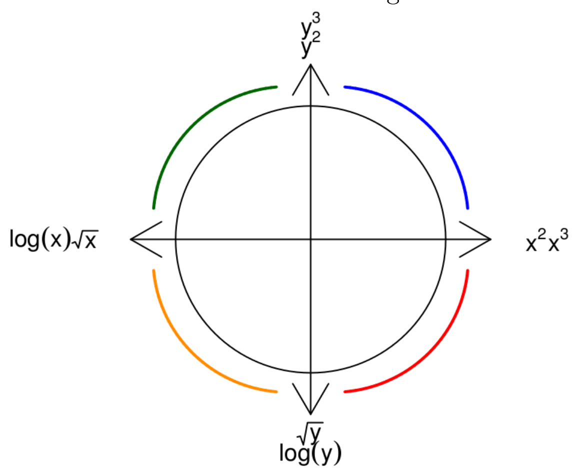
Agenda

1. Transformations
2. Log Coefficients

Transformations Tukey's Bulging Rule is a systematic approach for transforming variables. The idea is to move up or down the "ladder" in the direction indicated in the diagram.

$$\begin{aligned} 3 &: x^3 \\ 2 &: x^2 \\ 1 &: x \\ \frac{1}{2} &: \sqrt{x} \\ 0 &: \log x \\ -\frac{1}{2} &: -1/\sqrt{x} \\ -1 &: -1/x \\ -2 &: -1/x^2 \\ -3 &: -1/x^3 \end{aligned}$$



Transformations lab We'll go through the lab together. Most of the code isn't stuff you will need to know, but here are a few pieces that might be useful.

```
Rand = Rand %>%
  mutate(y_new = log(y))
xyplot(y_new ~ x, data=Rand)

require(manipulate)
manipulate(
  with(Rand, tukeyPlot(x, y, q.y))
  , q.y = slider(-3, 3, step=0.25, initial=1)
)

require(Stat2Data)
data(SpeciesArea)
xyplot(Species ~ Area, data=SpeciesArea)

manipulate(
  with(SpeciesArea, tukeyPlot(Area, Species, q.y, q.x))
  , q.y = slider(-3, 3, step=0.25, initial=1)
  , q.x = slider(-3, 3, step=0.25, initial=1)
)

xyplot(log(Species) ~ log(Area), data=SpeciesArea)

miniSpecies = SpeciesArea %>%
  slice(-c(1:10))

SpeciesArea = SpeciesArea %>%
  mutate(BigSmall = ifelse(Area>1500, "Big", "Small"))
```

```
xyplot(Species~Area, data=SpeciesArea, group=BigSmall, auto.key = TRUE)

ggplot(SpeciesArea) + geom_point(aes(x=Area, y=Species, col=BigSmall))
ggplot(SpeciesArea) + geom_point(aes(x=Area, y=Species, shape=BigSmall))
```

Interpreting log coefficients There are two commonly-used logs: log base 10, and natural log (base e). The book likes log base 10, but in this class we will be using *natural log*.

Some of the datasets in the Stat2Data package have pre-tranformed variables, like the `Caterpillars` data in the homework. Don't use the `LogMass` variable, instead, either create your own new variable using `mutate()` or just wrap the original variable name in `log()` in the model call.

```
# install.packages("fueleconomy")
require(fueleconomy)
m1 <- lm(log(hwy)~displ, data=vehicles)
coef(m1)

## (Intercept)       displ
##   3.5724757  -0.1332845
```

What is the equation of the line?

$$log(hwy) = 3.57 - 0.133 * displ$$

And our interpretation on the slope coefficient would be, for every one-litre increase in engine displacement, we would expect to see a 13.3% decrease in highway mileage. We can begin to transform back into the original data space.

$$\begin{aligned} log(hwy) &= 3.57 - 0.133 * displ \\ e^{log(hwy)} &= e^{3.57-0.133*displ} \\ hwy &= \frac{e^{3.57}}{e^{0.133*displ}} = \frac{35.52}{e^{0.133*displ}} \end{aligned}$$

Lets plug some numbers in for concreteness. If we plug in displ=4, $hwy = \frac{35.52}{1.70} = 20.89$, and with displ=5 (a one-litre increase) $hwy = \frac{35.5}{1.94} = 18.28$

Or, in R for more precision

```
exp(3.5724757)/(exp(0.1332845*4))
exp(3.5724757)/(exp(0.1332845*5))
20.8914 * 0.1332845
20.8914 - 2.7845
```

18.28 is approximately a 13.3% decrease from 20.89. Convenient, no?

What if we had done this with log10?

```
m2 <- lm(log10(hwy)~displ, data=vehicles)
coef(m2)

## (Intercept)       displ
##  1.55150650 -0.05788473
```

$$\begin{aligned} log_{10}(hwy) &= 1.55 - 0.058 * displ \\ 10^{log_{10}(hwy)} &= 10^{1.55-0.058*displ} \\ hwy &= \frac{10^{1.55}}{10^{0.058*displ}} = \frac{35.60}{10^{0.058*displ}} \end{aligned}$$

Plugging in displ = 4, $hwy = \frac{35.60}{1.70} = 20.89$, and displ = 5 $hwy = \frac{35.60}{1.94} = 18.28$

So, the predictions are the same. But, what of the coefficient interpretation? It's not so simple with log 10.