Red-Team-Archosaur

Red Team

```r
# Loading the necessary packages & libraries
library("ggplot2")
library("olsrr")
```

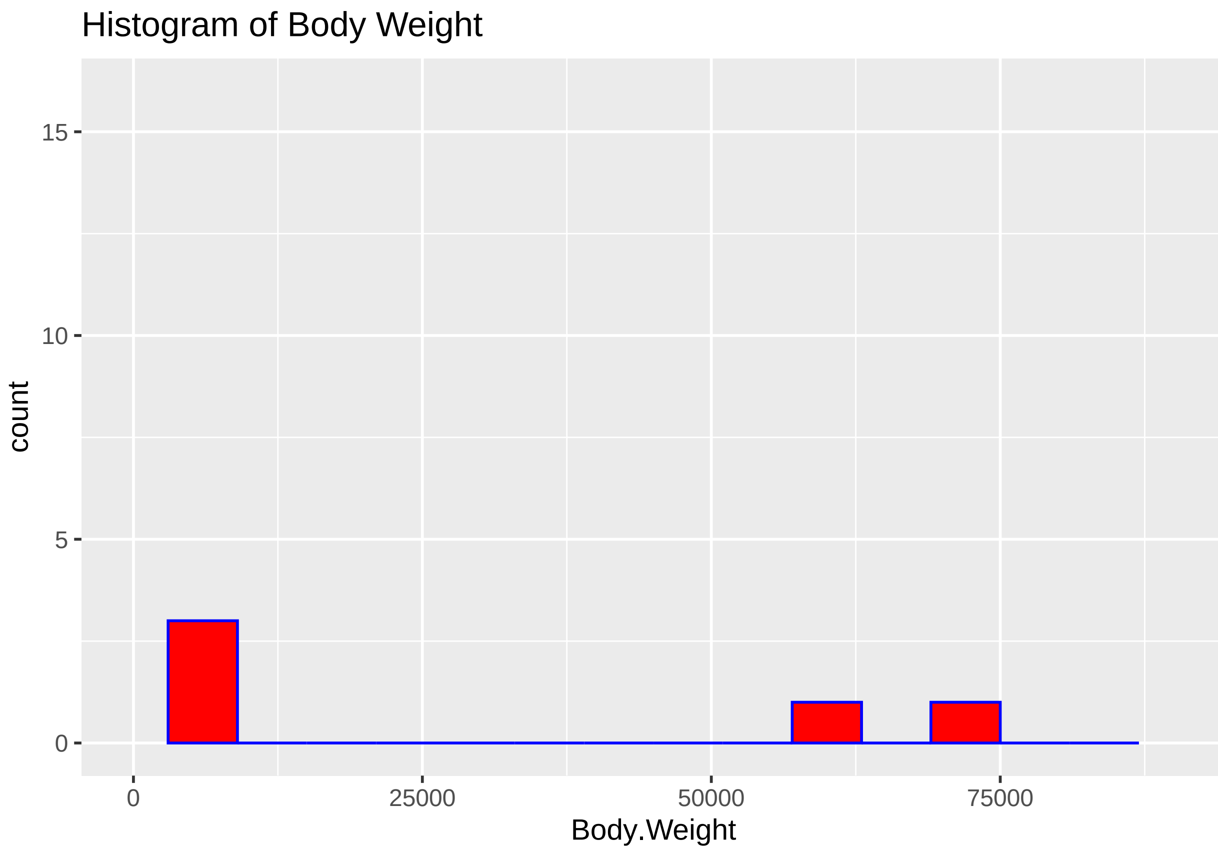
```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
# Loading the "archosaur" data set
data <- read.csv('archosaur.csv', header = TRUE)
```

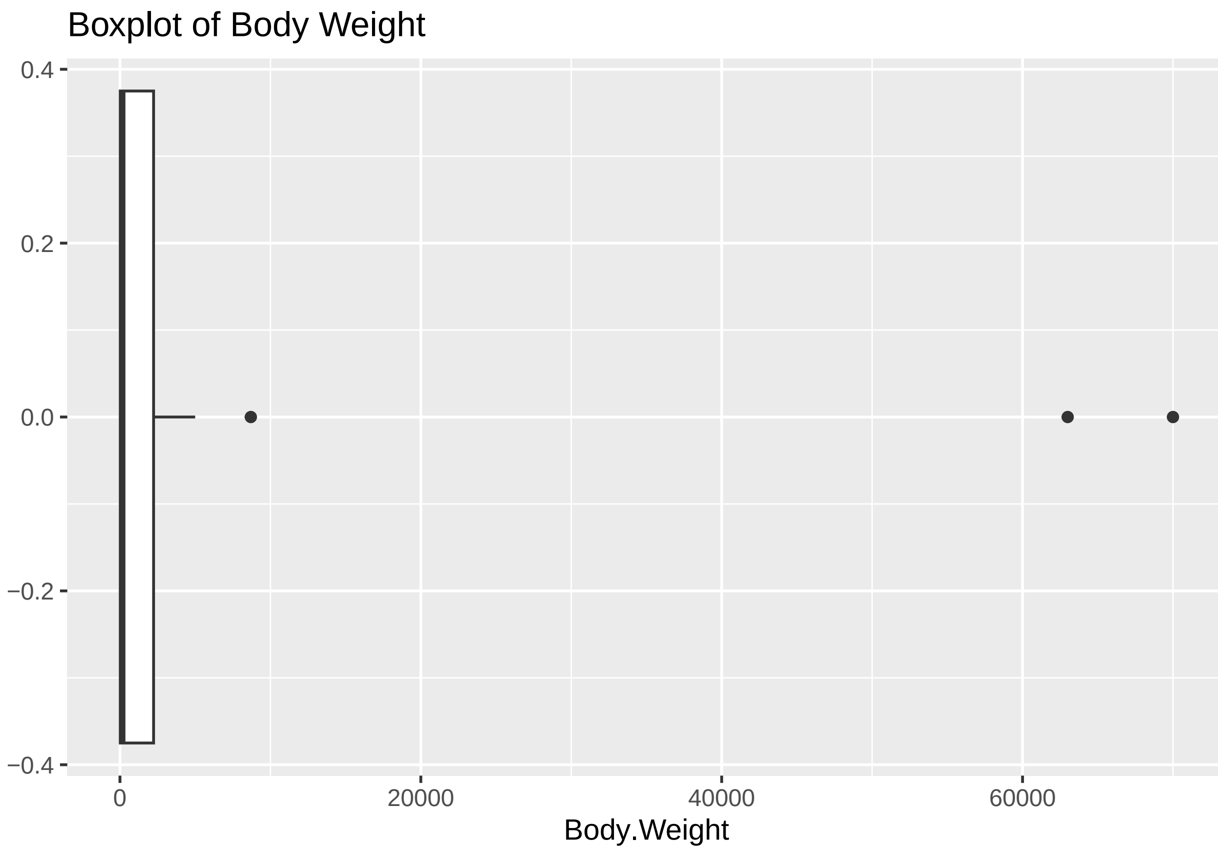
Body weight Graphs

```r
ggplot(data, aes(x = Body.Weight)) +
  geom_histogram(bins = 16, color = "blue", fill = "red") +
  ggtitle("Histogram of Body Weight") + scale_x_continuous(limits = c(0,90000))
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```




```
ggplot(data, aes(x = Body.Weight)) +
  geom_boxplot() +
  ggtitle("Boxplot of Body Weight")
```



Calculating the mean of Body weight

```
xbar <- mean(data$Body.Weight)
xbar
```

```
## [1] 7472.371
```

Calculating the standard deviation of body weight

```
sd <- sd(data$Body.Weight)
sd
```

```
## [1] 19770.46
```

```
# Set n to a value of 21
n = 21
```

Calculating the standard error mean of the body weight

```
standard_error_mean <- sd/sqrt(n)
standard_error_mean
```

```
## [1] 4314.268
```

```
margin <- qt(0.975,df = n-1) * sd/sqrt(n)

lowerinterval <- xbar - margin
lowerinterval
```

```
## [1] -1527.033
```

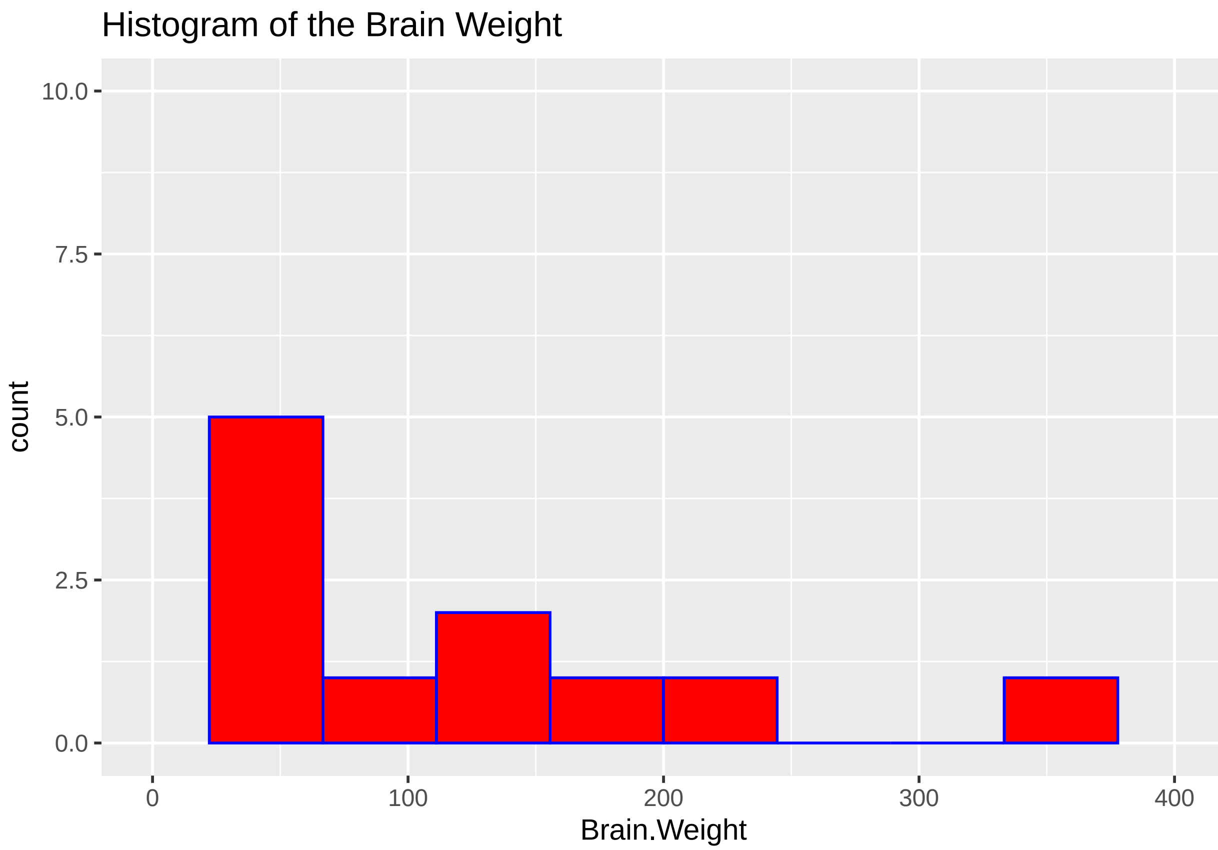
```
upperinterval <- xbar + margin
upperinterval
```

```
## [1] 16471.78
```

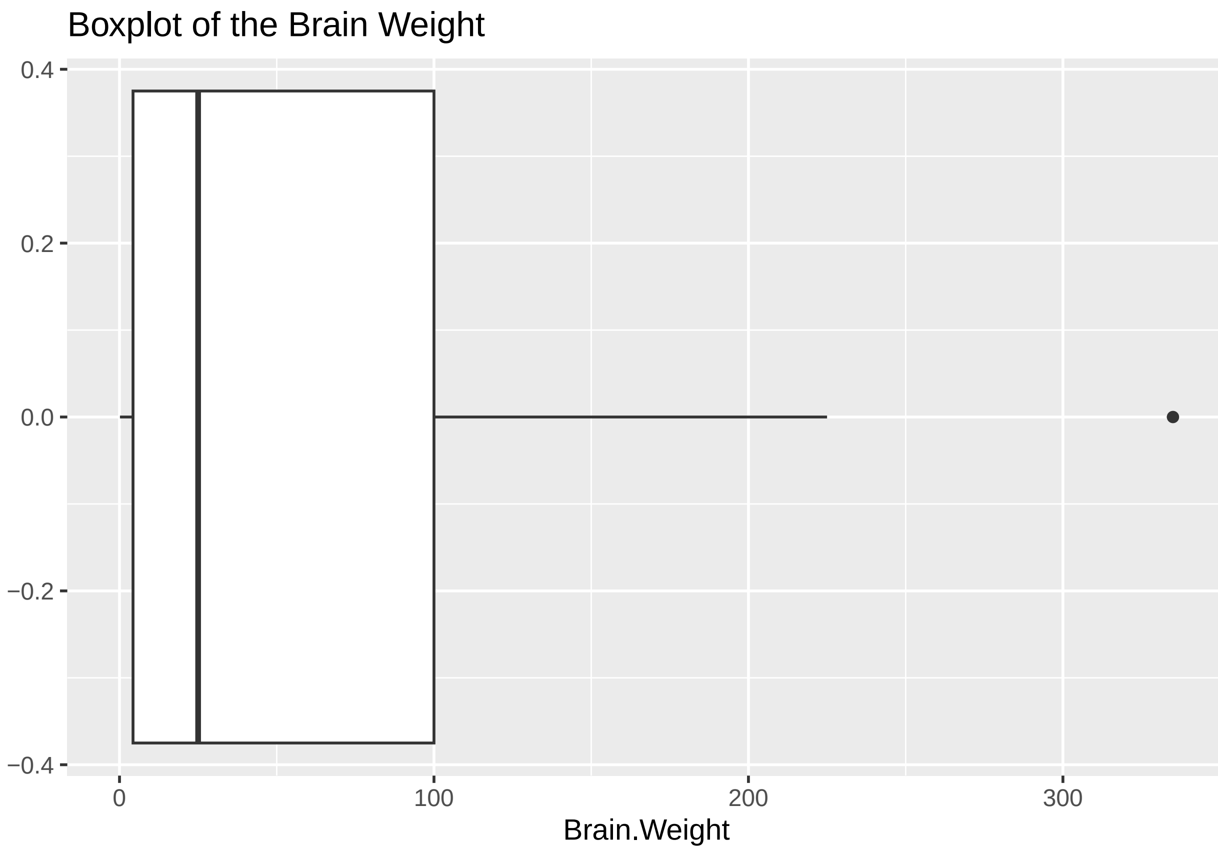
Brain Weight graphs

```
ggplot(data, aes(x = Brain.Weight)) +
  geom_histogram(bins = 10, color = "blue", fill = "red") +
  ggtitle("Histogram of the Brain Weight") +
  scale_x_continuous(limits = c(0,400))
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```




```
ggplot(data, aes(x = Brain.Weight)) +
  geom_boxplot() +
  ggtitle("Boxplot of the Brain Weight")
```



Calculating the mean of the Brain Weight

```
xbar <- mean(data$Brain.Weight)
xbar
```

```
## [1] 64.94086
```

Calculating the standard deviation of the Brain Weight

```
sd <- sd(data$Brain.Weight)
sd
```

```
## [1] 90.15867
```

```
# Set n to a value of 21
n = 21
```

Calculating the standard error mean of the brain weight

```
standard_error_mean <- sd/sqrt(n)
standard_error_mean
```

```
## [1] 19.67423
```

```
margin <- qt(0.975,df = n-1)*sd/sqrt(n)

lowerinterval <- xbar - margin
lowerinterval
```

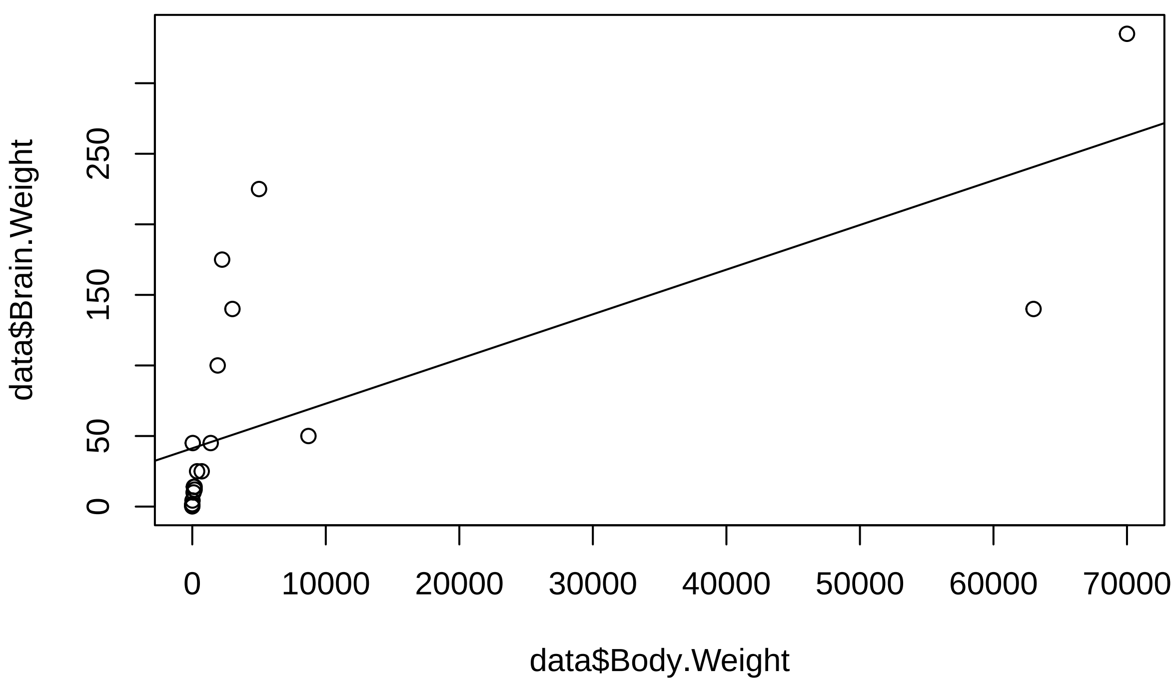
```
## [1] 23.90112
```

```
upperinterval <- xbar + margin
upperinterval
```

```
## [1] 105.9806
```

Regression Model between the body weight and data weight before the transformation occurs

```
plot(data$Brain.Weight~data$Body.Weight)
abline(lm(data$Brain.Weight~data$Body.Weight))
```

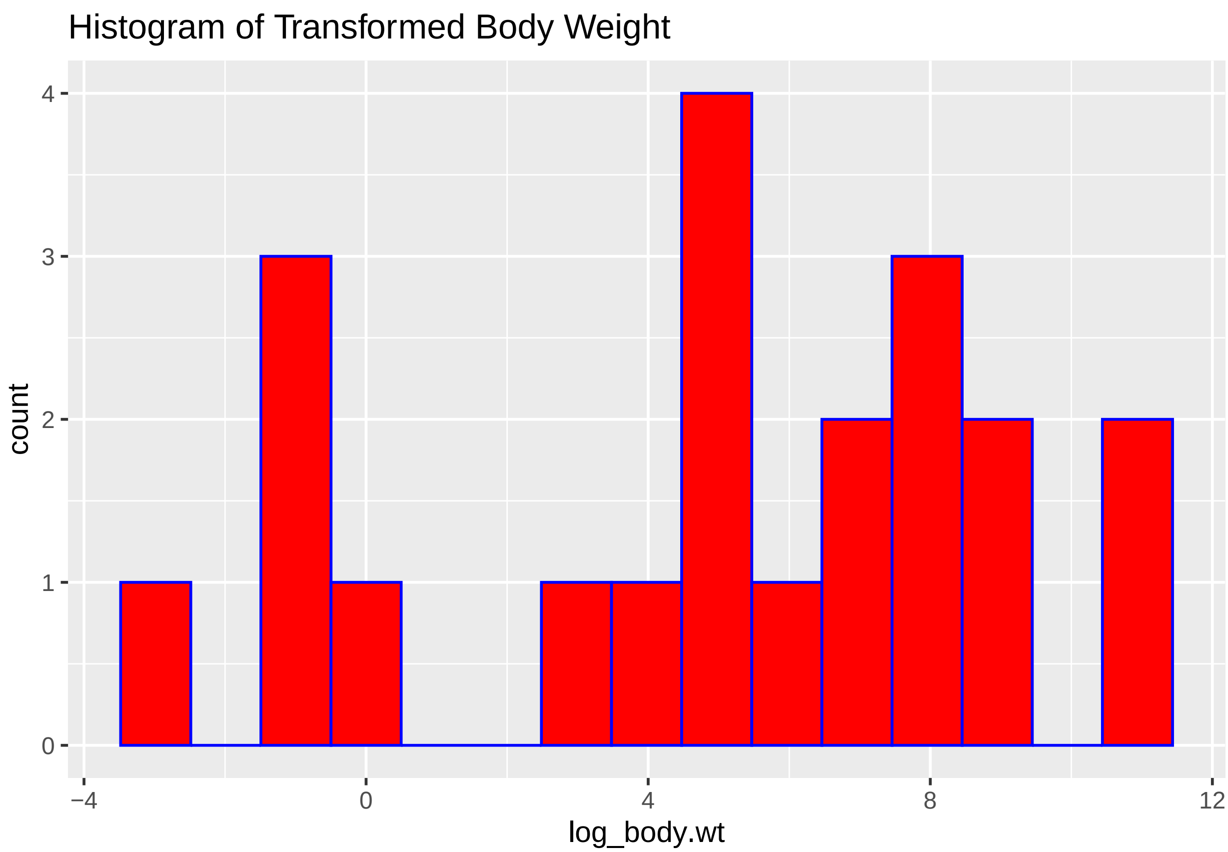


Transform ing the body weight and data weight

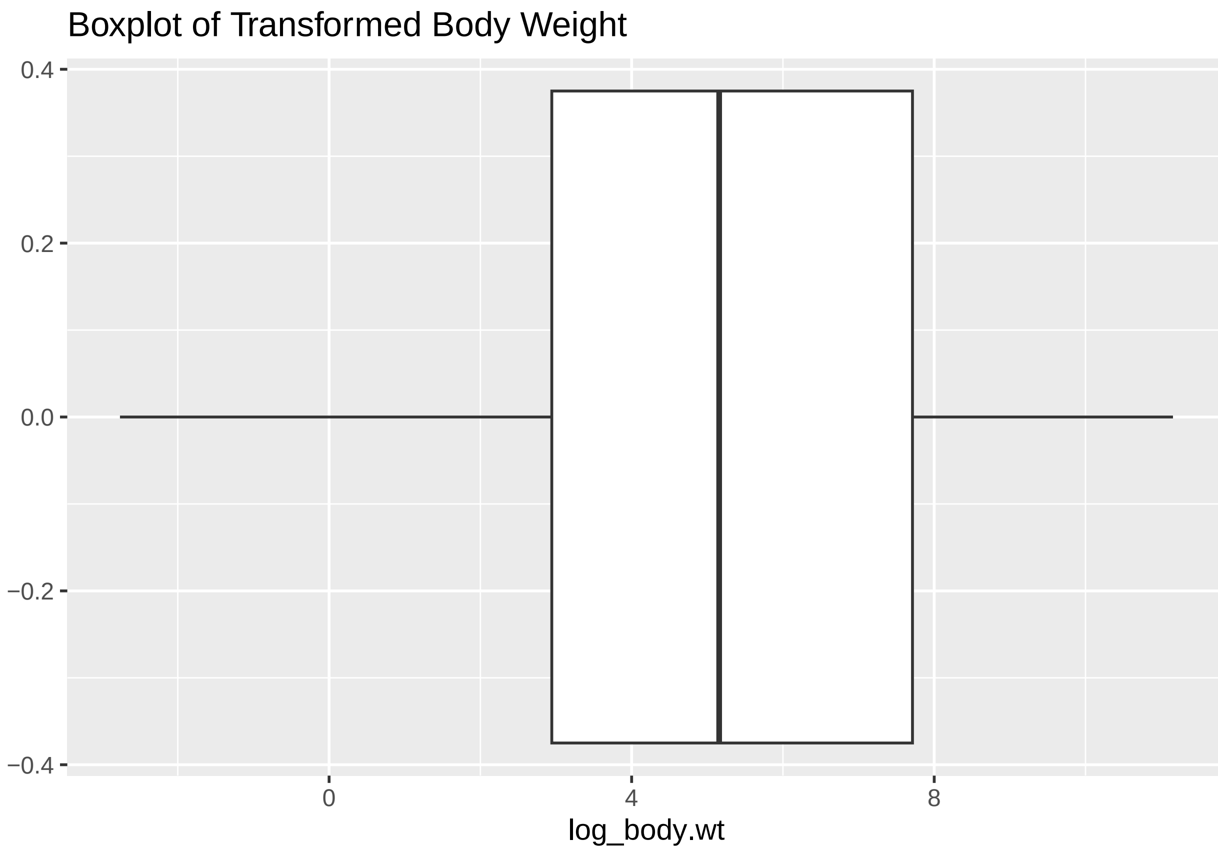
```
data$log_body.wt = log(data$Body.Weight)
data$log_data.wt = log(data$Brain.Weight)
```

Transformed body weight Graphs

```
ggplot(data, aes(x = log_body.wt)) +
  geom_histogram(bins = 15, color = "blue", fill = "red") +
  ggtitle("Histogram of Transformed Body Weight")
```




```
ggplot(data, aes(x = log_body.wt)) + geom_boxplot() +
  ggtitle("Boxplot of Transformed Body Weight")
```



Calculating the mean for the transformed body weight

```
xbar <- mean(data$log_body.wt)
xbar
```

```
## [1] 4.946397
```

Calculating the standard deviation for the transformed body weight

```
sd <- sd(data$log_body.wt)
sd
```

```
## [1] 4.044355
```

```
# Setting the n to a value of 21
n = 21
```

Calculating the standard error mean for the transformed body weight

```
standard_error_mean <- sd/sqrt(n)
standard_error_mean
```

```
## [1] 0.8825506
```

```
margin <- qt(0.975,df = n-1) * sd/sqrt(n)

lowerinterval <- xbar - margin
lowerinterval
```

```
## [1] 3.105429
```

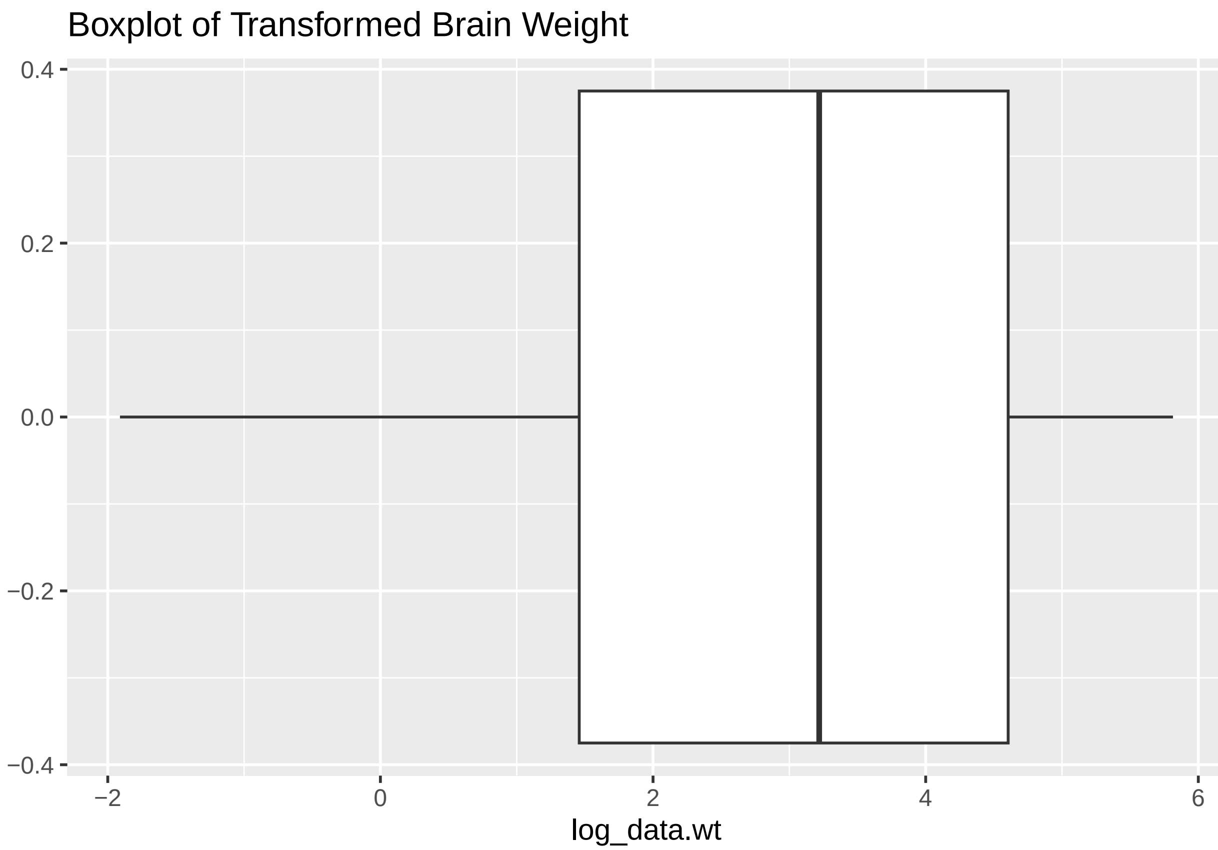
```
upperinterval <- xbar + margin
upperinterval
```

```
## [1] 6.787366
```

Transformed brain weight graphs

```
ggplot(data, aes(x = log_data.wt)) +
  geom_histogram(bins = 10, color = "blue", fill = "red") +
  ggtitle("Histogram of Transformed Brain Weight")
```


```
ggplot(data, aes(x = log_data.wt)) +
  geom_boxplot() +
  ggtitle("Boxplot of Transformed Brain Weight")
```



Calculating the mean for the transformed brain weight

```
xbar <- mean(data$log_data.wt)
xbar
```

```
## [1] 2.768455
```

Calculating the standard deviation for the transformed brain weight

```
sd <- sd(data$log_data.wt)
sd
```

```
## [1] 2.196627
```

```
# Setting n to a value of 21
n = 21
```

Calculating the standard mean for the transformed brain weight

```
standard_error_mean <- sd/sqrt(n)
standard_error_mean
```

```
## [1] 0.4793432
```

```
margin <- qt(0.975,df = n-1)*sd/sqrt(n)

lowerinterval <- xbar - margin
lowerinterval
```

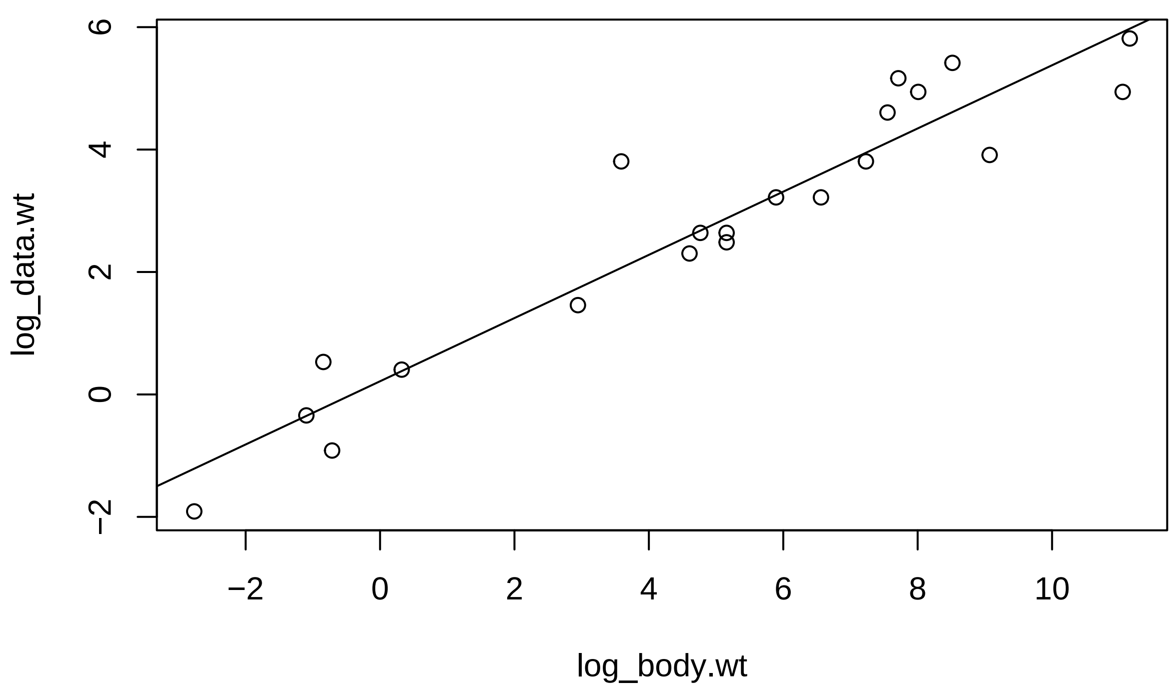
```
## [1] 1.768563
```

```
upperinterval <- xbar + margin
upperinterval
```

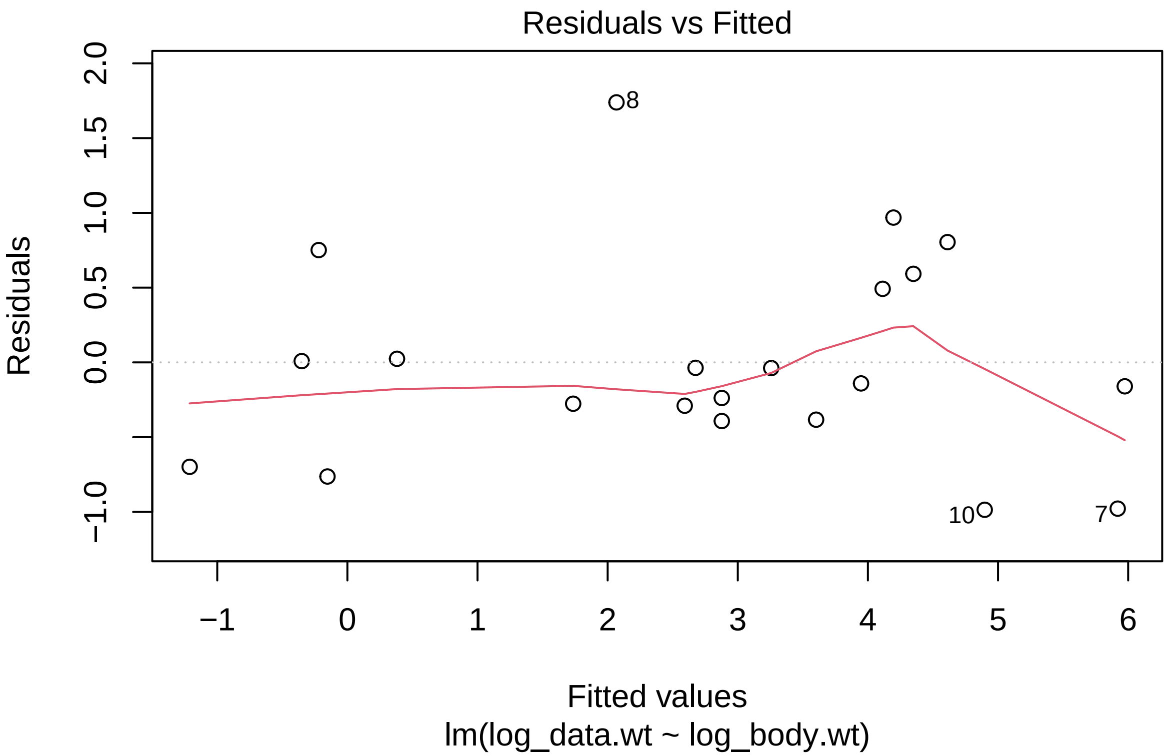
```
## [1] 3.768348
```

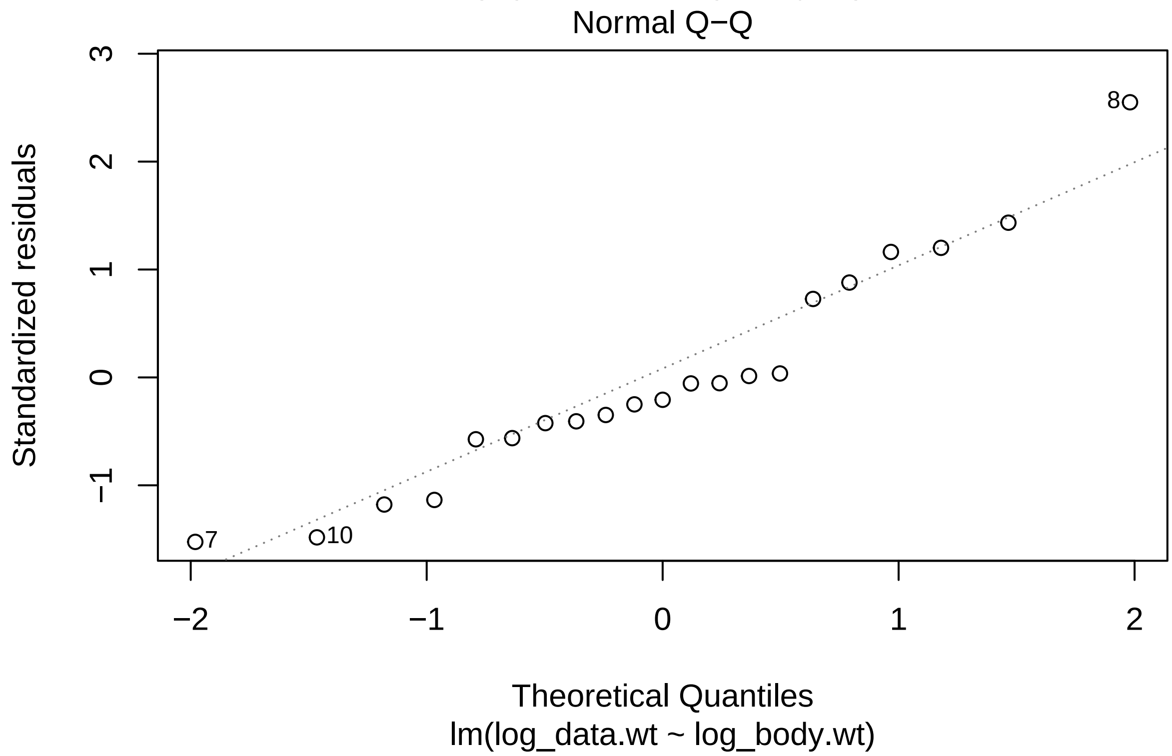
Regression Model of the transformed data

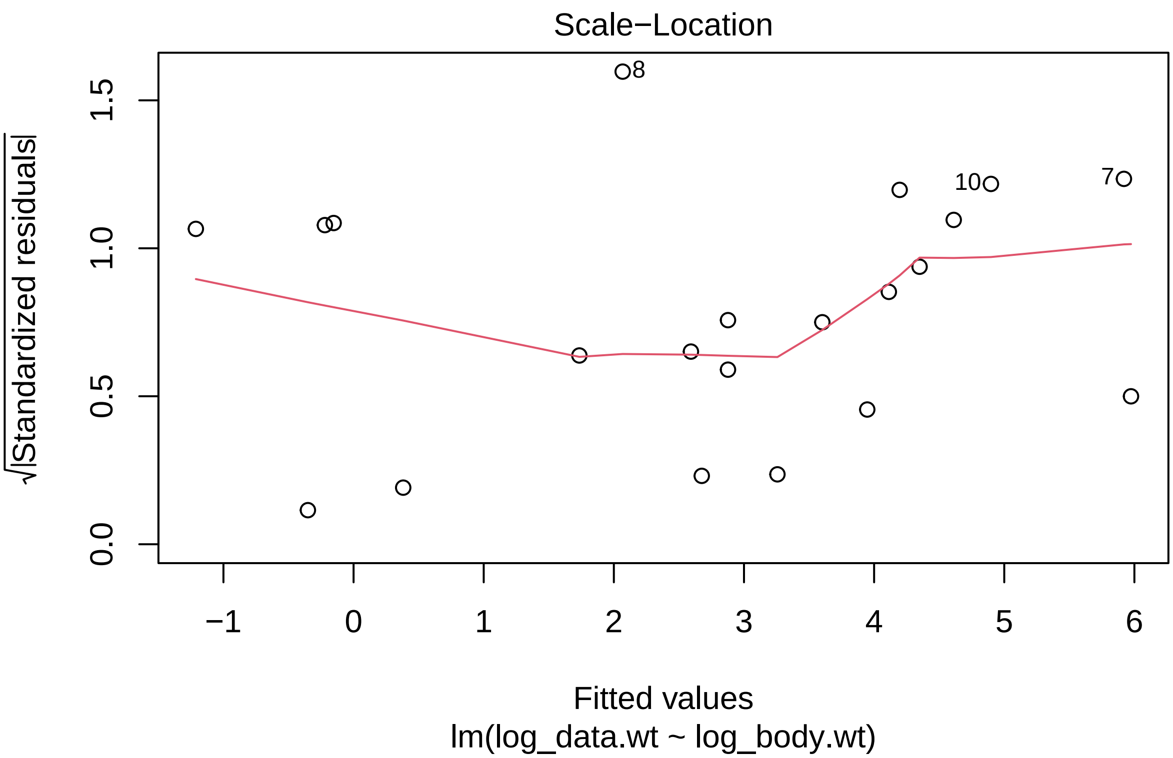
```
plot(log_data.wt~log_body.wt, dat = data)
abline(lm(log_data.wt~log_body.wt, dat = data))
```

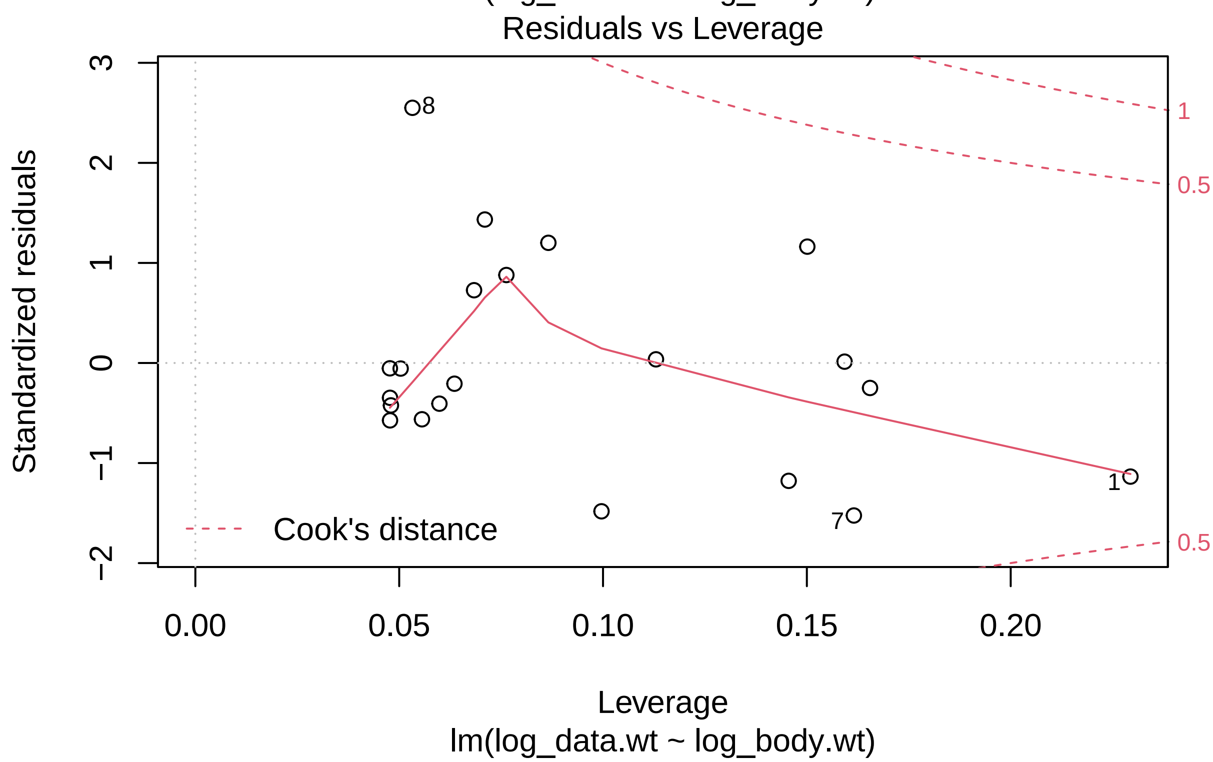


```r
result<-lm(log_data.wt~log_body.wt, dat = data)
plot(result)
```









Creating a lack of fit test

```r
lack <- lm(log_data.wt~log_body.wt, dat = data)
ols_pure_error_anova(lack)
```

```
##    Lack of Fit F Test
## ------------------------
## Response :   log_data.wt
## Predictor:   log_body.wt
##
##                         Analysis of Variance Table
## --------------------------------------------------------------------------
##                 DF      Sum Sq       Mean Sq      F Value        Pr(>F)
## --------------------------------------------------------------------------
## log_body.wt      1      87.17316      87.17316    7337.057    4.703166e-26
## Residual        19      9.330217     0.4910641
##  Lack of fit    18      9.318336     0.5176853    43.57175       0.1187295
##  Pure Error      1    0.01188122    0.01188122
## --------------------------------------------------------------------------
```

Parameter estimates

```r
ans <- lm(log_data.wt~log_body.wt, dat = data)
summary(ans)
```

```
##
## Call:
## lm(formula = log_data.wt ~ log_body.wt, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9856 -0.3831 -0.1405  0.4919  1.7389
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21507    0.24518   0.877    0.391
## log_body.wt  0.51621    0.03874  13.324 4.34e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7008 on 19 degrees of freedom
## Multiple R-squared:  0.9033, Adjusted R-squared:  0.8982
## F-statistic: 177.5 on 1 and 19 DF,  p-value: 4.341e-11
```

```r
confint(ans)
```

```
##                  2.5 %    97.5 %
## (Intercept) -0.2980876 0.7282302
## log_body.wt  0.4351187 0.5973031
```

Estimating the brain weight if the body size is set to 20 kilograms

```
value <- data.frame(20, log(20))
names(value) <- c("Body.Weight", "log_body.wt")
pred <- predict(ans, value)
pred
```

```
##        1
## 1.761501
```

```
exp(pred)
```

```
##        1
## 5.821168
```

Residual Model Graohs

```
ans <-lm(log_data.wt~log_body.wt, dat = data)
res <- resid(ans)
plot(fitted(ans), res)
abline(0,0)
```