

# The Causal Factors That Influence Sleep- Dominick Yacono

## Data and Problem Overview

In this project, we will build linear regression models to observe the causal factors across species that influence hours of sleep.

The data set `sleepdata.csv` includes data on 51 different species from at least 13 different orders of mammals. Each case in the data set represents a species of mammal; values of the variables for each mammal are average of characteristic values for the species.

Variable Name	Description
<code>species</code>	Species of mammal
<code>TS</code>	Total sleep, hrs/day
<code>BodyWt</code>	Body weight in kg
<code>BrainWt</code>	Brain weight in g
<code>Life</code>	Maximum life span, years
<code>GP</code>	Gestation time, days
<code>D</code>	Danger index, <code>D1</code> = relatively low danger from other animals, ..., <code>D5</code> = highest level of danger from other animals

## Exploratory Analysis

First, let's explore how the response variable `TS` depends on the predictors `BodyWt`, `BrainWt`, `Life`, `GP` and `D`.

Using scatter plots, we can observe whether `BodyWt`, `BrainWt`, `Life` and `GP` should be transformed to log scale before building our regression models.

Taking the logarithm of certain predictors can help adjust nonlinear relationships and make them more linear, which is easier to model using regression techniques. It can be helpful to transform either the predictor, the response, or both variables to the log scale to make them have a linear relationship. Let's transform the predictors to the natural log scale to see if we can create linearity.

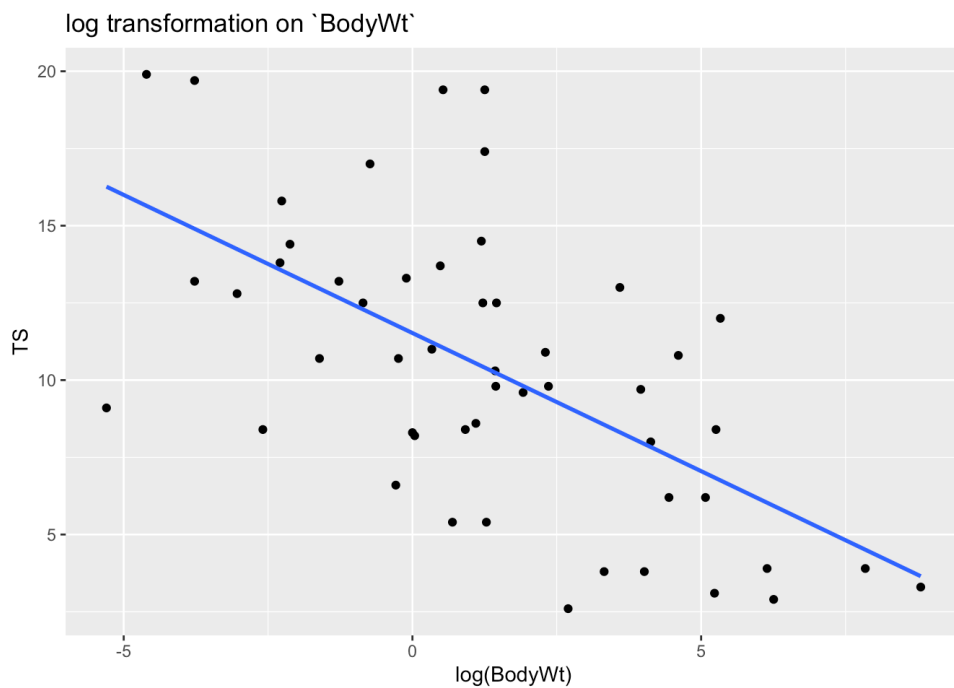
BodyWt :

```
ggplot(data = sleepdata, aes(x = BodyWt, y = TS)) + geom_point() + labs(title = "No log transformation on `BodyWt`")
```



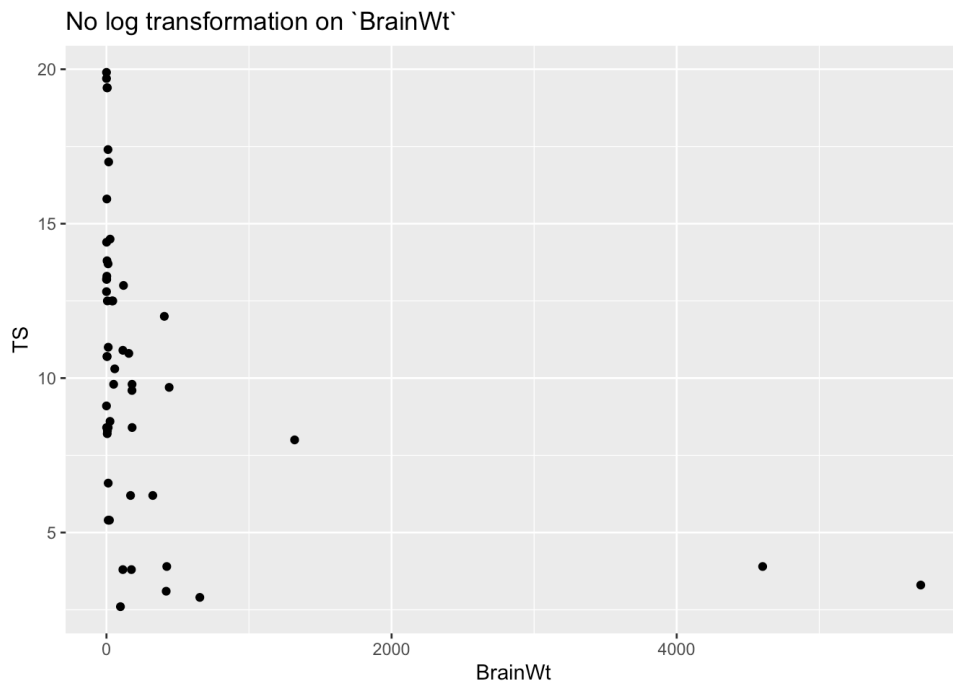
```
ggplot(data = sleepdata, aes(x = log(BodyWt), y = TS)) + geom_point() + labs(title = "log transformation on `BodyWt`") + geom_smooth(method="lm", se=F)
```

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



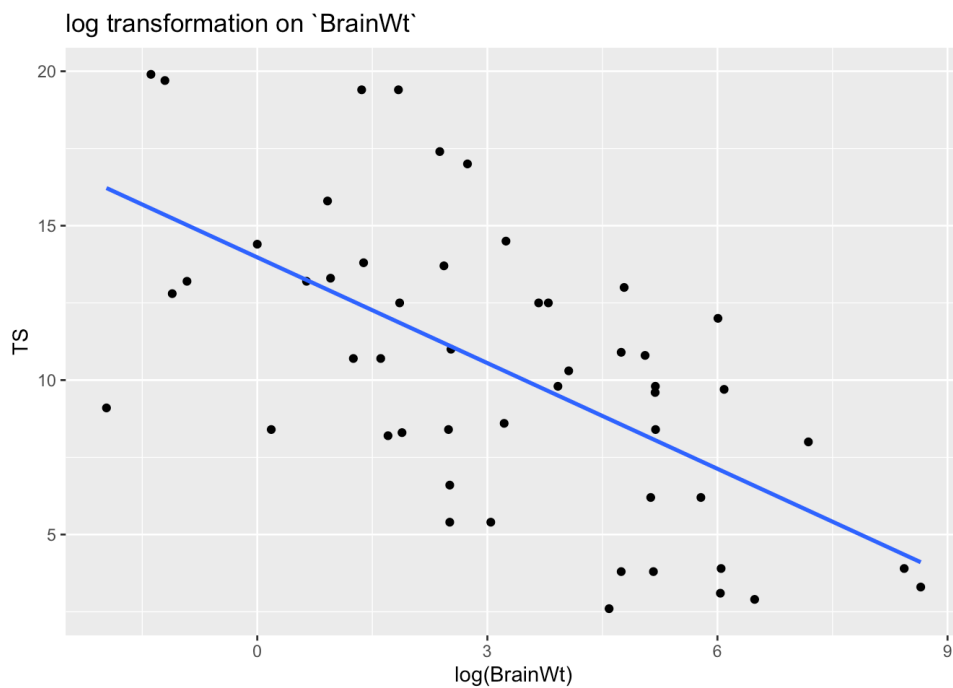
BrainWt

```
ggplot(data = sleepdata, aes(x = BrainWt, y = TS)) + geom_point() + labs(title = "No log transformation on `BrainWt`")
```



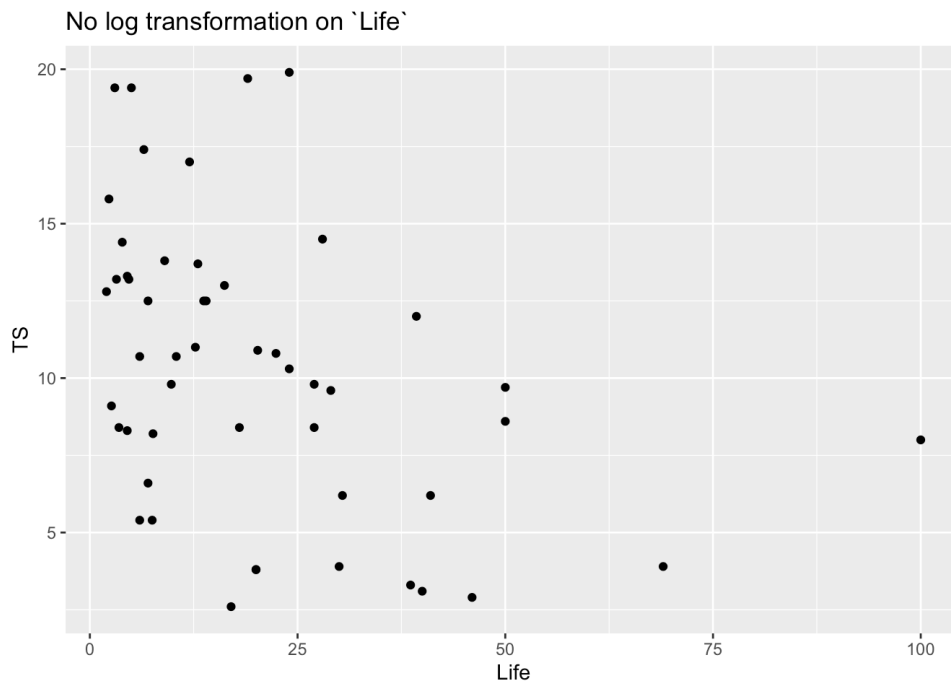
```
ggplot(data = sleepdata, aes(x = log(BrainWt), y = TS)) + geom_point() + labs(title = "log transformation on `BrainWt`") + geom_smooth(method="lm", se=F)
```

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



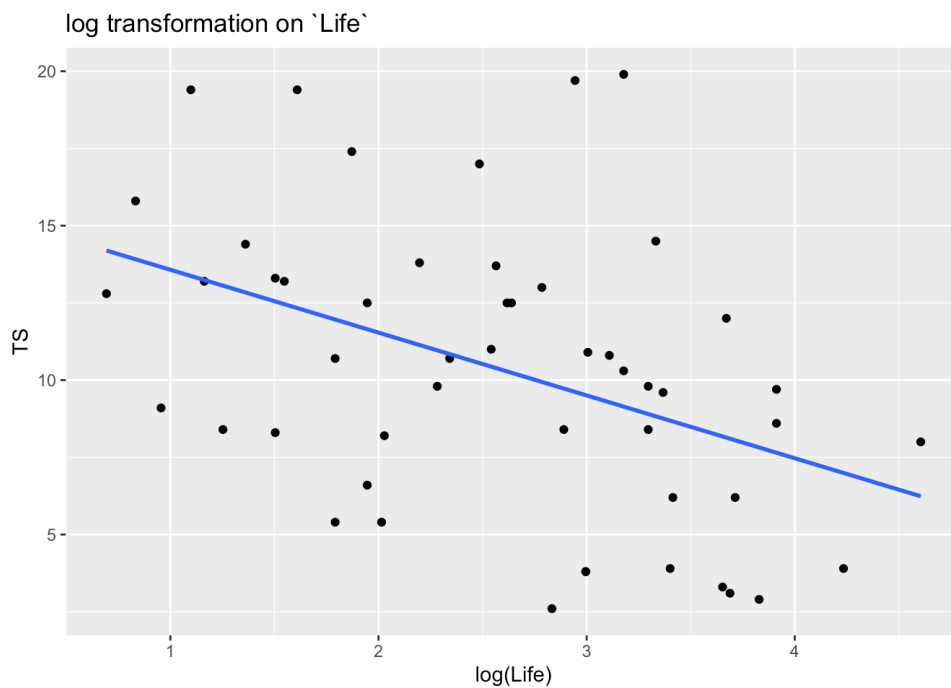
Life

```
ggplot(data = sleepdata, aes(x = Life, y = TS)) + geom_point() + labs(title = "No log transformation on `Life`")
```



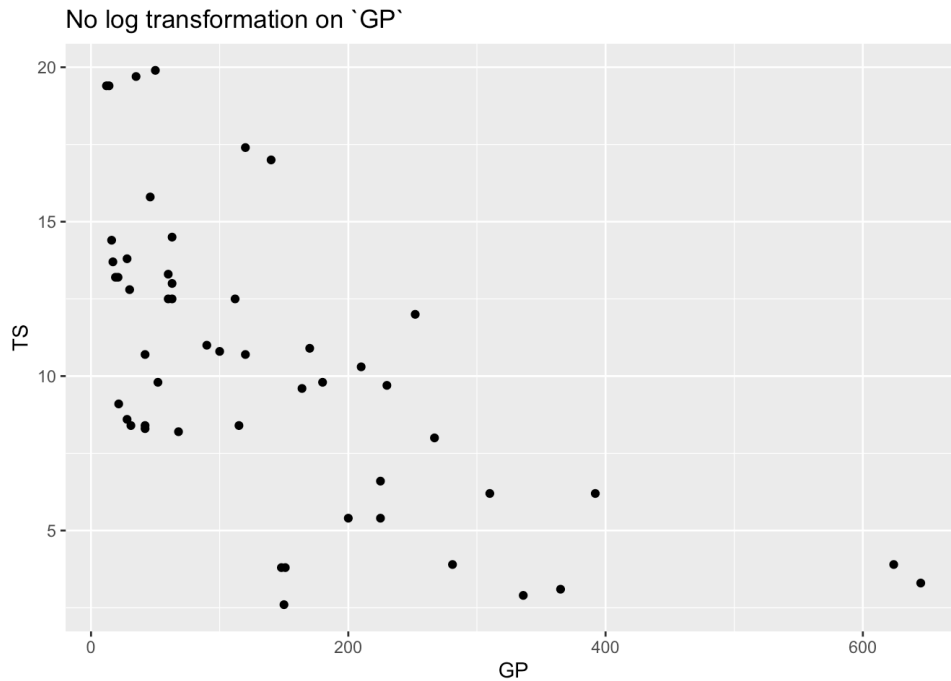
```
ggplot(data = sleepdata, aes(x = log(Life), y = TS)) + geom_point() + labs(title = "log transformation on `Life`")
) + geom_smooth(method="lm", se=F)
```

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



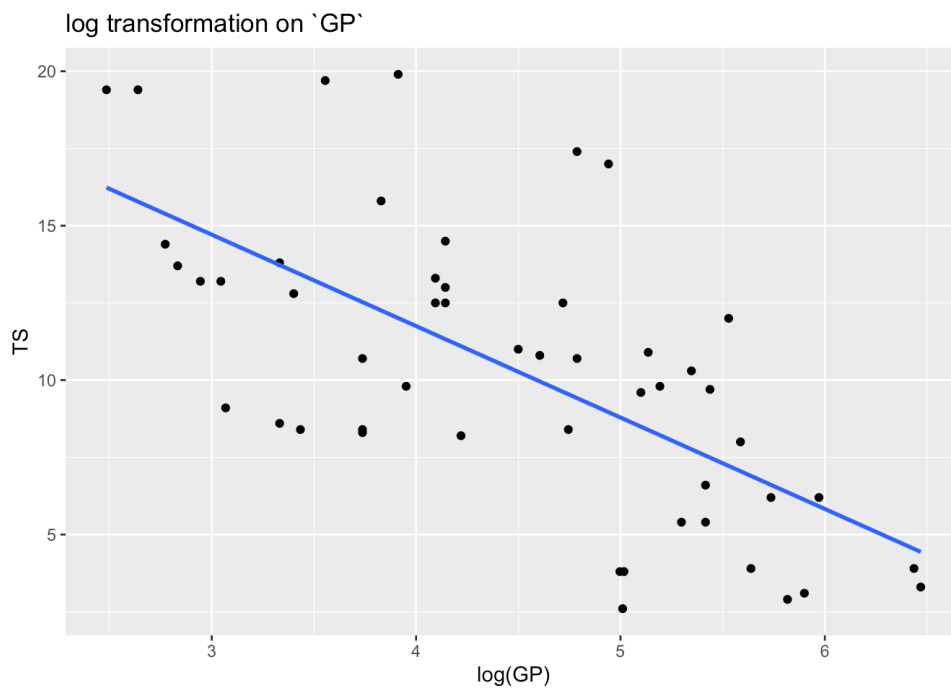
GP :

```
ggplot(data = sleepdata, aes(x = GP, y = TS)) + geom_point() + labs(title = "No log transformation on `GP`")
```



```
ggplot(data = sleepdata, aes(x = log(GP), y = TS)) + geom_point() + labs(title = "log transformation on `GP`") +
  geom_smooth(method="lm", se=F)
```

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



For the continuous predictors `BodyWt`, `BrainWt`, `Life` and `GP`, we see downward sloping linear trends between these predictors and `TS` once the predictors are put into log form. This is helpful for our linear model. It's also suggested to transform variables into log form when their set of values exceed one order of magnitude. `BodyWt`, `BrainWt`, `Life` and `GP` all have ranges of values exceeding one order of magnitude. Since we can observe linear trends between the log form of all the continuous predictors and `TS`, there is not much of a need to transform `TS` as well. Also, the values of `TS` do not exceed over one order of magnitude, so a log transformation of `TS` might not be necessarily helpful.

With that analysis out of the way, we will therefore use the natural logarithms  $\log(\text{BodyWt})$ ,  $\log(\text{BrainWt})$ ,  $\log(\text{Life})$  and  $\log(\text{GP})$  as the predictors instead of the original variables.

```
names(sleepdata)[names(sleepdata) == "BodyWt"] <- "ln(BodyWt)"
names(sleepdata)[names(sleepdata) == "BrainWt"] <- "ln(BrainWt)"
names(sleepdata)[names(sleepdata) == "Life"] <- "ln(Life)"
names(sleepdata)[names(sleepdata) == "GP"] <- "ln(GP)"

sleepdata$`ln(BodyWt)` = log(sleepdata$`ln(BodyWt)` )
sleepdata$`ln(BrainWt)` = log(sleepdata$`ln(BrainWt)` )
sleepdata$`ln(Life)` = log(sleepdata$`ln(Life)` )
sleepdata$`ln(GP)` = log(sleepdata$`ln(GP)` )
sleepdata
```

```
##          species    TS ln(BodyWt) ln(BrainWt) ln(Life) ln(GP) D
## 1 African_elephant  3.3  8.80297346  8.6503245  3.6532523  6.469250 D3
## 2 African_giant_pouched_rat  8.3  0.00000000  1.8870696  1.5040774  3.737670 D3
## 3 Arctic_Fox 12.5  1.21935391  3.7954892  2.6390573  4.094345 D1
## 4 Asian_elephant  3.9  7.84267147  8.4344635  4.2341065  6.436150 D4
## 5 Baboon 9.8  2.35612586  5.1901752  3.2958369  5.192957 D4
## 6 Big_brown_bat 19.7 -3.77226106 -1.2039728  2.9444390  3.555348 D1
## 7 Brazilian_tapir  6.2  5.07517382  5.1298987  3.4144426  5.971262 D4
## 8 Cat 14.5  1.19392247  3.2425924  3.3322045  4.143135 D1
## 9 Chimpanzee 9.7  3.95431592  6.0867747  3.9120230  5.438079 D1
## 10 Chinchilla 12.5 -0.85566611  1.8562980  1.9459101  4.718499 D4
## 11 Cow 3.9  6.14203741  6.0473722  3.4011974  5.638355 D5
## 12 Donkey 3.1  5.23164323  6.0378709  3.6888795  5.899897 D5
## 13 Eastern_American_mole  8.4 -2.59026717  0.1823216  1.2527630  3.737670 D1
## 14 Echidna 8.6  1.09861229  3.2188758  3.9120230  3.332205 D2
## 15 European_hedgehog 10.7 -0.24207156  1.2527630  1.7917595  3.737670 D2
## 16 Galago 10.7 -1.60943791  1.6094379  2.3418058  4.787492 D2
## 17 Goat 3.8  3.31998733  4.7449321  2.9957323  4.997212 D5
## 18 Golden_hamster 14.4 -2.12026354  0.0000000  1.3609766  2.772589 D2
## 19 Gorilla 12.0  5.33271879  6.0063532  3.6712245  5.529429 D1
## 20 Gray_seal 6.2  4.44265126  5.7838252  3.7135721  5.736572 D1
## 21 Gray_wolf 13.0  3.59264385  4.7833164  2.7850112  4.143135 D1
## 22 Ground_squirrel 13.8 -2.29263476  1.3862944  2.1972246  3.332205 D3
## 23 Guinea_pig 8.2  0.03922071  1.7047481  2.0281482  4.219508 D4
## 24 Horse 2.9  6.25575004  6.4846352  3.8286414  5.817111 D5
## 25 Jaguar 10.8  4.60517019  5.0562458  3.1090610  4.605170 D1
## 26 Lesser_short-tailed_shrew 9.1 -5.29831737 -1.9661129  0.9555114  3.068053 D4
## 27 Little_brown_bat 19.9 -4.60517019 -1.3862944  3.1780538  3.912023 D1
## 28 Human 8.0  4.12713439  7.1853870  4.6051702  5.587249 D1
## 29 Mouse 13.2 -3.77226106 -0.9162907  1.1631508  2.944439 D3
## 30 Musk_shrew 12.8 -3.03655427 -1.1086626  0.6931472  3.401197 D3
## 31 N._American_opossum 19.4  0.53062825  1.8405496  1.6094379  2.484907 D1
## 32 Nine-banded_armadillo 17.4  1.25276297  2.3795461  1.8718022  4.787492 D1
## 33 Owl_monkey 17.0 -0.73396918  2.7408400  2.4849066  4.941642 D2
## 34 Patas_monkey 10.9  2.30258509  4.7449321  3.0056826  5.135798 D4
## 35 Phanlanger 13.7  0.48242615  2.4336134  2.5649494  2.833213 D2
## 36 Pig 8.4  5.25749537  5.1929569  3.2958369  4.744932 D4
## 37 Rabbit 8.4  0.91629073  2.4932055  2.8903718  3.433987 D5
## 38 Raccoon 12.5  1.45582042  3.6686767  2.6173958  4.143135 D2
## 39 Rat 13.2 -1.27296568  0.6418539  1.5475625  3.044522 D3
## 40 Red_fox 9.8  1.44338333  3.9199912  2.2823824  3.951244 D1
## 41 Rhesus_monkey 9.6  1.91692261  5.1873858  3.3672958  5.099866 D2
## 42 Rock_hyrax_(Hetero._b) 6.6 -0.28768207  2.5095993  1.9459101  5.416100 D2
## 43 Rock_hyrax_(Procavia_hab) 5.4  1.28093385  3.0445224  1.7917595  5.416100 D3
## 44 Roe_deer 2.6  2.69665216  4.5870062  2.8332133  5.010635 D5
## 45 Sheep 3.8  4.01638302  5.1647860  2.9957323  5.017280 D5
## 46 Slow_loris 11.0  0.33647224  2.5257286  2.5416020  4.499810 D2
## 47 Tenrec 13.3 -0.10536052  0.9555114  1.5040774  4.094345 D2
## 48 Tree_hyrax 5.4  0.69314718  2.5095993  2.0149030  5.298317 D3
## 49 Tree_shrew 15.8 -2.26336438  0.9162907  0.8329091  3.828641 D2
## 50 Vervet 10.3  1.43270073  4.0604430  3.1780538  5.347108 D4
## 51 Water_opossum 19.4  1.25276297  1.3609766  1.0986123  2.639057 D1
```

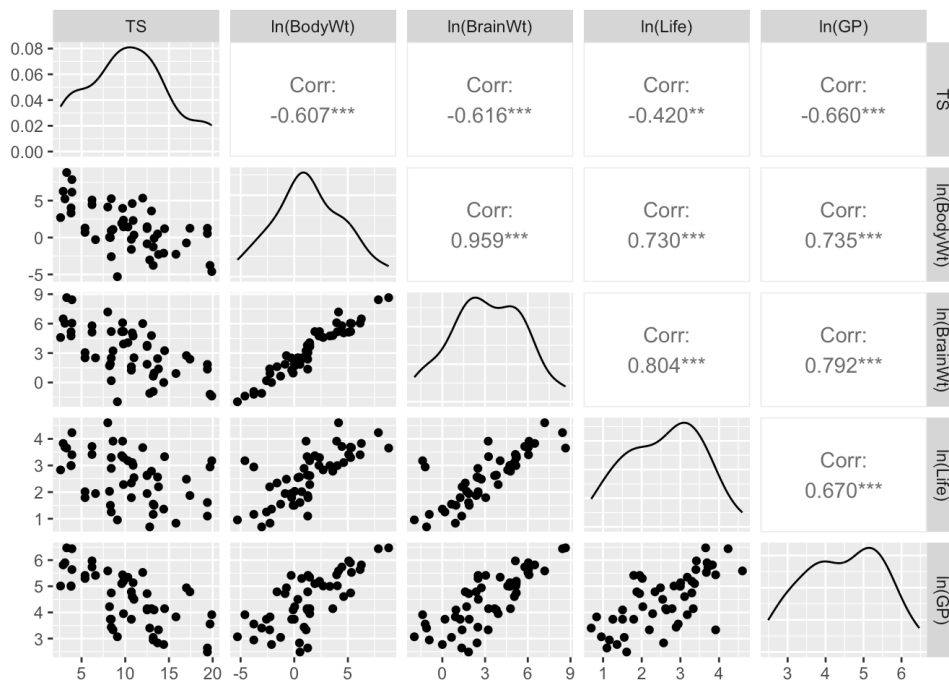
Next, we will use appropriate plots to explore the pairwise relationships between the response variable `TS` and the individual continuous predictors and between the individual continuous predictors themselves. Treating `D` as a categorical variable, we will use appropriate plots to

explore the relationship between the response  $TS$  and  $D$ .

Let's explore the relationships each individual predictor has with  $TS$ , and the relationships the continuous predictors have amongst themselves.

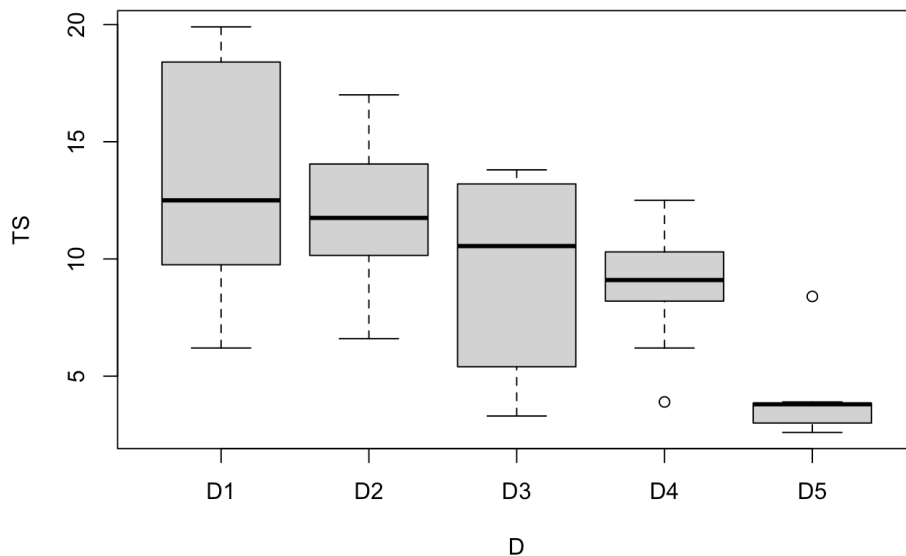
Continuous Predictors:

```
sleepdata_continuous <- sleepdata[, -c(7,7)]
sleepdata_continuous <- sleepdata_continuous[, -c(1,1)]
ggpairs(sleepdata_continuous)
```



The Categorical Predictor -  $D$ :

```
boxplot(TS ~ D, data = sleepdata)
```



Amongst the continuous predictors and the response variable, we see a variety of interesting relationships.  $TS$ , the response variable, is significantly negatively correlated with each one of the continuous predictors. We see negative correlations ranging from  $-.660$  to  $-.420$ . There are number of relationships between the predictors themselves as well.  $BodyWt$  has significant positive correlation with all other continuous predictors.  $\log(BrainWt)$  has significant positive correlation with all other continuous predictors.  $\log(Life)$  has significant positive correlation with all other continuous predictors.  $\log(GP)$  also has significant positive correlation with all other continuous predictors. In fact, all the continuous predictors are significantly positively correlated with one another. For the categorical variable  $D$ , we can see that as the danger level rises from  $D1$  to  $D5$ , there is a negative decreasing trend in  $TS$ .

### Understanding Adjusted Relationships

We will consider the full model that uses main effects only for  $\log(BodyWt)$ ,  $\log(BrainWt)$ ,  $\log(Life)$ ,  $\log(GP)$  and the categorical variable  $D$  to predict  $TS$ .

$$TS = \beta_0 + \beta_1 \ln(BodyWt) + \beta_2 \ln(BrainWt) + \beta_3 \ln(Life) + \beta_4 \ln(GP) + \beta_5 U_{D2} + \beta_6 U_{D3} + \beta_7 U_{D4} + \beta_8 U_{D5} + e_i$$

$$i = 1, \dots, n$$

where  $e_i$  is independently identically distributed and follows the following distribution:

$$N(0, \sigma^2)$$

```
main_model <- lm(TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D, data = sleepdataOriginal)
sum <- summary(main_model)
RSS.full = sum(resid(main_model)^2)

coefficients <- sum$coefficients

coefficients
```

```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept)  20.5577637   2.5165684   8.1689669 3.254293e-10
## log(BodyWt)   0.1011876   0.4727595   0.2140361 8.315546e-01
## log(BrainWt) -0.7987767   0.7115764  -1.1225453 2.680070e-01
## log(Life)     0.5756365   0.7532742   0.7641792 4.490326e-01
## log(GP)      -1.4749204   0.6466506  -2.2808614 2.769387e-02
## DD2          -1.9756427   1.1213856  -1.7617872 8.538312e-02
## DD3          -4.3761475   1.3251973  -3.3022611 1.964698e-03
## DD4          -3.1764274   1.2125726  -2.6195771 1.219858e-02
## DD5          -7.1567333   1.3220194  -5.4134858 2.755740e-06
```

First, let's analyze whether the categorical predictor  $D$  is useful in the full model. Perhaps the level of danger a mammal faces in the wild does not have a significant effect on time slept. Let's conduct the hypothesis test to assess this question.

The mean function for the full model:

$$E(TS | BodyWt_i, BrainWt_i, Life_i, GP_i, D_i) = \beta_0 + \beta_1 \ln(BodyWt_i) + \beta_2 \ln(BrainWt_i) + \beta_3 \ln(Life_i) + \beta_4 \ln(GP_i) + \beta_5 U_{D2i} + \beta_6 U_{D3i} + \beta_7 U_{D4i} + \beta_8 U_{D5i}$$

Hypothesis Test:

$H_0$  :

$$\beta_5 = \beta_6 = \beta_7 = \beta_8 = 0$$

vs

$H_1$  : at least one of

$$\beta_5, \beta_6, \beta_7, \beta_8 \neq 0$$

Since we are examining whether the type of danger level,  $D$ , has a statistically significant impact on the value of  $TS$  with all other predictors considered, we must conduct an F-test and test whether the coefficients for  $\beta_5$ ,  $\beta_6$ ,  $\beta_7$ , and  $\beta_8$  are all zero. First, we build a reduced model without these coefficients.

Reduced model:

```
main_model_noD <- lm(TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP), data = sleepdataOriginal)
summary(main_model_noD)
```



```
##
## Call:
## lm(formula = TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP),
##     data = sleepdataOriginal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.590 -2.126 -0.110  1.991  8.247
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.9662     3.1070   6.104 2.02e-07 ***
## log(BodyWt)   -0.3317     0.5696  -0.582  0.56320
## log(BrainWt)  -0.3621     0.8793  -0.412  0.68234
## log(Life)      1.1301     0.8893   1.271  0.21020
## log(GP)       -2.2248     0.7866  -2.828  0.00691 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.497 on 46 degrees of freedom
## Multiple R-squared:  0.4862, Adjusted R-squared:  0.4415
## F-statistic: 10.88 on 4 and 46 DF, p-value: 2.715e-06
```

```
RSS.reduced = sum(resid(main_model_noD)^2)
RSS.reduced
```

```
## [1] 562.526
```

Then, we conduct the F-test and plug in information from our “full” and “reduced” models.

F-stat:

$$F = \frac{\frac{RSS_{red} - RSS_{full}}{df_{red} - df_{full}}}{\frac{RSS_{full}}{df_{full}}}$$

$$F = \frac{\frac{562.526 - 314.5211}{46 - 42}}{\frac{314.5211}{42}}$$

$$F = 8.28$$

We find that out that the F-statistic is 8.28, which we can now use to calculate the p-value.

P-value:

```
pf(8.28, df1 = 46 - 42, df2 = 42, lower.tail = F)
```

```
## [1] 5.110506e-05
```

The p-value, 5.110506e-05, is below 0.05. Therefore, we can reject the null hypothesis at the 95% confidence level that  $D$  has no effect on the model (95% confidence)

The model with  $D$  takes into account a mammal's danger level index when it predicts the mammal's expected value of  $TS$ , alongside other factors. We can observe that mammal's with higher danger levels, like D5, are expected to have a lower value of  $TS$  compared to mammal's with lower danger levels (all other factors held constant and set to 0). For example, a mammal with danger level index D1 has an expected value of  $TS$  of 20.5578 (all other continuous factors at 0). On the other hand, a mammal with danger level index D5 has a lower expected value of  $TS$  of 13.4011 (all other continuous factors at 0). The model without  $D$  predicts that all mammals have an expected value of  $TS$  of 18.9662 (all other continuous factors at 0). Clearly, the model with  $D$  adds interesting insight not captured by the model without  $D$ . The result of the F-test informs us that the  $D$  predictor is valuable for our analysis. Some interesting observations are that in the model with  $D$ , we can see that the std. errors for the continuous predictors are lower than in the model without  $D$ . We can also see that the coefficients for D3, D4, and D5 are individually statistically significant when T-tests are conducted.

Leaving  $D$  in the model, let's describe the relationship between the predictors  $\log(\text{BodyWt})$  and  $\log(\text{BrainWt})$  and the response  $TS$  in the context of the MLR model with all predictors.

Using confidence intervals for our coefficients, let's describe the relationship between the predictors and the response  $TS$

```
confint(main_model)
```

```
##           2.5 %      97.5 %
## (Intercept) 15.4791231 25.6364043
## log(BodyWt) -0.8528798  1.0552550
## log(BrainWt) -2.2347959  0.6372426
## log(Life)    -0.9445323  2.0958053
## log(GP)      -2.7799142 -0.1699266
## DD2          -4.2386904  0.2874049
## DD3          -7.0505039 -1.7017911
## DD4          -5.6234979 -0.7293568
## DD5          -9.8246765 -4.4887901
```

All other factors held constant, a one unit increase in  $\log(\text{BodyWt})$  is associated with an expected change in  $\text{TS}$  between (-0.8528798, 1.0552550) (95% confidence). Therefore, we can not say for certain whether  $\log(\text{BodyWt})$  has a definitive positive or negative effect on  $\text{TS}$ .

All other factors held constant, a one unit increase in  $\log(\text{BrainWt})$  is associated with an expected change in  $\text{TS}$  between (-2.2347959, 0.6372426) (95% confidence). Therefore, we can not say for certain whether  $\log(\text{BrainWt})$  has a definitive positive or negative effect on  $\text{TS}$ .

All other factors held constant, a one unit increase in  $\log(\text{Life})$  is associated with an expected change in  $\text{TS}$  between (-0.9445323, 2.0958053) (95% confidence). Therefore, we can not say for certain whether  $\log(\text{BodyWt})$  has a definitive positive or negative effect on  $\text{TS}$ .

All other factors held constant, a one unit increase in  $\log(\text{GP})$  is associated with an expected change in  $\text{TS}$  between (-2.7799142, -0.1699266) (95% confidence). Therefore, we say it is likely  $\log(\text{BrainWt})$  has a negative effect on  $\text{TS}$ .

In the pairwise plots, we saw that the correlation coefficient between  $\log(\text{BodyWt})$  and  $\text{TS}$  was around 0.6. We saw that the correlation coefficient between  $\log(\text{BrainWt})$  and  $\text{TS}$  was around 0.6 as well. These correlation coefficients imply that  $\log(\text{BodyWt})$  and  $\log(\text{BrainWt})$  share a positive linear relationship with  $\text{TS}$ . That is, an increase in either  $\log(\text{BodyWt})$  and  $\log(\text{BrainWt})$  lead to a increase in  $\text{TS}$ . However, it's important to mention that these correlation coefficients do not take into account other possible factors that might influence their relationship with  $\text{TS}$ . This is a very big difference between studying the pairwise plots versus the linear regression models. The pairwise plots suggest a very positive correlation between  $\log(\text{BodyWt})$  and  $\text{TS}$  (and  $\log(\text{BrainWt})$  and  $\text{TS}$ ), but the main effects model takes into consideration other effects and suggests that  $\log(\text{BodyWt})$  and  $\log(\text{BrainWt})$  might actually have a negative relationship with  $\text{TS}$ , not a positive one.

## Model Search Methods

Time to analyze a number of different regression models using the main effects for  $\log(\text{BodyWt})$ ,  $\log(\text{BrainWt})$ ,  $\log(\text{Life})$ ,  $\log(\text{GP})$  and the variable  $D$ .

I used both the AIC and BIC criterion to determine the best model. I wanted to see if the measures would suggest the same model. AIC and BIC both quantify the balance between model fit and model complexity. They prefer models that fit data values to a model surface adequately while also not causing overfitting, which can happen if there is extreme model complexity. AIC/BIC returns a value for each model they analyze, and the model with the smallest value of AIC/BIC is preferred.

I wanted to use both AIC and BIC since they do have differences. AIC prefers slightly more complex models than BIC. If both the criterions suggest the same model, then I can be more assured I have a good pick.

Analyzing countless models can be difficult without a helpful algorithm. I used three guided search algorithms: forward selection, backward elimination, and stepwise regression. Forward selection starts by conducting the model criterion A(B)IC for the "null" model (just the intercept). It progresses to add variables and conduct the model criterion. If A(B)IC is larger for a model with an added regressor, it returns to the previous model. Backward elimination starts by conducting the model criterion A(B)IC for the "full" model (all regressors and intercept). It progresses to delete variables and conduct the model criterion. If A(B)IC is larger for a model with a deleted regressor, it returns to the previous model. Stepwise regression takes traits from both forward selection and backward elimination. It starts with any model, typically the null or full model. From there, it considers all possible models by adding or deleting a predictor. If the A(B)IC is larger for all these possible models, it returns to the current one and stops. If A(B)IC is smaller for one of these models, the algorithm updates the current best model with the new one and repeats the process.

Forward selection, backward elimination, and stepwise regression settle on a model where the value for A(B)IC is at its smallest and no other model can provide a smaller value. Similar to my choice to use both AIC and BIC, I chose to conduct all three of these methods so that I could see if the algorithms suggest the same model. No one single algorithm is objectively correct, but multiple algorithms could point to a similar answer!

```
null = lm(TS ~ 1, data = sleepdataOriginal)
full = lm(TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D, data = sleepdataOriginal)
n = dim(sleepdataOriginal)[1]
```

Forward Selection by AIC

```
stepAIC(null, scope = list(upper = full), direction = "forward", k = 2)
```

```
## Start:  AIC=158.39
## TS ~ 1
##
##           Df Sum of Sq  RSS   AIC
## + log(GP)      1    476.84 618.03 131.23
## + log(BrainWt)  1    416.10 678.77 136.01
## + log(BodyWt)   1    402.97 691.90 136.99
## + D             4    474.16 620.71 137.45
## + log(Life)     1    193.38 901.49 150.48
## <none>                 1094.87 158.40
##
## Step:  AIC=131.23
## TS ~ log(GP)
##
##           Df Sum of Sq  RSS   AIC
## + D             4   268.744 349.28 110.13
## + log(BodyWt)   1    35.151 582.88 130.24
## + log(BrainWt)  1    25.721 592.30 131.06
## <none>                 618.03 131.23
## + log(Life)     1     0.941 617.09 133.15
##
## Step:  AIC=110.13
## TS ~ log(GP) + D
##
##           Df Sum of Sq  RSS   AIC
## + log(BrainWt)  1   30.3864 318.90 107.48
## + log(BodyWt)   1   24.8139 324.47 108.37
## <none>                 349.28 110.13
## + log(Life)     1    2.0139 347.27 111.83
##
## Step:  AIC=107.48
## TS ~ log(GP) + D + log(BrainWt)
##
##           Df Sum of Sq  RSS   AIC
## <none>                 318.90 107.48
## + log(Life)     1    4.0315 314.86 108.84
## + log(BodyWt)   1    0.0015 318.89 109.48
```

```
##
## Call:
## lm(formula = TS ~ log(GP) + D + log(BrainWt), data = sleepdataOriginal)
##
## Coefficients:
## (Intercept)      log(GP)          DD2          DD3          DD4
##      21.2179      -1.4219      -2.1651      -4.7433      -3.2938
##          DD5 log(BrainWt)
##      -7.1699      -0.5319
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{BrainWt}_i) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Backward Elimination by AIC

```
stepAIC(full, direction = "backward", k = 2)
```

```
## Start:  AIC=110.78
## TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(BodyWt)   1     0.343 314.86 108.84
## - log(Life)     1     4.373 318.89 109.48
## - log(BrainWt)  1     9.436 323.96 110.29
## <none>                                314.52 110.78
## - log(GP)       1    38.958 353.48 114.74
## - D             4   248.005 562.53 132.43
##
## Step:  AIC=108.84
## TS ~ log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(Life)     1     4.032 318.90 107.48
## <none>                                314.86 108.84
## - log(BrainWt)  1    32.404 347.27 111.83
## - log(GP)       1    40.503 355.37 113.01
## - D             4   251.808 566.67 130.81
##
## Step:  AIC=107.48
## TS ~ log(BrainWt) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## <none>                                318.90 107.48
## - log(BrainWt)  1    30.386 349.28 110.13
## - log(GP)       1    37.615 356.51 111.17
## - D             4   273.409 592.30 131.06
```

```
##
## Call:
## lm(formula = TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
##
## Coefficients:
## (Intercept)  log(BrainWt)    log(GP)         DD2         DD3
##      21.2179      -0.5319      -1.4219      -2.1651      -4.7433
##          DD4          DD5
##      -3.2938      -7.1699
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{BrainWt}_i) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Stepwise Regression by AIC

```
stepAIC(full, direction = "both", k = 2)
```

```
## Start:  AIC=110.78
## TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(BodyWt)   1     0.343 314.86 108.84
## - log(Life)     1     4.373 318.89 109.48
## - log(BrainWt)  1     9.436 323.96 110.29
## <none>                                314.52 110.78
## - log(GP)       1    38.958 353.48 114.74
## - D             4   248.005 562.53 132.43
##
## Step:  AIC=108.84
## TS ~ log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(Life)     1     4.032 318.90 107.48
## <none>                                314.86 108.84
## + log(BodyWt)   1     0.343 314.52 110.78
## - log(BrainWt)  1    32.404 347.27 111.83
## - log(GP)       1    40.503 355.37 113.01
## - D             4   251.808 566.67 130.81
##
## Step:  AIC=107.48
## TS ~ log(BrainWt) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## <none>                                318.90 107.48
## + log(Life)     1     4.032 314.86 108.84
## + log(BodyWt)   1     0.001 318.89 109.48
## - log(BrainWt)  1    30.386 349.28 110.13
## - log(GP)       1    37.615 356.51 111.17
## - D             4   273.409 592.30 131.06
```

```
##
## Call:
## lm(formula = TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
##
## Coefficients:
## (Intercept) log(BrainWt)    log(GP)         DD2         DD3
##      21.2179      -0.5319      -1.4219      -2.1651      -4.7433
##           DD4         DD5
##      -3.2938      -7.1699
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{BrainWt}_i) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Forward Selection by BIC

```
stepAIC(null, scope = list(upper = full), direction = "forward", k = log(n))
```

```
## Start:  AIC=160.33
## TS ~ 1
##
##              Df Sum of Sq    RSS    AIC
## + log(GP)      1    476.84   618.03 135.09
## + log(BrainWt)  1    416.10   678.77 139.88
## + log(BodyWt)   1    402.97   691.90 140.85
## + D             4    474.16   620.71 147.11
## + log(Life)     1    193.38   901.49 154.35
## <none>          1094.87 160.33
##
## Step:  AIC=135.09
## TS ~ log(GP)
##
##              Df Sum of Sq    RSS    AIC
## + D           4    268.744 349.28 121.72
## <none>         618.03 135.09
## + log(BodyWt)  1     35.151 582.88 136.04
## + log(BrainWt) 1     25.721 592.30 136.86
## + log(Life)    1      0.941 617.09 138.95
##
## Step:  AIC=121.72
## TS ~ log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## + log(BrainWt) 1    30.3864 318.90 121.01
## <none>          349.28 121.72
## + log(BodyWt)  1    24.8139 324.47 121.89
## + log(Life)    1     2.0139 347.27 125.36
##
## Step:  AIC=121.01
## TS ~ log(GP) + D + log(BrainWt)
##
##              Df Sum of Sq    RSS    AIC
## <none>          318.90 121.01
## + log(Life)    1     4.0315 314.86 124.29
## + log(BodyWt)  1     0.0015 318.89 124.94
```

```
##
## Call:
## lm(formula = TS ~ log(GP) + D + log(BrainWt), data = sleepdataOriginal)
##
## Coefficients:
## (Intercept)      log(GP)          DD2          DD3          DD4
##      21.2179      -1.4219      -2.1651      -4.7433      -3.2938
##          DD5 log(BrainWt)
##      -7.1699      -0.5319
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{BrainWt}_i) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Backward Elimination by BIC

```
stepAIC(full, direction = "backward", k = log(n))
```

```
## Start:  AIC=128.17
## TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq  RSS   AIC
## - log(BodyWt)  1    0.343 314.86 124.29
## - log(Life)    1    4.373 318.89 124.94
## - log(BrainWt) 1    9.436 323.96 125.74
## <none>                                314.52 128.17
## - log(GP)      1   38.958 353.48 130.19
## - D            4  248.005 562.53 142.09
##
## Step:  AIC=124.29
## TS ~ log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq  RSS   AIC
## - log(Life)    1    4.032 318.90 121.01
## <none>                                314.86 124.29
## - log(BrainWt) 1   32.404 347.27 125.36
## - log(GP)      1   40.503 355.37 126.53
## - D            4  251.808 566.67 138.53
##
## Step:  AIC=121.01
## TS ~ log(BrainWt) + log(GP) + D
##
##              Df Sum of Sq  RSS   AIC
## <none>                                318.90 121.01
## - log(BrainWt) 1   30.386 349.28 121.72
## - log(GP)      1   37.615 356.51 122.76
## - D            4  273.409 592.30 136.86
```

```
##
## Call:
## lm(formula = TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
##
## Coefficients:
## (Intercept) log(BrainWt) log(GP) DD2 DD3
## 21.2179 -0.5319 -1.4219 -2.1651 -4.7433
## DD4 DD5
## -3.2938 -7.1699
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{BrainWt}_i) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Stepwise Regression by BIC

```
stepAIC(full, direction = "both", k = log(n))
```

```
## Start: AIC=128.17
## TS ~ log(BodyWt) + log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(BodyWt)  1     0.343 314.86 124.29
## - log(Life)    1     4.373 318.89 124.94
## - log(BrainWt) 1     9.436 323.96 125.74
## <none>                                314.52 128.17
## - log(GP)      1    38.958 353.48 130.19
## - D            4   248.005 562.53 142.09
##
## Step: AIC=124.29
## TS ~ log(BrainWt) + log(Life) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## - log(Life)    1     4.032 318.90 121.01
## <none>                                314.86 124.29
## - log(BrainWt) 1    32.404 347.27 125.36
## - log(GP)      1    40.503 355.37 126.53
## + log(BodyWt)  1     0.343 314.52 128.17
## - D            4   251.808 566.67 138.53
##
## Step: AIC=121.01
## TS ~ log(BrainWt) + log(GP) + D
##
##              Df Sum of Sq    RSS    AIC
## <none>                                318.90 121.01
## - log(BrainWt) 1    30.386 349.28 121.72
## - log(GP)      1    37.615 356.51 122.76
## + log(Life)    1     4.032 314.86 124.29
## + log(BodyWt)  1     0.001 318.89 124.94
## - D            4   273.409 592.30 136.86
```

```
##
## Call:
## lm(formula = TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
##
## Coefficients:
## (Intercept) log(BrainWt)    log(GP)          DD2          DD3
##      21.2179      -0.5319      -1.4219      -2.1651      -4.7433
##           DD4          DD5
##      -3.2938      -7.1699
```

Model chosen:  $y_i = \beta_0 + \beta_1 \log(\text{Brain}W_{t_i}) + \beta_2 \log(\text{GP}_i) + \beta_3 U_{D2i} + \beta_4 U_{D3i} + \beta_5 U_{D4i} + \beta_6 U_{D5i} + e_i$

Interestingly, forward selection for AIC, backward elimination for AIC, stepwise regression for AIC, forward selection for BIC, backward elimination for BIC and stepwise regression for BIC all suggest the same model.

The mean function for the suggested model:

When  $D = D1$

$$E(TS | \ln(\text{Body}W_t), \ln(\text{GP}), D = D1) = \beta_0 + \beta_1 \ln(\text{Body}W_{t_i}) + \beta_2 \ln(\text{GP}_i)$$

When  $D = D2$

$$E(TS | \ln(\text{Body}W_t), \ln(\text{GP}), D = D2) = \beta_0 + \beta_1 \ln(\text{Body}W_{t_i}) + \beta_2 \ln(\text{GP}_i) + \beta_3$$

When  $D = D3$

$$E(TS | \ln(\text{Body}W_t), \ln(\text{GP}), D = D3) = \beta_0 + \beta_1 \ln(\text{Body}W_{t_i}) + \beta_2 \ln(\text{GP}_i) + \beta_4$$

When  $D = D4$

$$E(TS | \ln(\text{Body}W_t), \ln(\text{GP}), D = D4) = \beta_0 + \beta_1 \ln(\text{Body}W_{t_i}) + \beta_2 \ln(\text{GP}_i) + \beta_5$$

When  $D = D5$

$$E(TS | \ln(\text{Body}W_t), \ln(\text{GP}), D = D5) = \beta_0 + \beta_1 \ln(\text{Body}W_{t_i}) + \beta_2 \ln(\text{GP}_i) + \beta_6$$



## Model Checking

We've now settled on using the model with main effects for  $\log(\text{BrainWt})$ ,  $\log(\text{GP})$  and the categorical variable  $D$  to predict sleep time. Let's analyze how well fitted the model is and if it meets the assumptions that linear regression models should meet.

Let's plot a residual plot, a histogram of the residuals, and a Q-Q plot to analyze:

A central assumption is that the residuals have constant variance across the whole range of fitted values. The residual plot and the green regression line suggest that this assumption is met for the fitted values between 0 and 10. However, the spread of the residuals starts to slightly vary between the fitted value of 10 and 20. Therefore, the assumption of constant variance is mostly but not entirely met.

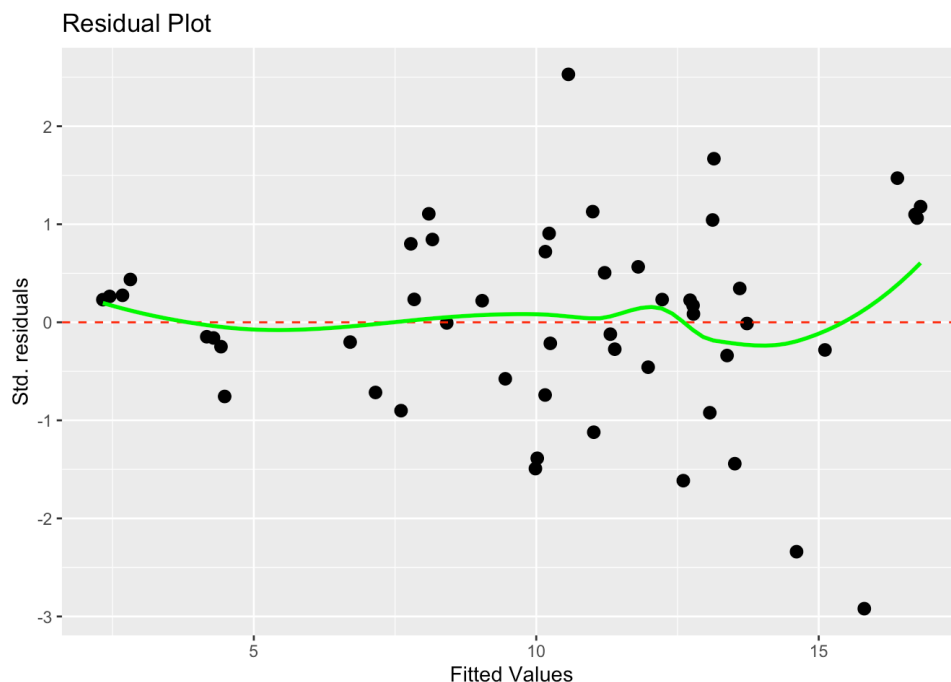
```
#Residual plot
```

```
residualInfo <- data.frame(Residuals = rstandard(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)),
  Fitted = fitted(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)))
```

```
residualplot <- ggplot(residualInfo, aes(x = Fitted, y = Residuals))+
  geom_point(shape = 16, size = 3, color = "black") +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", color = "green", se = FALSE) +
  labs(x = "Fitted Values", y = "Std. residuals") +
  ggtitle("Residual Plot")
```

```
residualplot
```

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



Another central assumption is that the residuals are centered around 0. We can see in the residual histogram that the residuals indeed are centered around 0.

```
#Hist plot
```

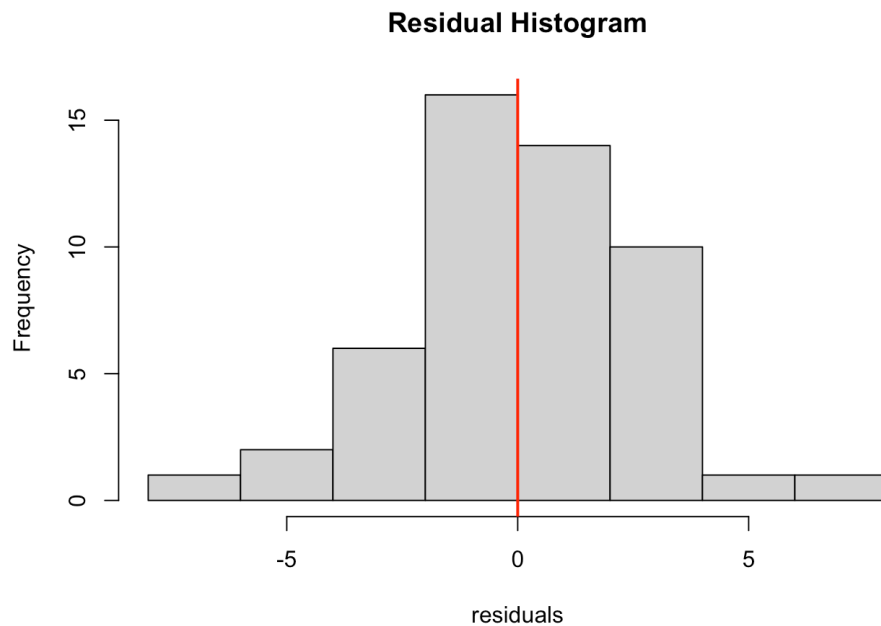
```
hist(resid(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)), xlab = "residuals", main = "Residual H  
istogram")
```

```
# Calculate the mean of the residuals
```

```
mean_resid <- mean(resid(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)))
```

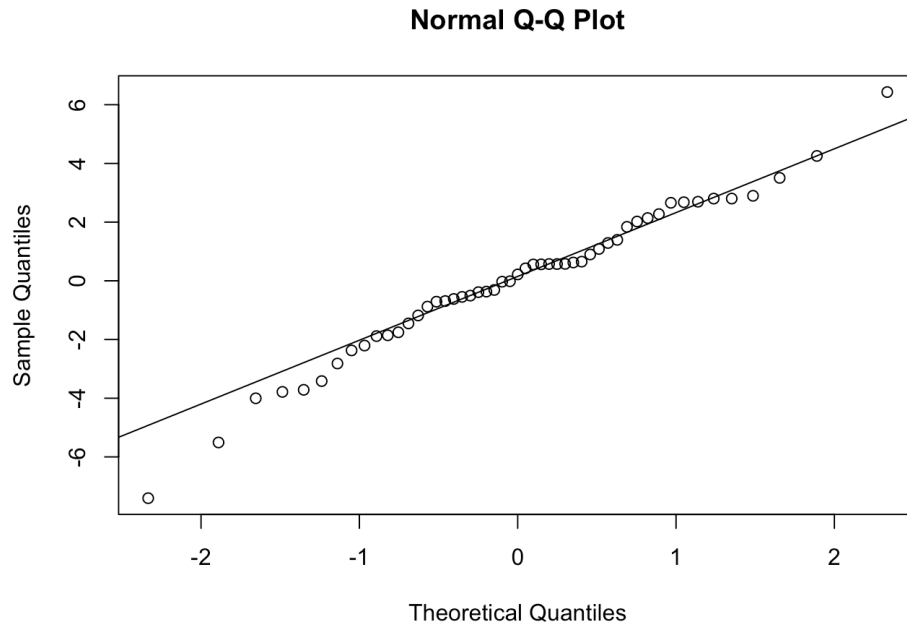
```
# Add a vertical line representing the mean
```

```
abline(v = mean_resid, col = "red", lwd = 2)
```



One final assumption we make is that the residuals follow a normal distribution. We can see in the Q-Q plot that the residuals follow close to a normal distribution, but not a perfect one. There are some outlier residuals that deviate from normality in the tail ends of the residual distribution. We can say that this assumption is partly but not entirely met.

```
qqnorm(resid(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal))); qqline(resid(lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)))
```



## Model Summarization

Let's summarize our fitted, final linear regression model

Using the estimated regression coefficients, let's describe how the predictors `BrainWt` and `GP` (on their original scales) are related to sleep time for mammals when adjusting for the other variables in the model.

The estimated regression coefficients:

```
finalModelLog = lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
sumfinalModelLog = summary(finalModelLog)
sumfinalModelLog$coefficients
```

```
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept)  21.2178793  2.2016833   9.637117 2.062947e-12
## log(BrainWt) -0.5318545  0.2597474  -2.047584 4.659935e-02
## log(GP)      -1.4218653  0.6241306  -2.278153 2.762844e-02
## DD2         -2.1650909  1.0607945  -2.041009 4.727657e-02
## DD3         -4.7432725  1.2132176  -3.909663 3.156525e-04
## DD4         -3.2938437  1.1831994  -2.783845 7.888554e-03
## DD5         -7.1698783  1.2766087  -5.616348 1.232254e-06
```

The coefficients reveal interesting insights on the effects of `BrainWt` and `GP` on `TS`. When adjusting for other variables in the model, a  $x * 100$  % increase in `BrainWt` leads to an expected  $-0.5319 * x$  change in sleep time for mammals. Likewise, when adjusting for other variables in the model, a  $x * 100$  % increase in `GP` leads to an expected  $-1.4219 * x$  change in sleep time for mammals.

Let's also describe how increases in the danger level index for a mammal are related to sleep time:

If a mammal has danger level index D2, then this leads to an expected decrease in total sleep by 2.165 hours compared to a mammal at level D1.  
 If a mammal has danger level index D3, then this leads to an expected decrease in total sleep by 4.743 hours compared to a mammal at level D1.  
 If a mammal has danger level index D4, then this leads to an expected decrease in total sleep by 3.294 hours compared to a mammal at level D1.  
 If a mammal has danger level index D5, then this leads to an expected decrease in total sleep by 7.17 hours compared to a mammal at level D1.

Now, let's make two effects plots: one for the predictor `BrainWt` (on its original scale) and one for the predictor `GP` (on its original scale). Each plot displays five separate regression functions (one for each level of `D`).

```
finalModelLog = lm(TS ~ log(BrainWt) + log(GP) + D, data = sleepdataOriginal)
finalModelOG = lm(TS ~ BrainWt + GP + D, data = sleepdataOriginal)
```

BrainWt effect plot:

```

#plot for `BrainWt`
gsize = 1000

newDatDD1 = data.frame(BrainWt = seq(from = min(sleepdataOriginal$BrainWt),
to = max(sleepdataOriginal$BrainWt), length.out = gsize),
GP = rep(mean(sleepdataOriginal$GP), gsize),
D = rep("D1",gsize))

newDatDD2 = data.frame(BrainWt = seq(from = min(sleepdataOriginal$BrainWt),
to = max(sleepdataOriginal$BrainWt), length.out = gsize),
GP = rep(mean(sleepdataOriginal$GP), gsize),
D = rep("D2",gsize))

newDatDD3 = data.frame(BrainWt = seq(from = min(sleepdataOriginal$BrainWt),
to = max(sleepdataOriginal$BrainWt), length.out = gsize),
GP = rep(mean(sleepdataOriginal$GP), gsize),
D = rep("D3",gsize))

newDatDD4 = data.frame(BrainWt = seq(from = min(sleepdataOriginal$BrainWt),
to = max(sleepdataOriginal$BrainWt), length.out = gsize),
GP = rep(mean(sleepdataOriginal$GP), gsize),
D = rep("D4",gsize))

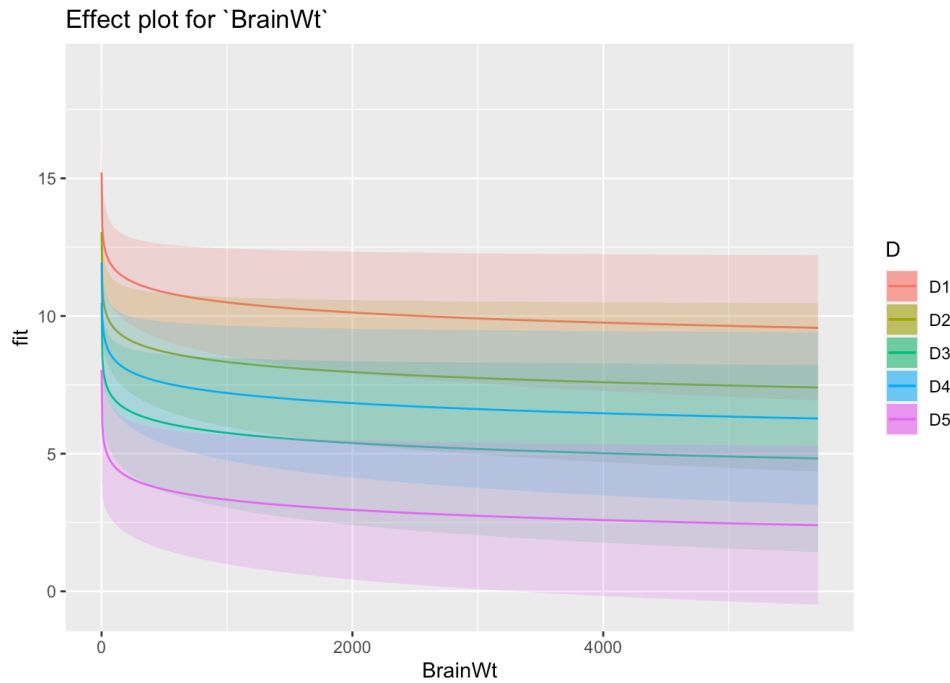
newDatDD5 = data.frame(BrainWt = seq(from = min(sleepdataOriginal$BrainWt),
to = max(sleepdataOriginal$BrainWt), length.out = gsize),
GP = rep(mean(sleepdataOriginal$GP), gsize),
D = rep("D5",gsize))

eplotinfoDD1 = predict(finalModelLog, newdata = newDatDD1, interval = "confidence")
eplotinfoDD2 = predict(finalModelLog, newdata = newDatDD2, interval = "confidence")
eplotinfoDD3 = predict(finalModelLog, newdata = newDatDD3, interval = "confidence")
eplotinfoDD4 = predict(finalModelLog, newdata = newDatDD4, interval = "confidence")
eplotinfoDD5 = predict(finalModelLog, newdata = newDatDD5, interval = "confidence")

# Combine the new dataframes
newDatCombined = rbind(newDatDD1, newDatDD2, newDatDD3, newDatDD4, newDatDD5)
eplotinfoCombined = rbind(eplotinfoDD1, eplotinfoDD2, eplotinfoDD3, eplotinfoDD4, eplotinfoDD5)

BrainWtGraph = bind_cols(newDatCombined, eplotinfoCombined)
ggplot(BrainWtGraph, aes(x = BrainWt, y = fit)) +
  geom_line(data = filter(BrainWtGraph, D %in% "D1"),aes(y = fit, color = D)) +
  geom_ribbon(data = filter(BrainWtGraph, D %in% "D1"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(BrainWtGraph, D %in% "D2"),aes(y = fit, color = D)) +
  geom_ribbon(data = filter(BrainWtGraph, D %in% "D2"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(BrainWtGraph, D %in% "D3"),aes(y = fit, color = D)) +
  geom_ribbon(data = filter(BrainWtGraph, D %in% "D3"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(BrainWtGraph, D %in% "D4"),aes(y = fit, color = D)) +
  geom_ribbon(data = filter(BrainWtGraph, D %in% "D4"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(BrainWtGraph, D %in% "D5"),aes(y = fit, color = D)) +
  geom_ribbon(data = filter(BrainWtGraph, D %in% "D5"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  ggtitle("Effect plot for `BrainWt`")

```



The effects plot for `BrainWt` reveals interesting information. The plot treats the mean function as a function of `TS` for a fixed value of `GP`. Here, the fixed value of `GP` is its sample average value, or the “typical” value. Therefore, the plot represents the adjusted effect of `BrainWt` when the other regressor is fixed at its “typical” values and the danger level, `D`, is given. We can see that as `BrainWt` increases and `GP` is held fixed at its typical value, the expected value of `TS` exponentially decreases for any level of `D`. The rate of decrease in `TS` is very high for changes in `BrainWt` across its lower domain of values, while the rate of decrease in `TS` is low for changes in `BrainWt` across its higher domain of values. The plot also reveals that mammals with lower danger index levels inherently have higher expected `TS` levels when `BrainWt` is 0 and `GP` is held at its fixed typical value.

GP effect plot:

```

#plot for `GP`
gsize = 1000

newDatDD1 = data.frame(GP = seq(from = min(sleepdataOriginal$GP),
                                to = max(sleepdataOriginal$GP), length.out = gsize),
                        BrainWt = rep(mean(sleepdataOriginal$BrainWt), gsize),
                        D = rep("D1", gsize))

newDatDD2 = data.frame(GP = seq(from = min(sleepdataOriginal$GP),
                                to = max(sleepdataOriginal$GP), length.out = gsize),
                        BrainWt = rep(mean(sleepdataOriginal$BrainWt), gsize),
                        D = rep("D2", gsize))

newDatDD3 = data.frame(GP = seq(from = min(sleepdataOriginal$GP),
                                to = max(sleepdataOriginal$GP), length.out = gsize),
                        BrainWt = rep(mean(sleepdataOriginal$BrainWt), gsize),
                        D = rep("D3", gsize))

newDatDD4 = data.frame(GP = seq(from = min(sleepdataOriginal$GP),
                                to = max(sleepdataOriginal$GP), length.out = gsize),
                        BrainWt = rep(mean(sleepdataOriginal$BrainWt), gsize),
                        D = rep("D4", gsize))

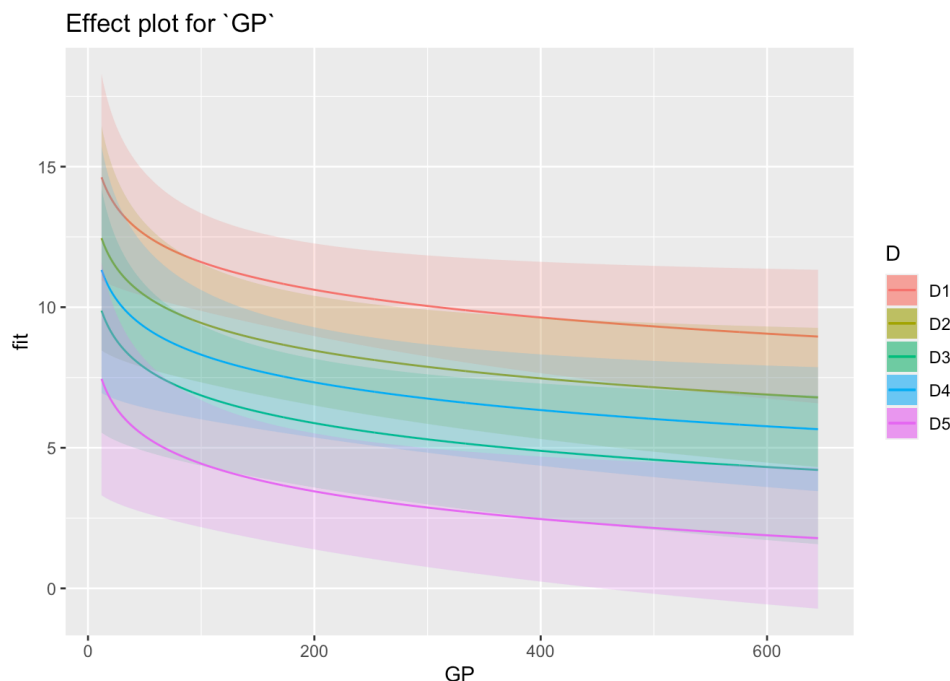
newDatDD5 = data.frame(GP = seq(from = min(sleepdataOriginal$GP),
                                to = max(sleepdataOriginal$GP), length.out = gsize),
                        BrainWt = rep(mean(sleepdataOriginal$BrainWt), gsize),
                        D = rep("D5", gsize))

eplotinfoDD1 = predict(finalModelLog, newdata = newDatDD1, interval = "confidence")
eplotinfoDD2 = predict(finalModelLog, newdata = newDatDD2, interval = "confidence")
eplotinfoDD3 = predict(finalModelLog, newdata = newDatDD3, interval = "confidence")
eplotinfoDD4 = predict(finalModelLog, newdata = newDatDD4, interval = "confidence")
eplotinfoDD5 = predict(finalModelLog, newdata = newDatDD5, interval = "confidence")

# Combine the new dataframes
newDatCombined = rbind(newDatDD1, newDatDD2, newDatDD3, newDatDD4, newDatDD5)
eplotinfoCombined = rbind(eplotinfoDD1, eplotinfoDD2, eplotinfoDD3, eplotinfoDD4, eplotinfoDD5)

GPGraph = bind_cols(newDatCombined, eplotinfoCombined)
ggplot(GPGraph, aes(x = GP, y = fit)) +
  geom_line(data = filter(GPGraph, D %in% "D1"), aes(y = fit, color = D)) +
  geom_ribbon(data = filter(GPGraph, D %in% "D1"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(GPGraph, D %in% "D2"), aes(y = fit, color = D)) +
  geom_ribbon(data = filter(GPGraph, D %in% "D2"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(GPGraph, D %in% "D3"), aes(y = fit, color = D)) +
  geom_ribbon(data = filter(GPGraph, D %in% "D3"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(GPGraph, D %in% "D4"), aes(y = fit, color = D)) +
  geom_ribbon(data = filter(GPGraph, D %in% "D4"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  geom_line(data = filter(GPGraph, D %in% "D5"), aes(y = fit, color = D)) +
  geom_ribbon(data = filter(GPGraph, D %in% "D5"), aes(ymin = lwr, ymax = upr, fill = D), alpha = 0.2) +
  ggtitle("Effect plot for `GP`")

```



The effects plot for `GP` reveals interesting information as well. The plot treats the mean function as a function of `TS` for a fixed value of `BrainWt`. Here, the fixed value of `BrainWt` is its sample average value, or the “typical” value. Therefore, the plot represents the adjusted effect of `GP` when the other regressor is fixed at its “typical” values and the danger level, `D`, is given. We can see that as `GP` increases and `BrainWt` is held fixed at its typical value, the expected value of `TS` exponentially decreases for any level of `D`. The rate of decrease in `TS` is very high for changes in `GP` across its lower domain of values, while the rate of decrease in `TS` is low for changes in `GP` across its higher domain of values. The plot also reveals that mammals with lower danger index levels inherently have lower expected `TS` levels when `GP` is 0 and `BrainWt` is held at its fixed typical value.

## A Brief Continuation: Ridge Regression Model

Lastly, we will create a Ridge Regression model using the final fitted model we just settled with. It is very similar to the traditional linear regression method, but with a slight difference. It corrects for overfitting on the training data by targeting multicollinearity. As we observed in the pairplots, there is high positive and negative correlation across the factors. This correlation can create multicollinearity, and it is a good idea to regularize our model to take this into account. What results is a model with less variance, meaning that unnecessary noise and information is filtered out.

Ridge Regression modifies ordinary least squares by adding a regularization term to the equation that determines coefficients  $\beta$ . The traditional equation for linear regression:  $\hat{\beta}_R = (X^T X)^{-1} X^T y$ . The new coefficients are given by:  $\hat{\beta}_R = (X^T X + \lambda I)^{-1} X^T y$

This penalty term,  $\lambda I$ , counteracts high  $\beta$  coefficients. This can be called coefficient shrinkage. The penalty term is added to the diagonals in the  $X^T X$  matrix, increasing the determinant away from 0. If the determinant of  $X^T X$  is close to 0, it is considered near-singular. This means that the  $\beta$  coefficients estimates from the matrix are highly inaccurate. By moving the determinant away from 0, the penalty term shrinks the predictor’s effect on the predicted value. This helps counteract multicollinearity and overfitting of the data.

Through the addition of the penalty term, higher  $\beta$  coefficients are reduced greater than smaller coefficients. The coefficients are brought closer to 0, reducing their overall effect on the outcome. A consequence of this is that the coefficients are not easy for us to interpret. Our model might more accurately predict the total sleep types of mammals get thanks to lower variance, but the results are not as easy to interpret.

The following code creates 2 ridge regression models of importance to us:

```
X <- model.matrix(finalModelLog)
y <- model.response(model.frame(finalModelLog))

ridge_model <- cv.glmnet(X, y, alpha = 0)
ridge_model
```

```
##  
## Call:  cv.glmnet(x = X, y = y, alpha = 0)  
##  
## Measure: Mean-Squared Error  
##  
##      Lambda Index Measure      SE Nonzero  
## min 0.3058   100   7.843 1.367         6  
## 1se 2.5983    77   9.164 1.481         6
```

The first model, indicated by “min”, represents the model that gives the lowest mean square error, indicated by “measure”. \ The second model, indicated by “1se”, represents the model that is slightly more regularized, indicated by “lambda”, but still within one standard error of the minimum MSE.

Now, we have three models! One least squares regression model, and two models that utilize ridge regression.