Group2

2024-02-12

Q1: Print the structure of your dataset

```
Billionaires_Statistics<- readxl::read_excel("Billionaires_Statistics.xlsx")
str(Billionaires_Statistics)
```

```
## tibble [2,640 × 35] (S3: tbl_df/tbl/data.frame)
##  $ rank                                    : num [1:2640] 1 2 3 4 5 6 7
8 9 10 ...
##  $ finalWorth                              : num [1:2640] 211000 180000
114000 107000 106000 104000 94500 93000 83400 80700 ...
##  $ category                                : chr [1:2640] "Fashion &
Retail" "Automotive" "Technology" "Technology" ...
##  $ personName                              : chr [1:2640] "Bernard
Arnault & family" "Elon Musk" "Jeff Bezos" "Larry Ellison" ...
##  $ age                                     : num [1:2640] 74 51 59 78 92
67 81 83 65 67 ...
##  $ country                                 : chr [1:2640] "France"
"United States" "United States" "United States" ...
##  $ city                                    : chr [1:2640] "Paris"
"Austin" "Medina" "Lanai" ...
##  $ source                                  : chr [1:2640] "LVMH" "Tesla,
SpaceX" "Amazon" "Oracle" ...
##  $ industries                              : chr [1:2640] "Fashion &
Retail" "Automotive" "Technology" "Technology" ...
##  $ countryOfCitizenship                    : chr [1:2640] "France"
"United States" "United States" "United States" ...
##  $ organization                            : chr [1:2640] "LVMH Moët
Hennessy Louis Vuitton" "Tesla" "Amazon" "Oracle" ...
##  $ selfMade                                : logi [1:2640] FALSE TRUE
TRUE TRUE TRUE TRUE ...
##  $ status                                  : chr [1:2640] "U" "D" "D"
"U" ...
##  $ gender                                  : chr [1:2640] "M" "M" "M"
"M" ...
##  $ birthDate                               : POSIXct[1:2640], format:
"1949-03-05 00:00:00" "1971-06-28 00:00:00" ...
##  $ lastName                                : chr [1:2640] "Arnault"
"Musk" "Bezos" "Ellison" ...
##  $ firstName                               : chr [1:2640] "Bernard"
"Elon" "Jeff" "Larry" ...
##  $ title                                   : chr [1:2640] "Chairman and
CEO" "CEO" "Chairman and Founder" "CTO and Founder" ...
##  $ date                                    : POSIXct[1:2640], format:
```

```
"2023-04-04 05:01:00" "2023-04-04 05:01:00" ...
##  $ state                                     : chr [1:2640] NA "Texas"
"Washington" "Hawaii" ...
##  $ residenceStateRegion                      : chr [1:2640] NA "South"
"West" "West" ...
##  $ birthYear                                 : num [1:2640] 1949 1971 1964
1944 1930 ...
##  $ birthMonth                                : num [1:2640] 3 6 1 8 8 10 2
1 4 3 ...
##  $ birthDay                                  : num [1:2640] 5 28 12 17 30
28 14 28 19 24 ...
##  $ cpi_country                               : num [1:2640] 110 117 117
117 117 ...
##  $ cpi_change_country                        : num [1:2640] 1.1 7.5 7.5
7.5 7.5 7.5 7.5 3.6 7.7 7.5 ...
##  $ gdp_country                               : num [1:2640] 2.72e+12
2.14e+13 2.14e+13 2.14e+13 2.14e+13 ...
##  $ gross_tertiary_education_enrollment       : num [1:2640] 65.6 88.2 88.2
88.2 88.2 88.2 88.2 40.2 28.1 88.2 ...
##  $ gross_primary_education_enrollment_country: num [1:2640] 102 102 102
102 102 ...
##  $ life_expectancy_country                   : num [1:2640] 82.5 78.5 78.5
78.5 78.5 78.5 78.5 75 69.4 78.5 ...
##  $ tax_revenue_country_country               : num [1:2640] 24.2 9.6 9.6
9.6 9.6 9.6 9.6 13.1 11.2 9.6 ...
##  $ total_tax_rate_country                    : num [1:2640] 60.7 36.6 36.6
36.6 36.6 36.6 36.6 55.1 49.7 36.6 ...
##  $ population_country                        : num [1:2640] 6.71e+07
3.28e+08 3.28e+08 3.28e+08 3.28e+08 ...
##  $ latitude_country                          : num [1:2640] 46.2 37.1 37.1
37.1 37.1 ...
##  $ longitude_country                         : num [1:2640] 2.21 -95.71 -
95.71 -95.71 -95.71 ...
```

Q2: List the variables in your dataset

```
names(Billionaires_Statistics)
```

```
##  [1] "rank"
##  [2] "finalWorth"
##  [3] "category"
##  [4] "personName"
##  [5] "age"
##  [6] "country"
##  [7] "city"
##  [8] "source"
##  [9] "industries"
## [10] "countryOfCitizenship"
## [11] "organization"
## [12] "selfMade"
```

https://github.com/Timmylicious/Billionaire_Statistics_Group2

```
## [13] "status"
## [14] "gender"
## [15] "birthDate"
## [16] "lastName"
## [17] "firstName"
## [18] "title"
## [19] "date"
## [20] "state"
## [21] "residenceStateRegion"
## [22] "birthYear"
## [23] "birthMonth"
## [24] "birthDay"
## [25] "cpi_country"
## [26] "cpi_change_country"
## [27] "gdp_country"
## [28] "gross_tertiary_education_enrollment"
## [29] "gross_primary_education_enrollment_country"
## [30] "life_expectancy_country"
## [31] "tax_revenue_country_country"
## [32] "total_tax_rate_country"
## [33] "population_country"
## [34] "latitude_country"
## [35] "longitude_country"
```

#Q3: Print the top 15 rows of your dataset

```
head(Billionaires_Statistics, 15)
```

```
## # A tibble: 15 × 35
##     rank finalWorth category    personName   age country city  source
industries
##    <dbl>      <dbl> <chr>       <chr>      <dbl> <chr>   <chr> <chr>
<chr>
##  1     1     211000 Fashion & … Bernard A…    74 France  Paris LVMH
Fashion &…
##  2     2     180000 Automotive  Elon Musk     51 United… Aust… Tesla…
Automotive
##  3     3     114000 Technology  Jeff Bezos    59 United… Medi… Amazon
Technology
##  4     4     107000 Technology  Larry Ell…    78 United… Lanai Oracle
Technology
##  5     5     106000 Finance & … Warren Bu…    92 United… Omaha Berks…
Finance &…
##  6     6     104000 Technology  Bill Gates    67 United… Medi… Micro…
Technology
##  7     7      94500 Media & En… Michael B…    81 United… New … Bloom…
Media & E…
##  8     8      93000 Telecom     Carlos Sl…    83 Mexico  Mexi… Telec…
Telecom
##  9     9      83400 Diversified Mukesh Am…    65 India   Mumb… Diver…
```

```
Diversifi…
## 10    10      80700 Technology  Steve Bal…    67 United… Hunt… Micro…
Technology
## 11    11      80500 Fashion & … Francoise…    69 France  Paris L'Oré…
Fashion &…
## 12    12      79200 Technology  Larry Page    50 United… Palo… Google
Technology
## 13    13      77300 Fashion & … Amancio O…    87 Spain   La C… Zara
Fashion &…
## 14    14      76000 Technology  Sergey Br…    49 United… Los … Google
Technology
## 15    15      68000 Food & Bev… Zhong Sha…    68 China   Hang… Bever… Food
& Be…
## # ℹ 26 more variables: countryOfCitizenship <chr>, organization <chr>,
## #   selfMade <lgl>, status <chr>, gender <chr>, birthDate <dttm>,
## #   lastName <chr>, firstName <chr>, title <chr>, date <dttm>, state
<chr>,
## #   residenceStateRegion <chr>, birthYear <dbl>, birthMonth <dbl>,
## #   birthDay <dbl>, cpi_country <dbl>, cpi_change_country <dbl>,
## #   gdp_country <dbl>, gross_tertiary_education_enrollment <dbl>,
## #   gross_primary_education_enrollment_country <dbl>, …
```

Q4: Write a user defined function using any of the variables from the data set.

```
BillionaireMean = mean(Billionaires_Statistics$rank)
Square = function(BillionaireMean){BillionaireMean^2}
```

Q5: Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset.

```
library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

filtered_data <- Billionaires_Statistics %>% filter(finalWorth > 10000)
```

Q6: Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset.

```
dependent_vars <- Billionaires_Statistics$countryOfCitizenship
independent_vars <- Billionaires_Statistics[, c("birthMonth", "birthDay")]
billionairescountry <- cbind(dependent_vars,independent_vars)
View(billionairescountry)
billionairescountry = as.data.frame(billionairescountry)
```

Q7: Remove missing values in your dataset

```
Billionaires_Statistics <- na.omit(Billionaires_Statistics)
head(Billionaires_Statistics)
```

```
## # A tibble: 6 × 35
##    rank finalWorth category     personName   age country city  source
industries
##   <dbl>      <dbl> <chr>        <chr>      <dbl> <chr>   <chr> <chr>
<chr>
## 1     2     180000 Automotive   Elon Musk     51 United… Aust… Tesla…
Automotive
## 2     3     114000 Technology   Jeff Bezos    59 United… Medi… Amazon
Technology
## 3     4     107000 Technology   Larry Ell…    78 United… Lanai Oracle
Technology
## 4     5     106000 Finance & I… Warren Bu…    92 United… Omaha Berks…
Finance &…
## 5     6     104000 Technology   Bill Gates    67 United… Medi… Micro…
Technology
## 6     7      94500 Media & Ent… Michael B…    81 United… New … Bloom…
Media & E…
## # ℹ 26 more variables: countryOfCitizenship <chr>, organization <chr>,
## #   selfMade <lgl>, status <chr>, gender <chr>, birthDate <dttm>,
## #   lastName <chr>, firstName <chr>, title <chr>, date <dttm>, state
<chr>,
## #   residenceStateRegion <chr>, birthYear <dbl>, birthMonth <dbl>,
## #   birthDay <dbl>, cpi_country <dbl>, cpi_change_country <dbl>,
## #   gdp_country <dbl>, gross_tertiary_education_enrollment <dbl>,
## #   gross_primary_education_enrollment_country <dbl>, …
```

#Q8: Identify and remove duplicated data in your dataset

```
duplicated(billionairescountry)
```

```
##    [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
```

https://github.com/Timmylicious/Billionaire_Statistics_Group2

```
##   [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
TRUE
##   [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [49] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [73] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##   [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [109]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
FALSE
##  [121]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
##  [133] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
FALSE
##  [145] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE
FALSE
##  [157] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
TRUE
##  [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [181]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [193] FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [217] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
##  [229] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
FALSE
##  [241]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
FALSE
##  [253] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [265] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [289] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [301]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
FALSE
##  [313] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
```

```
##  [325]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
FALSE
##  [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
##  [349]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
FALSE
##  [361] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
TRUE
##  [373] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
##  [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
TRUE
##  [409] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
FALSE
##  [421]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
FALSE
##  [433] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
TRUE
##  [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [457]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
##  [469] FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
FALSE
##  [481] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
FALSE
##  [493] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [505] FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [517] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [529]  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
FALSE
##  [541] FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [553] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [565] FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
##  [577] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [589] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
TRUE
##  [601] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
FALSE
##  [613] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
FALSE
```

```
##  [625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
FALSE
##  [637] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [649]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
FALSE
##  [661]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [673]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [685] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
FALSE
##  [697]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
TRUE
##  [709] FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
TRUE
##  [721]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
FALSE
##  [733] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
##  [745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
TRUE
##  [757] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
TRUE
##  [769] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
TRUE
##  [781]  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
##  [793] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
FALSE
##  [805] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
FALSE
##  [817]  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
FALSE
##  [829] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
FALSE
##  [841] FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
TRUE
##  [853] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
FALSE
##  [865] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
##  [877] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
TRUE
##  [889] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
##  [901]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
FALSE
##  [913] FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
```

```
##  [925]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
TRUE
##  [937] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
TRUE
##  [949] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
FALSE
##  [961] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
TRUE
##  [973] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
##  [985] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
TRUE
##  [997] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE
TRUE
## [1009] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
TRUE
## [1021] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
FALSE
## [1033] FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [1045]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [1057] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
TRUE
## [1069] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
FALSE
## [1081]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
FALSE
## [1093]  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
FALSE
## [1105] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
TRUE
## [1117] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [1129] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
FALSE
## [1141] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
TRUE
## [1153]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
TRUE
## [1165]  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
TRUE
## [1177] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
TRUE
## [1189] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
FALSE
## [1201] FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
TRUE
## [1213]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
FALSE
```

```
## [1225]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
FALSE
## [1237] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
TRUE
## [1249]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
## [1261] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
TRUE
## [1273] FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [1285] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
FALSE
## [1297] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
TRUE
## [1309]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
FALSE
## [1321]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
TRUE
## [1333] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
FALSE
## [1345] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
FALSE
## [1357]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
TRUE
## [1369]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
FALSE
## [1381]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
TRUE
## [1393] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
## [1405]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
FALSE
## [1417]  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
TRUE
## [1429]  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
TRUE
## [1441] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
TRUE
## [1453] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
FALSE
## [1465] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
TRUE
## [1477]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
FALSE
## [1489]  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
FALSE
## [1501]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
TRUE
## [1513] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
TRUE
```

```
## [1525] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
TRUE
## [1537] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
FALSE
## [1549] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
## [1561] FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
TRUE
## [1573]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE
TRUE
## [1585]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [1597] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
TRUE
## [1609] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
FALSE
## [1621] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
TRUE
## [1633]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
FALSE
## [1645]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
FALSE
## [1657] FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [1669]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
## [1681] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
TRUE
## [1693] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [1705] FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
TRUE
## [1717] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
FALSE
## [1729] FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
TRUE
## [1741] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
## [1753]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
TRUE
## [1765]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
FALSE
## [1777]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
## [1789]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
TRUE
## [1801]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
FALSE
## [1813]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
TRUE
```

```
## [1825] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
TRUE
## [1837]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
TRUE
## [1849] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [1861] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
TRUE
## [1873]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
TRUE
## [1885]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
TRUE
## [1897]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
FALSE
## [1909] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
## [1921]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
TRUE
## [1933] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE
## [1945] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [1957]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
TRUE
## [1969] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
FALSE
## [1981] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
FALSE
## [1993]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
FALSE
## [2005]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
TRUE
## [2017]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [2029]  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
FALSE
## [2041] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
TRUE
## [2053] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
TRUE
## [2065] FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [2077]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
TRUE
## [2089]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
## [2101]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
TRUE
## [2113] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
TRUE
```

```
## [2125]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
FALSE
## [2137] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
TRUE
## [2149]  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
FALSE
## [2161] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
TRUE
## [2173] FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
TRUE
## [2185]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
FALSE
## [2197]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
TRUE
## [2209] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
FALSE
## [2221] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
TRUE
## [2233] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
TRUE
## [2245]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
TRUE
## [2257] FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
TRUE
## [2269]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
TRUE
## [2281] FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [2293] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
FALSE
## [2305]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
FALSE
## [2317] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
FALSE
## [2329] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
FALSE
## [2341] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
FALSE
## [2353] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
FALSE
## [2365]  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
FALSE
## [2377] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
FALSE
## [2389]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
TRUE
## [2401]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
FALSE
## [2413]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
TRUE
```

```
## [2425]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
FALSE
## [2437] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
FALSE
## [2449]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
FALSE
## [2461] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [2473] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
FALSE
## [2485] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE
FALSE
## [2497]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
FALSE
## [2509]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE
TRUE
## [2521]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
FALSE
## [2533]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
TRUE
## [2545]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
TRUE
## [2557]  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE
TRUE
## [2569]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
TRUE
## [2581] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
TRUE
## [2593] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
TRUE
## [2605] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
FALSE
## [2617] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
FALSE
## [2629]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
FALSE
```

```
billionairescountry[!duplicated(billionairescountry$dependent_vars), ]
```

```
##            dependent_vars birthMonth birthDay
## 1                  France          3        5
## 2           United States          6       28
## 8                  Mexico          1       28
## 9                   India          4       19
## 13                  Spain          3       28
## 15                  China         12        1
## 22                 Canada          6       12
## 27                Germany          9       24
## 30                  Italy          9       21
## 33              Hong Kong          6       13
```

```
## 37                Austria          5        7
## 39                  Japan          2        7
## 43            Switzerland          6       27
## 52              Australia          2        9
## 56              Indonesia          4       17
## 58                 Russia          3        8
## 65                  Chile          1        1
## 67         United Kingdom          1        1
## 86                 Israel          6        2
## 100                Brazil          1        1
## 102        Czech Republic          7        4
## 103             Singapore          1        1
## 104                Sweden         10        4
## 116              Thailand          4       19
## 119           Netherlands          6       30
## 125               Nigeria          4       10
## 130                Cyprus          2        1
## 146              Malaysia         10        6
## 148  United Arab Emirates         10       10
## 158          South Africa          6        1
## 233           New Zealand          6        6
## 237           Philippines         12       13
## 247                Monaco          6        4
## 267               Belgium         10        1
## 271           South Korea          6       23
## 278                Taiwan          8        1
## 289                Norway          6       23
## 310                 Egypt          1       19
## 314               Denmark         11        4
## 340               Ireland          9        6
## 392  Eswatini (Swaziland)          1        6
## 403              Colombia          1       27
## 417                Poland          7       11
## 445               Ukraine          9       21
## 455                Greece          2       18
## 499                Turkey          9       26
## 555             Argentina         NA       NA
## 558               Georgia          2       18
## 580              Portugal          1       16
## 596            Kazakhstan         10       29
## 605               Algeria          1        1
## 638             Venezuela          7       23
## 651               Vietnam          8        5
## 686               Finland         11       14
## 727                Belize          4       21
## 1084              Lebanon         11       24
## 1106                 Oman          1        1
## 1222              Iceland          3       19
## 1334             Guernsey          8       30
## 1366        Liechtenstein          5        4
```

```
## 1447             Bulgaria          8        6
## 1565              Romania          5        9
## 1616             Zimbabwe          1       29
## 1649                Qatar          1        1
## 1657                Nepal          4       14
## 1818             Slovakia          6       27
## 1906              Morocco          1        1
## 1922              Hungary          3       20
## 1926             Tanzania          5        8
## 1972                 Peru         NA       NA
## 2096             Barbados          2       20
## 2173                Macau          9        1
## 2190              Estonia          9        1
## 2248  St. Kitts and Nevis          1        1
## 2376              Armenia          5       25
## 2589           Bangladesh          3        1
## 2608               Panama          6       15
```

#Q9: Reorder multiple rows in descending order

```
d_order <- Billionaires_Statistics[order(-Billionaires_Statistics$age),]
print(d_order)
```

```
## # A tibble: 238 × 35
##     rank finalWorth category    personName   age country city  source
industries
##    <dbl>      <dbl> <chr>       <chr>      <dbl> <chr>   <chr> <chr>
<chr>
##  1  1368       2200 Finance & … Charles M…    99 United… Los … Berks…
Finance &…
##  2  1217       2500 Food & Bev… S. Daniel…    98 United… Palm… Slim-… Food
& Be…
##  3   591       4600 Media & En… Charles D…    96 United… Oyst… Cable…
Media & E…
##  4   261       8000 Fashion & … Bernard M…    93 United… Atla… Home …
Fashion &…
##  5     5     106000 Finance & … Warren Bu…    92 United… Omaha Berks…
Finance &…
##  6    99      17100 Media & En… Rupert Mu…    92 United… New … Newsp…
Media & E…
##  7   365       6700 Finance & … George So…    92 United… Kato… Hedge…
Finance &…
##  8  1272       2400 Fashion & … Doris Fis…    91 United… San … Gap
Fashion &…
##  9    77      21000 Fashion & … Leonard L…    90 United… New … Estee…
Fashion &…
## 10    97      17400 Real Estate Donald Br…    90 United… Newp… Real … Real
Esta…
## # ℹ 228 more rows
## # ℹ 26 more variables: countryOfCitizenship <chr>, organization <chr>,
```

```
## #   selfMade <lgl>, status <chr>, gender <chr>, birthDate <dttm>,
## #   lastName <chr>, firstName <chr>, title <chr>, date <dttm>, state
<chr>,
## #   residenceStateRegion <chr>, birthYear <dbl>, birthMonth <dbl>,
## #   birthDay <dbl>, cpi_country <dbl>, cpi_change_country <dbl>,
## #   gdp_country <dbl>, gross_tertiary_education_enrollment <dbl>, …

nana <- Billionaires_Statistics[order(-Billionaires_Statistics$rank),]
print(nana)

## # A tibble: 238 × 35
##     rank finalWorth category    personName   age country city  source
industries
##    <dbl>      <dbl> <chr>       <chr>      <dbl> <chr>   <chr> <chr>
<chr>
##  1  2540       1000 Fashion & … William F…    66 United… San … Gap
Fashion &…
##  2  2540       1000 Sports      LeBron Ja…    38 United… Los … Baske…
Sports
##  3  2540       1000 Technology  Apoorva M…    36 United… San … Groce…
Technology
##  4  2540       1000 Media & En… Tyler Per…    53 United… Atla… Movie…
Media & E…
##  5  2405       1100 Fashion & … Sara Blak…    52 United… Atla… Spanx
Fashion &…
##  6  2405       1100 Finance & … Lloyd Bla…    68 United… New … Banki…
Finance &…
##  7  2405       1100 Technology  Ryan Bres…    28 United… Miami E-com…
Technology
##  8  2405       1100 Technology  Weili Dai     61 United… Las … Semic…
Technology
##  9  2405       1100 Fashion & … Robert Fi…    69 United… San … Gap
Fashion &…
## 10  2405       1100 Technology  David Hin…    78 United… Seat… Softw…
Technology
## # ℹ 228 more rows
## # ℹ 26 more variables: countryOfCitizenship <chr>, organization <chr>,
## #   selfMade <lgl>, status <chr>, gender <chr>, birthDate <dttm>,
## #   lastName <chr>, firstName <chr>, title <chr>, date <dttm>, state
<chr>,
## #   residenceStateRegion <chr>, birthYear <dbl>, birthMonth <dbl>,
## #   birthDay <dbl>, cpi_country <dbl>, cpi_change_country <dbl>,
## #   gdp_country <dbl>, gross_tertiary_education_enrollment <dbl>, …
```

#Q10: Rename some of the column names in your dataset

```
colnames(Billionaires_Statistics) <- c("RankNew", "FinalWorthNew",
"CategoryNew", "PersonNameNew", "AgeNew", "CountryNew", "City", "Source",
"Industries", "CountryOfCitizenship", "Organization", "SelfMade", "Status",
"Gender", "BirthDate", "LastName", "FirstName","Title", "Date", "State",
```

```
"ResidenceStateRegion", "BirthYear", "BirthMonth", "BirthDay", "CPI_Country",
"CPI_Change_Country", "GDP_Country",
"GrossTertiaryEducationEnrollment","GrossPrimaryEducationEnrollmentCountry","
LifeExpectancyCountry", "TaxRevenueCountryCountry", "TotalTaxRateCountry",
"PopulationCountry", "LatitudeCountry", "LongitudeCountry")
```

#Q11: Add new variables in your data frame by using a mathematical function (for e.g. – multiply an existing column by 2 and add it as a new variable to your data frame)

```
Billionaires_Statistics$New_Variable <- Billionaires_Statistics$AgeNew * 2
head(Billionaires_Statistics)
```

```
## # A tibble: 6 × 36
##   RankNew FinalWorthNew CategoryNew PersonNameNew AgeNew CountryNew City
Source
##     <dbl>         <dbl> <chr>       <chr>          <dbl> <chr>      <chr>
<chr>
## 1       2        180000 Automotive  Elon Musk         51 United St… Aust…
Tesla…
## 2       3        114000 Technology  Jeff Bezos        59 United St… Medi…
Amazon
## 3       4        107000 Technology  Larry Ellison     78 United St… Lanai
Oracle
## 4       5        106000 Finance & … Warren Buffe…     92 United St… Omaha
Berks…
## 5       6        104000 Technology  Bill Gates        67 United St… Medi…
Micro…
## 6       7         94500 Media & En… Michael Bloo…     81 United St… New …
Bloom…
## # ℹ 28 more variables: Industries <chr>, CountryOfCitizenship <chr>,
## #   Organization <chr>, SelfMade <lgl>, Status <chr>, Gender <chr>,
## #   BirthDate <dttm>, LastName <chr>, FirstName <chr>, Title <chr>,
## #   Date <dttm>, State <chr>, ResidenceStateRegion <chr>, BirthYear <dbl>,
## #   BirthMonth <dbl>, BirthDay <dbl>, CPI_Country <dbl>,
## #   CPI_Change_Country <dbl>, GDP_Country <dbl>,
## #   GrossTertiaryEducationEnrollment <dbl>, …
```

#Q12: Create a training set using random number generator engine.

```
 set.seed(1234)
billionairescountry %>% sample_frac(0.05, replace = FALSE)
```

```
##     dependent_vars birthMonth birthDay
## 1            India          8       30
## 2            China          1       21
## 3   United Kingdom          1        1
## 4            China         11        1
## 5            India         11        9
```

```
## 6    United States          2        8
## 7             China          1        1
## 8             China          3        1
## 9    United States          5       15
## 10         Malaysia         NA       NA
## 11            Japan         12       15
## 12        Singapore          7        1
## 13   United States          1       14
## 14        Singapore          2       27
## 15           Brazil          7        1
## 16   United States          7       19
## 17   United States         11        6
## 18   United States          1       12
## 19            China          1        1
## 20   United States         10       12
## 21          Germany         10       23
## 22        Australia          9       13
## 23        Hong Kong          3        1
## 24   United States          3        2
## 25            China         11       10
## 26          Germany          1       26
## 27          Germany          1        1
## 28            China          1        1
## 29   United States          4       24
## 30   United States          6       25
## 31           Russia          9       26
## 32           Canada         NA       NA
## 33            China          1        1
## 34   United States          3       16
## 35          Austria          3       27
## 36           Canada          1        1
## 37            China         12       28
## 38           Brazil         12        8
## 39   United States         10        8
## 40            China          1        1
## 41   United States          2        2
## 42            China          2       13
## 43          Germany          2       11
## 44          Germany          7       13
## 45   United States          5       23
## 46           Canada          1        1
## 47           Russia          9       25
## 48          Germany         11       22
## 49           Russia          1       29
## 50           Canada          8        4
## 51            India          7        5
## 52   United States          6        8
## 53           Brazil         11       30
## 54            China          1        1
## 55   United States          7        4
```

```
## 56           Japan          2       25
## 57           India          5       17
## 58   United States         12       21
## 59          Canada          9       10
## 60   United States          8       28
## 61   United States          2       18
## 62   United States         12       30
## 63   United States          5       30
## 64          Brazil          1        1
## 65         Austria          1        3
## 66           China          8        1
## 67           India          7       25
## 68       Singapore         11        5
## 69          Russia         12       15
## 70           India          1       18
## 71   United States         12        9
## 72     Switzerland          7       22
## 73           Chile          4        5
## 74   United States         12       10
## 75   United States          3       31
## 76  Czech Republic          7        4
## 77   United States          1        1
## 78   United States         11       27
## 79           India          4       17
## 80          Taiwan          5       25
## 81           China          1        1
## 82       Singapore         12        1
## 83        Malaysia         NA       NA
## 84           China          1        1
## 85   United States         12       29
## 86           China          1        1
## 87   United States          1       27
## 88          Mexico         10       19
## 89           China          1        1
## 90           China          7       31
## 91           China         11        6
## 92   United States          5        1
## 93     South Korea          1       11
## 94   United States          6        1
## 95   United States          2       29
## 96  United Kingdom         NA       NA
## 97           China          7       26
## 98           Japan          8       24
## 99           China          5        1
## 100  United States          5       31
## 101  United States          8       21
## 102          China          1        1
## 103  United States          1        1
## 104         Sweden         12        8
## 105  United States          2       26
```

```
## 106          China          5       18
## 107          China          7        1
## 108          Japan          3        5
## 109        Germany         NA       NA
## 110          India         12       10
## 111         Israel          1        1
## 112  United States          1        6
## 113         Mexico          8       13
## 114         Canada          3       29
## 115         Russia         11       20
## 116          China          9       16
## 117         Taiwan          4       21
## 118  United States          2       20
## 119      Indonesia          1       12
## 120           Oman         11       17
## 121          China         12       14
## 122    Switzerland          1        1
## 123  United States          9       29
## 124         Israel          9       24
## 125          India         10       10
## 126         Russia         10       24
## 127  United States         11       14
## 128  United States         10       12
## 129         Taiwan          1        1
## 130         Canada          6        2
## 131  United States          7       16
## 132          China          9        1
```

#Q13: Print the summary statistics of your dataset

```
summary(Billionaires_Statistics)
```

```
##     RankNew       FinalWorthNew    CategoryNew        PersonNameNew
##  Min.   :   2.0   Min.   :  1000   Length:238         Length:238
##  1st Qu.: 292.2   1st Qu.:  2100   Class :character   Class :character
##  Median : 748.0   Median :  3800   Mode  :character   Mode  :character
##  Mean   : 905.3   Mean   : 10638
##  3rd Qu.:1434.0   3rd Qu.:  7575
##  Max.   :2540.0   Max.   :180000
##      AgeNew       CountryNew           City              Source
##  Min.   :28.00   Length:238         Length:238         Length:238
##  1st Qu.:56.25   Class :character   Class :character   Class :character
##  Median :67.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :66.34
##  3rd Qu.:78.00
##  Max.   :99.00
##   Industries        CountryOfCitizenship Organization        SelfMade
##  Length:238         Length:238           Length:238         Mode :logical
##  Class :character   Class :character     Class :character   FALSE:40
##  Mode  :character   Mode  :character     Mode  :character   TRUE :198
```

```
##
##
##
##     Status             Gender            BirthDate
##  Length:238         Length:238         Min.   :1924-01-01 00:00:00.00
##  Class :character   Class :character   1st Qu.:1944-08-18 06:00:00.00
##  Mode  :character   Mode  :character   Median :1955-11-20 12:00:00.00
##                                        Mean   :1956-06-06 05:21:40.83
##                                        3rd Qu.:1966-03-30 18:00:00.00
##                                        Max.   :1994-05-20 00:00:00.00
##    LastName          FirstName            Title
##  Length:238         Length:238         Length:238
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##       Date                           State           ResidenceStateRegion
##  Min.   :2023-04-04 05:01:00.0   Length:238         Length:238
##  1st Qu.:2023-04-04 05:01:00.0   Class :character   Class :character
##  Median :2023-04-04 05:01:00.0   Mode  :character   Mode  :character
##  Mean   :2023-04-04 05:02:00.5
##  3rd Qu.:2023-04-04 05:01:00.0
##  Max.   :2023-04-04 09:01:00.0
##    BirthYear      BirthMonth        BirthDay      CPI_Country
##  Min.   :1924   Min.   : 1.000   Min.   : 1.00   Min.   :117.2
##  1st Qu.:1944   1st Qu.: 3.000   1st Qu.: 9.00   1st Qu.:117.2
##  Median :1955   Median : 7.000   Median :16.00   Median :117.2
##  Mean   :1956   Mean   : 6.538   Mean   :16.07   Mean   :117.2
##  3rd Qu.:1966   3rd Qu.: 9.000   3rd Qu.:24.00   3rd Qu.:117.2
##  Max.   :1994   Max.   :12.000   Max.   :31.00   Max.   :117.2
##  CPI_Change_Country  GDP_Country        GrossTertiaryEducationEnrollment
##  Min.   :7.5        Min.   :2.143e+13   Min.   :88.2
##  1st Qu.:7.5        1st Qu.:2.143e+13   1st Qu.:88.2
##  Median :7.5        Median :2.143e+13   Median :88.2
##  Mean   :7.5        Mean   :2.143e+13   Mean   :88.2
##  3rd Qu.:7.5        3rd Qu.:2.143e+13   3rd Qu.:88.2
##  Max.   :7.5        Max.   :2.143e+13   Max.   :88.2
##  GrossPrimaryEducationEnrollmentCountry LifeExpectancyCountry
##  Min.   :101.8                          Min.   :78.5
##  1st Qu.:101.8                          1st Qu.:78.5
##  Median :101.8                          Median :78.5
##  Mean   :101.8                          Mean   :78.5
##  3rd Qu.:101.8                          3rd Qu.:78.5
##  Max.   :101.8                          Max.   :78.5
##  TaxRevenueCountryCountry TotalTaxRateCountry PopulationCountry
##  Min.   :9.6              Min.   :36.6        Min.   :328239523
##  1st Qu.:9.6              1st Qu.:36.6        1st Qu.:328239523
##  Median :9.6              Median :36.6        Median :328239523
##  Mean   :9.6              Mean   :36.6        Mean   :328239523
```

```
##  3rd Qu.:9.6              3rd Qu.:36.6        3rd Qu.:328239523
##  Max.   :9.6              Max.   :36.6        Max.   :328239523
##  LatitudeCountry LongitudeCountry  New_Variable
##  Min.   :37.09   Min.   :-95.71   Min.   : 56.0
##  1st Qu.:37.09   1st Qu.:-95.71   1st Qu.:112.5
##  Median :37.09   Median :-95.71   Median :134.0
##  Mean   :37.09   Mean   :-95.71   Mean   :132.7
##  3rd Qu.:37.09   3rd Qu.:-95.71   3rd Qu.:156.0
##  Max.   :37.09   Max.   :-95.71   Max.   :198.0
```

#Q14: Use any of the numerical variables from the dataset and perform the following statistical functions (Mean, Median, Mode, Range)

```
mean(Billionaires_Statistics$FinalWorthNew)
```

```
## [1] 10637.82
```

```
median(Billionaires_Statistics$FinalWorthNew)
```

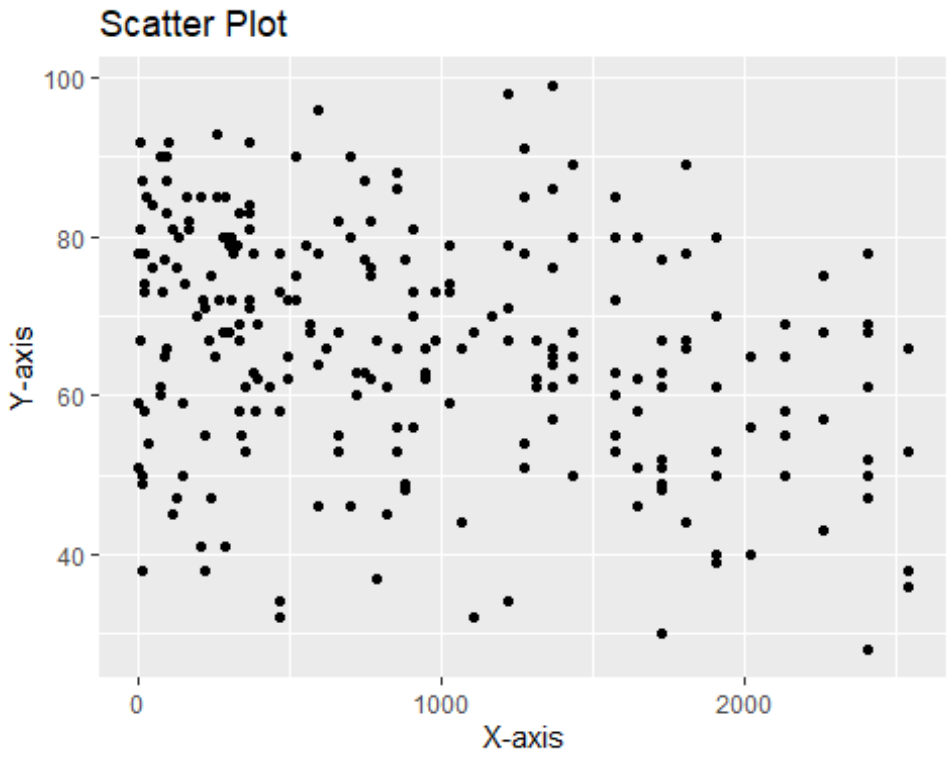
```
## [1] 3800
```

```
table_result=table(Billionaires_Statistics$RankNew)
Mode<-names(table_result)[table_result==max(table_result)]
print(Mode)
```

```
## [1] "1725"
```

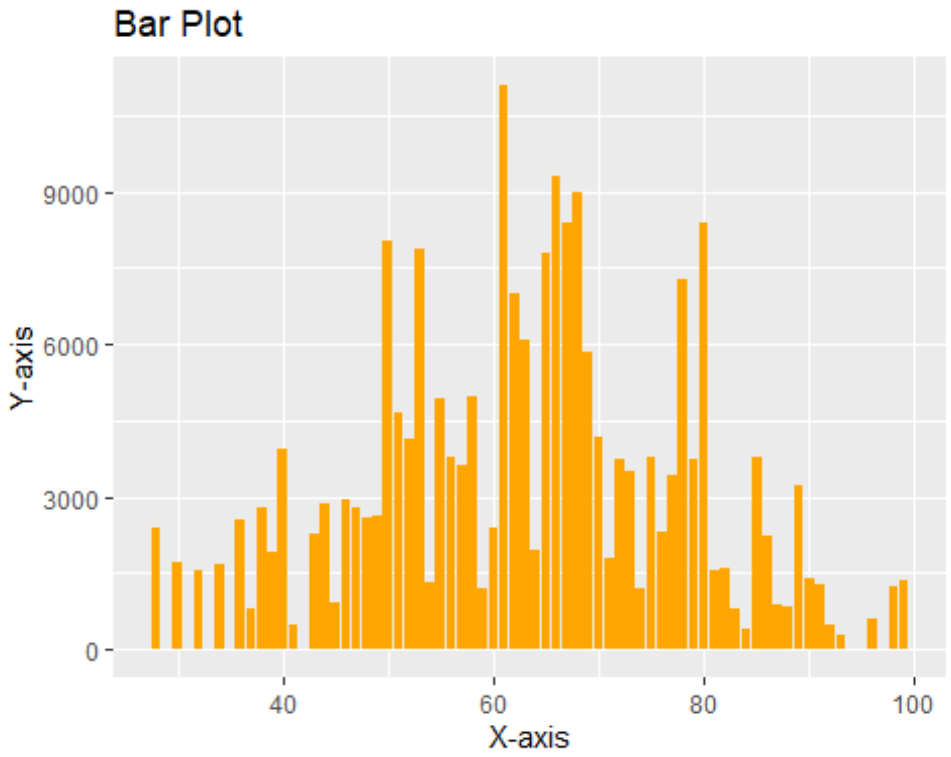
#Q15: Plot a scatter plot for any 2 variables in your dataset

```
library(ggplot2)
 ggplot(Billionaires_Statistics, aes(x = RankNew, y = AgeNew)) + geom_point()
+
  labs(title = "Scatter Plot", x = "X-axis", y = "Y-axis")
```



#Q16: Plot a bar plot for any 2 variables in your dataset

```
ggplot(Billionaires_Statistics, aes(x = AgeNew, y = RankNew)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(title = "Bar Plot", x = "X-axis", y = "Y-axis")
```



#Q17: Find the correlation between any 2 variables by applying least square linear regression model

```
variable1 <- "RankNew"
variable2 <- "FinalWorthNew"

linear_model <- lm(Billionaires_Statistics[[variable1]] ~
Billionaires_Statistics[[variable2]], data = Billionaires_Statistics)
correlation <- cor(Billionaires_Statistics[[variable1]],
fitted(linear_model))

cat("Correlation between", variable1, "and", variable2, ":", correlation,
"\n")

## Correlation between RankNew and FinalWorthNew : 0.4505491
```