Assignment8

White team

2023-10-06

packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(lmtest)
```

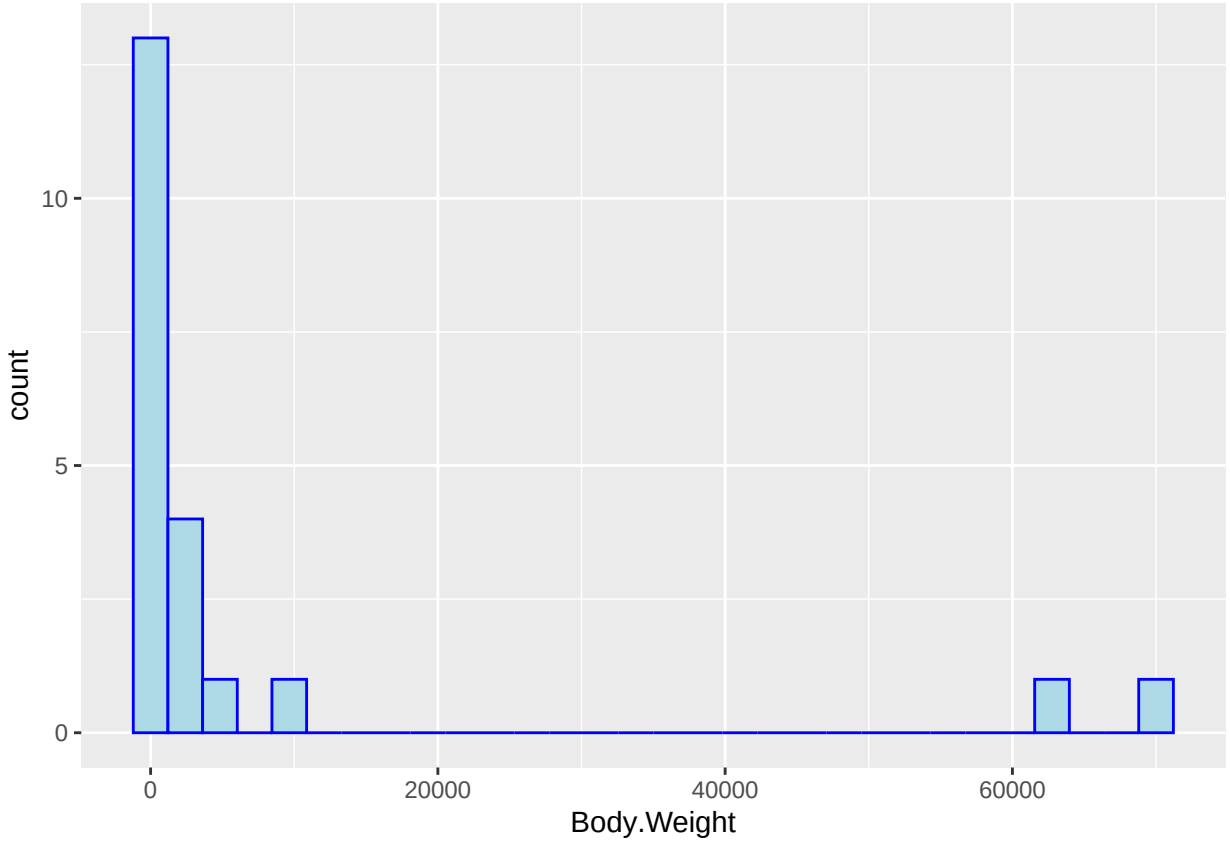
```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tibble)
```

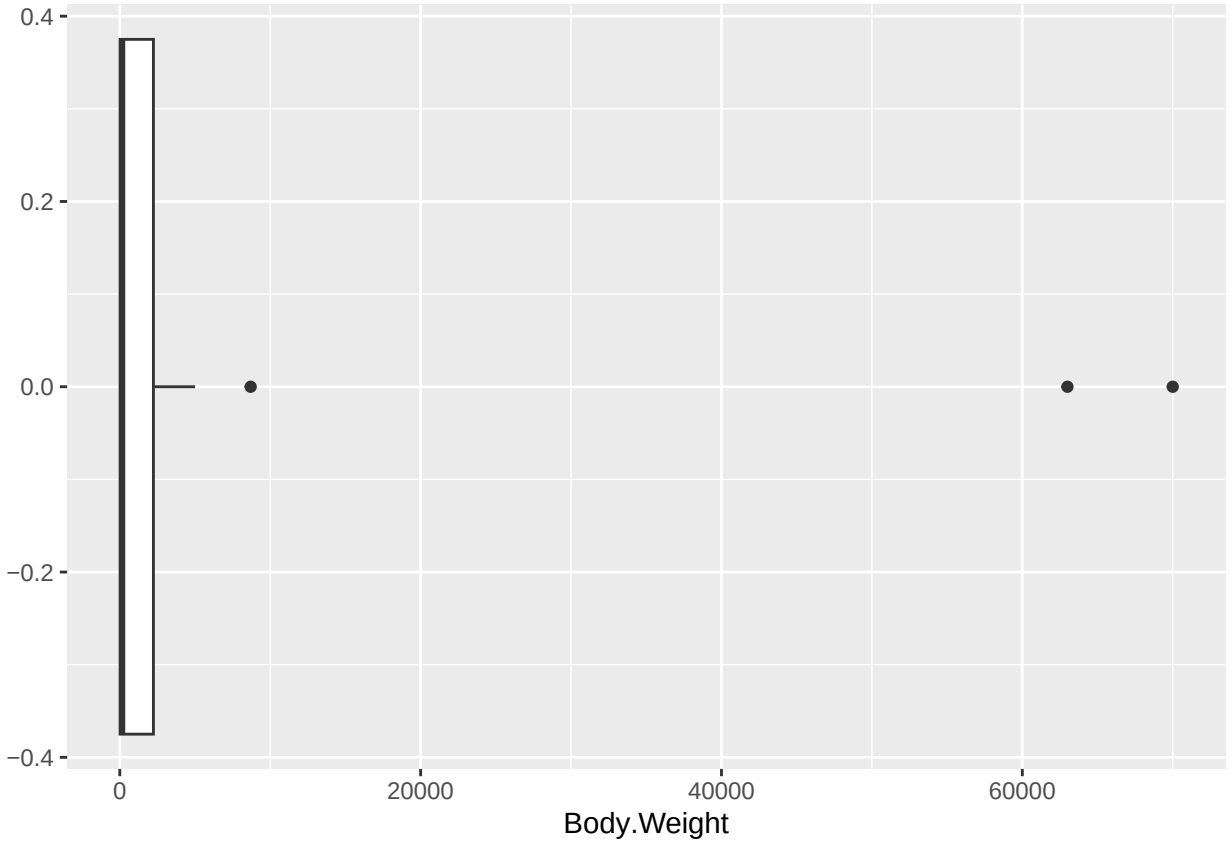
load data

```r
setwd("C:/Users/Chang/Downloads")  # Replace with the path to your directory


dat <- read.csv("archosaur.csv")

ggplot(dat, aes(x=Body.Weight)) +
  geom_histogram(fill="lightblue", color="blue" )
```

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

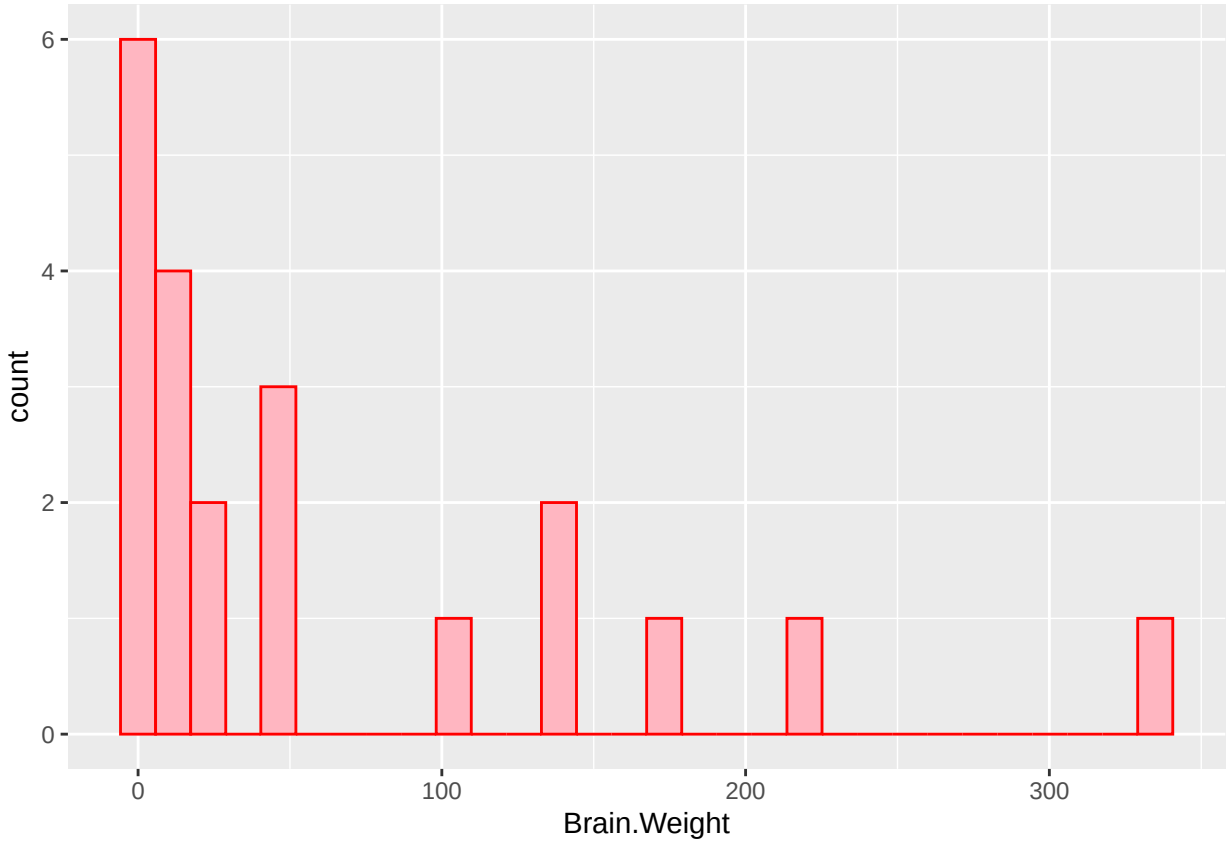



```r
ggplot(dat, aes(x=Body.Weight)) +
  geom_boxplot()
```

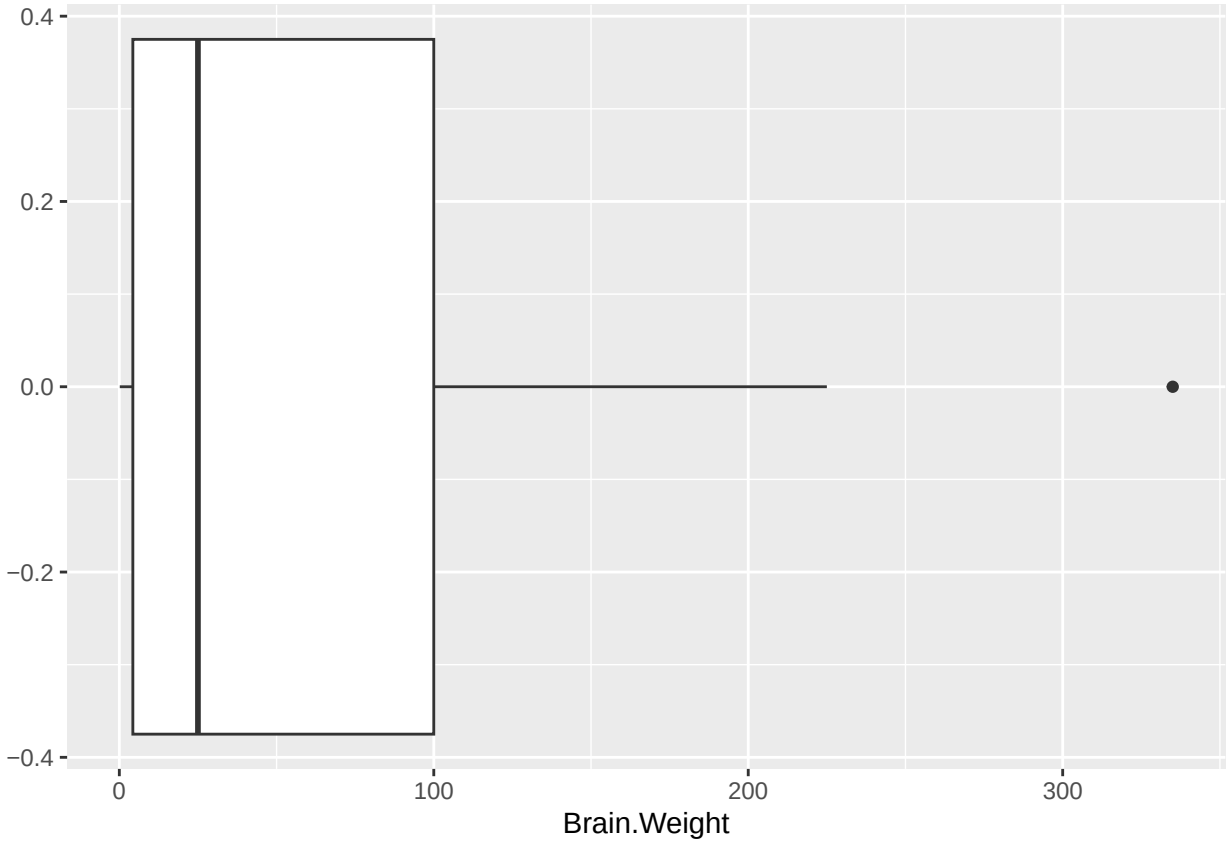



```
ggplot(dat, aes(x=Brain.Weight)) +
  geom_histogram(fill="lightpink", color="red")
```

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




```
ggplot(dat, aes(x=Brain.Weight)) +
  geom_boxplot()
```




```
summary(dat$Body.Weight)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.06    19.00   173.60  7472.37  2236.00 70000.00
```

```
summary(dat$Brain.Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.148   4.300  25.000  64.941 100.000 335.000
```

Fit the model(simple linear regression)

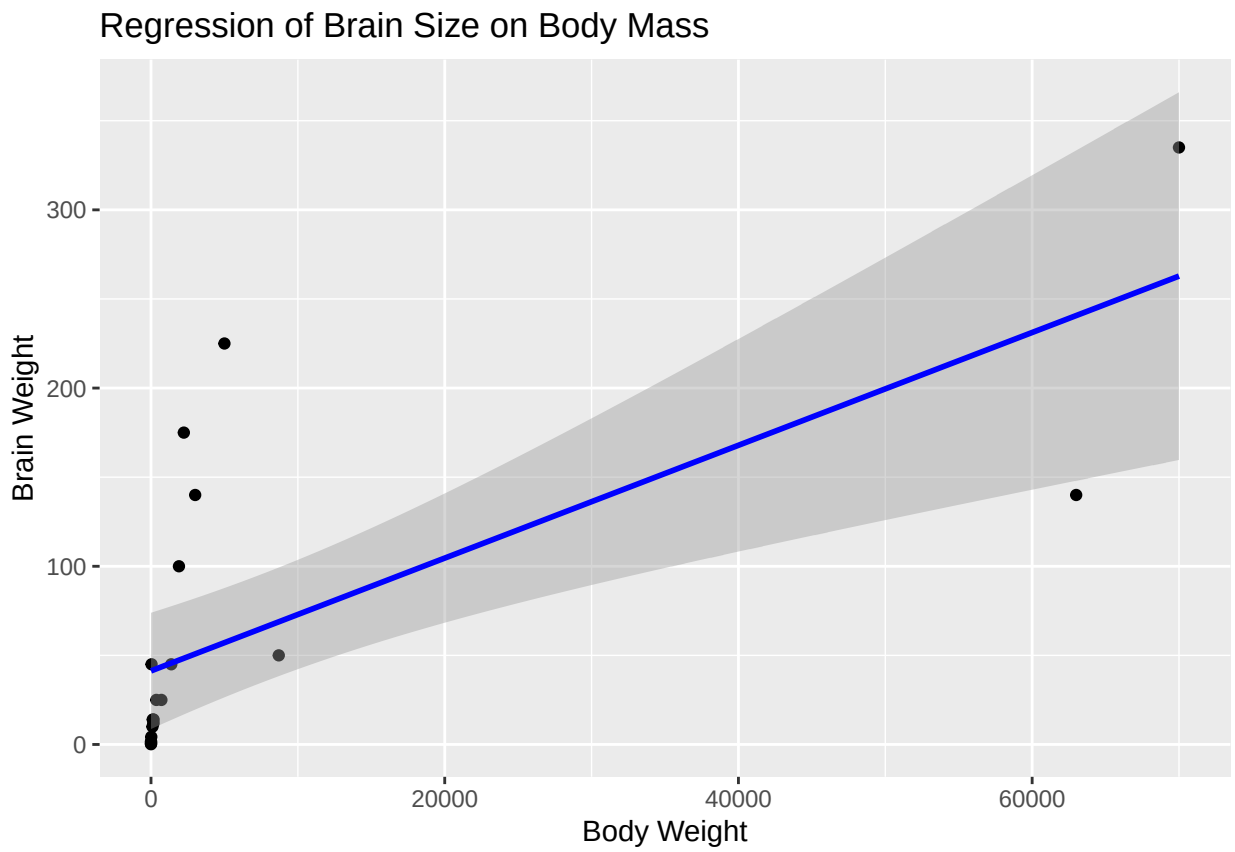
```
dat_lm <- lm(Brain.Weight~ Body.Weight,dat)
summary(dat_lm)
```

```
##
## Call:
## lm(formula = Brain.Weight ~ Body.Weight, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -100.671  -39.594  -27.665    3.593  167.884
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.129e+01  1.559e+01   2.649 0.015819 *
## Body.Weight 3.165e-03  7.533e-04   4.201 0.000484 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.6 on 19 degrees of freedom
## Multiple R-squared:  0.4816, Adjusted R-squared:  0.4543
## F-statistic: 17.65 on 1 and 19 DF,  p-value: 0.0004838
```

```
ggplot(dat, aes(Body.Weight,Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="Body Weight",
       y="Brain Weight",
       color="Legend")
```

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

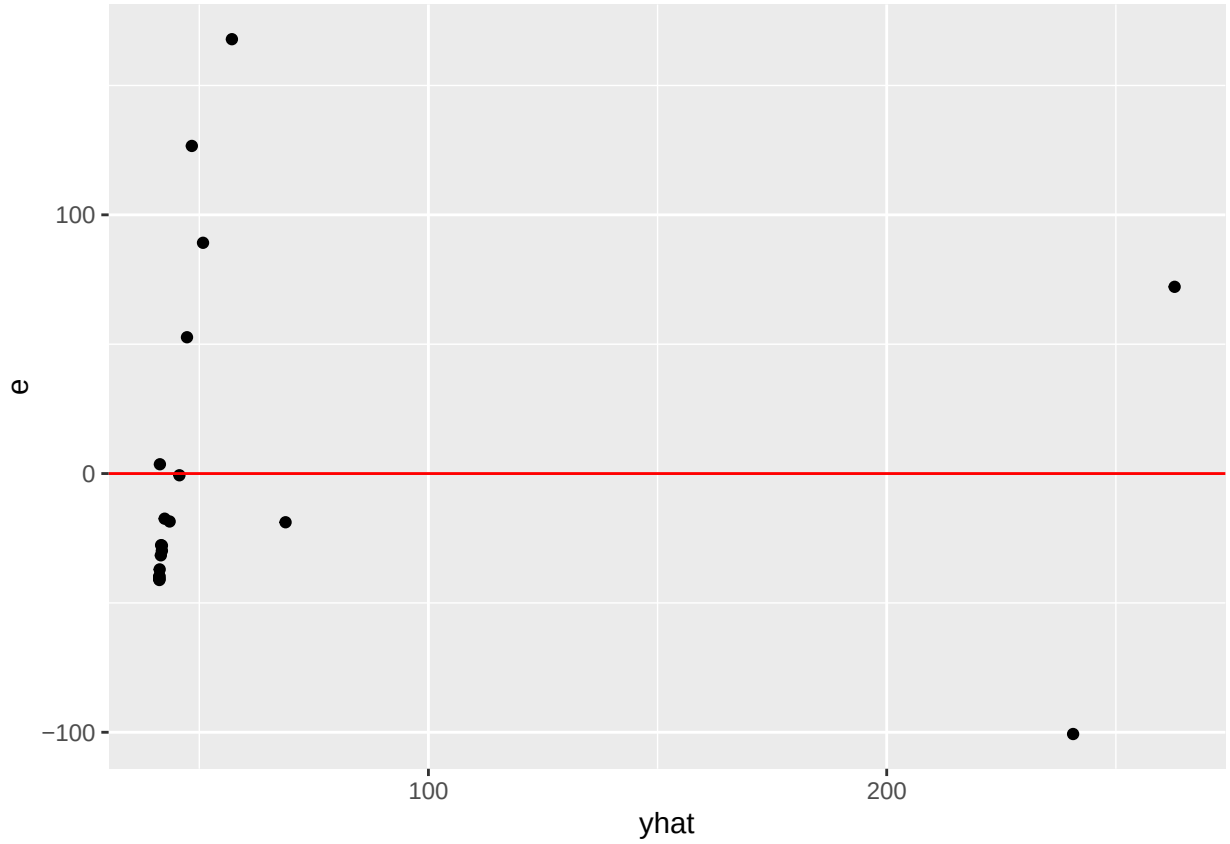


When we look at the scatterplot, without diagnostics, we get a clear sense of the right skewness of the distribution.

Diagnostics

1.residuals

```
#plot fitted values by residuals.
dat2 <- tibble(x =  dat$Body.Weight, yhat = dat_lm$fitted.values,
               e = dat_lm$residuals)
ggplot(dat2, aes(x = yhat, y = e))+
  geom_point()+
  geom_hline(yintercept = 0, col = "red")
```

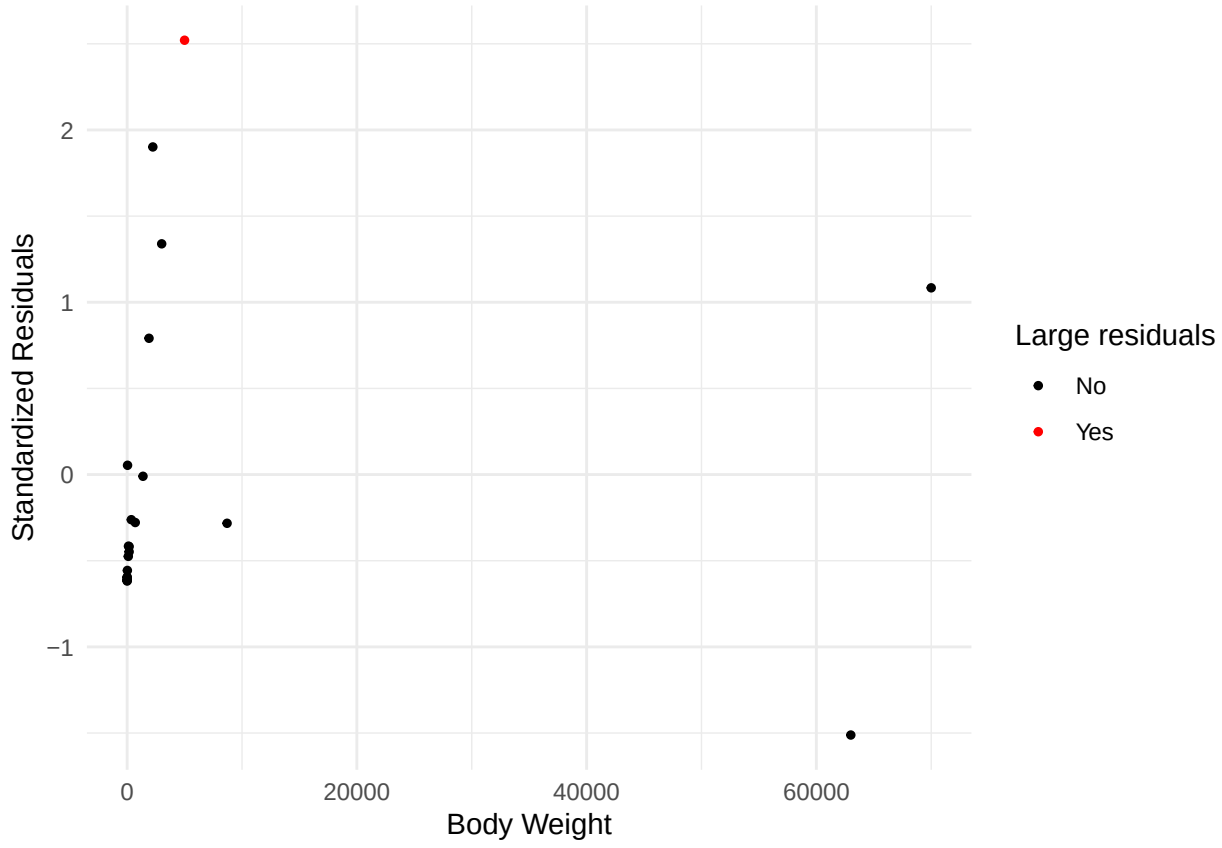


Examination of the residual plot reveals a non-random distribution of residuals. This suggests the presence of potential outliers or influential points that might be affecting the model fit. Notably, data points with smaller body weights exhibit distinctly large residuals, indicating that simply identifying and addressing a few influential points might not be sufficient to rectify the model.

2.semistudentized plot # also can check outlinear point

```
# Define a threshold
threshold <- 2
dat$e.star = dat_lm$residuals /summary(dat_lm)$sigma
# Create a plot
ggplot(dat, aes(x = Body.Weight, y = e.star)) +
```

```
  geom_point(aes(color = abs(e.star) > threshold), size = 1) +
  scale_color_manual(values = c("black", "red"),
                     name = "Large residuals",
                     breaks = c(FALSE, TRUE),
                     labels = c("No", "Yes")) +
  theme_minimal() +
  labs(y = "Standardized Residuals", x = "Body Weight")
```

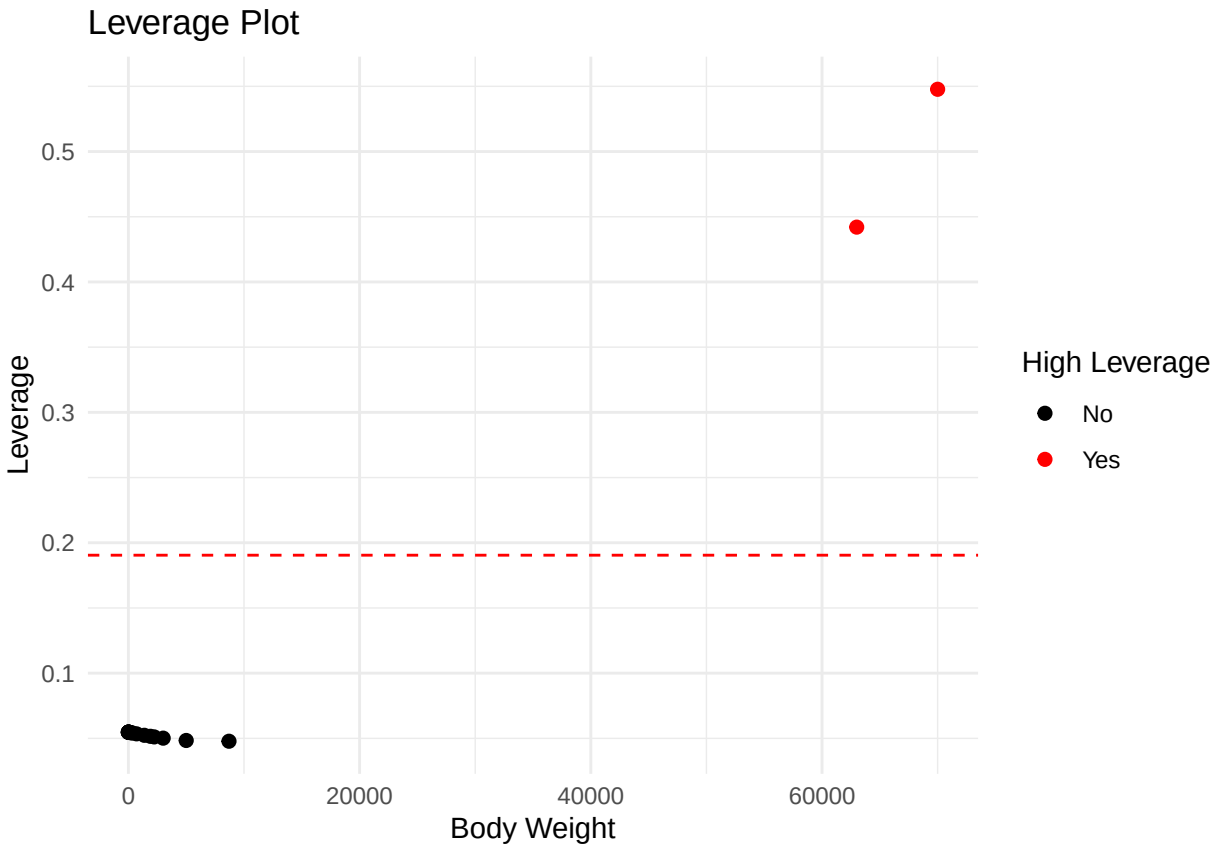



```
outliers <- dat %>% filter(abs(e.star) > threshold)

print(outliers)
```

```
##        Type     Details Body.Weight Brain.Weight   e.star
## 1 Dinosarus Triceratops        5000          225 2.520762
```

```
# Compute the leverage values
leverage_values <- hatvalues(dat_lm)

p <- length(coef(dat_lm))
n <- nrow(dat)
threshold_leverage <- 2*p/n


ggplot(dat, aes(x = Body.Weight, y = leverage_values)) +
```

```
  geom_point(aes(color = as.factor(leverage_values > threshold_leverage)), size = 2) +
  geom_hline(yintercept = threshold_leverage, linetype = "dashed", color = "red") +
  scale_color_manual(values = c("black", "red"),
                     name = "High Leverage",
                     breaks = c("FALSE", "TRUE"),
                     labels = c("No", "Yes")) +
  theme_minimal() +
  labs(y = "Leverage", x = "Body Weight", title = "Leverage Plot")
```




```
high_leverage_points <- dat[leverage_values > threshold_leverage, ]
print(high_leverage_points)
```

```
##        Type       Details Body.Weight Brain.Weight    e.star
## 7 Dinosaurs Tyrannosaurus       63000          140 -1.511564
## 9 Dinosaurs Brachiosaurus       70000          335  1.083724
```

We conducted a thorough examination of outliers and high leverage values within our dataset. Interestingly, there was no intersection between these two sets of points. Notably, some high-leverage data points were not classified as outliers. This indicates that while these points fit well with the regression line, they might disproportionately influence the overall model fit. A closer observation revealed that these high-leverage points consistently corresponded to larger BODY WEIGHT values. Given their potential undue influence on the model, we opted to exclude these data points and subsequently re-estimated our regression model.

Improvement of the model

1. exclude points with high_leverage.

```
dat_without_high_leverage <- dat[-which(leverage_values > threshold_leverage), ]


ggplot(dat_without_high_leverage, aes(Body.Weight,Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="Body Weight",
       y="Brain Weight",
       color="Legend")
```

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


```
dat_lm_without_high_leverage <- lm(Brain.Weight ~ Body.Weight, dat_without_high_leverage)
summary(dat_lm)
```

```
##
## Call:
## lm(formula = Brain.Weight ~ Body.Weight, data = dat)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -100.671  -39.594  -27.665    3.593  167.884
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.129e+01  1.559e+01   2.649 0.015819 *
## Body.Weight 3.165e-03  7.533e-04   4.201 0.000484 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.6 on 19 degrees of freedom
## Multiple R-squared:  0.4816, Adjusted R-squared:  0.4543
## F-statistic: 17.65 on 1 and 19 DF,  p-value: 0.0004838
```

```
summary(dat_lm_without_high_leverage)
```

```
##
## Call:
## lm(formula = Brain.Weight ~ Body.Weight, data = dat_without_high_leverage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.015  -23.834  -14.364    7.568  114.756
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.418823  14.581684   1.743  0.09935 .
## Body.Weight  0.016965   0.005779   2.936  0.00923 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.08 on 17 degrees of freedom
## Multiple R-squared:  0.3364, Adjusted R-squared:  0.2974
## F-statistic: 8.619 on 1 and 17 DF,  p-value: 0.009233
```

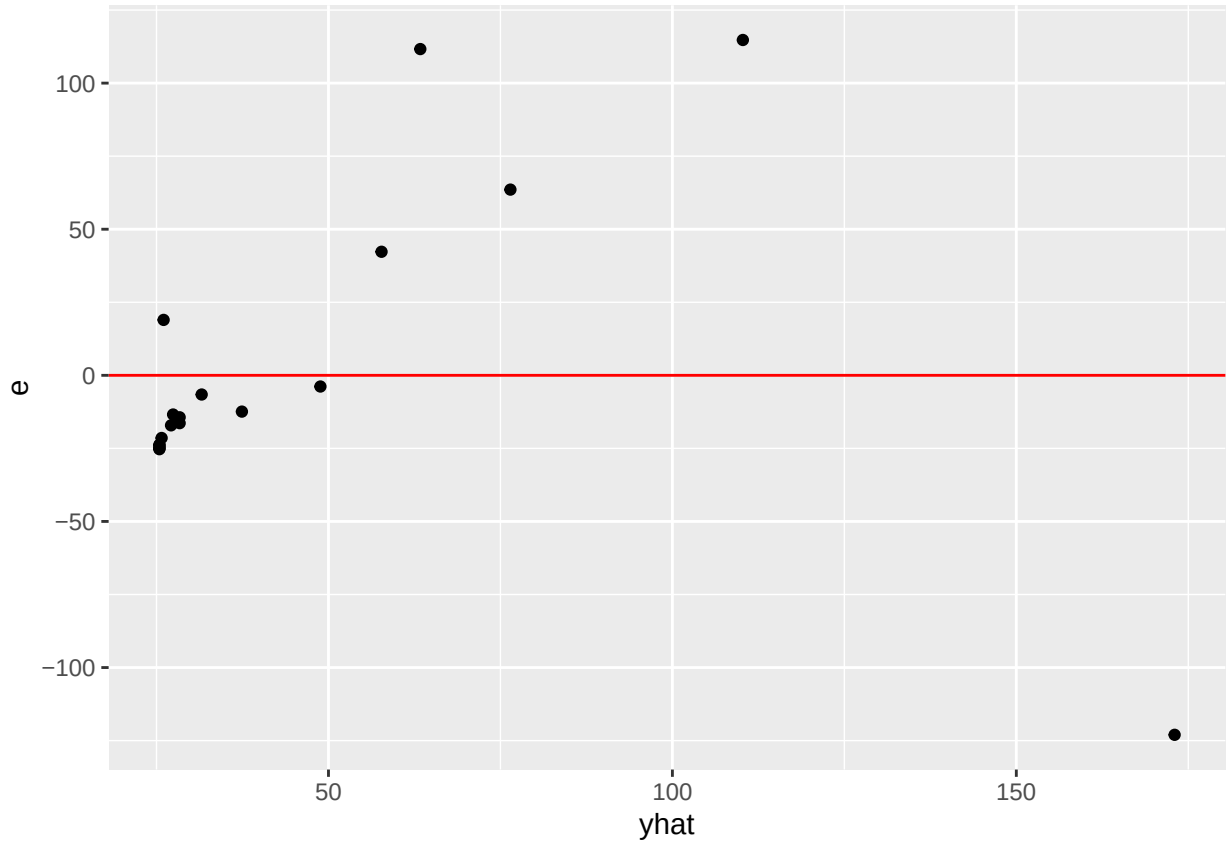
```
# Compare the two models (with and without high-leverage points)
AIC(dat_lm, dat_lm_without_high_leverage)
```

```
## Warning in AIC.default(dat_lm, dat_lm_without_high_leverage): models are not
## all fitted to the same number of observations
```

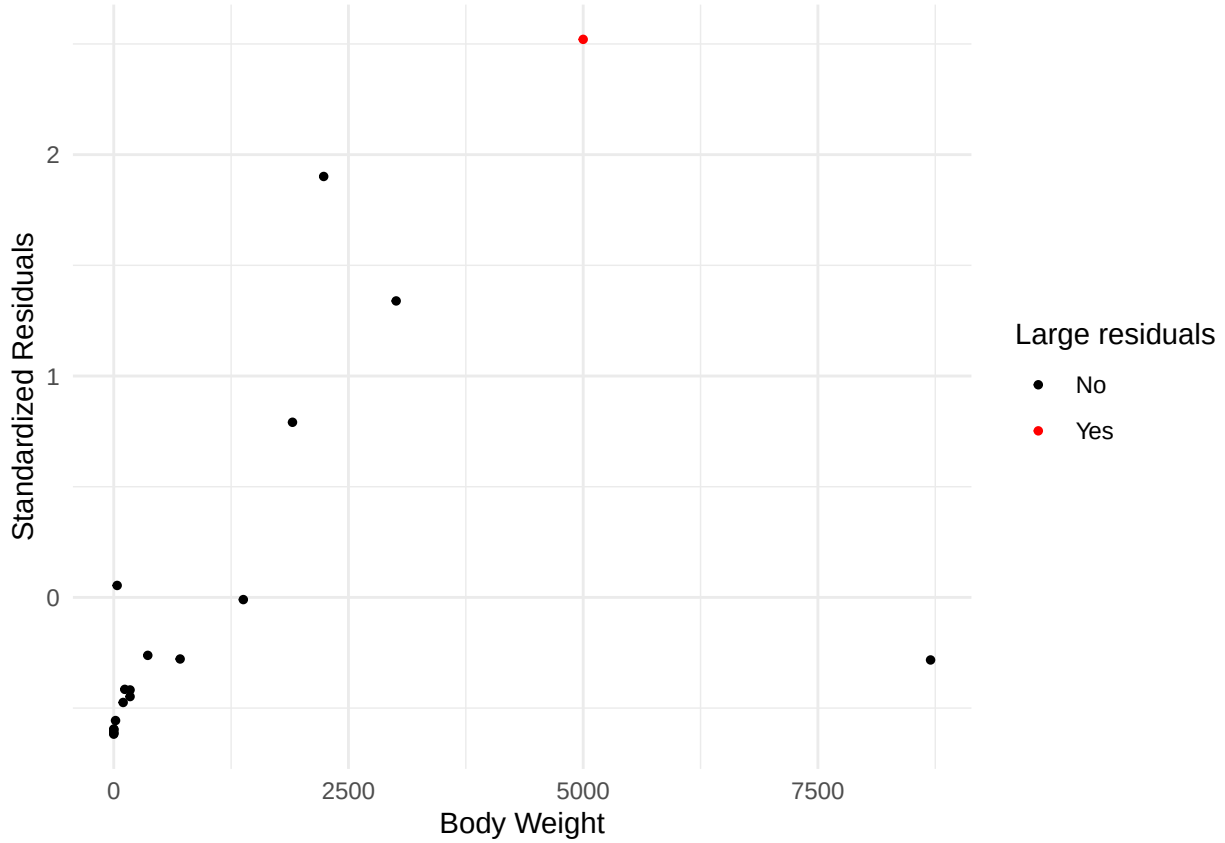
```
##                              df      AIC
## dat_lm                        3 239.8394
## dat_lm_without_high_leverage  3 210.1436
```

After excluding the high leverage points and refitting the linear regression model, there was a noticeable improvement in the model's fit as visualized in the scatterplot of the updated dataset. Although the r square value decreased to 0.33, indicating a reduced variance explanation, this might reflect a more genuine representation of the underlying relationship without the undue influence of the high leverage points. Importantly, the Akaike Information Criterion (AIC) dropped to 210.14, suggesting a better balance between model fit and complexity after the exclusion of these points.

```r
#plot fitted values by residuals.
dat_whl <- tibble(x =dat_lm_without_high_leverage$Body.Weight, yhat = dat_lm_without_high_leverage$fitt
                  e = dat_lm_without_high_leverage$residuals)
ggplot(dat_whl, aes(x = yhat, y = e))+
  geom_point()+
  geom_hline(yintercept = 0, col = "red")
```




```r
# Create a plot
ggplot(dat_without_high_leverage, aes(x = Body.Weight, y = e.star)) +
  geom_point(aes(color = abs(e.star) > threshold), size = 1) +
  scale_color_manual(values = c("black", "red"),
                     name = "Large residuals",
                     breaks = c(FALSE, TRUE),
                     labels = c("No", "Yes")) +
  theme_minimal() +
  labs(y = "Standardized Residuals", x = "Body Weight")
```



Upon examining the diagnostic plots, a distinct pattern in the residuals is evident, suggesting that the model's assumptions may not be entirely met even after addressing the high-leverage points. This persistent issue indicates the necessity of further model refinement. Several strategies should be explored to enhance the model's predictive capability and adherence to assumptions.

Furthermore, a deeper dive into the dataset highlighted another intriguing aspect: the potential influence of different species on our model. We discovered that the data points with high leverage were predominantly from species like Brachiosaurus and Tyrannosaurus, which are markedly distinct from other species in the dataset, such as Allosaurus, Scaaphognathus purdoni, Rhamphorhynchus, Pteranodon, and others.

Considering the vast Logarithmic Transformation of the Predictor Onlyevolutionary, ecological, and physiological differences between these species, it is plausible to hypothesize that pooling them together in a single model may not yield an accurate representation of the relationship between body weight and brain weight. For example, Brachiosaurus, a massive herbivore, would have had different evolutionary pressures compared to a predator like Tyrannosaurus, leading to variations in their brain-to-body weight ratios.

Given this insight, it may be more appropriate to segment our data based on specific dinosaur species or broader clades, and develop separate models for each group. This could offer a more nuanced understanding of the relationship under study, specific to each species or clade, rather than attempting a one-size-fits-all approach.(Considering that the sample sizes for the various categories in this dataset are so small, we were unable to make more specific attempts.)

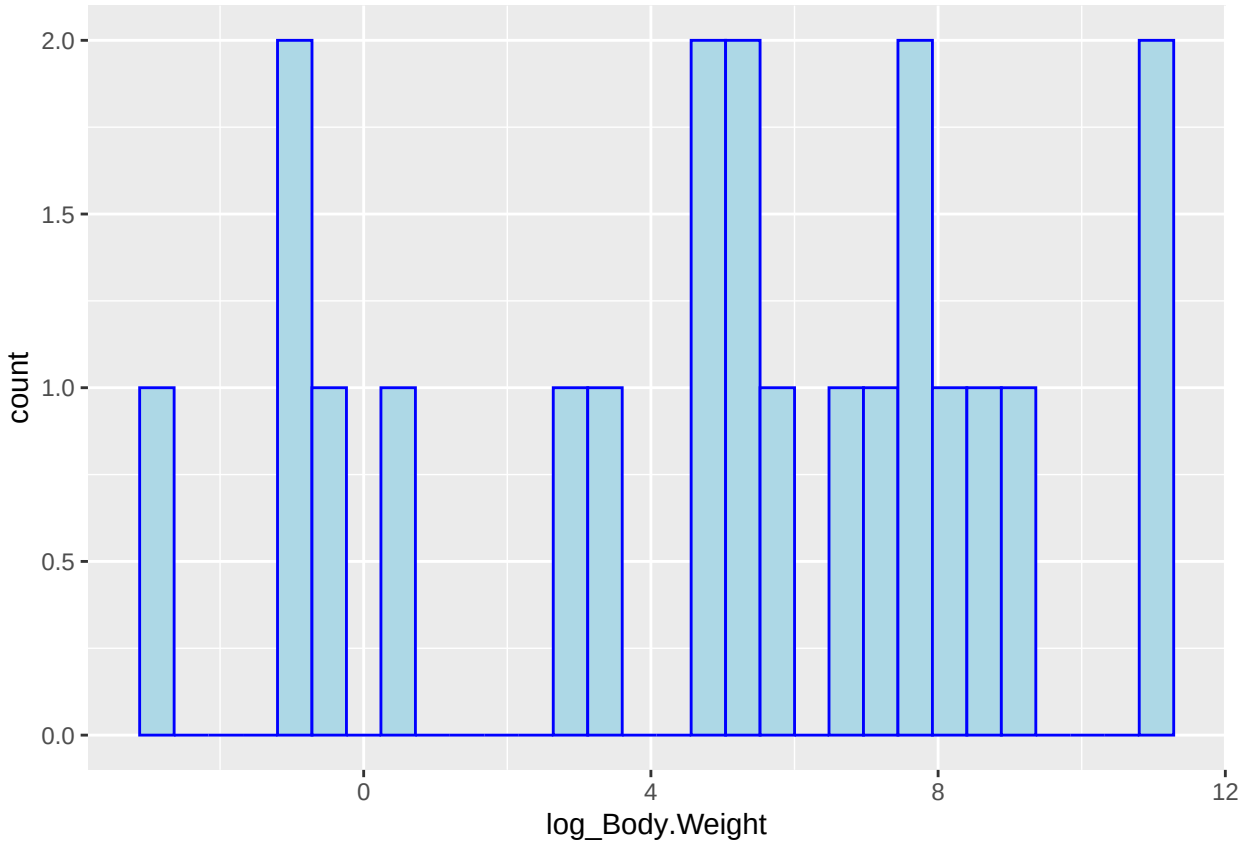
Our recommendation aligns with principles from the study of allometric scaling in vertebrates, which suggests that different species, due to their unique evolutionary trajectories and environmental pressures, might exhibit different relationships between physical and physiological characteristics.

2.Transformation

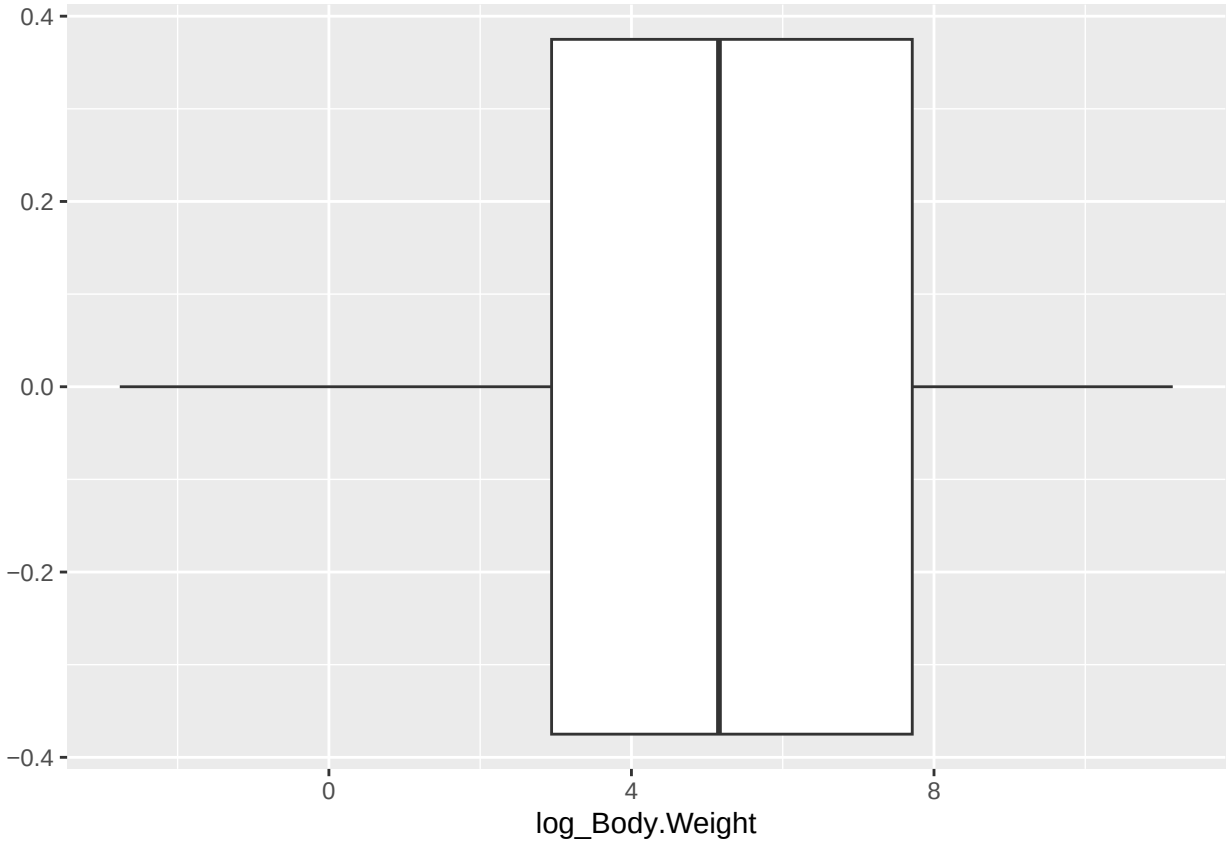
```r
dat$log_Body.Weight <- log(dat$Body.Weight)

#basic information of log_body.weight
ggplot(dat, aes(x=log_Body.Weight)) +
  geom_histogram(fill="lightblue", color="blue" )
```

1.Logarithmic Transformation of the Predictor Only

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




```r
ggplot(dat, aes(x=log_Body.Weight)) +
  geom_boxplot()
```




```
summary(dat$log_Body.Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.765   2.944   5.157   4.946   7.712  11.156
```

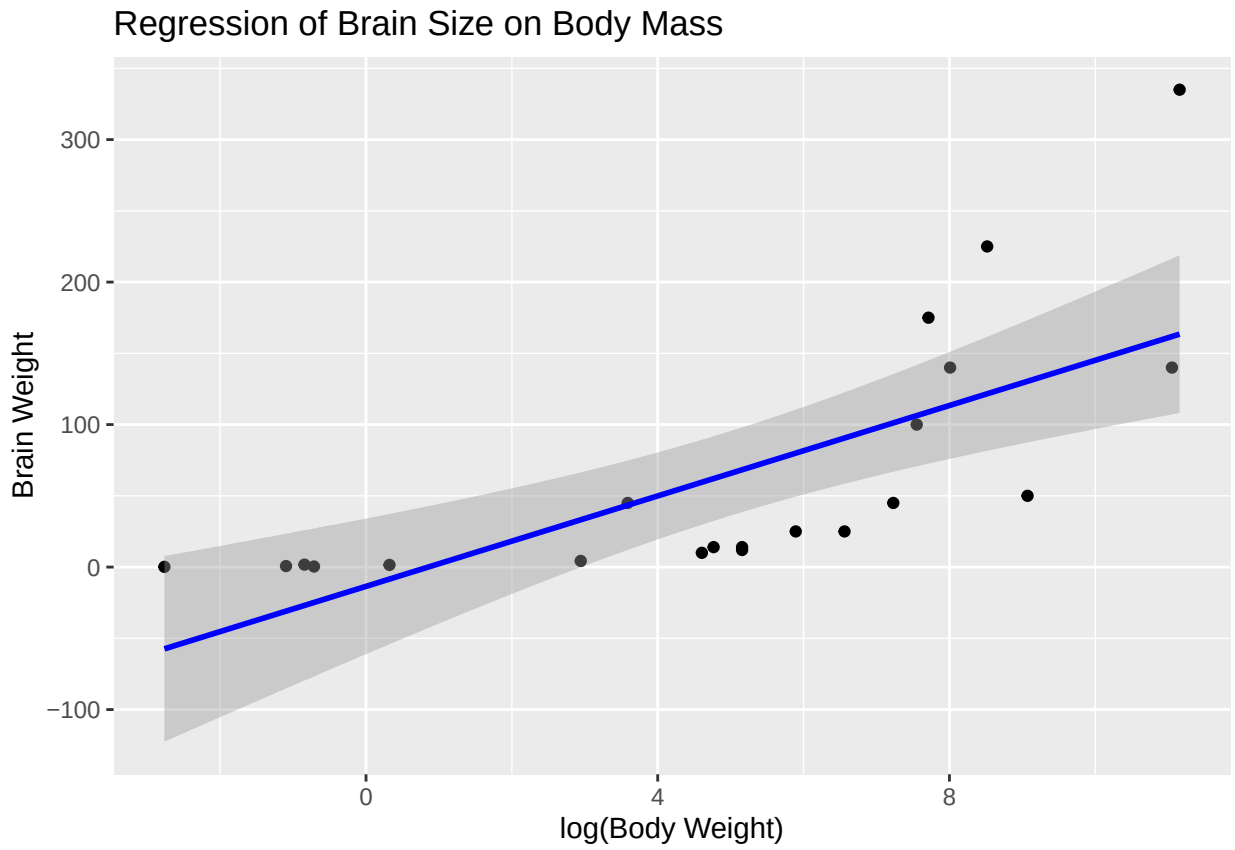
```
dat_lm_log_x <- lm(Brain.Weight ~ log_Body.Weight, data = dat)
summary(dat_lm_log_x)
```

```
##
## Call:
## lm(formula = Brain.Weight ~ log_Body.Weight, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -80.380 -54.278  -6.272  28.625 171.538
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -13.535     22.735  -0.595 0.558650
## log_Body.Weight   15.865      3.593   4.416 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.98 on 19 degrees of freedom
```

```
## Multiple R-squared:  0.5065, Adjusted R-squared:  0.4805
## F-statistic:  19.5 on 1 and 19 DF,  p-value: 0.0002968
```

```
ggplot(dat, aes(log_Body.Weight,Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="log(Body Weight)",
       y="Brain Weight",
       color="Legend")
```

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

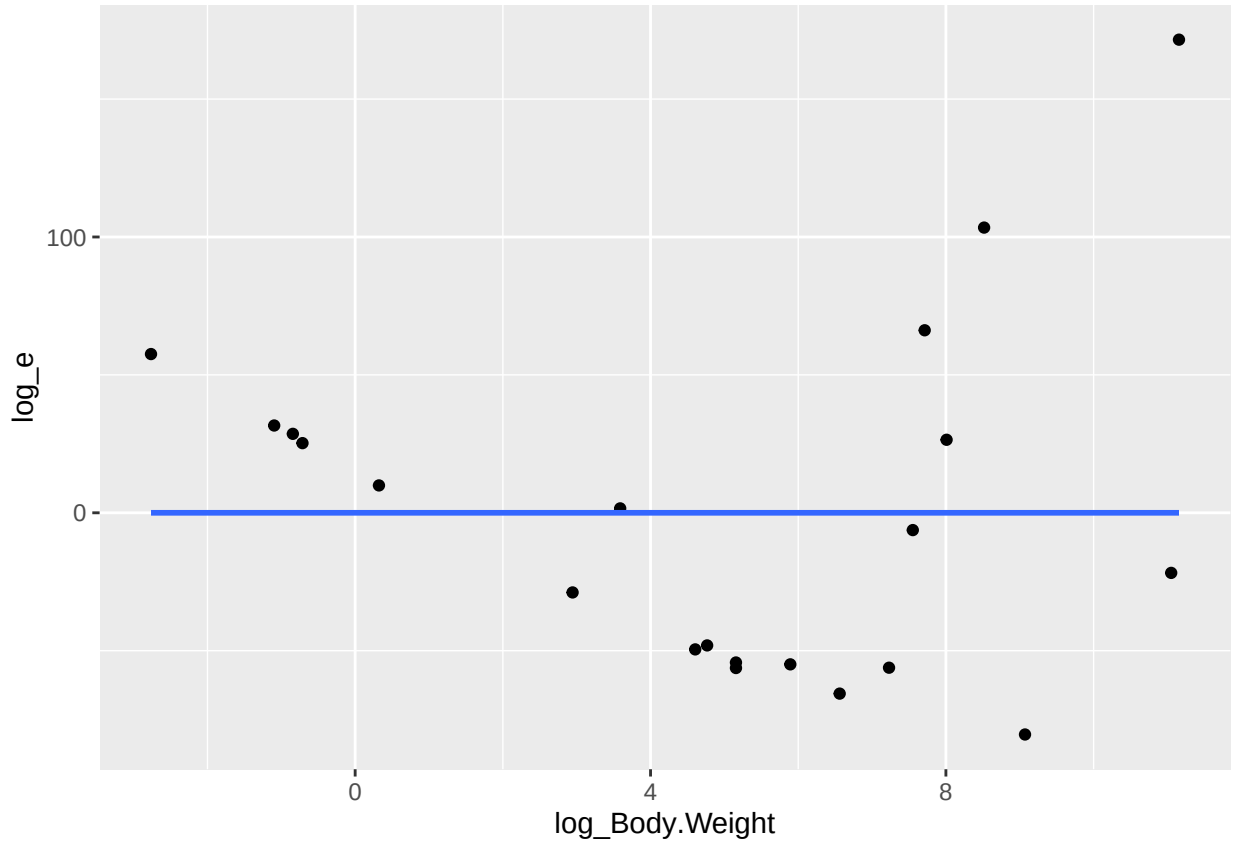


Upon examining the updated scatterplot, it's evident that the distribution of the individual data points aligns more closely with the assumptions of linear regression. Moving forward, we will rigorously assess this model by examining the four fundamental assumptions of linear regression

```
dat$log_e <- dat_lm_log_x %>%resid()
ggplot(dat, aes(log_Body.Weight,log_e))+
  geom_point()+
  geom_smooth(method = "lm", se = F)
```

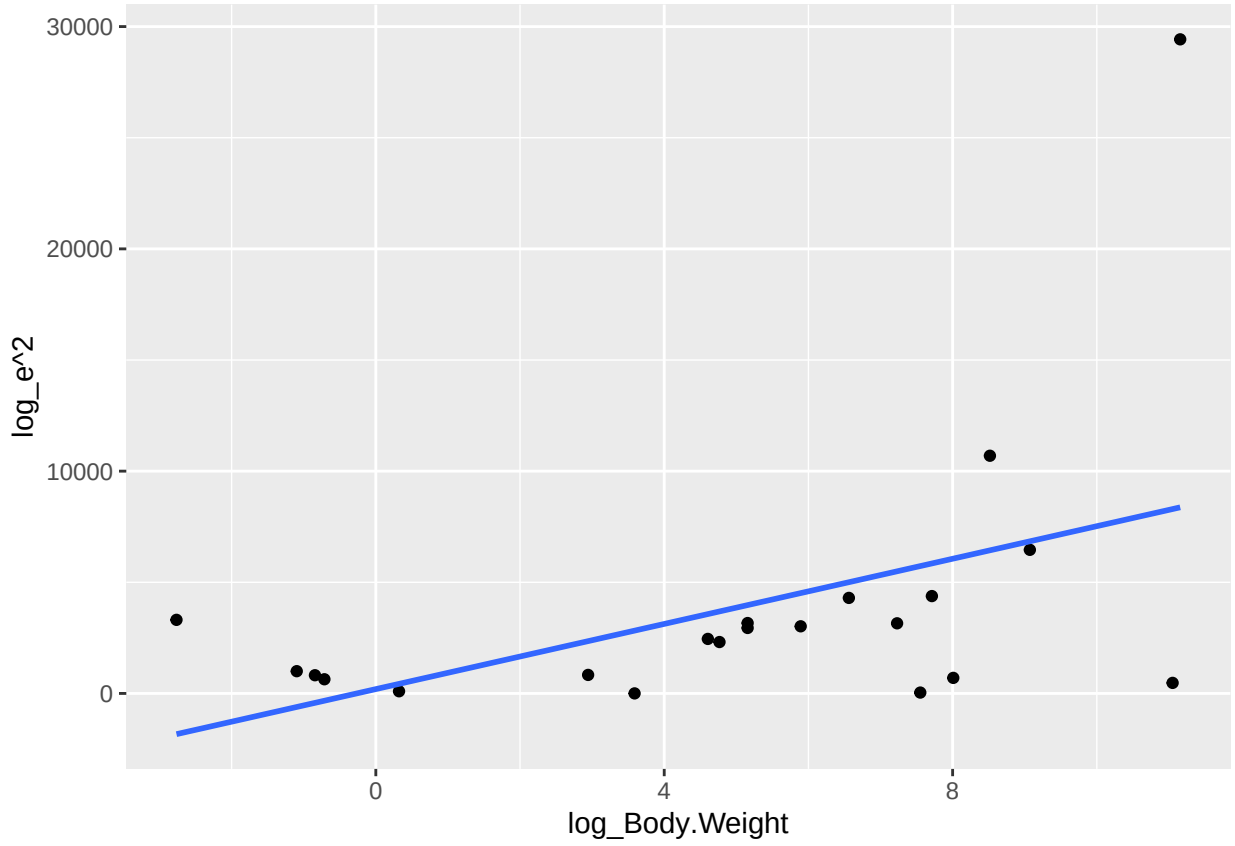
Diagnostics

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




```
ggplot(dat, aes(log_Body.Weight,log_e^2))+
  geom_point()+
  geom_smooth(method = "lm", se = F)
```

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




```
#breusch-Pagan test

bptest(Brain.Weight ~ log_Body.Weight, data = dat)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Brain.Weight ~ log_Body.Weight
## BP = 4.5195, df = 1, p-value = 0.03351
```

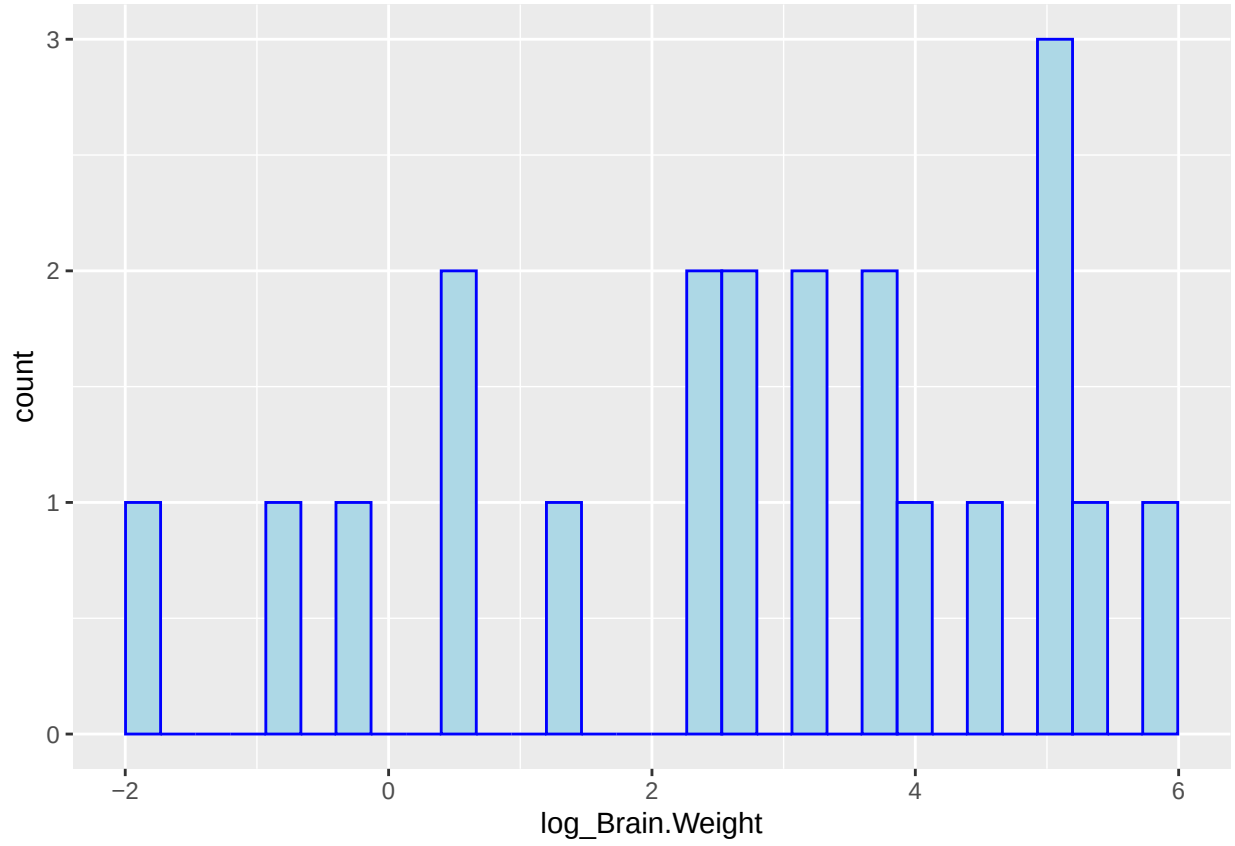
The residual plot displays a noticeable V-shape, suggesting that the relationship between the log of body weight and brain weight might not be strictly linear.The second residual plot, on the other hand, is not as obvious.Since the p value is smaller than 0.05, then there is sufficient evidence to conclude the variance is not constant through different values of x.

We observed that the model does not satisfy the homoscedasticity assumption, indicating that it might not be an ideal fit for the data. Instead of proceeding with the other three diagnostic tests, we chose to first address this issue. By transforming the dependent variable 'y', we aim to enhance the model's fit and subsequently reassess its assumptions.

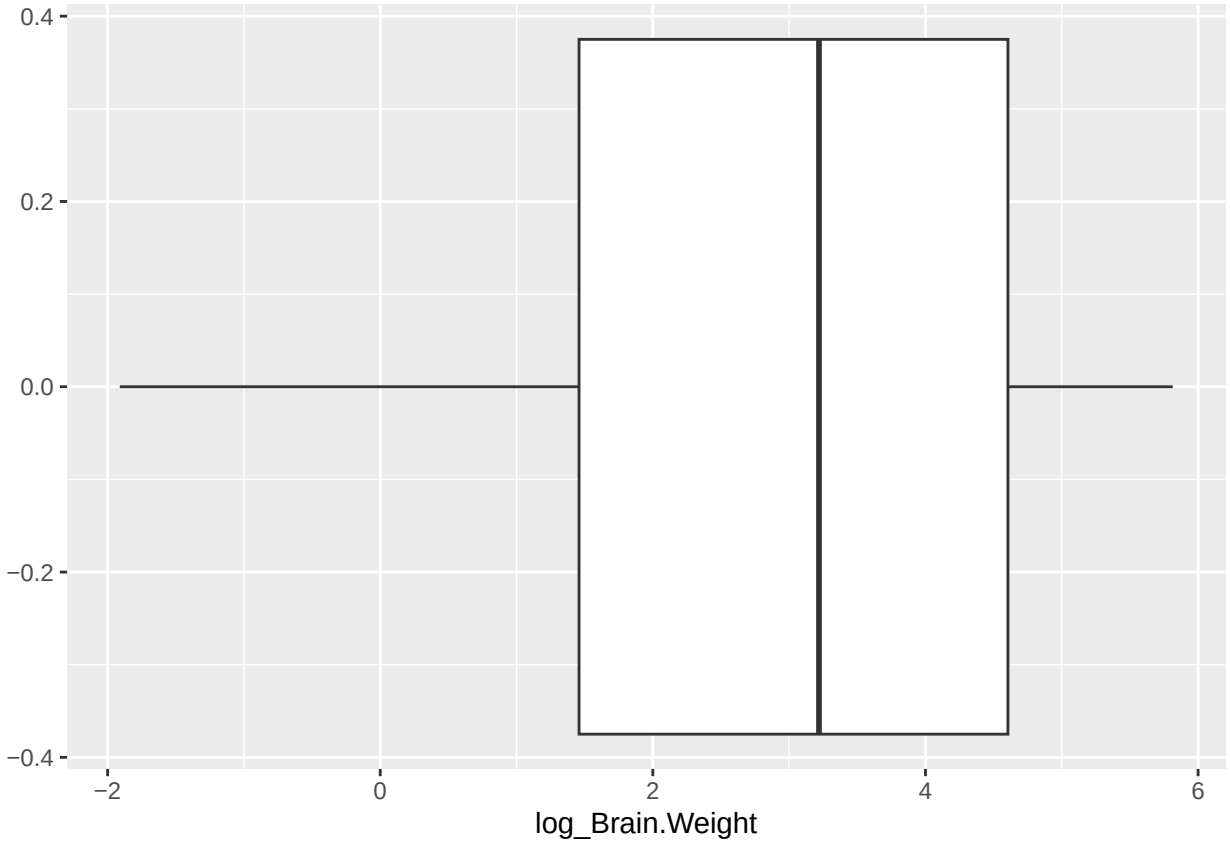
2.Logarithmic Transformation of the dependent variable. Keeping in mind the interpretability of the model, we opted to apply a logarithmic transformation solely to the response variable, brain weight, in our model refinement.

```r
dat$log_Brain.Weight <- log(dat$Brain.Weight)

#basic information of log_brain.weight
ggplot(dat, aes(x=log_Brain.Weight)) +
  geom_histogram(fill="lightblue", color="blue" )
```

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




```r
ggplot(dat, aes(x=log_Brain.Weight)) +
  geom_boxplot()
```




```
summary(dat$log_Brain.Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.911   1.459   3.219   2.768   4.605   5.814
```

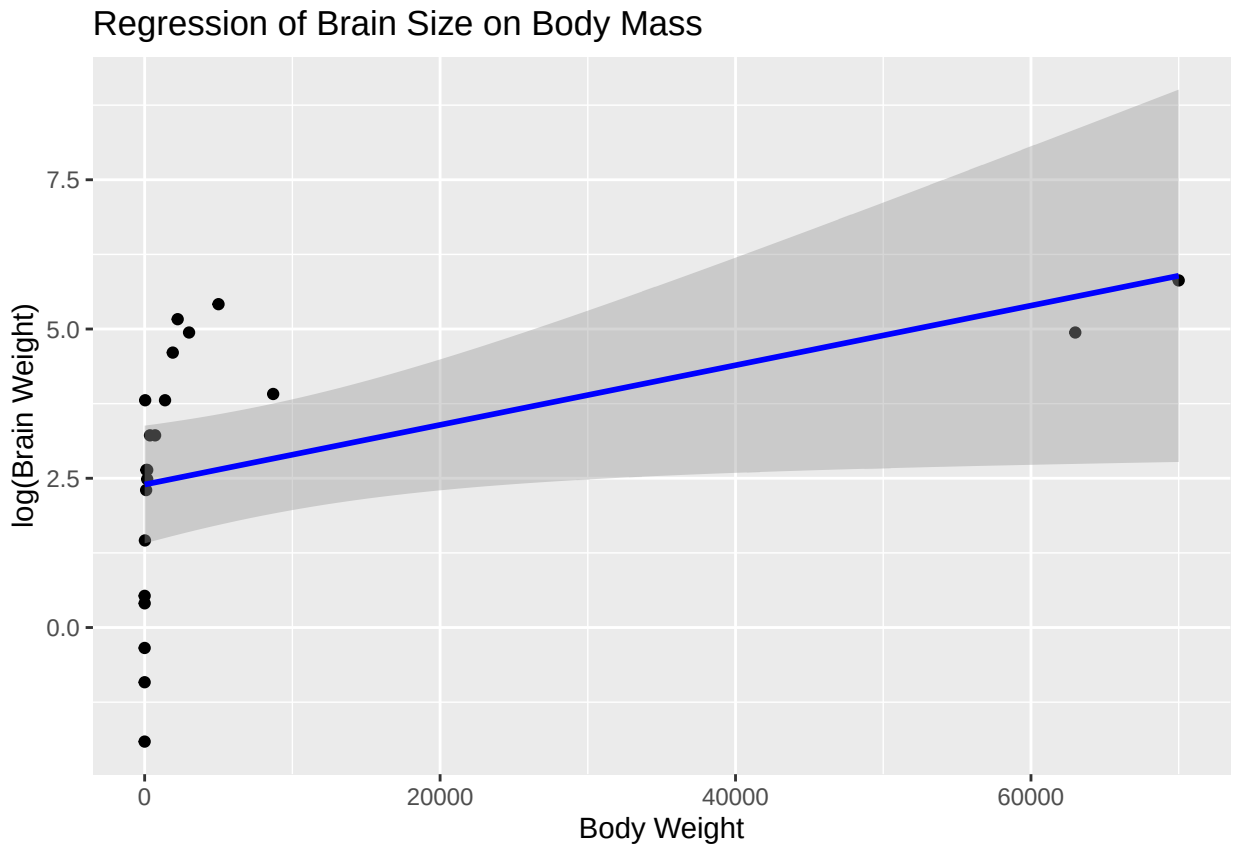
```
dat_lm_log_y <- lm(log_Brain.Weight ~ Body.Weight, data = dat)
summary(dat_lm_log_y)
```

```
##
## Call:
## lm(formula = log_Brain.Weight ~ Body.Weight, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3058 -0.9376  0.2352  1.3425  2.7711
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.395e+00  4.711e-01   5.084 6.59e-05 ***
## Body.Weight 4.995e-05  2.277e-05   2.194   0.0409 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.013 on 19 degrees of freedom
```

```
## Multiple R-squared:  0.2021, Adjusted R-squared:  0.1601
## F-statistic: 4.813 on 1 and 19 DF,  p-value: 0.04089
```

```
ggplot(dat, aes(Body.Weight,log_Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="Body Weight",
       y="log(Brain Weight)",
       color="Legend")
```

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

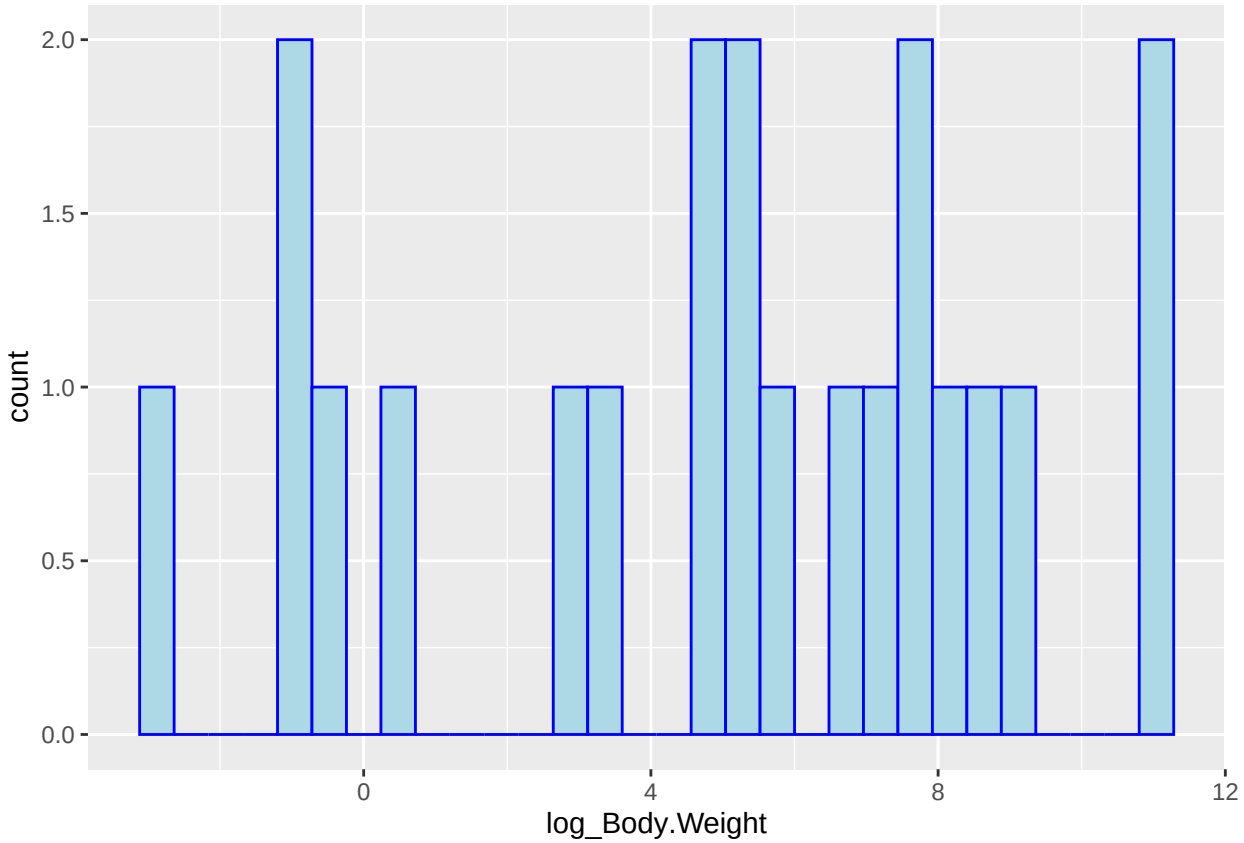


Examining the scatter plot, we observe a recurring issue from our initial model: the relationship between the logarithm of body weight and brain mass doesn't exhibit a clear linear trend. Moreover, the R-squared value has significantly declined, indicating a poorer fit than the initial model. There's room for further enhancement in our model. A potential avenue could be to consider logarithmic transformations for both the predictors and the response variable .

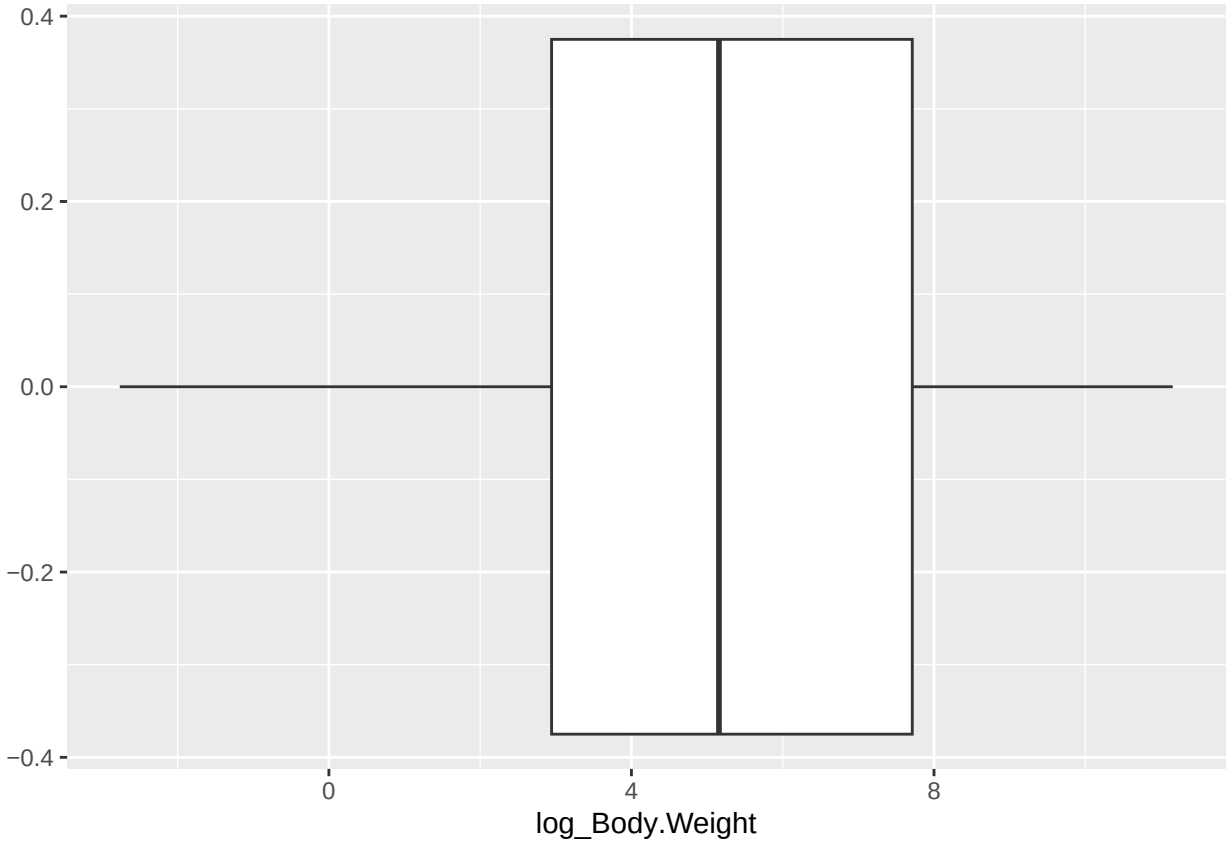
```
ggplot(dat, aes(x=log_Body.Weight)) +
  geom_histogram(fill="lightblue", color="blue" )
```

3.Logarithmic Transformation of the predictor and dependent variable.

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

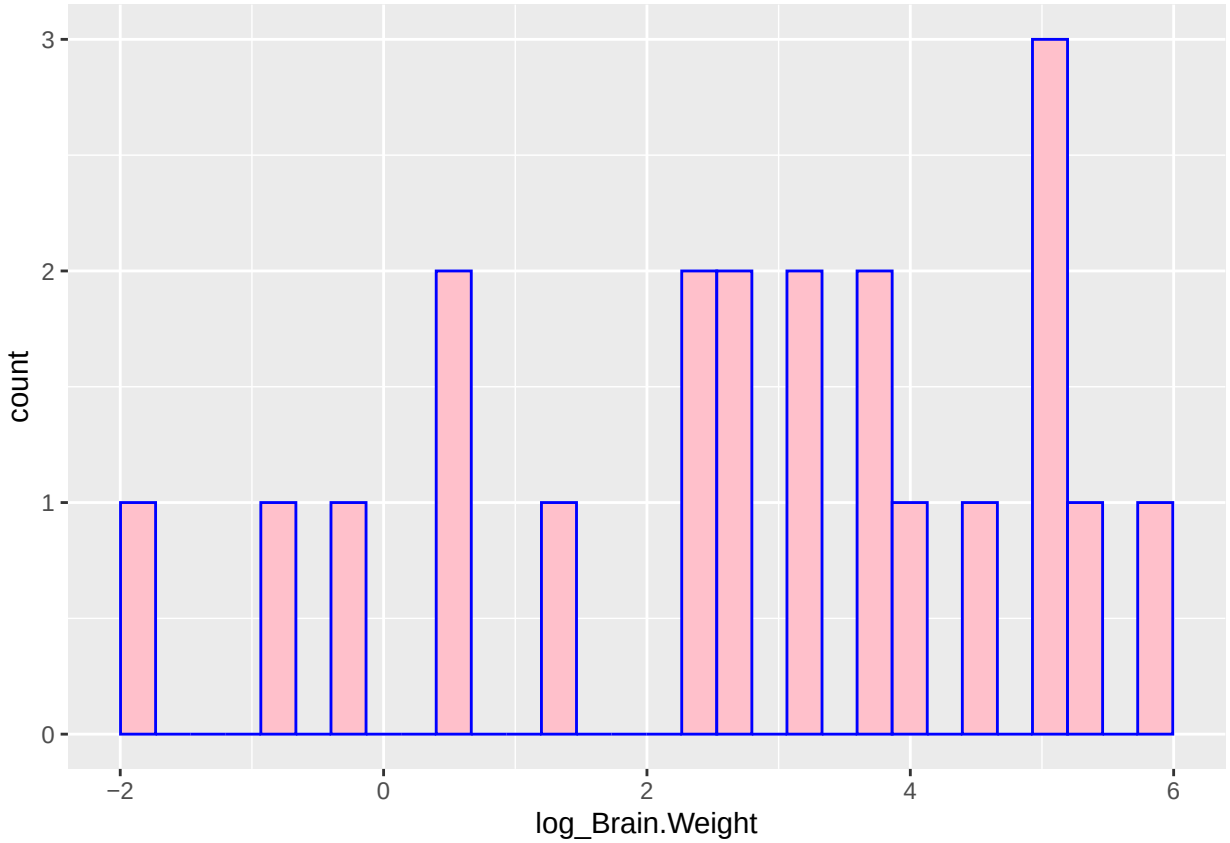



```
ggplot(dat, aes(x=log_Body.Weight)) +
  geom_boxplot()
```

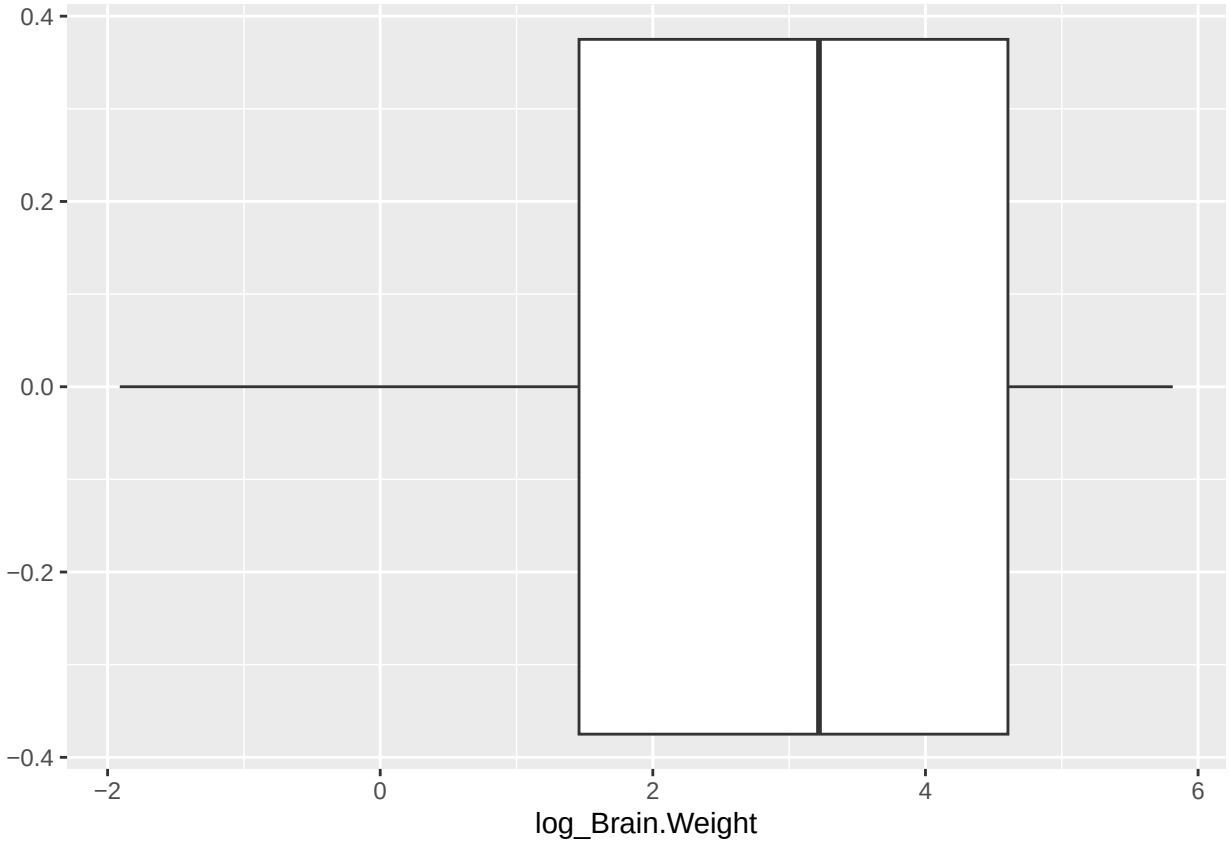



```
ggplot(dat, aes(x=log_Brain.Weight)) +
  geom_histogram(fill="pink", color="blue" )
```

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




```
ggplot(dat, aes(x=log_Brain.Weight)) +
  geom_boxplot()
```




```
summary(dat$log_Brain.Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.911   1.459   3.219   2.768   4.605   5.814
```

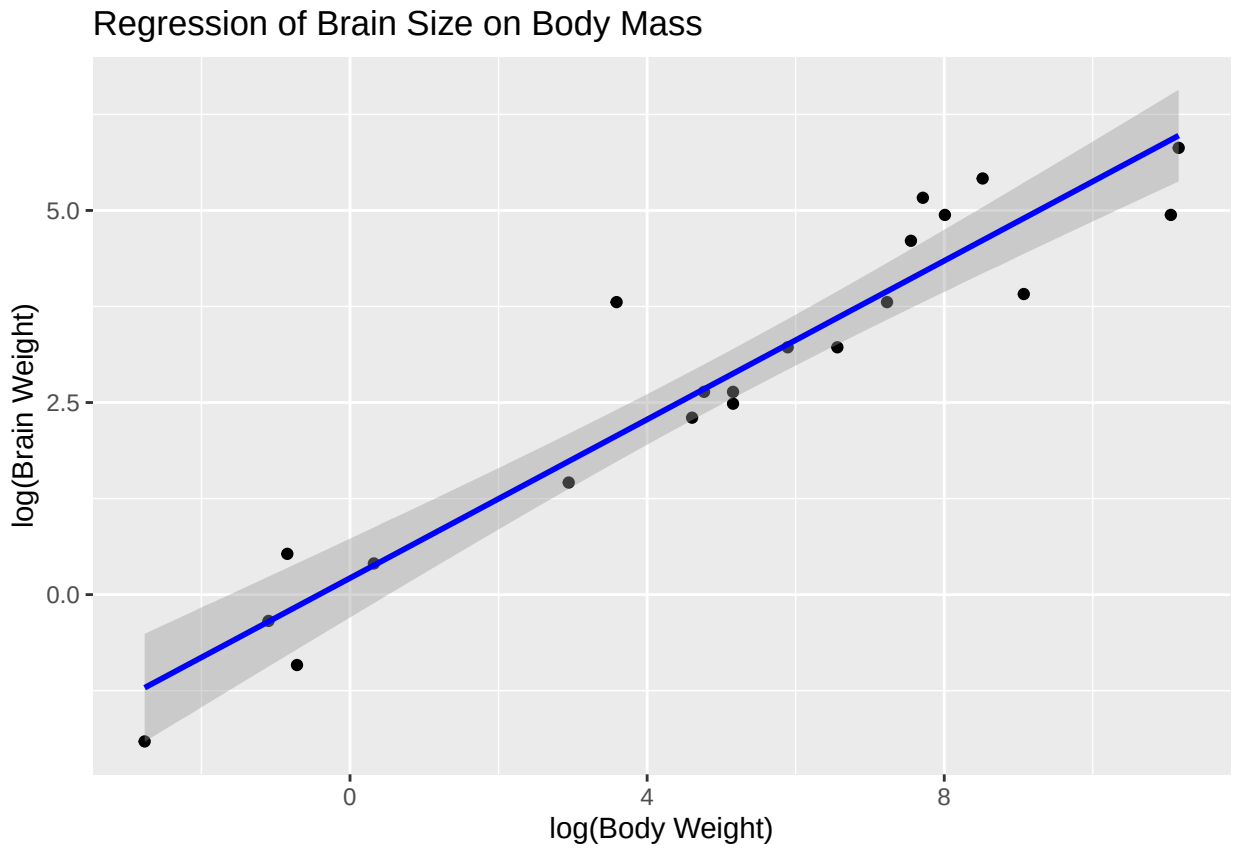
```
dat_lm_log_xy <- lm(log_Brain.Weight ~ log_Body.Weight, data = dat)
summary(dat_lm_log_xy)
```

```
##
## Call:
## lm(formula = log_Brain.Weight ~ log_Body.Weight, data = dat)
##
## 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.Weight  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
```

```
ggplot(dat, aes(log_Body.Weight,log_Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="log(Body Weight)",
       y="log(Brain Weight)",
       color="Legend")
```

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




```
summary(dat_lm_log_xy)
```

```
##
## Call:
## lm(formula = log_Brain.Weight ~ log_Body.Weight, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9856 -0.3831 -0.1405  0.4919  1.7389
##
## Coefficients:
```

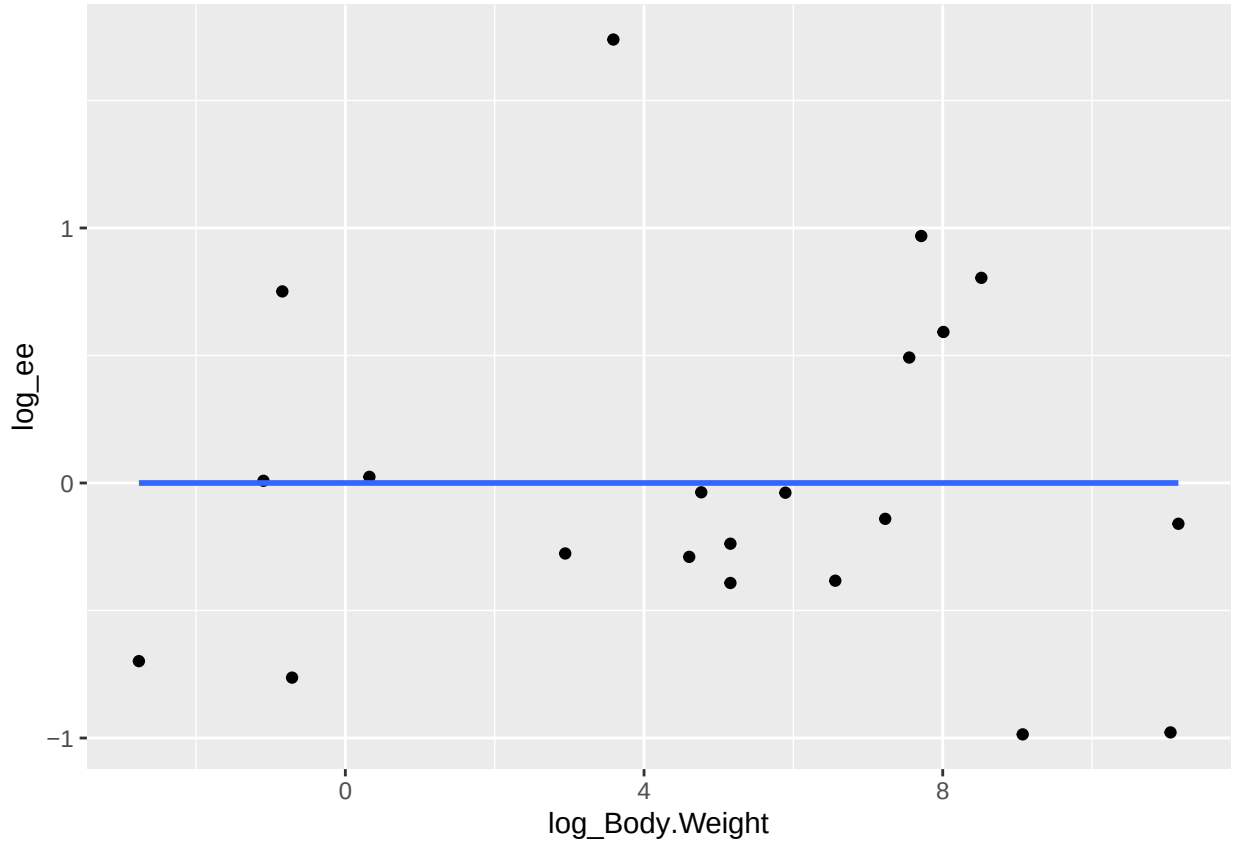
```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.21507    0.24518   0.877    0.391
## log_Body.Weight  0.51621    0.03874  13.324 4.34e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7008 on 19 degrees 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
```

The scatterplot for our model shows a trend that aligns well with a linear regression pattern, and the R-squared value has impressively increased to 0.9. It's now essential to revisit the diagnostics for this updated model.

Diagnostics Homoscedasticity

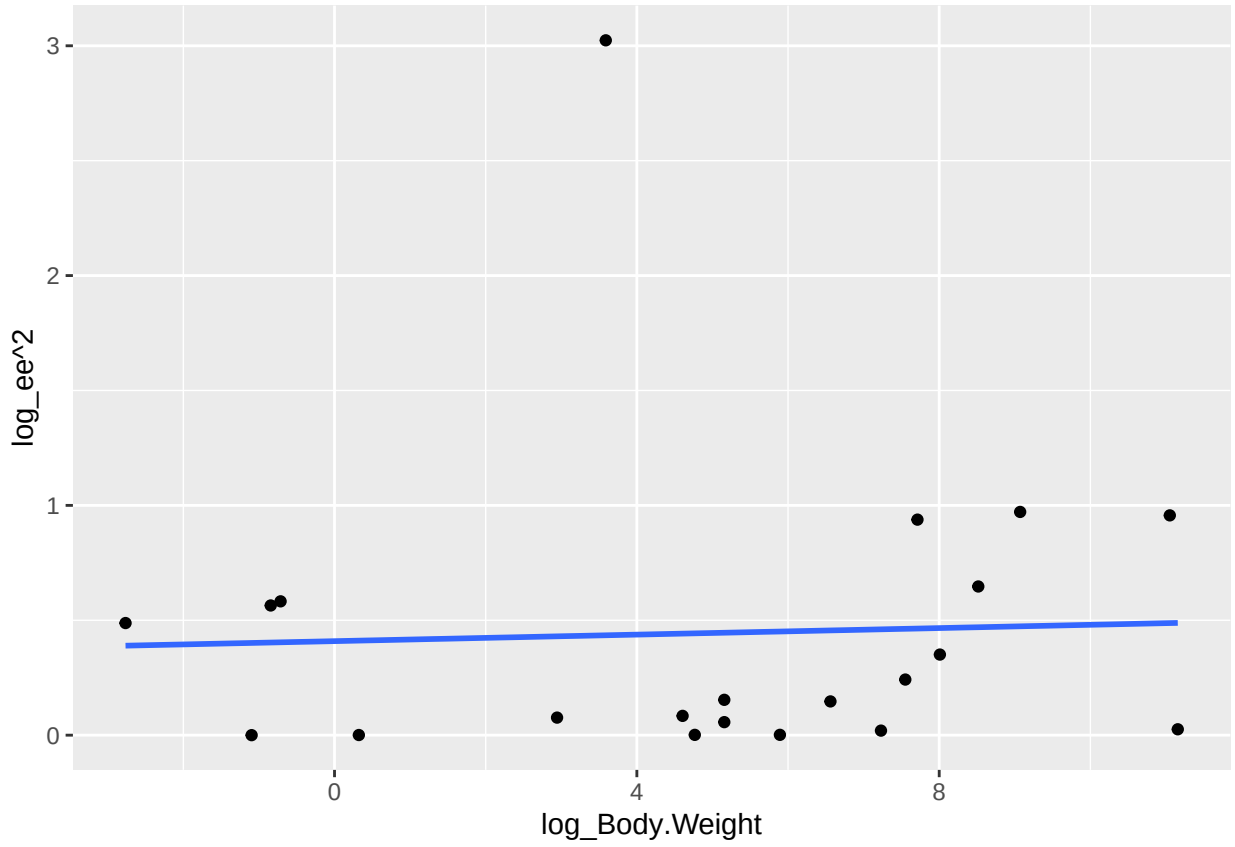
```
dat$log_ee <- dat_lm_log_xy %>%resid()
ggplot(dat, aes(log_Body.Weight,log_ee))+
  geom_point()+
  geom_smooth(method = "lm", se = F)
```

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



```
ggplot(dat, aes(log_Body.Weight,log_ee^2))+
  geom_point()+
  geom_smooth(method = "lm", se = F)
```

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



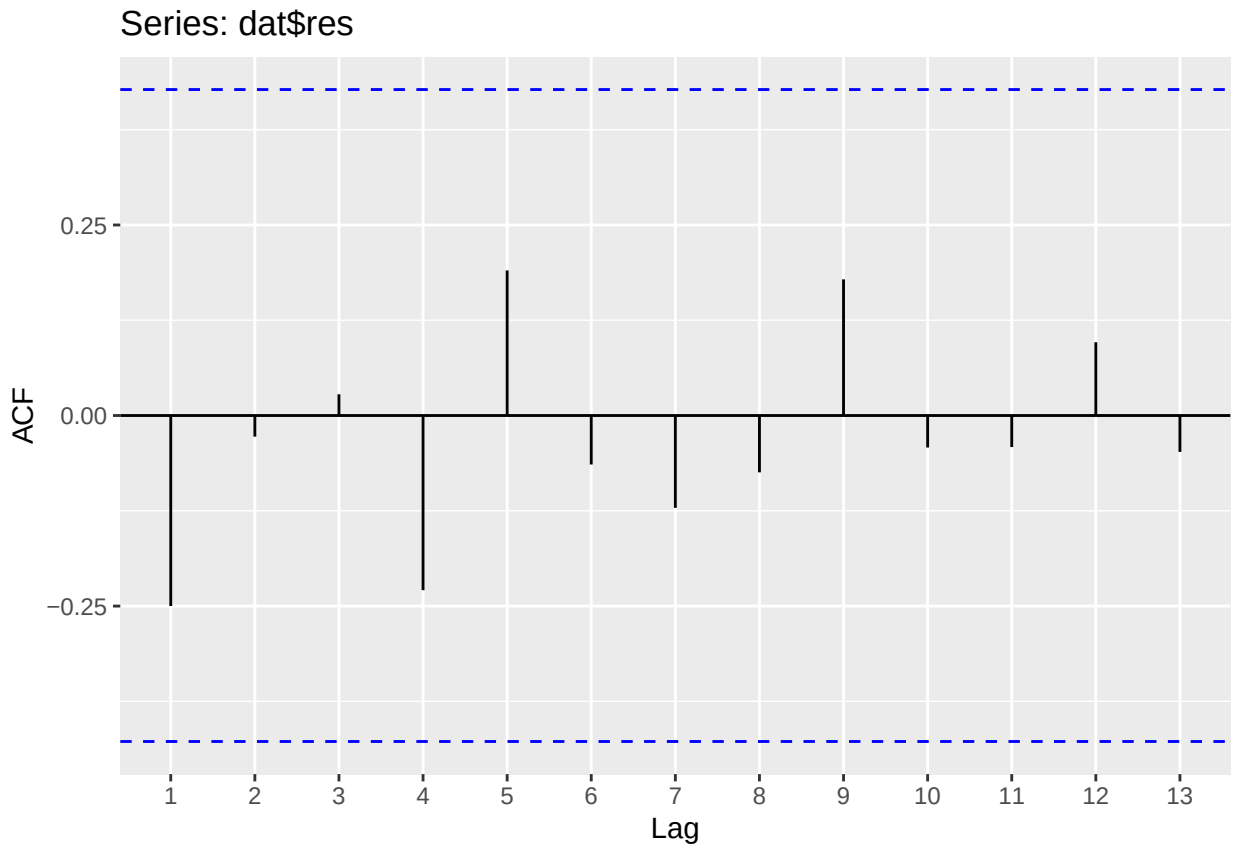

```
#breusch-Pagan test
library(lmtest)
bptest(log_Brain.Weight ~ log_Body.Weight, data = dat)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  log_Brain.Weight ~ log_Body.Weight
## BP = 0.037399, df = 1, p-value = 0.8467
```

From the residual plot, we don't observe any particular pattern; the residuals seem to be randomly scattered. Given the large p-value, we fail to reject the null hypothesis, indicating consistent variances of the residuals across the model.

Independence

```r
dat$res = dat_lm_log_xy%>%resid()
ggAcf(dat$res)
```




```r
Box.test(dat$res,type ="Ljung")
```

```
##
##  Box-Ljung test
##
## data:  dat$res
## X-squared = 1.5089, df = 1, p-value = 0.2193
```

```r
bgtest(dat_lm_log_xy)
```

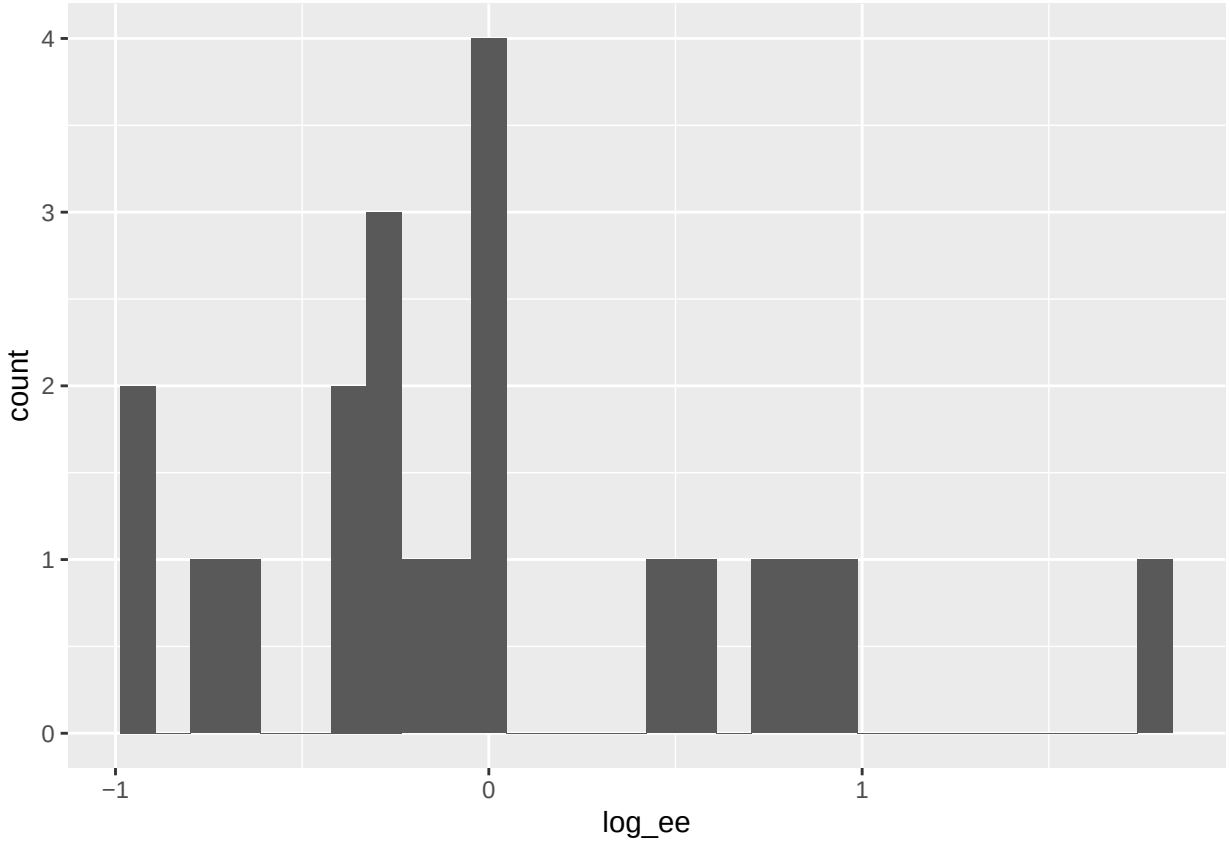
```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  dat_lm_log_xy
## LM test = 1.4792, df = 1, p-value = 0.2239
```

The ACF values at all lags fall outside the blue significance threshold, suggesting an absence of notable autocorrelation. Given that the p-values for all tests exceed 0.05, we possess substantial evidence to conclude that autocorrelation is not present.

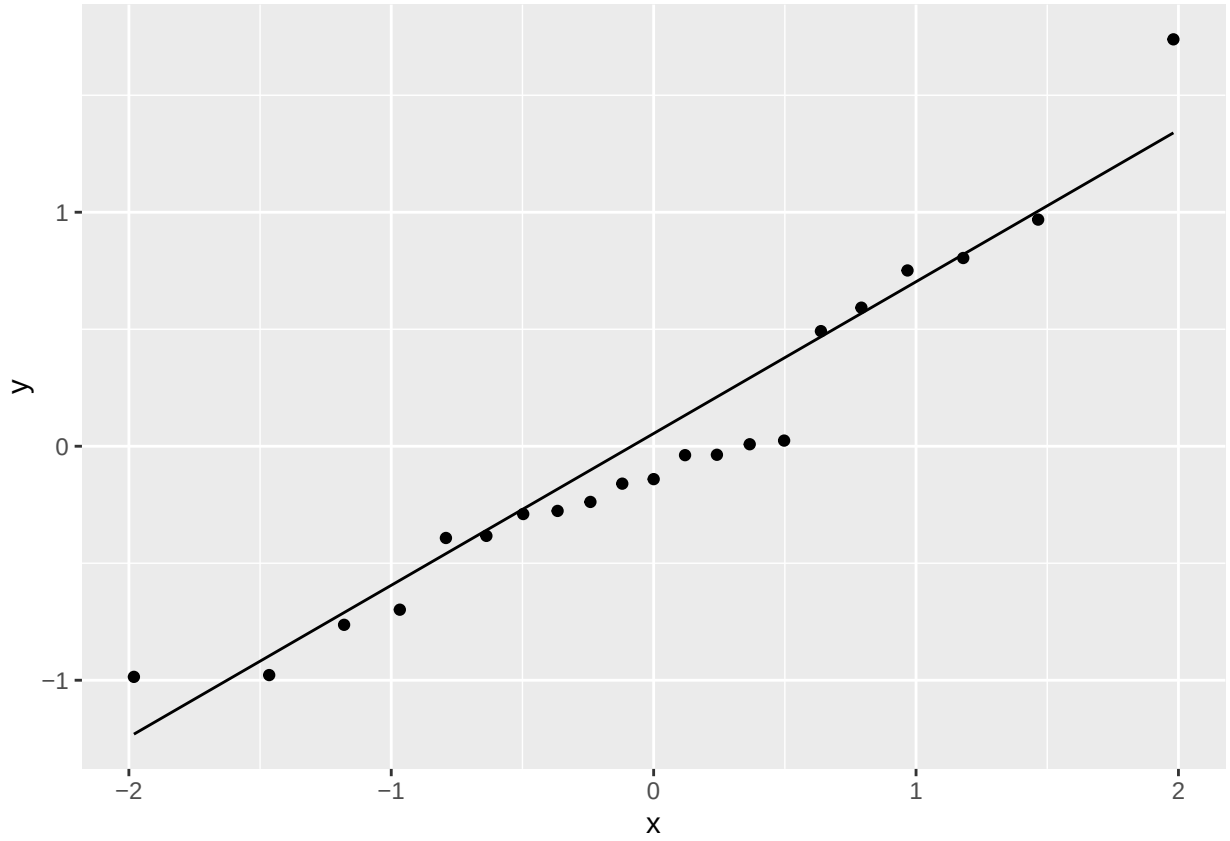
normality

```
ggplot(dat,aes(log_ee))+
  geom_histogram()
```

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




```
ggplot(dat,aes(sample =log_ee))+
  geom_qq()+
  geom_qq_line()
```




```
#shapiro-wilk normality test
shapiro.test(dat$log_ee)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dat$log_ee
## W = 0.93947, p-value = 0.2127
```

Based on the histogram and QQ plot, the residuals don't distinctly exhibit a normal distribution, likely attributable to our limited sample size. Given this small dataset, the Shapiro-Wilk test is a more appropriate choice for testing residual normality. Considering the test's p-value, we can conclude that the residuals are consistent with a normal distribution.

3.Based on the diagnostic tests and the R-squared value, we can reasonably infer that the model exhibits good fitness to the data.

4.In "Relative Brain Size and Behavior of Archosaurian Reptiles" (Annual Review of Ecology and Systematics 1977, by James A. Hopson) the author cites a widely observed power law relationship:log(Brain.weight) = log(k)+2/3 * log(Body.weight)

```
summary(dat_lm_log_xy)
```

```
##
## Call:
```

```
## lm(formula = log_Brain.Weight ~ log_Body.Weight, data = dat)
##
## 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.Weight  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
```

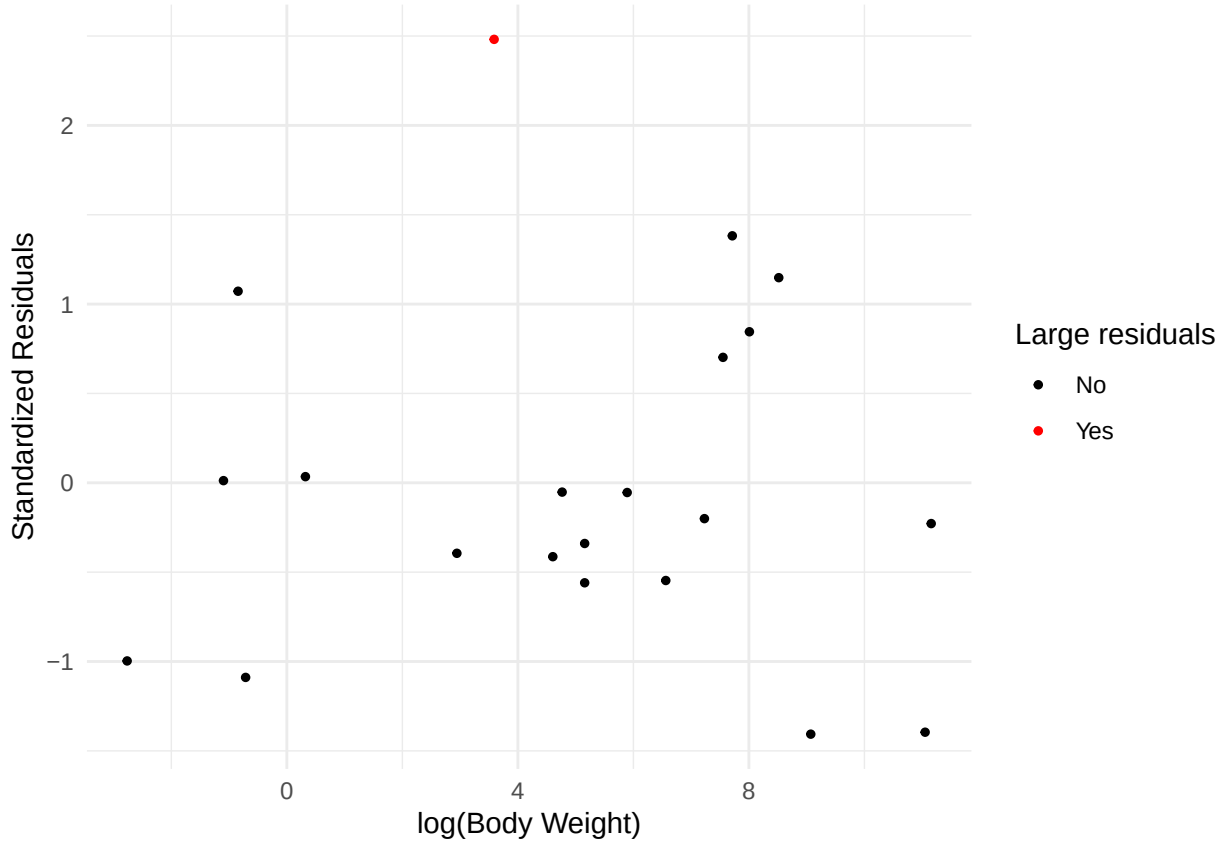
```
#the intercept is 0.2151 and slope is 0.5162
estimate <- 0.51621
se <- 0.03874
df <- 19

# Compute the critical t-value for 95% confidence level
t_critical <- qt(0.975, df)


lower_bound <- estimate - t_critical * se
upper_bound <- estimate + t_critical * se

cat("95% Confidence Interval for the slope:", lower_bound, "-", upper_bound)
```

```
## 95% Confidence Interval for the slope: 0.4351262 - 0.5972938
```

Since the value 2/3 (0.6667) does not fall within this interval, then it suggests that the slope in your data is significantly different from 2/3.

Due to the limited size of our dataset, our estimates are potentially more vulnerable to variance and influential points. With larger datasets, we might observe estimates that align more closely with the proposed relationships. Thus, it's imperative to further investigate the model's outliers and high leverage points

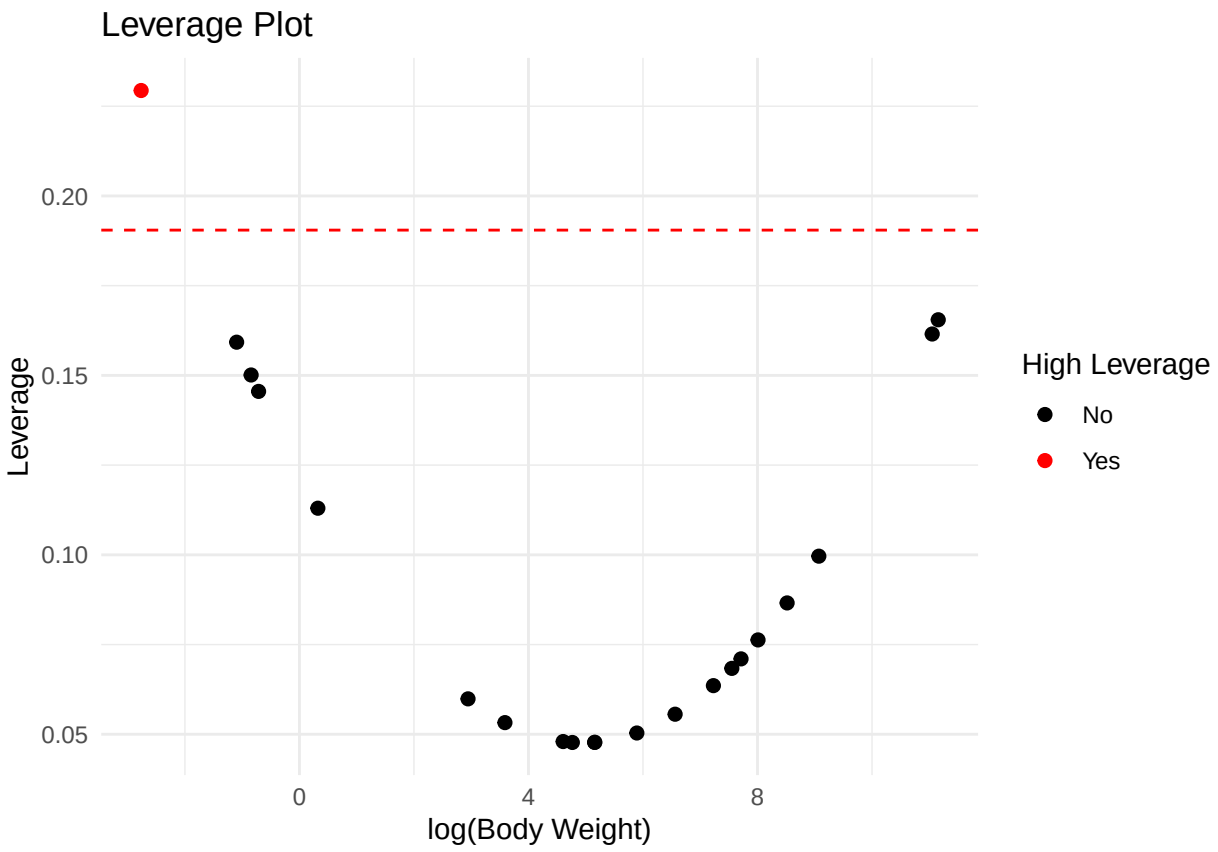
```
# Define a threshold
threshold <- 2
dat$ee.star = dat_lm_log_xy$residuals /summary(dat_lm_log_xy)$sigma
# Create a plot
ggplot(dat, aes(x = log_Body.Weight, y = ee.star)) +
  geom_point(aes(color = abs(ee.star) > threshold), size = 1) +
  scale_color_manual(values = c("black", "red"),
                     name = "Large residuals",
                     breaks = c(FALSE, TRUE),
                     labels = c("No", "Yes")) +
  theme_minimal() +
  labs(y = "Standardized Residuals", x = "log(Body Weight)")
```




```
log_outliers <- dat %>% filter(abs(ee.star) > threshold)

print(log_outliers)
```

```
##        Type          Details Body.Weight Brain.Weight     e.star
## 1 Dinosarus Stenonychosaurus        36.2           45 0.05394252
##   log_Body.Weight    log_e log_Brain.Weight  log_ee     res  ee.star
## 1        3.589059 1.593633         3.806662 1.73888 1.73888 2.481421
```

```
# Compute the leverage values
log_leverage_values <- hatvalues(dat_lm_log_xy)

P <- length(coef(dat_lm_log_xy))
N <- nrow(dat)
log_threshold_leverage <- 2*P/N


ggplot(dat, aes(x = log_Body.Weight, y = log_leverage_values)) +
  geom_point(aes(color = as.factor(log_leverage_values > log_threshold_leverage)), size = 2) +
  geom_hline(yintercept =log_threshold_leverage, linetype = "dashed", color = "red") +
  scale_color_manual(values = c("black", "red"),
                     name = "High Leverage",
                     breaks = c("FALSE", "TRUE"),
                     labels = c("No", "Yes")) +
  theme_minimal() +
  labs(y = "Leverage", x = "log(Body Weight)", title = "Leverage Plot")
```




```
log_high_leverage_points <- dat[log_leverage_values > log_threshold_leverage, ]
print(log_high_leverage_points)
```

```
##        Type Details Body.Weight Brain.Weight     e.star log_Body.Weight
## 1 Pterosaur elegans       0.063        0.148 -0.6177907       -2.764621
##      log_e log_Brain.Weight     log_ee        res    ee.star
## 1 57.54425        -1.910543 -0.6984871 -0.6984871 -0.9967571
```

Upon examining the model's high leverage points and outliers, I noticed a V-shaped trend in the leverage plot. This pattern suggests that observations with extremely low and high body weights exert significant influence on the model. Given our limited sample size, it's understandable that our estimated slope deviates from the theoretical slope of 2/3. My subsequent step will involve removing these influential points and refitting the model.

```
dat_lm_log_xy_whl <- dat[-which(log_leverage_values > log_threshold_leverage), ]

ggplot(dat_lm_log_xy_whl, aes(log_Body.Weight,log_Brain.Weight))+
  geom_point()+
  geom_smooth(method=lm, se=TRUE, color="blue")+
  labs(title="Regression of Brain Size on Body Mass",
       x="log(Body Weight)",
       y="log(Brain Weight)",
       color="Legend")
```

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


```
dat_lm_log_xy_whl <- lm(log_Brain.Weight ~ log_Body.Weight, dat_lm_log_xy_whl)
summary(dat_lm_log_xy)
```

```
##
## Call:
## lm(formula = log_Brain.Weight ~ log_Body.Weight, data = dat)
##
## 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.Weight  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
summary(dat_lm_log_xy_whl)
```

```
##
## Call:
## lm(formula = log_Brain.Weight ~ log_Body.Weight, data = dat_lm_log_xy_whl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9407 -0.3697 -0.1264  0.5244  1.6667
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.36391    0.27578    1.32    0.204
## log_Body.Weight  0.49485    0.04272   11.58 8.91e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6951 on 18 degrees of freedom
## Multiple R-squared:  0.8817, Adjusted R-squared:  0.8751
## F-statistic: 134.2 on 1 and 18 DF,  p-value: 8.906e-10
```

```r
# Compare the two models (with and without high-leverage points)
AIC(dat_lm_log_xy, dat_lm_log_xy_whl)
```

```
## Warning in AIC.default(dat_lm_log_xy, dat_lm_log_xy_whl): models are not all
## fitted to the same number of observations
```

```
##                   df      AIC
## dat_lm_log_xy      3 48.55887
## dat_lm_log_xy_whl  3 46.10272
```

From the results presented, even though there's a reduction in the AIC value, there's a notable discrepancy between the predicted slope of the revised model (with the high leverage point removed) and the theoretical value of 2/3. This suggests that factors other than just that specific point might be influencing the slope.