

# Exploring Data

Data & Graphical Summaries

**STAT5002**

*The University of Sydney*

Aug 2025



THE UNIVERSITY OF  
**SYDNEY**

# Exploring Data

## Topic 1: Data & Graphical Summaries

What type of data do we have & how can we visualise it?

## Topic 2: Numerical Summaries

What are the main features of the data?

# Outline

Initial data analysis

Identifying variables

Graphical summaries

- Barplot
- Histogram
- Scatter plot
- Boxplot

Logical operators

# Data story: what causes Australian road fatalities?

We are going to investigate [data from the Australian Bureau of Statistics \(ABS\)](#) (last updated Nov 2023).



# Variables

```
1 # Read in data
2 data = read.csv("data/2023fatalities.csv", header = TRUE)
3 # Names of Variables
4 names(data)

[1] "Crash.ID"                      "State"
[3] "Month"                          "Year"
[5] "Dayweek"                        "Time"
[7] "Crash.Type"                     "Bus.Involvement"
[9] "Heavy.Rigid.Truck.Involvement" "Articulated.Truck.Involvement"
[11] "Speed.Limit"                    "Road.User"
[13] "Gender"                         "Age"
[15] "National.Remoteness.Areas"      "SA4.Name.2021"
[17] "National.LGA.Name.2021"         "National.Road.Type"
[19] "Christmas.Period"              "Easter.Period"
[21] "Age.Group"                      "Day.of.week"
[23] "Time.of.day"                   "X"
```

## Data dictionary

# Statistical Thinking

Possible research questions:

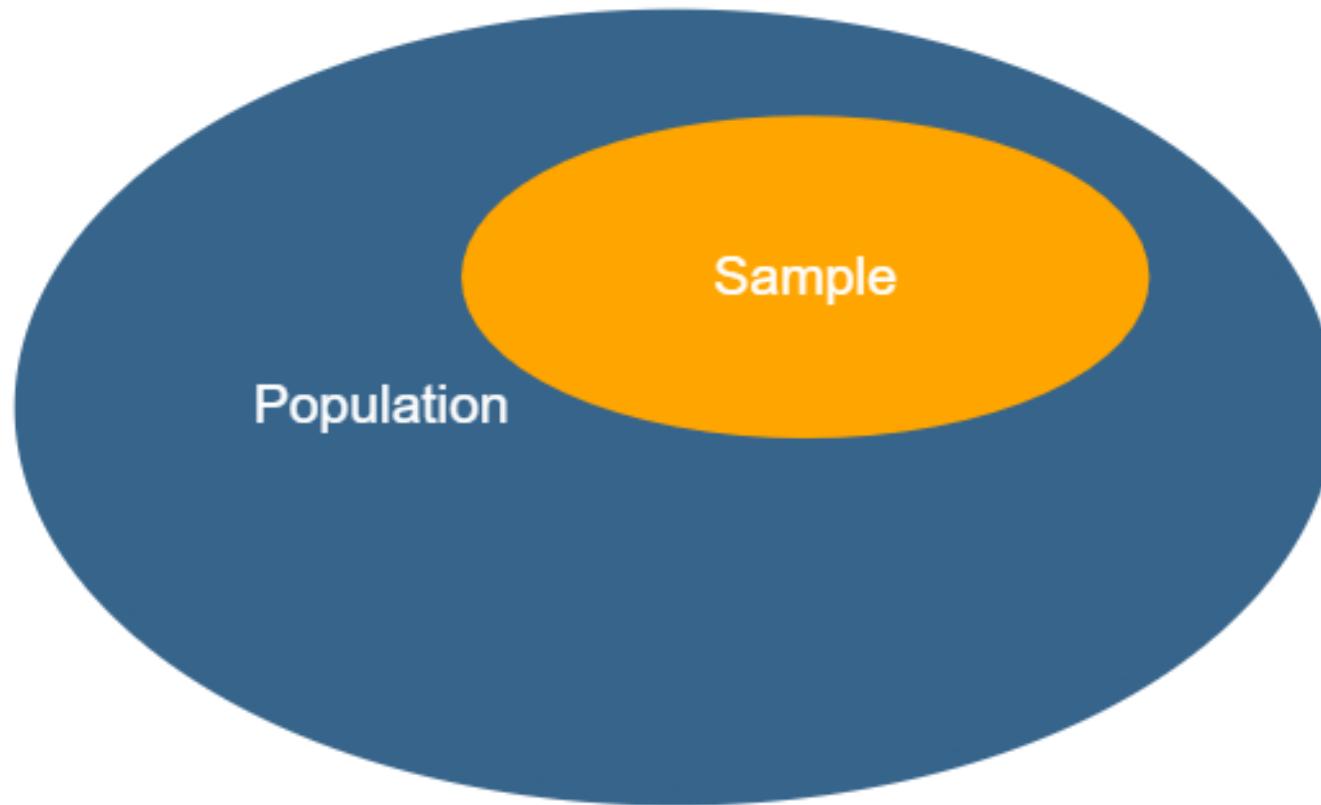
- How many road fatalities have there been so far this year, and how does it compare to last year?
- What is the most common day and time for a crash?
- Does gender affect the type of road fatality?
- What is the chance that a motorcycle rider is involved in a road fatality?
- How many people **wear seatbelts**?

# Initial data analysis

# Sample vs Population

Data is **information** about the set of **subjects** being studied (like road fatalities).

- The target population comprises all relevant subjects of interest.
- Most commonly, data refers to the **sample** (not the population), which is a manageable subset selected to make the study feasible.



# Initial data analysis (IDA)

**Initial data analysis** is a first general look at the data, without formally answering the research questions.

- IDA helps you to see whether the data can answer your research questions.
- IDA may lead to new research questions.
- IDA can
  - identify the data's main qualities;
  - suggest the population from which a sample derives.

# What's involved in IDA

Initial Data Analysis commonly involves:

- data background: checking the quality and integrity of the data
- data structure: what information has been collected?
- data wrangling: scraping, cleaning, tidying, reshaping, splitting, combining
- data summaries: graphical and numerical

Here we focus on **structure & graphical summaries** for qualitative and quantitative data.

# Structure of the data

# Variables

A **variable** measures or describes some attribute of the subjects.

- Data with  $p$  variables is said to have **dimension  $p$** .

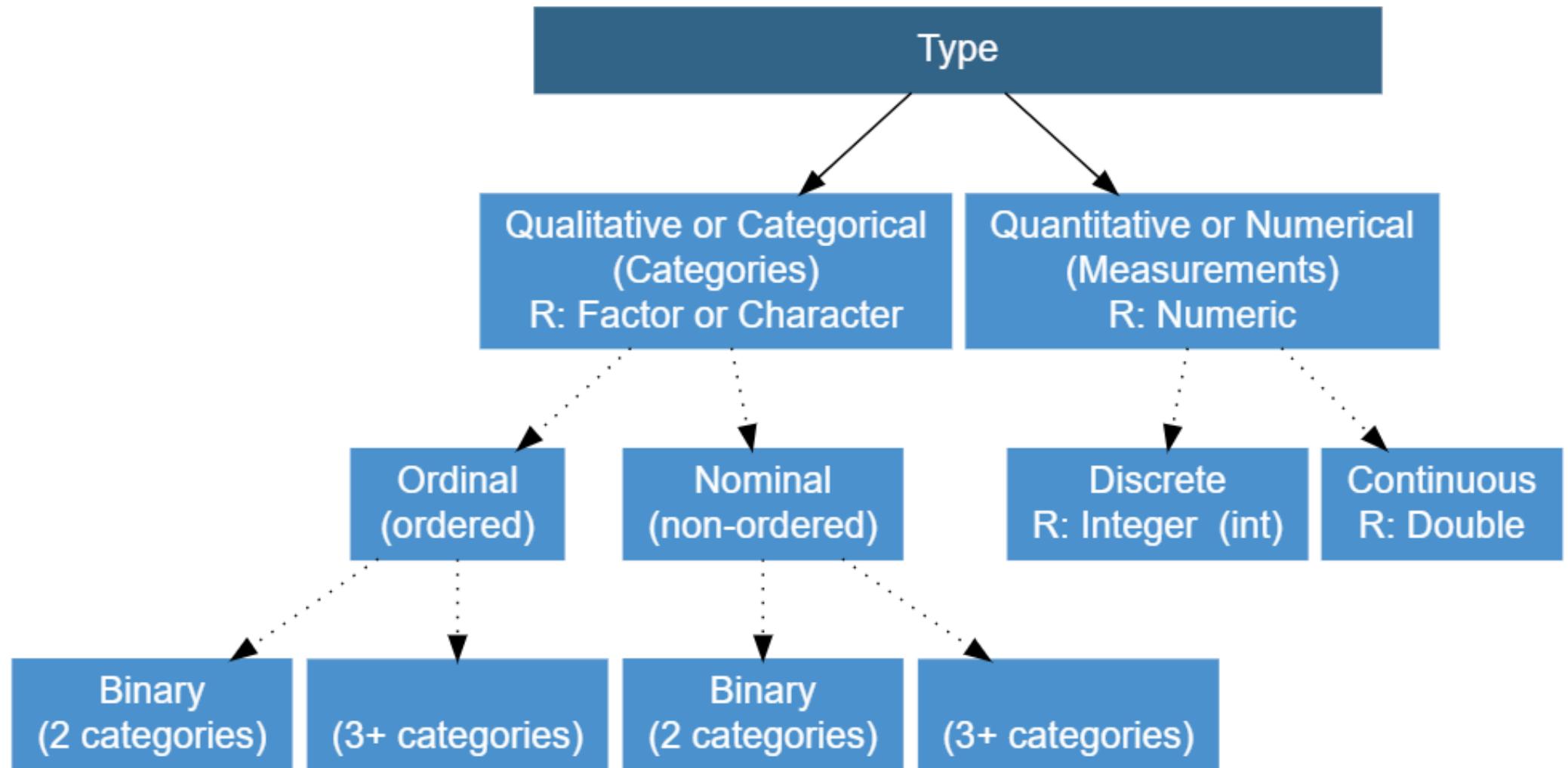
How many variables does the road fatality data have?

- The road fatality data has dimension  $p = 23$ , as the CrashID serves as an anonymous identifier.

```
1 # Size of Data (rows and columns)
2 dim(data)

[1] 55360    24
```

# Types of variables



# Statistical Thinking

Classify the variable Age in the Road Fatality Data.

- Technically Age is a quantitative, continuous variable, but here the ages have been reported as discrete 'integer' (by rounding down to the nearest year).
- Age may be also be recorded as a qualitative variable in a survey, as respondents may be more willing to give their age category. (e.g. 18-24)
- However, it is more precise to record quantitative data if possible.

# Change variable types in R

```
1 # Structure of Data (tells us how each variable is stored in R)
2 str(data, vec.len = 2)
```

```
'data.frame': 55360 obs. of 24 variables:  
$ Crash.ID : int 20237008 20234009 20233087 20233149 20233190 ...  
$ State : chr "NT" "SA" ...  
$ Month : int 10 10 10 10 10 ...  
$ Year : int 2023 2023 2023 2023 2023 ...  
$ Dayweek : chr "Friday" "Saturday" ...  
$ Time : chr "" "03:00" ...  
$ Crash.Type : chr "Single" "Single" ...  
$ Bus.Involvement : chr "No" "No" ...  
$ Heavy.Rigid.Truck.Involvement: chr "No" "No" ...  
$ Articulated.Truck.Involvement: chr "No" "No" ...  
$ Speed.Limit : chr "-9" "100" ...  
$ Road.User : chr "Driver" "Driver" ...  
$ Gender : chr "Female" "Male" ...  
$ Age : int 24 22 19 37 35 ...  
$ National.Remoteness.Areas : chr "" "Outer Regional Australia" ...  
$ SA4.Name.2021 : chr "" "Barossa - Yorke - Mid North" ...  
$ National.LGA.Name.2021 : chr "" "Yorke Peninsula" ...  
$ National.Road.Type : chr "" "Local Road" ...  
$ Christmas.Period : chr "No" "No" ...
```

# Change variable types in R

```
1 # Change qualitative variables stored as 'numeric' to 'factors'  
2 data$Crash.ID = as.factor(data$Crash.ID)  
3 data$Month = as.factor(data$Month)
```

```
1 # New structure of Data Display the first 5 variables using list.len=5  
2 str(data, list.len = 5)
```

```
'data.frame': 55360 obs. of 24 variables:  
 $ Crash.ID : Factor w/ 49903 levels "19891001","19891002",..: 49880 49646 49506 49568 49609  
49471 49805 49849 49568 49872 ...  
 $ State : chr "NT" "SA" "Qld" "Qld" ...  
 $ Month : Factor w/ 12 levels "1","2","3","4",..: 10 10 10 10 10 10 10 10 10 10 ...  
 $ Year : int 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...  
 $ Dayweek : chr "Friday" "Saturday" "Saturday" "Sunday" ...  
[list output truncated]
```

```
1 # Change quantitative variables stored as 'characters' to 'numeric'  
2 data$Speed.Limit = as.numeric(data$Speed.Limit)
```

```
1 # New structure of Data Display variables 11 to 15  
2 str(data[c(11, 12, 13, 14, 15)])
```

```
'data.frame': 55360 obs. of 5 variables:  
 $ Speed.Limit : num -9 100 80 60 100 70 60 80 60 60 ...  
 $ Road.User : chr "Driver" "Driver" "Driver" "Passenger" ...  
 $ Gender : chr "Female" "Male" "Male" "Male" ...  
 $ Age : int 24 22 19 37 35 32 29 51 39 33 ...  
 $ National.Remoteness.Areas: chr "" "Outer Regional Australia" "Inner Regional Australia" "Inner Regional  
Australia" ...
```

# Graphical summaries

# Graphical summaries

Once we've identified the variables, we can summarise the data, both graphically and numerically, in order to identify and highlight the main features of interest. We often start with graphical summaries because 'A (well-designed) picture is worth a thousand words.'

E.g. I didn't finish reading the "Lord of the Ring" books, but the movies are graphical summary the contents of the books. Yes, the specific details are omitted, but the movies told the same meaningful story in lesser time (11 hours vs 455,000 words.)

# Choosing a graphical summary

How to choose an appropriate graphical summary?

- The critical question is: 'What plot is the more informative?' or 'What plot will best highlight features of the data?' or 'What plot will best guide the next analysis?'
- To some extent we use trial and error. We try some standard forms and see what is revealed about the data. One graphical summary can suggest another, and often a combination will highlight different features of the data
- In practice we use computer packages like R to construct summaries.
- However, it is important to understand how to construct graphical summaries 'by hand', so that you understand how to interpret computer output and for your final exam.

# Graphical summaries

Barplot (qualitative data)

# Barplot (qualitative data)

Question: What was the most common day of road fatality?

Step 1: Build a frequency table

```
1 # Select the DayWeek variable from the whole data frame  
2 Dayweek = data$Dayweek  
3 # Produce a frequency table of fatalities per day of the week  
4 table(Dayweek)
```

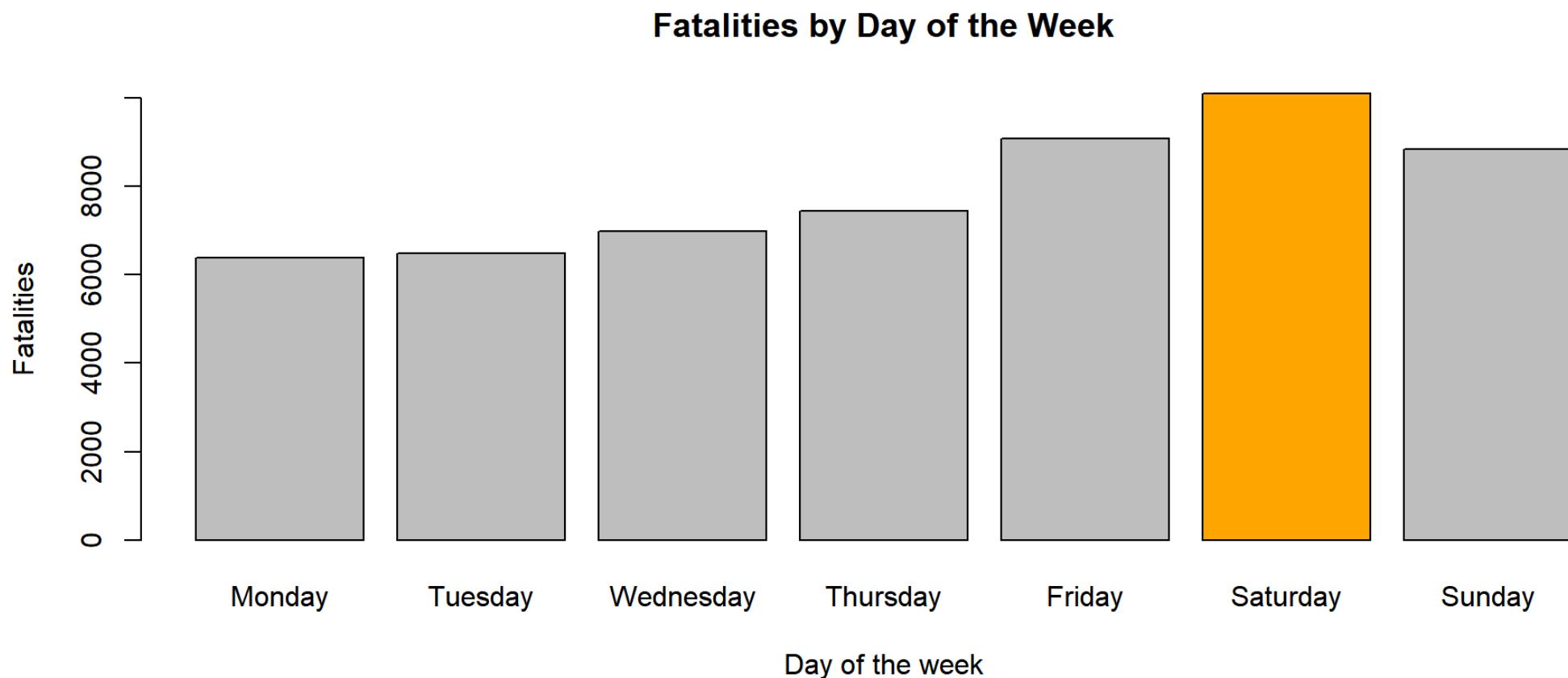
Dayweek	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
	9094	6382	10107	8855	7456	6483	6983

```
1 # Order days  
2 Dayweek = factor(Dayweek, levels = c("Monday", "Tuesday", "Wednesday", "Thursday",  
3 "Friday", "Saturday", "Sunday"))  
4 table(Dayweek)
```

Dayweek	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	6382	6483	6983	7456	9094	10107	8855

## Step 2: Produce a barplot

```
1 # Produce a barplot
2 barplot(table(Dayweek), col = mycols, main = "Fatalities by Day of the Week", ylab = "Fatalities",
3         xlab = "Day of the week")
```



# Statistical Thinking

What was the most common day of road fatality?

- Saturday

Why might that be the case?

- More volume of cars on the road, or people driving faster?

What data would you need to check your hypotheses?

- Data on volume and speed of cars on the road each day.

# Double barplot

Things get more interesting when we consider 2 qualitative variables.

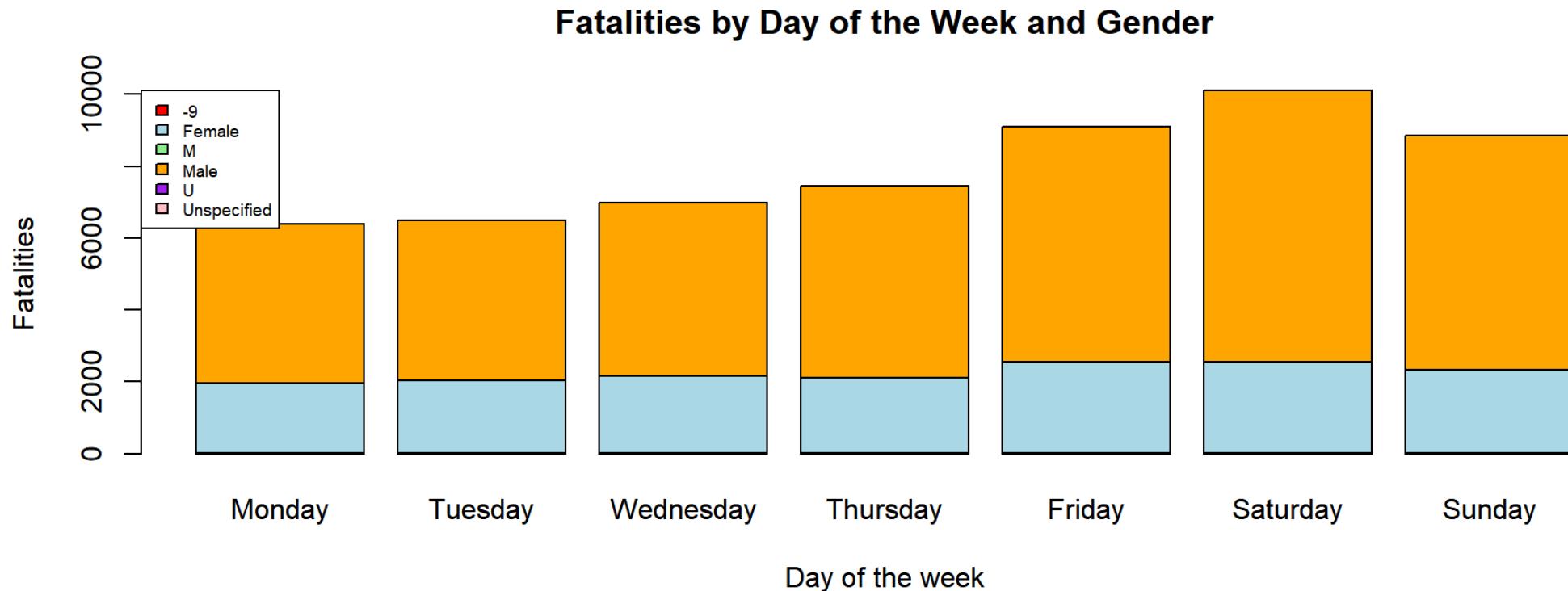
```
1 # Select Gender variable
2 Gender = data$Gender
3
4 # Produce a double frequency table (contingency table)
5 data1 = table(Gender, Dayweek)
6 data1
```

		Dayweek						
		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Gender								
-9		3	2	12	3	10	6	2
Female		1945	2034	2135	2094	2538	2555	2325
M		0	0	1	0	0	0	0
Male		4433	4447	4835	5359	6545	7541	6528
U		1	0	0	0	1	4	0
Unspecified		0	0	0	0	0	1	0

Note: Here Gender refers to biological sex as it was historically recorded in this dataset. [Read more](#).

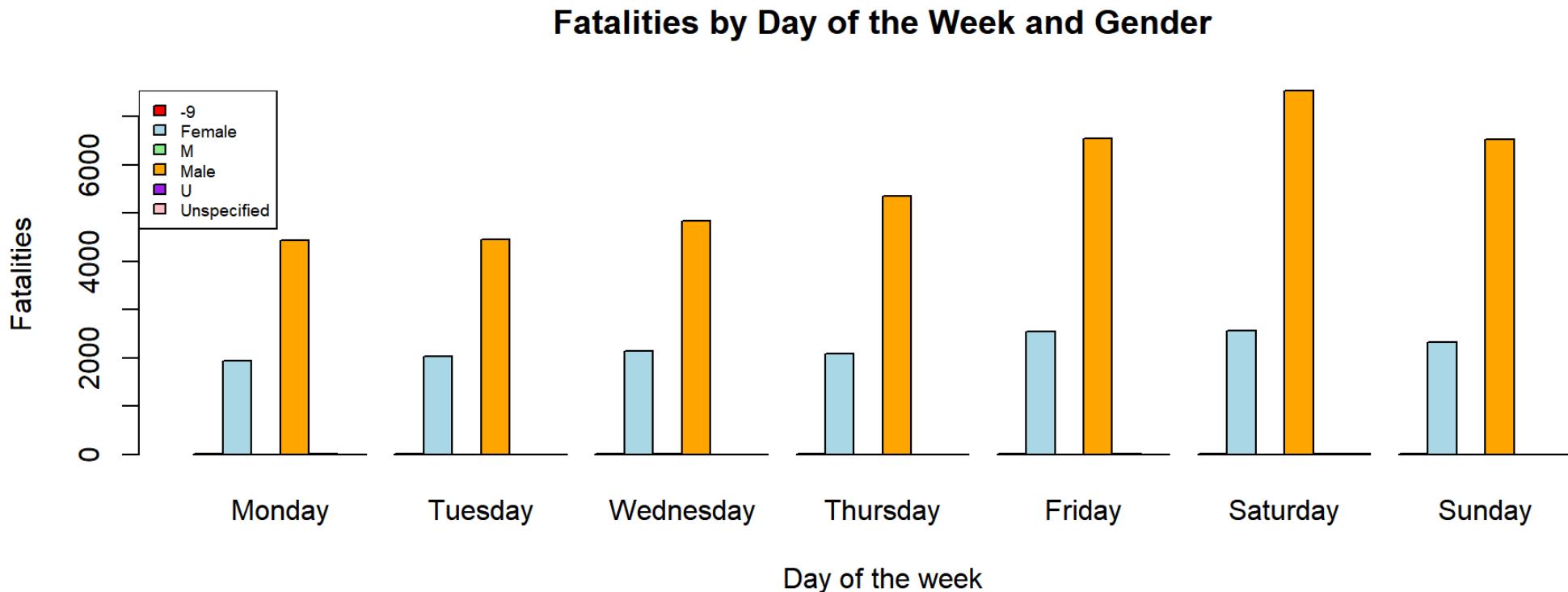
# Stacked barplot

```
1 barplot(data1, main = "Fatalities by Day of the Week and Gender", xlab = "Day of the week",
2         ylab = "Fatalities", col = c("red", "lightblue", "lightgreen", "orange", "purple",
3         "pink"))
4 legend("topleft", legend = rownames(data1), fill = c("red", "lightblue", "lightgreen",
5         "orange", "purple", "pink"), cex = 0.58)
```



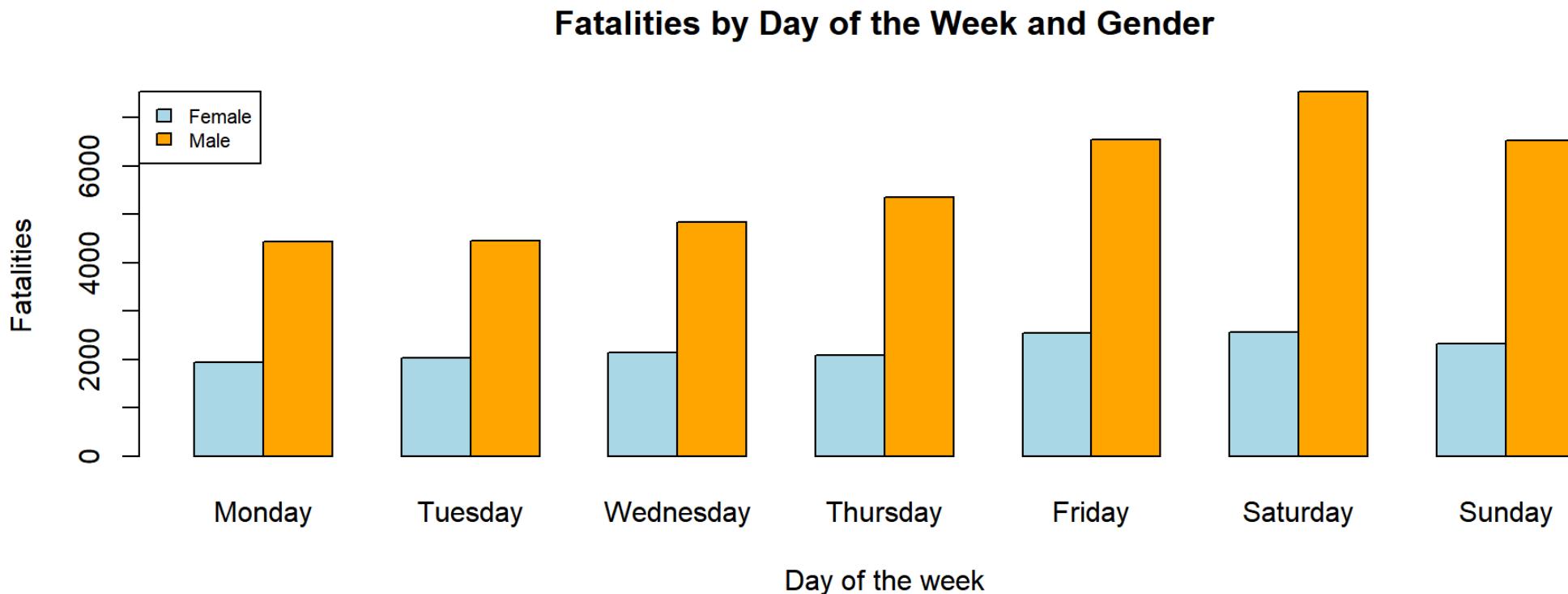
# Side-by-side barplot

```
1 barplot(data1, main = "Fatalities by Day of the Week and Gender", xlab = "Day of the week",
2         ylab = "Fatalities", col = c("red", "lightblue", "lightgreen", "orange", "purple",
3             "pink"), beside = TRUE)
4 legend("topleft", legend = rownames(data1), fill = c("red", "lightblue", "lightgreen",
5             "orange", "purple", "pink"), cex = 0.58)
```



# Side-by-side barplot ignoring '-9', 'M', 'U' and 'Unspecified'

```
1 barplot(data1[-c(1, 3, 5, 6), ], main = "Fatalities by Day of the Week and Gender",
2         xlab = "Day of the week", ylab = "Fatalities", col = c("lightblue", "orange"),
3         legend = rownames(data1[-c(1, 3, 5, 6), ]), beside = TRUE, args.legend = list(x = "topleft",
4                                         cex = 0.7))
```



# Statistical Thinking

Are these plots telling us anything useful? How could they be misread?

- There seems to be a similar proportion of gender fatalities across each day.
- We could posit that men are more likely to be involved in fatal accidents than women. However, perhaps there are more men on the road than women. More data is needed.

# Graphical summaries

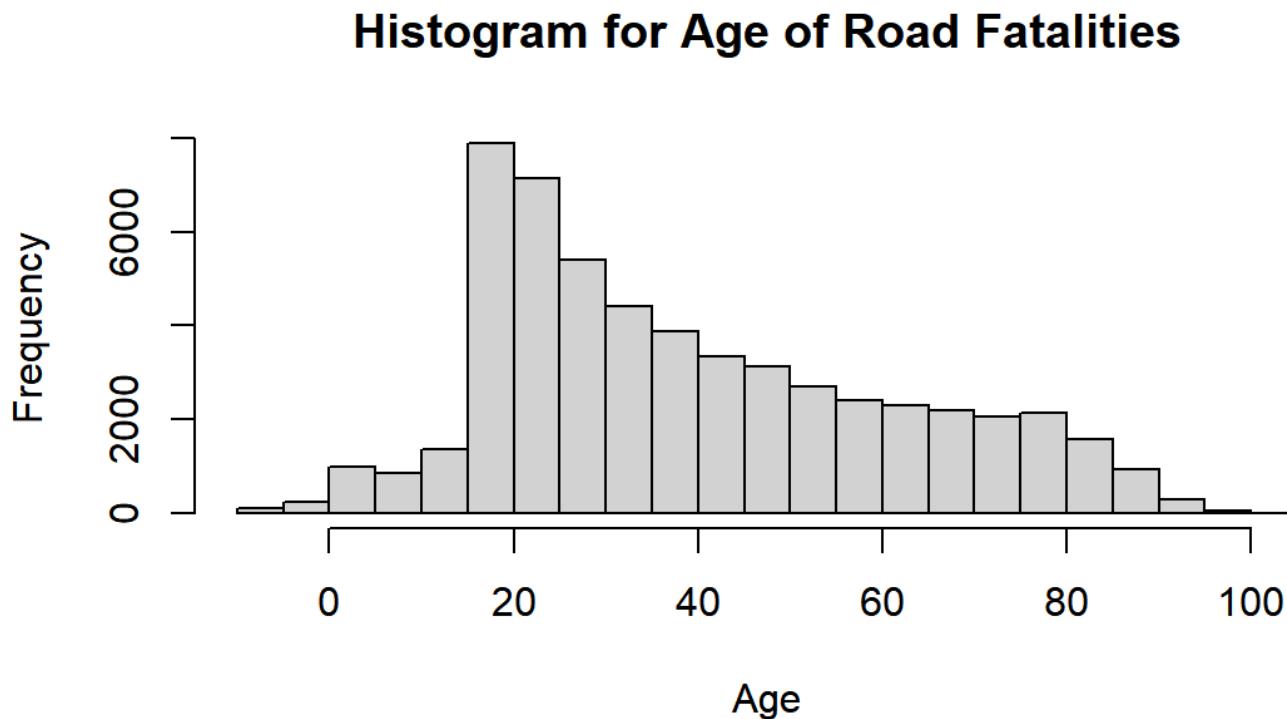
Histogram (quantitative data)

# Histogram

The frequency table can also be used to summarise a set of **quantitative** data, by collecting the data into **class intervals** (or ‘bins’). A histogram highlights the frequency of data in one class interval compared to another.

This is the default histogram generated by R for Age of Road Fatalities .

```
1 hist(data$Age, xlab = "Age", ylab = "Frequency", main = "Histogram for Age of Road Fatalities")
```



We can also provide **user-defined class intervals** and the **density scale**.

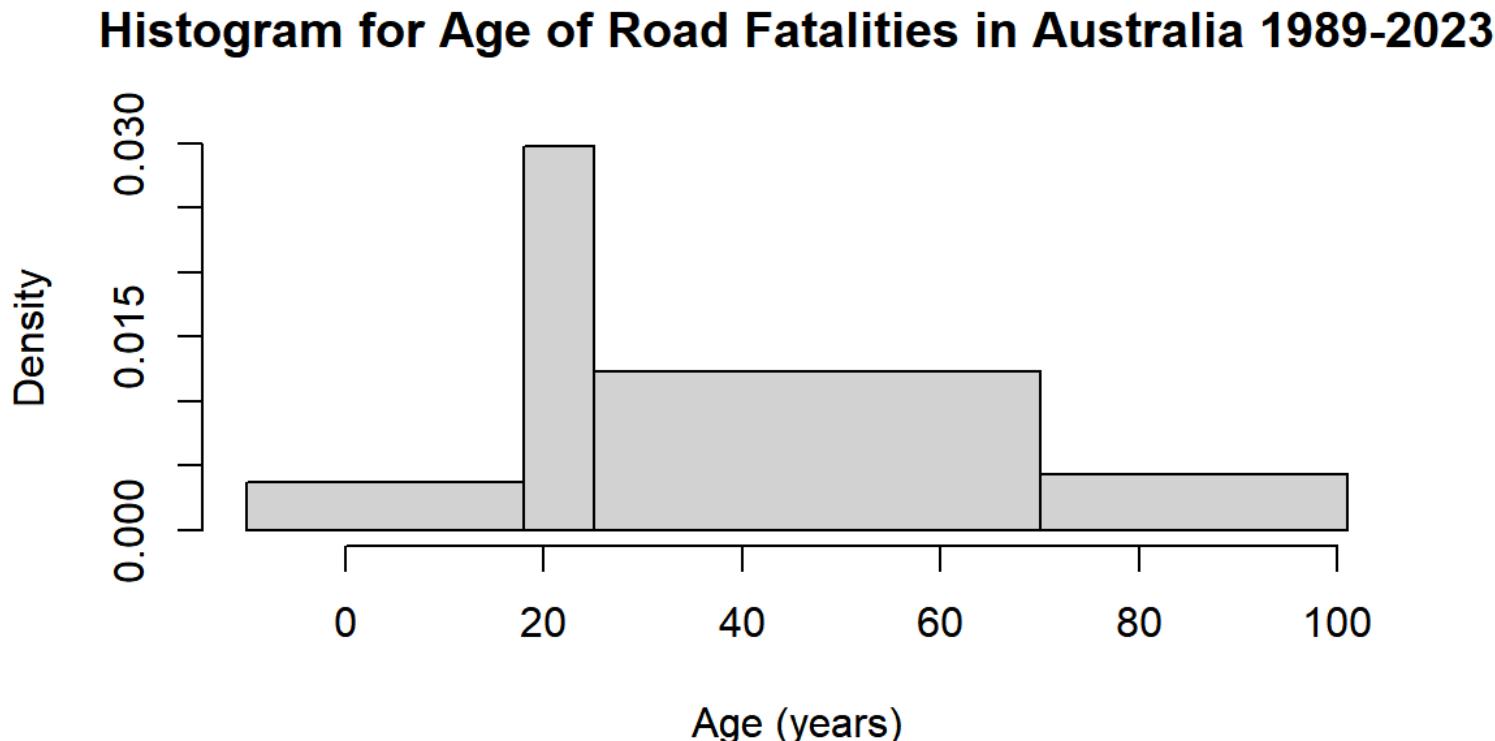
**Q:** What were the most common age groups at which a road fatality occurred?

```
1 # Select the variable Age
2 Age = data$Age
3
4 # Define end points for class intervals
5 breaks = c(-10, 18, 25, 70, 101)
6
7 # Build frequency table
8 table(cut(Age, breaks, right = F))
```

[-10,18)	[18,25)	[25,70)	[70,101)
5747	11541	30566	7504

# Histogram for Age of Road Fatalities in Australia 1989-2023

```
1 hist(Age, br = breaks, right = F, freq = F, xlab = "Age (years)", ylab = "Density",
2       main = "Histogram for Age of Road Fatalities in Australia 1989-2023")
```



- The horizontal scale is divided into **class intervals** with potentially unequal sizes.
- The **area of each block** represents the **proportion** of subjects in that particular class interval.

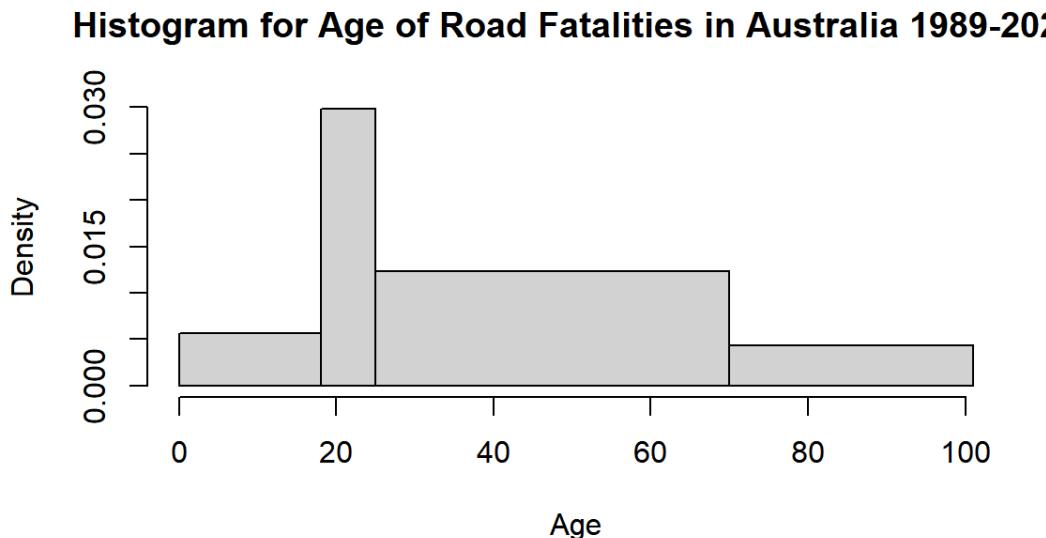
# Data cleaning

Why does the 1st block start below 0?

- **Data Dictionary**: missing values are coded as '-9'.
- It is better to replace the "-9" by "NA".

# Removing Missing Values

```
1 # Replacing the '-9' entries  
2 data$Age[data$Age == -9] = NA  
  
1 hist(data$Age, br = breaks, freq = F, right = F, xlab = "Age", ylab = "Density",  
2       main = "Histogram for Age of Road Fatalities in Australia 1989-2023")
```



How can we interpret this histogram?

- Why is the histogram tallest above [18,25)?
- Which age group have overall most fatalities? (should be [25,70), as it has the largest area)

# Details of density-scale histograms

We will mostly use the **density scale** instead of frequency scale. We will see why when looking at common mistakes.

## Density scale

The area of the whole histogram on the density scale is one (or, in percentage, 100%).

$$\text{area (proportion) of each block} = \frac{\text{number of subjects in the class interval}}{\text{total number of subjects}}$$

$$\text{height (density) of each block} = \frac{\text{proportion of the block}}{\text{length of the class interval}}$$

With the density scale on the vertical axis, the areas of the blocks represent percentage. The area under the histogram over an interval equals the percentage of cases in that interval. The total area under the histogram is 100%.

# Class Endpoints

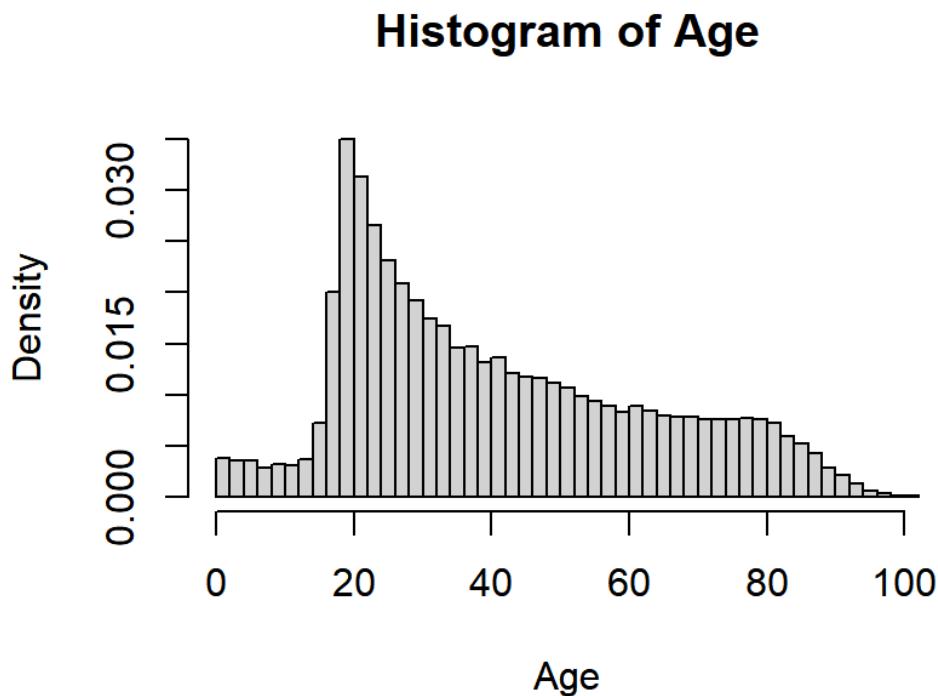
For continuous (quantitative) data, we need an **endpoint convention** for data points that fall on the border of two class intervals.

- If an interval contains the left endpoint but excludes the right endpoint, then an 18 year old would be counted in [18,25) not [0,18).
- We call this left-closed and right-open.
- Similarly, we can also have left-open and right-closed, e.g., (18,25].

# Number of class intervals

Think about how many class intervals (or the sizes of class intervals) you want to have.

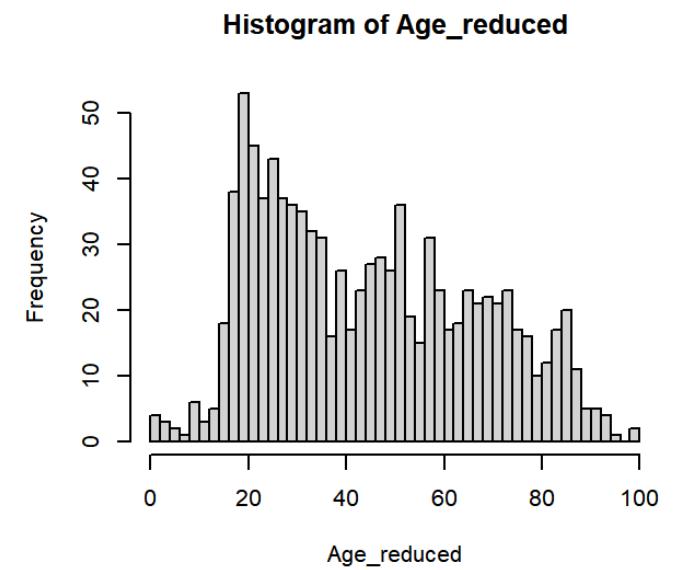
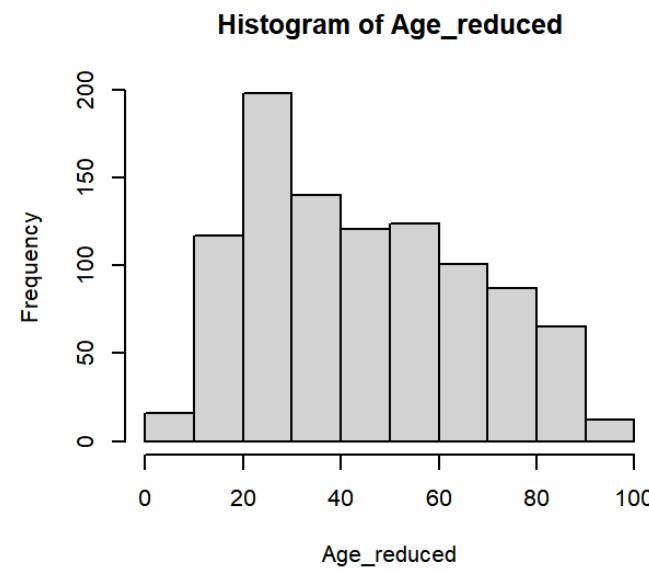
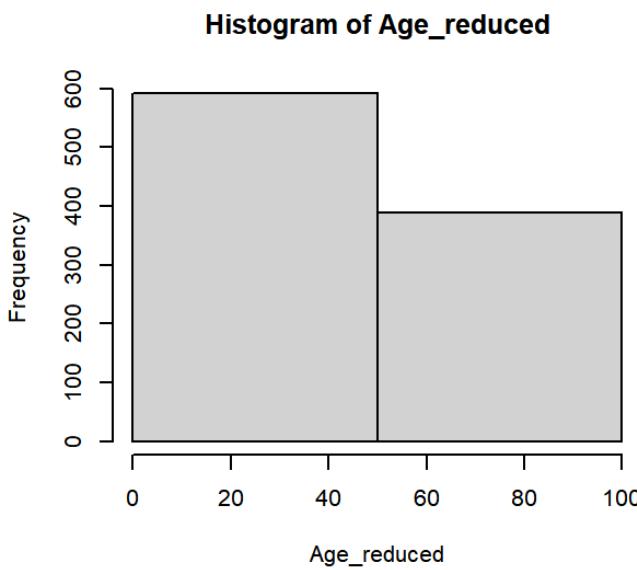
```
1 par(mfrow = c(1, 2)) # This puts the graphic output in 1 row with 2 columns
2 breaks = seq(0, 102, 2) # This sets the breaks to be every 2 numbers
3 hist(Age, br = breaks, freq = F, right = F, xlab = "Age", ylab = "Density")
4 breaks = c(0, 18, 25, 70, 101)
5 hist(Age, br = breaks, freq = F, right = F, xlab = "Age", ylab = "Density")
```



## Using too many or too few class intervals

This can hide the true pattern in the data. As a rule of thumb, use between 10-15 class intervals and make sure you consider the size of the data.

```
1 Age_reduced = Age[1:1000] # only look at subset of data
2 par(mfrow = c(1, 3))
3 hist(Age_reduced, breaks = 3)
4 hist(Age_reduced, breaks = 10)
5 hist(Age_reduced, breaks = 50)
```



# Produce a histogram by hand

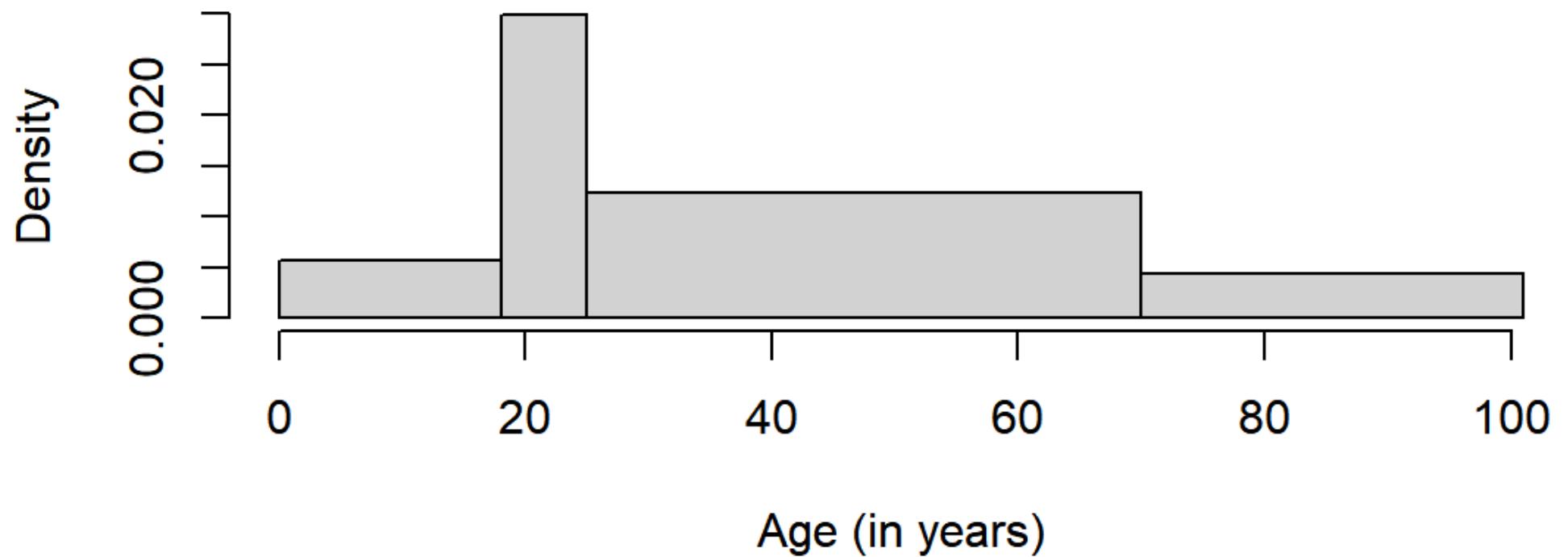
Step 1: Construct the distribution table.

Class intervals	Number of subjects in the interval	%	Height of block
[0,18)	5747	10.4	0.0058
[18,25)	11541	20.8	0.0298
[25,70)	30566	55.2	0.0123
[70,101)	7504	13.6	0.0044
	55360	100	

where Height of block = % per year.

Step 2: Draw the horizontal axis and blocks.

## Histogram for Age of Road Fatalities in Australia 1989-2020



# Summary in R

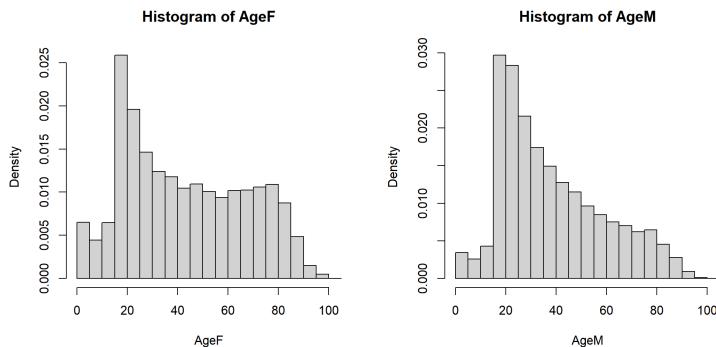
```
1 # Read in data
2 data = read.csv("data/2023Fatalities.csv", header = T)
3
4 # Cleaning
5 data$Age[data$Age == -9] = NA
6
7 # Choose a variable
8 Age = data$Age
9
10 # Choose the class intervals
11 breaks = c(0, 18, 25, 70, 101)
12
13 # Produce a histogram
14 hist(Age, br = breaks, freq = F, right = F, xlab = "Age (in years)", ylab = "Density",
15       main = "Histogram for Age of Road Fatalities in Australia 1989-2020")
```

## Note

- `freq=F` produces the histogram on the density scale.
- `right=F` makes the intervals right-open.

# Comparison among different categories of a variable

```
1 AgeF = data$Age[data$Gender == "Female"] # This selects just the female ages.  
2 AgeM = data$Age[data$Gender == "Male"]  
3 par(mfrow = c(1, 2)) # This puts the graphic output in 1 row with 2 columns  
4 hist(AgeF, freq = F)  
5 hist(AgeM, freq = F)
```



Do you notice any differences between men and women?

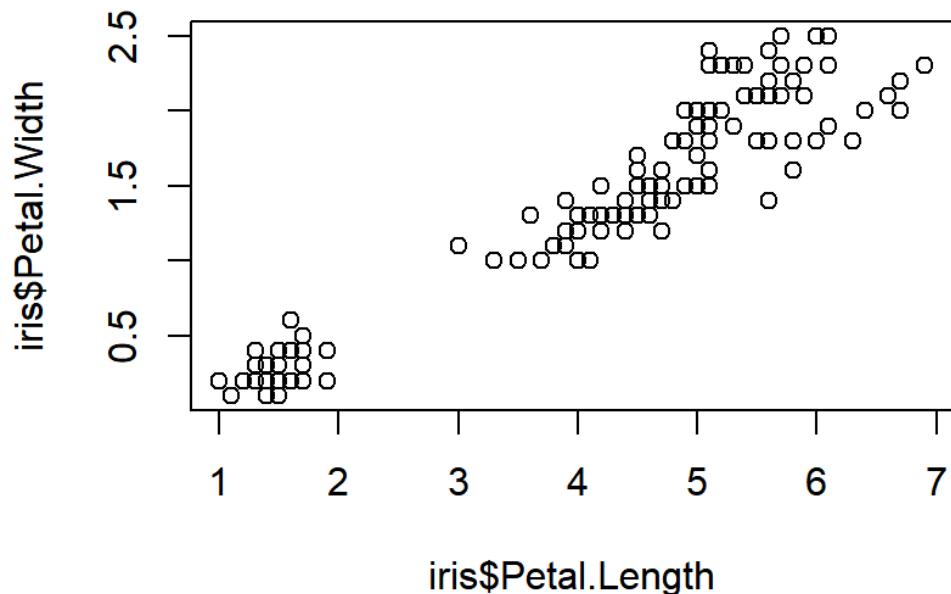
- We can see that for both men and women there is a peak around 16-20 years old, when people are able to first get their license in Australia.
- We can also see that for men, there is a higher peak around this age than for women. (approx. 0.25 vs 0.3).
- The distributions are also different, with men getting into less fatal road crashes as they get older, and women being in approximately the same amount of fatal road crashes between the ages of 30 and 80.

# Other graphical summaries

# Scatter plot

Scatter plots examine the relationship between **two quantitative variables**.

```
1 plot(iris$Petal.Length, iris$Petal.Width)
```



We can see that there is a relationship between petal length and petal width for these iris flowers. As the petal length increases, we can see that the petal width also increases.

# Boxplot

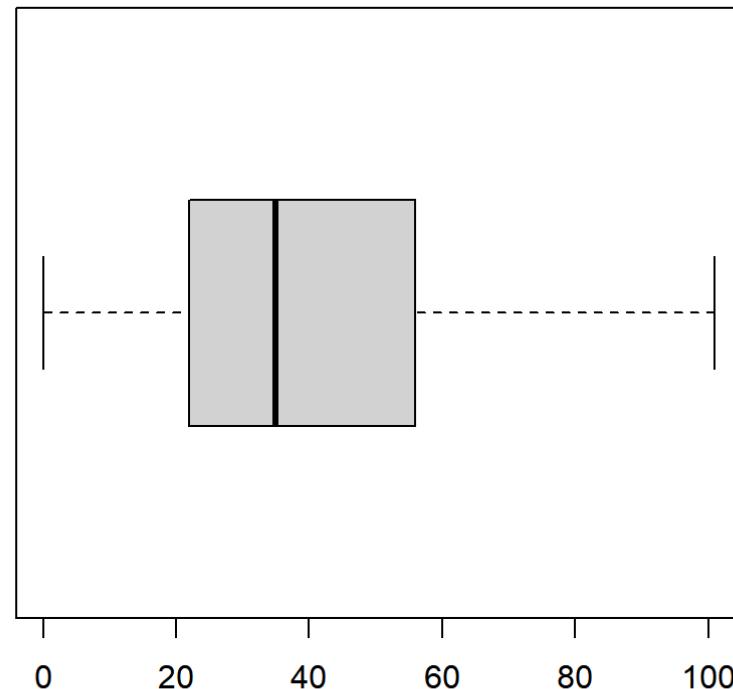
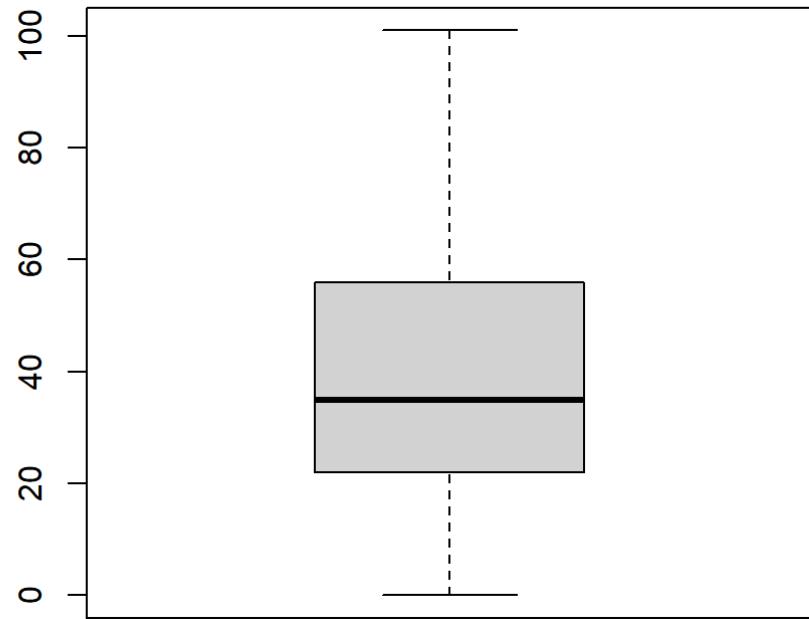
- Boxplots are a graphical representation of the **five numerical summaries** of a data set. It summarises centre, spread and outliers through plotting the median ('middle' data point), the middle 50% of the data in a box, the expected maximum and minimum in the whiskers, and any outliers.
- We will consider how to draw the box plot when we learn about the interquartile range (IQR) in a later lecture.

```
1 Age = data$Age  
2 summary(Age)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	22.00	35.00	39.99	56.00	101.00	116

# Boxplot in R

```
1 par(mfrow = c(1, 2))
2 boxplot(Age) # We can plot the box plot vertically
3 boxplot(Age, horizontal = T) # Or horizontally
```



# Statistical Thinking

What does the simple boxplot reveal about the age of fatalities?

- The box plot is fairly symmetric with no outliers.
- There does not seem to be any extreme ages for fatalities.

# Comparative box plots

A comparative boxplot splits up a quantitative variable by a qualitative variable.

```
1 Gender = data$Gender  
2 # Select each of the data entries in Age if the corresponding data entry in  
3 # Gender is Female  
4 summary(Age[Gender == "Female"])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	22.00	40.00	43.37	64.00	101.00	32

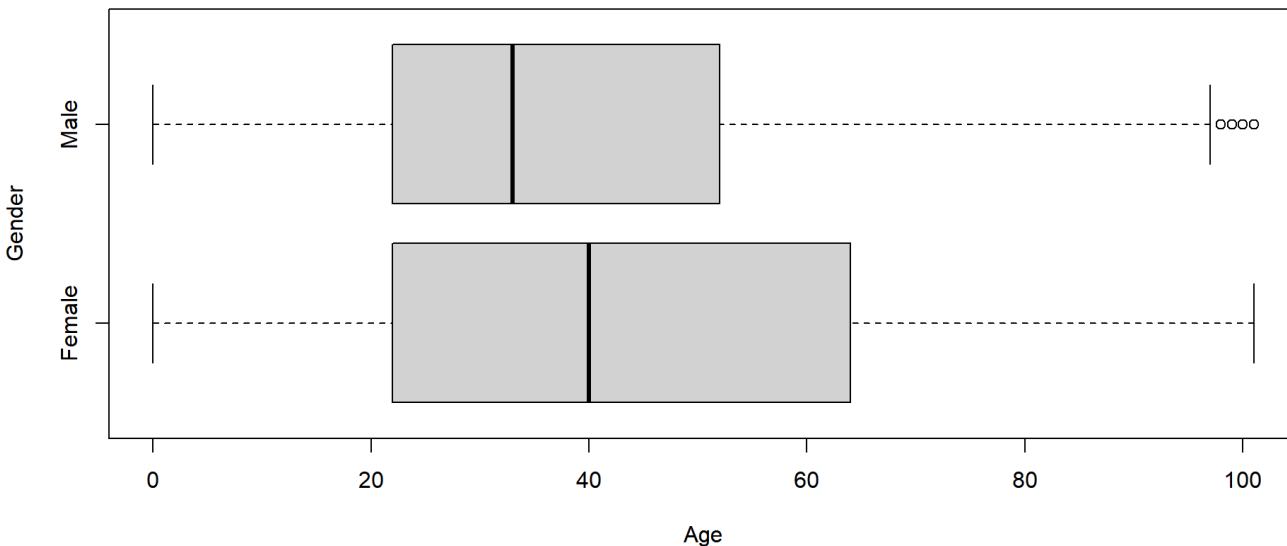
```
1 # Select each of the data entries in Age if the corresponding data entry in  
2 # Gender is Female  
3 summary(Age[Gender == "Male"])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	22.00	33.00	38.69	52.00	101.00	60

Here Age and Gender must have the same number of data points.

# Comparing by Gender

```
1 # Filtering the gender variable to only include 'female' and 'male' values
2 data$Gender = factor(data$Gender, levels = c("Female", "Male"))
3
4 # Selecting the gender variable
5 Gender = data$Gender
6
7 boxplot(Age ~ Gender, horizontal = T)
```



The median ('middle') age is fairly similar but higher for women than for men.

# Logical operators

# Basics of logical operators

The basic logical values in R are `TRUE` (or just `T`) and `FALSE` (or just `F`). These come up very often in R when you are checking an object, or comparing an object to a value or another object, as in  $x > 5$  or  $x > y$ .

Some commonly used logical operators:

<code>&gt;</code>	greater than	<code>≥</code>	greater than or equal to
<code>&lt;</code>	less than	<code>≤</code>	less than or equal to
<code>==</code>	equal to	<code>≠</code>	not equal to

Many of these are exactly what you would expect (like `>`) but remember to use **two** equal signs rather than one when assessing equality (`==` not `=`). If you use just one equal sign, R thinks you are trying to assign a value to an object.

```
1 x = 5 # This assigns the value 5 to x
2 x == 5 # This checks to see if x equals 5
```

```
[1] TRUE
```

# Combining logical conditions

You can combine logical conditions using `&` (and), `|` (or), and `!` (not).

The evaluation of `&` (and): both conditions need to be `TRUE` to have a `TRUE`

<code>&amp;</code>	<code>True</code>	<code>False</code>
<code>True</code>	<code>True</code>	<code>False</code>
<code>False</code>	<code>False</code>	<code>False</code>

Examples:

```
1 x = 10
2 is.numeric(x) & x < 20 # True and True
```

```
[1] TRUE
```

```
1 x = 10
2 is.numeric(x) & x < 0 # True and False
```

```
[1] FALSE
```

The evaluation of `|` (or): need to have at least one of the conditions to be `TRUE` to give a `TRUE` evaluation

I	True	False
True	True	True
False	True	False

Examples:

```
1 x = 10
2 !is.numeric(x) | x < 20 # False and True
```

```
[1] TRUE
```

```
1 x = 10
2 is.character(x) & x < 0 # False and False
```

```
[1] FALSE
```

# Data selection and counting

You can apply logical operators elementwise to vectors or matrices. This can be particularly useful for data selection and counting.

```
1 x = c(-1, 0, 1)
2 # Check each element of x against the condition (elementwise)
3 x <= 0
[1] TRUE TRUE FALSE
```

TRUE and FALSE in R also correspond to integers 1 (TRUE) and 0 (FALSE). This way, they are also useful for counting. For example, how many data points of  $x$  in the following case are less than 5?

```
1 x = 1:10
2 # Check each element of x against the condition (elementwise)
3 x <= 5
[1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE
1 sum(x <= 5) # sum over those TRUEs (data points <= 5)
[1] 5
```

# Combining logical conditions

## Example on data selection

```
1 # creating a data frame
2 rating = 1:5
3 animal = c("koala", "hedgehog", "sloth", "panda", "alligator")
4 country = c("Australia", "Italy", "Peru", "China", "USA")
5 avg_sleep_hours = c(21, 18, 17, 10, 15)
6 sleepers = data.frame(rating, animal, country, avg_sleep_hours)
7 str(sleepers)
```

```
'data.frame': 5 obs. of 4 variables:
 $ rating      : int 1 2 3 4 5
 $ animal       : chr "koala" "hedgehog" "sloth" "panda" ...
 $ country      : chr "Australia" "Italy" "Peru" "China" ...
 $ avg_sleep_hours: num 21 18 17 10 15
```

# Combining logical conditions

Q1: Filter `sleepers` data with rating less than 3

```
1 sleepers1 = sleepers[sleepers$rating < 3, ]  
2 dim(sleepers1)
```

```
[1] 2 4
```

```
1 str(sleepers1)
```

```
'data.frame': 2 obs. of 4 variables:  
 $ rating      : int 1 2  
 $ animal       : chr "koala" "hedgehog"  
 $ country      : chr "Australia" "Italy"  
 $ avg_sleep_hours: num 21 18
```

Q2: Filter `sleepers` data with rating more than 3 and sleeping hour more than 15

```
1 sleepers2 = sleepers[sleepers$rating > 3 & sleepers$avg_sleep_hours > 15, ]  
2 dim(sleepers2)
```

```
[1] 0 4
```

```
1 str(sleepers2)
```

```
'data.frame': 0 obs. of 4 variables:  
 $ rating      : int  
 $ animal       : chr  
 $ country      : chr  
 $ avg_sleep_hours: num
```

# Research question

# Statistical Thinking:

Consider the road fatalities data set.

1. How can we quantify the risk of each age group?
2. Which variables in our data might be useful?
3. Do we need additional data? What kind of data?

```
1 names(data)

[1] "Crash.ID"                      "State"
[3] "Month"                          "Year"
[5] "Dayweek"                        "Time"
[7] "Crash.Type"                     "Bus.Involvement"
[9] "Heavy.Rigid.Truck.Involvement" "Articulated.Truck.Involvement"
[11] "Speed.Limit"                    "Road.User"
[13] "Gender"                         "Age"
[15] "National.Remoteness.Areas"     "SA4.Name.2021"
[17] "National.LGA.Name.2021"        "National.Road.Type"
[19] "Christmas.Period"              "Easter.Period"
[21] "Age.Group"                     "Day.of.week"
[23] "Time.of.day"                   "X"
```

# 1) Who is more likely to be responsible for road safety?

What is the definition of **Road User**?

Road User	Road user type of killed person	Text	Driver Passenger Pedestrian Motorcycle rider Motorcycle pillion passenger Pedal cyclist (Note: includes pillion passenger) Other/-9

# 1) Only count those deaths where the person is the driver

```
1 data.driver = data[data$Road.User == "Driver", ]  
2 Age.driver = data.driver$Age  
3 breaks = c(0, 18, 25, 70, 101)  
4 # Produce a histogram  
5 hist(Age.driver, br = breaks, freq = F, right = F, xlab = "Age (in years)", ylab = "Density",  
6       main = "Age of Road Fatalities of drivers in Australia 1989-2020")
```



## 2) How many drivers are there in each age group?

Find driving licences data with ages: South Australia provides this information.

[data.gov.au](http://data.gov.au)

```
1 # Driver's licence data for SA Q4 2023
2 licence.sa = read.csv("data/drivers-licences-by-postcode-age-and-sex-q4-2023.csv",
3   header = T)
4 str(licence.sa)
```

```
'data.frame': 45975 obs. of 4 variables:
$ PostCode: chr  "0870" "0870" "0870" "0870" ...
$ Age      : int  19 21 23 24 24 25 26 26 27 28 ...
$ Sex      : chr  "Female" "Male" "Male" "Female" ...
$ Total    : chr  "1" "2" "1" "2" ...
```

```
1 # Convert data type of Total to numeric
2 licence.sa$Total = as.numeric(licence.sa$Total)
```

## Pooled data: Put ages into categories using `cut`.

```
1 breaks = c(0, 18, 25, 70, 101)
2 licence.sa$Age = cut(licence.sa$Age, breaks, right = F)
3 head(licence.sa)
```

	PostCode	Age	Sex	Total
1	0870	[18,25)	Female	1
2	0870	[18,25)	Male	2
3	0870	[18,25)	Male	1
4	0870	[18,25)	Female	2
5	0870	[18,25)	Male	1
6	0870	[25,70)	Male	1

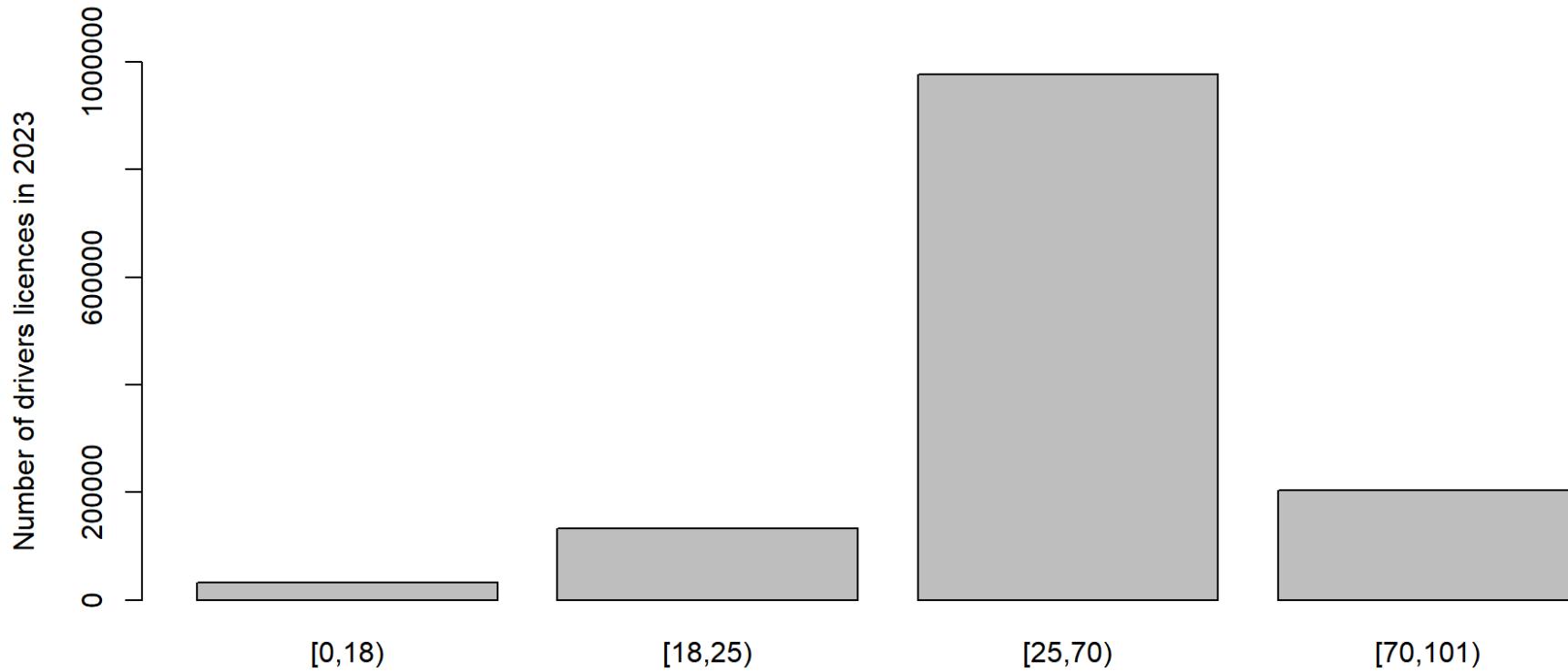
Pool the data for each age category using `aggregate`.

```
1 licence.sa.pooled = aggregate(Total ~ Age, sum, data = licence.sa)
2 head(licence.sa.pooled)
```

	Age	Total
1	[0,18)	33079
2	[18,25)	132769
3	[25,70)	976368
4	[70,101)	204619

## Plot the data with a barplot

```
1 Total = licence.sa.pooled$Total  
2 barplot(Total, names.arg = licence.sa.pooled$Age, ylab = "Number of drivers licences in 2023",  
3         ylim = c(0, 1000000))
```



## Re-visit Step 1)

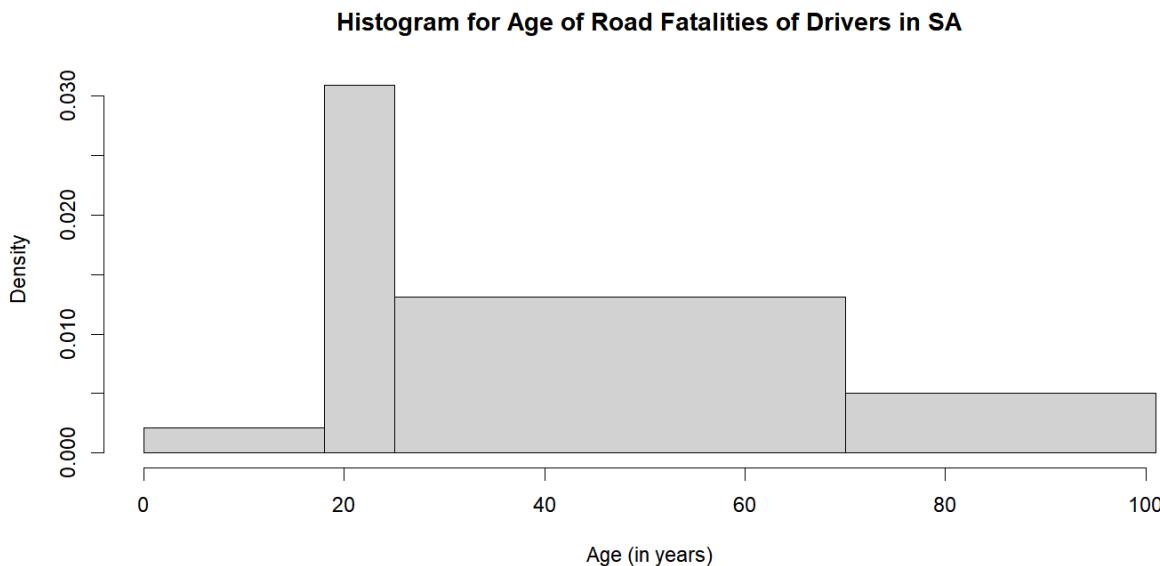
- We should filter the road deaths data for **South Australia** and for **drivers**.

```
1 data.sa = data[data$State == "SA" & data$Road.User == "Driver", ]  
2 dim(data.sa)
```

```
[1] 2226   24
```

- We have 2226 observed deaths. Plot the histogram of the ages of those fatalities.

```
1 hist(data.sa$Age, br = breaks, freq = F, right = F, xlab = "Age (in years)", ylab = "Density",  
2       main = "Histogram for Age of Road Fatalities of Drivers in SA")
```



- Pool the data for different age groups.

```
1 head(data.sa$Age)
```

```
[1] 22 86 54 36 24 53
```

```
1 data.sa$Age = cut(data.sa$Age, breaks, right = F)
```

```
2 head(data.sa$Age)
```

```
[1] [18,25) [70,101) [25,70) [25,70) [18,25) [25,70)
```

```
Levels: [0,18) [18,25) [25,70) [70,101)
```

```
1 data.sa.pooled = table(data.sa$Age)
```

```
2 data.sa.pooled
```

```
[0,18) [18,25) [25,70) [70,101)
```

```
84 481 1312 347
```

### 3) Derive death rates for different age groups.

Get death rate per 10,000 licences:

$$\text{death rate per 10000} = 10000 \times \frac{\text{number of deaths}}{\text{number of licences}}$$

```
1 death.rate = 10000 * data.sa.pooled/licence.sa.pooled$Total  
2 death.rate
```

```
[0,18) [18,25) [25,70) [70,101)  
25.39375 36.22834 13.43756 16.95835
```

#### Conclusion:

Death rate per licence for age group [18,25) is the highest, approximately three times higher than the death rate for age group [25,70)

# Summary

## Identifying variables

## Graphical summaries

- Barplot
- Histogram
- Scatter plot
- Boxplot

## Logical operators

---