Volpe R Course: Session 3

Instructors: Don Fisher, Dan Flynn, Jessie Yang
Course webpage: `http://bit.ly/volpeR`

4/13/2017

Overview

Last session we covered

- Comparing frequencies of events
  - Chi-square
- Assessing relationships between continuous variables
  - Correlation and regression
  - Simple linear regression and multiple regression

Today we will continue this work on Analysis of Variance, and dive more into the details of linear models

- Comparing means of groups of variables
  - T-test and ANOVA

Also, we will cover a few additional topics:

- Loops
- P-values

Homework: Data analysis

Homeworks received before today show great work! Some common themes:

- How to deal with outliers
- How to do repeated processing or analysis steps
- How to diagnose linear models

We will address the first two points especially.

Review: T-test: compare means of two groups

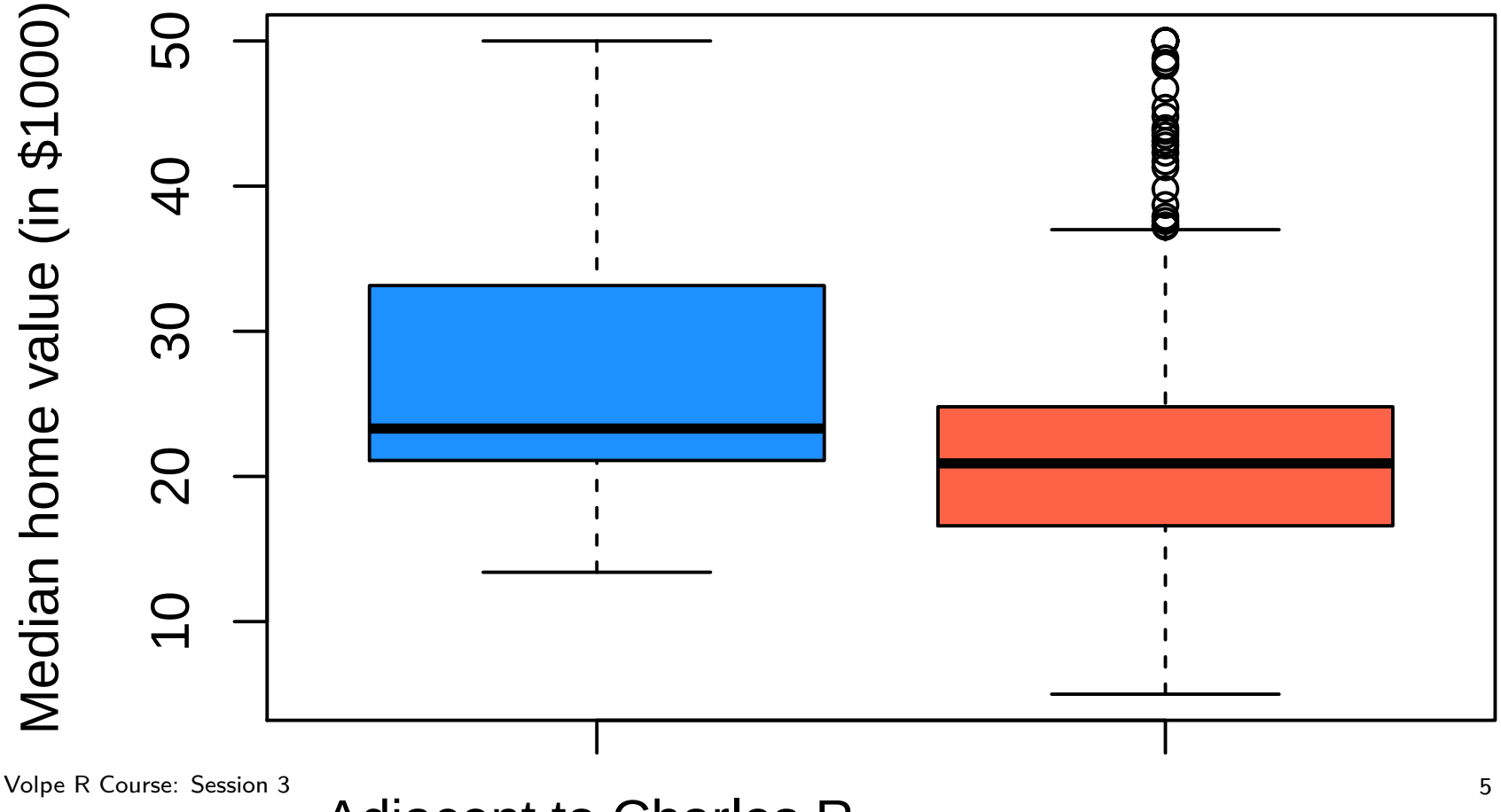
Comparing two groups is done in R with the function `t.test`. This tests if two groups have the same mean value. Here we are looking at housing prices for houses adjacent to the Charles river (`chas == 1`) or not:

```
with(Boston, t.test(medv[chas=="1"], medv[chas=="0"]))
```

```
##
##  Welch Two Sample t-test
##
## data:  medv[chas == "1"] and medv[chas == "0"]
## t = 3.1133, df = 36.876, p-value = 0.003567
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   2.215483 10.476831
## sample estimates:
## mean of x mean of y
##  28.44000  22.09384
```

T-Test: compare means of two groups

```
with(Boston, boxplot(medv[chas=="1"],
                     medv[chas=="0"],
                     names = c("Adjacent to Charles R.",
                               "Not adjacent to Charles R."),
                     col = c("dodgerblue", "tomato"),
                     ylab = "Median home value (in $1000)"
                     )
     )
```



Analysis of Variance (ANOVA): compare means of more than two groups

Analysis of variance is used for comparing means of more than two groups. Analysis of covariance (ANCOVA) combines features of ANOVA and regression. In fact, all of these models are related, in that they are linear models. A *generalized linear model* is the broadest category of these models.

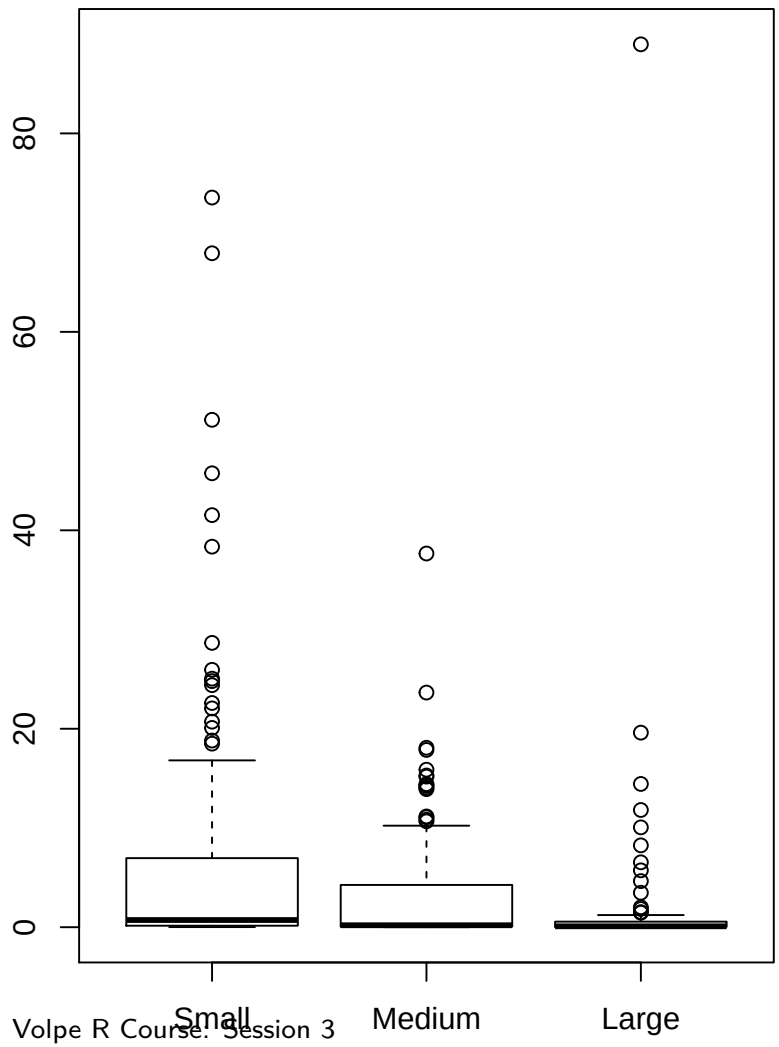
```
# create a three-category variable from the number of rooms per house

Boston$house.size <- cut(Boston$rm,
                         breaks = c(0, 6, 6.8, 10),
                         labels = c("Small", "Medium", "Large"))

m1 <- aov(crim ~ house.size, data = Boston)
summary(m1)
```

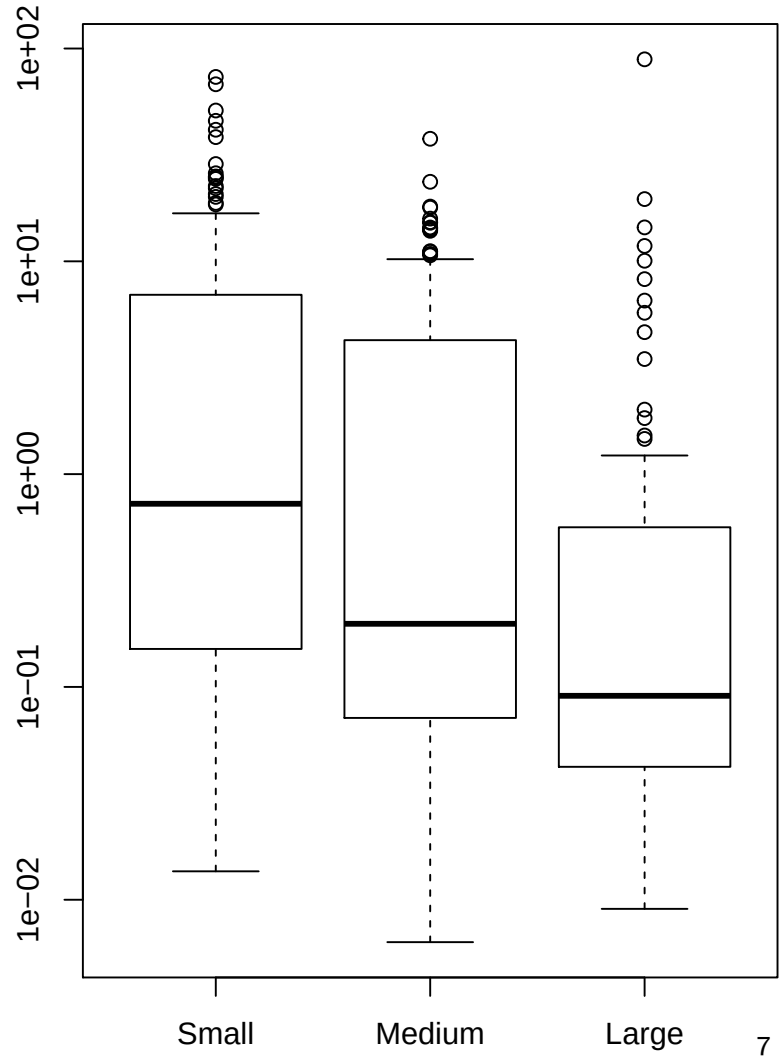
```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## house.size     2   1153   576.6   8.009 0.000377 ***
## Residuals    503  36210    72.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analysis of Variance (ANOVA): compare means of more than two groups

Probably need to consider distributions here, as well!

```
par(mfrow = c(1, 2))
boxplot(crim ~ house.size, data = Boston)
boxplot(crim ~ house.size, data = Boston, log = "y")
```

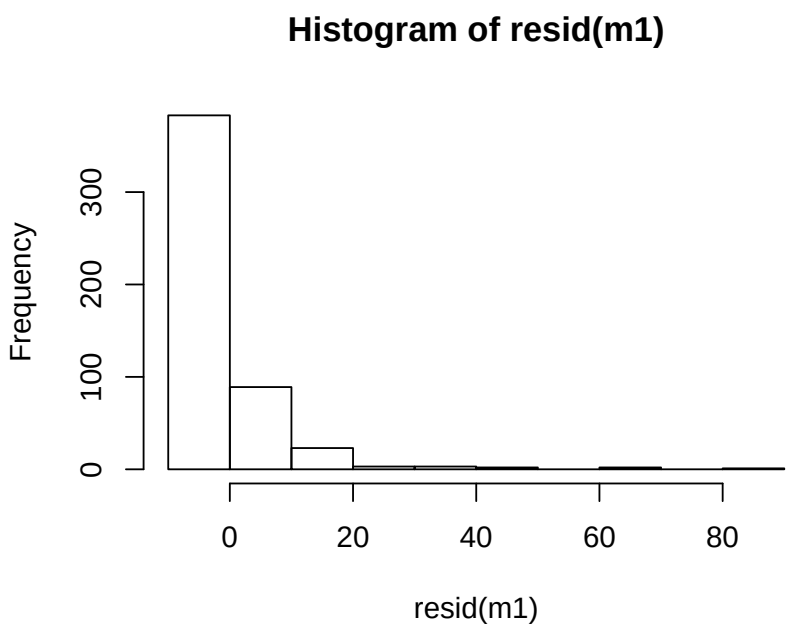


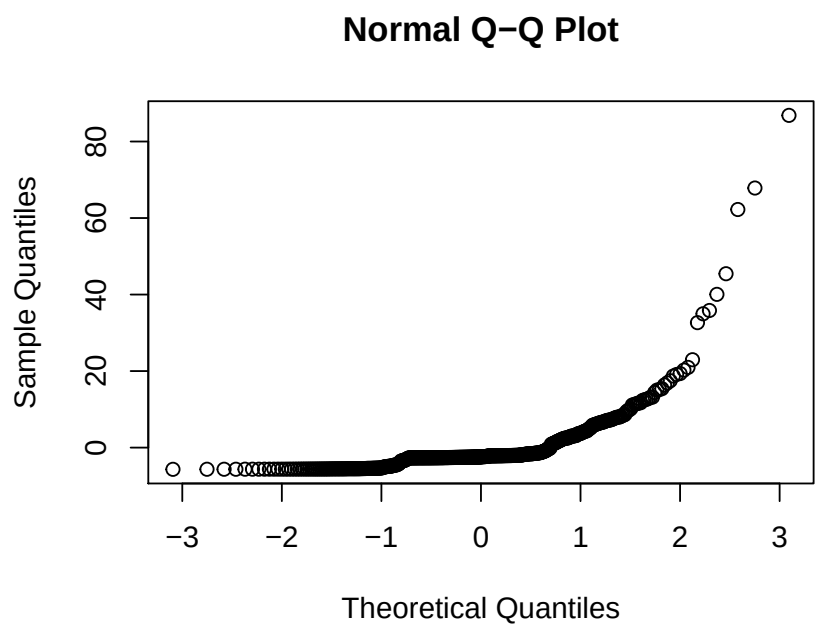


Analysis of Variance (ANOVA): compare means of more than two groups

Do we need to transform the data? As for the regresssion models above, the important point is that the *residuals of the model need to be normally distributed*. It is not important if the input data themselves are normal or not.

```
par(mfrow = c(1,2))
hist(resid(m1))
qqnorm(resid(m1)) # Not great!
```



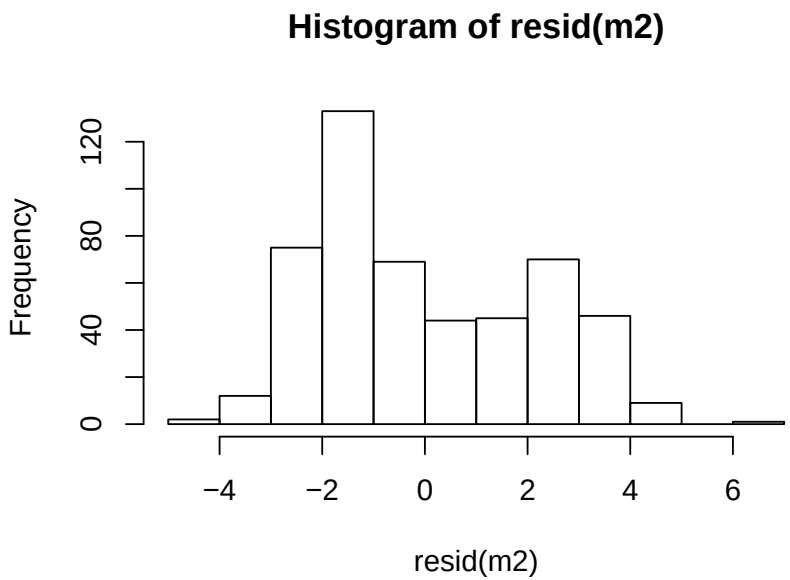


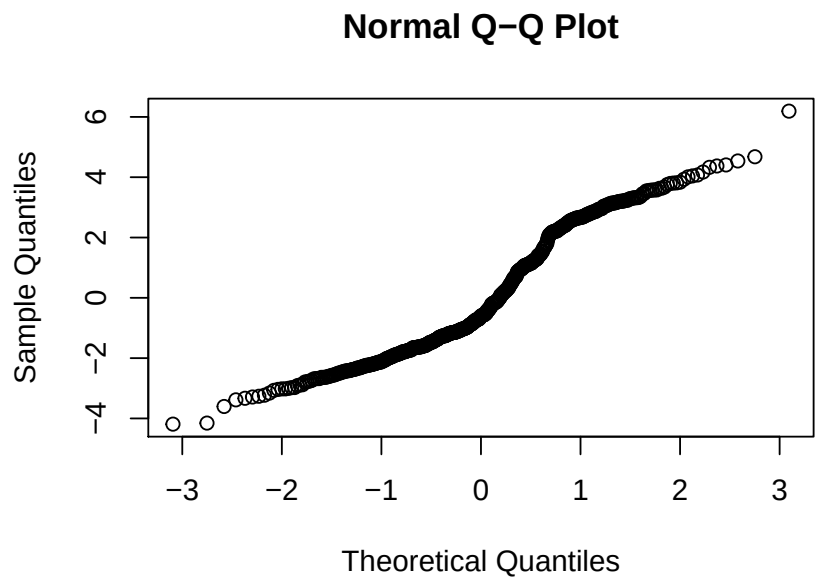
Analysis of Variance (ANOVA): compare means of more than two groups

```
m2 <- aov(log(crim) ~ house.size, data = Boston)
summary(m2)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## house.size      2  160.3   80.13   18.32 2.09e-08 ***
## Residuals     503 2200.3    4.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow = c(1,2))
hist(resid(m2))
qqnorm(m2$residuals) # Much better!
```





Histogram of log(ytest)

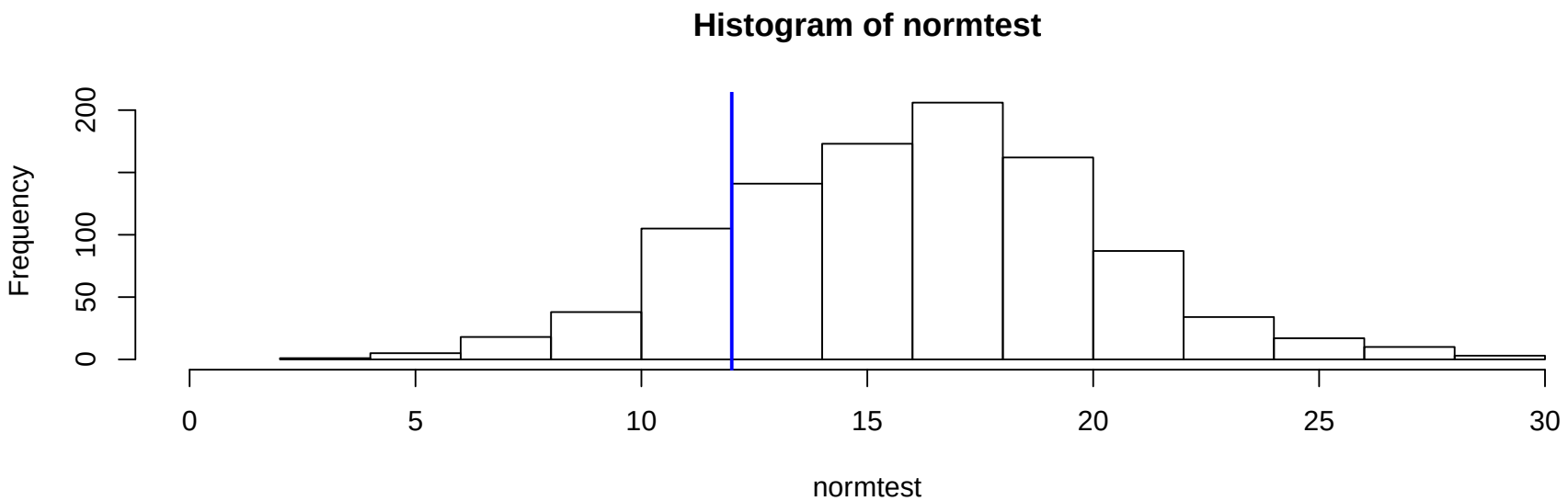
P-values

A quick digression back to P-values (and also covering loops)

Remember that a p-value says how often you would observe this same result (or one more extreme) by chance alone. A statistically significant result is one which would rarely occur by chance alone.

Consider a normal distribution:

```
normtest <- rnorm(1000, mean = 16, sd = 4)
hist(normtest, xlim = c(0, 30))
# A random test value: the sum of the last four digits of your phone number
lastfour = 7 + 4 + 1 + 0 # use your own!
abline(v = lastfour, lwd = 2, col = 'blue')
```



Does that observation fall between the upper or lower 2.5% of the distribution?

P-values and loops

```
upperlower <- quantile(normtest, c(0.025, 0.975))
(signif <- lastfour >= upperlower[1] | lastfour <= upperlower[2])
```

```
## 2.5%
## TRUE
```

Let's do an experiment, making taking multiple random draws. First, let's learn about loops.

```
for(i in 1:10){
  print(LETTERS[i])
  }

container <- vector()
for(indexvalue in 1:10){
  container <- c(container, LETTERS[rnorm(1, mean = 13, sd = 4)])
  }
container
```

Exercise 3.1

Make a loop which selects 5 random colors and create a plot using these colors!

P-values and loops

P-value experiment:

```
runs = 100

pvals = vector()
for(i in 1:runs){
  normtest <- rnorm(25, mean = 16, sd = 4)
  upperlower <- quantile(normtest, c(0.025, 0.975))
  signif <- lastfour >= upperlower[1] | lastfour <= upperlower[2]
  pvals = c(pvals, signif)
  }
summary(pvals)
```

```
##    Mode    TRUE    NA's
## logical     100       0
```

```
cat("p = ", round(length(pvals[pvals == TRUE])/length(pvals), 2))
```

```
## p =  1
```

Scaling data

For many of your data analysis tasks, you can use your data in the scale you were working in originally. Sometimes, your predictors will be in very different scales, so to interpret the influence of each predictor, it is better to re-scale the data.

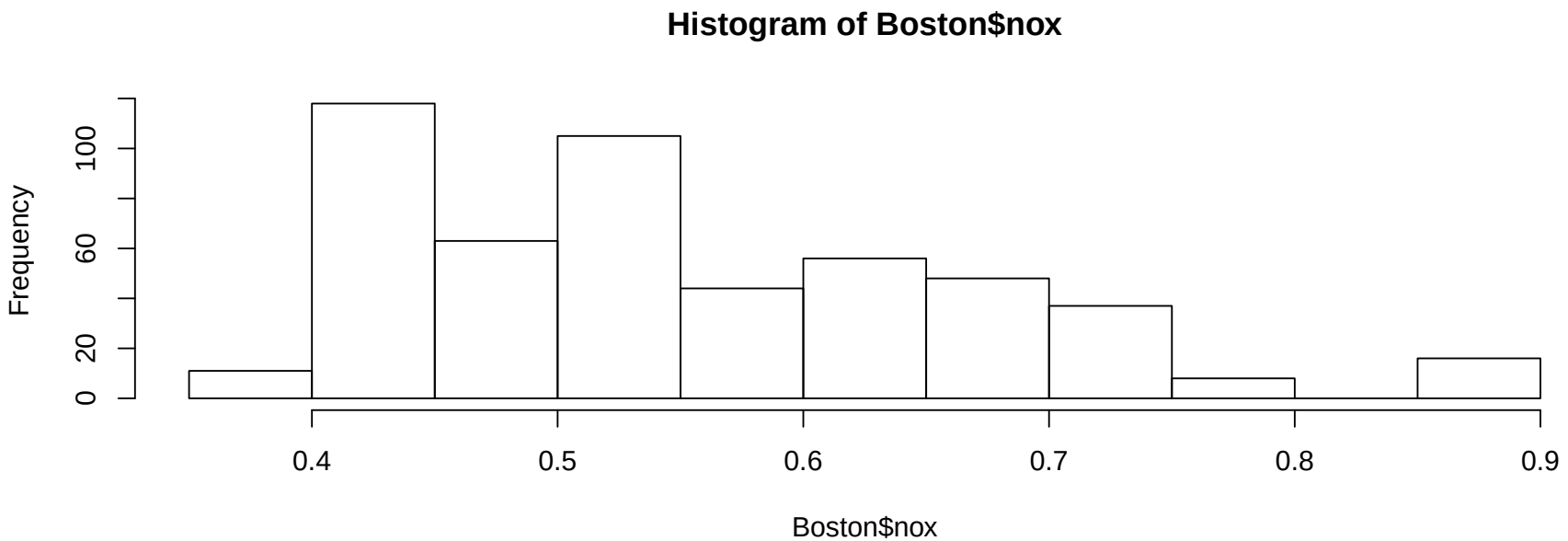
```
data(Boston)
m1 <- lm(medv ~ nox + crim, data = Boston)
Bo.scale <- Boston
Bo.scale[c("crim", "nox")] <- scale(Bo.scale[c("crim", "nox")])
m2 <- lm(medv ~ nox + crim, data = Bo.scale)
AIC(m1, m2) # these are the same models
coef(m1); coef(m2) # but now coefficients are more readily interpretable
```

You may see some text books recommending *always* re-scaling your predictors. This is also called *z-scoring* your predictors.

When are outliers "out"?

Detection of outliers can be accomplished mathematically, but the **interpretation** of these observations relies on understanding the data and intuition about the underlying processes.

```
hist(Boston$nox) # What would be considered an 'outlier'?
```




```
library(outliers) # install.packages('outliers')
out <- scores(Boston$nox, type = "chisq", prob = 0.95)
Boston$nox[out] # these are outliers, in a univariate sense.
```

```
##  [1] 0.871 0.871 0.871 0.871 0.871 0.871 0.871 0.871 0.871 0.871 0.871
## [12] 0.871 0.871 0.871 0.871 0.871
```

When are outliers "out"?

Interquartile range (IQR)

IQR is often used to detect outliers: distance between 25th percentile and 75th percentile of the data. Data within 1.5x the IQR from the mean is commonly used as a measure of what to keep.

```
iqr <- diff(quantile(Boston$nox, c(0.25, 0.75)))
low <- mean(Boston$nox) - 1.5*iqr
high <- mean(Boston$nox) + 1.5*iqr
Boston$nox[Boston$nox <= low | Boston$nox >= high]
# also see scores(..., type = "iqr")
```

When are outliers "out"?

Studentized residuals

An alternative test can be how much the *residuals* lie outside the predicted range. Residuals are the error $e_i = y_i - \hat{y}_i$

Where $\hat{y}_i$ represents the ith predicted value of y. The magnitude of the resiudals depends on the units of the dependent variable y. Dividing the residuals by the standard deviation of the data is called 'Studentizing' and shows values on a unit-less scale which can easily be tested. Values greather than 3 are often considered outliers.

```
studentized.resid <- residuals(m1)/(summary(m1)$sigma)
out <- studentized.resid[abs(studentized.resid) >= 3]
Boston[names(out),]
```

Exercise 3.2

Read in the Hubway data from the course page and look at tripduration. How would you decide which data should be considered outliers?

Homework

1. R skill: write two loops. One should loop generate random subsets of your data; Another should loop over rows in a data frame and perform some operation
2. Stats skill: test your data for outliers, using your models from Homework 2. Provide interpretation of your models, and try an analysis you didn't do in the previous round!

Document your work in a script as before, and upload to the Homework 3 folder on the course Google Drive.

As before: The more different things you try, the more feedback you will get from us!