R Module Day 2: Statistics

Drew Allen

Topics Covered

- Statistical Distributions
- Summary Statistics
- T tests
- Regression (simple linear, multiple linear)
- Analysis of Variance

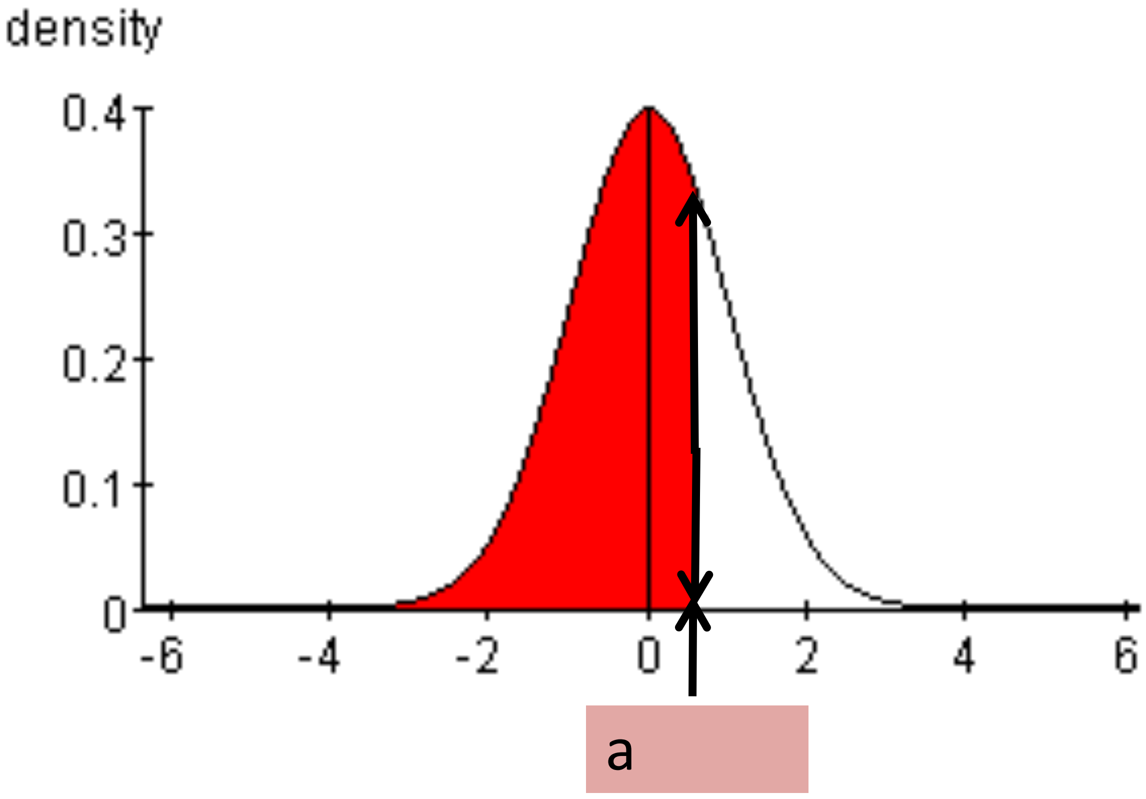
Statistical Distributions

Some Basic Definitions

- **Random Variable** – a variable whose value is not known with certainty
- **Random Variate** – particular outcome of a random variable
- **Probability** -- denotes the *relative frequency of occurrence* of a particular value
- **Probability distribution** yields the probability of
  - Each value of a random variable (**discrete distribution**)
  - the value of a random falling within a particular interval (**continuous distribution**)

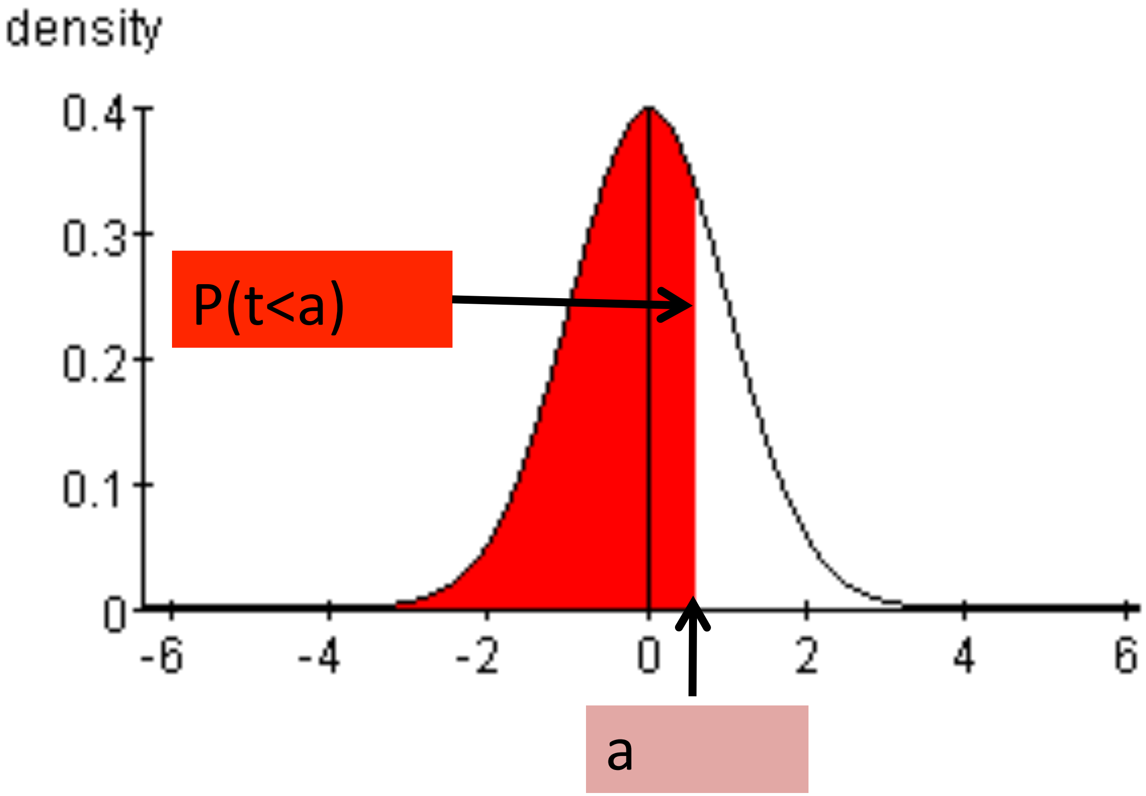
Probability density (i.e. height) at a

```
dnorm(a,mean=0,sd=1)
```



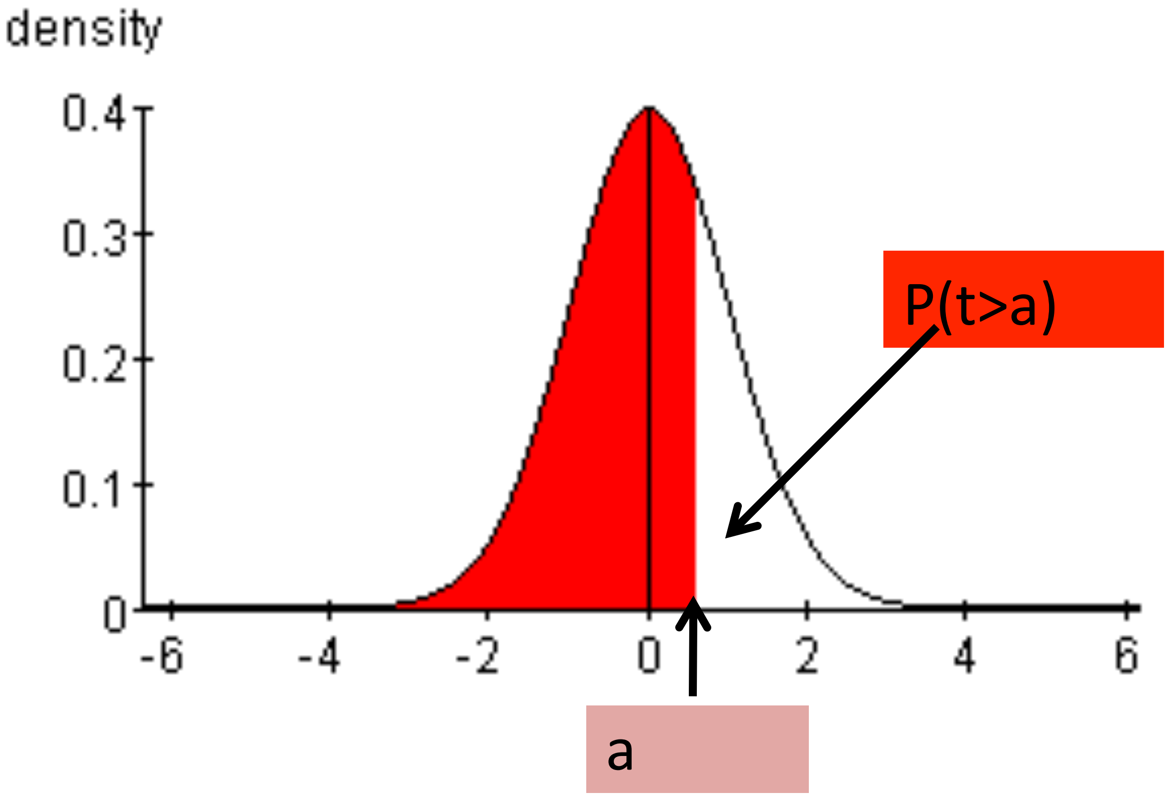
Probabilities from -∞ to a

```
pnorm(a,mean=0,sd=1,lower.tail=TRUE)
```



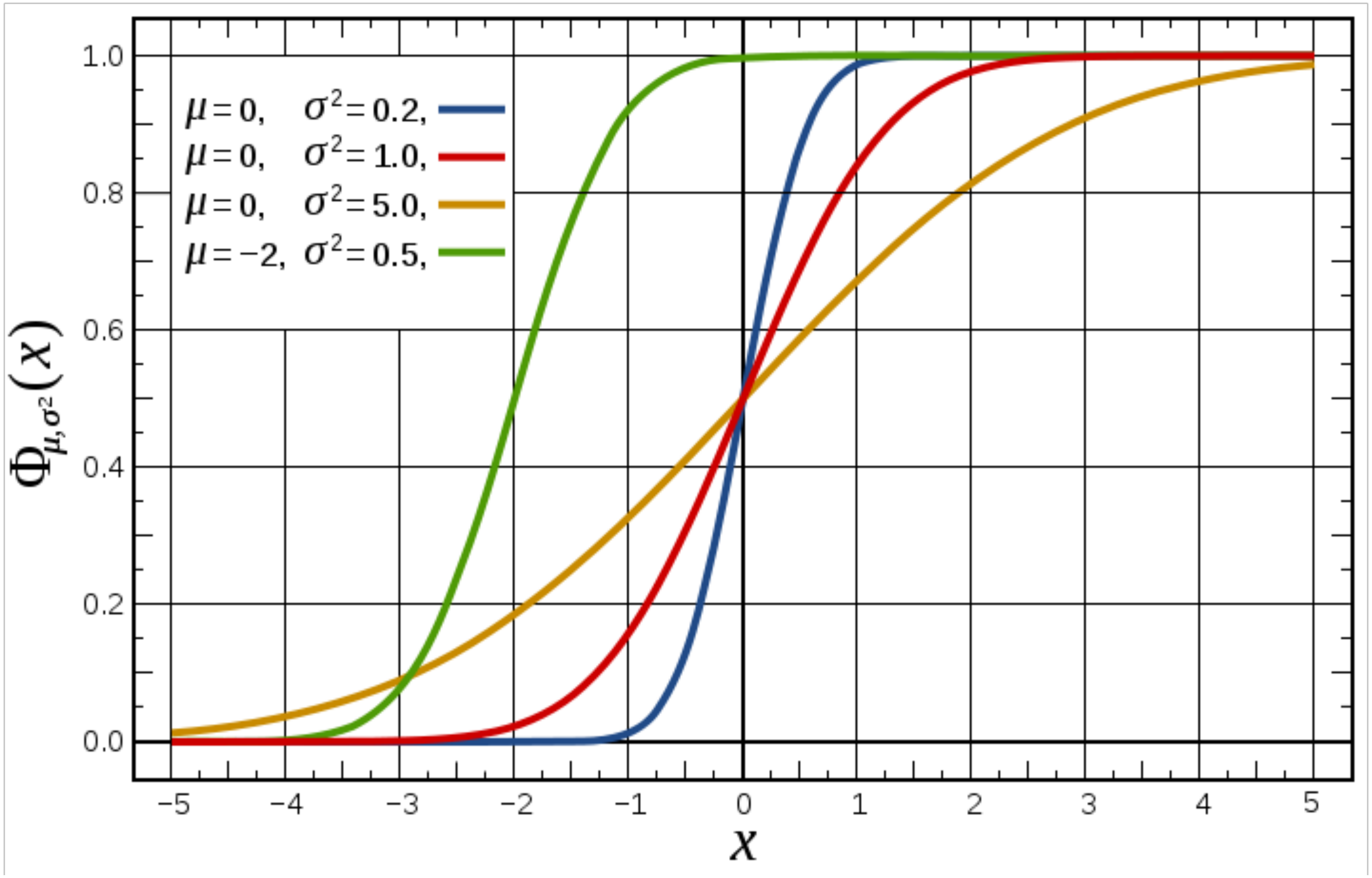
Probabilities from a to ∞

```
pnorm(a,mean=0,sd=1,lower.tail=FALSE)
```



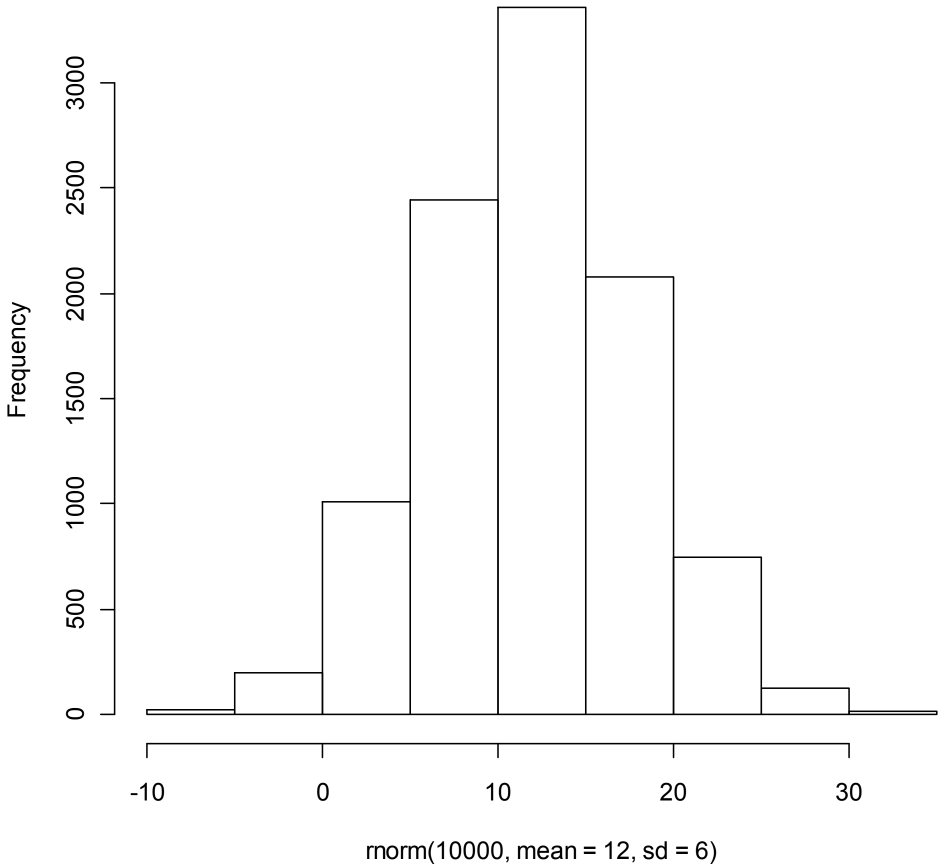
Probabilities from -∞ to a

```
qnorm(0.4,mean=-2,sd=sqrt(0.5))
```



Samples from a distribution

```
rnorm(1000,mean=12,sd=6)
```



Functions have required and optional arguments

- Works (no required arguments)
  - `q()`
- Doesn’t work:
  - `rnorm()`
- Does work (caution: computer assigns values for you some arguments!)
  - `rnorm(100)`
- Does work (all arguments specified by user)
  - `rnorm(100,mean=1,sd=4)`
  - `rnorm(mean=1,sd=4,n=100)`

Exercise 1: Using R as a Statistics Table

- Generate a sample of 1000 variates from a normal distribution of mean 10 and standard deviation 5 using `rnorm`
- For this sample, calculate what fraction of the points take values <5 (hint: use `length`)
- Using `pnorm`, calculate the theoretically predicted fraction of points that should take values < 5

Built-in Probability Distributions:

for a full list, type `?Distributions`

Continuous distributions

- Normal
- t
- Chi-squared
- F
- Exponential
- Uniform
- Beta
- Cauchy
- Logistic
- Lognormal
- Gamma
- Weibull

Discrete distributions

- Binomial
- Poisson
- Geometric
- Hypergeometric
- Negative binomial

Other Distributions Use Similar Syntax

NORMAL DISTRIBUTION

- dnorm(x, mean = 0, sd = 1, log = FALSE)
- pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
- qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
- rnorm(n, mean = 0, sd = 1)

UNIFORM DISTRIBUTION

- dunif(x, min=0, max=1, log = FALSE)
- punif(q, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
- qunif(p, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
- runif(n, min=0, max=1)

Exercise 2:

Using R as a Statistics Table

- What is the probability that a variate picked at random from gamma distribution with a shape of 3 and scale of 1 is < 0.68? [use `pgamma`]
- What is the probability that a variate selected at random from an exponential distribution with rate of 1 lies between 0.1 and 10? [use `pexp`]

Statistical distributions provide a means to perform simulations

- #using r for simulation of 1D random walker
- steps<-rnorm(n=10000,mean=0,sd=1)
- distance.from.origin <- cumsum(steps)
- plot(distance.from.origin,type='l')

Summary Statistics

Some Functions for Calculating Summary Statistics

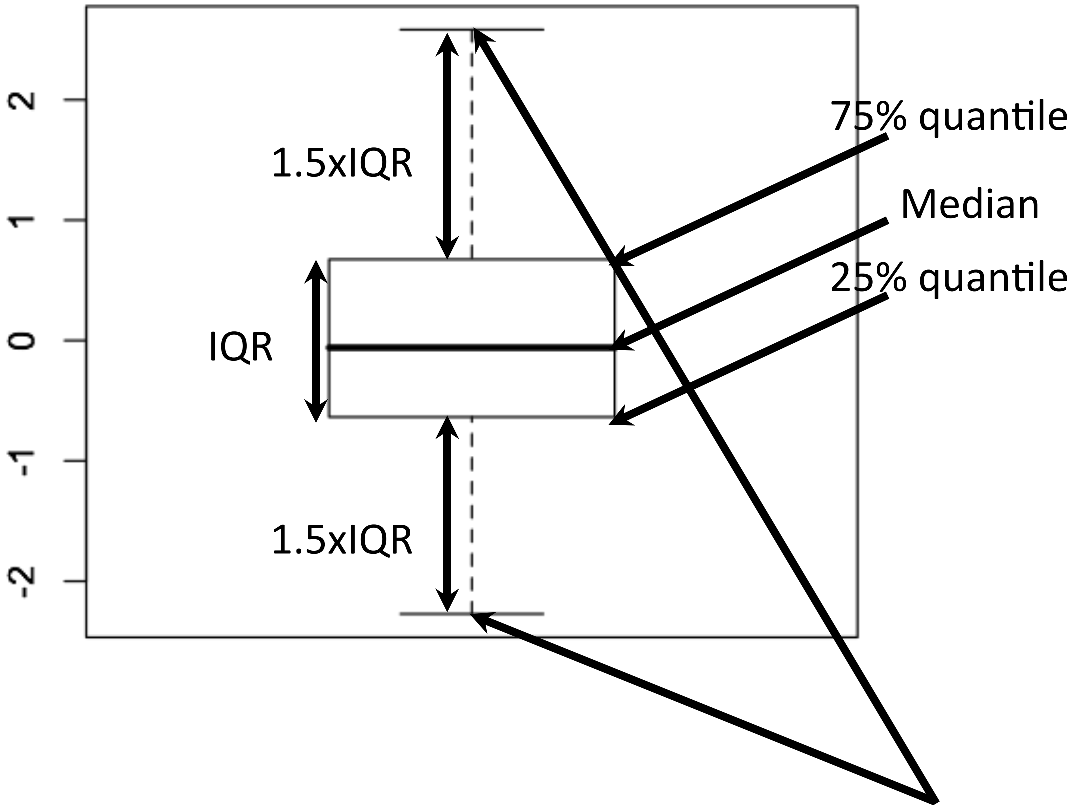
- Minimum: `min()`
- Maximum: `max()`
- Range (Minimum and Maximum): `range()`
- Mean: `mean()`
- Median: `median()`
- Quantiles: `quantile()`
- Interquartile range: `IQR()`
- Variance: `var()`
- Standard Deviation: `sd()`
- Summary: `summary()`
- Stem & Leaf Plot: `stem()`

- Boxplot: `boxplot()`
- QQ Plot: qqnorm(), qqline()

Functions for Calculating Summary Statistics

```
>x<-rnorm(100)

>boxplot(x)
```



1.5xIQR

IQR

1.5xIQR

75% quantile

Median

25% quantile

IQR= 75% quantile -25% quantile= Inter Quantile Range

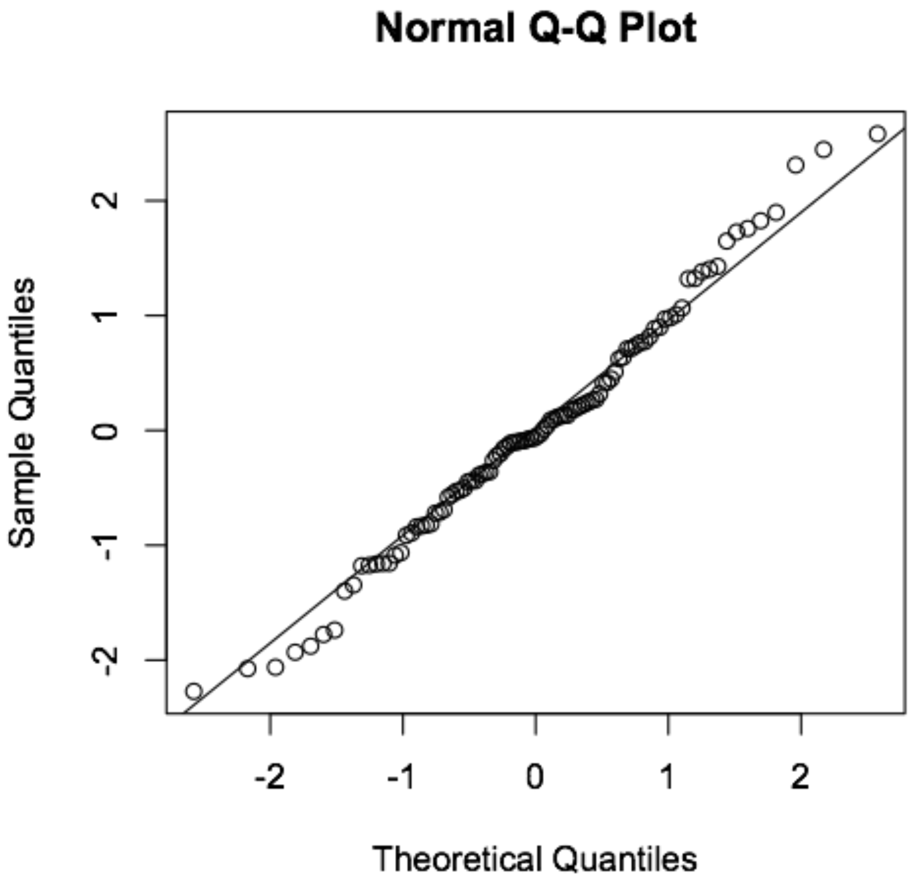
Everything above or below are considered outliers

QQ Plot

- Many statistical methods make some assumption about the distribution of the data (e.g. Normal)
- The quantile-quantile plot provides a way to visually verify such assumptions
- The QQ-plot shows the theoretical quantiles versus the empirical quantiles. If the distribution assumed (theoretical one) is indeed the correct one, we should observe a straight line.

QQ Plot

- `x<-rnorm(100)`
- `qqnorm(x)`
- `qqline(x)`



Functions for Calculating Summary Statistics

- Two functions are extremely useful for calculating summary statistics for subsets of data:
  - `apply()` (calculates function on a column-by – column or row-by-row basis)
  - `tapply()` (groups data in one column based on values in another column)

- Example Script:
  - `summary_statistics.R`

T test

What does Student's t distribution have to do with Guinness Stout?

VOLUME VI MARCH, 1908 No. 1

BIOMETRIKA.

THE PROBABLE ERROR OF A MEAN.

BY STUDENT.

Introduction.

ANY experiment may be regarded as forming an individual of a "population"

T distribution

- The t distribution was introduced by William Gosset, a chemist working for Guinness brewery in Ireland
- He published his work under the pen name "Student" because Guinness regarded the fact that they were using statistics to help with brewing to be a trade secret

T test Example:
Darwin’s Plant Growth Data

- Data are from Darwin's study of cross- and self-fertilization.
- Pairs of seedlings of the same age, one produced by cross-fertilization and the other by self-fertilization, were grown together so that the members of each pair were reared under nearly identical conditions.
- The data are the final heights of each plant after a fixed period of time, in inches.
- Darwin consulted the famous 19th century statistician Francis Galton about the analysis of these data

- Please download the following files:
  - `binary.csv`
  - `gala.txt`
  - `darwin.txt`

Exercise 3:

Darwin's Plant Growth Data

- Import `darwin.txt`
- Conduct a paired T test using the function `t.test()`
  - Type `?t.test` for some help
- Answer the following questions:
  - What is the mean difference, *m*, between the treatments?
  - What is the standard deviation, *s*, of the paired differences?
  - According to the t test, is the difference significant at the P = 0.05 level for the two-tailed test?
  - According to the non-parametric analogue of the t test (Mann-Whitney U), is the difference significant at the P = 0.05 level for the two-tailed test? **[Use `wilcox.test`]**

Exercise 3 Answers

- `m<-mean(darwin$crossfertilized-darwin$selffertilized)`
- `s<-sd(darwin$crossfertilized-darwin$selffertilized)`
- `t.test(darwin$crossfertilized,darwin$selffertilized,paired=TRUE)`
- `wilcox.test(darwin$crossfertilized,darwin$selffertilized,paired=TRUE)`

Mann-Whitney U Test

- This technique is non-parametric , meaning that they do not rely on assumptions that the data are drawn from a particlarly probability distribution.
- Non-parametric methods are particularly suited to data that are not normally distributed.
- Assumptions Mann-Whitney U Test include:
    - random samples from populations
    - independence within samples and mutual independence between samples
    - measurement scale is at least ordinal

Power Analysis

- **A very important part of planning research**

- **Power** is the conditional probability of rejecting the null hypothesis given that it is really false

- 1- Power = Type II error

Packages Allow You To Increase the Functionality of R

R has lots of statistical capabilities

- Full list of packages:
  - http://cran.r-project.org/web/packages/available_packages_by_name.html

- Task views are helpful:
  - http://cran.r-project.org/web/views/

Please add the following packages

- Please add the following packages
  - **`pwr:`** for performing power analysis

Exercise 4:
Darwin's Plant Growth Data

- Install the library `pwr`
- Calculate the estimated effect size as *d* = *m* / *s* for the `darwin.txt` data
- In the command window, learn how to conduct a power analysis using `?pwr.t.test`
- Using this function, calculate the statistical power of the test that Darwin conducted
- Now use this function to determine how large a sample size would be required to reject the null hypothesis at a significance level of 0.05 with 80% power

Answers

-
  ```
  m<-mean(darwin$crossfertilized-darwin$selffertilized)
  ```

-
  ```
  s<-sd(darwin$crossfertilized-darwin$selffertilized)
  ```

-
  ```
  pwr.t.test(n=16,d=m/s,sig.level=0.05,type='paired')
  ```

-
  ```
  pwr.t.test(d=m/s,sig.level=0.05,power=0.8,type='paired')
  ```

Linear Regression

Linear Regression

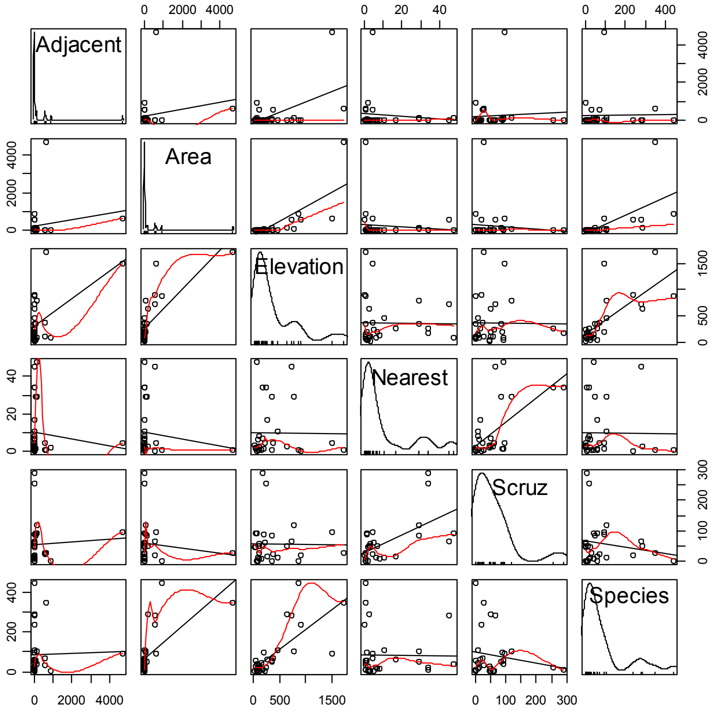
- Use `gala <- read.table(...,header=TRUE,row.names=1)` to import the dataset `gala`
- View the dataset

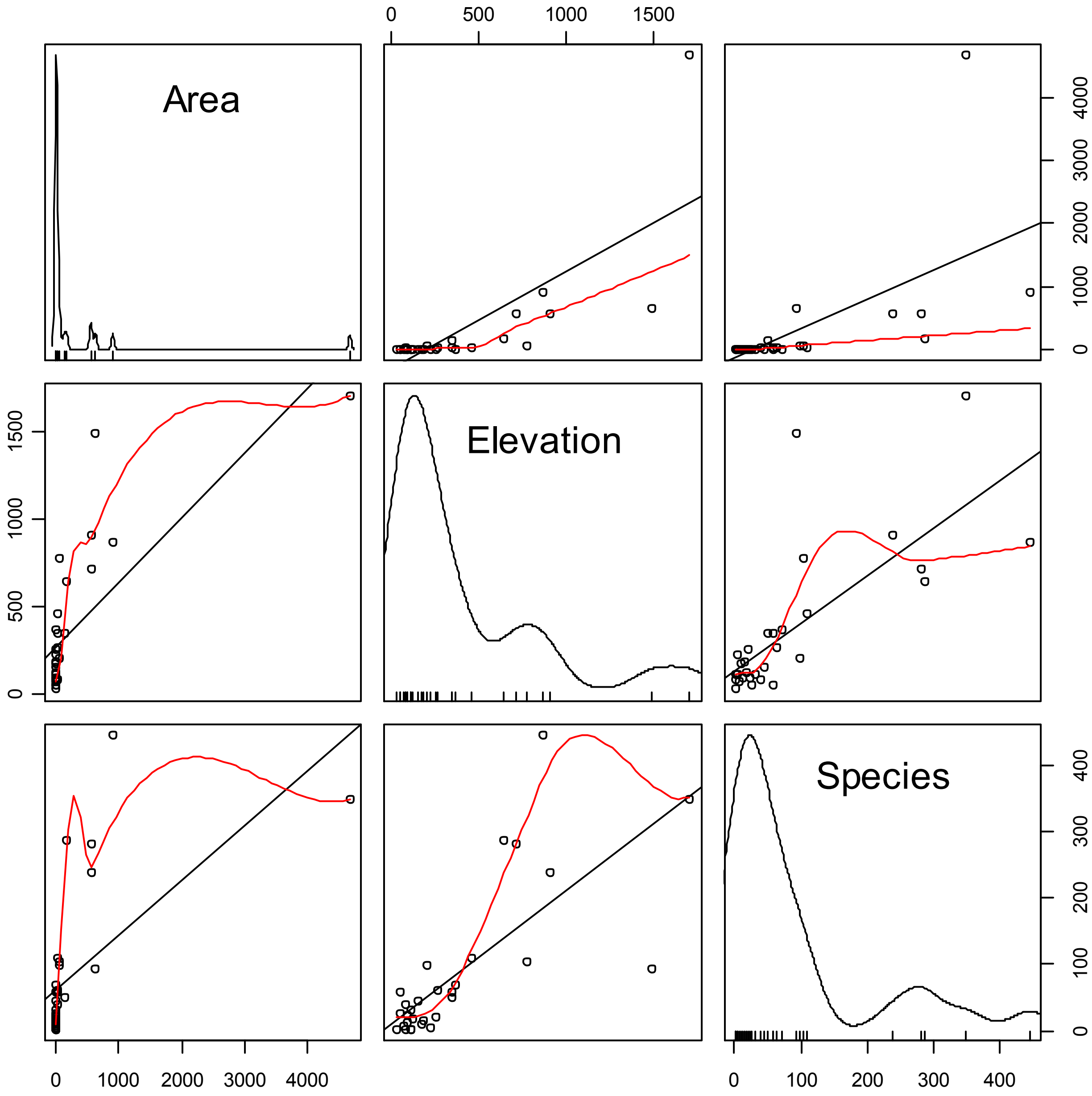
gala

- Source
  - M. P. Johnson and P. H. Raven (1973) "Species number and endemism: The Galapagos Archipelago revisited" Science, 179, 893-895
- Variables
  - **Species** the number of plant species found on the island
  - **Endemics** the number of endemic species
  - **Area** the area of the island (km^2)
  - **Elevation** the highest elevation of the island (m)
  - **Nearest** the distance from the nearest island (km)
  - **Scruz** the distance from Santa Cruz island (km)
  - **Adjacent** the area of the adjacent island (square km)

Investigate Distributions of Variables and Their Relationships

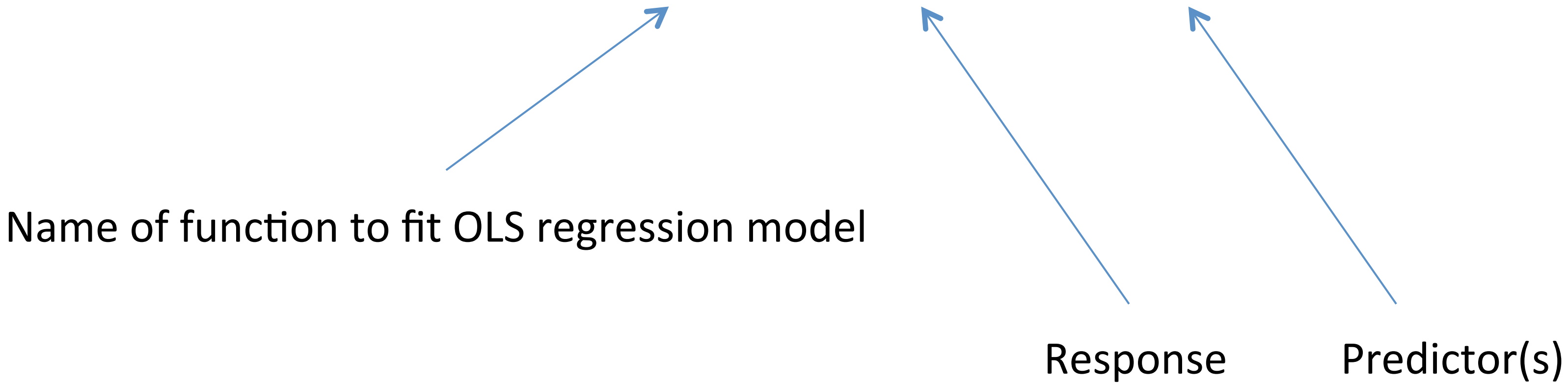
- Generate a plot similar to the one below by typing `plot(gala)`




Area
Elevation
Species
0
500
1000
1500
0
1000
2000
3000
4000
0
500
1000
1500
0
1000
2000
3000
4000
0
100
200
300
400
0
100
200
300
400

Ignore these issues and fit a linear model

- Now fit a linear regression model by typing:
  - `gala.model<-lm(Species~Area, data=gala)`



Name of function to fit OLS regression model

Response

Predictor(s)

- Let's look at the attributes of this object:
  - `str(gala.model)`

Extractor functions allow you to get information on `lm` objects

- `coef(gala.model)`
- `residuals(gala.model)`
- `fitted.values(gala.model)`
- `cooks.distance(gala.model)`
- `summary(gala.model)`
- `anova(gala.model)`

Assumptions of Linear Regression

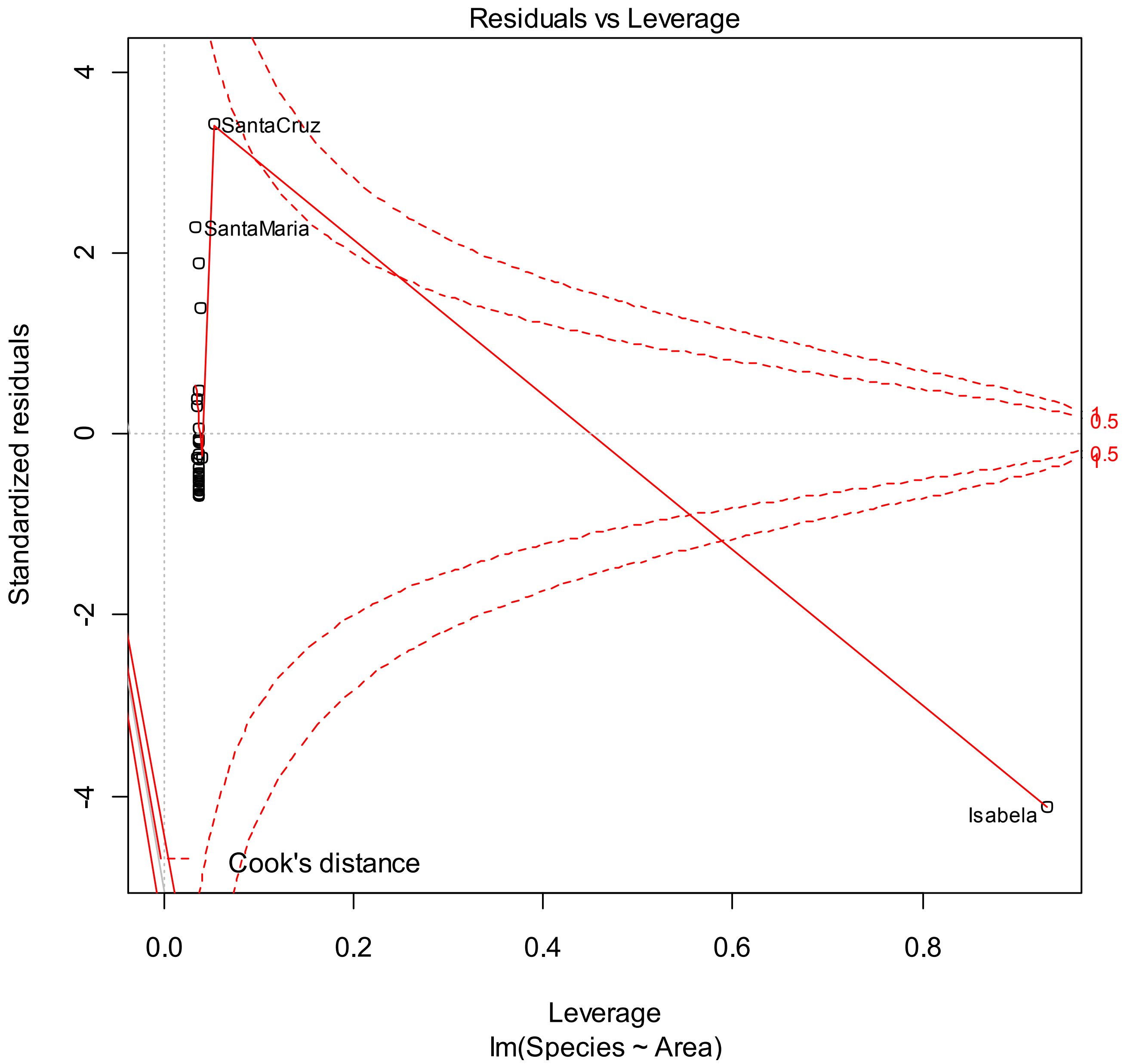
- **Linearity** of the relationship between dependent and independent variables
- **Independence** of the errors (no serial correlation)
- **homoscedasticity** (constant variance) of the errors
- **normality** of the error distribution

Let's evaluate these assumptions

- To evaluate assumptions type:
  - `plot(gala.model)`
- Theory:
  - Leverage is a measure of how far an independent variable deviates from its mean
  - Cook's distance
    - measures the influence of an observation on the overall model:

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j \ - \hat{Y}_{j(i)})^2}{p \ \mathrm{MSE}}.$$

    - Y_j is the prediction from the full regression model for observation *j*
    - $Y_{j\,(i)}$ is the prediction for observation *j* from a refitted regression model in which observation *i* has been omitted
  - As a rule of thumb, further consideration is given to points with distances $D_i > 4/n$


Residuals vs Leverage
Standardized residuals
Leverage
lm(Species ~ Area)
4
2
0
-2
-4
0.0
0.2
0.4
0.6
0.8
SantaCruz
SantaMaria
Isabela
Cook's distance
1
0.5
0.5
1

Exercise 5:

Independent analysis of `gala` data

- Transform species and area using the log10 transformation, e.g.
  - `gala$log.species<-log10(gala $Species)`
- Refit the linear model using the log transformed data and assess whether model assumptions are upheld
- Plot the data and model together using the functions `plot()` and `abline()`
- Inspect the coefficients using `summary()`

Fit of simple linear regression model

- `summary(gala.model)`

```
Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.26106    0.06822  18.484  < 2e-16 ***
log.area     0.38860    0.04160   9.342 4.23e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3406 on 28 degrees of freedom
Multiple R-squared: 0.7571,   Adjusted R-squared: 0.7484
F-statistic: 87.27 on 1 and 28 DF,  p-value: 4.23e-10
```

- 95% confidence interval for fitted slope:
  - lower CI: `0.38860 + qt(.025,28)* 0.04160`
  - Upper CI: `0.38860 - qt(.025,28)* 0.04160`
  - `confint(gala.model)`

Multiple linear regression

- Extending analyses to multiple linear regression is straightforward using `lm()`:
  - `lm(log.species~log.area+ log.elevation,data=gala)`
- Notation used for formulas:
  - Intercept only
    - `lm(y~1)`
  - Force-fit y versus x1 relationship through origin
    - `lm(y~x1-1)`
  - Include all variables in data.frame `gala`:
    - `lm(y~.,data=gala)`
  - x1, x2 and their interactions:
    - `lm(y~x1*x2,data=data)`
    - `lm(y~x1+x2+x1:x2,data=data)`

Formally testing effects of `log.elevation` after accounting for `log.area`

- Fit a new model that includes both `log.elevation` and `log.area`
- Null hypothesis: after account for the effects of area, elevation is not significant
- How do we test this null hypothesis?

- R knows what to do. Just type:
  - `anova(lm1,lm2)`

Automated Model Selection

- Several methods available:
  - Best subset selection
  - Stepwise selection

- Fit using multiple criteria:
  - Statistical significance [`logLik(lm1)-logLik(lm2)`]
  - AIC [`AIC(lm1) - AIC(lm2)`]
- Key issue: need to first specify a full model

- VERY controversial among statisticians due to multiple comparisons problem, but still useful for exploratory purposes

R Code for BE using `step()`

- Use R function `step`
- Need to define an *initial model* (the full model in this case, as produced by the R function lm) and a *scope* (a formula defining the full model)
- `ffa.lm = lm(ffa~., data=ffa.df)`
- `step(ffa.lm,direction=“backward”)`

Forward Selection (FS) using `step()`

- Start with a null model
- Fit all one-variable models in turn. Pick the model with the best AIC
- Then, fit all two variable models that contain the variable selected in 2. Pick the one for which the added variable gives the best AIC
- Continue in this way until adding further variables does not improve the AIC

R Code for FS using `step()`

- Use R function `step`
- As before, we need to define an *initial model* (the null model in this case and a *scope* (a formula defining the full model)
- `# R code: first make null model:`
- `ffa.lm = lm(ffa~., data=ffa.df)`
- `null.lm = lm(ffa~1, data=ffa.df)# then do FS`
- `step(null.lm, scope=formula(ffa.lm),`
- `   direction=“forward”)`

R Code Output (1 of 2)

```
> step(null.lm, scope=formula(ffa.lm),
direction="forward")
Start:  AIC=-49.16
ffa ~ 1

           Df Sum of Sq     RSS     AIC
+ weight    1   0.63906 0.91007 -57.799
+ age       1   0.20503 1.34410 -50.000
<none>                  1.54913 -49.161
+ skinfold  1   0.00145 1.54768 -47.179
```

Starts with constant term only

Results of all possible 1 (& 0) variable models. Pick weight (smallest AIC)

R Code Output (2 of 2)

```
Step:  AIC=-57.8
ffa ~ weight

           Df Sum of Sq     RSS     AIC
+ age       1  0.115900 0.79417 -58.524
<none>                  0.91007 -57.799
+ skinfold  1  0.007778 0.90230 -55.971

Step:  AIC= -58.52
 ffa ~ weight + age

           Df Sum of Sq     RSS     AIC
<none>                    0.794 -58.524
+ skinfold  1     0.003   0.791 -56.601
```

Exercise 6:

Choosing the best predictor of richness

- Using BE and function `step()`, determine the "best" model of species richness using the following potential predictors:
  - log.area
  - log.elevation
  - log.nearest
  - log.scruz [note: use log10(x+1) transform]
  - log.adjacent
- Recall:
  - `y.lm = lm(y~., data=data)`
  - `step(y.lm,direction="backward")`

Analysis of Variance/Covariance in R Three Issues

- Factor variable type:
  - http://www.ats.ucla.edu/stat/r/modules/factor_variables.htm
- Coding of factors:
  - http://www.ats.ucla.edu/stat/r/library/contrast_coding.htm
- Types of ANOVA:
  - http://goanna.cs.rmit.edu.au/~fscholer/anova.php

Factor Variable Type

- `ssize <- sample(0:2,40,replace=TRUE)`
- `ssize`
- `is.factor(ssize)`
- `ssize.f <- factor(ssize,labels=c('s','m', 'l'))`
- `is.factor(ssize.f)`
- `is.ordered(ssize.f)`
- `ssize.f <- factor(ssize,labels=c('s','m', 'l'),ordered=TRUE)`
- `is.ordered(ssize.f)`
- `ssize.f[41] <- 'x'`
- `levels(ssize.f) <- c('s','m','l','x')`
- `ssize.f[41] <- 'x'`

One-way ANOVA using `mtcars`

- ?mtcars

- summary(mtcars)

- str(mtcars)

Exercise 7: One-way ANOVA using `mtcars`

- Fit an lm model (`lm1`) that predicts mileage (`mpg`) based on the number of cylinders (`cyl`)
- Create a new variable (`cyl.f`) in the data.frame `mtcars` that treats the number of cylinders (`cyl`) as a factor variable
- Fit another lm model that predicts mileage based on (`lm2`)
- Compare the two models using `summary()`