

Introduction to R

Oliver Díaz & Yingbo Li

SAMSI-Duke

May 17, 2011



What is R?

- ▶ R is an interactive software package for statistical computing and graphics.
- ▶ Can be used as a programming language.
- ▶ Typically quite powerful and flexible.

Things to know

- Getting Started
- Vectors, sequences and matrices
- Data Manipulation
- Probability distributions in R
- Basic Statistics
- Input and Output from files
- Linear regression in R

Getting started

- ▶ To open R:
Click on a R icon or go to Start → All Programs → R → R 2.8.1
- ▶ To set up a working directory in R:
go to File → Change dir → the desired directory.
- ▶ To save a file in R:
go to File → Save Workspace → the desired directory → Save.
- ▶ To exit R
> `q()`
- ▶ For help with a specific function
> `help("function name")` # alternatively,
> `?function name`
- ▶ For help when you don't know the exact function, one can search by providing the key words
> `help.search("function name")`

Basic Syntax, vectors and matrices

```
# To create a vector and store it in the current directory
```

```
> v <- c(2, 7, 8, 1, 5); x <- c(1:5)
```

```
# To view the object v
```

```
> v
```

```
# element by element operations
```

```
> 2*v + (3*x + cos(v*x))* exp(-v)
```

```
# To create a sequence of numbers
```

```
> seq1 <- seq(1,3,length=5)
```

```
> seq2 <- seq(3,1,by=-0.5)
```

```
> seq3 <- rep(1:4,3)
```

```
# Let's create two 3-by-2 matrices
```

```
> A <- matrix(c(2,-1,4,-2,3,1),3,2) # by columns by default
```

```
> B <- matrix(c(2,-1,4,-2,3,1),3,2,byrow=T)
```

```
# Transpose of matrix A
```

```
> t(A) -> C
```

```
# Matrix multiplication
```

```
> A %*% C
```

```
# Matrix operations
```

```
> a <- A %*% C - B%*%t(B)
```

```
> y <- c(.4, -1); A %*% y -> b
```

```
# Solve equation  $a*x = b$ 
```

```
> solve(a,b)
```

To list the objects in the current directory

```
> ls()
```

To remove an object from the directory

```
> rm(x)
```

To remove everything in the working environment

```
> rm(list=ls())
```

Data Manipulation

```
# Let's create a vector x and display it
```

```
> x <- c(4, 3, 8, 1, 6); x
```

```
# To provide all elements in x except for the third one
```

```
> x[-3]
```

```
# To produce all numbers in the vector x whose indices are at least 2
```

```
> x[x>=2]
```

```
# To view vector x in reverse order
```

```
> rev(x)
```

```
# To view the first and third elements of x
```

```
> x[c(1,3)] # note that x[1,3] will fail
```

```
# To order x from the smallest to the largest (default)
```

```
> sort(x) # Try sort(x, decreasing=T)
```

Let's create a 3x2 matrix A

```
> A <- matrix(c(2,9,4,6,3,0),3,2)
```

```
# To view the dimension of A
```

```
> dim(A)
```

```
# To view the first and the third rows of A
```

```
> A[c(1,3),]
```

```
# To view the second column of A
```

```
> A[,2]
```

```
# To view the element in the 2nd row and first column of A
```

```
> A[2,1]
```

Probability distributions in R

In R one can calculate

- density or point probabilities (prefix `d`);
- cumulated probability or distribution functions (prefix `p`)

$$F(x) = \mathbb{P}[X \leq x];$$

- quantiles (prefix `q`)

$$z_q = \inf\{z : \mathbb{P}[X \leq z] \geq q\};$$

- pseudo-random numbers (prefix `r`).

Let's sample six uniformly distributed random numbers on $[0, 2]$

```
> y <- runif(6,0, 2)
```

Let's evaluate the distribution function at different points 'x' of a normal random variable whose mean and variance are 1.5 and 2^2 respectively.

```
> x<-seq(-2.5,5.5,0.2); pnorm(x,mean=1.5,sd=2)->P
```

```
> plot(x,P,'l') # plots x vs. P(x)
```

To find the 95% quantile of the Cauchy distribution with location 0 and scale 2

```
> qcauchy(.95,0,2)
```

To evaluate the density function of the standard normal probability law

```
> x <- seq(-4,4,by=.1); dP <- dnorm(x);
```

```
> plot(x,dP,'l') # To plot data points connected by lines
```

To randomly sample ten times from of a fair coin.

```
> coin<-c('Head', 'Tail')  
> sample(coin,10,replace=T)
```

To randomly sample ten times from a biased dice favoring six.

```
> dice <- 1:6; prob<-c(rep(.1, 5),.5)  
> sample(dice,10, replace=T, prob)
```

Let's create a sample from exponential distribution with mean 1

```
> x <- rexp(100,1) # 100 samples of exp(1)
```

To find the mean, variance, standard deviation, minimum, maximum and median of the sample x

```
> mean(x) # mean of sample x
> var(x) # variance of sample x
> sd(x) # standard deviation of x
> max(x) # maximum value of x
> min(x) # minimum value of x
> median(x) # median of x. Try quantile(x,.5)
```

To plot the histogram of the sample x

> hist(x)

The empirical cumulative distribution of a numerical sample X_n , $1 \leq n \leq N$ is given by

$$F_n(x) = \frac{1}{n} \sum_{n=1}^N \mathbf{1}_{(-\infty, x]}(X_n)$$

To plot the empirical cumulative distribution function of x

> plot(ecdf(x), do.points=F, verticals=T)

To find the value of the empirical distribution at say 2.3

> length(x[x<= 2.3])/length(x)

Basic Multidimensional Statistics

Let's create three sets of data

```
> x <- runif(5,0,1)
> y <- 2*x+ rnorm(5,0,1)
> z <- x-y+ rt(5,5get)
```

To compute the covariance and correlation of x and z

```
> cov(x,z)
> cor(x,z)
```

To compute the covariance matrix of (x, y, z)

```
> D <- cbind(x,y,z) # creates a matrix with columns x, y z
> cov(D) # produces the covariance matrix of x, y, z
> cov(D[,c(1,2)]) # covariance matrix of (x,y)
```

Input and Output from/to Files

- Input tables

- For most file types, e.g. .txt:

```
read.table(file, header = TRUE, sep = " ");
```

- For .csv files, i.e. "comma separated values"

```
read.csv(file, header = TRUE, sep = ",");
```

- Output data

- Save data matrix as files, e.g. .txt:

```
write.table(x, file, col.names = TRUE, row.names=TRUE);
```

- Save data matrix as .csv files:

```
write.csv(x, file, col.names = TRUE, row.names=TRUE);
```

Dataset: Used Car Prices

```
mercedes=read.table(file="mercedes.txt",header=TRUE);
```

Data columns:

- 1 Case number 1,2, ..., n=54
- 2 Asking price in pounds (NB: Note the "rounding" up to ..995, etc!)
- 3 Type/Model of car 0=model 500, 1=450, 2=380, 3=280, 4=200
- 4 Age of car in six-month units (based on advertised registration date)
- 5 Recorded mileage (in thousands of miles)
- 6 Vendor (0,1,2,3 are different dealerships, 4 means sale by owner)

Exploratory data analysis:

```
> mercedes[1:5,];
```

	Case	Price	Mod	Age	Mile	Vend
1	1	30495	0	1	7	0
2	2	22250	0	3	16	0
3	3	23995	0	3	8	1
4	4	18495	0	3	15	2
5	5	20950	0	2	26	4

```
> dim(mercedes);
```

```
[1] 54 6
```

```
> summary(mercedes$Price); # $ visit variable Price  
# in file mercedes
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4950	10560	15870	15060	18000	30500

```
> summary(mercedes);
```

Select the Mercedes Model 500 (Mod = 0), and output the sub-dataset.

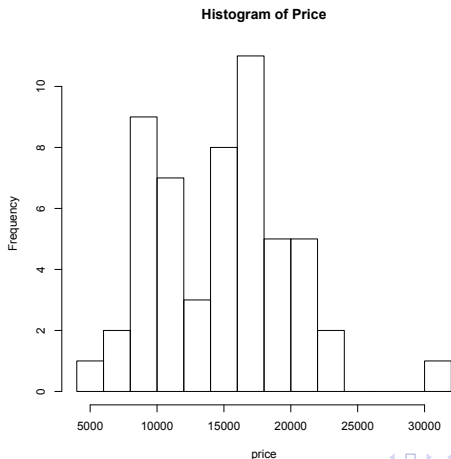
```
> mercedes500=mercedes[mercedes$Mod==0,];  
# $ visit variable Mod in mercedes
```

	Case	Price	Mod	Age	Mile	Vend
1	1	30495	0	1	7	0
2	2	22250	0	3	16	0
3	3	23995	0	3	8	1
4	4	18495	0	3	15	2
5	5	20950	0	2	26	4
6	6	21500	0	3	18	4
7	7	19995	0	5	24	0
8	8	18950	0	5	20	3

```
> write.csv(mercedes500,"mercedes500.csv",row.names=FALSE);
```

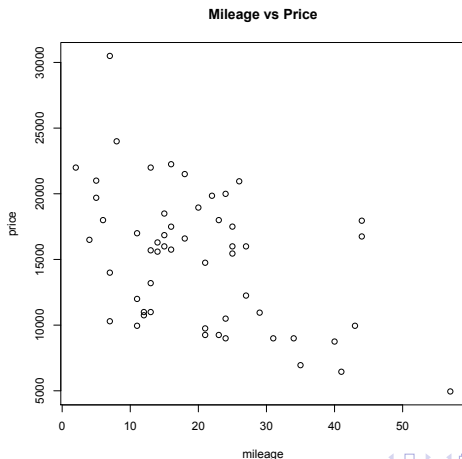
Histogram

```
> hist(mercedes$Price,breaks=10,xlab="price",  
+ main="Histogram of Price");
```



Scatter Plot: Mileage vs Price

```
> plot(mercedes$Mile, mercedes$Price, xlab="mileage",  
+ ylab="price", main="Mileage vs Price");
```



Regression

Regress price on mileage:

```
> car.lm=lm(Price~Mile, data=mercedes);
```

```
# or
```

```
> car.lm=lm(mercedes$Price~mercedes$Mile);
```

Regression Summary

```
> summary(car.lm);
```

Call:

```
lm(formula = Price ~ Mile, data = mercedes)
```

Residuals:

Min	1Q	Median	3Q	Max
-7541.3	-3807.6	-160.6	3020.1	12658.7

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	19302.0	1235.3	15.625	< 2e-16	***
Mile	-209.4	52.8	-3.966	0.000225	***

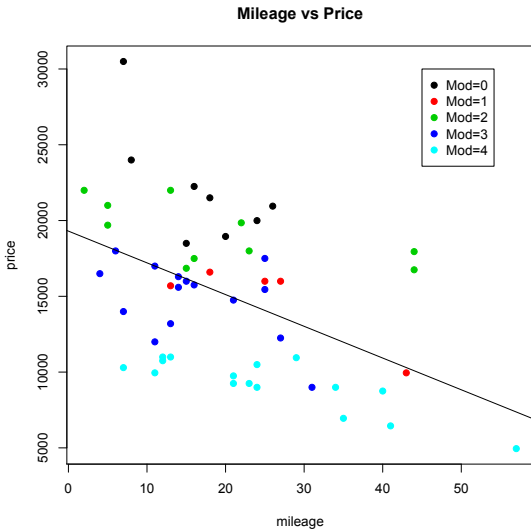
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4554 on 52 degrees of freedom

Multiple R-squared: 0.2322, Adjusted R-squared: 0.2175

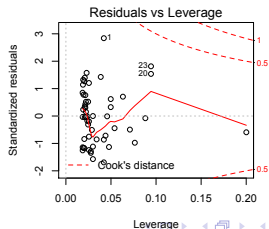
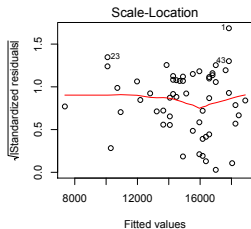
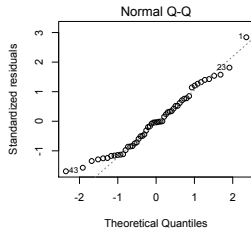
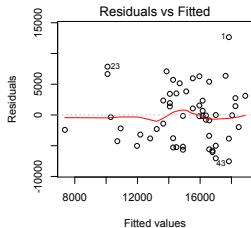
F-statistic: 15.73 on 1 and 52 DF, p-value: 0.0002247

```
> abline(car.lm$coef);
```



Diagnostic plots:

```
> par(mfrow=c(2,2));  
> plot(car.lm);
```



- 1 An Introduction to R (<http://cran.r-project.org/manuals.html>)
- 2 R project site (<http://www.R-project.org/>)
- 3 R faq (<http://cran.r-project.org/doc/FAQ/R-FAQ.html>)
- 4 UCLA's website (<http://www.ats.ucla.edu/stat/r/>)