R: A First Look

Srinivasa Rao

Michaelmas Term 2021

Ready to learn?

- Today's session takes place in a videocall using Microsoft Teams
- Can you see and hear the teacher?
- Please tell us if anything doesn't work
- Don't plan to multi-task
- Please silence mobile phones

Today's resources

- How will you display the videocall, browser window, slides (if desired)?
- Where are your course files?
- Is the software installed?

Find the resources for the workshop in our IT Learning Portfolio

Download the files (and more) from the IT Learning Portfolio at skills.it.ox.ac.uk/it-learning-portfolio

What is R?

- R is a statistical programming language
- Like most programming languages, it can be used for a lot of purposes...
- ... but it's specifically designed for dealing with data
- (Slightly stereotypically) it's the language of choice for academia
  - Industry tends towards Python

A little history...

- R is an adaptation of an earlier language called S (still around in the form of S-PLUS)
  - Superficially like S, but also inspired by a language called Scheme
- Created by Ross Ihaka and Robert Gentleman at the University of Auckland in 1993
- Made free and open-source in 1995, at the urging of Martin Mächler (ETH Zürich)
- Developed by the "R Core Team" since 1997
- Version 1.0.0 released on 29th February 2000
  - As of January 2021, on version 4.0.3 ("Bunny-Wunnies Freak Out")

Why use R?

- It's good at dealing with data
- Free and open-source = lots of packages (more on this later)
- It's relatively easy to use
- It's "interpreted" (you can run code a line at a time)

Why not use R?

- It's on the slower end: can take time to run with large datasets
  - An advanced technique is to write complex functions in other languages, and then use them from R
- It's less popular than Python in industry
  - Python is also free and open-source

What can you do with R?

These are examples of plots with R, from the R Graph Gallery

- Scatter plot, with densities
- Word cloud
- Choropleth
- Network
- Treemap

What is RStudio?

- RStudio is an "Integrated Development Environment" (IDE) for R
- Provides convenient way to edit code, run code, view plots, view help files, view data files, and so on
- Free as in "free beer", but not as in "free speech"

What is RStudio Cloud?

- A browser-based version of RStudio
- Like many cloud systems (Google Docs, Overleaf, Microsoft Office 365) you can access your work from any computer
- Free as in "free beer", but only for light use (e.g. a course!)
- Intensive use comes at a cost

Let's go to RStudio Cloud

- I'll send you a link through Teams and guide you through the process

The basics

On the first worksheet, we'll:

- study the basics of R
- learn how to carry out arithmetical calculations
- store values as variables
- learn what vectors are and how to work with them

Exercise 1 answer

```
lucky.number <- 44
lucky.number/2
```

```
## [1] 22
```

Exercise 2 answer

```
n <- 20
(1:n)^2
```

```
##   [1]   1   4   9  16  25  36  49  64  81 100 121
##  [12] 144 169 196 225 256 289 324 361 400
```

Exercise 2A answer

```
n <- 20
vector <- 1:n
vector*(vector+1)/2
```

```
##  [1]   1   3   6  10  15  21  28  36  45  55  66
## [12]  78  91 105 120 136 153 171 190 210
```

Four domains of application

- Data management
- Data analysis
- Data visualisation
- Data reporting

We'll look at all of these, to a greater or lesser degree.

Data management

On the second worksheet, we'll:

- Learn about data frames and import one
- Select columns and rows of a data frame
- Filter data by its values

Exercise 2B answer

The code without "`cars_dataset <-`" just loads the data, prints it back out, and then throws it away again. This means we can't do anything further with it.

Exercise 3 answer

```
my.cars <- cars_dataset[c(3,9,10),4:6]
```

```
##      disp  hp drat
## 3   108.0  93 3.85
## 9   140.8  95 3.92
## 10 167.6 123 3.92
```

Exercise 4 answer

```
cars_dataset$mpg[c(10:12, 14)]
```

```
## [1] 19.2 17.8 16.4 15.2
```

One thing to notice is that, in the portion `c(10:12, 14)`, we have embedded a vector (`10:12`) inside another vector. This is fine: R knows to treat this as one long vector. Alternatively, we could have written `c(10, 11, 12, 14)`.

Exercise 5 answer

```
cars_dataset[-16, -(1:3)]
```

I'm not printing the output on the slide, because it wouldn't fit!

Exercise 6 answer

```
cars_dataset[cars_dataset$mpg<15,c("Car", "mpg")]
```

```
##                          Car  mpg
## 7                 Duster 360 14.3
## 15  Cadillac Fleetwood 10.4
## 16 Lincoln Continental 10.4
## 17   Chrysler Imperial 14.7
## 24            Camaro Z28 13.3
```

```
#Cars with mpg less than 15
```

I've modified the R code relative to the exercise sheet, so that we only see certain columns (again, it wouldn't fit on the slide if I didn't).

Exercise 6 answer

```
cars_dataset[cars_dataset$carb>=6,c("Car", "carb")]
```

```
##                Car carb
## 30  Ferrari Dino    6
## 31 Maserati Bora    8
```

```
#Cars with at least six carburettors
```

Exercise 6 answer

```
cars_dataset[cars_dataset$vs!=0,c("Car", "vs")]
```

```
##                  Car vs
## 3       Datsun 710  1
## 4  Hornet 4 Drive  1
## 6           Valiant  1
## 8         Merc 240D  1
## 9          Merc 230  1
## 10         Merc 280  1
## 11        Merc 280C  1
## 18         Fiat 128  1
## 19     Honda Civic  1
## 20 Toyota Corolla  1
## 21  Toyota Corona  1
## 26       Fiat X1-9  1
## 28    Lotus Europa  1
## 32      Volvo 142E  1
```

```
#Cars without a V-shaped engine
```

Exercise 6 answer

On the previous slide: `!=` really does mean "not equal to": you don't need to (and shouldn't) type "`!==`".

```
cars_dataset[startsWith(cars_dataset$Car, "Merc"),c("Car",
```

```
##              Car  mpg
## 8     Merc 240D 24.4
## 9      Merc 230 22.8
## 10     Merc 280 19.2
## 11    Merc 280C 17.8
## 12   Merc 450SE 16.4
## 13   Merc 450SL 17.3
## 14 Merc 450SLC 15.2
```

```
#Cars starting with "Merc"
```

Exercise 6 answer

```
cars_dataset[cars_dataset$Car >= "T",c("Car", "mpg")]
```

```
##                  Car  mpg
## 6            Valiant 18.1
## 20 Toyota Corolla 33.9
## 21  Toyota Corona 21.5
## 32      Volvo 142E 21.4
```

```
#Cars from "T" onwards in the alphabet
```

Importing data

- We used the command `read.csv` to load the data frame
- This assumes that your CSV file:
  - has values separated by commas
  - has a row of column headers
- We can toggle these using function arguments:

```
cars_dataset <- read.csv("Data/mtcars.csv",
                         header = FALSE, sep = " ")
```

- To make sure you get it right:
  - Look at the CSV file before importing it (open it in RStudio or in Notepad/TextEdit)—your computer will default to Excel, which just imports it into cells
  - Check after importing using `head(cars_dataset)`

Data analysis and visualisation

On the third worksheet, we'll:

- Learn some exploratory data analysis methods using R
- Make simple plots
- (Optionally) Fit linear models and plot lines of best fit

Exercise 7 answer

```
apply(cars_dataset.no.names, 2, mean)
```

```
##         mpg         cyl        disp          hp
##  20.090625    6.187500 230.721875 146.687500
##        drat          wt        qsec          vs
##    3.596563    3.217250  17.848750   0.437500
##          am        gear        carb
##    0.406250    3.687500   2.812500
```

Exercise 8 answer

```
cars_dataset$gear.plus.carb <- cars_dataset$gear + cars_dat
cars_dataset[1:6, c("Car", "gear.plus.carb")] #print it
```

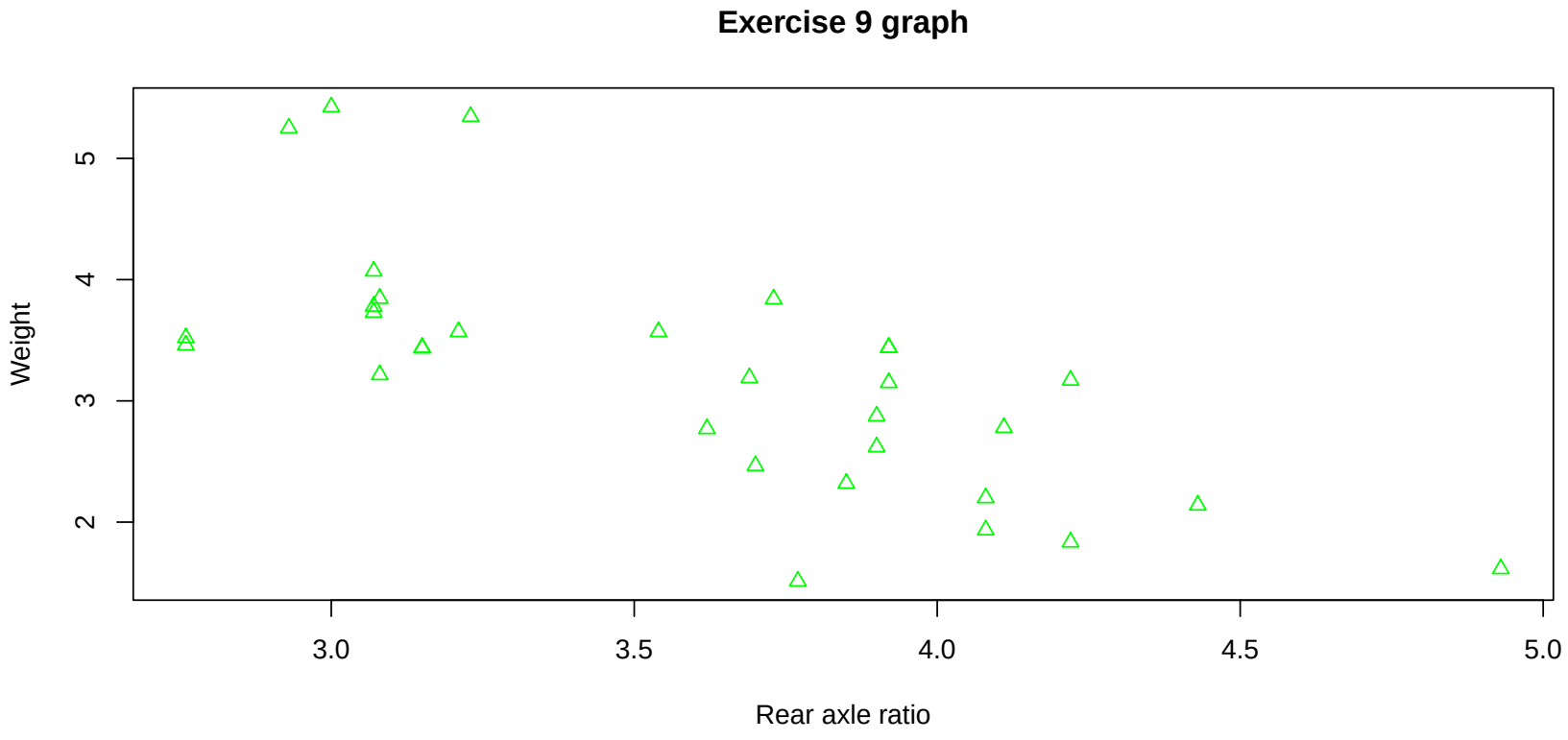
```
##                  Car gear.plus.carb
## 1          Mazda RX4              8
## 2      Mazda RX4 Wag              8
## 3         Datsun 710              5
## 4     Hornet 4 Drive              4
## 5  Hornet Sportabout              5
## 6            Valiant              4
```

Exercise 8A answer

The points are plotted in the order in which they appear in the data frame. So you'd normally only use a line plot when the points are already in some sort of order (e.g. by time), and except in certain specialised applications you'd normally put the ordered value on the x-axis.

Exercise 9 answer

```
plot(cars_dataset$drat, cars_dataset$wt, col = "green", pch
     xlab = "Rear axle ratio", ylab = "Weight",
     main = "Exercise 9 graph")
```

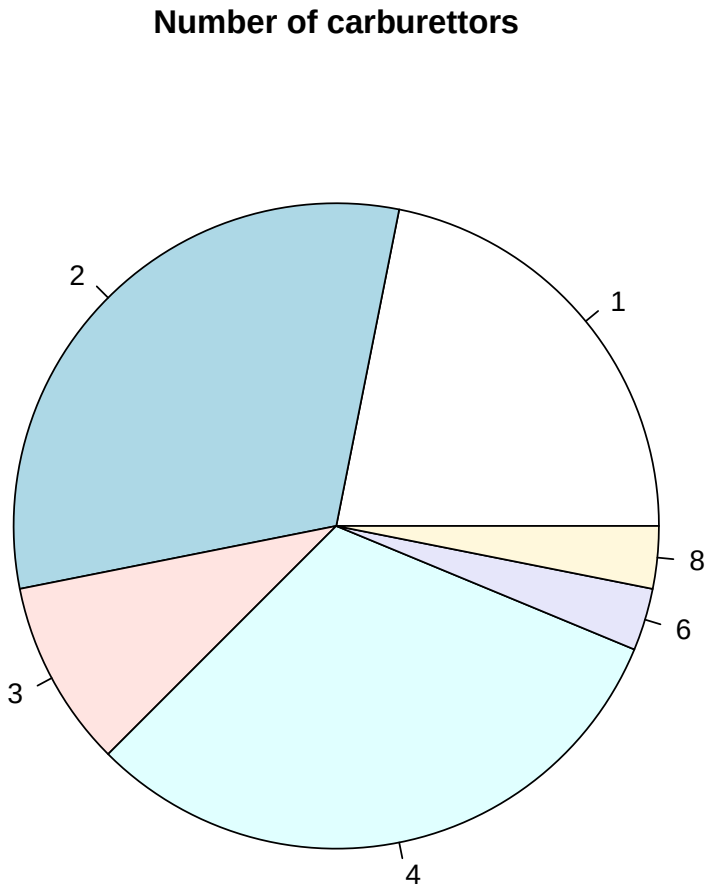


Exercise 10 answer

- (a) is a box plot of the values of MPG
- (b) is a combined box plot, showing values of Displacement and Weight on the same axis
- (c) is a histogram of the values of Displacement

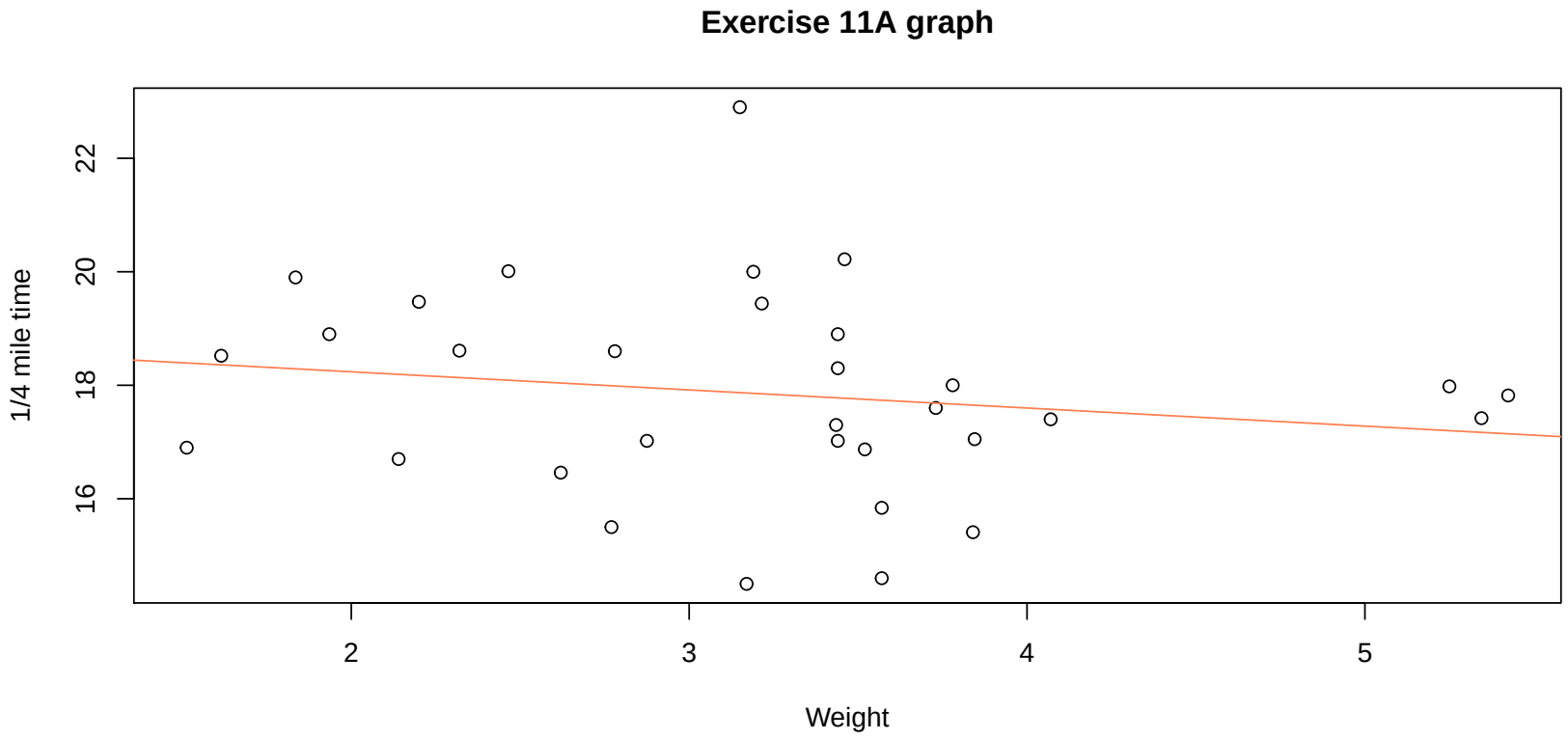
Exercise 11 answer

```
carb.tab <- table(cars_dataset$carb)
pie(carb.tab, main = "Number of carburettors")
```



Exercise 11A answer

```
mod11a <- lm(qsec~wt, data = cars_dataset)
plot(cars_dataset$wt, cars_dataset$qsec, xlab="Weight",
     ylab = "1/4 mile time",
     main = "Exercise 11A graph")
abline(mod11a, col = "coral")
```



Commenting code

- Write comments when writing code
- Will help other people understand it...
  - ... and maybe you when you look at it later!
- Remember: the # symbol is used for comments.

Data reporting

On the fourth worksheet, we'll:

- Comment some pre-written code
- (Optionally) Look at an R Markdown document

Exercise 12 answer

```
my.iris <- iris #Loads the dataset "iris"
head(my.iris) #Displays the top of the dataset
summary(my.iris) #Summarises the columns
plot(my.iris[,-5]) #Creates a "pairs plot" across the
 #continuous variables, omitting the "species" column
spec.tab <- table(my.iris$Species)  #Makes a table of
                    #values of "species", and stores it
barplot(spec.tab, xlab="Species", ylab = "Frequency")
                   #Plots a bar chart of species values
apply(my.iris[,-5], 2, sd)          #Finds the standard
                     #deviation of each numeric column
my.iris$Sepal.Ratio <-
  my.iris$Sepal.Length/my.iris$Sepal.Width
my.iris$Petal.Ratio <-
  my.iris$Petal.Length/my.iris$Petal.Width
     #Creates two new columns for the length-to-width
                           #ratio of sepals and petals
```

Exercise 12 answer

```
boxplot(my.iris$Sepal.Ratio, my.iris$Petal.Ratio,
        names = c("Sepal ratio", "Petal ratio"))
     #Makes a comparative box plot of the new columns
plot(my.iris$Sepal.Ratio, my.iris$Petal.Ratio,
     xlab= "Sepal ratio", ylab = "Petal ratio")
   #Makes a scatter plot of petal against sepal ratio
plot(my.iris$Sepal.Ratio, my.iris$Petal.Ratio,
     xlab= "Sepal ratio", ylab = "Petal ratio",
     col = my.iris$Species)
        #Changes the previous plot so that points are
                  #coloured according to their species
legend("topright",
       legend = c("setosa","versicolor","virginica"),
       col = 1:3, pch=1)
 #Adds a legend for the colours: the colours used are
    #colours 1, 2 and 3, colouring the species in the
      #order that they first appear in the data frame
```

Exercise 12 answer

```
my.iris.vir <-
  my.iris[my.iris$Species == "virginica",]
            #Filters the data only to virginica irises
plot(my.iris.vir$Sepal.Ratio, my.iris.vir$Petal.Ratio,
     xlab= "Sepal ratio", ylab = "Petal ratio",
     main = "Virginica only")
     #Repeats the plot above, but just for virginicas
mod.rat <- lm(Petal.Ratio~Sepal.Ratio,
              data = my.iris.vir)
#Creates a linear model for petal against sepal ratio
summary(mod.rat)
                    #Gives summary data for that model
abline(mod.rat, col="hotpink", lty=4)
                  #Adds a line of best fit to the plot
```

What is the tidyverse?

- An additional collection of functions that work in the same way
- Increasingly popular for working with datasets
- Many people find them more logical, easier to work with;
  - The tidyverse also provides "ggplot", which makes prettier plots than "base R"
- Others don't like them!
  - Some things are easier to do in base R; some things are easier with the tidyverse

The tidyverse

On the fifth worksheet, we'll:

- Learn what packages are, and load one
- Use pipes to chain up functions
- Use tidyverse packages to manipulate data and make plots

Exercise 13 answer

```
cars_starts_d <- cars_tib %>% select(starts_with("d"))
head(cars_starts_d)
#Selects only columns of mtcars that start with "d"

cars_contains_p <- cars_tib %>% select(contains("p"))
head(cars_contains_p)
#Selects only columns of mtcars that contain a "p"
```

I've omitted the output here. Note that the `starts_with` and `contains` functions (as well as `ends_with`) are case **in**sensitive by default.

Exercise 15 answer

```
cars_tib %>% select(!Car) %>%
  summarise(across(everything(),mean))
```

```
## # A tibble: 1 x 11
##     mpg   cyl  disp    hp  drat    wt  qsec    vs
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  20.1  6.19  231.  147.  3.60  3.22  17.8 0.438
## # ... with 3 more variables: am <dbl>, gear <dbl>,
## #   carb <dbl>
```

Instead of wrapping onto a second line, the default behaviour here is to omit what doesn't fit, and to inform you that some things are missing from the output. You probably didn't have this problem, since the console window is wider than these slides! (Unless you have a *very* low-resolution monitor.)

Exercise 16 answer

```
cars_tib <- cars_tib %>%
              mutate(gear_plus_carb = gear + carb)
```

This doesn't print anything because we've assigned the result back to `cars_tib`.

Exercise 18 answer

I've put my answer to this exercise in a separate R file for you to look at, since it's quite long!

Packages

- Extra functions for different purposes:
  - Saves you from writing your own code!
- Lots of packages on CRAN
- Syntax for installing and loading:

```
install.packages("package.name")
library(package.name)
```

- Installation is one-time, loading is necessary on each use
- To find them, use Google!
- Other sources of packages:
  - GitHub
  - Bioconductor

Where to learn more

- Type `swirl()` into the R console
- Other courses offered by IT Services
- LinkedIn Learning
  - Oxford has an institutional subscription
- Free online book: *R for Data Science* (also available to purchase in paper copy)
- Trial and error!

Downloading your work

- In the files pane, click on ".." to the right of the green arrow near the top
  - Repeat if necessary until "project" appears
- Then click on the empty square to the left of "project"
  - A tick should appear in the box
- Click "More", then "Export..."
- Choose a file name, and click "Download"
  - Make sure the file name ends in "`.zip`"
- Choose "Save file" and OK (on Windows; this step is slightly different but similar on Mac)

Copyrights