



THE UNIVERSITY OF  
**SYDNEY**

Lecture 1:  
Basics of statistical computing

**STAT5003**

# What will we cover in STAT5003?

Abstract  Concrete

Concepts related to  
statistical data  
analysis and  
statistical learning

Methods related to  
statistical data  
analysis and  
statistical learning

R computing for  
statistical data  
analysis and  
statistical learning

Descriptive

Inferential

Exploratory

Clustering

Smoothing

Classification

# STAT5003 lecture outlines

<b>Week 1</b> Basics of statistical computing	<b>Week 2</b> Exploratory data analysis and clustering	<b>Week 3</b> Density estimation	<b>Week 4</b> Data smoothing
<b>Week 5</b> Introduction to classification	<b>Week 6</b> Cross-validation and bootstrap	<b>Week 7</b> More on classification models	<b>Week 8</b> Feature and model selection
<b>Week 9</b> Combinatorial optimisation	<b>Week 10</b> Tree classifiers and ensembles	<b>Week 11</b> Model stability and diversity	<b>Week 12</b> Advanced topics

# Week 1: Basics of statistical computing

1. Review of R and R markdown
2. Review of basic statistical concepts
3. Generating survey data (email three questions you'd like to ask your fella classmates to:  
[stat5003usyd@gmail.com](mailto:stat5003usyd@gmail.com))

# A crash introduction on R

# Web sites + references

- R references:

- <http://www.R-project.org/>

- An introduction to R

- W.N. Venables, D.M. Smith and the R Development Core Team

- An introduction to statistics and R

- Introductory Statistics with R, Peter Dalgaard, 2008, Springer

- <http://www.bioconductor.org/>

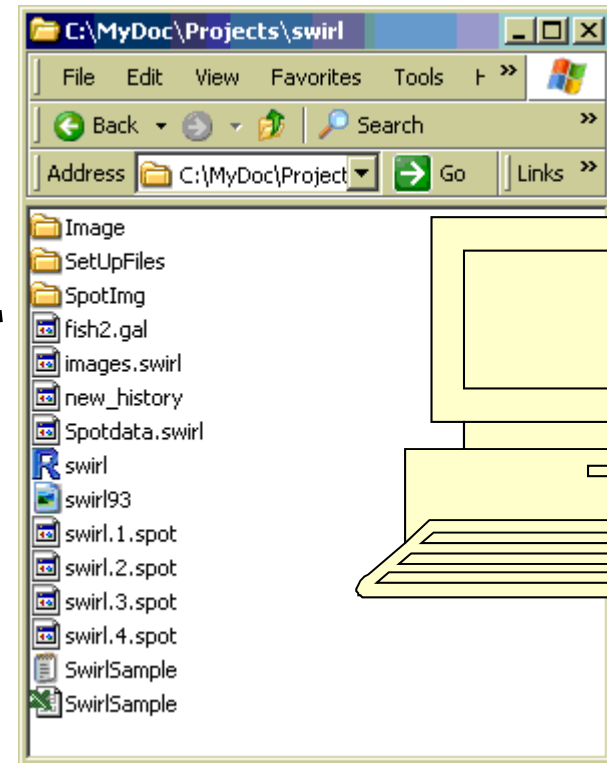
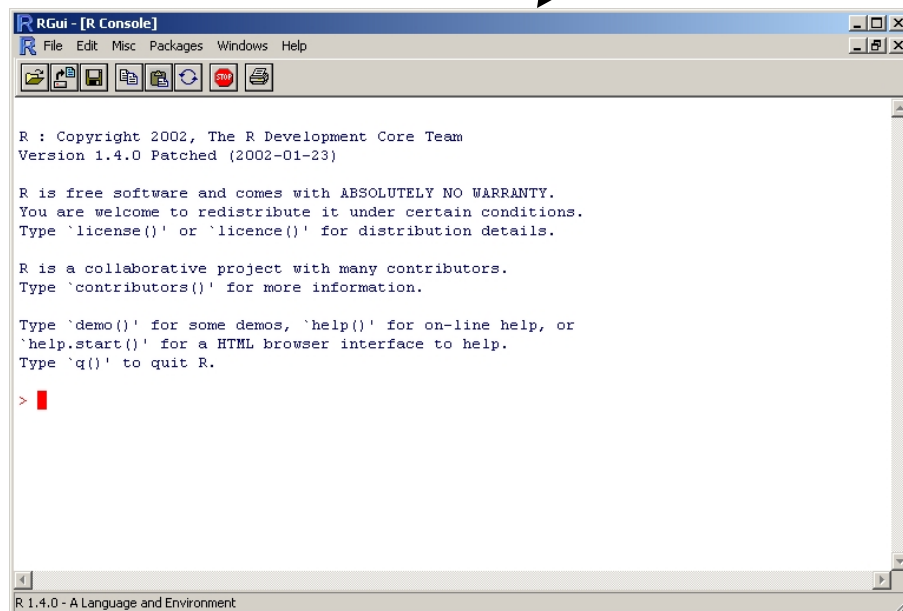
- Rstudio:

- <https://www.rstudio.com/>

Need to read files such as  
“data.txt” into the R programs.

Functions:

`read.delim`  
`scan`  
`read.table`



Save your workspace in R  
Using the function

`save.image`

You will only see  
`name.RData` or  
`.RData`

In your directory

# Download?

Download SetupR . exe from <http://cran.r-project.org/>,





# A few basics

- Working Directory
  - `getwd()`
  - `setwd()` or click on **File** and then click on **Change Dir**, use **Browse** to determine your working directory.
- Workspace
  - `save(a, b, file= "my.Rdata" )`: save objects a and b into the workspace “my.RData”
  - `save.image( "my.RData" )`: click on **File** and then click on **Save Workspace**
  - `load( "my.RData" )`: click on **File** and then click on **Load Workspace**
- Help
  - `help.start()`
  - `help()`: e.g. `help(plot)`

# Search paths + packages

```
search()
```

```
> search()
```

```
[1] ".GlobalEnv"      "package:ctest"  "Autoloads"
     "package:base"
```

```
library(cluster)
```

```
search()
```

```
> library(cluster)
```

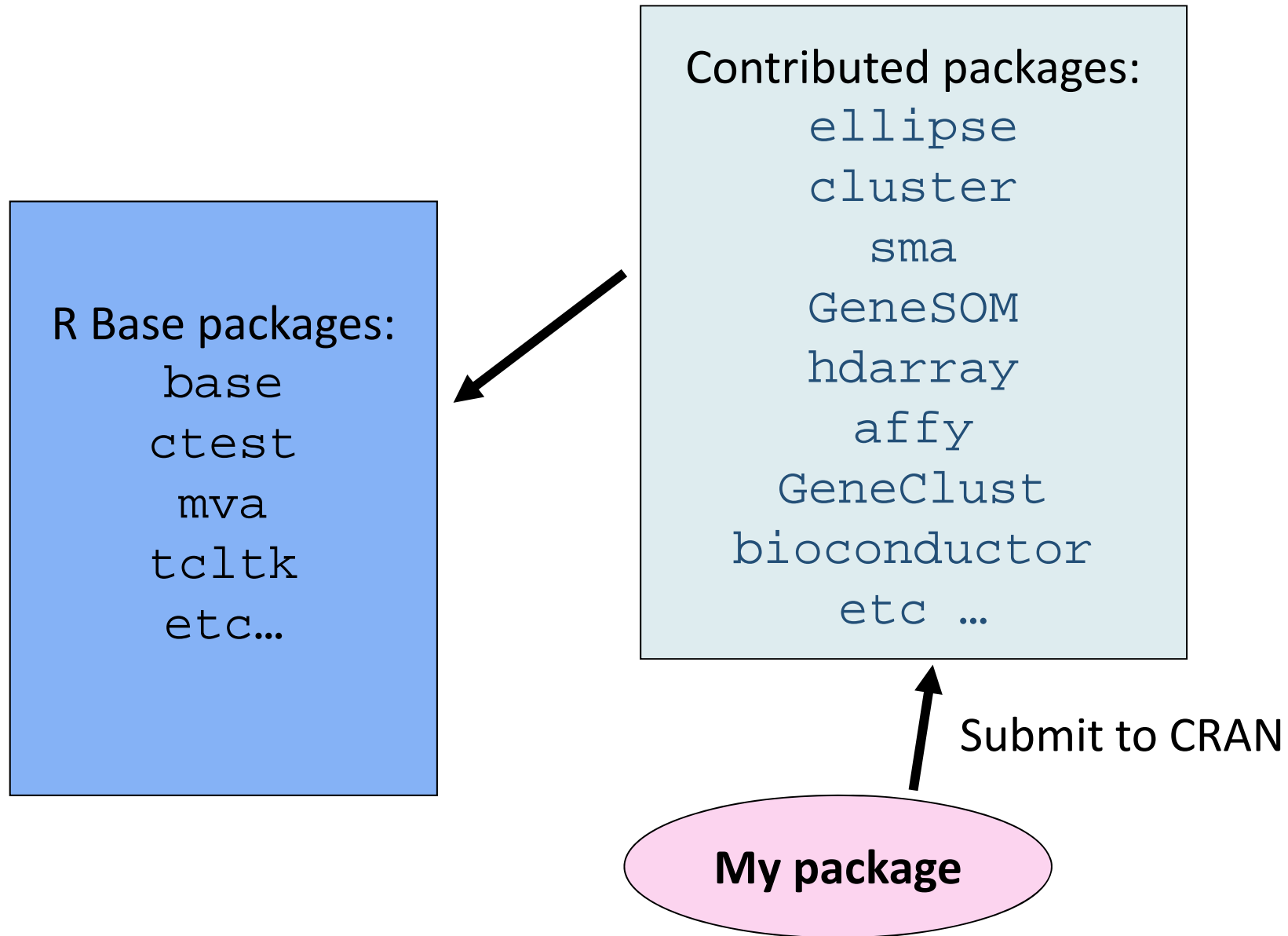
```
Loading required package: mva
```

```
> search()
```

```
[1] ".GlobalEnv"      "package:mva"    "package:cluster"
     "package:ctest"
[5] "Autoloads"       "package:base"
```

```
ls() : list objects in the GlovalEnv
```

```
ls(3) : list objects in search position number 3, in the above
        example, it is package:cluster
```



# Vectors and assignment

R operates on named data structures. The simplest such structure is the numeric vector, which is a single entity consisting of an ordered collection of numbers.

To set up a vector named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
x = c(10.4, 5.6, 3.1, 6.4, 21.7)
```

or

```
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an assignment statement using the function `c()`

This is a numeric vector

```
> is.numeric(x)
```

```
[1] TRUE
```

# Type of vectors

**numeric**

```
X <- c(1:5, 6, 9, 3, 10)  
is.numeric(X)
```

**vector**

**Character**

```
X <- c("a", "b", "c3", "4")  
is.character(X)
```

**logical**

```
X <- c(FALSE, FALSE, TRUE, FALSE)  
is.logical(X)
```

# Some R basics

- Install package by

`install.package(packagename);` example:

```
install.package(e1071)
```

- Loading a package, either by Menu “Packages”, or R command:

`library(packagename);` example:

```
library(e1071)
```

- Help (topic); example: `help(svm)`

- `demo(.)`, example:

```
demo(lm.glm, package = "stats", ask = TRUE)
```

# Some R basics

- `<-` is the symbol for ‘assign’
  - example: `x <- 14`
  - which is equivalent to: `x = 14`
- A function returns a value resulting from programming statements. A function may or may not have input arguments.
- Examples:
  - `abs(-14)`,
  - `max(2, 5, 6)`,
  - `c(2, 4, 7, 8)`, `help(abs)`

# Dealing with matrix

- Create a matrix

```
mymatrix <- matrix(1:20,5,4)
```

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20



# Index and sub-setting a matrix

- Create a matrix

```
mymatrix <- matrix(1:20,5,4)
```

- Sub-setting of vectors and matrices:

- `mymatrix[1,2],`
- `mymatrix[1,],`
- `mymatrix[,1],`
- `mymatrix[1:2,],`
- `mymatrix[c(1,3),]`

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

# Index and sub-setting a matrix

- Create a matrix  
`mymatrix <- matrix(1:20,5,4)`
- Sub-setting of vectors and matrices:
  - `mymatrix[1,2],`
  - `mymatrix[1,],`
  - `mymatrix[,1],`
  - `mymatrix[1:2,],`
  - `mymatrix[c(1,3),]`

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

# Index and sub-setting a matrix

- Create a matrix

```
mymatrix <- matrix(1:20,5,4)
```

- Sub-setting of vectors and matrices:

- mymatrix[1,2],
- mymatrix[1,],
- mymatrix[,1],
- mymatrix[1:2,],
- mymatrix[c(1,3),]

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

# Index and sub-setting a matrix

- Create a matrix

```
mymatrix <- matrix(1:20,5,4)
```

- Sub-setting of vectors and matrices:

- `mymatrix[1,2],`
- `mymatrix[1,],`
- `mymatrix[,1],`
- `mymatrix[1:2,],`
- `mymatrix[c(1,3),]`

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

# Index and sub-setting a matrix

- Create a matrix

```
mymatrix <- matrix(1:20,5,4)
```

- Sub-setting of vectors and matrices:

- `mymatrix[1,2],`
- `mymatrix[1,],`
- `mymatrix[,1],`
- `mymatrix[1:2,],`
- `mymatrix[c(1,3),]`

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

# Programming language and tools other than R

- **R (and RStudio)**
- **Weka** (Implemented in Java;  
[https://en.wikipedia.org/wiki/Weka\\_\(machine\\_learning\)](https://en.wikipedia.org/wiki/Weka_(machine_learning)))
- **Python** ([https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)))
- **Octave** ([https://en.wikipedia.org/wiki/GNU\\_Octave](https://en.wikipedia.org/wiki/GNU_Octave))

Collaboration and code distribution:

- **Github** (<https://en.wikipedia.org/wiki/GitHub>)
- **Google Docs** (<https://www.google.com/docs/about/>)

R markdown

# R Markdown Cheat Sheet

learn more at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)

rmarkdown 0.2.50 Updated: 8/14



**1. Workflow** R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

**i. Open** - Open a file that uses the .Rmd extension.



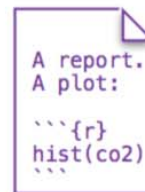
**ii. Write** - Write content with the easy to use R Markdown syntax



**iii. Embed** - Embed R code that creates output to include in the report



**iv. Render** - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.

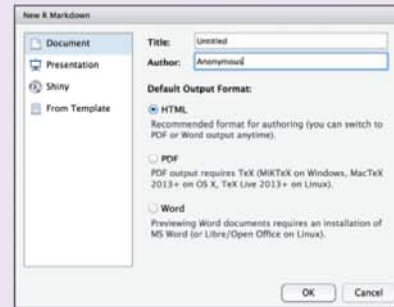




## 2. Open File

Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ► New File ► R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



## 4. Choose Output

Write a YAML header that explains what type of document to build from your R Markdown file.

### YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (---)

```
---
title: "Untitled"
author: "Anonymous"
output: html_document
---
```

This is the start of my report. The above is metadata saved in a YAML header.

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

**output: html\_document** ..... html file (web page)

**output: pdf\_document** ..... pdf document

**output: word\_document** ..... Microsoft Word .docx

**output: beamer\_presentation** ..... beamer slideshow (pdf)

**output: ioslides\_presentation** ..... ioslides slideshow (html)



## 3. Markdown

Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

### syntax

Plain text  
 End a line with two spaces to start a new paragraph.  
 \*italics\* and \_italics\_  
 \*\*bold\*\* and \_\_bold\_\_  
 superscript^2^  
 ~~strikethrough~~  
 [link](www.rstudio.com)

# Header 1  
 ## Header 2  
 ### Header 3  
 #### Header 4  
 ##### Header 5  
 ##### Header 6

endash: --  
 emdash: ---  
 ellipsis: ...  
 inline equation:  $A = \pi r^2$   
 image: ![path/to/smallorb.png]

horizontal rule (or slide break):

\*\*\*

> block quote

\* unordered list  
 \* item 2  
   + sub-item 1  
   + sub-item 2

1. ordered list  
 2. item 2  
   + sub-item 1  
   + sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

### becomes

Plain text  
 End a line with two spaces to start a new paragraph.  
*italics* and *italics*  
**bold** and **bold**  
 superscript<sup>2</sup>  
~~strikethrough~~  
[link](http://www.rstudio.com)

## Header 1

## Header 2

### Header 3

#### Header 4

#### Header 5

#### Header 6

endash: –  
 emdash: —  
 ellipsis: ...  
 inline equation:  $A = \pi r^2$



horizontal rule (or slide break):

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

- ordered list
- item 2
  - sub-item 1
  - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

**5. Embed Code** Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.

### inline code

Surround code with back ticks and r. R replaces inline code with its results.

Two plus two equals ``r 2 + 2``.  
Two plus two equals 4.

### code chunks

Start a chunk with ``{r}``.  
End a chunk with ``}``.

Here's some code  
`{r}`  
`dim(iris)`  
`}`  
Here's some code  
`dim(iris)`  
## [1] 150 5

### display options

Use knitr options to style the output of a chunk. Place options in brackets above the chunk.

Here's some code  
`{r eval=FALSE}`  
`dim(iris)`  
Here's some code  
`dim(iris)`  
Here's some code  
`{r echo=FALSE}`  
`dim(iris)`  
Here's some code  
`## [1] 150 5`

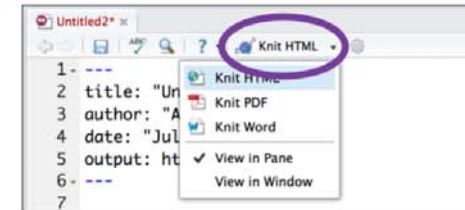
option	default	effect
eval	TRUE	Whether to evaluate the code and include its results
echo	TRUE	Whether to display code along with its results
warning	TRUE	Whether to display warnings
error	FALSE	Whether to display errors
message	TRUE	Whether to display messages
tidy	FALSE	Whether to reformat code in a tidy way when displaying it
results	"markup"	"markup", "asis", "hold", or "hide"
cache	FALSE	Whether to cache results for future renders
comment	"##"	Comment character to preface results with
fig.width	7	Width in inches for plots created in chunk
fig.height	7	Height in inches for plots created in chunk

For more details visit [yihui.name/knitr/](https://yihui.name/knitr/)

**6. Render** Use your .Rmd file as a blueprint to build a finished report.

Render your report in one of two ways

1. Run `rmarkdown::render("<file path>")`
2. Click the **knit HTML** button at the top of the RStudio scripts pane



When you render, R will

- execute each embedded code chunk and insert the results into your report
- build a new version of your report in the output file type
- open a preview of the output file in the viewer pane
- save the output file in your working directory

**7. Interactive Docs** Turn your report into an interactive Shiny document in 3 steps

1 Add runtime: shiny to the YAML header

```
title: "Line graph"
output: html_document
runtime: shiny
```

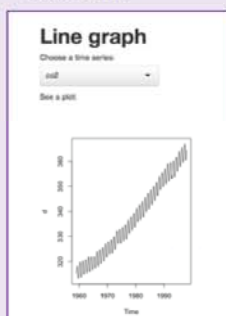
2 In the code chunks, add Shiny input functions to embed widgets. Add Shiny render functions to embed reactive output

```
title: "Line graph"
output: html_document
runtime: shiny

Choose a time series:
{r echo = FALSE}
selectInput("data", "",
  c("co2", "lh"))

See a plot:
{r echo = FALSE}
renderPlot({
  d <- get(input$data)
  plot(d)
})
```

3 Render with `rmarkdown::run` or click Run Document in RStudio



\* Note: your report will be a Shiny app, which means you must choose an html output format, like **html\_document** (for an interactive report) or **ioslides\_presentation** (for an interactive slideshow).

**8. Publish** Share your report where users can visit it online

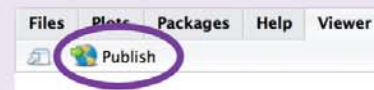
### Rpubs.com

Share non-interactive documents on RStudio's free R Markdown publishing site  
[www.rpubs.com](https://www.rpubs.com)

### ShinyApps.io

Host an interactive document on RStudio's server. Free and paid options  
[www.shinyapps.io](https://www.shinyapps.io)

Click the "Publish" button in the RStudio preview window to publish to [rpubs.com](https://www.rpubs.com) with one click.



**9. Learn More**

Documentation and examples - [rmarkdown.rstudio.com](https://rmarkdown.rstudio.com)

Further Articles - [shiny.rstudio.com/articles](https://shiny.rstudio.com/articles)

Blog - [blog.rstudio.com](https://blog.rstudio.com)

@rstudio



RStudio® and Shiny™ are trademarks of RStudio, Inc.  
CC BY RStudio info@rstudio.com  
844-448-1212 rstudio.com

# Review of basic statistical concepts

# Population

- Definition:
  - The set of data (numeric or otherwise) corresponding to the entire collection of units about which information is sought.
- Examples:
  - Blood pressure – Blood pressure readings of ALL people in Australia.
  - The number of languages spoken from ALL currently enrolled students in University of Sydney

# Sample

- Definition:
  - A subset of the population data that are actually collected in the course of a study.
- Examples:
  - Blood pressure readings of 1000 randomly selected people in Australia.
  - The number of languages spoken from 500 randomly selected students currently enrolled in University of Sydney.

*In most studies, it is difficult to obtain information about the whole population. That is why we rely on samples to make estimates and inferences related to the whole population.*

# Parameters vs statistics

- A **parameter** is a number that describes a population.
- A **statistic** is a number that describes a sample.
- *Parameters* are usually denoted using Greek letters  $\{\mu, \sigma\}$  while *statistics* are usually denoted using Roman letters  $\{x, s\}$ .
- A *parameter* is a fixed number (usually unknown). A *statistic* is a variable whose value varies from sample to sample.

# Descriptive statistics – numeric and graphics

Many methods are available for summarising data in both **numeric** and **graphical** form.

- **Numeric**

- Measure of *location*

- Mean, Median, Mode

- Measure of *spread*

- Standard deviation, MAD (median absolute deviation),  
IQR

- Others:

- Min, Max, Quartile, Five number summaries (used later in  
boxplot)

# Measure of locations – Mean

- Consider a sample of data drawn from some population.

$$\{x_1, x_2, \dots, x_n\}$$

Observations  
Sample size

- Definition of **Sample mean** = The sum of all the observations divided by the number of observations. It is written in symbols as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- Example:

Consider the following data set:

23, 34, 32, 33, 34, 22, 32, 29, 29, 34, 32, 31

Sample mean =  $365 / 12 = 30.4$



# Measure of locations – Median

The median of a set of data is a value  $\tilde{x}$  such that at least one half of the observations are less than or equal to  $\tilde{x}$  and at least one half of the observations are greater than or equal to  $\tilde{x}$ .

Definition of **Sample median** is:

- (a) The  $(n+1)/2$  th largest observation if  $n$  is odd.
- (b) The average of the  $(n/2)$ th and  $(n/2 + 1)$ th largest observation if  $n$  is even.

*Example:*

Consider the following data sets which consists of white-blood counts (x1000) taken on admission of all patients on a given day.

- (I) 7, 35, 5, 9, 8, 3, 10, 12, 8  
35, 12, 10, 9, 8, 8, 7, 5, 3

# Measure of locations – Mode

The *mode* is the most frequently occurring value among all the observations in a sample.

*Note:*

- If no entry is repeated, the data set has no mode.
- If two entries occur with the same greatest frequency, each entry is a mode and the data (biomodal).

*Example:*

Consider the white-blood cell (x1000) example: 7, 35, 5, 9, 8, 3, 10, 12, 8

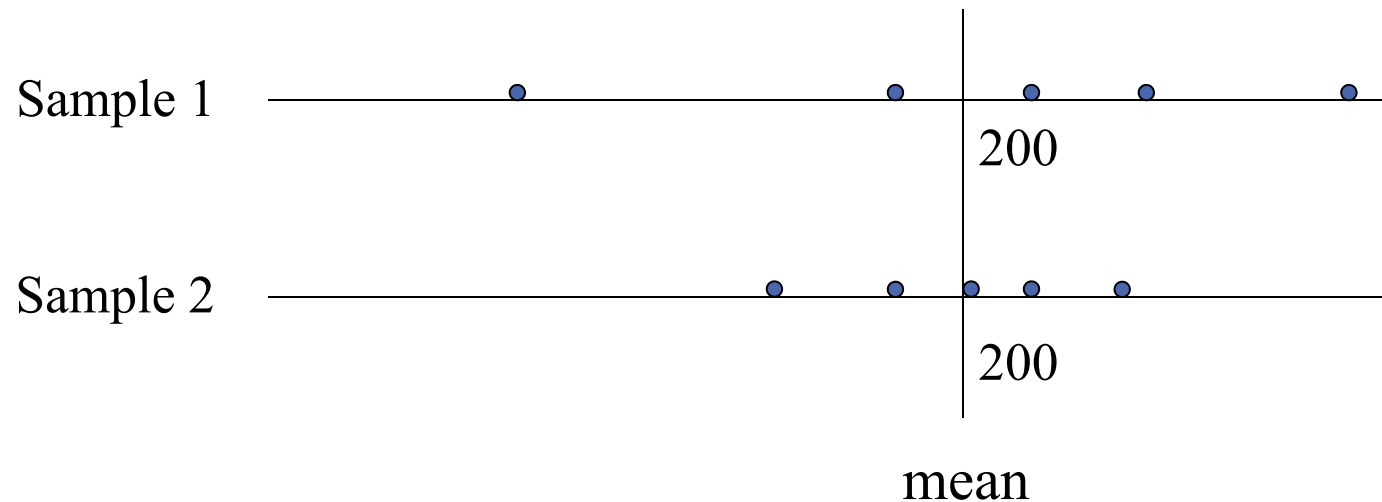
The mode is 8000, because it occurs more frequently (twice) than any other white-blood count (once).

# Median or mean?

- Both the median and the mean are measures of location, but which is preferable?
- For symmetric data, the mean is usually less variable from sample to sample than the median.
- For skewed data, the median is a better measure of location.
- The median does not react as much as the mean by outliers. This property of the median is known as ‘robustness’.
- The mean is easier to compute than the median and is much easier to handle theoretically.

# Measure of spread

Consider two samples shown below. Describe the difference ...



“variability (spread)”

- The *range* of a list is the largest value minus the smallest value. This gives a quick feeling for the overall spread – but is misleading because it is solely influenced by two most extreme values.

# Measure of spread – MAD

- Deviation: the difference between the individual sample points and the arithmetic mean.  $\{x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_n - \bar{x}\}$
- Consider the data set:  $\{177, 193, 195, 209, 226\}$   
The mean is  $(177 + 193 + 195 + 209 + 226) / 5 = 200$   
The deviations for this data set is:  
 $\{-23, -7, -5, 9, 26\}$
- The **MAD** is “Median Absolute Deviation” and is defined as median  $|y_i - \text{median}(y)|$

## **Example:**

1. The deviations from median (195) for this data set is  $\{-18, -2, 0, 14, 31\}$
2. The absolute value from the set of deviations is  $\{0, 2, 14, 18, 31\}$  and the median value is 14. Hence the MAD for the above data set is 14.

# Measure of spread – Standard deviation (I)

- The **standard deviation (SD) or variance** measures how spread out the data are around their mean.
- Steps to calculate sample variance,
  1. Find the mean of the data,
  2. Make a list of **deviations** from the mean.
  3. Calculate the average of the squares of deviations. (var)
- The **sample SD** is defined as:

$$\text{var} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

# Measure of spread – Standard deviation (II)

- Example**

Consider the data set: {177, 193, 195, 209, 226}

Step 1: The mean is 200

Step 2: The deviations for this data set is:

$$\{-23, -7, -5, 9, 26\}. \quad s^2 = [(-23)^2 + (-7)^2 + (-5)^2 + 9^2 + 26^2] / (5 - 1)$$

Step 3: Sample variance is:

$$= 1360 / 4$$

$$= 340$$

$$s = \sqrt{340} = 18.4$$

The **calculating formula**  
for **sample SD** is:

$$s^2 = \frac{\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2 / n}{n - 1}$$

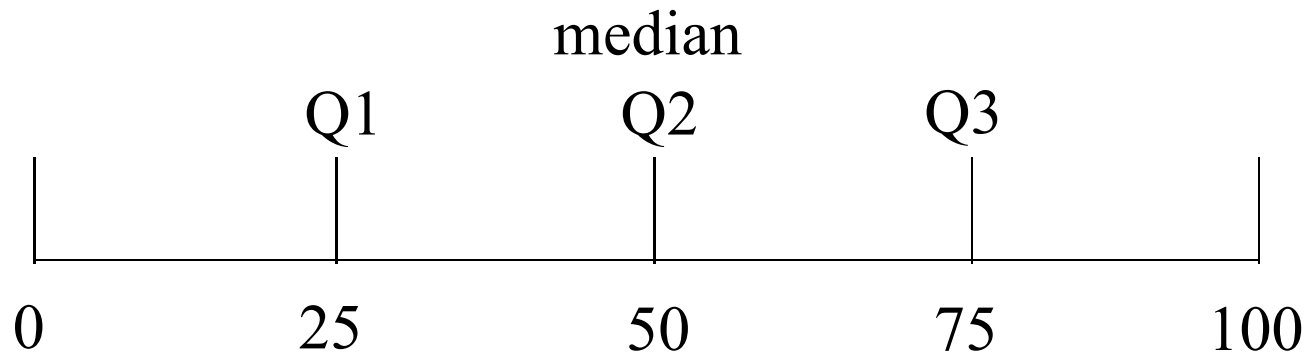
$$= \frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2}{n - 1}$$

$$= \frac{(177^2 + 193^2 + 195^2 + 209^2 + 226^2) - (5 \times 200^2)}{4}$$

$$= (201360 - 200000) / 4 = 1360 / 4$$

# Measure of spread – IQR

- The three **quartiles**, Q1, Q2, and Q3, approximately divide an ordered data set into four equal parts.



The **Inter-quartile range (IQR)** is defined as the upper quartile (Q3; 75th percentile) minus the lower quartile (Q1; 25th percentile). It is the width of the interval that contains the middle 50% of the data

$$\text{IQR} = Q3 - Q1$$

In R:

```
quantile(x)
```



# SD vs IQR

- Similar to median vs mean
- Sample standard deviations and the IQR ( $= Q3 - Q1$ ) are both measures of spread. The IQR is robust, like the median, but it is harder to handle theoretically than the standard deviation.

# Discrete distributions

For any random variable  $X$  with a discrete distribution, there is a sample space  $\Omega$  with finite number of possible values  $x = \{x_1, x_2, \dots\}$  and associated probabilities  $\{p_1, p_2, \dots\}$ .

The point probabilities for each value of  $x$  is  $f(x) = P(X = x)$  and the cumulative distribution function  $F(x) = P(X \leq x)$

Properties:

- There is a countable number of possible values;
- $\sum_i p_i = 1$

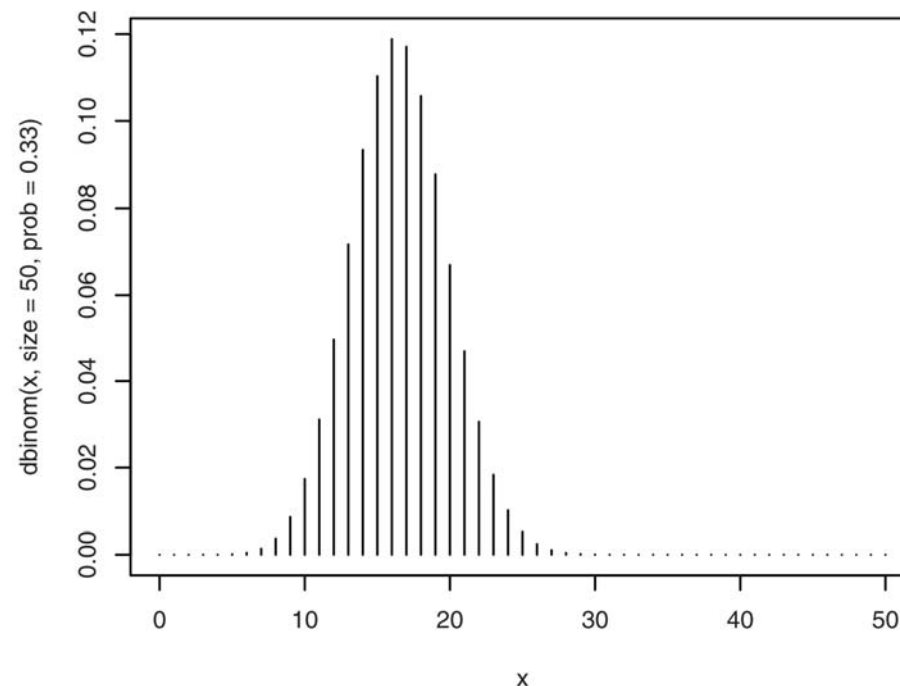
# Binomial distribution

$$f(x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

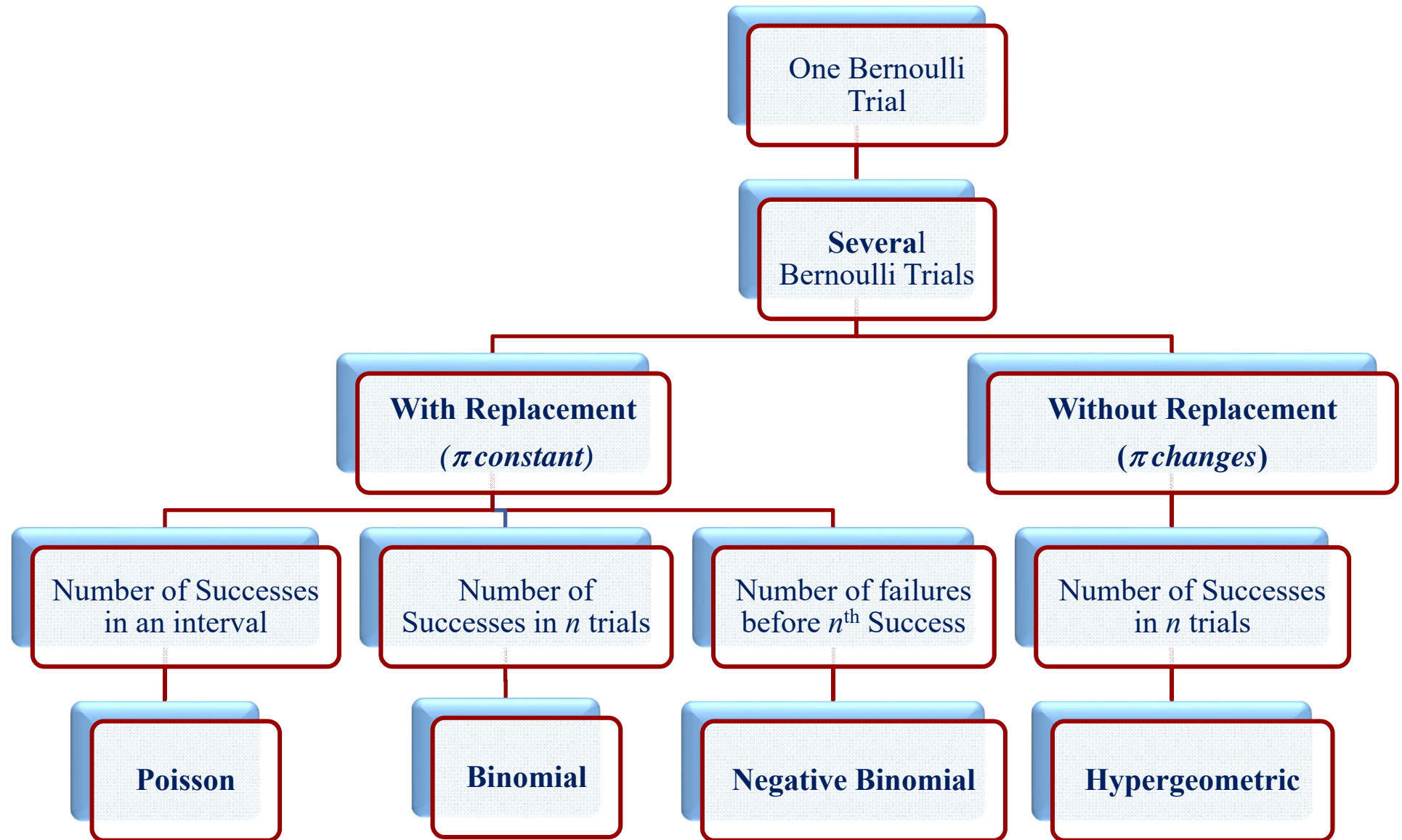
$\binom{n}{x}$  is known as binomial coefficients. The parameter  $p$  is the probability of a successful outcome in an individual trial (called a Bernoulli Trial).

In R: 

```
x <- 0:50  
plot(x, dbinom(x, size=50, prob=0.33), type="h")
```



# Summary of special discrete distributions



# Continues distributions

For any random variable  $X$  with a continues distribution, there is an infinite number of possible values; These values may be within a fixed interval such as the height of male in cm may be within the range of  $[50, 300]$ .

The point probabilities for each value of  $x$  is  $P(X = x) = 0$  and the cumulative distribution function  $F(x) = \int_{-\infty}^x f(x)dx$

Properties:

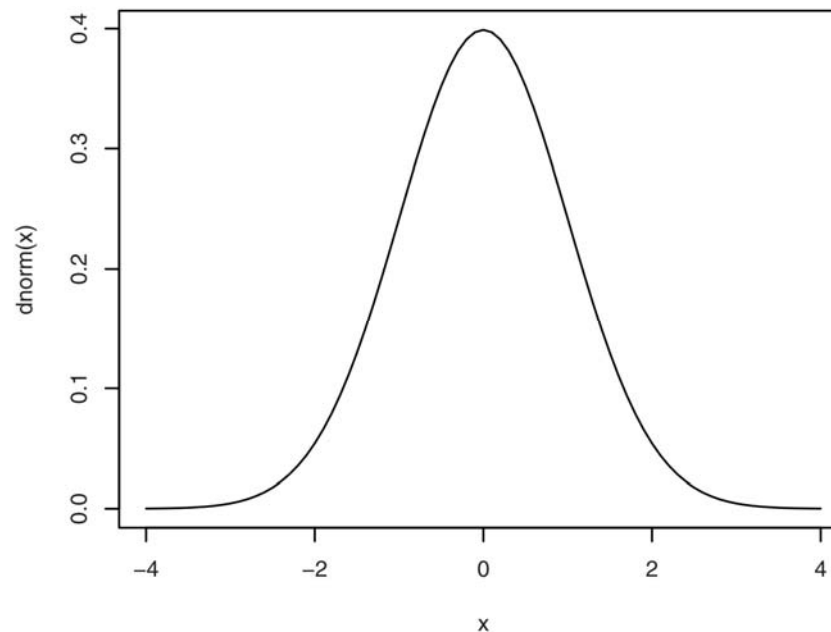
- There are infinite number of possible values;
- $f(x)$  is called the density function and its integration from  $-\infty$  to  $+\infty$  with respect to  $x$  is 1.

# Normal (Gaussian) distributions

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The parameter  $\mu$  and  $\sigma$  are the mean and standard deviation as defined in previous slides.

In R: `x <- seq(-4, 4, 0.1)`  
`plot(x, dnorm(x), type="l")`



References and some mathematical and statistical  
notations used in this lecture

# Statistical notation

We use capital letters to denote random variables, such as  $Y$  or  $X$ , and lowercase letters to represent specific realised values of random variables such as  $y$  or  $x$ .

- The probability density function of  $X$  is denoted  $f$ ;
- the cumulative distribution function is  $F$ .
- We use the notation  $X \sim f(x)$  to mean that  $X$  is distributed with density  $f(x)$ . Frequently, the dependence of  $f(x)$  on one or more parameters also will be denoted with a conditioning bar, as in  $f(x|\alpha, \beta)$ .
- The density functions for  $X$  and  $Y$  are  $f_X$  and  $f_Y$ , respectively.
- We use the same notation for distributions of discrete random variables and in the Bayesian context.




# Vectors and Matrices





- We use boldface to distinguish a vector  $\mathbf{x} = (x_1, \dots, x_p)$  or a matrix  $\mathbf{M}$  from a scalar variable  $x$  or a constant  $M$ .
- A vector-valued function  $\mathbf{f}$  evaluated at  $\mathbf{x}$  is also boldfaced, as in  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x}))$ .
- The transpose of  $\mathbf{M}$  is denoted  $\mathbf{M}^T$ .
- Unless otherwise specified, all vectors are considered to be column vectors, so, for example, an  $n \times p$  matrix can be written as  $\mathbf{M} = (\mathbf{x}_1 \dots \mathbf{x}_n)^T$ .
- Let  $\mathbf{I}$  denote an identity matrix, and  $\mathbf{1}$  and  $\mathbf{0}$  denote vectors of ones and zeros, respectively.

Example:  $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

# Vectors in R

- <http://adv-r.had.co.nz/Data-structures.html>
- <http://www.cyclismo.org/tutorial/R/types.html>
- <https://www.youtube.com/watch?v=YhQOV27pQfg> 
- <http://www.r-tutor.com/r-introduction/vector>
- <https://stat.ethz.ch/R-manual/R-devel/library/base/html/vector.html>
- <http://statistics.berkeley.edu/computing/r-vectors-matrices>

# Matrices in R

- <http://www.r-tutor.com/r-introduction/matrix>
- <http://www.r-tutor.com/r-introduction/matrix/matrix-construction>
- <http://www.statmethods.net/advstats/matrix.html>
- <https://www.youtube.com/watch?v=VRA08-4GzOA> 
- <https://www.youtube.com/watch?v=cR-hEUs1rRw> 
- <https://www.youtube.com/watch?v=92v44CN-Sz4> 
- <https://www.youtube.com/watch?v=1QYdrMRhNJs> 

# Derivatives

The derivative of a function  $f$ , evaluated at  $x$ , is denoted  $f'(x)$ .

When  $\mathbf{x} = (x_1, \dots, x_p)$ , the gradient of  $f$  at  $\mathbf{x}$  is

$$\mathbf{f}'(\mathbf{x}) = \left( \frac{df(\mathbf{x})}{dx_1}, \dots, \frac{df(\mathbf{x})}{dx_p} \right).$$

Derivatives in R:

<https://www.youtube.com/watch?v=X1QHNsoch98> 

1. Review of R and R markdown
2. Review of descriptive statistics
3. **Generating survey data** (email three questions you'd like to ask your fella classmates to:  
[stat5003usyd@gmail.com](mailto:stat5003usyd@gmail.com))