

Home Study 2: To be completed by Monday 26th October

MAS2905: Probability and Bayesian Methods with R

Newcastle University
`lee.fawcett@ncl.ac.uk`

Semester 1, 2020/21

What you should have done so far...

- Attended the first synchronous online lecture, 12-1pm Wednesday 21st October, and completed all associated activities
- Completed the video **Home Study 1**, including **Examples 2.2 and 2.3**: You should now be ready to check your answers!
- If you've not done the above, you must do so before completing the activities in this video

What you should have done so far...

- Attended the first synchronous online lecture, 12-1pm Wednesday 21st October, and completed all associated activities
- Completed the video **Home Study 1**, including **Examples 2.2** and **2.3**: You should now be ready to check your answers!
- If you've not done the above, you must do so before completing the activities in this video

Example 2.2: Solutions

- (a) How many people submitted votes for the films “*10 Things I Hate About You*”, “*Midnight in the Garden of Good and Evil*” and “*Zoolander*”?

```
1 which(movies$Title=="Midnight in the Garden of Good  
    and Evil")  
2 [1] 2372  
3 movies$Votes[2372]  
4 [1] 33728
```

Similarly, 249,949 for “*10 Things I Hate About You*”; 219,165 for “*Zoolander*”.

Example 2.2: Solutions

- (b) How many films are classified as “Action”? How many are classified as “Romance”? How many “Short” films are there in the database?

```
1 sum(movies$Action)
2 [1] 1270
```

Similarly, 804 for “Romance” and 0 for “Short”.

Example 2.2: Solutions

- (c) How many films were made between the Years 1970 and 2000 (inclusive)?

```
1 length(movies$Year[movies$Year >= 1970 & movies$Year  
2      <= 2000])  
[1] 1574
```

- (d) How many movies are longer than two hours in duration?

```
1 length(movies$Length[movies$Length > 120])  
2 [1] 1239
```

Example 2.2: Solutions

- (e) How many movies made since the year 2005 (not including 2005) are longer than three hours in duration?

```
1 movies2 = subset(movies, Year>2005)
2 length(movies2$Length[movies2$Length > 180])
3 [1] 8
```

Example 2.3: Solutions

- (a) How many tracks do we have data for, and on how many variables?

```
1 dim( music )  
2 [1] 424 10
```

Looks like 424 tracks, and 10 variables. Using the `head` command might help to confirm this.

Example 2.3: Solutions

- (b) How many tracks do we have from *The Beatles* in this dataframe? What about *Lady GaGa*?

```
1 attach(music) # Will give direct access to the columns
2 length(artist[artist=="The Beatles"])
3 [1] 19
```

Similarly, 2 tracks for *Lady GaGa*.

Example 2.3: Solutions

- (c) How many tracks have a tempo of at least 130 beats per minute?

```
1 # Have already attached the column headins in part (b)
2 length (tempo [tempo >=130])
3 [1] 121
```

Example 2.3: Solutions

- (d) Of the tracks with a tempo of at least 130 beats per minute, how many have been released since the year 2000 (including the year 2000)?

```
1 music2 = subset(music, tempo>=130)
2 length(music2$year[music2$year>=2000])
3 [1] 21
```

Or, perhaps even better:

```
1 music3 = subset(music, tempo>=130 & year>=2000)
2 dim(music3)
3 [1] 21 10
```

Example 2.3: Solutions

(e) What is the most popular key for tracks in this database?

```
1 table ( key )  
2 key  
3 A Ab B Bb C C# D E Eb F F# G  
4 43 27 28 41 42 32 43 38 23 46 20 41
```

So the answer is: Key of F! (Don't know if this is F Major or F Minor... I assume F Major!)

2.4 Graphical and numerical summaries

R has very powerful built-in functions for summarising datasets both **graphically** and **numerically**.

Most of these are fairly intuitive, so in this section we provide only a **brief review** of material from stage 1.

For example, we might want to explore the distribution of budgets in the `movies` dataframe.

As we know, there are some missing values in the dataset, but most of the built-in functions for data analysis in R have an argument we can invoke to ignore these.

2.4 Graphical and numerical summaries

```
1 # To find the mean of the movie budgets but ignoring the missing values
2 mean(movies$Budget, na.rm = TRUE)
3 [1] 41245023
4
5 # And the standard deviation
6 sd(movies$Budget, na.rm = TRUE)
7 [1] 68612118
8
9 # How about a range of numerical summaries with a single function?
10 summary(movies$Budget, na.rm = TRUE)
11 Min.    1st Qu.    Median      Mean   3rd Qu.      Max.    NA
12  0       9e+6     2.5e+7  4.125e+7   5e+7    1.9e+9   998
```

2.4 Graphical and numerical summaries

From the listing above we can make a few observations:

- The mean budget is larger than the median: $\sim$ US\$ 41M, compared to US\$ 25M. This indicates a **positively skewed** dataset, as the mean is being 'pulled up' by some very large budgets (of around USD\$1.9Bn?! Which movie is this?! Must be amazing...)
- The standard deviation seems extremely large; again, this could be due to a small number of extremely large budgets...
- ... or perhaps some unrealistically *small* budgets (of US\$ 0?!)

2.4 Graphical and numerical summaries

From the listing above we can make a few observations:

- The mean budget is larger than the median: $\sim$ US\$ 41M, compared to US\$ 25M. This indicates a **positively skewed** dataset, as the mean is being 'pulled up' by some very large budgets (of around USD\$1.9Bn?! Which movie is this?! Must be amazing...)
- The standard deviation seems extremely large; again, this could be due to a small number of extremely large budgets...
- ... or perhaps some unrealistically *small* budgets (of US\$ 0?!)

2.4 Graphical and numerical summaries

From the listing above we can make a few observations:

- The mean budget is larger than the median: $\sim$ US\$ 41M, compared to US\$ 25M. This indicates a **positively skewed** dataset, as the mean is being 'pulled up' by some very large budgets (of around USD\$1.9Bn?! Which movie is this?! Must be amazing...)
- The standard deviation seems extremely large; again, this could be due to a small number of extremely large budgets...
- ... or perhaps some unrealistically *small* budgets (of US\$ 0?!)

2.4 Graphical and numerical summaries

```
1 # Let's find out the position of the most expensive movie
2 which(movies$Budget == max(movies$Budget, na.rm=TRUE))
3 [1] 1211
4
5 # What's the name of this movie?
6 movies$Title[1211]
7 [1] "Enthiran"
8 # Never heard of it!
9
10 # Which movies were made for free?!
11 movies$Title[which(movies$Budget==0)]
12 [1] "Homefront"      "Just Friends"    "Knock Knock"
13 [4] "Now You See Me" "Safe House"      "Snitch"
```

2.4 Graphical and numerical summaries

What about some **graphical summaries**?

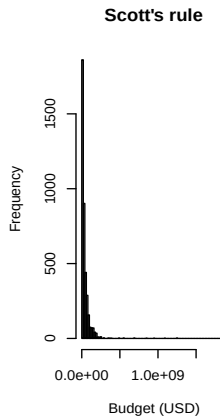
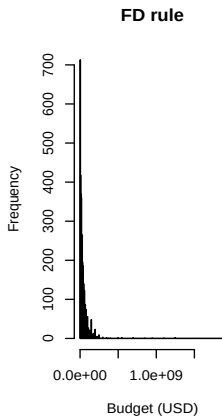
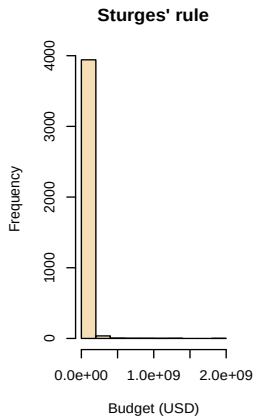
By far the most commonly-used graphics are **histograms** and **boxplots**.

Of course, there are so many more graphical options available in R; you should consult Chapter 3 of your MAS1802 notes for a more comprehensive coverage.

2.4 Graphical and numerical summaries

```
1 # Partitions the plotting space: 1 row, 3 columns
2 par(mfrow = c(1,3))
3
4 # Now we'll try three different methods for producing histograms
5
6 # 1. Sturges' rule; the default, tries to find an optimal number of bars
7 hist(movies$Budget, col="wheat", breaks="sturges",
8      xlab="Budget (USD)", main="Sturges' rule")
9
10 # 2. Freedman-Diaconis's rule; tries to optimise the class width
11 hist(movies$Budget, col="gold", breaks="fd",
12      xlab="Budget (USD)", main="FD rule")
13
14 # 3. Scott's rule; also tries to optimise the class width
15 hist(movies$Budget, breaks="scott",
16      xlab="Budget (USD)", main="Scott's rule")
```

2.4 Graphical and numerical summaries

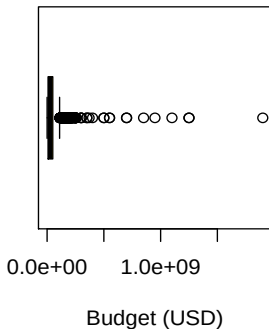


2.4 Graphical and numerical summaries

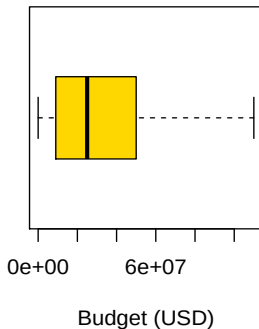
```
1 par(mfrow=c(1,2))
2 # Let's try a boxplot now
3 boxplot(movies$Budget, col="gold",
4         main="Boxplot of movie budgets", horizontal=TRUE, xlab
5         ="Budget (USD)")
6 # Outliers are spoiling things a bit. Let's remove them
7 boxplot(movies$Budget, col="gold",
8         main="Boxplot of movie budgets", horizontal=TRUE, xlab
9         ="Budget (USD)", outline=FALSE)
```

2.4 Graphical and numerical summaries

Boxplot of movie budgets



Boxplot of movie budgets



2.4 Graphical and numerical summaries

- The histograms reveal just how positively skewed the movie budget data are
- They also show how care should be taken with the method chosen to produce the histogram – of course, this is a subjective choice, and some experimentation might be necessary
- The boxplots also indicate just how skewed our dataset is – to get a handle on the distribution of most of our data, it might be preferable to produce the boxplot with the outliers removed

2.4 Graphical and numerical summaries

- The histograms reveal just how positively skewed the movie budget data are
- They also show how care should be taken with the method chosen to produce the histogram – of course, this is a subjective choice, and some experimentation might be necessary
- The boxplots also indicate just how skewed our dataset is – to get a handle on the distribution of most of our data, it might be preferable to produce the boxplot with the outliers removed

2.4 Graphical and numerical summaries

- The histograms reveal just how positively skewed the movie budget data are
- They also show how care should be taken with the method chosen to produce the histogram – of course, this is a subjective choice, and some experimentation might be necessary
- The boxplots also indicate just how skewed our dataset is – to get a handle on the distribution of most of our data, it might be preferable to produce the boxplot with the outliers removed

2.4 Graphical and numerical summaries

Command	Comment	Example
<code>table</code>	Contingency table	<code>table(x)</code>
<code>barplot</code>	Generate a bar chart	<code>barplot(table(x))</code>
<code>hist</code>	Histogram	<code>hist(x)</code>
<code>plot</code>	Scatterplot	<code>plot(x, y)</code>
<code>points</code>	Add points to a plot	<code>points(x, y)</code>
<code>lines</code>	Add lines to a plot	<code>lines(x, y)</code>
<code>boxplot</code>	Box and whisker plot	<code>boxplot</code>

Example 2.4



Load the `music` dataset into RStudio, just as you did in Example 2.3.

- (a) Find the mean and standard deviation for the duration of tracks in this dataset.
- (b) Produce histograms for track duration and danceability; put these histograms on the same panel of plots, use different colours for the histograms (you can choose whichever colours you like!) and make sure you provide suitable titles and labels on the axes.

Example 2.4

- (c) You are asked to investigate whether or not the duration of tracks has changed over time.
- (i) Use the `plot` command in R to produce a scatterplot of the year each track was released against track duration.
 - (ii) Try out the following code, and comment on what you see in the plots in part (c):

```
1 boxplot(music$duration ~ music$decade)
```

- (d) On average, how do energy scores for “Hip Hop” tracks compare to “Electronic” tracks?

Example 2.5

The code in the following listing will download COVID-19 data from the website of the *European Centre for Disease Prevention and Control*.¹

```
1 covid = read.csv("https://opendata.ecdc.europa.eu/  
covid19/casedistribution/csv", na.strings = "",  
2 fileEncoding = "UTF-8-BOM")
```

¹Note: This dataset is dynamic – it is updated every day, so the results you get will differ to those shown in the listings below.

Example 2.5

- (a) Use the `head` command to take a look at the first 6 rows of the dataframe, to get a feel for the information contained in the dataset. Also use the `dim` command.

```
1 head(covid)
2   dateRep day month year cases deaths
3   countriesAndTerritories geold
4 1 08/10/2020 8 10 2020 68 1
5   Afghanistan AF
6 2 07/10/2020 7 10 2020 62 2
7   Afghanistan AF
8 3 06/10/2020 6 10 2020 145 5
9   Afghanistan AF
10 4 05/10/2020 5 10 2020 44 0
11   Afghanistan AF
12 5 04/10/2020 4 10 2020 7 4
13   Afghanistan AF
14 6 03/10/2020 3 10 2020 5 0
15   Afghanistan AF
16 countryterritoryCode popData2019 continentExp
```

Example 2.5

```
1 dim(covid)
2 [1] 47689    12
```

- 47, 689 rows (on 08/10/20; will get bigger each day!)
- 12 columns – 12 variables...

```
1 colnames(covid)
2 [1] "dateRep"
3 [2] "day"
4 [3] "month"
5 [4] "year"
6 [5] "cases"
7 [6] "deaths"
8 [7] "countriesAndTerritories"
9 [8] "geold"
10 [9] "countryterritoryCode"
11 [10] "popData2019"
12 [11] "continentExp"
13 [12] "Cumulative_number_for_14_days_of_COVID.19_cases_
    per_100000"
```

Example 2.5

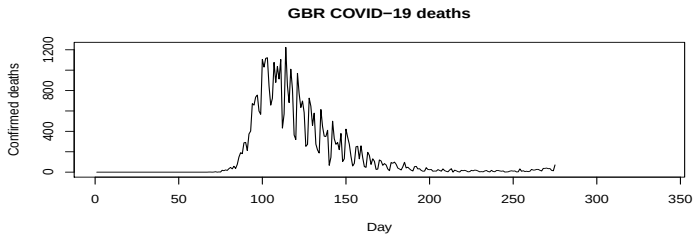
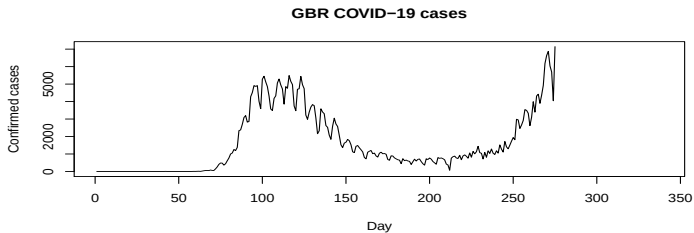
- (b) We can use the `attach` command to directly access the columns in the dataframe by their headings. This might make it a bit easier to isolate data for certain countries:

```
1 # Enable access to columns via their headings
2 attach(covid)
3
4 # Find the total number of cases, and deaths, globally (08/10/20)
5 sum(cases)
6 [1] 36194764
7 sum(deaths)
8 [1] 1053961
9
10 # What about Great Britain? (Data as of 08/10/20)
11 sum(cases[countryterritoryCode=="GBR"], na.rm=TRUE)
12 [1] 544275
13 sum(deaths[countryterritoryCode=="GBR"], na.rm=TRUE)
14 [1] 42515
```

Example 2.5

```
1 # What about a plot of cases/deaths through time in Great Britain?
2 par(mfrow=c(1,2))
3
4 plot(rev(cases[countryterritoryCode=="GBR"]), type="l",
      xlab="Day", ylab="Confirmed cases", main="GBR
      COVID-19 cases")
5
6 plot(rev(deaths[countryterritoryCode=="GBR"]), type="l",
      xlab="Day", ylab="Confirmed deaths", main="GBR
      COVID-19 deaths")
```

Example 2.5



Example 2.5

- (i) Find the total number of confirmed cases, and deaths, due to COVID-19, for three other countries of your choosing.

```
1 # For example, for USA:
2 sum( cases[ countryterritoryCode=="USA" ] , na.rm=TRUE)
3 [1] 7549794
4
5 sum( deaths[ countryterritoryCode=="USA" ] , na.rm=TRUE)
6 [1] 211801
```

Example 2.5

- (ii) Why is it necessary to use the `rev` command in lines 3 and 4 in the above listing?

Because the dataframe is updated from the top, meaning the first entry for each country is for the most recent time-point.

- (iii) For the three other countries you chose in part (i), produce plots showing confirmed cases and deaths through time.

```
1 # For example, for the USA:
2 par(mfrow=c(1,2))
3
4 plot(rev(cases[countryterritoryCode=="USA"]), type="l", xlab="Day", ylab="Confirmed cases", main="USA COVID-19 cases")
5
6 plot(rev(deaths[countryterritoryCode=="USA"]), type="l", xlab="Day", ylab="Confirmed deaths", main="USA COVID-19 deaths")
```

Example 2.5

- (ii) Why is it necessary to use the `rev` command in lines 3 and 4 in the above listing?

Because the dataframe is updated from the top, meaning the first entry for each country is for the most recent time-point.

- (iii) For the three other countries you chose in part (i), produce plots showing confirmed cases and deaths through time.

```
1 # For example, for the USA:
2 par(mfrow=c(1,2))
3
4 plot(rev(cases[countryterritoryCode=="USA"]), type=
      "l", xlab="Day", ylab="Confirmed cases", main="
      USA COVID-19 cases")
5
6 plot(rev(deaths[countryterritoryCode=="USA"]), type=
      "l", xlab="Day", ylab="Confirmed deaths", main=
      "USA COVID-19 deaths")
```

Example 2.5

To explore other country codes, try:

```
1 unique(countryterritoryCode)
```

Home Study 2: To be completed by Monday 26th October

- That's this Home Study activity completed. Well done!
- Before the end of **Monday 26th October**, you should:
 - Complete **Example 2.4**
 - Answers to the questions in Example 2.4 will be reviewed at the start of the video **Home Study 3**, to be released next week
 - Read through the lecture notes to review what we have covered this week: everything up to the end of Section 2.5
- See you next week! We have a computer practical session scheduled to take place on **Tuesday 27th October**

Home Study 2: To be completed by Monday 26th October

- That's this Home Study activity completed. Well done!
- Before the end of **Monday 26th October**, you should:
 - Complete **Example 2.4**
 - Answers to the questions in Example 2.4 will be reviewed at the start of the video **Home Study 3**, to be released next week
 - Read through the lecture notes to review what we have covered this week: everything up to the end of Section 2.5
- See you next week! We have a computer practical session scheduled to take place on **Tuesday 27th October**

Home Study 2: To be completed by Monday 26th October

- That's this Home Study activity completed. Well done!
- Before the end of **Monday 26th October**, you should:
 - Complete **Example 2.4**
 - Answers to the questions in Example 2.4 will be reviewed at the start of the video **Home Study 3**, to be released next week
 - Read through the lecture notes to review what we have covered this week: everything up to the end of Section 2.5
- See you next week! We have a computer practical session scheduled to take place on **Tuesday 27th October**

Home Study 2: To be completed by Monday 26th October

- That's this Home Study activity completed. Well done!
- Before the end of **Monday 26th October**, you should:
 - Complete **Example 2.4**
 - Answers to the questions in Example 2.4 will be reviewed at the start of the video **Home Study 3**, to be released next week
 - Read through the lecture notes to review what we have covered this week: everything up to the end of Section 2.5
- See you next week! We have a computer practical session scheduled to take place on **Tuesday 27th October**

Home Study 2: To be completed by Monday 26th October

- That's this Home Study activity completed. Well done!
- Before the end of **Monday 26th October**, you should:
 - Complete **Example 2.4**
 - Answers to the questions in Example 2.4 will be reviewed at the start of the video **Home Study 3**, to be released next week
 - Read through the lecture notes to review what we have covered this week: everything up to the end of Section 2.5
- See you next week! We have a computer practical session scheduled to take place on **Tuesday 27th October**