



THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

Residuals and autocorrelation 1

Lecture 5

STA 371G

Announcements

- Homework 1 due Thursday at 11:59 PM, in MyStatLab
- Submit an R script in Canvas that contains the commands you used for each problem

1. Transformations

2. Extrapolation

Residuals

Recall that the **residual** for the i th case in the data is $Y_i - \hat{Y}_i$.

- When the residual is *positive*, the actual Y -value is *higher* than our predicted Y -value.
- When the residual is *negative*, the actual Y -value is *lower* than our predicted Y -value.

Looking at residuals can tell us a lot about how well a model is working, and give us ideas for how to improve it.

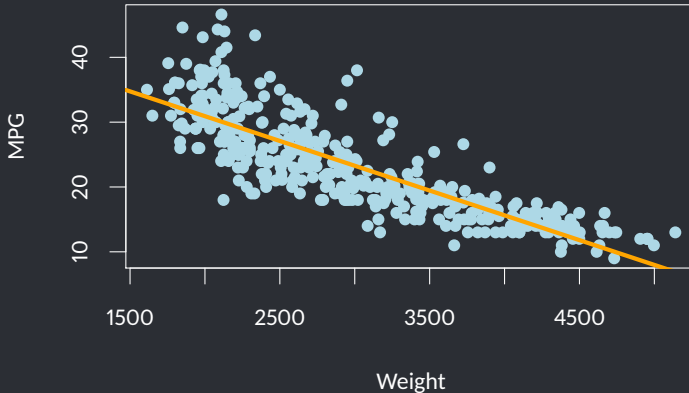
Mileage efficiency data set

The data set cars contains specs for 392 different cars. We'll focus on two variables:

- **MPG** is fuel efficiency, measured as miles per gallon
- **Weight** is the weight of the car, in pounds

What problems do you see here?

```
> plot(MPG ~ Weight, data=cars, pch=16, col="lightblue")  
> model <- lm(MPG ~ Weight, data=cars)  
> abline(model, col="orange", lwd=4)
```



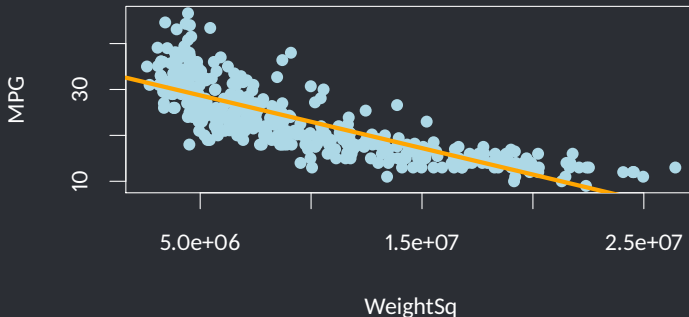
Using transformations to fix problems

- Sometimes, a violation of regression assumptions can be fixed by transforming one or the other of the variables (or both).
- When we transform a variable, we have to also transform our interpretation of the equation.

A bad example

What if we predict MPG from squared weight?

```
> cars$WeightSq <- cars$Weight^2  
> plot(MPG ~ WeightSq, data=cars, pch=16, col="lightblue")  
> sq.model <- lm(MPG ~ WeightSq, data=cars)  
> abline(sq.model, col="orange", lwd=4)
```



The log transformation

The **log** transformation is frequently useful in regression, because many nonlinear relationships are naturally exponential.

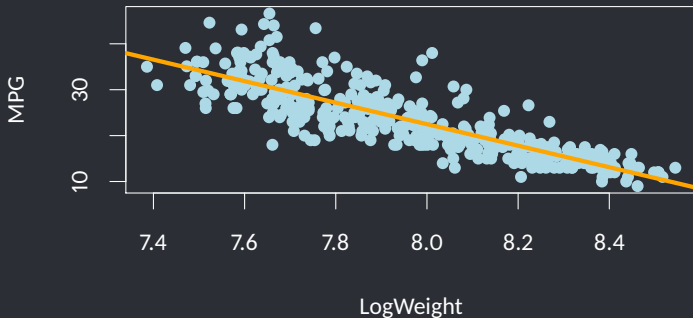
- $\log_b x = y$ when $b^y = x$.
- For example, $\log_{10} 1000 = 3$, $\log_{10} 100 = 2$, and $\log_{10} 10 = 1$.
- The natural log is \log_e , where $e \approx 2.72$ — when we say “log” we will usually mean “natural log.”

LOG!



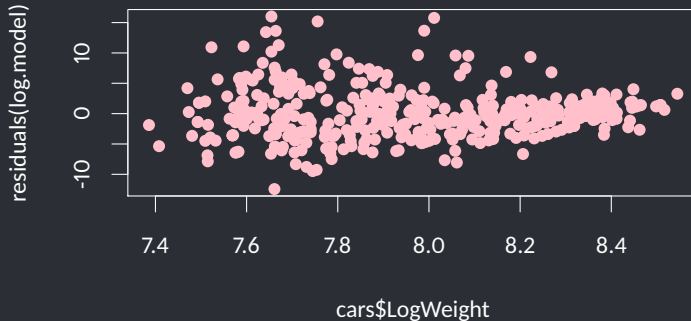
Applying a log transformation

```
> cars$LogWeight <- log(cars$Weight)
> plot(MPG ~ LogWeight, data=cars, pch=16, col="lightblue")
> log.model <- lm(MPG ~ LogWeight, data=cars)
> abline(log.model, col="orange", lwd=4)
```



Checking assumptions of our new model

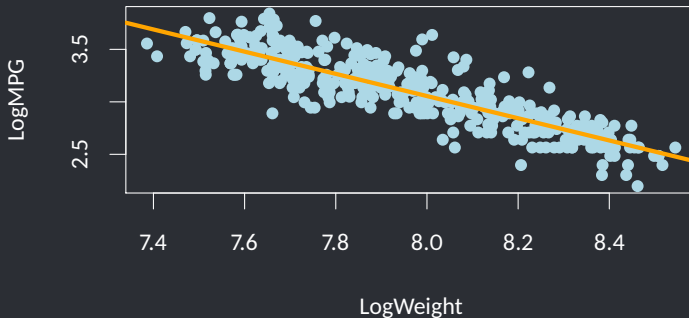
```
> plot(cars$LogWeight, residuals(log.model), pch=16, col="pink")
```



Linearity looks good, but homoscedasticity is still not satisfied!

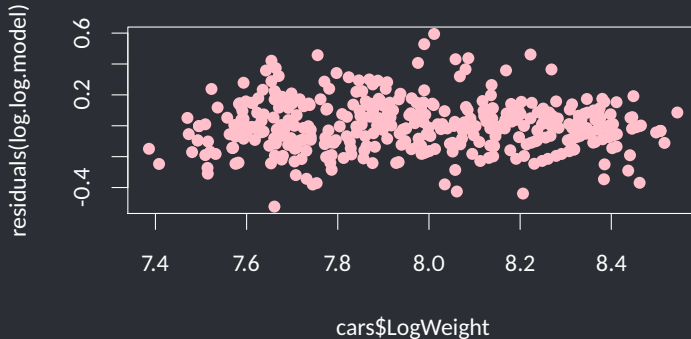
Applying a second log transformation

```
> cars$LogMPG <- log(cars$MPG)
> plot(LogMPG ~ LogWeight, data=cars, pch=16, col="lightblue")
> log.log.model <- lm(LogMPG ~ LogWeight, data=cars)
> abline(log.log.model, col="orange", lwd=4)
```



Checking assumptions of our new model

```
> plot(cars$LogWeight, residuals(log.log.model), pch=16, col="pink")
```



Much better—transforming MPG to $\log(\text{MPG})$ gives us both linearity and homoscedasticity!

Thinking about transformations

- Thinking about whether you want to stretch or squeeze one of the axes, and apply a transformation accordingly (e.g., \sqrt{x} or $\log x$ to squeeze; x^2 or e^x to stretch).

Thinking about transformations

- Thinking about whether you want to stretch or squeeze one of the axes, and apply a transformation accordingly (e.g., \sqrt{x} or $\log x$ to squeeze; x^2 or e^x to stretch).
- Transformations of Y can address both heteroscedasticity and nonlinearity; transformations of X can only address nonlinearity.

Thinking about transformations

- Thinking about whether you want to stretch or squeeze one of the axes, and apply a transformation accordingly (e.g., \sqrt{x} or $\log x$ to squeeze; x^2 or e^x to stretch).
- Transformations of Y can address both heteroscedasticity and nonlinearity; transformations of X can only address nonlinearity.
- You might need to transform both X and Y ; if so, start by transforming Y to address the heteroscedasticity, and then transform X to address nonlinearity if necessary.

Thinking about transformations

- Thinking about whether you want to stretch or squeeze one of the axes, and apply a transformation accordingly (e.g., \sqrt{x} or $\log x$ to squeeze; x^2 or e^x to stretch).
- Transformations of Y can address both heteroscedasticity and nonlinearity; transformations of X can only address nonlinearity.
- You might need to transform both X and Y ; if so, start by transforming Y to address the heteroscedasticity, and then transform X to address nonlinearity if necessary.
- It's OK to do a little trial and error!

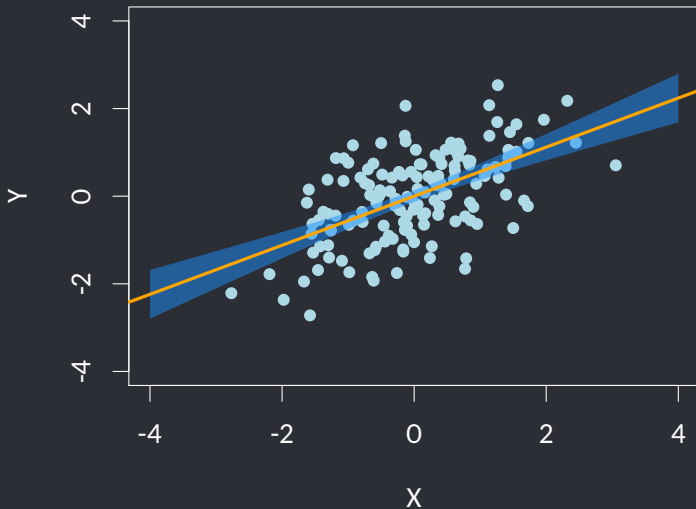
1. Transformations

2. Extrapolation

Going beyond the data

- It's natural to want to predict Y beyond the X values that we have in the data set (otherwise, why build a model in the first place?).
- But things get dicey when trying to predict Y for an X value that is far from the other X values in the data set: how can we be so sure that the observed trend continues?

The shaded area shows the 95% confidence interval for predicting the mean. As X moves away from \bar{X} , the CI becomes wider since we know our estimates are less precise.



There's nothing wrong with extrapolating a little bit beyond the data, but when you move beyond the data even the confidence intervals may *underestimate* the degree of uncertainty.