Homework_6

Claire Kraft

2024-10-02

I DID NOT SUBMIT THIS HOMEWORK AS I DID NOT COMPLETE THE ASSIGNMENT. COMMITING THIS TO THE REPO FOR MIDTERM EXAM REVIEW PURPROSES. THE CODE RESULTS AND INTERPEATIONS ARE AFTER THE 2024-10-03 THURSDAY TA SESSION.

Question 9.1

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

Conclusion: The TA's conclusion is that, PCA, in this case, didn't really help us build a better linear regression model as it didn't give us any stronger prediction power compared to the pre PCA model. My understanding is that the PCA's goal is to reduce dimension or sift through noise by simply pulling out the most impact features/independent variables.

```
# Set up
rm(list = ls())

# Helper
# Install.packages("GGally") # https://ggobi.github.io/ggally/
library(GGally)
```

```
## Loading required package: ggplot2
```

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

```
# Install.packages('corrplot') # https://www.rdocumentation.org/packages/corrplot/versions/0.94
library(corrplot)
```

```
## corrplot 0.94 loaded
```

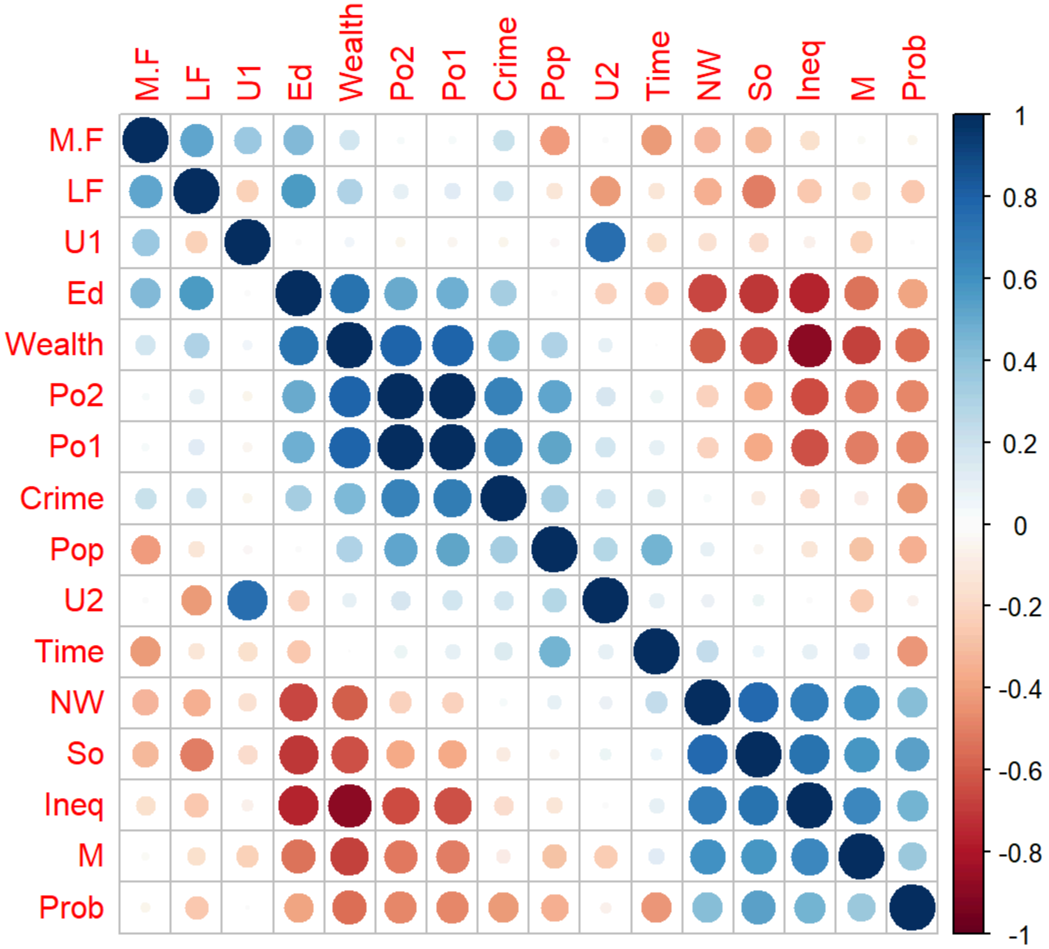
```
library(DAAG)
library(stats)

# Read in data
uscrime <- read.table("~/GitHub/omsa/ISYE 6501/Homework 06/uscrime.txt", stringsAsFactors = FALS
E, header = TRUE)
head(uscrime)
```

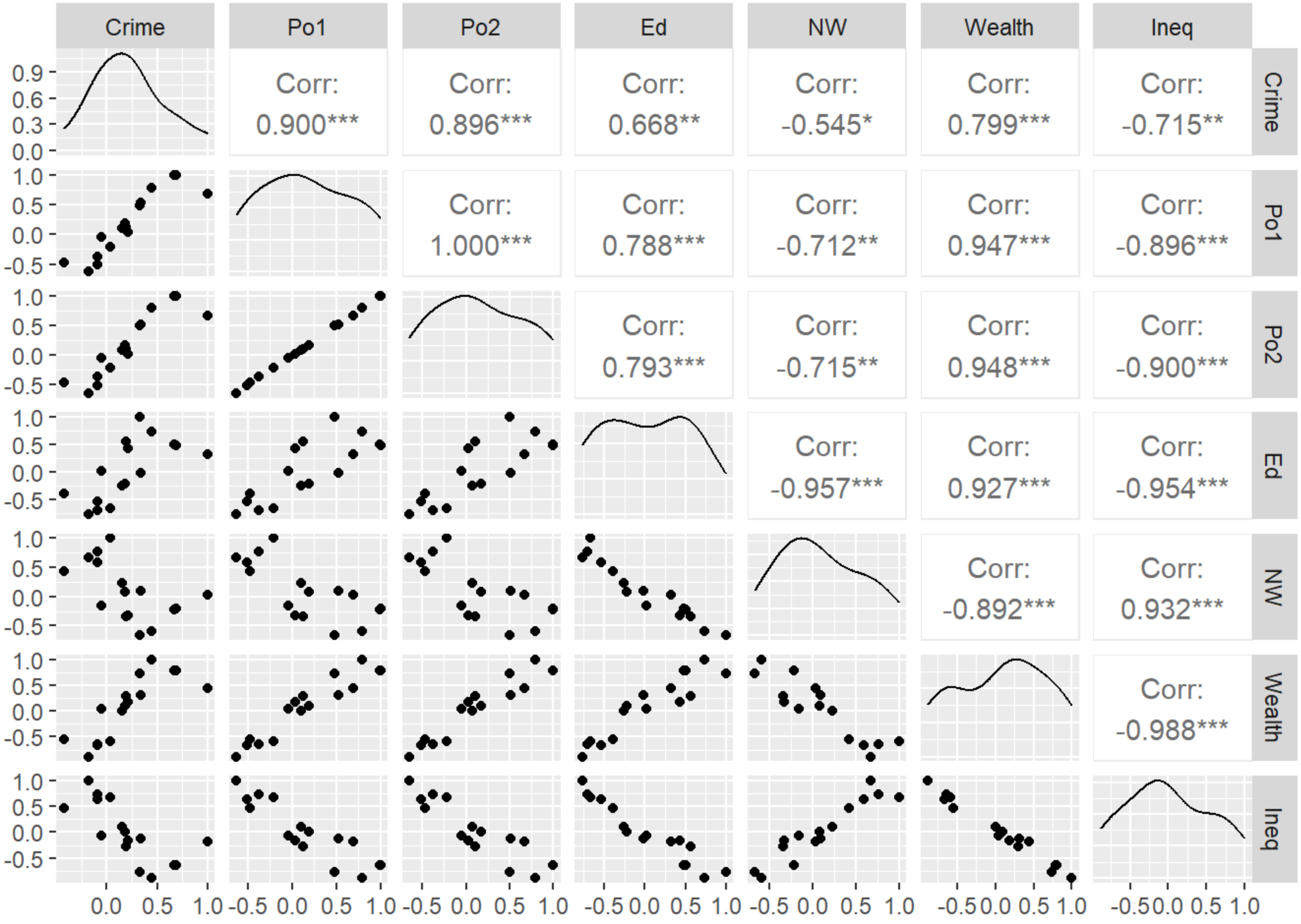
```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

EDA of the data to find the strongest correlations

```
# Draw correlations
# Look at reference 1
crime_data <- cor(uscrime)
corrplot(crime_data, method = "circle", order = 'AOE')
```



```
# Examine some of the strongest correlations-
# Look at reference 2
# ggpairs(uscrime, columns = c("Ed", "Ineq", "Po1")) # Choosing These Parameters Based On My Hw
5 Results
ggpairs(crime_data, columns = c('Crime', 'Po1', 'Po2', 'Ed', 'NW', 'Wealth', 'Ineq'))
```


Crime
Po1
Po2
Ed
NW
Wealth
Ineq
Corr: 0.900***
Corr: 0.896***
Corr: 0.668**
Corr: -0.545*
Corr: 0.799***
Corr: -0.715**
Corr: 1.000***
Corr: 0.788***
Corr: -0.712**
Corr: 0.947***
Corr: -0.896***
Corr: 0.793***
Corr: -0.715**
Corr: 0.948***
Corr: -0.900***
Corr: -0.957***
Corr: 0.927***
Corr: -0.954***
Corr: -0.892***
Corr: 0.932***
Corr: -0.988***
0.9
0.6
0.3
0.0
1.0
0.5
0.0
-0.5
0.0 0.5 1.0
-0.5 0.0 0.5 1.0
-0.50.0 0.5 1.0

```
ggpairs
```

```
## function (data, mapping = NULL, columns = 1:ncol(data), title = NULL, 
##     upper = list(continuous = "cor", combo = "box_no_facet", 
##         discrete = "count", na = "na"), lower = list(continuous = "points", 
##         combo = "facethist", discrete = "facetbar", na = "na"), 
##     diag = list(continuous = "densityDiag", discrete = "barDiag", 
##         na = "naDiag"), params = NULL, ..., xlab = NULL, ylab = NULL, 
##     axisLabels = c("show", "internal", "none"), columnLabels = colnames(data[columns]), 
##     labeller = "label_value", switch = NULL, showStrips = NULL, 
##     legend = NULL, cardinality_threshold = 15, progress = NULL, 
##     proportions = NULL, legends = stop("deprecated")) 
## {
##     warn_deprecated(!missing(legends), "legends")
##     warn_if_args_exist(list(...))
##     stop_if_params_exist(params)
##     isSharedData <- inherits(data, "SharedData")
##     data_ <- fix_data(data)
##     data <- fix_data_slim(data_, isSharedData)
##     if (!missing(mapping) && !is.list(mapping) && missing(columns)) {
##         columns <- mapping
##         mapping <- NULL
##     }
##     stop_if_bad_mapping(mapping)
##     columns <- fix_column_values(data, columns, columnLabels, 
##         "columns", "columnLabels")
##     stop_if_high_cardinality(data, columns, cardinality_threshold)
##     upper <- check_and_set_ggpairs_defaults("upper", upper, continuous = "cor", 
##         combo = "box_no_facet", discrete = "count", na = "na")
##     lower <- check_and_set_ggpairs_defaults("lower", lower, continuous = "points", 
##         combo = "facethist", discrete = "facetbar", na = "na")
##     diag <- check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", 
##         discrete = "barDiag", na = "naDiag", isDiag = TRUE)
##     axisLabels <- fix_axis_label_choice(axisLabels, c("show", 
##         "internal", "none"))
##     proportions <- ggmatrix_proportions(proportions, data, columns)
##     dataTypes <- plot_types(data, columns, columns, allowDiag = TRUE)
##     if (identical(axisLabels, "internal")) {
##         dataTypes$plotType[dataTypes$posX == dataTypes$posY] <- "label"
##     }
##     ggpairsPlots <- lapply(seq_len(nrow(dataTypes)), function(i) {
##         plotType <- dataTypes[i, "plotType"]
##         posX <- dataTypes[i, "posX"]
##         posY <- dataTypes[i, "posY"]
##         xColName <- dataTypes[i, "xVar"]
##         yColName <- dataTypes[i, "yVar"]
##         if (posX > posY) {
##             types <- upper
##         }
##         else if (posX < posY) {
##             types <- lower
##         }
##         else {
##             types <- diag
##         }
##         sectionAes <- add_and_overwrite_aes(add_and_overwrite_aes(aes(x = !!as.name(xColNam
e),
##             y = !!as.name(yColName)), mapping), types$mapping)
##         args <- list(types = types, sectionAes = sectionAes)
##         if (plotType == "label") {
##             args$label <- columnLabels[posX]
##         }
##         plot_fn <- ggmatrix_plot_list(plotType)
##         p <- do.call(plot_fn, args)
##         return(p)
##     })
##     plotMatrix <- ggmatrix(plots = ggpairsPlots, byrow = TRUE,
##         nrow = length(columns), ncol = length(columns), xAxisLabels = (if (axisLabels ==
##             "internal")
##             NULL
##         else columnLabels), yAxisLabels = (if (axisLabels ==
##             "internal")
##             NULL
##         else columnLabels), labeller = labeller, switch = switch,
##         showStrips = showStrips, showXAxisPlotLabels = identical(axisLabels,
##             "show"), showYAxisPlotLabels = identical(axisLabels,
##             "show"), title = title, xlab = xlab, ylab = ylab,
##         data = data_, gg = NULL, progress = progress, legend = legend,
##         xProportions = proportions, yProportions = proportions)
##     plotMatrix
## }
## <bytecode: 0x000002aef0a6c760>
## <environment: namespace:GGally>
```

Performing PCA, summarizing the results, extracting the eigenvectors, and visualizing the variance explained by each principal component in a scree plot.

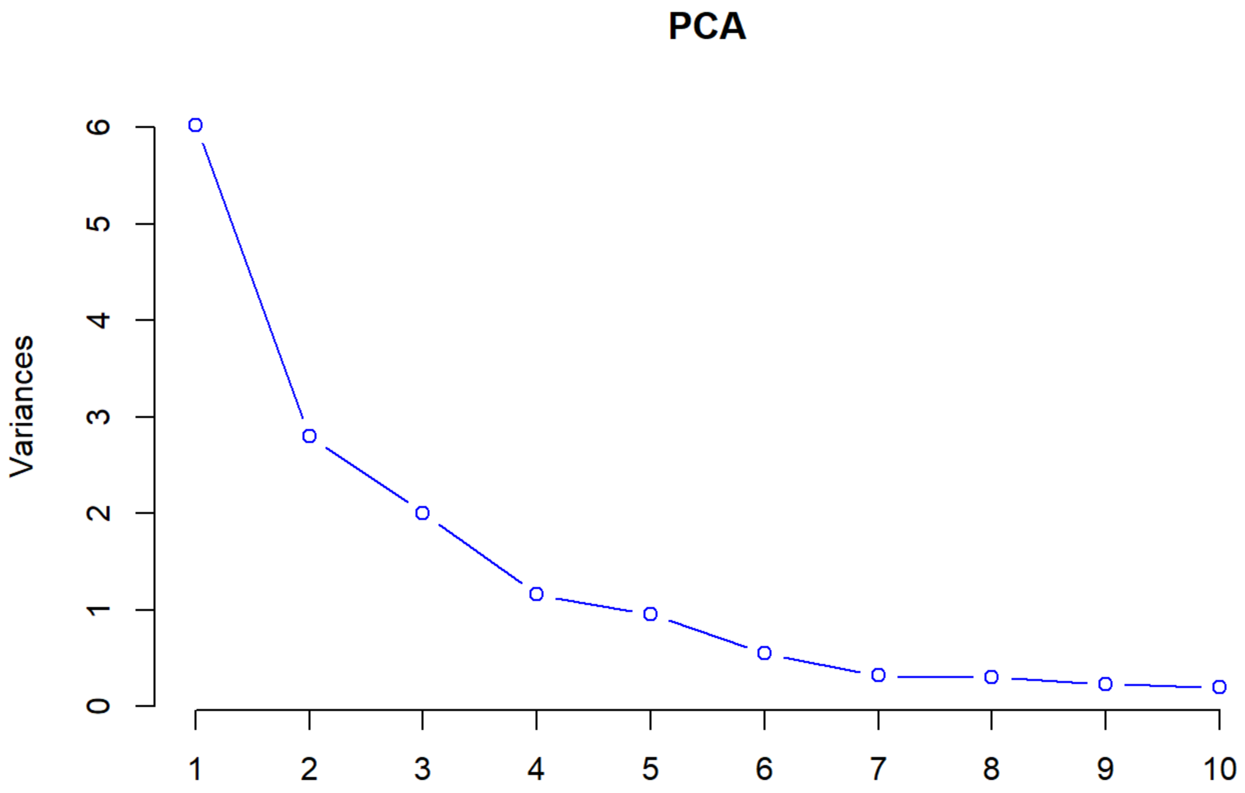
```
# PCA
PCA = prcomp(uscrime[,1:15], scale. = TRUE)
summary(PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142
##                            PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

```
# Eigenvector Matrix
PCA$rotation
```

```
##                PC1         PC2           PC3         PC4         PC5
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537 -0.35832737
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181 -0.12061130
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375 -0.02442839
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059 -0.23527680
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809 -0.20473383
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529 -0.39407588
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029 -0.57877443
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513 -0.08317034
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971 -0.36079387
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096 -0.13136873
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312 -0.13499487
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752  0.01167683
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612 -0.21672823
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389  0.16562829
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002 -0.14764767
##                 PC6         PC7         PC8         PC9        PC10        PC11
## M      -0.449132706 -0.15707378 -0.55367691  0.15474793 -0.01443093  0.39446657
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452  0.23397868
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452 -0.11821954
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978 -0.13042001
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780 -0.13885912
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073  0.38532827
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357 -0.28029732
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718  0.05849643
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966 -0.20695666
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278 -0.17857891
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100  0.47021842
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383  0.31955631
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576 -0.18278697
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006 -0.08978385
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326 -0.26363121
##               PC12        PC13        PC14          PC15
## M       0.16580189  0.05142365  0.04901705 -0.0051398012
## So     -0.05753357  0.29368483 -0.29364512 -0.0084369230
## Ed      0.47786536 -0.19441949  0.03964277  0.0280052040
## Po1     0.22611207  0.18592255 -0.09490151  0.6894155129
## Po2     0.19088461  0.13454940 -0.08259642 -0.7200270100
## LF      0.02705134  0.27742957 -0.15385625 -0.0336823193
## M.F    -0.23925913 -0.31624667 -0.04125321 -0.0097922075
## Pop    -0.18350385 -0.12651689 -0.05326383 -0.0001496323
## NW     -0.36671707 -0.22901695  0.13227774  0.0370783671
## U1     -0.09314897  0.59039450 -0.02335942 -0.0111359325
## U2      0.28440496 -0.43292853 -0.03985736 -0.0073618948
## Wealth -0.32172821  0.14077972  0.70031840  0.0025685109
## Ineq    0.43762828  0.12181090  0.59279037 -0.0177570357
## Prob    0.15567100  0.03547596  0.04761011 -0.0293376260
## Time    0.13536989  0.05738113 -0.04488401 -0.0376754405
```

```
# Use the first 4 PCs
screeplot(PCA, type = "lines", col = "blue")
```



Variance is the square of stdev

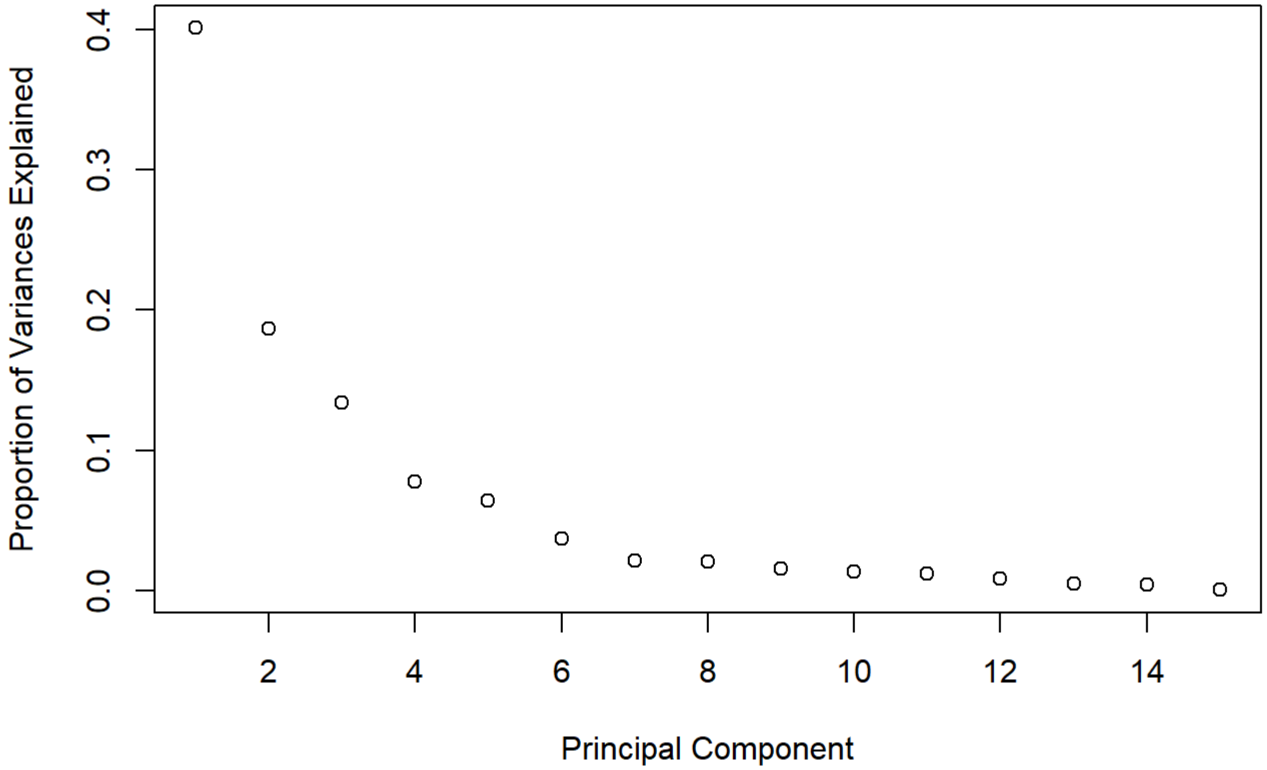
```
# Calculate variances and proportion of variances
variance <- PCA$sdev^2
variance
```

```
##  [1] 6.018952657 2.801847026 2.004944334 1.162207801 0.958298972 0.553193900
##  [7] 0.321818687 0.307401270 0.235155292 0.199880931 0.175685403 0.128190107
## [13] 0.069341691 0.058467765 0.004614165
```

```
# Plot the proportion of variances from PCA
propvvariance <- variance / sum(variance)
propvvariance
```

```
##  [1] 0.401263510 0.186789802 0.133662956 0.077480520 0.063886598 0.036879593
##  [7] 0.021454579 0.020493418 0.015677019 0.013325395 0.011712360 0.008546007
## [13] 0.004622779 0.003897851 0.000307611
```

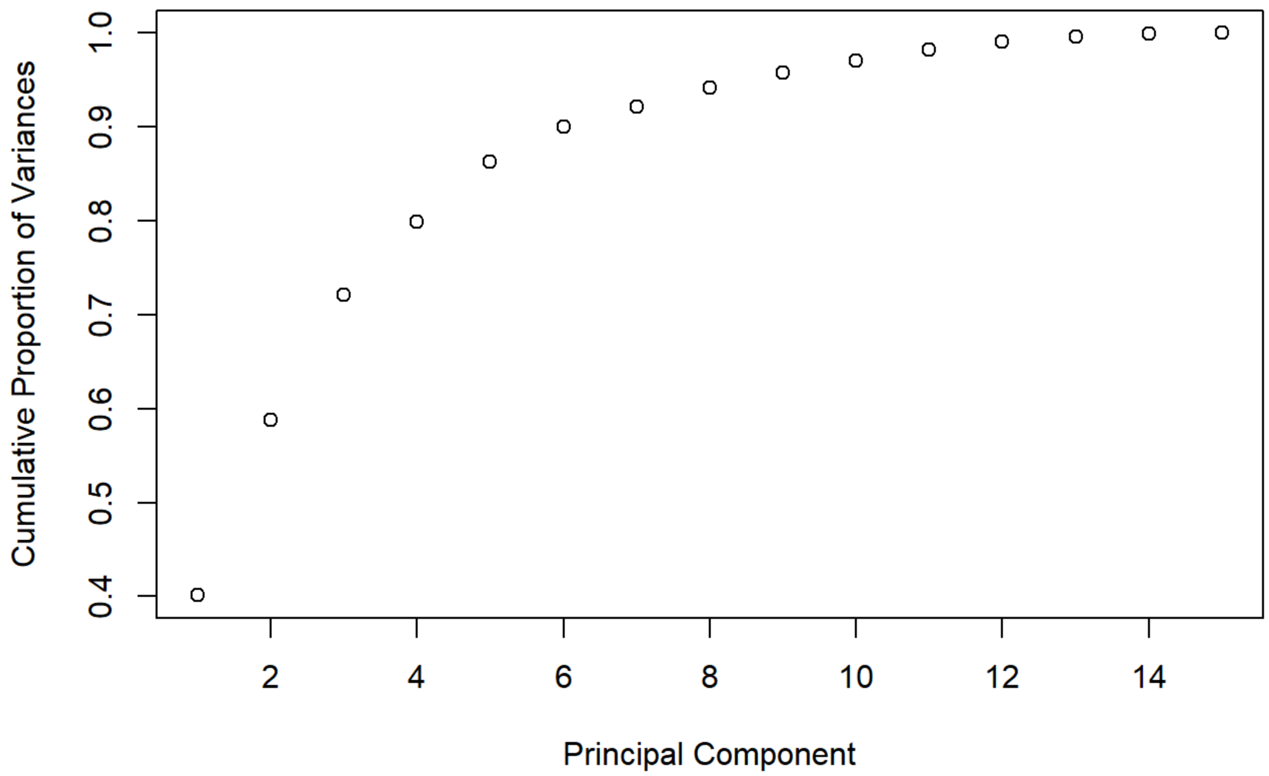
```
plot(propvvariance, xlab = 'Principal Component', ylab = 'Proportion of Variances Explained')
```



```
# Plot the cumulative sum proportion of variances from PCA
cumsum_propvvariance <- cumsum(propvvariance)
cumsum_propvvariance
```

```
##  [1] 0.4012635 0.5880533 0.7217163 0.7991968 0.8630834 0.8999630 0.9214176
##  [8] 0.9419110 0.9575880 0.9709134 0.9826258 0.9911718 0.9957945 0.9996924
## [15] 1.0000000
```

```
plot(cumsum_propvvariance, xlab = 'Principal Component', ylab = 'Cumulative Proportion of Varian
ces')
```



PCA using prcomp

```
## all from the TA
# Get the documentation on the `prcomp` function
?prcomp
```

```
## starting httpd help server ... done
```

```
# Select the first 4 principal components from the transformed data matrix `PCA$x` and assign to
`pcs`
pcs <- PCA$x[,1:4]

# Display the attributes of the `PCA$x` object, which contains the transformed data
attributes(PCA$x)
```

```
## $dim
## [1] 47 15
##
## $dimnames
## $dimnames[[1]]
## NULL
##
## $dimnames[[2]]
##  [1] "PC1"  "PC2"  "PC3"  "PC4"  "PC5"  "PC6"  "PC7"  "PC8"  "PC9"  "PC10"
## [11] "PC11" "PC12" "PC13" "PC14" "PC15"
```

```
# Print the `pcs` object, which contains the first 4 principal components of the transformed dat
a
pcs
```

```
##              PC1         PC2         PC3         PC4
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946
```

```
# Compute and display the covariance matrix of the transformed data `PCA$x`
cov(PCA$x)
```

```
##                PC1           PC2           PC3           PC4           PC5
## PC1   6.018953e+00 -1.098127e-15 -1.544752e-16  3.693830e-16  1.865063e-16
## PC2  -1.098127e-15  2.801847e+00  9.788366e-16  2.342961e-16  9.570724e-16
## PC3  -1.544752e-16  9.788366e-16  2.004944e+00  2.001072e-16  2.600435e-16
## PC4   3.693830e-16  2.342961e-16  2.001072e-16  1.162208e+00  1.333804e-17
## PC5   1.865063e-16  9.570724e-16  2.600435e-16  1.333804e-17  9.582990e-01
## PC6   2.879511e-16  6.154262e-17 -7.190865e-16 -1.718133e-16 -3.009062e-16
## PC7   1.627240e-16  4.240998e-16 -1.923180e-16  3.133954e-16  1.247445e-16
## PC8   1.152276e-15 -5.242942e-16 -3.370596e-16 -6.500971e-17  1.211072e-16
## PC9   1.126604e-15  4.433633e-16 -5.413940e-18  2.688967e-16 -4.532466e-17
## PC10 -2.072922e-16 -2.854120e-16  1.611829e-16  1.638547e-16  1.139697e-16
## PC11 -7.864484e-16  4.422602e-16  1.030251e-16  2.364312e-16  4.765422e-17
## PC12  4.040111e-16 -5.014590e-16 -1.069742e-16  1.373123e-16  5.958954e-17
## PC13 -2.131860e-16  1.285027e-16 -1.473578e-16  4.878262e-17  5.836425e-18
## PC14  4.636968e-16  2.651287e-16 -9.411935e-17 -1.756033e-16  3.617936e-18
## PC15 -4.177944e-16  7.093788e-16 -5.313080e-16 -1.893503e-16 -3.238465e-16
##                PC6           PC7           PC8           PC9          PC10
## PC1   2.879511e-16  1.627240e-16  1.152276e-15  1.126604e-15 -2.072922e-16
## PC2   6.154262e-17  4.240998e-16 -5.242942e-16  4.433633e-16 -2.854120e-16
## PC3  -7.190865e-16 -1.923180e-16 -3.370596e-16 -5.413940e-18  1.611829e-16
## PC4  -1.718133e-16  3.133954e-16 -6.500971e-17  2.688967e-16  1.638547e-16
## PC5  -3.009062e-16  1.247445e-16  1.211072e-16 -4.532466e-17  1.139697e-16
## PC6   5.531939e-01  1.444318e-16 -1.656222e-16  1.342535e-16 -5.587899e-17
## PC7   1.444318e-16  3.218187e-01  2.026634e-16 -1.724479e-16 -1.108387e-16
## PC8  -1.656222e-16  2.026634e-16  3.074013e-01  1.402558e-16 -4.379013e-17
## PC9   1.342535e-16 -1.724479e-16  1.402558e-16  2.351553e-01 -2.231368e-16
## PC10 -5.587899e-17 -1.108387e-16 -4.379013e-17 -2.231368e-16  1.998809e-01
## PC11 -1.118805e-16  2.867243e-18 -3.630722e-17  2.908814e-17 -5.216228e-18
## PC12 -1.598629e-16  4.157458e-17  2.490658e-17  6.072440e-17  7.513374e-17
## PC13  9.446981e-17  4.193918e-18  4.885288e-17 -5.371662e-18  3.302827e-17
## PC14  1.205486e-16  5.842075e-17  1.072506e-17  8.072195e-17  5.964485e-17
## PC15 -3.273878e-17  6.632968e-18  3.272132e-17  8.500337e-17 -1.068804e-16
##               PC11          PC12          PC13          PC14          PC15
## PC1  -7.864484e-16  4.040111e-16 -2.131860e-16  4.636968e-16 -4.177944e-16
## PC2   4.422602e-16 -5.014590e-16  1.285027e-16  2.651287e-16  7.093788e-16
## PC3   1.030251e-16 -1.069742e-16 -1.473578e-16 -9.411935e-17 -5.313080e-16
## PC4   2.364312e-16  1.373123e-16  4.878262e-17 -1.756033e-16 -1.893503e-16
## PC5   4.765422e-17  5.958954e-17  5.836425e-18  3.617936e-18 -3.238465e-16
## PC6  -1.118805e-16 -1.598629e-16  9.446981e-17  1.205486e-16 -3.273878e-17
## PC7   2.867243e-18  4.157458e-17  4.193918e-18  5.842075e-17  6.632968e-18
## PC8  -3.630722e-17  2.490658e-17  4.885288e-17  1.072506e-17  3.272132e-17
## PC9   2.908814e-17  6.072440e-17 -5.371662e-18  8.072195e-17  8.500337e-17
## PC10 -5.216228e-18  7.513374e-17  3.302827e-17  5.964485e-17 -1.068804e-16
## PC11  1.756854e-01  7.286497e-17  4.852983e-17 -5.965616e-18  2.151294e-17
## PC12  7.286497e-17  1.281901e-01  1.633553e-17  2.984866e-17 -8.215483e-17
## PC13  4.852983e-17  1.633553e-17  6.934169e-02 -2.230481e-17 -3.796144e-17
## PC14 -5.965616e-18  2.984866e-17 -2.230481e-17  5.846777e-02  1.200577e-17
## PC15  2.151294e-17 -8.215483e-17 -3.796144e-17  1.200577e-17  4.614165e-03
```

```
# Create a diagonal matrix from the variances (which are the squares of the standard deviations)
of the principal components
diag(PCA$sdev^2)
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]      [,6]      [,7]
##  [1,] 6.018953 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
##  [2,] 0.000000 2.801847 0.000000 0.000000 0.000000 0.0000000 0.0000000
##  [3,] 0.000000 0.000000 2.004944 0.000000 0.000000 0.0000000 0.0000000
##  [4,] 0.000000 0.000000 0.000000 1.162208 0.000000 0.0000000 0.0000000
##  [5,] 0.000000 0.000000 0.000000 0.000000 0.958299 0.0000000 0.0000000
##  [6,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.5531939 0.0000000
##  [7,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.3218187
##  [8,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
##  [9,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [10,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [11,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [12,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [13,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [14,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
## [15,] 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000
##            [,8]      [,9]     [,10]     [,11]     [,12]      [,13]      [,14]
##  [1,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [2,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [3,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [4,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [8,] 0.3074013 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##  [9,] 0.0000000 0.2351553 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
## [10,] 0.0000000 0.0000000 0.1998809 0.0000000 0.0000000 0.00000000 0.00000000
## [11,] 0.0000000 0.0000000 0.0000000 0.1756854 0.0000000 0.00000000 0.00000000
## [12,] 0.0000000 0.0000000 0.0000000 0.0000000 0.1281901 0.00000000 0.00000000
## [13,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.06934169 0.00000000
## [14,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.05846777
## [15,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000000 0.00000000
##             [,15]
##  [1,] 0.000000000
##  [2,] 0.000000000
##  [3,] 0.000000000
##  [4,] 0.000000000
##  [5,] 0.000000000
##  [6,] 0.000000000
##  [7,] 0.000000000
##  [8,] 0.000000000
##  [9,] 0.000000000
## [10,] 0.000000000
## [11,] 0.000000000
## [12,] 0.000000000
## [13,] 0.000000000
## [14,] 0.000000000
## [15,] 0.004614165
```

```
# Print the transformed data matrix `PCA$x`
PCA$x
```

```
##              PC1         PC2         PC3         PC4          PC5         PC6
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115  0.055283376  0.30733835
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709 -1.173199821 -0.58323731
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995  0.541345246  0.71872230
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331 -1.644746496  0.72948841
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305  0.041590320 -0.39409015
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635  1.374360960 -0.69225131
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873  0.718644387 -0.93107472
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387 -0.222781388  0.04912052
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150 -0.324292990  0.12683417
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026 -0.646056610  0.33130781
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210 -0.324083561  0.44365740
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964  0.358087724  0.25696957
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618  0.599881946  1.04761756
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173  0.741887592  0.61569775
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211  0.002969448  0.17074576
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187 -0.387454246 -0.20965321
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865  1.337784608 -0.25633983
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093  0.410914404  0.09317136
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766  1.356288600 -0.89044651
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758 -0.653038858  0.68615337
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548  0.084621572  0.45958300
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484 -0.689327990  0.13338054
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773  0.689939865  0.54002731
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784 -0.744371276  0.01453851
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518  0.803692524  1.52313508
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720 -2.895110075 -0.47766314
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615  1.830793964 -1.11656766
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146 -1.023229895 -0.74149513
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974  0.309946510  0.72486153
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834 -0.254715638 -0.42316550
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099 -0.470913997 -0.45866080
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980  0.331277622  1.30802615
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089  0.291863659 -0.13885903
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927  0.723474420 -0.42036754
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766  0.665406182  0.54144206
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698  0.020031154 -2.43356674
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399 -2.469679496  0.07239996
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576  0.028157162  1.06465126
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645  0.487259213 -0.20571166
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931 -0.645884788  0.63320546
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429  0.009855774 -1.03480444
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379  2.115630145 -0.02332805
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015 -0.867397522 -1.13982198
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770 -0.703116983 -0.65215040
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984  0.806659622 -0.48157983
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769  0.542466034  0.71712602
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946 -1.140712406  0.39563373
##                PC7          PC8          PC9         PC10          PC11
##  [1,] -0.566408161 -0.007801727  0.223509947  0.452743650 -0.0847454174
##  [2,]  0.195611187  0.154566472  0.436777195  0.212085890 -0.0339166059
##  [3,]  0.103306929  0.351138883  0.062992321 -0.067190215 -0.4814915573
##  [4,]  0.266994985 -1.547460841 -0.379541806  0.229223052  0.1098495110
##  [5,]  0.070507664 -0.543237437  0.224632448  0.477690842 -0.3295818584
##  [6,]  0.226482092  0.562323186  0.417722172  0.091009390  0.0102296864
##  [7,]  0.307507661  1.056861503 -1.160218292  0.791683164  0.2829470570
##  [8,]  0.911404993  0.693339330 -0.421314146  0.613278523 -0.3211719754
##  [9,] -0.417420968 -0.053270500  0.232662026  0.065541569  0.1212937342
## [10,]  0.009579488 -0.329270845 -0.123629746  0.200126861 -0.0005664179
## [11,] -0.182961180  0.587179568 -0.070907596 -0.556615080 -0.1727018439
## [12,] -0.462577031  0.307351101 -0.105197263 -0.132898969  0.2984659116
## [13,] -0.494631320  0.753702337 -0.384056907 -0.340154686 -0.3093005372
## [14,] -0.087093101 -0.046931419 -0.159138488  0.280005792  0.1705829803
## [15,]  1.040213660 -0.139392628 -0.147546022 -1.024276227  0.7966941694
## [16,]  0.262430717  0.641818600  0.526895635  0.828407330 -0.2016395195
## [17,] -0.754882880 -0.959968310  0.351808733 -0.046049514  0.1106976222
## [18,] -1.227238054  0.280226677 -0.412734008 -1.074780984  0.1309449295
## [19,]  0.387161139 -0.002276046  0.555855685  0.598093089  0.3873076362
## [20,] -0.401936004  0.240456772  0.341543809  0.229195572  0.7640552201
## [21,] -0.179283176  0.772072202 -0.344317021 -0.192047623 -0.2491916653
## [22,] -1.337728458  0.261648468  0.225568667  0.361253314 -1.2502555533
## [23,]  0.995827754  0.371597176  1.073655584  0.033997150 -0.0148920689
## [24,]  0.042135169 -0.210603749 -0.111463892  0.570729260 -0.2891751385
## [25,] -0.341012092  0.390172476 -0.015090214 -0.107776581  0.0126408264
## [26,] -0.110906098  0.991890307  0.232407672 -0.727397771 -0.1821057801
## [27,] -0.199196211 -0.044269305 -0.015729946 -0.046457518 -0.2413405035
## [28,]  0.113082804 -0.677219677  0.151930973  0.076617716 -0.4139560352
## [29,]  0.248081636 -0.844089307  0.230269486 -0.342149453 -0.8429456727
## [30,] -0.116127247 -0.891169193 -0.011731985 -0.435636015  0.0144413727
## [31,]  0.704852096 -0.538600585  0.439137868 -0.709658521 -0.5740441221
## [32,] -0.786980332 -0.067086938 -0.169888285  0.072917031  0.6056884273
## [33,]  0.767856496  0.027448832 -0.773125607  0.126124015  0.1459949892
## [34,]  0.181257930  0.115379461 -0.101718594  0.321007813 -0.4060548228
## [35,] -0.449541256 -0.276891496  0.007657702  0.202491328  0.0936192141
## [36,] -0.333843509  0.384707595  0.642612190 -0.727991803  0.1824929850
## [37,] -0.343611407  0.157984131  0.915881371  0.481641023  1.1919120577
## [38,]  0.863051754 -0.058247210  0.341385143 -0.133649827 -0.5185529852
## [39,]  0.966860079  0.059557654  0.039345212  0.034036490  0.2185933062
## [40,]  0.767470212 -0.704833575 -1.109887730  0.106827471  0.1951224135
## [41,] -0.589160590 -0.468876595 -0.528478950  0.430811630  0.1829897714
## [42,] -0.557413301 -0.963360913  0.485515025  0.007295728  0.4739341401
## [43,]  0.041128192 -0.573696577 -0.773992630 -0.447789368 -0.1172352964
## [44,] -0.442990964 -0.093002011 -0.515838387  0.241578722 -0.1363783451
## [45,]  0.233636019  0.379908278 -0.815127937 -0.541397364  0.2642920144
## [46,]  0.847914876  0.172381544  0.657987377 -0.480124036  0.1175554086
## [47,] -0.171412192  0.327844331 -0.167078790 -0.002371858  0.2888983375
##              PC12         PC13         PC14          PC15
##  [1,]  0.22096639  0.112616798  0.326964861 -0.0233840087
##  [2,]  0.35686524 -0.297516509  0.252356741  0.0607636781
##  [3,] -0.04701948 -0.052160542 -0.486551130 -0.0421174952
##  [4,]  0.17727101 -0.088381306  0.149678420 -0.0291749700
##  [5,]  0.41807551  0.722152235  0.131027187  0.0751493967
##  [6,] -0.70661980  0.135172709  0.194925675 -0.0155861048
##  [7,] -0.65196573 -0.168327740  0.145473719  0.0654492790
##  [8,]  0.49089082 -0.218057687 -0.623230400  0.0259344691
##  [9,] -0.29249322  0.242429444  0.026476592 -0.0252300906
## [10,] -0.21063943  0.257769674 -0.276967642 -0.0232404560
## [11,] -0.33472808 -0.238074383  0.255472039 -0.0992321732
## [12,] -0.26641418 -0.171319693  0.094123766 -0.0190525547
## [13,]  0.59785665  0.132203906  0.027925309  0.0148583070
## [14,]  0.18719968 -0.571485989  0.250689865 -0.0127642083
## [15,]  0.56068471 -0.217331625  0.037229143 -0.0452385996
## [16,] -0.16367226  0.082957159  0.137971468  0.0210413021
## [17,]  0.33986466  0.128534101 -0.246396571  0.0073811334
## [18,] -0.16259339  0.474477655  0.096820598 -0.0107830419
## [19,]  0.49141798 -0.110318335 -0.185686144 -0.1027680411
## [20,]  0.05854928 -0.173991982  0.041243802  0.0108009160
## [21,]  0.03436398  0.407556122  0.094462966  0.0062668835
## [22,]  0.15171519 -0.319206246  0.003834903  0.0005073113
## [23,]  0.08607424  0.037204214  0.545497655 -0.0129578778
## [24,] -0.20783571  0.240516367 -0.122497400  0.0342080182
## [25,]  0.37619331 -0.117057471 -0.105183565  0.0510978767
## [26,]  0.30036333 -0.137225797 -0.134072192  0.1184870411
## [27,] -0.51580918 -0.066145794 -0.186576416 -0.0791823778
## [28,]  0.24306271  0.140043507  0.629391628  0.0354269136
## [29,]  0.03561083  0.229673348 -0.234477116 -0.0387679658
## [30,] -0.36730664 -0.388569856 -0.025869303  0.0300544785
## [31,] -0.79220655 -0.007892720 -0.201914013 -0.0766956405
## [32,] -0.34195913 -0.154638372  0.085491563  0.0800132601
## [33,]  0.25911938  0.316086918 -0.024206874 -0.1045722437
## [34,]  0.25952688 -0.166191625  0.152140934 -0.0830313640
## [35,] -0.33281300 -0.047752123 -0.312239740  0.1013067365
## [36,]  0.47165172 -0.049320737 -0.382422475  0.0704633747
## [37,] -0.31784996  0.395326593 -0.238009619 -0.0858414347
## [38,] -0.25514910 -0.169135060 -0.013058191  0.0353381517
## [39,]  0.08796506 -0.030789317 -0.067516845  0.1026461875
## [40,] -0.05840207  0.137544171 -0.177710919  0.0704026331
## [41,] -0.26187866  0.058757893 -0.113235908  0.0939372094
## [42,]  0.33534399 -0.291642167  0.013605734  0.0399895760
## [43,] -0.26398492 -0.427157629  0.266115989  0.0276514754
## [44,]  0.17238472 -0.005592707  0.142206916 -0.1612571077
## [45,]  0.39144866  0.508852301  0.223930669 -0.0073779464
## [46,] -0.56753437  0.172018049  0.056680914  0.0850410458
## [47,]  0.01440895 -0.246609753 -0.223916593 -0.1659609523
```

```
# Print the `pcs` object again, which contains the first 4 principal components of the transform
ed data
pcs
```

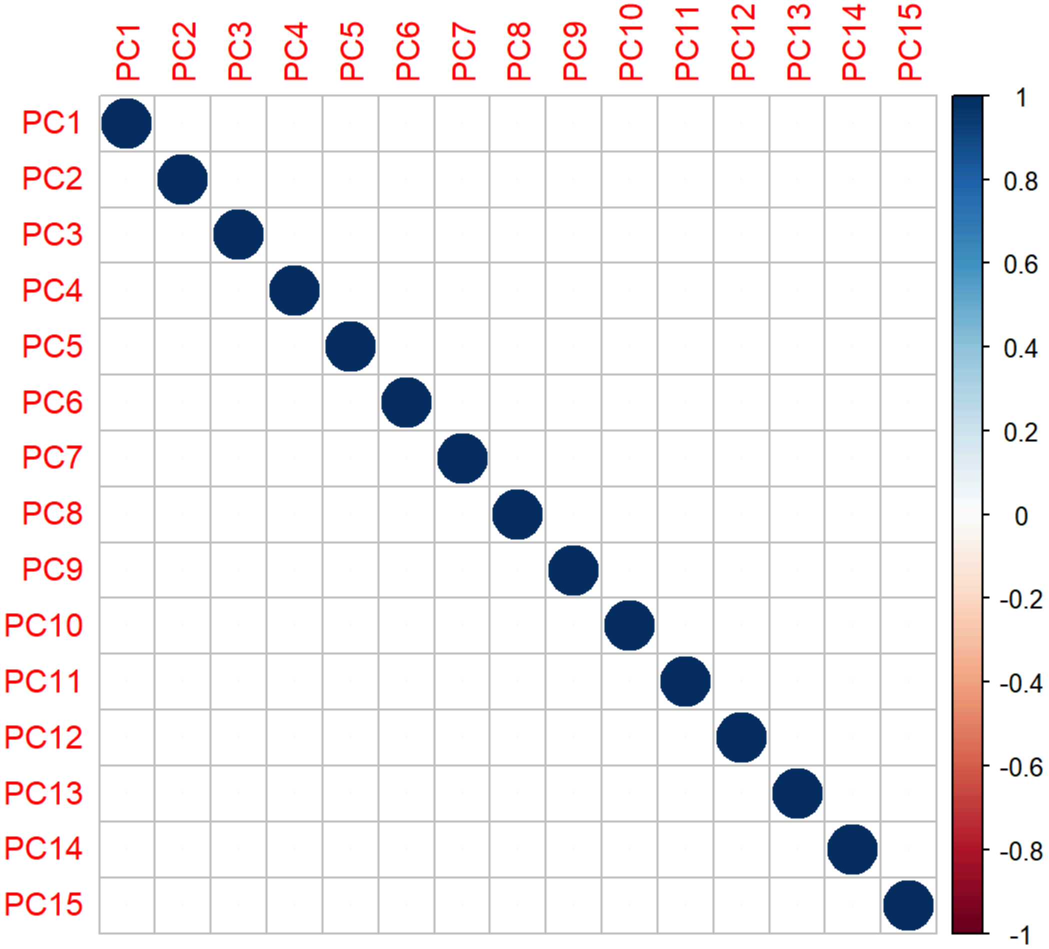
```
##              PC1         PC2         PC3         PC4
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946
```

```r
# Calculate the correlation matrix of the transformed data `PCA$x`
correlation <- cor(PCA$x)

# Print the correlation matrix
correlation
```

```
##               PC1           PC2           PC3           PC4           PC5
## PC1   1.000000e+00 -2.674053e-16 -4.446797e-17  1.396608e-16  7.765736e-17
## PC2  -2.674053e-16  1.000000e+00  4.129874e-16  1.298377e-16  5.840796e-16
## PC3  -4.446797e-17  4.129874e-16  1.000000e+00  1.310900e-16  1.876050e-16
## PC4   1.396608e-16  1.298377e-16  1.310900e-16  1.000000e+00  1.263862e-17
## PC5   7.765736e-17  5.840796e-16  1.876050e-16  1.263862e-17  1.000000e+00
## PC6   1.578046e-16  4.943278e-17 -6.827965e-16 -2.142774e-16 -4.132776e-16
## PC7   1.169191e-16  4.466221e-16 -2.394215e-16  5.124425e-16  2.246288e-16
## PC8   8.471166e-16 -5.649370e-16 -4.293413e-16 -1.087636e-16  2.231344e-16
## PC9   9.469639e-16  5.462106e-16 -7.884695e-18  5.143589e-16 -9.547874e-17
## PC10 -1.889891e-16 -3.813856e-16  2.546139e-16  3.399627e-16  2.604073e-16
## PC11 -7.647896e-16  6.303588e-16  1.735898e-16  5.232329e-16  1.161403e-16
## PC12  4.599447e-16 -8.367313e-16 -2.110088e-16  3.557463e-16  1.700170e-16
## PC13 -3.299901e-16  2.915364e-16 -3.952071e-16  1.718407e-16  2.264119e-17
## PC14  7.816555e-16  6.550527e-16 -2.748967e-16 -6.736474e-16  1.528453e-17
## PC15 -2.507007e-15  6.238920e-15 -5.523936e-15 -2.585697e-15 -4.870147e-15
##               PC6           PC7           PC8           PC9          PC10
## PC1   1.578046e-16  1.169191e-16  8.471166e-16  9.469639e-16 -1.889891e-16
## PC2   4.943278e-17  4.466221e-16 -5.649370e-16  5.462106e-16 -3.813856e-16
## PC3  -6.827965e-16 -2.394215e-16 -4.293413e-16 -7.884695e-18  2.546139e-16
## PC4  -2.142774e-16  5.124425e-16 -1.087636e-16  5.143589e-16  3.399627e-16
## PC5  -4.132776e-16  2.246288e-16  2.231344e-16 -9.547874e-17  2.604073e-16
## PC6   1.000000e+00  3.423091e-16 -4.016308e-16  3.722286e-16 -1.680444e-16
## PC7   3.423091e-16  1.000000e+00  6.443426e-16 -6.268662e-16 -4.370188e-16
## PC8  -4.016308e-16  6.443426e-16  1.000000e+00  5.216636e-16 -1.766598e-16
## PC9   3.722286e-16 -6.268662e-16  5.216636e-16  1.000000e+00 -1.029220e-15
## PC10 -1.680444e-16 -4.370188e-16 -1.766598e-16 -1.029220e-15  1.000000e+00
## PC11 -3.588790e-16  1.205844e-17 -1.562328e-16  1.431103e-16 -2.783576e-17
## PC12 -6.003186e-16  2.046892e-16  1.254683e-16  3.497507e-16  4.693773e-16
## PC13  4.823443e-16  2.807483e-17  3.346110e-16 -4.206628e-17  2.805453e-16
## PC14  6.702936e-16  4.258957e-16  7.999974e-17  6.884240e-16  5.517331e-16
## PC15 -6.480034e-16  1.721297e-16  8.688234e-16  2.580549e-15 -3.519377e-15
##              PC11          PC12          PC13          PC14          PC15
## PC1  -7.647896e-16  4.599447e-16 -3.299901e-16  7.816555e-16 -2.507007e-15
## PC2   6.303588e-16 -8.367313e-16  2.915364e-16  6.550527e-16  6.238920e-15
## PC3   1.735898e-16 -2.110088e-16 -3.952071e-16 -2.748967e-16 -5.523936e-15
## PC4   5.232329e-16  3.557463e-16  1.718407e-16 -6.736474e-16 -2.585697e-15
## PC5   1.161403e-16  1.700170e-16  2.264119e-17  1.528453e-17 -4.870147e-15
## PC6  -3.588790e-16 -6.003186e-16  4.823443e-16  6.702936e-16 -6.480034e-16
## PC7   1.205844e-17  2.046892e-16  2.807483e-17  4.258957e-16  1.721297e-16
## PC8  -1.562328e-16  1.254683e-16  3.346110e-16  7.999974e-17  8.688234e-16
## PC9   1.431103e-16  3.497507e-16 -4.206628e-17  6.884240e-16  2.580549e-15
## PC10 -2.783576e-17  4.693773e-16  2.805453e-16  5.517331e-16 -3.519377e-15
## PC11  1.000000e+00  4.855385e-16  4.396871e-16 -5.886122e-17  7.555887e-16
## PC12  4.855385e-16  1.000000e+00  1.732642e-16  3.447779e-16 -3.377997e-15
## PC13  4.396871e-16  1.732642e-16  1.000000e+00 -3.503023e-16 -2.122262e-15
## PC14 -5.886122e-17  3.447779e-16 -3.503023e-16  1.000000e+00  7.309460e-16
## PC15  7.555887e-16 -3.377997e-15 -2.122262e-15  7.309460e-16  1.000000e+00
```

```
# Plot the correlation matrix using the `corrplot` package
corrplot(correlation)
```



Build a regression on the first 4 PCs. Unsclae and Un

```
# Combine the first 4 principal components and the crime data (5th column) into a new data matri
x
pc_crime <- cbind(pcs, uscrime[,5])

# Print the newly created data matrix
pc_crime
```

```
##              PC1         PC2         PC3         PC4     
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115  5.6
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709  9.5
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995  4.4
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331 14.1
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305 10.1
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635 11.5
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873  7.9
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387 10.9
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150  6.2
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026  6.8
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210 11.6
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964  7.1
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618  6.0
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173  6.1
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211  5.3
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187  7.7
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865  6.3
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093 11.5
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766 12.8
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758 10.5
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548  6.7
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484  4.4
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773  8.3
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784  7.3
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518  5.7
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720 14.3
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615  7.1
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146  7.6
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974 15.7
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834  5.4
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099  5.4
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980  8.1
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089  6.4
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927  9.7
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766  8.7
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698  9.8
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399  5.6
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576  4.7
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645  5.4
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931  7.4
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429  6.6
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379  5.4
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015  7.0
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770  9.6
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984  4.1
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769  9.7
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946  9.1
```

```
# Convert the combined data matrix to a data frame (commented out)
# as.data.frame(pc_crime)

# Fit a linear model with the crime data (5th column) as the response variable and the principal
components as predictors
model <- lm(V5 ~ ., data = as.data.frame(pc_crime))

# Summarize the linear model to view the results
summary(model)
```

```
##
## Call:
## lm(formula = V5 ~ ., data = as.data.frame(pc_crime))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6037 -0.5085  0.0338  0.4547  2.0946
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.02340    0.11537  69.545  < 2e-16 ***
## PC1          0.86958    0.04753  18.294  < 2e-16 ***
## PC2         -0.73808    0.06967 -10.594 1.95e-13 ***
## PC3          0.14838    0.08236   1.802   0.0788 .
## PC4          0.98404    0.10817   9.097 1.75e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7909 on 42 degrees of freedom
## Multiple R-squared:  0.9269, Adjusted R-squared:   0.92
## F-statistic: 133.2 on 4 and 42 DF,  p-value: < 2.2e-16
```

```
# Get coefficients in terms of original data from PCA coefficients
# PCA coefficients for linear regression

# Extract the intercept (beta0) from the linear model coefficients.
beta0 <- model$coefficients[1]

# Extract the regression coefficients (betas) from the linear model, excluding the intercept.
betas <- model$coefficients[2:5]

# Display the intercept (beta0).
beta0
```

```
## (Intercept)
##    8.023404
```

```
# Display the regression coefficients (betas).
betas
```

```
##        PC1        PC2        PC3        PC4 
##  0.8695770 -0.7380754  0.1483771  0.9840373
```

```
# Transform PCA coefficients into coefficients for the original variables.
# Extract the eigenvectors (rotation matrix) corresponding to the first 4 principal components.
PCA$rotation[,1:4] # This is a 15 x 4 matrix.
```

```
##                PC1         PC2           PC3         PC4
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002
```

```
# Display the regression coefficients (betas) again for clarity.
betas
```

```
##        PC1        PC2        PC3        PC4 
##  0.8695770 -0.7380754  0.1483771  0.9840373
```

```
# Compute the regression coefficients (alphas) for the original variables by multiplying the rot
ation matrix and the betas.
# This is a matrix multiplication: 15 x 4 matrix (rotation) multiplied by 4 x 1 matrix (betas) r
esulting in a 15 x 1 matrix (alphas).
alphas <- PCA$rotation[,1:4] %*% betas

# Transpose the resulting alphas vector to match typical output format for coefficients.
t(alphas)
```

```
##               M        So        Ed       Po1       Po2         LF        M.F
## [1,] -0.304902 0.1192733 0.2254494 0.8029726 0.8194409 -0.1832533 -0.2096841
##            Pop        NW         U1         U2    Wealth       Ineq      Prob
## [1,] 0.4230202 0.1601283 -0.2483984 0.06857863 0.5045825 -0.3771979 0.1257321
##            Time
## [1,] -0.2361487
```

```
# Calculate the original alpha coefficients by dividing alphas by the standard deviations of the
original variables.
original_alpha <- alphas / sapply(uscrime[,1:15], sd)

# Get documentation on `sapply`.
?sapply

# Display the standard deviations of the original variables.
sapply(uscrime[,1:15], sd)
```

```
##            M           So           Ed          Po1          Po2           LF
##   1.25676339   0.47897516   1.11869985   2.97189736   2.79613186   0.04041181
##          M.F          Pop           NW           U1           U2       Wealth
##   2.94673654  38.07118801  10.28288187   0.01802878   0.84454499 964.90944200
##         Ineq         Prob         Time
##   3.98960606   0.02273697   7.08689519
```

```
# Adjust the constant term (beta0) by subtracting the sum of (alphas * mean / standard deviatio
n) for the original variables.
original_beta0 <- beta0 - sum(alphas * sapply(uscrime[,1:15], mean) / sapply(uscrime[,1:15], s
d))

# Display the means of the original variables.
sapply(uscrime[,1:15], mean)
```

```
##            M           So           Ed          Po1          Po2           LF
## 1.385745e+01 3.404255e-01 1.056383e+01 8.500000e+00 8.023404e+00 5.611915e-01
##          M.F          Pop           NW           U1           U2       Wealth
## 9.830213e+01 3.661702e+01 1.011277e+01 9.546809e-02 3.397872e+00 5.253830e+03
##         Ineq         Prob         Time
## 1.940000e+01 4.709138e-02 2.659792e+01
```

```
# Transpose the original alpha coefficients for better readability.
t(original_alpha)
```

```
##               M        So       Ed       Po1       Po2        LF         M.F
## [1,] -0.2426089 0.2490177 0.201528 0.2701885 0.2930623 -4.534647 -0.07115808
##             Pop         NW        U1         U2       Wealth        Ineq
## [1,] 0.01111129 0.01557232 -13.77788 0.08120186 0.0005229325 -0.09454515
##          Prob        Time
## [1,] 5.529851 -0.03332189
```

```
# Display the adjusted constant term.
original_beta0
```

```
## (Intercept)
##    14.25125
```

References:

[1] Wei, T., & Simko, V. (2021, November 18). An Introduction to corrplot Package. Cran.r-Project.org. https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html (https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html)

[2] TA office hour