Statistics

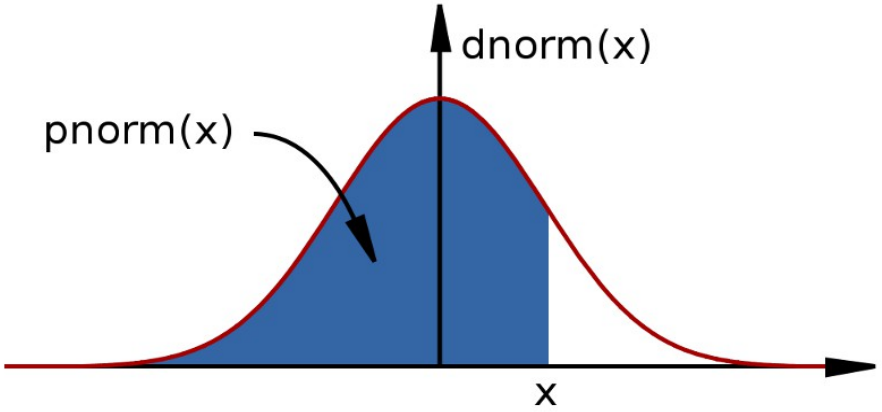
5

Built-in support for statistics

- R is a statistical programming language
  - Classical statistical tests are built-in
  - Statistical modeling functions are built-in
  - Regression analysis is fully supported
  - Additional mathematical packages are available
    - MASS, Waves, sparse matrices, etc

Pseudo-random numbers and distributions

- mostly commonly used distributions are built-in, functions have stereotypical names, e.g. for normal distribution:
  - pnorm - cummulative distribution for x
  - qnorm - inverse of pnorm (from probability gives x)
  - dnorm - distribution density
  - rnorm - random number from normal distribution



- available for variety of distributions: punif (uniform), pbinom (binomial), pnbinom (negative binomial), ppois (poisson), pgeom (geometric), phyper (hyper-geometric), pt (T distribution), pf (F distribution) ...

Two sample tests
Basic data analysis

- Comparing 2 variances
  - Fisher's F test

`var.test()`

- Comparing 2 sample means with normal errors
  - Student's t test

`t.test()`

- Comparing 2 means with non-normal errors
  - Wilcoxon's rank test

`wilcox.test()`

- Comparing 2 proportions
  - Binomial test

`prop.test()`

- Correlating 2 variables
  - Pearson's / Spearman's rank correlation

`cor.test()`

- Testing for independence of 2 variables in a contingency table
  - Chi-squared

`chisq.test()`

  - Fisher's exact test

`fisher.test()`

Comparison of 2 data sets example
Basic data analysis

- Men, on average, are taller than women.
  - The steps
    1. Determine whether variances in each data series are different
       - Variance is a measure of sampling dispersion, a first estimate in determining the degree of difference
         - *Fisher's F test*
    2. Comparison of the mean heights.
       - Determine probability that mean heights really are drawn from different sample populations
         - *Student's t test, Wilcoxon's rank sum test*
    3. Review significance of finding
       - What's the likelihood of getting our t statistic?
         - *What's the critical t value?*

1. Comparison of 2 data sets
Fisher's F test

- Read in the data file into a new object, heightData

```
heightData<-read.csv("10_heightData.csv",header=T)
```

- Do the two sexes have the same variance?

```
var.test(heightData$Female,heightData$Male)
```

```
F test to compare two variances

data:  heightData$Female and heightData$Male
F = 1.0073, num df = 99, denom df = 99, p-value = 0.9714
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.6777266 1.4970241
sample estimates:
ratio of variances
           1.00726
```

2. Comparison of 2 data sets
Student's t test

- Student's t test is appropriate for comparing the difference in mean height in our data.
  - Remember a t test = $\dfrac{\text{difference in two sample means}}{\text{standard error of the difference of the means}}$

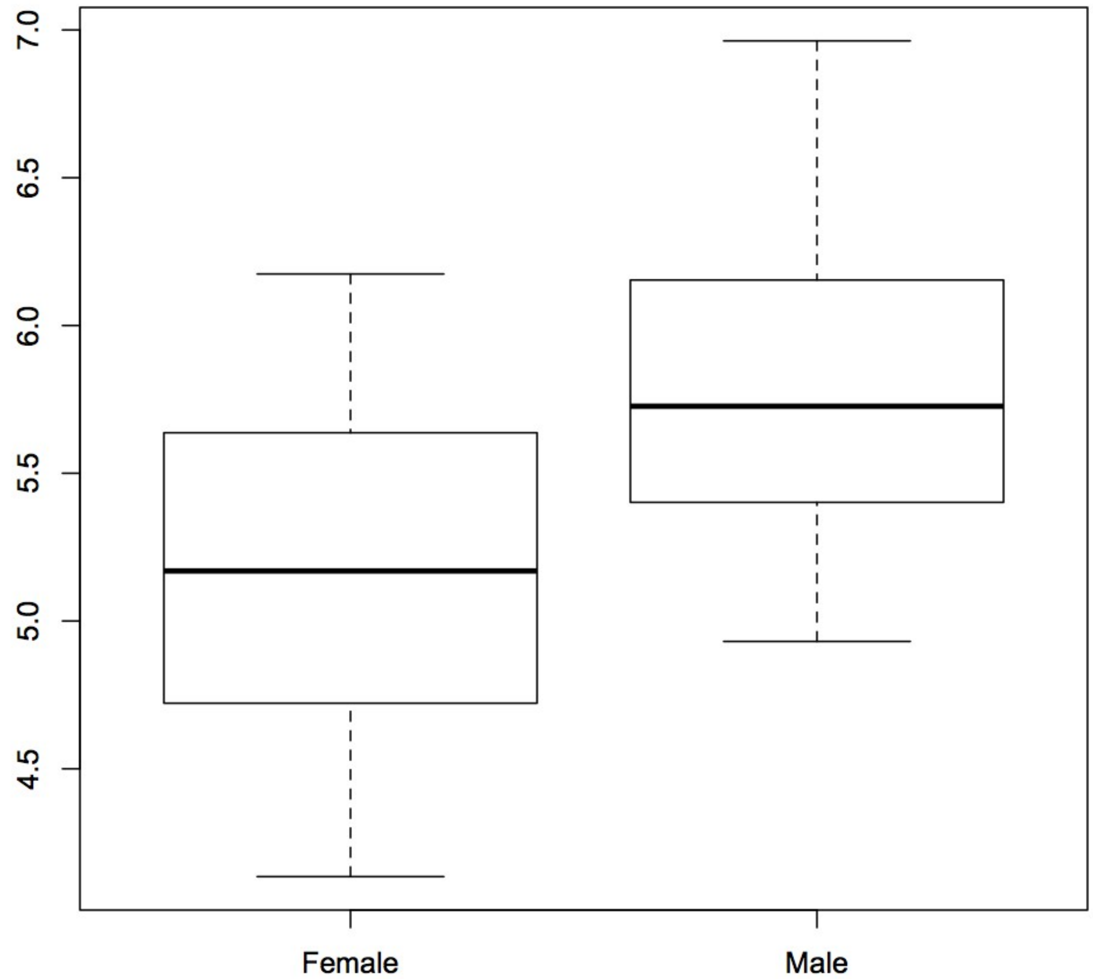
```
t.test(heightData$Female,heightData$Male)
```

```
Welch Two Sample t-test

data:  heightData$Female and heightData$Male
t = -8.4508, df = 197.997, p-value = 6.217e-15
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.7788497 -0.4841288
sample estimates:
mean of x mean of y
 5.168725  5.800214
```

3. Comparison of 2 data sets
Review findings

```
> boxplot(heightData)
```



Linear regression
Basic data analysis

- Linear modeling is supported by the function `lm()`
  - `example(lm)` # the output assumes you know a fair bit about the subject
- lm is really useful for plotting lines of best fit to XY data in order to determine, intercept, gradient & Pearson's correlation coefficient
  - This is very easy in R
- Three steps to plotting with a best fit line
  - Plot XY scatter-plot data
  - Fit a linear model
  - Add bestfit line data to plot with abline() function

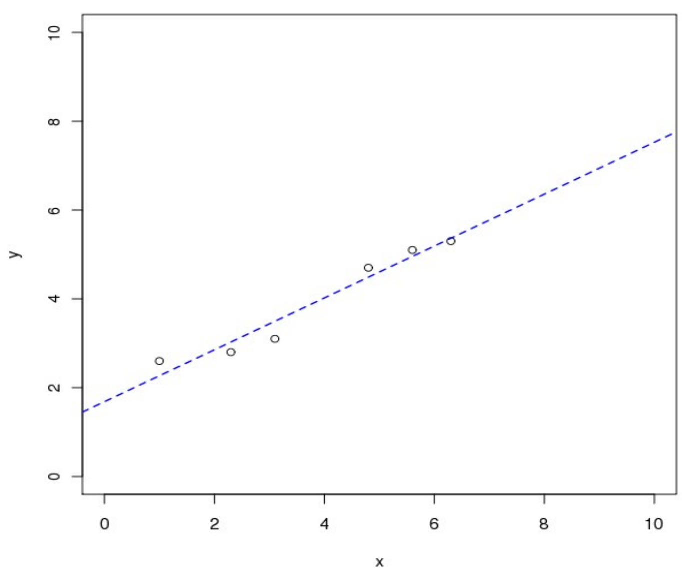
Typical linear regression analysis
Basic data analysis

X	Y
1.0	2.6
2.3	2.8
3.1	3.1
4.8	4.7
5.6	5.1
6.3	5.3

```
> x<-c(1, 2.3, 3.1, 4.8, 5.6, 6.3)
> y<-c(2.6, 2.8, 3.1, 4.7, 5.1, 5.3)
> plot(y~x, xlim=c(0,10),ylim=c(0,10))
```

Note formula notation
(y is given by x)

```
> myModel<-lm(y~x)
> abline(myModel,lty=2,lwd=1.5,col="blue")
```



Get the coefficients of the fit from:
`summary.lm(myModel)` **and**
`coef(myModel)`
`resid(myModel)`
`fitted(myModel)`

Get QC of fit from
`plot(myModel)`

Find out about the fit data from
`names(myModel)`

The linear model object
Basic data analysis

- Summary data describing the linear fit is given by
  - summary.lm(myModel)

```
> summary.lm(myModel)

Call:
lm(formula = y ~ x)

Residuals:
       1        2        3        4        5        6
 0.33159 -0.22785 -0.39520  0.21169  0.14434 -0.06458

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.68422    0.29056   5.796   0.0044 **
x            0.58418    0.06786   8.608   0.0010 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3114 on 4 degrees of freedom
Multiple R-squared: 0.9488,   Adjusted R-squared: 0.936
F-statistic:  74.1 on 1 and 4 DF,  p-value: 0.001001
```

Y intercept

Gradient

Good fit: would happen 1 in 1000 by chance

R^2 , with pValue

Exercise
Work through the previous tests

You should:

1) Undertake the variance and t.test 'height exercise'
2) Make sure you are able to understand the F and t statistics
3) Generate the simple box plot
4) Access the help and arguments information for each function used

help("t.test")

Shortcut ... ?t.test

args(t.test)