Assignement_1_33059047

Tamanna Das

2023-03-25

Introduction

This report is based on the dataset National Waste report 2022 Australia provided by the department of Climate change, Energy the Environment and water in Australia. There are two datasets provided which are wastes.csv and Year_State_ID.csv containing the waste types, their categorization, the state and year in which they were analyzed, the ultimate destination of the waste,and the quantity of waste generated for the wstes dataset and in the year_state_id we have data of particular years and states.

Installing packages and library

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
```

Loading the two files and reading the data of the files that is -

1.Wastes.csv

2.Year_State_Id.csv

```
wastes<-read_csv("wastes.csv")
```

```
## Rows: 36717 Columns: 8
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): Category, Type, Stream, Fate, Core_Non-core
## dbl (3): Case_ID, Year_State_ID, Tonnes
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Year_State_Id<-read_csv("Year_State_Id.csv")
```

```
## Rows: 130 Columns: 3
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): Year, State
## dbl (1): ID
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Question 1.

Reading the data wastes: We have already read the wastes data above here I am going to print the data to see how it looks.

Code:

```
print(wastes)
```

```
## # A tibble: 36,717 × 8
##    Case_ID Year_State_ID Category             Type  Stream Fate   Tonnes Core_…¹
##      <dbl>         <dbl> <chr>                <chr> <chr>  <chr>   <dbl> <chr>
##  1   34658          1001 Biosolids            Bios… C&I    Disp… 2.02e+4 Core w…
##  2   34659          1001 Biosolids            Bios… C&I    Recy… 0       Core w…
##  3   34662          1001 Building and demoli… Asph… C&D    Recy… 0       Core w…
##  4   34663          1001 Building and demoli… Asph… C&I    Recy… 0       Core w…
##  5   34664          1001 Building and demoli… Asph… MSW    Recy… 0       Core w…
##  6   34666          1001 Building and demoli… Bric… C&D    Recy… 1.42e+5 Core w…
##  7   34667          1001 Building and demoli… Bric… C&I    Recy… 3.83e+0 Core w…
##  8   34668          1001 Building and demoli… Bric… MSW    Recy… 0       Core w…
##  9   34670          1001 Building and demoli… Cera… C&D    Recy… 3.62e+4 Core w…
## 10   34671          1001 Building and demoli… Cera… C&I    Recy… 5.31e-1 Core w…
## # … with 36,707 more rows, and abbreviated variable name ¹`Core_Non-core`
## # ℹ Use `print(n = ...)` to see more rows
```

Finding the number of columns and rows in the dataset:

Code:

```
print(ncol(wastes))
```

```
## [1] 8
```

```
print(nrow(wastes))
```

```
## [1] 36717
```

Answer as seen above: The dataset contains 8 columns and 36717 rows as seen above

Explanation: To arrive at this answer we used the function ncol and nrow which is used to find the number of columns and rows respectively in a dataset.

Displaying the first 10 records and last 10 records of the dataset

Code:

```
first_10<-head(wastes,10)
print(first_10)
```

```
## # A tibble: 10 × 8
##    Case_ID Year_State_ID Category             Type  Stream Fate   Tonnes Core_…¹
##      <dbl>         <dbl> <chr>                <chr> <chr>  <chr>   <dbl> <chr>
##  1   34658          1001 Biosolids            Bios… C&I    Disp… 2.02e+4 Core w…
##  2   34659          1001 Biosolids            Bios… C&I    Recy… 0       Core w…
##  3   34662          1001 Building and demoli… Asph… C&D    Recy… 0       Core w…
##  4   34663          1001 Building and demoli… Asph… C&I    Recy… 0       Core w…
##  5   34664          1001 Building and demoli… Asph… MSW    Recy… 0       Core w…
##  6   34666          1001 Building and demoli… Bric… C&D    Recy… 1.42e+5 Core w…
##  7   34667          1001 Building and demoli… Bric… C&I    Recy… 3.83e+0 Core w…
##  8   34668          1001 Building and demoli… Bric… MSW    Recy… 0       Core w…
##  9   34670          1001 Building and demoli… Cera… C&D    Recy… 3.62e+4 Core w…
## 10   34671          1001 Building and demoli… Cera… C&I    Recy… 5.31e-1 Core w…
## # … with abbreviated variable name ¹`Core_Non-core`
```

Answer: The output of the first 10 records as seen above

EXplanation: To arrive at the output I used the head function specifying the number of rows to be displayed showing the first 10 records.

last 10 records -

Code:

```
last_10<-tail(wastes,10)
print(last_10)
```

```
## # A tibble: 10 × 8
##    Case_ID Year_State_ID Category              Type  Stream Fate  Tonnes Core_…¹
##      <dbl>         <dbl> <chr>                 <chr> <chr>  <chr>  <dbl> <chr>
##  1   88872          1122 Plastics              Poly… MSW    Recy…      0 Core w…
##  2   88874          1122 Plastics              Poly… C&D    Recy…      0 Core w…
##  3   88875          1122 Plastics              Poly… C&I    Recy…      0 Core w…
##  4   88876          1122 Plastics              Poly… MSW    Recy…      0 Core w…
##  5   88878          1122 Textiles, leather & … Leat… C&D    Recy…      0 Core w…
##  6   88879          1122 Textiles, leather & … Leat… C&I    Recy…      0 Core w…
##  7   88880          1122 Textiles, leather & … Leat… MSW    Recy…      0 Core w…
##  8   88882          1122 Textiles, leather & … Text… C&D    Recy…      0 Core w…
##  9   88883          1122 Textiles, leather & … Text… C&I    Recy…      0 Core w…
## 10   88884          1122 Textiles, leather & … Text… MSW    Recy…      0 Core w…
## # … with abbreviated variable name ¹`Core_Non-core`
```

Answer: The ouput of the last 10 records as seen above

Explanation: To arrive at the output I used the tail function specifying the number of rows to be displayed showing the last 10 records.

Question 2.

Finding unique “Type” values in the file -

Code:

```
Type_values<-unique(wastes$Type)
print(Type_values)
```

```
##  [1] "Biosolids"
##  [2] "Asphalt"
##  [3] "Bricks, concrete and pavers"
##  [4] "Ceramics, tiles and pottery"
##  [5] "Plasterboard & cement sheeting"
##  [6] "Rubble"
##  [7] "Soil, sand and rock not contaminated above any threshold requiring classification as co
ntaminated soils (N120)"
##  [8] "Glass from food and beverage containers"
##  [9] "Other glass"
## [10] "Acids (B)"
## [11] "Alkalis (C)"
## [12] "Asbestos (N220)"
## [13] "Clinical and pharmaceutical (R)"
## [14] "Contaminated soils (N120)"
## [15] "Food-derived hazardous wastes (K100, K110)"
## [16] "Inorganic chemicals (D)"
## [17] "Oils (J)"
## [18] "Organic chemicals (M)"
## [19] "Organic solvents (G)"
## [20] "Other"
## [21] "Other hazardous organic wastes (K140, K190)"
## [22] "Other miscellaneous (other T)"
## [23] "Other soil/sludges (other N)"
## [24] "Paints, resins, inks, organic sludges (F)"
## [25] "Pesticides (H)"
## [26] "Plating and heat treatment (A)"
## [27] "Reactive chemicals (E)"
## [28] "Tyres (T140)"
## [29] "Aluminium"
## [30] "Iron and steel"
## [31] "Non-ferrous metals (ex. aluminium)"
## [32] "Food organics"
## [33] "Garden organics"
## [34] "Other organics"
## [35] "Timber"
## [36] "Certified compostable plastics"
## [37] "High density polyethylene (HDPE) (2)"
## [38] "Low density polyethylene (LDPE) (4)"
## [39] "Other plastics (7)"
## [40] "Polyethylene terephthalate (PET) (1)"
## [41] "Polypropylene (PP) (5)"
## [42] "Polystyrene (PS) (6)"
## [43] "Polyvinyl chloride (PVC) (3)"
## [44] "Textiles"
## [45] "Ash"
## [46] "Cardboard"
## [47] "Newsprint & magazines"
## [48] "Office paper"
## [49] "Polymer coated paperboard"
```

```
## [50] "Leather & rubber (excl. tyres)"
## [51] "Unclassified materials"
```

Answer: As we can see above there are 51 unique “Type” of waste materials in the dataset.The names of the unique “Type” of waste materials are aslo displayed above.

Explanation: To find the unique Type values I chose the column “Type” because it refers to the detailed classification of waste material. And to achieve this I counted the unique values in the column type of the dataset using unique function and then I printed it out to see the unique type values.

Type values containing the keywords “chemicals” & “organics”

Code:

```
library(stringr)
```

Explanation for using stringr: str_count function is used to count the occurences of the word chemicals and Organics in the dataset

Code: To find the no of occurences of the word chemicals and organics in the dataset

```
##To avoid the error of getting reached getOption("max print")
options(max.print = 30000)

x<-sum(str_count(wastes,"\\borganics\\b"),na.rm = TRUE)
```

```
## Warning in stri_count_regex(string, pattern, opts_regex = opts(pattern)):
## argument is not an atomic vector; coercing
```

```
y<-sum(str_count(wastes,"\\bchemicals\\b"),na.rm = TRUE)
```

```
## Warning in stri_count_regex(string, pattern, opts_regex = opts(pattern)):
## argument is not an atomic vector; coercing
```

```
print("No of occurences of the word organics is:")
```

```
## [1] "No of occurences of the word organics is:"
```

```
print(x)
```

```
## [1] 3156
```

```
print("No of occurences of the word chemicals is:")
```

```
## [1] "No of occurences of the word chemicals is:"
```

```
print(y)
```

```
## [1] 4416
```

Answer: Can be seen above that is no of occurences of organcis and chemicals are : 3156 & 4416 respectively

Explanation: I have used the str_count function to count the occurences and I have used the format "\b the word to match\b" as a method of regular expression to match chemicals and organics in the dataset where \b is a word boundary anchor that matches the empty string at the beginning or end of a word.

Question 3.

finding the number of columns containing missing values and naming those

code:

```
colSums(is.na(wastes))
```

```
##       Case_ID Year_State_ID      Category          Type        Stream
##             0             0             0             0             0
##          Fate        Tonnes Core_Non-core
##             0            11            14
```

Answer: We can see above that there are two columns with missing values i.e. Tonnes and Core_Non-core

Explanation:colSums is a function that calculates the sum of each column in our dataset and inside that I have used is.na to calculate the count of na values in each column

Finding the missingness percentage

code:

```
colMeans(is.na(wastes))*100
```

```
##       Case_ID Year_State_ID      Category          Type        Stream
##    0.00000000    0.00000000    0.00000000    0.00000000    0.00000000
##          Fate        Tonnes Core_Non-core
##    0.00000000    0.02995887    0.03812948
```

Answer: Here we see that the missing percentage for Tonnes is 0.02995887% & for Core_Non-core is 0.03812948%.

Explanation: is.na is used to create a logical expression to findout NA/missing values in the dataset and with colmeans to findout out means of each column of the logical expression. Multiplying it with 100 gives us the missingness %ages of Tonnes and Core_Non-core.

Question 4.

Finding no. of rows containing duplicates and printing them out

Code: No of rows containing duplicates

```
sum(duplicated(wastes))
```

```
## [1] 3
```

Code: The duplicated rows are:

```
duplicate_rows<-wastes[duplicated(wastes),]
print(duplicate_rows)
```

```
## # A tibble: 3 × 8
##   Case_ID Year_State_ID Category Type                Stream Fate  Tonnes Core_…¹
##     <dbl>         <dbl> <chr>    <chr>               <chr>  <chr>  <dbl> <chr>
## 1   88850          1122 Plastics High density polye… C&D    Recy…      0 Core w…
## 2   88851          1122 Plastics High density polye… C&I    Recy…      0 Core w…
## 3   88852          1122 Plastics High density polye… MSW    Recy…      0 Core w…
## # … with abbreviated variable name ¹`Core_Non-core`
```

Answer: In the above we can see that there are 3 rows in the dataset wastes that are duplicates of each other. Namely: the rows with case_ID as: 88850,88851 & 88852

Explanation: Here I have used duplicated function to check for the rows that have duplicated values with sum function to see the number of duplicated rows and then have printed out the duplicated rows.

Removing dupliacted rows and creating a new dataframe

Code:

```
wastes_new <- wastes[!duplicated(wastes), ]
wastes_new
```

```
## # A tibble: 36,714 × 8
##    Case_ID Year_State_ID Category             Type  Stream Fate   Tonnes Core_…¹
##      <dbl>         <dbl> <chr>                <chr> <chr>  <chr>   <dbl> <chr>
##  1   34658          1001 Biosolids            Bios… C&I    Disp… 2.02e+4 Core w…
##  2   34659          1001 Biosolids            Bios… C&I    Recy… 0       Core w…
##  3   34662          1001 Building and demoli… Asph… C&D    Recy… 0       Core w…
##  4   34663          1001 Building and demoli… Asph… C&I    Recy… 0       Core w…
##  5   34664          1001 Building and demoli… Asph… MSW    Recy… 0       Core w…
##  6   34666          1001 Building and demoli… Bric… C&D    Recy… 1.42e+5 Core w…
##  7   34667          1001 Building and demoli… Bric… C&I    Recy… 3.83e+0 Core w…
##  8   34668          1001 Building and demoli… Bric… MSW    Recy… 0       Core w…
##  9   34670          1001 Building and demoli… Cera… C&D    Recy… 3.62e+4 Core w…
## 10   34671          1001 Building and demoli… Cera… C&I    Recy… 5.31e-1 Core w…
## # … with 36,704 more rows, and abbreviated variable name ¹`Core_Non-core`
## # ℹ Use `print(n = ...)` to see more rows
```

Answer: I have removed the duplicates and created a new dataframe wastes_new

Explanation: I have done this by simply applying a !mark to the function of duplicated(wastes) to remove the 3 rows that appeared in the duplicated(wastes) function.

Question 5.

Finding unique cases in the data file

Code:

```
nrow(wastes_new)
```

```
## [1] 36714
```

Answer: There are 36714 unique cases

Explanation: Since we removed the duplicates in the above Q4. we have only the cases which are unique so we print the no of rows of the new dataframe wastes_new.

Finding the number of core and Non core cases

Code:

```
sum(str_count(wastes_new,"\\bCore\\b"),na.rm = TRUE)
```

```
## Warning in stri_count_regex(string, pattern, opts_regex = opts(pattern)):
## argument is not an atomic vector; coercing
```

```
## [1] 36575
```

```
sum(str_count(wastes_new,"\\bNon-core\\b"),na.rm = TRUE)
```

```
## Warning in stri_count_regex(string, pattern, opts_regex = opts(pattern)):
## argument is not an atomic vector; coercing
```

```
## [1] 125
```

Answer: The number of core cases are 36575 and the number of Non-core cases are 125

Explanation: I have again used the str_count method and sum along with the regex expressions “\bCore\b” & “\bNon-core\b” to identify the cases and to not count missing values i have used na.rm = TRUE.

Question 6.

Finding the amount of tones of waste for different waste sources from various categories.

stream, category & Tones

Code:

```
wastes_new3<-wastes_new
```

```
table_frac_new<-wastes_new3 %>%
  group_by(Category,Stream)%>%
  summarise(total_tonnes = sum(Tonnes,na.rm=TRUE))%>%
  mutate(frac = round(total_tonnes/sum(total_tonnes),4))
```

```
## `summarise()` has grouped output by 'Category'. You can override using the
## `.groups` argument.
```

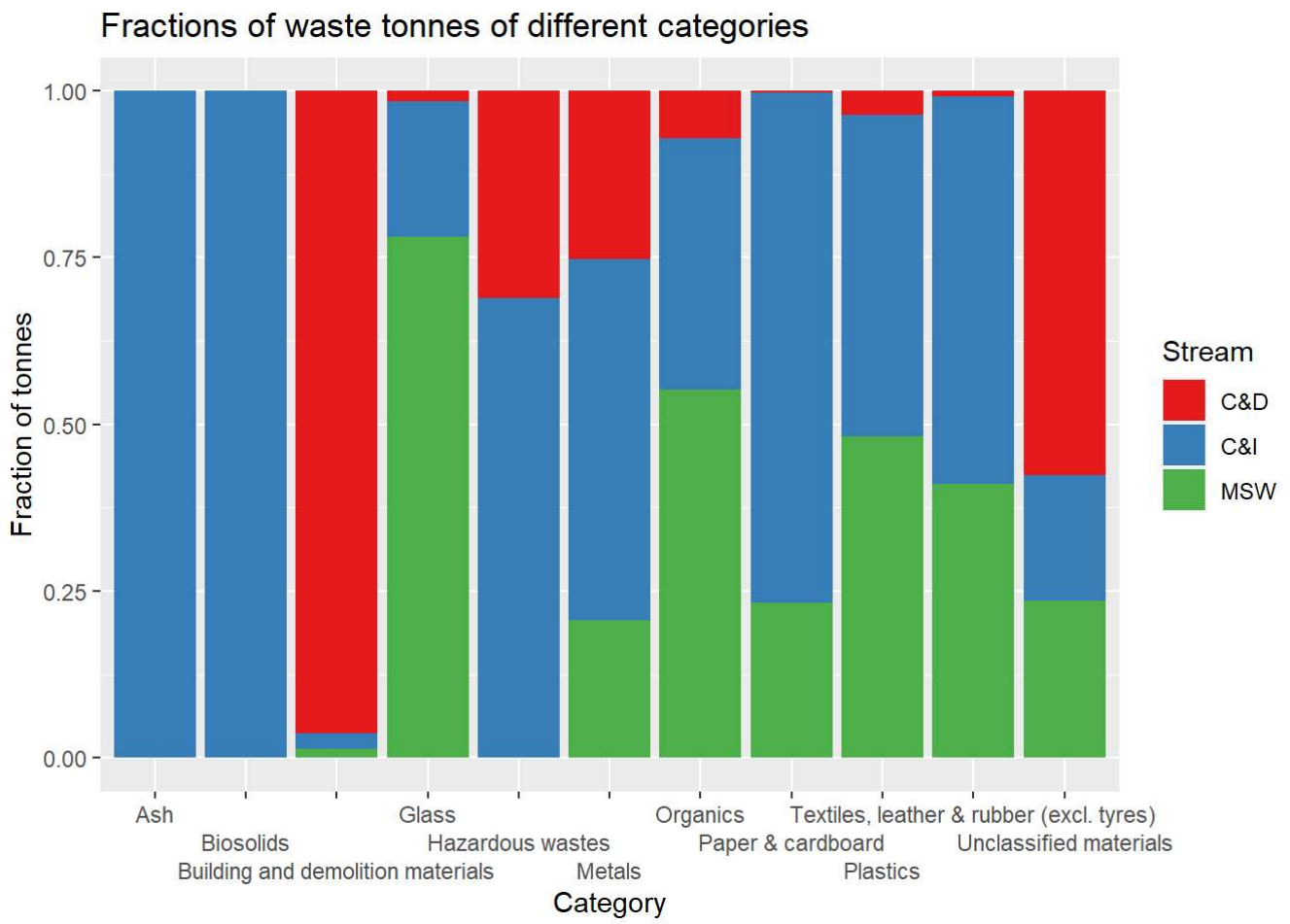
```
write.table(table_frac_new,"C:/Users/Tamanna Das/Desktop/table_frac.csv",sep=",",row.names=FALS
E)
```

```
colSums(is.na(table_frac_new))
```

```
##     Category       Stream total_tonnes         frac
##            0            0            0            0
```

Plotting frac vs category

```
ggplot(data = table_frac_new,aes(x = Category, y = frac, fill = Stream)) + geom_bar(stat = "iden
tity", position = "stack")+scale_x_discrete(guide=guide_axis(n.dodge=3)) +labs(title = "Fraction
s of waste tonnes of different categories", x = "Category", y = "Fraction of tonnes")+scale_fill
_brewer(palette = "Set1")
```



Answer: I have drawn a chart showing the fraction value specific to the category. Here we can observe that Categories such as Ash and Biosolids have the highest percentage of fraction of wastes of C&I waste source, Building & demolition materials have the highest fraction of wastes for C&D waste source and Glass category have the highest waste fraction for MSW waste source.

Explanation: First I created a dataset call table_frac_new there I grouped by Category and STream and then summarised for total tonnes after that I added a col named frac to find the total tonnes for that group divided by the sum of tonnes for all groups to calculate fraction.Post that I created a ggplot using geom_bar.

Question 7.

compute total tonnes for each Fate value, and then sort the total tonnes in a descending manner and display the data showing two columns (Fate and total tonnes).

Code:

```
fate<-aggregate(wastes_new$Tonnes, by=list(Fate = wastes_new$Fate), FUN = sum, na.rm = TRUE, na.action = NULL)
```

```
colnames(fate)[colnames(fate) == "x"] ="Total tonnes"
```

```
fate %>% arrange(desc(`Total tonnes`))
```

```
##                Fate Total tonnes
## 1         Recycling    341913950
## 2          Disposal    134596697
## 3 Long-term storage     89131044
## 4   Energy recovery     20556464
## 5       Waste reuse      3868426
```

Answer: Displayed the columns fate and total_tonnes as seen above.The fate value showing the largest total tonnes is Recycling.

Explanation: I used the function aggregate to sum the tonnes by fate and I saved it in a table where I renmaed the sum of tonnes col by Total_tonnes and then I arranged it by descending order of total tonnes.

Question 8.

Write code to only keep data records that are about the Category of Organics, and then compute total tonnes against Type and Stream. Display the data and save them in a file named “wastes_organics_type_stream.csv”

Code:

```
Category_Organics<-wastes_new %>%
  filter(Category == "Organics")
#print(Category_Organics)
```

```
wastes_organics_type_stream<-Category_Organics %>% group_by(Type,Stream) %>%
  summarise(sum =sum(Tonnes,na.rm = TRUE),
            .groups = 'drop') %>%
  as.data.frame()
```

```
print(wastes_organics_type_stream)
```

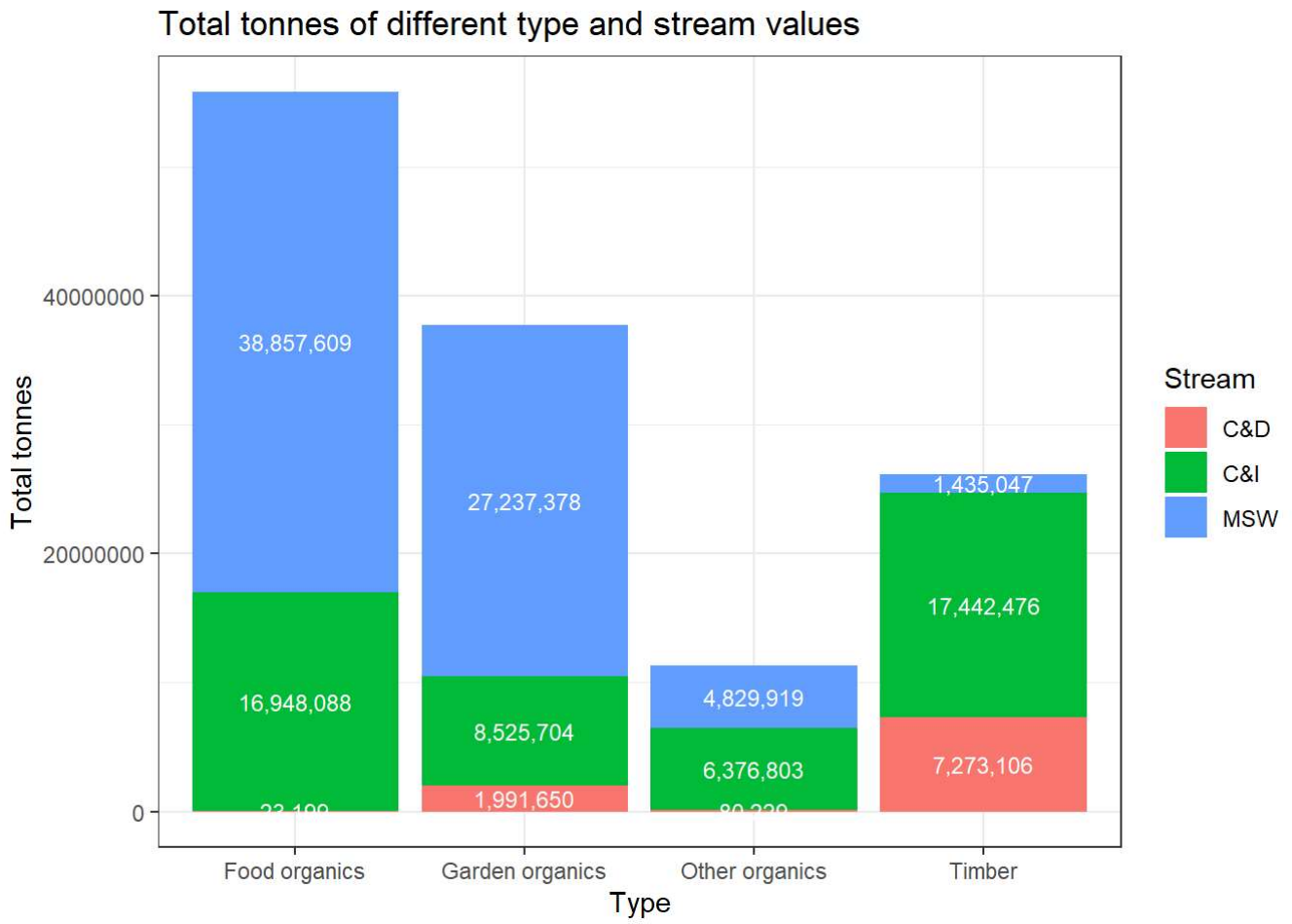
```
##               Type Stream         sum
## 1    Food organics    C&D    23198.68
## 2    Food organics    C&I 16948088.16
## 3    Food organics    MSW 38857609.32
## 4  Garden organics    C&D  1991650.10
## 5  Garden organics    C&I  8525703.68
## 6  Garden organics    MSW 27237377.58
## 7   Other organics    C&D    80229.13
## 8   Other organics    C&I  6376803.10
## 9   Other organics    MSW  4829919.39
## 10          Timber    C&D  7273105.67
## 11          Timber    C&I 17442475.93
## 12          Timber    MSW  1435046.65
```

```
write.table(wastes_organics_type_stream,"C:/Users/Tamanna Das/Desktop/wastes_organics_type_strea
m.csv",sep=",",row.names=FALSE)
```

Draw a chart to show the total tonnes of different Type and Stream values in “wastes_organics_type_stream.csv”

Code:

```
options(scipen = 999)
ggplot(wastes_organics_type_stream, aes(x = Type,y = sum, fill = Stream, group = 1 ))+
  geom_col()+geom_text(aes(label = scales::comma(sum)),
            position = position_stack(vjust = 0.5),
            size = 3, color = "white")+
  theme_bw()+labs(x="Type",y="Total tonnes",title = "Total tonnes of different type and stream v
alues")
```



Answer: Here we can observe that C&D stream has the highest waste tonnes for Timber type waste, C&I stream has highest waste tonnes for timber type waste and MSW stream has highest waste in the type food organics.

Explanation: I have used filter first to filter out the category of Organics then I have used group by type and stream and summarised total tonnes and then I have written that data to csv. Post that I plotted a stacked column plot by color - stream and x axis as Type values while Y axis as total tonnes.

Question 9.

Write code to add two new columns named “year” and “State” and fill it with corresponding values, which can be retrieved from the file Year_State_ID.csv. Display the first 10 data records of the updated data.

Code:

```
merged_data<-wastes_new%>%
  inner_join(Year_State_Id, by=c('Year_State_ID'='ID')

  )
```

```
head(merged_data,10)
```

```
## # A tibble: 10 × 10
##    Case_ID Year_State_ID Category Type  Stream Fate   Tonnes Core_…¹ Year  State
##      <dbl>         <dbl> <chr>    <chr> <chr>  <chr>   <dbl> <chr>   <chr> <chr>
##  1   34658          1001 Biosoli… Bios… C&I    Disp… 2.02e+4 Core w… 2020… ACT
##  2   34659          1001 Biosoli… Bios… C&I    Recy… 0       Core w… 2020… ACT
##  3   34662          1001 Buildin… Asph… C&D    Recy… 0       Core w… 2020… ACT
##  4   34663          1001 Buildin… Asph… C&I    Recy… 0       Core w… 2020… ACT
##  5   34664          1001 Buildin… Asph… MSW    Recy… 0       Core w… 2020… ACT
##  6   34666          1001 Buildin… Bric… C&D    Recy… 1.42e+5 Core w… 2020… ACT
##  7   34667          1001 Buildin… Bric… C&I    Recy… 3.83e+0 Core w… 2020… ACT
##  8   34668          1001 Buildin… Bric… MSW    Recy… 0       Core w… 2020… ACT
##  9   34670          1001 Buildin… Cera… C&D    Recy… 3.62e+4 Core w… 2020… ACT
## 10   34671          1001 Buildin… Cera… C&I    Recy… 5.31e-1 Core w… 2020… ACT
## # … with abbreviated variable name ¹`Core_Non-core`
```

Answer: Displaying the first 10 records of the merged data as seen above

Explanation: In the above code I have first merged the data sets waste_new_final with Year_State_id by the common column name in both data sets namely year_state_id in wastes_new_final and ID in Year_State_Id.And then I displayed the first 10 records using head function.

Question 10.

Write code to display the statistical information, i.e., Min, Max, and Mean, of waste tonnes for each state.

Code:

```
options(scipen = 999)
summary_stats<-merged_data%>%
  group_by(State)%>%
  summarise(min_Tonnes = min(Tonnes,na.rm = TRUE), max_Tonnes = max(Tonnes,na.rm = TRUE), mean_T
onnes = mean(Tonnes,na.rm = TRUE))
```

```
summary_stats%>%slice_max(summary_stats$mean_Tonnes)
```

```
## # A tibble: 1 × 4
##   State min_Tonnes max_Tonnes mean_Tonnes
##   <chr>      <dbl>      <dbl>       <dbl>
## 1 NSW            0    4814588      46840.
```

Which state has the largest Mean value of waste tonnes?

Answer: NSW has the largest mean value of waste tonnes. I have also displayed the statistical information as seen above.

Explanation: In the merged data I have grouped by State column and then I summarised the statistical info required using the functions mean, max, min of waste tonnes. Then I arranged it in descending order of mean tonnes to see which state has the highest mean tonnes

Question 11.

Write code to display the most recycled waste Type and the most disposed waste Type with the corresponding year.

Code:

```
filter_recycling<-merged_data%>%filter(merged_data$Fate == "Recycling")%>%group_by(Type,Year)%>%
summarise(sum =sum(Tonnes,na.rm = TRUE),
            .groups = 'drop')

filter_recycling %>% slice_max(filter_recycling$sum)
```

```
## # A tibble: 1 × 3
##   Type                        Year            sum
##   <chr>                       <chr>         <dbl>
## 1 Bricks, concrete and pavers 2020-2021 10478681.
```

Answer: The most recycled waste type with the corresponding years is Bricks, concrete and pavers (2020-2021).

Explanation: Here I have first filtered the waste type recycling and grouped by Type & Year and then summarised tottal tonnes and then amongst those I have sliced out the max and which year it was..The same was done with type=disposed.

Type “Disposed”

```
filter_disposed<-merged_data%>%filter(merged_data$Fate == "Disposal")%>%group_by(Type,Year)%>%su
mmarise(sum =sum(Tonnes,na.rm = TRUE),
            .groups = 'drop')

filter_disposed %>% slice_max(filter_disposed$sum)
```

```
## # A tibble: 1 × 3
##   Type          Year          sum
##   <chr>         <chr>       <dbl>
## 1 Food organics 2008-2009 4089186
```

Answer: The most disposed waste type is Food organics in the year (2008-2009)

Write code to display the most increased waste Type over years.

Code:

```
df3<-merged_data %>%
  group_by(Type,Year)%>%summarise(sum =sum(Tonnes,na.rm = TRUE),
            .groups = 'drop')
```

```
growth_rate<-df3%>%group_by(Type)%>%arrange(Year,.by_group = TRUE)%>%mutate(Growth = (sum - lag
(sum,na.rm=TRUE))/lag(sum,na.rm=TRUE))%>%
   mutate(Growth = replace(Growth, is.infinite(Growth), NA))
```

```
growth_rate%>%
  filter(Year != min(Year))%>%
  group_by(Type)%>%
  summarise(avg_growth = mean(Growth,na.rm = TRUE))%>%
  slice_max(order_by = avg_growth)
```

```
## # A tibble: 1 × 2
##   Type         avg_growth
##   <chr>             <dbl>
## 1 Office paper       99.8
```

Answer: Displaying the most increased waste type over the years that is Office paper

Explanation: In the above code after grouping by type and year and summarising the total tonnes I have calculated a year-on-year growth rate for each group, I have also replaced the term where infinity occurs due to previous value becomming 0 as NA and then calculated the avergae growth rate by doing mean of growth rates for each group and then sliced out the max value.

Question 12.

Write code to only keep data records where the Category value is Hazardous wastes, the Type value is Tyres (T140) and tonnes value is more than 0

Code:

```
Category_Hazardous<-merged_data %>%
  filter(Category == "Hazardous wastes" & Type == "Tyres (T140)" & Tonnes >0)
```

```
print(Category_Hazardous)
```

```
## # A tibble: 392 × 10
##    Case_ID Year_State_ID Category  Type  Stream Fate  Tonnes Core_…¹ Year  State
##      <dbl>         <dbl> <chr>     <chr> <chr>  <chr>  <dbl> <chr>   <chr> <chr>
##  1   35081          1001 Hazardou… Tyre… C&I    Ener…  2801. Core w… 2020… ACT
##  2   35082          1001 Hazardou… Tyre… C&I    Disp…  1452. Core w… 2020… ACT
##  3   35083          1001 Hazardou… Tyre… C&I    Disp…   104. Core w… 2020… ACT
##  4   35084          1001 Hazardou… Tyre… C&I    Recy…  1245. Core w… 2020… ACT
##  5   35830          1003 Hazardou… Tyre… C&I    Ener… 48222. Core w… 2020… NSW
##  6   35831          1003 Hazardou… Tyre… C&I    Disp… 45684. Core w… 2020… NSW
##  7   35832          1003 Hazardou… Tyre… C&I    Disp…  2644. Core w… 2020… NSW
##  8   35833          1003 Hazardou… Tyre… C&I    Recy… 44521. Core w… 2020… NSW
##  9   36371          1004 Hazardou… Tyre… C&I    Ener…  1069. Core w… 2020… NT
## 10   36372          1004 Hazardou… Tyre… C&I    Disp…  1817. Core w… 2020… NT
## # … with 382 more rows, and abbreviated variable name ¹`Core_Non-core`
## # ℹ Use `print(n = ...)` to see more rows
```

then write code to add a new column named “Tonnes_range” and fill it with one of the following values based on the corresponding “Tonnes” value as mentioned below:

1.[0,10000) 2.[10000,20000) 3.[20000,40000) 4.[40000,80000)

```
Category_Hazardous$Tonnes_range<-ifelse(Category_Hazardous$Tonnes >= 0 & Category_Hazardous$Tonn
es < 10000, "[0,10000)", NA)

Category_Hazardous$Tonnes_range<-ifelse(Category_Hazardous$Tonnes >= 10000 & Category_Hazardous
$Tonnes < 20000, "[10000,20000)", Category_Hazardous$Tonnes_range)

Category_Hazardous$Tonnes_range<-ifelse(Category_Hazardous$Tonnes >= 20000 & Category_Hazardous
$Tonnes < 40000, "[20000,40000)", Category_Hazardous$Tonnes_range)

Category_Hazardous$Tonnes_range<-ifelse(Category_Hazardous$Tonnes >= 40000 & Category_Hazardous
$Tonnes < 80000, "[40000,80000)", Category_Hazardous$Tonnes_range)
```

```
print(Category_Hazardous)
```

```
## # A tibble: 392 × 11
##    Case_ID Year_…¹ Categ…² Type  Stream Fate  Tonnes Core_…³ Year  State Tonne…⁴
##      <dbl>   <dbl> <chr>   <chr> <chr>  <chr>  <dbl> <chr>   <chr> <chr> <chr>
##  1   35081    1001 Hazard… Tyre… C&I    Ener…  2801. Core w… 2020… ACT   [0,100…
##  2   35082    1001 Hazard… Tyre… C&I    Disp…  1452. Core w… 2020… ACT   [0,100…
##  3   35083    1001 Hazard… Tyre… C&I    Disp…   104. Core w… 2020… ACT   [0,100…
##  4   35084    1001 Hazard… Tyre… C&I    Recy…  1245. Core w… 2020… ACT   [0,100…
##  5   35830    1003 Hazard… Tyre… C&I    Ener… 48222. Core w… 2020… NSW   [40000…
##  6   35831    1003 Hazard… Tyre… C&I    Disp… 45684. Core w… 2020… NSW   [40000…
##  7   35832    1003 Hazard… Tyre… C&I    Disp…  2644. Core w… 2020… NSW   [0,100…
##  8   35833    1003 Hazard… Tyre… C&I    Recy… 44521. Core w… 2020… NSW   [40000…
##  9   36371    1004 Hazard… Tyre… C&I    Ener…  1069. Core w… 2020… NT    [0,100…
## 10   36372    1004 Hazard… Tyre… C&I    Disp…  1817. Core w… 2020… NT    [0,100…
## # … with 382 more rows, and abbreviated variable names ¹Year_State_ID,
## #   ²Category, ³`Core_Non-core`, ⁴Tonnes_range
## # ℹ Use `print(n = ...)` to see more rows
```

Then, for each state, display a chart to show the number of cases of different score_range values. What do you observe?

```
Count_range<-Category_Hazardous%>%
  count(State,Tonnes_range)
```

```
  ggplot(Count_range,aes(x = Count_range$Tonnes_range,y = n,fill = State))+geom_col()+geom_text
(aes(label = scales::comma(n)),
            position = position_stack(vjust = 0.5),
            size = 3, color = "white")+
  theme_bw()+labs(x="Tonnes Range",y="No of cases of different score range",title = "No of cases
of different score range across cities")
```

```
## Warning: Use of `Count_range$Tonnes_range` is discouraged. Use `Tonnes_range` instead.
## Use of `Count_range$Tonnes_range` is discouraged. Use `Tonnes_range` instead.
```

Answer: ACT,NT & TAS state has the lowest amount of tonnes geneartion amongst all beacuse the no of cases of ACT,NT & Tas's tonnes generation is only in (0,10000) which is the lowest range of tonnes.While generation of tonnes in (40000-80000) range is highest for NSW that means NSW is the state generating most amount of waste materials.

Explanation: First I have filtered the data according to our need and then I have created a column named tonnes_range where I placed a value based on the tonnes column as specified in the question by using if else statements and then post that I did a count of state and tonnes range by group to see the count of tonnes range by state and then used ggplot to plot the data.

Question 13.

Write code to draw a chart showing a yearly trend of total waste tonnes of food organics for each state. To draw the chart, please convert the Year-Year formats of all Year values into Year formats. For example, convert 2006-2007 into 2006 and 2020-2021 into 2020. What do you observe from the chart?

Code:

```
merged_foodorganics<-merged_data %>%
  filter(Type == "Food organics")%>%
  group_by(State,Year)%>%summarise(sum =sum(Tonnes,na.rm = TRUE), .groups = 'drop')
```

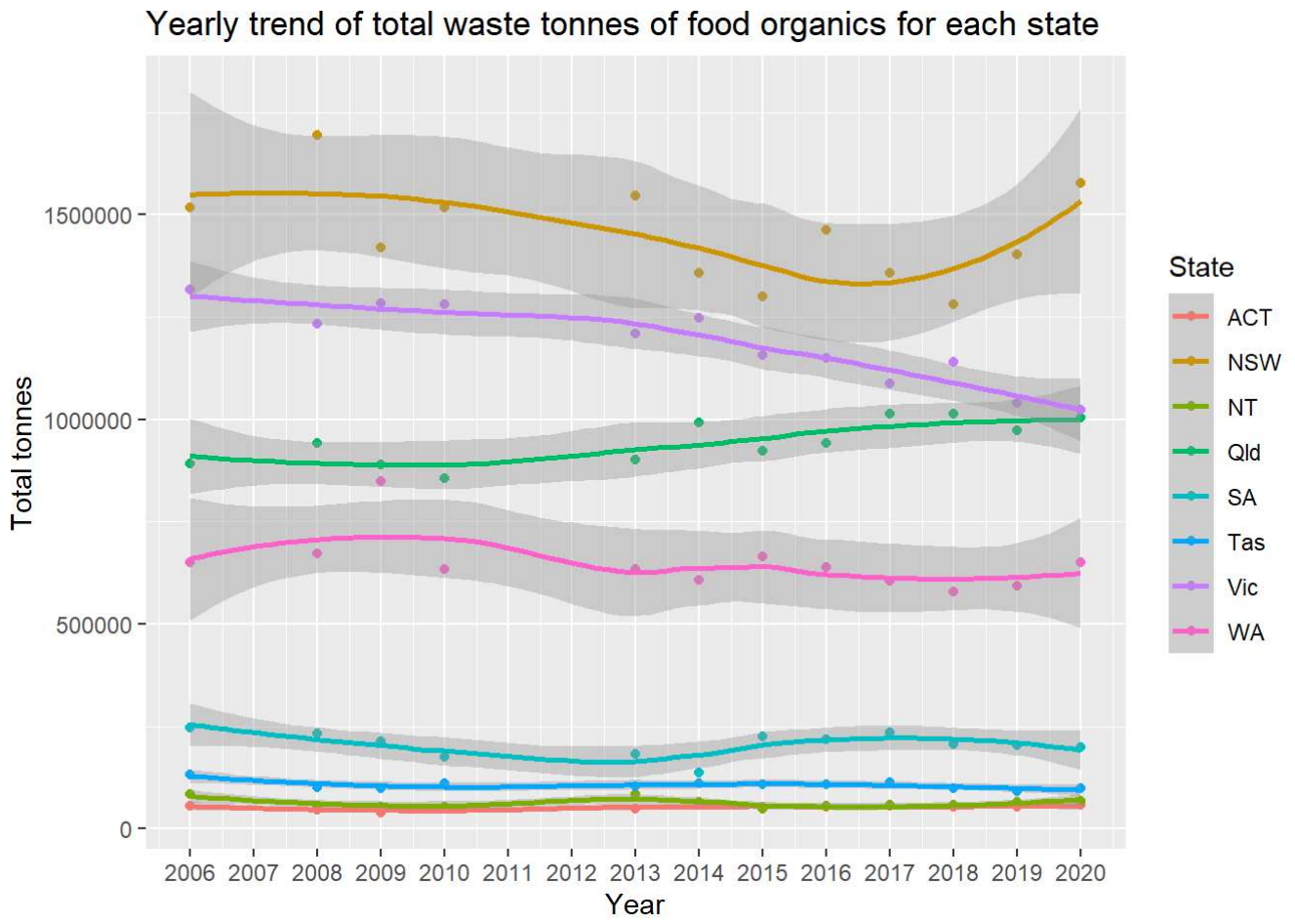
```
merged_foodorganics$Extracted_Year=substring(merged_foodorganics$Year,1,4)
merged_foodorganics$Extracted_Year <- as.numeric(merged_foodorganics$Extracted_Year)

merged_foodorganics
```

```
## # A tibble: 96 × 4
##    State Year        sum Extracted_Year
##    <chr> <chr>     <dbl>          <dbl>
##  1 ACT   2006-2007 55924           2006
##  2 ACT   2008-2009 46602           2008
##  3 ACT   2009-2010 38175           2009
##  4 ACT   2010-2011 49398           2010
##  5 ACT   2013-2014 47498           2013
##  6 ACT   2014-2015 60292           2014
##  7 ACT   2015-2016 51788           2015
##  8 ACT   2016-2017 52326           2016
##  9 ACT   2017-2018 54273           2017
## 10 ACT   2018-2019 52355           2018
## # … with 86 more rows
## # ℹ Use `print(n = ...)` to see more rows
```

```
options(scipen = 999)
ggplot(merged_foodorganics,aes(x = Extracted_Year,y = sum,color = State))+geom_point()+ geom_smo
oth(method = "loess")+theme(strip.text.x = element_text(size=8, angle=75),
          strip.text.y = element_text(size=12, face="bold"),
          strip.background = element_rect(colour="red", fill="#CCCCFF"))+labs(x="Year",y="Total
tonnes",title = "Yearly trend of total waste tonnes of food organics for each state")+scale_x_co
ntinuous(breaks = seq(2006, 2022))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Answer: Here we can observe that amongst all states NSW has the highest total tonnes and has also increased the most However, states like NT,Tas,ACT has the lowest waste generation of food organics and have stayed consistent throught the years.VIC state on the other hand has the decreased their food organics waste generation over the years.

Explanation: Filtered food organics and have grouped by state and year and then I have summed the total tonnes and then I have added a column named extracted year where I have used substring method to extract 2006 from 2006-2007 based on substring(1,4) counting the letters post that for plotting the data I have converted the year into numeric type. Post that I plotted my graph showing yearly trend of total tonnes of food organics using geom_poin and geom_smooth to see the trend line.

Question 14.

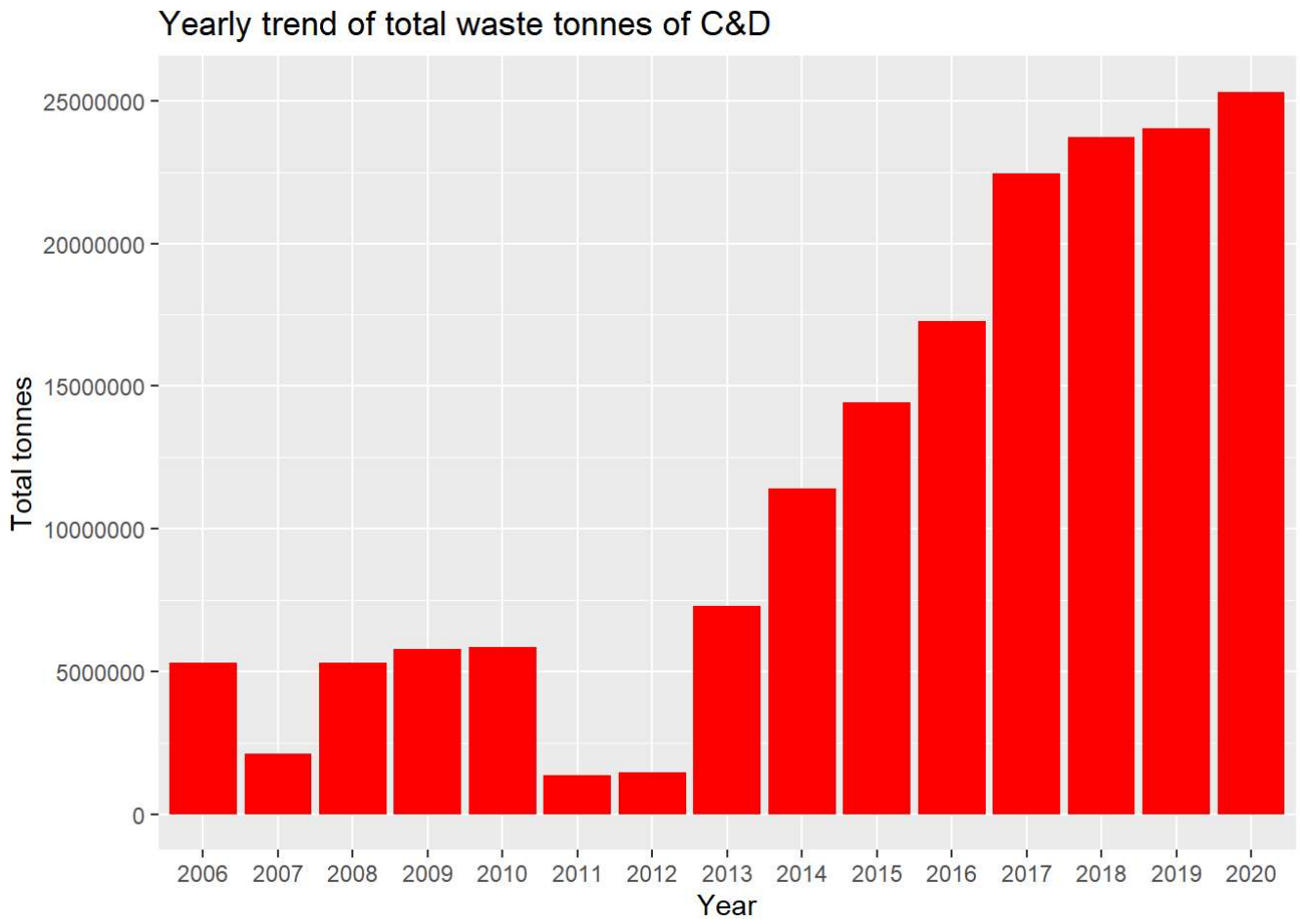
Write code to draw a chart showing a yearly trend of total waste tonnes of C&D. After that, please find a factor affecting the yearly trend of C&D waste on Google or other sources. This factor data also has to be yearly data.

Code:

```
merged_C_D<-merged_data %>%
  filter(Stream == "C&D")%>%
  group_by(Year)%>%summarise(sum =sum(Tonnes,na.rm = TRUE), .groups = 'drop')
```

```
merged_C_D$Extracted_Year=substring(merged_C_D$Year,1,4)
```

```
options(scipen = 999)
ggplot(merged_C_D,aes(x = Extracted_Year,y = sum))+ geom_col(fill = "red")+theme(strip.text.x =
element_text(size=8, angle=75),
          strip.text.y = element_text(size=12, face="bold"),
          strip.background = element_rect(colour="red", fill="#CCCCFF"))+labs(x="Year",y="Total
tonnes",title = "Yearly trend of total waste tonnes of C&D")
```



Answer: Yearly trend of total waste tonnes of C&D have increased over the years but we can also see a slight decrease in 2007,2011 & 2012.

Explanation:I have again filtered C&D waste and then used group function over year and summarized the total tonnes post that I did extraction of year using substring method as done in the previous question and plotted my graph.

With the existing rate of migration and population growth (ABS, 2018b), it is expected that C & D waste generation will continue to grow steadily in the coming years. (https://sbenrc.com.au/app/uploads/2019/10/CIB-WBC-Jun2019-ConstructionDemolitionWasteManagementAustralia.pdf)

population growth rate data
(https://datacommons.org/tools/timeline#&place=country/AUS&statsVar=Count_Person)

```
Population_growth_rate = c(20.5,20.8,21.2,21.7,22,22.3,22.7,23.1,23.4,23.8,24.1,24.5,24.9,25.3,25.6)
merged_C_D_2<-cbind(merged_C_D, Population_growth_rate)
merged_C_D_2$sum <- merged_C_D_2$sum / 10^6
print(merged_C_D_2)
```

```
##         Year       sum Extracted_Year Population_growth_rate
## 1  2006-2007  5.314915           2006                  20.5
## 2  2007-2008  2.095537           2007                  20.8
## 3  2008-2009  5.314525           2008                  21.2
## 4  2009-2010  5.789373           2009                  21.7
## 5  2010-2011  5.850830           2010                  22.0
## 6  2011-2012  1.339393           2011                  22.3
## 7  2012-2013  1.463712           2012                  22.7
## 8  2013-2014  7.279393           2013                  23.1
## 9  2014-2015 11.394013           2014                  23.4
## 10 2015-2016 14.414755           2015                  23.8
## 11 2016-2017 17.258592           2016                  24.1
## 12 2017-2018 22.461497           2017                  24.5
## 13 2018-2019 23.706643           2018                  24.9
## 14 2019-2020 24.021548           2019                  25.3
## 15 2020-2021 25.313728           2020                  25.6
```

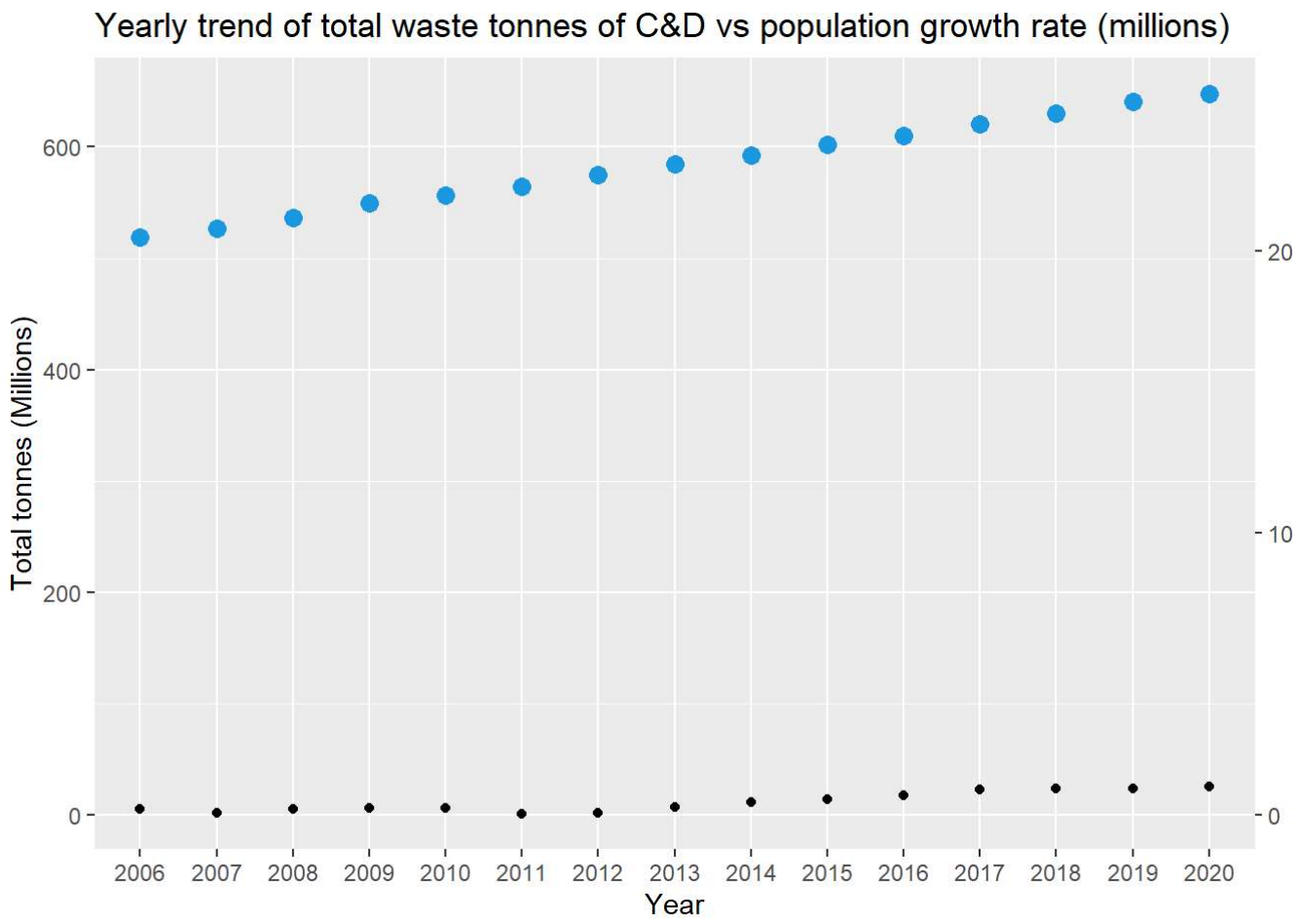
```
ggp1<-ggplot(merged_C_D_2,aes(x = Extracted_Year,y = sum))+ geom_point(fill = "red")+theme(strip.text.x = element_text(size=8, angle=75),
          strip.text.y = element_text(size=12, face="bold"),
          strip.background = element_rect(colour="red", fill="#CCCCFF"))+labs(x="Year",y="Total tonnes (Millions)",title = "Yearly trend of total waste tonnes of C&D vs population growth rate (millions)")
```

```
ggp2<-ggp1+geom_point(aes(x = Extracted_Year, y= merged_C_D_2$Population_growth_rate*max(sum), group = 1),
            col = "#1b98e0", lwd = 3)+scale_y_continuous(sec.axis = sec_axis(~ . / max(merged_C_D_2$sum)))+scale_color_manual(name = "Variables", values = c("Population growth rate" = "#1b98e0", "Waste tonnes of C&D" = "black"))
```

Fig1

```
print(ggp2)
```

```
## Warning: Use of `merged_C_D_2$Population_growth_rate` is discouraged. Use
## `Population_growth_rate` instead.
```




```
ggplot(merged_C_D_2, aes(x = Population_growth_rate, y = sum)) +
  geom_point() +
  labs(x = "Population Growth Rate", y = "Total Waste Tonnes of C&D",
       title = "Relationship between Population Growth and total Waste Tonnes of C&D") +
  theme_bw()
```

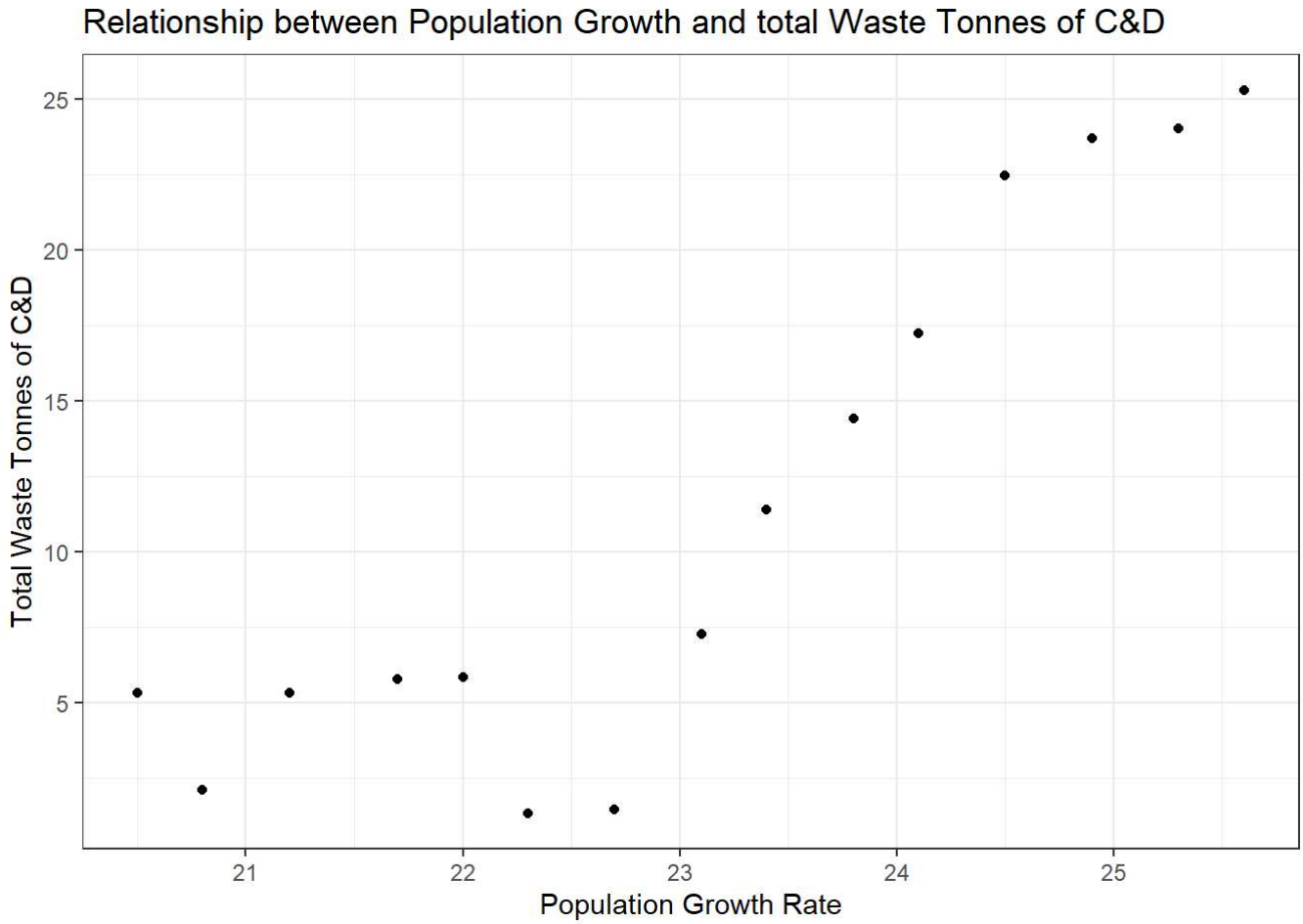


Fig 2

Answer: We can see here in fig 2 that in general if we compare the total waste tonnes of C&D yearly and yearly population growth rate it seems that the total tonnes of C&D waste generation has increased over the years with increase in population growth. However there are few years where the C&D waste has decreased despite of increase in population growth as seen in fig1 like the years 2007, 2011, 2012 but since then it has shown a contant increase.

Explanation: I had C&D waste by year from above plot so to that I added population growth rate for each year using cbin method and also converted the total c&D waste to millions by dividing it with 10^6.Then I created a ggplot for Extracted year and sum of tonnes of C&D using geom_poin to that I added another ggplot of geom_point to see the relation between two points for fig1. For fig 2 I tried to show the relationship between C&D waste and population growth rate from 2006-2020.

Question 15.

Staewise and year wise, core vs non-core waste generation

Code:

```
Core_Noncore<-merged_data %>%
  group_by(`Core_Non-core`,State,Year) %>%
  filter(!is.na(`Core_Non-core`)) %>%
  summarise(total_count=n(),.groups = 'drop') %>%
  as.data.frame()

head(Core_Noncore,40)
```

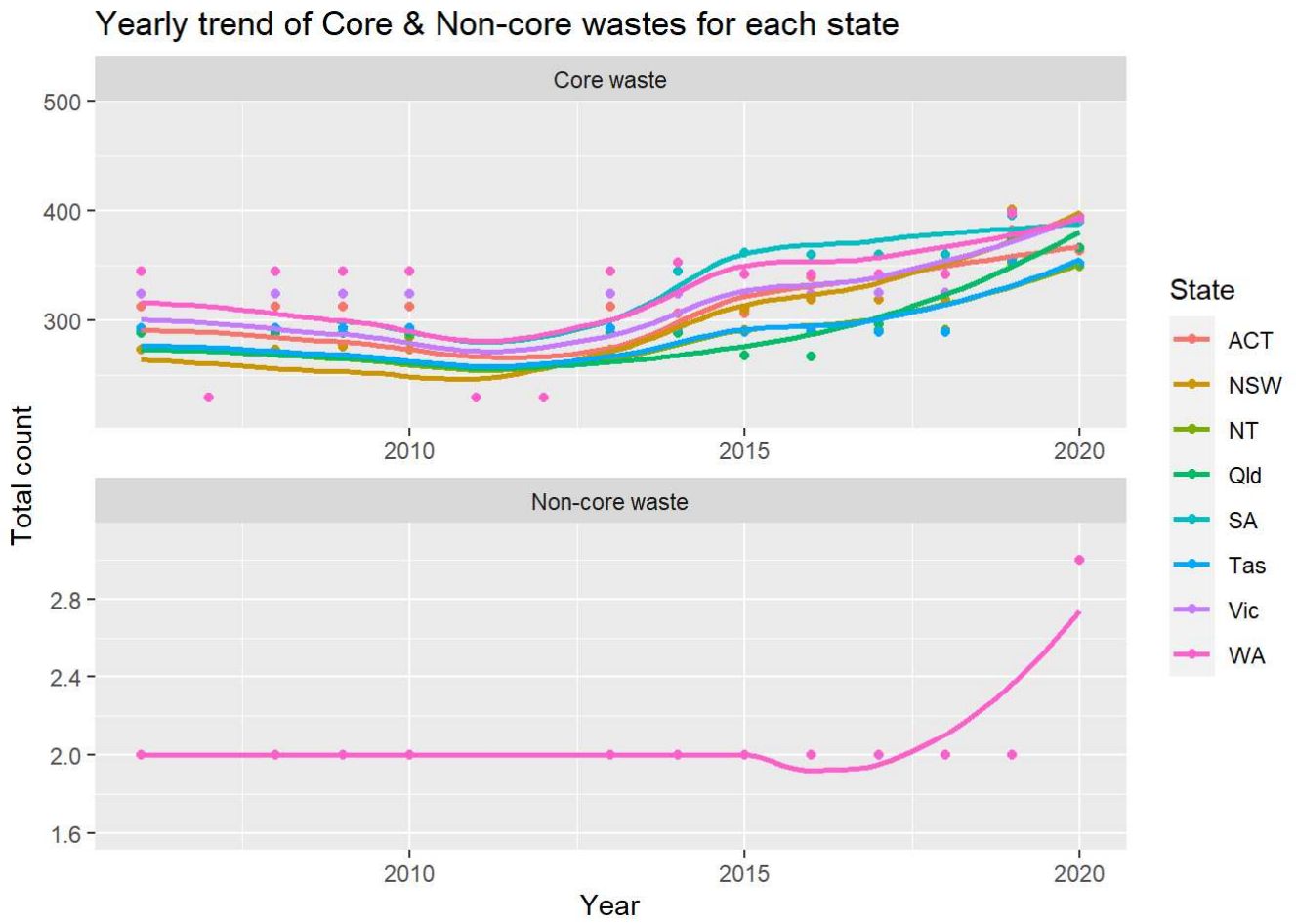
```
##    Core_Non-core State      Year total_count
## 1     Core waste   ACT 2006-2007         312
## 2     Core waste   ACT 2007-2008         229
## 3     Core waste   ACT 2008-2009         312
## 4     Core waste   ACT 2009-2010         312
## 5     Core waste   ACT 2010-2011         312
## 6     Core waste   ACT 2011-2012         229
## 7     Core waste   ACT 2012-2013         229
## 8     Core waste   ACT 2013-2014         312
## 9     Core waste   ACT 2014-2015         306
## 10    Core waste   ACT 2015-2016         306
## 11    Core waste   ACT 2016-2017         339
## 12    Core waste   ACT 2017-2018         340
## 13    Core waste   ACT 2018-2019         319
## 14    Core waste   ACT 2019-2020         382
## 15    Core waste   ACT 2020-2021         363
## 16    Core waste   NSW 2006-2007         273
## 17    Core waste   NSW 2007-2008         229
## 18    Core waste   NSW 2008-2009         273
## 19    Core waste   NSW 2009-2010         276
## 20    Core waste   NSW 2010-2011         273
## 21    Core waste   NSW 2011-2012         229
## 22    Core waste   NSW 2012-2013         229
## 23    Core waste   NSW 2013-2014         273
## 24    Core waste   NSW 2014-2015         325
## 25    Core waste   NSW 2015-2016         310
## 26    Core waste   NSW 2016-2017         319
## 27    Core waste   NSW 2017-2018         319
## 28    Core waste   NSW 2018-2019         319
## 29    Core waste   NSW 2019-2020         401
## 30    Core waste   NSW 2020-2021         395
## 31    Core waste    NT 2006-2007         291
## 32    Core waste    NT 2007-2008         229
## 33    Core waste    NT 2008-2009         291
## 34    Core waste    NT 2009-2010         288
## 35    Core waste    NT 2010-2011         285
## 36    Core waste    NT 2011-2012         229
## 37    Core waste    NT 2012-2013         229
## 38    Core waste    NT 2013-2014         288
## 39    Core waste    NT 2014-2015         288
## 40    Core waste    NT 2015-2016         291
```

```
Core_Noncore$Extracted_Year=substring(Core_Noncore$Year,1,4)
Core_Noncore$Extracted_Year <- as.numeric(Core_Noncore$Extracted_Year)
```

```
options(scipen = 999)
ggplot(Core_Noncore,aes(x = Extracted_Year,y = Core_Noncore$total_count,color = State))+geom_poi
nt()+geom_smooth(alpha = 0)+ labs(x="Year",y="Total count",title = "Yearly trend of Core & Non-c
ore wastes for each state")+facet_wrap(Core_Noncore$`Core_Non-core` ~ ., ncol=1, scales = "fre
e")
```

```
## Warning: Use of `Core_Noncore$total_count` is discouraged. Use `total_count` instead.
## Use of `Core_Noncore$total_count` is discouraged. Use `total_count` instead.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Answer: Non-core waste is present only in the state WA. When we look at the graph of core-wastes, Core waste generation have increased in NSW over the years significantly and a significant increase in core waste can also be seen in SA.

Explanation: Here I have grouped core, Non-core and state and summarised its count I have also handled NA values when summarising. THen I have extracted the year from 2006-2007 into 2006 in a separate col named extracted year and converted that to numeric too for plotting. post that I plotted a facet wrap graph for core and non-core showing the yearly trend of Core_Non-core waste generation over the years in various Australian states.

————END————————————

References for calculating growth rate in question 11 (https://stackoverflow.com/questions/19824601/how-calculate-growth-rate-in-long-format-data-frame)