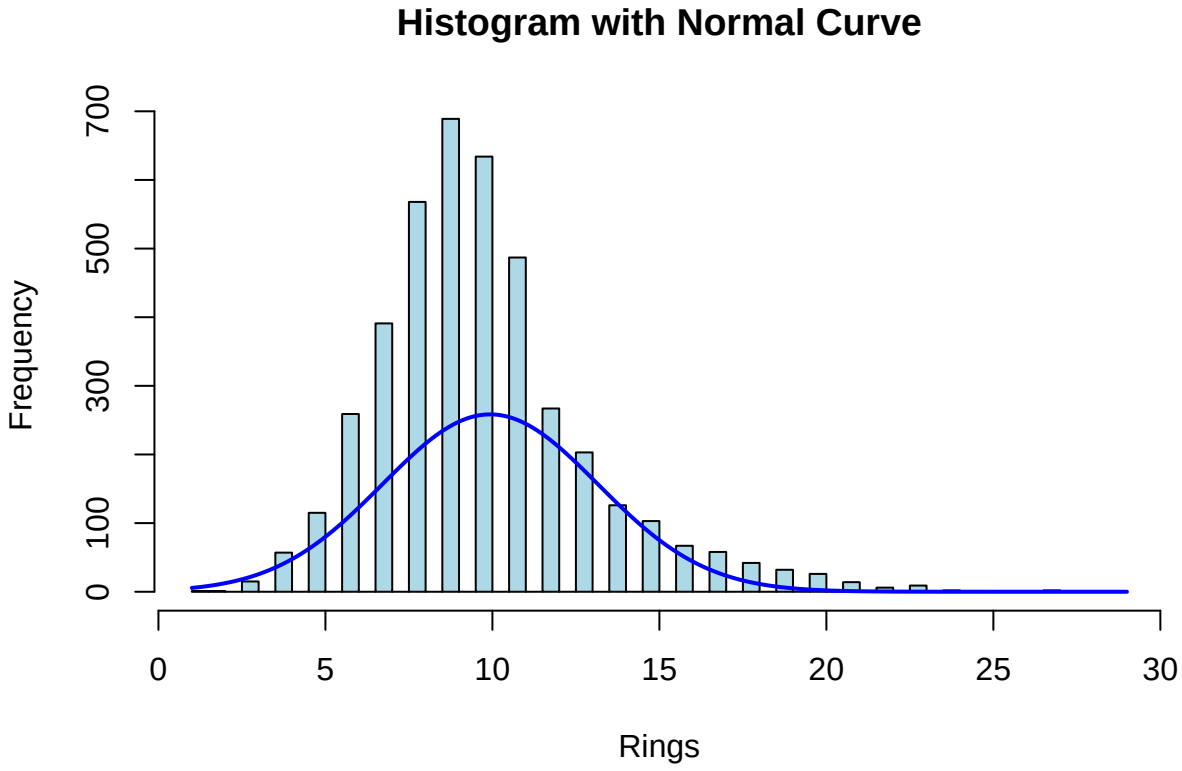
```r
sprintf("Standard deviation of the Rings variable: %.3f", rings.std)
```

```
## [1] "Standard deviation of the Rings variable: 3.224"
```

We observe a range [1.0,29.0] , approximatedly equal mean and median ,but the maximum value could indicate some outliers. We also observe that the variance of these data seems to be quite big relative to the range of the data.

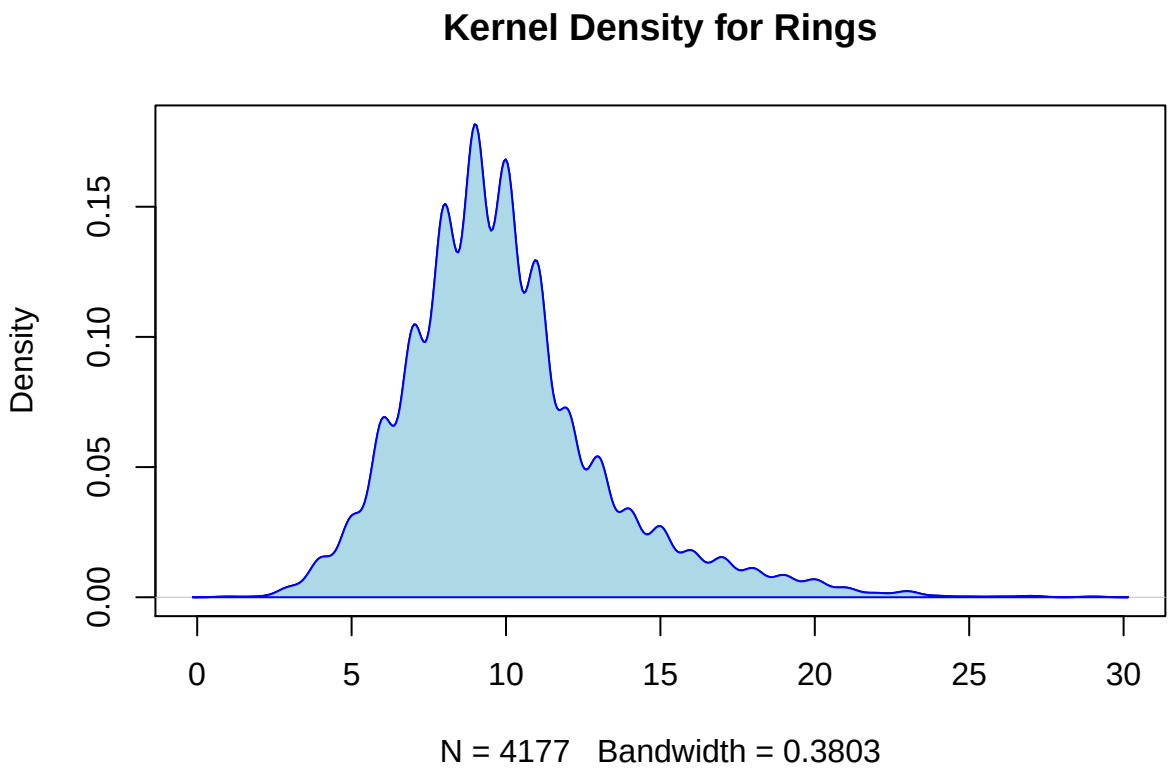
Now plot the density

```r
# plot a histogram along with a normal curve
h <- hist(rings, breaks = 50 , col="light blue", xlab="Rings", main="Histogram with Normal Curve")
xfit <- seq(min(rings), max(rings),length = 4177)
yfit <- dnorm(xfit, mean=rings.mean, sd=rings.std)
yfit <- yfit*diff(h$mids[1:2])*length(rings)
lines(xfit, yfit, col="blue", lwd=2)
```



Although it looks to be somehow bell-shaped, we see that it deviates from the regular normal distribution with same sample mean and std. In fact, we suspect it could come from a χ^2 distribution instead. Next we draw the density plot:

```r
# Filled Denisty plot
d <- density(rings)
plot(d, main="Kernel Density for Rings")
polygon(d, col="light blue", border = "blue")
```

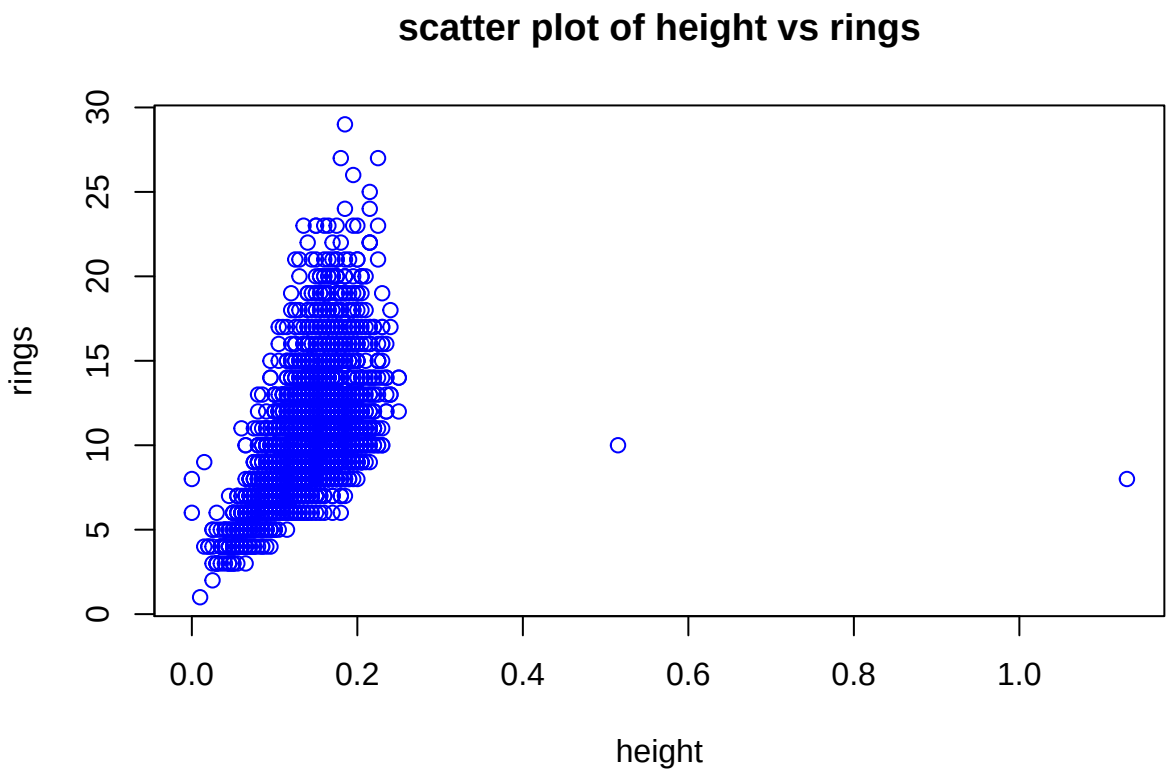


We can observe that indeed, the Rings random variable does not appear to be normally distributed.

Scatterplot

Now we bring our analysis to both variables in a scatterplot

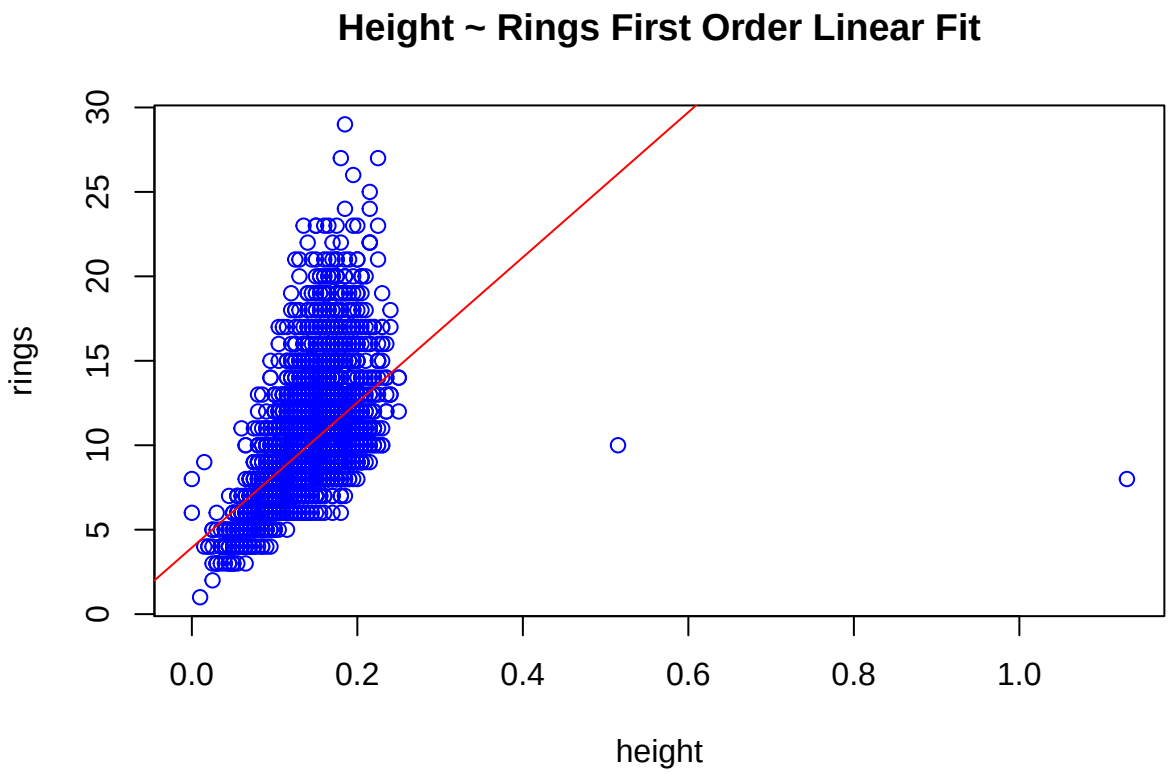
```
# Draw scatter plot
plot(height, rings, col = "blue", main= "scatter plot of height vs rings")
```



We observe a cluster of points which appear to have an positive (but not necessarily linear) correlation pattern. We also observe there are two significantly outliers, which do not follow the group trend. We suspect that a linear relationship might not be the most adequate in this case.

First order linear fit

```
# Fit a linear model, plot and then obtain summary.
lm.abalone <- lm(rings ~ height)
plot(height, rings, xlab="height", ylab="rings", main="Height ~ Rings First Order Linear Fit", col = "bl
abline(coef(lm.abalone), col="red")
```



We now examine the model's fit summary:

```
summary(lm.abalone)
```

```
##
## Call:
## lm(formula = rings ~ height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.496  -1.657  -0.607   0.839  17.112
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9385     0.1443   27.30   <2e-16 ***
## height       42.9714     0.9904   43.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.677 on 4175 degrees of freedom
## Multiple R-squared:  0.3108, Adjusted R-squared:  0.3106
## F-statistic:  1882 on 1 and 4175 DF,  p-value: < 2.2e-16
```

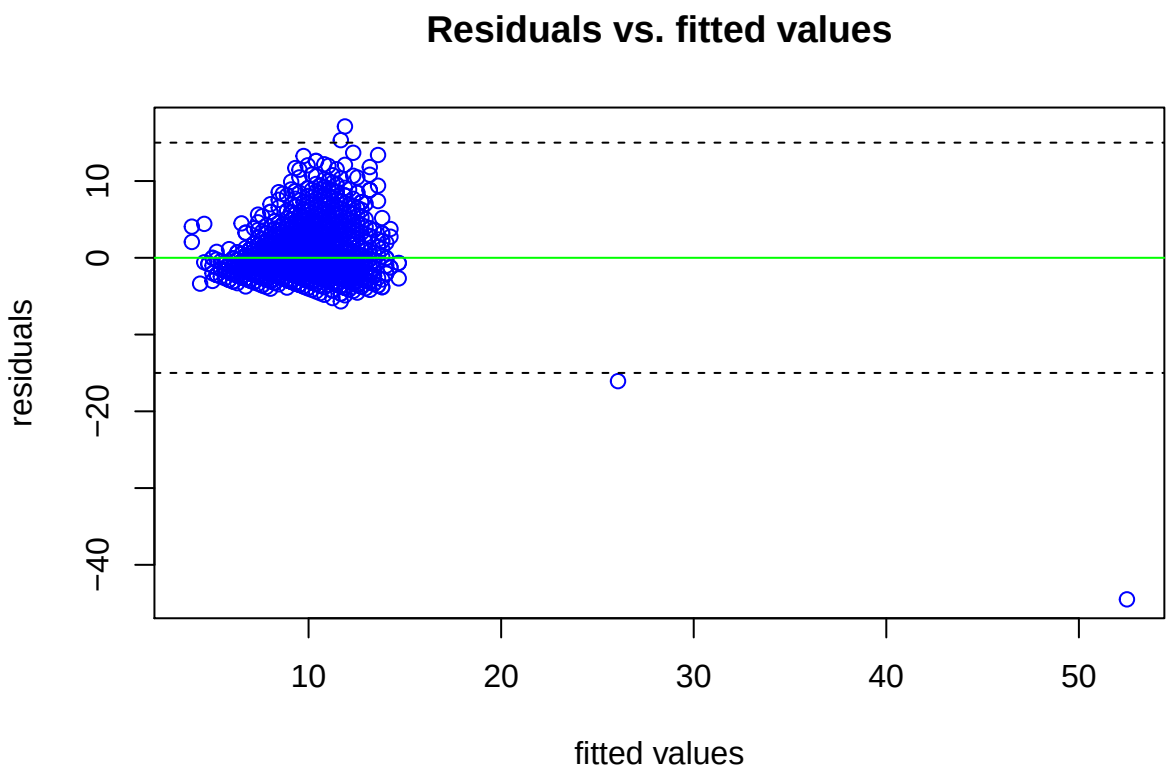
We notice that although the $\hat{\beta}_0$ and $\hat{\beta}_1$ estimates both have significant p-values, looking at the model's Adjusted R-square we observe that the current fit only explains the variance in the data very poorly (0.31), so it is overall a very bad fit.

Diagnostics

We now examine the model more closely but examining the residuals, as well as normality assumptions.

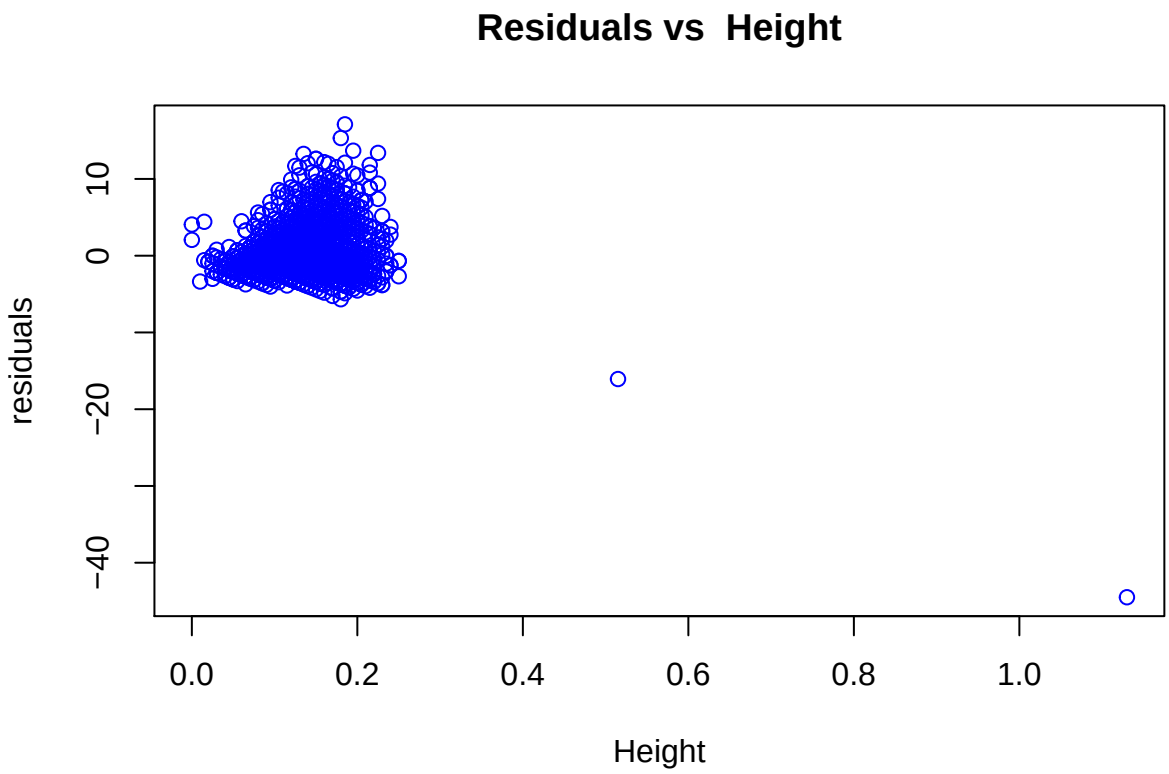
Residuals plot

```
res.abalone <- residuals(lm.abalone) # extract the residuals
fitted.abalone <- lm.abalone$fitted.values # extract the fitted values
plot(fitted.abalone, res.abalone, xlab= "fitted values" , ylab = "residuals",
     main="Residuals vs. fitted values", col = "blue") # plot the residuals vs fitted values
abline(h=0, col="green")
abline(h=c(-15,15), lty=2)
```



Just as we suspected, we can observe two significant outliers as well as a clustering of points with a pattern that does not seem symmetrically and randomly distributed aout 0. This implies that the **linear relationship assumption might now be appropriate**.

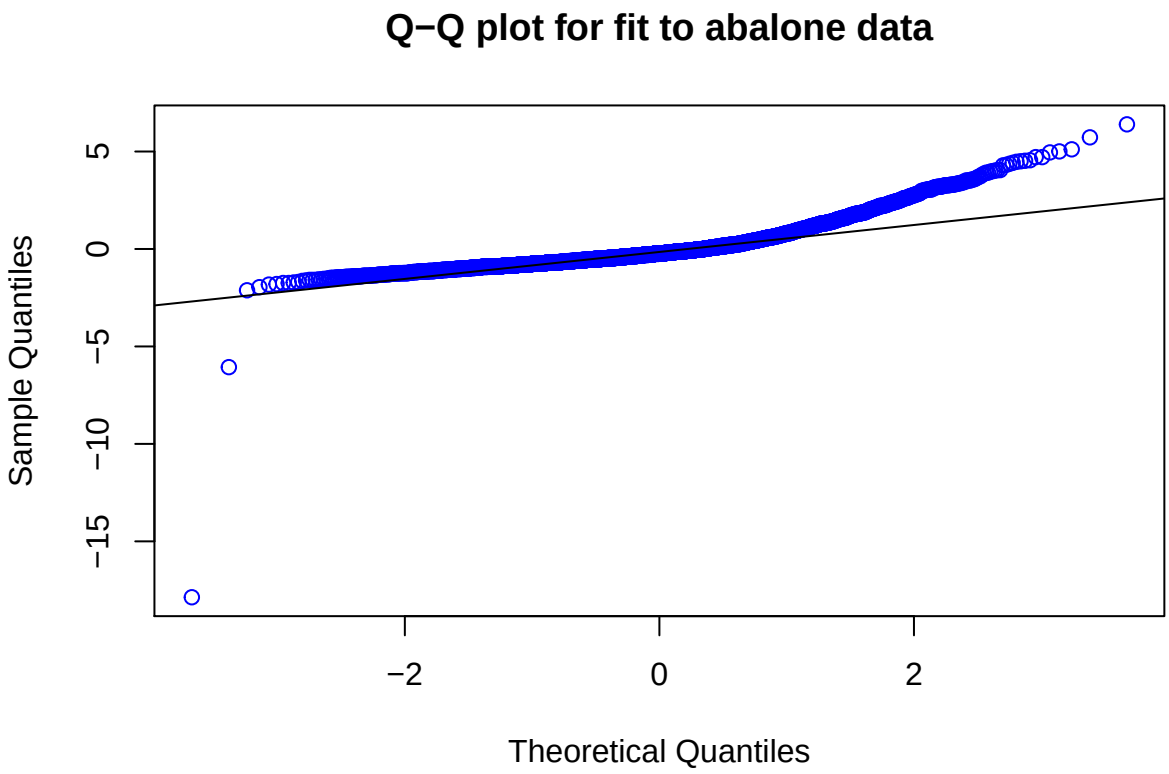
```
res.abalone <- residuals(lm.abalone) # extract the residuals
plot(abalone$Height , lm.abalone$residuals, xlab= "Height" , ylab = "residuals",
     main="Residuals vs  Height", col = "blue") # plot the residuals vs Height
```



Plotting the residuals vs. Height also yields a very similar plot, which again indicates the presence of outliers, as well as the fact that the 0-mean and constant variance assumptions are violated.

Q-Q Plot

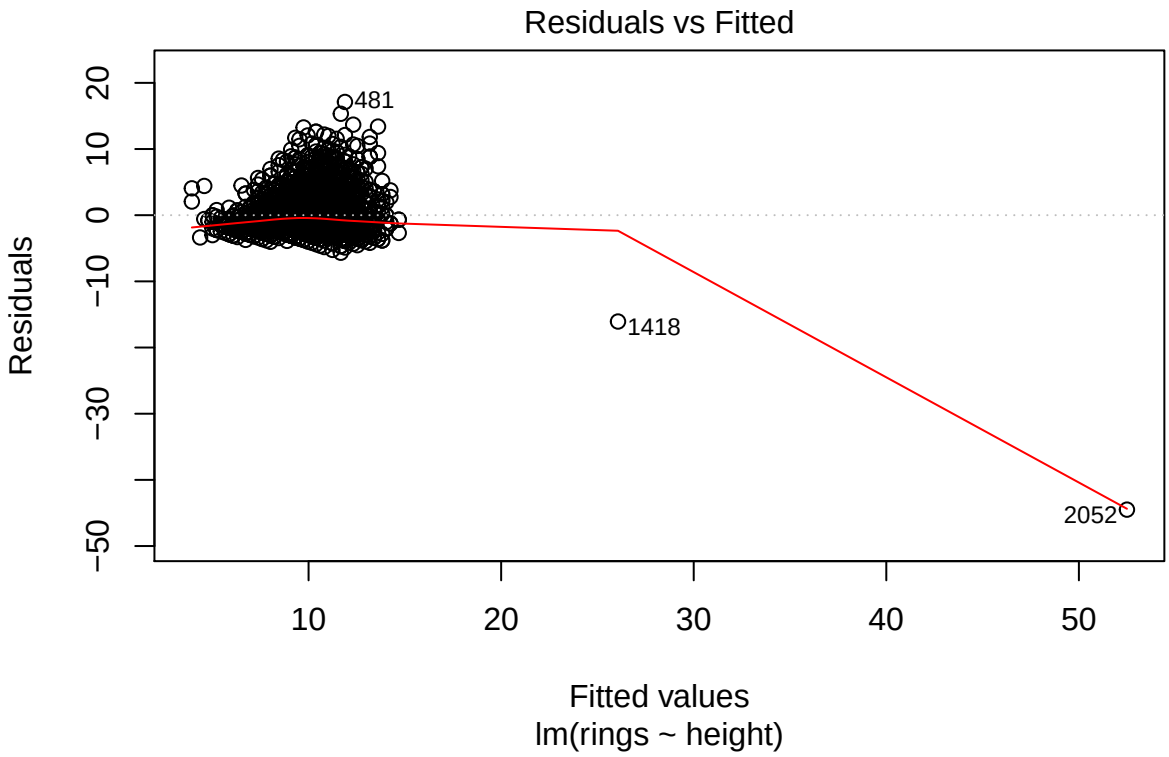
```
## QQ-plot to test for normality
res.abalone <- rstandard(lm.abalone) # obtain the standard residuals
qqnorm(res.abalone, main="Q-Q plot for fit to abalone data", col="blue")  # plot the qq-plot
qqline(res.abalone)
```

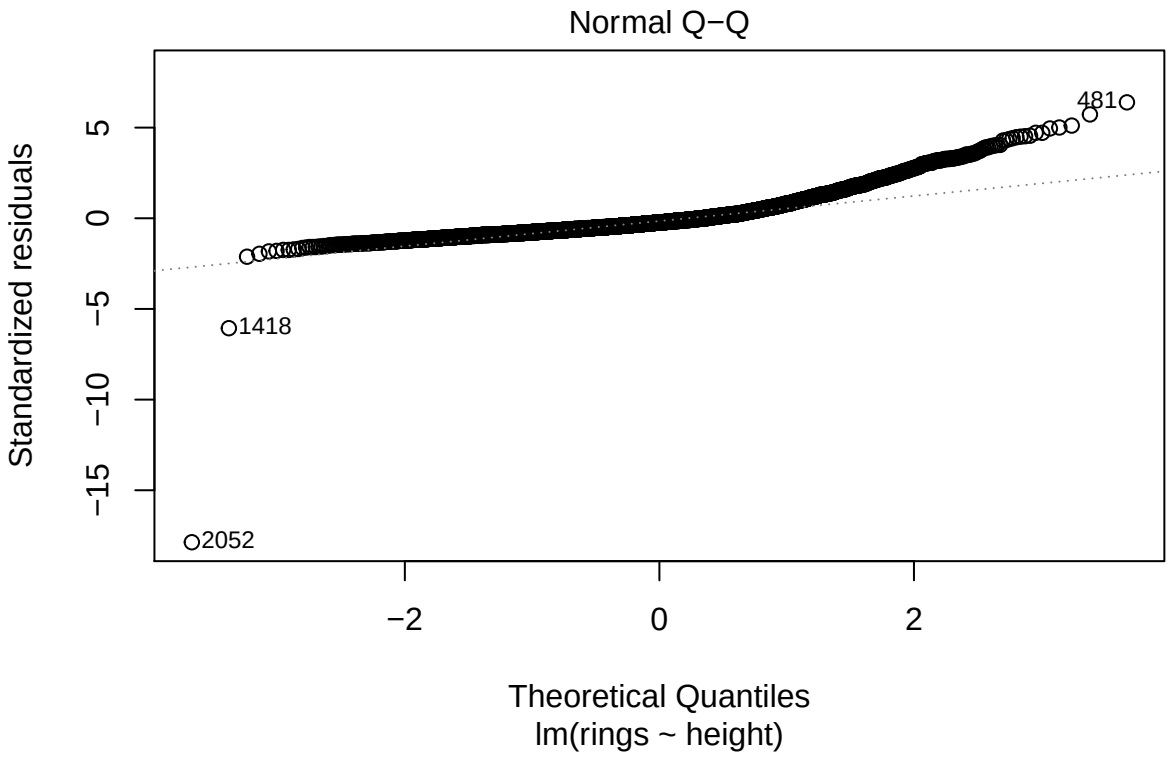


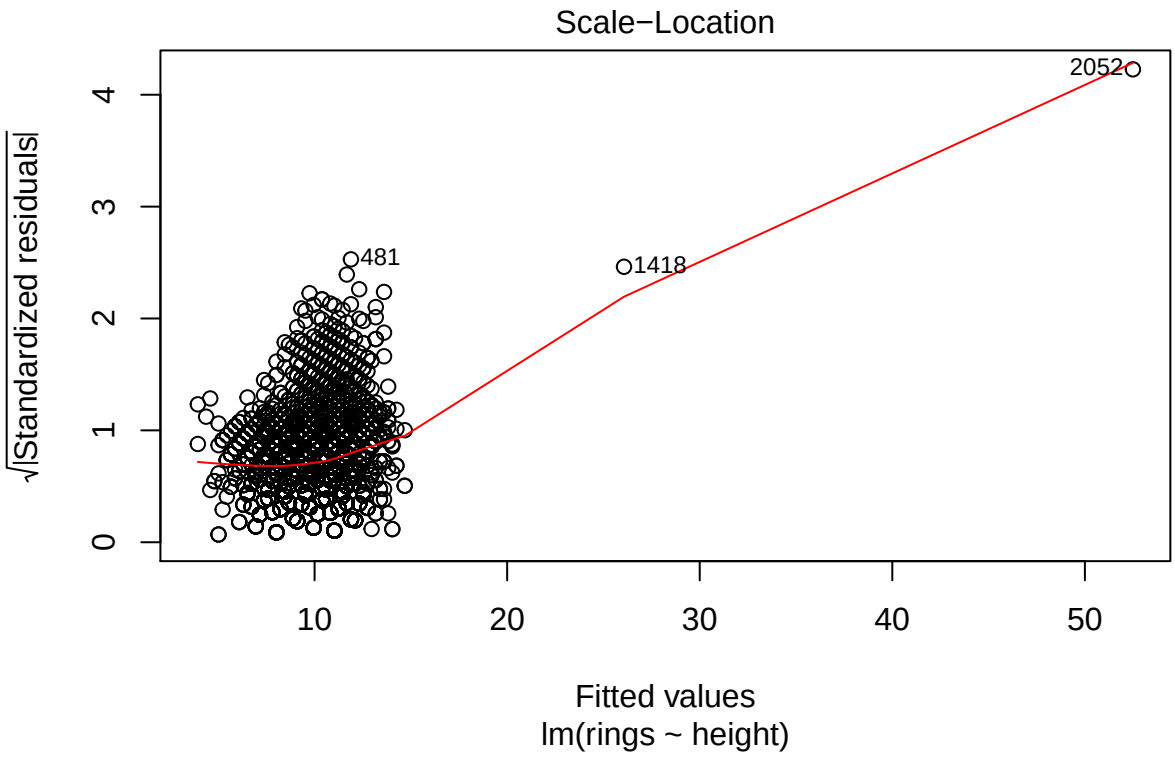
We can observe from the Q-Q Plot that since most of the points are on top of the line and even follow a certain pattern, this indicates that the normality assumptions are not met

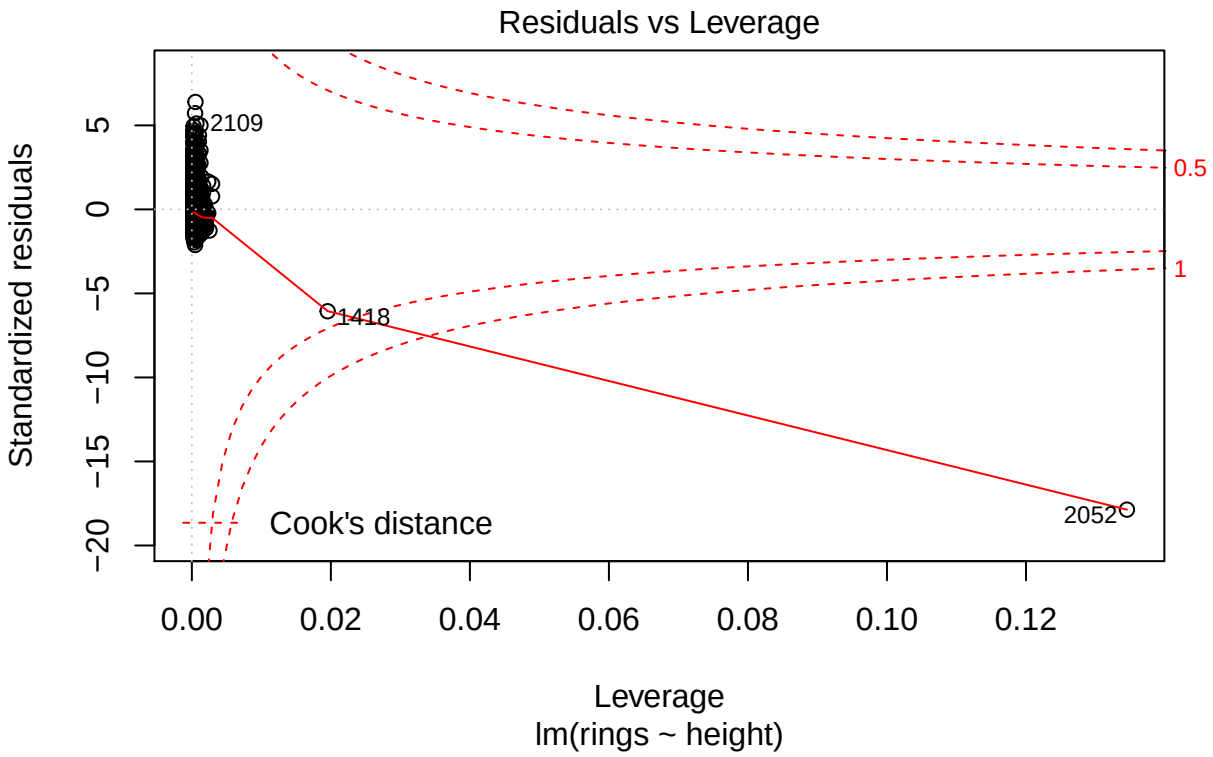
In summary

```
plot(lm.abalone)
```


Residuals vs Fitted
Residuals
20
10
0
−10
−30
−50
481
1418
2052
10
20
30
40
50
Fitted values
lm(rings ~ height)


Normal Q−Q
Standardized residuals
5
0
−5
−10
−15
481
1418
2052
−2
0
2
Theoretical Quantiles
lm(rings ~ height)


Scale−Location
√|Standardized residuals|
0
1
2
3
4
10
20
30
40
50
481
1418
2052
Fitted values
lm(rings ~ height)



Looking a the forth plot, we observe that there are at least two significant outliers, which have a really big cook distance.

Transforming the data

Although the assumptions are not directly met, we can modify the data to improve fit. We do so by first getting rid of outliers, and then applying an exponential transformation to the feature variable.

```
library(outliers)

df <- data.frame(height, rings)
# obtian outliers using z-score
outliers <- scores(height, type="chisq", prob=0.99)  # beyond 95th %ile based on z-scores
# remove the rows whose z-score is less than 0.99
df <- df[which(outliers == FALSE),]
sprintf("Previous number of observations: %d ", length(height))
```

```
## [1] "Previous number of observations: 4177 "
```

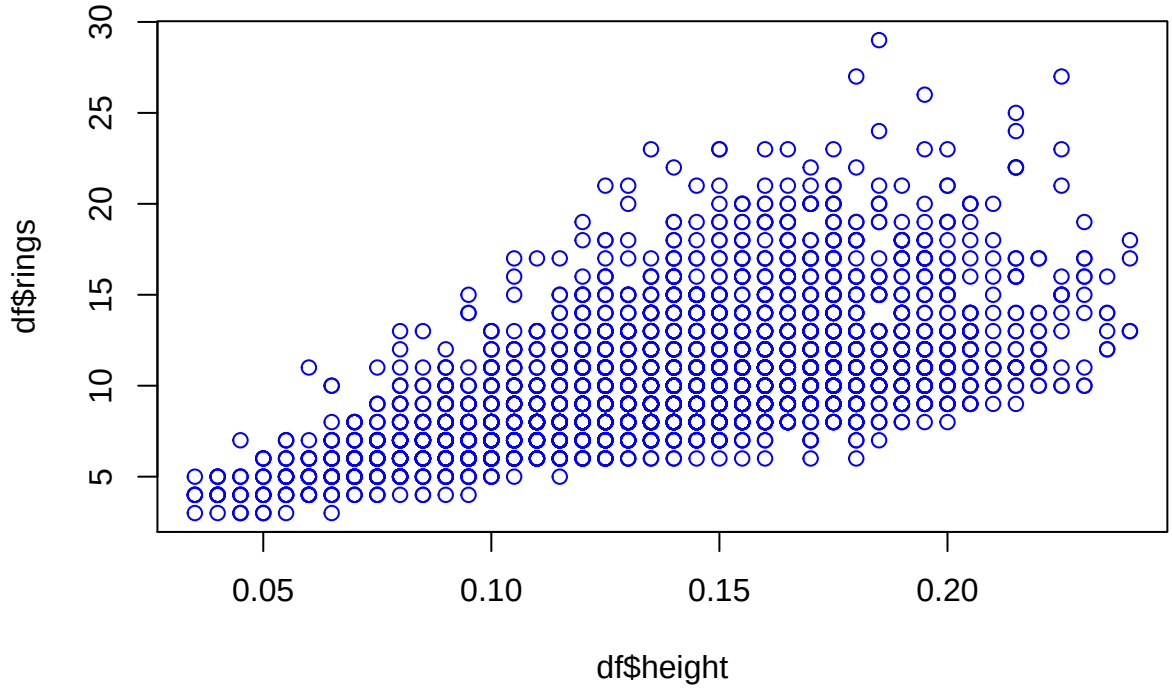
```
sprintf("Current number of observations: %d", nrow(df))
```

```
## [1] "Current number of observations: 4154"
```

Recall that we originally had 4177 observations, after eliminating potential outliers, we are left with 4154. Even if some of them were not actual outliers, at least they were potential outliers, and since the amount of

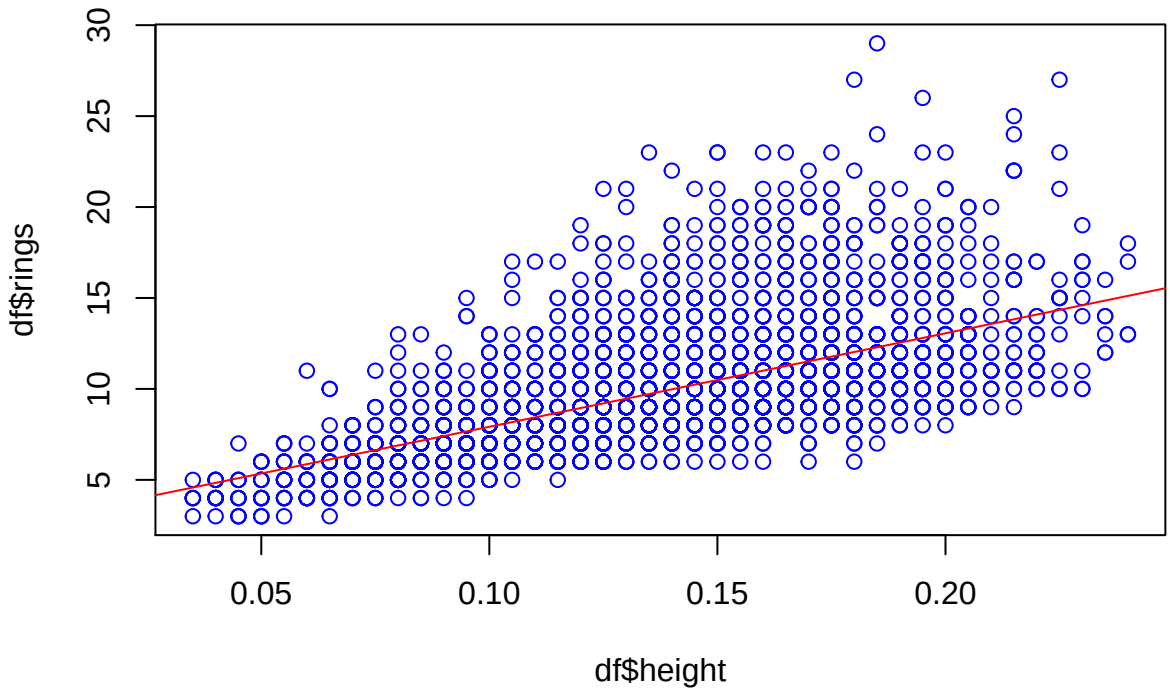
data we lost is relateively small to the size of the dataset, we can be sure we haven't loss that much relevant information. Let's plot the data and see how it looks:

```
plot(df$height, df$rings, col="blue")
```



Now we can concentrate on the big cluster of data without extreme outliers. We estimate that the relationship between the two given variables is in fact more complicated than just linear, so given the shape of the data, we will attemp to apply log transformations to both the rings variable and the height variable, and then fit a model of the log(log(rings)) as a function of the height and log(height). Notice that the variance is quite big with respect to the rings variable.

```
# straight regression fit ?
mod.abalone <- lm(rings ~ height, data = df)
plot(df$height, df$rings,  col="blue")
abline(coef(mod.abalone), col="red")
```




```
summary(mod.abalone)
```

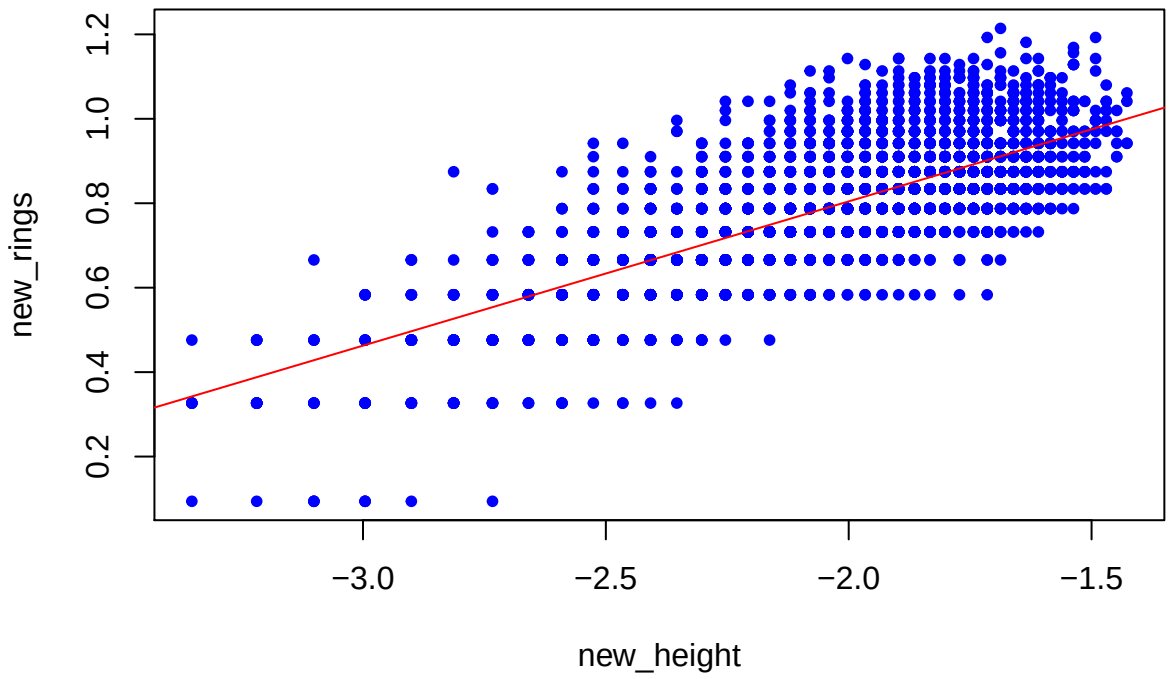
```
##
## Call:
## lm(formula = rings ~ height, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0312 -1.6628 -0.5451  0.8233 16.7118
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.7805     0.1524   18.25   <2e-16 ***
## height       51.3929     1.0536   48.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.558 on 4152 degrees of freedom
## Multiple R-squared:  0.3643, Adjusted R-squared:  0.3642
## F-statistic:  2379 on 1 and 4152 DF,  p-value: < 2.2e-16
```

We can observe that the variance of Y seems to increase as height increases, but we can improve the fit by applying a transformation to Y. In particular, we let

$$Y' := log(log(Y)) \; X' := log(X)$$

```
# otbain filtered variables
new_height <- log(df$height)
new_rings <- log(log(df$rings))

plot(new_height, new_rings, col= "blue", pch=20)
lm_new.abalone <- lm(new_rings ~ new_height )
abline(coef(lm_new.abalone), col="red")
```




```
# xp <- seq(0.01,0.30, by=0.01)
# m.fit <- cbind(rep(1,length(xp)), xp, log(xp)) %*% coef(lm_new.abalone)
# lines(xp,m.fit, col="red", lty=2)
# legend(0.25,0.4,c("Straight Line","Quadratic"),lty=c(1,2),col=c("black","red"))
```

We observe that we have improved the model quite a bit, but the variance seems to be too big to begin with. (ref: analysis of the rings variable above).

```
summary(lm_new.abalone)
```

```
##
## Call:
## lm(formula = new_rings ~ new_height)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.45981 -0.06355 -0.01007  0.05862  0.34808
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.487666   0.010238   145.3   <2e-16 ***
## new_height  0.341634   0.005031    67.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09975 on 4152 degrees of freedom
## Multiple R-squared:  0.5262, Adjusted R-squared:  0.5261
## F-statistic:  4611 on 1 and 4152 DF,  p-value: < 2.2e-16
```

```
summary(aov(lm_new.abalone))
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## new_height     1  45.88   45.88    4611 <2e-16 ***
## Residuals   4152  41.31    0.01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observing the two tables above we can see that the intercept, the filtered height and log of height attributes have very significant p-values. In addition, we see that the amount of variance explained in the Adjusted R-square statistic improved from 0.3106 in the raw , first order linear model to 0.5261 in the current model with logarithmically transformet features and target variables. In addition, the F-statistic's p-value $< \alpha = 0.01$ indicates that the model's fit is significant.

Interpretation

Letting Y =number of rings and X_1 =height, we are assuming that the true model takes the form

$$Y' = \beta_0 + \beta_1 X_1' \equiv log(log(Y)) = \theta_0 + \theta_1 log(X_1)$$

Where $\hat{\beta}_0 = 1.487666$ and $\hat{\beta}_1 = 0.341634$. Under this transformation, we wee that per each unit of transformed height ($X_1' = log(X)$) increase, transformed rings ($Y' = log(log(Y))$) increases by an amount of ≈ 0.341634.

Confidence intervals

We can obtain the confidence 95% confidence intevals for β_0, β_1 as

$$C.I.(\beta_0) = \hat{\beta}_0 \pm t_{\alpha/2,n-1}\sqrt{\hat{\sigma}^2\left(\frac{1}{n} + \frac{\bar{x_1}^2}{S_{xx}}\right)}$$

$$C.I.(\beta_1) = \hat{\beta}_1 \pm t_{\alpha/2,n-1}\sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$$

```
# Calulate the confidence intervals
(confint(lm_new.abalone))
```

```
##                 2.5 %    97.5 %
## (Intercept) 1.4675935 1.5077386
## new_height  0.3317698 0.3514981
```

```
# COnfidence intervals for beta_0, beta_1 and beta_2

t_alpha_half <- qt(.95, 49) # .95th quantile of the t-distribution
(beta_0_hat <- coef(lm_new.abalone)[1]) # get beta 0  hat
```

```
## (Intercept)
##    1.487666
```

```
(beta_1_hat <- coef(lm_new.abalone)[2]) # get beta 1 hat
```

```
## new_height
##  0.3416339
```

Is there a statistically significant relationship between the height and the number of rings?

We test for the transformed hypothesis that

$$\begin{cases} H_0 : \beta_1 = 0 \\ H_a : \beta_1 \neq 0 \end{cases}$$

At the $\alpha = 0.01$ significance level, we see that since we have p-value$< 2e - 16 < 0.01$, we **reject** the null hypothesis that $\beta_1 = 0$ and therefore conclude there is a significantly statistical relationship between height and the number of rings (and hence, the age) of abalones. In other words, knowing the predictor "height" is somehow informativpredice of the actual number of rings of the abalone.

Predictions

We know find a point estimate and a 99% confidence interval for the average number of rings for abalones with height at 0.128. For this, we apply the same transformations to the new input, and inverse transformations to the output according to our model.

```
# tranform the new data accordingly
new_X <- log(0.128)

# Create new observation
new_data <- data.frame(new_height <- new_X)
exp(exp(predict(lm_new.abalone, newdata = new_data, interval = "prediction", level=0.99))) # predict an
```

```
##        fit      lwr      upr
## 1 8.963832 5.452023 17.05044
```

So we predict that the average number of rings for abalones with height 0.128 is approximatedly 9, being between approx. 6 and 14 95% of the time.

Question 2

Suppose that

$$Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i \ , \ i = 1, \ldots, n$$

where $\epsilon_i \sim N(0, \sigma^2)$. Notice that there is no intercept. Suppose that

$$\sum_{i=1}^{n} X_{i1} X_{i2} = 0$$

Show that the least squares estimators $\hat{\beta}_1$ and $\hat{\beta}_2$ from the multiple regression are the same as if we were to fit separate, simple regressions on X_1 and X_2.

Proof

If the true model were given by

$$Y = \beta_k X_k + \epsilon \ , \ k = 1, 2$$

then $\mathbb{E}(Y) = \beta_k X_k$ and so we model $\hat{\beta}_k X_k$. In order to find $\hat{\beta}_k$, we solve the minimization problem

$$\hat{\beta}_k = \underset{\beta_k}{arg\ min}\ \widehat{MSE}(\beta_k) = \underset{\beta_k}{arg\ min}\ \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Taking derivatives

$$\frac{d\widehat{MSE}(\beta_k)}{d\beta_k} = \frac{d}{d\beta_k}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right) = \frac{d}{d\beta_k}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_k x_i)^2\right)$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_k x_{ik}) x_{ik} := 0$$

$$\implies \sum_{i=1}^{n} x_{ik} y_i - \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 = 0$$

$$\implies \hat{\beta}_k = \frac{\sum_{i=1}^{n} x_{ik} y_i}{\sum_{i=1}^{n} x_{ik}^2}$$

Now, if we consider the true model to be

$$Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i \ , \ i = 1, \ldots, n$$

Then we model $\mathbb{E}(Y) = \beta_1 X_1 + \beta_2 X_2$ with $\hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$, and solve the problem

$$(\hat{\beta}_1, \hat{\beta}_1) = \underset{\beta_1, \beta_2}{arg\ min}\ \widehat{MSE}(\beta_1, \beta_2) = \underset{\beta_1, \beta_2}{arg\ min}\ \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

So taking partials wtr to β_1 we have

$$\frac{\partial \widehat{MSE}(\beta_1, \beta_2)}{\partial \beta_1} = \frac{\partial}{\partial \beta_1}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)$$

$$= \frac{\partial}{\partial \beta_1}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2\right)$$

$$= -\frac{2}{n}\sum_{i=1}^{n}(y_i - \beta_1 x_{i1} - \beta_2 x_{i2})x_{i1}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_{i1}y_i - \beta_1 x_{i1}^2 - \beta_2 x_{i1}x_{i2}) := 0$$

$$= \frac{1}{n}\sum_{i=1}^{n} x_{i1}y_i - \beta_1 \frac{1}{n}\sum_{i=1}^{n} x_{i1}^2 - \beta_2 \frac{1}{n}\sum_{i=1}^{n} x_{i1}x_{i2} := 0$$

But since

$$\sum_{i=1}^{n} X_{i1}X_{i2} = 0$$

$$\implies \frac{1}{n}\sum_{i=1}^{n} x_{i1}y_i - \beta_1 \frac{1}{n}\sum_{i=1}^{n} x_{i1}^2 = 0$$

$$\implies \hat{\beta}_1 = \frac{\frac{1}{n}\sum_{i=1}^{n} x_{i1}y_i}{\frac{1}{n}\sum_{i=1}^{n} x_{i1}^2} = \frac{\sum_{i=1}^{n} x_{i1}y_i}{\sum_{i=1}^{n} x_{i1}^2}$$

A similar derivation for $\hat{\beta}_2$ yields

$$\implies \hat{\beta}_2 = \frac{\sum_{i=1}^{n} x_{i2}y_i}{\sum_{i=1}^{n} x_{i2}^2}$$

So these are exactly the same least square estimates as before. ■

Question 3

Load the stackloss data:

```
data("stackloss") # load data
names(stackloss) # display names
```

```
## [1] "Air.Flow"   "Water.Temp" "Acid.Conc." "stack.loss"
```

The `stackloss` data is a data frame with 21 observations on 4 variables.

```
head(stackloss, 10)
```

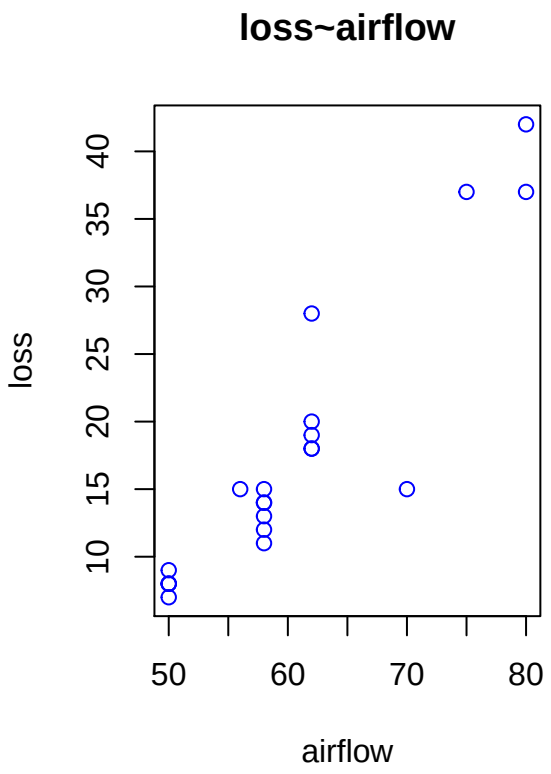
```
##    Air.Flow Water.Temp Acid.Conc. stack.loss
## 1        80         27         89         42
## 2        80         27         88         37
## 3        75         25         90         37
## 4        62         24         87         28
## 5        62         22         87         18
## 6        62         23         87         18
## 7        62         24         93         19
## 8        62         24         93         20
## 9        58         23         87         15
## 10       58         18         80         14
```

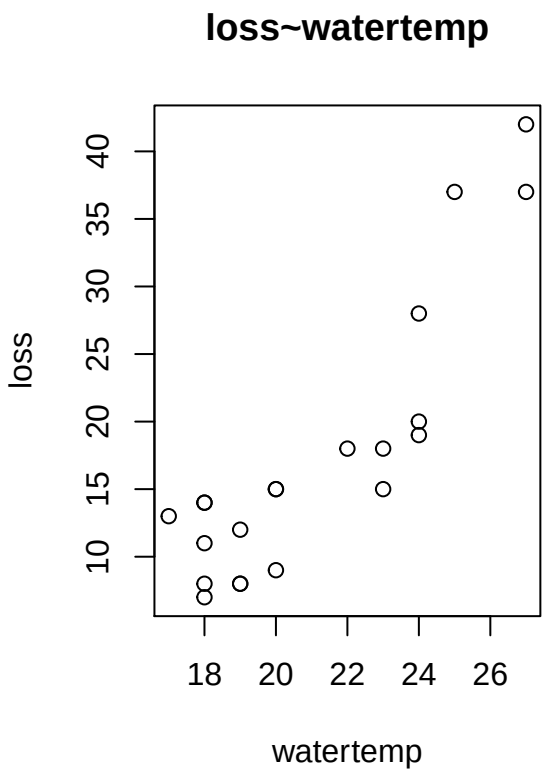
1. Plot the data

We will first generate three different plots for each of the predictors and the response variable.

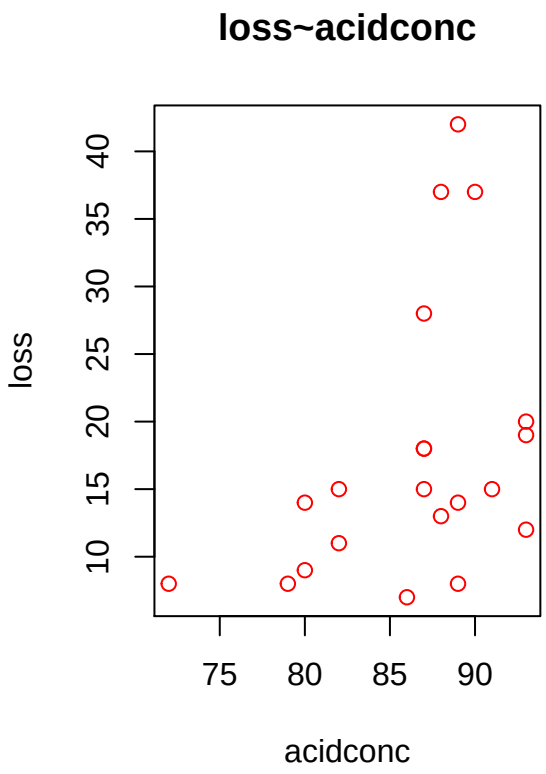
```
# store the variables in simpler names
airflow <- stackloss$Air.Flow
watertemp <- stackloss$Water.Temp
acidconc <- stackloss$Acid.Conc.
loss <- stackloss$stack.loss

# plot the different predictors against the response
par(mfrow= c(1,2))
plot(airflow, loss, pch = 21, main="loss~airflow", col='blue' )
plot(watertemp, loss, pch = 21, main="loss~watertemp", col='black')
```



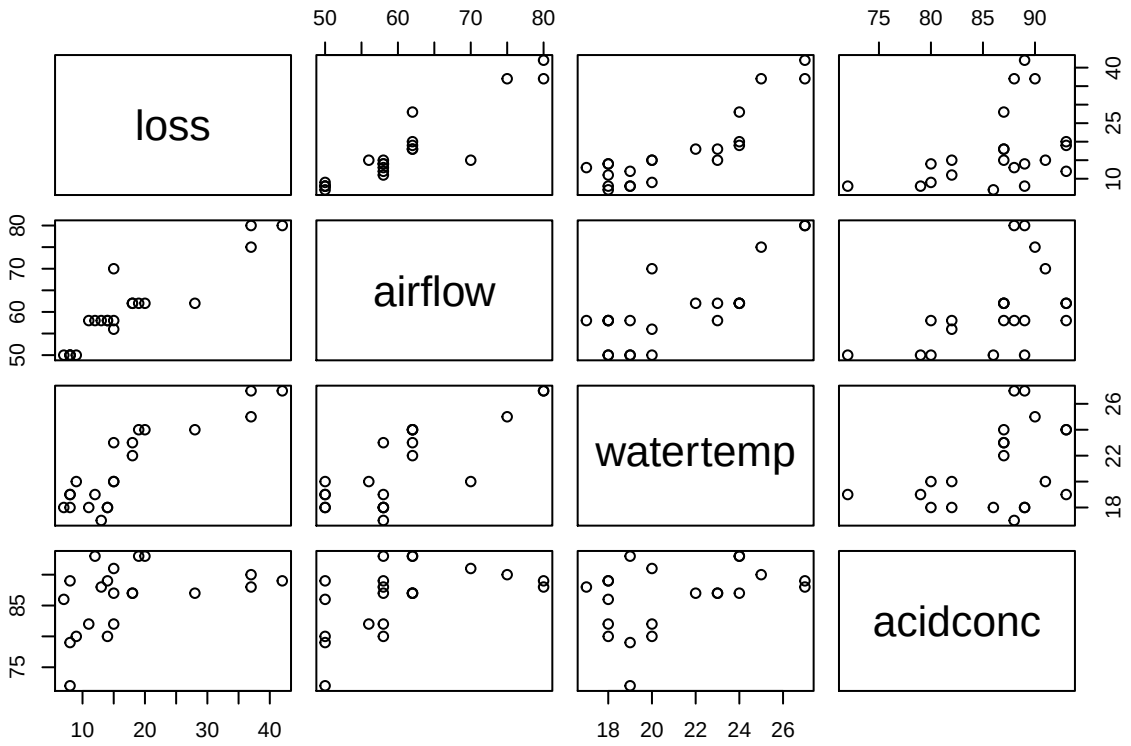



```
plot(acidconc, loss, pch = 21, main="loss~acidconc", col='red')
```



Now we produce a pairplot to visualize all variables at the same time.

```
par(mfrow = c(1,1))
pairs(cbind(loss, airflow, watertemp, acidconc))
```



2. Fitting the model

Letting

- X_1 = Air flow
- X_2 = Water temperature
- X_3 = Acid concentration
- Y = stack loss

We fit the model for the real model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

```
# fit the model & otput summary
fit.stackloss <- lm(loss ~ airflow + watertemp + acidconc)
summary(fit.stackloss)
```

```
##
## Call:
## lm(formula = loss ~ airflow + watertemp + acidconc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -7.2377 -1.7117 -0.4551  2.3614  5.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
## airflow       0.7156     0.1349   5.307  5.8e-05 ***
## watertemp     1.2953     0.3680   3.520  0.00263 **
## acidconc     -0.1521     0.1563  -0.973  0.34405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

We observe from the t-statistics of each of the betas that all , except for acidconc have a p-value less than $\alpha = 0.5$, which indicates that they are stastistically significant.

Confidence interval for model coefficients

Given that

$$\widehat{Var}(\hat{\beta}) = \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$$

Then, for each β_j, $j = 0, 1, \ldots, k$, we have that

$$ese(\hat{\beta}_j) = \left[\hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}$$

Then a $100(1-\alpha)\%$ **confidence interval** for β_j is given by

$$C.I.(\beta_j) = \hat{\beta}_j \pm t_{\alpha/2,n-p}\sqrt{\left[\hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}}$$

In order to find these for each of the coefficients of the model, with $\alpha = 0.1$, we find first find the point $t_{\alpha/2,n-p} = t_{0.5,n-p}$, and the procede to use the ese's in the formula. We obtain the following confidence intervals:

```
# (t_alpha_half <- qt(.95, 49)) # .95th quantile
(confint(fit.stackloss, level = 0.90))
```

```
##                    5 %        95 %
## (Intercept) -60.6140306 -19.2253183
## airflow       0.4810400   0.9502404
## watertemp     0.6550686   1.9355036
## acidconc     -0.4240127   0.1197676
```

Confidence prediction interval

Given a multiple linear regression model, for a new point $x_0 = [1, x_{01}, \ldots, x_{0k}] \in \mathbb{R}^{1xp}$, the fitted value is

$$\hat{m}(x_0) = x_0^T \hat{\beta} = \begin{bmatrix} 1 & x_{01} & \dots & x_{0k} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix}$$

For this , we have

$$\mathbb{E}[\hat{m}(x_0)] = x_0^T \beta$$

$$\mathbb{V}[\hat{m}(x_0)] = \sigma^2 x_0 (\mathbf{X}^T \mathbf{X})^{-1} x_0^T$$

A **prediction interval** for a new observation is given by

$$P.I.(y_0) = \hat{m}(x_0) \pm t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 x_0 (\mathbf{X}^T \mathbf{X})^{-1} x_0^T}$$

Now we want to construct a prediction interval for a new observation when `Airflow = 58`, `Water temperature = 20` and `Acid = 86`.

```
y_new <- data.frame(airflow=58, watertemp=20, acidconc=86)
predict(fit.stackloss, newdata = y_new, interval = "prediction", level = 0.99)
```

```
##        fit      lwr      upr
## 1 14.41064 4.759959 24.06133
```

Thefore predict for the input values a stackloss of 14.41064, with a prediction interval of [4.759959,24.06133].

Hypothesis test

We want to test for

$$\begin{cases} H_0 : \beta_3 = 0 \\ H_1 : \beta_3 \neq 0 \end{cases}$$

with the statistic (under H_0)

$$T_3 = \frac{\hat{\beta}_3 - \beta_3}{ese(\hat{\beta}_3)} = \frac{\hat{\beta}_3}{[\hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}]_{33}} \sim t_{n-2}$$

and under the normal noise assumption, $T_3 \sim t_{n-p}$, and we **reject**H_0 if

$$|T_j| > t_{\alpha/2, n-p} \iff \mathbb{P}(|T| > |T_j|) \equiv p - value < \alpha$$

Here, p=17, and $\alpha = 0.10$, so the $t_{0.05,17}$ quantile is

```
(t_alpha_half <- qt(.95, 17)) # .95th quantile
```

```
## [1] 1.739607
```

From the model summary, we see that $T_3 = -0.973 \implies |T_3| = 0.973 < t_{0.05,17}$, and p-val= 0.34405 > 0.1, so in both cases we **fail to reject** H_0, which indicates that the coefficient might not be significant on its own for the model.

Question 4

Load the data

```
data("ChickWeight")
names(ChickWeight)
```

```
## [1] "weight" "Time"   "Chick"  "Diet"
```

```
attach(ChickWeight)
```

Quick look of the data:

```
head(ChickWeight, 10)
```

```
##    weight Time Chick Diet
## 1      42    0     1    1
## 2      51    2     1    1
## 3      59    4     1    1
## 4      64    6     1    1
## 5      76    8     1    1
## 6      93   10     1    1
## 7     106   12     1    1
## 8     125   14     1    1
## 9     149   16     1    1
## 10    171   18     1    1
```

```
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   578 obs. of  4 variables
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

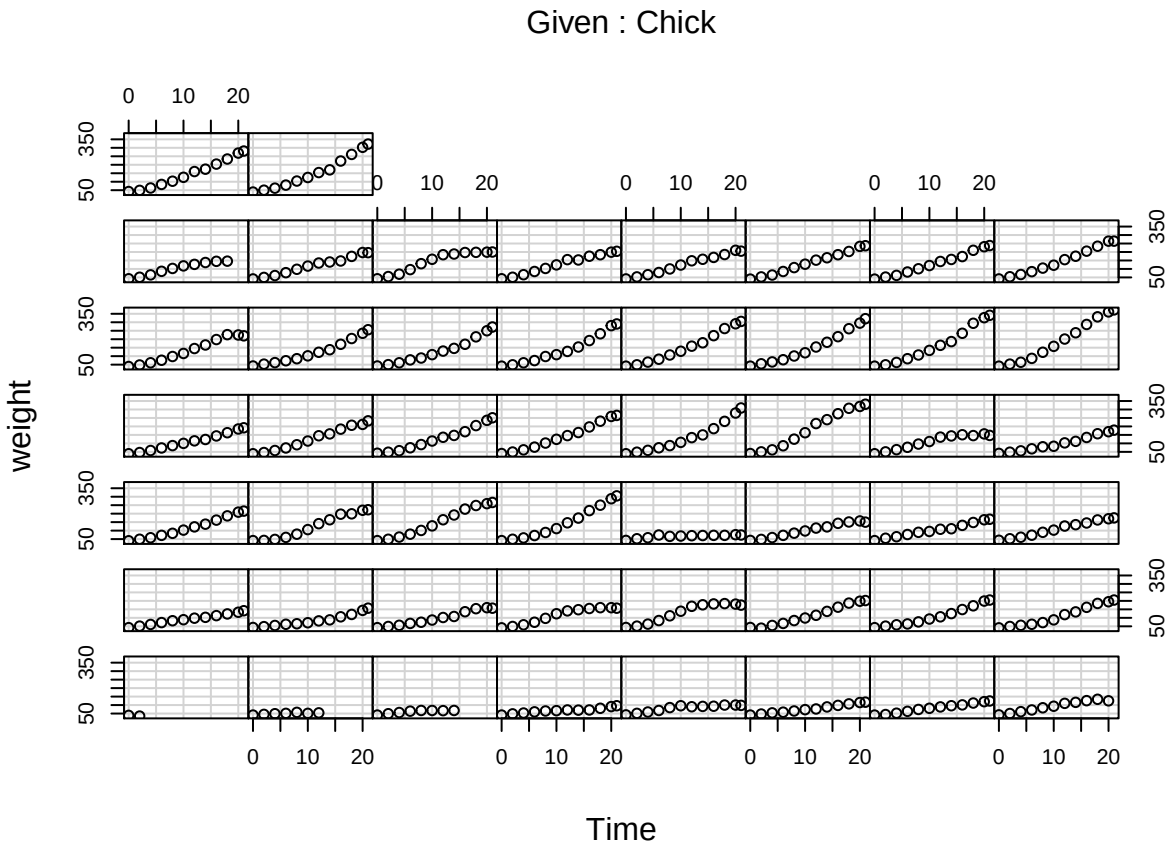
From the summary, we observe that the data contains 578 observations with two numeric values (`weight` and `Time`), as well as an ordinal or perhaps categorical factor `Chick` and a categorical factor `Diet` with four levels: i.e., four different kinds of feed.

Looking more closely, we see that these are in fact measures for 50 chicks which were taken at different times, and to which different diets were provided, and then the weights recorded.

(1) Plotting the data

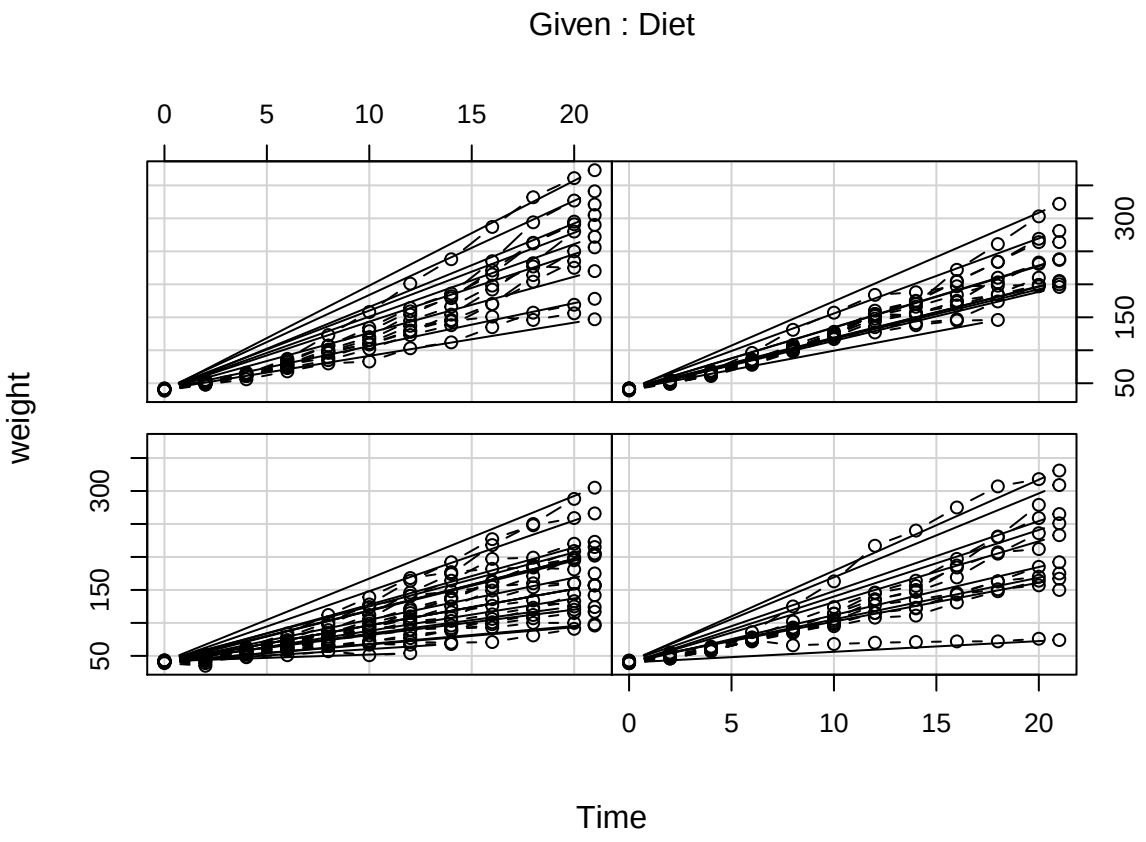
We can first plot the weight of each chick as a function of time:

```
coplot(weight ~ Time | Chick, data = ChickWeight, type = "b",
show.given = FALSE)
```



We can also plot simple lienar regression plots for different diets and the different chicks. We see that most of them seem to show a positive correlation.

```
coplot(weight ~ Time | Diet, data = ChickWeight, type = "b",
show.given = FALSE)
```



(2) Fitting data from an observation.

We want to extract the data corresponding to the sixth chick and fit a linear model using only time to predict the weight of the chick.

```
# extract data from the 6th chick
chick6 <- ChickWeight[ChickWeight$Chick == 6, ]
print(chick6)
```

```
##    weight Time Chick Diet
## 61     41    0     6    1
## 62     49    2     6    1
## 63     59    4     6    1
## 64     74    6     6    1
## 65     97    8     6    1
## 66    124   10     6    1
## 67    141   12     6    1
## 68    148   14     6    1
## 69    155   16     6    1
## 70    160   18     6    1
## 71    160   20     6    1
## 72    157   21     6    1
```

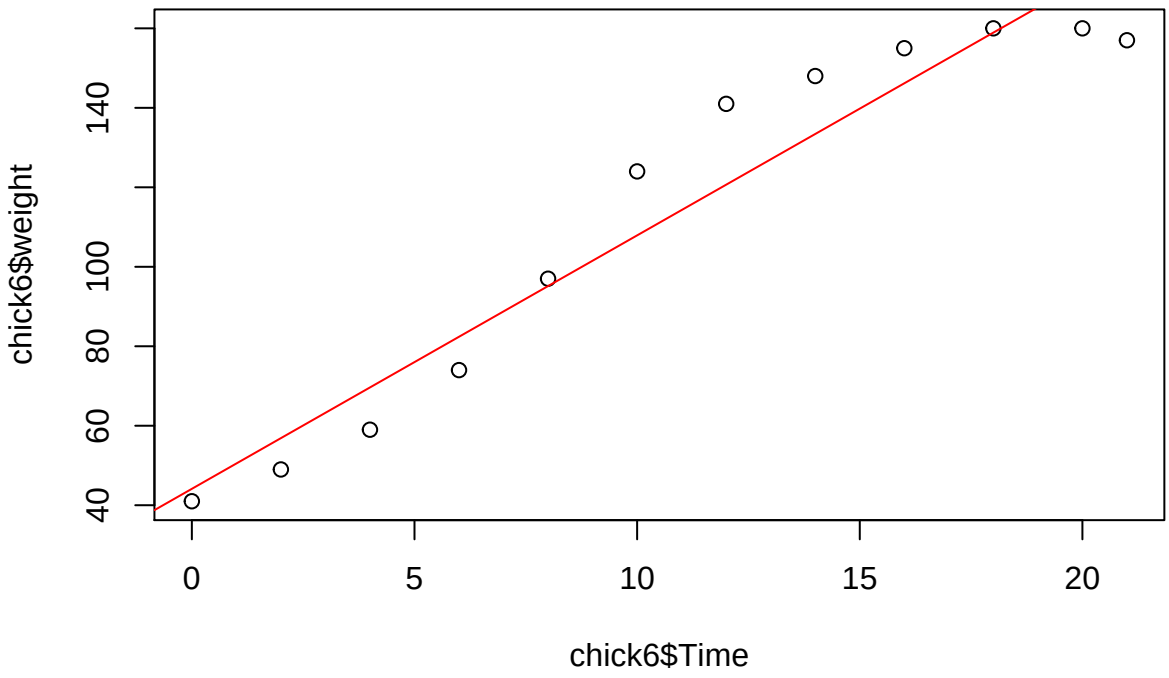
Now we will fit a model using `Time` only.

```r
fit.chick6 <- lm(weight ~Time, data = chick6)
summary(fit.chick6)
```

```
##
## Call:
## lm(formula = weight ~ Time, data = chick6)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.062  -8.953  -1.026  10.268  20.340
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  44.1234     7.3514   6.002 0.000132 ***
## Time          6.3780     0.5722  11.147 5.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.43 on 10 degrees of freedom
## Multiple R-squared:  0.9255, Adjusted R-squared:  0.9181
## F-statistic: 124.3 on 1 and 10 DF,  p-value: 5.825e-07
```

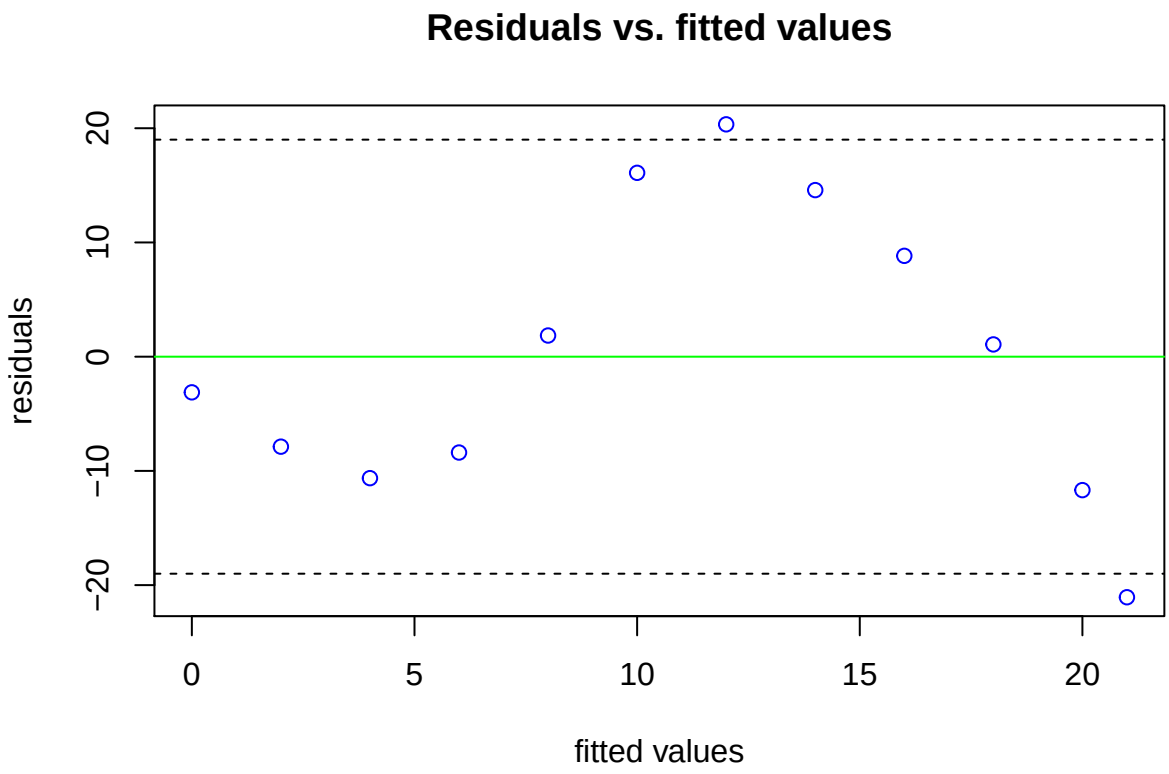
We observe that both coefficients are significant, the F-staistic also outputs a significant p-value, and the Adjisted R-squared is an indication of the amount of variance explained by the model, i.e.,The model seems to fit quite adequately, however from plotting (below) it is evident that process generating the data might not be linear.

```r
plot(chick6$Time, chick6$weight)
abline(coef(fit.chick6), col="red")
```



We can now check the residuals vs. time:

```
res.chick6 <- residuals(fit.chick6) # extract the residuals
# fitted.chick6 <- fit.chick6$fitted.values # extract fitted values
# plot the residuals vs fitted values
plot(chick6$Time , res.chick6 , xlab="fitted values", ylab = "residuals", main="Residuals vs. fitted va
abline(h=0, col="green")
abline(h=c(-19,19), lty=2)
```



Indeed, we observe that the linearity assumption might not be repected.

Fitting a polynomial model

We will now fit a polynomial model and see whether the fit improves.

```
fit.chick6poly <- lm(weight ~ Time + I(Time^2) + I(Time^2), data = chick6)
summary(fit.chick6poly)
```

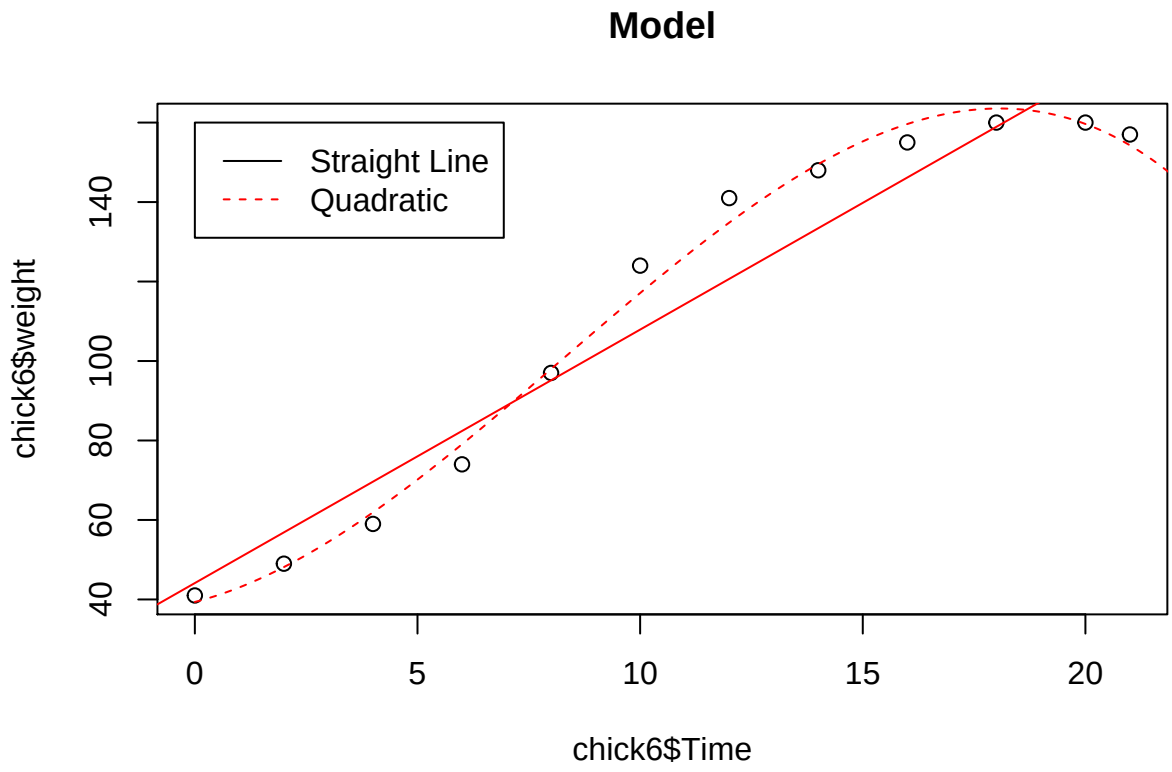
```
##
## Call:
## lm(formula = weight ~ Time + I(Time^2) + I(Time^2), data = chick6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8885  -7.1549   0.9567   5.6730  13.1453
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.85468    6.99297   3.983  0.00319 **
## Time        11.41146    1.52774   7.470 3.81e-05 ***
## I(Time^2)   -0.23430    0.06866  -3.413  0.00772 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.346 on 9 degrees of freedom
## Multiple R-squared:  0.9675, Adjusted R-squared:  0.9603
```

```
## F-statistic: 134.1 on 2 and 9 DF,  p-value: 2.002e-07
```

We observe that all the coefficients have significant p-values and the adjusted R-square also improved greatly (from ~0.92 to ~0.96). The F-test also yields a p-value $< \alpha = 0.5$, say.

The model fit can be observed below:

```
fit.chick6poly <- lm(weight ~ Time + I(Time^2) + I(Time^3), data = chick6)
# Fitted values for polynomial regression
xp <- seq(0,22, by=0.1) # generate a seq of numbers
polyfit <- cbind(rep(1,length(xp)), xp, xp^2, xp^3) %*% coef(fit.chick6poly)
plot(chick6$Time, chick6$weight)
abline(coef(fit.chick6), col="red")
 # simple linear regression
lines(xp, polyfit, col="red", lty=2) # poly regression model
legend(0,160,c("Straight Line","Quadratic"),lty=c(1,2),col=c("black","red"))
title("Model")
```



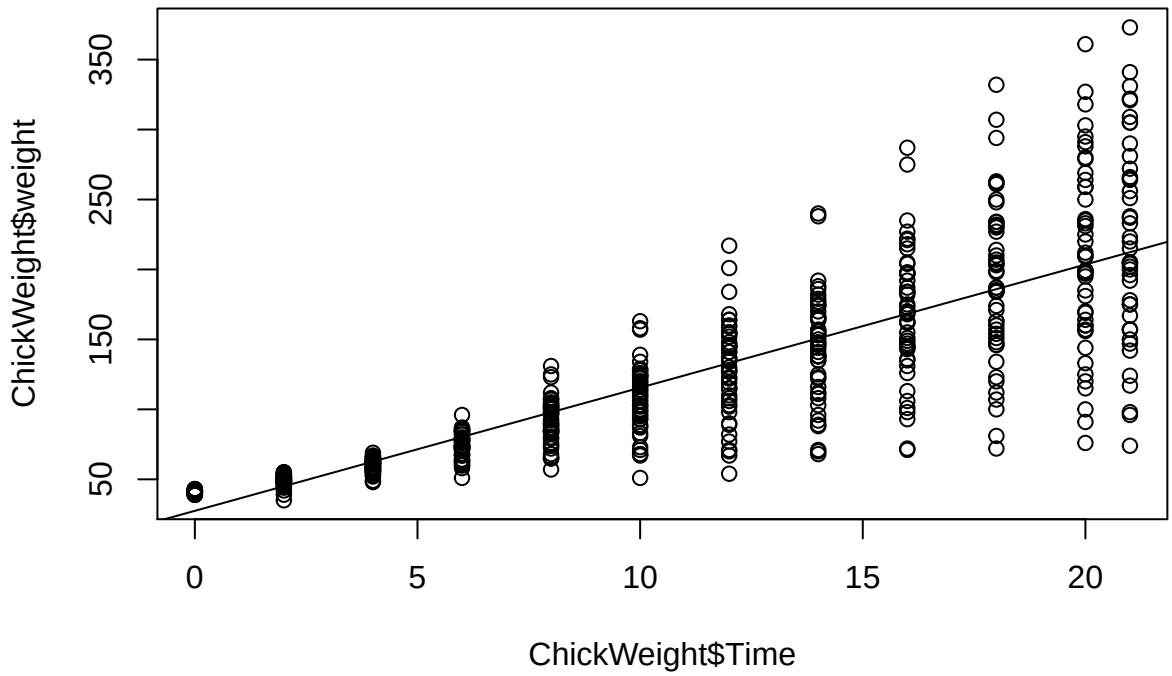
(3) Using all Weight data.

We will now use the data for all chicks, trying to predict weight from time.

```
fit.chickweight <- lm(weight ~ Time, data = ChickWeight)
summary(fit.chickweight)
```

```
##
## Call:
## lm(formula = weight ~ Time, data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -138.331  -14.536    0.926   13.533  160.669
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.4674     3.0365   9.046   <2e-16 ***
## Time          8.8030     0.2397  36.725   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.91 on 576 degrees of freedom
## Multiple R-squared:  0.7007, Adjusted R-squared:  0.7002
## F-statistic:  1349 on 1 and 576 DF,  p-value: < 2.2e-16
```

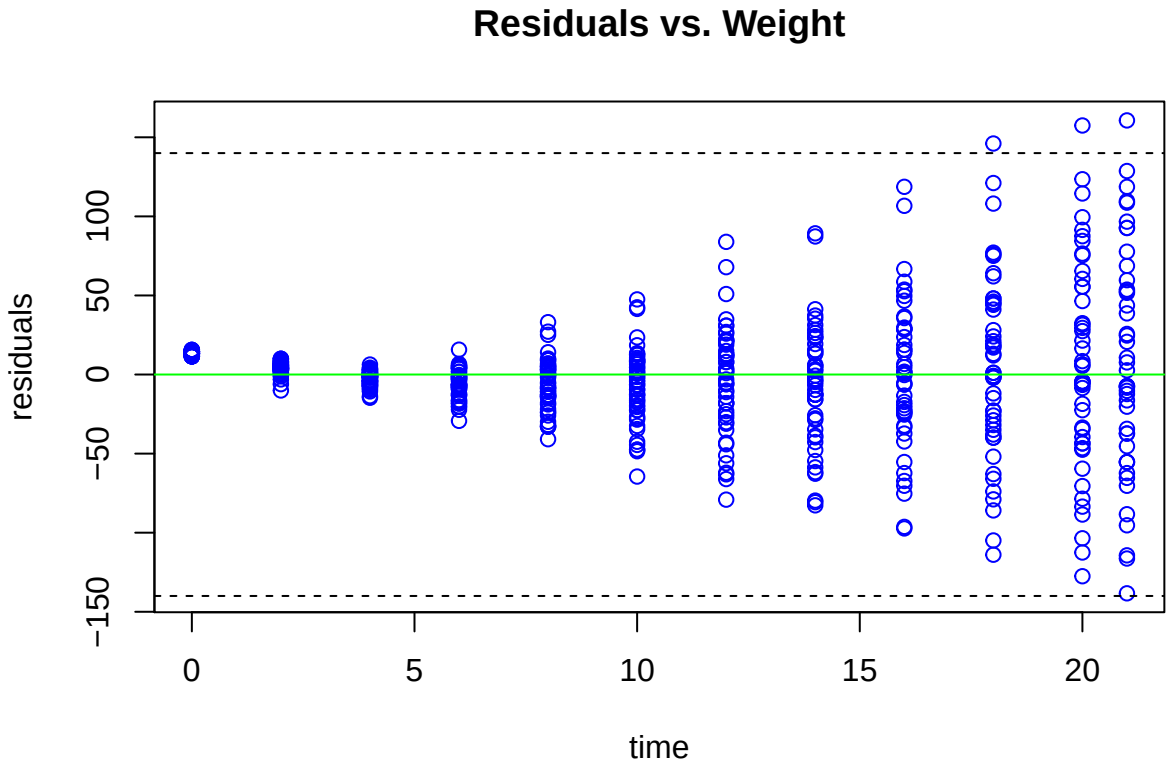
```
plot(ChickWeight$Time, ChickWeight$weight)
abline(coef(fit.chickweight))
```



We observe immediately that the model has poorer fit than the model with data for only one chick. From plotting the fit, we see that the weight variation becomes bigger as time increases.

We can verify the residuals:

```
res.chickweights <- residuals(fit.chickweight) # extract the residuals
# fitted.chick6 <- fit.chick6$fitted.values # extract fitted values
# plot the residuals vs fitted values
plot(ChickWeight$Time , res.chickweights , xlab="time", ylab = "residuals", main="Residuals vs. Weight"
abline(h=0, col="green")
abline(h=c(-140,140), lty=2)
```



Indeed, we observe that the variance increases as time increases. We will now attempt to fit a polynomial function here too and see if it helps.

```
fit.chickweightspoly <- lm(weight ~ Time + I(Time^2) , data = ChickWeight)
summary(fit.chickweightspoly)
```

```
##
## Call:
## lm(formula = weight ~ Time + I(Time^2), data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -147.952  -12.507    0.518   11.126  151.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.13394    4.08288   9.340  < 2e-16 ***
## Time         5.45963    0.89962   6.069 2.34e-09 ***
```

```
## I(Time^2)    0.15684    0.04071   3.852  0.00013 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.45 on 575 degrees of freedom
## Multiple R-squared:  0.7083, Adjusted R-squared:  0.7073
## F-statistic:   698 on 2 and 575 DF,  p-value: < 2.2e-16
```

```
fit.chickweightspoly <- lm(weight ~ Time + I(Time^2) + I(Time^3) , data = ChickWeight)
summary(fit.chickweightspoly)
```

```
##
## Call:
## lm(formula = weight ~ Time + I(Time^2) + I(Time^3), data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -145.434  -12.197    0.107   11.829  153.566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.893364   4.791190   8.535   <2e-16 ***
## Time         3.386492   2.088136   1.622    0.105
## I(Time^2)    0.415158   0.238316   1.742    0.082 .
## I(Time^3)   -0.008170   0.007426  -1.100    0.272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.45 on 574 degrees of freedom
## Multiple R-squared:  0.7089, Adjusted R-squared:  0.7074
## F-statistic: 465.9 on 3 and 574 DF,  p-value: < 2.2e-16
```

```
fit.chickweightspoly <- lm(weight ~ Time + I(Time^2) + I(Time^4) , data = ChickWeight)
summary(fit.chickweightspoly)
```

```
##
## Call:
## lm(formula = weight ~ Time + I(Time^2) + I(Time^4), data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -145.184  -11.970    0.416   12.052  153.816
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.5838954  4.6564128   8.716   <2e-16 ***
## Time         3.8562997  1.7199234   2.242   0.0253 *
## I(Time^2)    0.3062767  0.1425714   2.148   0.0321 *
## I(Time^4)   -0.0001926  0.0001761  -1.094   0.2745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.45 on 574 degrees of freedom
```

```
## Multiple R-squared:  0.7089, Adjusted R-squared:  0.7074
## F-statistic: 465.9 on 3 and 574 DF,  p-value: < 2.2e-16
```

We see that even after fittinga 4th degree polynomial, we cannot improve the fit. This is because , as we saw before, the variance increases more and more as time decreases, so the range of the weight becomes wider , and thus a polynomial fit will not really help in this case.

Repeating the anlysis including Diet

Now we will include the `Diet` categorical variable and fit the model again.

```
# tranform the diet variable as a factor
diet_categ <- as.factor(ChickWeight$Diet)

# fit the model with Time and Diet
#  Note that R automatically encoded the Diet variable as categorical
fit.chickweightall <- lm(weight ~Time + Diet, data=ChickWeight)
summary(fit.chickweightall)
```

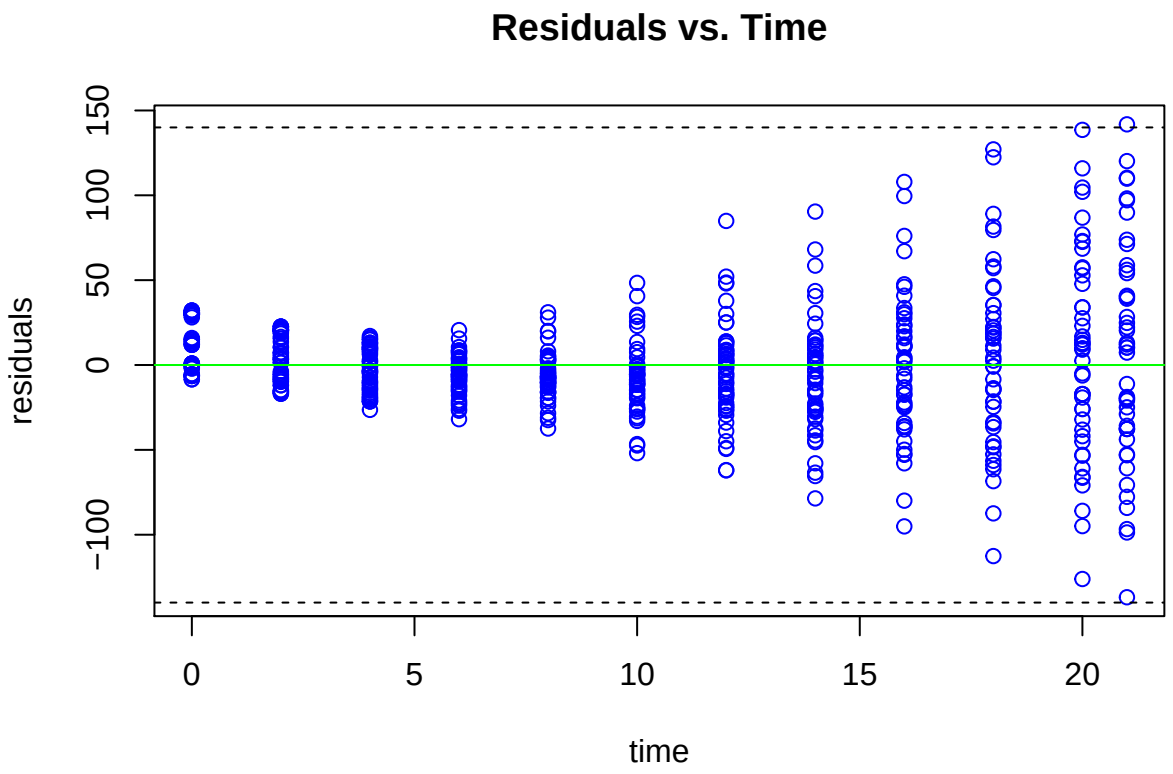
```
##
## Call:
## lm(formula = weight ~ Time + Diet, data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -136.851  -17.151   -2.595   15.033  141.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.9244     3.3607   3.251  0.00122 **
## Time          8.7505     0.2218  39.451  < 2e-16 ***
## Diet2        16.1661     4.0858   3.957 8.56e-05 ***
## Diet3        36.4994     4.0858   8.933  < 2e-16 ***
## Diet4        30.2335     4.1075   7.361 6.39e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.99 on 573 degrees of freedom
## Multiple R-squared:  0.7453, Adjusted R-squared:  0.7435
## F-statistic: 419.2 on 4 and 573 DF,  p-value: < 2.2e-16
```

We observe that by fitting the `Diet` variable, we increase the quality of the model, obtaining a higher Adjusted R-Square than before. This, along with the significant ($<\alpha = 0.5$) p-values indicate that the inclusion of the categorical values are statistically significant contributions to the model.

We now proceed to plot the residuals to check the model assumptions.

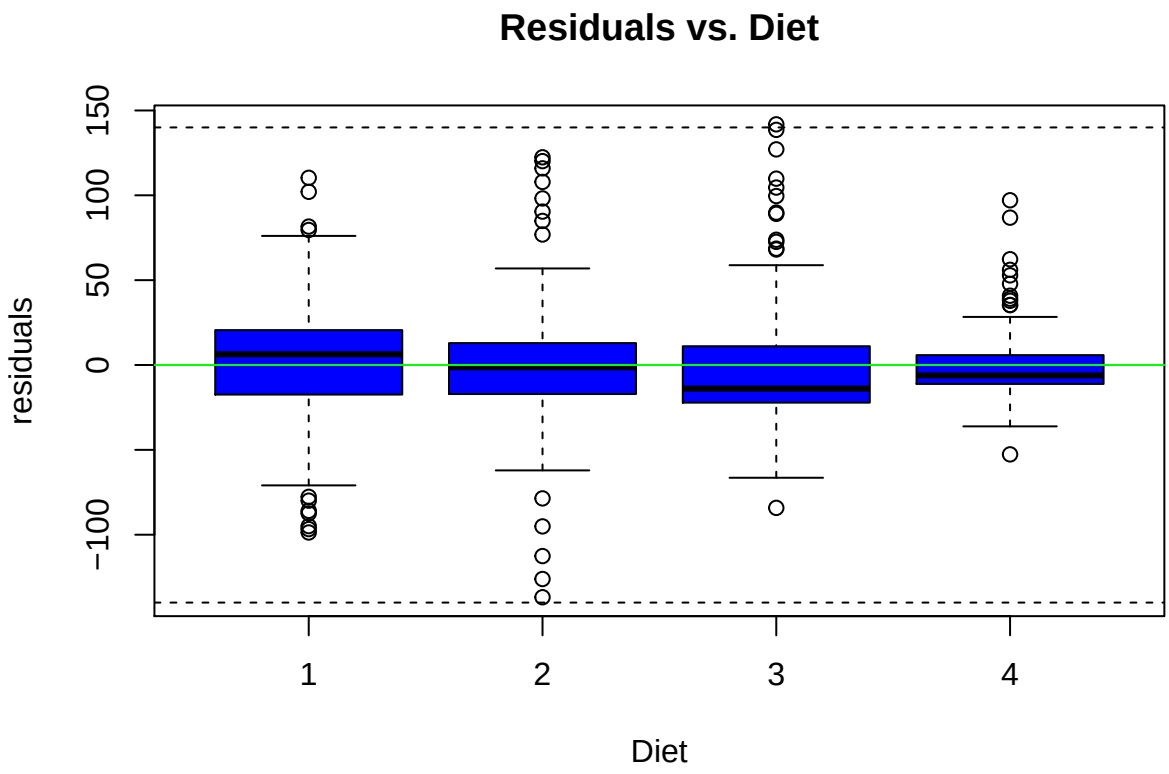
```
res.chickweightsall <- residuals(fit.chickweightall)

# Plot of the Residuals vs. the Time predictor
plot(ChickWeight$Time , res.chickweightsall , xlab="time", ylab = "residuals", main="Residuals vs. Time
abline(h=0, col="green")
abline(h=c(-140,140), lty=2)
```



Here we observe that althought the distribution seems to be centered around 0, the model's variance seems to increase as Time increases, just as we determined before. This is an indication that the constant variance assumption is not respected. This could be improved by applying a transformation.

```
# Plot if the Residuals vs. the Diet
plot(ChickWeight$Diet , res.chickweightsall , xlab="Diet", ylab = "residuals", main="Residuals vs. Diet
abline(h=0, col="green")
abline(h=c(-140,140), lty=2)
```



From the boxplots we can also observe that although all the groups except for Diet3 seem to have a distribution centered around 0 and approximatedly bell shape (judging from the whiskers), there also seems to be a significant amount of outliers. In particular for group 3, the normal distribution assumption might not be respected. In general, the residuals also seem to show non-constant variance across diets.

Question 5

We will work with data that relate to a study of 25 cigarrete brands: in the dataset, for each brand,

- X_1 is the tar content (mg), denoted `TAR`;
- X_2 is the nicotine content (mg), denoted `NICOTINE`;
- X_3 is the weight (g), denoted `WEIGHT`;
- Y is the amount of Carbon Monoxide (mg) produced in a standarized volume, denoted `CD`.

```
library(readr)
cigs <- read_csv("C:/Users/jairp/Desktop/BackUP/McGill-20180719T015111Z-001/McGill/7. Fall 2019/MATH 423
```

```
## Parsed with column specification:
## cols(
##   TAR = col_double(),
##   NICOTINE = col_double(),
##   WEIGHT = col_double(),
##   CO = col_double()
## )
```

```r
head(cigs, 10)
```

```
## # A tibble: 10 x 4
##      TAR NICOTINE WEIGHT    CO
##    <dbl>    <dbl>  <dbl> <dbl>
##  1  14.1     0.86  0.985  13.6
##  2  16       1.06  1.09   16.6
##  3  29.8     2.03  1.16   23.5
##  4   8       0.67  0.928  10.2
##  5   4.1     0.4   0.946   5.4
##  6  15       1.04  0.888  15
##  7   8.8     0.76  1.03    9
##  8  12.4     0.95  0.922  12.3
##  9  16.6     1.12  0.937  16.3
## 10  14.9     1.02  0.886  15.4
```

Regression models constructed to study the ability of the predictors to capture the variation in response are predicted. The most complex model is the multiple regression model

$$\mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

The fitting of this model is

```r
# fit the model and output the summary
fit.cig_full <- lm(CO ~ TAR + NICOTINE + WEIGHT, data=cigs)
summary(fit.cig_full)
```

```
##
## Call:
## lm(formula = CO ~ TAR + NICOTINE + WEIGHT, data = cigs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89261 -0.78269  0.00428  0.92891  2.45082
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2022     3.4618   0.925 0.365464
## TAR           0.9626     0.2422   3.974 0.000692 ***
## NICOTINE     -2.6317     3.9006  -0.675 0.507234
## WEIGHT       -0.1305     3.8853  -0.034 0.973527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.446 on 21 degrees of freedom
## Multiple R-squared:  0.9186, Adjusted R-squared:  0.907
## F-statistic: 78.98 on 3 and 21 DF,  p-value: 1.329e-11
```

All models to be considered are nested within this one. We will compute the following F-statistics:

1. The F-statistic for comparing the two models:

$$\begin{cases} H_0 : \mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \\ \\ H_1 : \mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \end{cases}$$

Which in this case is equivalent to

$$\begin{cases} H_0 : \beta_3 = 0 \\ \\ H_1 : \beta_3 \neq 0 \end{cases}$$

Which can easily be tested as follows

```
# drop X3
fit.cig_reduced <- lm(CO ~ TAR + NICOTINE, data=cigs)
anova(fit.cig_reduced, fit.cig_full) # F-test: reduced vs full
```

```
## Analysis of Variance Table
##
## Model 1: CO ~ TAR + NICOTINE
## Model 2: CO ~ TAR + NICOTINE + WEIGHT
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     22 43.895
## 2     21 43.893  1 0.0023573 0.0011 0.9735
```

We see that

$$F_1 = \frac{\overline{SS_R}(\beta_3|\beta_0,\beta_1,\beta_2)/1}{MS_{Res}} = \frac{\overline{SS_R}(\beta_3|\beta_0,\beta_1,\beta_2)/1}{SS_{Res}(\beta_0,\beta_1,\beta_2,\beta_3)/(n-4)}$$

$$= \frac{(\overline{SS_R}(\beta_0,\beta_1,\beta_2,\beta_3) - \overline{SS_R}(\beta_0,\beta_1,\beta_2))/1}{SS_{Res}(\beta_0,\beta_1,\beta_2,\beta_3)/(n-4)}$$

$$= \frac{(SS_{Res}(\beta_0,\beta_1,\beta_2) - SS_{Res}(\beta_0,\beta_1,\beta_2,\beta_3))/1}{SS_{Res}(\beta_0,\beta_1,\beta_2,\beta_3)/(n-4)}$$

$$\frac{(43.895 - 43.893)/1}{43.893/21}$$

$$= 0.0011$$

At the $\alpha = 0.1$ level of confidence, say, since p-val=0.9735, we **fail to reject** H_0, which is indicative that including `Weight` into the model is not statistically significant.

2. The F-statistic for comparing the two models:

$$\begin{cases} H_0 : \mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 \\ \\ H_1 : \mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \end{cases}$$

if it is known that the predictor X_3 is not included.

```r
m_01 <- lm(CO ~ TAR, data = cigs) # fit model under H0
m_012 <- lm(CO ~ TAR + NICOTINE, data = cigs )  # fit model under H1
anova(m_01,m_012)
```

```
## Analysis of Variance Table
##
## Model 1: CO ~ TAR
## Model 2: CO ~ TAR + NICOTINE
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     23 44.869
## 2     22 43.895  1   0.97415 0.4882  0.492
```

Once again, by observing that we get a p-value which is quite big, we fail to reject the null hypothesis and conclude that given that we didn't include X_3, the model $\mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2$ is statistically more appropriate that the one without X_2.

3. The F-statistic for comparing the two models:

$$\begin{cases} H_0 : \mathbb{E}[Y|X] = \beta_0 \\ \\ H_1 : \mathbb{E}[Y|X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \end{cases}$$

This time , we will perform a global F-test model given that we didn't include X_3 in the first place.

```r
m_012 <- lm(CO ~ TAR + NICOTINE, data = cigs )
summary(m_012)
```

```
##
## Call:
## lm(formula = CO ~ TAR + NICOTINE, data = cigs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89941 -0.78470 -0.00144  0.91585  2.43064
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0896     0.8438   3.662 0.001371 **
## TAR           0.9625     0.2367   4.067 0.000512 ***
## NICOTINE     -2.6463     3.7872  -0.699 0.492035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.413 on 22 degrees of freedom
## Multiple R-squared:  0.9186, Adjusted R-squared:  0.9112
## F-statistic: 124.1 on 2 and 22 DF,  p-value: 1.042e-12
```

In this case, we obtain a F statistic of 124.1, and in particular, we obtain a p-value of 1.042e-12 $< \alpha = 0.01$, say, and so we reject the null hypothesis that the intercept-only model is more adequate than the model including predictors X_1 and X_2.

Page 1

December 5, 2019 11:53 PM

Study questions

Study questions 3.2.1

Referring to Study question 1.5.2 (Figure 1.10) and the parameters listed therein,

(a) Compute P(y|do(x)) for all values of x and y, by simulating the intervention do(x) on the model.
(b) Compute P(y|do(x)) for all values of x and y, using the adjustment formula (3.5)
(c) Compute the ACE

$$ACE = P(y_1|do(x_1)) - P(y_1|do(x_0))$$

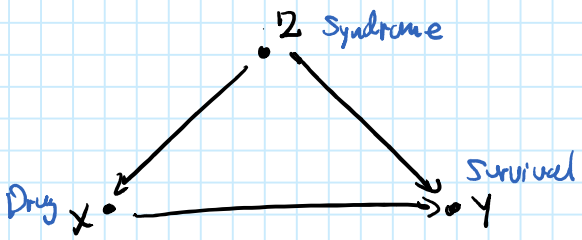
Population of patients contain fraction r of individuals who suffer from syndrome Z, $P(Z)=r$, which makes them uncomfortable to take a life-prolonging drug X.

$$Z = \begin{cases} 1, & \text{syndrome} \\ 0, & \text{absence of syndrome} \end{cases}$$

$$Y = \begin{cases} 1, & \text{death} \\ 0, & \text{survival} \end{cases} \qquad X = \begin{cases} 1, & \text{take the drug} \\ 0, & \text{not taking the drug} \end{cases}$$

Assume
- $P(Z=1) = r$
- $P(Y=1 \mid X=0, Z=0) = p_1$
- $P(Y=1 \mid X=1, Z=0) = p_2$
- $P(Y=1 \mid X=0, Z=1) = p_3$
- $P(Y=1 \mid X=1, Z=1) = p_4$
- $P(X=1 \mid Z=0) = q_1$
- $P(X=1 \mid Z=1) = q_2$

(b)



$$P(Y=y \mid do(X=x)) = \sum_z P(Y=y \mid X=x, Z=z) P(Z=z)$$

Want to estimate

$$P(Y=1 \mid do(X=1)) - P(Y=1 \mid do(X=0))$$

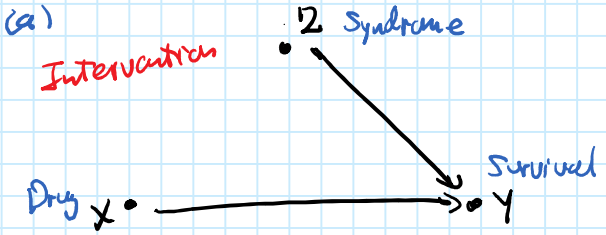
$$P(Y=1 \mid do(X=1)) = P(Y=1 \mid X=1, Z=1) P(Z=1) + P(Y=1 \mid X=1, Z=0) P(Z=0)$$
$$= (p_4)(r) + (p_2)(1-r)$$

$$P(Y=1 \mid do(X=0)) = P(Y=1 \mid X=0, Z=1) P(Z=1) + P(Y=1 \mid X=0, Z=0) P(Z=0)$$
$$= (p_3)(r) + (p_1)(1-r)$$

$$P(Y=0 \mid do(X=1)) = P(Y=0 \mid X=1, Z=1) P(Z=1) + P(Y=0 \mid X=1, Z=0) P(Z=0)$$
$$= (1-p_4)(r) + (1-p_2)(1-r)$$

$$P(Y=0 \mid do(X=0)) = P(Y=0 \mid X=0, Z=0) P(Z=0) + P(Y=0 \mid X=0, Z=1) P(Z=0)$$
$$= (1-p_3)(r) + (1-p_1)(1-r)$$

(a) Intervention



Note that from this model and by definition,

(1) $P_m(Z=z) = P(Z=z)$

(2) $P_m(Y=y \mid Z=z, X=x) = P_m(Y=y \mid Z=z) = P(Y=y \mid Z=z)$

(3) $P_m(Z=z \mid X=x) = P_m(Z=z) = P(Z=z)$

$$\therefore P(Y=y \mid do(X=x)) = P_m(Y=y \mid X=x) \quad \text{def}$$
$$= \sum_z P_m(Y=y \mid X=x, Z=z) P_m(Z=z \mid X=x) \quad \text{Bayes}$$
$$= \sum_z P_m(Y=y \mid X=x, Z=z) P_m(Z=z) \quad (2)(3)$$
$$= \sum_z P(Y=y \mid X=x, Z=z) P(Z=z) \quad (1)$$

So the values are the same as in (b)

Page 2

December 6, 2019 12:57 PM

(a) $ACE = P(Y=1 \mid do(X=1)) - P(Y=1 \mid do(X=0))$

$= (\rho_4)(r) + (\rho_2)(1-r) - (\rho_3)(r) - (\rho_1)(1-r)$

$= \boxed{r(\rho_4 - \rho_3) + (1-r)(\rho_2 - \rho_1)}$

Let $P(X=1) = a$

$\Rightarrow \boxed{P(Z=0 \mid X=1)} = \dfrac{P(Z=0)\,P(X=1 \mid Z=0)}{P(X=1)}$

$= \boxed{\dfrac{(1-r)(q_1)}{(a)}}$

$\boxed{P(Z=1 \mid X=1)} = \dfrac{P(Z=1)\,P(X=1 \mid Z=1)}{P(X=1)}$

$= \boxed{\dfrac{(r)(q_2)}{(1-a)}} := b$

$\boxed{P(Z=0 \mid X=0)} = \dfrac{P(Z=0)\,P(X=0 \mid Z=0)}{P(X=0)}$

$= \boxed{\dfrac{(1-r)(1-q_1)}{(1-a)}}$

$\boxed{P(Z=1 \mid X=0)} = \dfrac{P(Z=1)\,P(X=0 \mid Z=1)}{P(X=0)}$

$= \boxed{\dfrac{(r)(1-q_2)}{(1-a)}} := d$

$\boxed{a = P(X=1) = P(Z=1)P(X=1 \mid Z=1) + P(Z=0)P(X=1 \mid Z=0)}$

$\boxed{RD} = P(Y=1 \mid X=1) - P(Y=1 \mid X=0)$

$\boxed{P(Y=1 \mid X=1)} = P(Y=1 \mid X=1, Z=1)\,P(Z=1 \mid X=1) + P(Y=1 \mid X=1, Z=0)\,P(Z=0 \mid X=1)$

$= \boxed{(\rho_4)(b) + (\rho_2)(1-b)}$

$\boxed{P(Y=1 \mid X=0)} = P(Y=1 \mid X=0, Z=1)\,P(Z=1 \mid X=0) + P(Y=1 \mid X=0, Z=0)\,P(Z=1 \mid X=0)$

$= \boxed{(\rho_3)(b) + (\rho_1)(1-b)}$

$\boxed{RD} = P(Y=1 \mid X=1) - P(Y=1 \mid X=0)$

$= \boxed{(b)(\rho_4 - \rho_3) + (1-b)(\rho_2 - \rho_1)}$

But clearly, $\boxed{ACE \neq RD}$

We could minimize the difference with the parameters r, q_2, since these constitute the ratios appearing in both equations.

(d)