308A3

Tianhao You 260830663

2021/1/27

Load libraries and data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Load and modify data

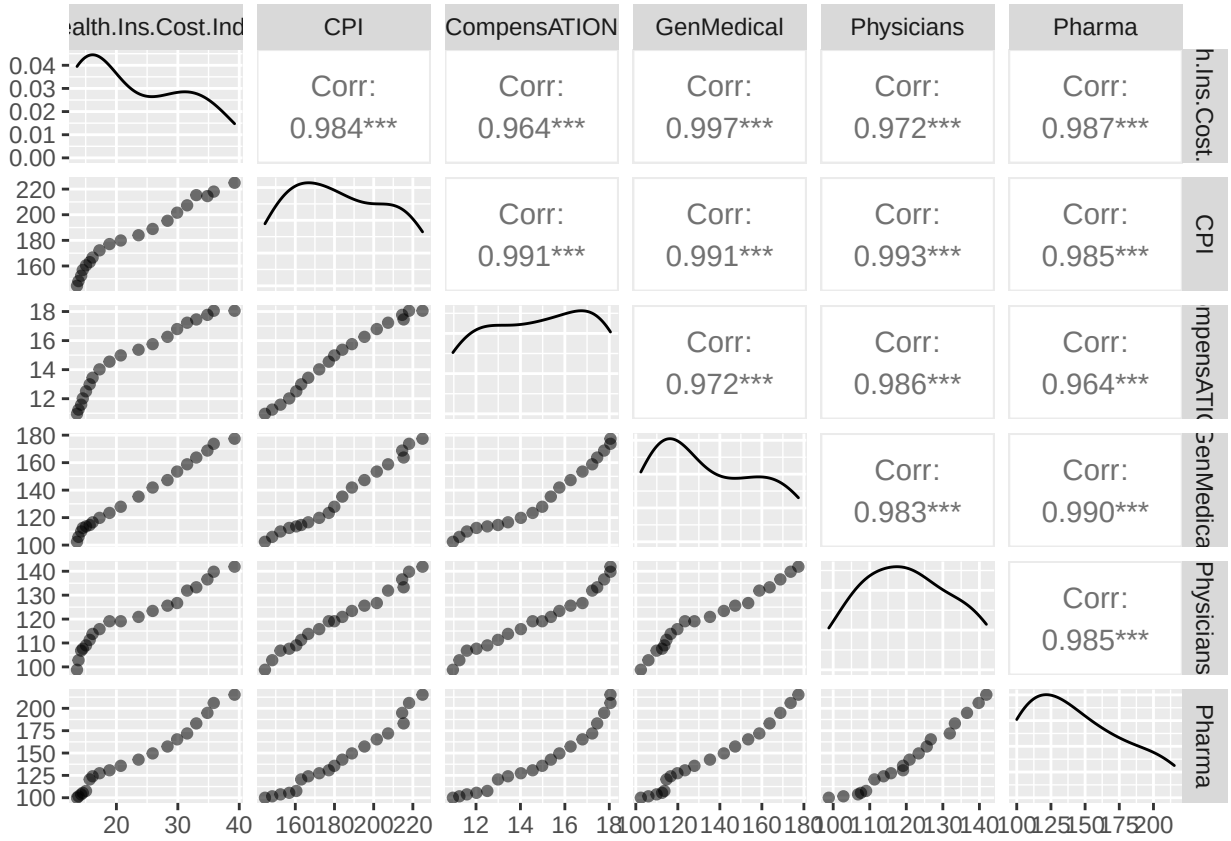
```
indices <- read.csv2("~/Desktop/InsurEconIndices.csv")
indices <- as_tibble(indices)
indices <- indices[3:8]
indices
```

```
## # A tibble: 19 x 6
##    Health.Ins.Cost.Index   CPI CompensATION GenMedical Physicians Pharma
##                    <dbl> <dbl>        <dbl>      <dbl>      <dbl>  <dbl>
##  1                  13.6  144.         11.0       102.       98.8   100
##  2                  13.8  148.         11.2       106       103.    102.
##  3                  14.3  152.         11.6       110.      107.    104.
##  4                  14.5  157.         12.0       112.      108.    106.
##  5                  15.1  160.         12.5       114.      109     108.
##  6                  15.7  163          13.0       115.      111.    120.
##  7                  16.1  167.         13.4       117.      114.    124
##  8                  17.3  172.         14.0       120.      116.    127.
##  9                  18.9  177.         14.6       123.      119.    131.
## 10                  20.7  180.         15.0       128.      119.    136.
```

```
## 11                23.6  184          15.4       135.       121.     143.
## 12                25.9  189.         15.8       142.       123.     150.
## 13                28.3  195.         16.3       147.       126.     157.
## 14                29.9  202.         16.8       154.       127.     165.
## 15                31.5  207.         17.2       159.       132.     172.
## 16                33.0  215.         17.5       164.       133.     183.
## 17                34.8  215.         17.8       169.       137.     195.
## 18                35.8  218.         18.1       174.       140.     206
## 19                39.2  225.         18.1       177.       142.     215.
```

Q1.

```
ggpairs(indices[1:6], aes(alpha = 0.4))
```




```
corI <- cor(indices[1:6],  method = "pearson")
corI
```

```
##                       Health.Ins.Cost.Index       CPI CompensATION GenMedical
## Health.Ins.Cost.Index             1.0000000 0.9842963    0.9638955  0.9966360
## CPI                               0.9842963 1.0000000    0.9908052  0.9911303
## CompensATION                      0.9638955 0.9908052    1.0000000  0.9724983
## GenMedical                        0.9966360 0.9911303    0.9724983  1.0000000
## Physicians                        0.9719992 0.9931699    0.9855722  0.9832603
## Pharma                            0.9874122 0.9847262    0.9638404  0.9901850
##                       Physicians    Pharma
## Health.Ins.Cost.Index  0.9719992 0.9874122
```

```
## CPI                    0.9931699 0.9847262
## CompensATION           0.9855722 0.9638404
## GenMedical             0.9832603 0.9901850
## Physicians             1.0000000 0.9846452
## Pharma                 0.9846452 1.0000000
```

From the plot that we constructed, we can find that the corr between the permutations of 6 indices are all more than 0.9, which means they all have strong correlations. Also, the graph seems to have the same trend, as the x-axis becomes larger and larger, y-axis increases as follows. Thus, we can foucs on the one relations with highest correlation, and the others are redundant.

Q2.

```
# build the matrix with two cols
A = matrix(c(indices$GenMedical,indices$Physicians),ncol = 2)
```

a. sample covariance matrix

```
sample_cov <- cov(A)
sample_cov
```

```
##          [,1]     [,2]
## [1,] 613.2077 308.0442
## [2,] 308.0442 160.0595
```

b. eigenvalues of cov and eigenvectors

```
ev = eigen(sample_cov)
ev
```

```
## eigen() decomposition
## $values
## [1] 769.03004   4.23715
##
## $vectors
##            [,1]       [,2]
## [1,] -0.8923315  0.4513807
## [2,] -0.4513807 -0.8923315
```

c.

From the result above,

$\lambda_1 = 769.03004$ $v_1 =$ [-0.8923315, -0.4513807]

$\lambda_2 = 4.23715$ $v_2 =$ [0.4513807, -0.8923315]

d.

$Y_1 = -0.8923315X_1^* - 0.4513807X_2^*$

e.

```
pA <- prcomp(A, scale = FALSE)
pA
```

```
## Standard deviations (1, .., p=2):
## [1] 27.731391  2.058434
##
```

```
## Rotation (n x k) = (2 x 2):
##            PC1        PC2
## [1,] 0.8923315 -0.4513807
## [2,] 0.4513807  0.8923315
```

```
pA$sdev
```

```
## [1] 27.731391  2.058434
```

```
pA_var <- pA$sdev ^2
pA_var
```

```
## [1] 769.03004   4.23715
```

```
pA_ve <- pA_var/sum(pA_var)
pA_ve
```

```
## [1] 0.994520458 0.005479542
```

It shows that 99.45% of the 1st principal components should be kept.

```
# We can easily find the result with summary() function
summary(pA)
```

```
## Importance of components:
##                             PC1     PC2
## Standard deviation     27.7314 2.05843
## Proportion of Variance  0.9945 0.00548
## Cumulative Proportion   0.9945 1.00000
```

Q3.

In Q3, we use the standardized data.

```
# standardize the data
st_d <- prcomp(indices[1:6], scale = TRUE)
summary(st_d)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.4320 0.23345 0.14397 0.07731 0.05096 0.03782
## Proportion of Variance 0.9858 0.00908 0.00345 0.00100 0.00043 0.00024
## Cumulative Proportion  0.9858 0.99488 0.99833 0.99933 0.99976 1.00000
```

From the graph, we can find that the proportion of first principal component is 98.58%, which reached 95% of the total variability in the data.

Q4.

```
M <- prcomp(indices[1:6], scale = FALSE)
summary(M)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     53.1623 3.88983 2.62778 1.13885 0.48849 0.25074
```

```
## Proportion of Variance  0.9917 0.00531 0.00242 0.00046 0.00008 0.00002
## Cumulative Proportion   0.9917 0.99702 0.99944 0.99989 0.99998 1.00000
```

By comparing the results, we can find that each result of standardized data is lower than the one that is not standardized. There are differences between the two datas. The proportion of first principal component is 99.17%, it also reached 95%.