STAT 213
Transformations

Colin Reimer Dawson

Oberlin College

June 21, 2021

What to Do If Conditions are Violated?

What if we have...

- Lack of normality of residuals
- Patterns (e.g. curvature) in residuals
- Non-constant variance ("heteroskedasticity")
- Outliers: influential points, large residuals

Transformations and Outliers

Data Transformations

Can (sometimes) be used to

- "Unskew" residual distribution
- "Unbend" non-linear relationships
- Stabilize (equalize) variance of residuals
- Reduce influence of outliers

Example: Year Length on Different Planets

Cases: Planets in our solar system

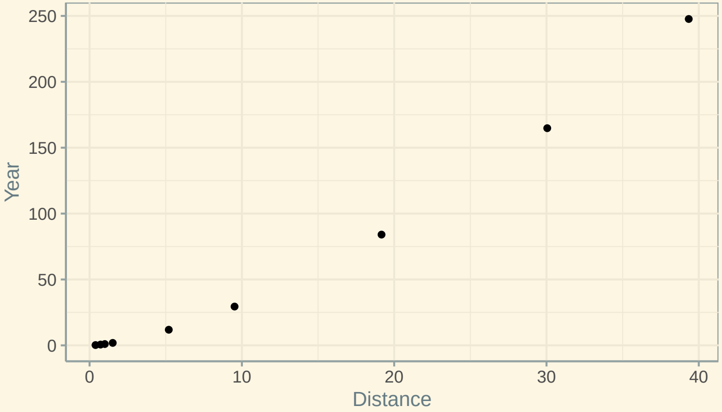
Y : `Length` (days) of a year on each planet

X : `Distance` (km) from the sun

Can we model `Length` as a function of `Distance`?

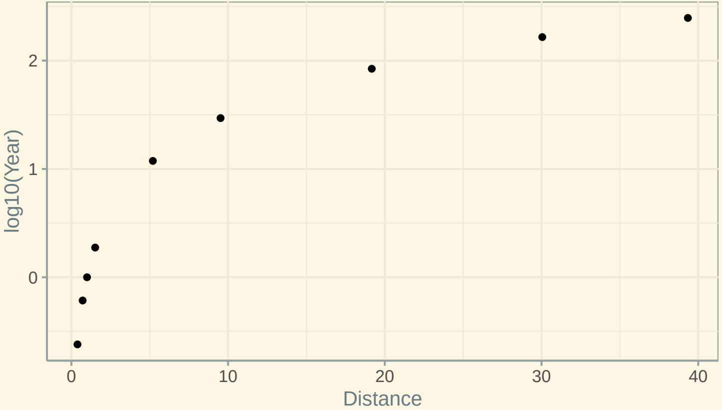
Example: Year Length on Different Planets

```
library(mosaic)
## Note: syntax to read data from a file on the web
Planets <- read.file("http://colindawson.net/data/Planets.csv")
gf_point(Year ~ Distance, data = Planets) # not linear
```



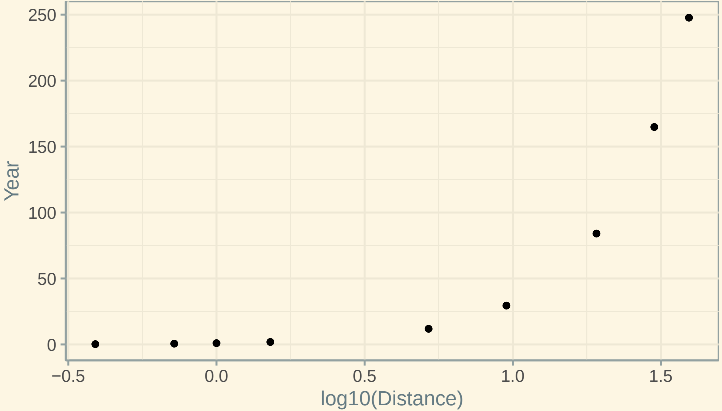
Transforming Y

```
gf_point(log10(Year) ~ Distance, data = Planets) ## overcorrected
```



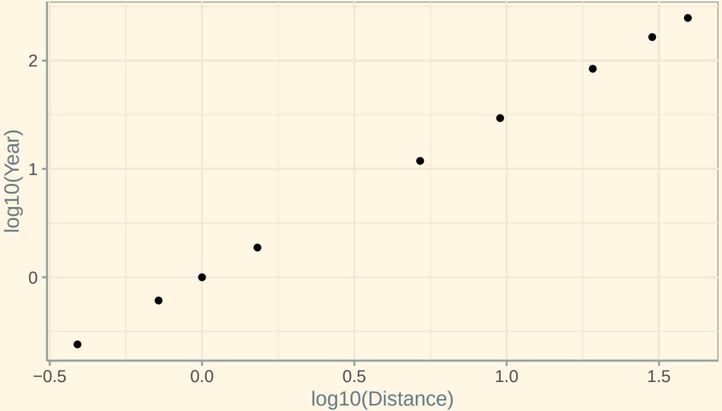
Transforming X

```
gf_point(Year ~ log10(Distance), data = Planets) ## wrong direction
```



Transforming X and Y

```
gf_point(log10(Year) ~ log10(Distance), data = Planets) ## linear!
```



Interpreting the Transformed Relationship

```
LogLogModel <- lm(log10(Year) ~ log10(Distance), data = Planets)
coefficients(LogLogModel)


        (Intercept) log10(Distance)
       -0.001491341     1.502061101
```

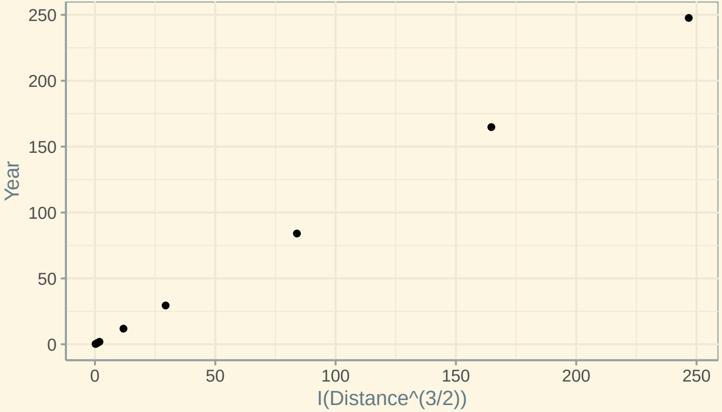
- "For each one unit increase in $\log_{10}$(`Distance`), the log of the `Year` length increases by 1.5 units"
- More understandably: "Each time distance is multiplied by 10^1, year length is multiplied by $10^{1.5}$"
- In this case, $\hat{\beta}_0 \approx 0$, so

$$\widehat{\log_{10}(\text{Year})} \approx 1.5 \cdot \log_{10}(\text{Distance})$$

$$\text{Year} \approx \text{Distance}^{3/2}$$

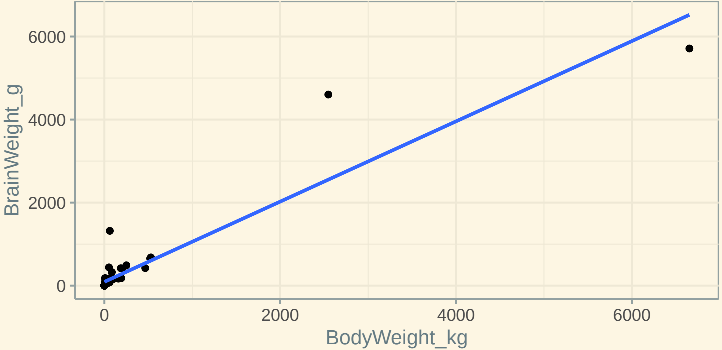
Year Length and Distance

```
gf_point(Year ~ I(Distance^(3/2)), data = Planets)
```



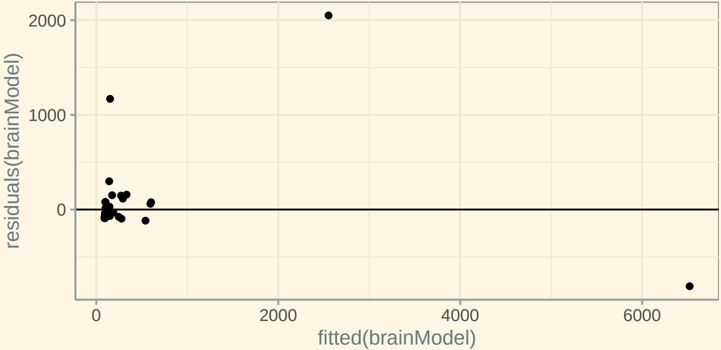
Brain and Body Weight of Terrestrial Mammals

```r
library(mosaic)
BrainBodyWeight <- read.file("http://colindawson.net/data/BrainBodyWeight.csv")
gf_point(BrainWeight_g ~ BodyWeight_kg, data = BrainBodyWeight) %>%
    gf_smooth(method = "lm")
```



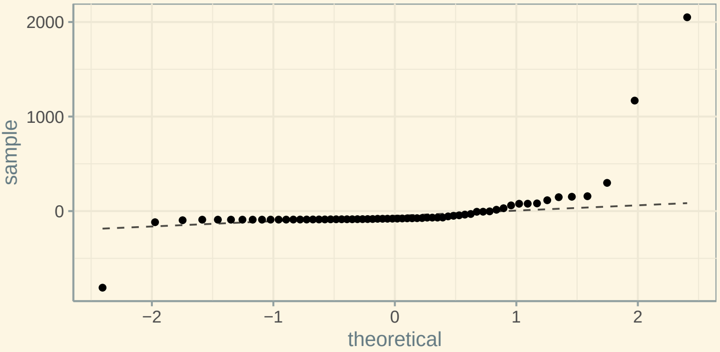
Brain and Body Weight of Terrestrial Mammals

```
brainModel <- lm(BrainWeight_g ~ BodyWeight_kg, data = BrainBodyWeight)
gf_point(residuals(brainModel) ~ fitted(brainModel)) %>%
    gf_hline(yintercept = ~0)
```



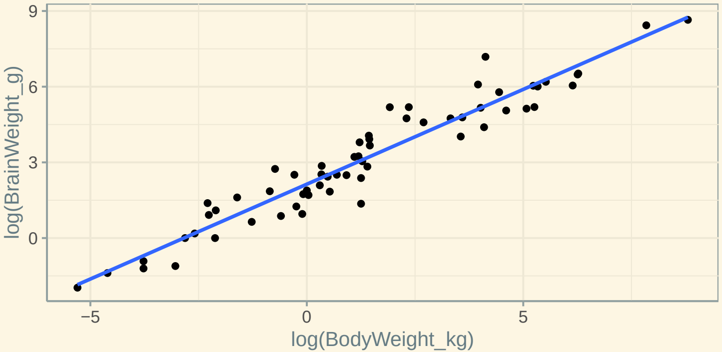
Brain and Body Weight of Terrestrial Mammals

```
gf_qq(~residuals(brainModel)) %>%
    gf_qqline()
```



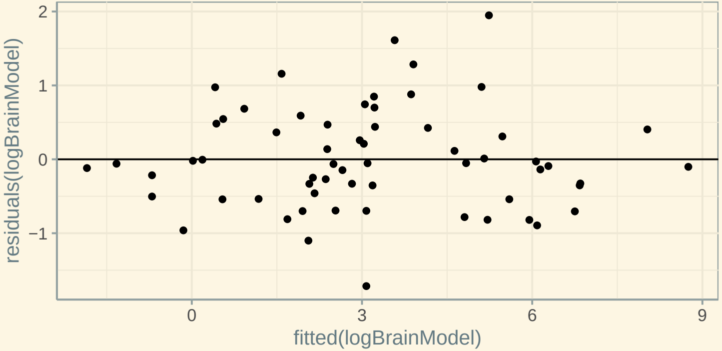
Log Brain and Log Body Weight

```
gf_point(log(BrainWeight_g) ~ log(BodyWeight_kg), data = BrainBodyWeight) %>%
    gf_smooth(method = "lm")
```



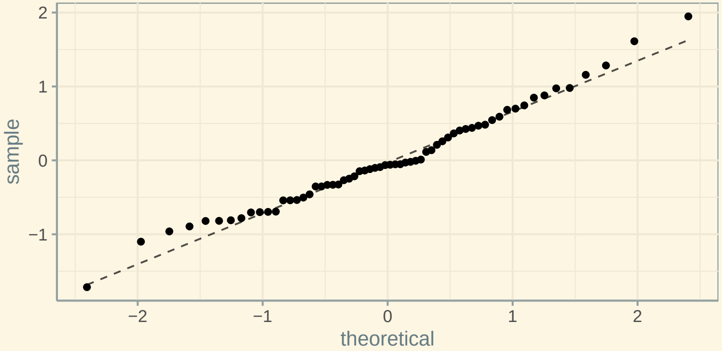
Log Brain and Log Body Weight

```
logBrainModel <-
    lm(log(BrainWeight_g) ~ log(BodyWeight_kg), data = BrainBodyWeight)
## residuals by fitted
gf_point(residuals(logBrainModel) ~ fitted(logBrainModel)) %>%
    gf_hline(yintercept = ~0)
```



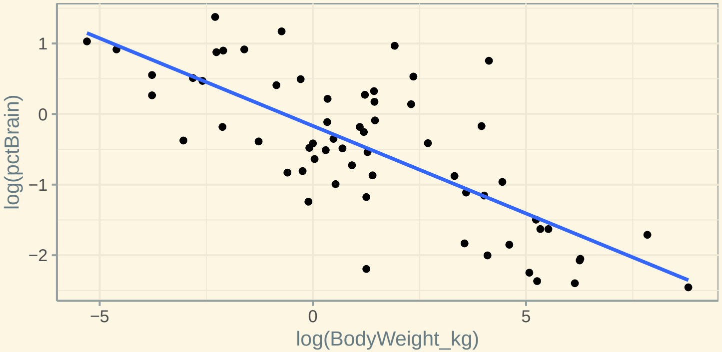
Log Brain and Log Body Weight

```
## QQ Plot
gf_qq(~residuals(logBrainModel)) %>%
    gf_qqline()
```

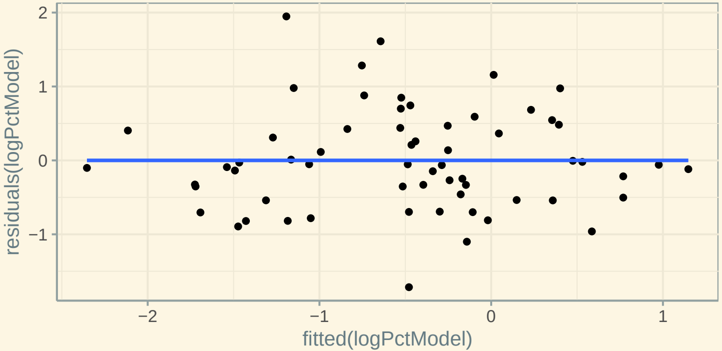


Percent Brain Weight by Body Weight

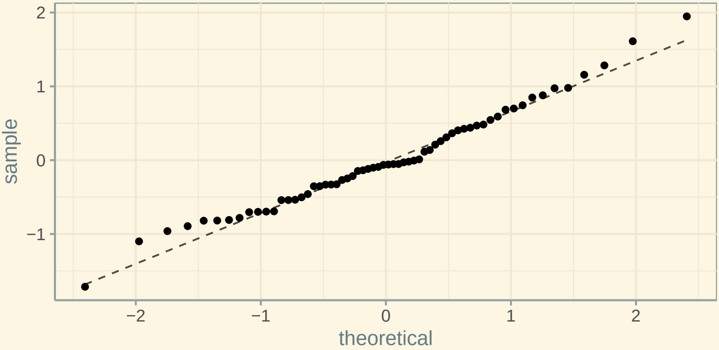
```
library(mosaic)
## Making a new variable out of old ones
BrainBodyWeight_new <- mutate(
    BrainBodyWeight,
    pctBrain = 100 * (BrainWeight_g / (BodyWeight_kg * 1000)))
gf_point(log(pctBrain) ~ log(BodyWeight_kg), data = BrainBodyWeight_new) %>%
    gf_smooth(method = "lm")
```



Percent Brain Weight By Body Weight



Percent Brain Weight By Body Weight



Key Points: Transformations

- Transformations can be used to address **skewed residuals**, **nonlinearity**, **nonconstant variance**
- Best if the transformation is motivated by **knowledge of the context**
- Typically use concave transformations ($\log$, sqrt) with right-skewed variables
- Less common, but sometimes use convex transformations ($\exp$, powers) with left-skewed variables
- Log turns multiplicative (proportional) change into additive change (one unit difference in log scale corresponds to a constant *ratio* in the original scale)