

ETC5512: Wild Caught Data

Australian election data

Lecturer: *Kate Saunders*

Department of Econometrics and Business Statistics

✉ ETC5512.Clayton-x@monash.edu

📅 Week 5



Australian election data

- Much like the census, **election** attempts to collect the data from the population.
- In Australia, it is compulsory by law to vote in elections if you are an Australian citizen (or eligible British subject) aged 18 years old or over and have lived in your address for at least one month.
- The Australian Electoral Commission (AEC) is an independent federal agency in charge of federal Australian elections and provides the geographical boundaries of the electoral divisions.



1. When was the last federal election in Australia?
2. How often is the federal election conducted in Australia?
3. How many electoral divisions are there in the last federal election?
4. What is the population for the Australian federal election?



Today you will:

- Learn about Australian election data
- Look at the 2022 election results
- Learn to visualise the election results spatially in a few ways
- Learn about reprojecting geographic data into different coordinate reference systems



From a coding perspective:

- This will require learning about **mapping in R**.
- You will also need to learn about different **mapping projections**

2022 Australian Federal Election

- Parliament of Australia comprises two houses:
 - **Senate** (upper house) comprising 76 senators
 - **House of Representatives** (lower house) comprising 151 members
- Government is formed by the party or coalition with majority of the seats in the lower house
- The 2022 Australian Federal Election was held on Sat 21st May 2022
- The next federal election will be likely be held in 2025

- Major parties in Australia:

- Coalition:




- Some minor parties in Australia:










Ballots

- **House of Representatives** uses the instant-runoff voting system
- Senate uses the single transferable voting system

 House of Representatives
Ballot Paper


State
Electoral Division of Division Name

Number the boxes from 1 to 8 in the order of your choice

	2	SURNAME, Given Names INDEPENDENT
	3	SURNAME, Given Names PARTY
	7	SURNAME, Given Names PARTY
	4	SURNAME, Given Names PARTY
	1	SURNAME, Given Names PARTY
	5	SURNAME, Given Names PARTY
	6	SURNAME, Given Names PARTY
	8	SURNAME, Given Names PARTY

Remember... number every box to make your vote count









SAMPLE

 Senate Ballot Paper
State – Election of 6 Senators

You may vote in one of two ways

Either

Above the line
By numbering at least 6 of these boxes in the order of your choice (with number 1 as your first choice).

A	B	C	D	E	F	G
		 			 	
5	2	1		3	6	4
PARTY	PARTY	PARTY	PARTY	PARTY	PARTY	

Or

Below the line
By numbering at least 12 of these boxes in the order of your choice (with number 1 as your first choice).

PARTY	PARTY	PARTY	PARTY	PARTY	PARTY	PARTY	UNGROUPED
<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names INDEPENDENT
<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names INDEPENDENT
<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	
	<input type="checkbox"/> SURNAME Given Names PARTY				<input type="checkbox"/> SURNAME Given Names PARTY	<input type="checkbox"/> SURNAME Given Names PARTY	

SAMPLE

2022 Australian Federal Election Data

- Get the distribution of preferences by candidate by division for the 2022 Australian Federal Election

 <https://results.aec.gov.au>



1. 2022 federal election
2. Downloads
3. Distribution of preferences by candidate by division

- Or refer directly to the link:

<https://results.aec.gov.au/27966/Website/Downloads/HouseDopByDivisionDownload-27966.csv>

House of Representative Voting Data

```
library(tidyverse)
votes <- read_csv("https://results.aec.gov.au/27966/Website/Downloads/HouseDopByDivisionDownload")
glimpse(votes)

## Rows: 35,096
## Columns: 14
## $ StateAb      <chr> "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT...
## $ DivisionID   <dbl> 318, 318, 318, 318, 318, 318, 318, 318, 318, 318, 318...
## $ DivisionNm    <chr> "Bean", "Bean", "Bean", "Bean", "Bean", "Bean", "Bean...
## $ CountNumber   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ BallotPosition <dbl> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, ...
## $ CandidateID   <dbl> 36239, 36239, 36239, 36239, 37455, 37455, 37455, 3745...
## $ Surname       <chr> "CONWAY", "CONWAY", "CONWAY", "CONWAY", "AMBARD", "AM...
## $ GivenNm       <chr> "Sean", "Sean", "Sean", "Sean", "Benjamin", "Benjamin...
## $ PartyAb       <chr> "UAPP", "UAPP", "UAPP", "UAPP", "ON", "ON", "ON", "ON...
## $ PartyNm       <chr> "United Australia Party", "United Australia Party", "...
## $ Elected      <chr> "N", "N", "N", "N", "N", "N", "N", "N", "Y", "Y", "Y"...
## $ HistoricElected <chr> "N", "N", "N", "N", "N", "N", "N", "N", "Y", "Y", "Y"...
## $ CalculationType <chr> "Preference Count", "Preference Percent", "Transfer C...
## $ CalculationValue <dbl> 2831.00, 2.88, 0.00, 0.00, 2680.00, 2.72, 0.00, 0.00, ...
```

Electoral district of Monash

- Let's have a look at the electoral district named "Monash"

```
## Rows: 224
## Columns: 14
## $ StateAb      <chr> "VIC", "VIC", "VIC", "VIC", "VIC", "VIC", "VIC", "VIC...
## $ DivisionID   <dbl> 323, 323, 323, 323, 323, 323, 323, 323, 323, 323, 323...
## $ DivisionNm   <chr> "Monash", "Monash", "Monash", "Monash", "Monash", "Mo...
## $ CountNumber  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ BallotPosition <dbl> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, ...
## $ CandidateID  <dbl> 36561, 36561, 36561, 36561, 36737, 36737, 36737, 3673...
## $ Surname      <chr> "MORGAN", "MORGAN", "MORGAN", "MORGAN", "BROADBENT", ...
## $ GivenNm      <chr> "Mat", "Mat", "Mat", "Mat", "Russell", "Russell", "Ru...
## $ PartyAb      <chr> "GVIC", "GVIC", "GVIC", "GVIC", "LP", "LP", "LP", "LP...
## $ PartyNm      <chr> "The Greens", "The Greens", "The Greens", "The Greens...
## $ Elected      <chr> "N", "N", "N", "N", "Y", "Y", "Y", "Y", "N", "N", "N" ...
## $ HistoricElected <chr> "N", "N", "N", "N", "Y", "Y", "Y", "Y", "N", "N", "N" ...
## $ CalculationType <chr> "Preference Count", "Preference Percent", "Transfer C...
## $ CalculationValue <dbl> 9533.00, 9.86, 0.00, 0.00, 36546.00, 37.79, 0.00, 0.0...
```

District: Monash

```
votes_monash <- votes %>%  
  # get the preference count only  
  filter(CalculationType == "Preference Count") %>%  
  # get the Monash division  
  filter(DivisionNm == "Monash")  
glimpse(votes_monash)
```

```
## Rows: 56  
## Columns: 14  
## $ StateAb      <chr> "VIC", "VIC", "VIC", "VIC", "VIC", "VIC", "VIC", "VIC"  
## $ DivisionID   <dbl> 323, 323, 323, 323, 323, 323, 323, 323, 323, 323, 323  
## $ DivisionNm   <chr> "Monash", "Monash", "Monash", "Monash", "Monash", "Monash", "Monash", "Monash"  
## $ CountNumber  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2  
## $ BallotPosition <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8, 1, 2  
## $ CandidateID  <dbl> 36561, 36737, 36065, 37629, 37637, 36455, 36914, 3603...  
## $ Surname      <chr> "MORGAN", "BROADBENT", "LEONARD", "HICKEN", "WELSH", ...  
## $ GivenNm      <chr> "Mat", "Russell", "Deb", "Allan", "David Matthew", "J...  
## $ PartyAb      <chr> "GVIC", "LP", "IND", "ON", "CYA", "ALP", "LDP", "UAPP...  
## $ PartyNm      <chr> "The Greens", "Liberal", "Independent", "Pauline Hans...  
## $ Elected     <chr> "N", "Y", "N", "N", "N", "N", "N", "N", "N", "N", "Y", "N"...  
## $ HistoricElected <chr> "N", "Y", "N", "N", "N", "N", "N", "N", "N", "N", "Y", "N"...
```

House of Representatives
Ballot Paper

State
Electoral Division of Division Name

Number the boxes from 1 to 8 in the order of your choice

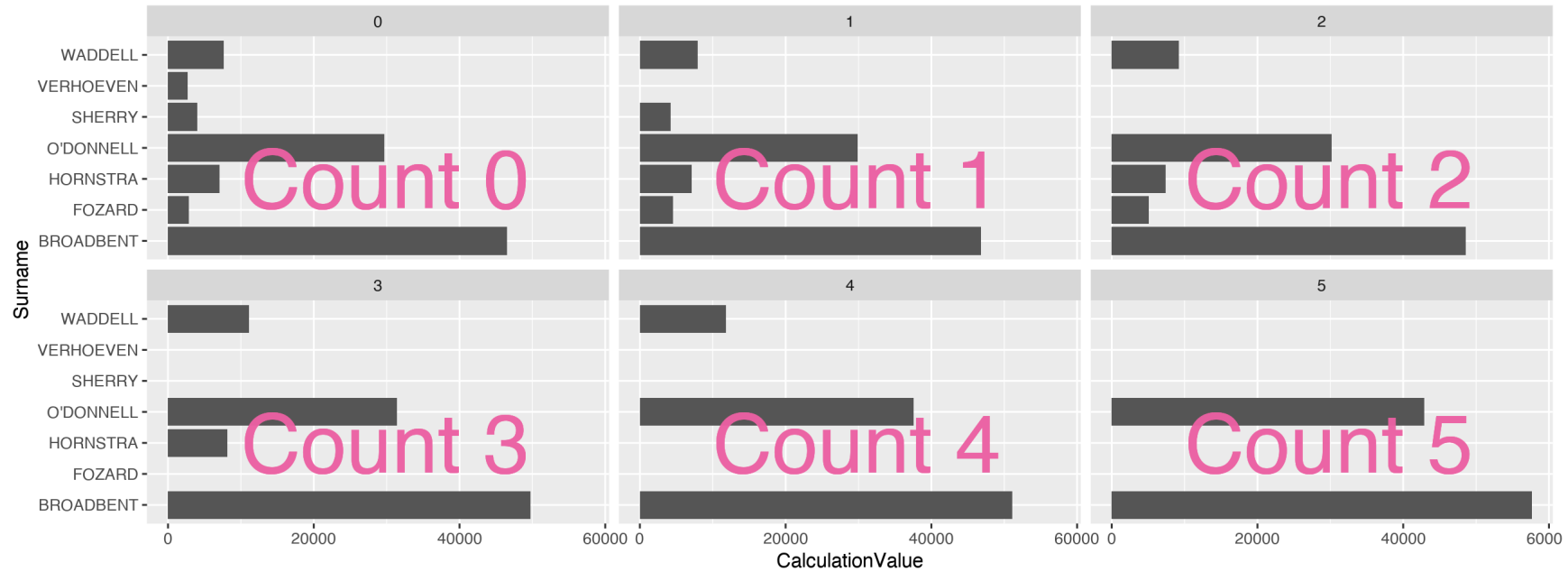
2	SURNAME, Given Names INDEPENDENT
3	SURNAME, Given Names PARTY
7	SURNAME, Given Names PARTY
4	SURNAME, Given Names PARTY
1	SURNAME, Given Names PARTY
5	SURNAME, Given Names PARTY
6	SURNAME, Given Names PARTY
8	SURNAME, Given Names PARTY

Remember... number every box to make your vote count

SAMPLE

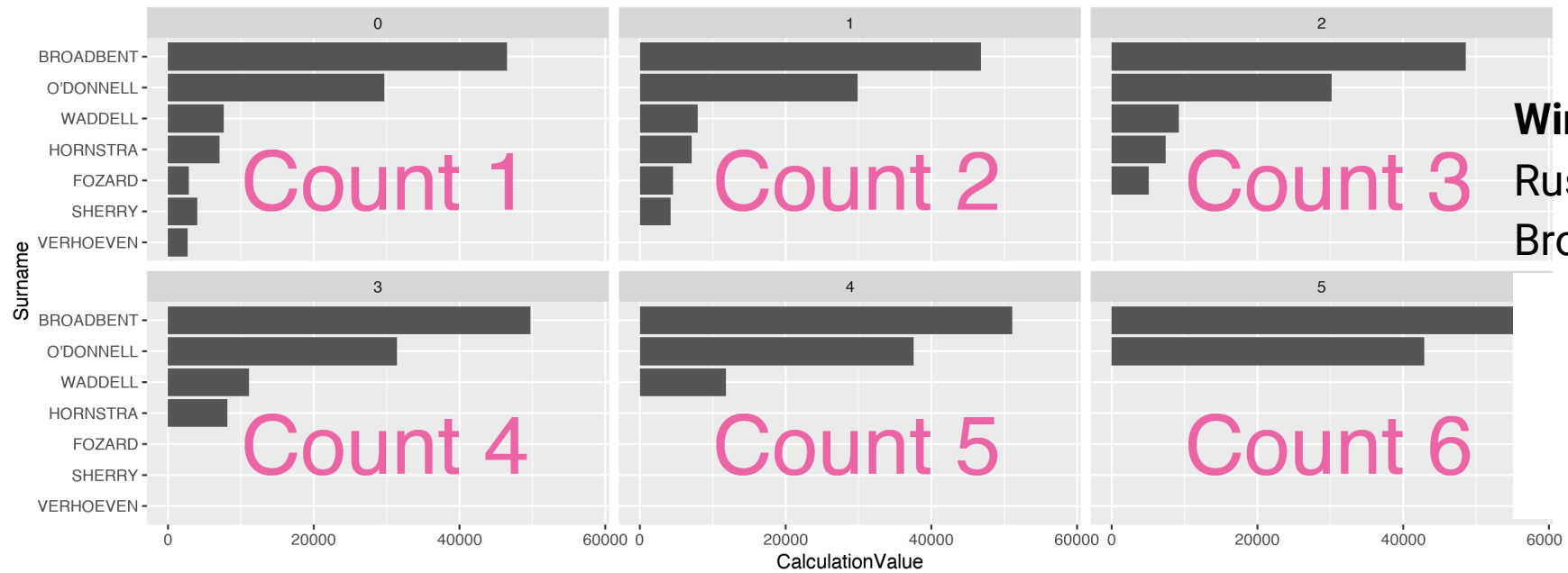
Visualising the counts

```
ggplot(votes_monash) +  
  geom_col(aes(x = CalculationValue, y = Surname)) +  
  geom_text(aes(label = paste("Count", CountNumber)),  
    x = 10000, y = 3, size = 16, color = "#ee64a4",  
    alpha = 0.4, hjust = "left"  
  ) +  
  facet_wrap(~CountNumber)
```



... but better to order candidates by counts

```
mutate(votes_monash, Surname = fct_reorder(Surname, CalculationValue, sum)) %>%  
  ggplot() +  
  geom_col(aes(x = CalculationValue, y = Surname)) +  
  geom_text(aes(label = paste("Count", CountNumber + 1)),  
    x = 10000, y = 3, size = 16, color = "#ee64a4", alpha = 0.4, hjust = "left"  
  ) +  
  facet_wrap(~CountNumber)
```

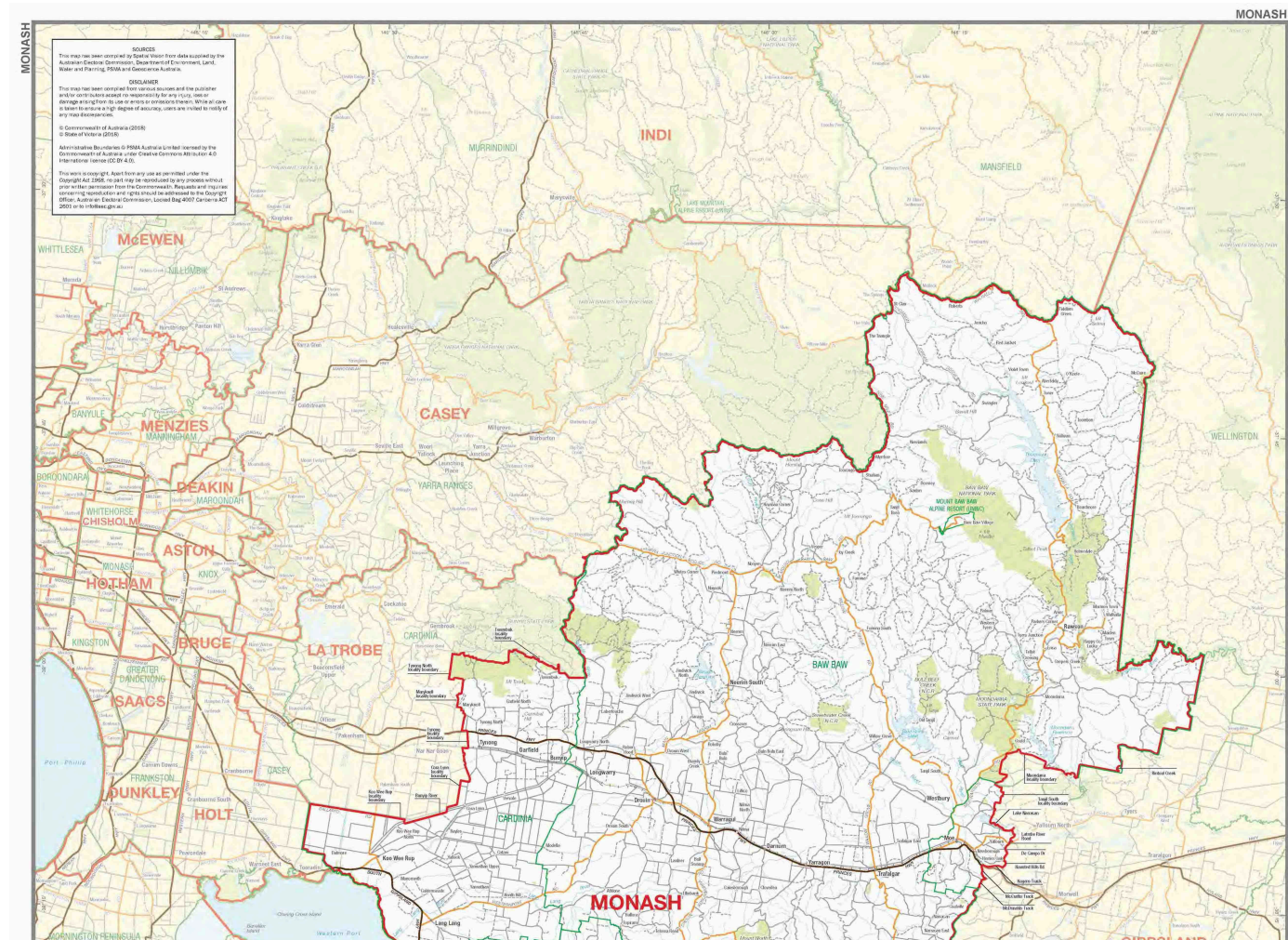


Winner:
Russel
Broadbent



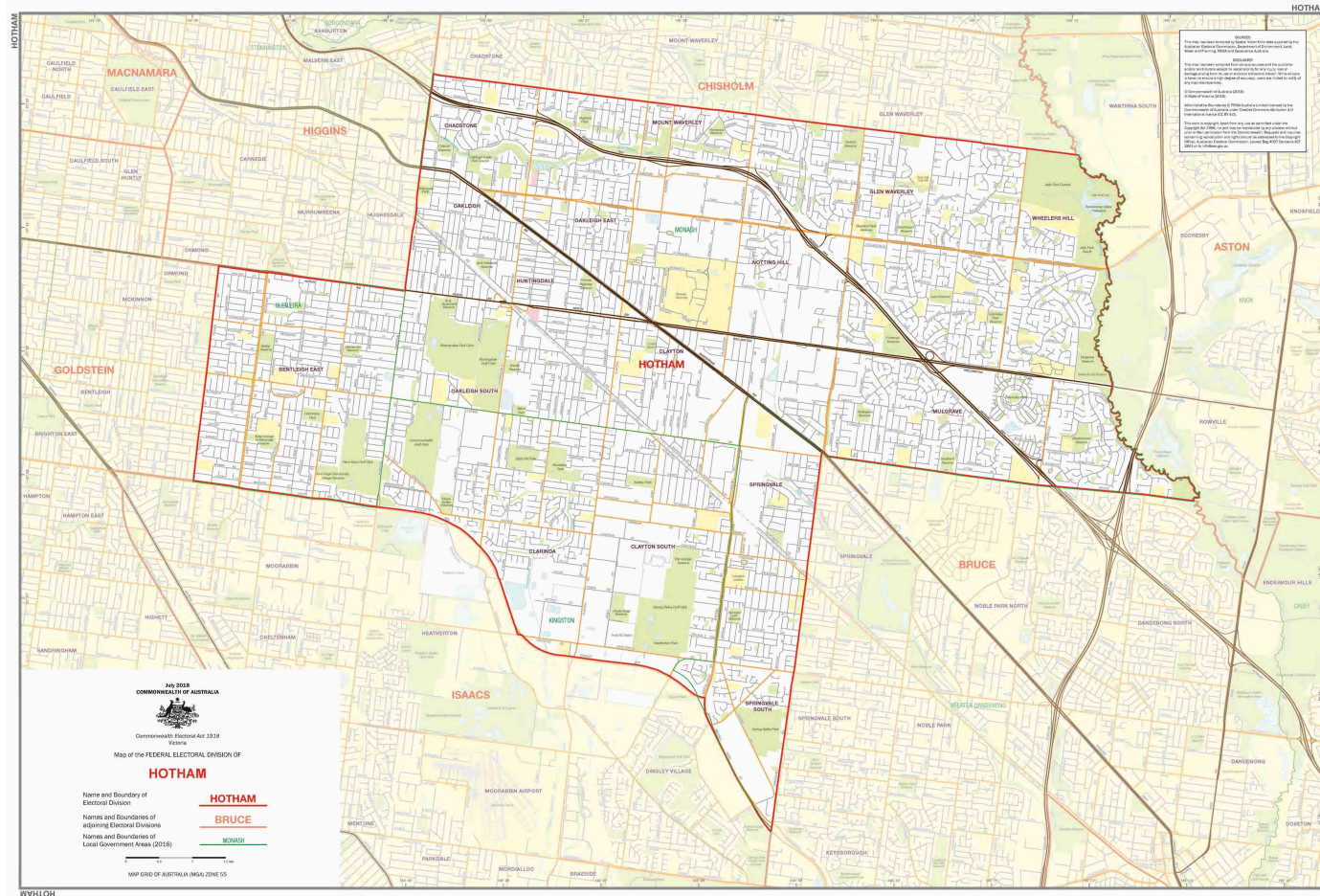
Where is the electoral district of **Monash**?

- ...*doesn't* include Monash Clayton campus



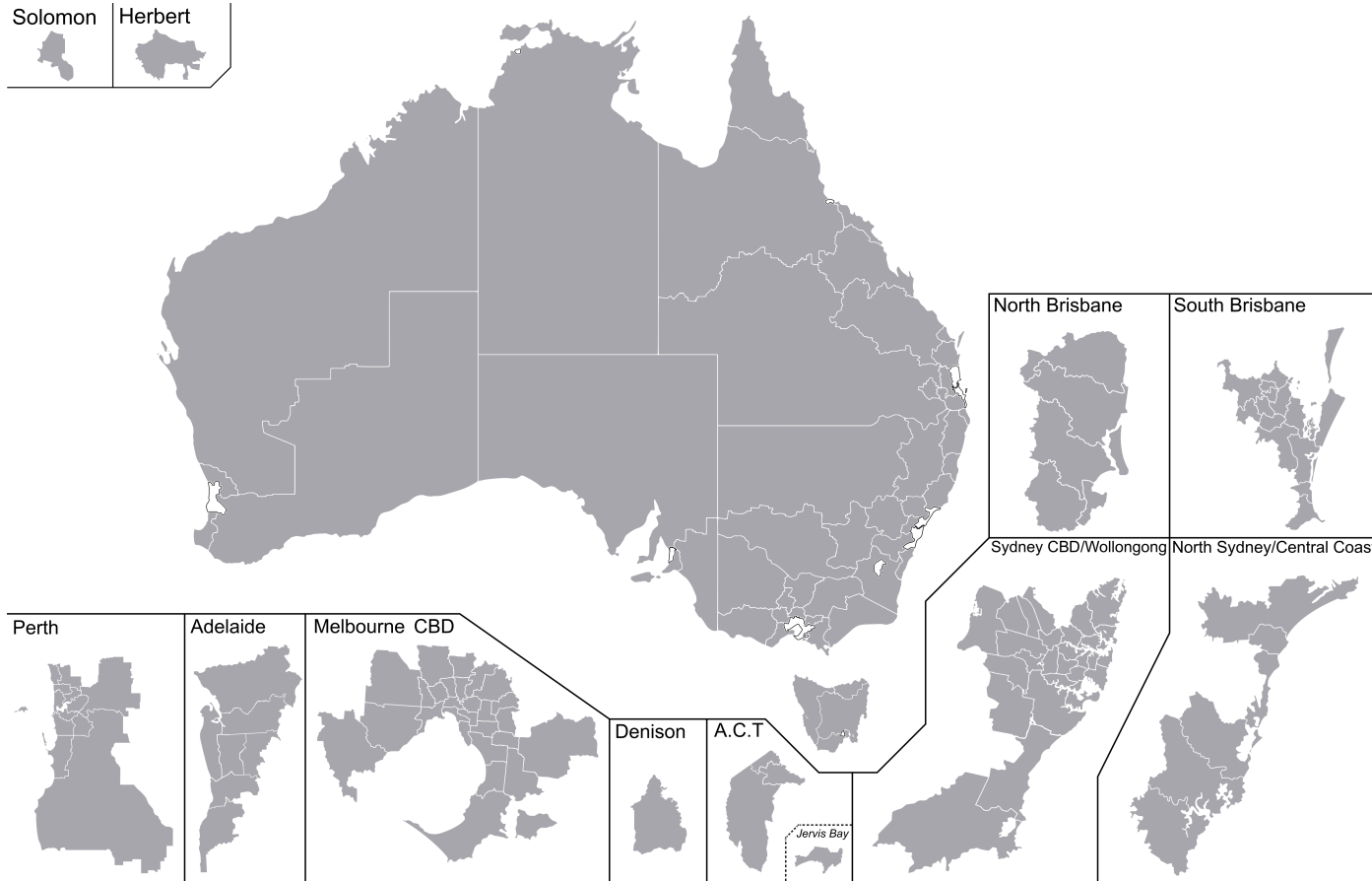
Electoral district of **Hotham**

- Does include Monash Clayton campus



Australian Electorates Divisions

There are 151 electorates in 2019.



The **geographical boundaries of the electoral divisions** are determined by the **Redistribution Committee** and are **redrawn** every so often to ensure similar number of electors in each electoral division for a given state or territory.



This means that the geographical boundaries could be different across years.

Federal electoral boundary GIS data

- GIS (Geographic Information System) is a framework that capture and inspect geographical data.
- This data is found at

 <https://www.aec.gov.au/electorates/gis/licence.htm>

- Agree to the license to get to the download page

“

The Licensee must make End-users aware the data was sourced from the Australian Electoral Commission and is used under licence.

Note: the federal electoral boundary is provided by Australian Electoral Commission

© Commonwealth of Australia (Australian Electoral Commission) 2021

- We download the ESRI zip file for Victoria.
- To work with spatial data, we use the [sf](#) R-package.

Working with spatial data

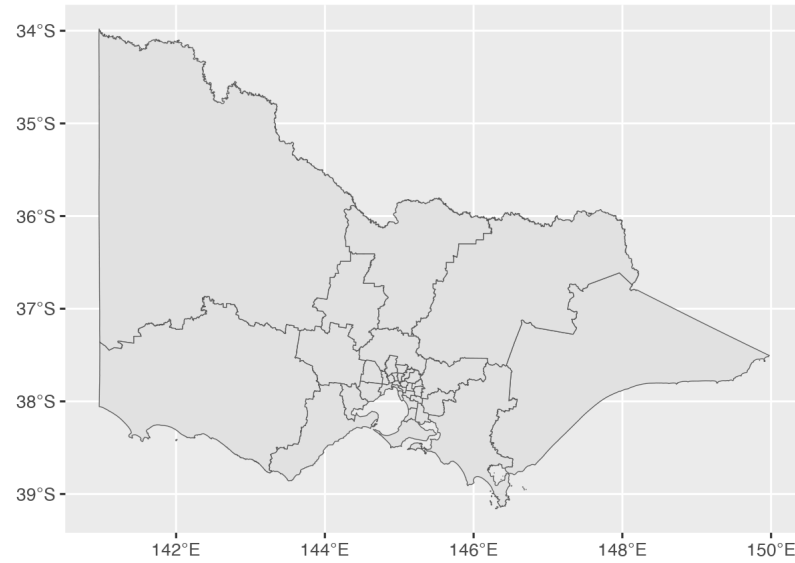
```
library(sf)
aec_map <- read_sf(here::here("data/vic-july-2021-esri/E_VIC21_region.shp"))
# aec_map
```

Geometry object and visualisation in as **ggplot**

```
aec_map$geometry[[1]]
```

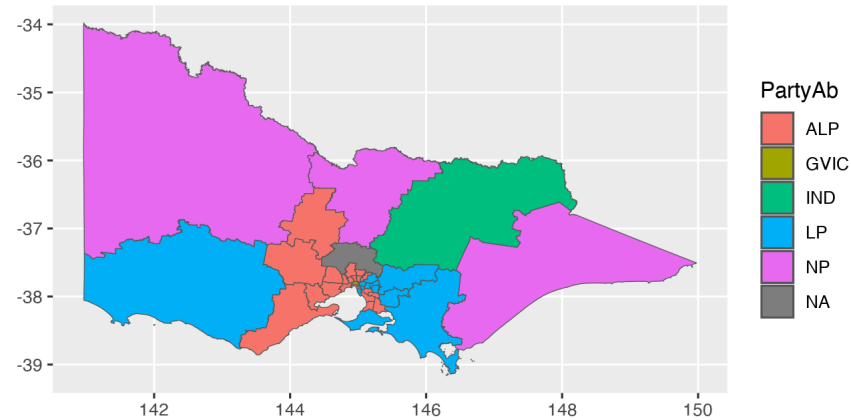
```
## MULTIPOLYGON Z (((145.3476 -37.85941 0, 145.3468 -37.8595 0, 145.3458 -37.859
```

```
ggplot(aec_map) +  
  geom_sf()
```



Integrating data of election winners

```
winners <- votes %>%  
  # get the winner  
  filter(Elected == "Y" & CountNumber == 0 & CalculationType == "Preference Count") %>%  
  # join the data  
  right_join(aec_map, by = c("DivisionNm" = "Elect_div")) %>%  
  select(DivisionNm, PartyAb, PartyNm, geometry)  
  
ggplot(winners) +  
  geom_sf(aes(fill = PartyAb, geometry = geometry))
```



Is there something wrong here?

Investigating missing observation

```
winners %>%
  filter(is.na(PartyAb))

## # A tibble: 1 × 4
##   DivisionNm PartyAb PartyNm geometry
##   <chr>      <chr>   <chr>   <MULTIPOLYGON [°]>
## 1 Mcewen    <NA>    <NA>    Z (((145.3664 -37.54513 0, 145.3663 -37.5459 0, 14...
```

```
--

votes %>%
  # approximate string matching (or fuzzy matching)
  filter(agrepl("McEwen", DivisionNm))

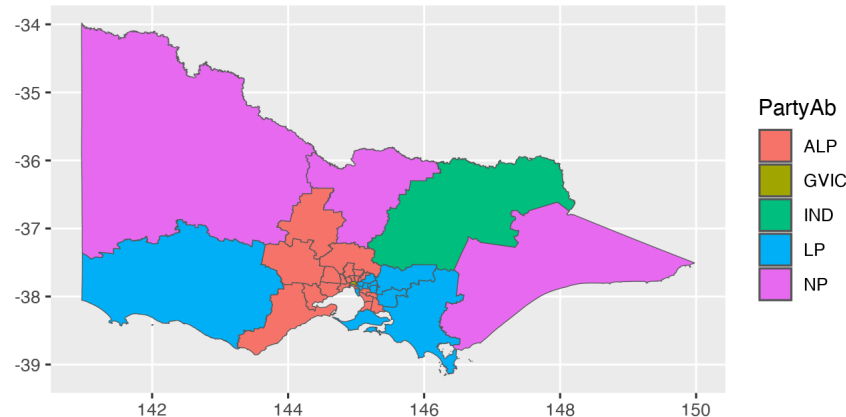
## # A tibble: 168 × 14
##   StateAb DivisionID DivisionNm CountNumber BallotPosition CandidateID Surname
##   <chr>      <dbl> <chr>      <dbl>      <dbl>      <dbl> <chr>
## 1 VIC          226 McEwen          0          1        36547 BARKER
## 2 VIC          226 McEwen          0          1        36547 BARKER
## 3 VIC          226 McEwen          0          1        36547 BARKER
## 4 VIC          226 McEwen          0          1        36547 BARKER
## 5 VIC          226 McEwen          0          2        36054 McRAE
## 6 VIC          226 McEwen          0          2        36054 McRAE
## 7 VIC          226 McEwen          0          2        36054 McRAE
## 8 VIC          226 McEwen          0          2        36054 McRAE
## 9 VIC          226 McEwen          0          3        37619 NEIL
## 10 VIC         226 McEwen          0          3        37619 NEIL
```

So what went wrong here?

Victoria map of election winners

```
winners_fix <- votes %>%
  mutate(DivisionNm = ifelse(DivisionNm == "McEwen", "Mcewen", DivisionNm)) %>%
  # get the winner
  filter(Elected == "Y" & CountNumber == 0 & CalculationType == "Preference Count") %>%
  # join the data
  right_join(aec_map, by = c("DivisionNm" = "Elect_div")) %>%
  select(DivisionNm, PartyAb, PartyNm, geometry)

ggplot(winners_fix) +
  geom_sf(aes(fill = PartyAb, geometry = geometry))
```



Maps visualisation

National map of election winners

```
ausmap <- read_sf(here::here("data/2021-Cwlth_electoral_boundaries_ESRI/2021_ELB_region.shp"))

all_winners <- votes %>%
  mutate(DivisionNm = case_when(
    DivisionNm == "McEwen" ~ "Mcewen",
    DivisionNm == "McPherson" ~ "Mcpherson",
    DivisionNm == "Eden-Monaro " ~ "Eden-monaro",
    DivisionNm == "McMahon" ~ "Mcmahon",
    DivisionNm == "O'Connor" ~ "O'connor",
    TRUE ~ DivisionNm
  )) %>%
  # Another way to select the winner
  filter(Elected == "Y") %>%
  group_by(DivisionID) %>%
  slice(1) %>%
  ungroup() %>%
  # then join the map data
  right_join(ausmap, by = c("DivisionNm" = "Elect_div"))
```

Code fix for previous slide to remove NAs on map

```
ausmap <- read_sf(here::here("data/2021-Cwlth_electoral_boundaries_ESRI/2021_ELB_region.shp"))

# Wrangle vote data: Want the winner for each electoral division
electoral_winners = votes |>
  # Make everything capitals to fix mismatches
  mutate(DivisionNm = toupper(DivisionNm)) |>
  # Filter to winners
  filter(Elected == "Y") |>
  # Select only the key variables we need for plotting
  select(PartyAb, DivisionID, DivisionNm, Elected) |>
  # We get a copy of the winner for each count, only need one
  distinct()

# Wrangle map data
ausmap = ausmap |>
  # Make everything capitals to fix mismatches
  mutate(Elect_div = toupper(Elect_div))

#Combine map data and electoral winners
all_winners = electoral_winners |>
  left_join(ausmap, by = c("DivisionNm" = "Elect_div"))
```

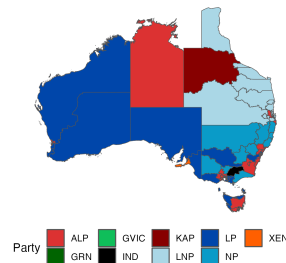
Using colors wisely

```
auscolours <- c(
  "ALP" = "#DE3533", "LNP" = "#ADD8E6", "KAP" = "#8B0000",
  "GVIC" = "#10C25B", "XEN" = "#ff6300", "LP" = "#0047AB",
  "NP" = "#0a9cca", "IND" = "#000000", "GRN" = "#006400"
)
# update this from lecture to include Greens

aus_winners_chloropleth_map <- ggplot(all_winners) +
  geom_sf(aes(fill = PartyAb, geometry = st_zm(geometry))) +
  scale_fill_manual(name = "Party", values = auscolours) +
  theme_void() +
  theme(legend.position = "bottom")

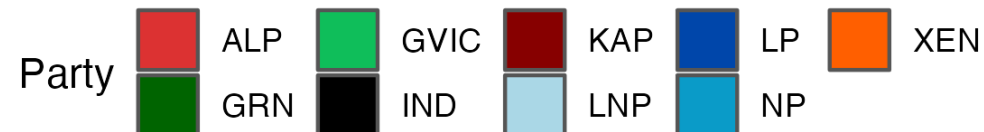
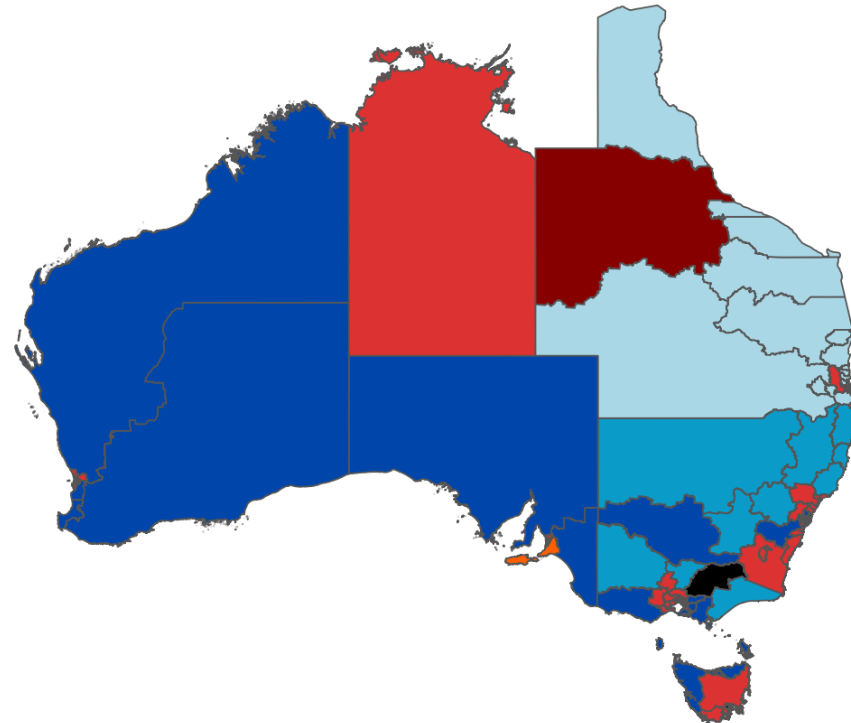
ggsave(aus_winners_chloropleth_map,
       file = "images/aus_winners_chloropleth_map.png")

# aus_winners_chloropleth_map
```



Choropleth Map

Which party won from looking at this map and by how much?



Liberal/National Coalition:

56

Labor: **75**

Greens: **4**

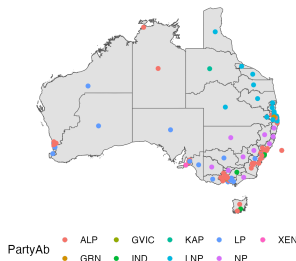
Katter's Australian: **1**

Centre Alliance: **1**

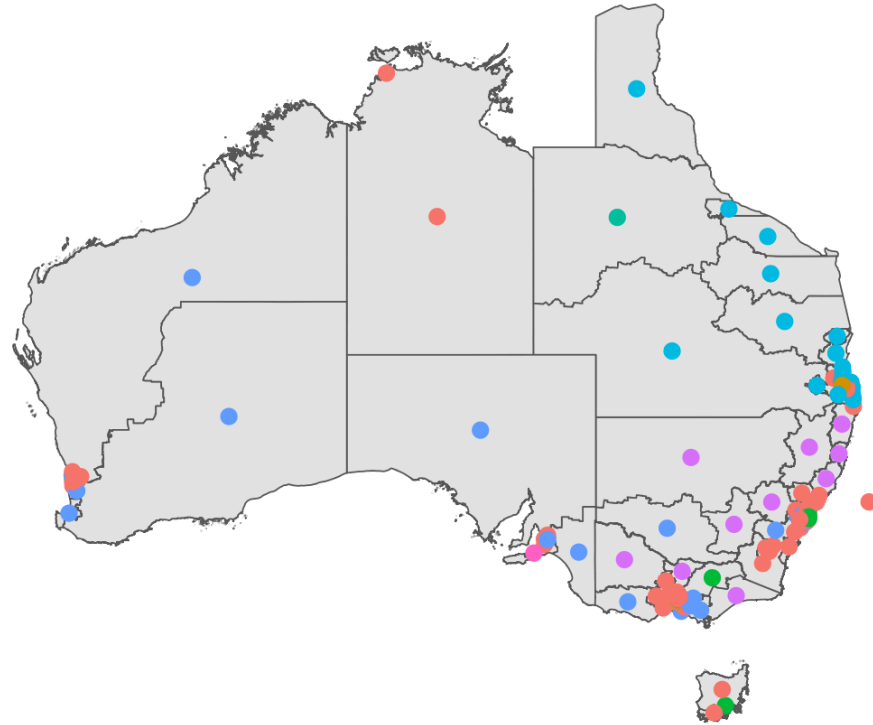
Independents: **10**

Mapping the centroids

```
all_winners_centroid <- all_winners %>%  
  # some issues with a duplicate edge  
  # slice(-79,-80) |> # for original code  
  slice(-93, -94) |> # for update post lecture  
  mutate(centroid = st_centroid(geometry))  
  
aus_winners_centroid_map <- ggplot(all_winners_centroid) +  
  geom_sf(aes(geometry = st_zm(geometry))) +  
  geom_sf(aes(geometry = centroid, color = PartyAb)) +  
  theme_void() +  
  theme(legend.position = "bottom")  
  
ggsave(aus_winners_centroid_map,  
       file = "images/aus_winners_centroid_map.png")  
  
# aus_winners_centroid_map
```



Non-Contiguous, Dorling Cartogram



PartyAb

● ALP	● GVIC	● KAP	● LP	● XEN
● GRN	● IND	● LNP	● NP	

Improving the national map of election winners

```
library(sugarbag)

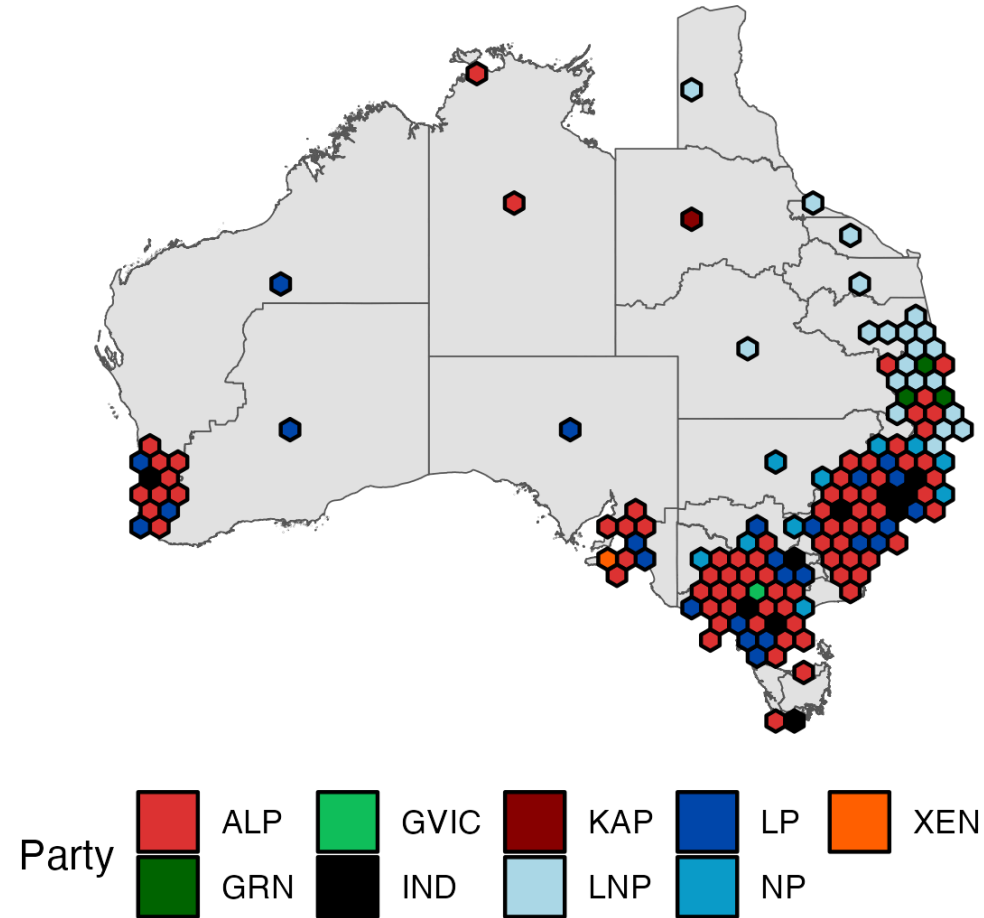
centroids <- all_winners_centroid %>%
  bind_cols(sf::st_coordinates(all_winners_centroid$centroid)) %>%
  select(DivisionNm, longitude = X, latitude = Y)

grid <- create_grid(centroids = centroids, hex_size = 0.9, buffer_dist = 5)

hex_allocated <- allocate(
  centroids = centroids,
  sf_id = "DivisionNm",
  hex_grid = grid,
  hex_size = 0.9, # same size used in create_grid
  hex_filter = 10,
  focal_points = capital_cities,
  width = 30, verbose = TRUE
)

hex_map <- hex_allocated %>%
  fortify_hexagon(hex_size = 0.9, sf_id = "DivisionNm") |>
  left_join(all_winners_centroid, by = "DivisionNm")
```

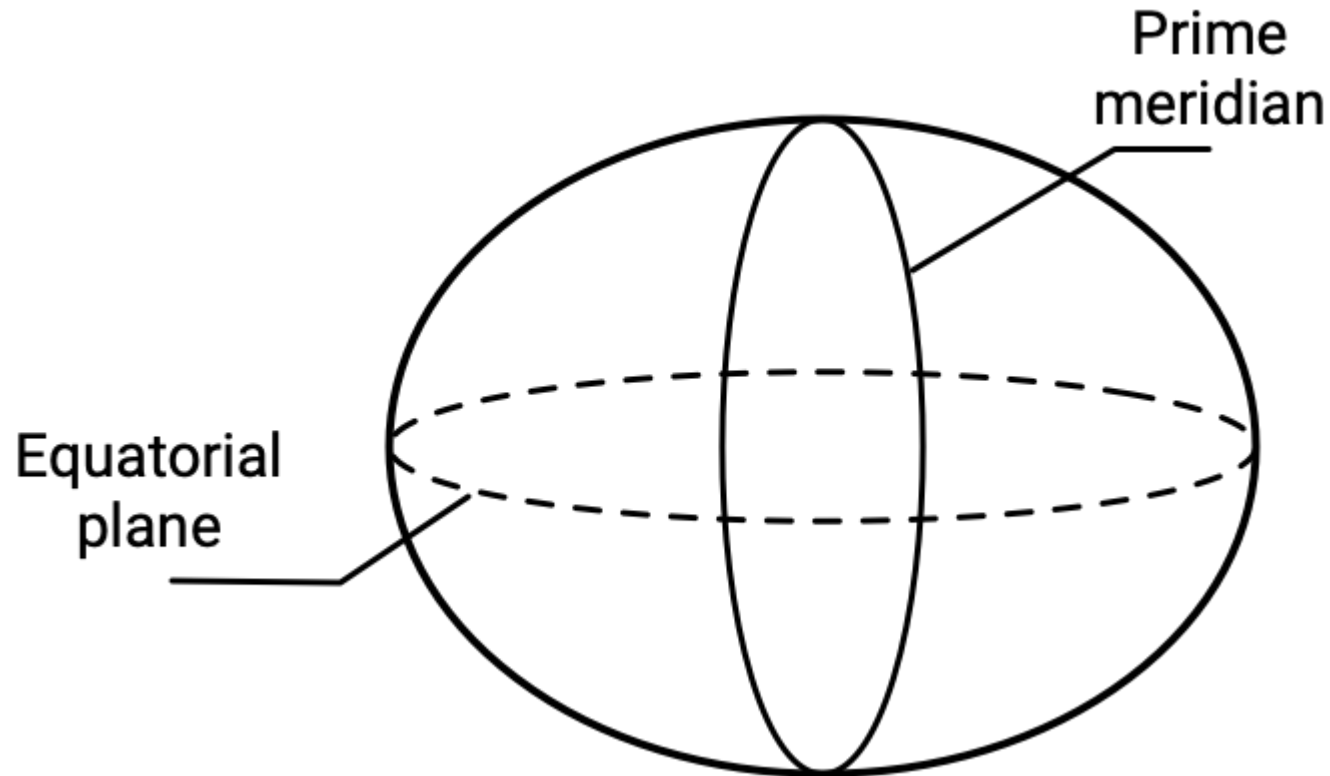

Tessellated Hexagon Map



Coordinate reference system (CRS)

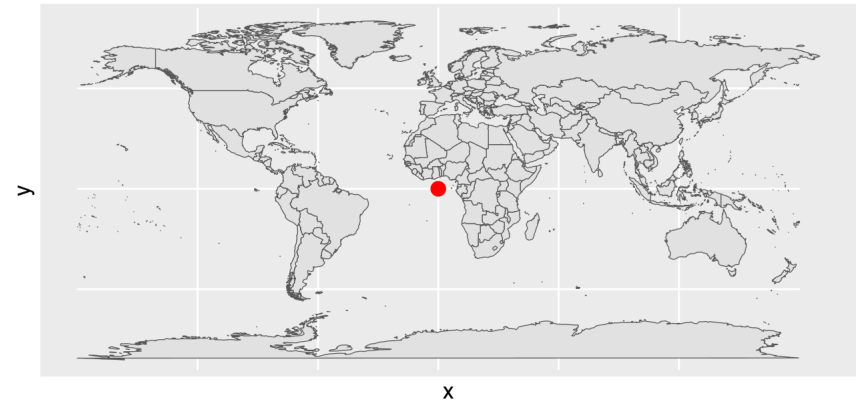
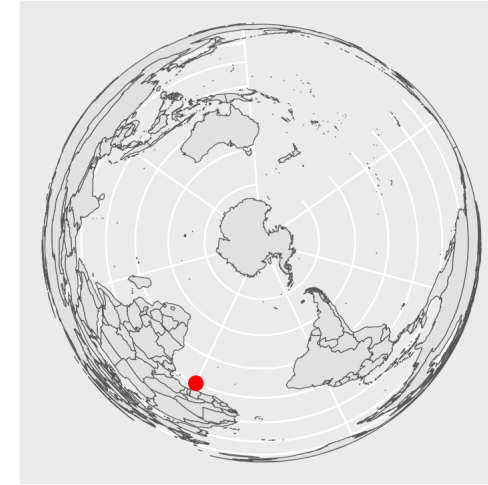
Geographic coordinate reference systems

- **Geographic CRSs** identify a location on the Earth's surface by *longitude* and *latitude*.
- **Longitude** is the East-West direction in angular distance from the Prime Meridian plane.
- **Latitude** is the angular distance North or South of the equatorial plane.



Projected coordinate reference systems

- All projected CRSs are based on a geographic CRS.
- Map projections convert the three-dimensional surface of the Earth into Easting and Northing (x and y) values (typically meters) in a projected CRS.
- These projected CRSs are based on Cartesian coordinates on an implicitly flat surface.
- Some deformations are introduced in the process, e.g. area, direction, distance or shape, while preserving one or two of these properties.

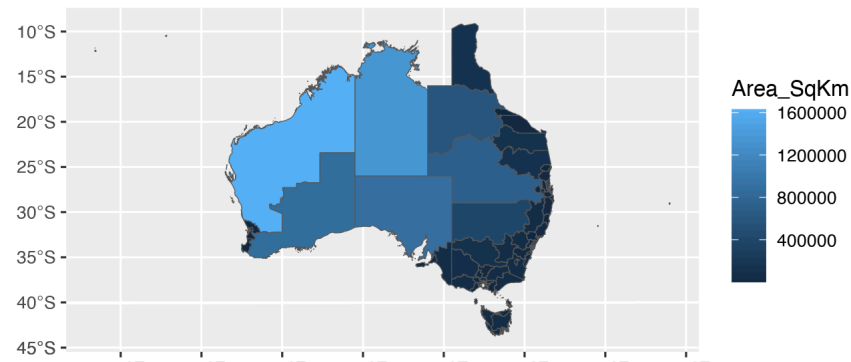


Well Known Text (WKT)

- Open Geospatial Consortium (OGC) developed an open standard format for describing CRSs called **WKT**

```
st_crs(ausmap)

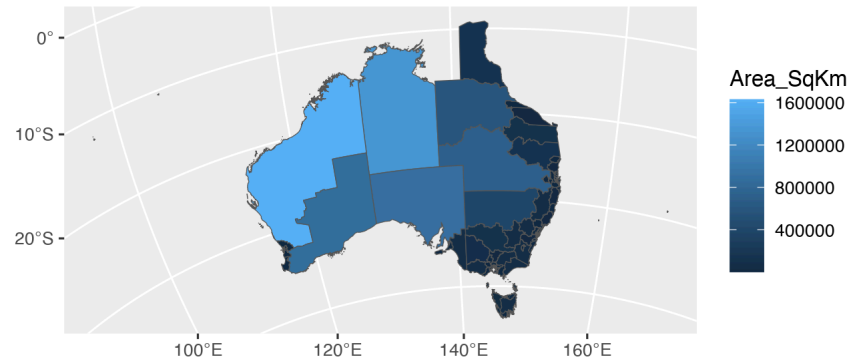
## Coordinate Reference System:
##   User input: GDA94
##   wkt:
##   GEOGCRS["GDA94",
##     DATUM["Geocentric Datum of Australia 1994",
##       ELLIPSOID["GRS 1980",6378137,298.257222101,
##         LENGTHUNIT["metre",1]]],
##   ggplot(ausmap) +
##     geom_sf(aes(fill = Area_SqKm))
```



Changing map projections

- Map projections may be modified in multiple methods (it's beyond this unit to delve deep into this).
- Below uses the Lambert azimuthal equal-area projection centered on the longitude and latitude of (rough) Melbourne coordinates via [proj4string](#):

```
ausmap %>%  
  st_transform(crs = "+proj=laea +x_0=0 +y_0=0 +lon_0=145 +lat_0=-38") %>%  
  ggplot() +  
  geom_sf(aes(fill = Area_SqKm))
```





Summary

- We had a look at the 2022 federal election data
- We looked at visualising this data spatially in various ways
- We learnt more about mapping in R
- We looked at reprojecting geographic data into different coordinate reference systems

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



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Lecturer: *Kate Saunders*

Department of Econometrics and Business Statistics

✉ ETC5512.Clayton-x@monash.edu

📅 Week 5