Linear regression model

Chandan Kumar Pandey

2022-10-16

Linear regression with only one predictor.

Student marks dataset and exploration.

The data set contain the mark scored by student in exam, hours of study and number of course they have opted for. The question we are asking in this case if, **How does number of hours of studies will impact score?**

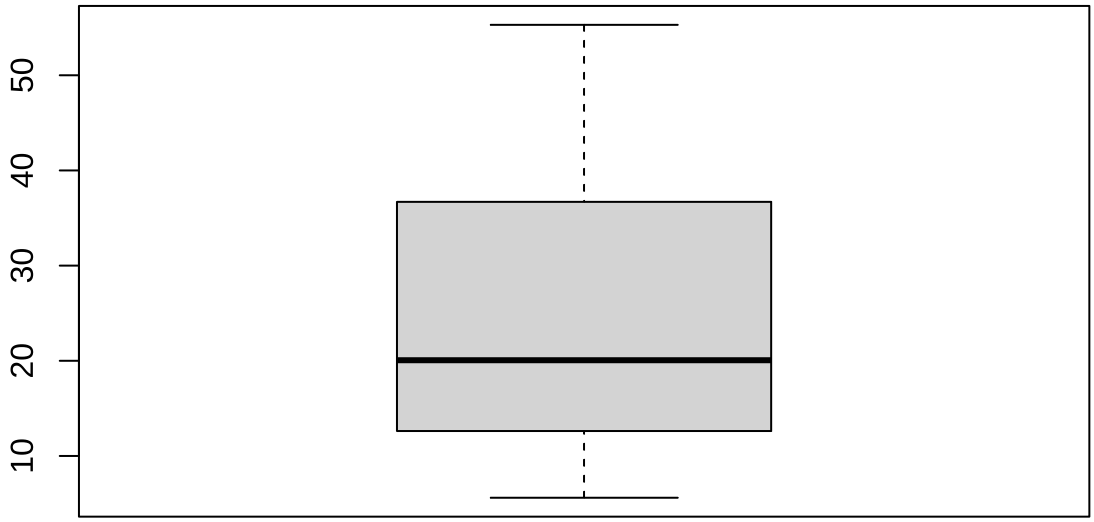
```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

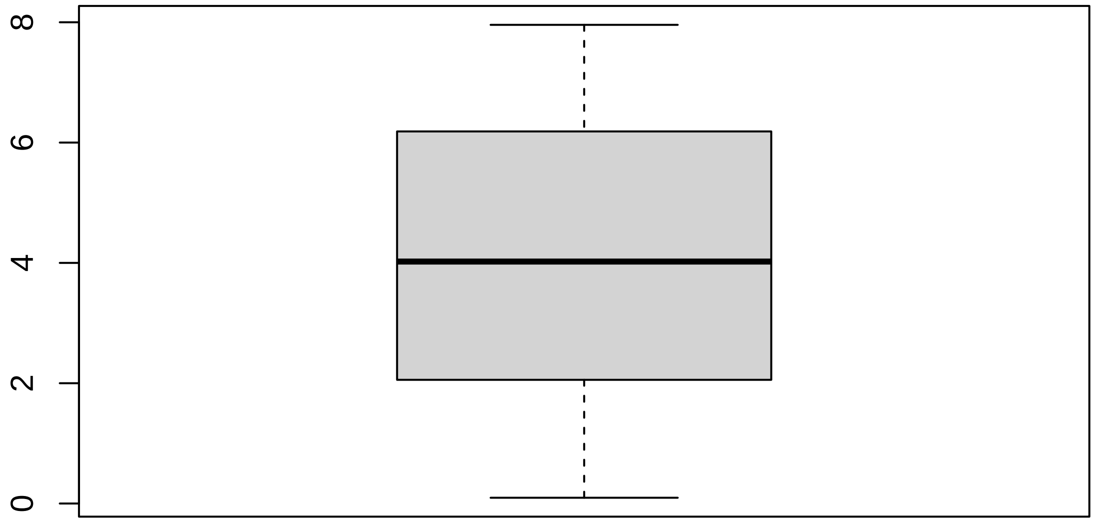
```
#Read the data set
Marks <- read.csv("data/Student_Marks.csv",header = T)
# Basic exploration of the marks data set.
summary(Marks)
```

```
##  number_courses   time_study        Marks
##  Min.   :3.00   Min.   :0.096   Min.   : 5.609
##  1st Qu.:4.00   1st Qu.:2.058   1st Qu.:12.633
##  Median :5.00   Median :4.022   Median :20.059
##  Mean   :5.29   Mean   :4.077   Mean   :24.418
##  3rd Qu.:7.00   3rd Qu.:6.179   3rd Qu.:36.676
##  Max.   :8.00   Max.   :7.957   Max.   :55.299
```

```
#graphical representations.
boxplot(Marks$Marks)
```



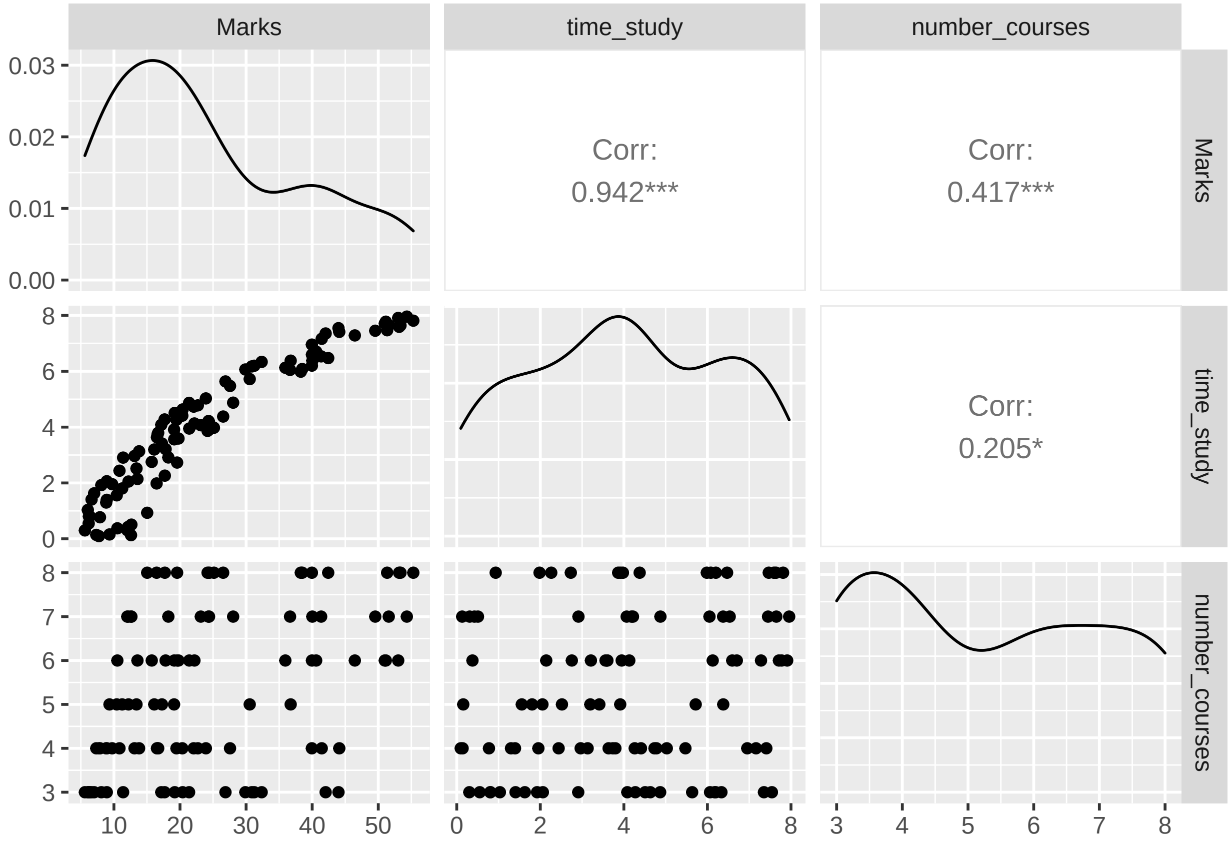
```
boxplot(Marks$time_study)
```



```
str(Marks)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ number_courses: int  3 4 4 6 8 6 3 5 4 3 ...
##  $ time_study    : num  4.508 0.096 3.133 7.909 7.811 ...
##  $ Marks         : num  19.2 7.73 13.81 53.02 55.3 ...
```

```
# The first set of data exploration is pair plot which check for
#correlation among predictor and repose.
ggpairs(Marks[,c(3,2,1)])
```



Linear regression fitting.

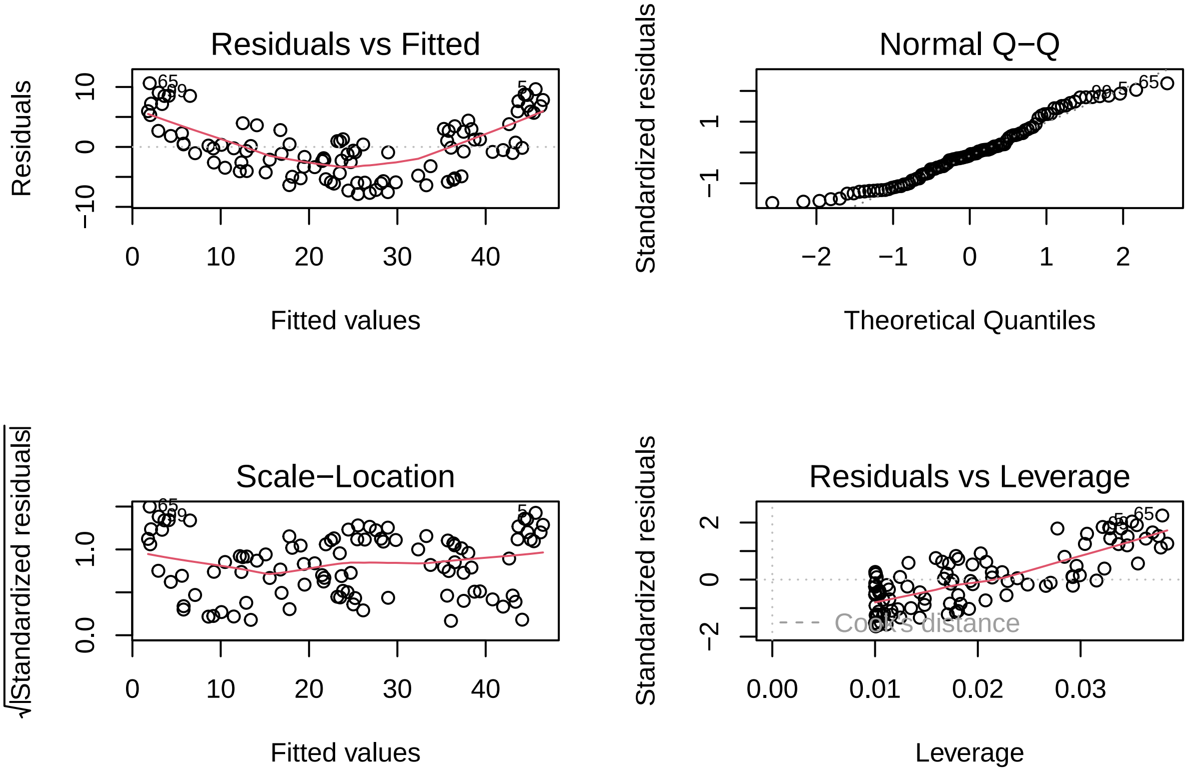
Based on Pairwise plot it is clear that Marks do follow a linear trend with time of study. In the next step we will model the linear regression

$$y = \beta 0 + \beta 1X + \epsilon$$

where

$$\epsilon = N(\mu, \sigma)$$

```
model <- lm(Marks~time_study,data = Marks)
#model's assumptions validation.
par(mfrow=c(2,2))
plot(model)
```

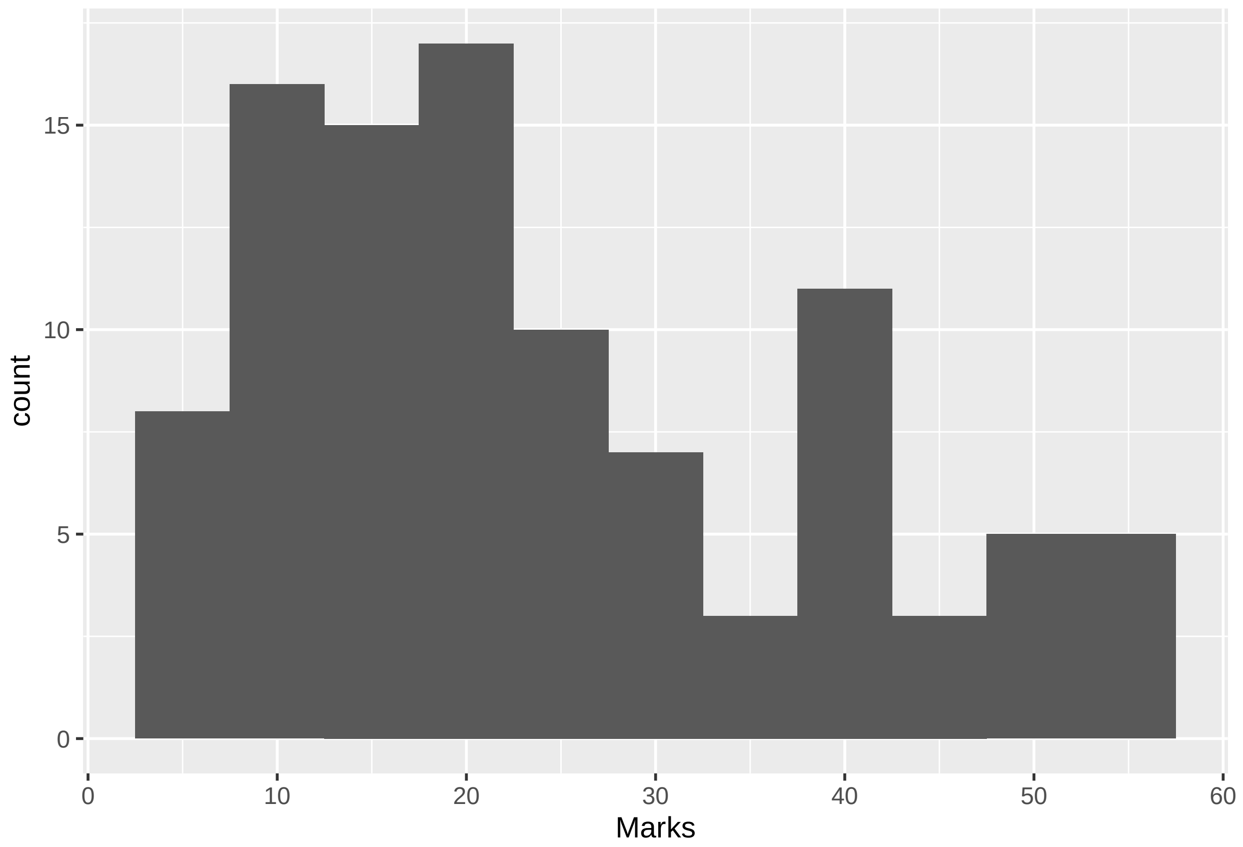


Assumption of linear regression.

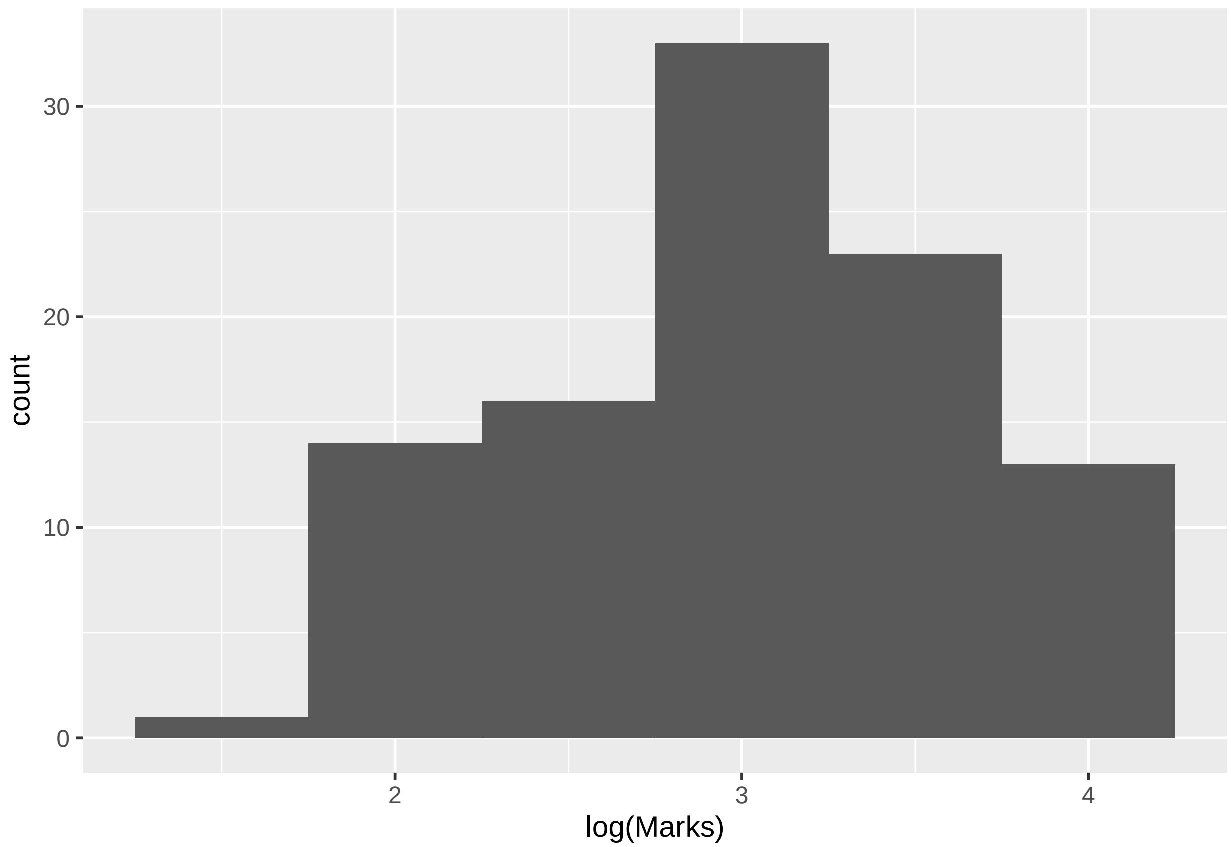
1. Linear relation.
2. Normality
3. Homogeneity of residuals variance

Based on out graphs above, the plot of Residuals vs Fitted value we can see that out data in not homogeneous. We can later confirm this by ploting histogram of Marks. **One way to overcome this problem is to do *transformation.***

```
library(ggplot2)
ggplot(data = Marks,aes(x=Marks))+geom_histogram(binwidth = 5)
```



```
##Log transformation
ggplot(data = Marks,aes(x=log(Marks)))+geom_histogram(binwidth = 0.5)
```



```
##lets check the normality by using shapiro test
shapiro.test(Marks$Marks)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Marks$Marks
## W = 0.91427, p-value = 7.082e-06
```

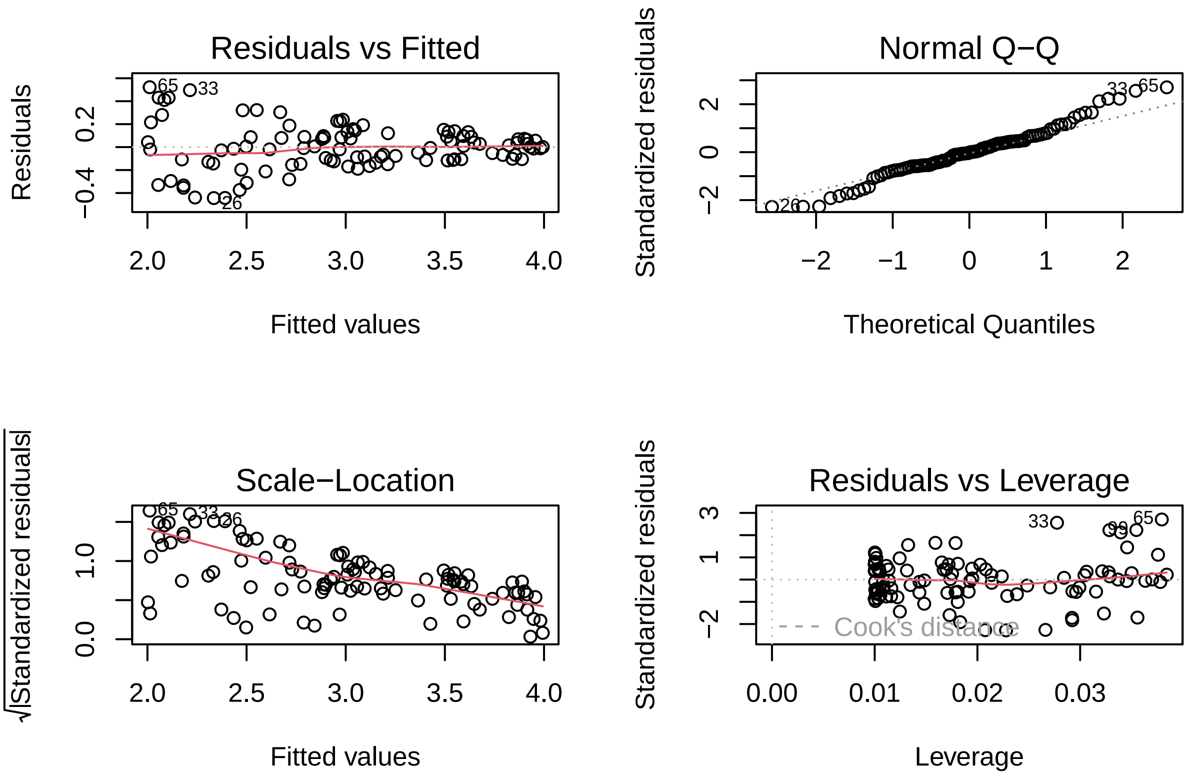
```
shapiro.test(log(Marks$Marks))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(Marks$Marks)
## W = 0.96222, p-value = 0.005805
```

```
## The value p-value have increase but not to significant level that
## distribution become normal.
```

The value p-value have increase but not to significant level that distribution become normal. However, given with data the model will improve significantly

```
model1 <- lm(log(Marks)~time_study,data = Marks)
#model's assumptions validation.
par(mfrow=c(2,2))
plot(model1)
```



```
summary(model1)
```

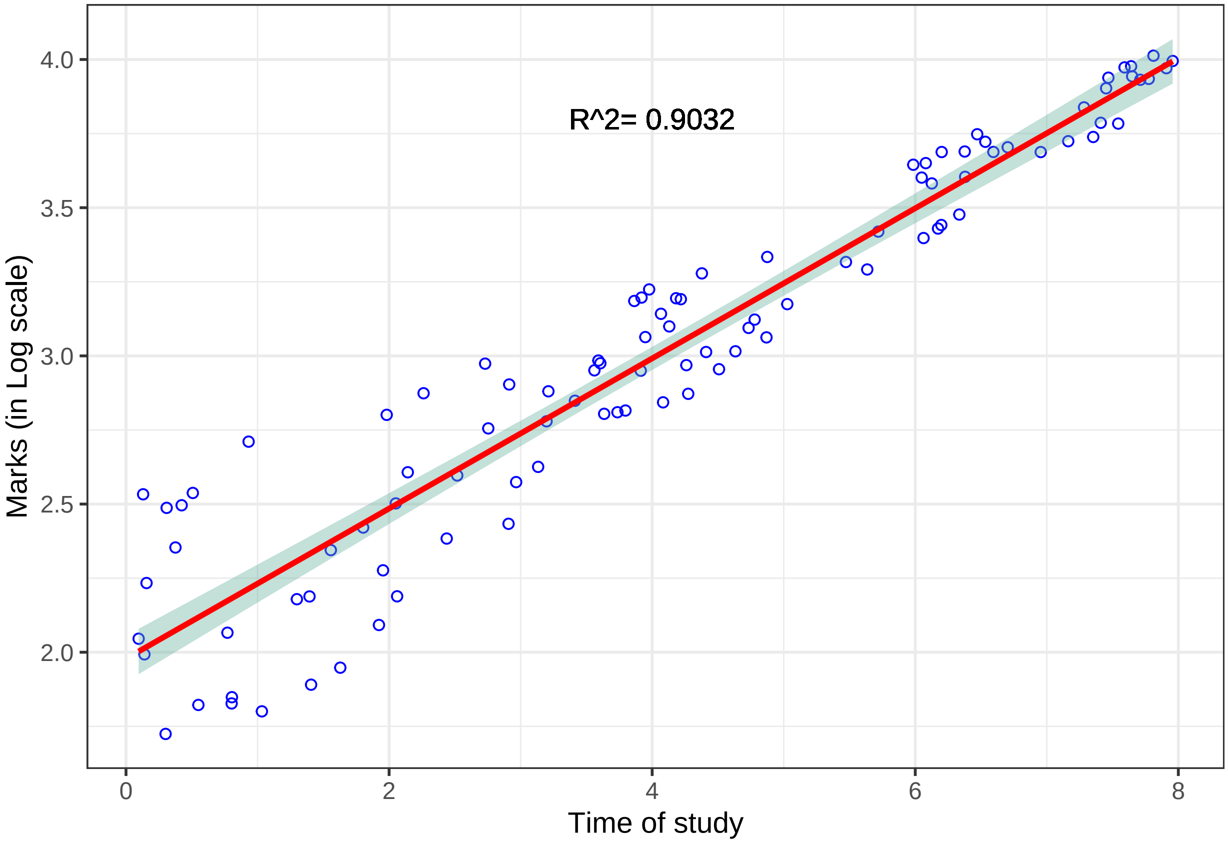
```
##
## Call:
## lm(formula = log(Marks) ~ time_study, data = Marks)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44413 -0.11272 -0.00365  0.09239  0.52174
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.978317   0.039250    50.4   <2e-16 ***
## time_study  0.253277   0.008331    30.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1967 on 98 degrees of freedom
## Multiple R-squared:  0.9041, Adjusted R-squared:  0.9032
## F-statistic: 924.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

interpretation of model with log transformation.

$$log(Marks) = 1.978 + 0.253 * studytime + \epsilon$$

```
plot_model2 <- ggplot(data = Marks,aes(x=time_study,y=log(Marks)))
plot_model2+geom_point(pch=1,col="blue")+
  geom_smooth(method=lm , color="red", fill="#69b3a2", se=TRUE)+
  theme_bw()+
  labs(x="Time of study",y="Marks (in Log scale)")+
  geom_text(mapping = aes(x=4,y=3.8,
                          label=paste("R^2=",round(summary(model1)$adj.r.squared,4))))
```

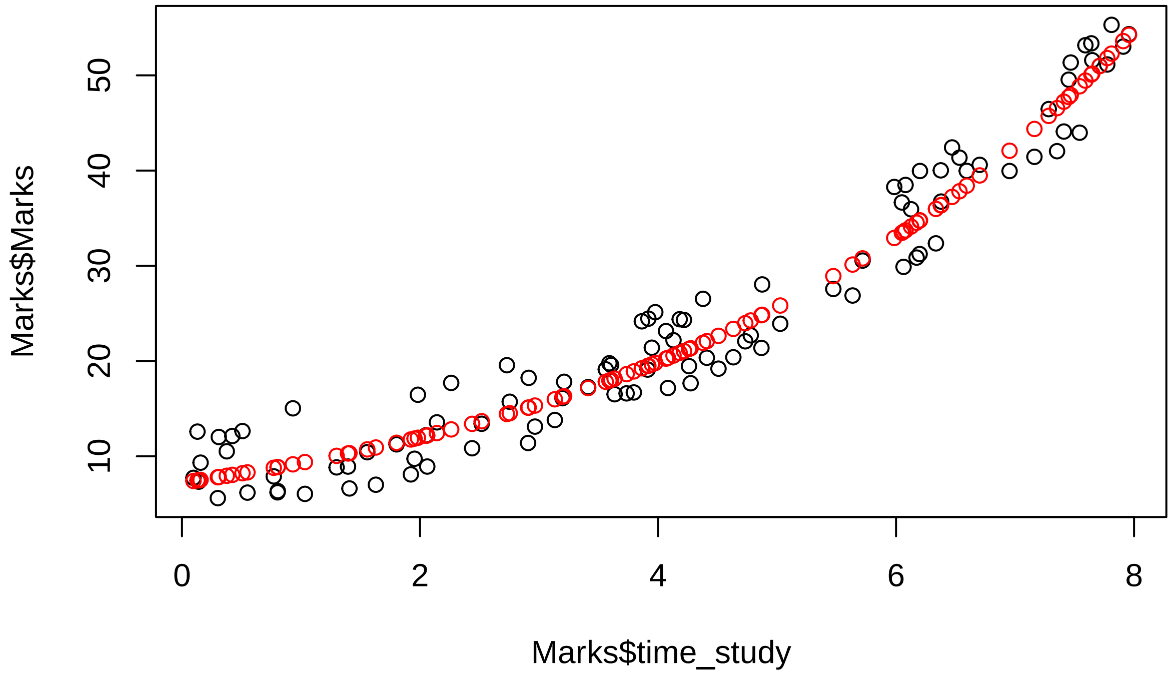
```
## `geom_smooth()` using formula 'y ~ x'
```



In order to back transform from log scale to linear scale we need to look in the use the formula

$$Marks = \exp^{B1*time+B0}$$

```
exp_pred<-exp(predict(model1,newdata = Marks))
plot(Marks$Marks~Marks$time_study)
points(exp_pred~Marks$time_study,col="red")
```



Model with more that one parameter.

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \epsilon$$

where

$$\epsilon = N(\mu, \sigma)$$

Let us use the fish morphometric data and ask question is fish length can predict its weight.

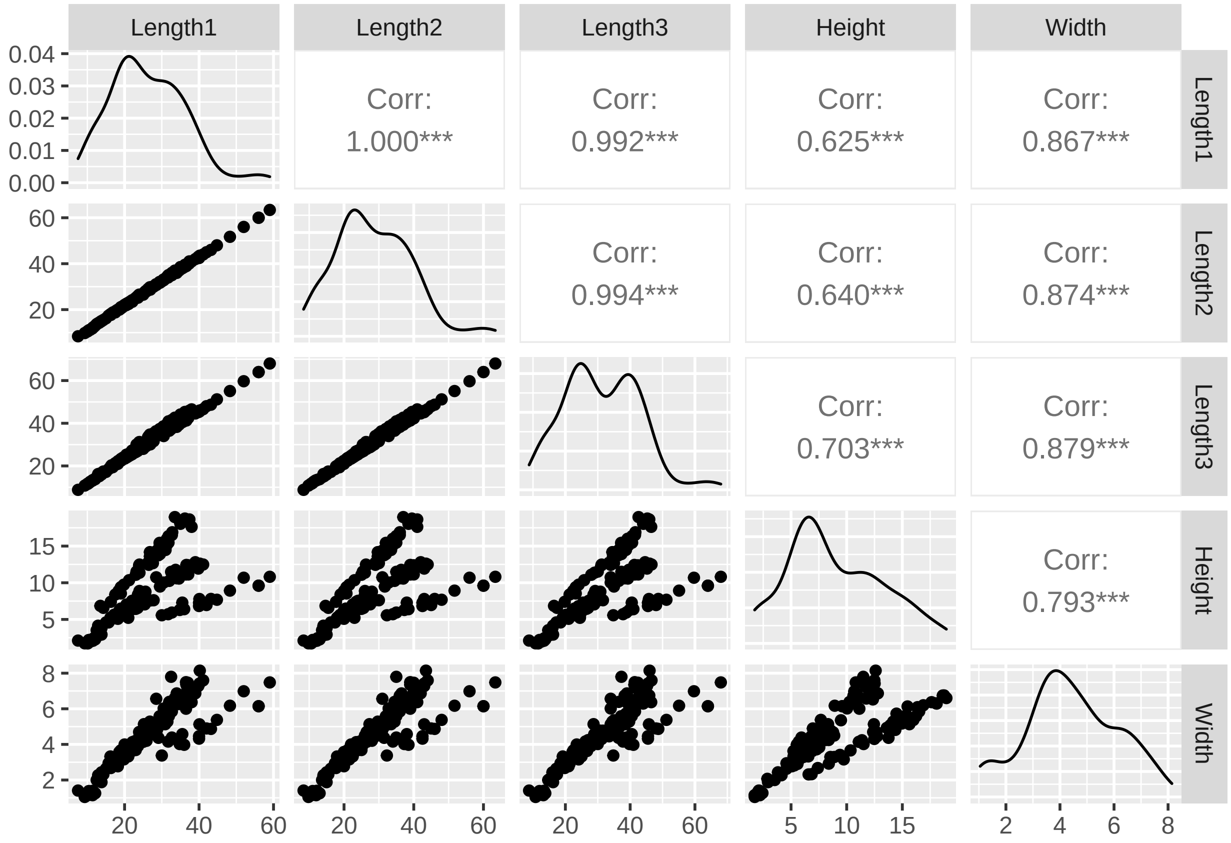
```
fish <- read.csv("data/Fish_morphometrics.csv",header = T)
# Summary and str of data
summary(fish)
```

```
##    Species              Weight          Length1         Length2
##  Length:159         Min.   :   0.0   Min.   : 7.50   Min.   : 8.40
##  Class :character   1st Qu.: 120.0   1st Qu.:19.05   1st Qu.:21.00
##  Mode  :character   Median : 273.0   Median :25.20   Median :27.30
##                     Mean   : 398.3   Mean   :26.25   Mean   :28.42
##                     3rd Qu.: 650.0   3rd Qu.:32.70   3rd Qu.:35.50
##                     Max.   :1650.0   Max.   :59.00   Max.   :63.40
##     Length3          Height           Width
##  Min.   : 8.80   Min.   : 1.728   Min.   :1.048
##  1st Qu.:23.15   1st Qu.: 5.945   1st Qu.:3.386
##  Median :29.40   Median : 7.786   Median :4.248
##  Mean   :31.23   Mean   : 8.971   Mean   :4.417
##  3rd Qu.:39.65   3rd Qu.:12.366   3rd Qu.:5.585
##  Max.   :68.00   Max.   :18.957   Max.   :8.142
```

```
str(fish)
```

```
## 'data.frame':    159 obs. of  7 variables:
##  $ Species: chr  "Bream" "Bream" "Bream" "Bream" ...
##  $ Weight : num  242 290 340 363 430 450 500 390 450 500 ...
##  $ Length1: num  23.2 24 23.9 26.3 26.5 26.8 26.8 27.6 27.6 28.5 ...
##  $ Length2: num  25.4 26.3 26.5 29 29 29.7 29.7 30 30 30.7 ...
##  $ Length3: num  30 31.2 31.1 33.5 34 34.7 34.5 35 35.1 36.2 ...
##  $ Height : num  11.5 12.5 12.4 12.7 12.4 ...
##  $ Width  : num  4.02 4.31 4.7 4.46 5.13 ...
```

```
# pairwise plot
ggpairs(fish[,3:7])
```

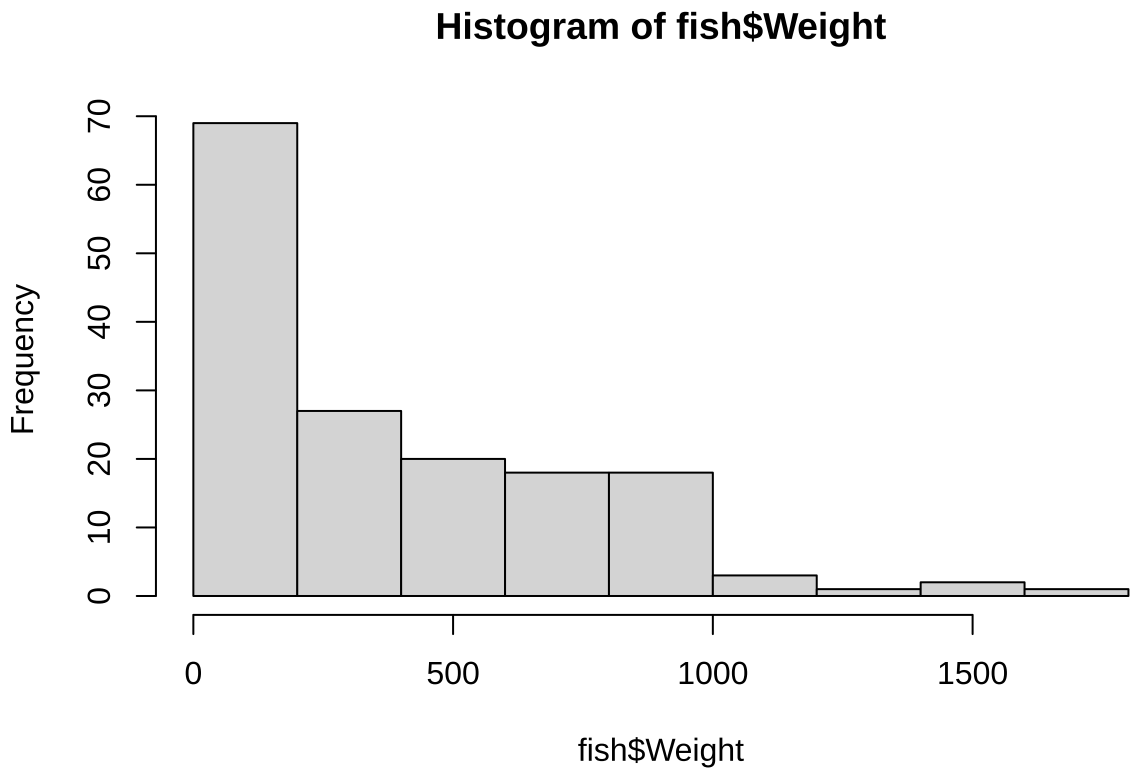


```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
fish_group<-fish%>%group_by(Species)%>%summarise(count=n(),mean(Weight),mean(Length3),sd(Weight))
fish<-fish%>%mutate(avg_length = (Length1+Length2+Length3)/3)
hist(fish$Weight)
```



```
#Install.packages("devtools")
devtools::install_github("cardiomoon/ggiraphExtra")
```

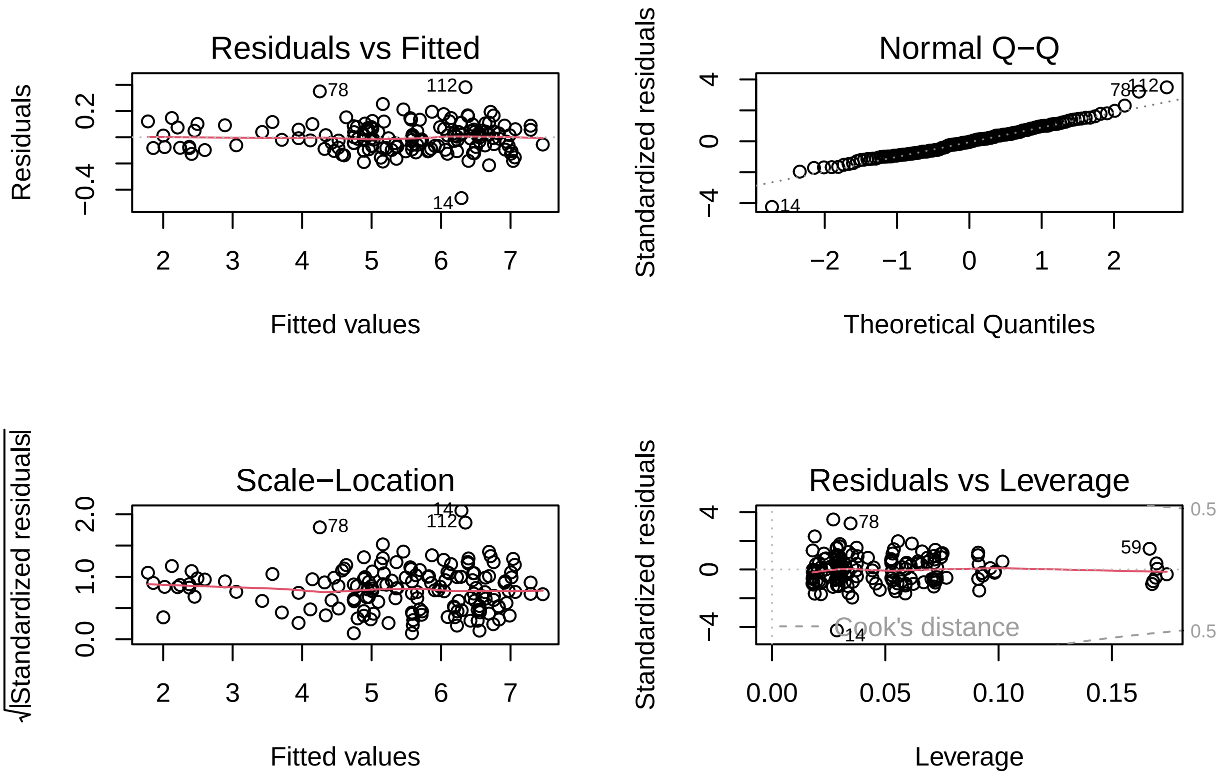
```
## Skipping install of 'ggiraphExtra' from a github remote, the SHA1 (c2c1ce81) has not changed since la
##   Use `force = TRUE` to force installation
```

```
library(ggiraphExtra)
require(ggiraph)
```

```
## Loading required package: ggiraph
```

```
model_fish1 <- lm(Weight~avg_length+Species,data = fish)
fish <- fish[fish$Weight!=0,] ## wrong data point
model_log <- lm(log(Weight)~avg_length+Species,data = fish)
#we make log log transformation
```

```
fish <- fish%>%mutate(log_w = log(Weight),log_avg_len=log(avg_length))
model_log_log <- lm(log_w~log_avg_len+Species,data = fish)
par(mfrow=c(2,2))
plot(model_log_log)
```



```
summary.lm(model_log_log)
```

```
##
## Call:
## lm(formula = log_w ~ log_avg_len + Species, data = fish)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46466 -0.07484 -0.00418  0.06379  0.38221
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4.57249    0.13214 -34.603  < 2e-16 ***
## log_avg_len        3.10950    0.03718  83.625  < 2e-16 ***
## SpeciesParkki      0.12178    0.04276   2.848  0.00502 **
## SpeciesPerch      -0.12573    0.02569  -4.894 2.53e-06 ***
## SpeciesPike       -0.86944    0.03447 -25.226  < 2e-16 ***
## SpeciesRoach      -0.14879    0.03510  -4.239 3.91e-05 ***
## SpeciesSmelt      -0.79600    0.05204 -15.295  < 2e-16 ***
## SpeciesWhitefish   0.02280    0.04919   0.464  0.64362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1111 on 150 degrees of freedom
## Multiple R-squared:  0.9933, Adjusted R-squared:  0.993
## F-statistic:  3174 on 7 and 150 DF,  p-value: < 2.2e-16
```

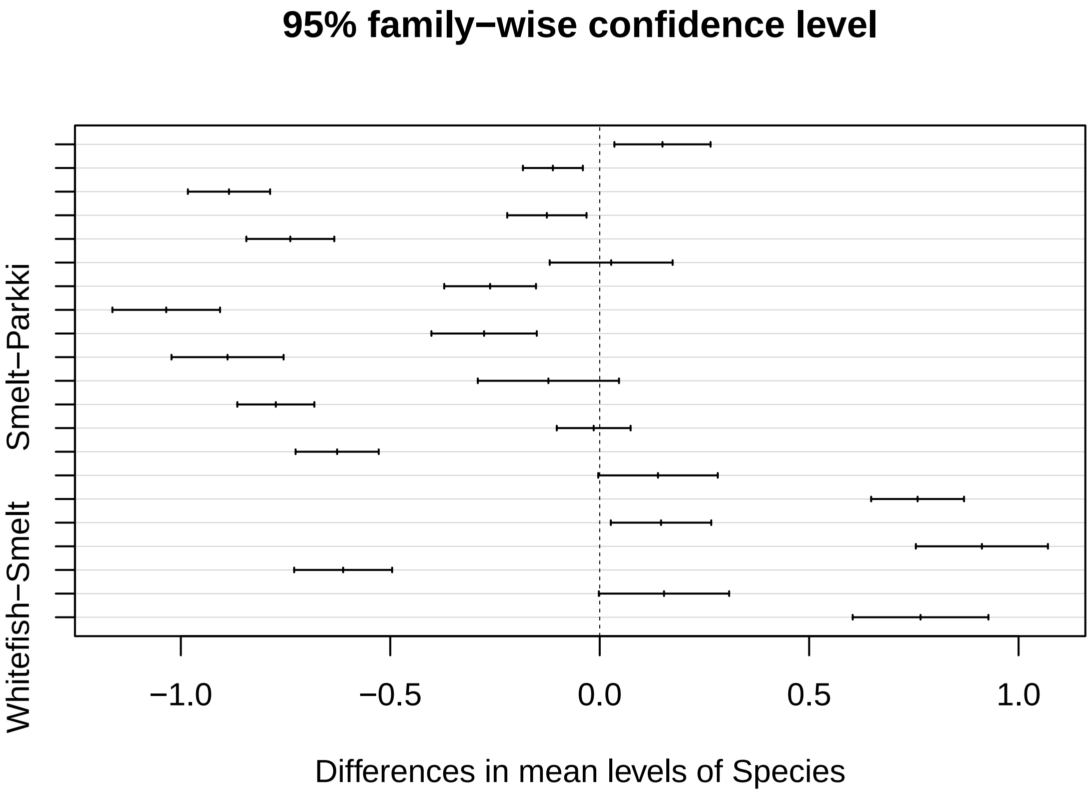
```
summary.aov(model_log_log)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## log_avg_len    1 258.61  258.61 20949.4 <2e-16 ***
## Species        6  15.66    2.61   211.4 <2e-16 ***
## Residuals    150   1.85    0.01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

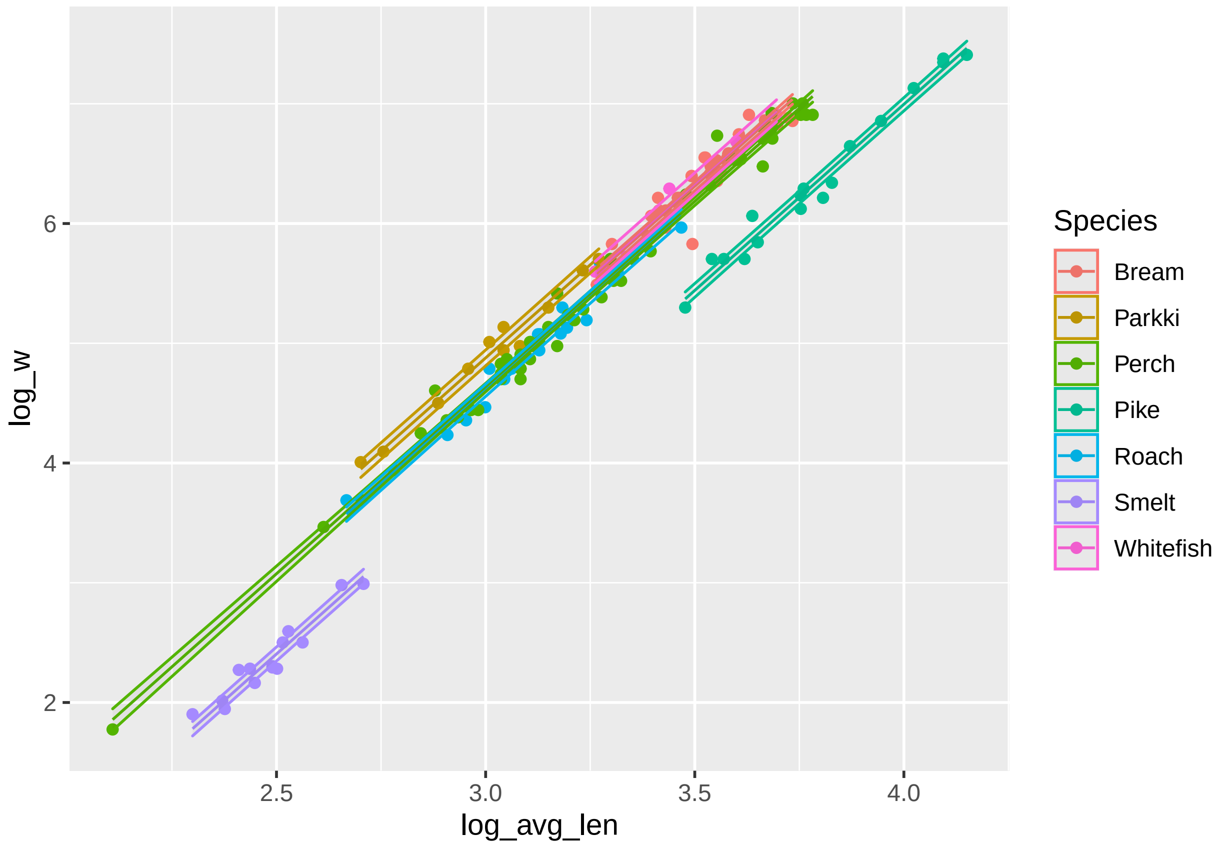
```
model_log_log_aov <- aov(log_w~log_avg_len+Species,data = fish)
par(mfrow=c(1,1))
#posthoc
Tuk <-TukeyHSD(model_log_log_aov, "Species",data = fish)
```

```
## Warning in replications(paste("~", xx), data = mf): non-factors ignored:
## log_avg_len
```

```
plot(Tuk)
```



```
#predicting the values
pred_value<-predict.lm(model_log_log,interval = "confidence")
pred_value <- as.data.frame(pred_value)
fish <- cbind(fish,pred_value)
## plotting the model
plot <- ggplot(data = fish, aes(x=log_avg_len,y=log_w,col=Species))
plot+geom_point()+
  geom_line(aes(y=fit))+
  geom_ribbon(aes(ymin=lwr,ymax=upr),alpha=0.05)
```


log_w
log_avg_len
Species
Bream
Parkki
Perch
Pike
Roach
Smelt
Whitefish