MONASH University

ETC5512: Wild Caught Data

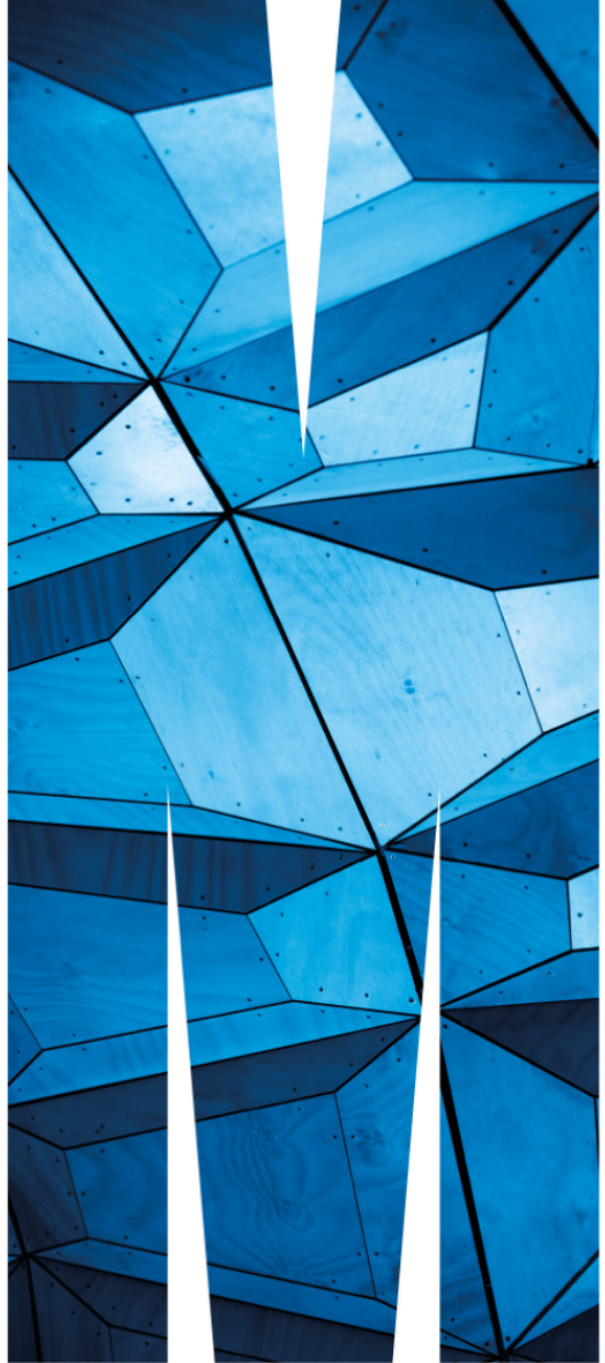
Australian census

Lecturer: *Kate Saunders*

Department of Econometrics and Business Statistics

ETC5512.Clayton-x@monash.edu

Week 4



Recall from lecture 2:

> *Collecting data on the entire population is normally too expensive or infeasible!*
>
> *Therefore, we often can only collect data about a subset of the population.*

If we can collect data about the entire population, that is called a **census**

	Sample survey	Census
Advantages	• Reduces cost • Timely collection of data	• Data available, even for small geographical areas or subpopulations • Statistics are not subject to sampling error • Better accuracy and details
Disadvantages	• Lack of data on sub-population (particularly minorities) or small geographical areas • Requires careful construction of sampling design • Estimates are subject to sampling error • The estimates may not be accurate or reliable • Estimating and communicating precision of estimates is difficult	• Expensive or infeasible • Time consuming to collect all data

Today is all about the Australian Census:

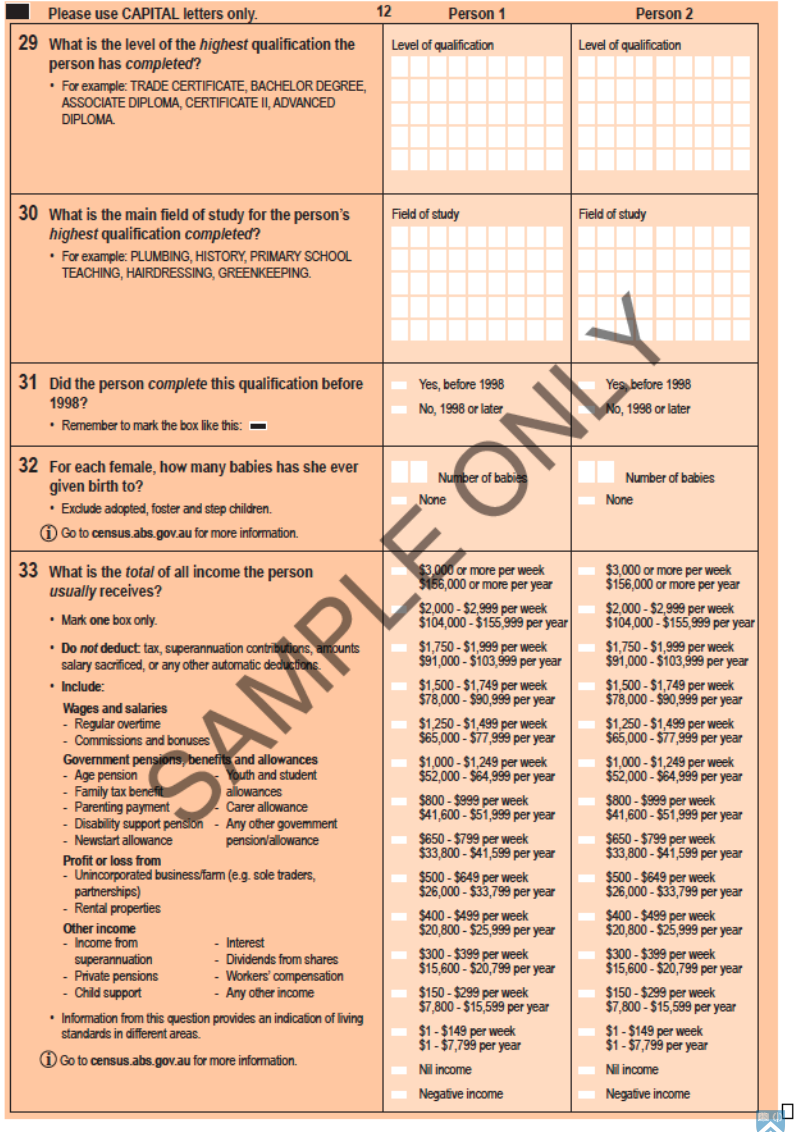
- Learn about what the census is and how it is collected
- Learn what data on population demographics is collected
- Learn how the census data is stored and how to access it

From a coding perspective:

- Learn about organising your data the **tidy data** way.
- Learn to **manipulate strings** and a bit about regular expressions.

What is the Australian census?

- The first Australian census was held in 1911.
- The next census is in 2026.
- Counts **every person and household** in Australia. *(well almost everyone, the 2021 census had a 96% participation rate but that is very high.)*
- Comprehensive snapshot of the country and **tells the story of how we are changing**.
- The Australia Bureau of Statistics (ABS) is legislated to collect and disseminate census data under the *ABS Act 1975* and *Census and Statistics Act 1905*.
- For more details refer to the (ABS) Website.



Please use CAPITAL letters only. 12

	Person 1	Person 2
29 What is the level of the *highest* qualification the person has *completed*? • For example: TRADE CERTIFICATE, BACHELOR DEGREE, ASSOCIATE DIPLOMA, CERTIFICATE II, ADVANCED DIPLOMA.	Level of qualification	Level of qualification
30 What is the main field of study for the person's *highest* qualification *completed*? • For example: PLUMBING, HISTORY, PRIMARY SCHOOL TEACHING, HAIRDRESSING, GREENKEEPING.	Field of study	Field of study
31 Did the person *complete* this qualification before 1998? • Remember to mark the box like this: ▬	Yes, before 1998 No, 1998 or later	Yes, before 1998 No, 1998 or later
32 For each female, how many babies has she ever given birth to? • Exclude adopted, foster and step children. ⓘ Go to **census.abs.gov.au** for more information.	Number of babies None	Number of babies None
33 What is the *total* of all income the person *usually* receives? • Mark **one** box only. • **Do *not* deduct:** tax, superannuation contributions, amounts salary sacrificed, or any other automatic deductions. • **Include:** **Wages and salaries** - Regular overtime - Commissions and bonuses **Government pensions, benefits and allowances** - Age pension - Family tax benefit - Parenting payment - Disability support pension - Newstart allowance - Youth and student allowances - Carer allowance - Any other government pension/allowance **Profit or loss from** - Unincorporated business/farm (e.g. sole traders, partnerships) - Rental properties **Other income** - Income from superannuation - Private pensions - Child support - Interest - Dividends from shares - Workers' compensation - Any other income • Information from this question provides an indication of living standards in different areas. ⓘ Go to **census.abs.gov.au** for more information.	$3,000 or more per week $156,000 or more per year $2,000 - $2,999 per week $104,000 - $155,999 per year $1,750 - $1,999 per week $91,000 - $103,999 per year $1,500 - $1,749 per week $78,000 - $90,999 per year $1,250 - $1,499 per week $65,000 - $77,999 per year $1,000 - $1,249 per week $52,000 - $64,999 per year $800 - $999 per week $41,600 - $51,999 per year $650 - $799 per week $33,800 - $41,599 per year $500 - $649 per week $26,000 - $33,799 per year $400 - $499 per week $20,800 - $25,999 per year $300 - $399 per week $15,600 - $20,799 per year $150 - $299 per week $7,800 - $15,599 per year $1 - $149 per week $1 - $7,799 per year Nil income Negative income	$3,000 or more per week $156,000 or more per year $2,000 - $2,999 per week $104,000 - $155,999 per year $1,750 - $1,999 per week $91,000 - $103,999 per year $1,500 - $1,749 per week $78,000 - $90,999 per year $1,250 - $1,499 per week $65,000 - $77,999 per year $1,000 - $1,249 per week $52,000 - $64,999 per year $800 - $999 per week $41,600 - $51,999 per year $650 - $799 per week $33,800 - $41,599 per year $500 - $649 per week $26,000 - $33,799 per year $400 - $499 per week $20,800 - $25,999 per year $300 - $399 per week $15,600 - $20,799 per year $150 - $299 per week $7,800 - $15,599 per year $1 - $149 per week $1 - $7,799 per year Nil income Negative income

SAMPLE ONLY

What is the Australian Bureau of Statistics (ABS)?

- ABS is the independent statistical agency of the Government of Australia.
- If you are from outside Australia, find the statistical government agency in your country 🔧, e.g.
  - in 🇯🇵 Japan, this is the Statistics Bureau of Japan,
  - in 🇨🇳 China, the National Bureau of Statistics of China,
  - in 🇮🇳 India, the Ministry of Statistics and Programme Implementation, and
  - in 🇳🇿 New Zealand, the Statistics New Zealand.
- ABS provides key statistics on a wide range of economic, population, environmental and social issues, to assist and encourage informed decision making, research and discussion within governments and the community.

Why do we do a census?

- The census is not cheap to do. The 2021 census **cost of $565 million**. That's roughly $22 per person.
- However the **census provides value for money and it is important**.
- An independent report found that for every $1 invested in the Census, $6 of value is generated to the Australian economy.
- The census **data tells us about the economic, social and cultural** make-up of the country.
- Need census **data to make decisions and plan for the future**
- It informs planning for schools, health care, transport and infrastructure. It is also used to help plan local services for individuals, families and communities.

How is the census conducted?

The ABS contacts households in a few different ways:

- **Letters and paper forms are delivered** in some areas
- In other areas, **visits were made to households**.

Then households complete the Census form, either submitting it online or sending it back in the mail.

ABS provides a range of supports and resources to help everyone to fill in the census.

Take a moment to think about **what challenges might arise if you try to survey everyone**.

Hint: Think about smaller communities, their sub-groups and their different needs.

How can we survey everyone?

It is no small task!

- Resources for people in the deaf/hard of hearing and blind/low vision communities
  e.g. audio guides and braille information packs

- To support Aboriginal and Torres Straight Islanders to fill in the census there are urban and regional pop-up hubs. *This includes extra face-to-face support*

- For migrants, refugees, and international visitors there are language supports available.

- Additional efforts are made to survey in locations to reach without a fixed address
  e.g. FIFO workers (Fly in Fly Out), Grey Nomads, People experiencing homelessness.

More details on the ABS website: here

What is in the census?

There are questions about:

- age
- country of birth
- religion
- ancestry
- language used at home
- work
- education

Breakout Session

Investigate what data is collected in the census.

Use the quick stats summary for Clayton here.

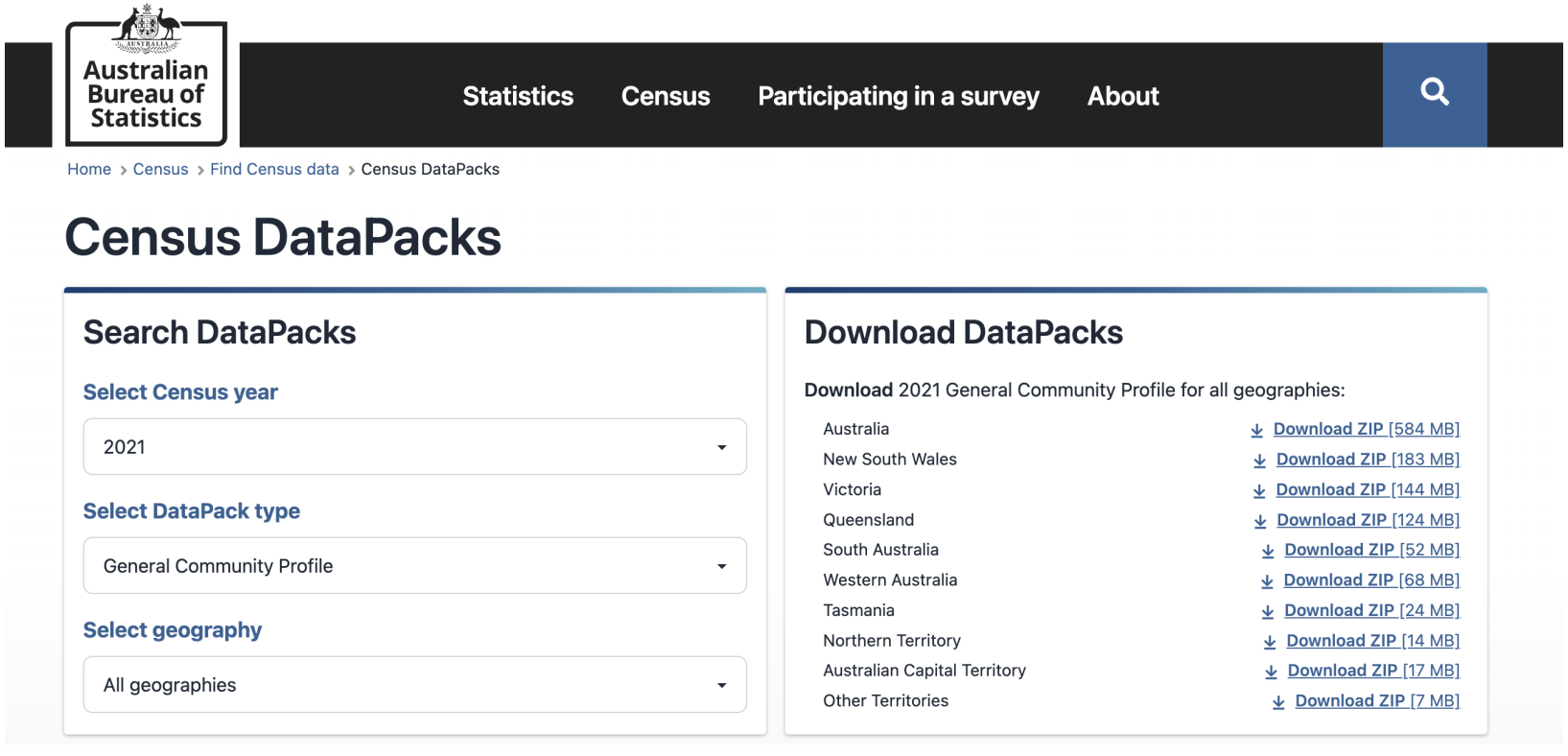
Are there any weird variables, or variables that surprise you? What do you learn about where you live?

Getting the ABS Census Data

https://www.abs.gov.au/census/find-census-data

There are two main types of data that you can download:

- **DataPacks** https://www.abs.gov.au/census/find-census-data/datapacks
- **GeoPackages** https://www.abs.gov.au/census/find-census-data/geopackages



Navigating ABS Census data

- DataPacks are only available for the 2011, 2016 and 2021 census.
- ABS aims for **census data to be comparable and compatible with previous censuses**.
- Questions and classifications are reviewed to reflect changes in the Australian society.
  e.g. In 2021, ABS did not to ask about home internet connection as people now have other options like mobile devices and that data was no longer considered relevant to society.
- There are small differences in the available data between years.
  Variables can be added, updated and removed.
- There are also sometimes data corrections at a later date.
- Here are links to:
  (i) what's new in 2021 - there were 56 new additions!
  (ii) consultation for changes in 2026 and
  (iii) an example of a 2026 proposed change

Reality of any data analysis

Navigating data and deducing what it is often requires you to do some "**detective work**" 🕵️

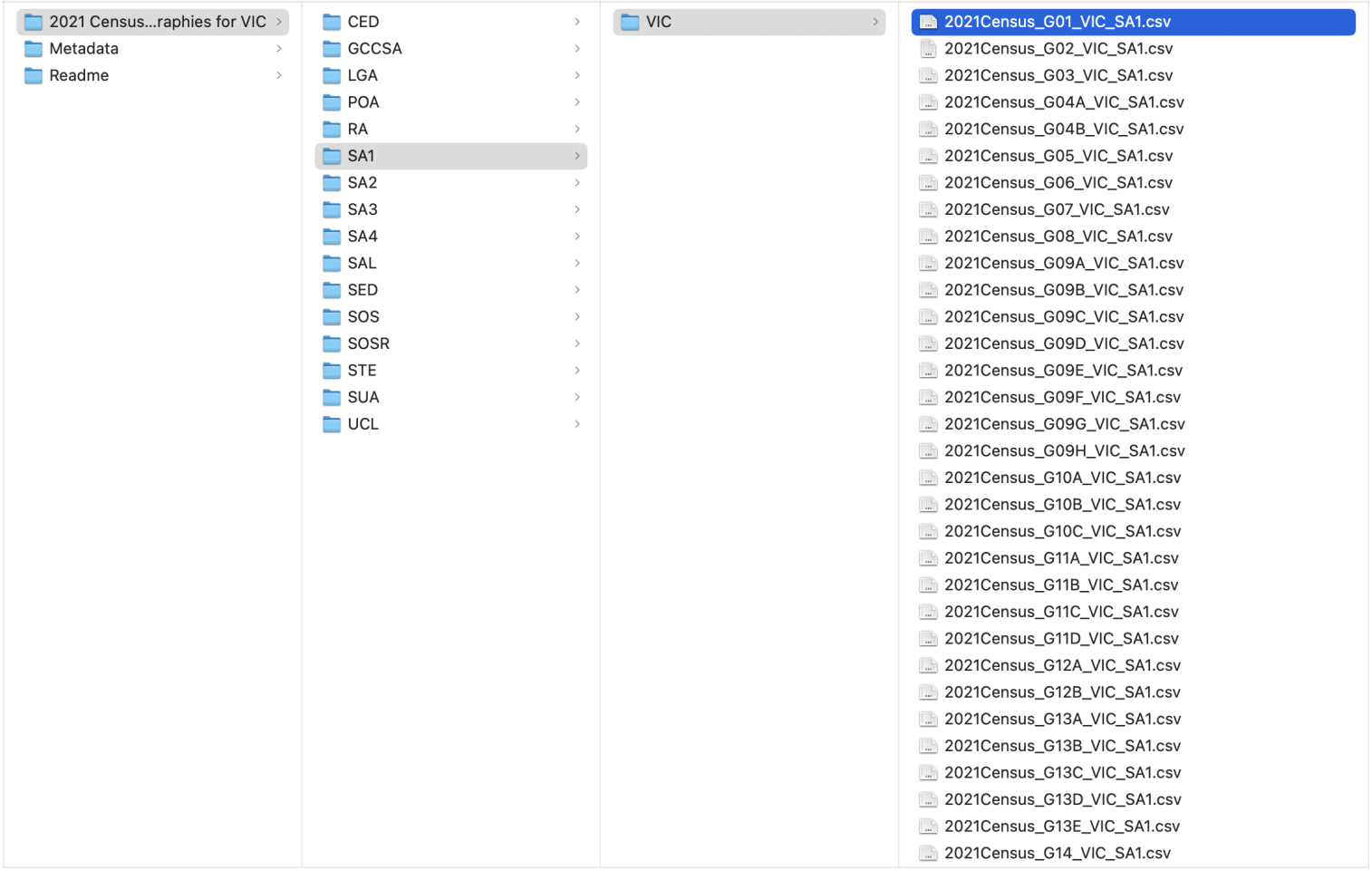
- Much like real detective work, **just locating the data and understanding the data variables can take a long time**

- **Cleaning and wrangling of the data is not glamorous**;
  There's far more attention in "catching criminals" / praise for the cool discoveries from statistical analysis.

Let's get delve into 'grunt work' of an analysis with the census data!

Data Structure and what's in it?

Datapack data structure



- The data is nested within folders. *Click on the folder name to see folders and files nested within.*

- Preserve the data in the original structure as much as you can! *Good practice not to modify the raw data and it's structure*

Where do we get started??

What is stored in each of these folders/files??

Read Me and Meta Data

Download the 2021 Census data containing the General Community Profile for all geographies in Victoria.

We need some description or understanding of the variables.
It will be near impossible to extract meaningful information from the data without it.

Breakout Session

Then take some time to review the read me and the meta data folders.

- Which folder contains demographic information about each suburb?
- What is LGA short for?
- Where can I find information about how much rent people pay?
- What is contained in variable G17?

Table G17

There are few things to note:

- There are 201 columns in G17A and G17B and 81 columns in G17C.
- Perhaps there is an export limitation for a data that contains more than 200 columns, thus it is broken up into different csv files.
- Which means that you have to join the tables G17A, G17B and G17C as one *(you'll do this in the tutorial 🔧)*.

❓ But what does the data show?

Tables G17A-G17C

2021Census_G17A_VIC_STE.csv

```
##   STE_CODE_2021 M_Neg_Nil_income_15_19_yrs M_Neg_Nil_income_20_24_yrs
## 1             2                      88386                      21186
```

2021Census_G17B_VIC_STE.csv

```
##   STE_CODE_2021 F_300_399_15_19_yrs F_300_399_20_24_yrs
## 1             2                8810               19537
```

2021Census_G17C_VIC_STE.csv

```
##   STE_CODE_2021 P_650_799_15_19_yrs P_650_799_20_24_yrs
## 1             2                7670               45029
```

Tidy Data

What is Tidy Data?

Tidy Data Principles

1. Each variable must have its own column
2. Each observation must have its own row
3. Each value must have its own cell

So what about the ABS Census Data?

- The table header in fact contains information!
- E.g. `F_400_499_15_19_yrs` is female aged 15-19 years old who earn $400-499 per week (in Victoria).
- The number in the cells are the **counts**.
- Is the data tidy?

Tidying the ABS 2016 Census Data

- Ideally we want the data to look like:

```
##   age_min age_max gender income_min income_max count
## 1      15      19 female        400        499  4020
```

- Putting data into a tidy format makes the data analysis easier.
- You can include other information, e.g. geography code (useful if combining with other geographical area) or average age/income.
- Note some categories do not have upper bounds, e.g. `M_3000_more_85ov`. In R, `-Inf` and `Inf` are used to represent $-\infty$ and ∞, respectively.
- You'll wrangle the data into the tidy form in tutorial 🔧
- This will require getting the pieces of information from the column names and organising them using string manipulation.

Manipulating strings

Manipulating strings

- The `stringr` package provides a set of functions designed to help with string manipulation.

```
library(tidyverse) # includes `stringr`
```

- Main functions in `stringr` begin with the **prefix with `str_`** and the first input into the functions is a string (or a vector of strings)
- What do you think `str_trim` and `str_squish` do?

```
str_trim(c("    Apple ", "  Goji    Berry     "))
```

```
## [1] "Apple"         "Goji    Berry"
```

```
str_squish(c("    Apple ", "  Goji    Berry     "))
```

```
## [1] "Apple"      "Goji Berry"
```

- Click here for a cheat sheet for `stringr` functions.

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0.

Gagolewski M. and others (2020). R package stringi: Character string processing facilities.

Some other examples

These are `stringr` functions we'll need for our census application.

Splitting strings by a pattern:

```
str_split(string = "Hi_everyone_in_ETC5512", pattern = "_")

## [[1]]
## [1] "Hi"       "everyone" "in"       "ETC5512"
```

Replacing parts of strings with a different pattern:

```
str_replace(string = "we_want_fourwords", pattern = "rw", replace = "r_w")

## [1] "we_want_four_words"
```

Deleting parts of strings that aren't imporant:

```
str_remove(string = "we_want_to_remove_the_extra_stuff", pattern = "to_remove_th

## [1] "we_want_stuff"
```

To get more control over the kinds of patterns we can match, we need regular expressions.

Regular expressions Part 1

- **Regular expression**, or **regex**, is a string of characters that define a search pattern for text
- Regular expression is... hard, but comes up often enough that it's worth learning

```
ozanimals <- c("koala", "kangaroo", "kookaburra", "numbat")
```

= Basic match

```
str_detect(ozanimals, "oo")

## [1] FALSE  TRUE  TRUE FALSE

str_extract(ozanimals, "oo")

## [1] NA   "oo" "oo" NA
```

```
str_match(ozanimals, "oo")

##      [,1]
## [1,] NA
## [2,] "oo"
## [3,] "oo"
## [4,] NA
```

Regular expressions Part 2

Meta-characters

- `"."` a wildcard to match any character except a new line

```
str_starts(c("color", "colouur", "colour", "red-column"), "col...")
```

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

- `"(.|.)"` a marked subexpression with alternate possibilites marked with `|`

```
str_replace(c("lovelove", "move", "stove", "drove"), "(l|dr|st)o", "ha")
```

```
## [1] "havelove" "move"     "have"     "have"
```

- `"[...]"` matches a single character contained in the bracket

```
str_replace_all(c("cake", "cookie", "lamington"), "[aeiou]", "_")
```

```
## [1] "c_k_"      "c__k__"    "l_m_ngt_n"
```

Regular expressions Part 3

Meta-character quantifiers

- "?" zero or one occurence of preceding element

```
str_extract(c("color", "colouur", "colour", "red"), "colou?r")
```

```
## [1] "color"  NA       "colour" NA
```

- "*" zero or more occurence of preceding element

```
str_extract(c("color", "colouur", "colour", "red"), "colou*r")
```

```
## [1] "color"   "colouur" "colour"  NA
```

- "+" one or more occurence of preceding element

```
str_extract(c("color", "colouur", "colour", "red"), "colou+r")
```

```
## [1] NA        "colouur" "colour"  NA
```

Regular expressions Part 4

- `"{n}"` preceding element is matched exactly `n` times

```
str_replace(c("banana", "bananana", "bana", "bananananana"), "ba(na){2}", "-")
```

```
## [1] "-"     "-na"   "bana"  "-nana"
```

- `"{min,}"` preceding element is matched `min` times or more

```
str_replace(c("banana", "bananana", "bana", "bananananana"), "ba(na){2,}", "-")
```

```
## [1] "-"    "-"    "bana" "-"
```

- `"{min,max}"` preceding element is matched at least `min` times but no more than `max` times

```
str_replace(c("banana", "bananana", "bana", "bananananana"), "ba(na){1,2}", "-")
```

```
## [1] "-"     "-na"   "-"     "-nana"
```

Regular expressions Part 5

Character classes

- `[:alpha:]` or `[A-Za-z]` to match alphabetic characters
- `[:alnum:]` or `[A-Za-z0-9]` to match alphanumeric characters
- `[:digit:]` or `[0-9]` or `\\d` to match a digit
- `[^0-9]` to match non-digits
- `[a-c]` to match a, b or c
- `[A-Z]` to match uppercase letters
- `[a-z]` to match lowercase letters
- `[:space:]` or `[ \t\r\n\v\f]` to match whitespace characters
- and more...

View matches with regular expressions

```
str_view(c("banana", "bananana", "bana", "banabanana"), "ba(na){1,2}")
```

```
## [1] | <banana>
## [2] | <banana>
## [3] | <bana>
## [4] | <bana><b
```

- When a function in `stringr` ends with `_all`, all matches of the pattern are considered
- The one *without* `_all` only considers the first match

```
str_view_all(c("banana", "bananana", "bana", "banabanana"), "ba(na){1,2}")
```

```
## [1] | <banana>
## [2] | <banana>na
## [3] | <bana>
## [4] | <bana><banana>
```

Weird characters

Characters we use to define the regex, e.g. *,.,!,?,),] need to be defined differently when we are trying to match them.

This doesn't work:

```
str_extract("Let's get the character and the brackets (A)", "([:alpha:])")

## [1] "L"

str_view("Let's get the character and the brackets (A)", "([:alpha:])")

## [1] | <L><e><t>'<s> <g><e><t> <t><h><e> <c><h><a><r><a><c><t><e><r> <a><n><d
```

But this does.

```
str_extract("Let's get the character and the brackets (A)", "\\([:alpha:]\\)" )

## [1] "(A)"
```

To match a bracket `(` we need to use `\\(` in stringr. It tells R we are looking for the bracket as part of the pattern and not to look for the backslash. The same goes for other special characters:

Back to Census

Raw Data vs. Aggregated Data

- Although the data collected was from individual households, with each person in the household surveyed (see sample form here), the downloaded data are **aggregated**.
- Aggregate data presents summary statistics from the **raw data**. *(e.g. a common summary statistic is the mean)*.
- When the summary statistics are counts then it is often called **frequency data**.
- The raw data collected would be similar to the form

household_id	person	gender	age	maritial_status	income_per_week
1	John Smith	F	40	Married	400-499
1	Jane Smith	M	39	Married	300-399
1	David Smith	M	10	Never married	Nil
1	Mary Smith	F	8	Never married	Nil
2	John Citizen	M	32	Never married	400-499
2	Jane Citizen	F	33	Never married	1750-1999

What you lose in aggregate data

- For aggregate data, there are less scope for you to draw insights conditioned on other variables.
- *e.g. Based on frequency data alone, you cannot answer questions like: How many middle income families have 2 children?*
- Raw data are desirable if you can get hold of it!

Trust and skepticism

- By the way, did you notice anything odd about the dummy data presented in the last slide?
- John Smith was recorded as female and Jane Smith as male. Data may have been incorrectly recorded.
- How much do you trust the aggregate data?
- Remember to have a healthy dose of skepticism in your data.

Data Confidentiality

- The data is not just aggregated, but it is also anonymised
- E.g. in `2021_GCP_Sequential_Template_R2.xlsx`, Sheet "G17", footnote says "*Please note that there are **small random adjustments** made to all cell values to protect the confidentiality of data. These adjustments may cause the sum of rows or columns to differ by small amounts from table totals.*"

> Do you think that you'll get the same numbers if you use the ones from different geographical code? E.g. SA1 and STE.

- You can check this in the tutorial 🔧

Summary

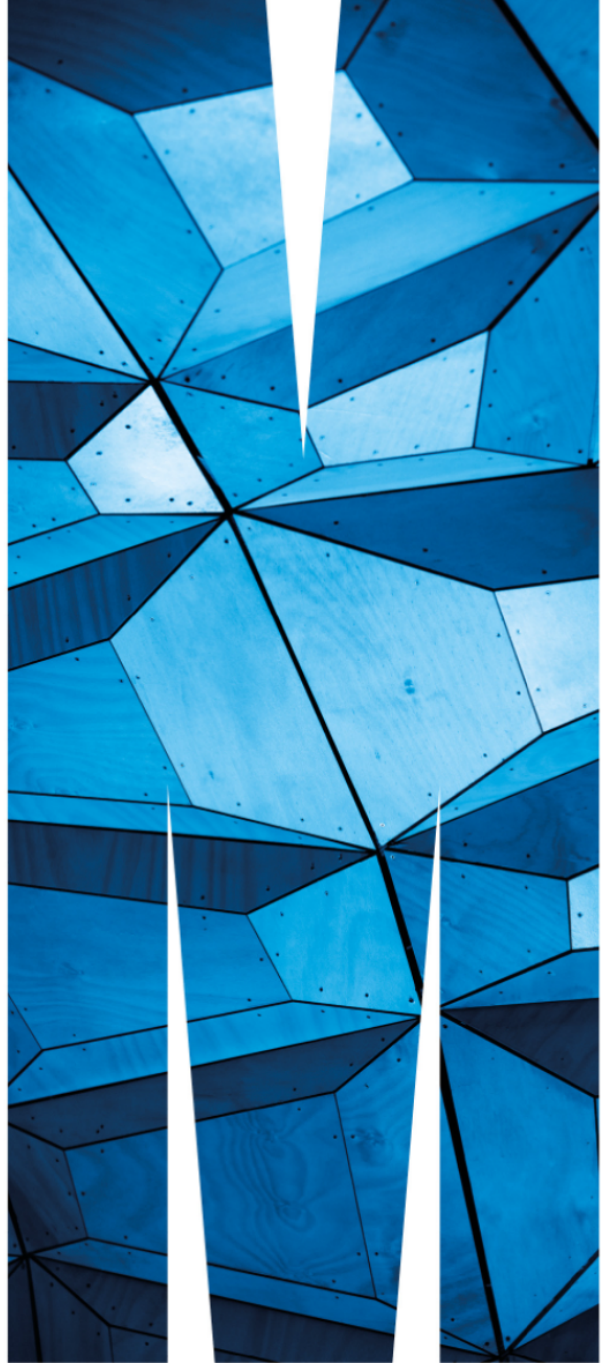
- We went through how to locate and understand the data available in the 2021 Australian census.
- We know some limitations with this data.
- We learnt about what tidy data is.
- We learnt a little about how to manipulate strings

Answers to break out questions

- Which folder contains demographic information about each suburb?
  In the file `2021AboutDataPacks_readme.txt` you find out that folders represent different geographical sub-regions. SAL represents suburbs and locaties and in the previous census was called SSC.
- What is LGA short for?
  Local Government Areas
- Where can I find information about how much rent people pay?
  In the file `2021_GCP_Sequential_Template_R2` there is a list of variables and what is contained in each table. G40 contains the rental information (organised by landlord type).
- What is contained in variable G17?
  G17 contains information about the total personal income organised by age and sex.

Slides developed by Dr. Emi Tanaka and updated by Dr. Kate Saunders

MONASH University



Lecturer: *Kate Saunders*

Department of Econometrics and Business Statistics

ETC5512.Clayton-x@monash.edu

Week 4

Case study 🔍 Aussie Local Government Area

```
LGA <- ozmaps::abs_lga %>% pull(NAME)
LGA[1:7]
```

```
## [1] "Broken Hill (C)" "Waroona (S)"     "Toowoomba (R)"   "West Arthur (S)"
## [5] "Moreton Bay (R)" "Etheridge (S)"   "Cleve (DC)"
```

C = Cities	A = Areas	RC = Rural Cities
B = Boroughs	S = Shires	DC = District Councils
M = Municipalities	T = Towns	AC = Aboriginal Councils
RegC = Regional Councils		

🎯 **Extract the LGA status from the LGA names**

How?

Extracting the string

```
str_extract(LGA, "\\(.+\\)")
```

```
##   [1] "(C)"          "(S)"          "(R)"          "(S)"          "(R)"
##   [6] "(S)"          "(DC)"         "(R)"          "(DC)"         "(C)"
##  [11] "(DC)"         "(S)"          "(S)"          "(S)"          "(S)"
##  [16] "(A)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [21] "(A)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [26] "(DC)"         "(S)"          "(S)"          "(S)"          "(S)"
##  [31] "(S)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [36] "(R)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [41] "(S)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [46] "(AC)"         "(S)"          "(S)"          "(S)"          "(S)"
##  [51] "(A)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [56] "(S)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [61] "(C)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [66] "(C)"          "(S)"          "(S)"          "(S)"          "(S)"
##  [71] "(R)"          "(R)"          "(S)"          "(B)"          "(DC)"
##  [76] "(M)"          "(A)"          "(C)"          "(S)"          "(S)"
```

- What is `"\\(.+\\)"`???
- This is a pattern expressed as **regular expression** or **regex** for short
- Note in R, you have to add an extra `\` when `\` is included in the pattern (yes this means that you can have a lot of backslashes... just keep adding `\` until it works! Enjoy this xkcd comic.)
- From R v4.0.0 onwards, you can use raw string to elimiate all the extra `\`, e.g. `r"(\(.+\))" is the same as "\\(.+\\)"`

Back to Extracting the string

```
str_extract(LGA, "\\(.+\\)") %>%
  table()

## .
##       (A)       (AC)        (B)        (C)  (C) (NSW)   (C) (SA) (C) (Vic.)
##       100          2          1        120          2          1          2
##      (DC)  (DC) (SA)        (M) (M) (Tas.)        (R)  (R) (Qld)       (RC)
##        40          1         23          4         38          1          7
##     (RegC)        (S)  (S) (Qld)        (T)
##         1        182          1         12
```

Where the same Local Government Area name appears in different States or Territories, the State or Territory abbreviation appears in parenthesis after the name. Local Government Area names are therefore unique.

-Australian Bureau of Statistics

Retry Extracting the string

```
str_extract(LGA, "\\([^)]+\\)") %>%
  # remove the brackets
  str_replace_all("[\\(\\)]", "") %>%
  table()

## .
##   A  AC   B   C  DC   M   R  RC RegC   S   T
## 100   2   1 125  41  27  39   7    1 183  12
```

- `"[ ]"` for single character match
- We want to match `(` and `)` but these are meta-characters
- So we need to escape it to have it as a literal: `\(` and `\)`
- But we must escape the escape character... so it's actually `\\( \\)`

R v4.0.0 Extracting the string

```
str_extract(LGA, r"(\([^)]+\))") %>%
  # remove the brackets
  str_replace_all(r"([\(\)])", "") %>%
  table()
```

```
## .
##   A  AC   B   C  DC   M   R  RC RegC   S   T
## 100   2   1 125  41  27  39   7    1 183  12
```

- If using R v4.0.0 onwards, you can use the raw string version instead