

# Summarising data

Melbourne Statistical Consulting Platform  
University of Melbourne  
April 2024

# Summarising data

Summaries of:

- one numerical variable
- one numerical and one categorical variable
- one categorical variable
- two categorical variables

# Summarising data

Summaries of:

- **one numerical variable**
- one numerical and one categorical variable
- one categorical variable
- two categorical variables

# Descriptive statistics for numerical data

- measures of location (mean, median, quartiles)
- measures of spread (variance, standard deviation, IQR)
- many other less common summary statistics for numerical data

# Descriptive statistics for numerical data

Some adolescent dental health data:

```
dental_decay <-  
  read_csv("../5 data/dentaldecay.csv")
```

Rows: 220 Columns: 6

— Column specification —————

Delimiter: ","

chr (4): Fluoridation, Caries, Brush...

dbl (2): PatientID, Age

**i** Use ``spec()`` to retrieve the full column specification

**i** Specify the column types or set ``show_col_types``

# Descriptive statistics for numerical data

Base R provides a basic `summary` function

```
summary(dental_decay$Age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    \
  8.00  11.00   13.00   12.63  14.00    \
   Max.                                     \
 17.00
```

As well as individual functions to obtain specific summaries

```
mean(dental_decay$Age)
```

```
[1] 12.62727
```

```
sd(dental_decay$Age)
```

```
[1] 2.05788
```

# Descriptive statistics for numerical data

The `summarise()` function collapses a data frame down to a single row.

```
summarise(dental_decay,  
          Mean_Age = mean(Age),  
          SD_Age = sd(Age))
```

```
# A tibble: 1 × 2  
  Mean_Age SD_Age  
    <dbl>   <dbl>  
1    12.6    2.06
```

# Useful R functions for summarising data

- `sum()`
- `mean()`, `sd()`
- `median()`, `quantile()`, `min()`, `max()`
- `rank()`
- `n()`
- `sum(!is.na())` (counts number of non-missing observations)

Most built-in R functions deal conservatively with missing data: if there is even a single missing value, the function will return `NA`.

Most of the functions on the left accept the option `na.rm = TRUE` to exclude missing observations.



# Descriptive statistics for numerical data

We can also make a nice-looking table using the `gt()` function (in the `gt` package)

```
library(gt)
gt(summarise(dental_decay,
             Mean_Age = mean(Age),
             SD_Age = sd(Age)))
```

Mean_Age	SD_Age
12.62727	2.05788

# An introduction to pipes

## Brackets

Using brackets inside brackets like `gt(summarise(...))` is a common programming approach. But...

- it's not very elegant
- you don't actually do the outside thing first
- it is harder to see immediately what is going on
- this doesn't scale easily to lots of steps



Source: Wikipedia

# An introduction to pipes

## Brackets

Using brackets inside brackets like `gt(summarise(...))` is a common programming approach. But...

## Actual steps

1. Start with the source of the data
2. Calculate summary statistics
3. Present them in a table

## Pipes

- `%>%` operator looks kinda like a pipe if you squint
- if you find that hard to type, Ctrl+Shift+M will insert `%>%`
- `%>%` needs to go at the end of each line (not the start), but not the final line of the sequence

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
gt()
```

# An introduction to pipes

## Brackets

```
gt(summarise(dental_decay,  
             Mean_Age = mean(Age),  
             SD_Age = sd(Age),  
             n = n()))
```

Mean_Age	SD_Age	n
12.62727	2.05788	220

## Pipes

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt()
```

Mean_Age	SD_Age	n
12.62727	2.05788	220

# Descriptive statistics for numerical data

We can also make a nice-looking table using the `gt()` function, using the `fmt_number()` function to control rounding.

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age)) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```

Mean_Age	SD_Age
12.6	2.1

# Summarising data

Summaries of:

- one numerical variable
- **one numerical and one categorical variable**
- one categorical variable
- two categorical variables

Introducing `group_by()`

# Grouping output

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```

Mean_Age	SD_Age	n
12.6	2.1	220

# Grouping output

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```

Mean_Age	SD_Age	n
12.6	2.1	220

```
dental_decay %>%  
  group_by(Caries) %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```



# Grouping output

```
dental_decay %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```

Mean_Age	SD_Age	n
12.6	2.1	220

```
dental_decay %>%  
  group_by(Caries) %>%  
  summarise(Mean_Age = mean(Age),  
            SD_Age = sd(Age),  
            n = n()) %>%  
  gt() %>%  
  fmt_number(c(Mean_Age, SD_Age),  
            decimals = 1)
```

Caries	Mean_Age	SD_Age	n
No	12.4	2.0	163
Yes	13.3	2.0	57

# Summarising data

Summaries of:

- one numerical variable
- one numerical and one categorical variable
- **one categorical variable**
- two categorical variables

# Frequency tables for categorical data

```
dental_decay %>%  
  group_by(Caries) %>%  
  summarise(n = n()) %>%  
  gt()
```

Caries	n
No	163
Yes	57

We recommend using the `tabyl` function in the `janitor` package.

```
library(janitor)  
dental_decay %>%  
  tabyl(Caries) %>%  
  gt()
```

Caries	n	percent
No	163	0.7409091
Yes	57	0.2590909

# Frequency tables for categorical data

```
dental_decay %>%  
  tabyl(Brush) %>%  
  gt()
```

Brush	n	percent
Once or less a day	116	0.5272727
Twice or more a day	104	0.4727273

# Frequency tables for categorical data

We can `adorn` our tables in a range of ways.

```
dental_decay %>%  
  tabyl(Brush) %>%  
  adorn_totals("row") %>%  
  gt()
```

Brush	n	percent
Once or less a day	116	0.5272727
Twice or more a day	104	0.4727273
Total	220	1.0000000

# Frequency tables for categorical data

We can `adorn` our tables in a range of ways.

```
dental_decay %>%  
  tabyl(Brush) %>%  
  adorn_totals("row") %>%  
  adorn_pct_formatting() %>%  
  gt()
```

Brush	n	percent
Once or less a day	116	52.7%
Twice or more a day	104	47.3%
Total	220	100.0%

# Adornment options

- `adorn_ns()`: add underlying Ns to a tabyl displaying percentages
- `adorn_pct_formatting()`: format a tabyl with decimals as percentages
- `adorn_percentages()`: Convert a tabyl of counts to percentages
- `adorn_rounding()`: Round the numeric columns in a tabyl
- `adorn_title()`: Add column name to the top of a two-way tabyl
- `adorn_totals()`: Append a totals row and/or column to a tabyl

These functions can also be applied to data frames more broadly.

# Summarising data

Summaries of:

- one numerical variable
- one numerical and one categorical variable
- one categorical variable
- **two categorical variables**



# Crosstabulations for two categorical variables

```
dental_decay %>%  
  tabyl(Brush, Caries) %>%  
  gt()
```

Brush	No	Yes
Once or less a day	83	33
Twice or more a day	80	24

# Crosstabulations for two categorical variables

```
dental_decay %>%  
  tabyl(Brush, Caries) %>%  
  adorn_percentages("row") %>%  
  gt()
```

Brush	No	Yes
Once or less a day	0.7155172	0.2844828
Twice or more a day	0.7692308	0.2307692

# Crosstabulations for two categorical variables

```
dental_decay %>%  
  tabyl(Brush, Caries) %>%  
  adorn_percentages("row") %>%  
  adorn_pct_formatting(digits = 0) %>%  
  adorn_ns(position = "front") %>%  
  gt()
```

Brush	No	Yes
Once or less a day	83 (72%)	33 (28%)
Twice or more a day	80 (77%)	24 (23%)

# Extension: tbl\_summary

A simple and quick way to summarise your data

```
library(gtsummary)

dental_decay %>%
  tbl_summary()
```

Needs a bit of work!

Characteristic	<sup>1</sup> "> <b>N = 220</b> <sup>1</sup>
PatientID	111 (56, 165)
Fluoridation	120 (55%)
Caries	57 (26%)
Brush	
Once or less a day	116 (53%)
Twice or more a day	104 (47%)
Age	13.00 (11.00, 14.00)
Gender	
Female	98 (45%)
Male	122 (55%)
<sup>1</sup> Median (IQR); n (%)	

# Extension: tbl\_summary

Also by a grouping variable.

```
library(gtsummary)

dental_decay %>%
  tbl_summary(by="Gender")
```

Characteristic	<sup>1</sup> > <b>Female</b> , N = 98 <sup>1</sup>	<sup>1</sup> > <b>Male</b> , N = 122 <sup>1</sup>
PatientID	112 (58, 153)	109 (55, 180)
Fluoridation	45 (46%)	75 (61%)
Caries	27 (28%)	30 (25%)
Brush		
Once or less a day	53 (54%)	63 (52%)
Twice or more a day	45 (46%)	59 (48%)
Age	13.00 (11.00, 14.00)	13.00 (11.00, 14.00)
<sup>1</sup> Median (IQR); n (%)		

Particularly useful starting point for the classic "Table 1" for patient characteristics by treatment for a clinical trial.

## Exercise 2.1.