

# STAT 213

## Transformations and Influential Points

Colin Reimer Dawson

Oberlin College

February 12, 2018

# Outline

Transformations

Influence and Outliers

## The Simple Linear Model

$$Y = \beta_0 + \beta_1 \cdot X + \varepsilon$$

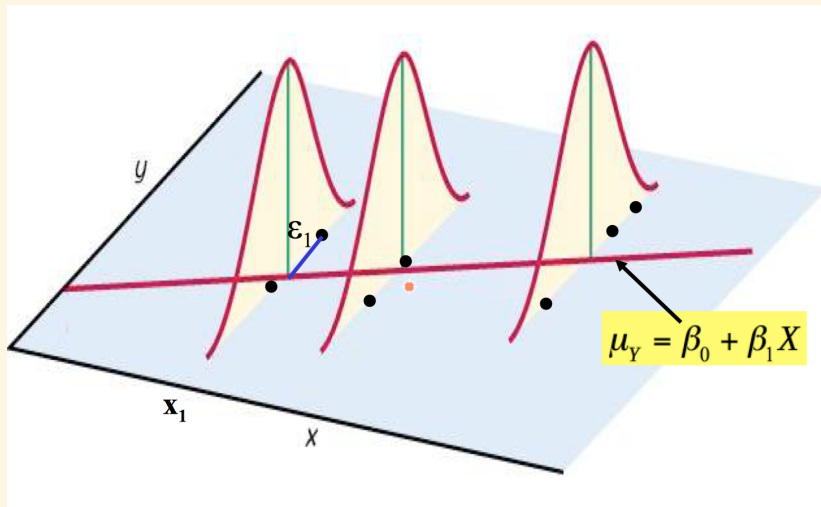
aka

Response = Intercept + Slope · Predictor + Random Error

Standard form: Assume the  $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$  and are independent

Parameters to estimate:  $\beta_0$ ,  $\beta_1$  and  $\sigma_\varepsilon$

## SLM Visualized



## Conditions for SLM

### Pattern

1. Mean  $Y$  at each  $X$  is a linear function of  $X$ :

$$\mu_Y(X) = f(X) = \beta_0 + \beta_1 X$$

### Residuals

2. Zero mean: Residuals centered at 0
3. Constant variance: Same variability at all  $X$  (Homoskedasticity)
4. Independence: No relationship among errors
5. Normality (for standard form): At each  $X$ ,  $Y$ s are Normally distributed

## QQ Plots De-mystified

- Quantile-Quantile plots and Normal probability plots are tools to assess extent to which residuals are normally distributed
- Each dot is (the residual for) one data point
- Unfortunately not much standardization across software in exactly what each axis represents
- Common thread: look at each part (e.g., percentile) of the residual distribution, and compare reality to expectation
- Play with the demo applet and recognize common shapes

# Demo: Checking Conditions

Demo

# 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 (“heteroscedasticity”)
- Outliers: influential points, large residuals



# Transformations and Outliers

## Data Transformations

Can be used to

- “Unskew” residual distribution
- Address non-linearity
- Stabilize (homogenize) variance
- Reduce influence of outliers

# Example: Year Length on Different Planets

Cases: Planets in our solar system

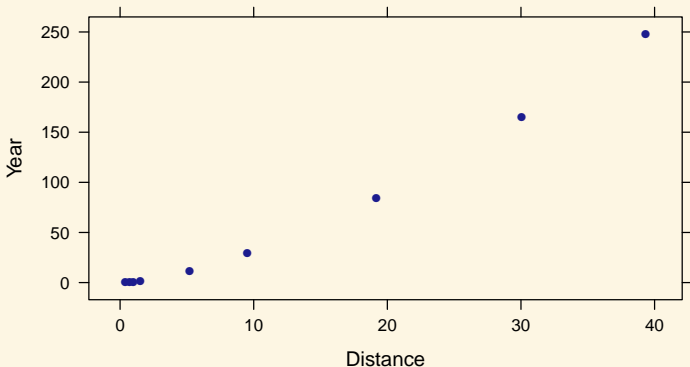
$Y$  : Length (days) of a year on each planet

$X$  : Distance (km) from the sun

Can we model  $Y$  as a function of  $X$ ?

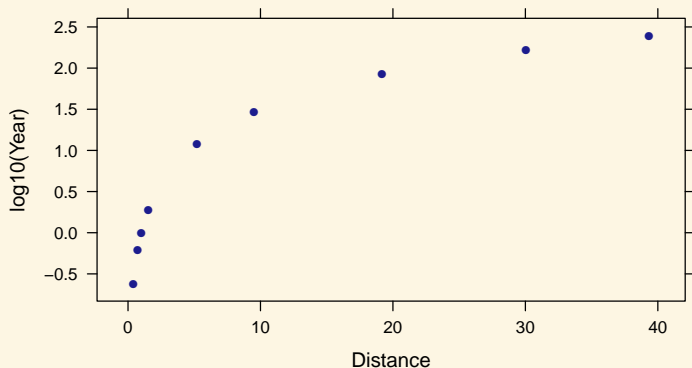
# Example: Year Length on Different Planets

```
library("mosaic")  
Planets <- read.file("http://colindawson.net/data/Planets.csv")  
xyplot(Year ~ Distance, data = Planets) # not linear
```



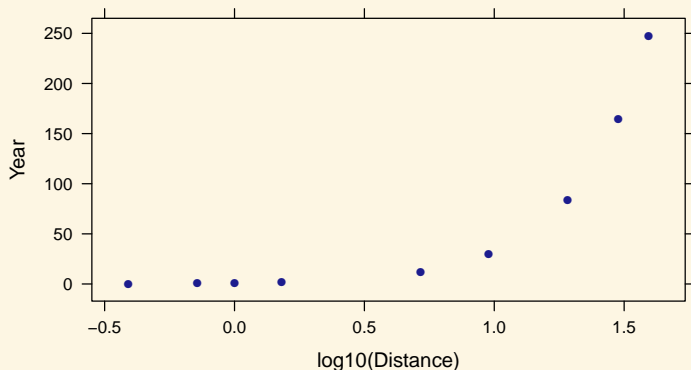
# Transforming $X$ and $Y$

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



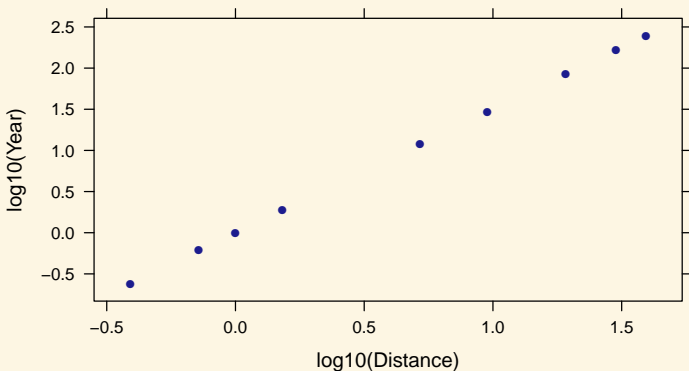
# Transforming $X$ and $Y$

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



# Transforming $X$ and $Y$

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



# Interpreting the Transformed Relationship

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

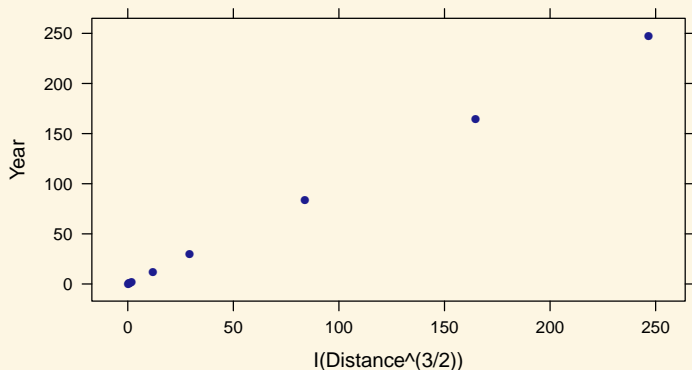
```
(Intercept) log10(Distance)
-0.001491341  1.502061101
```

- “For each one unit increase in  $\log_{10}(Distance)$ , the log 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

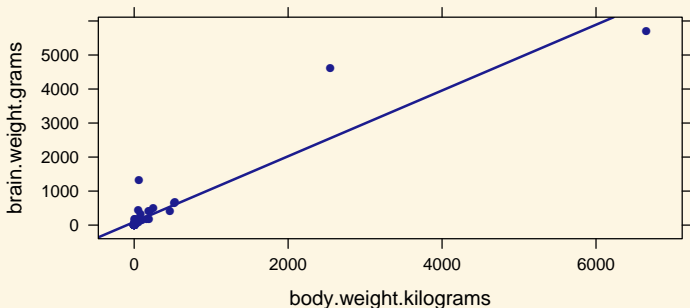
```
xyplot(Year ~ I(Distance^(3/2)), data = Planets)
```





# Brain and Body Weight of Terrestrial Mammals

```
library("mosaic")
BrainBodyWeight <- read.file("http://colindawson.net/data/BrainBodyWeight.csv")
xyplot(
  brain.weight.grams ~ body.weight.kilograms,
  data = BrainBodyWeight, type = c("p", "r"))
```

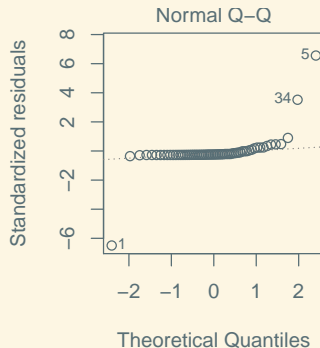
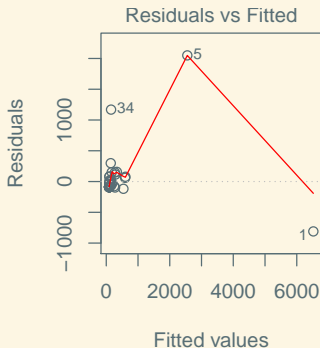


## Aside: R Code in Slides

- I often provide the R code used to get the results in a slide
- This is mainly for your reference later, not intended for you to follow the details during class

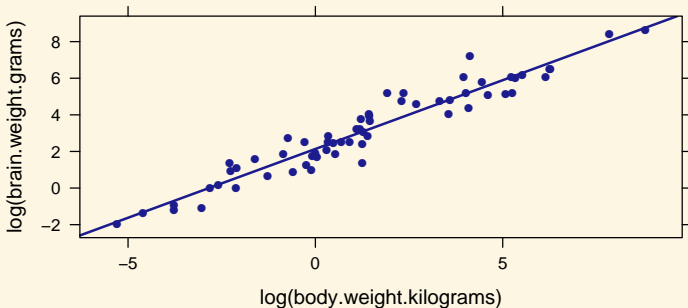
# Brain and Body Weight of Terrestrial Mammals

```
brain.model <-
  lm(brain.weight.grams ~ body.weight.kilograms, data = BrainBodyWeight)
par(mfrow = c(1,2)) # to create a 1-by-2 plotting grid
plot(brain.model, which = 1) #residuals by predicted
plot(brain.model, which = 2) #quantile-quantile
```



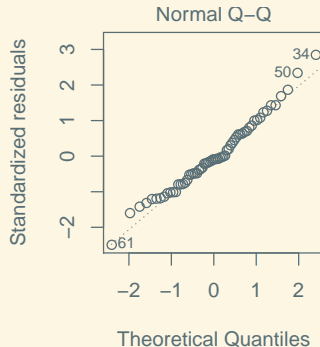
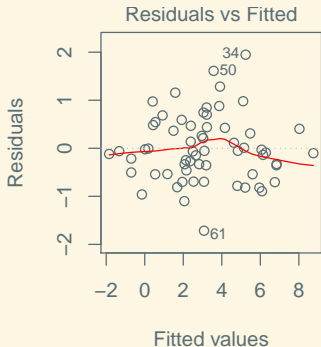
# Log Brain and Log Body Weight

```
xyplot(  
  log(brain.weight.grams) ~ log(body.weight.kilograms),  
  data = BrainBodyWeight, type = c("p", "r"))
```



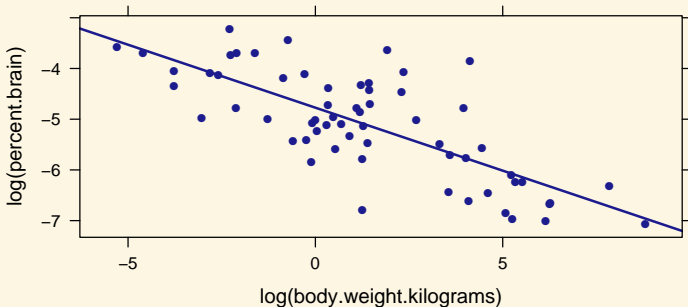
# Log Brain and Log Body Weight

```
log.brain.model <-  
  lm(log(brain.weight.grams) ~ log(body.weight.kilograms),  
     data = BrainBodyWeight)  
par(mfrow = c(1,2))  
plot(log.brain.model, which = 1) #residuals by predicted  
plot(log.brain.model, which = 2) #quantile-quantile
```

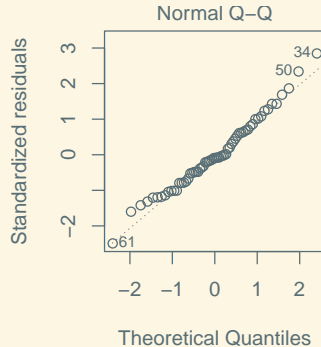
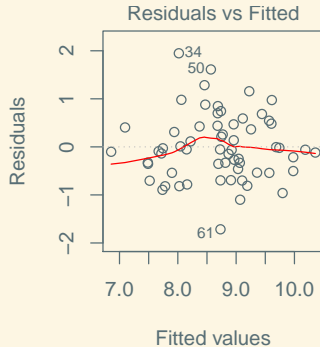


# Percent Brain Weight by Body Weight

```
library(mosaic)
mutate(
  BrainBodyWeight,
  percent.brain = brain.weight.grams / (body.weight.kilograms * 1000)
) %>%
xyplot(
  log(percent.brain) ~ log(body.weight.kilograms),
  data = ., type = c("p", "r")
)
```



# 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) to address right-skew
- Less common, but sometimes use convex transformations (exp, powers) to address left-skew
- Log turns multiplicative (proportional) change into additive change (one unit difference in log scale corresponds to a constant *ratio* in the original scale)



# Influential Points

Handout

# Unusual Cases

## Influential point

An **influential point** is a data point that by itself has a large effect on the fitted regression line.

How do we measure “effect on the fit”?

# Kinds of Influential Points

## Outliers

An **outlier** is a data point that is unusually far from the trend line (i.e., it has an unusually large residual).

## Leverage

A point has high **leverage** if it has an unusually extreme value on a predictor (explanatory variable). (Think of a see-saw)

# Unusual Cases

## Detecting Unusual Cases

- Residual plots
- Standardized/Studentized residuals
- Leverage measurement
- Cook's distance

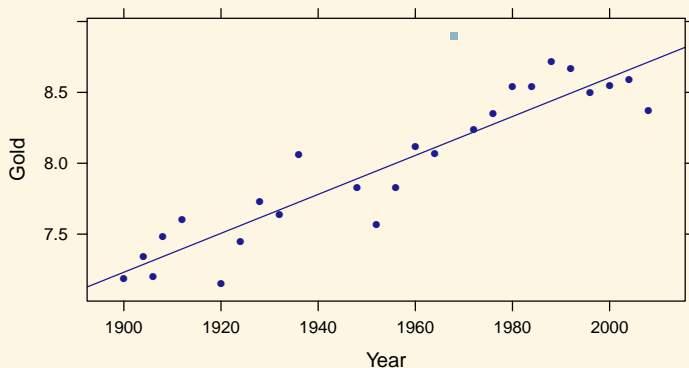
# Raw vs. Standardized Residuals

How do we tell if a residual is unusually large?

$$\begin{array}{ll} Y = GPA & \varepsilon_i = 2.6 \text{ is very large} \\ Y = SAT & \varepsilon_i = 2.6 \text{ is very small} \end{array}$$

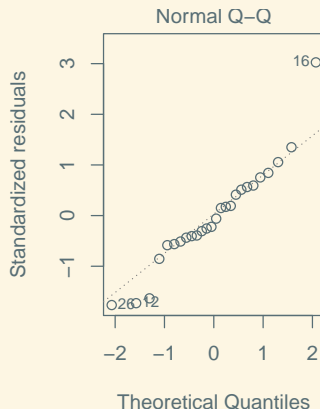
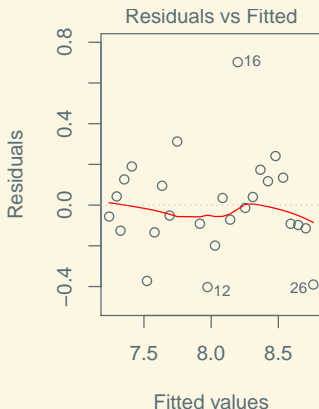
# Men's Long Jump

```
library(Stat2Data)
data(LongJumpOlympics)
xyplot(
  Gold ~ Year, data = LongJumpOlympics, type = c("p", "r"),
  groups = (Year == 1968) ## highlight the outlier
)
```



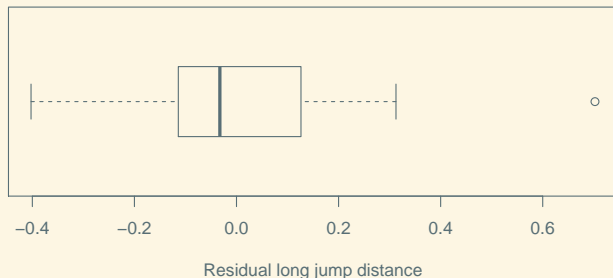
# Men's Long Jump: Residuals

```
long.jump.model <- lm(Gold ~ Year, data = LongJumpOlympics)
par(mfrow = c(1,2))
plot(long.jump.model, which = 1)
plot(long.jump.model, which = 2)
```



# Men's Long Jump: Box Plot of Residuals

```
boxplot(residuals(long.jump.model), horizontal = TRUE,  
        xlab = "Residual long jump distance")
```



The outlier has a residual which is more than one and a half times the IQR beyond the upper quartile.



# Standardized and Studentized Residuals

## Standardized Residuals

$$\frac{y_i - \hat{y}_i}{\hat{\sigma}_\varepsilon \sqrt{1 - h_i}} \quad (1)$$

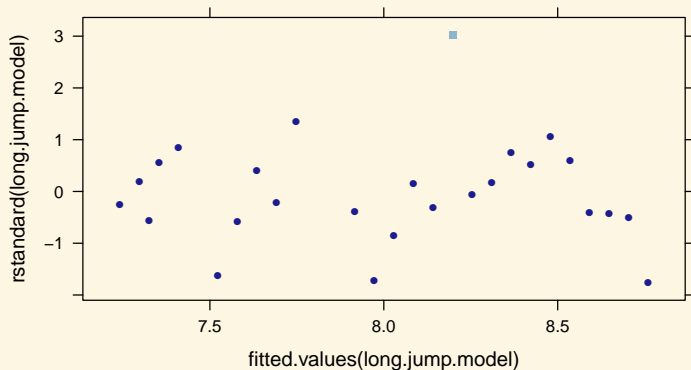
## “Studentized” Residuals

$$\frac{y_i - \hat{y}_i}{\hat{\sigma}_\varepsilon^{(i)} \sqrt{1 - h_i}} \quad (2)$$

where  $\hat{\sigma}_\varepsilon^{(i)}$  is standard deviation of all residuals *other than i*, and  $h_i$  is called the “leverage” of point  $i$  (more on this soon)

# Standardized Residuals

```
xyplot(rstandard(long.jump.model) ~ fitted.values(long.jump.model),  
       groups = abs(rstandard(long.jump.model)) > 2)
```



# Standardized Residuals

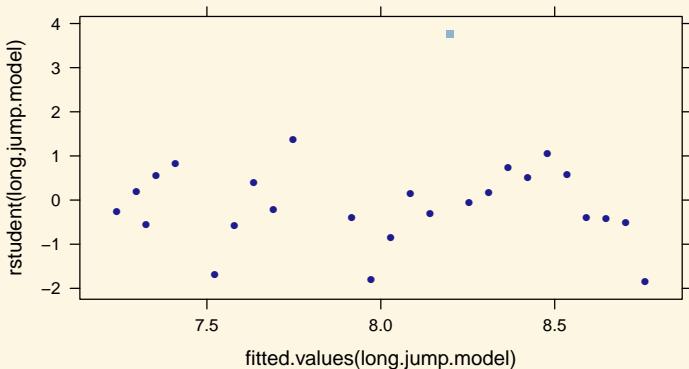
Keep an eye out for standardized residuals in excess of  $\pm 2$  or  $\pm 3$ .  
When normality is satisfied, only 5% will exceed  $\pm 2$ , and less than 0.2% exceed  $\pm 3$ .

## Studentized Residuals

- Worry: a large residual will inflate the estimated standard deviation of the residuals, and therefore lead to underestimating the standardized residual.
- Solution: “Studentize” the residuals by fitting the model with that case removed, then find new  $\hat{\sigma}_\epsilon$ .

# Studentized Residuals

```
xyplot(rstudent(long.jump.model) ~ fitted.values(long.jump.model),  
       groups = abs(rstudent(long.jump.model)) > 2)
```



# Influence

Two characteristics contribute to influence of a data point on regression line:

1. Distance in  $Y$  from trend (think: residual for line fit w/o that point)
2. Distance of  $X$  from  $\bar{X}$  (think: distance from center on a see-saw)

# Leverage

## Leverage

$$h_i = \frac{1}{n} \left( 1 + \frac{(x_i - \bar{x})^2}{\frac{1}{n} \sum_{i'=1}^n (x_{i'} - \bar{x})^2} \right) \approx \frac{1}{n} (1 + z_{x,i}^2)$$

where  $z_{x,i}$  is the  $z$ -score of the  $i$ th data point in the  $x$  direction.  $h_i$  measures influence of a point with value  $x_i$  on regression line.

Typical leverage:  $2/n$ . Keep an eye out for leverage in excess of  $4/n$  or  $6/n$ .

# Standardized and Studentized Residuals

## Standardized Residuals

$$\frac{y_i - \hat{y}_i}{\hat{\sigma}_\varepsilon \sqrt{1 - h_i}} \quad (3)$$

## “Studentized” Residuals

$$\frac{y_i - \hat{y}_i}{\hat{\sigma}_\varepsilon^{(i)} \sqrt{1 - h_i}} \quad (4)$$

where  $\hat{\sigma}_\varepsilon^{(i)}$  is standard deviation of all residuals *other than*  $i$ , and  $h_i$  is the leverage of point  $i$  (more on this soon).



## Cook's Distance

Another measure of influence that combines leverage and size of residual is **Cook's Distance**.

### Cook's Distance

$$D_i := \frac{\sum_{j=1}^J (\hat{y}_j - \hat{y}_{j(i)})^2}{(k+1)\hat{\sigma}_\varepsilon^2}$$

Equivalently

$$D_i := \frac{\text{stdres}_i}{k+1} \left[ \frac{h_i}{1-h_i} \right]$$

where  $k$  is the number of predictors,  $\hat{y}_j$  is the predicted value of data point  $j$  with all data,  $\hat{y}_{j(i)}$  is the predicted value if data point  $i$  is removed,  $\text{stdres}_i$  is the standardized residual,  $\hat{\sigma}_\varepsilon^2$  is the estimated variance of the residuals, and  $h_i$  is the leverage.

# Cook's Distance for Long Jumps

```
plot(long.jump.model, which = 4)
```

