Volpe R Course: Session 2

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

3/30/2017

Review

Last session we covered essential material on

- Basic R operations
- Summarizing data; writing functions
- Getting data in
- Indexing

Today we will move to data analysis in R, but will aim to review some of that material as we go.

Homework: Writing a function

Excercise 1:

Write a function to calculate the minimum, median, and standard deviation of a vector, and show the result on the console.

Some possible solutions:

```
summarize.1 <- function(x){
  print(min(x))
  print(median(x))
  print(sd(x))
  }

summarize.2 <- function(x){
  min.x <- min(x)
  med.x <- median(x)
  sd.x <- sd(x)
  print(data.frame(Min = min.x, Median = med.x, SD = sd.x))
}
```

Homework: Writing a function

```r
x = seq(1, 100, by = 7)
summarize.1(x)
```

```
## [1] 1
## [1] 50
## [1] 31.30495
```

```r
summarize.2(x)
```

```
##   Min Median       SD
## 1   1     50 31.30495
```

Homework: Writing a function

Some of your answers!

```
# This solution is more complex than we have time to discuss today!
sumstats = function(x) {
  str_out = sprintf("min:\t%s\nmedian:\t%s\nstdev:\t%s",
                    min(x), median(x), sd(x))
  cat(str_out)
  }

# This solution is similar to summarize.1, but with added text
descriptive <- function(x) {
  cat("vector minimum: ", min(x), "\n")
  cat("vector median: ", median(x), "\n")
  cat("vector standard deviation: ", sd(x), "\n")
  }
```

```r
sumstats(x)
```

```
## min: 1
## median:  50
## stdev:   31.3049516849971
```

```r
descriptive(x)
```

```
## vector minimum:  1
## vector median:  50
## vector standard deviation:  31.30495
```

See `?sprintf` and `?cat` to understand these solutions.

Lessons:

- There are many ways to solve problems in R!
- `print()` and `cat()` can be used return output to the console

The standard statistical toolbox

These tools are the heart of all statistical analysis. We are assuming you have encountered these before in other courses, but feel free to ask general questions.

Today we'll cover three types of standard tools:

- Comparing frequencies of events
  - Chi-square
- Assessing relationships between continuous variables
  - Correlation and regression
- Comparing means of groups of variables
  - T-test and ANOVA

Comparing frequencies of events

This is one of the simplest tests: are some events more common than others? The Chi-squared test is used to test independence of variables, as well as in goodness-of-fit measures.

Simple case:

Class	Number
Freshmen	32
Sophomores	28
Juniors	20
Seniors	21

We only have four data points; while we might suspect there is a trend in enrollment here, we can only do a simple Chi-squared test, which is calculated as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Comparing frequencies of events

In the given case, the expected value in each should be 0.25 x the total, 101 students = 25.25. We could manually calculate the Chi-squared statistic as follows:

```
chidat <- data.frame(Class = 1:4, Number = c(32, 28, 20, 21))
(E <- sum(chidat$Number)/nrow(chidat))
```

```
## [1] 25.25
```

```
# Manual calculation
(32 - E)^2 / E + (28 - E)^2 / E +
  (20 - E)^2 / E + (21 - E)^2 / E
```

```
## [1] 3.910891
```

```
chisq.test(chidat$Number)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  chidat$Number
## X-squared = 3.9109, df = 3, p-value = 0.2712
```

Comparing frequencies of events

In the case that the expected values are not identical, you can specify that in the `chisq.test` function. For example, perhaps the course is an upper-level course, and freshmen are expected to be only 15% of the registered students. We can specify the expected proportion as a vector `p`

```
chisq.test(chidat$Number, p = c(0.15, 0.25, 0.30, 0.30))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  chidat$Number
## X-squared = 25.396, df = 3, p-value = 1.276e-05
```

Exercise 2.1:

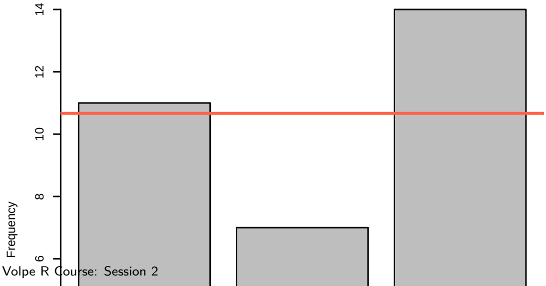
Look at the built in R data file `mtcars`, which we looked at last time. Use the `table` function to look at the frequency of cars with different number of cylinders in the engines, variable `cyl`. Carry out a Chi-squared test to see if the frequencies statistically different from equal. What do you conclude?

Exercise 2.1

```
cylfreq <- table(mtcars$cyl)
chisq.test(cylfreq)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  cylfreq
## X-squared = 2.3125, df = 2, p-value = 0.3147
```

```
par(cex = 0.5)
barplot(cylfreq, xlab = "Number of Cylinders", ylab = "Frequency")
abline(h = sum(cylfreq)/3, lwd = 2, col = "tomato")
```



Relationship between continuous variables

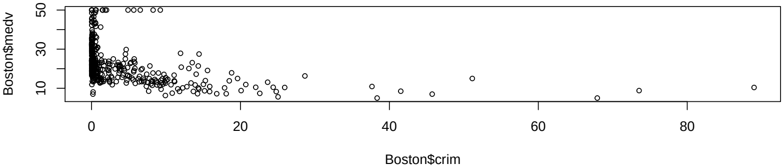
Load the Boston housing price data (already in R). These are from a 1978 research paper.

```
library(MASS)
summary(Boston)
```

There are two ways to think about continuous variables: correlation and regression.

Examine the correlation between crime rates and median value of owner-occupied houses:

```
cor(Boston$crim, Boston$medv)
plot(Boston$crim, Boston$medv, cex = 0.75)
```



`plot(medv ~ crim, data = Boston)` Would be another option

Relationship between continuous variables

We also often want to know if this relationship is different from what we would expect by chance:

```
cor.test(Boston$crim, Boston$medv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Boston$crim and Boston$medv
## t = -9.4597, df = 504, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4599064 -0.3116859
## sample estimates:
##        cor
## -0.3883046
```

Regression

For regression, we are assuming a causal relationship between the variables. Correlation only gives one output, the strength of the relationship expressed as `r`. For regression, we use workhorse of R analysis, `lm`:

```
house.crime <- lm(medv ~ crim, data = Boston)
summary(house.crime)
```

```
##
## Call:
## lm(formula = medv ~ crim, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.957  -5.449  -2.007   2.512  29.800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## crim        -0.41519    0.04389   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.484 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

Regression

There is a lot to unpack with these results. Recall that regression minimizes the sum of squares between the observed values and the expected values. This method is called *Ordinary Least Squares (OLS)*, finding the slope coefficient which best matches the predictor and response variables.

The key values to look at in this case are the **Estimate** for the predictor variable, `crim`, as well as for the intercept. We also want to know if these results are significantly different from the null hypothesis of no relationship.

```
coef(house.crime)
```

```
## (Intercept)        crim
##  24.0331062  -0.4151903
```

```
# And the p-value:
summary(house.crime)$coefficients[2,4]
```

```
## [1] 1.173987e-19
```

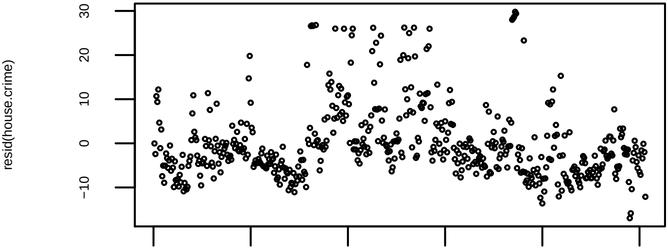
Regression

We also want to know if this is a good model. Let's look at the measure of goodness of fit. This should look familiar!

```
summary(house.crime)$r.squared
cor(Boston$crim, Boston$medv)^2
```

Let's also plot the residuals. You can also use plot(house.crime) for a lot of output:

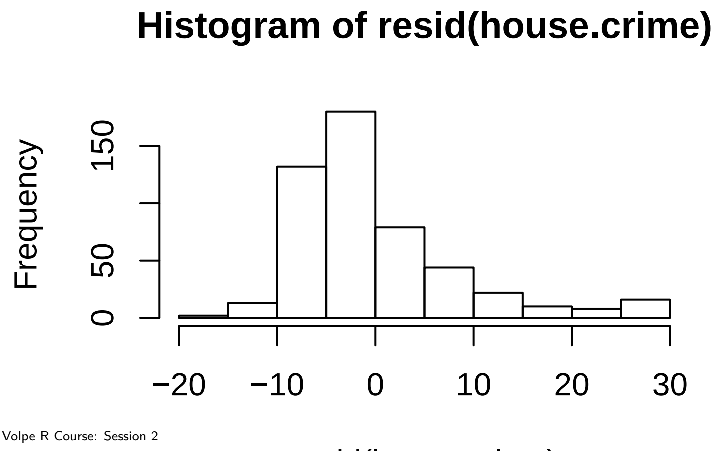
```
plot(resid(house.crime))
```



Regression

Those look acceptable. If we plot the model with `plot(house.crime)` we see the Q-Q plot. The Q-Q plot should mostly be straight, as it is. If we are very concerned with normality of residuals, you can log-transform the response (most common) or predictors. Note that the interpretation of the coefficients is then different.

```r
hist(resid(house.crime))
```



Regression

Look at how average NOx concentrations in an area affect median housing prices:

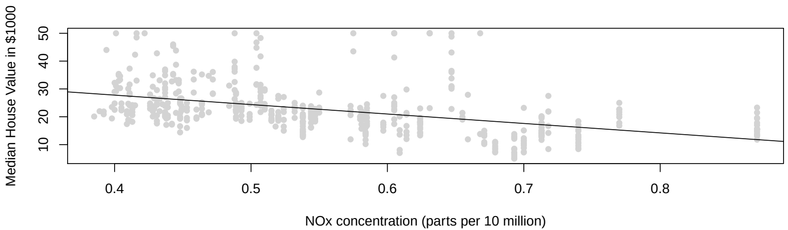
```r
nox.house <- lm(medv ~ nox, data = Boston)
summary(nox.house)
```

```
##
## Call:
## lm(formula = medv ~ nox, data = Boston)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -13.691  -5.121  -2.161   2.959  31.310
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.346      1.811   22.83   <2e-16 ***
## nox          -33.916      3.196  -10.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.323 on 504 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.181
## F-statistic: 112.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

Regression

Plot it:

```r
plot(medv ~ nox,
     pch = 16,
     col = "lightgrey",
     ylab = "Median House Value in $1000",
     xlab = "NOx concentration (parts per 10 million)",
     data = Boston)

# Add the trendline using "a B line" abline()
abline(nox.house)
```



Regression

We can use these coefficients to test scenarios. For example what if NOx concentrations are reduced by half?

1/2 x mean NOx concentration x slope of relationship + intercept.

```
half.nox.price <- 0.5 * mean(Boston$nox) *
  nox.house$coefficients[2] + nox.house$coefficients[1]
current.nox.price <- mean(Boston$medv)

# Print out on the console the expected housing prices
# under the half NOx scenario and 'current'
paste("$",1000*round(c(half.nox.price, current.nox.price), 2), sep="")
```

```
## [1] "$31940" "$22530"
```

Multiple regression

Multiple regression is deceptively easy in R – just add more variables! Let's conduct a multiple regression with both NOx concentration and mean # of rooms

```r
multiple1 <- lm(medv ~ nox + rm, data = Boston)
summary(multiple1)
```

```
##
## Call:
## lm(formula = medv ~ nox + rm, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.889  -3.287  -0.636   2.518  39.638
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.2059     3.3393  -5.452 7.82e-08 ***
## nox         -18.9706     2.5304  -7.497 2.97e-13 ***
## rm            8.1567     0.4173  19.546  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.281 on 503 degrees of freedom
## Multiple R-squared:  0.5354, Adjusted R-squared:  0.5336
## F-statistic: 289.9 on 2 and 503 DF,  p-value: < 2.2e-16
```

Multiple regression

Exercise 2.2:

Carry out a multiple regression for median house price and two predictor variables, using the Boston data set. See `?Boston` for a description of the variables. See if you can get a better model than the `nox` + `rm` model above.

Exercise 2.2

How do we compare models? If we have the same response data in each model, we can compare using an index called AIC, Akaike Information Criterion. Of course you can look at R^2 as well, but there are pitfalls with only looking at R^2.

```
summary(multiple2 <- lm(medv ~ age + dis, data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ age + dis, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.661  -5.145  -1.900   2.173  31.114
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.3982     2.2991  14.526  < 2e-16 ***
## age          -0.1409     0.0203  -6.941  1.2e-11 ***
## dis          -0.3170     0.2714  -1.168    0.243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.524 on 503 degrees of freedom
## Multiple R-squared:  0.1444, Adjusted R-squared:  0.141
## F-statistic: 42.45 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
AIC(multiple1, multiple2)
```

T-test: compare means of two groups

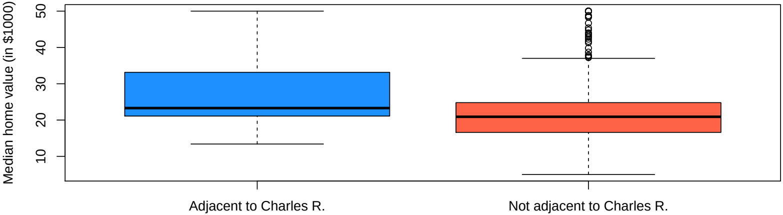
Comparing two groups is done in R with the function `t.test`. This tests if two groups have the same mean value. Here we are looking at housing prices for houses adjacent to the Charles river (`chas == 1`) or not:

```r
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.

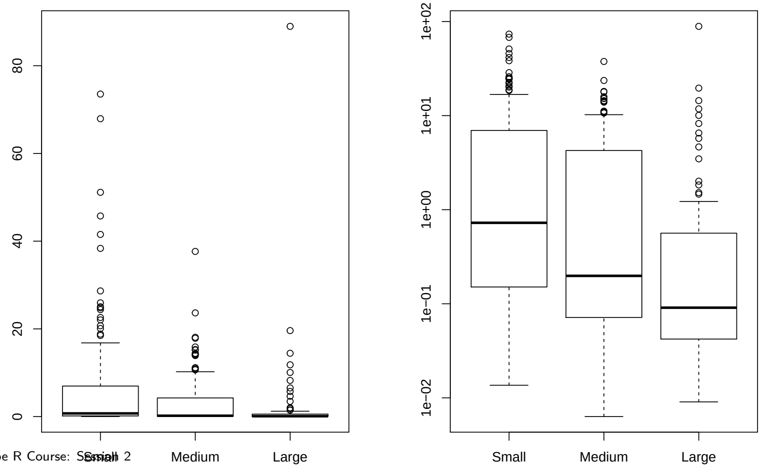
```r
# create a three-category variable from the number of rooms per house

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!

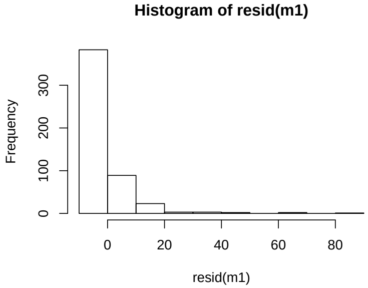
```r
par(mfrow = c(1, 2))
boxplot(crim ~ house.size, data = Boston)
boxplot(crim ~ house.size, data = Boston, log = "y")
```

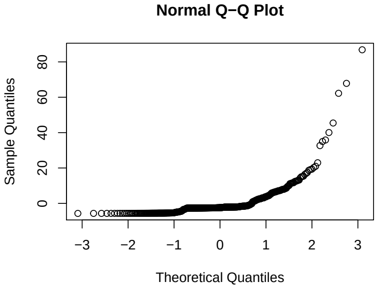


Analysis of Variance (ANOVA): compare means of more than two groups

Do we need to transform the data? As for the regression 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!
```



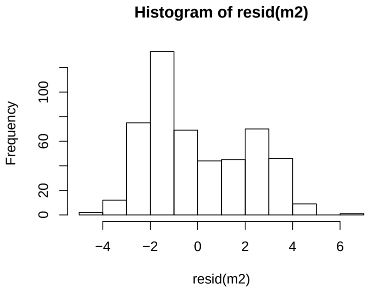


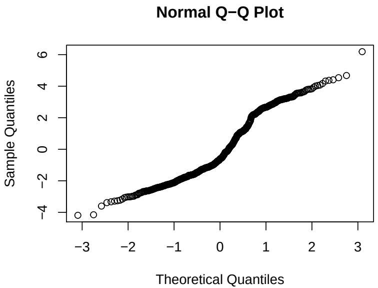
Analysis of Variance (ANOVA): compare means of more than two groups

```
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!
```





Homework

Take your dataset you used for Homework 1, or another dataset if you decide to change, and carry out at least two statistical tests.

Document your work in a script as before, and upload to the Homework 2 folder on the course Google Drive.

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