Canonical Correlation Analysis

Blake Pappas

December 17, 2023

Load the Data and Libraries

```
Packages <- c("ggplot2","GGally", "ellipse", "RColorBrewer",
              "CCA", "CCP")
lapply(Packages, library, character.only = TRUE)
```

```
## [[1]]
## [1] "ggplot2"   "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "GGally"    "ggplot2"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "ellipse"   "GGally"    "ggplot2"   "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "RColorBrewer" "ellipse"      "GGally"       "ggplot2"      "stats"
##  [6] "graphics"     "grDevices"    "utils"        "datasets"     "methods"
## [11] "base"
##
## [[5]]
##  [1] "CCA"          "fields"       "viridis"      "viridisLite"  "spam"
##  [6] "fda"          "deSolve"      "fds"          "RCurl"        "rainbow"
## [11] "pcaPP"        "MASS"         "splines"      "RColorBrewer" "ellipse"
## [16] "GGally"       "ggplot2"      "stats"        "graphics"     "grDevices"
## [21] "utils"        "datasets"     "methods"      "base"
##
## [[6]]
##  [1] "CCP"          "CCA"          "fields"       "viridis"      "viridisLite"
##  [6] "spam"         "fda"          "deSolve"      "fds"          "RCurl"
## [11] "rainbow"      "pcaPP"        "MASS"         "splines"      "RColorBrewer"
## [16] "ellipse"      "GGally"       "ggplot2"      "stats"        "graphics"
## [21] "grDevices"    "utils"        "datasets"     "methods"      "base"
```

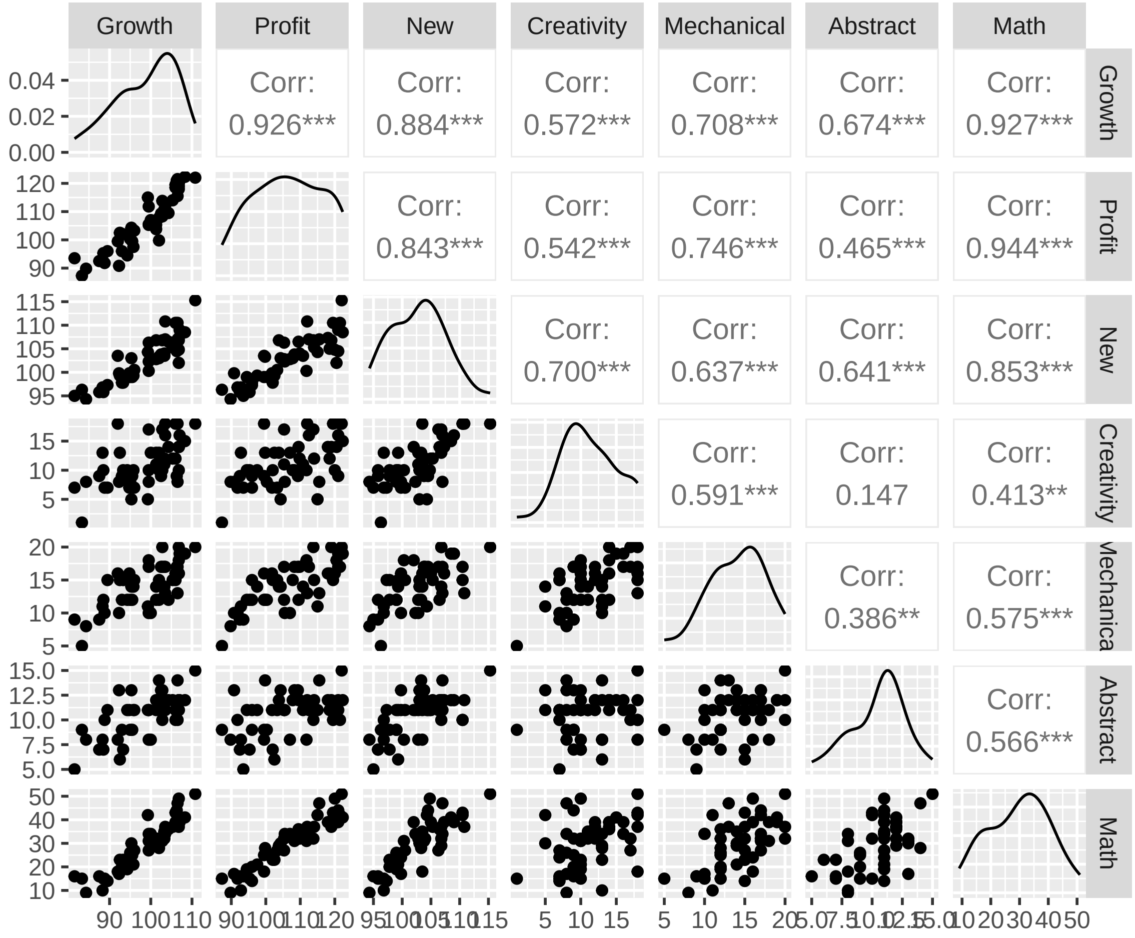
```
dat1 <- read.table("sales.txt") # Loads the data set
colnames(dat1) <- c("Growth", "Profit", "New",
                    "Creativity", "Mechanical", "Abstract", "Math") # Gives column names to the variabl
```

Summarize the Data

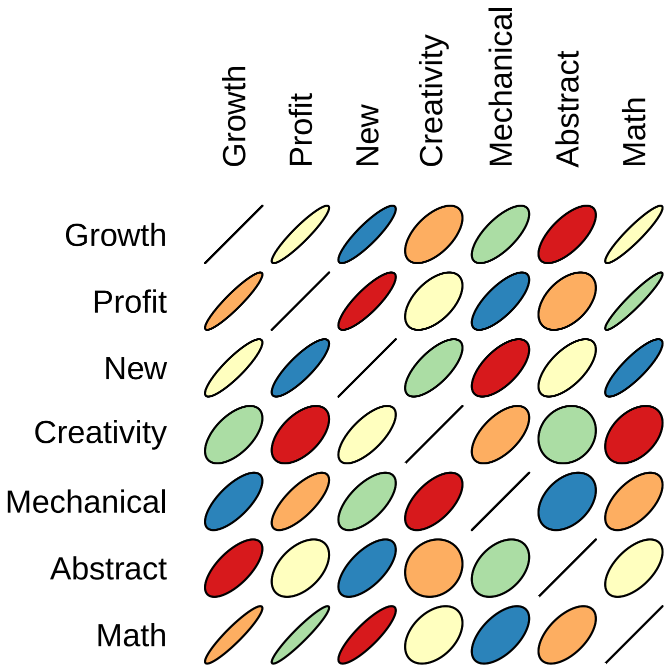
```
library(GGally)
summary(dat1) # Summarizes each variable in the data set
```

```
##      Growth           Profit            New           Creativity   
##  Min.   : 81.50   Min.   : 87.3   Min.   : 94.30   Min.   : 1.00  
##  1st Qu.: 93.55   1st Qu.: 99.5   1st Qu.: 99.08   1st Qu.: 8.25  
##  Median :100.65   Median :106.2   Median :103.15   Median :10.00  
##  Mean   : 98.84   Mean   :106.6   Mean   :102.81   Mean   :11.22  
##  3rd Qu.:105.05   3rd Qu.:114.8   3rd Qu.:106.45   3rd Qu.:14.00  
##  Max.   :110.80   Max.   :122.3   Max.   :115.30   Max.   :18.00  
##    Mechanical       Abstract          Math      
##  Min.   : 5.00   Min.   : 5.00   Min.   : 9.00  
##  1st Qu.:12.00   1st Qu.: 9.00   1st Qu.:21.50  
##  Median :15.00   Median :11.00   Median :31.50  
##  Mean   :14.18   Mean   :10.56   Mean   :29.76  
##  3rd Qu.:17.00   3rd Qu.:12.00   3rd Qu.:37.00  
##  Max.   :20.00   Max.   :15.00   Max.   :51.00
```

```
ggpairs(dat1) # Shows the scatterplot matrix of all variables in the data set
```



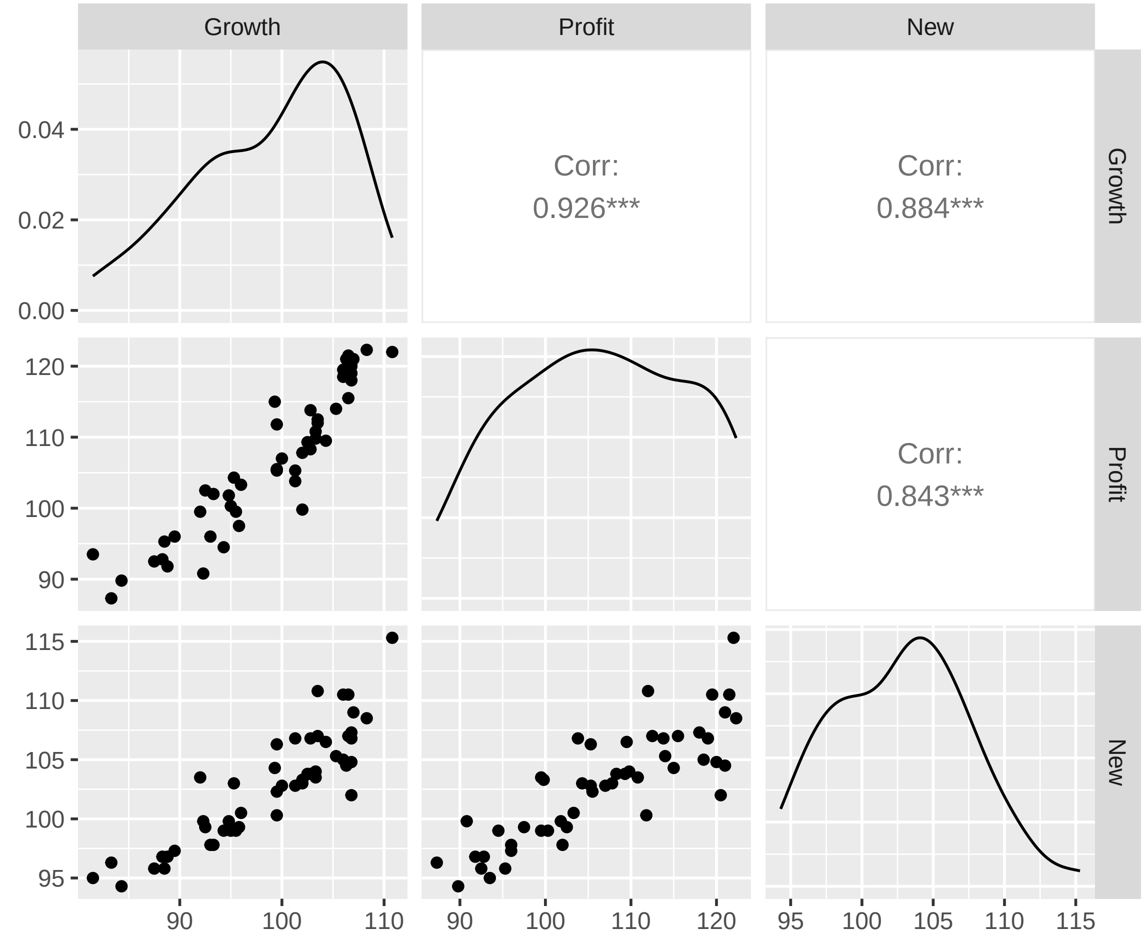
```r
my_colors <- brewer.pal(5, "Spectral")
plotcorr(cor(dat1), col = my_colors) # Visualizes the correlations based on ellipses. The narrower the
```



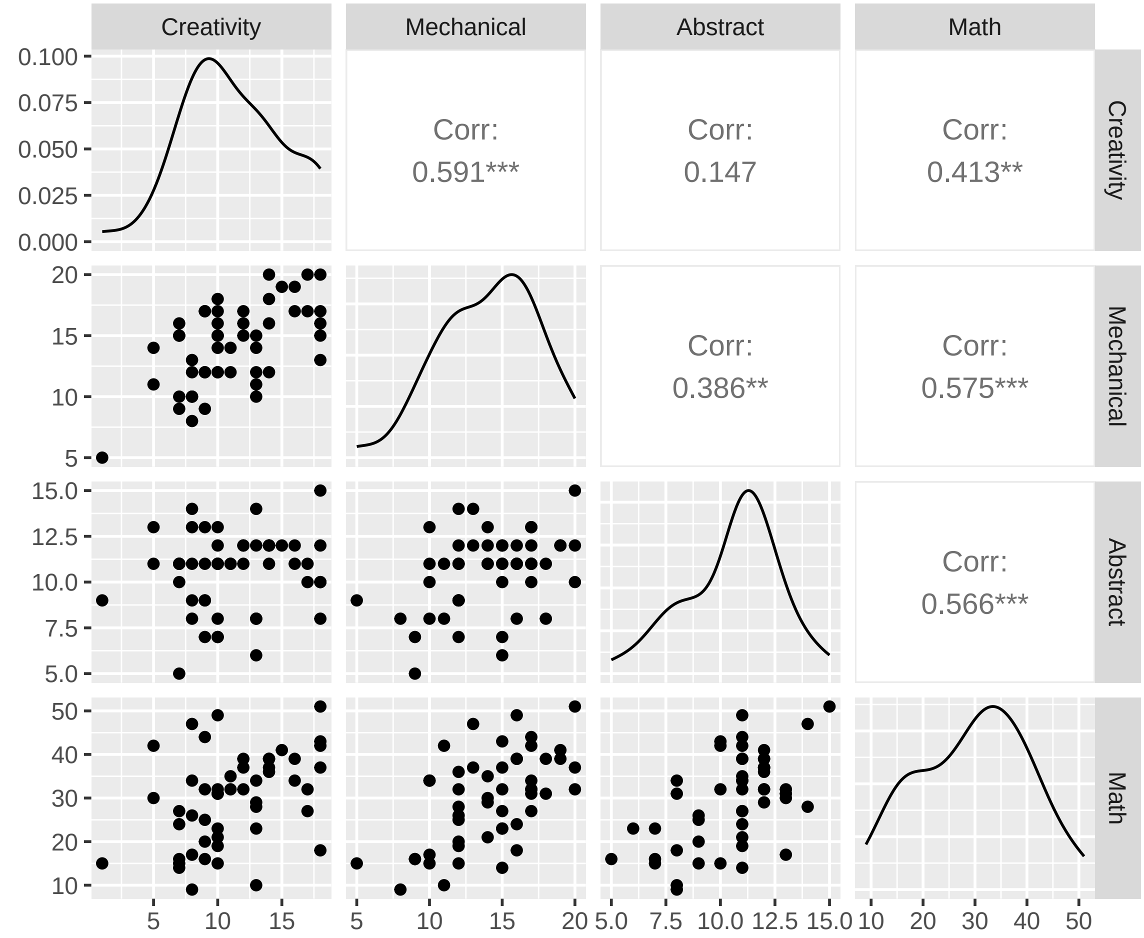
Let’s Examine *sales* and *intelligence*

```r
sales <- dat1[, 1:3] # Separates the sales variables into their own subset
intelligence <- dat1[, 4:7] # Separates the intelligence variables into their own subset

ggpairs(sales) # Shows the scatterplot matrix of the sales subset
```



```r
ggpairs(intelligence) # Shows the scatterplot matrix of the intelligence subset
```



```
# Calculate the Cross-Covariance Structure
matcor(sales, intelligence) # $Xcor calculates sigma(X), $Ycor calculates sigma(Y), and $XYcor calculat
```

```
## $Xcor
##           Growth    Profit       New
## Growth 1.0000000 0.9260758 0.8840023
## Profit 0.9260758 1.0000000 0.8425232
## New    0.8840023 0.8425232 1.0000000
##
## $Ycor
##             Creativity Mechanical  Abstract      Math
## Creativity   1.0000000  0.5907360 0.1469074 0.4126395
## Mechanical   0.5907360  1.0000000 0.3859502 0.5745533
## Abstract     0.1469074  0.3859502 1.0000000 0.5663721
## Math         0.4126395  0.5745533 0.5663721 1.0000000
##
## $XYcor
##                Growth    Profit       New Creativity Mechanical  Abstract
## Growth      1.0000000 0.9260758 0.8840023  0.5720363  0.7080738 0.6744073
## Profit      0.9260758 1.0000000 0.8425232  0.5415080  0.7459097 0.4653880
## New         0.8840023 0.8425232 1.0000000  0.7003630  0.6374712 0.6410886
## Creativity  0.5720363 0.5415080 0.7003630  1.0000000  0.5907360 0.1469074
## Mechanical 0.7080738 0.7459097 0.6374712  0.5907360  1.0000000 0.3859502
## Abstract   0.6744073 0.4653880 0.6410886  0.1469074  0.3859502 1.0000000
## Math       0.9273116 0.9442960 0.8525682  0.4126395  0.5745533 0.5663721
##                 Math
## Growth     0.9273116
## Profit     0.9442960
## New        0.8525682
## Creativity 0.4126395
## Mechanical 0.5745533
## Abstract   0.5663721
## Math       1.0000000
```

Test $H_0 : \Sigma_{XY} = 0$

```
# Tests of Canonical Dimensions
rho <- cc(sales, intelligence)$cor

## Define Number of Observations, Number of Variables In the First Set, and Number of Variables In the
n <- dim(sales)[1]
p <- length(sales)
q <- length(intelligence)

## Calculate p-Values Using the F-Approximations of Different Test Statistics
# library(CCP)
p.asym(rho, n, p, q, tstat = "Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##                 stat    approx df1      df2      p.value
## 1 to 3:  0.002148472 87.391525  12 114.0588 0.000000e+00
## 2 to 3:  0.195241267 18.526265   6  88.0000 8.248957e-14
## 3 to 3:  0.852846693  3.882233   2  45.0000 2.783536e-02
```

Canonical Correlation Analysis Using *cc* Function from *CCA* Package

```
# Calculate the Canonical Correlation
cc1 <- cc(sales, intelligence)
names(cc1)
```

```
## [1] "cor"    "names"  "xcoef"  "ycoef"  "scores"
```

```
cc1$cor # Canonical Correlation
```

```
## [1] 0.9944827 0.8781065 0.3836057
```

```
cc1$xcoef # Sales Matrix
```

```
##               [,1]       [,2]       [,3]
## Growth -0.06237788 -0.1740703  0.3771529
## Profit -0.02092564  0.2421641 -0.1035150
## New    -0.07825817 -0.2382940 -0.3834151
```

```
cc1$ycoef # Intelligence Matrix
```

```
##                   [,1]        [,2]        [,3]
## Creativity -0.06974814 -0.19239132 -0.24655659
## Mechanical -0.03073830  0.20157438  0.14189528
## Abstract   -0.08956418 -0.49576326  0.28022405
## Math       -0.06282997  0.06831607 -0.01133259
```

```
cc1$scores # Matrices
```

```
## $xscores
##              [,1]         [,2]        [,3]
##  [1,]  0.97838292 -0.362539552  0.81938141
##  [2,]  1.40651588 -0.410239408  0.05351720
##  [3,]  0.66973709  0.044672581  0.66847466
##  [4,] -0.40689705 -2.063089470 -0.30840196
##  [5,] -0.23688307 -0.310765017  0.99852234
##  [6,]  0.65494914 -0.844131320  1.14501451
##  [7,]  0.65528867 -0.236093843  0.93986313
##  [8,] -2.04552806 -1.334870222 -1.86845037
##  [9,] -0.35985473 -0.519574441  0.94175512
## [10,] -0.72379436  2.167475467  1.87763089
## [11,] -0.43808377 -0.291022268  0.89837606
## [12,]  0.04665613  1.736460981  0.67680067
## [13,] -0.74183310 -0.386875380 -0.45592909
## [14,]  0.02197133 -0.265760835  0.56211507
## [15,] -0.07973516 -0.108696870  0.40371149
## [16,]  1.96716552  1.701082028 -2.18552760
## [17,] -0.12525281 -0.746667218  1.06998582
## [18,] -0.41988032  0.070288835  0.98656859
## [19,]  0.25428846  0.007931722 -1.16609979
## [20,] -0.28687624 -1.267369772 -0.95084222
## [21,]  1.43024767  0.727850022 -0.03851613
## [22,] -0.32086304  1.593024005 -1.26353824
## [23,]  1.55121735  0.223860881 -0.12582704
## [24,] -0.75246285  0.068144053  0.71947930
## [25,] -1.29453998  0.585685240 -0.78260154
## [26,]  0.83411571  1.038223887  0.31143725
## [27,] -1.08624858  0.299106887  0.10431852
## [28,] -0.93245443  1.379170139  0.85582619
## [29,]  0.97434443 -1.976531657  0.32682226
## [30,] -0.89871368  1.779857589  0.67875923
## [31,] -1.31816090  0.039068369 -1.57960457
## [32,]  1.41677918 -0.081040167 -0.23857427
## [33,]  0.42719503  0.239653495  0.15995996
## [34,]  0.83477033 -1.238029918  1.00485465
## [35,] -1.39120112  0.436361392 -1.59805812
## [36,] -0.99174366 -0.182594033  0.36498468
## [37,]  0.52144946 -0.699170907 -2.10553997
## [38,] -0.09295538 -2.319565933  1.71161788
## [39,] -1.36370547  0.793354173 -0.23516470
## [40,] -1.06804514  0.660417991  0.19251105
## [41,] -0.36206701 -0.225189261  0.72509423
## [42,]  0.75616793  0.941121130 -0.61716522
## [43,] -0.70972028  0.097445954 -0.77782264
## [44,]  1.88288869 -0.423444109 -1.36329886
## [45,]  0.58821767  0.252097550  0.13103950
## [46,] -1.02875134 -1.413474738 -1.86114887
## [47,]  1.23583458  0.365853533 -0.30894632
## [48,]  1.92471305  0.484483866 -0.47810330
## [49,] -0.68982939 -1.133476875  0.34804581
## [50,] -0.86681530  1.107521445  0.63269334
##
## $yscores
##              [,1]        [,2]        [,3]
##  [1,]  0.97479103  0.09430244 -0.08851950
##  [2,]  1.40034960 -0.76140727  0.45769014
##  [3,]  0.66755933  0.69659017  0.09004153
##  [4,] -0.19984043 -1.14455925 -0.05227647
##  [5,] -0.20982423 -0.16086269  0.79529079
##  [6,]  0.60160796 -0.61815056  0.49782999
##  [7,]  0.66064116  0.43588278 -0.14518246
##  [8,] -2.38396289 -0.88140585  0.15766738
##  [9,] -0.29803503 -0.32179325  1.37063799
## [10,] -0.93127733  0.64827089 -0.12480193
## [11,] -0.41079711 -0.14249657  0.58596818
## [12,]  0.11904755  2.35859742  0.11141301
## [13,] -0.72588545 -0.27966647 -0.70314741
## [14,]  0.04726778 -0.15155656  0.27603833
## [15,] -0.03278039  0.37377660 -1.79741676
## [16,]  1.81607881  1.58415070 -1.09665800
## [17,] -0.09779329 -0.46221390 -0.15717568
## [18,] -0.34775982  0.14588306  0.09261709
## [19,]  0.20575055 -0.03287585  2.18906768
## [20,] -0.35582377 -0.95027027 -0.87037584
## [21,]  1.39832111  0.55185730 -0.83886122
## [22,] -0.27686589  1.17372033  1.06694262
## [23,]  1.49745418  0.20784154 -1.02932306
## [24,] -0.66347038 -0.20406500  0.24551465
## [25,] -1.19107609 -0.03070063 -0.19579577
## [26,]  0.80346642  1.70310899 -0.50383613
## [27,] -0.95936491 -0.25064113 -0.12836843
## [28,] -1.21910791  1.69784810  0.46430792
## [29,]  0.93624898 -2.30445623  1.02914053
## [30,] -0.86594820  1.75023346  0.90942276
## [31,] -1.27981072  0.04300999 -1.86226853
## [32,]  1.44440068 -1.06423465 -1.38353924
## [33,]  0.40313424  0.57049419  1.31139946
## [34,]  0.78874451 -1.15793146  0.23670462
## [35,] -1.27845734  0.37784268 -1.56714538
## [36,] -1.13042931 -0.14601431  1.39507261
## [37,]  0.43932868 -0.47179081 -1.99750652
## [38,] -0.25466221 -2.60755059  0.23571368
## [39,] -1.24698790  0.29832283  0.02809563
## [40,] -0.95665816  0.41902427  0.46187787
## [41,] -0.29111687 -0.06108586  1.60586199
## [42,]  0.68378617  1.62169828 -1.52372994
## [43,] -0.67262436  0.49179647 -0.78157702
## [44,]  2.06209412 -0.11916799  0.94732921
## [45,]  0.56088586  0.56712037  1.48729252
## [46,] -1.02048263 -1.76156170 -1.51761542
## [47,]  1.21992390 -0.31761469  1.45872318
## [48,]  1.94818626 -0.77531725 -0.56510955
## [49,] -0.64792179 -1.26188686 -0.66195176
## [50,] -0.73030445  0.62990478  0.08452069
##
## $corr.X.xscores
##              [,1]          [,2]         [,3]
## Growth -0.9798776  0.0006477883  0.199598477
## Profit -0.9464085  0.3228847489 -0.007504408
## New    -0.9518620 -0.1863009724 -0.243414776
##
## $corr.Y.xscores
##                  [,1]       [,2]        [,3]
## Creativity -0.6348095 -0.1894059 -0.24988439
## Mechanical -0.7171837  0.2086069  0.02598458
## Abstract   -0.6436782 -0.4402237  0.22027544
## Math       -0.9388771  0.1734549  0.03614570
##
## $corr.X.yscores
##              [,1]          [,2]         [,3]
## Growth -0.9744713  0.0005688272  0.076567107
## Profit -0.9411869  0.2835272081 -0.002878734
## New    -0.9466102 -0.1635921013 -0.093375287
##
## $corr.Y.yscores
##                  [,1]       [,2]        [,3]
## Creativity -0.6383313 -0.2156981 -0.65140953
## Mechanical -0.7211626  0.2375644  0.06773775
## Abstract   -0.6472493 -0.5013329  0.57422365
## Math       -0.9440859  0.1975329  0.09422619
```

Check

Compute the eigenvalues and eigenvectors of

$$\Sigma_X^{-1/2}\Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}\Sigma_X^{-1/2}$$

and

$$\Sigma_Y^{-1/2}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}\Sigma_Y^{-1/2}$$

```r
library(expm)
a <- solve(sqrtm(var(dat1[, 1:3]))) %*% var(dat1)[1:3, 4:7] %*% solve(var(dat1[, 4:7])) %*% var(dat1)[4

eigen(a)$values
```

```
## [1] 0.9889958 0.7710711 0.1471533
```

```
cc1$cor^2
```

```
## [1] 0.9889958 0.7710711 0.1471533
```

```
u_vec <- eigen(a)$vectors

u_vec[, 1] %*% solve(sqrtm(var(dat1[, 1:3])))
```

```
##              [,1]        [,2]        [,3]
## [1,] -0.06237788 -0.02092564 -0.07825817
```

```
cc1$xcoef[, 1]
```

```
##      Growth      Profit         New
## -0.06237788 -0.02092564 -0.07825817
```

```
b <- solve(sqrtm(var(dat1[, 4:7]))) %*% var(dat1)[4:7, 1:3] %*% solve(var(dat1[, 1:3])) %*% var(dat1)[1

eigen(b)$values
```

```
## [1] 9.889958e-01 7.710711e-01 1.471533e-01 7.771561e-16
```

```
cc1$cor^2
```

```
## [1] 0.9889958 0.7710711 0.1471533
```

```
v_vec <- eigen(b)$vectors

v_vec[, 1] %*% solve(sqrtm(var(dat1[, 4:7])))
```

```
##             [,1]      [,2]       [,3]       [,4]
## [1,] 0.06974814 0.0307383 0.08956418 0.06282997
```

```
cc1$ycoef[, 1]
```

```
##  Creativity  Mechanical    Abstract        Math
## -0.06974814 -0.03073830 -0.08956418 -0.06282997
```

Compute the Correlations Between $\{(U_i, V_i)\}_{i=1}^{3}$ and $\{X_i\}_{i=1}^{3}$ and $\{Y_j\}_{i=1}^{4}$

```
# Compute Canonical Loadings
cc2 <- comput(sales, intelligence, cc1)

# Display Canonical Loadings
cc2$corr.X.xscores # Correlations Between X's and U's
```

```
##              [,1]          [,2]         [,3]
## Growth -0.9798776  0.0006477883  0.199598477
## Profit -0.9464085  0.3228847489 -0.007504408
## New    -0.9518620 -0.1863009724 -0.243414776
```

```
cc2$corr.Y.xscores # Correlations Between Y ’s and V ’s
```

```
##                  [,1]       [,2]        [,3]
## Creativity -0.6348095 -0.1894059 -0.24988439
## Mechanical -0.7171837  0.2086069  0.02598458
## Abstract   -0.6436782 -0.4402237  0.22027544
## Math       -0.9388771  0.1734549  0.03614570
```

```
cc2$corr.X.yscores # Correlations Between X’s and V ’s
```

```
##              [,1]          [,2]         [,3]
## Growth -0.9744713  0.0005688272  0.076567107
## Profit -0.9411869  0.2835272081 -0.002878734
## New    -0.9466102 -0.1635921013 -0.093375287
```

```
cc2$corr.Y.yscores # Correlations Between Y ’s and U’s
```

```
##                  [,1]       [,2]        [,3]
## Creativity -0.6383313 -0.2156981 -0.65140953
## Mechanical -0.7211626  0.2375644  0.06773775
## Abstract   -0.6472493 -0.5013329  0.57422365
## Math       -0.9440859  0.1975329  0.09422619
```

```
# Check
cc1$xcoef[, 1] %*% var(dat1[, 1:3]) %*% diag(diag(var(dat1[, 1:3]))^(-0.5), 3)
```

```
##             [,1]       [,2]      [,3]
## [1,] -0.9798776 -0.9464085 -0.951862
```

```
cc2$corr.X.xscores[, 1]
```

```
##     Growth     Profit        New
## -0.9798776 -0.9464085 -0.9518620
```

```
cc1$ycoef[, 1] %*% var(dat1[, 4:7]) %*% diag(diag(var(dat1[, 4:7]))^(-0.5), 4)
```

```
##             [,1]       [,2]       [,3]       [,4]
## [1,] -0.6383313 -0.7211626 -0.6472493 -0.9440859
```

```
cc2$corr.Y.yscores[, 1]
```

```
## Creativity Mechanical   Abstract       Math
## -0.6383313 -0.7211626 -0.6472493 -0.9440859
```

Car Example

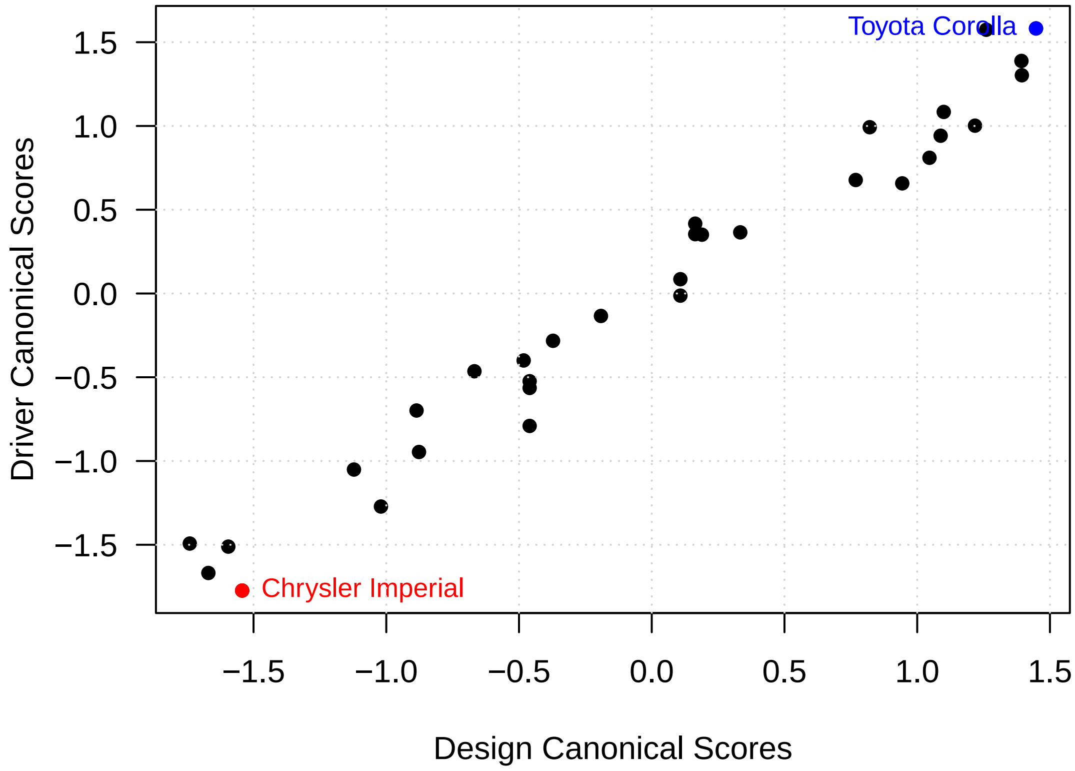
```r
vars <- colnames(mtcars)
design_vars <- which(vars %in% c("cyl", "disp", "carb", "drat", "gear", "vs"))
design <- mtcars[, design_vars]
driver_vars <- which(vars %in% c("mpg", "hp", "wt", "qsec", "am"))
driver <- mtcars[, driver_vars]
cancor(design, driver)
```

```
## $cor
## [1] 0.9850787 0.8471577 0.5796657 0.4137175 0.2548956
## 
## $xcoef
##               [,1]          [,2]         [,3]          [,4]         [,5]
## cyl  -0.0100612390 -0.0569725135 -0.189103848 -0.2479140068 -0.102007942
## disp -0.0009495869  0.0004777465  0.002595693  0.0012817966 -0.001723557
## drat  0.0168545333 -0.1114867476 -0.017162071  0.0115810636 -0.261447690
## vs   -0.0031559628  0.2008177747  0.187622423 -0.5772281070 -0.162098510
## gear  0.0092210270 -0.2019148908  0.045291715 -0.1449428572 -0.202846030
## carb -0.0412607448  0.0209358523  0.085311493 -0.0001123744  0.148096201
##               [,6]
## cyl  -0.0389624753
## disp  0.0002254826
## drat -0.5075998495
## vs   -0.0705387787
## gear  0.3172873054
## carb -0.0679174163
## 
## $ycoef
##               [,1]          [,2]        [,3]         [,4]         [,5]
## mpg  -0.0006532741 -0.0009491906 0.012578147  0.009797823 -0.077407171
## hp   -0.0008829583 -0.0003685299 0.002683266 -0.005001656 -0.001609325
## wt   -0.1051231423  0.0258415492 0.163035807  0.353350819 -0.189553488
## qsec  0.0277569776  0.0550748950 0.129469152 -0.103853140  0.072405438
## am    0.0317021519 -0.2174647296 0.430690734  0.194815128  0.333154777
## 
## $xcenter
##        cyl       disp       drat         vs       gear       carb 
##   6.187500 230.721875   3.596563   0.437500   3.687500   2.812500 
## 
## $ycenter
##       mpg        hp        wt      qsec        am 
##  20.09062 146.68750   3.21725  17.84875   0.40625
```

```r
cc(design, driver)
```

```
## $cor
## [1] 0.9850787 0.8471577 0.5796657 0.4137175 0.2548956
## 
## $names
## $names$Xnames
## [1] "cyl"  "disp" "drat" "vs"   "gear" "carb"
## 
## $names$Ynames
## [1] "mpg"  "hp"   "wt"   "qsec" "am"
##
## $names$ind.names
##  [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
##  [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
##  [7] "Duster 360"          "Merc 240D"           "Merc 230"
## [10] "Merc 280"            "Merc 280C"           "Merc 450SE"
## [13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
## [16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
## [22] "Dodge Challenger"    "AMC Javelin"         "Camaro Z28"
## [25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"
## [28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
## [31] "Maserati Bora"       "Volvo 142E"
##
##
## $xcoef
##              [,1]        [,2]        [,3]         [,4]         [,5]
## cyl  -0.056018608 -0.31720953 -1.05288566 -1.380326772  0.567956182
## disp -0.005287076  0.00265998  0.01445221  0.007136742  0.009596359
## drat  0.093842070 -0.62073194 -0.09555437  0.064480633  1.455679133
## vs   -0.017571657  1.11810605  1.04463744 -3.213870084  0.902526305
## gear  0.051340506 -1.12421453  0.25217360 -0.807007675  1.129398896
## carb -0.229730104  0.11656589  0.47499429 -0.000625674 -0.824564751
##
## $ycoef
##              [,1]         [,2]       [,3]        [,4]         [,5]
## mpg  -0.003637276 -0.005284870 0.07003216  0.05455197  0.430984888
## hp   -0.004916104 -0.002051888 0.01493980 -0.02784804  0.008960343
## wt   -0.585300885  0.143879657 0.90774496  1.96737410  1.055389153
## qsec  0.154544311  0.306644038 0.72085373 -0.57822981 -0.403136420
## am    0.176510111 -1.210792371 2.39798452  1.08468473 -1.854927293
##
## $scores
## $scores$xscores
##                           [,1]        [,2]        [,3]         [,4]        [,5]
## Mazda RX4            0.1638176 -1.01906175 -0.66783496  0.926788638 -1.36454541
## Mazda RX4 Wag        0.1638176 -1.01906175 -0.66783496  0.926788638 -1.36454541
## Datsun 710           1.2177093  0.27648334  0.31085379  0.101114525  0.30396815
## Hornet 4 Drive       0.1890118  1.64323935  0.19431710 -0.831670189  0.62906257
## Hornet Sportabout   -0.6678966  0.23514682 -1.01366086  0.353381966  1.11861005
## Valiant              0.3334559  1.75409424 -0.25202841 -1.087816465 -0.15343461
## Duster 360          -1.1217263  0.43103469 -0.06940554  0.355999456 -0.44317870
## Merc 240D            0.7683547  0.59530755  1.36043727  0.366363851 -0.38212616
## Merc 230             0.8211321  0.43684532  1.25319173  0.339087621 -0.10393848
## Merc 280             0.1079410  0.10684550  0.48472819 -2.231552596 -0.35997319
## Merc 280C            0.1079410  0.10684550  0.48472819 -2.231552596 -0.35997319
## Merc 450SE          -0.4599623  0.17740098 -1.74789823 -0.253315803 -0.63042247
## Merc 450SL          -0.4599623  0.17740098 -1.74789823 -0.253315803 -0.63042247
## Merc 450SLC         -0.4599623  0.17740098 -1.74789823 -0.253315803 -0.63042247
## Cadillac Fleetwood  -1.7401546  0.90275736  1.57599711  1.137259941  0.22402338
## Lincoln Continental -1.6701407  0.82738637  1.39588179  1.056132686  0.21076460
## Chrysler Imperial   -1.5428155  0.63141843  1.08486010  0.928228399  0.35364362
## Fiat 128             1.3942043  0.05577758 -0.13457345 -0.093161459  0.35760102
## Honda Civic          1.2601012 -0.36325861  0.21584300 -0.060388820  0.74157446
## Toyota Corolla       1.4475240 -0.05134073 -0.25778785 -0.138373407  0.48846377
## Toyota Corona        1.0883189  1.52599341  0.24788508  0.984804679 -0.92766667
## Dodge Challenger    -0.4824378  0.36551314 -1.58338744  0.028491371  0.14784811
## AMC Javelin         -0.3718204  0.08618796 -1.82298457 -0.046275565  0.58121394
## Camaro Z28          -1.0200577  0.08165429 -0.26361590  0.318161969  0.21781086
## Pontiac Firebird    -0.8859486  0.38499725 -0.42888369  0.634337987  1.40056688
## Fiat X1-9            1.3926182  0.05657558 -0.13023778 -0.091020437  0.36047993
## Porsche 914-2        1.0462888 -2.17657813  0.11572487  2.632531949  0.66860510
## Lotus Europa         1.1000157 -0.71582049  0.85923252 -0.803741241  0.36855493
## Ford Pantera L      -0.8766811 -2.46827345  0.20836186 -1.257121128  3.19948778
## Ferrari Dino        -0.1912717 -1.77623925  0.34429931 -0.006576086 -2.43581156
## Maserati Bora       -1.5950604 -1.71291114  1.45070554 -1.660307736 -1.56845099
## Volvo 142E           0.9436462  0.26623866  0.94888266  0.210031457 -0.01736736
## 
## $scores$yscores
##                            [,1]       [,2]        [,3]        [,4]        [,5]
## Mazda RX4            0.41680233 -1.1602193 -0.60385124  1.34331735 -1.10864425
## Mazda RX4 Wag        0.35409542 -0.9518093  0.03130182  1.52118905 -1.06527641
## Datsun 710           1.00168953 -0.5187292  0.54574215  0.08152132 -1.66855733
## Hornet 4 Drive       0.35112532  1.0478667 -0.28557053 -0.27208384  0.34528701
## Hornet Sportabout   -0.46429070  0.2190576 -1.04379406 -0.38752172  0.97710281
## Valiant              0.36485470  1.3499991  0.19328779 -0.28187773 -1.17764090
## Duster 360          -1.05086535 -0.2444568 -1.03875043 -1.63884367  0.32089488
## Merc 240D            0.67736381  1.2986264 -0.41159978  0.85528514  0.95600409
## Merc 230             0.99254257  2.1228824  2.02352804 -1.90654489 -0.64919160
## Merc 280            -0.01265523  0.7156176 -0.86295458  0.34771840  0.21064280
## Merc 280C            0.08516354  0.9070029 -0.52848736 -0.07559225 -0.63461790
## Merc 450SE          -0.79051820  0.4281222 -0.28436531  0.36748688  0.54234261
## Merc 450SL          -0.56388059  0.4357756 -0.38579890 -0.36796950  0.49076941
## Merc 450SLC         -0.52368963  0.5767254 -0.19913770 -0.61545186 -0.52278397
## Cadillac Fleetwood  -1.49261648  0.7561658  1.15817084  1.33010209 -0.80801807
## Lincoln Continental -1.66834697  0.7116189  1.35017982  1.48646151 -0.47027510
## Chrysler Imperial   -1.77330776  0.5240916  1.51536169  1.37918370  1.59524389
## Fiat 128             1.30265069 -0.2676147  1.32568121  0.62375491  1.75362431
## Honda Civic          1.57433462 -0.6037999 -0.23938209  0.30292806  0.53178668
## Toyota Corolla       1.58219976 -0.1941492  1.39442985 -0.23329945  1.83257560
## Toyota Corona        0.94173687  1.1408902 -0.74270675 -1.70972564 -0.74942858
## Dodge Challenger    -0.39975454  0.2527801 -1.67690032  0.37823747 -0.48115582
## AMC Javelin         -0.28245873  0.3739927 -1.46510118 -0.05399374 -0.87350802
## Camaro Z28          -1.27171336 -0.3321813 -1.17365855 -0.91356582  0.34821372
## Pontiac Firebird    -0.69851987  0.2838857 -0.61951566  0.41919388  1.60793377
## Fiat X1-9            1.38821527 -0.4535771  0.31707816  0.15377671 -0.49428898
## Porsche 914-2        0.81005698 -1.1431255 -0.80025924  1.06207532  0.27269414
## Lotus Europa         1.08379120 -1.2404042 -0.58842758 -0.65974249  1.62379891
## Ford Pantera L      -0.94618614 -1.9706172  0.41909647 -1.01356551 -0.59926143
## Ferrari Dino        -0.13417363 -1.5595180 -0.27966417  0.31248365 -0.54118297
## Maserati Bora       -1.51098561 -2.0238570  1.85897958 -2.30529156  0.07397704
## Volvo 142E           0.65734020 -0.4810424  1.09708800  0.47035424 -1.63906031
## 
## $scores$corr.X.xscores
##            [,1]        [,2]       [,3]         [,4]        [,5]
## cyl  -0.9332686  0.01412088 -0.3431599 -0.085215124  0.06115658
## disp -0.9424652  0.18920878 -0.0430631  0.117537554  0.23939445
## drat  0.6418747 -0.51877788  0.2852497  0.002088073  0.12781781
## vs    0.7790313  0.36441654  0.3574816 -0.356355409  0.06049635
## gear  0.3828007 -0.79798477  0.4003617 -0.182157946 -0.05402306
## carb -0.6717294 -0.49824957  0.2394798 -0.194397353 -0.42892252
##
## $scores$corr.Y.xscores
##            [,1]       [,2]       [,3]        [,4]         [,5]
## mpg   0.8915483 -0.1506162 0.06576487  0.01347160  0.093771816
## hp   -0.9006692 -0.2239542 0.04551045 -0.12258440  0.003514662
## wt   -0.8717485  0.3375017 0.08292516  0.07763919 -0.012638355
## qsec  0.5767753  0.6341267 0.17921691 -0.00169870 -0.009073941
## am    0.4826987 -0.6883227 0.16779894  0.04438408 -0.016936476
##
## $scores$corr.X.yscores
##            [,1]        [,2]       [,3]          [,4]        [,5]
## cyl  -0.9193430  0.01196261 -0.1989180 -0.0352549903  0.01558855
## disp -0.9284024  0.16028967 -0.0249622  0.0486273465  0.06102060
## drat  0.6322971 -0.43948668  0.1653495  0.0008638726  0.03258020
## vs    0.7674071  0.30871828  0.2072198 -0.1474304788  0.01542026
## gear  0.3770888 -0.67601895  0.2320760 -0.0753619353 -0.01377024
## carb -0.6617063 -0.42209596  0.1388182 -0.0804255923 -0.10933048
##
## $scores$corr.Y.yscores
##            [,1]       [,2]       [,3]         [,4]        [,5]
## mpg   0.9050529 -0.1777901 0.11345311  0.032562318  0.36788316
## hp   -0.9143119 -0.2643596 0.07851155 -0.296299747  0.01378863
## wt   -0.8849532  0.3983930 0.14305686  0.187662310 -0.04958247
## qsec  0.5855119  0.7485344 0.30917287 -0.004105941 -0.03559865
## am    0.4900103 -0.8125084 0.28947537  0.107281129 -0.06644474
```

```r
ccs <- cc(design, driver)
descc1 <- ccs$scores$xscores[, 1]; drivcc1 <- ccs$scores$yscores[, 1]
sdr <- sort(drivcc1)
sdr <- sdr[c(1, length(sdr))] # First and Last
ext <- match(sdr, drivcc1)
plot(descc1, drivcc1, xlab = "Design Canonical Scores",
     ylab = "Driver Canonical Scores", las = 1,
     pch = 16)
points(descc1[ext], drivcc1[ext], pch = 16, col = c("red", "blue"))
text(descc1[ext], drivcc1[ext], labels = rownames(mtcars)[ext],
     pos = c(4, 2), cex = 0.8, col = c("red", "blue"))
grid()
```



```
cancor(design, driver)$cor; cor(descc1, drivcc1)
```

```
## [1] 0.9850787 0.8471577 0.5796657 0.4137175 0.2548956
```

```
## [1] 0.9850787
```