Data Transformations

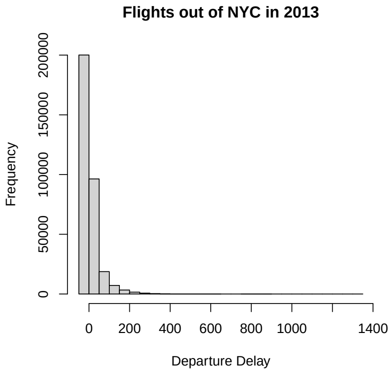
Ryan Miller

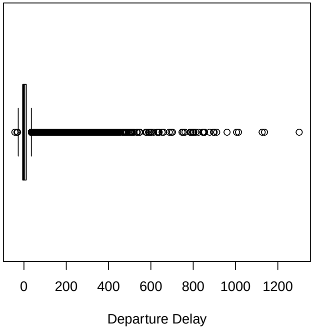
Introduction

- It's foolish to think you can jump straight into modeling before first spending some time to understand your data via visualization
  - You might mistakenly use a highly skewed variable in a model that assumes normality
  - You might overlook missing values, extreme outliers, or recording errors
  - You might choose a model family that cannot accommodate how your variables are related (ie: linear vs. polynomial, etc.)

Understanding your Response Variable

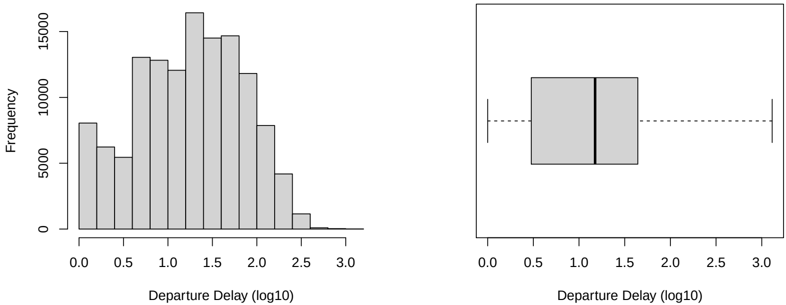
- *Linear regression* models tend to work best when the response variable is Normally distributed
  - You can fit a straight-line to *any data*
  - Normality helps allow for *valid statistical inference*





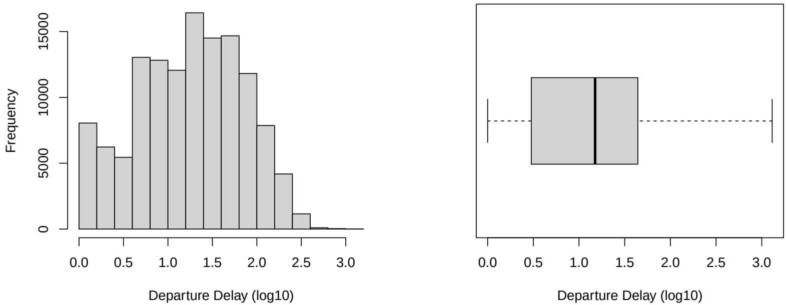
Log-Transformations

Taking base-10 logarithm of the delays makes a big difference. But how do we interpret the transformed variable?



Log-Transformations

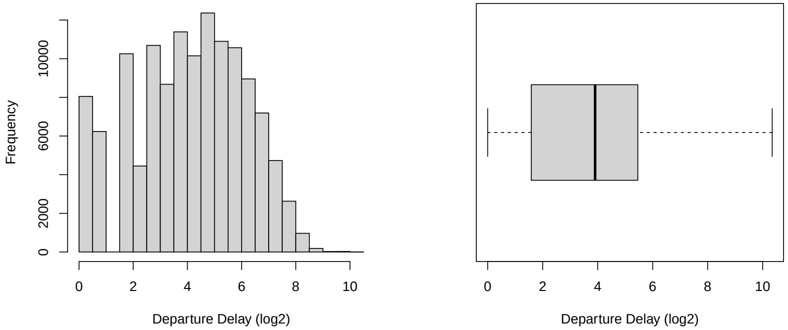
Taking base-10 logarithm of the delays makes a big difference. But how do we interpret the transformed variable?



$log_{10}(1) = 0$, $log_{10}(10) = 1$, $log_{10}(100) = 2$, so each 1-unit increment on the log_{10} scale corresponds to a 10-fold change on the original scale

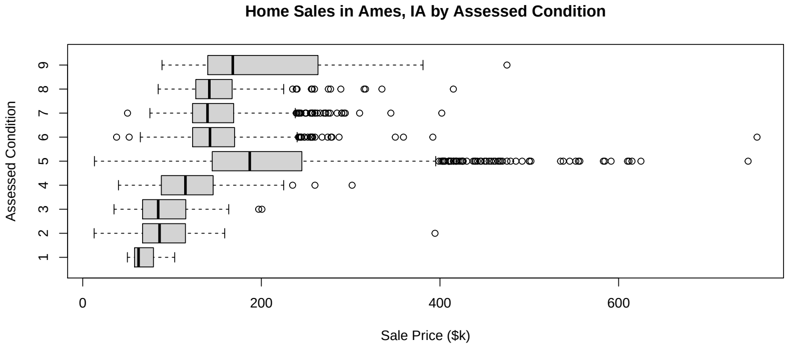
Log-transformations

We can also consider taking the base-2 logarithm if it's more sensible to consider 2-fold changes (doublings)



Category Merging

- A major goal of modeling is provide a reasonable simplification of the relationships seen in data
  - A potential challenge is categorical data with many categories



Do we really need to use 9 different condition ratings?

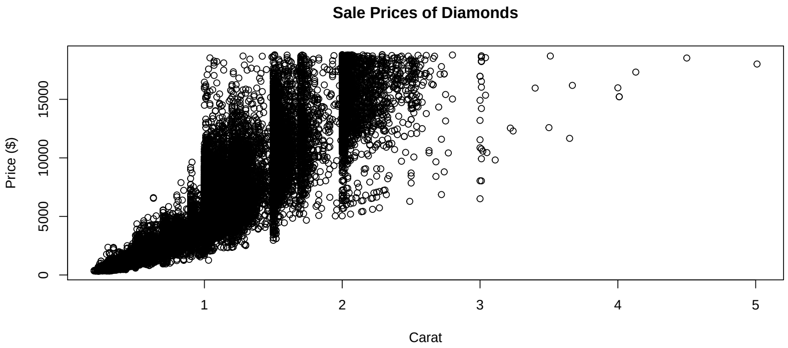
Category Merging

- By expressing condition as "Poor" (1-4) or "Acceptable" (5-10) we can retain the essence of the relationship between condition and price

```r
library(forcats)
AmesHousing$OverallCond2 <- fct_collapse(factor(AmesHousing$OverallCond),
                                         Acceptable = c("5","6","7","8","9","10"),
                                         Poor = c("1","2","3","4"))
boxplot(AmesHousing$SalePrice/1000 ~ AmesHousing$OverallCond2, horizontal = TRUE)
```

Cutting

- Sometimes the relationship between a numeric predictor and a response variable is complicated



- Notice the big price jumps at 1.0 carats, 1.5 carats, and 2.0 carats

Cutting

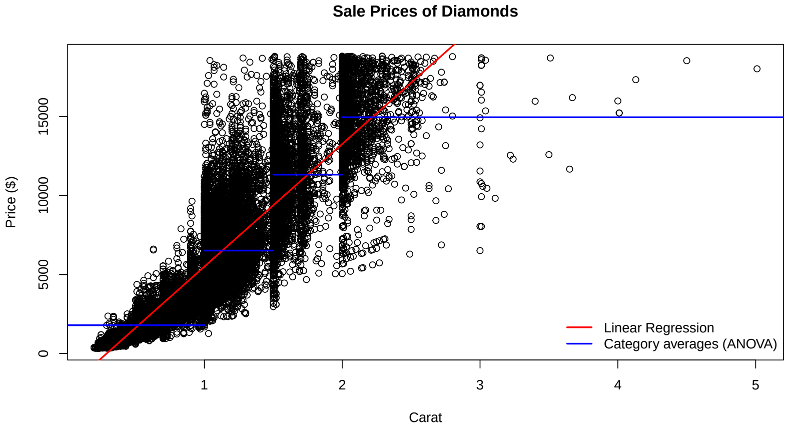
- In a situation like this one, it be sensible to express the explanatory variable using categories (rather than forcibly impose a linear relationship)
  - We can group diamonds into the following categories $(0,1), [1,1.5), [1.5,2), [2,\infty)$ using the `cut` function

```
diamonds$carat_cat <- cut(diamonds$carat, breaks = c(0,1,1.5,2,Inf))
table(diamonds$carat_cat)
```

```
##
##   (0,1] (1,1.5] (1.5,2] (2,Inf]
##   36438   12060    3553    1889
```

Cutting

There are pros and cons to each approach, but we can look at these models visually to determine which is more useful:



Closing Remarks

- In pursuit of this goal we should acknowledge the trade-off between accuracy and simplicity

  "The goal of a model is to provide a low-dimensional summary of a dataset"

Closing Remarks

- In pursuit of this goal we should acknowledge the trade-off between accuracy and simplicity

  "The goal of a model is to provide a low-dimensional summary of a dataset"

- Sometimes a simpler model should be favored, even if it is slightly less precise
  - Log-transformations prior to linear regression can be preferable to polynomials or complex algorithms
  - Category merging and cutting can each aide in interpretation, even if it some details are lost