Stat 344 – Test 3

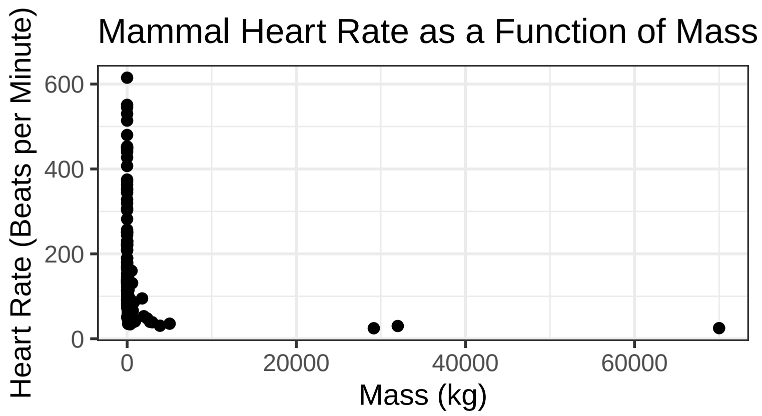
Trey Tipton

April 11, 2022

```r
mammals <- read.csv('https://sldr.netlify.app/data/mammal-hr.csv')
```

Scatter Plot

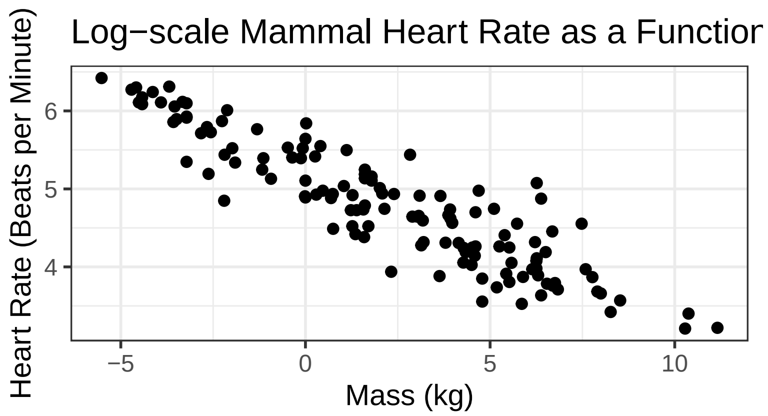
```r
gf_point(heart_rate ~ mass_kg, data = mammals) %>%
  gf_labs(title = "Mammal Heart Rate as a Function of Mass",
          x = "Mass (kg)", y = "Heart Rate (Beats per Minute)")
```



Transformations

Based on our scatter plot of heart rate as a function of mass in kilograms, they do not have a linear relationship, so we should try to make transformations on the data. We should first try moving x down or y down the ladder based on the shape of the curve. Let's try putting both on the log scale.

```r
gf_point(log(heart_rate) ~ log(mass_kg), data = mammals) %>%
  gf_labs(title = "Log-scale Mammal Heart Rate as a Function of Mass",
          x = 'Mass (kg)', y = "Heart Rate (Beats per Minute)")
```



This transformation appeared to make the data have a linear slope. Therefore, we have reason to proceed with creating a linear model using the log-scale version of the heart rates and masses.

Model, Confidence Intervals, and R-squared

a.)

```
mammals <- mammals |>
  mutate(log_y = log(heart_rate)) %>%
  mutate(log_x = log(mass_kg))

xbar <- mean(mammals$log_x)
ybar <- mean(mammals$log_y)
sy <- sd(mammals$log_y)
sx <- sd(mammals$log_x)

r <- cor(mammals$log_y ~ mammals$log_x)

beta1 <- r*(sy/sx)
beta1
```

```
## [1] -0.1903638
```

```
beta0 <- ybar - beta1*xbar
beta0
```

```
## [1] 5.221026
```

```
resids <- mammals$log_y - (beta0 + beta1*mammals$log_x)
sum <- sum(resids^2)
sigma <- sqrt(sum/length(mammals$log_y))
sigma
```

```
## [1] 0.3294668
```

The estimate for the model slope is $\hat{\beta}_1 = -0.1903638$, the intercept is $\hat{\beta}_0 = 5.221026$, and the residual standard deviation is $\hat{\sigma} = 0.3294668$.

Our model equation is:

$y_{HeartRate} = 5.221026 - 0.1903638 x_{Mass(kg)} + \epsilon$, where $\epsilon \sim Norm(0, 0.3294668)$

b.)

```r
log_y <- mammals$log_y
log_x <- mammals$log_x
n <- length(log_y)
sxx <- sum((log_x - mean(log_x))^2)
RSS <- sum((log_y - (beta0 + beta1*log_x))^2)
MSE <- RSS / (n - 2)
SE_beta1 <- (sqrt(MSE))/(sqrt(sxx))
t_star <- qt(0.975, df = (n - 2))

upper <- beta1 + t_star*SE_beta1
lower <- beta1 - t_star*SE_beta1

conf_int <- c(lower, upper)
conf_int
```

```
## [1] -0.2054505 -0.1752772
```

A 95% confidence interval for the slope estimate of our model is (-0.2054505, -0.1752772). This means that we can be 95% confident that the true slope of the linear relationship between all mammals' heart rate and mass (converted to the log-scale) is contained in the interval (-0.2054505, -0.1752772).

c.)

```r
r_squared <- (((n*sum(log_x*log_y)) - (sum(log_x)*sum(log_y))) /
                (sqrt(n*sum(log_x^2) - (sum(log_x)^2))*sqrt(n*sum(log_y^2) - (sum(log_y)^2))))^2
r_squared
```

```
## [1] 0.8285157
```

The R^2 value for this model is 0.8285157. This means that 82.8% of the variation in the heart rate of mammals is explained by the model.

Checking answers using lm() and confint():

```r
mammals_model <- lm(log(heart_rate) ~ log(mass_kg), data = mammals)
msummary(mammals_model)
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.221026   0.033624  155.28   <2e-16 ***
## log(mass_kg) -0.190364   0.007625  -24.96   <2e-16 ***
##
## Residual standard error: 0.332 on 129 degrees of freedom
## Multiple R-squared:  0.8285, Adjusted R-squared:  0.8272
## F-statistic: 623.3 on 1 and 129 DF,  p-value: < 2.2e-16
```

```r
confint(mammals_model, level = 0.95)
```

```
##                    2.5 %     97.5 %
## (Intercept)   5.1545005  5.2875519
## log(mass_kg) -0.2054505 -0.1752772
```

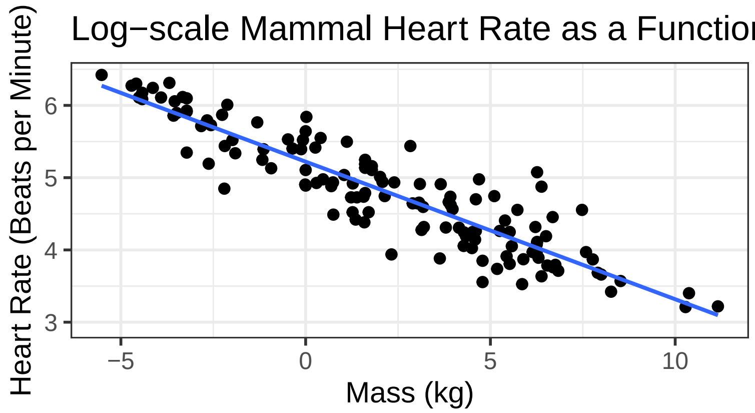
Using R, our model and confidence interval confirm our answers that were solved analytically.

Model Assesment

a.) Checking Conditions:

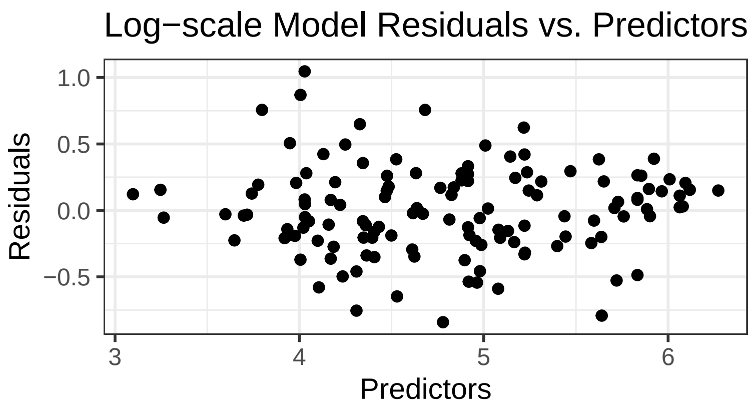
```r
mammals <- mammals %>%
  mutate(preds = predict(mammals_model), resids = resid(mammals_model))

gf_point(log(heart_rate) ~ log(mass_kg), data = mammals) %>%
  gf_lm() %>%
  gf_labs(title = "Log-scale Mammal Heart Rate as a Function of Mass with Regression Line",
          x = 'Mass (kg)', y = "Heart Rate (Beats per Minute)")
```



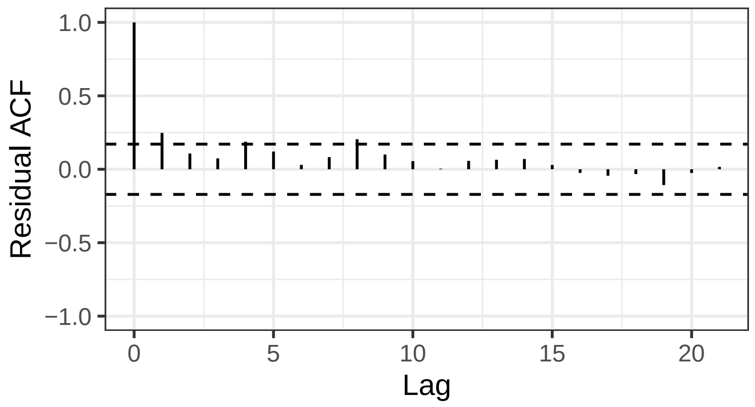
The scatter plot of the log-scale heart rate of the mammals as a function of mass in kilograms helps to check the conditions of lack of non-linearity and constant error variance. The data appears to have a downward linear trend and the residuals are evenly scattered on the y-scale above and below the regression line. Therefore these conditions are passed, but to make sure there is nothing out of the ordinary, let's look at the residuals vs. predictors scatter plot.

```r
gf_point(resids ~ preds, data = mammals) %>%
  gf_labs(title = "Log-scale Model Residuals vs. Predictors",
          x = 'Predictors', y = "Residuals")
```



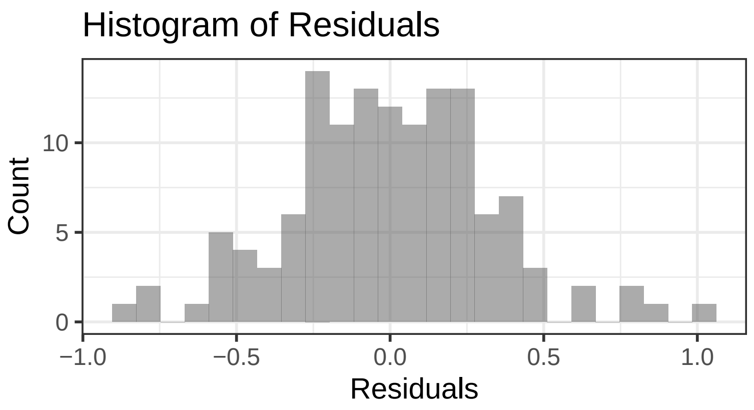
This scatter plot also checks the conditions of lack of non-linearity and constant error variance. Because the points are randomly scattered and there is no non-linear trend, the first condition is passed. In addition, since the points are also scattered randomly across the y-axis and there is no trends, the model also passes the constant error variance condition.

```r
s245::gf_acf(~mammals_model) %>%
  gf_lims(y = c(-1, 1))
```



This ACF plot checks the condition of independence among residuals. Since all of the lines are mostly within the limits of -1 and 1, we can say the the model passes the independence condition.

```r
gf_histogram(~resids, data = mammals)%>%
  gf_labs(title = "Histogram of Residuals",
          x = 'Residuals', y = "Count")
```



The histogram of residuals checks the normality of residuals condition. Since the histogram is normal, unimodal, and symmetric, our model passes the normality condition.

Therefore, all of our conditions are passed, and we can say that our model is reliable.

b.)

Since our model passes all LINE conditions, we can trust the results that the model produced. This means we can also trust the calculations we did analytically including RSS, MSE, etc. assuming we did the math correctly, and our estimates for the slope, intercept, and residual standard deviation are accurate. Therefore, our model is a reliable equation that represents the linear relationship between heart rate and mass in the population of mammals.

Bayesian Estimation

a.) Fitting a model via grid search typically only works if there is minimal parameters. For this model, we are looking for two parameters, but a grid search method would still be difficult to carry out because it tries all possible combinations of the parameters. Since 100-1000 points are made per parameter, this means that at minimum 10,000 different models are created and the best is picked from that. If more parameters are included, this number grows exponentially. I do not believe that with the software I have that this could be completed.

b.)

```
# fit model
my_bayesian_fit <- stan_glm(log_y ~ log_x, data = mammals, refresh = 0)
# show summary
coef(my_bayesian_fit)
```

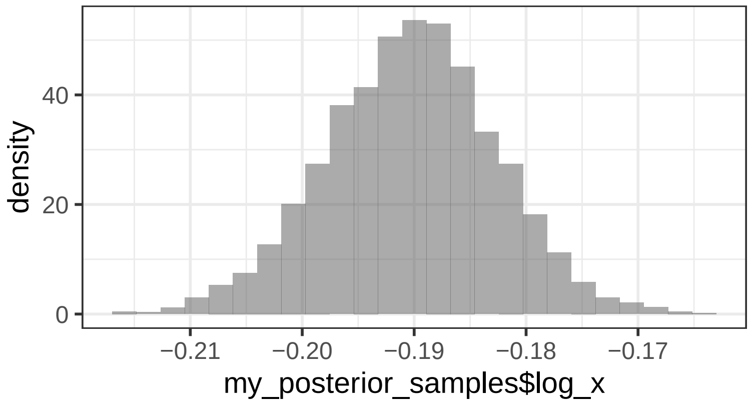
```
## (Intercept)       log_x
##   5.2207630  -0.1901674
```

```
# get posterior samples
my_posterior_samples <- as.data.frame(my_bayesian_fit)
```

c.) Unif(0, 6.43). Since I do not know much about the heart rates of animals and how that relates to their mass, I would use a uniform distribution as the prior for the slope parameter. I might expect the (both normal and log-scale) heart rates to follow a uniform distribution anyways. Therefore, I would choose non-informative prior so that it does alter the posterior.

d.)

```
gf_dhistogram(~my_posterior_samples$log_x)
```



Since our posterior sample of the slope parameter is normal, unimodal, and symmetric, we can use the mean as the point estimate and the middle 95% as the interval estimate.

```
mean(my_posterior_samples$log_x)
```

```
## [1] -0.1903002
```

```
cdata(~my_posterior_samples$log_x, 0.95)
```

```
##            lower     upper central.p
## 2.5% -0.2058748 -0.175639      0.95
```

Using the middle 95% of the data, we were able to find a 95% interval estimate using bayesian inference; that interval is (-0.2054696-0.1749021). The point estimate for the slope parameter is -0.1904199.