# Module 4 - Assignment 1

## Pierce, Michaela

### Data Transformation

# Load required packages  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ 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

# Load the dataset  
state\_income <- read\_csv("state\_income.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,  
## e.g.:  
## dat <- vroom(...)  
## problems(dat)

## Rows: 32526 Columns: 18  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (7): State\_Name, State\_ab, County, City, Place, Type, Primary  
## dbl (11): id, State\_Code, Zip\_Code, Area\_Code, ALand, AWater, Lat, Lon, Mean...  
##   
## 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.

#### State Incomes

In this section, I will create a subset of data from the state\_income.csv file. I will be using the following variables (columns).

1. State\_Name
2. State\_ab
3. County
4. City
5. Type
6. ALand
7. Mean
8. Median
9. Stdev

# Create state\_income2 dataset with selected variables and rearrange column order  
# using select() command to help get remaining columns  
state\_income2 <- state\_income %>%  
 select(State\_ab, everything()) %>%  
 select(State\_Name, State\_ab, County, City, Type, ALand, Mean, Median, Stdev) %>%  
 rename(State = State\_Name) # Rename the State\_Name column to State  
  
# using head() command to show first 10 rows of data of state\_income2  
head(state\_income2, 10)

## # A tibble: 10 × 9  
## State State\_ab County City Type ALand Mean Median Stdev  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Alabama AL Mobile County Chickasaw City 1.09e7 38773 30506 33101  
## 2 Alabama AL Barbour County Louisville City 2.61e7 37725 19528 43789  
## 3 Alabama AL Shelby County Columbiana City 4.48e7 54606 31930 57348  
## 4 Alabama AL Mobile County Satsuma City 3.69e7 63919 52814 47707  
## 5 Alabama AL Mobile County Dauphin Is… Town 1.62e7 77948 67225 54270  
## 6 Alabama AL Cullman County Cullman Town 8.91e6 50715 42643 35886  
## 7 Alabama AL Escambia County East Brewt… City 8.83e6 33737 23610 28256  
## 8 Alabama AL Elmore County Coosada Town 1.02e7 46319 40242 38941  
## 9 Alabama AL Morgan County Eva Town 1.05e7 57994 39591 47235  
## 10 Alabama AL Talladega County Sylacauga CDP 4.52e7 54807 41712 51359

# Rename columns  
# Ex.) state\_income2 <- rename(state\_income2, new\_variable\_name = old\_varaibale\_name)   
  
state\_income2 <- rename(state\_income2,  
 SquareArea = ALand,  
 IncomeMean = Mean,  
 IncomeMedian = Median,  
 IncomeStDev = Stdev)  
  
# Display the first 10 rows of the dataset  
head(state\_income2, 10)

## # A tibble: 10 × 9  
## State State\_ab County City Type SquareArea IncomeMean IncomeMedian  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Alabama AL Mobile County Chic… City 10894952 38773 30506  
## 2 Alabama AL Barbour Coun… Loui… City 26070325 37725 19528  
## 3 Alabama AL Shelby County Colu… City 44835274 54606 31930  
## 4 Alabama AL Mobile County Sats… City 36878729 63919 52814  
## 5 Alabama AL Mobile County Daup… Town 16204185 77948 67225  
## 6 Alabama AL Cullman Coun… Cull… Town 8913021 50715 42643  
## 7 Alabama AL Escambia Cou… East… City 8826252 33737 23610  
## 8 Alabama AL Elmore County Coos… Town 10222339 46319 40242  
## 9 Alabama AL Morgan County Eva Town 10544874 57994 39591  
## 10 Alabama AL Talladega Co… Syla… CDP 45178321 54807 41712  
## # ℹ 1 more variable: IncomeStDev <dbl>

# Create a new dataset containing only North Carolina data using filter()  
NC\_income <- state\_income2 %>%  
 filter(State == "North Carolina")  
  
# Display the first 10 rows of the new dataset  
head(NC\_income, 10)

## # A tibble: 10 × 9  
## State State\_ab County City Type SquareArea IncomeMean IncomeMedian  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 North Carolina NC Alama… Elon CDP 3515396 89973 300000  
## 2 North Carolina NC Johns… Wend… Town 23956770 67438 300000  
## 3 North Carolina NC Samps… Sted… Town 1353212 43538 25196  
## 4 North Carolina NC Hende… Hend… CDP 2625120 38120 31430  
## 5 North Carolina NC Beauf… Pine… Town 4121722 30468 17951  
## 6 North Carolina NC Davie… Clem… Town 5903422 97561 80720  
## 7 North Carolina NC Blade… Blad… Town 5737410 38588 20838  
## 8 North Carolina NC Samps… Clin… CDP 8562785 34778 23603  
## 9 North Carolina NC Lee C… Broa… Town 3350431 60384 52298  
## 10 North Carolina NC Guilf… Burl… City 75533002 54337 300000  
## # ℹ 1 more variable: IncomeStDev <dbl>

#### NC Incomes

I will be using the NC\_income dataset to create summaries of the incomes within North Carolina including summaries by county, city and type.

# Arrange the NC\_income dataset by County in ascending order using arrange()  
NC\_income <- arrange(NC\_income, County)  
  
# Display the first 10 rows of the arranged dataset  
head(NC\_income, 10)

## # A tibble: 10 × 9  
## State State\_ab County City Type SquareArea IncomeMean IncomeMedian  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 North Carolina NC Alama… Elon CDP 3515396 89973 300000  
## 2 North Carolina NC Alama… Meba… City 23213152 67397 55632  
## 3 North Carolina NC Alama… Hend… Track 12734435 57073 41022  
## 4 North Carolina NC Alama… Ahos… Track 199246026 54071 42038  
## 5 North Carolina NC Alama… Red … Track 93319263 30673 20786  
## 6 North Carolina NC Alama… Stat… Track 10829691 40174 27569  
## 7 North Carolina NC Alama… Supp… Track 29875162 45625 32324  
## 8 North Carolina NC Alama… Stat… Track 37718022 55177 48504  
## 9 North Carolina NC Alama… Moor… Track 13853696 106274 83085  
## 10 North Carolina NC Alama… Moor… Track 7037037 93463 79991  
## # ℹ 1 more variable: IncomeStDev <dbl>

summary1 <- group\_by(NC\_income, County)   
summary1 <- summarise(summary1, mean = mean(IncomeMean))   
  
summary2 <- NC\_income %>%   
 group\_by(City) %>%   
 summarise(mean = mean(IncomeMean))  
  
summary1

## # A tibble: 49 × 2  
## County mean  
## <chr> <dbl>  
## 1 Alamance County 58430.  
## 2 Anson County 36559.  
## 3 Avery County 41915   
## 4 Beaufort County 40029.  
## 5 Bladen County 35796   
## 6 Brunswick County 53794.  
## 7 Burke County 50283.  
## 8 Caldwell County 50050   
## 9 Camden County 53950   
## 10 Carteret County 50700   
## # ℹ 39 more rows

summary2

## # A tibble: 335 × 2  
## City mean  
## <chr> <dbl>  
## 1 Aberdeen 71839   
## 2 Advance 96650   
## 3 Ahoskie 51360   
## 4 Albemarle 61028.  
## 5 Andrews 39213   
## 6 Angier 58414.  
## 7 Apex 113786.  
## 8 Arapahoe 58334   
## 9 Arden 85641   
## 10 Ash 50636   
## # ℹ 325 more rows

Summary1 presents the average income for each county. It organizes the NC\_income dataset according to counties and computes the mean of IncomeMean for each county.

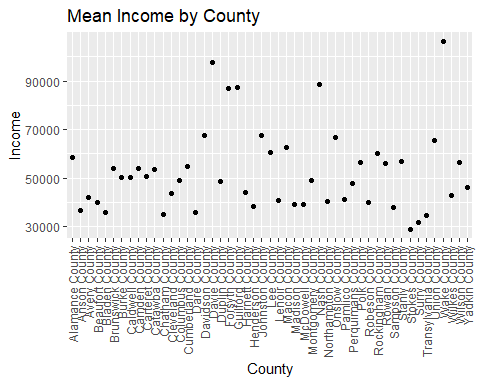
Summary2 showcases the average income for each city. It organizes the NC\_income dataset by city and computes the mean of IncomeMean for each city.

# Create summary3 dataset using pipe operator  
summary3 <- NC\_income %>%  
 group\_by(Type) %>%  
 summarise(mean = mean(IncomeMean))  
  
summary3

## # A tibble: 5 × 2  
## Type mean  
## <chr> <dbl>  
## 1 CDP 45853.  
## 2 City 55884.  
## 3 Town 53116.  
## 4 Track 58381.  
## 5 Village 0

#### Income Visualization

library(ggplot2)  
  
# Scatterplot for summary1 dataset (County income data)  
ggplot(data = summary1, aes(x = County, y = mean)) +  
 geom\_point() +  
 labs(y = "Income", title = "Mean Income by County") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1))



Based on the scatterplot, Wake County has the largest average income and Stokes County has the lowest average income. Surry County has the second lowest income.

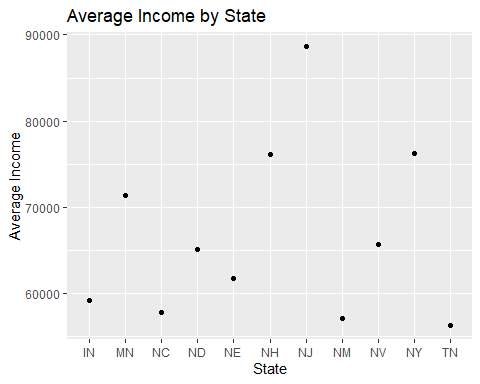
# Create AvgStateIncome dataset by grouping state\_income2 by State\_ab and summarizing using mean of IncomeMean  
AvgStateIncome <- state\_income2 %>%  
 group\_by(State\_ab) %>%  
 summarise(AvgIncome = mean(IncomeMean))  
  
AvgStateIncome

## # A tibble: 52 × 2  
## State\_ab AvgIncome  
## <chr> <dbl>  
## 1 AK 77670.  
## 2 AL 53613.  
## 3 AR 52060.  
## 4 AZ 62578.  
## 5 CA 78127.  
## 6 CO 73323.  
## 7 CT 89227.  
## 8 DC 90668.  
## 9 DE 70142.  
## 10 FL 60888.  
## # ℹ 42 more rows

# Filter AvgStateIncome to include only states with "N" in the abbreviation  
AvgStateIncome\_N <- filter(AvgStateIncome, grepl("N", State\_ab))  
  
AvgStateIncome\_N

## # A tibble: 11 × 2  
## State\_ab AvgIncome  
## <chr> <dbl>  
## 1 IN 59154.  
## 2 MN 71404.  
## 3 NC 57750.  
## 4 ND 65137.  
## 5 NE 61719.  
## 6 NH 76114.  
## 7 NJ 88658.  
## 8 NM 57127.  
## 9 NV 65684.  
## 10 NY 76201.  
## 11 TN 56272.

# Scatterplot for AvgStateIncome dataset  
ggplot(data = AvgStateIncome\_N, aes(x = State\_ab, y = AvgIncome)) +  
 geom\_point() +  
 labs(x = "State", y = "Average Income", title = "Average Income by State")



Based on the scatterplot, New Jersey (NJ) had the largest average income, while Tennessee (TN) had the smallest average income.