FUNDAMENTALS OF DATA SCIENCE

— Assignment 1—

Mrunali Vikas Patil

2023-10-02

#PROBLEM SET 1

Import data in R

```
MyData = read.csv("adult.csv", header = T)
```

Question A :

```
summary(MyData)

##      age         workclass            fnlwgt         education        
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561      
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character  
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character  
##  Mean   :38.58                      Mean   : 189778                     
##  3rd Qu.:48.00                      3rd Qu.: 237051                     
##  Max.   :90.00                      Max.   :1484705                     
##  education.num   marital.status      occupation        relationship      
##  Min.   : 1.00   Length:32561       Length:32561       Length:32561      
##  1st Qu.: 9.00   Class :character   Class :character   Class :character  
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :10.08                                                           
##  3rd Qu.:12.00                                                           
##  Max.   :16.00                                                           
##      race               sex             capital.gain    capital.loss   
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0  
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0  
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0  
##                                        Mean   : 1078   Mean   :  87.3  
##                                        3rd Qu.:    0   3rd Qu.:   0.0  
##                                        Max.   :99999   Max.   :4356.0  
##  hours.per.week  native.country     income.bracket    
##  Min.   : 1.00   Length:32561       Length:32561      
```

```
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

```r
summary(MyData$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.58   48.00   90.00
```

Min value for variable age is 17 where as max value is 90. Median and mean are similar hence we can say that data might be normally distributed.

Difference between Mean vs 1st quartile and 3rd quartile vs mean is around 20.

NA values are absent.

```r
summary(MyData$education.num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    9.00   10.00   10.08   12.00   16.00
```

Median and Mean value for for variable education num are similar. Also difference between mean vs 1st quartile and mean vs 3rd quartile is 1 and 2 respectively.

As difference is similar, we can predict that distribution must me close to normal.

```r
library(psych)
describe(MyData$education.num)
```

```
##    vars     n  mean   sd median trimmed  mad min max range  skew kurtosis   se
## X1    1 32561 10.08 2.57     10   10.19 1.48   1  16    15 -0.31     0.62 0.01
```

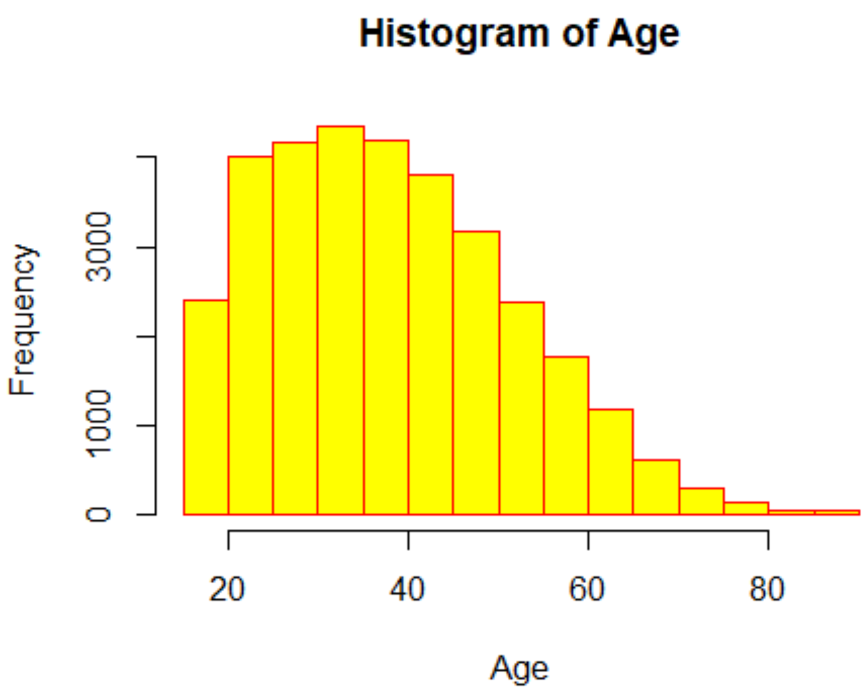
Question B:

We will use scatterplot to compare above two variables

```
plot(x = MyData$age,y = MyData$education.num, xlab = "Age", ylab = "Education Num",
main = "Age vs Education Num")
```

by looking at the scatter plot, we can conclude that there is no linear relation between the two variables.

```
hist(MyData$age, xlab = "Age",col = "yellow",border = "red", main = "Histogram of Age")
```




```
hist(MyData$education.num, xlab = "Education Num",col = "yellow",border = "red", main =
"Histogram of Education Num")
```

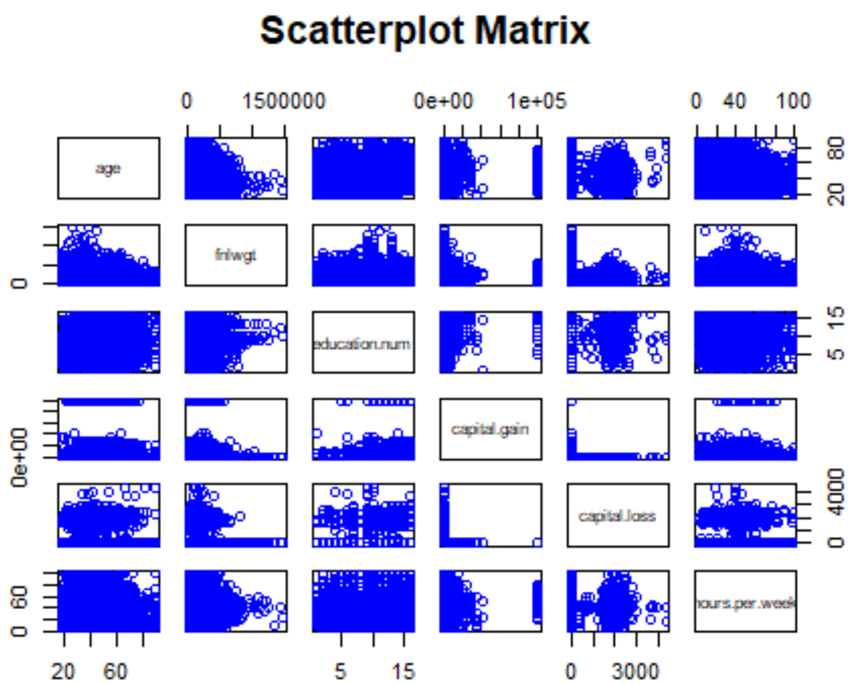
As per the histogram, we can say that data is not normally distributed as we have tail at right side of the graph. Hence data is positively skewed. # For variable education.num, we cannot say anything about normality of the data as data is disbursed randomly.

Question c :
By looking at the summary of the data, we found below mentioned numerical variables.

age fnlwgt education.num capital.gain capital.loss hours.per.week

Creating scatterplot matrix for above variables.

```
pairs(~age+fnlwgt+education.num+capital.gain+capital.loss+hours.per.week,data = MyData,main = "Scatterplot Matrix",col = "blue")
```



As per the matrix, we discovered that there is no linear relation between all variables.
With the help of scatterplot matrix, we can compare all numeric variables with each other in a single graph instead of creating separate scatterplot for two different variables.

Question D :
As per the summary, we found below mentioned variables as the categorical variables.

workclass education marital.status occupation relationship race sex native.country income.bracket

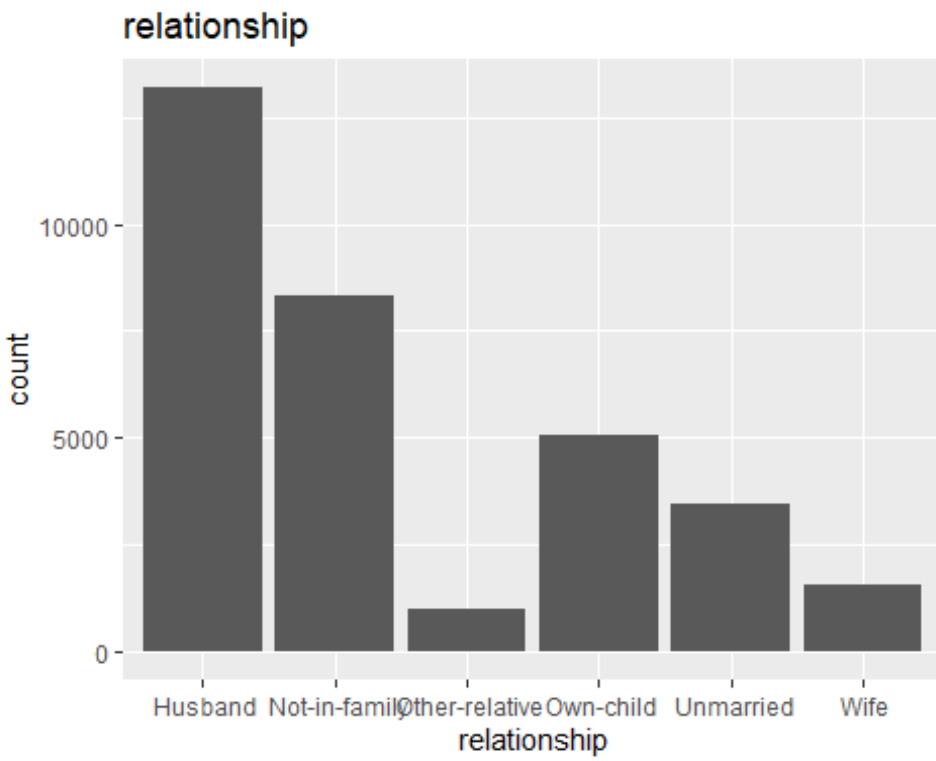
1. Checking categories of categorical variables with the help of bar charts
Using ggplot2 library to create bar chart and used count function to check count by occupation

```
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.3     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ ggplot2::%+%()   masks psych::%+%()
## ✖ ggplot2::alpha() masks psych::alpha()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2)
MyData %>% group_by(occupation) %>% count()

## # A tibble: 15 × 2
## # Groups:   occupation [15]
##    occupation          n
##    <chr>           <int>
```

```
## 1 " ?"                1843
## 2 " Adm-clerical"      3770
## 3 " Armed-Forces"         9
## 4 " Craft-repair"      4099
## 5 " Exec-managerial"   4066
## 6 " Farming-fishing"    994
## 7 " Handlers-cleaners" 1370
## 8 " Machine-op-inspct" 2002
## 9 " Other-service"     3295
## 10 " Priv-house-serv"    149
## 11 " Prof-specialty"    4140
## 12 " Protective-serv"    649
## 13 " Sales"             3650
## 14 " Tech-support"       928
## 15 " Transport-moving"  1597

MyData %>% ggplot(aes(x = occupation)) + geom_bar() + labs(title = "Occupation")
```

Count and bar plot for variable relationship.

```
library(ggplot2)
MyData %>% group_by(relationship) %>% count()

## # A tibble: 6 × 2
## # Groups:   relationship [6]
##   relationship          n
##   <chr>             <int>
## 1 " Husband"        13193
## 2 " Not-in-family"   8305
## 3 " Other-relative"   981
## 4 " Own-child"       5068
## 5 " Unmarried"       3446
## 6 " Wife"            1568

MyData %>% ggplot(aes(x = relationship)) + geom_bar() + labs(title = "relationship")
```



Question E :

```r
MyData_New <- table(MyData$race,MyData$income.bracket)
MyData_New

##
##                    <=50K  >50K
##   Amer-Indian-Eskimo   275    36
##   Asian-Pac-Islander   763   276
##   Black               2737   387
##   Other                246    25
##   White              20699  7117

MyData_New <- as.data.frame(MyData_New)
print(MyData_New)

##                  Var1  Var2  Freq
## 1  Amer-Indian-Eskimo <=50K   275
## 2  Asian-Pac-Islander <=50K   763
## 3               Black <=50K  2737
## 4               Other <=50K   246
## 5               White <=50K 20699
## 6  Amer-Indian-Eskimo  >50K    36
## 7  Asian-Pac-Islander  >50K   276
## 8               Black  >50K   387
## 9               Other  >50K    25
## 10              White  >50K  7117

names(MyData_New) <- c("Race", "IncomeBracket", "Frequency")

library(ggplot2)

ggplot(MyData_New, aes(x = Race, y = Frequency, fill = IncomeBracket)) + geom_bar(stat =
"identity") + labs(x = "Race", y = "Frequency", title = "Race vs. IncomeBracket") +
scale_fill_manual(values = c(" <=50K" = "skyblue", " >50K" = "yellow")) +
theme_minimal()
```

As per the bar chart, we can see people with White race have more number of records. And there are more people who falls under <50k bracket.

#PROBLEM SET 2

Question-1 :

```
Population_even = read.csv("population_even.csv", header = T)
Population_odd = read.csv("population_odd.csv", header = T)
Population_Data = merge(x = Population_even, y = Population_odd, by = "STATE")
```

Joining two data tables to merge into one data frame basen on the common variable

Question-2:

A) {r}colnames(Population_Data) # As we do not have duplicate state column, we cannot delete one.

B) Renaming columns to just years.

```
Population_Data <- Population_Data %>% rename("2010" = "POPESTIMATE2010","2011"
= "POPESTIMATE2011","2012" = "POPESTIMATE2012","2013" =
"POPESTIMATE2013","2014" = "POPESTIMATE2014","2015" =
"POPESTIMATE2015","2016" = "POPESTIMATE2016","2017" =
"POPESTIMATE2017","2018" = "POPESTIMATE2018","2019" = "POPESTIMATE2019")
colnames(Population_Data)

##  [1] "STATE"  "NAME.x" "2010"   "2012"   "2014"   "2016"   "2018"   "NAME.y"
##  [9] "2011"   "2013"   "2015"   "2017"   "2019"
```

C) Reordering columns according to year

```
Population_Data <- Population_Data[, c("STATE", "NAME.x", "2010", "2011","2012","2013",
"2014", "2015","2016", "2017", "2018", "2019")]
Population_Data

##   STATE               NAME.x     2010     2011     2012     2013     2014
## 1     1              Alabama  4785437  4799069  4815588  4830081  4841799
## 2     2               Alaska   713910   722128   730443   737068   736283
## 3     4              Arizona  6407172       NA  6554978  6632764  6730413
## 4     5             Arkansas  2921964  2940667  2952164  2959400  2967392
## 5     6           California 37319502 37638369 37948800 38260787 38596972
## 6     8             Colorado  5047349  5121108  5192647  5269035  5350101
## 7     9          Connecticut  3579114  3588283  3594547  3594841  3594524
## 8    10             Delaware   899593   907381   915179   923576   932487
## 9    11 District of Columbia   605226   619800   634924   650581   662328
```

```
## 10   12          Florida 18845537 19053237 19297822 19545621 19845911
## 11   13          Georgia  9711881  9802431  9901430  9972479 10067278
## 12   15           Hawaii  1363963  1379329  1394804  1408243  1414538
## 13   16            Idaho  1570746  1583910  1595324  1611206  1631112
## 14   17         Illinois 12840503 12867454 12882510 12895129 12884493
## 15   18          Indiana  6490432  6516528  6537703  6568713  6593644
## 16   19             Iowa  3050745  3066336  3076190  3092997  3109350
## 17   20           Kansas  2858190  2869225  2885257  2893212  2900475
## 18   21         Kentucky  4348181  4369821  4386346  4404659  4414349
## 19   22        Louisiana  4544532  4575625  4600972  4624527  4644013
## 20   23            Maine  1327629  1328284  1327729  1328009  1330513
## 21   24         Maryland  5788645  5839419  5886992  5923188  5957283
## 22   25    Massachusetts  6566307  6613583  6663005  6713315  6762596
## 23   26         Michigan  9877510  9882412  9897145  9913065  9929848
## 24   27        Minnesota  5310828  5346143  5376643  5413479  5451079
## 25   28      Mississippi  2970548  2978731  2983816  2988711  2990468
## 26   29         Missouri  5995974  6010275  6024367  6040715  6056202
## 27   30          Montana   990697   997316  1003783  1013569  1021869
## 28   31         Nebraska  1829542  1840672  1853303  1865279  1879321
## 29   32           Nevada  2702405  2712730  2743996  2775970  2817628
## 30   33    New Hampshire  1316762  1320202  1324232  1326622  1333341
## 31   34       New Jersey  8799446  8828117  8844942  8856972  8864525
## 32   35       New Mexico  2064552  2080450  2087309  2092273  2089568
## 33   36         New York 19399878 19499241 19572932 19624447 19651049
## 34   37   North Carolina  9574323  9657592  9749476  9843336  9932887
## 35   38     North Dakota   674715   685225   701176   722036   737401
## 36   39             Ohio 11539336 11544663 11548923       NA 11602700
## 37   40         Oklahoma  3759944  3788379  3818814  3853214  3878187
## 38   41           Oregon  3837491  3872036  3899001  3922468  3963244
## 39   42     Pennsylvania 12711160 12745815 12767118 12776309 12788313
## 40   44     Rhode Island  1053959  1053649  1054621  1055081  1055936
## 41   45   South Carolina  4635649  4671994  4717354  4764080  4823617
## 42   46     South Dakota   816166   823579   833566   842316   849129
## 43   47        Tennessee  6355311  6399291  6453898  6494340  6541223
## 44   48            Texas 25241971 25645629 26084481 26480266 26964333
## 45   49             Utah  2775332  2814384  2853375  2897640  2936879
## 46   50          Vermont   625879   627049   626090   626210   625214
## 47   51         Virginia  8023699  8101155  8185080  8252427  8310993
## 48   53       Washington  6742830  6826627  6897058  6963985  7054655
## 49   54    West Virginia  1854239  1856301  1856872  1853914  1849489
```

```
## 50   55         Wisconsin  5690475  5705288  5719960  5736754  5751525
## 51   56           Wyoming   564487   567299   576305   582122   582531
## 52   72       Puerto Rico  3721525  3678732  3634488  3593077  3534874
##        2015     2016     2017     2018     2019
## 1   4852347  4863525  4874486  4887681  4903185
## 2    737498   741456   739700   735139   731545
## 3   6829676  6941072  7044008  7158024  7278717
## 4   2978048  2989918  3001345  3009733  3017804
## 5  38918045 39167117 39358497 39461588 39512223
## 6   5450623  5539215  5611885  5691287  5758736
## 7   3587122  3578141  3573297  3571520  3565287
## 8    941252   948921   956823   965479   973764
## 9    675400   685815   694906   701547   705749
## 10 20209042 20613477 20963613 21244317 21477737
## 11 10178447 10301890 10410330 10511131 10617423
## 12  1422052  1427559  1424393  1420593  1415872
## 13       NA  1682380  1717715  1750536  1787065
## 14 12858913 12820527 12778828 12723071 12671821
## 15  6608422  6634304  6658078  6695497  6732219
## 16  3120960  3131371  3141550  3148618  3155070
## 17  2909011  2910844  2908718  2911359  2913314
## 18  4425976  4438182  4452268  4461153  4467673
## 19  4664628  4678135  4670560  4659690  4648794
## 20  1328262  1331317  1334612  1339057  1344212
## 21  5985562  6003323  6023868  6035802  6045680
## 22  6794228  6823608  6859789  6882635  6892503
## 23  9931715  9950571  9973114  9984072  9986857
## 24  5482032  5522744  5566230  5606249  5639632
## 25  2988471  2987938  2988510  2981020  2976149
## 26  6071732  6087135  6106670  6121623  6137428
## 27  1030475  1040859       NA  1060665  1068778
## 28  1891277  1905616  1915947  1925614  1934408
## 29  2866939  2917563  2969905  3027341  3080156
## 30  1336350  1342307  1348787  1353465  1359711
## 31  8867949  8870827  8885525  8886025  8882190
## 32  2089291  2091630  2091784  2092741  2096829
## 33 19654666 19633428 19589572 19530351 19453561
## 34 10031646 10154788 10268233 10381615 10488084
## 35   754066   754434   754942   758080   762062
## 36 11617527 11634370 11659650 11676341 11689100
```

```
## 37  3909500  3926331  3931316  3940235  3956971
## 38  4015792  4089976  4143625  4181886  4217737
## 39 12784826 12782275 12787641 12800922 12801989
## 40  1056065  1056770  1055673  1058287  1059361
## 41  4891938  4957968  5021268  5084156  5148714
## 42   853988   862996   872868   878698   884659
## 43  6591170  6646010  6708799  6771631  6829174
## 44 27470056 27914410 28295273 28628666 28995881
## 45  2981835  3041868  3101042  3153550  3205958
## 46   625216   623657   624344   624358   623989
## 47  8361808  8410106  8463587  8501286  8535519
## 48  7163657  7294771  7423362  7523869  7614893
## 49  1842050  1831023  1817004  1804291  1792147
## 50  5760940  5772628  5790186  5807406       NA
## 51   585613   584215   578931   577601   578759
## 52  3473232  3406672  3325286  3193354  3193694
```

Question-3

```
which(is.na(Population_Data))

## [1] 159 296 377 495 622
```

Finding which state has NA value in a respective column.

```
Population_Data$NAME.x[is.na(Population_Data$'2011')]

## [1] "Arizona"
```

pulling out states of respective years with missing values

```
State_2011 <- Population_Data$NAME.x[is.na(Population_Data$'2011')]
State_2013 <- Population_Data$NAME.x[is.na(Population_Data$'2013')]
State_2015 <- Population_Data$NAME.x[is.na(Population_Data$'2015')]
State_2017 <- Population_Data$NAME.x[is.na(Population_Data$'2017')]
State_2019 <- Population_Data$NAME.x[is.na(Population_Data$'2019')]
```

Replacing missing values by mean of surrounding years

```
Population_Data$'2011'[is.na(Population_Data$'2011')]<-
mean(c(Population_Data$'2010'[Population_Data$NAME.x==State_2011],Population_Data
$'2012'[Population_Data$NAME.x==State_2011]))
```

```r
Population_Data$'2013'[is.na(Population_Data$'2013')]<-
mean(c(Population_Data$'2012'[Population_Data$NAME.x==State_2013],Population_Data
$'2014'[Population_Data$NAME.x==State_2013]))
Population_Data$'2015'[is.na(Population_Data$'2015')]<-
mean(c(Population_Data$'2014'[Population_Data$NAME.x==State_2015],Population_Data
$'2016'[Population_Data$NAME.x==State_2015]))
Population_Data$'2017'[is.na(Population_Data$'2017')]<-
mean(c(Population_Data$'2016'[Population_Data$NAME.x==State_2017],Population_Data
$'2018'[Population_Data$NAME.x==State_2017]))
Population_Data$'2019'[is.na(Population_Data$'2019')]<-
mean(c(Population_Data$'2018'[Population_Data$NAME.x==State_2019],Population_Data
$'2017'[Population_Data$NAME.x==State_2019]))
```

Question-4

```r
library(tidyverse)
```

A) Finding each state's maximum annual population:

```r
max_population <- Population_Data %>%
rowwise() %>%
mutate(max_population = max(c_across(starts_with("201")), na.rm = TRUE)) %>%
select(NAME.x, max_population)
head(max_population)

## # A tibble: 6 × 2
## # Rowwise:
##   NAME.x     max_population
##   <chr>              <dbl>
## 1 Alabama          4903185
## 2 Alaska          741456
## 3 Arizona         7278717
## 4 Arkansas         3017804
## 5 California      39512223
## 6 Colorado         5758736
```

B)Finding the total population for each state throughout all years:

```r
sum_population <- Population_Data %>%
rowwise() %>%
mutate(sum_population = sum(c_across(starts_with("201")), na.rm = TRUE)) %>%
```

```r
select(NAME.x, sum_population)
head(sum_population)
```

```
## # A tibble: 6 × 2
## # Rowwise:
##   NAME.x     sum_population
##   <chr>               <dbl>
## 1 Alabama          48453198
## 2 Alaska            7325170
## 3 Arizona          68057899
## 4 Arkansas         29738435
## 5 California      386181900
## 6 Colorado         54031986
```

Refer last column to see max population and total population state wise. We just replaced max function by sum function to get sum of the population.

Question-5:

Get the total US population for one single year

```r
Total_Population_2024 <- sum(Population_Data$'2014')
Total_Population_2024
```

```
## [1] 321835882
```

#PROBLEM SET 3

Reshaping the data

```
Population_Data_New <- Population_Data %>%
pivot_longer(cols = starts_with("20"), names_to = "year", values_to = "population") %>%
mutate(year = as.integer(str_extract(year, "\\d+")))
```

Need to select 4 states

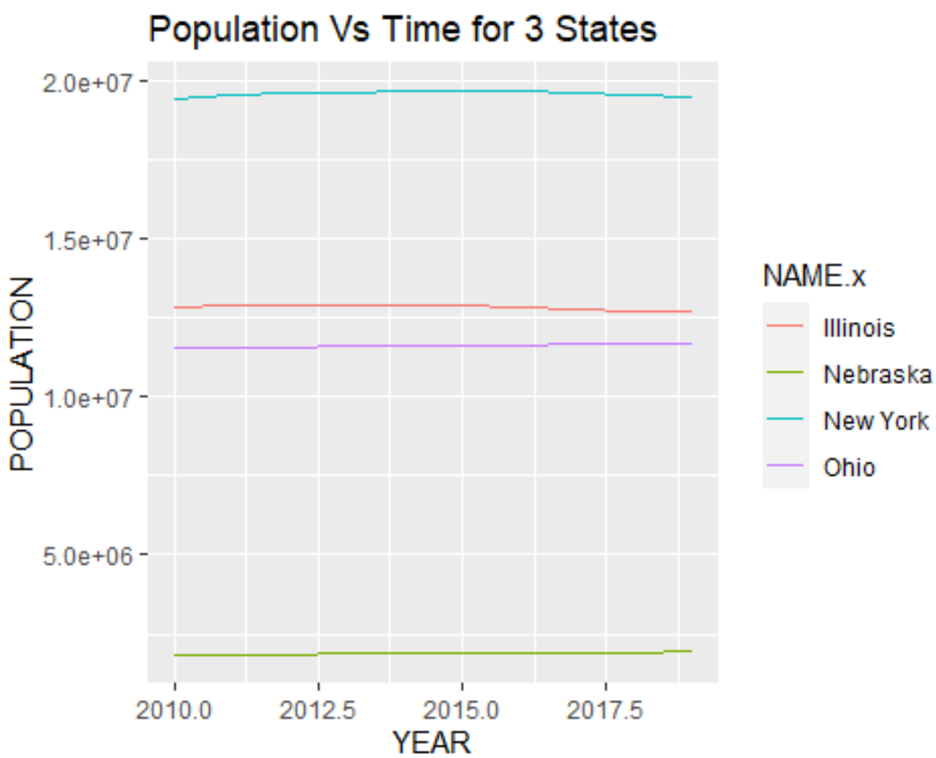
```
States <- c("Nebraska", "Ohio", "Illinois","New York")
```

Filter the data for the chosen states.

```
Data_states <- Population_Data_New %>% filter(NAME.x %in% States)

library(ggplot2)
```

Construct a line graph.

```
ggplot(Data_states, aes(x = year, y = population, color = NAME.x)) +
geom_line() +
labs(title = "Population Vs Time for 3 States",
x = "YEAR",
y = "POPULATION")
```



Data_states

```
## # A tibble: 40 × 4
##    STATE NAME.x    year population
##    <int> <chr>    <int>      <dbl>
##  1    17 Illinois  2010   12840503
##  2    17 Illinois  2011   12867454
##  3    17 Illinois  2012   12882510
##  4    17 Illinois  2013   12895129
##  5    17 Illinois  2014   12884493
##  6    17 Illinois  2015   12858913
##  7    17 Illinois  2016   12820527
##  8    17 Illinois  2017   12778828
##  9    17 Illinois  2018   12723071
## 10    17 Illinois  2019   12671821
## # ℹ 30 more rows
```

PROBLEM SET- 4

Question A :

#Describe two ways in which data can be dirty, and for each one, provide a potential solution.

–>Dirty Data and Potential Solutions

1)Missing Values

Problem –>Missing data can make analysis biased or incomplete, it occurs when when a dataset has certain values that are either incomplete or not recorded at all.

Solution –>Filling in missing values can be done using imputation techniques like mean imputation or model-based imputation.

2)Outliers

Problem - Outliers may have an impact on the results of statistical modeling and data analysis. It may have significant effects on the mean and standard deviation.

Solution -First, identify outliers using IQR approach and The method of standard deviation. Outliers can be handled in numerous ways, including capping, removing them, or considering them as missing numbers.

Question B :

i)Customers who make similar purchases can be grouped together using clustering techniques. K-Means clustering, KNN, etc.

ii)Classification algorithms, like decision trees or logistic regression, are used to predict binary outcomes, such as if a customer is willing to buy milk or not.

iii)An approach known as Association Rule Mining is frequently used in the field of data mining to find groups of products that are frequently purchased together. The goal of association rule mining is to find intriguing connections or patterns in huge datasets.

Question C :

a. Organizing the customers of a company according to education level. -> Yes, this activity involves data mining because we are classifying clients based on criteria like education level. From it, we can derive results like which level of education attracts the most customers, etc.

b. Computing the total sales of a company. -> As we are not discovering any patterns, this is not a data mining activity.

c. Sorting a student database according to identification numbers. -> No. This is not a data mining task as we are only ordering the data here.

d. Predicting the outcomes of tossing a (fair) pair of dice. -> Although it is a process of making a prediction, we cannot call it a data mining process because no relationships or patterns are being found. We are merely forecasting some of the results.

e. Predicting the future stock price of a company using historical records. -> Yes, this is a data mining process since we must first analyze historical data and, if necessary, perform extensive preprocessing on the data. In order to decide whether or not to maintain certain variables, we must also identify patterns and relationships between them. The future stock price can thus be predicted. ## R Markdown