Investigating size of Indigenous Australian Population and Australian Immunisation rates

Author : Prashant Jajoria

07 September 2020

Contents

Introduction

This report provides insights into the relationship between the distribution and age of Indigenous Australians, their relations and trends over time. It helps to answer various questions pertaining to the distribution of Indigenious Australians and the Immunisation of Australian population. The Dataset was cleaned and wrangled using R programming language. It uses charts like boxplots, line garphs and Motion charts wherever its deemed appropriate to help support the answer.

About the datasets

The two datasets of focus are obtained from the Australian Bureau of Statistics (ABS), having yearly data about estimated population of Indigenous Australians, grouped by indigenous regions, between 2016 to 2031. Second dataset having information of estimated resident population of Indigenous Australians, grouping by state or territory, between 2006 and 2031. The third dataset contains yearly data regarding the number of 1, 2 and 5 year-old Australian children fully or partially immunised in various Primary Health Network (PHN) areas.

Task A: Investigating the size of the Indigenous Australian Population

A1: Investigating the Distribution of Indigenous Australians

Using R to read, wrangle and analyse the data in Data1 Loading library Tidyverse, a collection of useful packages for Data Analysis. Tidyverse has some of the most versatile R packages: ggplot2, dplyr, tidyr, readr, purrr, and tibble, etc

```
library(tidyverse)
```

Tidyverse can be installed by using the following command.

```
install.packages("tidyverse")
```

Reading dataset IndigAusPopData_by_region (Data1) into R. We use the read_csv() function, passing the csv file name as the parameter and storing in variable data1.

```
data1 = read_csv("IndigAusPopData_byregion.csv")
```

Having a glimpse of the dataset. Using the glimpse() function which belong to tibble package of Tidyverse.

```
glimpse(data1)
```

```
## Rows: 8,288
## Columns: 7
## $ INDIGENOUS_REGION   <dbl> 801, 801, 801, 801, 801, 801, 801, 801, 801, 801, ~
## $ `Indigenous region` <chr> "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "~
## $ Age                 <chr> "Oct-14", "Oct-14", "Oct-14", "Oct-14", "Oct-14", ~
## $ TIME                <dbl> 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 20~
## $ Value               <dbl> 694, 696, 697, 722, 776, 792, 812, 857, 859, 841, ~
## $ `Projection series` <chr> "Series A", "Series A", "Series A", "Series A", "S~
## $ Frequency           <chr> "Annual", "Annual", "Annual", "Annual", "Annual", ~
```

Changing column names to upper case for consistency.

```
# Rename column where names is "Indigenous region"
names(data1)[names(data1) == "Indigenous region"] <- "Region"

# Rename column where names is "Projection series"
names(data1)[names(data1) == "Projection series"] <- "Projection_series"

# Changing column names to upper case for consistency
names(data1) = toupper(names(data1))
```

Group the data by Age. We use the group_by() function from dplyr package. Grouping by age to get the age groups in the data.

```
age_group = data1 %>% group_by(AGE) %>% count(AGE)
age_group["AGE"]
```

```
## # A tibble: 14 x 1
## # Groups:   AGE [14]
##    AGE
##    <chr>
##  1 0 - 4
##  2 15 - 19
##  3 20 - 24
##  4 25 - 29
##  5 30 - 34
##  6 35 - 39
##  7 40 - 44
##  8 45 - 49
##  9 5-Sep
## 10 50 - 54
## 11 55 - 59
## 12 60 - 64
## 13 65 and over
## 14 Oct-14
```

Problem encountered : Age range of '5 - 9' and '10 - 14' missing We can observe that age range of '5 - 9' and '10 - 14' are missing from the group. This shows a possible mistake during data capture.

Solution : The groups '5 Sep' and 'Oct-14' contain the data for age groups '5 - 9' and '10 - 14'. As the count of values in the two groups is 592, which is the count of values for other columns, confirms that it has values for the missing age range. During capturing of data the age range of '10 - 14' is mistakenly entered as 'Oct-14'. We can see that the last two numbers are 14 indicating its the data for age range for '10-14'. Similarly for age range '5-9', the value is not properly captured. So when this data is loaded in R the it shows value as '5-Sep'. This indicate that its the value for age group '5-9' as the first digit is 5. Replacing the data of Age column to proper values, i.e 'Oct-14' to age range '10-14' and '5 Sept' to '5-9'. We use the mutate() function from dplyr package along with replace() to change the Age column.

```
# Replacing the Age rows with '10-14'
data1 = data1 %>%
     mutate(AGE=replace(AGE, AGE=='Oct-14', '10-14')) %>%
     as.data.frame()
```

```
# Replacing the Age rows with '5-9'
data1 = data1 %>%
     mutate(AGE=replace(AGE, AGE=='5-Sep', '5-9')) %>%
     as.data.frame()
```

Checking for NA values before answering the questions. A good way to do this is using the is.na() function that comes from the base package. Calculating the sum of count of all the NAs tells us whether NAs are present in the dataset or not.

```
# Checking for NA
sum(is.na(data1))
```

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

As the sum of is.na() is False, the dataset is good to go for answering the questions.

Question a : What regions have the maximum and minimum total Indigenous populations in 2016 and 2031?

- **Filtering data of Indigenous populations for year 2016**.

Firstly, filtering th data to keep only the data for year 2016. To do this we use the filter() function with parameter TIME == 2016. Then we group the data of 2016 by region to have data for each region. Lastly we use the summarise() to get the total of all the regions and sort the data using arrange() based on TOTAL number of Indigenous population.

```
# using filter() to get the data for year 2016
data_2016 = filter(data1,TIME==2016)

# group the data for 2016 by Region
by_region_2016 = group_by(data_2016,REGION)

# summarise to get the total of Indigenous populations. Using arrange() to sort the data.
by_region_2016 = summarise(by_region_2016,
                                TOTAL = sum(VALUE) ) %>%
                                                    arrange(TOTAL)
by_region_2016
```

```
## # A tibble: 37 x 2
##    REGION                TOTAL
##    <chr>                 <dbl>
##  1 Port Lincoln - Ceduna  2738
##  2 Tennant Creek          4374
##  3 Kununurra              6040
##  4 West Kimberley         6053
##  5 Broome                 6057
##  6 Alice Springs          6807
##  7 Torres Strait          7403
##  8 Kalgoorlie             7510
##  9 ACT                    7513
## 10 Geraldton              8502
## # ... with 27 more rows
```

Name of Region with Minimum total Indigenous populations in year 2016: Using the filter() function to get the data for only the minimum row.

```
by_region_2016 %>% filter(TOTAL == min(TOTAL))
```

```
## # A tibble: 1 x 2
##   REGION                TOTAL
##   <chr>                 <dbl>
## 1 Port Lincoln - Ceduna  2738
```

Port Lincoln - Ceduna region from the state of South Australia has the lowest total Indigenous populations in year 2016.

Name of Region with Maximum total Indigenous populations in year 2016: Using the filter() function to get the data for only the maximum row.

```
by_region_2016 %>% filter(TOTAL == max(TOTAL))
```

```
## # A tibble: 1 x 2
##   REGION                       TOTAL
##   <chr>                        <dbl>
## 1 NSW Central and North Coast 85169
```

NSW Central and North Coast region from the state of New South Wales has the highest total Indigenous populations in year 2016.

- **Filtering data of Indigenous populations for year 2031**:

Carrying the same steps for wrangling data for year 2031. Firstly, filtering th data to keep only the data for year 2031 To do this we use the filter() function with parameter TIME == 2031. Then we group the data of 2031 by region to have data for each region. Lastly we use the summarise() to get the total of all the regions and sort the data using arrange() based on TOTAL number of Indigenous population.

```
# using filter() to get the data for year 2031
data_2031 = filter(data1,TIME==2031)

# group the data for 2031 by Region
by_region_2031 = group_by(data_2031,REGION)
by_region_2031 = summarise(by_region_2031,
                           TOTAL = sum(VALUE) ) %>%
                                                arrange(TOTAL)
by_region_2031
```

```
## # A tibble: 37 x 2
##    REGION                TOTAL
##    <chr>                 <dbl>
##  1 Port Lincoln - Ceduna  2881
##  2 Tennant Creek          4778
##  3 Kununurra              6932
##  4 Alice Springs          7548
##  5 West Kimberley         7789
##  6 Broome                 7909
```

```
##  7 Torres Strait          8378
##  8 Mount Isa              9518
##  9 Katherine             10105
## 10 Kalgoorlie            10179
## # ... with 27 more rows
```

Name of Region with Maximum total Indigenous populations in year 2031:

Using the filter() function to get the data for only the Maximum row.

```
by_region_2031 %>% filter(TOTAL == max(TOTAL))
```

```
## # A tibble: 1 x 2
##   REGION    TOTAL
##   <chr>     <dbl>
## 1 Brisbane 129835
```

Brisbane region from the state of Queensland has the highest total Indigenous populations in year 2031.

Name of Region with Minimum total Indigenous populations in year 2031:

Using the filter() function to get the data for only the Minimum row.

```
by_region_2031 %>% filter(TOTAL == min(TOTAL))
```

```
## # A tibble: 1 x 2
##   REGION                TOTAL
##   <chr>                 <dbl>
## 1 Port Lincoln - Ceduna  2881
```

Port Lincoln - Ceduna region from the state of South Australia has the lowest total Indigenous populations in year 2016.

Question b :What region/s have the maximum and minimum growth or decay rates of their total Indigenous population between 2016 and 2031?

As we already have the total of population for year 2016 and 2031, we join the two dataframe to calculate growth or decay between 2016 and 2031. Using the inner_join() function with parameter by = REGION, as both the dataframe have the same region names. Finally renaming the columns to proper names.

```
total_2016_2031 = inner_join(by_region_2016, by_region_2031, by = "REGION")

# renaming the columns
names(total_2016_2031)[names(total_2016_2031) == "TOTAL.x" ] = "TOTAL_2016"
names(total_2016_2031)[names(total_2016_2031) == "TOTAL.y" ] = "TOTAL_2031"

total_2016_2031
```

```
## # A tibble: 37 x 3
##    REGION                TOTAL_2016 TOTAL_2031
##    <chr>                      <dbl>      <dbl>
##  1 Port Lincoln - Ceduna       2738       2881
```

```
##  2 Tennant Creek               4374       4778
##  3 Kununurra                   6040       6932
##  4 West Kimberley              6053       7789
##  5 Broome                      6057       7909
##  6 Alice Springs               6807       7548
##  7 Torres Strait               7403       8378
##  8 Kalgoorlie                  7510      10179
##  9 ACT                         7513      11638
## 10 Geraldton                   8502      10267
## # ... with 27 more rows
```

Calculating the growth or decay rates between 2016 and 2031. Sorting the resultant dataframe by the percenatge decay/growth.

```
# create new column named RATE to store the percenatge decay/growth.
total_2016_2031['RATE'] = ( ( total_2016_2031["TOTAL_2031"] - total_2016_2031["TOTAL_2016"] ) /
                            total_2016_2031["TOTAL_2016"])*100
# sort the data
arrange(total_2016_2031,by=RATE)
```

```
## # A tibble: 37 x 4
##    REGION                TOTAL_2016 TOTAL_2031  RATE
##    <chr>                      <dbl>      <dbl> <dbl>
##  1 Katherine                  11063      10105 -8.66
##  2 Port Lincoln - Ceduna       2738       2881  5.22
##  3 Mount Isa                   9003       9518  5.72
##  4 Tennant Creek               4374       4778  9.24
##  5 North-Western NSW           9848      10854 10.2
##  6 Alice Springs               6807       7548 10.9
##  7 Torres Strait               7403       8378 13.2
##  8 Kununurra                   6040       6932 14.8
##  9 Port Augusta                9403      11171 18.8
## 10 Darwin                     18309      22044 20.4
## # ... with 27 more rows
```

Getting the minimum growth or decay rates of total Indigenous populations between 2016 and 2031.

```
# using filter() to get the minimum growth or decay rates
total_2016_2031 %>% filter(RATE == min(RATE))
```

```
## # A tibble: 1 x 4
##   REGION    TOTAL_2016 TOTAL_2031  RATE
##   <chr>          <dbl>      <dbl> <dbl>
## 1 Katherine      11063      10105 -8.66
```

Katherine region from the state of Northern Territory has a decay rates of 8.65% of total Indigenous populations between 2016 and 2031.

Getting the maximum growth or decay rates of total Indigenous populations between 2016 and 2031:

```
total_2016_2031 %>% filter(RATE == max(RATE))
```

```
## # A tibble: 1 x 4
##   REGION TOTAL_2016 TOTAL_2031  RATE
##   <chr>       <dbl>      <dbl> <dbl>
## 1 ACT          7513      11638  54.9
```

Australian Capital Territory has the maximum growth rate of 54.9% of total Indigenous populations between 2016 and 2031.

Question c : Plot and describe the growth or decay of the total Indigenous populations for the capitals of the 8 state/territories across all time periods.

Lets have a look at the regions from the given dataset

```
by_region = data1 %>% group_by(REGION) %>% count(REGION)
by_region["REGION"]
```

```
## # A tibble: 37 x 1
## # Groups:   REGION [37]
##    REGION
##    <chr>
##  1 ACT
##  2 Adelaide
##  3 Alice Springs
##  4 Apatula
##  5 Brisbane
##  6 Broome
##  7 Cairns - Atherton
##  8 Cape York
##  9 Darwin
## 10 Dubbo
## # ... with 27 more rows
```

We can observe that we do not have data for all the capitals of the 8 state/territories.

- For example, capital of Australian Capital Territory is Canberra, is not given in the dataset.
- For state of New South Wales, the data of Sydeney is only for Wollongong region.
- We have the data for Tasmania, not for Hobart the capital of the state.

Assumption : To answer the questions, we assume the data for Sydney, Canberra and Hobart to be the data of their state/regions.

Defining variable called capitals to store the names of the 8 state capitals of Australia.

```
capitals = list("ACT","Sydney - Wollongong", "Darwin", "Brisbane", "Adelaide",
                "Tasmania", "Melbourne", "Perth")
```

Filtering the data frame to keep only the data of the capital of the states using filter() function.

```
data1_capitals = filter(data1, REGION %in% capitals)
glimpse(data1_capitals)
```

```
## Rows: 1,792
## Columns: 7
## $ INDIGENOUS_REGION <dbl> 801, 801, 801, 801, 801, 801, 801, 801, 801, 801, 80~
## $ REGION            <chr> "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "AC~
## $ AGE               <chr> "10-14", "10-14", "10-14", "10-14", "10-14", "10-14"~
## $ TIME              <dbl> 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024~
## $ VALUE             <dbl> 694, 696, 697, 722, 776, 792, 812, 857, 859, 841, 83~
## $ PROJECTION_SERIES <chr> "Series A", "Series A", "Series A", "Series A", "Ser~
## $ FREQUENCY         <chr> "Annual", "Annual", "Annual", "Annual", "Annual", "A~
```

Aggregating the data of capitals to get the sum of Indeginious population over the years of a region. Sorting the data and having a glimpse of the result.

```
by_region_year = aggregate(data1_capitals$VALUE,
                           by=list(data1_capitals$REGION, data1_capitals$TIME), FUN=sum)

# renaming the columns
names(by_region_year)[names(by_region_year) == "Group.1" ] = "REGION"
names(by_region_year)[names(by_region_year) == "Group.2" ] = "TIME"
names(by_region_year)[names(by_region_year) == "x" ] = "VALUE"

# sorting the data by region name in ascending order
by_region_year <- by_region_year %>% arrange(REGION)

# using glimpse() to know about the data
glimpse(by_region_year)
```

```
## Rows: 128
## Columns: 3
## $ REGION <chr> "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", "ACT", ~
## $ TIME   <dbl> 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 202~
## $ VALUE  <dbl> 7513, 7730, 7956, 8189, 8431, 8680, 8936, 9201, 9474, 9757, 100~
```

** FUNCTION DEFINED ** : As we need the data of Growth/Decay of Indeginious population different state capitals, its a good approach to define a function to do this, as the steps are repetitive. Function save a lot of time and makes the task easy.

Parameters: by_region_year - Dataframe having information of all the capital region - Region name

The input to the function is the dataframe which has been grouped by region and year. Region for which we need the Growth/Decay of Indeginious population. We iterate throught the rows to get the change in indeginious population. Storing this change in a new vector called rate. Similarly storing the cor(responding year in a new vector named period. Capturing the region name in vector region. Combining the three vectors using column bind i.e. cbind() and returning the resultant data frame.

```
get_rate_growth_capitals = function(data = by_region_year, region){

  # Filter the dataframe for a particular region.
  data = filter(data, REGION == region)
```

```
  # defining empty vectors
  rate = period = region = c()

  # Looping over the filtered data of capital, to calculate the % change in population
  for(i in 1:nrow(data) )
  {
    # Run unitl we reach the last row in the dataframe
    if ( i != nrow(data) )
    {
      # storing the time period
      period[i] = paste(substr(data[i,"TIME"],3,4) , substr(data[i+1,"TIME"],3,4) , sep = "-")

      # calculating the % change in indeginious population
      rate[i] = round((( data[i+1,"VALUE"] - data[i,"VALUE"] ) / data[i,"VALUE"] ) * 100,3)

      # capturing the name of the region
      region[i] = data[i,"REGION"]
    }
  }

  # binding all the 3 vectors together - column bind
  df <- cbind(region,period,rate)

  # returning the data as a Dataframe
  return(data.frame(df))
}
```

Getting the data of % change in Indeginious population of Australian Capital Region (ACT):

```
  act = get_rate_growth_capitals(region="ACT")
```

Getting the data of % change in Indeginious population of Sydney (Wollongong):

```
  sydney = get_rate_growth_capitals(region="Sydney - Wollongong")
```

Getting the data of % change in Indeginious population of Darwin:

```
  darwin = get_rate_growth_capitals(region="Darwin")
```

Getting the data of % change in Indeginious population of Brisbane:

```
  brisbane = get_rate_growth_capitals(region="Brisbane")
```

Getting the data of % change in Indeginious population of Adelaide:

```
  adelaide = get_rate_growth_capitals(region="Adelaide")
```

Getting the data of % change in Indeginious population of Tasmania:

```
tasmania = get_rate_growth_capitals(region="Tasmania")
```

Getting the data of % change in Indeginious population of Melbourne :

```
victoria = get_rate_growth_capitals(region="Melbourne")
```

Getting the data of % change in Indeginious population of Perth:

```
perth = get_rate_growth_capitals(region="Perth")
```

Combining the data of all the State capitals to a single dataframe using row bind. Now we have the data of all the states in a Long format. This makes long format is good for visualizing the data later.

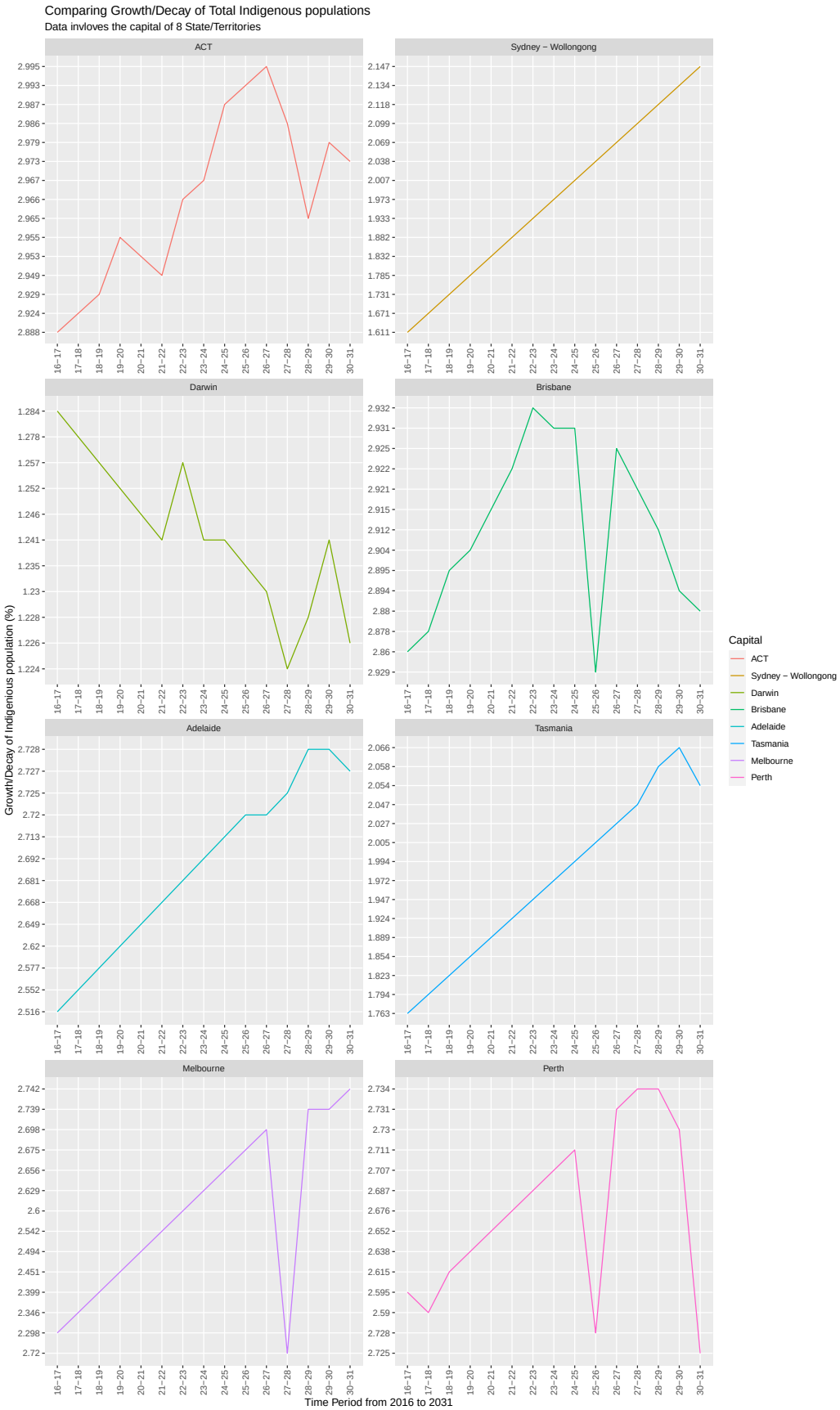
```
# row bind
rate_growth_decay = rbind(act,sydney,darwin,brisbane,adelaide,tasmania,victoria,perth)

# get the glimpse of the data
glimpse(rate_growth_decay)
```

```
## Rows: 120
## Columns: 3
## $ region <fct> ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT, ACT~
## $ period <fct> 16-17, 17-18, 18-19, 19-20, 20-21, 21-22, 22-23, 23-24, 24-25, ~
## $ rate   <fct> 2.888, 2.924, 2.929, 2.955, 2.953, 2.949, 2.966, 2.967, 2.987, ~
```

Making Line plot for each State capital over the years from 2016 to 2031. Using the Grammer of graphics to construct a faceted line plot. The color of each plot distinguishes the region. Keeping the x-axis at an angle for better readibility of the plot.

A line graph is appropriate for this king of analysis as it a Time series data. And change in the value over time can be best understood using a Line chart. Growth/Decay of Indigenious population (%) is a continuous variable making it suitable for y-axis in out plot. On the other hand year is discrete and can be used on the x-axis.

```
rate_growth_decay %>%
  ggplot( aes(x=period, y=rate, group=region, color=region)) +
  geom_line() +
  facet_wrap(~ region, scales = "free", ncol = 2) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(y="Growth/Decay of Indigenious population (%)",
  x = "Time Period from 2016 to 2031") +
  labs(title = "Comparing Growth/Decay of Total Indigenous populations",
       subtitle = "Data invloves the capital of 8 State/Territories ")+
  scale_color_discrete(name = "Capital") # rename the legend title
```


Comparing Growth/Decay of Total Indigenous populations
Data invloves the capital of 8 State/Territories
ACT
Sydney – Wollongong
Darwin
Brisbane
Adelaide
Tasmania
Melbourne
Perth
Growth/Decay of Indigenious population (%)
Time Period from 2016 to 2031
Capital
ACT
Sydney – Wollongong
Darwin
Brisbane
Adelaide
Tasmania
Melbourne
Perth

Findings/Insights

Trends of the Growth / Decay of the total Indigenous populations for the capitals of the 8 state/territories across all time periods.

- **ACT** : We can see that % change in Indeginious population of Australian Capital Region (ACT) rise gradually from the year 2016 up till 2026, having a slight drop in between in the years 2021-2022. It peaks during 2026 and 2027 having an percentage increase of 2.995%, and decreases later on.
- **Sydney - Wollongong** : The % change for Sydney - Wollongong region follows a linear relation having alomost a constant increase throughout the years.
- **Darwin** : For Darwin region the % change in population of Indeginious people is negative. It sees a gradual decrease overall, except for and increase of 1.257% and 1.241% during the years 2022-2023 and 2029-2030 respectively.
- **Brisbane** : Brisbane has a trend of increase in the population initially and then a gradual decrease from 2026-2027 during the end of the period.
- **Adelaide** : The indeginious population in Adelaide is expected to increse steadily in the given time period with peak increase of about 2.78% during year 2027-2028.
- **Tasmania** : Tasmania also has a similar increasing trend as Adelaide, with a peak increase of 2.066% during the years 2029-2030.
- **Melbourne** : Melbourne, capital of Victoria, also sees good increase in number of Indigenious population with increasing rate in the range of 2.6 to 2.9 and peak increase of 2.742% during 2030-31.
- **Perth** : The capital of Western Australia, Perth follows similar trend as Victoria, with numbers gradually increasing and reaching the maximum of 2.74% during 2027 and 2028.

A2 :Investigating the Ages of Indigenous Australians

Question 1. Which region has the highest percentage of children in its total 2016 population ?

Considering the given definition of a Child, the ABS commonly considers children to be under 15 years of age.

Using filter() for year 2016 and aggregating the data by region to get the sum of population for each region. Renaming the column for clarity.

```
# filtering the data to get values only for year 2016
data_2016 = data1 %>% filter(TIME == '2016')

# calculating the sum of values of a Region
by_region_2016 = aggregate(data_2016$VALUE, by = list(data_2016$REGION), FUN = sum)

# Rename column where names is "Group.1"
names(by_region_2016)[names(by_region_2016) == "Group.1"] <- "REGION"
names(by_region_2016)[names(by_region_2016) == "x"] <- "VALUE_TOTAL"

# have a glimpse
glimpse(by_region_2016)
```

```
## Rows: 37
## Columns: 2
## $ REGION      <chr> "ACT", "Adelaide", "Alice Springs", "Apatula", "Brisbane",~
## $ VALUE_TOTAL <dbl> 7513, 30124, 6807, 10191, 84454, 6057, 30050, 10579, 18309~
```

Filter the data for range 0 - 4, 5 - 9 and 10 - 14 to get the data of children for different regions. Summing up the data for each region using the aggregate() function on Value column to get data of children.

```
# filtering the data to get values only for year 2016 and age range
data_child_2016 = data1 %>% filter(TIME == '2016', AGE %in%  c('0-4','5-9','10-14') )

# group the data for region
by_region_child_2016 = aggregate(data_child_2016$VALUE, by = list(data_child_2016$REGION), FUN = sum)

# Rename column where names is "Group.1"
names(by_region_child_2016)[names(by_region_child_2016) == "Group.1"] <- "REGION"
names(by_region_child_2016)[names(by_region_child_2016) == "x"] <- "VALUE_CHILD"

# have a glimpse
glimpse(by_region_child_2016)
```

```
## Rows: 37
## Columns: 2
## $ REGION      <chr> "ACT", "Adelaide", "Alice Springs", "Apatula", "Brisbane",~
## $ VALUE_CHILD <dbl> 1490, 6968, 1294, 1818, 19512, 1322, 7066, 2263, 3747, 353~
```

Combine the data of total population and child population for all the regions using column bind.

```
# column bind the data of child and total population.
agg_df_child_2016 = cbind(by_region_2016,by_region_child_2016)

# select columns that are needed.
agg_df_child_2016 = subset(agg_df_child_2016,
                           select = c("REGION","VALUE_CHILD","VALUE_TOTAL"))
```

Calculating the percentage of child population out of the total population for all regions and storing in a new column called "PERCENTAGE_OF_CHILDREN"

```
agg_df_child_2016["PERCENTAGE_OF_CHILDREN"] = (agg_df_child_2016["VALUE_CHILD"] /
                                                 agg_df_child_2016["VALUE_TOTAL"]) * 100
```

Getting the region with highest percentage of child population out of the total population. For this we use the filter() function along with max() on the PERCENTAGE_OF_CHILDREN column.

```
max_agg_df_child_2016 = agg_df_child_2016 %>%
                        filter(PERCENTAGE_OF_CHILDREN == max(PERCENTAGE_OF_CHILDREN))
max_agg_df_child_2016
```

```
##             REGION VALUE_CHILD VALUE_TOTAL PERCENTAGE_OF_CHILDREN
## 1 Toowoomba - Roma        5596       21350               26.21077
```

Toowoomba - Roma region of Queensland state has the highest percentage of child population out of the total population in 2016.

Question 2. Calculate and discuss which state or territory has the highest percentage of children in its total 2006, 2016 and 2031 populations

Reading dataset IndigAusPopData_bystate (Data2) into R.

```
data2 = read_csv("IndigAusPopData_bystate.csv")
```

Having a glimpse of the dataset :

```
glimpse(data2)
```

```
## Rows: 528
## Columns: 28
## $ Age    <chr> "0", "0", "0", "0", "0", "0", "0", "0", "1", "1", "1", "1", "1"~
## $ State  <chr> "NSW", "Vic", "QLD", "SA", "WA", "Tas", "NT", "ACT", "NSW", "Vi~
## $ `2006` <dbl> 6024, 1255, 5324, 1007, 2391, 679, 1446, 158, 5691, 1156, 5183,~
## $ `2007` <dbl> 6243, 1361, 5566, 980, 2306, 692, 1462, 155, 6038, 1244, 5306, ~
## $ `2008` <dbl> 6147, 1394, 5619, 1023, 2362, 702, 1458, 148, 6257, 1351, 5548,~
## $ `2009` <dbl> 6161, 1375, 5608, 997, 2354, 723, 1521, 157, 6159, 1383, 5601, ~
## $ `2010` <dbl> 6182, 1371, 5583, 1053, 2257, 664, 1565, 170, 6173, 1364, 5590,~
## $ `2011` <dbl> 6240, 1333, 5522, 1049, 2264, 626, 1559, 155, 6193, 1360, 5565,~
## $ `2012` <dbl> 6348, 1323, 5552, 998, 2208, 660, 1501, 157, 6245, 1335, 5490, ~
## $ `2013` <dbl> 6376, 1467, 5514, 1037, 2272, 588, 1477, 187, 6353, 1326, 5520,~
## $ `2014` <dbl> 6212, 1409, 5413, 986, 2317, 595, 1443, 173, 6381, 1472, 5479, ~
```

```
## $ `2015` <dbl> 6423, 1375, 5500, 993, 2292, 621, 1463, 175, 6217, 1413, 5382, ~
## $ `2016` <dbl> 6597, 1480, 5295, 986, 2300, 644, 1395, 178, 6426, 1381, 5467, ~
## $ `2017` <dbl> 6466, 1431, 5432, 992, 2304, 623, 1423, 178, 6573, 1460, 5329, ~
## $ `2018` <dbl> 6639, 1483, 5561, 1011, 2345, 636, 1423, 184, 6444, 1414, 5463,~
## $ `2019` <dbl> 6805, 1538, 5689, 1031, 2379, 651, 1418, 190, 6616, 1465, 5594,~
## $ `2020` <dbl> 6974, 1591, 5821, 1054, 2418, 666, 1413, 198, 6781, 1519, 5723,~
## $ `2021` <dbl> 7142, 1646, 5961, 1075, 2461, 681, 1407, 203, 6950, 1570, 5856,~
## $ `2022` <dbl> 7309, 1699, 6108, 1094, 2504, 696, 1401, 209, 7119, 1624, 5996,~
## $ `2023` <dbl> 7466, 1755, 6248, 1113, 2542, 710, 1397, 216, 7284, 1677, 6144,~
## $ `2024` <dbl> 7614, 1807, 6380, 1131, 2578, 724, 1390, 223, 7441, 1732, 6284,~
## $ `2025` <dbl> 7756, 1852, 6507, 1151, 2610, 738, 1379, 230, 7589, 1782, 6417,~
## $ `2026` <dbl> 7895, 1898, 6629, 1171, 2640, 751, 1369, 238, 7732, 1827, 6545,~
## $ `2027` <dbl> 8043, 1946, 6760, 1189, 2671, 765, 1358, 243, 7870, 1872, 6669,~
## $ `2028` <dbl> 8179, 1996, 6889, 1206, 2701, 780, 1349, 249, 8017, 1920, 6800,~
## $ `2029` <dbl> 8302, 2042, 7004, 1224, 2727, 793, 1339, 255, 8152, 1969, 6929,~
## $ `2030` <dbl> 8410, 2086, 7110, 1238, 2750, 802, 1329, 261, 8277, 2014, 7048,~
## $ `2031` <dbl> 8517, 2122, 7205, 1254, 2767, 808, 1316, 268, 8386, 2058, 7152,~
```

FUNCTION DEFINED : As we need to calculate the state or territory that has the highest percentage of children in its total 2006, 2016 and 2031 populations, we can define a function to do this, as the steps are repetitive.

Defining a fucntion to get the maximum percentage of child population in a given year.

Parameter: year - year for which the highest percentage of children is needed.

Firstly, we get the information about the age, state and year from the dataset we read. Filter the data to include only the rows where the age is in the range of 0-14. Store this in a new dataframe. Perform aggreagtion by State, to get the sum of population of children in each state.

Now, we do the same step of aggregating to get the data for total population for given year. Aa the data is for the same regions, we can column bind the two dataframes into a new column. Subsetting this new dataframe and caluclating the % of child popluation for the given year. Using max function to get state with the highest % of child population and reurning this value.

```
get_max_child_pop_percentage <- function(year){

  # select Age, State and year column for the read csv dataset
  data2_year = data2[c("Age","State", year)]

  # filter to keep the rows of childdren
  data_age_year = data2_year %>% filter(Age %in%  as.list(0:14) )

  # sum up the data to get the total child population for each state
  by_child_age_year = aggregate(data_age_year[,3], by = list(data_age_year$State), FUN = sum)

  # Rename column where names is "Group.1"
  names(by_child_age_year)[names(by_child_age_year) == "Group.1"] <- "REGION"
  names(by_child_age_year)[names(by_child_age_year) == year] <- "VALUE_CHILD"

  # sum up the data to get the total population for each state
  by_region_data_year = aggregate(data2_year[,3], by = list(data2_year$State), FUN = sum)

  # Rename column where names is "Group.1"
  names(by_region_data_year)[names(by_region_data_year) == "Group.1"] <- "REGION"
  names(by_region_data_year)[names(by_region_data_year) == year] <- "VALUE_TOTAL"
```

```
  # column bind the total and child population for each state
  agg_data2_child = cbind(by_child_age_year,by_region_data_year)

  # Add year column to the new df
  agg_data2_child["YEAR"] = year

  # subset to select the needed columns
  agg_data2_child = subset(agg_data2_child,
                           select = c("REGION","YEAR","VALUE_CHILD","VALUE_TOTAL"))

  # calculate the % child population
  agg_data2_child["PERCENTAGE_OF_CHILDREN"] = (agg_data2_child["VALUE_CHILD"] /
                                                 agg_data2_child["VALUE_TOTAL"]) * 100

  # get the max child population
  max_agg_df_child_year = agg_data2_child %>%
                          filter(PERCENTAGE_OF_CHILDREN == max(PERCENTAGE_OF_CHILDREN))
  # return the max child population
  return(max_agg_df_child_year)
}
```

State or territory that has the highest percentage of children in its total 2006

Making a function call to get_max_child_pop_percentage() with year = 2006 to get the region with maximum percentage of child population.

```
get_max_child_pop_percentage(year=2006)
```

```
##   REGION YEAR VALUE_CHILD VALUE_TOTAL PERCENTAGE_OF_CHILDREN
## 1    QLD 2006       69861      175267               39.85976
```

Queensland with 39.85% has the heighest % of child population in 2006.

State or territory that has the highest percentage of children in its total 2016

Making a function call to get_max_child_pop_percentage() with year = 2016 to get the region with maximum percentage of child population.

```
get_max_child_pop_percentage(year=2016)
```

```
##   REGION YEAR VALUE_CHILD VALUE_TOTAL PERCENTAGE_OF_CHILDREN
## 1    QLD 2016       79410      221276               35.88731
```

Queensland with 35.88% has the heighest % of child population in 2016.

State or territory that has the highest percentage of children in its total 2031

Making a function call to get_max_child_pop_percentage() with year = 2031 to get the region with maximum percentage of child population.

```
get_max_child_pop_percentage(year=2031)
```

```
##   REGION YEAR VALUE_CHILD VALUE_TOTAL PERCENTAGE_OF_CHILDREN
## 1    QLD 2031       96730      304395               31.77779
```

Queensland with 31.77% has the heighest % of child population in 2031.

Question 3. Motion Chart comparing the total Indigenous Australian population of each region to the percentage of Indigenous Australian children in each state/territory

Wrangling the data for Motion chart

Firstly, we aggregate the data read from dataset 1 to get the total of population for each region and year. As the regions do not have a state named, we classify them to the stes they belong.

```
by_region_ind_pop = aggregate(data1$VALUE, by = list(data1$REGION,data1$TIME), FUN = sum)

# RENAME the columns
names(by_region_ind_pop)[names(by_region_ind_pop) == "Group.1"] <- "Region"
names(by_region_ind_pop)[names(by_region_ind_pop) == "Group.2"] <- "YEAR"
names(by_region_ind_pop)[names(by_region_ind_pop) == "x"] <- "Ind_Population"

#by_region_ind_pop
```

Making a list of regions for each state. This will be used to classfiy the data later.

```
# naming the regions of each state

act = c('ACT')
south_australia = c('Adelaide','Port Augusta','Port Lincoln - Ceduna')
north_terittory = c('Alice Springs','Apatula','Darwin','Jabiru - Tiwi',
                    'Katherine','Nhulunbuy','Tennant Creek')
queensland = c('Brisbane','Cairns - Atherton', 'Cape York', 'Mount Isa',
               'Rockhampton', 'Toowoomba - Roma', 'Torres Strait', 'Townsville - Mackay')
western_australia = c('Broome', 'Geraldton', 'Kalgoorlie', 'Kununurra', 'Perth',
                      'Riverina - Orange', 'South-Western WA', 'South Hedland', 'West Kimberley')
new_south_wales = c('Dubbo', 'North-Eastern NSW', 'North-Western NSW',
                    'NSW Central and North Coast', 'South-Eastern NSW', 'Sydney - Wollongong')
tasmania = c('Tasmania')
victoria = c('Melbourne', 'Victoria exc. Melbourne')
```

Using mutate() along with case_when() to add a state value to each region.

```
# add the respective state to the regions- This will be helpful to merge the dataframe later on

by_region_ind_pop_new = by_region_ind_pop %>%
                        mutate( STATE = case_when(
                                        by_region_ind_pop$Region  %in% act ~ 'ACT',
                                        by_region_ind_pop$Region  %in% south_australia ~ 'SA',
                                        by_region_ind_pop$Region  %in% north_terittory ~ 'NT',
                                        by_region_ind_pop$Region  %in% queensland ~ 'QLD',
                                        by_region_ind_pop$Region  %in% western_australia ~ 'WA',
                                        by_region_ind_pop$Region  %in% new_south_wales ~ 'NSW',
                                        by_region_ind_pop$Region  %in% tasmania ~ 'TAS',
                                        by_region_ind_pop$Region  %in% south_australia ~ 'SA',
                                        by_region_ind_pop$Region  %in% victoria ~ 'VIC'
                                    )
                                )
```

The data read from dataset 2 is in wide format. This has to be converted into long format for Motion chart to work. Using gather() function from tidyverse package to convert the data to long format. Making new

column Year and value to store the corresponding year and population for each state. Finally converting the state name to upper case as we have to merge this with other dataset.

```
# Convert the data2 into a long form
data2_long <- data2 %>%
  gather(key = YEAR, value = POPULATION, -Age, -State) %>%
  # use select() to rearrange the columns
  select(YEAR, everything(), POPULATION)

# convert state name to upper case
data2_long$State = toupper(data2_long$State)
```

The dataset 1 do not have data for years before 2016, we remove the corresponding rows from our consideration of dataset 2.

```
# filter data2 before 2016 -  as we do not have that in data 1

years = as.list(2016:2031)

data2_long = filter(data2_long, YEAR %in% years)
#data2_long
```

Filtering our long form data to get the rows for only child poplation.

```
# get the data for child age range only
child_age_range = as.list(0:14)

data2_long1 = filter(data2_long, Age %in% child_age_range)
#data2_long1
```

Aggregting the dataframe of child population to get their sum for each year and state. This will be needed for calculating the percentage of child population.

```
# child population each state / year
by_year_state_child = aggregate( data2_long1$POPULATION ,
                                 by = list( data2_long1$YEAR , data2_long1$State ) , FUN = sum)

# RENAME
names(by_year_state_child)[names(by_year_state_child) == "Group.1"] <- "Year"
names(by_year_state_child)[names(by_year_state_child) == "Group.2"] <- "Region"
names(by_year_state_child)[names(by_year_state_child) == "x"] <- "Child_Population"

#by_year_state_child
```

Aggregting the dataframe of total population to get their sum for each year and state. This will be needed for calculating the percentage of child population.

```
# child population each state / year
by_year_state_all_pop = aggregate( data2_long$POPULATION ,
                                   by = list( data2_long$YEAR , data2_long$State ) , FUN = sum)

# RENAME
```

```
names(by_year_state_all_pop)[names(by_year_state_all_pop) == "Group.1"] <- "YEAR"
names(by_year_state_all_pop)[names(by_year_state_all_pop) == "Group.2"] <- "STATE"
names(by_year_state_all_pop)[names(by_year_state_all_pop) == "x"] <- "Total_Population"

#by_year_state_all_pop
```

As the two data frames are for same states and years, we can column bind to get a new data frame. This will be useful for calculating the percentage of child population for each state and each year.

```
  # binding the 2 dataframe
  df_per_child_pop <- cbind(by_year_state_child,by_year_state_all_pop)

  df_per_child_pop$CHILD_POPULATION_PERCENTAGE = (df_per_child_pop$Child_Population /
                                                    df_per_child_pop$Total_Population) * 100


  df_per_child_pop = subset( df_per_child_pop , select =
                               c( 'YEAR' , 'STATE' , 'CHILD_POPULATION_PERCENTAGE') )
  df_per_child_pop$YEAR = as.numeric(df_per_child_pop$YEAR)
  #df22
```

Performing inner join between the two dataframes having information from dataset 1 and dataset 2. Inner join can be performed as the state name and the year are common between the two dataframes.

```
joined_df = inner_join(by_region_ind_pop_new, df_per_child_pop, by = c("YEAR","STATE"))
#joined_df
```

Using the newly formed dataframe to plot a Motion chart. In this the x-axis is the total population and y-axis is the % of child population. The size of bubble is the the Total population for a year. Reporting Year is used as time variable.

Motion chart is a best fit for this kind of analysis as the change in values can be identified easily. Also in motion chart we have used 4 different aspects of the data in the same plot. The total population for bubble size, state for colour of regions, year for time variation, % of child population on the y-axis and Total population for x-axis. Adding all the 4 elements for any other graph is not appropriate and is not easy to comprefend the data. Motion chart thus comes to the rescue by allowing to use these 4 elements together.

```
# load the googleVis library
library('googleVis')

# make theb plot
Md <- gvisMotionChart(joined_df, idvar='Region', timevar="YEAR",
                      xvar = 'Ind_Population', yvar = 'CHILD_POPULATION_PERCENTAGE', sizevar = 'Ind_Popu
# display the plot
plot(Md)
```

Question 4.a. Which region's population overtakes that of another region in the same state/territory? In which year/s does this happen?

The population of region of Apatula overtakes the region of Katherine, both belonging to Northern Territory state. This happens during the years 2024 and 2031.

Question 4.b. Is there generally a relationship between the Indigenous Australian population size and percentage of children in the population? If so, what kind of relationship? Explain your answer.

As from the motion chart, we can observe an inverse relationship between Indigenous Australian population size and percentage of children in the population. As the Indigenous Australian population size increases the percentage of children in the population decreases. For example the Indigenous Australian population size for Perth was 38,919 and the percentage child poulation was 33.4% in 2016. This decreased and the percentage of children population was 28.4% in 2031.

Question 4.c. Colour is commonly used in data visualisation to help understand data. Which aspect of this data would you use colour for in your plot and why?

Color adds to the visual element of a grahical representation. So choosing a good group for colour can help interpret the data. For the Motion chart I used the state. As colouring the regions belonging to same state help understand better the movement for regions in each state.

Question 4.d. Are there any other interesting things you notice in the data or any changes you would recommend for the Motion Chart?

One intersting thing that I notice is that the population of Brisbane overtake the population of NSW Central and North Coast during the years 2016 to 2031. The Motion chart can be improved to have a dropdown to select the Age range. And this Age range could be lotted againt the total population in the Motion chart.

Task B: Exploratory Analysis on Australian Immunisation rates

B1. Values and Variables

Using R to read, wrangle and analyse the data in Data3 Loading library Tidyverse, a collection of useful packages for Data Analysis. Tidyverse has some of the most versatile R packages: ggplot2, dplyr, tidyr, readr, purrr, and tibble, etc

```
library(tidyverse)
```

Tidyverse can be installed by using the following command.

```
install.packages("tidyverse")
```

Reading dataset AusImmunisationData (Data3) into R using the read_csv() method.

```
data3 = read_csv("AusImmunisationData.csv")
```

Having a look at the dataset uding the head() to see the top rows:

```
head(data3)
```

```
## # A tibble: 6 x 16
##   State `PHN code` `PHN area name` `Reporting Year` `Age group` `Number of regi~
##   <chr> <chr>      <chr>           <chr>            <chr>                   <dbl>
```

```
## 1 ACT   PHN801     Australian Cap~ 2015-16          2 years                  5679
## 2 ACT   PHN801     Australian Cap~ 2014-15          2 years                  5525
## 3 ACT   PHN801     Australian Cap~ 2016-17          2 years                  5761
## 4 ACT   PHN801     Australian Cap~ 2012-13          1 year                   5381
## 5 ACT   PHN801     Australian Cap~ 2013-14          1 year                   5513
## 6 ACT   PHN801     Australian Cap~ 2012-13          2 years                  5318
## # ... with 10 more variables: 'Number fully immunised' <dbl>, 'Number not fully
## #   immunised' <dbl>, 'Number of registered IndigAus children' <chr>, 'Number
## #   IndigAus fully immunised' <chr>, 'Number IndigAus not fully
## #   immunised' <chr>, 'Interpret with caution (#)' <chr>, X13 <lgl>, X14 <lgl>,
## #   X15 <lgl>, X16 <lgl>
```

Problem encountered : Extra columns named X13, X14, X15, X16 in the dataset when read in R.

```
names(data3)
```

```
##  [1] "State"
##  [2] "PHN code"
##  [3] "PHN area name"
##  [4] "Reporting Year"
##  [5] "Age group"
##  [6] "Number of registered children"
##  [7] "Number fully immunised"
##  [8] "Number not fully immunised"
##  [9] "Number of registered IndigAus  children"
## [10] "Number IndigAus fully immunised"
## [11] "Number IndigAus not fully immunised"
## [12] "Interpret with caution (#)"
## [13] "X13"
## [14] "X14"
## [15] "X15"
## [16] "X16"
```

Solution : Subset the data to not include these 4 columns using select() function as they do not contain any data.

```
data3 = data3 %>% select("State":"Interpret with caution (#)")
```

Question 1. How many PHN areas does the data cover?

Counting distinct PHN Areas using n_distinct() function for column named 'PHN area name'. This returns list of unique PHN codes.

```
# Counting distinct PHN Areas using n_distinct() function
data3 %>% summarise(PHN_AREAS_COUNT = n_distinct(data3[3]))
```

```
## # A tibble: 1 x 1
##   PHN_AREAS_COUNT
##             <int>
## 1              31
```

The dataset covers 31 Primary Health Network(PHN) areas.

Question 2. What are the possible values for 'PHN code'?

Displaying the different PHN code using unique() function on "PHN code" column. Converting the data to a dataframe using as.dta.frame()

```
# Counting distinct PHN code using unique()
phn_code  = data3 %>% select("PHN code")
phn_code = unique(phn_code)
as.data.frame(phn_code)
```

```
##    PHN code
## 1    PHN801
## 2    PHN103
## 3    PHN107
## 4    PHN106
## 5    PHN101
## 6    PHN109
## 7    PHN105
## 8    PHN110
## 9    PHN104
## 10   PHN108
## 11   PHN102
## 12   PHN701
## 13   PHN306
## 14   PHN305
## 15   PHN302
## 16   PHN307
## 17   PHN304
## 18   PHN303
## 19   PHN301
## 20   PHN401
## 21   PHN402
## 22   PHN601
## 23   PHN202
## 24   PHN204
## 25   PHN201
## 26   PHN203
## 27   PHN206
## 28   PHN205
## 29   PHN501
## 30   PHN502
## 31   PHN503
```

The datset covers the following 31 PHN codes : PHN801, PHN103, PHN107, PHN106, PHN101, PHN109, PHN105, PHN110, PHN104, PHN108, PHN102, PHN701, PHN306, PHN305, PHN302, PHN307, PHN304, PHN303, PHN301

Question 3. Calculate the percentage of Australian children that are fully immunised.

The following definition of Immunisation rate will be used for further calculations and answering the questions.

Immunisation rate =
$$\frac{\text{Number of Australians Immunises}}{\text{Total number of Australian population}} * 100$$

Calculating the Immunisation Rate of Australian Children by dividing the no. of children fully immunised divided by the total number of children. Storing the result of Immunisation Rate of Australian Children in a new column in data3 dataframe.

```
data3["Immunisation Rate non IndigAus"] = ( data3["Number fully immunised"] /
                                            data3["Number of registered children"] ) * 100

head(data3)
```

```
## # A tibble: 6 x 13
##   State `PHN code` `PHN area name` `Reporting Year` `Age group` `Number of regi~
##   <chr> <chr>      <chr>           <chr>            <chr>                   <dbl>
## 1 ACT   PHN801     Australian Cap~ 2015-16          2 years                  5679
## 2 ACT   PHN801     Australian Cap~ 2014-15          2 years                  5525
## 3 ACT   PHN801     Australian Cap~ 2016-17          2 years                  5761
## 4 ACT   PHN801     Australian Cap~ 2012-13          1 year                   5381
## 5 ACT   PHN801     Australian Cap~ 2013-14          1 year                   5513
## 6 ACT   PHN801     Australian Cap~ 2012-13          2 years                  5318
## # ... with 7 more variables: `Number fully immunised` <dbl>, `Number not fully
## #   immunised` <dbl>, `Number of registered IndigAus children` <chr>, `Number
## #   IndigAus fully immunised` <chr>, `Number IndigAus not fully
## #   immunised` <chr>, `Interpret with caution (#)` <chr>, `Immunisation Rate
## #   non IndigAus` <dbl>
```

Question Calculating the average Immunisation Rate of Australian children.

Using the mean() function on the column "Immunisation Rate non IndigAus" to get the average Immunisation Rate of Australian children. As the columns under consideration do not have any NA values we do not use the na.rm paramater for mean().

```
mean(data3[["Immunisation Rate non IndigAus"]])
```

```
## [1] 92.06154
```

The average Immunisation Rate of Australian children is 92.06154%

Question Calculating the maximum Immunisation Rate of Australian children

Using the max() function on the column "Immunisation Rate non IndigAus" to get the maximum Immunisation Rate of Australian children. As the columns under consideration do not have any NA values we do not use the na.rm paramater for max().

```
max(data3[["Immunisation Rate non IndigAus"]])
```

```
## [1] 96.15255
```

The maximum Immunisation Rate of Australian children is 96.15255%

Question Calculating the minimum Immunisation Rate of Australian children. Using the min() function on the column "Immunisation Rate non IndigAus" to get the minimum Immunisation Rate of Australian children.As the columns under consideration do not have any NA values we do not use the na.rm paramater for min().

```
min(data3[["Immunisation Rate non IndigAus"]])
```

```
## [1] 86.09929
```

The minimum Immunisation Rate of Australian children is 86.09929%

Question 3: Calculate the percentage of Indigenous Australian children that are fully immunised.

Carrying out similar steps for Indigenous Australian children.

Preparing the data of Immunisation of Australian and Indeginious children.

Problem encountered : The column name 'Number of registered IndigAus children' and 'Number IndigAus fully immunised' have some rows with values 'NP'. These 'NP' values indicate that the region has no population of Indeginious children.

Solution : Using mutate() with ifelse() to conditionally replace the rows having NP with 0. And making a new column. Renaming the column names back to get the original form of dataframe.

```
# create a vector of column names
cols_ind = c("Number of registered IndigAus  children","Number IndigAus fully immunised")

# replacing the rows with 'NP' values with 0.
data3 = data3 %>%
  mutate( total = ifelse(data3[["Number of registered IndigAus  children"]] == 'NP',
                         0 , data3[["Number of registered IndigAus  children"]]),
          val = ifelse(data3[["Number IndigAus fully immunised"]] == 'NP',
                         0 , data3[["Number IndigAus fully immunised"]]) )

# selecting only the required columns.
# Excluding the old columns "Number of registered IndigAus  children" and "Number IndigAus fully immuni
data3 = data3 %>% select(-one_of(cols_ind))

# Rename the columns
names(data3)[names(data3) == "total"] <- "Number of registered IndigAus  children"
names(data3)[names(data3) == "val"] <- "Number IndigAus fully immunised"

head(data3)
```

```
## # A tibble: 6 x 13
##   State `PHN code` `PHN area name` `Reporting Year` `Age group` `Number of regi~
##   <chr> <chr>      <chr>           <chr>            <chr>                   <dbl>
## 1 ACT   PHN801     Australian Cap~ 2015-16          2 years                  5679
## 2 ACT   PHN801     Australian Cap~ 2014-15          2 years                  5525
## 3 ACT   PHN801     Australian Cap~ 2016-17          2 years                  5761
## 4 ACT   PHN801     Australian Cap~ 2012-13          1 year                   5381
## 5 ACT   PHN801     Australian Cap~ 2013-14          1 year                   5513
## 6 ACT   PHN801     Australian Cap~ 2012-13          2 years                  5318
## # ... with 7 more variables: `Number fully immunised` <dbl>, `Number not fully
## #   immunised` <dbl>, `Number IndigAus not fully immunised` <chr>, `Interpret
## #   with caution (#)` <chr>, `Immunisation Rate non IndigAus` <dbl>, `Number of
## #   registered IndigAus children` <chr>, `Number IndigAus fully
## #   immunised` <chr>
```

Problem encountered : The column name 'Number of registered IndigAus children' and 'Number IndigAus fully immunised' have some rows with values that contains number that are seperated by comma (eg: 1,234). For these rows the as.numeric() fails. And thus we have to relace these (commas) with no space.

Solution : Using mutate() with ifelse() to conditionally replace the comma with blank space. And making a new columns. Renaming the column names back to get the original form of dataframe.

```
# replace commas(,) inside of numbers with no space. And store it in new column.
data3 = data3 %>%
        mutate(
          total=ifelse(nchar(data3[["Number of registered IndigAus  children"]]) == 5,
                       sub(",","",data3[["Number of registered IndigAus  children"]]) ,
                       data3[["Number of registered IndigAus  children"]]),

          val= ifelse(nchar(data3[["Number IndigAus fully immunised"]]) == 5,
                      sub(",","",data3[["Number IndigAus fully immunised"]]) ,
                      data3[["Number IndigAus fully immunised"]])
         )

# columns to not select
cols_ind = c("Number of registered IndigAus  children", "Number IndigAus fully immunised")

# Using select() to get the required columns.
data3 = data3 %>% select(-one_of(cols_ind))

# Rename column.
names(data3)[names(data3) == "total"] <- "Number of registered IndigAus  children"
names(data3)[names(data3) == "val"] <- "Number IndigAus fully immunised"

head(data3)
```

```
## # A tibble: 6 x 13
##   State `PHN code` `PHN area name` `Reporting Year` `Age group` `Number of regi~
##   <chr> <chr>      <chr>           <chr>            <chr>                   <dbl>
## 1 ACT   PHN801     Australian Cap~ 2015-16          2 years                  5679
## 2 ACT   PHN801     Australian Cap~ 2014-15          2 years                  5525
## 3 ACT   PHN801     Australian Cap~ 2016-17          2 years                  5761
## 4 ACT   PHN801     Australian Cap~ 2012-13          1 year                   5381
## 5 ACT   PHN801     Australian Cap~ 2013-14          1 year                   5513
## 6 ACT   PHN801     Australian Cap~ 2012-13          2 years                  5318
## # ... with 7 more variables: `Number fully immunised` <dbl>, `Number not fully
## #   immunised` <dbl>, `Number IndigAus not fully immunised` <chr>, `Interpret
## #   with caution (#)` <chr>, `Immunisation Rate non IndigAus` <dbl>, `Number of
## #   registered IndigAus children` <chr>, `Number IndigAus fully
## #   immunised` <chr>
```

Problem encountered : The columns 'Number of registered IndigAus children' and `Number IndigAus fully immunised` are a list of values. The list cannot be handled by as.numeric() function. And as.numeric() function accepts single elements.

Solution : So we unlist the elements using unlist() and use these values with as.numeric() to get the result. Storing the Immunisation rates of Indigenious Australians in an new column in our data frame.

```r
# Calculating the Immunisation Rate for Indeginious Australian children.
data3["Immunisation Rate IndigAus"] = ( as.numeric(unlist(data3[13])) /
                                        as.numeric(unlist(data3[12])) ) * 100
head(data3)
```

```
## # A tibble: 6 x 14
##   State `PHN code` `PHN area name` `Reporting Year` `Age group` `Number of regi~
##   <chr> <chr>      <chr>           <chr>            <chr>                   <dbl>
## 1 ACT   PHN801     Australian Cap~ 2015-16          2 years                  5679
## 2 ACT   PHN801     Australian Cap~ 2014-15          2 years                  5525
## 3 ACT   PHN801     Australian Cap~ 2016-17          2 years                  5761
## 4 ACT   PHN801     Australian Cap~ 2012-13          1 year                   5381
## 5 ACT   PHN801     Australian Cap~ 2013-14          1 year                   5513
## 6 ACT   PHN801     Australian Cap~ 2012-13          2 years                  5318
## # ... with 8 more variables: `Number fully immunised` <dbl>, `Number not fully
## #   immunised` <dbl>, `Number IndigAus not fully immunised` <chr>, `Interpret
## #   with caution (#)` <chr>, `Immunisation Rate non IndigAus` <dbl>, `Number of
## #   registered IndigAus children` <chr>, `Number IndigAus fully
## #   immunised` <chr>, `Immunisation Rate IndigAus` <dbl>
```

Question : Calculating the average Immunisation Rate of Indeginious Australian children. Using the mean() function on the column "Immunisation Rate IndigAus" to get the average Immunisation Rate of Indeginious children. As the columns under consideration do have any NA values we do not use the na.rm paramater for mean() to not includde them in the calculations. These NAs are due to division by 0.

```r
mean(data3[["Immunisation Rate IndigAus"]], na.rm = TRUE)
```

```
## [1] 89.92436
```

The avearge Immunisation Rate of Indeginious Australian children is 89.92436%

Question : Calculating the maximum Immunisation Rate of Indeginious Australian children Using the max() function on the column "Immunisation Rate IndigAus" to get the maximum Immunisation Rate of Indeginious children. As the columns under consideration do have any NA values we do not use the na.rm paramater for max() to not includde them in the calculations. These NAs are due to division by 0.

```r
max(data3[["Immunisation Rate IndigAus"]], na.rm = TRUE)
```

```
## [1] 97.68977
```

The maximum Immunisation Rate of Indeginious Australian children is 97.689%

Question : Calculating the minimum Immunisation Rate of Indeginious Australian children Using the min() function on the column "Immunisation Rate IndigAus" to get the minimum Immunisation Rate of Indeginious children. As the columns under consideration do have any NA values we do not use the na.rm paramater for min() to not includde them in the calculations. These NAs are due to division by 0.

```r
min(data3[["Immunisation Rate IndigAus"]] , na.rm = TRUE)
```

```
## [1] 73.77049
```

The minimum Immunisation Rate of Indeginious Australian children is 73.7704%

Using the names() function to get the column names of the dataframe. Renaming the columns before further analysis.

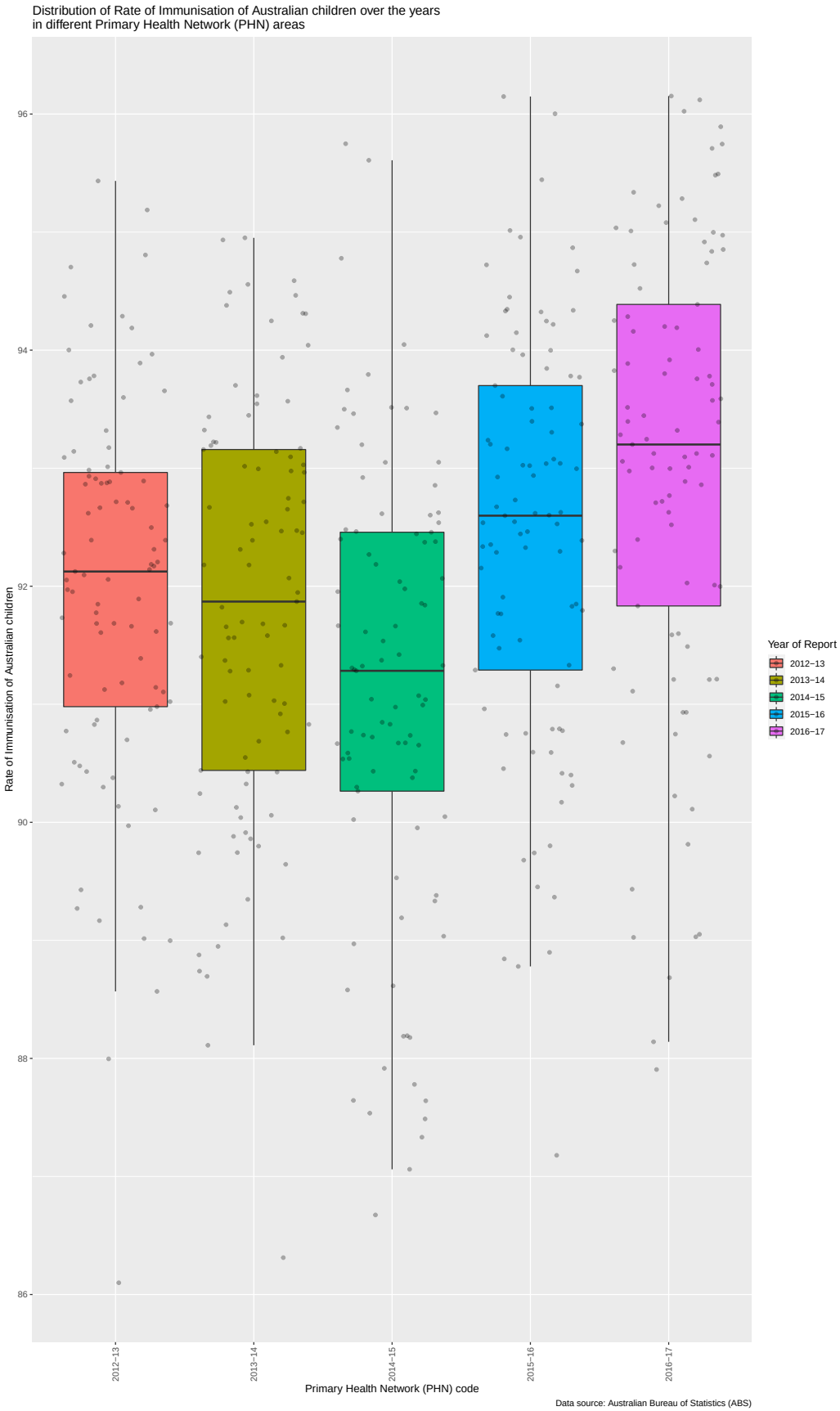
```
# Rename column before plotting the graphs
names(data3)[names(data3) == "PHN code"] <- "phn_code"
names(data3)[names(data3) == "PHN area name"] <- "phn_area_name"
names(data3)[names(data3) == "Immunisation Rate non IndigAus"] <- "immunisation_rate_non_IndigAus"
names(data3)[names(data3) == "Immunisation Rate IndigAus"] <- "immunisation_rate_IndigAus"
names(data3)[names(data3) == "Reporting Year"] <- "reporting_year"
names(data3)[names(data3) == "Age group"] <- "age_group"
```

B2. Variation in rates over Time, Age and Location

Question 1. Have the immunisation rates improved over time? Are the median immunisation rates increasing, decreasing or staying the same?

Using the data of immunisation rates from data3, we generate boxplots of different years. Boxplots are useful in this case as we can see the median of the immunisation rates over the years. Also it shows the Interquartile region i.e. extent of data covered. Here x-axis is the reporting year and y-axis is the rate of immunisation.

```
# set the width of figure of this code chunk
opts_current$set(fig.width=50)

# set the height of figure of this code chunk
opts_current$set(fig.height =100)

data3 %>%
  # fill the box by reporting year
  ggplot( aes(x = reporting_year ,  y = immunisation_rate_non_IndigAus, fill = reporting_year) ) +
  # hide the outliers
  geom_boxplot(outlier.alpha = 0) +
  # adding jitter to make the overlapping data visible
  geom_jitter(alpha = 0.3) +
  # rotate the x-axis by 90 degree
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  # add title and caption
  labs(title = "Distribution of Rate of Immunisation of Australian children over the years \nin differe
       caption = "Data source: Australian Bureau of Statistics (ABS)") +
  # add labels
  labs(y="Rate of Immunisation of Australian children",
       x = "Primary Health Network (PHN) code")+
  # rename the legend title
  scale_fill_discrete(name="Year of Report")
```

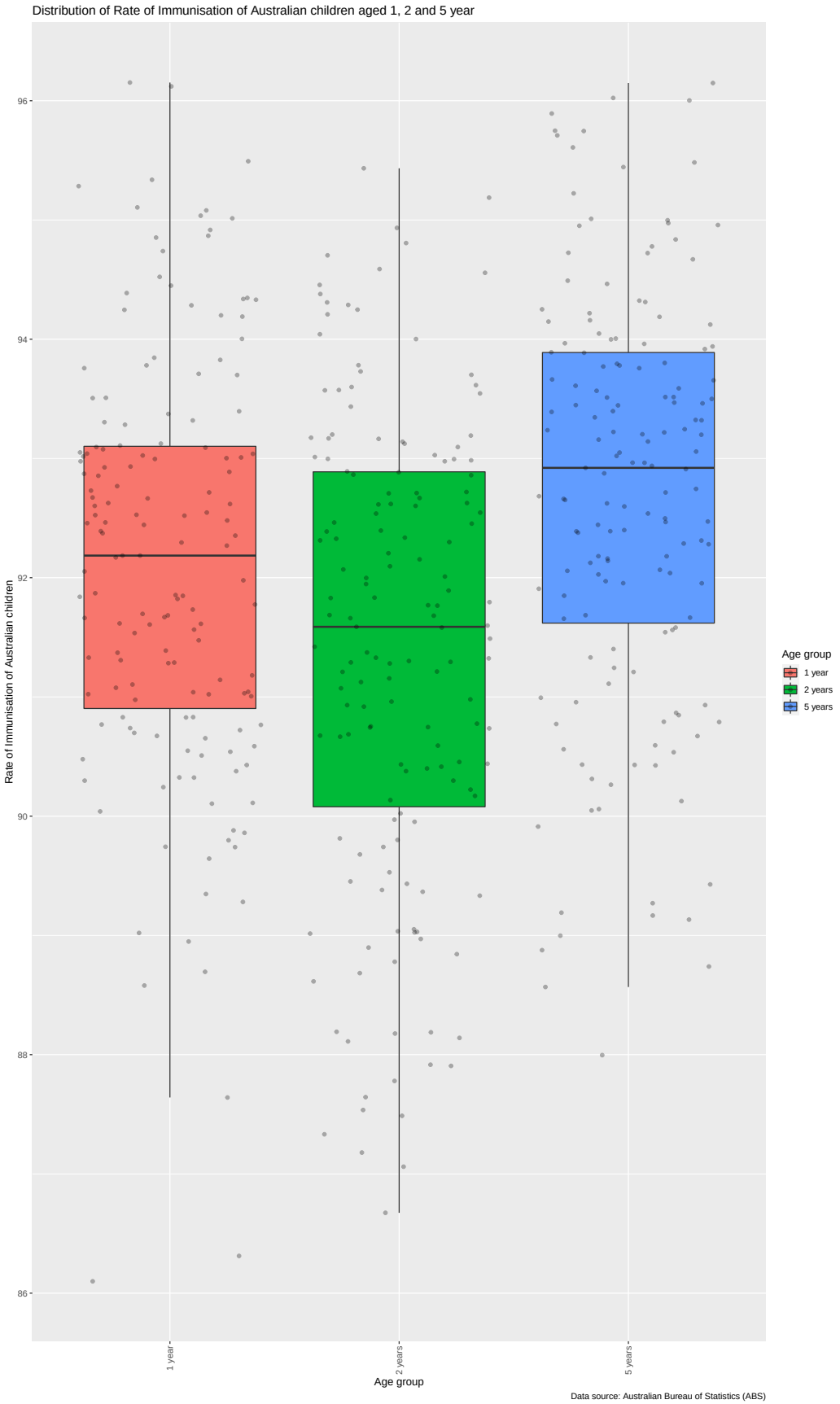

Distribution of Rate of Immunisation of Australian children over the years
in different Primary Health Network (PHN) areas
Rate of Immunisation of Australian children
96
94
92
90
88
86
2012–13
2013–14
2014–15
2015–16
2016–17
Primary Health Network (PHN) code
Year of Report
2012–13
2013–14
2014–15
2015–16
2016–17
Data source: Australian Bureau of Statistics (ABS)

Overall there is an increase in the Rate of Immunisation of Australian children over the years. The median Rate of Immunisation of Australian children decreased from year 2012-13 to 2014-15. It increased later from year 2014-2015 and finally in 2016-17 it was the highest with median rate around 93%.

Question 2. How do the immunisation rates vary with the age of the child?

Using the data of immunisation rates from data3, we generate boxplots of different age groups. Boxplots are useful in this case as we can see the median of the immunisation rates for different age groups. Also it shows the Interquartile region i.e. extent of data covered. Here the x-axis is the age group and y-axis is the rate of immunisation.

```
# set the width of figure of this code chunk
opts_current$set(fig.width=50)

# set the height of figure of this code chunk
opts_current$set(fig.height =100)

data3 %>%
  # fill the box by age group
  ggplot( aes(x = age_group ,  y = immunisation_rate_non_IndigAus, fill = age_group) ) +
  # hide the outliers
  geom_boxplot(outlier.alpha = 0) +
  # adding jitter to make the overlapping data visible
  geom_jitter(alpha = 0.3) +
  # rotate the x-axis by 90 degree
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  # adding title and caption
  labs(title = "Distribution of Rate of Immunisation of Australian children aged 1, 2 and 5 year",
       caption = "Data source: Australian Bureau of Statistics (ABS)") +
  # adding x and y labels to the figure
  labs(y="Rate of Immunisation of Australian children",
       x = "Age group")+
  # rename the legend title
  scale_fill_discrete(name="Age group")
```


Distribution of Rate of Immunisation of Australian children aged 1, 2 and 5 year
Rate of Immunisation of Australian children
96
94
92
90
88
86
1 year
2 years
5 years
Age group
Age group
1 year
2 years
5 years
Data source: Australian Bureau of Statistics (ABS)

We can see that the median rate of immunisation for age group of 2 years is the lowest with value of just under 92%. The median rate of immunisation for age group of 1 year is better than that of age group of 2 years. The rate of immunisation for 2 years old is just above 92%. The rate of immunisation for 5 years old is highest with value of around 93%.

Question 3. What is the median rate per state/territory?

Data Wrangling steps : Preparing the data in the correct format before using it for plot.

Making a list of regions for each State/territory. This will be used to classfiy the data later.

```
# naming the regions of each state
act = c('Australian Capital Territory')

south_australia = c('Adelaide','Country SA')

north_terittory = c('Northern Territory')

queensland =c('Brisbane North','Brisbane South','Central Queensland, Wide Bay and Sunshine Coast', 'Dar

western_australia = c('Country WA','Perth North','Perth South')

new_south_wales = c('Central and Eastern Sydney', 'Hunter New England and Central Coast','Murrumbidgee'

tasmania = c('Tasmania')

victoria = c('Eastern Melbourne', 'Gippsland','Murray','North Western Melbourne','South Eastern Melbour
```

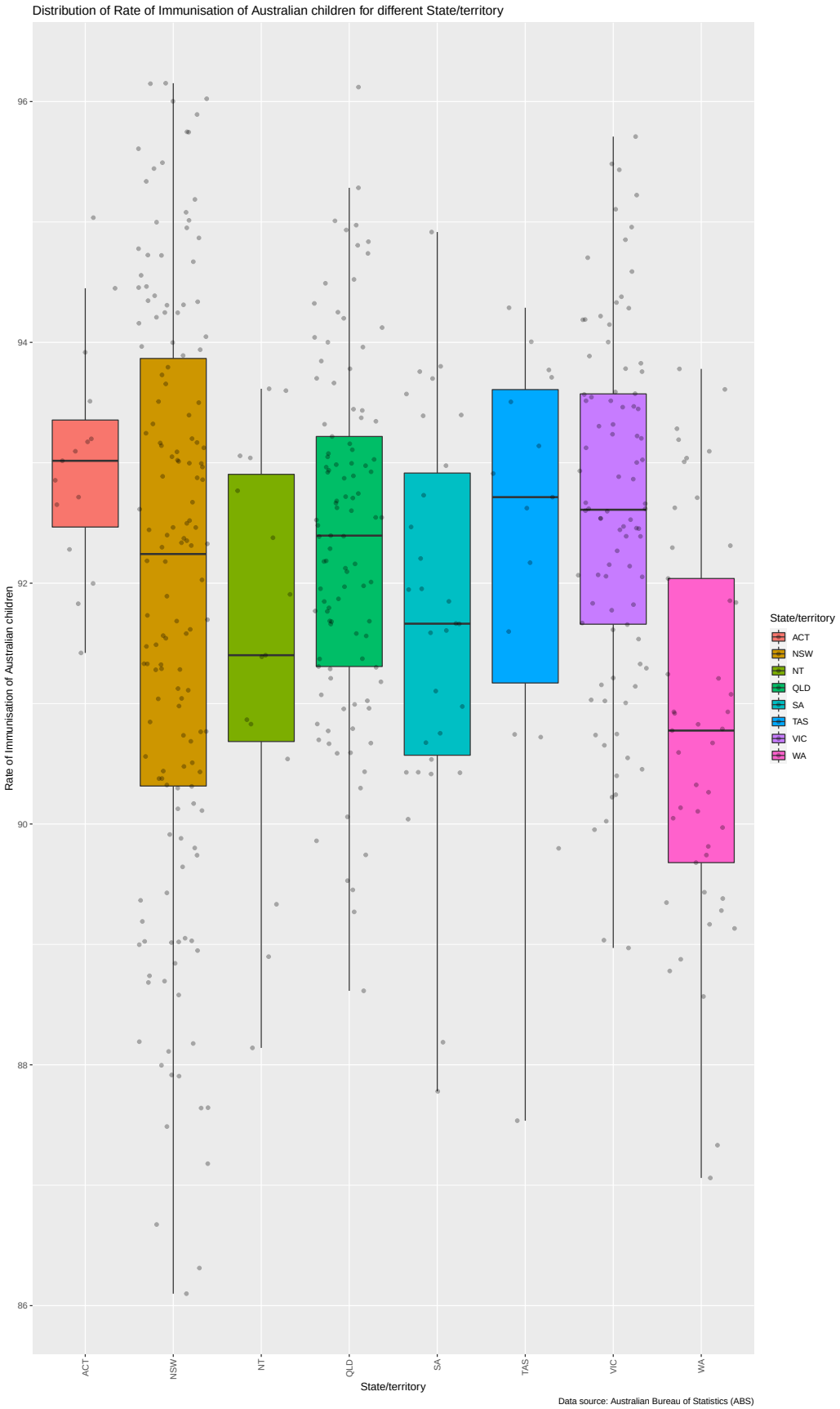
Using mutate() along with case_when() to add a state value to each region.

```
# add the respective state to the regions- This will be helpful to merge the dataframe later on

data3 = data3 %>%
                        mutate( STATE = case_when(
                                      data3$phn_area_name  %in% act ~ 'ACT',
                                      data3$phn_area_name  %in% south_australia ~ 'SA',
                                      data3$phn_area_name  %in% north_terittory ~ 'NT',
                                      data3$phn_area_name  %in% queensland ~ 'QLD',
                                      data3$phn_area_name  %in% western_australia ~ 'WA',
                                      data3$phn_area_name  %in% new_south_wales ~ 'NSW',
                                      data3$phn_area_name  %in% tasmania ~ 'TAS',
                                      data3$phn_area_name  %in% south_australia ~ 'SA',
                                      data3$phn_area_name  %in% victoria ~ 'VIC'
                                  )
                                )
```

Using the data of immunisation rates from data3, we generate boxplots of different State/territory. Boxplots are useful in this case as we can see the median of the immunisation rates for different State/territory. Also it shows the Interquartile region i.e. extent of data covered. Here the x-axis is the State/territory and y-axis is the rate of immunisation.

```
# set the width of figure of this code chunk
opts_current$set(fig.width=50)
```

```r
# set the height of figure of this code chunk
opts_current$set(fig.height =100)

data3 %>%
  # fill the box by state
  ggplot( aes(x = STATE ,  y = immunisation_rate_non_IndigAus, fill = STATE) ) +
  # hide the outliers
  geom_boxplot(outlier.alpha = 0) +
  # adding jitter to make the overlapping data visible
  geom_jitter(alpha = 0.3) +
  # rotate the x-axis by 90 degree
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  # addin title and caption for the figure
  labs(title = "Distribution of Rate of Immunisation of Australian children for different State/territo
       caption = "Data source: Australian Bureau of Statistics (ABS)") +
  # add x and y axis label to the figure
  labs(y="Rate of Immunisation of Australian children",
       x = "State/territory") +
  # rename the legend title
  scale_fill_discrete(name="State/territory")
```


Distribution of Rate of Immunisation of Australian children for different State/territory
Rate of Immunisation of Australian children
96
94
92
90
88
86
ACT
NSW
NT
QLD
SA
TAS
VIC
WA
State/territory
Data source: Australian Bureau of Statistics (ABS)
State/territory
ACT
NSW
NT
QLD
SA
TAS
VIC
WA

Median immunisation rates for various State/territory

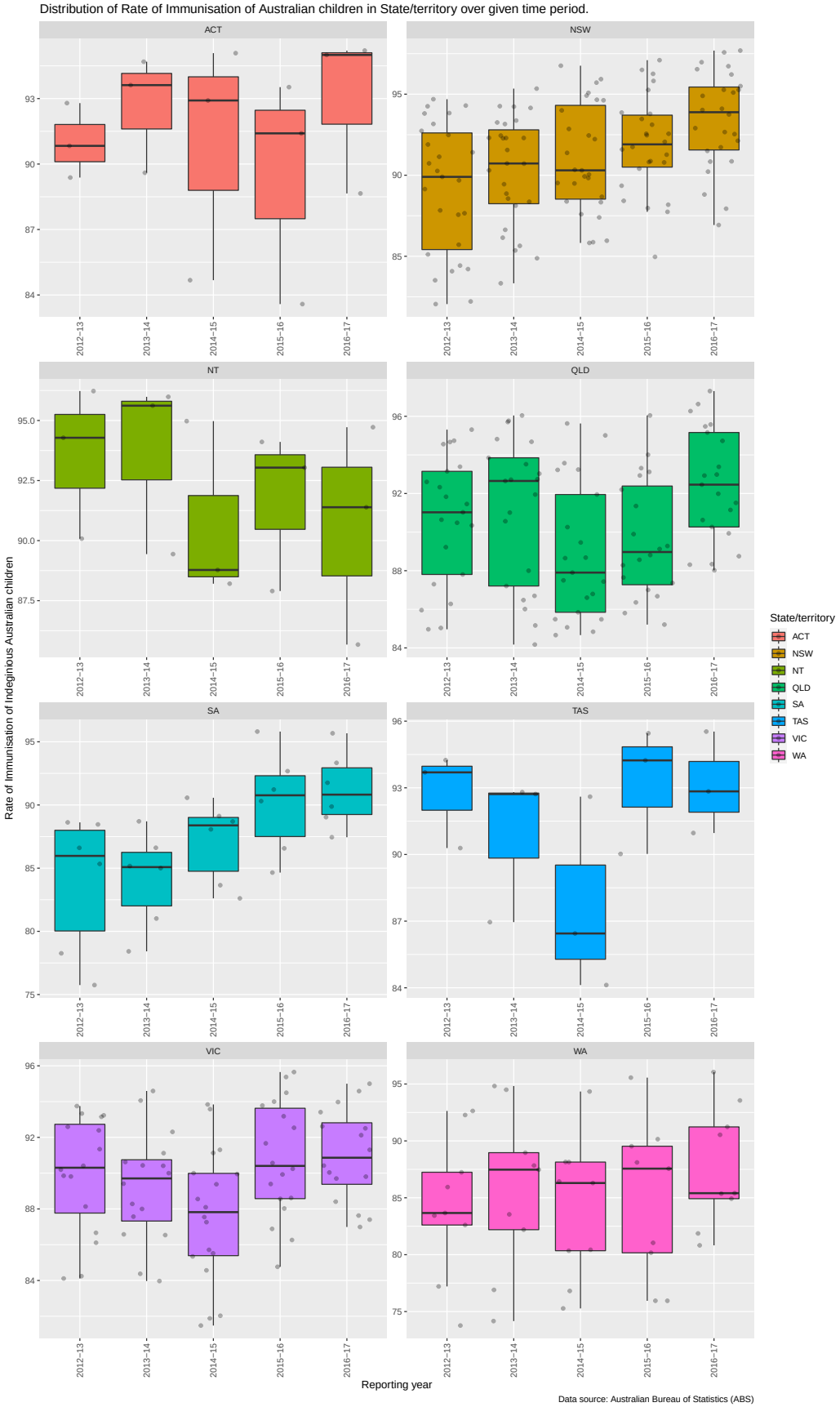
- **Australian Capital Region** : The medain rate of immunisation for ACT is just above 93%. This is the highest for any State/territory.
- **New South Wales** : The medain rate of immunisation for New South Wales is just around 92%
- **Northern territory** : The medain rate of immunisation for Northern territory is just above 91%
- **Queensland** : The medain rate of immunisation for Queensland is just above 92%
- **South Australia** : The medain rate of immunisation for South Australia is around 91.75%
- **Tasmania** : The medain rate of immunisation for Tasmania is just above 92.75%
- **Victoria** : The medain rate of immunisation for Victoria is around 92.75%
- **Westen Australia** : The medain rate of immunisation for Westen Australia is just above 91.75%. This is the lowest for any State/territory.

Question 4. Which states or territories seem most consistent in their immunisation rates?

Using the data of immunisation rates from data3, we generate boxplots of different State/territory over the years. Boxplots are useful in this case as we can see the median of the immunisation rates for different State/territory. Also it shows the Interquartile region i.e. extent of data covered. Here the x-axis is the reporting year and y-axis is the rate of immunisation. Facet is the State/territory.

The scale in this case is "free" as we have facet of state, and can be analysed independently.

```
# set the width of figure of this code chunk
opts_current$set(fig.width=50)
# set the height of figure of this code chunk
opts_current$set(fig.height =80)

data3 %>%
  # fill the box by state
  ggplot(aes(x = reporting_year ,  y = immunisation_rate_IndigAus, fill=STATE)) +
  # hide the outliers
  geom_boxplot(outlier.alpha = 0) +
  # adding jitter to make the overlapping data visible
  geom_jitter(alpha = 0.3) +
  # rotate the x-axis by 90 degree
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  # facaet the plot by State/territory
  facet_wrap(~ STATE, scales = "free", ncol=2) +
  # adding title and caption for the figure
  labs(title = "Distribution of Rate of Immunisation of Australian children in State/territory over giv
       caption = "Data source: Australian Bureau of Statistics (ABS)")+
  # add x and y axis label to the figure
  labs(y="Rate of Immunisation of Indeginious Australian children",
       x = "Reporting year") +
  # rename the legend title
  scale_fill_discrete(name="State/territory")
```


Distribution of Rate of Immunisation of Australian children in State/territory over given time period.
ACT
NSW
NT
QLD
SA
TAS
VIC
WA
Rate of Immunisation of Indeginious Australian children
Reporting year
2012–13
2013–14
2014–15
2015–16
2016–17
State/territory
Data source: Australian Bureau of Statistics (ABS)

The state of Northern territory and Queensland have been most consistent in their immunisation rates. We can conclude this from seeing the region of Interquartie region covered by the Boxplots. The size of Box plot is the population of that have been immunised. This is consistent for only these two states of Northern territory and Queensland over the years.

References

- Get a glimpse of your data — glimpse. (2020). Retrieved 13 September 2020, from https://tibble.tidyverse.org/reference/glimpse.html
- Group by one or more variables — group_by. (2020). Retrieved 13 September 2020, from https://dplyr.tidyverse.org/reference/group_by.html
- Dhana, K. (2020). How to Deal with Missing Values in R. Retrieved 13 September 2020, from https://datascienceplus.com/missing-values-in-r
- contributors, D. (2020). Data visualization with ggplot2. Retrieved 13 September 2020, from https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html