R Assignment

Group

2023-06-07

```
library(readxl)
 World_City_Populations_2023 <- read_excel("C:/Users/rishi/Downloads/World City Populations 2023.xlsx",
     sheet = "World City Populations 2023")
```

```
## New names:
## * `` -> `...8`
```

Structure of data

```
str(World_City_Populations_2023)
```

```
## tibble [778 x 8] (S3: tbl_df/tbl/data.frame)
##  $ Pop2023   : num [1:778] 37194104 32941308 29210808 23209616 22619736 ...
##  $ Pop2022   : num [1:778] 37274000 32065760 28516904 22478116 22429800 ...
##  $ City      : chr [1:778] "Tokyo" "Delhi" "Shanghai" "Dhaka" ...
##  $ Country   : chr [1:778] "Japan" "India" "China" "Bangladesh" ...
##  $ Continent : chr [1:778] "Asia" "Asia" "Asia" "Asia" ...
##  $ growthRate: num [1:778] -0.0021 0.0273 0.0243 0.0325 0.0085 0.0089 0.0199 0.0203 0.016 -0.0024 ..
##  $ rank      : num [1:778] 1 2 3 4 5 6 7 8 9 10 ...
##  $ ...8      : num [1:778] NA NA NA NA NA NA NA NA NA NA ...
```

variables

```
variables <- names(World_City_Populations_2023)
variables
```

```
## [1] "Pop2023"    "Pop2022"    "City"       "Country"    "Continent"
## [6] "growthRate" "rank"       "...8"
```

15 Rows

```
head(World_City_Populations_2023, 15)
```

```
## # A tibble: 15 x 8
##     Pop2023  Pop2022 City        Country    Continent     growthRate  rank  ...8
##       <dbl>    <dbl> <chr>       <chr>      <chr>              <dbl> <dbl> <dbl>
##  1 37194104 37274000 Tokyo       Japan      Asia             -0.0021     1    NA
##  2 32941308 32065760 Delhi       India      Asia              0.0273     2    NA
##  3 29210808 28516904 Shanghai    China      Asia              0.0243     3    NA
```

```
##  4 23209616 22478116 Dhaka       Bangladesh Asia              0.0325     4    NA
##  5 22619736 22429800 Sao Paulo   Brazil     South America     0.0085     5    NA
##  6 22281442 22085140 Mexico City Mexico     North America     0.0089     6    NA
##  7 22183200 21750020 Cairo       Egypt      Africa            0.0199     7    NA
##  8 21766214 21333332 Beijing     China      Asia              0.0203     8    NA
##  9 21296516 20961472 Mumbai      India      Asia              0.016      9    NA
## 10 19013434 19059856 Osaka       Japan      Asia             -0.0024    10    NA
## 11 17340704 16874740 Chongqing   China      Asia              0.0276    11    NA
## 12 17236230 16839950 Karachi     Pakistan   Asia              0.0235    12    NA
## 13 16315534 15628085 Kinshasa    DR Congo   Africa            0.044     13    NA
## 14 15945912 15387639 Lagos       Nigeria    Africa            0.0363    14    NA
## 15 15847768 15636243 Istanbul    Turkey     Asia              0.0135    15    NA
```

User Defined Functions

```
calculatePopulationChange <- function(city) {
  pop2022 <- World_City_Populations_2023$Population2022[World_City_Populations_2023$City == city]
  pop2023 <- World_City_Populations_2023$Population2023[World_City_Populations_2023$City == city]

  if (length(pop2022) > 0 && length(pop2023) > 0) {
    change <- (pop2023 - pop2022) / pop2022 * 100
    return(change)
  } else {
    return("City not found!")
  }
}
```

filter data

```
filtered_data <- World_City_Populations_2023[World_City_Populations_2023$growthRate > 0.05, ]
```

new data frame

```
dependent_variable <- World_City_Populations_2023$Pop2023
independent_variable <- World_City_Populations_2023$growthRate
new_data <- data.frame(Dependent = dependent_variable, Independent = independent_variable)
```

removing missing

```
World_City_Populations_2023 <- na.omit(World_City_Populations_2023)
```

Removing Duplicate Items

```
World_City_Populations_2023 <- unique(World_City_Populations_2023)
```

Reordering Rows

```
World_City_Populations_2023 <- World_City_Populations_2023[order(World_City_Populations_2023$rank, decr
```

Rename Column

```
colnames(World_City_Populations_2023) <- c("Population2023", "Population2022", "City", "Country", "Conti
```

Add new variable

```
World_City_Populations_2023$Population_double_2022 <- World_City_Populations_2023$Population2022 * 2
```

Training Set

```
set.seed(123)
train_indices <- sample(1:nrow(World_City_Populations_2023), size = 0.8 * nrow(World_City_Populations_20
train_set <- World_City_Populations_2023[train_indices, ]
```

Summary

```
summary(World_City_Populations_2023)
```

```
##  Population2023    Population2022        City             Country
##  Min.   : 871449   Min.   : 876728   Length:12          Length:12
##  1st Qu.: 925113   1st Qu.: 938253   Class :character   Class :character
##  Median : 964631   Median : 962009   Mode  :character   Mode  :character
##  Mean   :1199393   Mean   :1203479
##  3rd Qu.:1400430   3rd Qu.:1399781
##  Max.   :2264876   Max.   :2276533
##   Continent           GrowthRate             Rank             NA
##  Length:12          Min.   :-0.027100   Min.   :227.0   Min.   :227.0
##  Class :character   1st Qu.:-0.007325   1st Qu.:386.5   1st Qu.:386.5
##  Mode  :character   Median :-0.001000   Median :603.0   Median :603.0
##                     Mean   :-0.002883   Mean   :518.4   Mean   :518.4
##                     3rd Qu.: 0.003650   3rd Qu.:639.8   3rd Qu.:639.8
##                     Max.   : 0.014000   Max.   :676.0   Max.   :676.0
##  Population_double_2022
##  Min.   :1753456
##  1st Qu.:1876506
##  Median :1924017
##  Mean   :2406958
##  3rd Qu.:2799561
##  Max.   :4553066
```

Mean, Median ,Mode

```
mean_population2023 <- mean(World_City_Populations_2023$Population2023)
median_population2023 <- median(World_City_Populations_2023$Population2023)
range_population2023 <- range(World_City_Populations_2023$Population2023)
print(mean_population2023)
```

```
## [1] 1199393
```

```
print(median_population2023)
```

```
## [1] 964631
```

```
print(range_population2023)
```

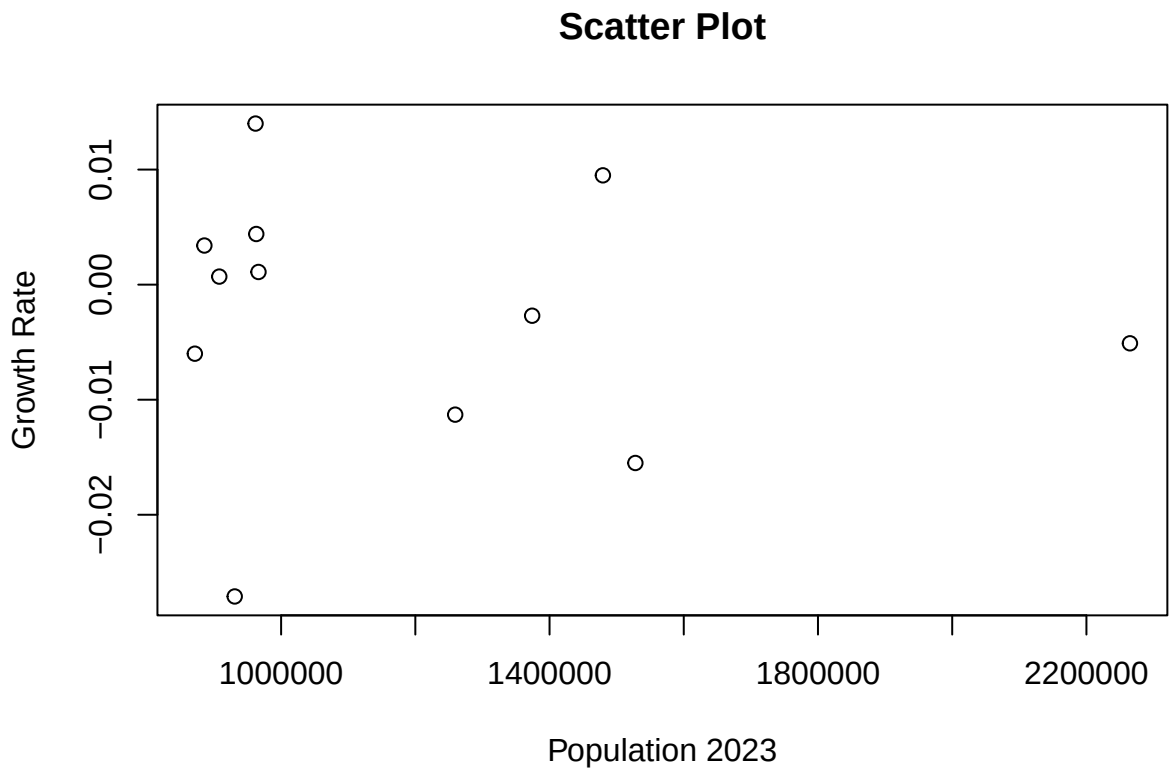
```
## [1]  871449 2264876
```

```
# to find mode
rank <- World_City_Populations_2023$Rank
tt <- table(as.vector(rank))
mode_pop <- names(tt) [tt == max(tt)]
print(mode_pop)
```

```
##  [1] "227" "351" "361" "395" "441" "601" "605" "607" "636" "651" "670" "676"
```

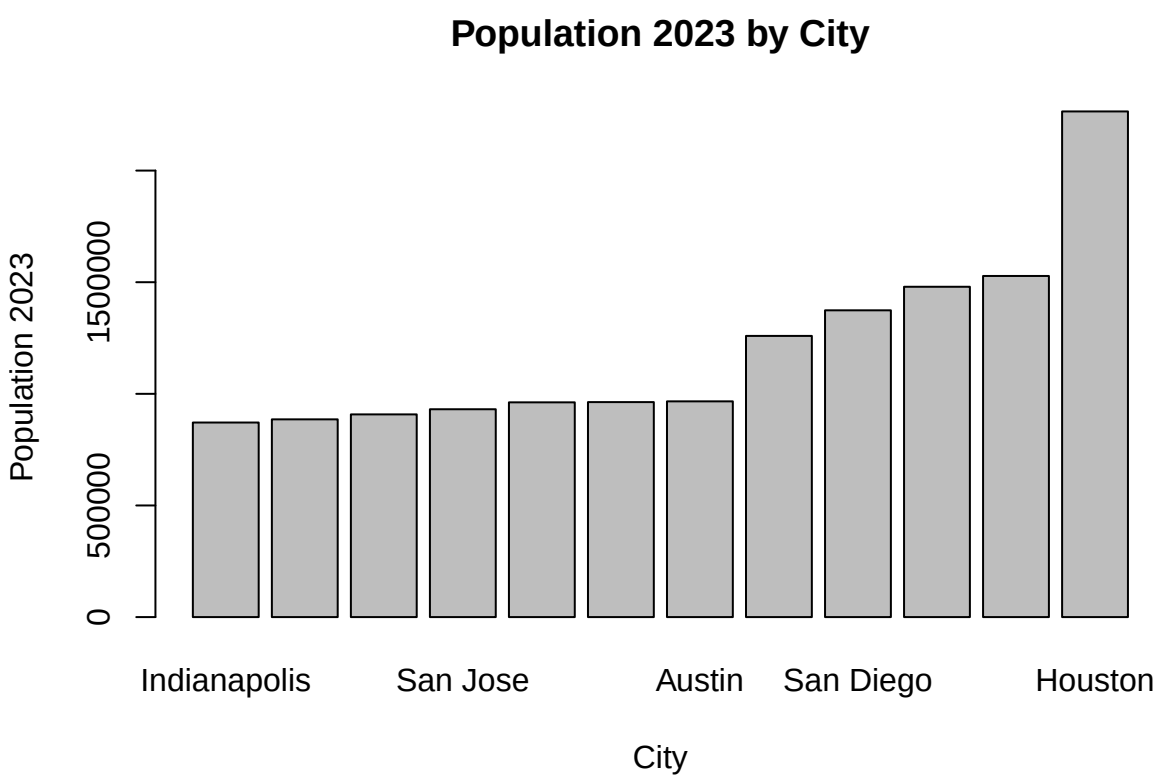
scatter plot

```
plot(World_City_Populations_2023$Population2023, World_City_Populations_2023$GrowthRate, xlab = "Popula
```



BAr Plot

```
barplot(World_City_Populations_2023$Population2023, names.arg = World_City_Populations_2023$City, xlab =
```



Correlation

```
correlation <- cor(World_City_Populations_2023$Population2023, World_City_Populations_2023$GrowthRate)
print(correlation)
```

```
## [1] -0.1105717
```