code_along_correlation

2025-09-24

The main point of this is to introduce you to correlation matrices and heatmaps in R

```
#load libraries
library(ggplot2)
library(reshape2) #install.packages('reshape2)

#read in the data
colleges <- read.csv("https://remiller1450.github.io/data/Colleges2019_Complete.csv")

#and look at the first few rows
head(colleges)
```

```
##   X                          Name        City State Enrollment Private
## 1 1 Abilene Christian University     Abilene    TX       3524 Private
## 2 3           Adelphi University Garden City    NY       5307 Private
## 3 4               Adrian College      Adrian    MI       1781 Private
## 4 5      AdventHealth University     Orlando    FL       1166 Private
## 5 8     Alabama A & M University      Normal    AL       4990  Public
## 6 9     Alabama State University  Montgomery    AL       3903  Public
##        Region Adm_Rate ACT_median ACT_Q1 ACT_Q3  Cost Net_Tuition
## 1  South West   0.5696         24     21     21 48046       16177
## 2    Mid East   0.7418         25     22     22 49008       24971
## 3 Great Lakes   0.6481         23     19     19 51626       14136
## 4  South East   0.8689         20     18     18 24338       15360
## 5  South East   0.8986         18     16     16 22489        7413
## 6  South East   0.9774         18     16     16 21476       10160
##   Avg_Fac_Salary PercentFemale PercentWhite PercentBlack PercentHispanic
## 1          69804     0.6118200       0.7946       0.0814          0.1635
## 2         111339     0.7211121       0.6669       0.1785          0.1292
## 3          72873     0.4221106       0.8861       0.0692          0.0318
## 4          69759     0.8251058       0.7622       0.1395          0.1338
## 5          63909     0.5640301       0.4684       0.4798          0.0379
## 6          69786     0.6134185       0.4269       0.5232          0.0409
##   PercentAsian FourYearComp_Males FourYearComp_Females Debt_median
## 1       0.0287          0.4115756            0.5283019       16000
## 2       0.0673          0.6114650            0.6998855       19500
## 3       0.0121          0.2320917            0.3319838       18468
## 4       0.0259          0.4761905            0.4132231       16646
## 5       0.0148          0.1471572            0.2313665       15000
## 6       0.0141          0.1282051            0.2679211       18950
##   Salary10yr_median
## 1             43000
## 2             58500
## 3             38600
```

```
## 4             56000
## 5             31000
## 6             27700
```

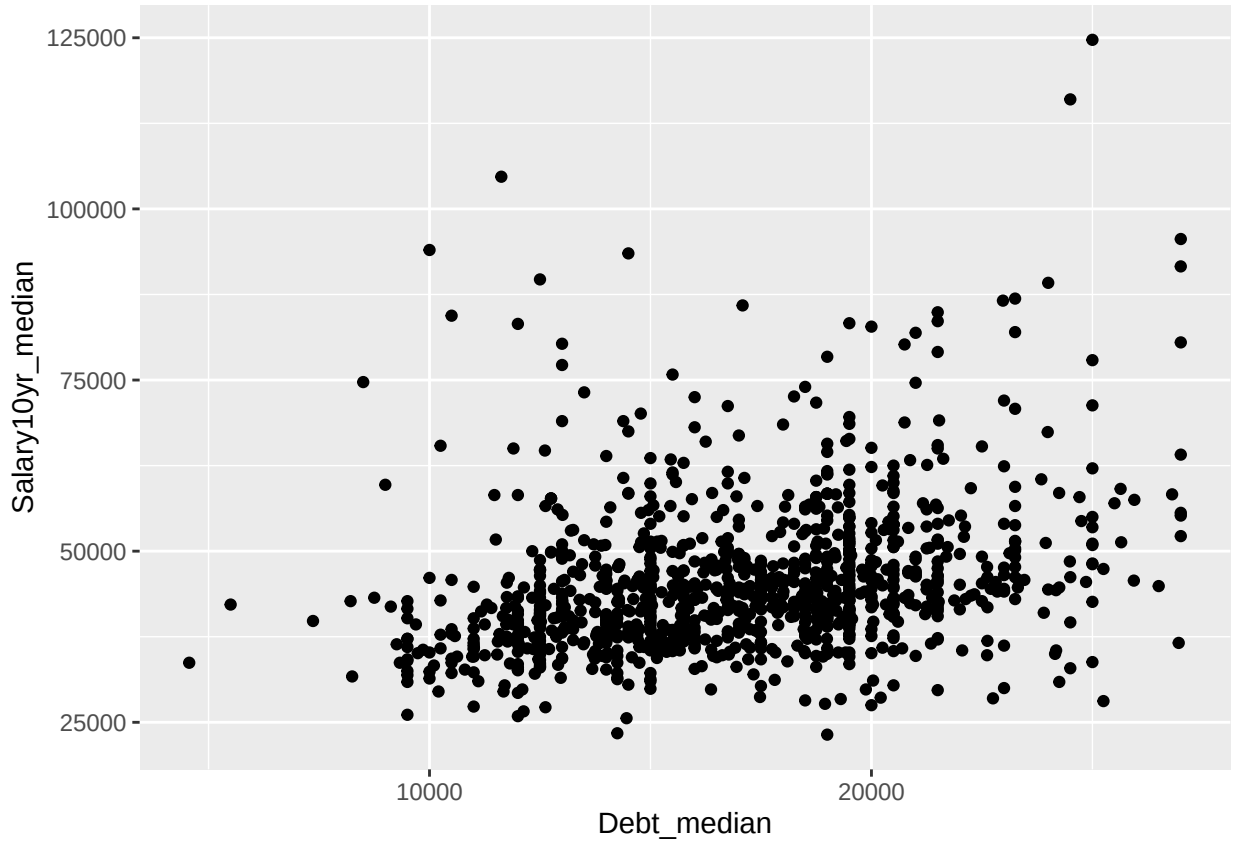
```
cor(colleges$Debt_median,   #this is pearson's correaltion (the default)
    colleges$Salary10yr_median)
```

```
## [1] 0.3062557
```

```
cor(colleges$Debt_median,
    colleges$Salary10yr_median,
    method = "spearman")  #this is spearman's correaltion
```

```
## [1] 0.3644316
```

```
ggplot(data = colleges,  #standard scatter plot from previous labs
       aes(x = Debt_median,
           y = Salary10yr_median)) +
  geom_point()
```



more advanced coding

We can find a matrix of correlations for multiple variables at once. Note that a variable will be perfectly correlated with itself (correlation = 1)

```r
univ <- colleges[, c(19:23)] #grab a subset of the data made up of numeric vars.
head(univ) #and look at the first few rows
```

```
##   PercentAsian FourYearComp_Males FourYearComp_Females Debt_median
## 1       0.0287          0.4115756            0.5283019       16000
## 2       0.0673          0.6114650            0.6998855       19500
## 3       0.0121          0.2320917            0.3319838       18468
## 4       0.0259          0.4761905            0.4132231       16646
## 5       0.0148          0.1471572            0.2313665       15000
## 6       0.0141          0.1282051            0.2679211       18950
##   Salary10yr_median
## 1             43000
## 2             58500
## 3             38600
## 4             56000
## 5             31000
## 6             27700
```

```r
my_cor <- cor(univ) #calculate the correlation MATRIX. Note the code changed only in that
          #we fed it a data frame from the beginning and not two seperate
          #numeric vectors (like we did in the previous example)

round(cor(univ),
      digits = 2) #the table is hard to read so let's use the round() function
```

```
##                     PercentAsian FourYearComp_Males FourYearComp_Females
## PercentAsian                1.00               0.29                 0.24
## FourYearComp_Males          0.29               1.00                 0.93
## FourYearComp_Females        0.24               0.93                 1.00
## Debt_median                 0.01               0.40                 0.39
## Salary10yr_median           0.35               0.69                 0.64
##                     Debt_median Salary10yr_median
## PercentAsian               0.01              0.35
## FourYearComp_Males         0.40              0.69
## FourYearComp_Females       0.39              0.64
## Debt_median                1.00              0.31
## Salary10yr_median          0.31              1.00
```

```r
                    #to make it a bit more manageble. I like either 2 or 3
                    #decimal places myself, purely personal taste thing



#same as above but for spearman's correaltion
cor(univ, method = 'spearman')
```

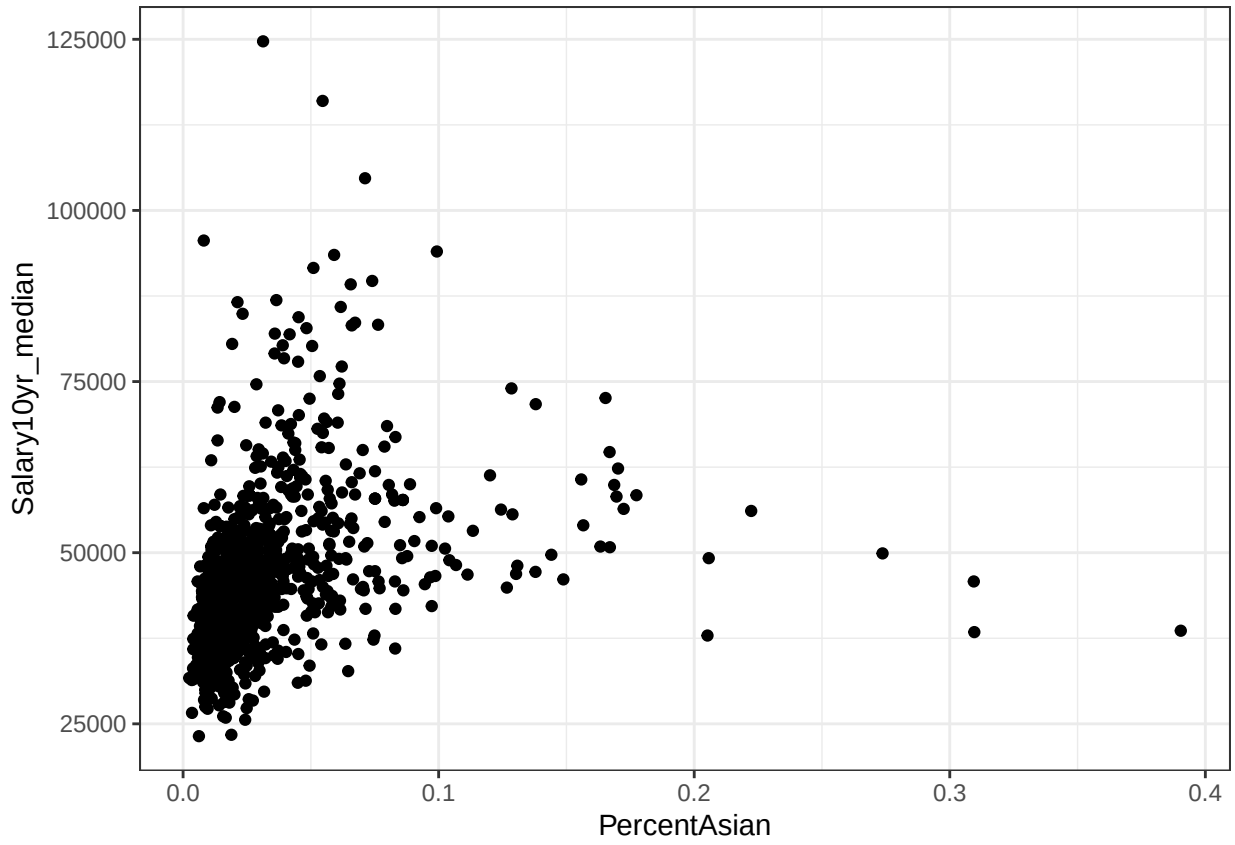
```
##                     PercentAsian FourYearComp_Males FourYearComp_Females
## PercentAsian           1.0000000          0.4771757            0.4238205
## FourYearComp_Males     0.4771757          1.0000000            0.9239659
## FourYearComp_Females   0.4238205          0.9239659            1.0000000
## Debt_median            0.2058505          0.4465556            0.4311662
## Salary10yr_median      0.5836339          0.7146055            0.6718770
```

```
##                       Debt_median Salary10yr_median
## PercentAsian            0.2058505         0.5836339
## FourYearComp_Males      0.4465556         0.7146055
## FourYearComp_Females    0.4311662         0.6718770
## Debt_median             1.0000000         0.3644316
## Salary10yr_median       0.3644316         1.0000000
```

```
round(cor(univ, method = 'spearman'),
      digits = 2)
```

```
##                       PercentAsian FourYearComp_Males FourYearComp_Females
## PercentAsian                  1.00               0.48                 0.42
## FourYearComp_Males            0.48               1.00                 0.92
## FourYearComp_Females          0.42               0.92                 1.00
## Debt_median                   0.21               0.45                 0.43
## Salary10yr_median             0.58               0.71                 0.67
##                       Debt_median Salary10yr_median
## PercentAsian                 0.21               0.58
## FourYearComp_Males           0.45               0.71
## FourYearComp_Females         0.43               0.67
## Debt_median                  1.00               0.36
## Salary10yr_median            0.36               1.00
```

```
#and a scatter plot to figure out which of
#the two correlations is better to use
ggplot(data = univ,
       aes(x = PercentAsian,
           y = Salary10yr_median)) +
  geom_point() +
  theme_bw()
```



Graphing correlation matrices

```
#You may need to install the reshape2 package
#install.packages('reshape2')
library(reshape2)


#the melt() functon makes a table into long format
#and then we look at the first few rows of the data
tidy_cor <- melt(my_cor)
head(tidy_cor)
```

```
##                   Var1               Var2      value
## 1         PercentAsian       PercentAsian 1.00000000
## 2   FourYearComp_Males       PercentAsian 0.28972112
## 3 FourYearComp_Females       PercentAsian 0.24279413
## 4          Debt_median       PercentAsian 0.01434663
## 5    Salary10yr_median       PercentAsian 0.34673178
## 6         PercentAsian FourYearComp_Males 0.28972112
```

```
#column names.....melt does weird things
colnames(tidy_cor) #look at the names
```

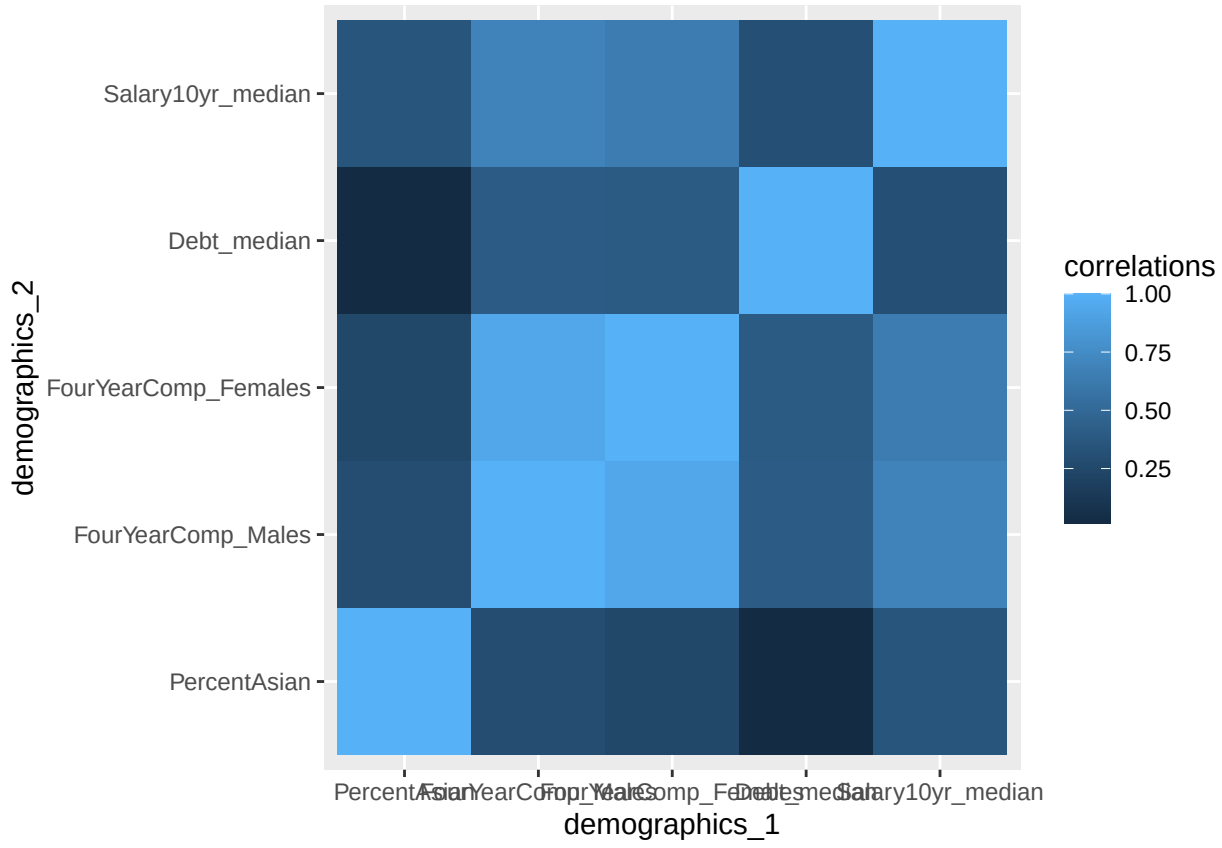
```
## [1] "Var1"  "Var2"  "value"
```

```
colnames(tidy_cor) <- c('demographics_1', 'demographics_2', 'correlations')
#the above line let's us overwrite the names
head(tidy_cor)
```

```
##         demographics_1     demographics_2 correlations
## 1         PercentAsian       PercentAsian   1.00000000
## 2   FourYearComp_Males       PercentAsian   0.28972112
## 3 FourYearComp_Females       PercentAsian   0.24279413
## 4          Debt_median       PercentAsian   0.01434663
## 5    Salary10yr_median       PercentAsian   0.34673178
## 6         PercentAsian FourYearComp_Males   0.28972112
```

```
#beginning of the ggplot is pretty standard
ggplot(data = tidy_cor,
       aes(x = demographics_1,
           y = demographics_2)) +

  #this geom_ is new to you guys. It makes a
  #series of tiles that then can be filled in
  #by the value of correlations
  geom_tile(aes(fill = correlations))
```



```
#NOTE: we cannot use color as it'll do something else
#Go ahead and run the coee to see what happens!
```