S17403 - Mini Project 02

S17403

2023-08-01

Import data set, EDA and data cleaning

```
library(stats)

library(readr)
data <- read_csv("E:/Sumedha(important)/4th Year 1st Sem/Statistics/ST 405 - Multivariate methods II/pro
                 skip = 1)
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 246 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (14): day, month, year, Temperature, RH, Ws, Rain, FFMC, DMC, DC, ISI, B...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#View(data)
problems()

new_data <- data[-c(123, 124), ]

#View(new_data)
# Assuming your dataset is stored in 'data', check the structure
str(new_data)
```

```
## tibble [244 x 14] (S3: tbl_df/tbl/data.frame)
##  $ day        : chr [1:244] "01" "02" "03" "04" ...
##  $ month      : chr [1:244] "06" "06" "06" "06" ...
##  $ year       : chr [1:244] "2012" "2012" "2012" "2012" ...
##  $ Temperature: chr [1:244] "29" "29" "26" "25" ...
##  $ RH         : chr [1:244] "57" "61" "82" "89" ...
##  $ Ws         : chr [1:244] "18" "13" "22" "13" ...
##  $ Rain       : chr [1:244] "0" "1.3" "13.1" "2.5" ...
##  $ FFMC       : chr [1:244] "65.7" "64.4" "47.1" "28.6" ...
##  $ DMC        : chr [1:244] "3.4" "4.1" "2.5" "1.3" ...
##  $ DC         : chr [1:244] "7.6" "7.6" "7.1" "6.9" ...
##  $ ISI        : chr [1:244] "1.3" "1" "0.3" "0" ...
```

```
##  $ BUI        : chr [1:244] "3.4" "3.9" "2.7" "1.7" ...
##  $ FWI        : chr [1:244] "0.5" "0.4" "0.1" "0" ...
##  $ Classes    : chr [1:244] "not fire" "not fire" "not fire" "not fire" ...
```

```
head(new_data)
```

```
## # A tibble: 6 x 14
##   day   month year  Temperature RH    Ws    Rain  FFMC  DMC   DC    ISI   BUI
##   <chr> <chr> <chr> <chr>       <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 01    06    2012  29          57    18    0     65.7  3.4   7.6   1.3   3.4
## 2 02    06    2012  29          61    13    1.3   64.4  4.1   7.6   1     3.9
## 3 03    06    2012  26          82    22    13.1  47.1  2.5   7.1   0.3   2.7
## 4 04    06    2012  25          89    13    2.5   28.6  1.3   6.9   0     1.7
## 5 05    06    2012  27          77    16    0     64.8  3     14.2  1.2   3.9
## 6 06    06    2012  31          67    14    0     82.6  5.8   22.2  3.1   7
## # ... with 2 more variables: FWI <chr>, Classes <chr>
## # i Use `colnames()` to see all variable names
```

```
# Check for missing values
sum(is.na(new_data))
```

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

```
# Remove rows with any missing values
new_data<- na.omit(new_data)
# Check for missing values
sum(is.na(new_data))
```

```
## [1] 0
```

Convert char into numerical data

```
library(car)
```

```
## Loading required package: carData
```

```
library(dplyr)
```

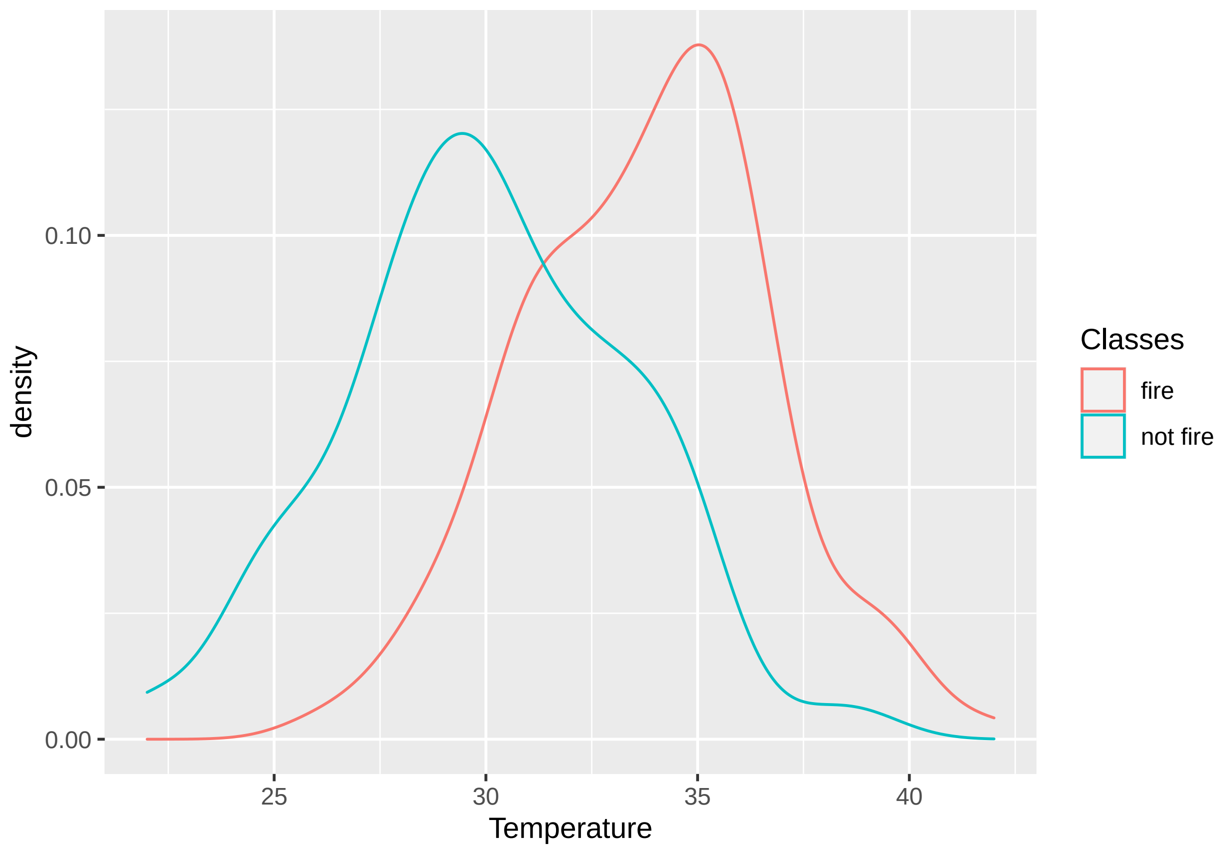
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
new_data <- new_data %>%
  mutate_at(vars(FFMC,DMC,DC,ISI,BUI,FWI), as.numeric)
```

```r
new_data <- new_data %>%
  mutate_at(vars(Temperature, RH,Ws,Rain), as.numeric)

# Check the class and structure of the data frame
str(new_data)
```

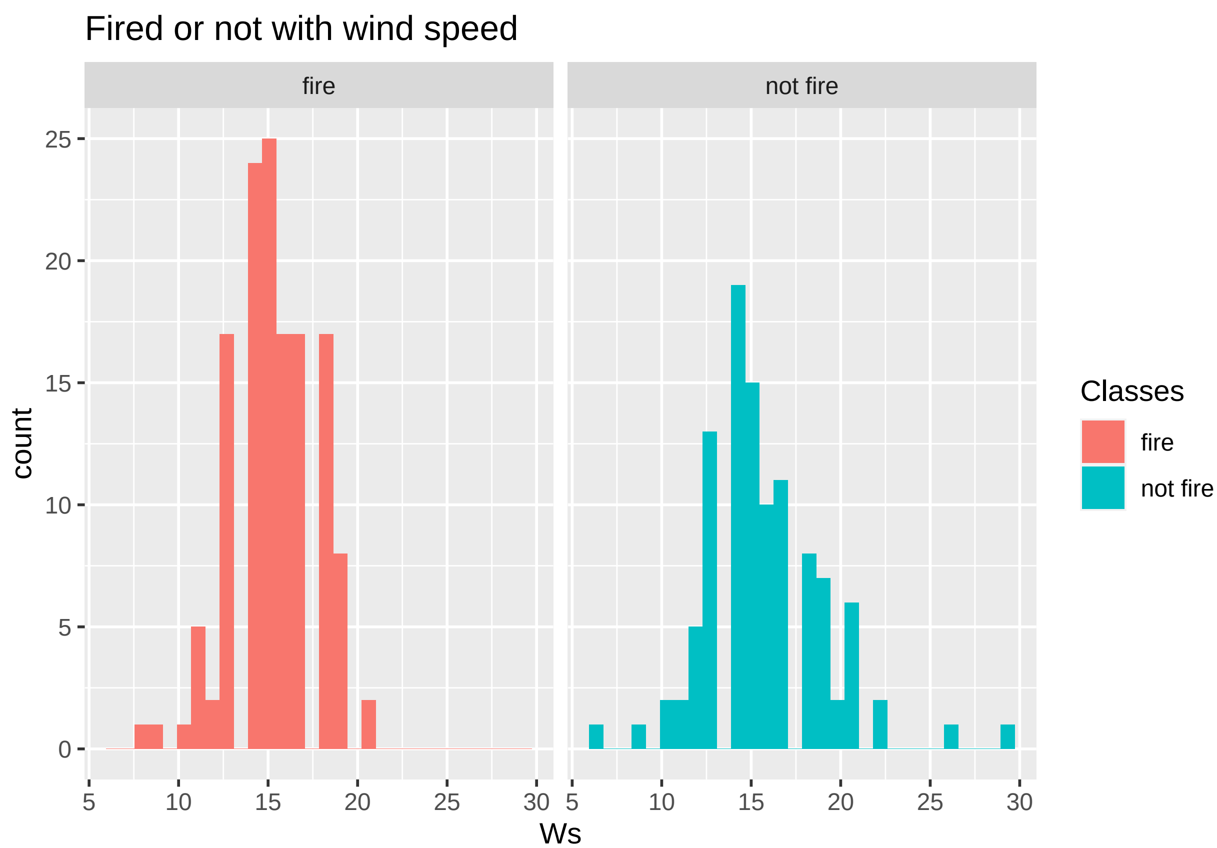
```
## tibble [243 x 14] (S3: tbl_df/tbl/data.frame)
##  $ day        : chr [1:243] "01" "02" "03" "04" ...
##  $ month      : chr [1:243] "06" "06" "06" "06" ...
##  $ year       : chr [1:243] "2012" "2012" "2012" "2012" ...
##  $ Temperature: num [1:243] 29 29 26 25 27 31 33 30 25 28 ...
##  $ RH         : num [1:243] 57 61 82 89 77 67 54 73 88 79 ...
##  $ Ws         : num [1:243] 18 13 22 13 16 14 13 15 13 12 ...
##  $ Rain       : num [1:243] 0 1.3 13.1 2.5 0 0 0 0 0.2 0 ...
##  $ FFMC       : num [1:243] 65.7 64.4 47.1 28.6 64.8 82.6 88.2 86.6 52.9 73.2 ...
##  $ DMC        : num [1:243] 3.4 4.1 2.5 1.3 3 5.8 9.9 12.1 7.9 9.5 ...
##  $ DC         : num [1:243] 7.6 7.6 7.1 6.9 14.2 22.2 30.5 38.3 38.8 46.3 ...
##  $ ISI        : num [1:243] 1.3 1 0.3 0 1.2 3.1 6.4 5.6 0.4 1.3 ...
##  $ BUI        : num [1:243] 3.4 3.9 2.7 1.7 3.9 7 10.9 13.5 10.5 12.6 ...
##  $ FWI        : num [1:243] 0.5 0.4 0.1 0 0.5 2.5 7.2 7.1 0.3 0.9 ...
##  $ Classes    : chr [1:243] "not fire" "not fire" "not fire" "not fire" ...
##  - attr(*, "na.action")= 'omit' Named int 166
##   ..- attr(*, "names")= chr "166"
```

```r
library(ggplot2)
ggplot(new_data, aes(x=Temperature, col=Classes)) + geom_density()
```




```
ggplot(new_data, aes(x=Ws, fill=Classes)) + geom_histogram() + facet_wrap(~Classes) + ggtitle("Fired or
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Summary of the data set

```
summary(new_data)
```

```
##      day                month               year           Temperature
##  Length:243         Length:243         Length:243         Min.   :22.00
##  Class :character   Class :character   Class :character   1st Qu.:30.00
##  Mode  :character   Mode  :character   Mode  :character   Median :32.00
##                                                           Mean   :32.15
##                                                           3rd Qu.:35.00
##                                                           Max.   :42.00
##        RH              Ws             Rain              FFMC
##  Min.   :21.00   Min.   : 6.00   Min.   : 0.000   Min.   :28.60
##  1st Qu.:52.50   1st Qu.:14.00   1st Qu.: 0.000   1st Qu.:71.85
##  Median :63.00   Median :15.00   Median : 0.000   Median :83.30
##  Mean   :62.04   Mean   :15.49   Mean   : 0.763   Mean   :77.84
##  3rd Qu.:73.50   3rd Qu.:17.00   3rd Qu.: 0.500   3rd Qu.:88.30
##  Max.   :90.00   Max.   :29.00   Max.   :16.800   Max.   :96.00
##       DMC              DC              ISI              BUI
##  Min.   : 0.70   Min.   :  6.90   Min.   : 0.000   Min.   : 1.10
##  1st Qu.: 5.80   1st Qu.: 12.35   1st Qu.: 1.400   1st Qu.: 6.00
##  Median :11.30   Median : 33.10   Median : 3.500   Median :12.40
##  Mean   :14.68   Mean   : 49.43   Mean   : 4.742   Mean   :16.69
##  3rd Qu.:20.80   3rd Qu.: 69.10   3rd Qu.: 7.250   3rd Qu.:22.65
##  Max.   :65.90   Max.   :220.40   Max.   :19.000   Max.   :68.00
##       FWI          Classes
```

```
##  Min.   : 0.000   Length:243
##  1st Qu.: 0.700   Class :character
##  Median : 4.200   Mode  :character
##  Mean   : 7.035
##  3rd Qu.:11.450
##  Max.   :31.100
```

```
xtabs(~Classes, data = new_data)
```

```
## Classes
##     fire not fire
##      137      106
```

```
#str(new_data)
```

```
sd_data<-new_data[, c("Temperature", "RH","Ws","Rain","FFMC", "DMC","DC","ISI","BUI","FWI")]
sd_data <-apply(sd_data,2,scale)
```

Extract the two multivariate sets of variables

```
Weather <- new_data[, c("Temperature", "RH","Ws","Rain")]
FWI_compo <- new_data[, c("FFMC", "DMC","DC","ISI","BUI","FWI")]
```

```
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.2.3
```

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

```
library(CCA)
```

```
## Warning: package 'CCA' was built under R version 4.2.3
```

```
## Loading required package: fda
```

```
## Warning: package 'fda' was built under R version 4.2.3
```

```
## Loading required package: splines
```

```
## Loading required package: fds
```

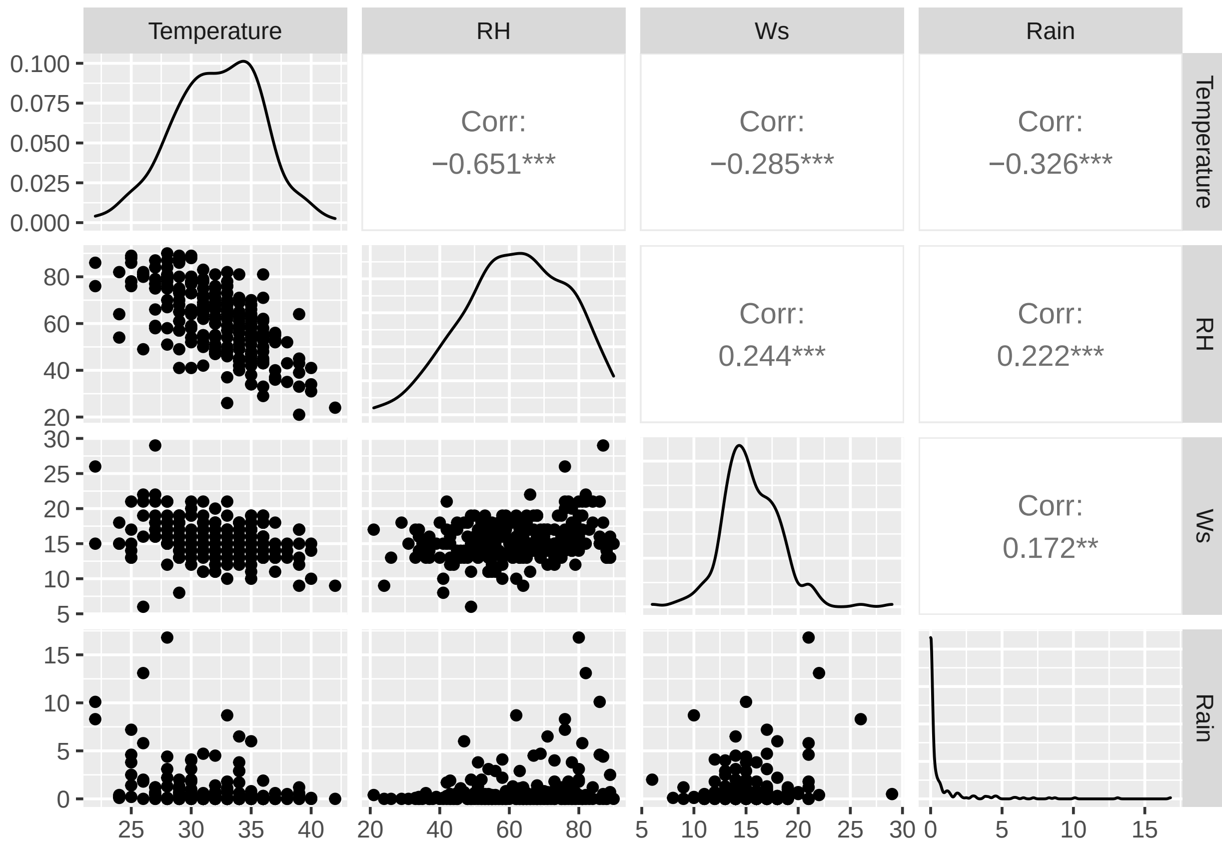
```
## Warning: package 'fds' was built under R version 4.2.3
```

```
## Loading required package: rainbow
```

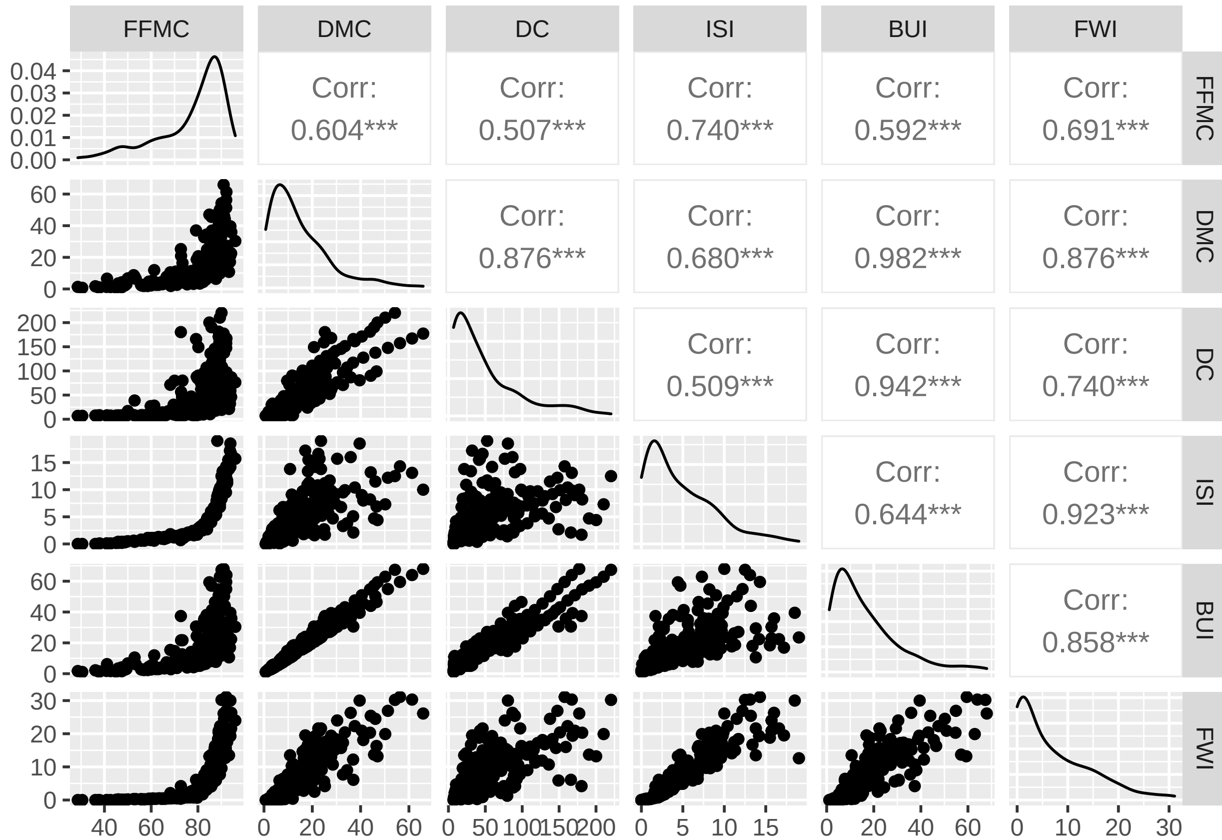
```
## Warning: package 'rainbow' was built under R version 4.2.3

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.2.2

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: pcaPP

## Warning: package 'pcaPP' was built under R version 4.2.3

## Loading required package: RCurl

## Loading required package: deSolve

##
## Attaching package: 'fda'

## The following object is masked from 'package:graphics':
##
##     matplot

## Loading required package: fields

## Warning: package 'fields' was built under R version 4.2.3

## Loading required package: spam

## Warning: package 'spam' was built under R version 4.2.3

## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: viridis

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

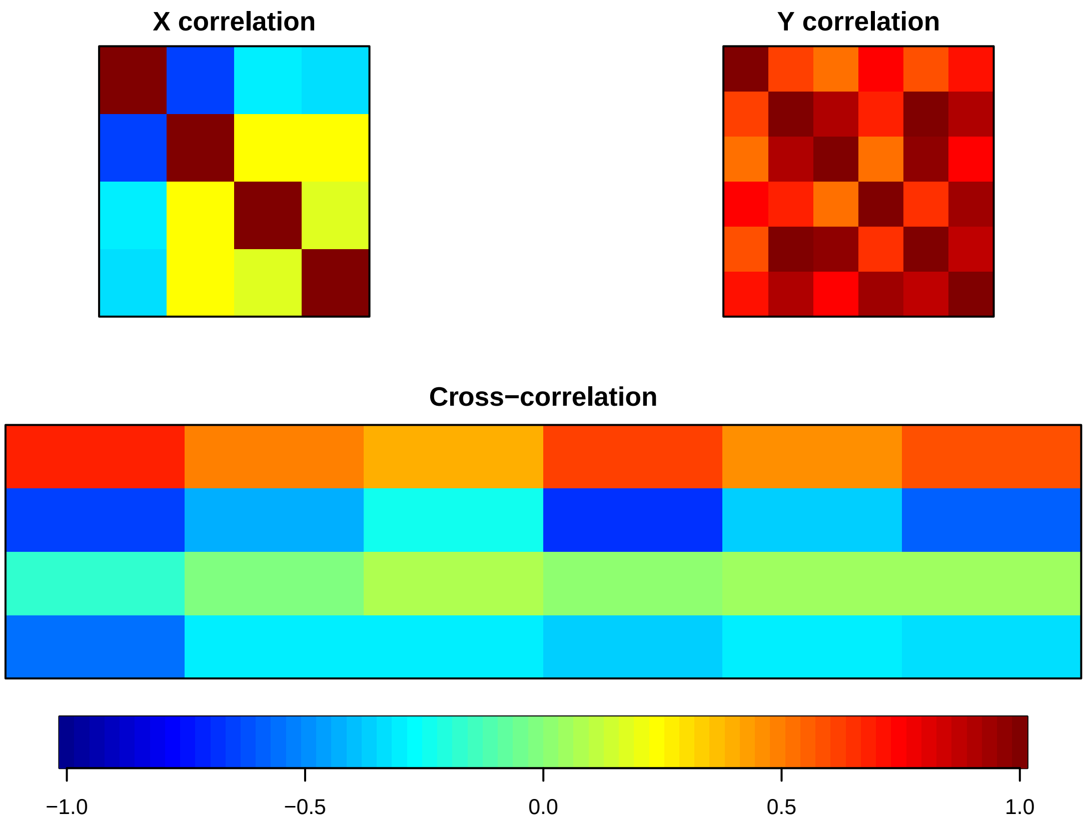
```r
ggpairs(Weather)
```




```r
ggpairs(FWI_compo)
```




```r
library("CCA")
correl <- matcor(Weather, FWI_compo)
img.matcor(correl, type = 2)
```



Correlations

```r
# correlations
matcor(Weather, FWI_compo)
```

```
## $Xcor
##             Temperature         RH         Ws       Rain
## Temperature   1.0000000 -0.6514003 -0.2845099 -0.3264919
## RH           -0.6514003  1.0000000  0.2440484  0.2223561
## Ws           -0.2845099  0.2440484  1.0000000  0.1715062
## Rain         -0.3264919  0.2223561  0.1715062  1.0000000
##
## $Ycor
##           FFMC       DMC        DC       ISI       BUI       FWI
## FFMC 1.0000000 0.6036076 0.5073967 0.7400068 0.5920110 0.6911320
## DMC  0.6036076 1.0000000 0.8759247 0.6804543 0.9822485 0.8758642
## DC   0.5073967 0.8759247 1.0000000 0.5086432 0.9419885 0.7395206
## ISI  0.7400068 0.6804543 0.5086432 1.0000000 0.6440926 0.9228949
## BUI  0.5920110 0.9822485 0.9419885 0.6440926 1.0000000 0.8579731
## FWI  0.6911320 0.8758642 0.7395206 0.9228949 0.8579731 1.0000000
##
## $XYcor
##             Temperature         RH            Ws       Rain       FFMC
## Temperature   1.0000000 -0.6514003 -0.2845098897 -0.3264919  0.6765681
## RH           -0.6514003  1.0000000  0.2440483822  0.2223561 -0.6448735
## Ws           -0.2845099  0.2440484  1.0000000000  0.1715062 -0.1665483
```

```
## Rain         -0.3264919  0.2223561  0.1715061807  1.0000000 -0.5439062
## FFMC          0.6765681 -0.6448735 -0.1665482728 -0.5439062  1.0000000
## DMC           0.4856869 -0.4085192 -0.0007209737 -0.2887729  0.6036076
## DC            0.3762835 -0.2269411  0.0791345143 -0.2980231  0.5073967
## ISI           0.6038706 -0.6866670  0.0085316891 -0.3474839  0.7400068
## BUI           0.4597895 -0.3538405  0.0314384118 -0.2998515  0.5920110
## FWI           0.5666699 -0.5809567  0.0323677727 -0.3244216  0.6911320
##                      DMC          DC          ISI         BUI         FWI
## Temperature  0.4856869230  0.37628353  0.603870559  0.45978947  0.56666988
## RH          -0.4085191880 -0.22694112 -0.686667043 -0.35384055 -0.58095675
## Ws          -0.0007209737  0.07913451  0.008531689  0.03143841  0.03236777
## Rain        -0.2887729260 -0.29802308 -0.347483929 -0.29985152 -0.32442156
## FFMC         0.6036076410  0.50739666  0.740006828  0.59201101  0.69113197
## DMC          1.0000000000  0.87592466  0.680454326  0.98224849  0.87586416
## DC           0.8759246607  1.00000000  0.508643247  0.94198846  0.73952056
## ISI          0.6804543264  0.50864325  1.000000000  0.64409260  0.92289493
## BUI          0.9822484891  0.94198846  0.644092598  1.00000000  0.85797310
## FWI          0.8758641588  0.73952056  0.922894934  0.85797310  1.00000000
```

Canonical correlation Analysis

```r
#Canonical correlation
cc1 <- cc(Weather, FWI_compo)

# display the canonical correlations
cc1$cor
```

```
## [1] 0.8399173 0.3984355 0.3112653 0.1344204
```

Canonical coefficients

```r
# raw canonical coefficients
cc1$xcoef
```

```
##                    [,1]        [,2]        [,3]        [,4]
## Temperature -0.10315053  0.12792789 -0.13175043 -0.31483591
## RH           0.03914075  0.06353169 -0.02597180 -0.04134852
## Ws          -0.05990063 -0.11178467 -0.34726856  0.05402229
## Rain         0.18212156 -0.32917360  0.06639048 -0.36760036
```

```r
cc1$ycoef
```

```
##              [,1]        [,2]        [,3]        [,4]
## FFMC -0.046469484  0.07486498  0.05677798  0.02356368
## DMC  -0.060439405 -0.04485567  0.22563936 -0.12473490
## DC   -0.006339360  0.01314634 -0.01848025  0.02735178
## ISI  -0.115751724 -0.29407436 -0.25209715  0.12069574
## BUI   0.078758418  0.02663832 -0.18309962 -0.07538721
## FWI   0.002871459 -0.01597301  0.06038108  0.03522416
```

The raw canonical coefficients are interpreted in a manner analogous to interpreting regression coefficients i.e., for the variable FFMC, a one unit increase in FFMC leads to a 0.0464

decrease in the first canonical variate of set 2 when all of the other variables are held constant. FWI leads to a 0.002871 increase in the dimension 1 for the FWI_Compo set with the other predictors held constant.

Next, we will use compute to compute the loadings of the variables on the canonical dimensions (variates). These loadings are correlations between variables and the canonical variates.

```
# compute canonical loadings
cc2 <- comput(Weather,FWI_compo,cc1)

# display canonical loadings
cc2[3:6]
```

```
## $corr.X.xscores
##                   [,1]       [,2]         [,3]         [,4]
## Temperature -0.8234979  0.1551730  0.007214859 -0.545637621
## RH           0.8641844  0.4164061 -0.282441272  0.004258191
## Ws           0.1422818 -0.3295025 -0.911488404  0.200930036
## Rain         0.5871815 -0.6553635  0.035980706 -0.473731929
##
## $corr.Y.xscores
##            [,1]        [,2]        [,3]         [,4]
## FFMC -0.7978544  0.11749991  0.01521896 -0.002188395
## DMC  -0.5240900  0.03121606 -0.11253096 -0.091760375
## DC   -0.3945849  0.13249945 -0.20935845 -0.059185475
## ISI  -0.7527288 -0.14015675 -0.07874522 -0.011576566
## BUI  -0.4921211  0.06790534 -0.15408038 -0.082661709
## FWI  -0.6730546 -0.08053571 -0.12187813 -0.047260416
##
## $corr.X.yscores
##                   [,1]        [,2]         [,3]          [,4]
## Temperature -0.6916702  0.06182643  0.002245735 -0.0733448208
## RH           0.7258435  0.16591097 -0.087914171  0.0005723876
## Ws           0.1195049 -0.13128550 -0.283714724  0.0270090934
## Rain         0.4931839 -0.26112009  0.011199546 -0.0636792297
##
## $corr.Y.yscores
##            [,1]        [,2]        [,3]        [,4]
## FFMC -0.9499202  0.29490320  0.04889385 -0.01628023
## DMC  -0.6239781  0.07834659 -0.36152746 -0.68263733
## DC   -0.4697902  0.33254931 -0.67260449 -0.44030133
## ISI  -0.8961940 -0.35176771 -0.25298426 -0.08612210
## BUI  -0.5859162  0.17042994 -0.49501300 -0.61494919
## FWI  -0.8013344 -0.20212986 -0.39155704 -0.35158667
```

The above correlations are between observed variables and canonical variables which are known as the canonical loadings. These canonical variates are actually a type of latent variable.

In general, the number of canonical dimensions is equal to the number of variables in the smaller set; however, the number of significant dimensions may be even smaller. Canonical dimensions, also known as canonical variates, are latent variables that are analogous to factors obtained in factor analysis. For this particular model there are four canonical dimensions of which only the first two are statistically significant. For statistical test we use R package "CCP".

TESTS

```
library(CCP)
# tests of canonical dimensions
rho <- cc1$cor
## Define number of observations, number of variables in first set, and number of variables in the secon
n <- dim(Weather)[1]
p <- length(Weather)
q <- length(FWI_compo)

## Calculate p-values using the F-approximations of different test statistics:
p.asym(rho, n, p, q, tstat = "Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##               stat    approx df1      df2      p.value
## 1 to 4:  0.2197308 18.451704  24 814.0499 0.000000e+00
## 2 to 4:  0.7460161  4.825319  15 646.3724 5.978211e-09
## 3 to 4:  0.8867957  3.637285   8 470.0000 4.019205e-04
## 4 to 4:  0.9819312  1.447571   3 236.0000 2.296535e-01
```

wilk lambda =0.2197308,F=18.452,d.f=24 : p<0.0001.Here we reject the null hypothesis that there is no relationship between the two sets of variables,and can conclude that two sets of variables are dependent.

same as second, third and fourth canonical variate pair is correlated(All the P values are lower than 0.001).Therefore All four canonical variate pairs are significantly correlated and dependent on one another.

```
p.asym(rho, n, p, q, tstat = "Hotelling")
```

```
##  Hotelling-Lawley Trace, using F-approximation:
##                stat    approx df1 df2      p.value
## 1 to 4:  2.70952656 26.135642  24 926 0.000000e+00
## 2 to 4:  0.31438985  4.894002  15 934 2.778319e-09
## 3 to 4:  0.12568137  3.699745   8 942 2.883511e-04
## 4 to 4:  0.01840133  1.456772   3 950 2.248796e-01
```

```
p.asym(rho, n, p, q, tstat = "Pillai")
```

```
##  Pillai-Bartlett Trace, using F-approximation:
##                stat    approx df1 df2      p.value
## 1 to 4:  0.97916684 12.749428  24 944 0.000000e+00
## 2 to 4:  0.27370579  4.661788  15 952 1.039101e-08
## 3 to 4:  0.11495494  3.550690   8 960 4.591673e-04
## 4 to 4:  0.01806884  1.464167   3 968 2.227964e-01
```

```
p.asym(rho, n, p, q, tstat = "Roy")
```

```
##  Roy's Largest Root, using F-approximation:
##              stat   approx df1 df2 p.value
## 1 to 1:  0.705461 94.20871   6 236       0
##
##  F statistic for Roy's Greatest Root is an upper bound.
```

Estimates of Canonical Correlation

```r
# Get the squared canonical correlations
squared_canonical_correlations <- cc1$cor^2

# Print the squared canonical correlations
print(squared_canonical_correlations)
```

```
## [1] 0.70546105 0.15875085 0.09688610 0.01806884
```

70.54% of the variation in U1 is explained by the variation in V1 ,15.87% of the variation in U2 is explained by the variation in V2 and 9.68% of the variation in U3 is explained by the variation in V3 but only 1.80% of the variation in U4 is explained by V4.

this first one is very high canonical correlation and implies that only first one canonical correlation is important.

Standardized canonical coefficients

```r
# standardized psych canonical coefficients diagonal matrix of Weather sd's
s1 <- diag(sqrt(diag(cov(Weather))))
s1 %*% cc1$xcoef
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.3742342  0.4641274 -0.4779958 -1.1422371
## [2,]  0.5803853  0.9420581 -0.3851140 -0.6131224
## [3,] -0.1684038 -0.3142698 -0.9763057  0.1518775
## [4,]  0.3648271 -0.6594028  0.1329939 -0.7363795
```

```r
# standardized acad canonical coefficients diagonal matrix of acad FWI_Compo's
s2 <- diag(sqrt(diag(cov(FWI_compo))))
s2 %*% cc1$ycoef
```

```
##             [,1]       [,2]       [,3]       [,4]
## [1,] -0.66682042  1.0742856  0.8147436  0.3381304
## [2,] -0.74902795 -0.5558982  2.7963575 -1.5458446
## [3,] -0.30216945  0.6266283 -0.8808725  1.3037392
## [4,] -0.48085973 -1.2216536 -1.0472705  0.5013983
## [5,]  1.12060793  0.3790212 -2.6052185 -1.0726410
## [6,]  0.02136529 -0.1188483  0.4492695  0.2620878
```

consider the variable FFMC , a one standard deviation increase in reading leads to a 0.66 standard deviation decrease in the score on the first canonical variate for set 2 when the other variables in the model are held constant.