

STAT-S631

Exam 2

John Koo

Statement

On my honor, I have not had any form of communication about this exam with any other individual (including other students, teaching assistants, instructors, etc.).

Signed: *John Koo*

Writeup (Interpretations and Explanations)

See code below—I often refer to the output of code in the writeup as if they were right above the explanations/interpretations.

Problem 1

Part a

[Boxplots—see code below]

From a boxplot of each of the levels of `Type`, `Energy` for `Type = "non-echolocating bats"` and `"non-echolocating birds"` is significantly different than the baseline, `Type = "echolocating bats"`, for usual values of α . However, the plot suggests that there isn't a significant difference between `"non-echolocating bats"` and `"non-echolocating birds"`. We can test this as follows:

[F test for $\beta_2 = \beta_3$ for model with just `Type`—please see code below]

So we fail to reject the null hypothesis that $\beta_1 = \beta_2$. In other words, we cannot say that the difference in `Energy` is statistically significant between `Types "non-echolocating bats"` and `"non-echolocating birds"`.

Part b

But since they are both significantly different from the baseline, we can say it is reasonable to use `Type` as a regressor. We might want to try this model:

[Model with $\beta_2 = \beta_3$ and anova comparing this model with the first model—see code below]

Problem 2

Part a

[Scatterplot of `Energy` vs `log(Mass)` and anova comparing linear vs quadratic models]

From the scatterplot, it appears that there is no reason to use higher order terms. The ANOVA test confirms this.

Part b

[Summary of model $\text{Energy} \sim \log(\text{Mass})$]

A linear model using **Mass** appears to be appropriate in this case. The t test for β_1 is significant for reasonable values of α , with the p -value below machine precision.

Problem 3

Part a

[Scatterplot of **Energy** vs $\log(\text{Mass})$ with points colored by **Type**]

From the scatterplot, it appears that changes in **Energy** explained by **Type** is pretty much all captured by **Mass**. We can test this:

[Type II test for full model and anova comparing full model vs linear model $\text{Energy} \sim \log(\text{Mass})$]

Type II test (**Anova**):

- A model excluding $\log(\text{Mass})$ (and the interaction term, per the marginality principle), i.e., the model $\text{Energy} \sim \text{Type}$, is significantly different than the full model.
- A model excluding **Type** (and the interaction term, per the marginality principle), i.e., the model $\text{Energy} \sim \text{Mass}$, is not significantly different than the full model.
- A model excluding the interaction term, i.e., $\text{Energy} \sim \text{Mass} + \text{Type}$, is not significantly different than the full model.

Part b

Type I test (**anova**): The model $\text{Energy} \sim \log(\text{Mass})$ is not significantly different from the full model. This is consistent with the scatterplot. So we can conclude that the most appropriate model is $\text{Energy} \sim \log(\text{Mass})$.

Problem 4

Part a

Our final model is the linear model $\text{Energy} \sim \log(\text{Mass})$.

[Scatterplot of residuals of linear model vs **Mass**]

[Non-constant variance test $\sim \log(\text{Mass})$]

[Non-constant variance test $\sim \text{Type}$]

Visually, there appears to be no reason to believe that that variance isn't consistent. The non-constant variance tests confirm this.

Part b

[See code below for how these were computed]

OLS CI for β_1 : (0.740, 0.877)

HC3 estimate CI for β_1 : (0.744, 0.873)

Code and outputs

```
# packages
import::from(magrittr, `%>%`, `%<>%`)
dp <- loadNamespace('dplyr')
library(ggplot2)
import::from(car, Anova, ncvTest)

# plotting stuff
theme_set(theme_bw())

# read data
flight.df <- read.table('~/.dev/stats-hw/stat-s631/takehome2.txt')
head(flight.df)
```

	Mass		Type	Energy
1	779.0	non-echolocating	bats	3.7773481
2	628.0	non-echolocating	bats	3.5496174
3	258.0	non-echolocating	bats	3.1484534
4	315.0	non-echolocating	bats	3.1090610
5	24.3	non-echolocating	birds	0.9001613
6	35.0	non-echolocating	birds	1.3686394

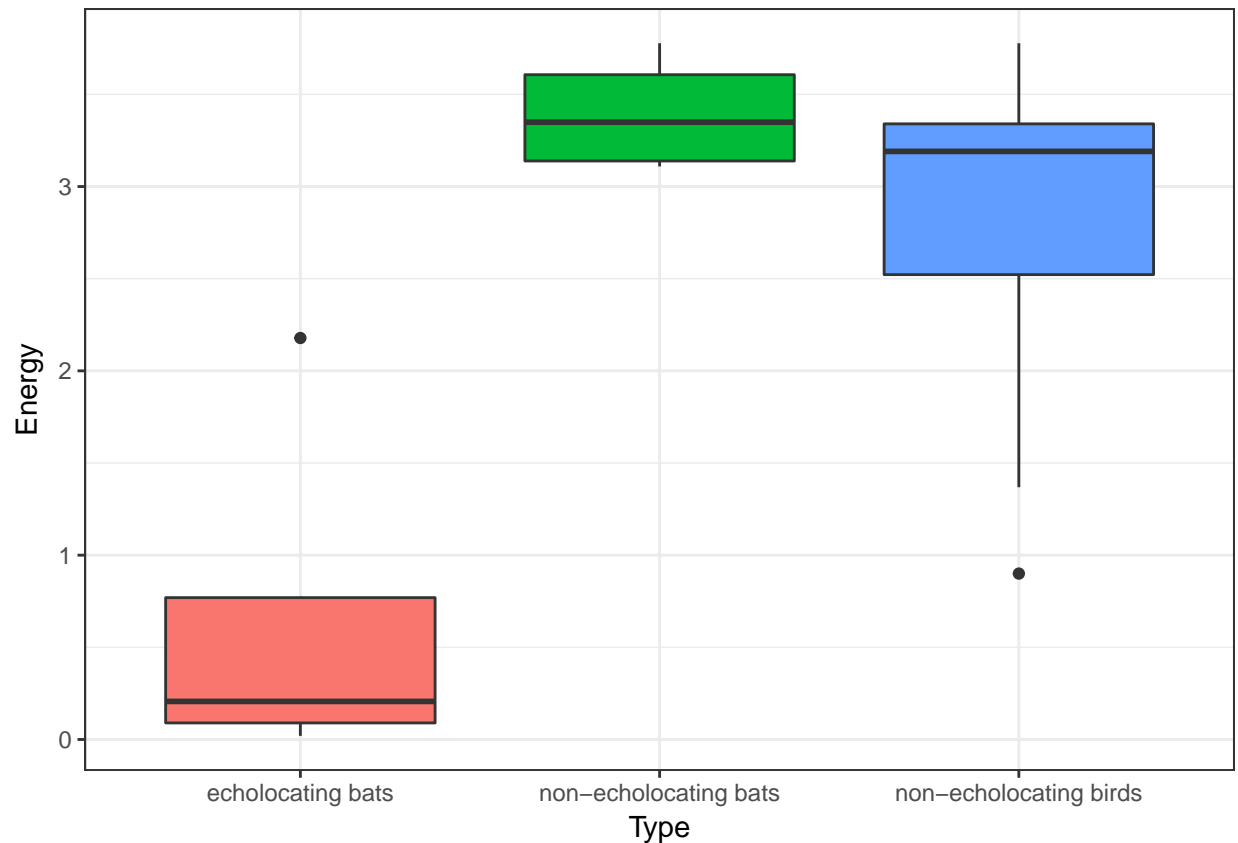
```
summary(flight.df)
```

	Mass		Type		Energy
Min.	: 6.70	echolocating	bats	: 4	Min. :0.0198
1st Qu.:	63.35	non-echolocating	bats	: 4	1st Qu.:1.9758
Median	:266.50	non-echolocating	birds	:12	Median :3.1179
Mean	:262.68				Mean :2.4822
3rd Qu.:	391.00				3rd Qu.:3.3399
Max.	:779.00				Max. :3.7773

Problem 1

Part a

```
ggplot(flight.df) +
  geom_boxplot(aes(x = Type, y = Energy, group = Type, fill = Type)) +
  scale_colour_brewer(palette = 'Set1') +
  guides(fill = FALSE)
```



```
mod.1 <- lm(Energy ~ Type, data = flight.df)
summary(mod.1)
```

Call:

```
lm(formula = Energy ~ Type, data = flight.df)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.88718	-0.39944	0.02359	0.49323	1.52531

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.6528	0.4224	1.546	0.140585
Type _{non-echolocating bats}	2.7433	0.5973	4.593	0.000259 ***
Type _{non-echolocating birds}	2.1345	0.4877	4.377	0.000411 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8447 on 17 degrees of freedom
 Multiple R-squared: 0.5953, Adjusted R-squared: 0.5477
 F-statistic: 12.5 on 2 and 17 DF, p-value: 0.0004576

```
L <- c(0, 1, -1)
c <- 0
beta.hat <- mod.1$coefficients
V.hat <- vcov(mod.1)
```

```
F.stat <- t(L %*% beta.hat - c) %*%
  solve(L %*% V.hat %*% L) %*%
  (L %*% beta.hat - c)
F.stat
```

```
      [,1]
[1,] 1.558157
```

```
1 - pf(F.stat, 1, mod.1$df.residual)
```

```
      [,1]
[1,] 0.2288543
```

Part b

```
flight.df %<>%
  dp$mutate(type = dp$if_else(Type != 'echolocating bats',
                              'non-echolocating bats/birds',
                              'echolocating bats'))
mod.2 <- lm(Energy ~ type, data = flight.df)
summary(mod.2)
```

Call:

```
lm(formula = Energy ~ type, data = flight.df)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.0394	-0.3994	0.1981	0.4803	1.5253

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.6528	0.4289	1.522	0.145309
typenon-echolocating bats/birds	2.2867	0.4795	4.769	0.000153 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8577 on 18 degrees of freedom

Multiple R-squared: 0.5582, Adjusted R-squared: 0.5337

F-statistic: 22.74 on 1 and 18 DF, p-value: 0.0001534

```
anova(mod.2, mod.1) # should be the same as before
```

Analysis of Variance Table

Model 1: Energy ~ type

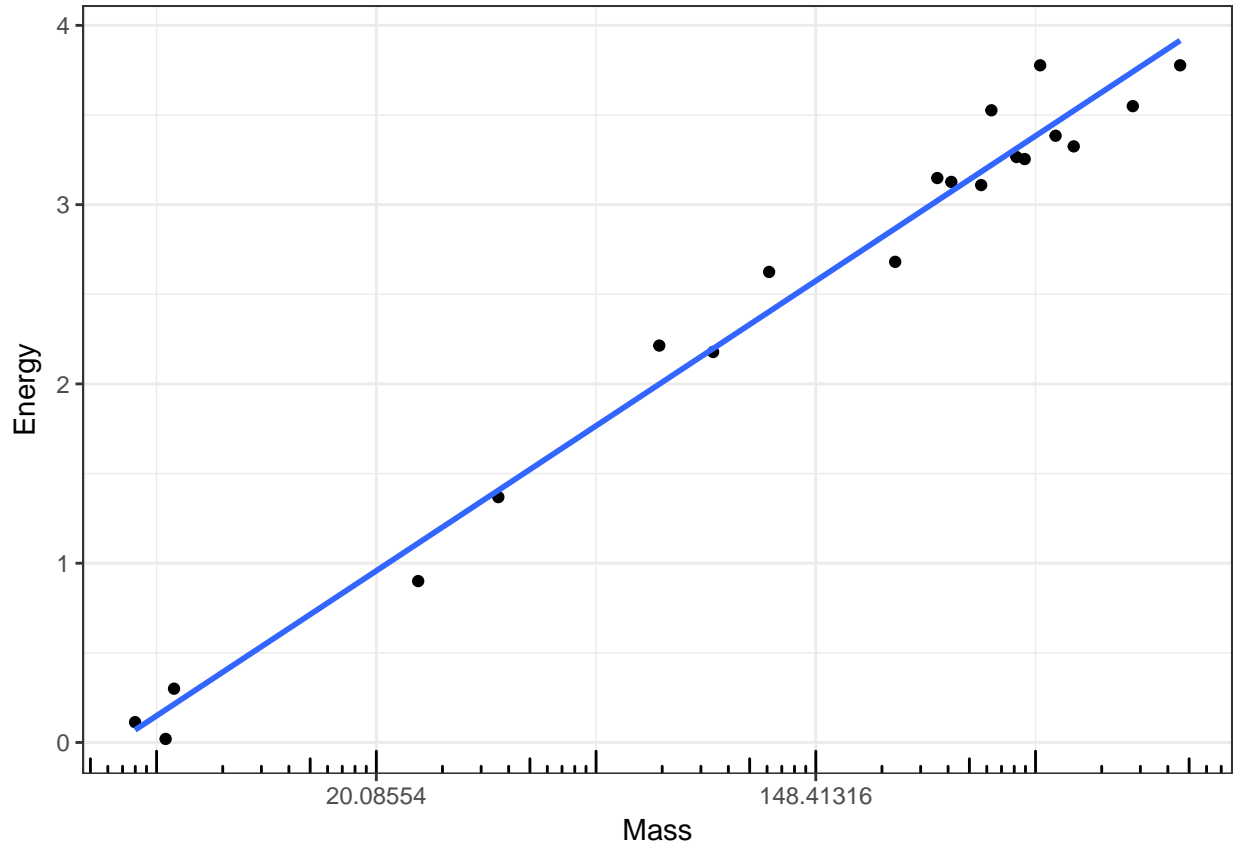
Model 2: Energy ~ Type

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	18	13.242				
2	17	12.130	1	1.1118	1.5582	0.2289

Problem 2

Part a

```
ggplot(flight.df) +  
  geom_point(aes(x = Mass, y = Energy)) +  
  scale_x_continuous(trans = 'log') +  
  annotation_logticks(sides = 'b') +  
  stat_smooth(aes(x = Mass, y = Energy), method = 'lm', se = FALSE)
```



```
lin.mod <- lm(Energy ~ log(Mass), data = flight.df)  
quad.mod <- lm(Energy ~ log(Mass) + I(log(Mass) ** 2), data = flight.df)  
anova(quad.mod)
```

Analysis of Variance Table

Response: Energy

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
log(Mass)	1	29.3919	29.3919	920.597	3.024e-16 ***
I(log(Mass)^2)	1	0.0401	0.0401	1.257	0.2778
Residuals	17	0.5428	0.0319		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Part b

```
summary(lin.mod)
```

Call:

```
lm(formula = Energy ~ log(Mass), data = flight.df)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.21143	-0.14422	-0.04284	0.09681	0.37695

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.46826	0.13716	-10.71	3.1e-09 ***
log(Mass)	0.80861	0.02684	30.13	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.18 on 18 degrees of freedom

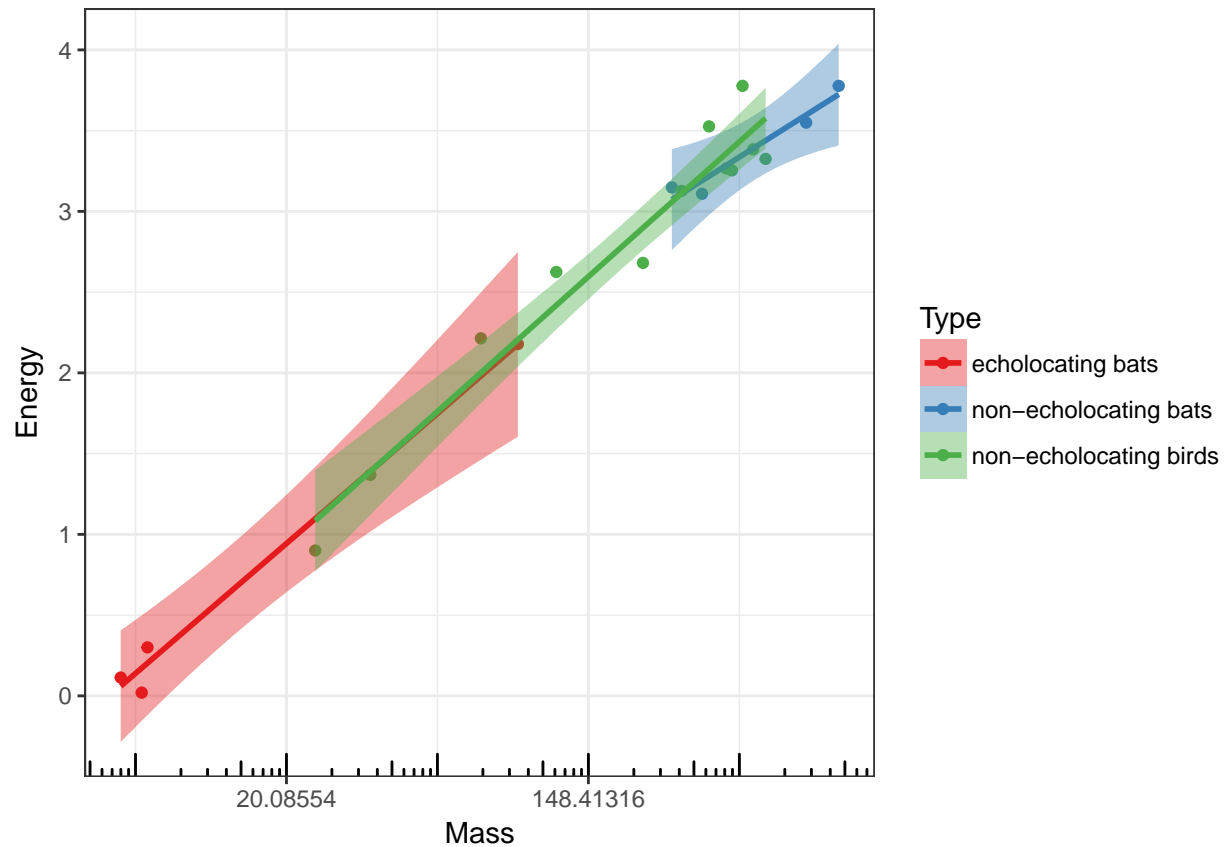
Multiple R-squared: 0.9806, Adjusted R-squared: 0.9795

F-statistic: 907.6 on 1 and 18 DF, p-value: < 2.2e-16

Problem 3

Part a

```
ggplot(flight.df) +  
  geom_point(aes(x = Mass, y = Energy, colour = Type)) +  
  scale_colour_brewer(palette = 'Set1') +  
  scale_x_continuous(trans = 'log') +  
  annotation_logticks(sides = 'b') +  
  stat_smooth(aes(x = Mass, y = Energy, fill = Type, colour = Type),  
              method = 'lm') +  
  scale_fill_brewer(palette = 'Set1')
```



Parts a and b

```
full.mod <- lm(Energy ~ log(Mass) * Type, data = flight.df)
Anova(full.mod)
```

Anova Table (Type II tests)

Response: Energy

	Sum Sq	Df	F value	Pr(>F)
log(Mass)	11.5770	1	321.0305	4.748e-11 ***
Type	0.0296	2	0.4100	0.6713
log(Mass):Type	0.0484	2	0.6718	0.5265
Residuals	0.5049	14		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
anova(lin.mod, full.mod)
```

Analysis of Variance Table

Model 1: Energy ~ log(Mass)

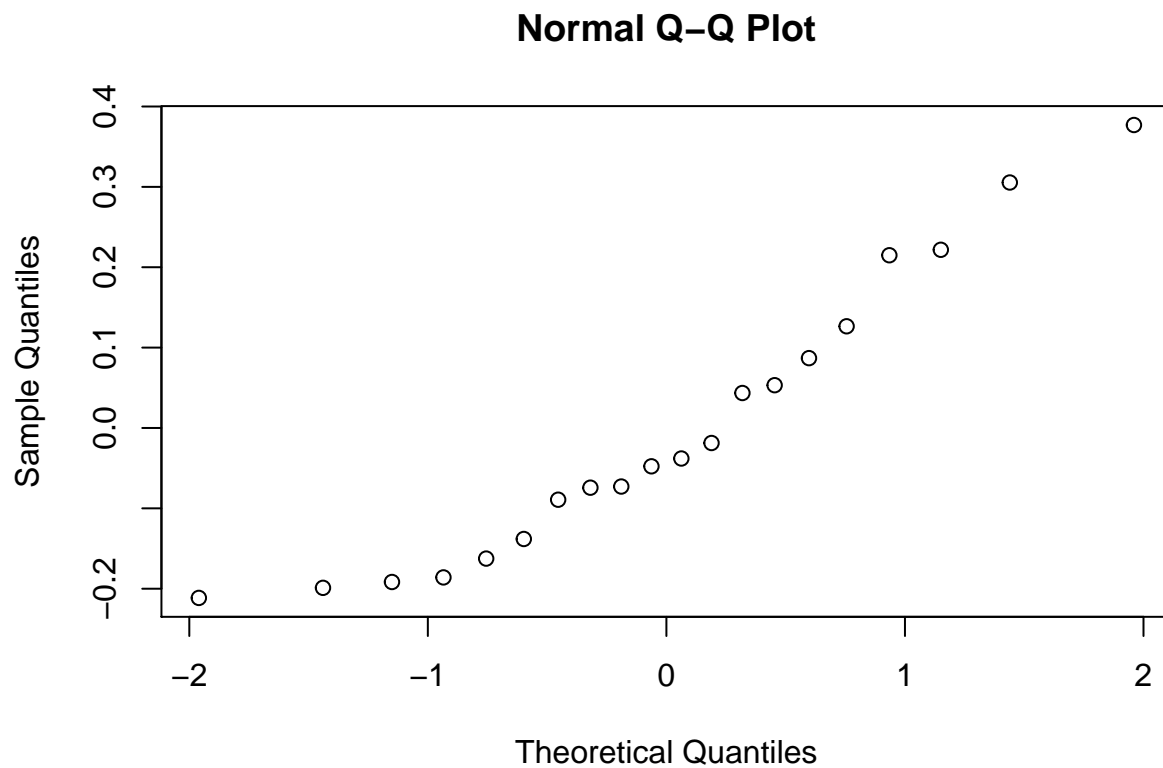
Model 2: Energy ~ log(Mass) * Type

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	18	0.58289				
2	14	0.50487	4	0.078023	0.5409	0.7084

Problem 4

Part a

```
# quick tests for normality
qqnorm(lin.mod$residuals)
```



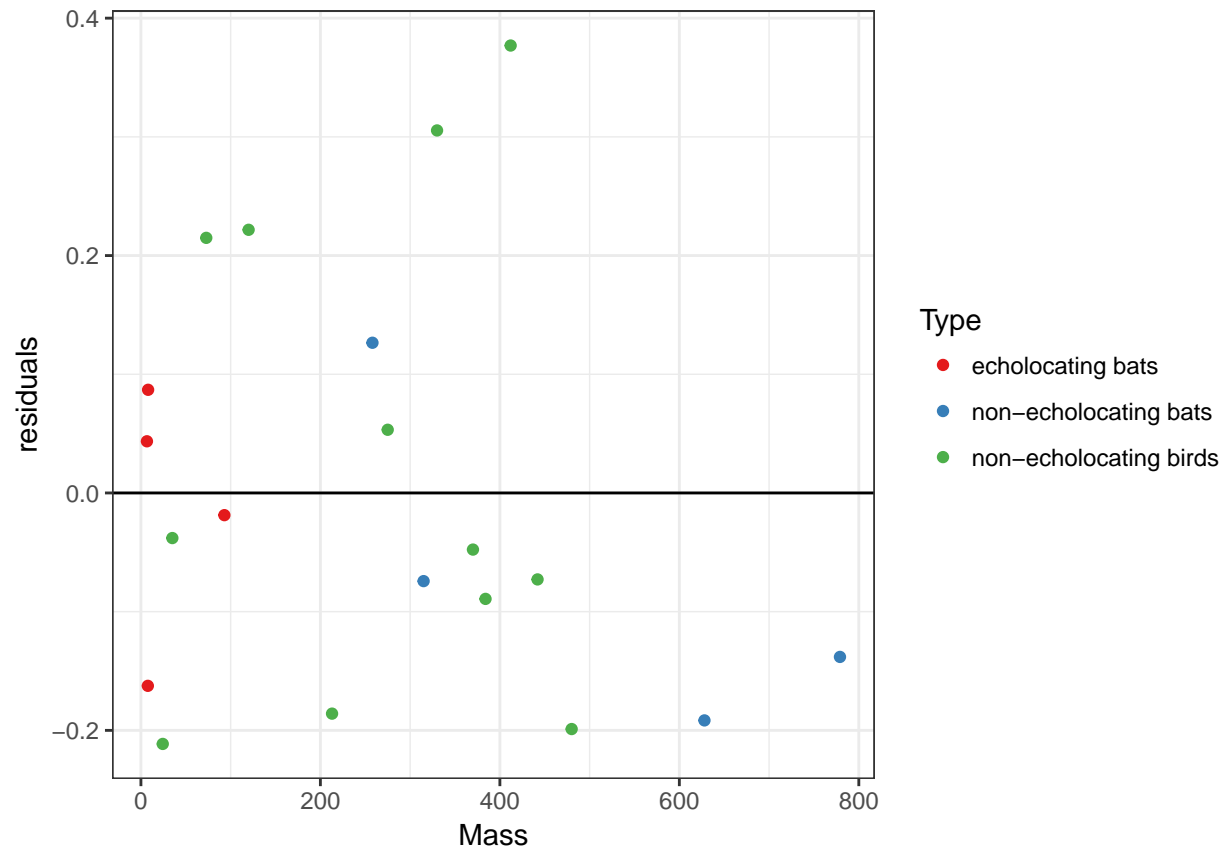
```
shapiro.test(lin.mod$residuals)
```

Shapiro-Wilk normality test

```
data: lin.mod$residuals
W = 0.92436, p-value = 0.1202
```

```
flight.df %<>%
  dp$mutate(energy.pred = predict(lin.mod, flight.df),
            resid = Energy - energy.pred)

ggplot(flight.df) +
  geom_point(aes(x = Mass, y = resid, colour = Type)) +
  geom_abline(slope = 0) +
  labs(y = 'residuals') +
  scale_colour_brewer(palette = 'Set1')
```



```
ncvTest(lin.mod, ~ log(Mass), data = flight.df)
```

```
Non-constant Variance Score Test
Variance formula: ~ log(Mass)
Chisquare = 0.6779911    Df = 1    p = 0.4102793
```

```
ncvTest(lin.mod, data = flight.df)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.6779911    Df = 1    p = 0.4102793
```

```
ncvTest(lin.mod, ~ Type, data = flight.df)
```

```
Non-constant Variance Score Test
Variance formula: ~ Type
Chisquare = 1.879569    Df = 2    p = 0.3907121
```

Part b

```
alpha <- .98

# OLS CI
mass.t <- unname(lin.mod$coefficients['log(Mass)'])
mass.se <- summary(lin.mod)$coefficients['log(Mass)', 'Std. Error']
t.98 <- qt(.5 + alpha / 2, lin.mod$df.residual)
c('lower' = mass.t - t.98 * mass.se,
```

```

'estimate' = mass.t,
'upper' = mass.t + t.98 * mass.se)

      lower estimate      upper
0.7401039 0.8086098 0.8771156
# same thing
confint(lin.mod, 'log(Mass)', .98)

      1 %      99 %
log(Mass) 0.7401039 0.8771156
# model matrix
X <- model.matrix(~ log(Mass), data = flight.df)

# response
Y <- flight.df$Energy

# projection matrix
H <- X %*% solve(t(X) %*% X) %*% t(X)

# sandwich term
S <- t(X) %*% diag(residuals(lin.mod) ** 2 / (1 - diag(H)) ** 2) %*% X

# compute standard errors
var.beta.hat <- solve(t(X) %*% X) %*% S %*% solve(t(X) %*% X)
diag(sqrt(var.beta.hat))

(Intercept)    log(Mass)
  0.12121494  0.02529946
# equivalent
diag(sqrt(sandwich::vcovHC(lin.mod)))

(Intercept)    log(Mass)
  0.12121494  0.02529946
# HC3 CI
se <- diag(sqrt(var.beta.hat))['log(Mass)']
c('lower' = mass.t - se * t.98,
  'estimate' = mass.t,
  'upper' = mass.t + se * t.98)

lower.log(Mass)      estimate upper.log(Mass)
      0.7440359      0.8086098      0.8731836

```