Exploring Data

Data & Graphical Summaries

01 August 2024

Module1 Exploring Data

Data & Graphical Summaries

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

Numerical Summaries

What are the main features of the data?

Outline

Data story

What causes Australian road fatalities?

We are going to investigate 🔗 data from the Australian Bureau of Statistics (ABS) (last updated Nov 2023)

🔗 ABC Animation

```r
# Read in data
data = read.csv("data/2023fatalities.csv", header = TRUE)
# Names of Variables
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 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

Initial data analysis (IDA)

Data is **information** about the set of **subjects** being studied (like road fatalities). Most commonly, data refers to the **sample**, not the population.

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 pose other 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.

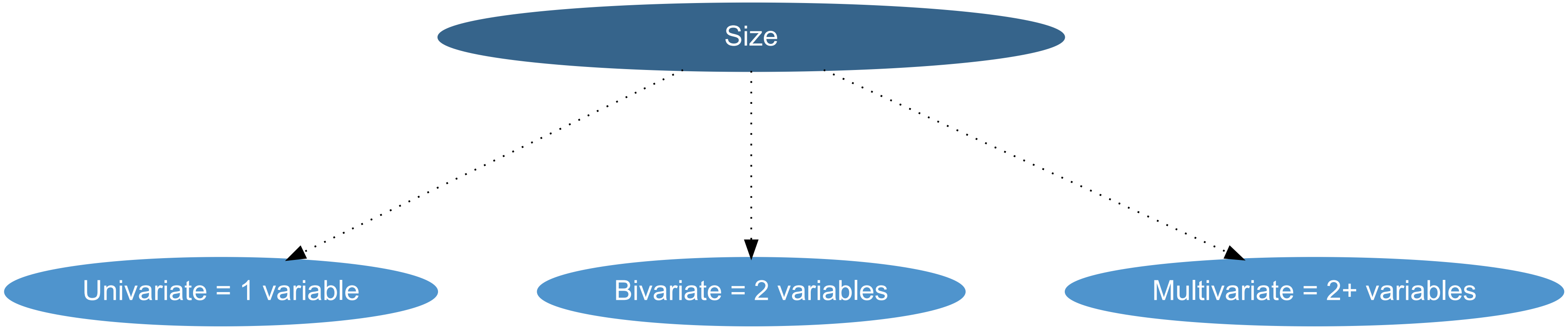
How many variables does the road fatality data have?

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

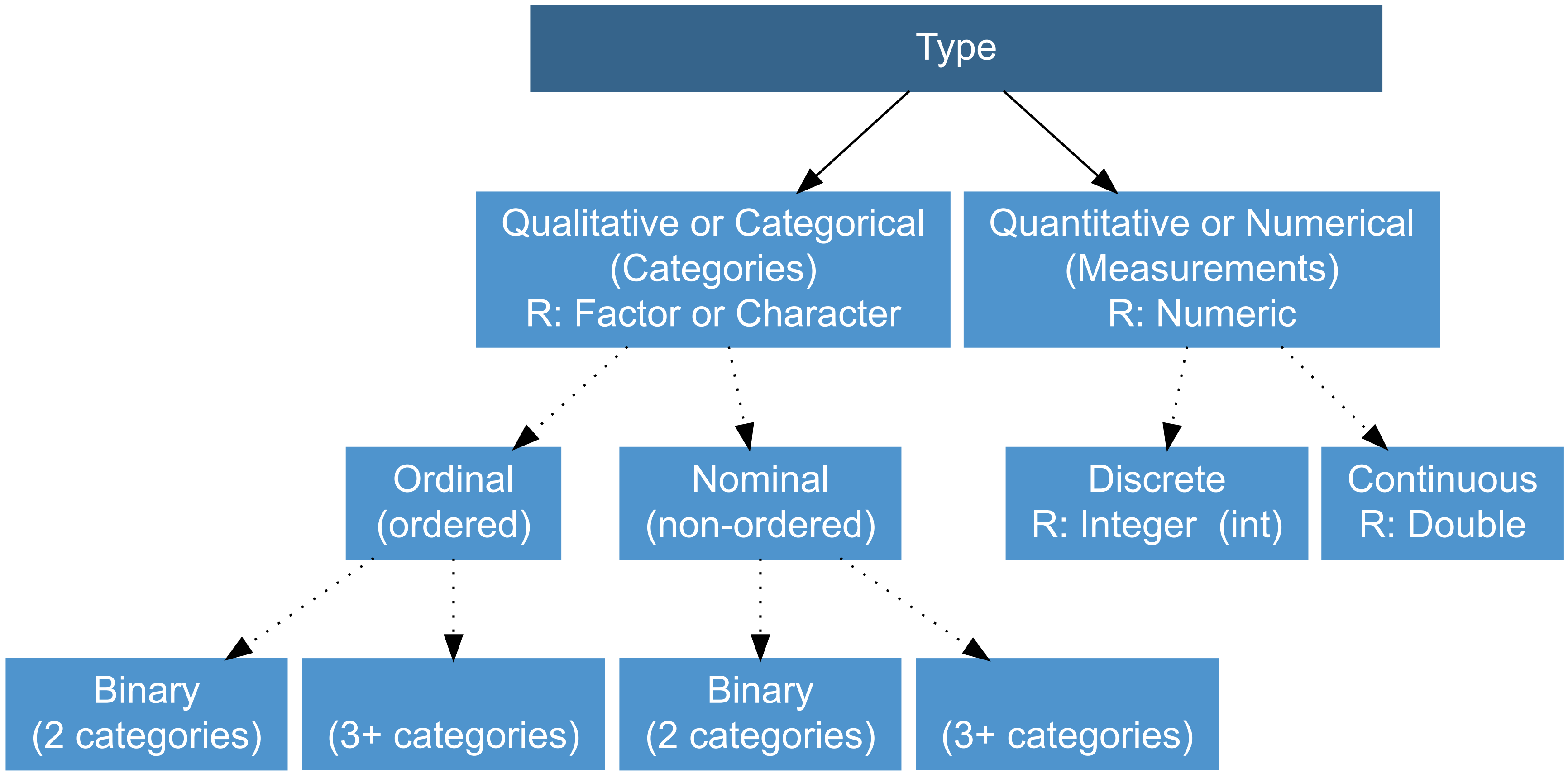
```
# Size of Data
dim(data)
```

```
## [1] 55360    24
```

Number of variables



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. However, it is optimal to record quantitative data if possible.

Suggest a similar variable.

Change variable types in R

```r
# Structure of Data
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" ...
##  $ Easter.Period                 : chr  "No" "No" ...
##  $ Age.Group                     : chr  "17_to_25" "17_to_25" ...
##  $ Day.of.week                   : chr  "Weekend" "Weekend" ...
##  $ Time.of.day                   : chr  "Night" "Night" ...
##  $ X                             : logi  NA NA NA ...
```

```r
# Change type of Variables
data$Crash.ID = as.factor(data$Crash.ID)
data$Month = as.factor(data$Month)
```

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

```
## 'data.frame':    55360 obs. of  24 variables:
##  $ Crash.ID                    : Factor w/ 49903 levels "19891001","19891002",..: 49880 49646 49506 49
##  $ State                       : chr  "NT" "SA" ...
##  $ Month                       : Factor w/ 12 levels "1","2","3","4",..: 10 10 10 10 10 ...
##  $ Year                        : int  2023 2023 2023 2023 2023 ...
##  $ Dayweek                     : chr  "Friday" "Saturday" ...
##   [list output truncated]
```

```r
# Change type of Variables
data$Speed.Limit = as.numeric(data$Speed.Limit)
```

```
## Warning: NAs introduced by coercion
```

```r
# New structure of Data display the first 5 variables using list.len
str(data[c(11, 12, 13, 14, 15)], vec.len = 2)
```

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

Graphical summaries

Graphical summaries

Once we identify 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

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

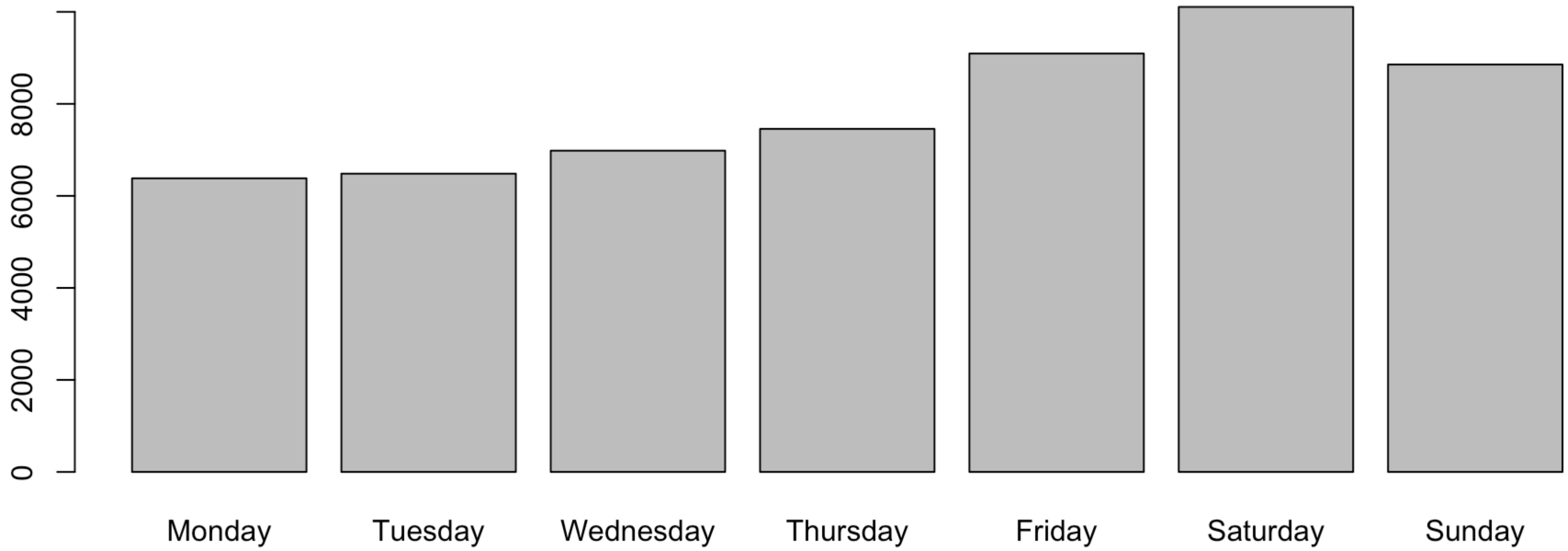
```
## Dayweek
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##      9094      6382     10107      8855      7456      6483      6983
```

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

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

Step 2: produce a barplot

```
# Produce a barplot
barplot(table(Dayweek))
```



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.

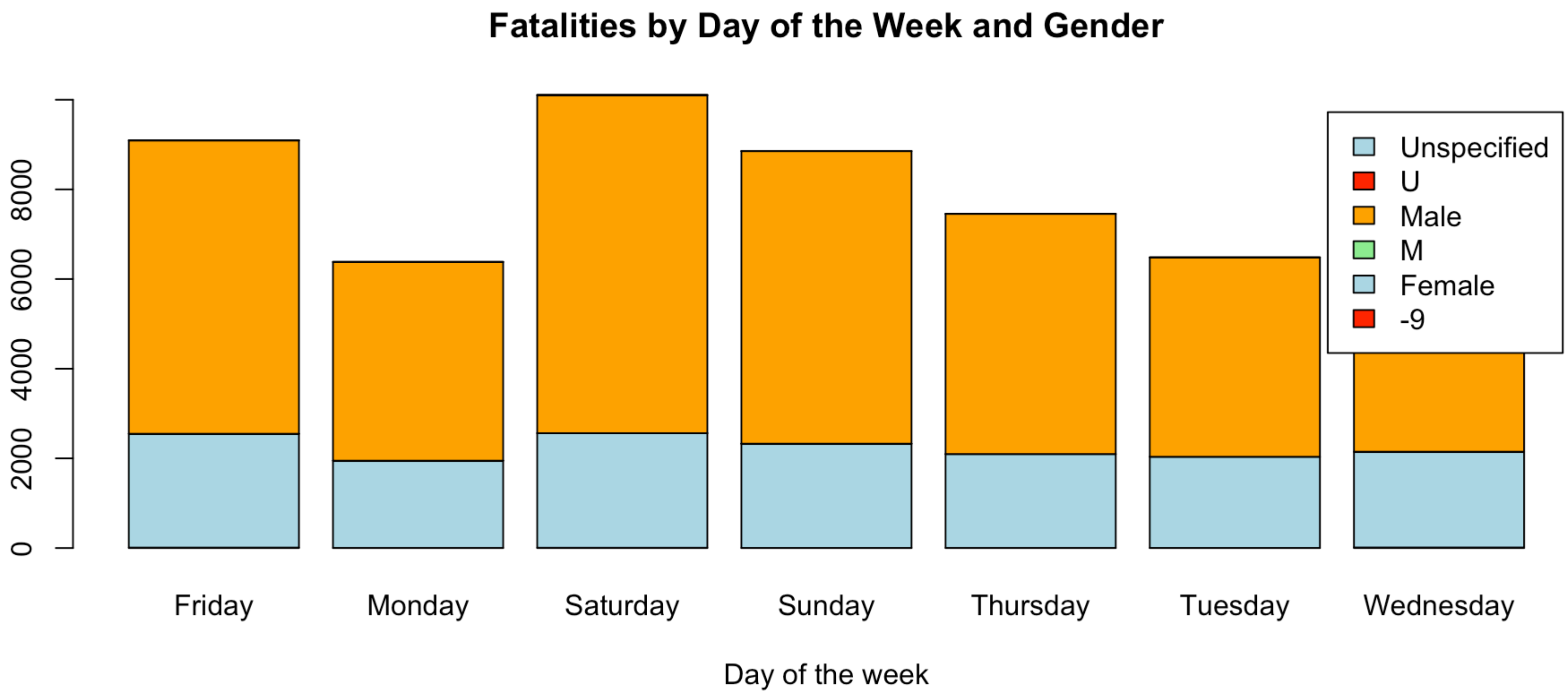
```
# Select DayWeek and Gender variables
Dayweek = data$Dayweek
Gender = data$Gender

# Produce a double frequency table (contingency table)
data1 = table(Gender, Dayweek)
data1
```

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

Note: Here Gender refers to biological sex as it was historically recorded in this dataset. Read more.

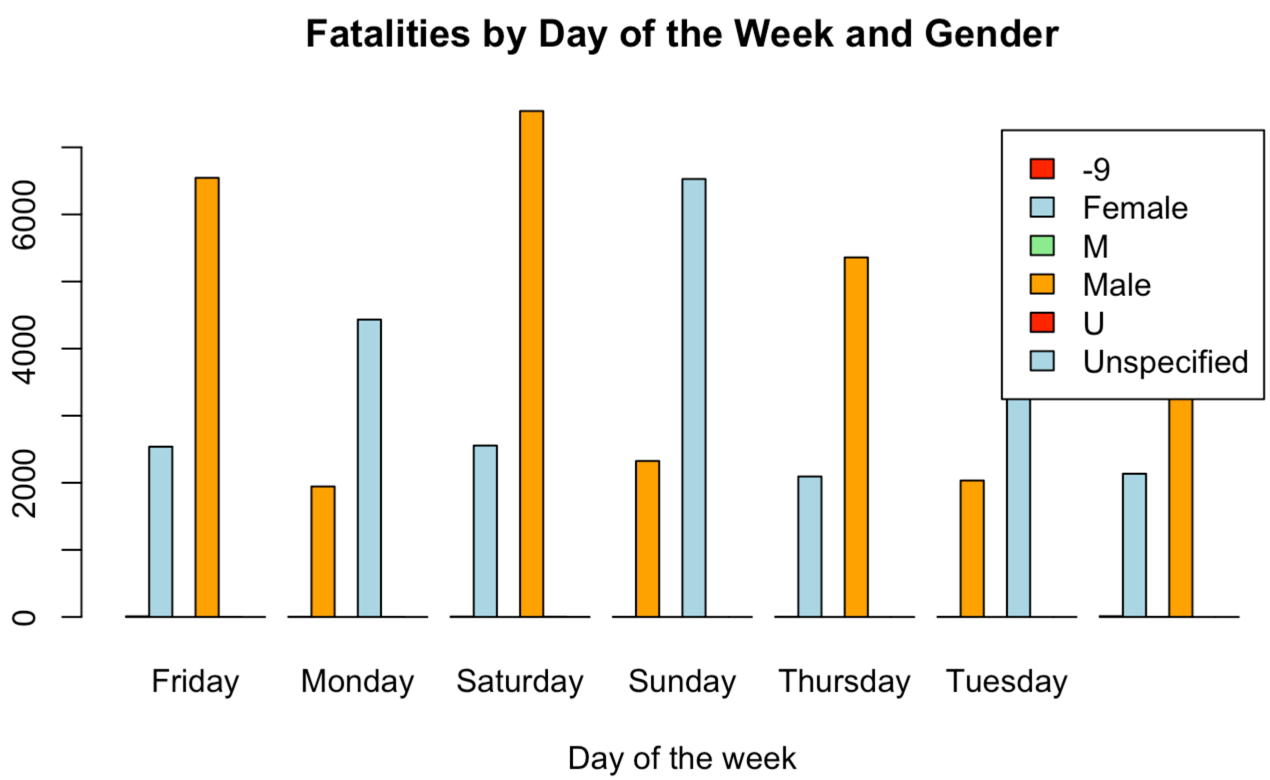
Stacked barplot

```
barplot(data1, main = "Fatalities by Day of the Week and Gender", xlab = "Day of the week",
    col = c("red", "lightblue", "lightgreen", "orange"), legend = rownames(data1))
```



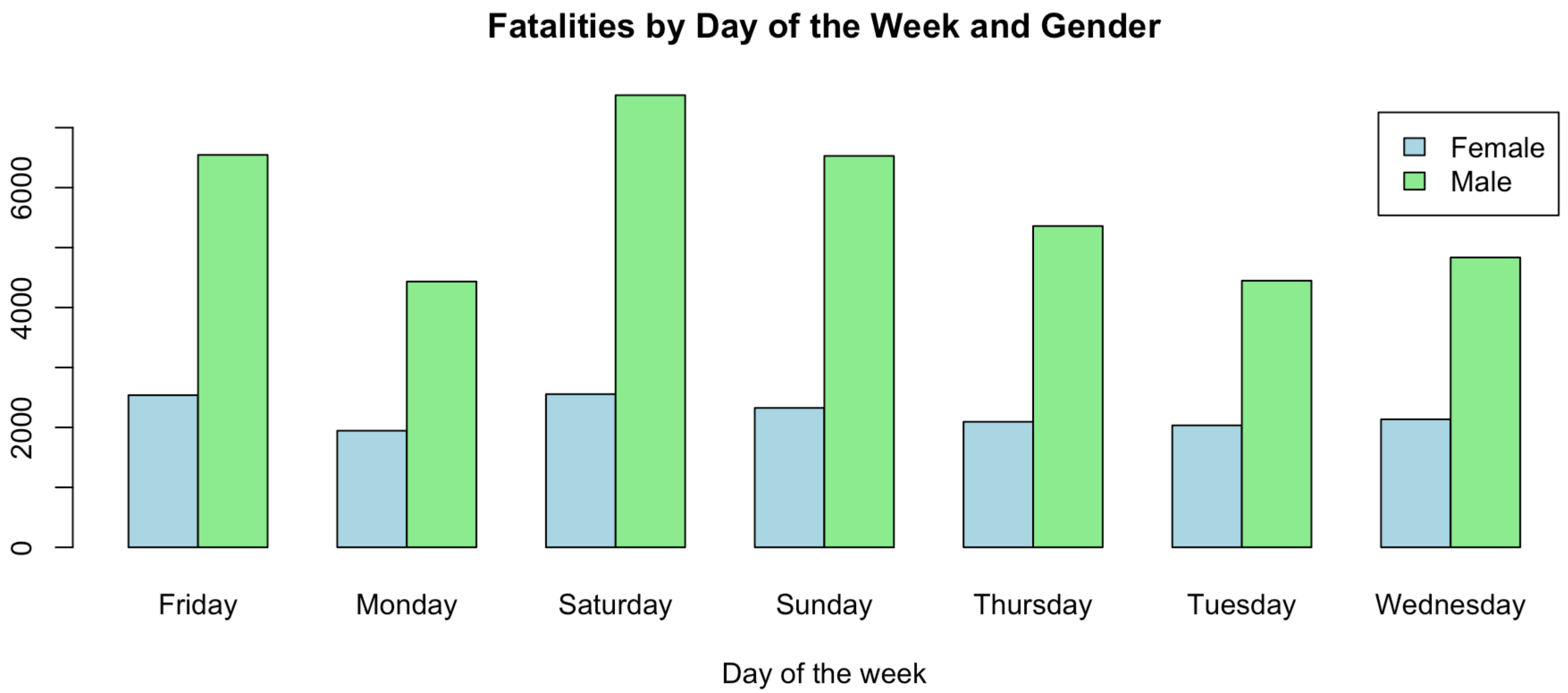
Side-by-side barplot

```
barplot(data1, main = "Fatalities by Day of the Week and Gender", xlab = "Day of the week",
    col = c("red", "lightblue", "lightgreen", "orange"), legend = rownames(data1),
    beside = TRUE)
```



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

```
barplot(data1[-c(1, 3, 5, 6), ], main = "Fatalities by Day of the Week and Gender",
    xlab = "Day of the week", col = c("lightblue", "lightgreen"), legend = rownames(data1[-c(1,
        3, 5, 6), ]), beside = TRUE)
```



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

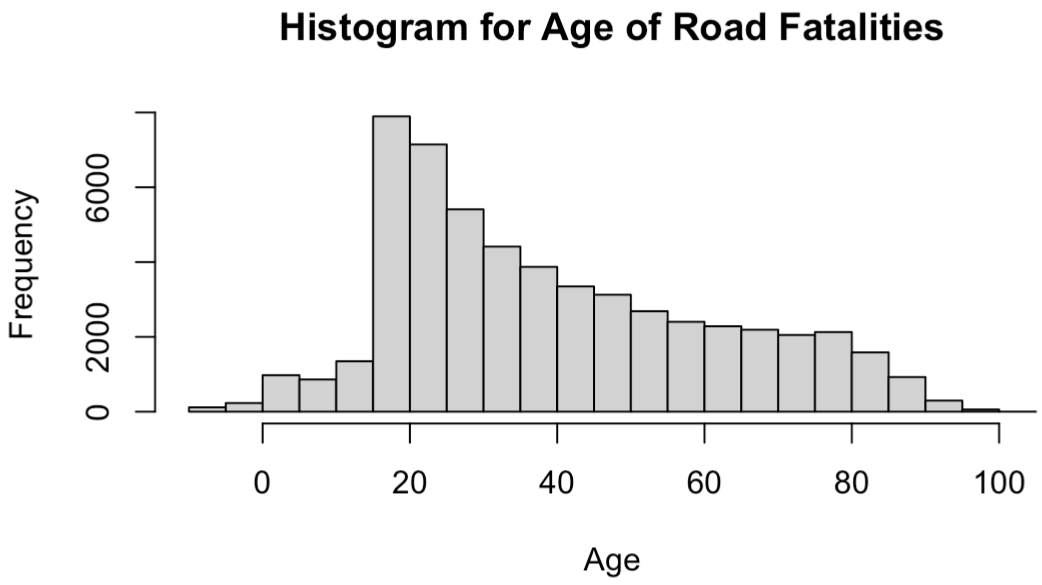
Histogram (quantitative data)

Histogram

The frequency table can also be used to summarise a set of **quantitative** data, by collecting the datq 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.

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



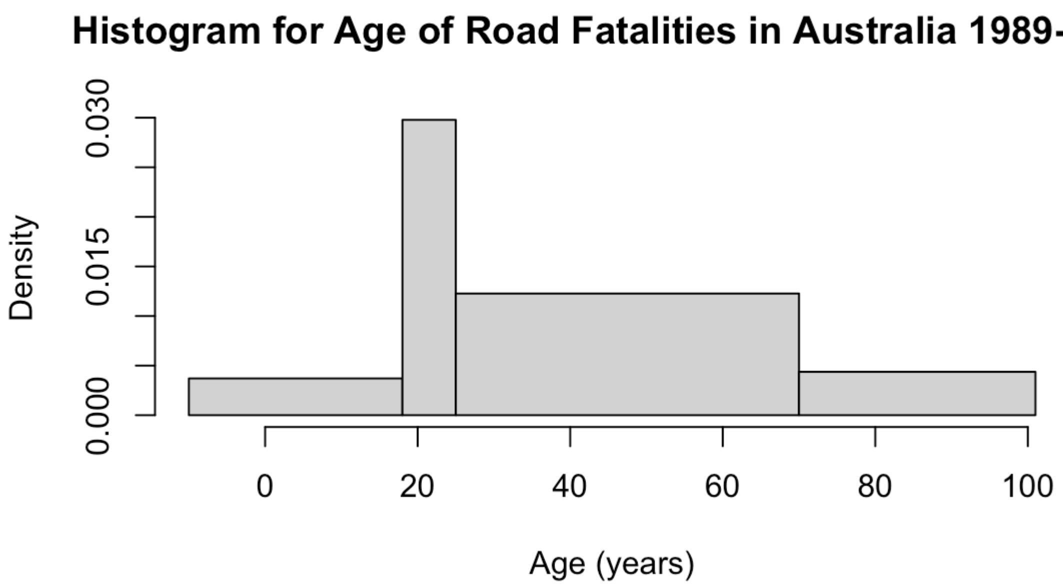
We can also provide used-defined class intervals and the **density scale**.

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

```
# Select the variable Age
Age = data$Age

# Define end points for class intervals
breaks = c(-10, 18, 25, 70, 101)

# Build frequency table
table(cut(Age, breaks, right = F))
```

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

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

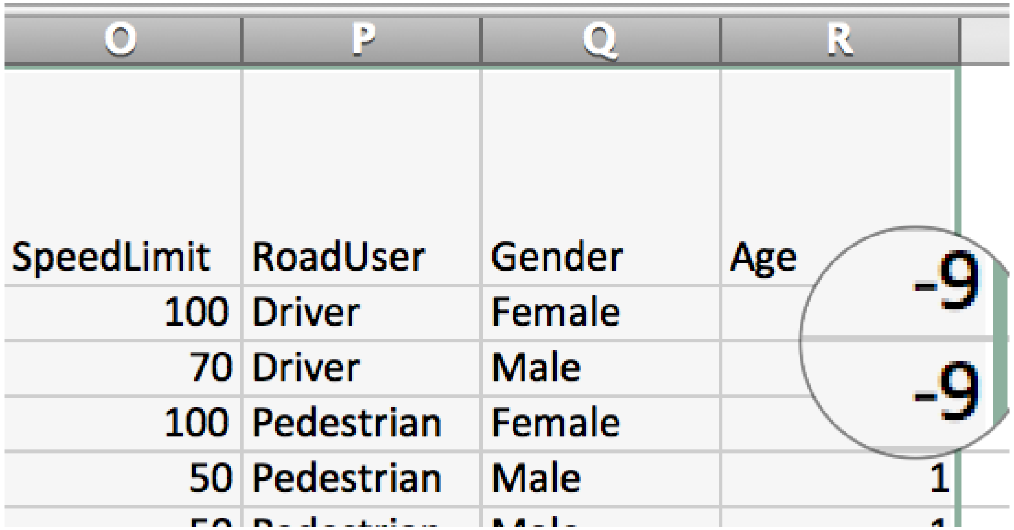


- The horizontal scale is divided into **class intervals** with potentially unequal sizes.
- The **area of each block** represents the **proportion** of subjects in that particular class interval.

Data cleaning

Why does the 1st block start below 0?

- Data Dictionary: missing values are coded as '-9'.



O	P	Q	R
SpeedLimit	RoadUser	Gender	Age
100	Driver	Female	-9
70	Driver	Male	-9
100	Pedestrian	Female	
50	Pedestrian	Male	1

- It is better to replace the "-9" by "NA".

```r
# Replacing the '-9' entries
data$Age[data$Age == -9] = NA
```

```r
hist(data$Age, br = breaks, freq = F, right = F, xlab = "Age", ylab = "Density",
    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

1. We will mostly use the **density scale** instead of frequency scale. It has advantages for later modelling.

Density scale

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

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

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

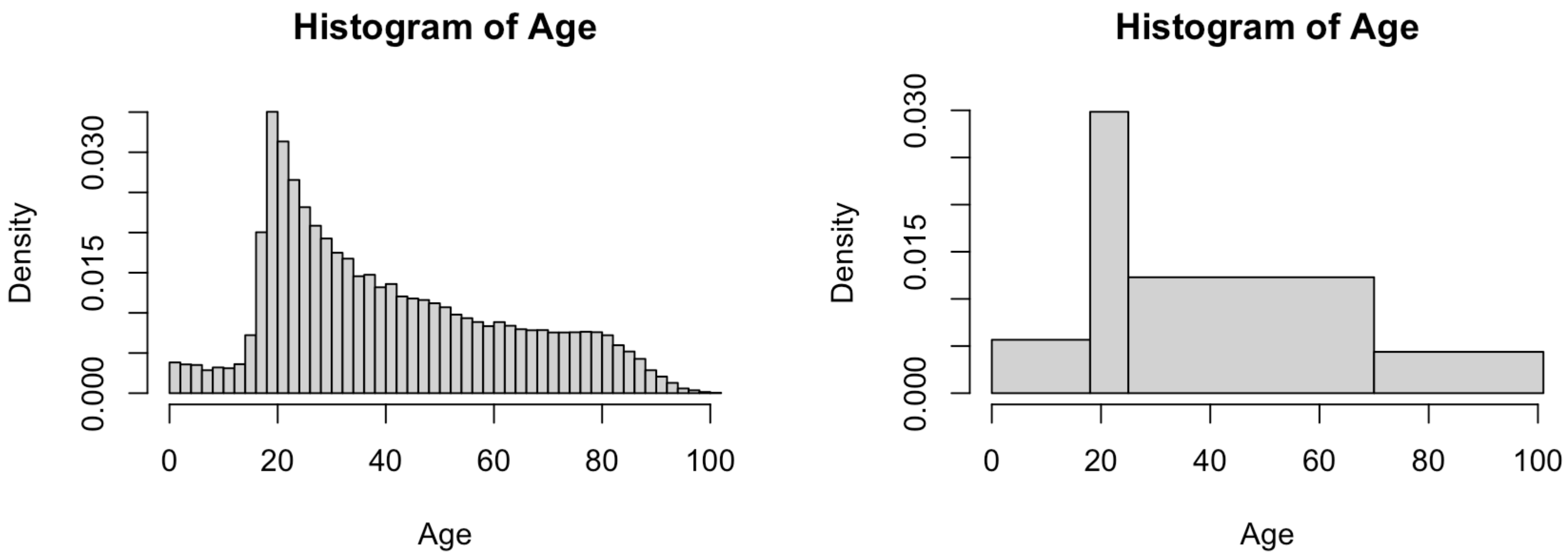
2. For continuous (quantitative) data, we need an **endpoint convention** for data points that fall on the border of two class intervals.

- 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].

3. Number of class intervals

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

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



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.

The speedy way in R

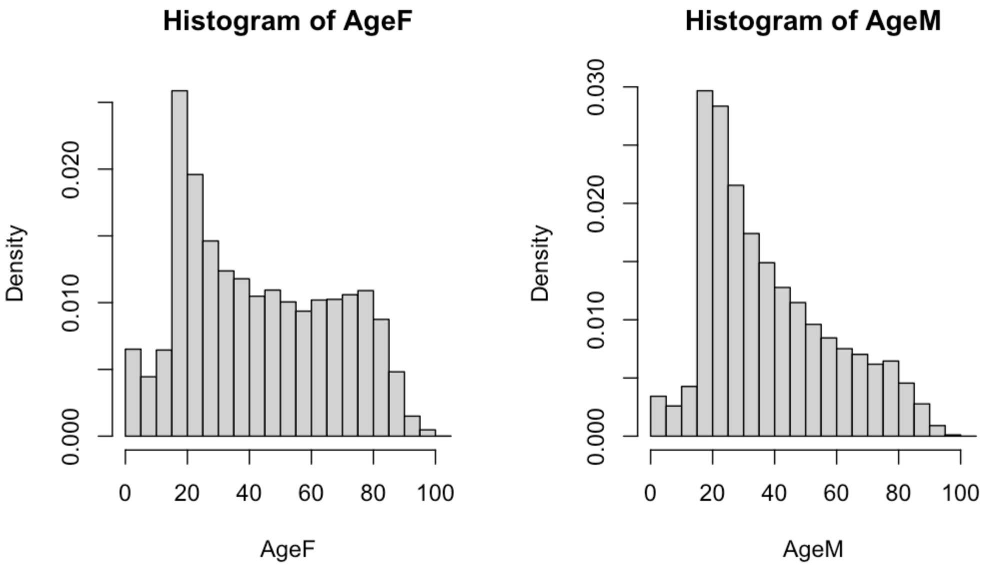
```r
# Read in data
data = read.csv("data/2023Fatalities.csv", header = T)

# Cleaning
data$Age[data$Age == -9] = NA

# Choose a variable
Age = data$Age

# Choose the class intervals
breaks = c(0, 18, 25, 70, 101)

# Produce a histogram
hist(Age, br = breaks, freq = F, right = F, xlab = "Age (in years)", ylab = "Density",
    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.

Controlling for a variable

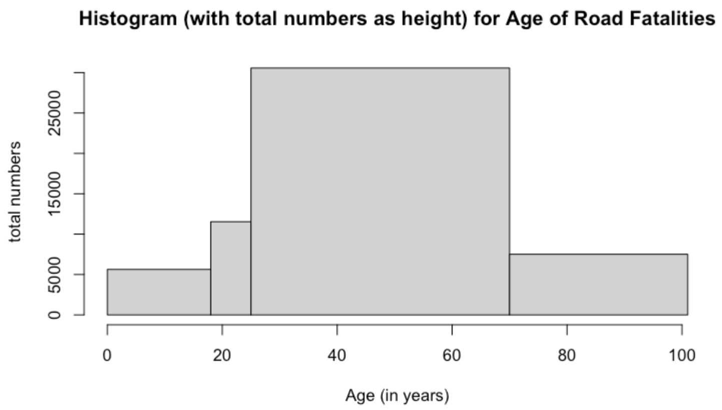
```
AgeF = data$Age[data$Gender == "Female"]  # This selects just the female ages.
AgeM = data$Age[data$Gender == "Male"]
par(mfrow = c(1, 2))  # This puts the graphic output in 1 row with 2 columns
hist(AgeF, freq = F)
hist(AgeM, freq = F)
```



💬 Do you notice any differences between men and women?

Mistake 1: the block heights are equal to the percentages

- Here we wrongly use the **percentage** (or the **total number** of subjects in a class interval) as the heights.
- Unless the class intervals are the same size, in both cases this will makes larger class intervals look like a larger overall %.

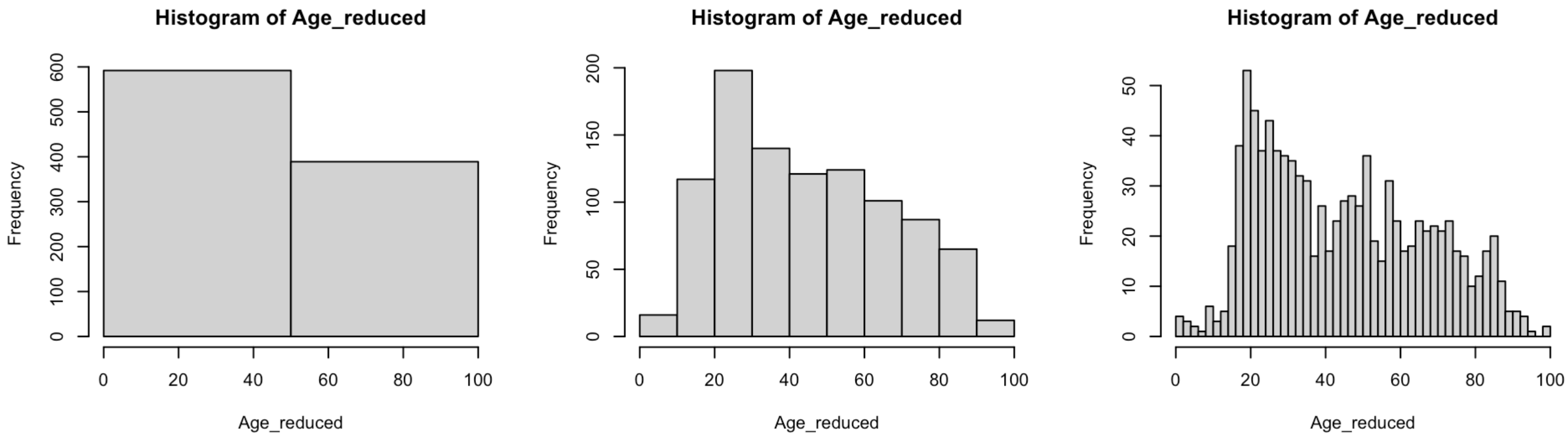


Solution: Use density as the height, especially if class intervals are **not** the same size. Don't use percentage or total numbers.

Mistake 2: Use 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.

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

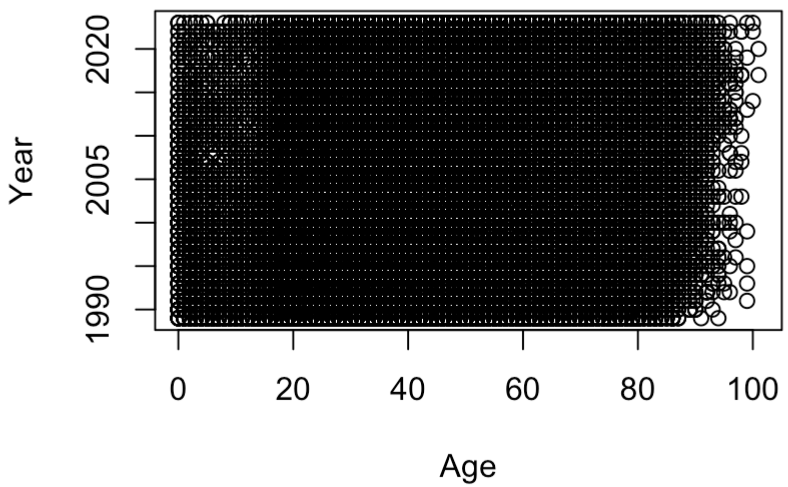


Other graphical summaries

Scatter plot

The scatter plot examines the relationship between two quantitative variables.

```
Year = data$Year
plot(Age, Year)
```

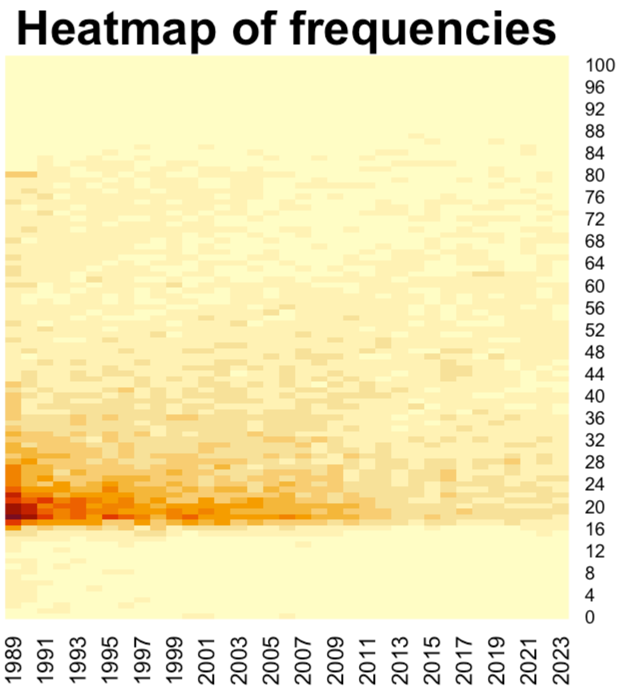


There have been fatalities for nearly every combination… so it is hard to say whether there is a relationship between Age and Year in fatalities.

Heatmap

A heatmap might be a good choice here. A heatmap is especially useful when a contingency table is not practical due to too many different values.

```
heatmap(table(Age, Year), Rowv = NA, Colv = NA, scale = "none", main = "Heatmap of frequencies")
```

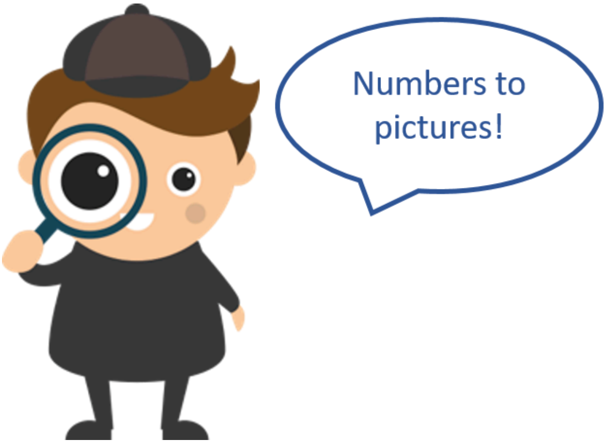


Boxplot

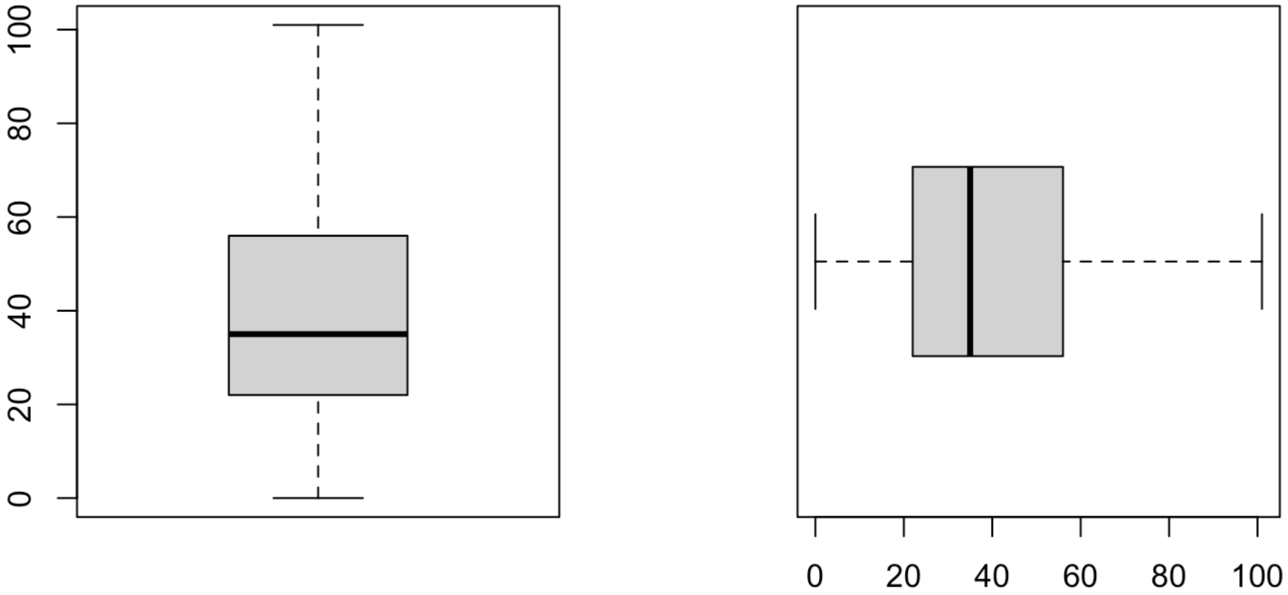
- The boxplot plots the five number summary of a data set. That is, the median ('middle' data point), the middle 50% of the data in a box, the expected maximum and minimum in the whiskers, and determines any outliers.
- We will consider how to draw the box plot when we learn about the interquartile range (IQR) in a later lecture.

```
Age = data$Age
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
```



```r
par(mfrow = c(1, 2))
boxplot(Age)
boxplot(Age, horizontal = T)
```



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.

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

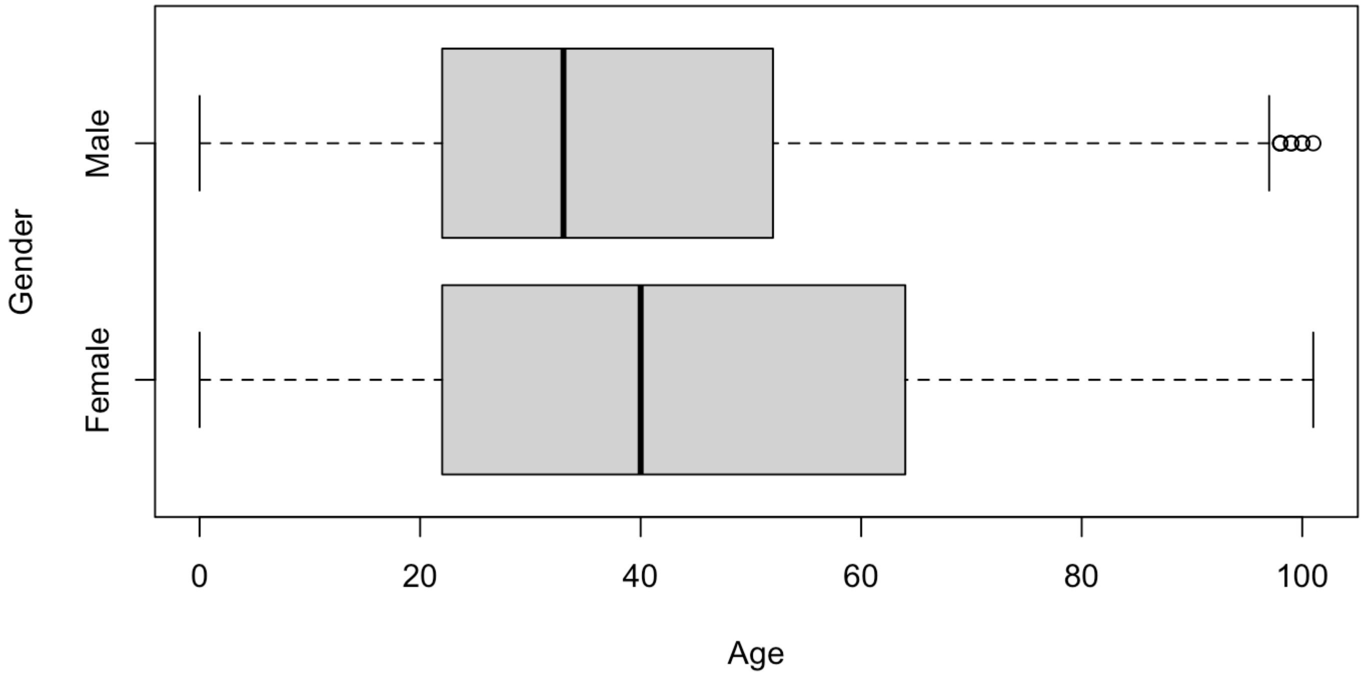
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   22.00   40.00   43.37   64.00  101.00      32
```

```
# Select each of the data entries in Age if the corresponding data entry in
# Gender is Female
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.

```r
data$Gender = factor(data$Gender, levels = c("Female", "Male"))
Gender = data$Gender
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:

`>`	greater than	`>=`	greater than or equal to
`<`	less than	`<=`	less than or equal to
`==`	equal to	`!=`	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.

```r
x = 5  # This assigns the value 5 to x
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`

`&`	True	False
True	True	False
False	False	False

Examples:

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

```
## [1] TRUE
```

```
x = 10
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

`|`	True	False
True	True	True
False	True	False

Examples:

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

```
## [1] TRUE
```

```
x = 10
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.

```
x = c(-1, 0, 1)
# Check each element of x against the condition (elementwise)
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?

```
x = 1:10
# Check each element of x against the condition (elementwise)
x <= 5
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```
sum(x <= 5)  # sum over those TRUEs (data points <= 5)
```

```
## [1] 5
```

Example on data selection

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

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

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

```
## [1] 2 4
```

```
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

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

```
## [1] 0 4
```

```
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?

```
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"
```

Strategy:

- Only count those deaths where person is driver.
- Find driving licences data with age information.
- Combine information and derive a death rate per driving licence for different age groups.

1) Only count those deaths where person is driver

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 person is driver

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

2) Find driving licences data with ages

South Australia provides this information.

data.gov.au

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

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

```
## Warning: NAs introduced by coercion
```

Pooled data

Put ages into categories using `cut`.

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

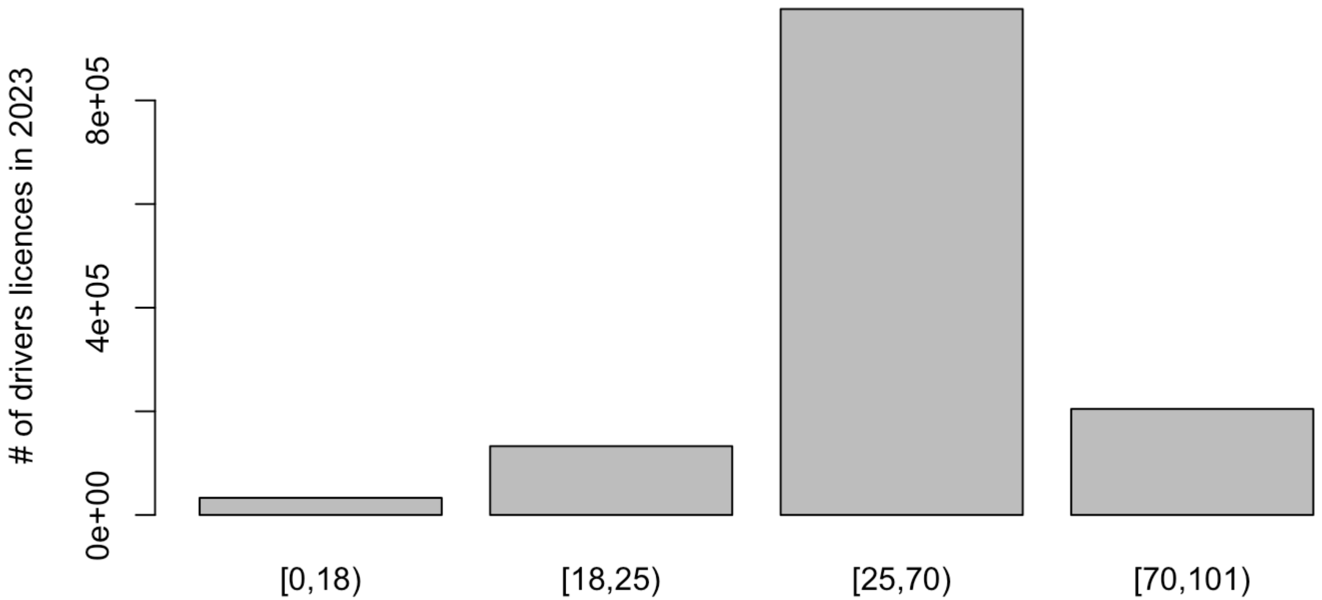
Pool the data for each age category using `aggregate`.

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

```
##        Age  Total
## 1   [0,18)  33079
## 2  [18,25) 132769
## 3  [25,70) 976352
## 4 [70,101) 204619
```

Plot the data with a barplot

```
Total = licence.sa.pooled$Total
barplot(Total, names.arg = licence.sa.pooled$Age, ylab = "# of drivers licences in 2023")
```



Re-visit 1)

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

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

```
## [1] 2226   24
```

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

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

- Pool the data for different age groups.

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

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

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

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

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

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

```
##
##   [0,18)  [18,25)  [25,70) [70,101)
## 25.39375 36.22834 13.43778 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

Some useful R Functions

`names`, `dim`, `str`, `table`, `barplot`, `hist`, `plot`, `boxplot`, `as.factor`, `as.numeric`, `cut`, `sum`

Logical operators

`<`, `<=`, `>`, `>=`, `==`, `!=`, `&`, `|`, `!`