BACS_HW_Week14_106071041

106071041

2021/5/30

```
cars <- read.table("auto-data.txt", header = F, na.strings = "?")
names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "origin", "car_na
me")
```

```
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(horsepower), log(weight), log(acceleration), model_year, ori
gin))
```

```
head(cars_log)
```

```
##   log.mpg. log.cylinders. log.horsepower. log.weight. log.acceleration.
## 1 2.890372       2.079442        4.867534    8.161660          2.484907
## 2 2.708050       2.079442        5.105945    8.214194          2.442347
## 3 2.890372       2.079442        5.010635    8.142063          2.397895
## 4 2.772589       2.079442        5.010635    8.141190          2.484907
## 5 2.833213       2.079442        4.941642    8.145840          2.351375
## 6 2.708050       2.079442        5.288267    8.375860          2.302585
##   model_year origin
## 1         70      1
## 2         70      1
## 3         70      1
## 4         70      1
## 5         70      1
## 6         70      1
```

Question 1 | Indirect Effect & Direct Effect

a. Direct Effects

i. Model 1: Regress log.weight. over log.cylinders. only and report the coefficient

number of cylinders has a significant direct effect on weight

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

```
summary(cy_weight_regr)[4]
```

```
## $coefficients
##                 Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)    6.6036502 0.03711549 177.92166  0.000000e+00
## log.cylinders. 0.8201241 0.02212817  37.06244 8.330974e-131
```

ii. Model 2: Regress log.mpg. over log.weight. and all control variables and report the coefficient

weight has a significant direct effect on mpg

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

```
summary(mpg_weight_regr)[4]
```

```
## $coefficients
##                      Estimate  Std. Error    t value      Pr(>|t|)
## (Intercept)        7.43115547 0.312247834  23.798902  4.173116e-78
## log.weight.       -0.87660818 0.028697020 -30.547011 1.006403e-105
## log.acceleration.  0.05150802 0.036652496   1.405307  1.607219e-01
## model_year         0.03273393 0.001695554  19.305742  7.558672e-59
## factor(origin)2    0.05799137 0.017885258   3.242412  1.286685e-03
## factor(origin)3    0.03233252 0.018278851   1.768849  7.769672e-02
```

b. Indirect Effect | cylinders on mpg

-0.7195467

```
cy_weight_regr$coefficients[2]
```

```
## log.cylinders.
##      0.8201241
```

```
mpg_weight_regr$coefficients[2]
```

```
## log.weight.
##  -0.8766082
```

```
unname(cy_weight_regr$coefficients[2] * mpg_weight_regr$coefficients[2])
```

```
## [1] -0.7189275
```

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

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

(-0.7820571, -0.6604221)

```
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(3237823)
indirect <- replicate(2000,
boot_mediation(cy_weight_regr, mpg_weight_regr, cars_log))
```

```
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7831568 -0.6570331
```

Question 2 | PCA

Important: remove any rows that have missing values.

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

```
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(horsepower), log(weight), log(acceleration)))
```

a. Let’s analyze the principal components of the four collinear variables

i.Create a new data.frame of the four log-transformed variables with high multicollinearity

(Give this smaller data frame an appropriate name – what might they jointly mean?)

```
components <- cars_log[-1]
```

```
head(components)
```

```
##   log.cylinders. log.horsepower. log.weight. log.acceleration.
## 1       2.079442        4.867534    8.161660          2.484907
## 2       2.079442        5.105945    8.214194          2.442347
## 3       2.079442        5.010635    8.142063          2.397895
## 4       2.079442        5.010635    8.141190          2.484907
## 5       2.079442        4.941642    8.145840          2.351375
## 6       2.079442        5.288267    8.375860          2.302585
```

ii. How much variance of the four variables is explained by their first principal component? (Use the eigenvalues only)

0.7862857

```
eigen(cor(components))$values[1]/sum(eigen(cor(components))$values)
```

```
## [1] 0.7862857
```

iii. Looking at the values and valence (positive/negative) of the first principal component's eigenvector, what would you call the information captured by this component?

3.14 out of 4 of the differences between the regression and the real data can be explained by PC1 which equals 78.6% of the variance.

```
eigen(cor(components))
```

```
## eigen() decomposition
## $values
## [1] 3.14514298 0.65890413 0.14717943 0.04877345
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]
## [1,] -0.5188336 -0.31000501  0.7748667  0.1851763
## [2,] -0.5462486  0.06853368 -0.4983861  0.6697215
## [3,] -0.5167694 -0.42734364 -0.3622813 -0.6473632
## [4,]  0.4066616 -0.84650897 -0.1412273  0.3132152
```

b. Regression analysis on cars_log:

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

```
cars_log_pca <- prcomp(components, scale. = TRUE)
```

```
cars_log_pca
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7734551 0.8117291 0.3836397 0.2208471
##
## Rotation (n x k) = (4 x 4):
##                          PC1         PC2        PC3        PC4
## log.cylinders.    -0.5188336 -0.31000501  0.7748667 -0.1851763
## log.horsepower.   -0.5462486  0.06853368 -0.4983861 -0.6697215
## log.weight.       -0.5167694 -0.42734364 -0.3622813  0.6473632
## log.acceleration.  0.4066616 -0.84650897 -0.1412273 -0.3132152
```

```
cars_log$PC1 <- cars_log_pca$x[,1]
```

```
head(cars_log["PC1"])
```

```
##         PC1
## 1 -2.094000
## 2 -2.665453
## 3 -2.481171
## 4 -2.284026
## 5 -2.482900
## 6 -3.566677
```

```
head(cars_log)
```

```
##   log.mpg. log.cylinders. log.horsepower. log.weight. log.acceleration.
## 1 2.890372       2.079442        4.867534    8.161660          2.484907
## 2 2.708050       2.079442        5.105945    8.214194          2.442347
## 3 2.890372       2.079442        5.010635    8.142063          2.397895
## 4 2.772589       2.079442        5.010635    8.141190          2.484907
## 5 2.833213       2.079442        4.941642    8.145840          2.351375
## 6 2.708050       2.079442        5.288267    8.375860          2.302585
##         PC1
## 1 -2.094000
## 2 -2.665453
## 3 -2.481171
## 4 -2.284026
## 5 -2.482900
## 6 -3.566677
```

ii. Regress mpg over the the column with PC1 scores (replaces cylinders, displacement, horsepower, and weight), as well as acceleration, model_year and origin

```
PC1_mpg_regr <- lm(cars$mpg ~ cars_log$PC1 + cars$acceleration + cars$model_year + factor(cars$origin))
```

```
summary(PC1_mpg_regr)
```

```
##
## Call:
## lm(formula = cars$mpg ~ cars_log$PC1 + cars$acceleration + cars$model_year +
##     factor(cars$origin))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3697  -1.9081  -0.1705   1.7354  13.3958
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -17.77079    4.01321  -4.428 1.24e-05 ***
## cars_log$PC1           3.68719    0.16837  21.899  < 2e-16 ***
## cars$acceleration     -0.80039    0.08818  -9.076  < 2e-16 ***
## cars$model_year        0.69796    0.04970  14.043  < 2e-16 ***
## factor(cars$origin)2   1.29868    0.52524   2.473  0.01385 *
## factor(cars$origin)3   1.98225    0.51761   3.830  0.00015 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.294 on 386 degrees of freedom
## Multiple R-squared:  0.8241, Adjusted R-squared:  0.8219
## F-statistic: 361.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

iii. Try running the regression again over the same independent variables, but this time with everything standardized. How important is this new column relative to other columns?

the coefficient and std. error of everything got obviously smaller after standardized. Scale problem was eliminated by standardization.

```
PC1_mpg_regr_again <- lm(scale(cars$mpg) ~ scale(cars_log$PC1) + scale(cars$acceleration) + scale(cars$model_year) + factor
(cars$origin))
```

```
summary(PC1_mpg_regr_again)
```

```
##
## Call:
## lm(formula = scale(cars$mpg) ~ scale(cars_log$PC1) + scale(cars$acceleration) +
##     scale(cars$model_year) + factor(cars$origin))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45672 -0.24447 -0.02184  0.22235  1.71631
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.08005    0.03007  -2.662  0.00809 **
## scale(cars_log$PC1)       0.83780    0.03826  21.899  < 2e-16 ***
## scale(cars$acceleration) -0.28292    0.03117  -9.076  < 2e-16 ***
## scale(cars$model_year)    0.32942    0.02346  14.043  < 2e-16 ***
## factor(cars$origin)2      0.16639    0.06729   2.473  0.01385 *
## factor(cars$origin)3      0.25397    0.06632   3.830  0.00015 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4221 on 386 degrees of freedom
## Multiple R-squared:  0.8241, Adjusted R-squared:  0.8219
## F-statistic: 361.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

Question 3 | An online marketing firm

```
security_questions <- read_excel("security_questions.xlsx", sheet = "data")
```

a. How much variance did each extracted factor explain?

Please refer to the first row of the summary: PC1 explained 4.5803 variances, PC2 2.01574,….etc.

```
security_questions_pca <- prcomp(security_questions)
```

```
summary(security_questions_pca)
```

```
## Importance of components:
##                           PC1     PC2    PC3     PC4     PC5    PC6     PC7
## Standard deviation     4.5803 2.01574 1.6194 1.30124 1.25295 1.2341 1.07068
## Proportion of Variance 0.5097 0.09871 0.0637 0.04113 0.03814 0.0370 0.02785
## Cumulative Proportion  0.5097 0.60836 0.6721 0.71319 0.75133 0.7883 0.81618
##                            PC8    PC9    PC10    PC11    PC12   PC13    PC14
## Standard deviation     1.03349 0.9940 0.93530 0.88795 0.81779 0.8166 0.76556
## Proportion of Variance 0.02595 0.0240 0.02125 0.01915 0.01625 0.0162 0.01424
## Cumulative Proportion  0.84213 0.8661 0.88738 0.90653 0.92278 0.9390 0.95322
##                           PC15    PC16    PC17    PC18
## Standard deviation     0.74400 0.72833 0.65653 0.64084
## Proportion of Variance 0.01345 0.01289 0.01047 0.00998
## Cumulative Proportion  0.96667 0.97955 0.99002 1.00000
```

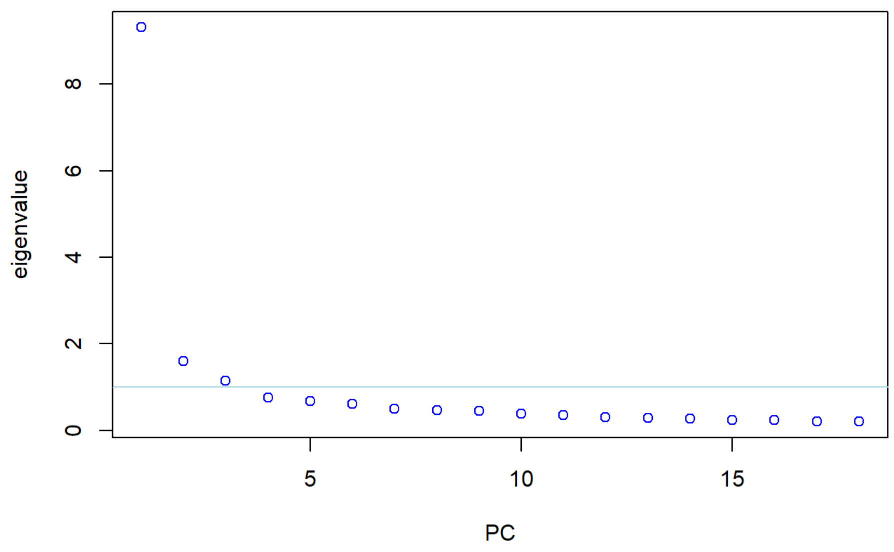
b. How many dimensions would you retain, according to the criteria we discussed?

(show a single visualization with scree plot of data, eigenvalue = 1 cutoff)

i. Eigenvalues ≥ 1

3.

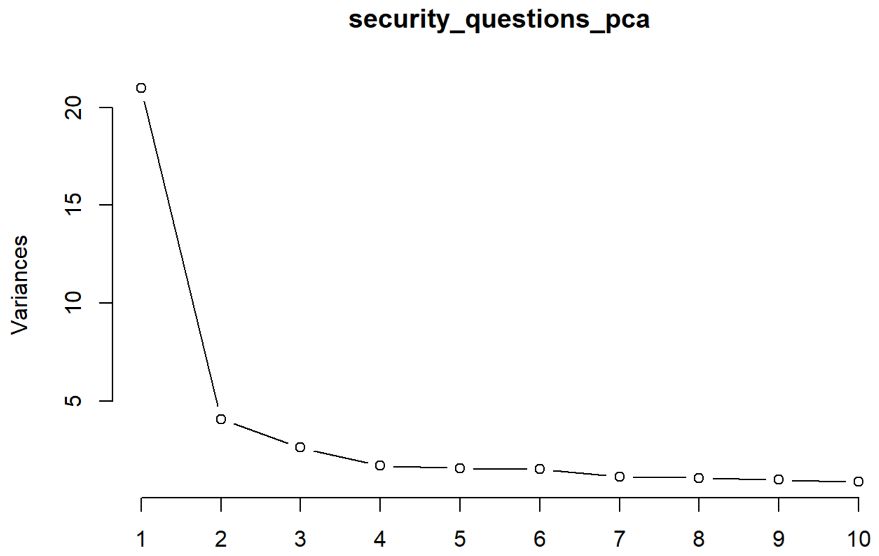
```
plot(eigen(cor(security_questions))$values, ylab = "eigenvalue", xlab = "PC", col = "blue")
abline(h=1, col="lightblue")
```



ii. Scree plot

1.

```
screeplot(security_questions_pca, type="lines")
```



c. (ungraded) Can you interpret what any of the principal components mean?

For PC1, all Questions can explain the data negatively. For PC1+PC2 combined, some questions get positive relation with the data, some become more negative and explain 60% variance of the data.

```
eigen(cor(security_questions))
```

```
## eigen() decomposition
## $values
##  [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
##
## $vectors
##             [,1]         [,2]         [,3]         [,4]         [,5]
##  [1,] -0.2677422  0.110341691 -0.001973491  0.126220668 -0.048468417
##  [2,] -0.2204272  0.010886972  0.083171536  0.258122218  0.093887919
##  [3,] -0.2508767  0.025878543  0.083648794 -0.399268076 -0.061766335
##  [4,] -0.2042919 -0.508981768  0.100759585  0.040690031 -0.072913141
##  [5,] -0.2261544  0.024745268 -0.505845415  0.052574743 -0.193207848
##  [6,] -0.2237681  0.082805088  0.193281966 -0.004209098  0.611348765
##  [7,] -0.2151891  0.251398450  0.302354487  0.327318232  0.008596733
##  [8,] -0.2576225 -0.033526840 -0.320109219  0.076017162  0.209097752
##  [9,] -0.2369512  0.183342667  0.189853454 -0.124795087  0.025138160
## [10,] -0.2248660  0.078103267 -0.496820932 -0.034236123 -0.249119125
## [11,] -0.2467645  0.206580870  0.160903091  0.264607608 -0.210724202
## [12,] -0.2065785 -0.504591429  0.113342400  0.060346524  0.052819352
## [13,] -0.2333066  0.051159791  0.078658760 -0.602543012 -0.030357718
## [14,] -0.2659342  0.078910404  0.146232765 -0.362581586 -0.086718158
## [15,] -0.2307289 -0.008373326 -0.310161141  0.069411508  0.513508897
## [16,] -0.2482681  0.160524168  0.170839887  0.204337585 -0.342722070
## [17,] -0.2023781 -0.525747030  0.102652280  0.080754652 -0.157376900
## [18,] -0.2643810  0.089915229 -0.060800871  0.051492827 -0.024214541
##                [,6]        [,7]         [,8]         [,9]       [,10]
##  [1,]  0.1826730451  0.47564502  0.011877666 -0.158945743 -0.02559547
##  [2,]  0.7972988590 -0.10381142  0.370484027  0.018906337  0.01758985
##  [3,]  0.1343170710 -0.29794768 -0.045361944  0.046160967 -0.62920376
##  [4,] -0.0683434170 -0.07323286 -0.082718228  0.034011814 -0.13146697
##  [5,]  0.1493338250 -0.19273010 -0.188948821  0.218690034  0.09878156
##  [6,]  0.0551361412  0.06503361 -0.538423059  0.331476460 -0.04348905
##  [7,] -0.0562329401 -0.45399251 -0.229822767 -0.236185029  0.31439194
##  [8,] -0.2005009349  0.06635056  0.204619876 -0.232217507  0.08234563
##  [9,] -0.2696485391 -0.12766155  0.452229009  0.595761520  0.25923949
## [10,]  0.0232597277 -0.15613131 -0.250158309  0.141066357  0.09604999
## [11,] -0.1928970917  0.01757216 -0.170741343 -0.289466716 -0.12972901
## [12,] -0.0454546580  0.03110171  0.005586284  0.007633808  0.16822370
## [13,]  0.0949114194  0.03589479 -0.013028375 -0.281562536  0.49131061
## [14,] -0.0006735609  0.07224998  0.032286752 -0.224017714 -0.12173004
## [15,] -0.2572918341 -0.15806779  0.305772284 -0.250812042 -0.19230189
## [16,] -0.2189544787  0.03885431  0.186064954  0.134618480 -0.21266262
## [17,] -0.0527365890 -0.02827931 -0.038609734  0.023978170  0.09198523
## [18,] -0.0327588454  0.58413134 -0.079484842  0.184214340 -0.01232082
##              [,11]         [,12]       [,13]       [,14]        [,15]
##  [1,]  0.261433547  0.3655136121 -0.09437152  0.21538278  0.107191422
##  [2,] -0.141511628 -0.1423173350 -0.01439656 -0.14151031 -0.124321587
##  [3,]  0.215411545  0.0711375730  0.07897104  0.38275058 -0.173199162
##  [4,]  0.182772484  0.0001075882  0.32083974 -0.53718169 -0.009053271
##  [5,] -0.090154465  0.0962621836  0.41176540  0.13779948  0.420108616
##  [6,] -0.230188841  0.1679270706 -0.06866003 -0.12229591 -0.076584623
##  [7,]  0.441121206  0.0404427953 -0.01046519  0.03486607  0.164646045
##  [8,]  0.218910615  0.3074295739  0.08286262 -0.07220809 -0.517381497
##  [9,]  0.125837984 -0.1387657899  0.06167134  0.06636535 -0.103891810
## [10,]  0.006787801 -0.1568738426 -0.54451920 -0.17543121 -0.275471410
## [11,] -0.395639123 -0.4128696157  0.22239835  0.14404891 -0.308218564
## [12,] -0.072388580 -0.1181594259 -0.39416050  0.46427132  0.147423769
## [13,] -0.306206763  0.1388173302  0.19909498  0.01118762 -0.042881369
## [14,]  0.134853427 -0.2306763906 -0.29401321 -0.38305994  0.322075542
## [15,] -0.178156051 -0.1589461038 -0.01621655  0.01470750  0.336177176
## [16,] -0.383866578  0.4817217034 -0.17169894 -0.17403268  0.168614520
## [17,] -0.083760590  0.0503178068  0.03431935  0.09260499 -0.096523523
## [18,]  0.229097907 -0.3832085961  0.19580495  0.02702597  0.077981920
##             [,16]       [,17]       [,18]
##  [1,] -0.26663363  0.15892454  0.49709414
##  [2,]  0.04539846 -0.01378516 -0.07954338
##  [3,]  0.10905667  0.08731092 -0.07451547
##  [4,] -0.26266355  0.39030988  0.02091260
##  [5,] -0.20508811 -0.26389562 -0.07356419
##  [6,] -0.04426883 -0.11718533  0.02443898
##  [7,]  0.19302912  0.07574440 -0.08656284
##  [8,] -0.08324463 -0.31696165 -0.32212598
##  [9,] -0.19386537 -0.01929777  0.22424357
## [10,]  0.07402245  0.24996841  0.14445897
## [11,] -0.28230295 -0.05599291  0.11746105
## [12,] -0.29758805  0.08367724 -0.38027121
## [13,]  0.11740772  0.26739129 -0.04166051
## [14,] -0.16553236 -0.50553644 -0.01188146
## [15,]  0.18191811  0.22010115  0.21302663
## [16,]  0.17538230  0.09232084 -0.26436304
```

```
## [17,]  0.51310849 -0.39101042  0.42651093
## [18,]  0.42203495  0.12287014 -0.30773331
```