HW14

106022113

5/26/2021

Question1. Check whether weight mediates the relationship between cylinders and mpg

```
auto <- read.table("auto-data.txt",header=FALSE, na.strings = "?",stringsAsFactors =F)
names(auto) <- c("mpg","cylinders","displacement","horsepower","weight","acceleration","model_year","or
cars_log <- with(auto, data.frame(log(mpg),log(cylinders),log(displacement),log(horsepower),log(weight)
```

a. Try computing the direct effects first

```
model1 <- lm(data=cars_log, log.weight.~log.cylinders.)
summary(model1)
```

i. Regress log.weight. over log.cylinders and report coefficient

```
##
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35473 -0.09076 -0.00147  0.09316  0.40374
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.60365    0.03712  177.92   <2e-16 ***
## log.cylinders.  0.82012    0.02213   37.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1329 on 396 degrees of freedom
## Multiple R-squared:  0.7762, Adjusted R-squared:  0.7757
## F-statistic:  1374 on 1 and 396 DF,  p-value: < 2.2e-16
```

ANSWER : Yes, it has significant effect on weight.

```r
model2 <- lm(data= cars_log,log.mpg.~log.weight.+log.acceleration.+model_year+factor(origin))
summary(model2)
```

ii. Regress log.mpg. over log.weight. and control variables

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38275 -0.07032  0.00491  0.06470  0.39913
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.431155   0.312248  23.799  < 2e-16 ***
## log.weight.       -0.876608   0.028697 -30.547  < 2e-16 ***
## log.acceleration.  0.051508   0.036652   1.405  0.16072
## model_year         0.032734   0.001696  19.306  < 2e-16 ***
## factor(origin)2    0.057991   0.017885   3.242  0.00129 **
## factor(origin)3    0.032333   0.018279   1.769  0.07770 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 392 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8841
## F-statistic: 606.8 on 5 and 392 DF,  p-value: < 2.2e-16
```

ANSWER : Yes, it has significant effect on mpg

b. What is the indirect effect of cylinders on mpg?

```r
model3<- lm(log.mpg. ~log.weight.+log.cylinders.  ,data = cars_log)
summary(model3)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.cylinders., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59242 -0.10298 -0.00572  0.09914  0.61654
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     10.04134    0.40493  24.798  < 2e-16 ***
## log.weight.     -0.81999    0.06094 -13.456  < 2e-16 ***
## log.cylinders.  -0.25176    0.05673  -4.438 1.18e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1612 on 395 degrees of freedom
## Multiple R-squared:  0.7759, Adjusted R-squared:  0.7747
## F-statistic: 683.7 on 2 and 395 DF,  p-value: < 2.2e-16
```

```
indirect_coeff <- model1$coefficients[2]*model2$coefficients[2]
paste("Indirect coefficients: ",indirect_coeff)
```

```
## [1] "Indirect coefficients:  -0.718927457998107"
```

Since cylinders aren't significant in affecting mpg, it is a indirect factor.

c. Bootstrap the confidence interval of indirect effect of cylinders on mpg

```
boot_mediation<-function(model1, model2, dataset) {
  boot_index<-sample(1:nrow(dataset), replace=TRUE)
  data_boot<-dataset[boot_index, ]
  regr1 <-lm(model1, data_boot)
  regr2 <-lm(model2, data_boot)
  return(regr1$coefficients[2] * regr2$coefficients[2])
  }
set.seed(42)
intxns<-replicate(2000, boot_mediation(model1, model2, cars_log))
quantile(intxns, probs=c(0.025, 0.975))
```

i. What is the 95% CI of the indirect effect of log.cylinders. on log.mpg.

```
##       2.5%      97.5%
## -0.7784044 -0.6610106
```

Question 2. Revisit multicollinearity

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

a. Analyze principle components of the four collinear variables

```
collinear_var <- cars_log[,c("log.cylinders.","log.displacement.","log.horsepower.","log.weight.")]
```

i. Create new data frame of the four log transformed variables with high multicollinearity
They are collinear.

```
summary(prcomp(collinear_var,scale. = T))
```

ii. How much variance of the four variables explained by their first PC?

```
## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.9168 0.43316 0.32238 0.18489
## Proportion of Variance 0.9186 0.04691 0.02598 0.00855
## Cumulative Proportion  0.9186 0.96547 0.99145 1.00000
```

```
eigenval <- eigen(cor(collinear_var))$values
eigenval[1]/sum(eigenval) #same as PCA reports
```

```
## [1] 0.9185647
```

```
prcomp(collinear_var,scale. = F)
```

iii. Observe values and valence of first PC eigenvector, what would you call the information captured by this component?

```
## Standard deviations (1, .., p=4):
## [1] 0.73122637 0.15173927 0.09535464 0.07272012
##
## Rotation (n x k) = (4 x 4):
##                          PC1         PC2        PC3         PC4
## log.cylinders.    -0.3944484  0.32615343 -0.6895416  0.51241263
## log.displacement. -0.7221160  0.36134848  0.1626248 -0.56703525
## log.horsepower.   -0.4322835 -0.87289692 -0.2158783 -0.06766477
## log.weight.       -0.3689037 -0.03319916  0.6719242  0.64134686
```

The vector that captures the most orthogoanl varaince is the first principle component. While each principe compoent's magnitude is the variance captured by PC relative to average original data dimension.

b. Revisit regression analysis on cars_log

```
cars_log$PC1 <- prcomp(cars_log, scale. = F)$x[,1]
```

i. Store the scores of first PC as a new column of cars_log

```
pc_regr <- lm(data = cars_log, log.mpg. ~ PC1+log.acceleration.+model_year+factor(origin))
summary(pc_regr)
```

ii. Regress mpg over the column wiht PC1 scores

```
##
## Call:
## lm(formula = log.mpg. ~ PC1 + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42623 -0.05333  0.00096  0.04864  0.39217
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       340.82898    8.58642  39.694  < 2e-16 ***
## PC1                 4.47778    0.11240  39.838  < 2e-16 ***
## log.acceleration.  -0.28591    0.03331  -8.584 2.27e-16 ***
## model_year         -4.43313    0.11232 -39.469  < 2e-16 ***
## factor(origin)2    -0.22934    0.01869 -12.269  < 2e-16 ***
## factor(origin)3    -0.41664    0.02269 -18.364  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09413 on 386 degrees of freedom
## Multiple R-squared:  0.9244, Adjusted R-squared:  0.9234
## F-statistic: 943.4 on 5 and 386 DF,  p-value: < 2.2e-16
```

```r
cars_log$PC1Scale <- prcomp(collinear_var, scale. = T)$x[,1]
pc_regr2 <- lm(data = cars_log, log.mpg.~ PC1Scale+log.acceleration.+model_year+factor(origin))
summary(pc_regr2)
```

iii. Run regression again but standardized

```
##
## Call:
## lm(formula = log.mpg. ~ PC1Scale + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51137 -0.06050 -0.00183  0.06322  0.46792
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.398114   0.166554   8.394 8.99e-16 ***
## PC1Scale           0.145663   0.005057  28.804  < 2e-16 ***
## log.acceleration. -0.191482   0.041722  -4.589 6.02e-06 ***
## model_year         0.029180   0.001810  16.122  < 2e-16 ***
## factor(origin)2    0.008272   0.019636   0.421    0.674
## factor(origin)3    0.019687   0.019395   1.015    0.311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1199 on 386 degrees of freedom
```

```
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8756
## F-statistic: 551.6 on 5 and 386 DF,  p-value: < 2.2e-16
```

Estimator of PC after standardized dropped significantly.

Question 3.

```
security <- read.csv("security_questions.csv")
```

a. How much variance did each extracted factor explain?

```
summary(prcomp(security,scale. = T))
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.0514 1.26346 1.07217 0.87291 0.82167 0.78209 0.70921
## Proportion of Variance 0.5173 0.08869 0.06386 0.04233 0.03751 0.03398 0.02794
## Cumulative Proportion  0.5173 0.60596 0.66982 0.71216 0.74966 0.78365 0.81159
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.68431 0.67229 0.6206 0.59572 0.54891 0.54063 0.51200
## Proportion of Variance 0.02602 0.02511 0.0214 0.01972 0.01674 0.01624 0.01456
## Cumulative Proportion  0.83760 0.86271 0.8841 0.90383 0.92057 0.93681 0.95137
##                           PC15   PC16   PC17   PC18
## Standard deviation     0.48433 0.4801 0.4569 0.4489
## Proportion of Variance 0.01303 0.0128 0.0116 0.0112
## Cumulative Proportion  0.96440 0.9772 0.9888 1.0000
```

b. How many dimensions would you retain?

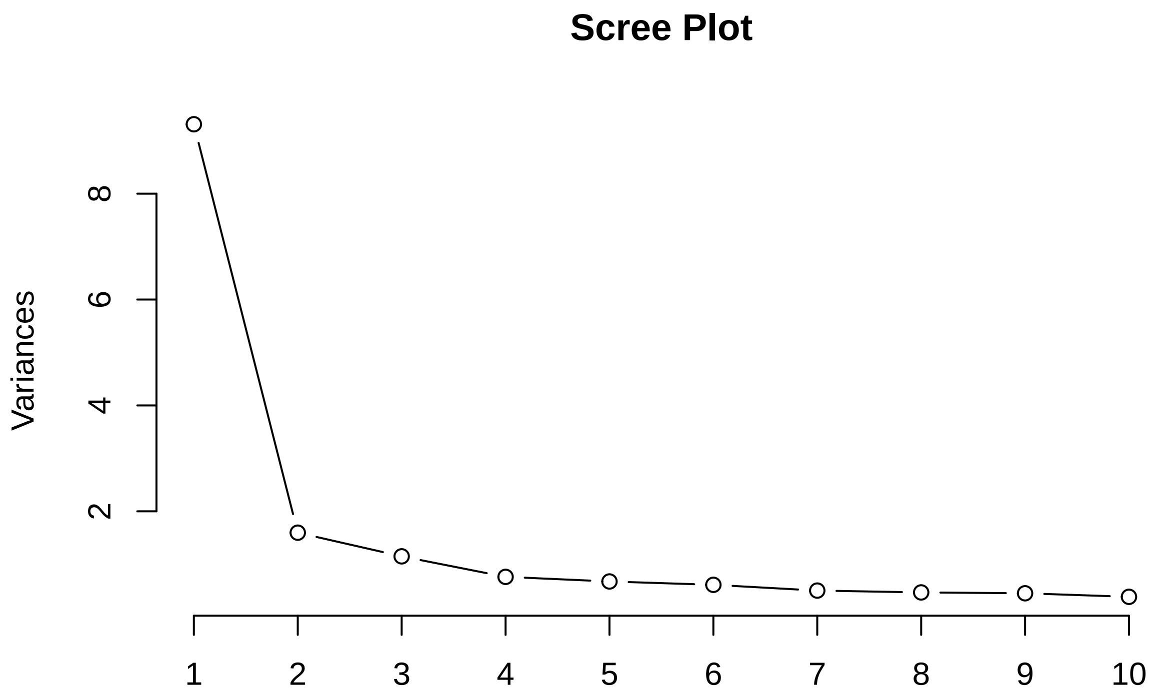
```
eigen(cor(security))$values
```

i. Eigenvalues>=1

```
##  [1] 9.3109533 1.5963320 1.1495582 0.7619759 0.6751412 0.6116636 0.5029855
##  [8] 0.4682788 0.4519711 0.3851964 0.3548816 0.3013071 0.2922773 0.2621437
## [15] 0.2345788 0.2304642 0.2087471 0.2015441
```

Three factors have eigenvlaues >= 1

```
screeplot(prcomp(security,scale. = T),type = "line",main = "Scree Plot")
```



ii. Scree plot
Roughly three factors explains most of the variance

c. Can you interpret what any of the PC means?

The first PC can explain two-thirds of the whole data varaince, which is also the average score of questions.