Math 533- Assignment#3

Aymen Rumi

12/7/2019

Question 1:

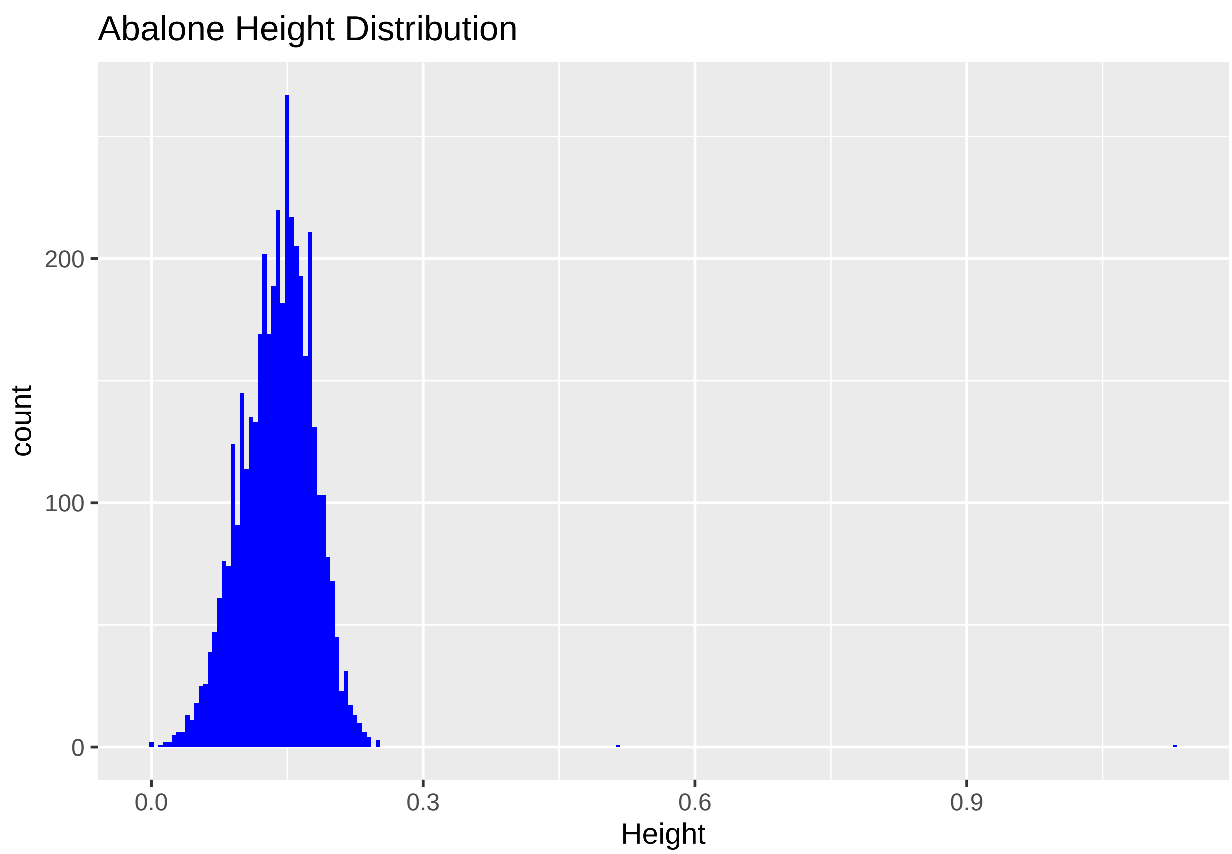
Hypothesis: I believe that a simple linear regression model with normal error assumption is appropriate to describe the relationship between the height of abalones and their ages, and particularly, that a larger height is associated with an older age, we will use data from `abalone.csv` to test this hypotheis

```
# importing data

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

Univariate Analysis: Height:

```
# function we will use
Summary_Table<-function(data,variable)
{
    data %>% summarise(Avg = mean(variable),
        Med = median(variable),
        Q25 = quantile(variable,0.25), Q75 = quantile(variable,0.75),
        StD = sd(variable), Var=var(variable), Min=min(variable),
        Max=max(variable))%>%kable()
}
```

```
#plotting distribution of Height

ggplot(abalone,aes(x=Height))+geom_bar(fill="blue")+
  scale_fill_viridis_d()+ggtitle("Abalone Height Distribution")
```

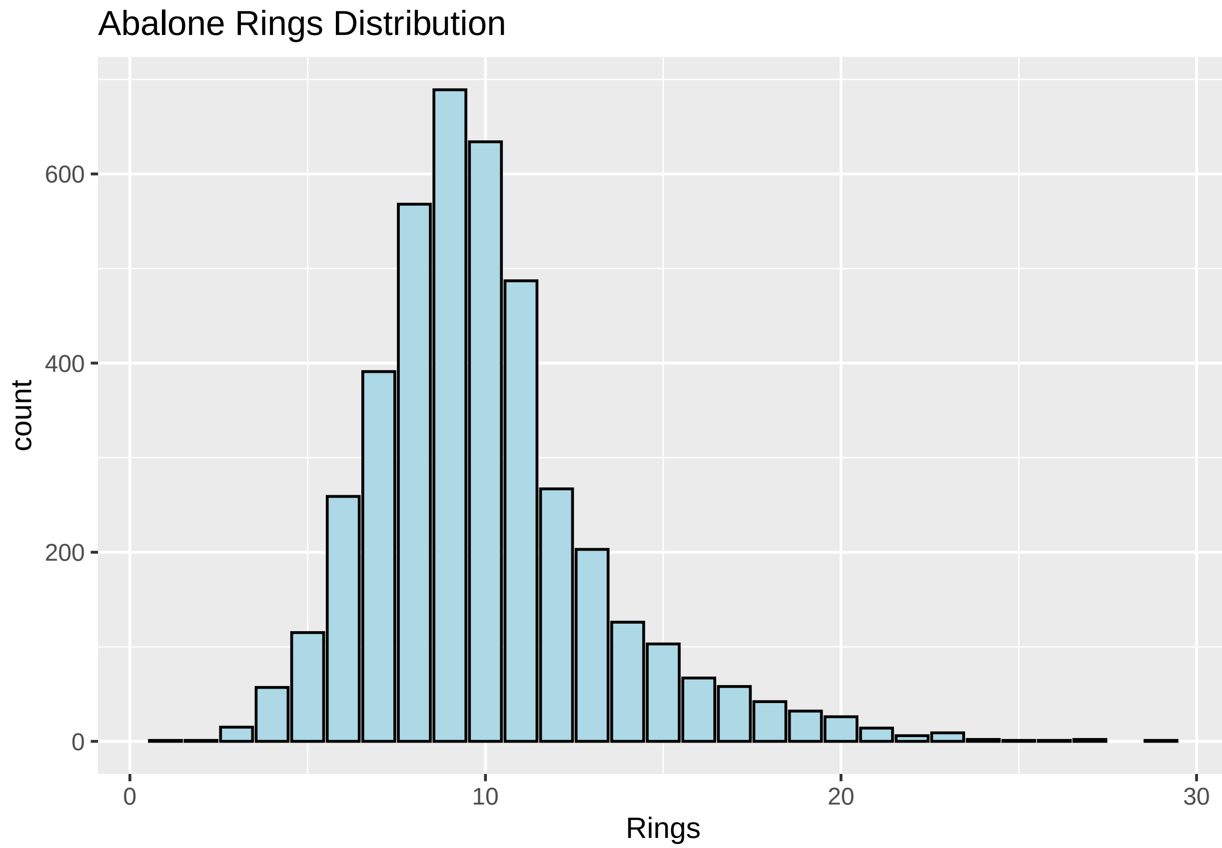



```r
#summary table of Height
Summary_Table(abalone,abalone$Height)
```

Avg	Med	Q25	Q75	StD	Var	Min	Max
0.1395164	0.14	0.115	0.165	0.0418271	0.0017495	0	1.13

Univariate Analysis: Rings:

```r
#plotting distribution of Rings

ggplot(abalone,aes(x=Rings))+geom_bar(fill="lightblue",color="black")+
  scale_fill_viridis_d()+ggtitle("Abalone Rings Distribution")
```

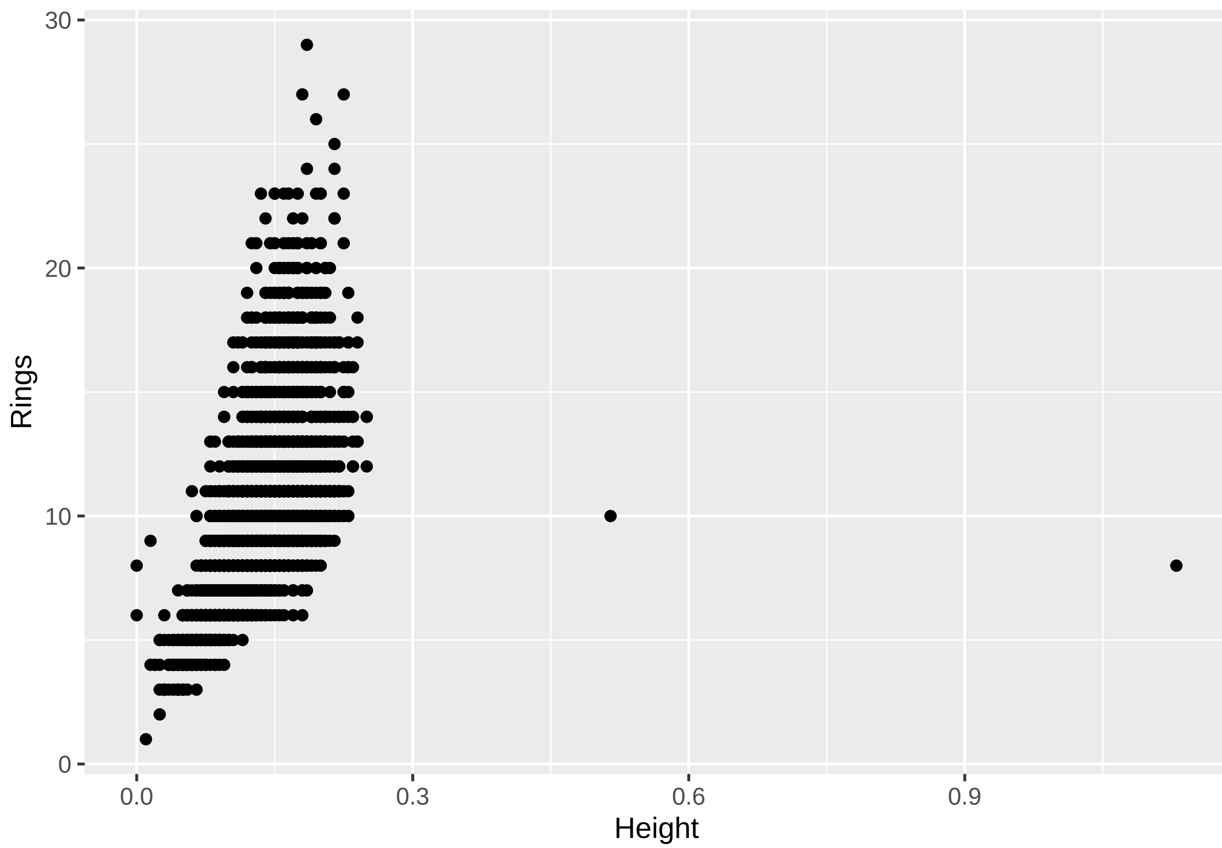



```
#summary table of Rings
Summary_Table(abalone,abalone$Rings)
```

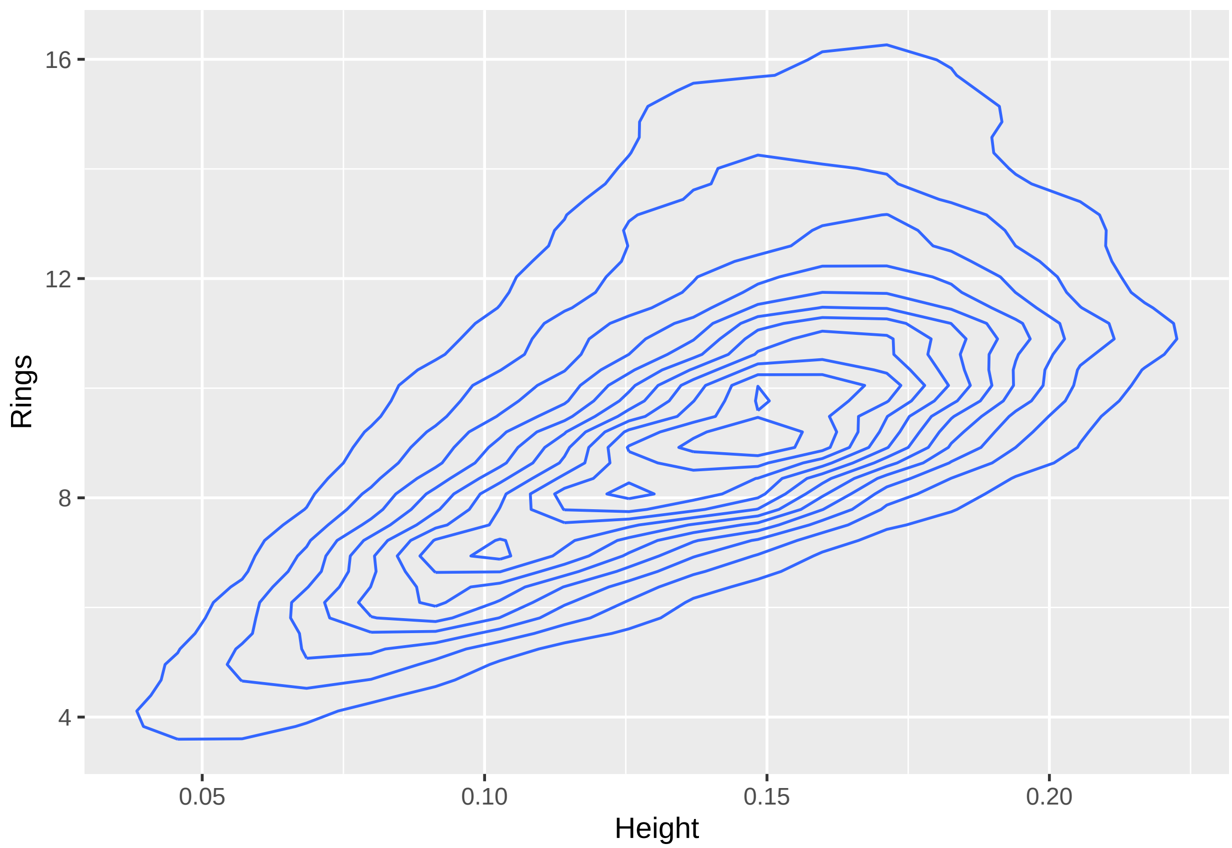
Avg	Med	Q25	Q75	StD	Var	Min	Max
9.933685	9	8	11	3.224169	10.39527	1	29

Data ScatterPlot & Other Visuals

```
#data visuals for Height vs Rings

ggplot(abalone,aes(x=Height,y=Rings))+geom_point()
```

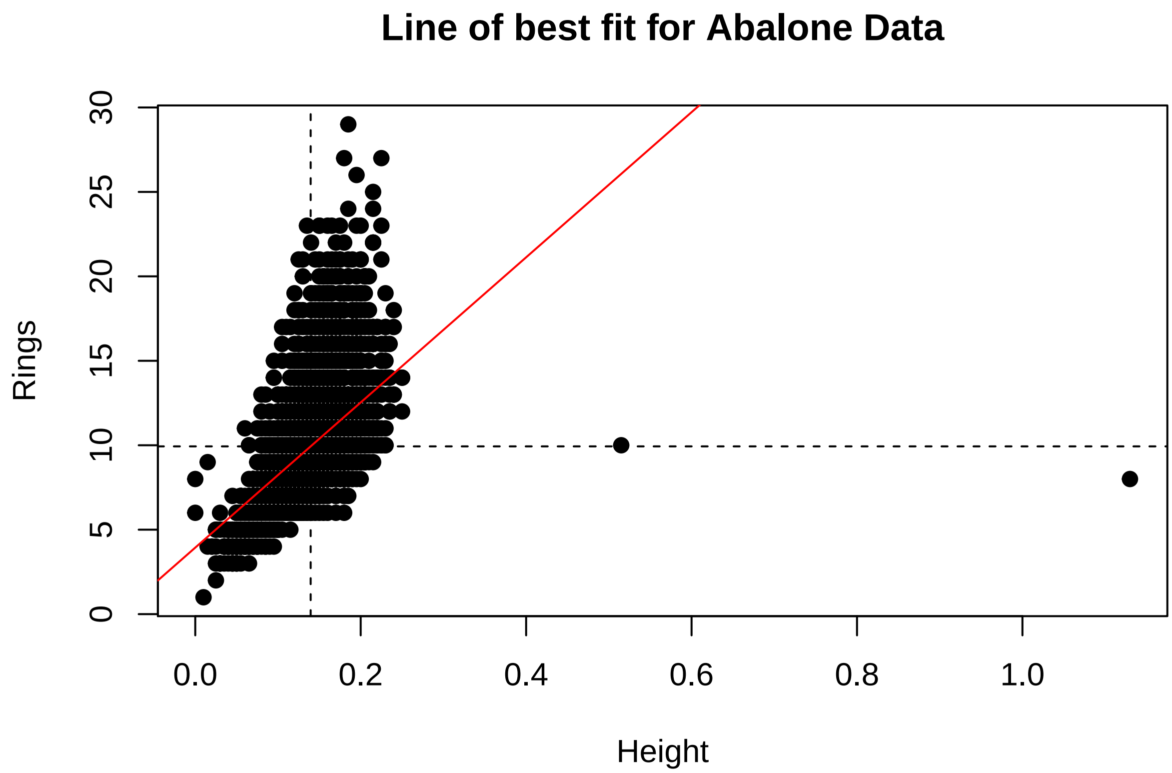



```
ggplot(abalone,aes(x=Height,y=Rings,fill = ..level..), geom = "polygon")+geom_density_2d()
```



Fitting Simple Linear Regression Line

```
#plotting with regression line

plot(abalone$Height,abalone$Rings,pch=19,xlab='Height',ylab='Rings')

abline(v=mean(abalone$Height),h=mean(abalone$Rings),lty=2)
fit.RP<-lm(abalone$Rings~abalone$Height)

title('Line of best fit for Abalone Data')

abline(coef(fit.RP),col='red')
```

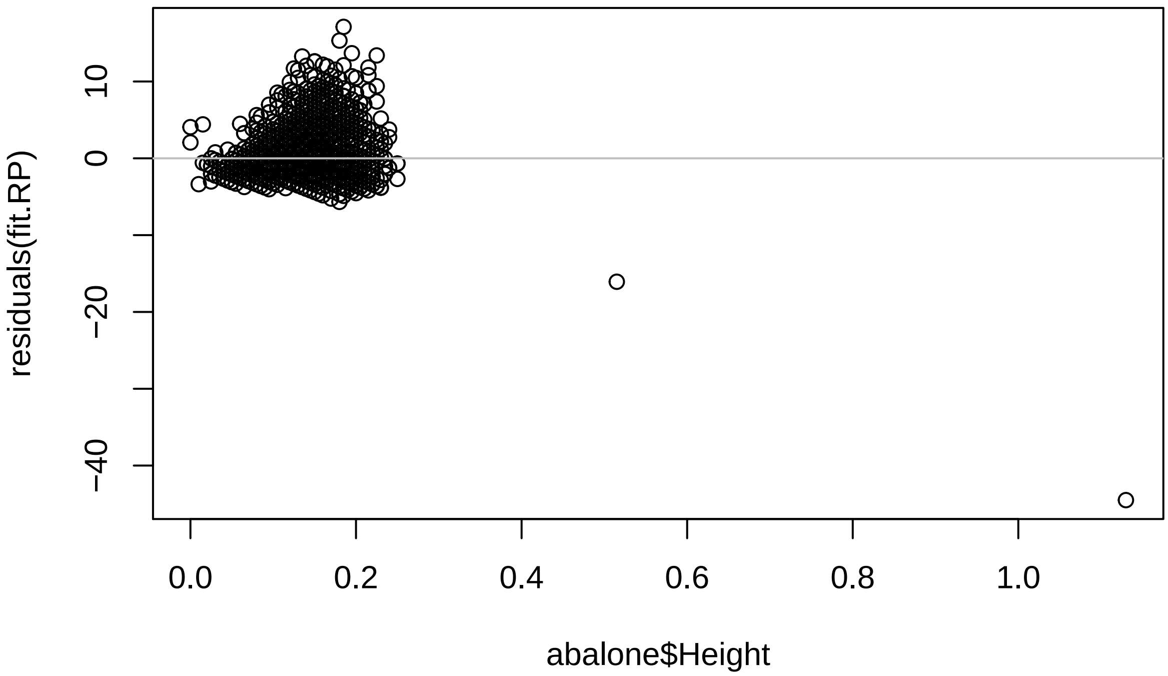



```r
summary(fit.RP)
```

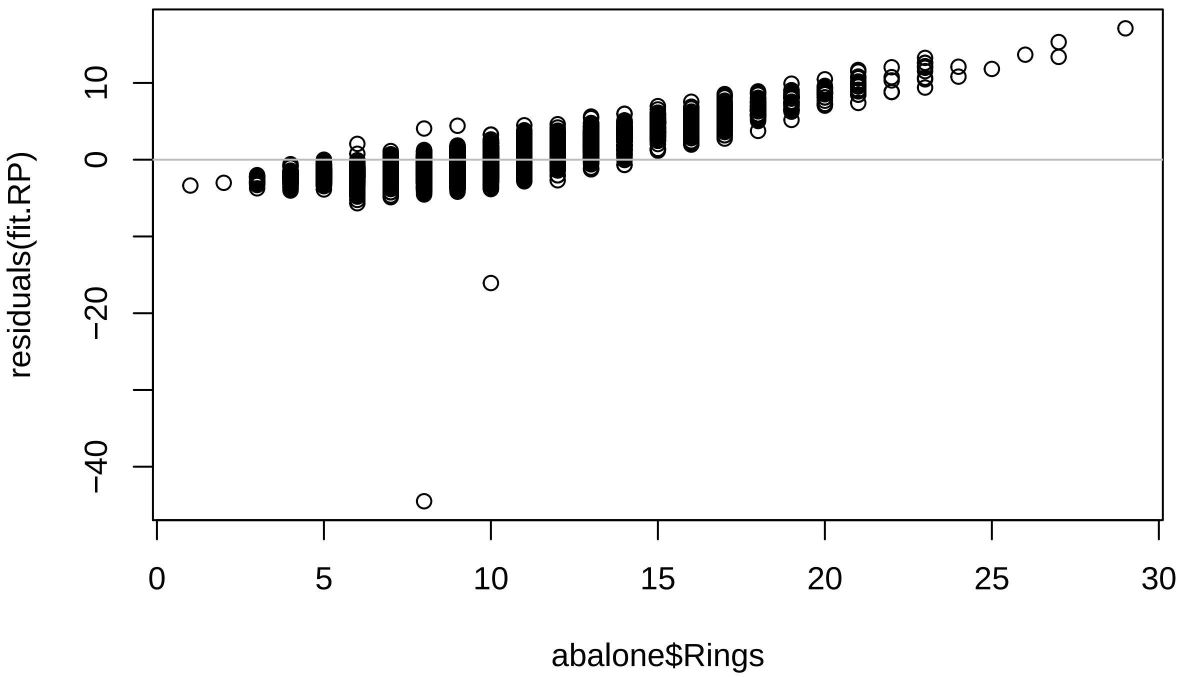
```
##
## Call:
## lm(formula = abalone$Rings ~ abalone$Height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.496  -1.657  -0.607   0.839  17.112
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.9385     0.1443   27.30   <2e-16 ***
## abalone$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
```

Model Aquecuacy Checking & Diagnostic

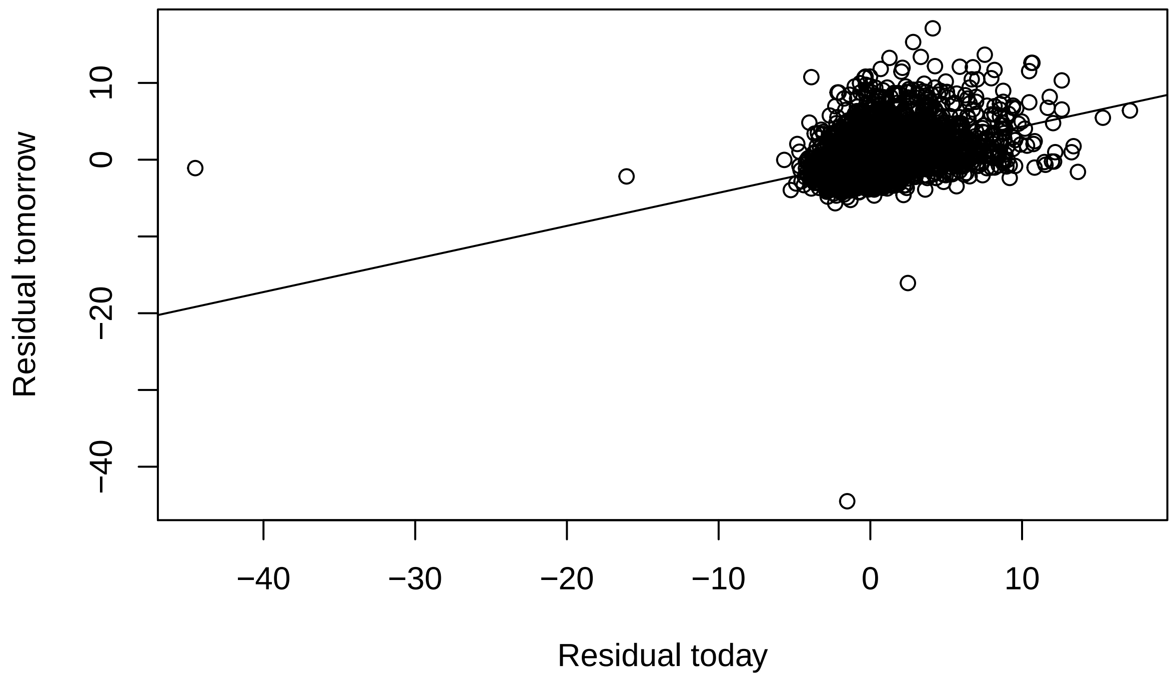
```r
#plotting residual vs Regressor
plot(abalone$Height, residuals(fit.RP))
abline(h = 0, col = "grey")
```



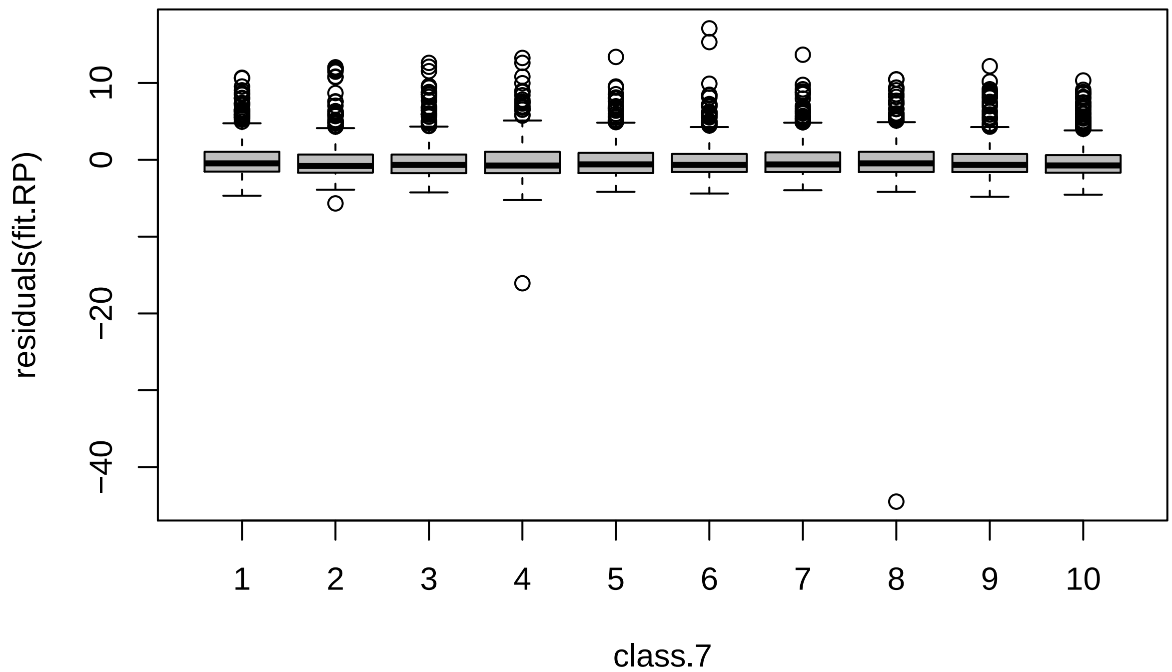

```r
#plotting residual vs Prediction Variable

plot(abalone$Rings, residuals(fit.RP))
abline(h = 0, col = "grey")
```



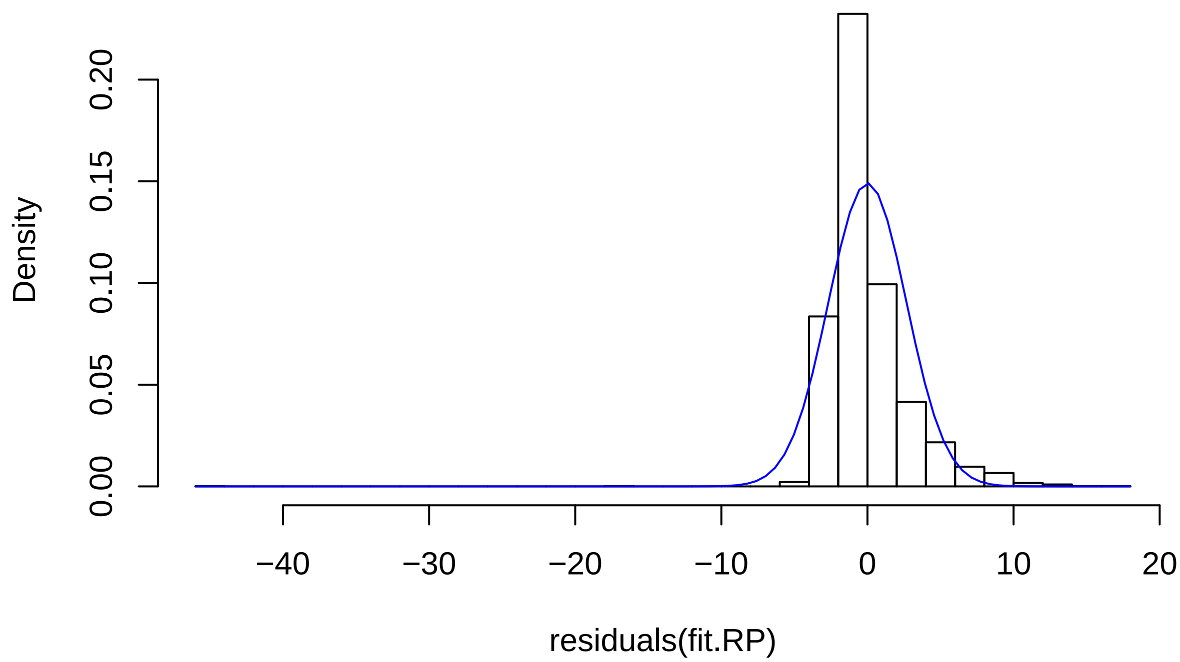

```r
#plotting Residual vs Residual

plot(head(residuals(fit.RP), -1),
tail(residuals(fit.RP), -1), xlab = "Residual today",
ylab = "Residual tomorrow")
abline(lm(tail(residuals(fit.RP),
-1) ~ head(residuals(fit.RP),
-1)))
```



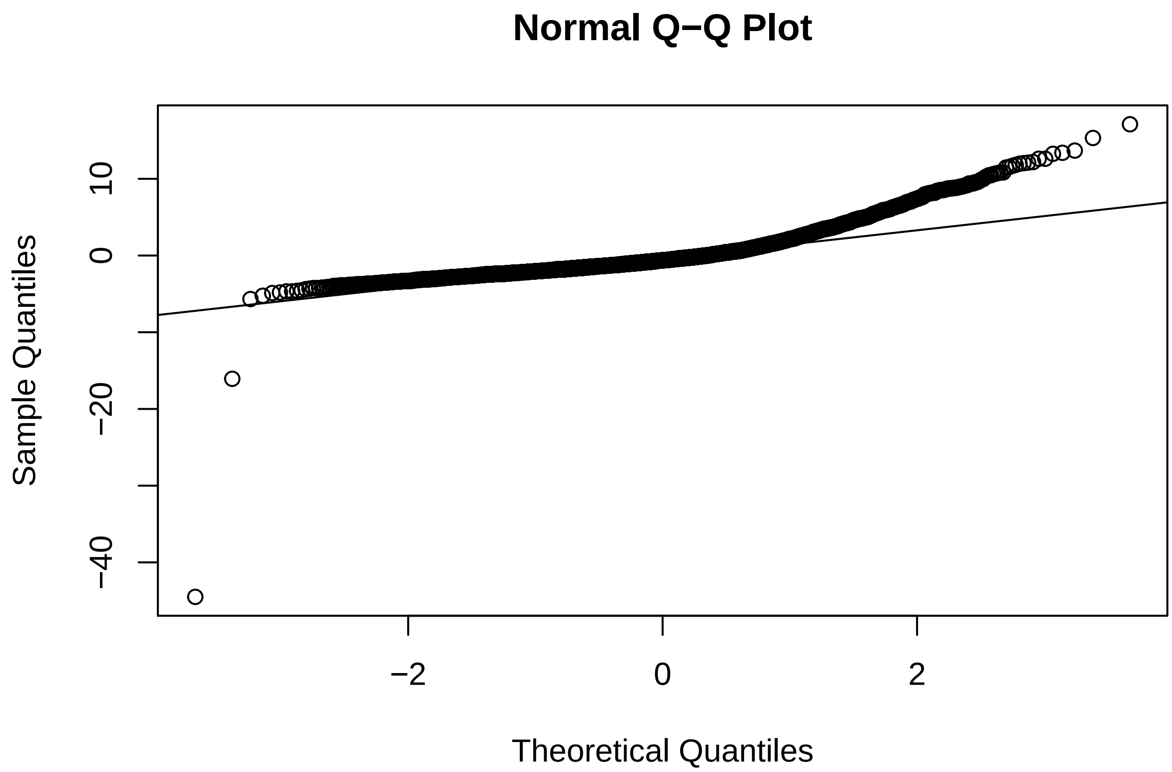

```
#plotting Residual Boxplots
n<-length(residuals(fit.RP))
x<-runif(n,0,100)
class.7<-cut(x,breaks=seq(0,100,by=10),labels=FALSE)
boxplot(residuals(fit.RP)~class.7,col='gray')
```




```r
#plotting Residual Distribution with Normal Distribution
hist(residuals(fit.RP), breaks = 40,
freq = FALSE,
main = "")
curve(dnorm(x, mean = 0, sd = sd(residuals(fit.RP))),
add = TRUE, col = "blue")
```



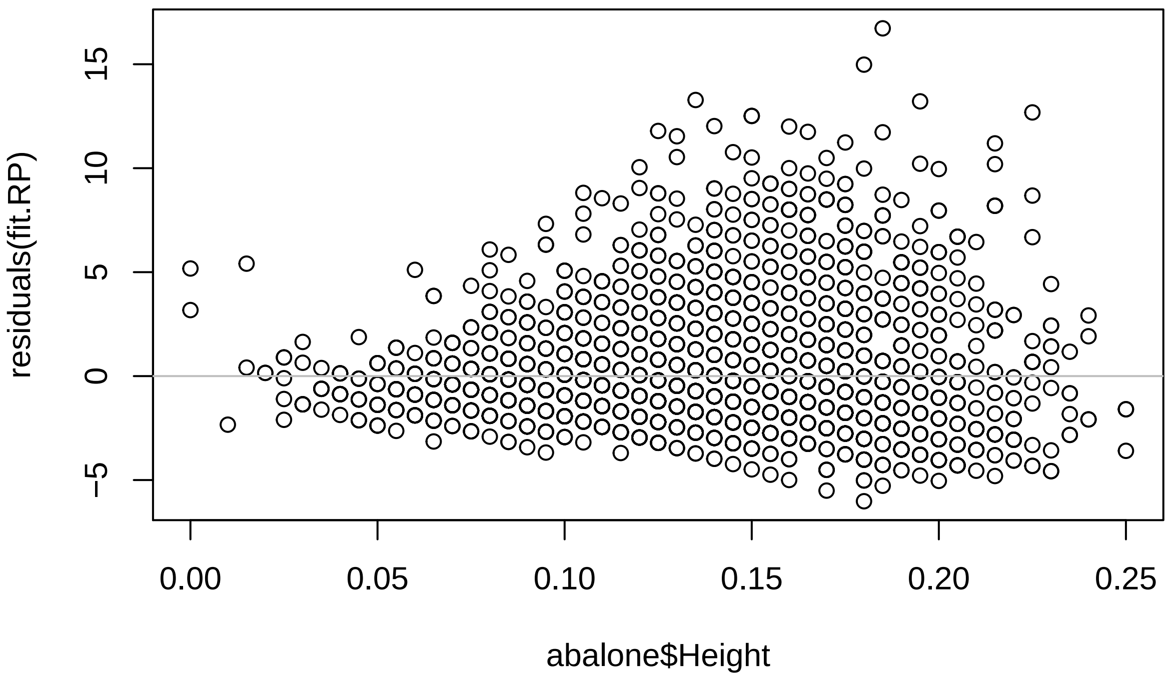

```
#plotting quartile
qqnorm(residuals(fit.RP))
qqline(residuals(fit.RP))
```



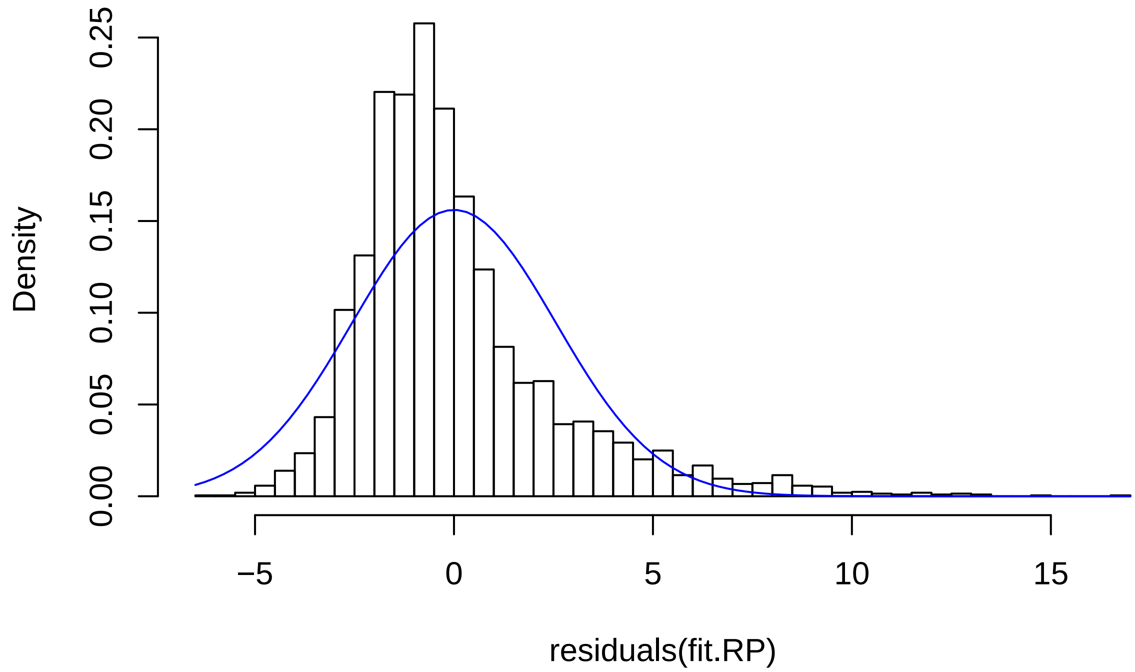
There seems to be outliers present and also a positive skew in the distribution of the residuals, we will fight this with a log transformation

Model Re-Fitting & Transformation

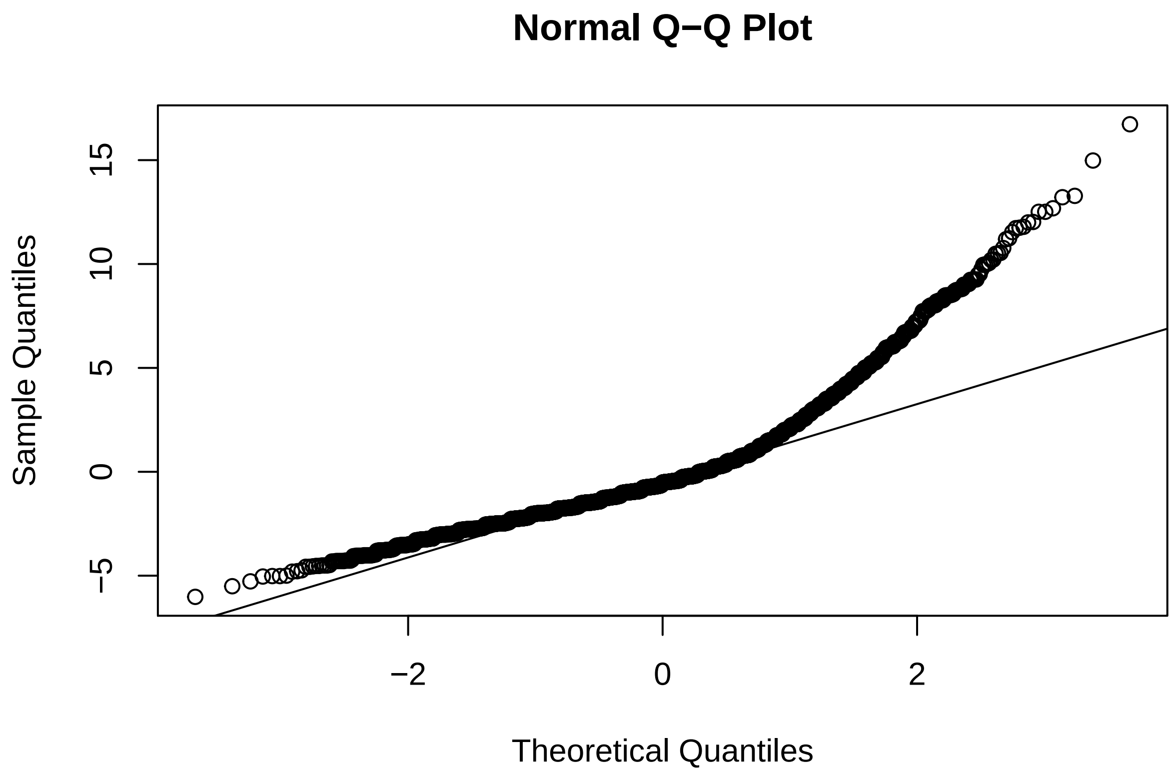
```
# removing outliers
abalone<-abalone%>%filter(Height<0.5)

fit.RP<-lm(abalone$Rings~abalone$Height)

#plotting data again with no outliers
plot(abalone$Height, residuals(fit.RP))
abline(h = 0, col = "grey")
```



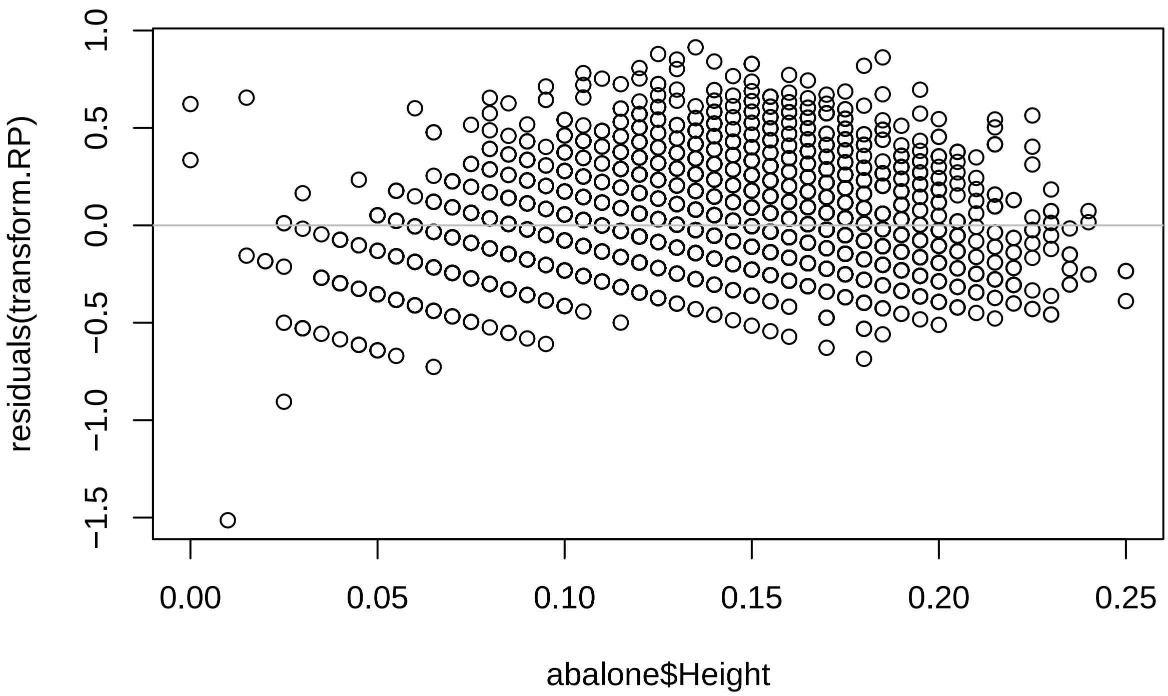

```r
#plotting residuals
hist(residuals(fit.RP), breaks = 40,
freq = FALSE,
main = "")
curve(dnorm(x, mean = 0, sd = sd(residuals(fit.RP))),
add = TRUE, col = "blue")
```



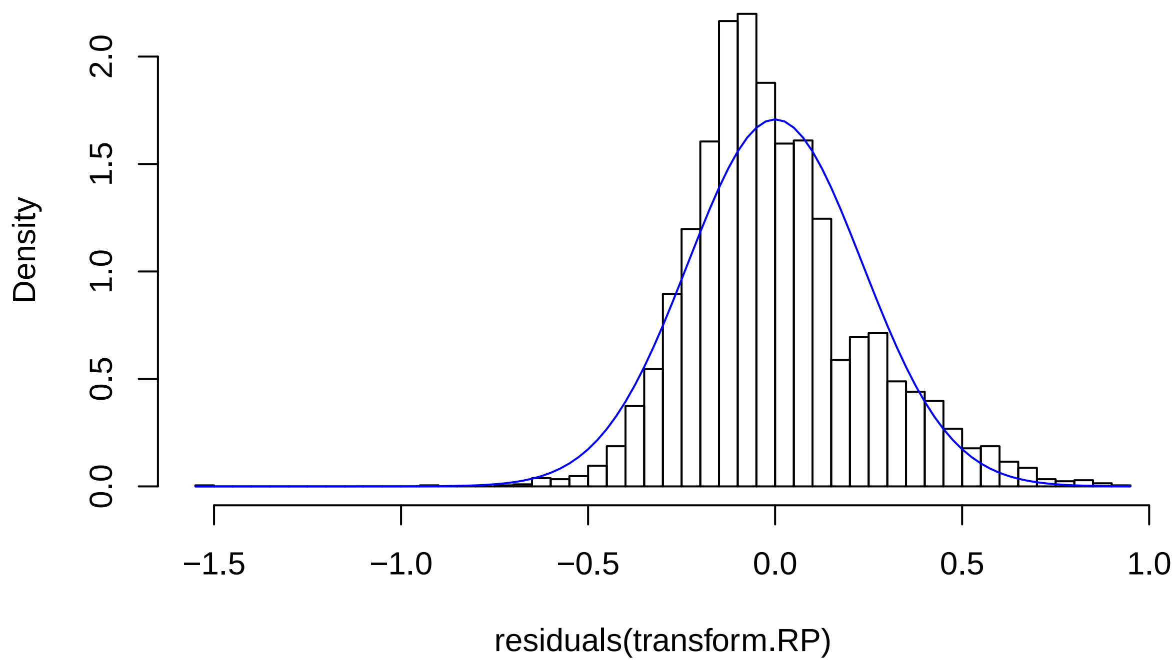

```
#plotting quantile
qqnorm(residuals(fit.RP))
qqline(residuals(fit.RP))
```



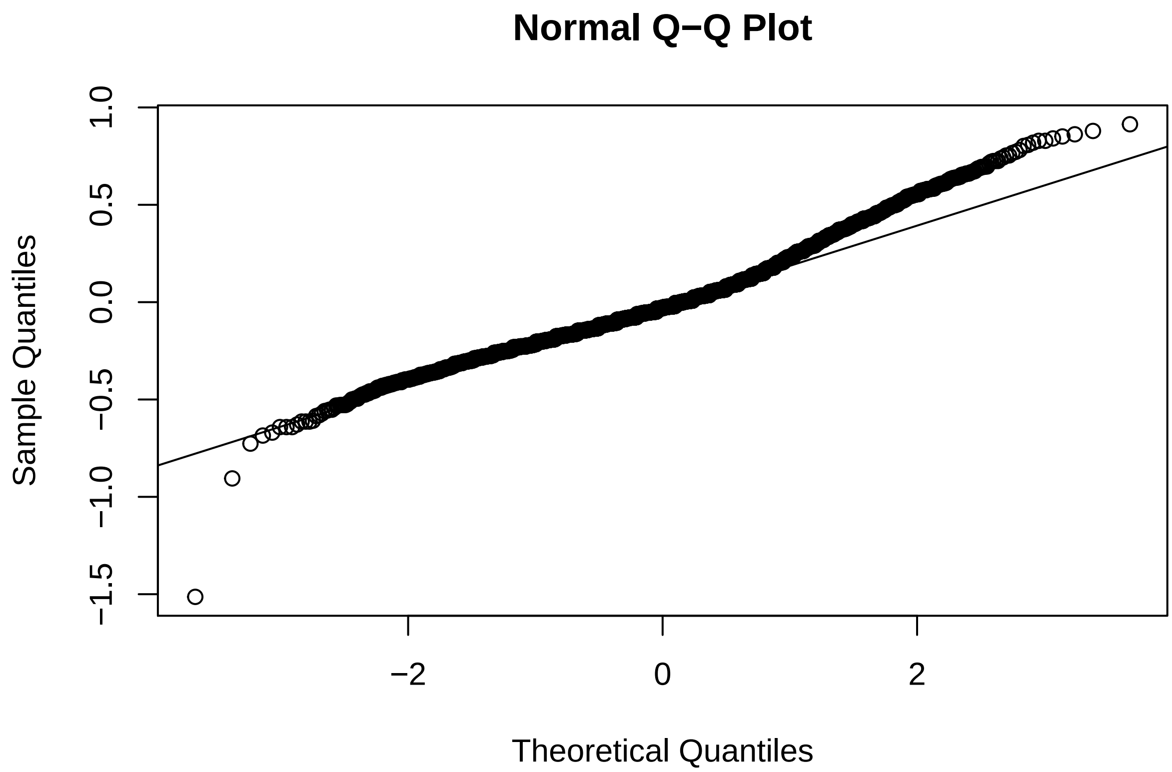

```
# log transformation to remove skewness of data
transform.RP<-lm(log(abalone$Rings)~abalone$Height)

#plotting residuals
plot(abalone$Height, residuals(transform.RP))
abline(h = 0, col = "grey")
```




```r
#plotting density vs gaussian
hist(residuals(transform.RP), breaks = 40,
freq = FALSE,
main = "")
curve(dnorm(x, mean = 0, sd = sd(residuals(transform.RP))),
add = TRUE, col = "blue")
```




```r
#plotting quantile
qqnorm(residuals(transform.RP))
qqline(residuals(transform.RP))
```



Functions for Coming Questions

```
#functions needed

Confidence_Interval<-function(estimate,standard_error,alpha,length)
{
  interval=c(estimate+(qt(p=(alpha/2),df=length-2,lower.tail = T))*(standard_error),
             estimate+(qt(p=(alpha/2),df=length-2,lower.tail = F))*(standard_error))

  return (interval)
}

SXX<-function(x) {
  return (sum((x - mean(x))^2))
}


SST<-function(x,y) {
return (sum(y^2)-((sum(y)^2)/length(y))) }

SXY<-function(x,y) {
return (sum((x - mean(x))*(y-mean(y)))) }

B1_hat<-function(x,y) {
return (SXY(x,y)/SXX(x)) }
```

```
SSres<-function(x,y) {
  return (SST(x,y)-(B1_hat(x,y)*SXY(x,y)))
}


MSres<-function(x,y) {
  return (SSres(x,y)/(length(x)-2))
}




Point_Estimate<-function(B1,B0,x)
{
  return (B1*x+B0)
}

MeanResponse_CI<-function(B1,B0,x0,x,y,alpha)
{
  estimate<-Point_Estimate(B1,B0,x0)


  interval<-c(estimate+(qt(p=(alpha/2),df=length(x)-2,lower.tail = T))*(sqrt(MSres(x,y)*((1/length(x))+

  return (interval)
}

Prediction_CI<-function(B1,B0,x0,x,y,alpha)
{
  {
  estimate<-Point_Estimate(B1,B0,x0)


  interval<-c(estimate+(qt(p=(alpha/2),df=length(x)-2,lower.tail = T))*(sqrt(MSres(x,y)*(1+(1/length(x)

  return (interval)
}
}
```

Confidence Interval

```
#Confidence Interval for B1
Confidence_Interval(summary(transform.RP)[["coefficients"]][, "Estimate"][2],
                    summary(transform.RP)[["coefficients"]][, "Std. Error"][2],0.05,length(abalone$Heig
```

```
## abalone$Height abalone$Height
##       5.482899       5.851252
```

```
#Confidence Interval for B0
Confidence_Interval(summary(transform.RP)[["coefficients"]][, "Estimate"][1],
                    summary(transform.RP)[["coefficients"]][, "Std. Error"][1], 0.05,length(abalone$Hei
```

```
## (Intercept) (Intercept)
##    1.430312    1.483506
```

Statistical Significance

```
(qt(p=(0.05/2),df=length(abalone$Height)-2,lower.tail = T))
```

```
## [1] -1.960533
```

```
(qt(p=(0.05/2),df=length(abalone$Height)-2,lower.tail = F))
```

```
## [1] 1.960533
```

```
summary(transform.RP)[["coefficients"]][, "t value"][2]
```

```
## abalone$Height
##       60.32516
```

Since our t value is larger than our t quartile limits, we can conclude that B1 is statistically significant

Mean Response Confidence Interval

```
#we will construct a  95% confidence interval for the average number of rings for abalones with height 

Point_Estimate(summary(transform.RP)[["coefficients"]][, "Estimate"][2],summary(transform.RP)[["coeffic
```

```
## abalone$Height
##       2.182295
```

```
MeanResponse_CI(summary(fit.RP)[["coefficients"]][, "Estimate"][2],summary(fit.RP)[["coefficients"]][, 
```

```
## abalone$Height abalone$Height
##       9.281826       9.443392
```

Prediction Confidence Interval

```
#we will find the predicted value and a 99% prediction interval for height=0.138

Prediction_CI(summary(fit.RP)[["coefficients"]][, "Estimate"][2],summary(fit.RP)[["coefficients"]][, "E
```

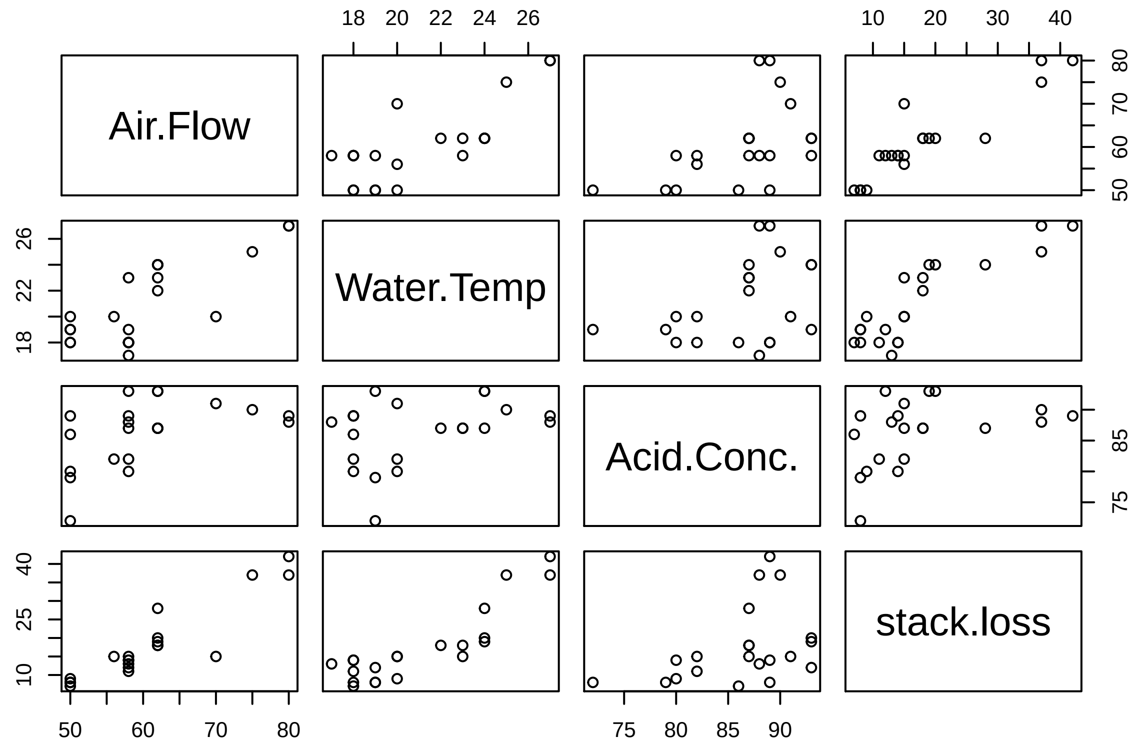
```
## abalone$Height abalone$Height
##       4.349753      14.375466
```

Conclusions: The dataset contains outliers and the distribution of the residuals contains a positive skew, thus we should look more into the model assumptions we have made of constant variance and mean error of 0. We can try to sample more spread of data for Height as it was very clusters from what weve seen in our univariate distributions

Question 3:

1.) Plotting the Data

```r
#importing and plotting data
data(stackloss)
plot(stackloss)
```



2.) Fitting Multiple Linear Regression

```r
fit.MR <- lm ( stack.loss ~
+     Air.Flow + Water.Temp + Acid.Conc.,data=stackloss)


# summary of multiple regression fit
summary(fit.MR)
```

```
##
## Call:
## lm(formula = stack.loss ~ +Air.Flow + Water.Temp + Acid.Conc.,
##     data = stackloss)
##
## 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 **
```

```
## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
## Water.Temp    1.2953     0.3680   3.520  0.00263 **
## Acid.Conc.   -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
```

Functions for Coming Questions

```r
C<-function(j,hat_matrix)
{
  diag(hat_matrix)[j]
}

Coefficient_ConfidenceInterval<-function(hat_matrix,B,length,alpha,dof,sigma,coefficient)
{

  interval=c(B[coefficient]+(qt(p=(alpha/2),df=dof,lower.tail = T))*(sqrt(sigma*C(coefficient,hat_matri
             B[coefficient]+(qt(p=(alpha/2),df=dof,lower.tail = F))*(sqrt(sigma*C(coefficient,hat_matri

  return (interval)

}
```

3.) Constructing 90% CI for Coefficients

```r
X <- cbind(constant = 1, as.matrix(stackloss$Air.Flow),as.matrix(stackloss$Water.Temp),
           as.matrix(stackloss$Acid.Conc.))

Y <- as.matrix(stackloss$stack.loss)

B<-solve(t(X)%*%X)%*%t(X)%*%Y

sigma_hat<-(t(Y)%*%Y-t(B)%*%t(X)%*%Y)/(length(Y)-4)

hat_matrix<-solve((t(X)%*%X))

#CI for all 4 prediction coefficients in order of B0 to B3
Coefficient_ConfidenceInterval(hat_matrix,B,length(B),0.1,length(Y)-3,sigma_hat,1)
```

```
## [1] -60.54809 -19.29126
```

```r
Coefficient_ConfidenceInterval(hat_matrix,B,length(B),0.1,length(Y)-3,sigma_hat,2)
```

```
## [1] 0.4817875 0.9494929
```

```r
Coefficient_ConfidenceInterval(hat_matrix,B,length(B),0.1,length(Y)-3,sigma_hat,3)
```

```
## [1] 0.6571086 1.9334636
```

```r
Coefficient_ConfidenceInterval(hat_matrix,B,length(B),0.1,length(Y)-3,sigma_hat,4)
```

```
## [1] -0.4231463  0.1189013
```

4.) 99% prediction interval for a new observation when Airflow = 58, Water temperature = 20 and Acid = 86

```r
newdata = data.frame(Air.Flow=58,
     Water.Temp=20,
     Acid.Conc.=86)

predict(fit.MR, newdata, interval="prediction", level=0.99)
```

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

Test the null hypothesis H0 : B3 = 0

```r
(qt(p=(0.1/2),df=length(Y)-3,lower.tail = T))
```

```
## [1] -1.734064
```

```r
(qt(p=(0.1/2),df=length(Y)-3,lower.tail = F))
```

```
## [1] 1.734064
```

```r
summary(fit.MR)[["coefficients"]][, "t value"][4]
```

```
## Acid.Conc.
## -0.9733098
```

```r
summary(fit.MR)[["coefficients"]][,"Pr(>|t|)"][4]
```

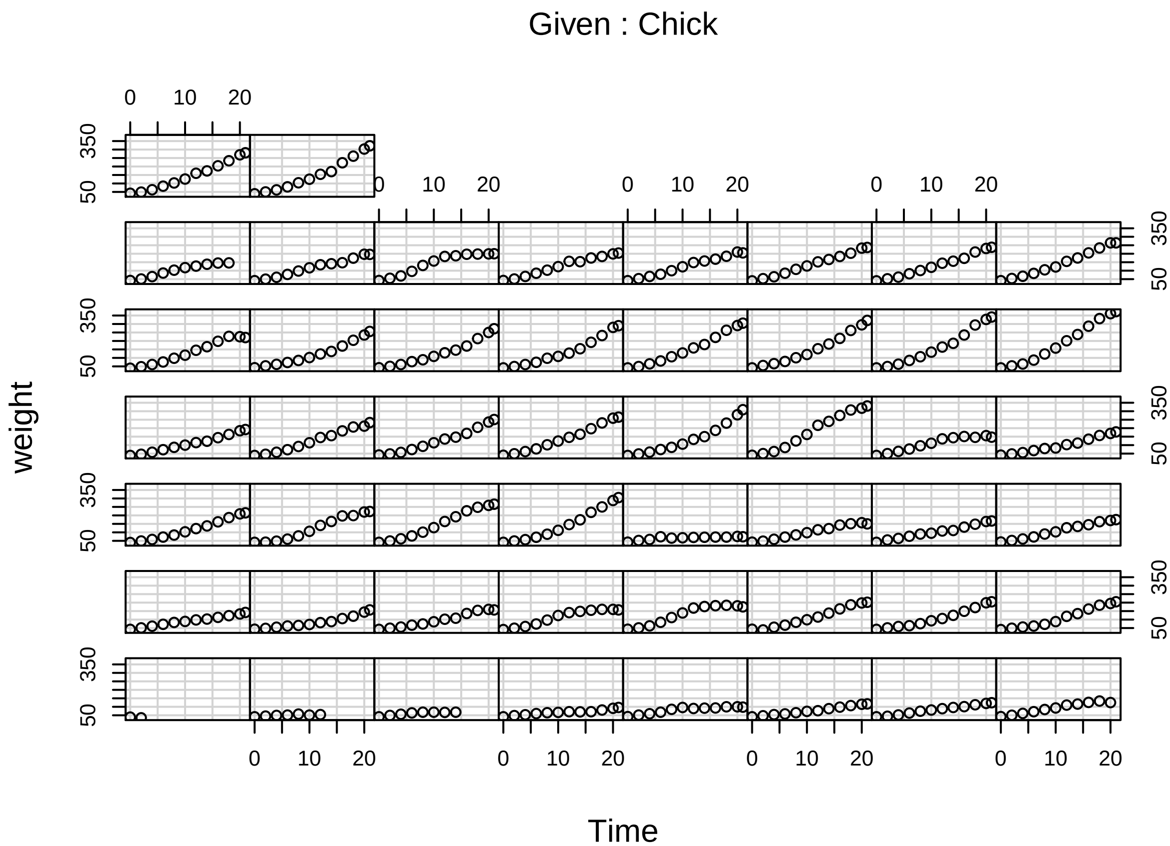
```
## Acid.Conc.
##  0.3440461
```

since out t value is smaller than out t range values we can accept our null and conclude that B3 is not statistically significant

Question 4:

1.) Plotting Data

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



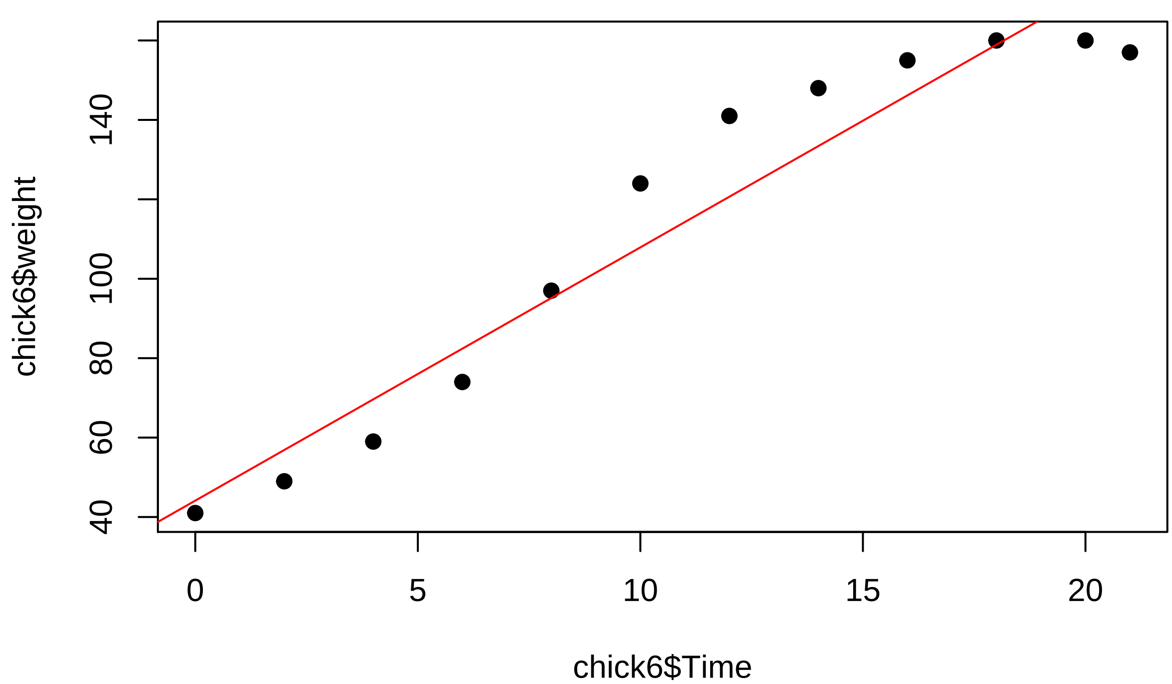
Functions we will use for upcoming questions

```
Polynomial_Regression<-function(exponent,data)
{
  for (i in 2:exponent)
  {
    poly.line<-lm(data$weight~poly(data$Time,i))

    plot(x=data$Time, y=data$weight,pch=19)
    points(x=data$Time, fitted(poly.line), col = "blue")
    plot(x=data$Time, resid(poly.line),pch=19)
    abline(h = 0)

  }
}

Polynomial_Regression_MLR<-function(exponent,data)
{
  for (i in 2:exponent)
  {
    fit.mult<-lm(weight~poly(Time,i)+Diet,data=chick_diet)
    plot(chick_diet$Time, residuals(fit.mult),pch=19)

    abline(h = 0, col = "grey")

    plot(chick_diet$Diet, residuals(fit.mult),pch=19)
```

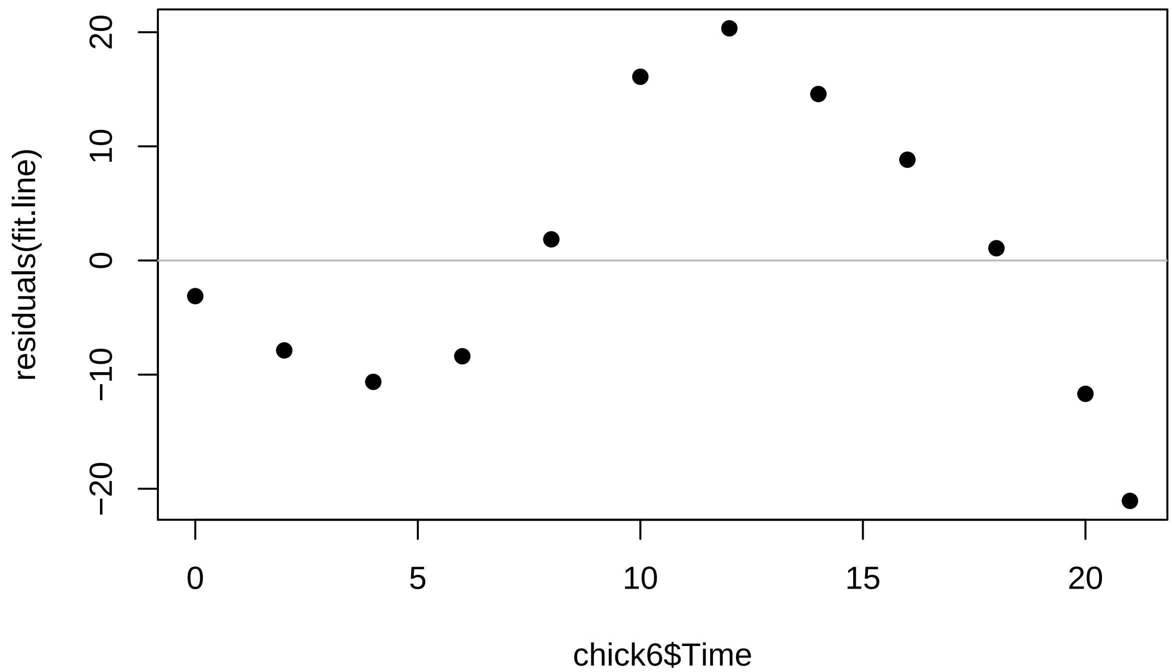
```
    abline(h = 0, col = "grey")

  }
}
```

2.) Fit Linear Regression on chick#6, Show Linear and Polynomial Regression & Show Residuals

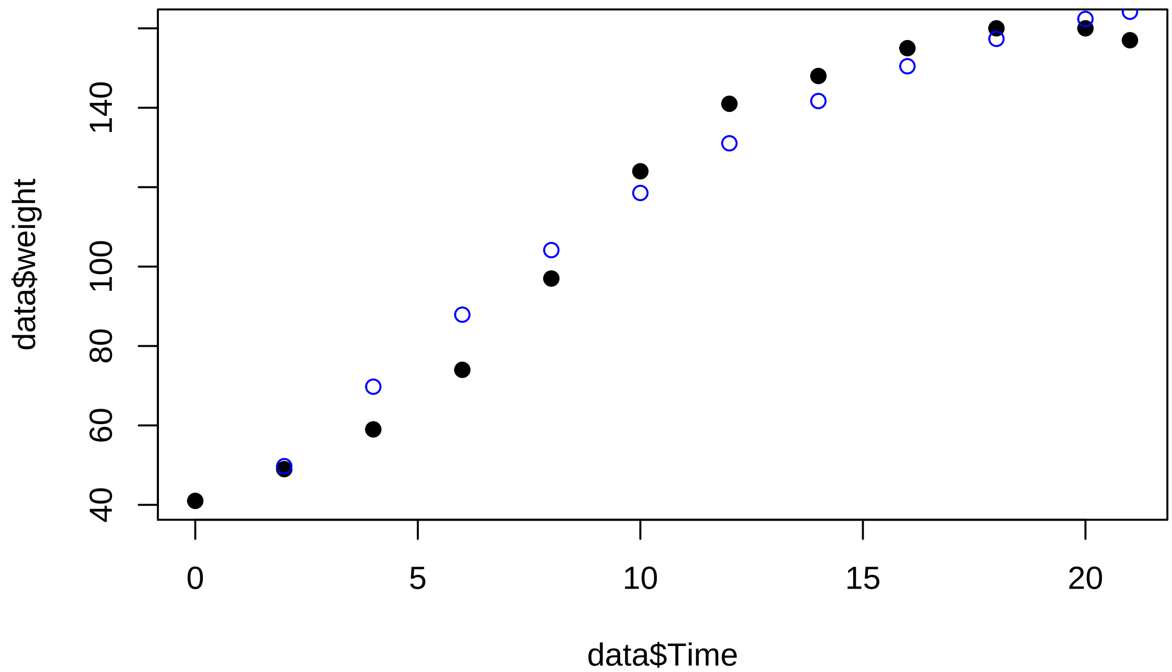
```
chick6<-ChickWeight%>%filter(Chick==6)

plot(x=chick6$Time,y=chick6$weight,pch=19)

fit.line<-lm(chick6$weight~chick6$Time)

abline(coef(fit.line),col='red')
```

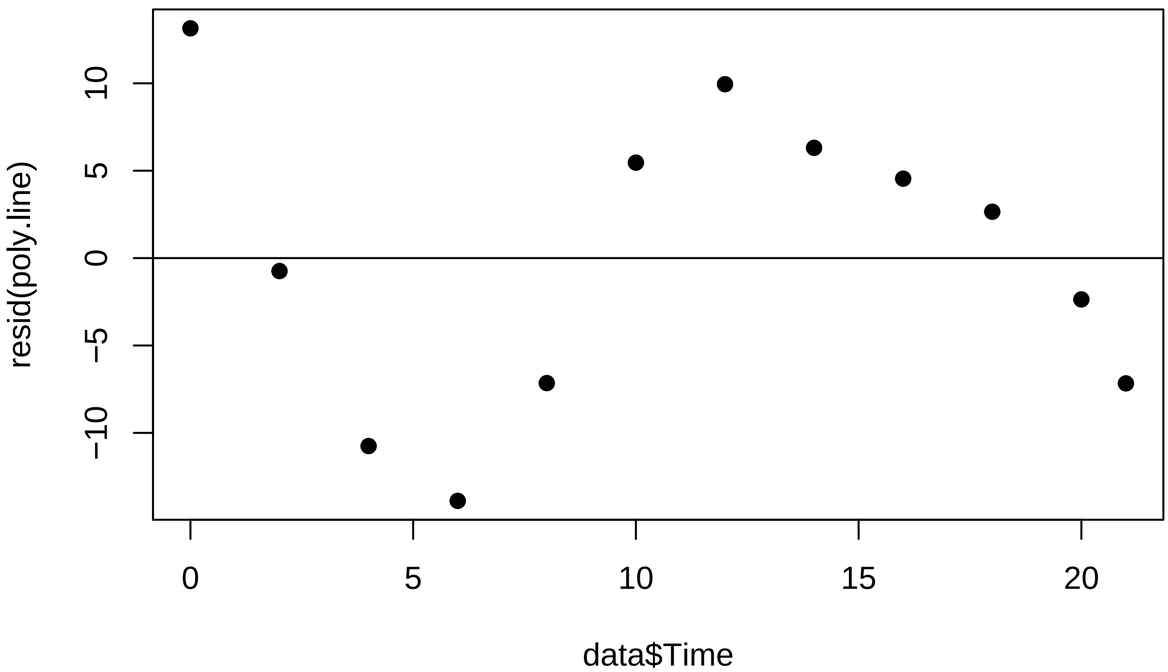



```
plot(chick6$Time, residuals(fit.line),pch=19)

abline(h = 0, col = "grey")
```

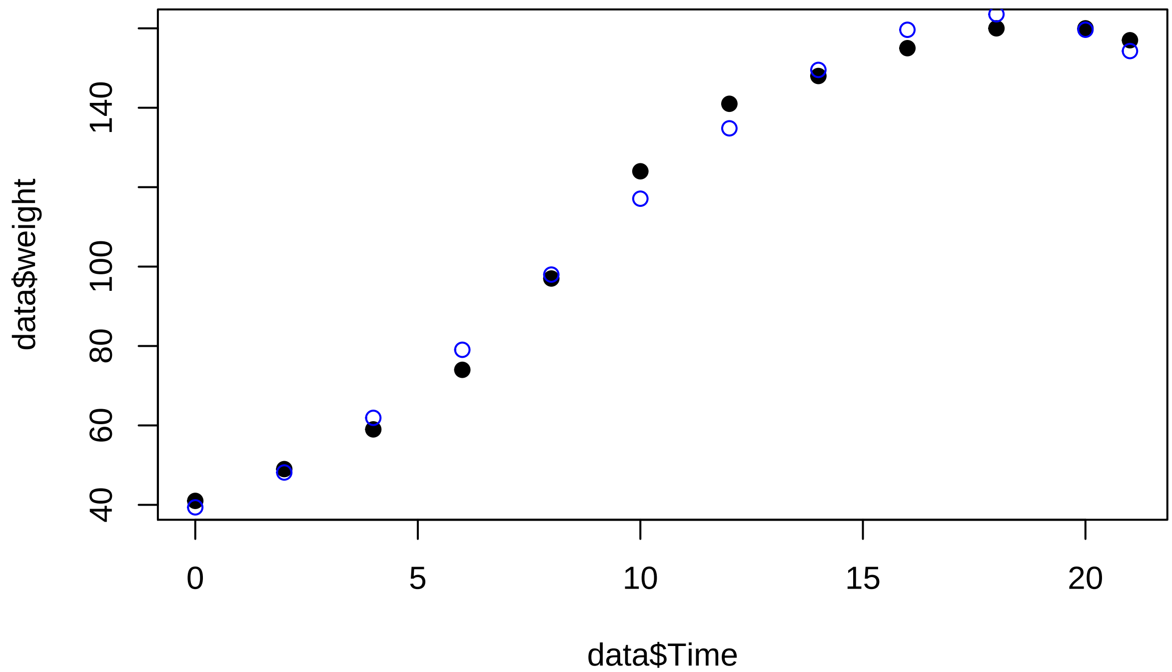


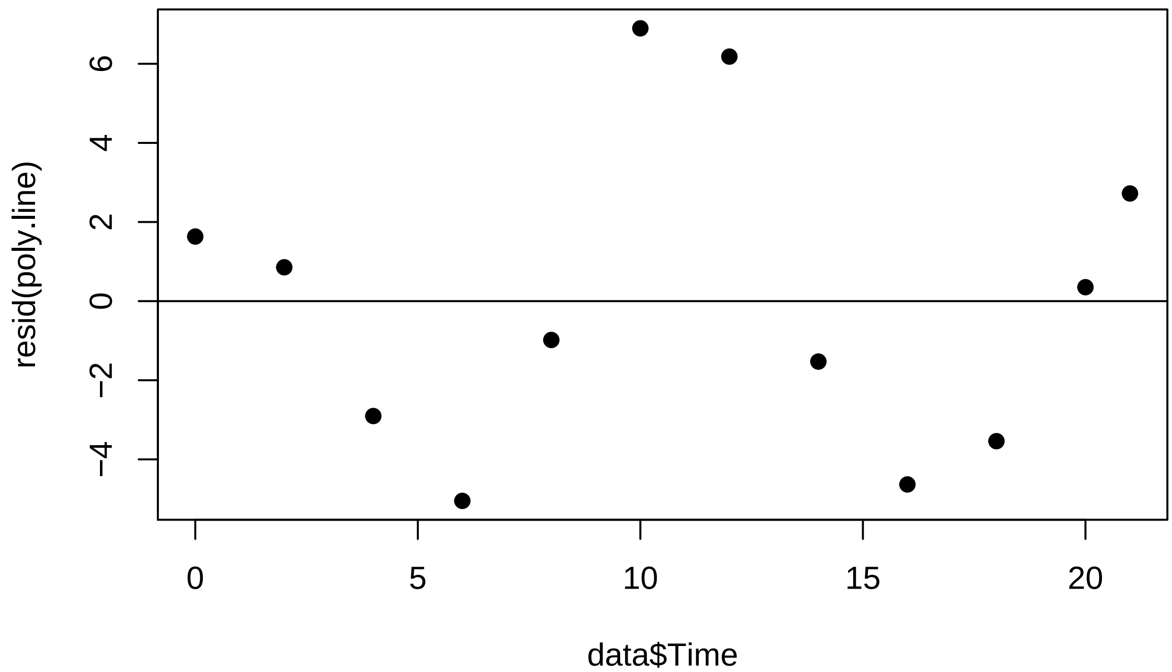

```r
#I try with all polynomials up to power 5
Polynomial_Regression(5,chick6)
```

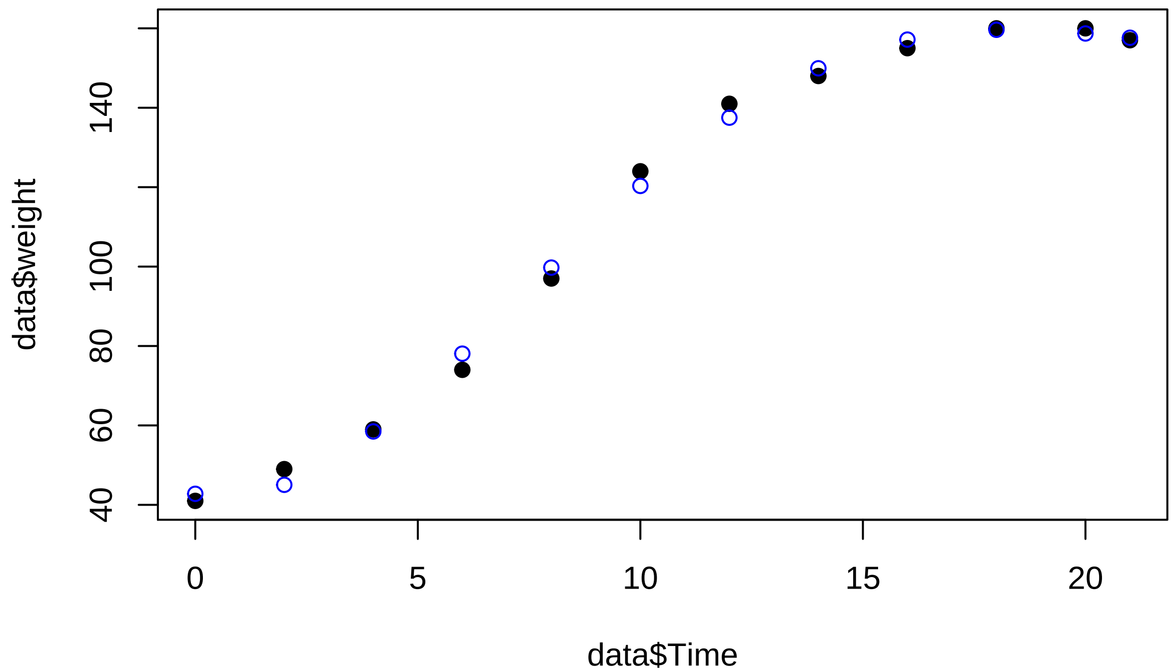

data$weight
40
60
80
100
140
0
5
10
15
20
data$Time



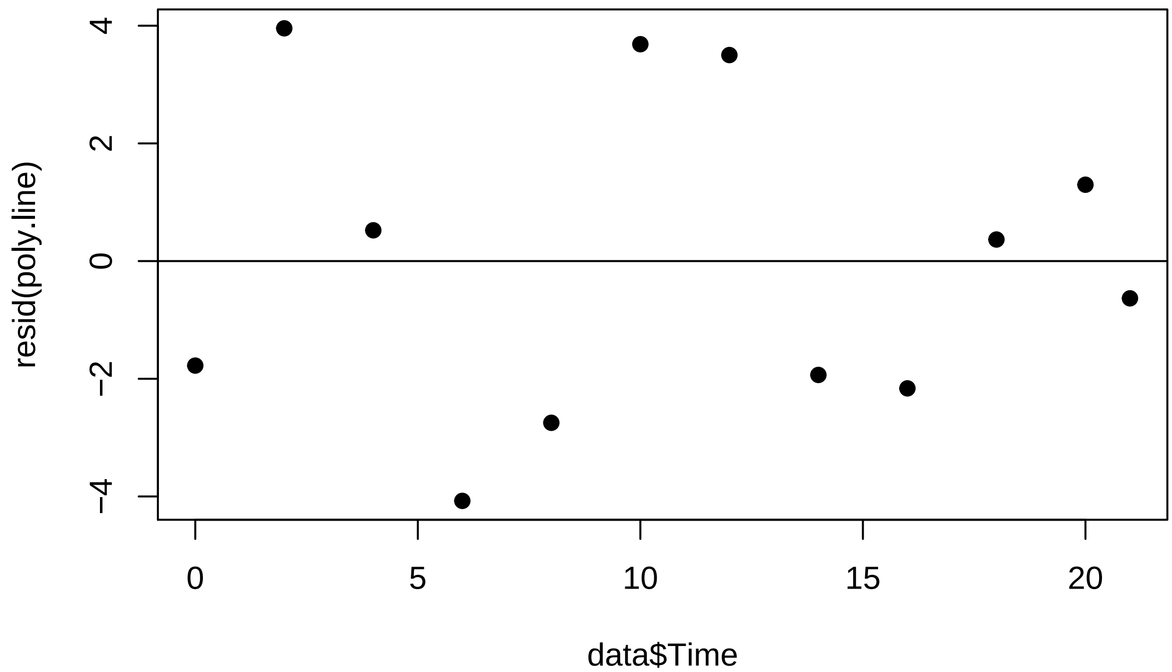
resid(poly.line)
−10
−5
0
5
10
0
5
10
15
20
data$Time

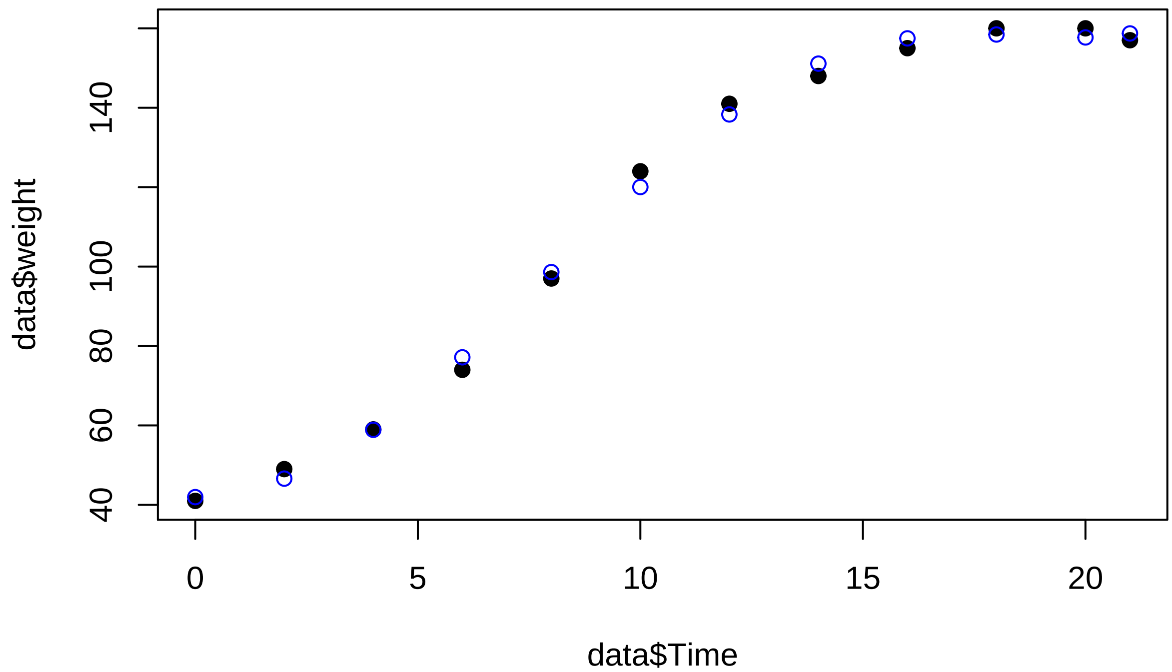

data$weight
40
60
80
100
140
0
5
10
15
20
data$Time

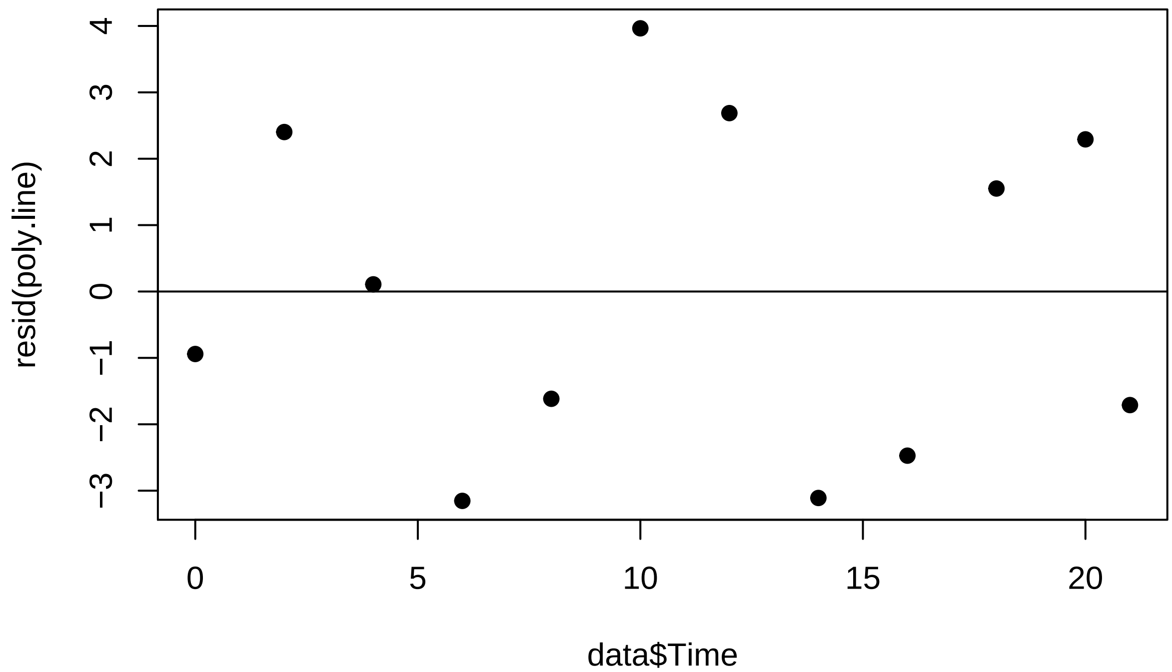

resid(poly.line)
−4
−2
0
2
4
6
0
5
10
15
20
data$Time



40
60
80
100
140
data$weight
0
5
10
15
20
data$Time

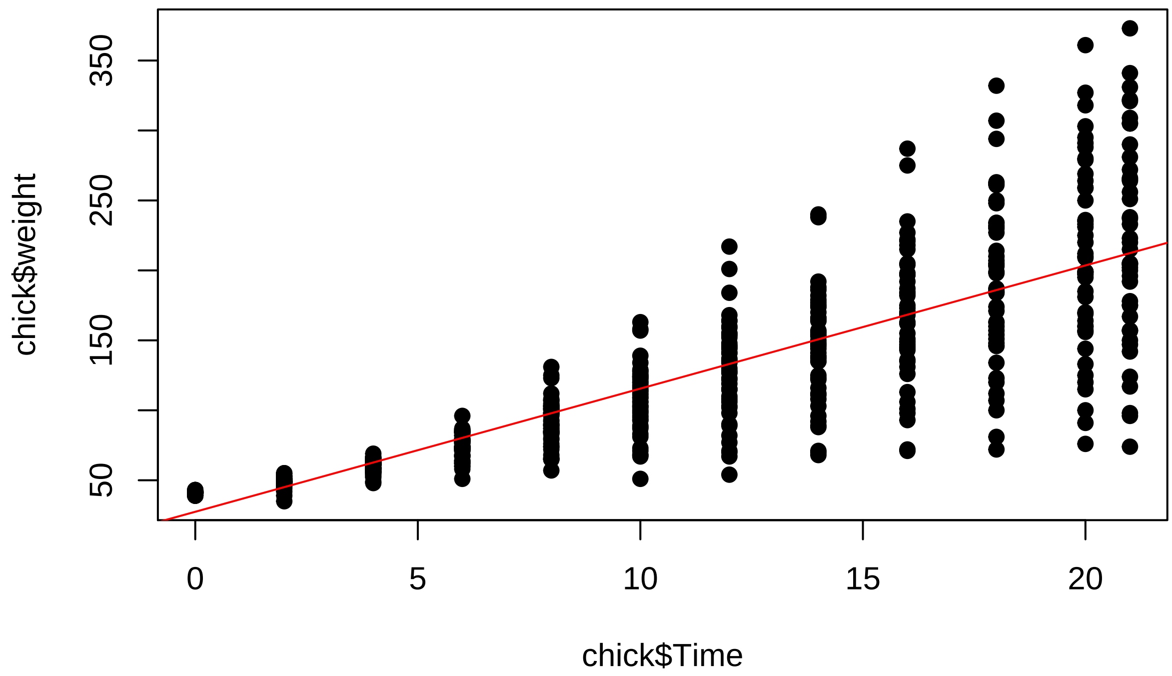

resid(poly.line)
4
2
0
−2
−4
0
5
10
15
20
data$Time


140
100
80
60
40
data$weight
0
5
10
15
20
data$Time

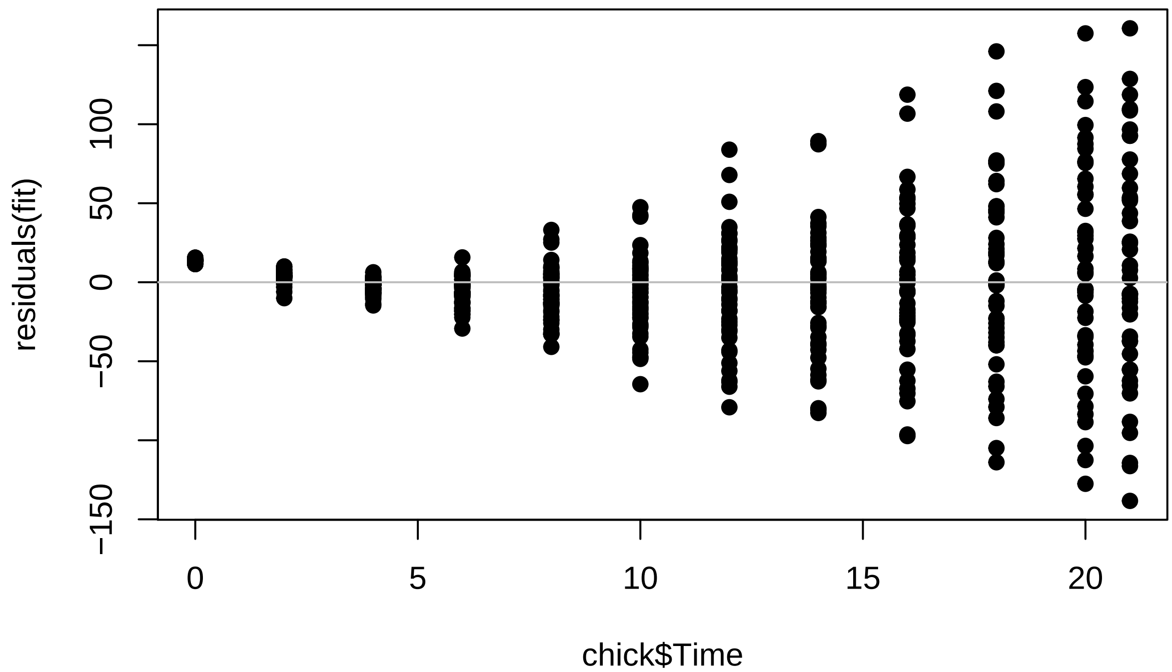


3.) Fit Linear Regression all data, Show Linear and Polynomial Regression & Show Residuals

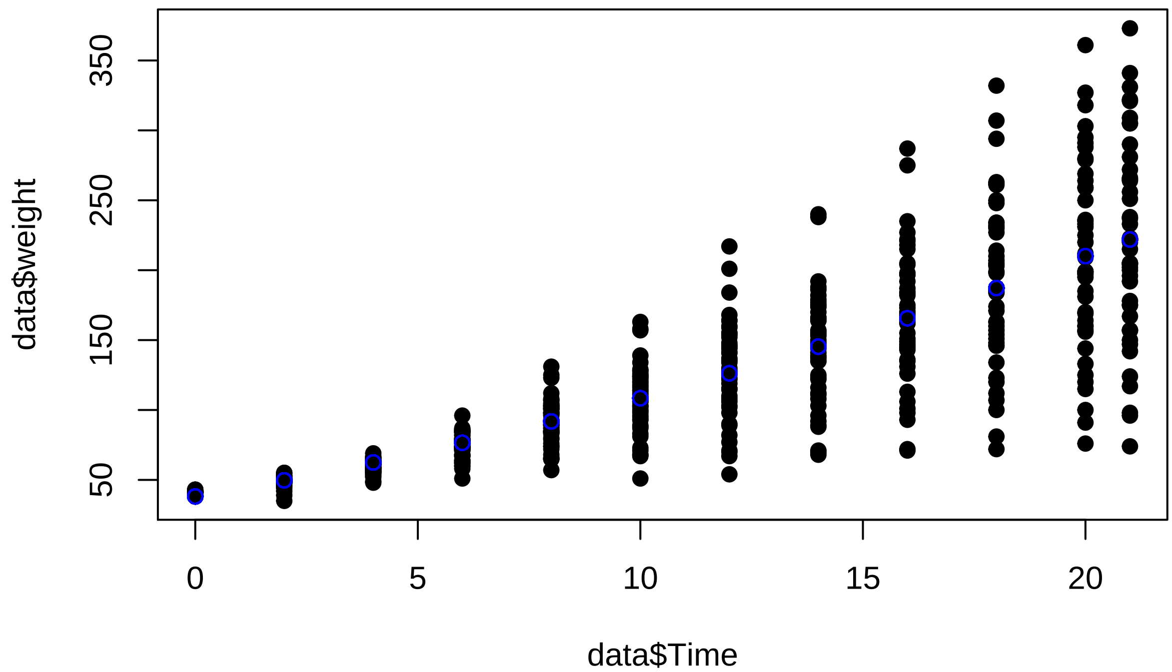
```
chick<-ChickWeight%>%select(weight,Time)

plot(x=chick$Time,y=chick$weight,pch=19)

fit<-lm(chick$weight~chick$Time)

abline(coef(fit),col='red')
```



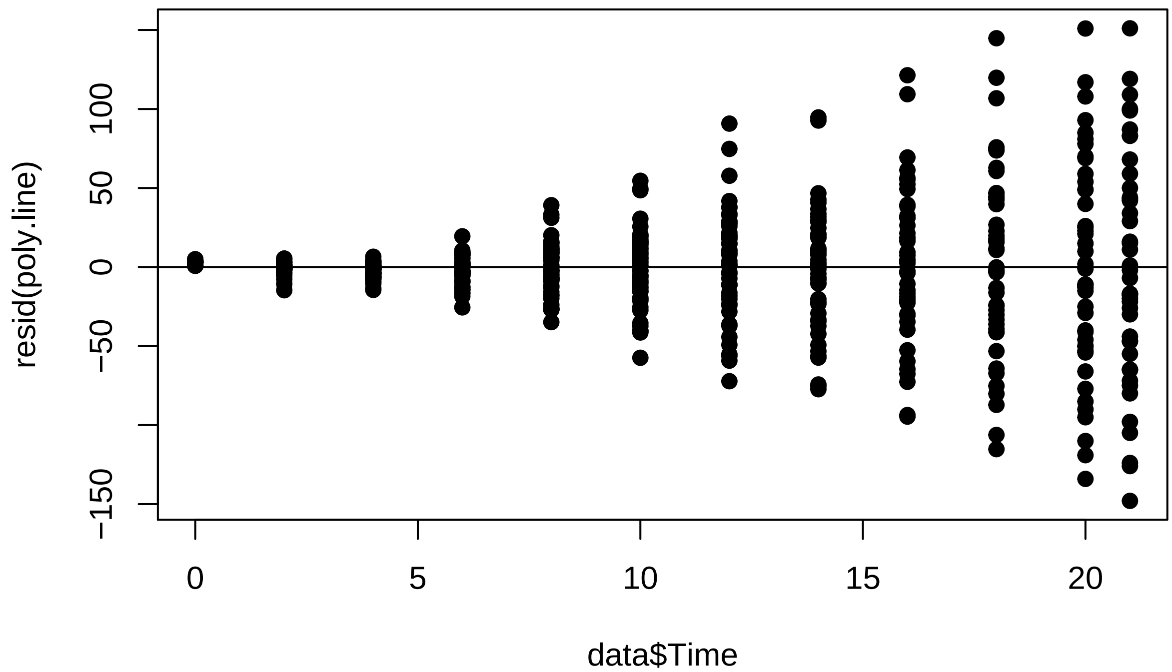

```r
plot(chick$Time, residuals(fit),pch=19)

abline(h = 0, col = "grey")
```



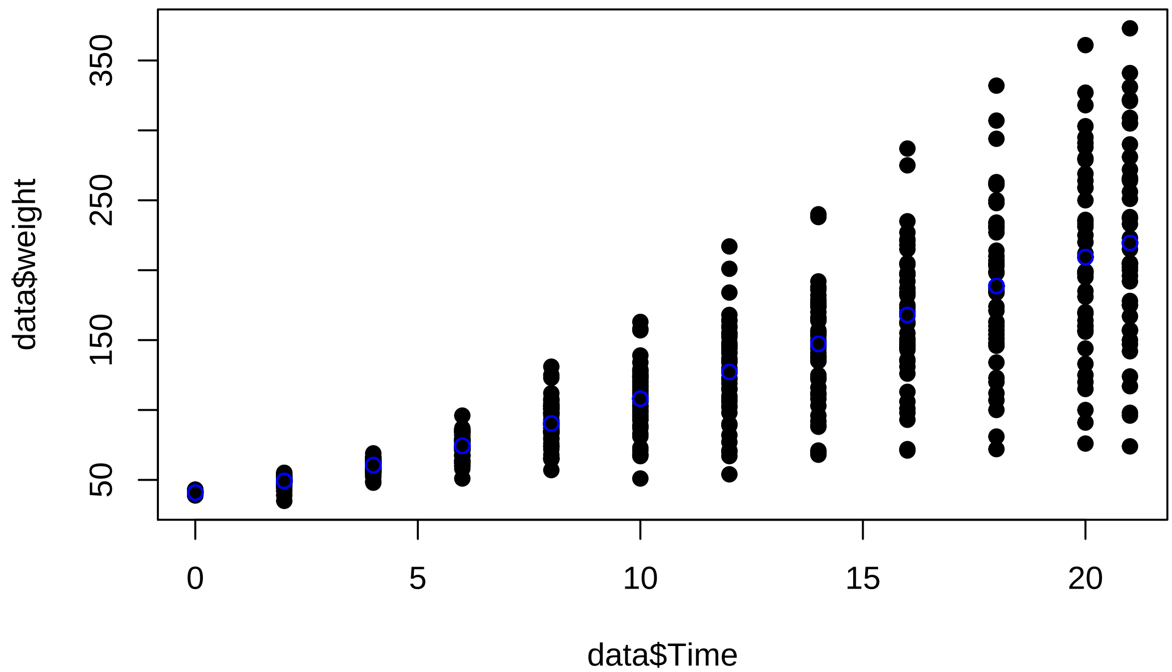

```r
#I try with all polynomials up to power 5
Polynomial_Regression(5,chick)
```

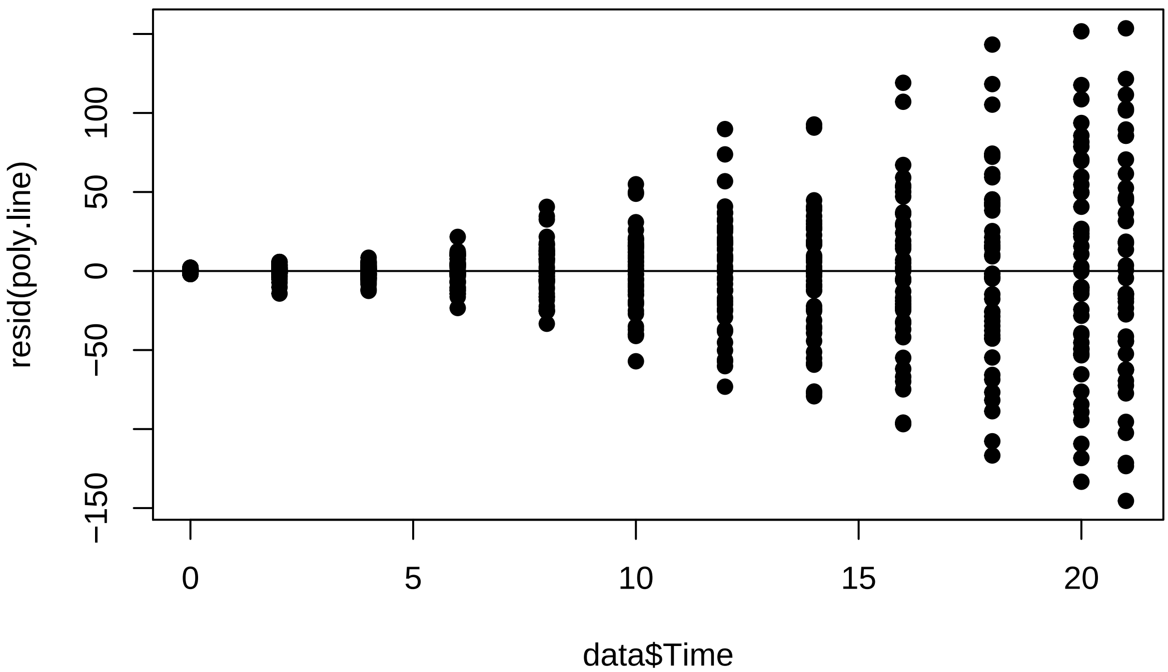


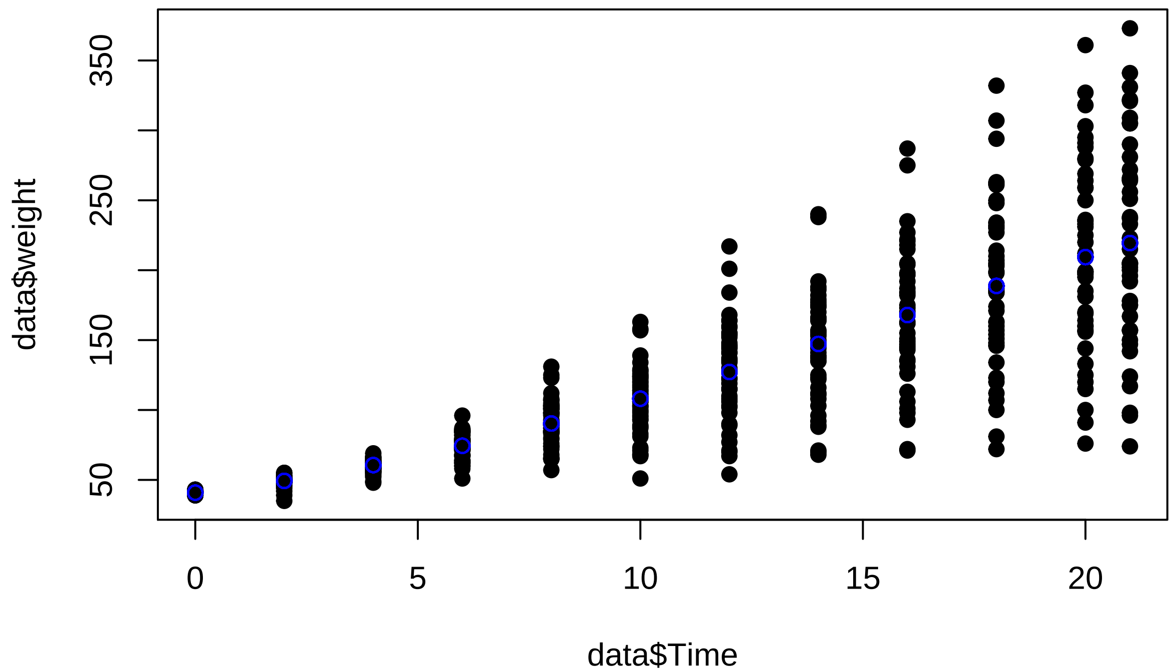
data$weight
50
150
250
350
0
5
10
15
20
data$Time



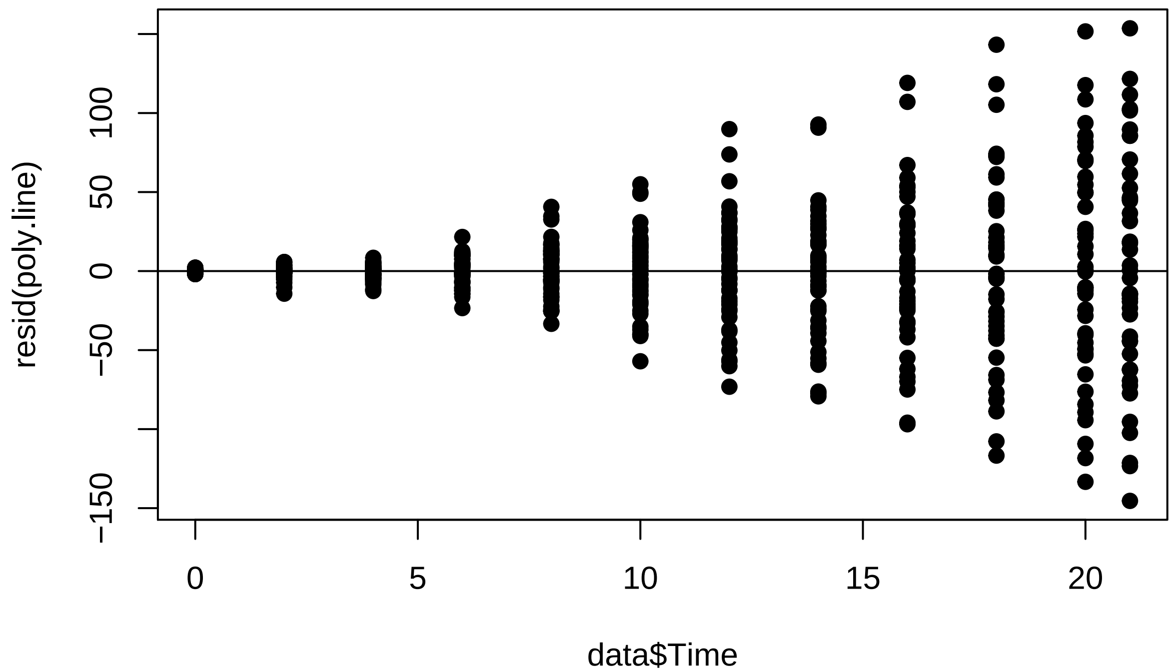
resid(poly.line)
−150
−50
0
50
100
0
5
10
15
20
data$Time

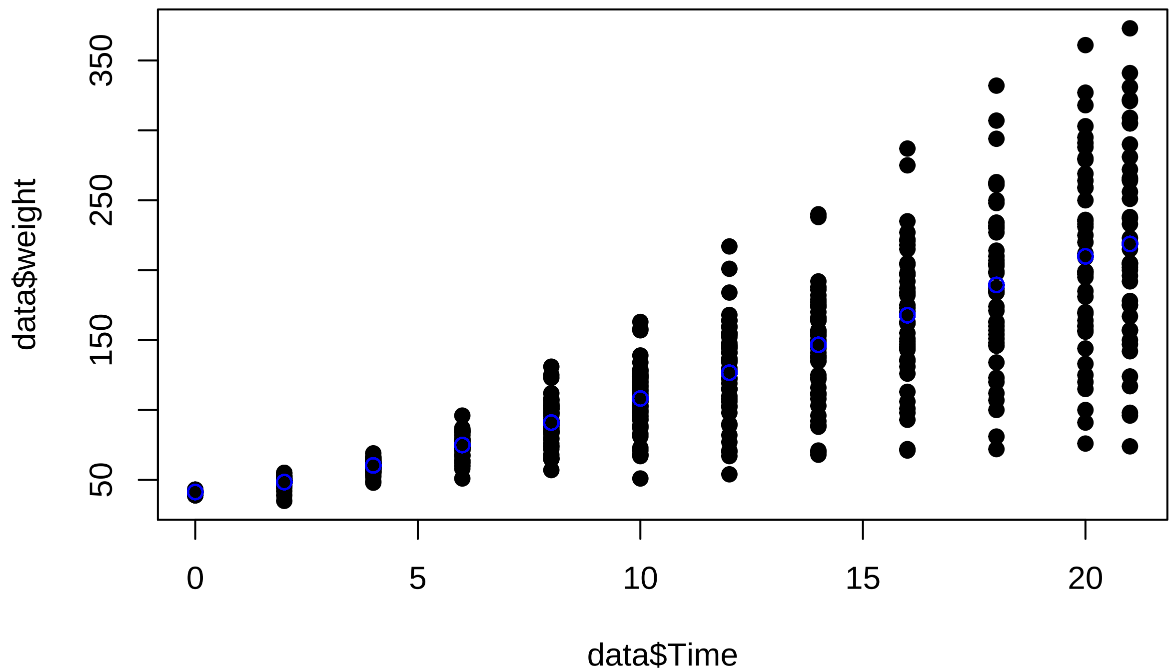

data$weight
350
250
150
50
0
5
10
15
20
data$Time

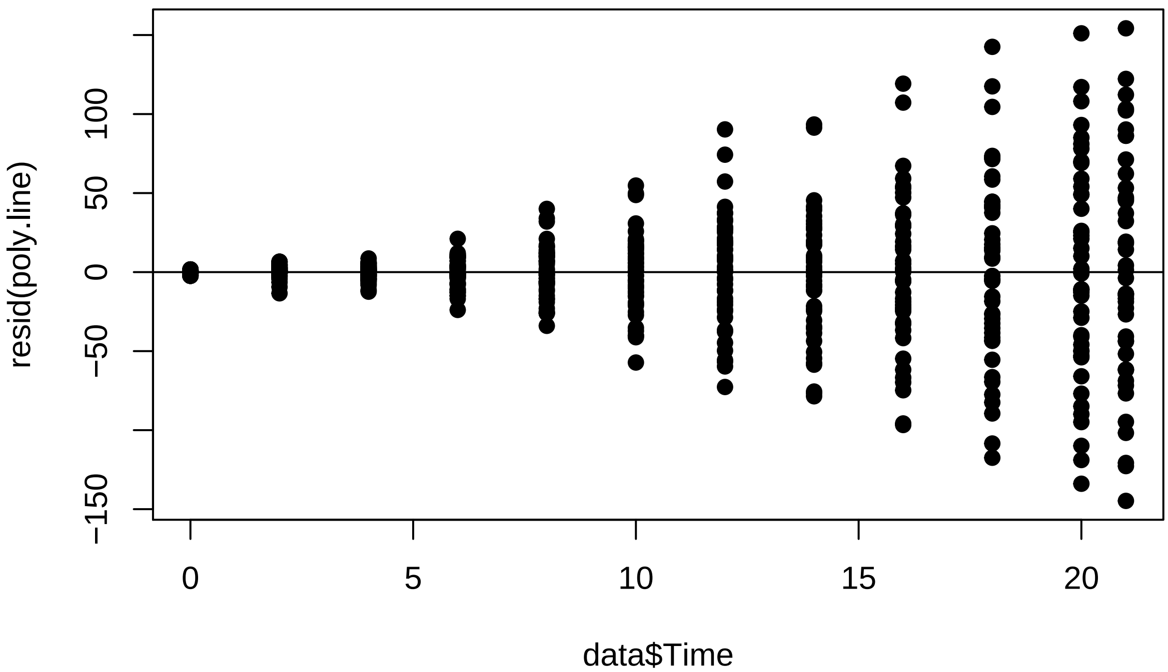

resid(poly.line)
−150
−50
0
50
100
0
5
10
15
20
data$Time



350
250
150
50
data$weight
0
5
10
15
20
data$Time

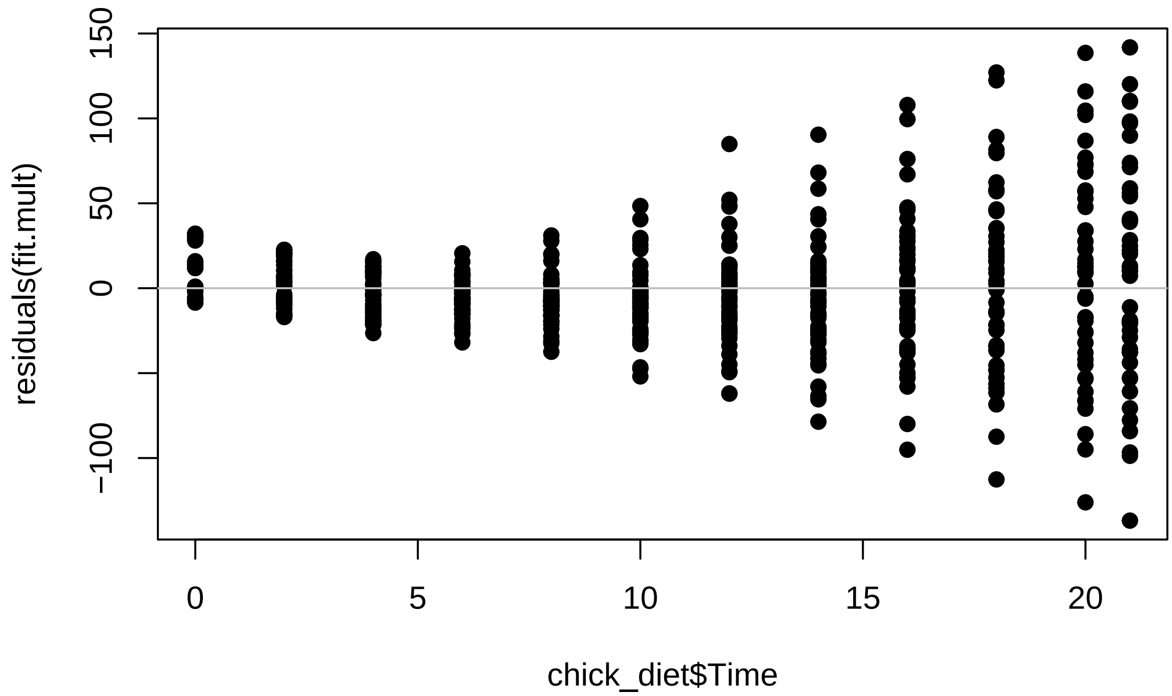

resid(poly.line)
−150
−50
0
50
100
0
5
10
15
20
data$Time


data$weight
50
150
250
350
0
5
10
15
20
data$Time

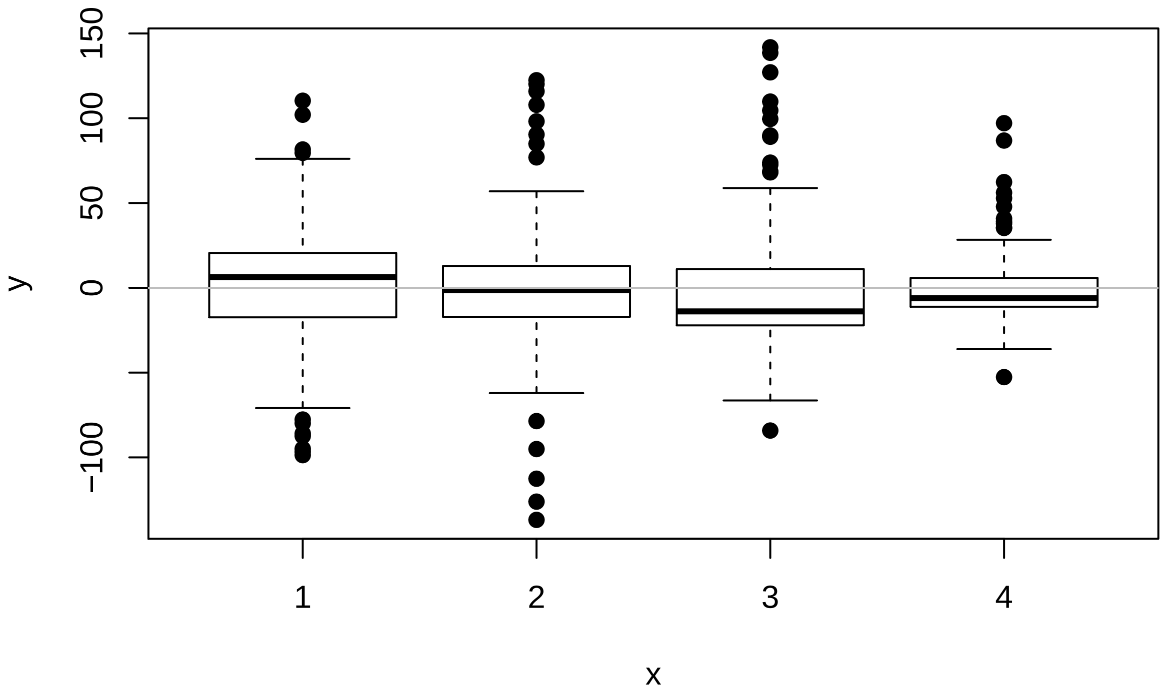


4.) Fit Multiple Regression with Diet, Show Linear and Polynomial Regression & Show Residuals

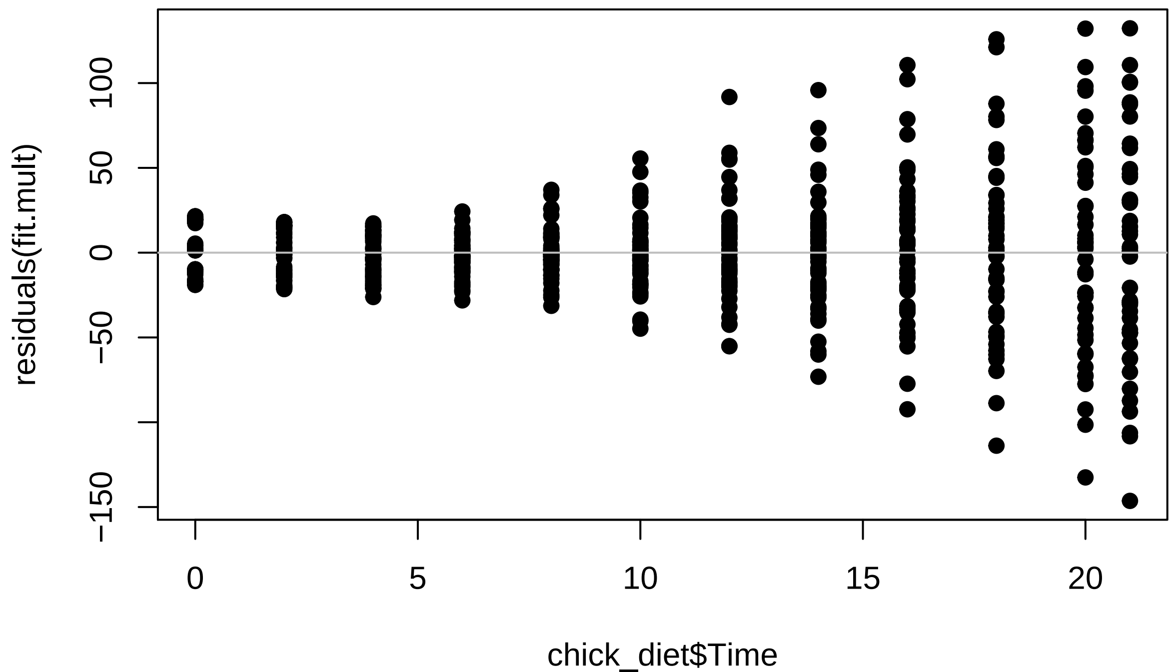
```
chick_diet<-ChickWeight%>%select(weight,Time,Diet)

fit.mult<-lm(weight~Time+Diet,data=chick_diet)

plot(chick_diet$Time, residuals(fit.mult),pch=19)

abline(h = 0, col = "grey")
```



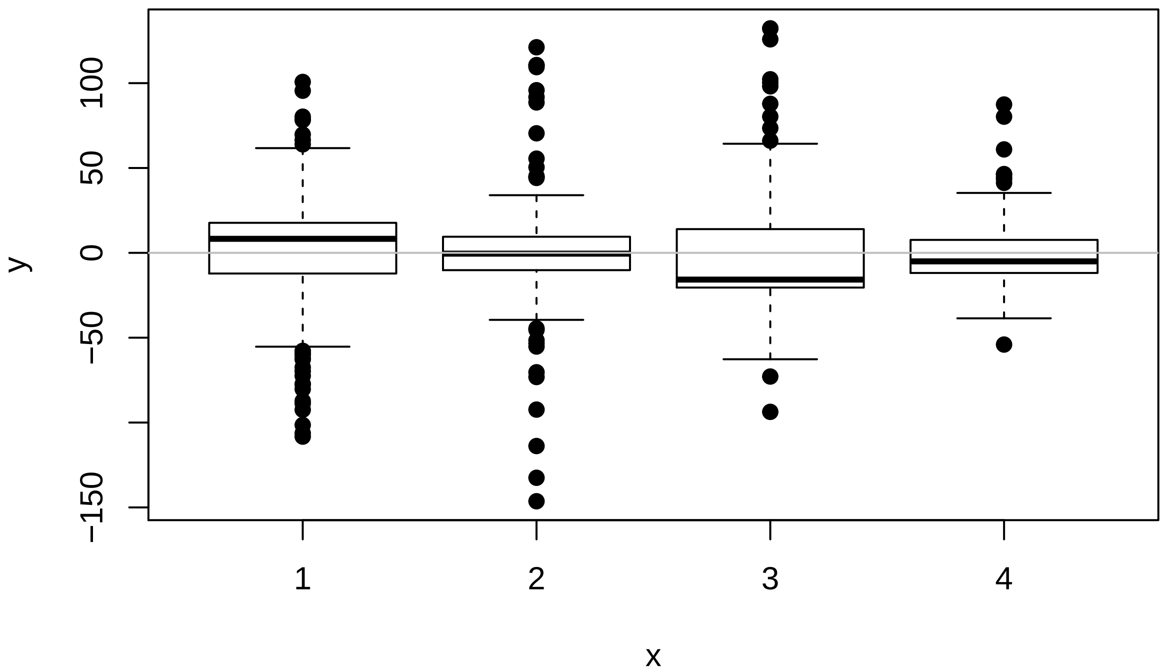

```r
plot(chick_diet$Diet, residuals(fit.mult),pch=19)

abline(h = 0, col = "grey")
```

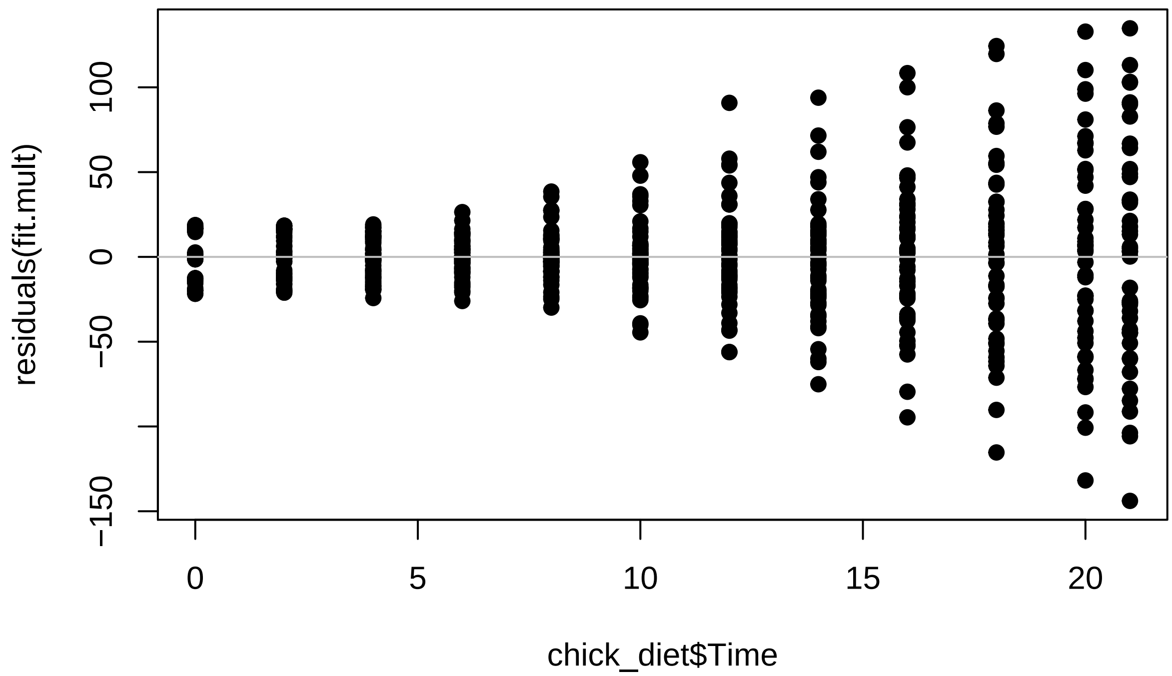


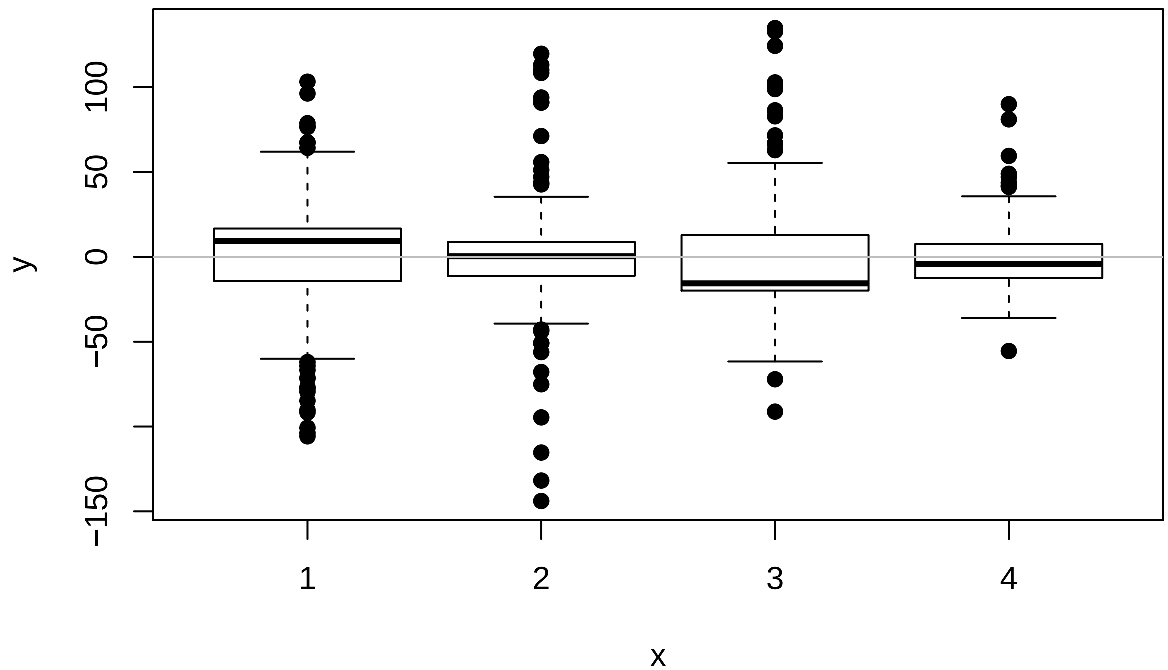

```
Polynomial_Regression_MLR(3,chick_diet)
```



residuals(fit.mult)
100
50
0
−50
−150
0
5
10
15
20
chick_diet$Time


100
50
0
−50
−150
y
1
2
3
4
x


residuals(fit.mult)
100
50
0
−50
−150
0
5
10
15
20
chick_diet$Time



Question 5:

1.) F test statistic: for B0 +B1X1 +B2X2 vs B0 +B1X1 +B2X2 +B3X3

```
file2 <- "http://www.math.mcgill.ca/yyang/regression/data/cigs.csv"
cigs <- read.csv(file2, header = TRUE)

m_reduced <- lm(CO ~ TAR+NICOTINE, data = cigs) # fit the reduced model
m_full <- lm(CO ~ TAR+NICOTINE+WEIGHT, data = cigs) # fit the full model


anova(m_full,m_reduced)[["F"]][2]
```

```
## [1] 0.001127825
```

2.) F test statistic: for B0 +B1X1 vs B0 +B1X1 +B2X2

```
m_reduced <- lm(CO ~ TAR, data = cigs) # fit the reduced model
m_full <- lm(CO ~ TAR+NICOTINE, data = cigs) # fit the full model


anova(m_full,m_reduced)[["F"]][2]
```

```
## [1] 0.4882394
```

3.) F test statistic: for B0 vs B0 +B1X1 +B2X2

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

summary(m_full)[["fstatistic"]][1]
```

```
##    value
## 124.1102
```