# R Notebook sandbox: Playing with Correlation

## Contents



# Correlation

## Review of Basic Differences Testing

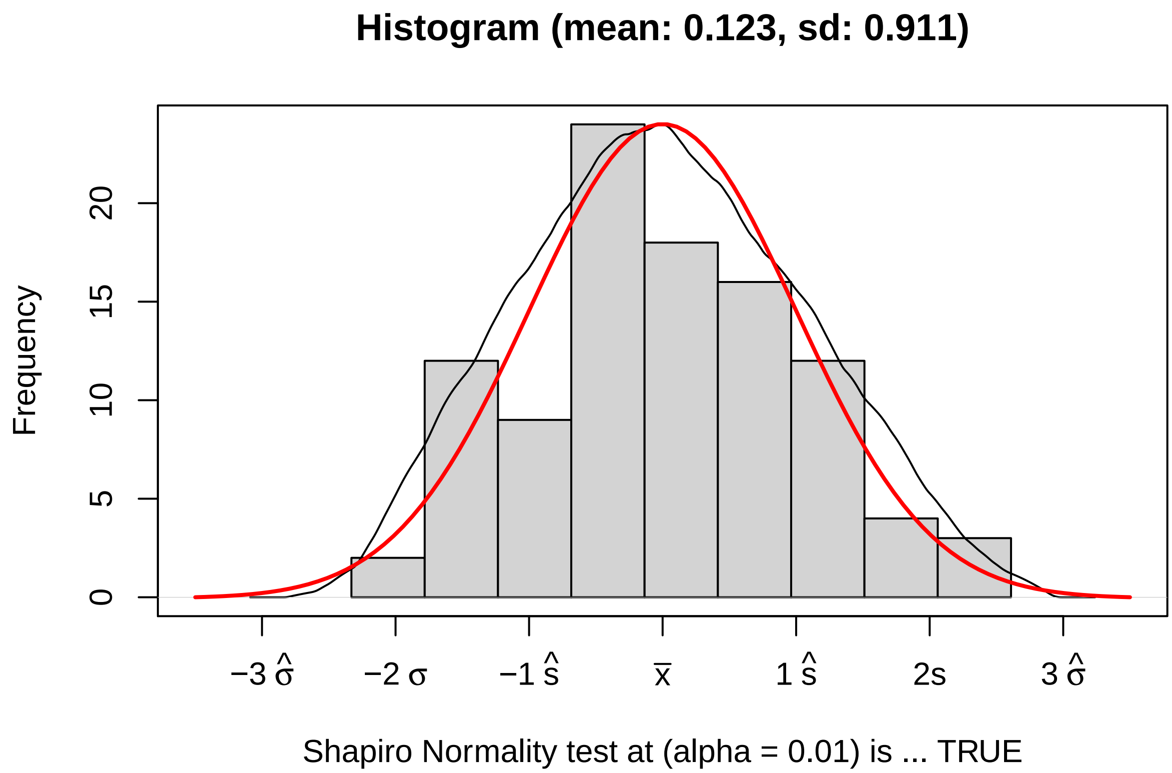
### Variance

Variance measures the spread of a variable. Consider the following three examples:

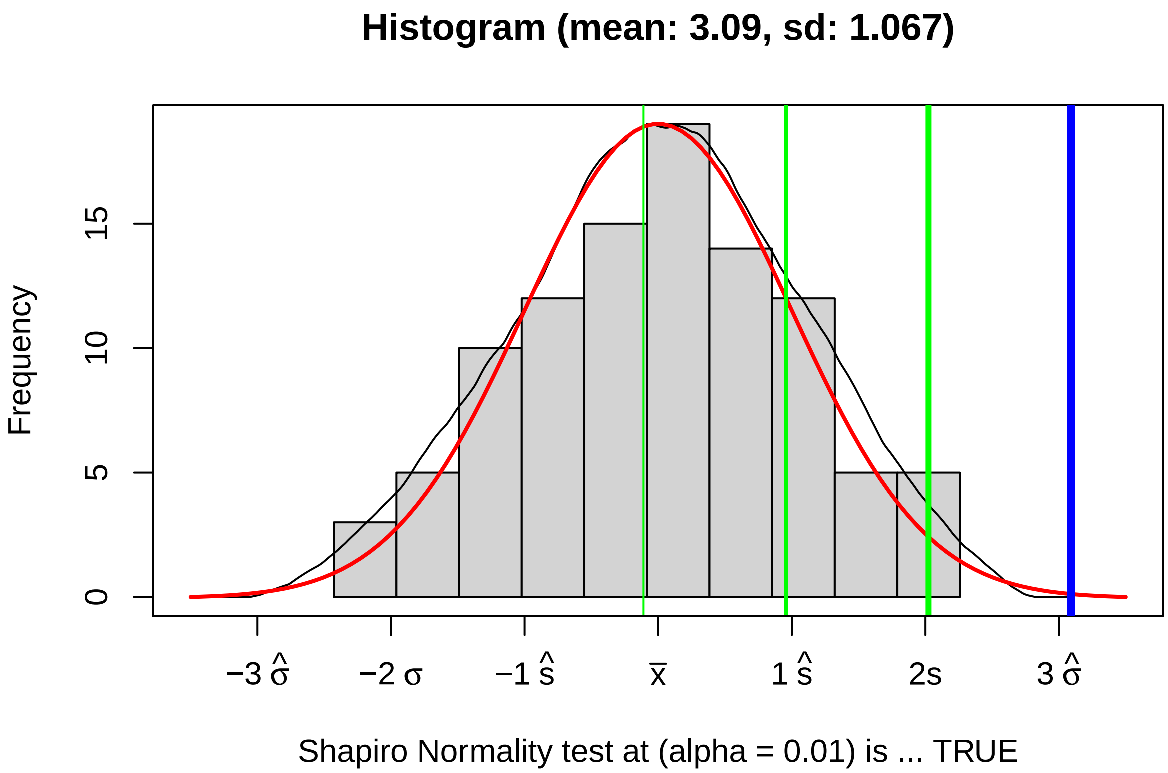
```
library(humanVerseWSU);
# You need R tools for this to work:  https://cran.r-project.org/bin/windows/Rtools/
# You may want to see if you have the latest version...
# library(devtools);
# detach(package:humanVerseWSU);
# install_github("MonteShaffer/humanVerseWSU/humanVerseWSU");
# Choose (3) None to minimize headaches ....


normalDiagnosticPlot = function(x,  normalityTest=TRUE,
                                    showDensity=TRUE,
                                    showNormal=TRUE,
                                    showSDs=FALSE,
                                    showAxis=TRUE
                                )
  {
  xx = na.omit(x);
  x.stats = doStatsSummary(x);
  # x.table = table(x);

  # library(KernSmooth); # install.packages("KernSmooth", dependencies=TRUE);
  # bin.count = dpih(xx);
```

```r
  # mybreaks = 100 * bin.count;

  mxlim = c(x.stats$mean - 3.5 * x.stats$sd ,
            x.stats$mean + 3.5 * x.stats$sd );
  h = hist(xx, breaks="Sturges", plot=F);
  mylim = c(0, max(h$counts));

  myMain = paste0(  "Histogram (mean: ",
                  round(x.stats$mean,digits=3),
                  ", sd: ",
                  round(x.stats$sd,digits=3),
                  ")"
                  );



mxlab = "";
  if(normalityTest)
    {
    isNormal = NULL;
    if(x.stats$shapiro.is.normal$`0.10`) { isNormal = 0.10; }
    if(x.stats$shapiro.is.normal$`0.05`) { isNormal = 0.05; }
    if(x.stats$shapiro.is.normal$`0.01`) { isNormal = 0.01; }

    isNormalResult = FALSE;
    if(!is.null(isNormal)) { isNormalResult = TRUE;}
    if(is.null(isNormal)) { isNormal = 0.05;}

    mxlab = paste0("Shapiro Normality test at (alpha = ",
                isNormal, ") is ... ",isNormalResult);
    }


### Histogram
  hist(xx, breaks="Sturges",  xlim=mxlim, ylim=mylim,
      xlab=mxlab, xaxt='n', main=myMain);

  if(showDensity)
    {
    par(new=T); # overlay
  ### Density Plot (remember first reading?)
    plot( density(xx, kernel="epanechnikov") ,
            xlim=mxlim,
            main="",
            xlab="",
            ylab="",
            xaxt='n',
            yaxt='n'
        );
    }


  if(showNormal)
```

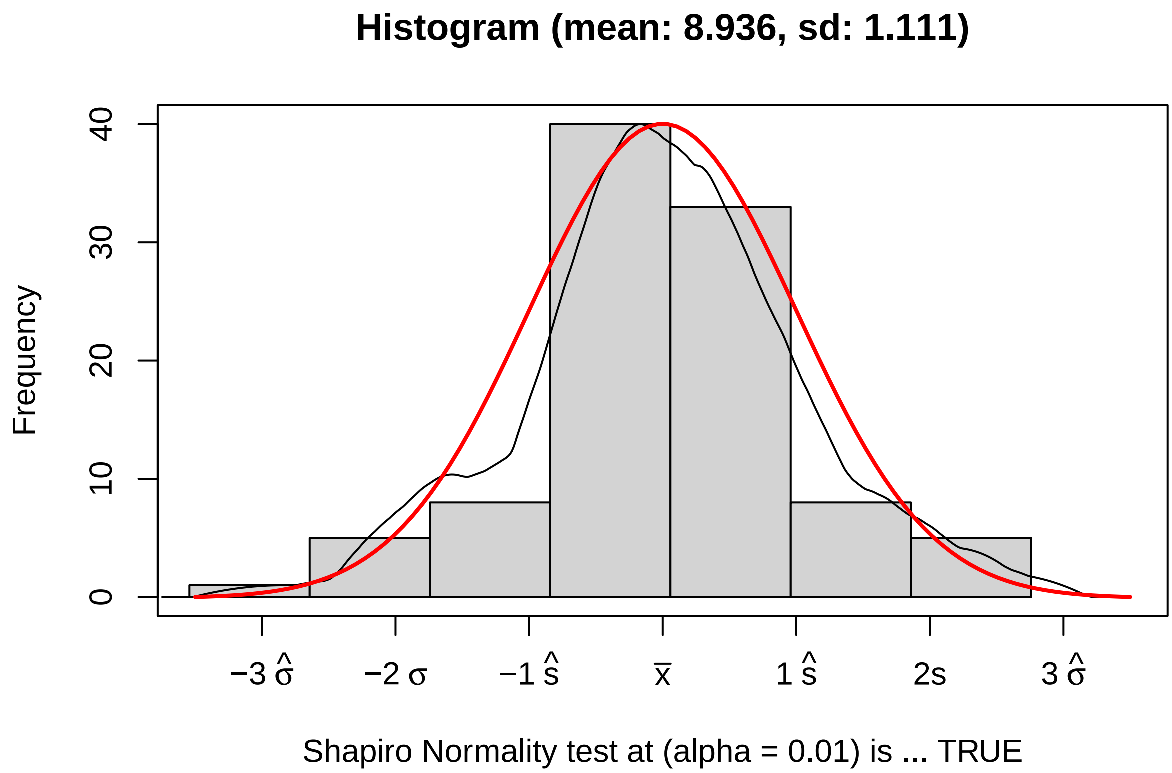
```
    {
    par(new=T); # overlay
  ### Normal Curve
    xt = seq(-3.5,3.5, length=100);
            yt = dnorm(xt);

    plot( xt, yt,
          type="l",
          lwd=2,
          col = "red",
          axes=F,
          xlab="",
          ylab=""
        );
    }


  if(showSDs)
    {
  ### vertical lines at sd's of data ...
    abline(v=x.stats$mean,lwd=4,col="blue");
      abline(v=x.stats$mean - 1 * x.stats$sd , col="green",lwd=3);
      abline(v=x.stats$mean + 1 * x.stats$sd , col="green",lwd=3);
      abline(v=x.stats$mean - 2 * x.stats$sd , col="green",lwd=2);
      abline(v=x.stats$mean + 2 * x.stats$sd , col="green",lwd=2);
      abline(v=x.stats$mean - 3 * x.stats$sd , col="green",lwd=1);
      abline(v=x.stats$mean + 3 * x.stats$sd , col="green",lwd=1);
    }


  if(showAxis)
    {
  ### axis labels showing the ability to use expression
    axis(1, at = -3:3, labels = c( expression("-3"~hat(sigma) ), expression("-2"~sigma ), expression("-
        #axis(1, at = -3:3, labels = c("-3s", "-2s", "-1s", "hat(mu)", "1s", "2s", "3s"))
    }


  }

x.0.1 = rnorm(100,0,1);   normalDiagnosticPlot(x.0.1);
```




```r
x.3.1 = rnorm(100,3,1);    normalDiagnosticPlot(x.3.1, showSDs=TRUE);
```

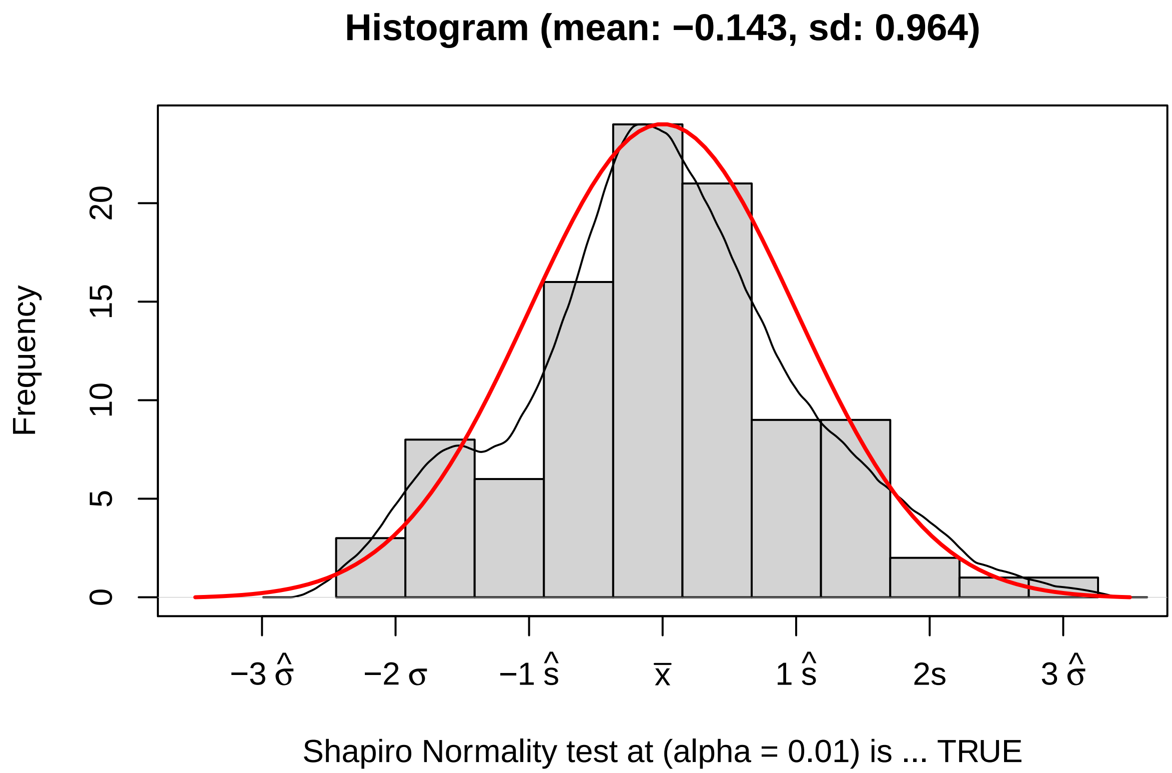



```
x.9.1 = rnorm(100,9,1);   normalDiagnosticPlot(x.9.1);
```

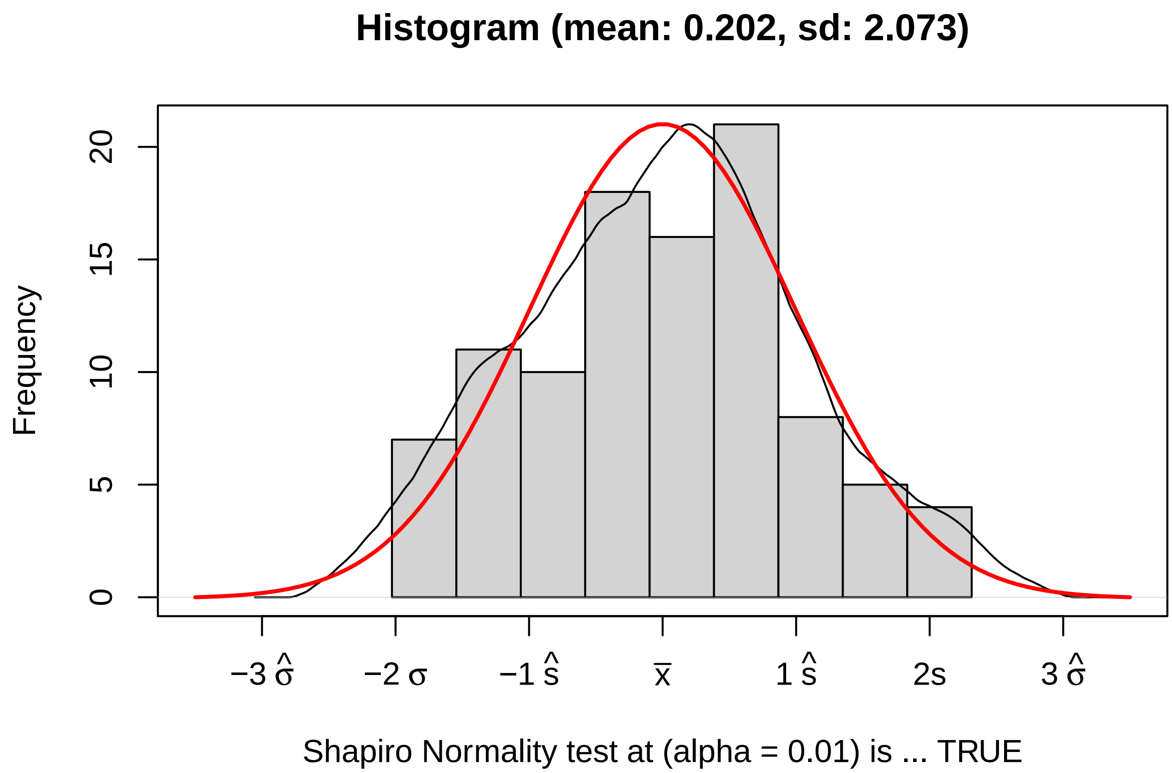



Notice that they all have about the same "spread" yet have different mean values.

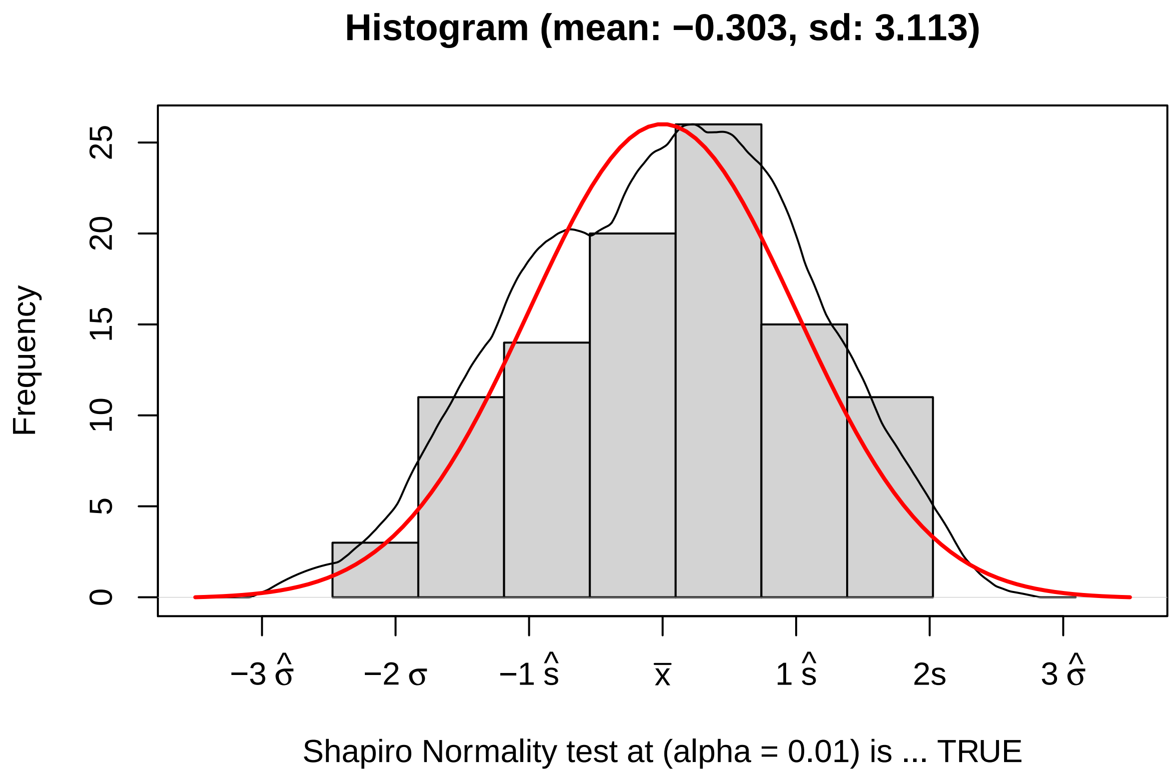
```
x.0.1b = rnorm(100,0,1);    normalDiagnosticPlot(x.0.1b);
```




```
x.0.2 = rnorm(100,0,2);   normalDiagnosticPlot(x.0.2);
```




```
x.0.3 = rnorm(100,0,3);   normalDiagnosticPlot(x.0.3);
```




Notice that they all have different amounts of "spread" yet have about the same mean values.

### Student's t-test: Compare means of two independent samples

TODO: Review the one-sided and two-sided t-test https://en.wikipedia.org/wiki/Student%27s_t-test. Demonstrate your knowledge of those two tests by applying it to one or more of the x-data prepared above (e.g., 'x.0.1').

```
#t.test(x.0.1, x.0.1b, paired=FALSE, var.equal=FALSE);
```

### Welch's t-test: Compare means of two independent samples

We cannot always assume the two samples we want to compare have the same variance. With Student's t-test the pooled variance is used. For Welch's t-test, a different form is used https://en.wikipedia.org/wiki/Welch%27s_t-test.

```
t.test(x.0.1, x.0.1b, paired=FALSE, var.equal=FALSE);
```

```
##
##  Welch Two Sample t-test
##
## data:  x.0.1 and x.0.1b
## t = 2.0071, df = 197.37, p-value = 0.04611
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.004642271 0.527730796
## sample estimates:
##  mean of x  mean of y
```

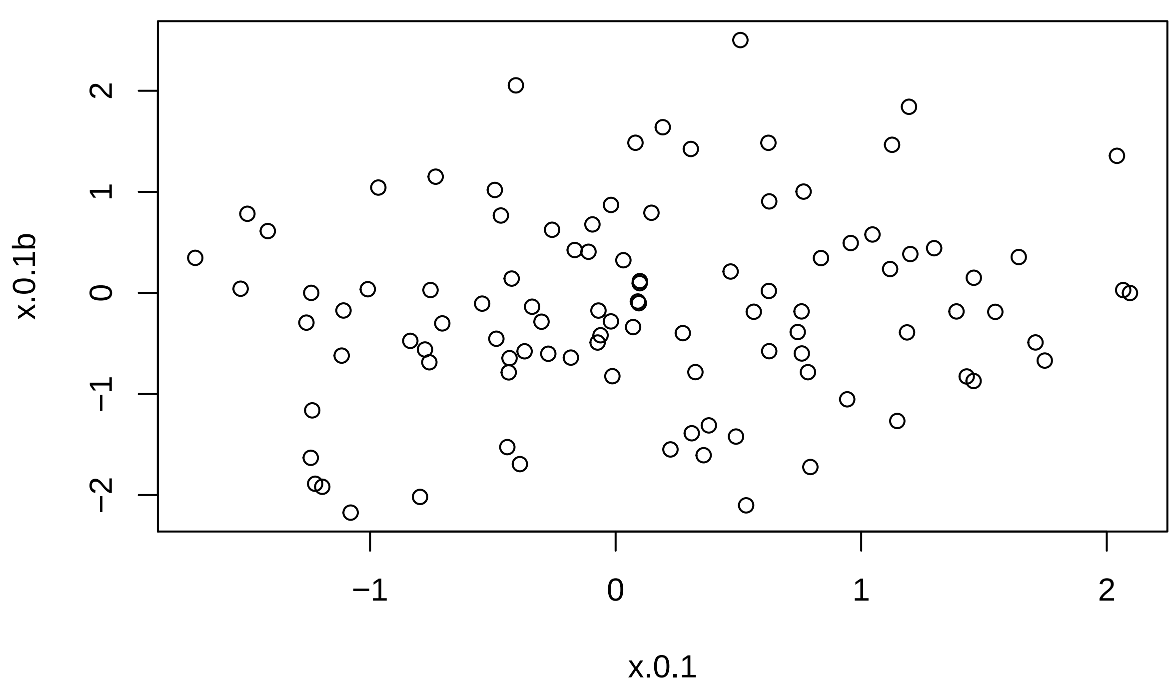
```
##  0.1231149 -0.1430716
```

## Interdependence of data

Above we review when two "independent" samples may have different means or variances. But what happens when the two samples have some degree of "codependency" or "interdependence"?

### Rather Independent

```r
plot(x.0.1, x.0.1b);
```




```r
    cor(x.0.1, x.0.1b, method="pearson");  # default
```

```
## [1] 0.1417192
```

```r
    cor(x.0.1, x.0.1b, method="kendall");
```

```
## [1] 0.07151515
```

```r
    cor(x.0.1, x.0.1b, method="spearman");
```

```
## [1] 0.1058146
```

```r
cor.test(x.0.1, x.0.1b, method="pearson");  # default
```

```
##
##  Pearson's product-moment correlation
##
## data:  x.0.1 and x.0.1b
```

```
## t = 1.4173, df = 98, p-value = 0.1596
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05626511  0.32897978
## sample estimates:
##       cor
## 0.1417192
```

```
cor.test(x.0.1, x.0.1b, method="kendall");
```

```
##
##  Kendall's rank correlation tau
##
## data:  x.0.1 and x.0.1b
## z = 1.0543, p-value = 0.2918
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.07151515
```

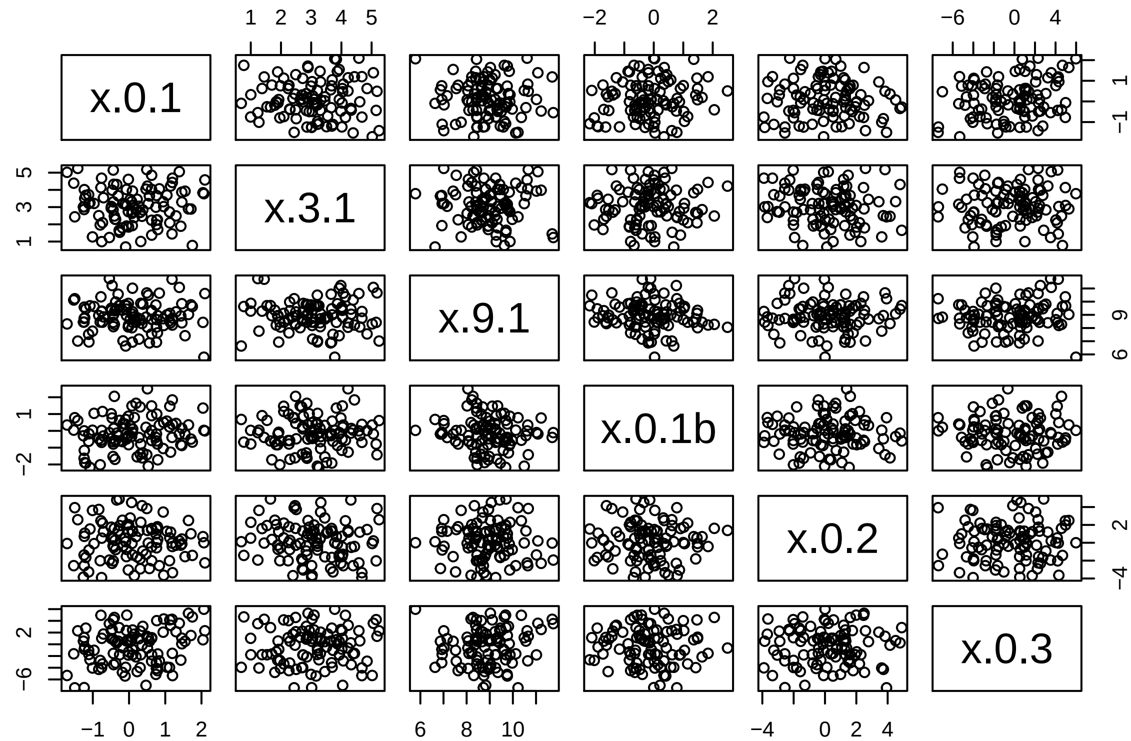
```
cor.test(x.0.1, x.0.1b, method="spearman");
```

```
##
##  Spearman's rank correlation rho
##
## data:  x.0.1 and x.0.1b
## S = 149016, p-value = 0.2942
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1058146
```

```
x.df = data.frame( cbind(x.0.1, x.3.1, x.9.1, x.0.1b, x.0.2, x.0.3) );
    cor(x.df);
```

```
##              x.0.1       x.3.1       x.9.1      x.0.1b       x.0.2       x.0.3
## x.0.1   1.00000000 -0.00962073 -0.02285625  0.14171921 -0.04764616  0.24738443
## x.3.1  -0.00962073  1.00000000 -0.01351532  0.03666927 -0.12923814  0.04086212
## x.9.1  -0.02285625 -0.01351532  1.00000000 -0.13086408 -0.01378132  0.12040782
## x.0.1b  0.14171921  0.03666927 -0.13086408  1.00000000 -0.01192729 -0.03776085
## x.0.2  -0.04764616 -0.12923814 -0.01378132 -0.01192729  1.00000000  0.01999602
## x.0.3   0.24738443  0.04086212  0.12040782 -0.03776085  0.01999602  1.00000000
```

```
plot(x.df);
```




```r
symnum( cor(x.df) ); # remove noise based on cutpoints
```

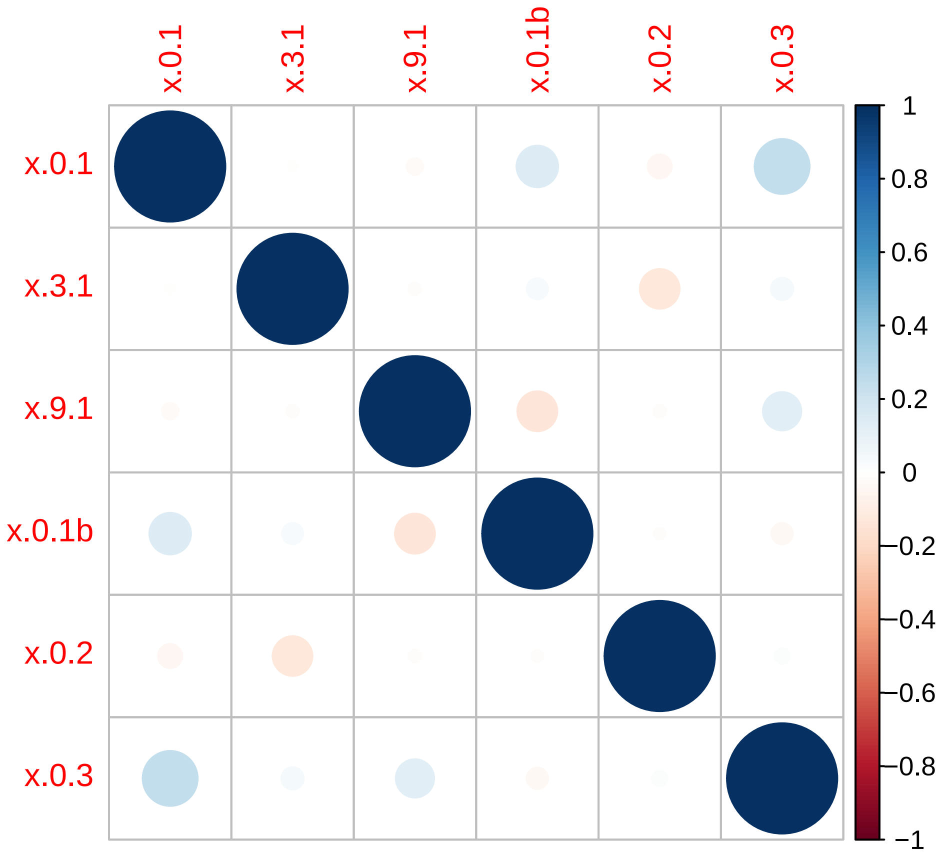
```
##        x.0.1 x.3 x.9 x.0.1b x.0.2 x.0.3
## x.0.1  1
## x.3.1        1
## x.9.1            1
## x.0.1b               1
## x.0.2                       1
## x.0.3                             1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```r
symnum( cor(x.df),
        diag = TRUE,
        corr = TRUE,
        cutpoints=c(0.1,0.4,0.7,0.9),
        symbols = c(" ",".","*","**","***")
      );
```

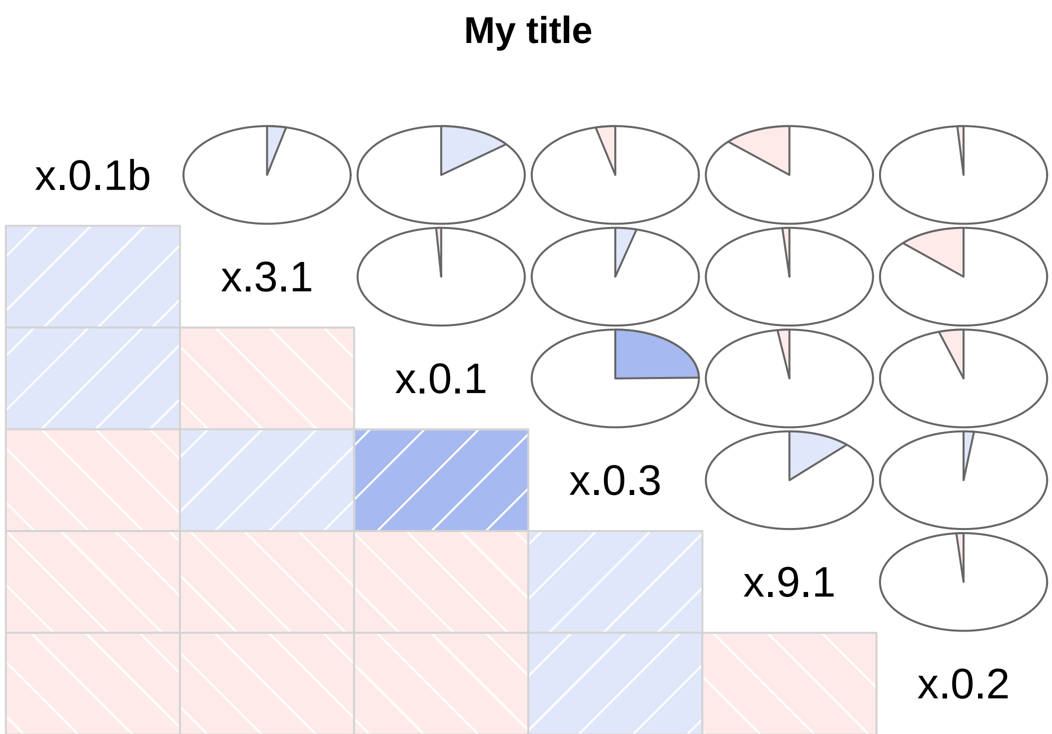
```
##        x.0.1 x.3 x.9 x.0.1b x.0.2 x.0.3
## x.0.1  1
## x.3.1        1
## x.9.1            1
## x.0.1b .         .   1
## x.0.2        .              1
## x.0.3  .         .                1
## attr(,"legend")
```

```
## [1] 0 ' ' 0.1 '.' 0.4 '*' 0.7 '**' 0.9 '***' 1
```

```
    corrplot::corrplot( (cor(x.df)) );

# https://www.statmethods.net/advgraphs/correlograms.html
library(corrgram); # install.packages("corrgram",dependencies=TRUE);
```




```
    corrgram(x.df,
             order=TRUE,
             lower.panel=panel.shade,
             upper.panel=panel.pie,
             text.panel=panel.txt,
             main="My title");
```




```r
library(Hmisc); # install.packages("Hmisc",dependencies=TRUE);
    rcorr(as.matrix(x.df), type="pearson");
```

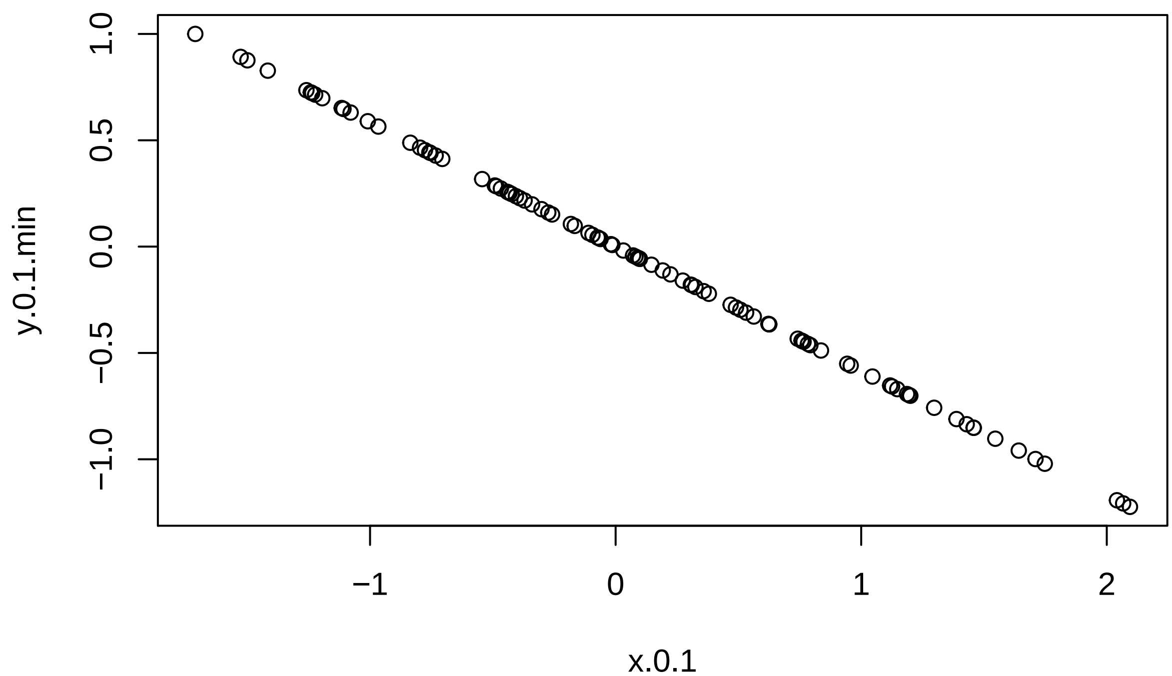
```
##        x.0.1 x.3.1 x.9.1 x.0.1b x.0.2 x.0.3
## x.0.1   1.00 -0.01 -0.02   0.14 -0.05  0.25
## x.3.1  -0.01  1.00 -0.01   0.04 -0.13  0.04
## x.9.1  -0.02 -0.01  1.00  -0.13 -0.01  0.12
## x.0.1b  0.14  0.04 -0.13   1.00 -0.01 -0.04
## x.0.2  -0.05 -0.13 -0.01  -0.01  1.00  0.02
## x.0.3   0.25  0.04  0.12  -0.04  0.02  1.00
##
## n= 100
##
##
## P
##        x.0.1  x.3.1  x.9.1  x.0.1b x.0.2  x.0.3
## x.0.1         0.9243 0.8214 0.1596 0.6378 0.0131
## x.3.1  0.9243        0.8938 0.7172 0.2000 0.6865
## x.9.1  0.8214 0.8938        0.1944 0.8918 0.2328
## x.0.1b 0.1596 0.7172 0.1944        0.9062 0.7092
## x.0.2  0.6378 0.2000 0.8918 0.9062        0.8435
## x.0.3  0.0131 0.6865 0.2328 0.7092 0.8435
```

```r
# corrplot::corrplot(
# sweep
# symnum
```

Notice that since the data was randomly generated "independently", the "correlation" is close to zero.

**Functionally Dependent**

```
# requires functions-standardize.R of humanVerseWSU

y.0.1.min = standardizeToMin(x.0.1);

plot(x.0.1, y.0.1.min);
```




```
cor(x.0.1, y.0.1.min, method="pearson");  # default
```

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

```
cor(x.0.1, y.0.1.min, method="kendall");
```

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

```
cor(x.0.1, y.0.1.min, method="spearman");
```

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

```
cor.test(x.0.1, y.0.1.min, method="pearson");  # default
```

```
##
##  Pearson's product-moment correlation
##
## data:  x.0.1 and y.0.1.min
## t = -Inf, df = 98, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -1 -1
## sample estimates:
## cor
##  -1
```

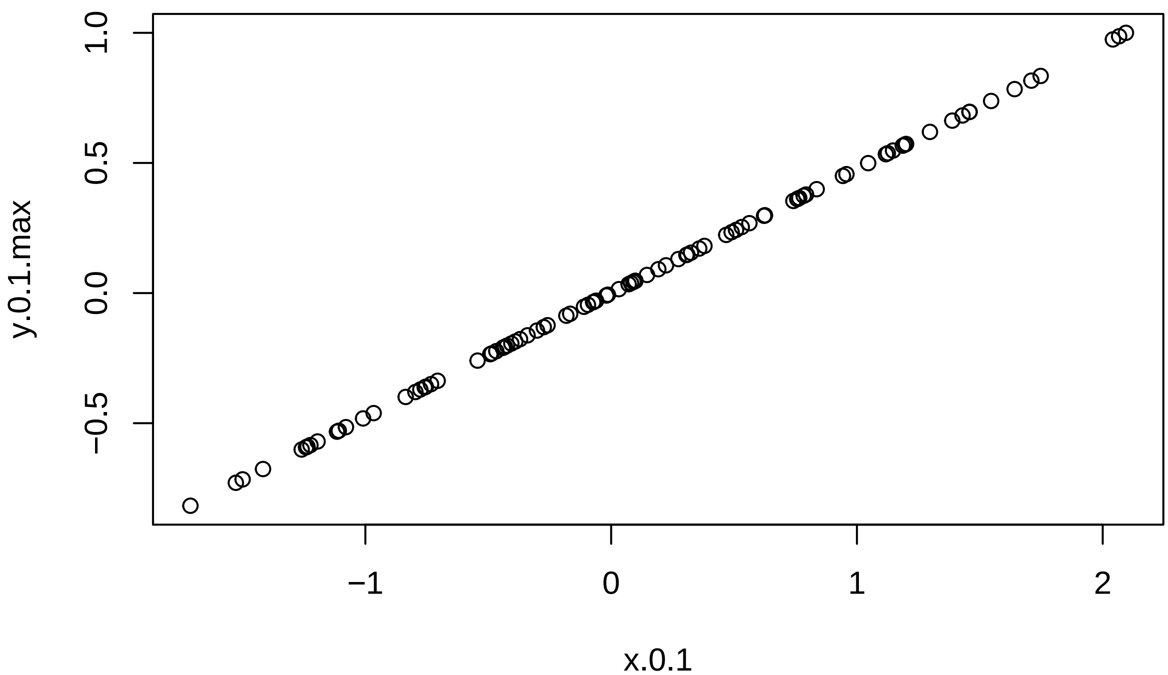
```r
cor.test(x.0.1, y.0.1.min, method="kendall");
```

```
##
##  Kendall's rank correlation tau
##
## data:  x.0.1 and y.0.1.min
## z = -14.742, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
## tau
##  -1
```

```r
cor.test(x.0.1, y.0.1.min, method="spearman");
```

```
##
##  Spearman's rank correlation rho
##
## data:  x.0.1 and y.0.1.min
## S = 333300, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##  -1
```

```r
y.0.1.max = standardizeToMax(x.0.1);

plot(x.0.1, y.0.1.max);
```




```r
cor(x.0.1, y.0.1.max, method="pearson");  # default
```

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

```r
cor(x.0.1, y.0.1.max, method="kendall");
```

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

```r
cor(x.0.1, y.0.1.max, method="spearman");
```

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

```r
cor.test(x.0.1, y.0.1.max, method="pearson");  # default
```

```
##
##  Pearson's product-moment correlation
##
## data:  x.0.1 and y.0.1.max
## t = Inf, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  1 1
## sample estimates:
## cor
##   1
```

```r
cor.test(x.0.1, y.0.1.max, method="kendall");
```

```
##
```

```
##  Kendall's rank correlation tau
##
## data:  x.0.1 and y.0.1.max
## z = 14.742, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
## tau
##   1
```

```r
cor.test(x.0.1, y.0.1.max, method="spearman");
```

```
##
##  Spearman's rank correlation rho
##
## data:  x.0.1 and y.0.1.max
## S = 0, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##   1
```

QUESTION(s) to PONDER: Why does one have a negative slope and the other have a positive slope? Does one (1) mean "perfect correlation"? How does that relate to being a linear combination of a vector basis?

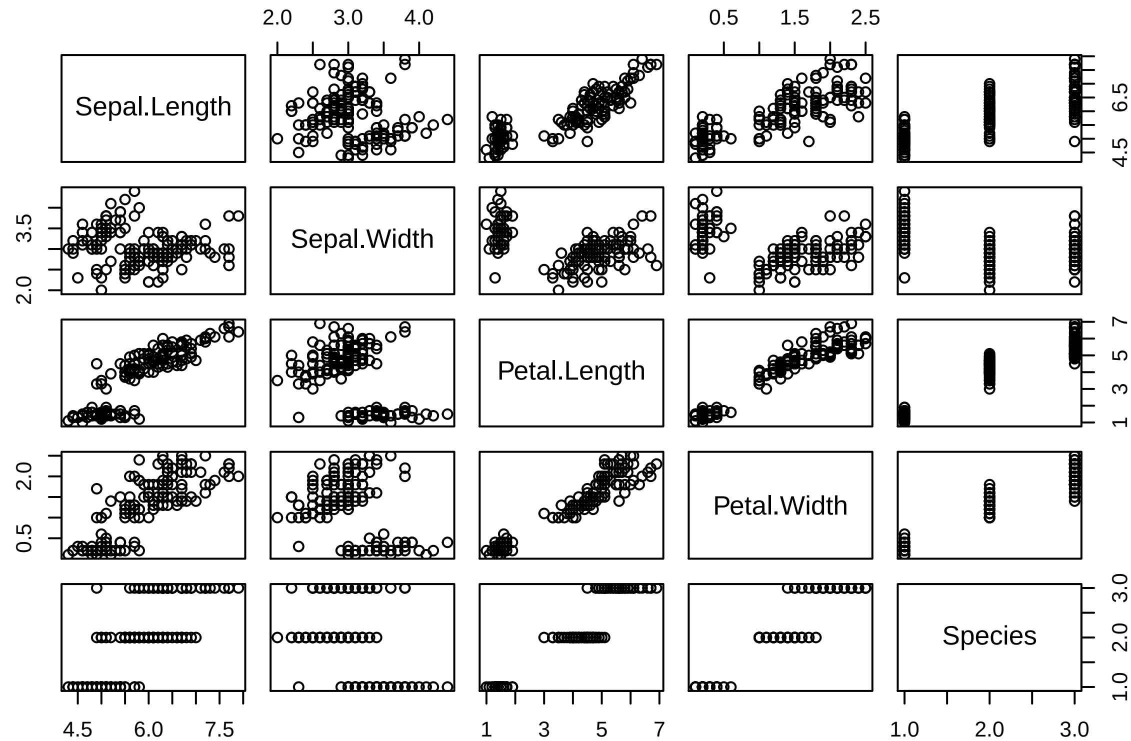
### Somewhere in Between

```r
iris.df = iris;
iris.df$Species = as.numeric(iris.df$Species); # if not, non-numeric, throws an error ...

round( cor(iris.df), digits=2);
```

### Iris

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## Sepal.Length         1.00       -0.12         0.87        0.82    0.78
## Sepal.Width         -0.12        1.00        -0.43       -0.37   -0.43
## Petal.Length         0.87       -0.43         1.00        0.96    0.95
## Petal.Width          0.82       -0.37         0.96        1.00    0.96
## Species              0.78       -0.43         0.95        0.96    1.00
```

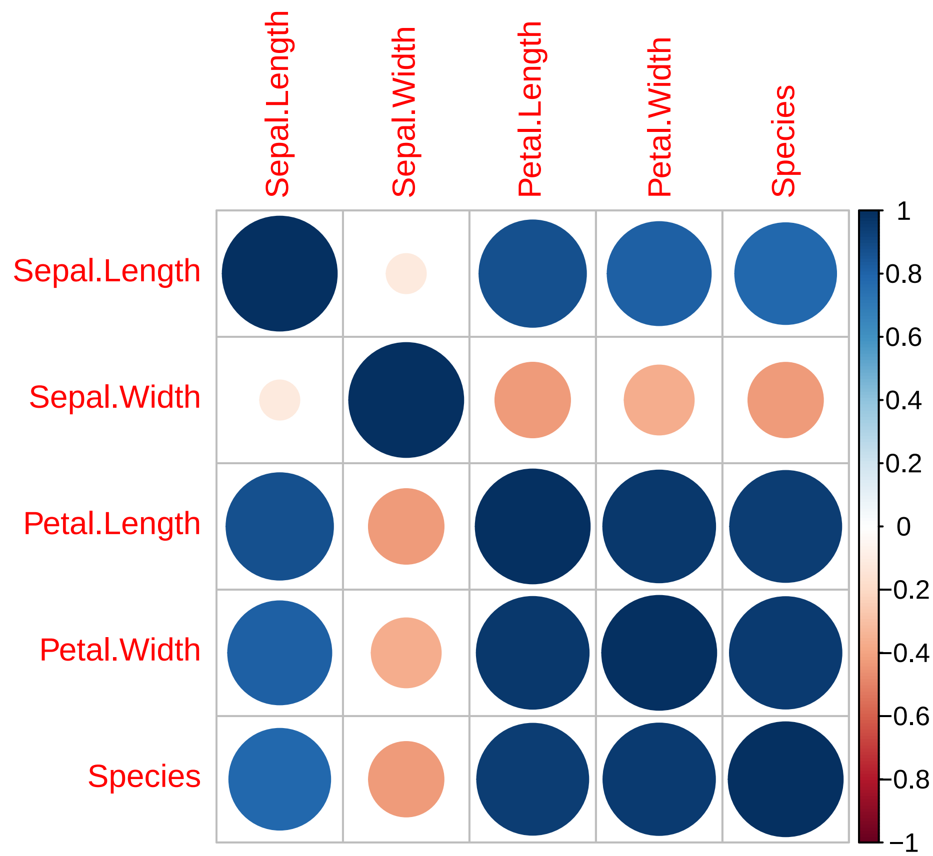
```r
plot(iris.df);
```



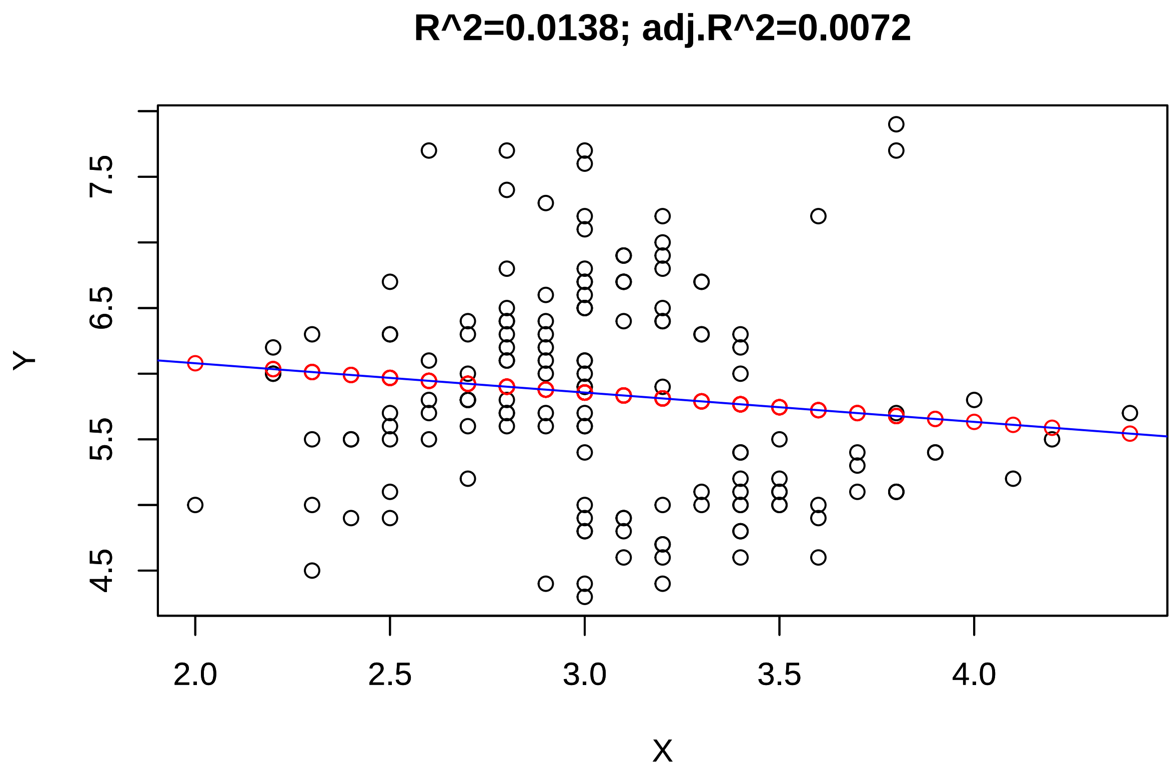

```r
symnum( cor(iris.df),
        diag = TRUE,
        corr = TRUE,
        cutpoints=c(0.4,0.7,0.9),
        symbols = c(" ",".","*","**")
      );
```

```
##              S.L S.W P.L P.W Sp
## Sepal.Length 1
## Sepal.Width      1
## Petal.Length *   .   1
## Petal.Width  *       **  1
## Species      *   .   **  **  1
## attr(,"legend")
## [1] 0 ' ' 0.4 '.' 0.7 '*' 0.9 '**' 1
```

```r
corrplot::corrplot( (cor(iris.df)) );
```




```
# Let's suppose we want to consider that
# Sepal.Length is a function of Sepal.Width

Y = iris.df$Sepal.Length;
X = iris.df$Sepal.Width;
myData = data.frame(cbind(Y,X));


plot(X,Y, xlim=range(X), ylim=range(Y) );
  linear.model = lm(Y~X, myData );
  linear.model.summary = summary(linear.model);
  Y.predicted = predict(linear.model);
par(new=TRUE);
  myMain = paste0("R^2=",
          round(linear.model.summary$r.squared, digits=4),
              "; adj.R^2=",
          round(linear.model.summary$adj.r.squared, digits=4)
                );
plot(X,Y.predicted, main=myMain, ylab="", xlim=range(X), ylim=range(Y), col="red");
  abline(linear.model, col="blue");
```




```r
cor(X, Y, method="pearson");  # default
```

```
## [1] -0.1175698
```

```r
cor(X, Y, method="kendall");
```

```
## [1] -0.07699679
```

```r
cor(X, Y, method="spearman");
```

```
## [1] -0.1667777
```

```r
cor.test(X, Y, method="pearson");  # default
```

```
##
##  Pearson's product-moment correlation
##
## data:  X and Y
## t = -1.4403, df = 148, p-value = 0.1519
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.27269325  0.04351158
## sample estimates:
##        cor
## -0.1175698
```

```r
cor.test(X, Y, method="kendall");
```

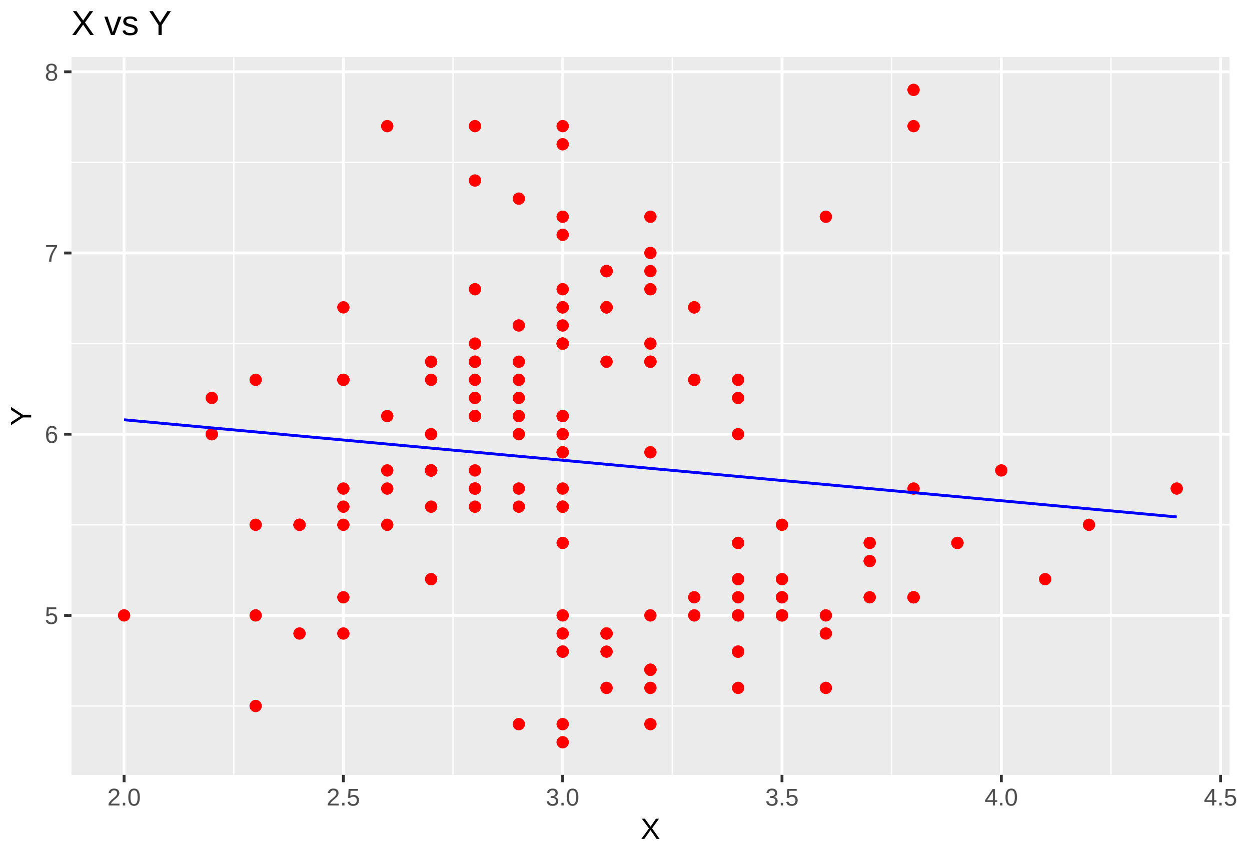
```
##
```

```
##  Kendall's rank correlation tau
##
## data:  X and Y
## z = -1.3318, p-value = 0.1829
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##         tau
## -0.07699679
```

```r
cor.test(X, Y, method="spearman");
```

```
##
##  Spearman's rank correlation rho
##
## data:  X and Y
## S = 656283, p-value = 0.04137
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1667777
```

```r
#####################################################
# different visualization
library(ggplot2);

ggplot() + geom_point(aes(x = X,
                y = Y), colour = 'red') +
geom_line(aes(x = X,
y = Y.predicted), colour = 'blue') +

ggtitle('X vs Y') +
xlab('X') +
ylab('Y')
```




```
library(devtools);
path.github = "https://raw.githubusercontent.com/MonteShaffer/humanVerseWSU/master/";
source_url( paste0(path.github, "humanVerseWSU/R/functions-dataframe.R") );


library(devtools);
source_url("http://md5.mshaffer.com/WSU_STATS419/denzel");
  str(denzel);
```

**Will & Denzel**

```
## List of 4
##  $ nmid      : chr "nm0000243"
##  $ name      : chr "Denzel Washington"
##  $ countfilms:List of 2
##   ..$ totalcount: num 61
##   ..$ pagecount : num 50
##  $ movies.50 :'data.frame':  50 obs. of  11 variables:
##   ..$ rank      : num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ title     : chr [1:50] "American Gangster" "Training Day" "Inside Man" "The Equalizer" ...
##   ..$ ttid      : chr [1:50] "tt0765429" "tt0139654" "tt0454848" "tt0455944" ...
##   ..$ year      : num [1:50] 2007 2001 2006 2014 2004 ...
##   ..$ rated     : chr [1:50] "R" "R" "R" "R" ...
##   ..$ minutes   : num [1:50] 157 122 129 132 146 138 126 118 125 115 ...
##   ..$ genre     : chr [1:50] "Biography, Crime, Drama" "Crime, Drama, Thriller" "Crime, Drama, Myste
```

```
##   ..$ ratings   : num [1:50] 7.8 7.7 7.6 7.2 7.7 7.3 7 6.9 7.7 6.7 ...
##   ..$ metacritic: num [1:50] 76 69 76 57 47 76 59 53 66 52 ...
##   ..$ votes     : num [1:50] 384284 382395 332289 326479 324413 ...
##   ..$ millions  : num [1:50] 130.2 76.6 88.5 101.5 77.9 ...
```

```r
source_url("http://md5.mshaffer.com/WSU_STATS419/will");
  str(will);
```

```
## List of 4
##  $ nmid      : chr "nm0000226"
##  $ name      : chr "Will Smith"
##  $ countfilms:List of 2
##   ..$ totalcount: num 111
##   ..$ pagecount : num 50
##  $ movies.50 :'data.frame':  50 obs. of  11 variables:
##   ..$ rank      : num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ title     : chr [1:50] "I Am Legend" "Suicide Squad" "Independence Day" "Men in Black" ...
##   ..$ ttid      : chr [1:50] "tt0480249" "tt1386697" "tt0116629" "tt0119654" ...
##   ..$ year      : num [1:50] 2007 2016 1996 1997 2004 ...
##   ..$ rated     : chr [1:50] "PG-13" "PG-13" "PG-13" "PG-13" ...
##   ..$ minutes   : num [1:50] 101 123 145 98 115 117 92 88 106 118 ...
##   ..$ genre     : chr [1:50] "Action, Adventure, Drama" "Action, Adventure, Fantasy" "Action, Adventu
##   ..$ ratings   : num [1:50] 7.2 6 7 7.3 7.1 8 6.4 6.2 6.8 6.6 ...
##   ..$ metacritic: num [1:50] 65 40 59 71 59 64 49 49 58 58 ...
##   ..$ votes     : num [1:50] 675193 588111 520657 507618 491489 ...
##   ..$ millions  : num [1:50] 256 325 306 251 145 ...
```

```r
imdb.df = rbind(denzel$movies.50, will$movies.50);

## requires latest humanVerseWSU
## rank & votes & so on ...
imdb.rv = removeAllColumnsBut( imdb.df,
            c("rank","year","minutes",
              "ratings","metacritic","votes","millions") );
dim(imdb.rv);  # 100 movies
```

```
## [1] 100   7
```

```r
imdb.rv = removeNAsFromDataFrame(imdb.rv, c("metacritic","millions") );

dim(imdb.rv);  # 85 movies, is it worth it?
```

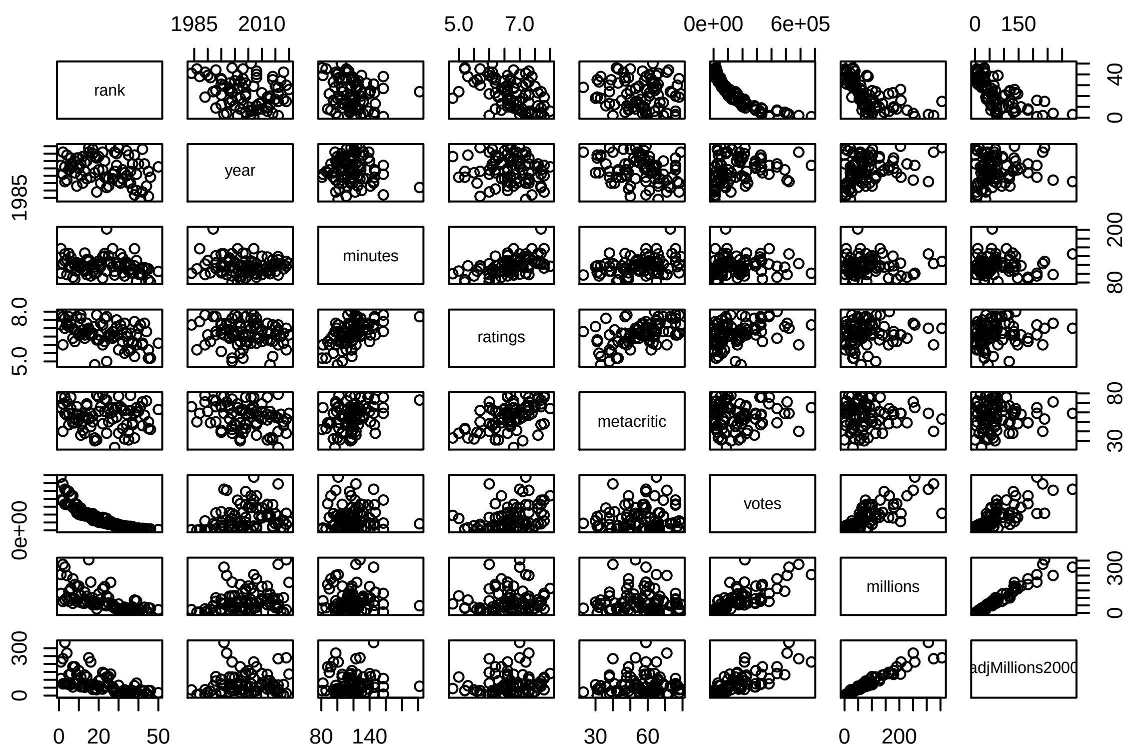
```
## [1] 85  7
```

```r
loadInflationData(); # requires functions-inflation.R in humanVerseWSU ...

imdb.rv = standardizeDollarsInDataFrame(imdb.rv, 2000, "millions", "year", "adjMillions2000");



round( cor(imdb.rv), digits=2);
```

```
##                   rank  year minutes ratings metacritic votes millions
## rank              1.00 -0.26   -0.21   -0.46      -0.09 -0.88    -0.66
## year             -0.26  1.00   -0.08   -0.12      -0.29  0.22     0.26
## minutes          -0.21 -0.08    1.00    0.51       0.31  0.06     0.01
## ratings          -0.46 -0.12    0.51    1.00       0.52  0.33     0.11
```

```
## metacritic      -0.09 -0.29    0.31    0.52       1.00  0.06      0.01
## votes           -0.88  0.22    0.06    0.33       0.06  1.00      0.77
## millions        -0.66  0.26    0.01    0.11       0.01  0.77      1.00
## adjMillions2000 -0.66  0.08    0.02    0.14       0.05  0.76      0.96
##                 adjMillions2000
## rank                      -0.66
## year                       0.08
## minutes                    0.02
## ratings                    0.14
## metacritic                 0.05
## votes                      0.76
## millions                   0.96
## adjMillions2000            1.00
```

```
plot(imdb.rv);
```

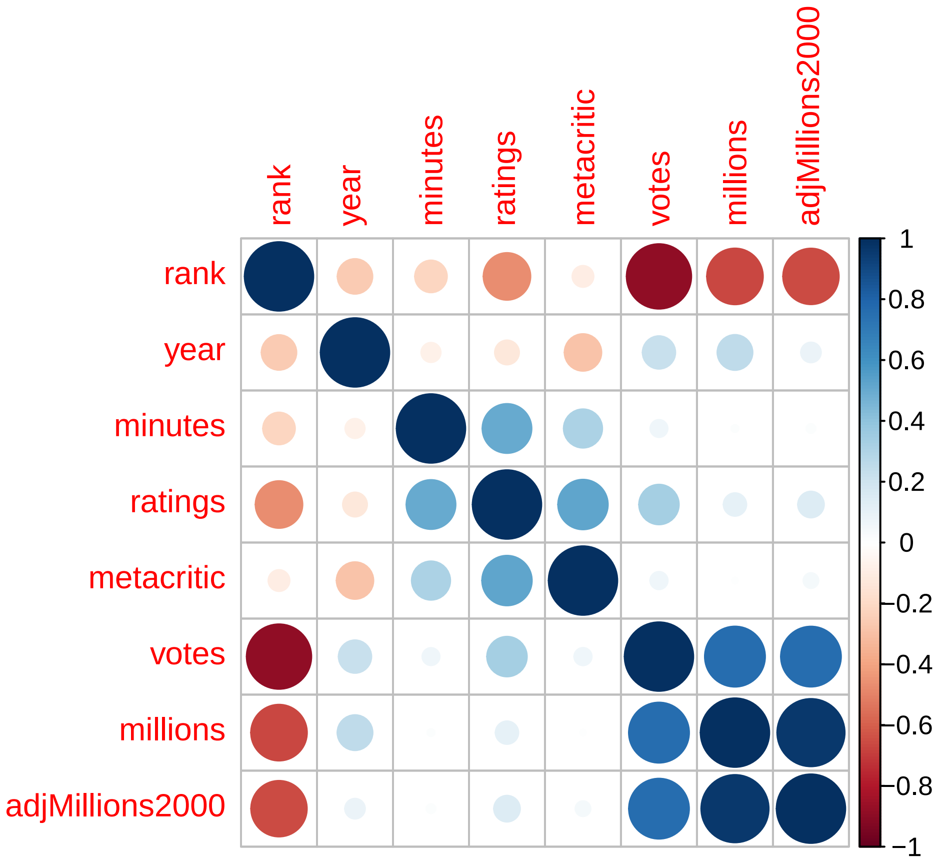



```
    symnum( cor(imdb.rv),
            diag = TRUE,
            corr = TRUE,
            cutpoints=c(0.4,0.7,0.9),
            symbols = c(" ",".","*","**")
          );
```

```
##                 rn y mn rt mt v ml a
## rank            1
## year               1
## minutes              1
```

```
## ratings          .     .  1
## metacritic                .  1
## votes           *             1
## millions        .             * 1
## adjMillions2000 .             * ** 1
## attr(,"legend")
## [1] 0 ' ' 0.4 '.' 0.7 '*' 0.9 '**' 1
```

```
    corrplot::corrplot( (cor(imdb.rv)) );
```




Can you read the results above? Are you able to describe the details? Is a correlation strong/weak? Is it positive or negative? What exactly does that all mean?

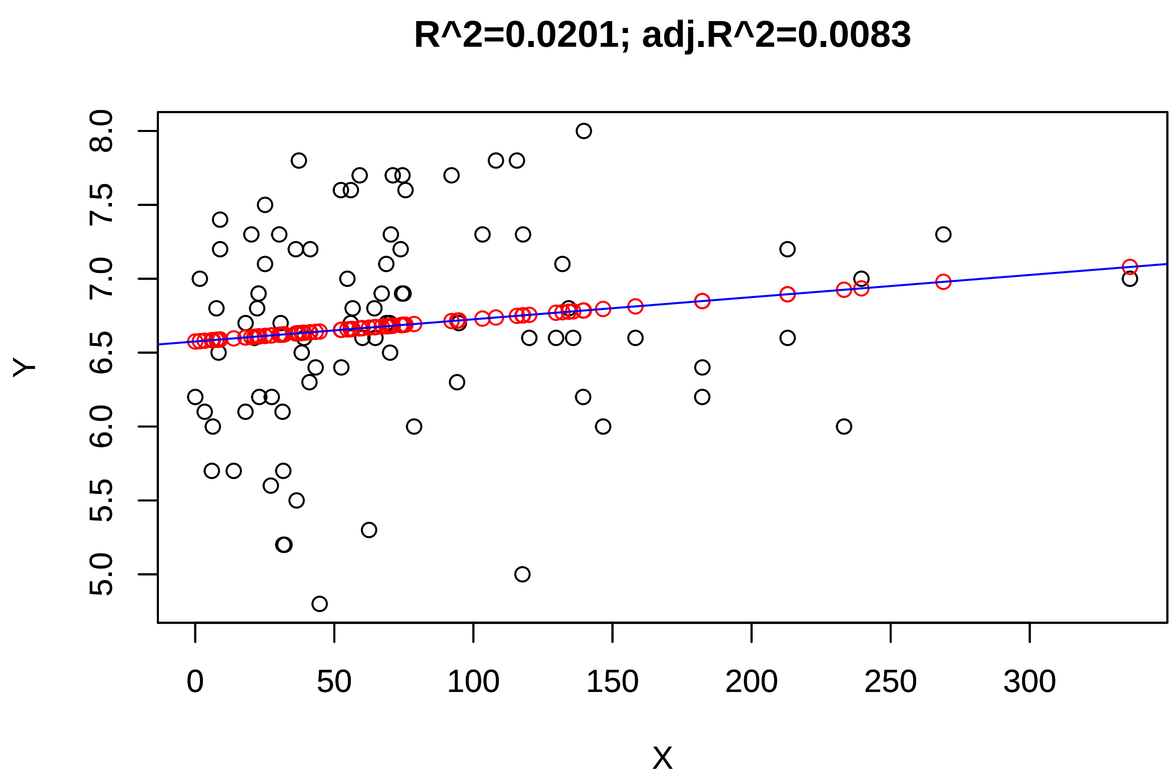
**rank vs. minutes** –WRITE SOMETHING HERE–

**millions vs. year** –WRITE SOMETHING HERE–

**ratings vs. adjMillions2000** –WRITE SOMETHING HERE–

```
# Let's suppose we want to consider that
# Ratings is a function of adjMillions2000






Y = imdb.rv$ratings;
```

```r
X = imdb.rv$adjMillions2000;

myData = data.frame(cbind(Y,X));


plot(X,Y, xlim=range(X), ylim=range(Y) );
  linear.model = lm(Y~X, myData );
  linear.model.summary = summary(linear.model);
  Y.predicted = predict(linear.model);
par(new=TRUE);
  myMain = paste0("R^2=",
          round(linear.model.summary$r.squared, digits=4),
              "; adj.R^2=",
          round(linear.model.summary$adj.r.squared, digits=4)
                );
plot(X,Y.predicted, main=myMain, ylab="", xlim=range(X), ylim=range(Y), col="red");
  abline(linear.model, col="blue");
```




```r
cor(X, Y, method="pearson");  # default
```

```
## [1] 0.1418799
```

```r
cor(X, Y, method="kendall");
```

```
## [1] 0.1306547
```

```r
cor(X, Y, method="spearman");
```

```
## [1] 0.1891664
```

```r
cor.test(X, Y, method="pearson");  # default
```

```
##
##  Pearson's product-moment correlation
##
## data:  X and Y
## t = 1.3058, df = 83, p-value = 0.1952
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07346566  0.34458441
## sample estimates:
##       cor
## 0.1418799
```

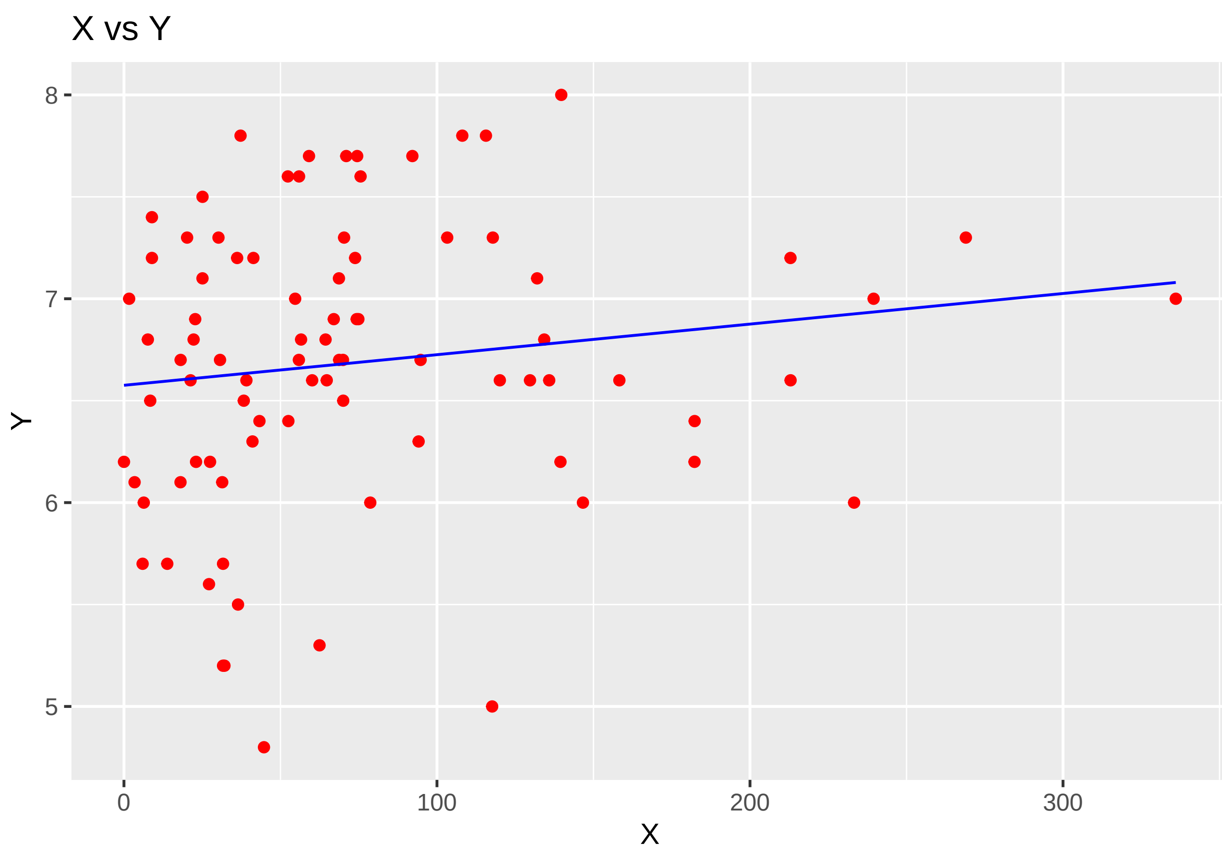
```r
cor.test(X, Y, method="kendall");
```

```
##
##  Kendall's rank correlation tau
##
## data:  X and Y
## z = 1.7378, p-value = 0.08224
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.1306547
```

```r
cor.test(X, Y, method="spearman");
```

```
##
##  Spearman's rank correlation rho
##
## data:  X and Y
## S = 82981, p-value = 0.08294
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1891664
```

```r
#####################################################
# different visualization
library(ggplot2);

ggplot() + geom_point(aes(x = X,
                y = Y), colour = 'red') +
geom_line(aes(x = X,
y = Y.predicted), colour = 'blue') +

ggtitle('X vs Y') +
xlab('X') +
ylab('Y')
```




```
### 3-D plot
library(rgl);
Z = imdb.rv$millions2000;
plot3d(x=X, y=Y, z=Z,
      type="p", col="red",
      xlab="X", ylab="Y", zlab="Z",
      size=5, lwd=15, box=F
      );

# try this one ...
X = rnorm(100,0,1);
Y = rnorm(100,0,1);
Z = X+Y;
plot3d(x=X, y=Y, z=Z,
      type="p", col="red",
      xlab="X", ylab="Y", zlab="Z",
      size=5, lwd=15, box=F
      );
# what do you notice?
# how is this simular to plot(X,z-scores) on a previous notebook?

# functionally, notice that RStudio doesn't embed locally.  Is there a new 3D plotting tool that does?
```

```
library(devtools);
measure.instructor = read.csv("http://md5.mshaffer.com/WSU_STATS419/measure-04343803d489fe8ee2c5f6ab97a
```

```r
getOne = c("hand.length", "hand.width", "hand.elbow", "elbow.armpit", "arm.reach", "foot.length", "floo
n.rows = dim(measure.instructor)[1];


for(one in getOne)
  {
  measure.instructor[one] = NA;
  }

for(i in 1:n.rows)
  {
  measure.row = measure.instructor[i,];
  for(one in getOne)
    {
    nidx = getIndexOfDataFrameColumns(measure.instructor, one);

    myleft = paste0(one,".left");
      lidx = getIndexOfDataFrameColumns(measure.row, myleft);
    myright = paste0(one,".right");
      ridx = getIndexOfDataFrameColumns(measure.row, myleft);

      print(paste0(
                  "left: ",myleft," --> ",lidx,
                  " right: ",myright," --> ",ridx
                  )
            );

      row.m = mean(
            c(as.numeric(unlist(measure.row[lidx])),
            as.numeric(unlist(measure.row[ridx]))),
            na.rm=TRUE);

    measure.instructor[i,nidx] =  row.m;
    }
  }
```

#### Measure

```
## [1] "left: hand.length.left --> 7 right: hand.length.right --> 7"
## [1] "left: hand.width.left --> 9 right: hand.width.right --> 9"
## [1] "left: hand.elbow.left --> 11 right: hand.elbow.right --> 11"
## [1] "left: elbow.armpit.left --> 13 right: elbow.armpit.right --> 13"
## [1] "left: arm.reach.left --> 15 right: arm.reach.right --> 15"
## [1] "left: foot.length.left --> 18 right: foot.length.right --> 18"
## [1] "left: floor.kneepit.left --> 20 right: floor.kneepit.right --> 20"
## [1] "left: floor.hip.left --> 22 right: floor.hip.right --> 22"
## [1] "left: floor.armpit.left --> 25 right: floor.armpit.right --> 25"
## [1] "left: hand.length.left --> 7 right: hand.length.right --> 7"
## [1] "left: hand.width.left --> 9 right: hand.width.right --> 9"
## [1] "left: hand.elbow.left --> 11 right: hand.elbow.right --> 11"
## [1] "left: elbow.armpit.left --> 13 right: elbow.armpit.right --> 13"
## [1] "left: arm.reach.left --> 15 right: arm.reach.right --> 15"
## [1] "left: foot.length.left --> 18 right: foot.length.right --> 18"
## [1] "left: floor.kneepit.left --> 20 right: floor.kneepit.right --> 20"
```

```
## [1] "left: floor.hip.left --> 22 right: floor.hip.right --> 22"
## [1] "left: floor.armpit.left --> 25 right: floor.armpit.right --> 25"
```

```
str(measure.instructor); # lot's of columns ...
```

```
## 'data.frame':    2 obs. of  46 variables:
##  $ data_collector       : chr  "04343803d489fe8ee2c5f6ab97a101e9" "04343803d489fe8ee2c5f6ab97a101e9"
##  $ person_id            : chr  "1c2408654ef5a2fe1fc962088312266c" "8a4108d8610658f282267a704288398d"
##  $ side                 : chr  "right" "right"
##  $ height.NA            : int  178 NA
##  $ head.height.NA       : int  21 17
##  $ head.circumference.NA: int  59 51
##  $ hand.length.left     : num  19.5 NA
##  $ hand.length.right    : int  19 13
##  $ hand.width.left      : int  22 NA
##  $ hand.width.right     : num  21.5 13
##  $ hand.elbow.left      : int  45 NA
##  $ hand.elbow.right     : int  46 NA
##  $ elbow.armpit.left    : int  30 NA
##  $ elbow.armpit.right   : int  31 NA
##  $ arm.reach.left       : num  226 NA
##  $ arm.reach.right      : int  226 NA
##  $ arm.span.NA          : int  176 NA
##  $ foot.length.left     : num  26.5 NA
##  $ foot.length.right    : num  26.5 NA
##  $ floor.kneepit.left   : num  43.5 NA
##  $ floor.kneepit.right  : int  43 NA
##  $ floor.hip.left       : int  94 NA
##  $ floor.hip.right      : num  93.5 NA
##  $ floor.navel.NA       : num  98.5 NA
##  $ floor.armpit.left    : int  136 NA
##  $ floor.armpit.right   : int  134 NA
##  $ units                : chr  "cm" "cm"
##  $ writing              : chr  "right" "both"
##  $ eye                  : chr  "right" NA
##  $ eye_color            : chr  "blue" "green"
##  $ swinging             : chr  "left" "both"
##  $ age                  : num  47 4.5
##  $ gender               : chr  "male" "male"
##  $ quality              : int  10 7
##  $ minutes              : int  23 NA
##  $ ethnicity            : chr  "white" "white"
##  $ notes                : chr  "possible ancestry may include: scottish, wea native american" "possil
##  $ hand.length          : num  19.5 NaN
##  $ hand.width           : num  22 NaN
##  $ hand.elbow           : num  45 NaN
##  $ elbow.armpit         : num  30 NaN
##  $ arm.reach            : num  226 NaN
##  $ foot.length          : num  26.5 NaN
##  $ floor.kneepit        : num  43.5 NaN
##  $ floor.hip            : num  94 NaN
##  $ floor.armpit         : num  136 NaN
```

You should have a dataset of 10 persons. Once I merge all of the datasets of each student participant, I will provide you a larger dataset.

You should read the "Vitrian Man" reading found in the Dropbox for this week. This would be one set of research activities you could perform for your project 1; that is, review the correlations of various components of the body. It would be a good idea to write some code below to begin studying the correlations.

```
## do some initial exploration of your measure-data

## you should be thinking about "one" good research question for PROJECT 01 ... sub-questions may be ap

## At the top-level of your "WSU_STATS419_FALL2020" repository add a folder "PROJECT-01" if you haven't t

## maybe add a section to your 01_notebook in your project folder called # Correlation and perform the
```

```
personality.raw = readRDS( system.file("extdata", "personality-raw.rds", package="humanVerseWSU") );

cleanupPersonalityDataFrame = function(df)
  {
  df = removeColumnsFromDataFrame(personality.raw, "V00");
  dim(df);  # 838


  ywd.cols = c("year","week","day");
  ywd = convertDateStringToFormat( df$date_test,
                                    c("%Y","%W","%j"),
                                    ywd.cols,
                                    "%m/%d/%Y %H:%M"
                                  );

  ndf = replaceDateStringWithDateColumns(df,"date_test",ywd);
  ndf = sortDataFrameByNumericColumns(ndf, ywd.cols, "DESC");
  ndf = removeDuplicatesFromDataFrame(ndf, "md5_email");

  dim(ndf); # 678
  ndf;
  }


personality.clean = cleanupPersonalityDataFrame(personality.raw);

### let's examine the data in total

personality.Vs = removeColumnsFromDataFrame(personality.clean,c("md5_email","year","week","day"));


round( cor(personality.Vs), digits=2);
```

**Personality**

```
##       V01   V02   V03   V04   V05   V06   V07   V08   V09   V10   V11   V12
## V01  1.00  0.35 -0.19  0.07  0.20  0.17  0.19  0.05  0.25  0.35  0.10  0.46
## V02  0.35  1.00 -0.44  0.12  0.18  0.21  0.13  0.01  0.23  0.44  0.10  0.30
## V03 -0.19 -0.44  1.00 -0.03 -0.01  0.11 -0.10 -0.01 -0.09 -0.29  0.03 -0.25
## V04  0.07  0.12 -0.03  1.00  0.22  0.11  0.26  0.08  0.28  0.22  0.07  0.21
## V05  0.20  0.18 -0.01  0.22  1.00  0.39  0.18  0.23  0.50  0.27 -0.20  0.17
## V06  0.17  0.21  0.11  0.11  0.39  1.00  0.04  0.20  0.34  0.23 -0.14  0.05
```

```
## V07  0.19  0.13 -0.10  0.26  0.18  0.04  1.00  0.13  0.14  0.27  0.15  0.28
## V08  0.05  0.01 -0.01  0.08  0.23  0.20  0.13  1.00  0.23  0.11 -0.13  0.12
## V09  0.25  0.23 -0.09  0.28  0.50  0.34  0.14  0.23  1.00  0.34 -0.15  0.23
## V10  0.35  0.44 -0.29  0.22  0.27  0.23  0.27  0.11  0.34  1.00  0.02  0.33
## V11  0.10  0.10  0.03  0.07 -0.20 -0.14  0.15 -0.13 -0.15  0.02  1.00  0.18
## V12  0.46  0.30 -0.25  0.21  0.17  0.05  0.28  0.12  0.23  0.33  0.18  1.00
## V13  0.24  0.32 -0.27  0.16  0.11  0.04  0.31  0.17  0.14  0.32  0.28  0.40
## V14  0.45  0.65 -0.52  0.16  0.25  0.16  0.19  0.10  0.34  0.50  0.03  0.44
## V15  0.08  0.08 -0.03  0.36  0.30  0.21  0.22  0.23  0.33  0.20  0.05  0.15
## V16  0.23  0.32 -0.07  0.25  0.52  0.42  0.16  0.25  0.59  0.39 -0.21  0.16
## V17  0.22  0.26 -0.06  0.23  0.55  0.55  0.10  0.21  0.56  0.32 -0.12  0.16
## V18  0.24  0.31 -0.11  0.26  0.39  0.40  0.18  0.17  0.42  0.38 -0.06  0.23
## V19  0.25  0.29 -0.18  0.25  0.15  0.10  0.27  0.11  0.24  0.36  0.19  0.35
## V20  0.02  0.00  0.15 -0.16 -0.22 -0.16  0.04 -0.12 -0.22 -0.04  0.41  0.02
## V21  0.19  0.28 -0.18  0.14 -0.09 -0.09  0.23  0.00 -0.05  0.19  0.52  0.35
## V22  0.16  0.13 -0.13  0.39  0.27  0.13  0.34  0.18  0.26  0.30  0.08  0.30
## V23  0.24  0.34 -0.11  0.15  0.52  0.51  0.10  0.19  0.43  0.41 -0.13  0.11
## V24 -0.10  0.00  0.19  0.27  0.28  0.21  0.11  0.13  0.23  0.12 -0.10 -0.08
## V25 -0.02  0.09  0.11 -0.01 -0.11  0.02  0.11 -0.09 -0.15 -0.03  0.41  0.00
## V26  0.37  0.25 -0.11  0.27  0.39  0.16  0.26  0.15  0.44  0.34 -0.04  0.32
## V27  0.25  0.29 -0.17  0.13  0.23  0.11  0.17  0.05  0.29  0.40  0.02  0.26
## V28  0.10 -0.09  0.11  0.16  0.30  0.17  0.10  0.23  0.38  0.09 -0.26  0.10
## V29  0.04  0.16 -0.02  0.02 -0.17 -0.10  0.16 -0.08 -0.11  0.00  0.55  0.17
## V30  0.18  0.16 -0.06  0.21  0.44  0.33  0.11  0.22  0.48  0.25 -0.24  0.14
## V31  0.14  0.16 -0.10  0.30  0.11  0.04  0.33  0.10  0.13  0.22  0.22  0.32
## V32  0.14  0.17 -0.06  0.17  0.15  0.05  0.17  0.03  0.18  0.22  0.08  0.17
## V33  0.07  0.10  0.02  0.16  0.30  0.22  0.08  0.19  0.31  0.17 -0.10  0.04
## V34  0.27  0.31 -0.21  0.25  0.18  0.07  0.21  0.11  0.23  0.37  0.13  0.38
## V35 -0.04  0.00  0.15  0.20  0.10  0.00  0.10  0.08  0.07  0.02  0.15  0.11
## V36  0.14  0.23 -0.07  0.30  0.29  0.15  0.17  0.14  0.28  0.27  0.09  0.16
## V37  0.22  0.27 -0.11  0.12  0.00 -0.05  0.24 -0.02  0.00  0.16  0.38  0.35
## V38  0.19  0.26 -0.20  0.29  0.18  0.06  0.29  0.09  0.20  0.32  0.11  0.29
## V39  0.16  0.16 -0.03  0.24  0.29  0.17  0.14  0.13  0.32  0.21 -0.06  0.13
## V40  0.22  0.25 -0.13  0.26  0.22  0.15  0.19  0.03  0.25  0.30  0.10  0.24
## V41  0.19  0.23 -0.14  0.22  0.03 -0.03  0.21  0.00  0.07  0.20  0.25  0.28
## V42  0.25  0.24 -0.20  0.26  0.15  0.06  0.20  0.07  0.20  0.30  0.11  0.33
## V43  0.15  0.21 -0.07  0.24  0.24  0.15  0.21  0.10  0.25  0.27  0.11  0.17
## V44  0.24  0.23 -0.16  0.32  0.17  0.12  0.23  0.15  0.27  0.35  0.10  0.28
## V45  0.11  0.19 -0.12  0.21  0.34  0.26  0.15  0.20  0.35  0.28  0.03  0.07
## V46  0.09  0.12 -0.01  0.12  0.31  0.28  0.07  0.21  0.32  0.16 -0.07  0.06
## V47  0.11  0.15  0.00  0.17  0.42  0.33  0.11  0.19  0.32  0.23 -0.05  0.00
## V48  0.06 -0.04  0.10  0.09  0.18  0.10  0.04  0.15  0.26  0.08 -0.15  0.03
## V49  0.26  0.30 -0.18  0.21  0.19  0.04  0.23  0.16  0.17  0.25  0.22  0.33
## V50  0.24  0.31 -0.21  0.03  0.11 -0.01  0.15  0.05  0.16  0.26  0.12  0.19
## V51  0.42  0.29 -0.21  0.18  0.18  0.08  0.20  0.12  0.24  0.33  0.09  0.43
## V52  0.12  0.15 -0.11  0.36  0.22  0.15  0.23  0.15  0.23  0.25  0.10  0.19
## V53  0.15  0.23 -0.02  0.17  0.38  0.33  0.07  0.16  0.43  0.23 -0.10  0.08
## V54  0.25  0.31 -0.14  0.22  0.35  0.22  0.11  0.14  0.39  0.34 -0.02  0.16
## V55  0.03  0.08  0.05  0.08  0.16  0.18  0.06  0.08  0.15  0.09  0.03  0.01
## V56  0.22  0.18 -0.09  0.17  0.26  0.33  0.07  0.15  0.34  0.23 -0.06  0.12
## V57  0.28  0.29 -0.16  0.20  0.22  0.15  0.19  0.12  0.26  0.34  0.11  0.30
## V58  0.19  0.26 -0.15  0.23  0.30  0.26  0.12  0.17  0.33  0.31 -0.06  0.17
## V59  0.21  0.23 -0.18  0.25  0.11 -0.01  0.30  0.05  0.14  0.28  0.23  0.32
## V60  0.09  0.11  0.07  0.18  0.33  0.36  0.03  0.17  0.34  0.16 -0.10  0.05
```

```
##       V13   V14   V15   V16   V17   V18   V19   V20   V21   V22   V23   V24
## V01  0.24  0.45  0.08  0.23  0.22  0.24  0.25  0.02  0.19  0.16  0.24 -0.10
## V02  0.32  0.65  0.08  0.32  0.26  0.31  0.29  0.00  0.28  0.13  0.34  0.00
## V03 -0.27 -0.52 -0.03 -0.07 -0.06 -0.11 -0.18  0.15 -0.18 -0.13 -0.11  0.19
## V04  0.16  0.16  0.36  0.25  0.23  0.26  0.25 -0.16  0.14  0.39  0.15  0.27
## V05  0.11  0.25  0.30  0.52  0.55  0.39  0.15 -0.22 -0.09  0.27  0.52  0.28
## V06  0.04  0.16  0.21  0.42  0.55  0.40  0.10 -0.16 -0.09  0.13  0.51  0.21
## V07  0.31  0.19  0.22  0.16  0.10  0.18  0.27  0.04  0.23  0.34  0.10  0.11
## V08  0.17  0.10  0.23  0.25  0.21  0.17  0.11 -0.12  0.00  0.18  0.19  0.13
## V09  0.14  0.34  0.33  0.59  0.56  0.42  0.24 -0.22 -0.05  0.26  0.43  0.23
## V10  0.32  0.50  0.20  0.39  0.32  0.38  0.36 -0.04  0.19  0.30  0.41  0.12
## V11  0.28  0.03  0.05 -0.21 -0.12 -0.06  0.19  0.41  0.52  0.08 -0.13 -0.10
## V12  0.40  0.44  0.15  0.16  0.16  0.23  0.35  0.02  0.35  0.30  0.11 -0.08
## V13  1.00  0.36  0.20  0.16  0.11  0.18  0.59  0.05  0.41  0.30  0.11  0.00
## V14  0.36  1.00  0.16  0.36  0.29  0.33  0.34 -0.07  0.27  0.22  0.36 -0.03
## V15  0.20  0.16  1.00  0.31  0.34  0.25  0.26 -0.10  0.13  0.38  0.26  0.25
## V16  0.16  0.36  0.31  1.00  0.58  0.44  0.27 -0.20 -0.12  0.21  0.49  0.28
## V17  0.11  0.29  0.34  0.58  1.00  0.50  0.18 -0.22 -0.06  0.20  0.59  0.19
## V18  0.18  0.33  0.25  0.44  0.50  1.00  0.25 -0.23  0.01  0.23  0.47  0.21
## V19  0.59  0.34  0.26  0.27  0.18  0.25  1.00 -0.03  0.31  0.29  0.19  0.07
## V20  0.05 -0.07 -0.10 -0.20 -0.22 -0.23 -0.03  1.00  0.25 -0.09 -0.20 -0.11
## V21  0.41  0.27  0.13 -0.12 -0.06  0.01  0.31  0.25  1.00  0.15 -0.02 -0.09
## V22  0.30  0.22  0.38  0.21  0.20  0.23  0.29 -0.09  0.15  1.00  0.20  0.21
## V23  0.11  0.36  0.26  0.49  0.59  0.47  0.19 -0.20 -0.02  0.20  1.00  0.25
## V24  0.00 -0.03  0.25  0.28  0.19  0.21  0.07 -0.11 -0.09  0.21  0.25  1.00
## V25  0.11  0.00 -0.02 -0.10 -0.07 -0.03  0.05  0.32  0.26  0.06 -0.05  0.07
## V26  0.22  0.34  0.26  0.43  0.28  0.27  0.27 -0.13  0.08  0.33  0.26  0.15
## V27  0.25  0.37  0.11  0.36  0.29  0.22  0.29  0.01  0.13  0.13  0.25  0.11
## V28  0.02  0.03  0.20  0.36  0.25  0.18  0.10 -0.21 -0.23  0.16  0.13  0.24
## V29  0.31  0.08  0.02 -0.14 -0.11 -0.07  0.20  0.40  0.45  0.07 -0.09 -0.04
## V30  0.07  0.25  0.31  0.48  0.40  0.31  0.13 -0.19 -0.11  0.21  0.36  0.25
## V31  0.37  0.20  0.21  0.16  0.09  0.13  0.33  0.05  0.29  0.29  0.07  0.13
## V32  0.23  0.20  0.18  0.22  0.15  0.13  0.27  0.03  0.12  0.19  0.15  0.10
## V33  0.00  0.18  0.25  0.34  0.32  0.22  0.14 -0.05 -0.05  0.13  0.24  0.19
## V34  0.39  0.38  0.21  0.21  0.17  0.23  0.40 -0.01  0.28  0.28  0.20  0.03
## V35  0.12 -0.02  0.18  0.08  0.01  0.05  0.16  0.06  0.12  0.16  0.00  0.26
## V36  0.23  0.25  0.29  0.31  0.30  0.32  0.27 -0.03  0.11  0.29  0.29  0.23
## V37  0.40  0.24  0.04 -0.01 -0.03  0.03  0.29  0.15  0.44  0.08 -0.02 -0.05
## V38  0.31  0.33  0.24  0.26  0.17  0.22  0.35 -0.03  0.25  0.31  0.21  0.11
## V39  0.11  0.21  0.22  0.35  0.29  0.27  0.20 -0.09  0.00  0.26  0.26  0.17
## V40  0.30  0.31  0.27  0.27  0.20  0.22  0.34 -0.02  0.20  0.25  0.22  0.13
## V41  0.34  0.23  0.11  0.08  0.04  0.07  0.28  0.11  0.41  0.13  0.03  0.04
## V42  0.27  0.32  0.24  0.19  0.20  0.25  0.30 -0.02  0.26  0.30  0.18  0.09
## V43  0.26  0.25  0.24  0.30  0.23  0.23  0.28  0.00  0.14  0.25  0.22  0.11
## V44  0.26  0.30  0.23  0.29  0.22  0.20  0.36 -0.02  0.23  0.32  0.20  0.13
## V45  0.15  0.23  0.33  0.41  0.37  0.23  0.25 -0.06  0.04  0.28  0.35  0.17
## V46  0.05  0.19  0.26  0.34  0.34  0.23  0.12 -0.09 -0.05  0.21  0.30  0.23
## V47  0.09  0.19  0.25  0.37  0.42  0.32  0.16 -0.15 -0.07  0.21  0.43  0.21
## V48  0.00  0.06  0.12  0.25  0.16  0.10  0.04 -0.10 -0.11  0.05  0.12  0.19
## V49  0.51  0.35  0.22  0.20  0.14  0.17  0.44  0.02  0.35  0.31  0.12  0.03
## V50  0.31  0.36  0.08  0.23  0.13  0.15  0.30  0.13  0.26  0.09  0.13  0.00
## V51  0.32  0.39  0.10  0.22  0.20  0.22  0.28 -0.01  0.24  0.24  0.18  0.01
## V52  0.23  0.17  0.38  0.26  0.22  0.18  0.30 -0.03  0.15  0.33  0.21  0.18
## V53  0.06  0.26  0.28  0.49  0.47  0.32  0.17 -0.12 -0.05  0.20  0.41  0.18
```

```
## V54  0.14  0.38  0.30  0.45  0.40  0.29  0.19 -0.07  0.03  0.19  0.37  0.14
## V55  0.07  0.10  0.21  0.21  0.20  0.15  0.13  0.02  0.01  0.14  0.19  0.23
## V56  0.10  0.25  0.29  0.39  0.38  0.40  0.18 -0.12  0.02  0.19  0.39  0.14
## V57  0.34  0.39  0.24  0.31  0.24  0.27  0.34  0.00  0.23  0.24  0.23  0.12
## V58  0.15  0.35  0.28  0.35  0.33  0.35  0.21 -0.15  0.02  0.26  0.35  0.19
## V59  0.42  0.29  0.19  0.14  0.07  0.12  0.35  0.05  0.38  0.27  0.07  0.06
## V60 -0.03  0.16  0.27  0.40  0.40  0.32  0.10 -0.14 -0.12  0.20  0.33  0.26
##       V25   V26   V27   V28   V29   V30   V31   V32   V33   V34   V35   V36
## V01 -0.02  0.37  0.25  0.10  0.04  0.18  0.14  0.14  0.07  0.27 -0.04  0.14
## V02  0.09  0.25  0.29 -0.09  0.16  0.16  0.16  0.17  0.10  0.31  0.00  0.23
## V03  0.11 -0.11 -0.17  0.11 -0.02 -0.06 -0.10 -0.06  0.02 -0.21  0.15 -0.07
## V04 -0.01  0.27  0.13  0.16  0.02  0.21  0.30  0.17  0.16  0.25  0.20  0.30
## V05 -0.11  0.39  0.23  0.30 -0.17  0.44  0.11  0.15  0.30  0.18  0.10  0.29
## V06  0.02  0.16  0.11  0.17 -0.10  0.33  0.04  0.05  0.22  0.07  0.00  0.15
## V07  0.11  0.26  0.17  0.10  0.16  0.11  0.33  0.17  0.08  0.21  0.10  0.17
## V08 -0.09  0.15  0.05  0.23 -0.08  0.22  0.10  0.03  0.19  0.11  0.08  0.14
## V09 -0.15  0.44  0.29  0.38 -0.11  0.48  0.13  0.18  0.31  0.23  0.07  0.28
## V10 -0.03  0.34  0.40  0.09  0.00  0.25  0.22  0.22  0.17  0.37  0.02  0.27
## V11  0.41 -0.04  0.02 -0.26  0.55 -0.24  0.22  0.08 -0.10  0.13  0.15  0.09
## V12  0.00  0.32  0.26  0.10  0.17  0.14  0.32  0.17  0.04  0.38  0.11  0.16
## V13  0.11  0.22  0.25  0.02  0.31  0.07  0.37  0.23  0.00  0.39  0.12  0.23
## V14  0.00  0.34  0.37  0.03  0.08  0.25  0.20  0.20  0.18  0.38 -0.02  0.25
## V15 -0.02  0.26  0.11  0.20  0.02  0.31  0.21  0.18  0.25  0.21  0.18  0.29
## V16 -0.10  0.43  0.36  0.36 -0.14  0.48  0.16  0.22  0.34  0.21  0.08  0.31
## V17 -0.07  0.28  0.29  0.25 -0.11  0.40  0.09  0.15  0.32  0.17  0.01  0.30
## V18 -0.03  0.27  0.22  0.18 -0.07  0.31  0.13  0.13  0.22  0.23  0.05  0.32
## V19  0.05  0.27  0.29  0.10  0.20  0.13  0.33  0.27  0.14  0.40  0.16  0.27
## V20  0.32 -0.13  0.01 -0.21  0.40 -0.19  0.05  0.03 -0.05 -0.01  0.06 -0.03
## V21  0.26  0.08  0.13 -0.23  0.45 -0.11  0.29  0.12 -0.05  0.28  0.12  0.11
## V22  0.06  0.33  0.13  0.16  0.07  0.21  0.29  0.19  0.13  0.28  0.16  0.29
## V23 -0.05  0.26  0.25  0.13 -0.09  0.36  0.07  0.15  0.24  0.20  0.00  0.29
## V24  0.07  0.15  0.11  0.24 -0.04  0.25  0.13  0.10  0.19  0.03  0.26  0.23
## V25  1.00 -0.19  0.04 -0.20  0.37 -0.17  0.06  0.08 -0.14  0.03  0.10  0.04
## V26 -0.19  1.00  0.25  0.33 -0.10  0.38  0.22  0.18  0.25  0.31  0.10  0.26
## V27  0.04  0.25  1.00  0.10  0.03  0.19  0.15  0.23  0.07  0.23  0.07  0.13
## V28 -0.20  0.33  0.10  1.00 -0.24  0.31  0.16  0.16  0.24  0.05  0.14  0.09
## V29  0.37 -0.10  0.03 -0.24  1.00 -0.19  0.24  0.12 -0.07  0.16  0.15  0.12
## V30 -0.17  0.38  0.19  0.31 -0.19  1.00  0.12  0.13  0.28  0.14  0.07  0.24
## V31  0.06  0.22  0.15  0.16  0.24  0.12  1.00  0.27  0.14  0.37  0.35  0.38
## V32  0.08  0.18  0.23  0.16  0.12  0.13  0.27  1.00  0.21  0.40  0.22  0.26
## V33 -0.14  0.25  0.07  0.24 -0.07  0.28  0.14  0.21  1.00  0.22  0.14  0.39
## V34  0.03  0.31  0.23  0.05  0.16  0.14  0.37  0.40  0.22  1.00  0.16  0.41
## V35  0.10  0.10  0.07  0.14  0.15  0.07  0.35  0.22  0.14  0.16  1.00  0.23
## V36  0.04  0.26  0.13  0.09  0.12  0.24  0.38  0.26  0.39  0.41  0.23  1.00
## V37  0.15  0.17  0.13 -0.05  0.40 -0.06  0.41  0.14 -0.08  0.28  0.21  0.10
## V38  0.04  0.31  0.19  0.16  0.13  0.18  0.41  0.29  0.30  0.44  0.19  0.45
## V39 -0.06  0.36  0.10  0.22 -0.07  0.28  0.19  0.20  0.45  0.30  0.13  0.45
## V40  0.00  0.27  0.26  0.14  0.08  0.24  0.38  0.41  0.24  0.49  0.19  0.41
## V41  0.10  0.16  0.20  0.05  0.28 -0.01  0.45  0.17  0.00  0.34  0.22  0.17
## V42 -0.04  0.28  0.23  0.13  0.14  0.16  0.39  0.32  0.23  0.46  0.18  0.38
## V43  0.06  0.22  0.22  0.14  0.08  0.22  0.33  0.41  0.23  0.42  0.19  0.34
## V44 -0.01  0.29  0.15  0.13  0.11  0.18  0.43  0.30  0.22  0.41  0.18  0.43
## V45  0.04  0.19  0.16  0.16  0.03  0.29  0.26  0.26  0.44  0.32  0.10  0.42
## V46 -0.01  0.14  0.10  0.19 -0.07  0.22  0.21  0.23  0.36  0.22  0.18  0.34
```

```
## V47 -0.04  0.19  0.12  0.15 -0.01  0.27  0.12  0.28  0.47  0.24  0.12  0.42
## V48 -0.07  0.22  0.09  0.45 -0.18  0.22  0.03  0.21  0.31  0.14  0.05  0.18
## V49  0.08  0.30  0.17  0.11  0.24  0.13  0.48  0.39  0.20  0.53  0.26  0.38
## V50  0.11  0.18  0.38  0.06  0.15  0.10  0.23  0.34  0.13  0.38  0.10  0.20
## V51 -0.03  0.28  0.27  0.06  0.10  0.13  0.27  0.21  0.14  0.45  0.07  0.28
## V52  0.04  0.27  0.13  0.11  0.13  0.18  0.44  0.23  0.29  0.39  0.20  0.39
## V53 -0.05  0.23  0.17  0.19 -0.04  0.36  0.16  0.28  0.51  0.26  0.11  0.41
## V54 -0.05  0.30  0.19  0.20 -0.01  0.34  0.23  0.29  0.42  0.36 -0.02  0.46
## V55  0.12  0.10  0.05  0.06  0.02  0.18  0.11  0.15  0.24  0.12  0.26  0.24
## V56 -0.05  0.15  0.14  0.14 -0.04  0.24  0.18  0.21  0.32  0.24  0.10  0.33
## V57  0.07  0.29  0.37  0.08  0.13  0.22  0.31  0.39  0.29  0.54  0.13  0.39
## V58 -0.07  0.21  0.15  0.13 -0.06  0.29  0.21  0.21  0.42  0.31  0.13  0.43
## V59  0.07  0.28  0.18  0.06  0.28  0.10  0.51  0.27  0.11  0.42  0.26  0.36
## V60 -0.02  0.17  0.15  0.26 -0.12  0.32  0.13  0.22  0.42  0.15  0.15  0.31
##       V37   V38   V39   V40   V41   V42   V43   V44   V45   V46   V47   V48
## V01  0.22  0.19  0.16  0.22  0.19  0.25  0.15  0.24  0.11  0.09  0.11  0.06
## V02  0.27  0.26  0.16  0.25  0.23  0.24  0.21  0.23  0.19  0.12  0.15 -0.04
## V03 -0.11 -0.20 -0.03 -0.13 -0.14 -0.20 -0.07 -0.16 -0.12 -0.01  0.00  0.10
## V04  0.12  0.29  0.24  0.26  0.22  0.26  0.24  0.32  0.21  0.12  0.17  0.09
## V05  0.00  0.18  0.29  0.22  0.03  0.15  0.24  0.17  0.34  0.31  0.42  0.18
## V06 -0.05  0.06  0.17  0.15 -0.03  0.06  0.15  0.12  0.26  0.28  0.33  0.10
## V07  0.24  0.29  0.14  0.19  0.21  0.20  0.21  0.23  0.15  0.07  0.11  0.04
## V08 -0.02  0.09  0.13  0.03  0.00  0.07  0.10  0.15  0.20  0.21  0.19  0.15
## V09  0.00  0.20  0.32  0.25  0.07  0.20  0.25  0.27  0.35  0.32  0.32  0.26
## V10  0.16  0.32  0.21  0.30  0.20  0.30  0.27  0.35  0.28  0.16  0.23  0.08
## V11  0.38  0.11 -0.06  0.10  0.25  0.11  0.11  0.10  0.03 -0.07 -0.05 -0.15
## V12  0.35  0.29  0.13  0.24  0.28  0.33  0.17  0.28  0.07  0.06  0.00  0.03
## V13  0.40  0.31  0.11  0.30  0.34  0.27  0.26  0.26  0.15  0.05  0.09  0.00
## V14  0.24  0.33  0.21  0.31  0.23  0.32  0.25  0.30  0.23  0.19  0.19  0.06
## V15  0.04  0.24  0.22  0.27  0.11  0.24  0.24  0.23  0.33  0.26  0.25  0.12
## V16 -0.01  0.26  0.35  0.27  0.08  0.19  0.30  0.29  0.41  0.34  0.37  0.25
## V17 -0.03  0.17  0.29  0.20  0.04  0.20  0.23  0.22  0.37  0.34  0.42  0.16
## V18  0.03  0.22  0.27  0.22  0.07  0.25  0.23  0.20  0.23  0.23  0.32  0.10
## V19  0.29  0.35  0.20  0.34  0.28  0.30  0.28  0.36  0.25  0.12  0.16  0.04
## V20  0.15 -0.03 -0.09 -0.02  0.11 -0.02  0.00 -0.02 -0.06 -0.09 -0.15 -0.10
## V21  0.44  0.25  0.00  0.20  0.41  0.26  0.14  0.23  0.04 -0.05 -0.07 -0.11
## V22  0.08  0.31  0.26  0.25  0.13  0.30  0.25  0.32  0.28  0.21  0.21  0.05
## V23 -0.02  0.21  0.26  0.22  0.03  0.18  0.22  0.20  0.35  0.30  0.43  0.12
## V24 -0.05  0.11  0.17  0.13  0.04  0.09  0.11  0.13  0.17  0.23  0.21  0.19
## V25  0.15  0.04 -0.06  0.00  0.10 -0.04  0.06 -0.01  0.04 -0.01 -0.04 -0.07
## V26  0.17  0.31  0.36  0.27  0.16  0.28  0.22  0.29  0.19  0.14  0.19  0.22
## V27  0.13  0.19  0.10  0.26  0.20  0.23  0.22  0.15  0.16  0.10  0.12  0.09
## V28 -0.05  0.16  0.22  0.14  0.05  0.13  0.14  0.13  0.16  0.19  0.15  0.45
## V29  0.40  0.13 -0.07  0.08  0.28  0.14  0.08  0.11  0.03 -0.07 -0.01 -0.18
## V30 -0.06  0.18  0.28  0.24 -0.01  0.16  0.22  0.18  0.29  0.22  0.27  0.22
## V31  0.41  0.41  0.19  0.38  0.45  0.39  0.33  0.43  0.26  0.21  0.12  0.03
## V32  0.14  0.29  0.20  0.41  0.17  0.32  0.41  0.30  0.26  0.23  0.28  0.21
## V33 -0.08  0.30  0.45  0.24  0.00  0.23  0.23  0.22  0.44  0.36  0.47  0.31
## V34  0.28  0.44  0.30  0.49  0.34  0.46  0.42  0.41  0.32  0.22  0.24  0.14
## V35  0.21  0.19  0.13  0.19  0.22  0.18  0.19  0.18  0.10  0.18  0.12  0.05
## V36  0.10  0.45  0.45  0.41  0.17  0.38  0.34  0.43  0.42  0.34  0.42  0.18
## V37  1.00  0.31 -0.05  0.22  0.47  0.31  0.19  0.22 -0.03 -0.03 -0.10 -0.10
## V38  0.31  1.00  0.32  0.46  0.26  0.43  0.39  0.35  0.34  0.25  0.23  0.14
## V39 -0.05  0.32  1.00  0.32  0.07  0.27  0.27  0.28  0.38  0.28  0.44  0.29
```

```
## V40  0.22  0.46  0.32  1.00  0.28  0.46  0.47  0.41  0.29  0.29  0.32  0.19
## V41  0.47  0.26  0.07  0.28  1.00  0.33  0.23  0.29  0.10  0.03 -0.08 -0.02
## V42  0.31  0.43  0.27  0.46  0.33  1.00  0.42  0.38  0.31  0.27  0.24  0.12
## V43  0.19  0.39  0.27  0.47  0.23  0.42  1.00  0.37  0.37  0.36  0.31  0.21
## V44  0.22  0.35  0.28  0.41  0.29  0.38  0.37  1.00  0.32  0.31  0.22  0.02
## V45 -0.03  0.34  0.38  0.29  0.10  0.31  0.37  0.32  1.00  0.46  0.48  0.21
## V46 -0.03  0.25  0.28  0.29  0.03  0.27  0.36  0.31  0.46  1.00  0.44  0.21
## V47 -0.10  0.23  0.44  0.32 -0.08  0.24  0.31  0.22  0.48  0.44  1.00  0.26
## V48 -0.10  0.14  0.29  0.19 -0.02  0.12  0.21  0.02  0.21  0.21  0.26  1.00
## V49  0.41  0.44  0.29  0.46  0.35  0.39  0.42  0.38  0.30  0.21  0.22  0.13
## V50  0.30  0.31  0.13  0.37  0.34  0.34  0.39  0.28  0.25  0.16  0.19  0.09
## V51  0.27  0.35  0.22  0.34  0.34  0.47  0.32  0.43  0.21  0.16  0.18  0.14
## V52  0.20  0.40  0.27  0.38  0.24  0.42  0.34  0.36  0.40  0.29  0.28  0.07
## V53 -0.12  0.26  0.46  0.34 -0.01  0.27  0.35  0.32  0.55  0.49  0.59  0.31
## V54  0.03  0.38  0.44  0.41  0.10  0.30  0.37  0.37  0.50  0.41  0.49  0.29
## V55 -0.01  0.17  0.25  0.22  0.03  0.14  0.25  0.15  0.29  0.33  0.23  0.18
## V56 -0.01  0.24  0.32  0.28  0.05  0.33  0.31  0.34  0.41  0.43  0.51  0.13
## V57  0.20  0.40  0.32  0.48  0.26  0.41  0.39  0.42  0.38  0.30  0.35  0.23
## V58 -0.02  0.34  0.36  0.31  0.08  0.32  0.33  0.35  0.45  0.44  0.43  0.22
## V59  0.48  0.50  0.15  0.40  0.44  0.42  0.35  0.38  0.25  0.13  0.11  0.05
## V60 -0.15  0.16  0.37  0.25 -0.03  0.18  0.29  0.21  0.47  0.49  0.52  0.35
##       V49   V50   V51   V52   V53   V54   V55   V56   V57   V58   V59   V60
## V01  0.26  0.24  0.42  0.12  0.15  0.25  0.03  0.22  0.28  0.19  0.21  0.09
## V02  0.30  0.31  0.29  0.15  0.23  0.31  0.08  0.18  0.29  0.26  0.23  0.11
## V03 -0.18 -0.21 -0.21 -0.11 -0.02 -0.14  0.05 -0.09 -0.16 -0.15 -0.18  0.07
## V04  0.21  0.03  0.18  0.36  0.17  0.22  0.08  0.17  0.20  0.23  0.25  0.18
## V05  0.19  0.11  0.18  0.22  0.38  0.35  0.16  0.26  0.22  0.30  0.11  0.33
## V06  0.04 -0.01  0.08  0.15  0.33  0.22  0.18  0.33  0.15  0.26 -0.01  0.36
## V07  0.23  0.15  0.20  0.23  0.07  0.11  0.06  0.07  0.19  0.12  0.30  0.03
## V08  0.16  0.05  0.12  0.15  0.16  0.14  0.08  0.15  0.12  0.17  0.05  0.17
## V09  0.17  0.16  0.24  0.23  0.43  0.39  0.15  0.34  0.26  0.33  0.14  0.34
## V10  0.25  0.26  0.33  0.25  0.23  0.34  0.09  0.23  0.34  0.31  0.28  0.16
## V11  0.22  0.12  0.09  0.10 -0.10 -0.02  0.03 -0.06  0.11 -0.06  0.23 -0.10
## V12  0.33  0.19  0.43  0.19  0.08  0.16  0.01  0.12  0.30  0.17  0.32  0.05
## V13  0.51  0.31  0.32  0.23  0.06  0.14  0.07  0.10  0.34  0.15  0.42 -0.03
## V14  0.35  0.36  0.39  0.17  0.26  0.38  0.10  0.25  0.39  0.35  0.29  0.16
## V15  0.22  0.08  0.10  0.38  0.28  0.30  0.21  0.29  0.24  0.28  0.19  0.27
## V16  0.20  0.23  0.22  0.26  0.49  0.45  0.21  0.39  0.31  0.35  0.14  0.40
## V17  0.14  0.13  0.20  0.22  0.47  0.40  0.20  0.38  0.24  0.33  0.07  0.40
## V18  0.17  0.15  0.22  0.18  0.32  0.29  0.15  0.40  0.27  0.35  0.12  0.32
## V19  0.44  0.30  0.28  0.30  0.17  0.19  0.13  0.18  0.34  0.21  0.35  0.10
## V20  0.02  0.13 -0.01 -0.03 -0.12 -0.07  0.02 -0.12  0.00 -0.15  0.05 -0.14
## V21  0.35  0.26  0.24  0.15 -0.05  0.03  0.01  0.02  0.23  0.02  0.38 -0.12
## V22  0.31  0.09  0.24  0.33  0.20  0.19  0.14  0.19  0.24  0.26  0.27  0.20
## V23  0.12  0.13  0.18  0.21  0.41  0.37  0.19  0.39  0.23  0.35  0.07  0.33
## V24  0.03  0.00  0.01  0.18  0.18  0.14  0.23  0.14  0.12  0.19  0.06  0.26
## V25  0.08  0.11 -0.03  0.04 -0.05 -0.05  0.12 -0.05  0.07 -0.07  0.07 -0.02
## V26  0.30  0.18  0.28  0.27  0.23  0.30  0.10  0.15  0.29  0.21  0.28  0.17
## V27  0.17  0.38  0.27  0.13  0.17  0.19  0.05  0.14  0.37  0.15  0.18  0.15
## V28  0.11  0.06  0.06  0.11  0.19  0.20  0.06  0.14  0.08  0.13  0.06  0.26
## V29  0.24  0.15  0.10  0.13 -0.04 -0.01  0.02 -0.04  0.13 -0.06  0.28 -0.12
## V30  0.13  0.10  0.13  0.18  0.36  0.34  0.18  0.24  0.22  0.29  0.10  0.32
## V31  0.48  0.23  0.27  0.44  0.16  0.23  0.11  0.18  0.31  0.21  0.51  0.13
## V32  0.39  0.34  0.21  0.23  0.28  0.29  0.15  0.21  0.39  0.21  0.27  0.22
```

```
## V33  0.20  0.13  0.14  0.29  0.51  0.42  0.24  0.32  0.29  0.42  0.11  0.42
## V34  0.53  0.38  0.45  0.39  0.26  0.36  0.12  0.24  0.54  0.31  0.42  0.15
## V35  0.26  0.10  0.07  0.20  0.11 -0.02  0.26  0.10  0.13  0.13  0.26  0.15
## V36  0.38  0.20  0.28  0.39  0.41  0.46  0.24  0.33  0.39  0.43  0.36  0.31
## V37  0.41  0.30  0.27  0.20 -0.12  0.03 -0.01 -0.01  0.20 -0.02  0.48 -0.15
## V38  0.44  0.31  0.35  0.40  0.26  0.38  0.17  0.24  0.40  0.34  0.50  0.16
## V39  0.29  0.13  0.22  0.27  0.46  0.44  0.25  0.32  0.32  0.36  0.15  0.37
## V40  0.46  0.37  0.34  0.38  0.34  0.41  0.22  0.28  0.48  0.31  0.40  0.25
## V41  0.35  0.34  0.34  0.24 -0.01  0.10  0.03  0.05  0.26  0.08  0.44 -0.03
## V42  0.39  0.34  0.47  0.42  0.27  0.30  0.14  0.33  0.41  0.32  0.42  0.18
## V43  0.42  0.39  0.32  0.34  0.35  0.37  0.25  0.31  0.39  0.33  0.35  0.29
## V44  0.38  0.28  0.43  0.36  0.32  0.37  0.15  0.34  0.42  0.35  0.38  0.21
## V45  0.30  0.25  0.21  0.40  0.55  0.50  0.29  0.41  0.38  0.45  0.25  0.47
## V46  0.21  0.16  0.16  0.29  0.49  0.41  0.33  0.43  0.30  0.44  0.13  0.49
## V47  0.22  0.19  0.18  0.28  0.59  0.49  0.23  0.51  0.35  0.43  0.11  0.52
## V48  0.13  0.09  0.14  0.07  0.31  0.29  0.18  0.13  0.23  0.22  0.05  0.35
## V49  1.00  0.39  0.34  0.34  0.24  0.33  0.18  0.26  0.46  0.27  0.48  0.14
## V50  0.39  1.00  0.35  0.21  0.22  0.29  0.15  0.20  0.46  0.24  0.27  0.14
## V51  0.34  0.35  1.00  0.22  0.25  0.32  0.08  0.28  0.46  0.32  0.32  0.15
## V52  0.34  0.21  0.22  1.00  0.25  0.34  0.14  0.30  0.32  0.27  0.37  0.23
## V53  0.24  0.22  0.25  0.25  1.00  0.59  0.31  0.50  0.41  0.53  0.13  0.59
## V54  0.33  0.29  0.32  0.34  0.59  1.00  0.25  0.44  0.42  0.50  0.24  0.43
## V55  0.18  0.15  0.08  0.14  0.31  0.25  1.00  0.26  0.21  0.28  0.09  0.36
## V56  0.26  0.20  0.28  0.30  0.50  0.44  0.26  1.00  0.31  0.50  0.21  0.45
## V57  0.46  0.46  0.46  0.32  0.41  0.42  0.21  0.31  1.00  0.40  0.31  0.33
## V58  0.27  0.24  0.32  0.27  0.53  0.50  0.28  0.50  0.40  1.00  0.21  0.47
## V59  0.48  0.27  0.32  0.37  0.13  0.24  0.09  0.21  0.31  0.21  1.00  0.01
## V60  0.14  0.14  0.15  0.23  0.59  0.43  0.36  0.45  0.33  0.47  0.01  1.00
```

```
#plot(personality.Vs);  # too big ...
    symnum( cor(personality.Vs),
            diag = TRUE,
            corr = TRUE,
            cutpoints=c(0.15,0.30,0.60,0.90),
            symbols = c(" ",".","*","**","***")
          );
```

```
##     V01 V02 V03 V04 V05 V06 V07 V08 V09 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19
## V01 1
## V02 *   1
## V03 .   *   1
## V04             1
## V05 .   .       .   1
## V06 .   .           *   1
## V07 .           .   .       1
## V08                 .   .       1
## V09 .   .       .   *   *       .   1
## V10 *   *   .   .   .   .   .       *   1
## V11                 .       .       .       1
## V12 *   .   .   .   .       .       .   *   .   1
## V13 .   *   .   .           *   .       *   .   *   1
## V14 *   **  *   .   .   .   .       *   *       *   *   1
## V15             *   .   .   .   .   *   .       .   .   .   1
## V16 .   *       .   *   *   .   .   *   *   .   .   .   *   *   1
## V17 .   .       .   *   *       .   *   *       .       .   *   *   1
```

```
## V18 .   *       .   *   *   .   .   *   *       .   .   *   .   *   *   1
## V19 .   .   .   .           .       .   *   .   *   *   *   .   .   .   .   1
## V20         .   .   .   .           .       *                   .   .   .
## V21 .   .   .               .           .   *   *   *   .                   *
## V22 .           *   .       *   .   .   .       .   .   .   *   .   .   .   .
## V23 .   *       .   *   *       .   *   *               *   .   *   *   *   .
## V24         .   .   .   .           .                       .   .   .   .
## V25                                         *
## V26 *   .       .   *   .   .   .   *   *       *   .   *   .   *   .   .   .
## V27 .   .   .       .       .       .   *       .   .   *       *   .   .   .
## V28             .   .   .       .   *       .               .   *   .   .
## V29     .           .       .               *   .   *                       .
## V30 .   .       .   *   *       .   *   .   .           .   *   *   *   *
## V31     .       *           *           .   .   *   *   .   .   .           *
## V32     .       .   .       .       .   .       .   .   .   .   .           .
## V33             .   .   .       .   *   .               .   .   *   *   .
## V34 .   *   .   .   .       .       .   *       *   *   *   .   .   .   .   *
## V35         .   .                       .               .               .
## V36     .       .   .       .       .   .       .   .   .   .   *   .   *   .
## V37 .   .                   .           .   *   *   *   .                   .
## V38 .   .   .   .   .       .       .   *       .   *   *   .   .   .   .   *
## V39 .   .       .   .   .           *   .               .   .   *   .   .   .
## V40 .   .       .   .   .   .       .   *       .   .   *   .   .   .   .   *
## V41 .   .       .           .           .   .   .   *   .                   .
## V42 .   .   .   .           .       .   .       *   .   *   .   .   .   .   *
## V43 .   .       .   .       .       .   .       .   .   .   .   .   .   .   .
## V44 .   .   .   *   .       .       .   *       .   .   .   .   .   .   .   *
## V45     .       .   *   .   .   .   *   .               .   *   *   *   .   .
## V46                 *   .       .   *   .               .   .   *   *   .
## V47             .   *   *       .   *   .               .   .   *   *   *   .
## V48                 .           .   .                           .   .
## V49 .   *   .   .   .       .   .   .   .   .   *   *   *   .   .       .   *
## V50 .   *   .               .       .   .       .   *   *       .           .
## V51 *   .   .   .   .       .       .   *       *   *   *       .   .   .   .
## V52             *   .   .   .   .   .   .       .   .   .   *   .   .   .   *
## V53     .       .   *   *       .   *   .               .   .   *   *   *   .
## V54 .   *       .   *   .           *   *       .       *   *   *   *   .   .
## V55                 .   .           .                       .   .   .
## V56 .   .       .   .   *           *   .               .   .   *   *   *   .
## V57 .   .   .   .   .   .   .       .   *       *   *   *   .   *   .   .   *
## V58 .   .       .   .   .       .   *   *       .       *   .   *   *   *   .
## V59 .   .   .   .           *           .   .   *   *   .   .               *
## V60             .   *   *       .   *   .               .   .   *   *   *
##     V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38
## V01
## V02
## V03
## V04
## V05
## V06
## V07
## V08
## V09
## V10
```

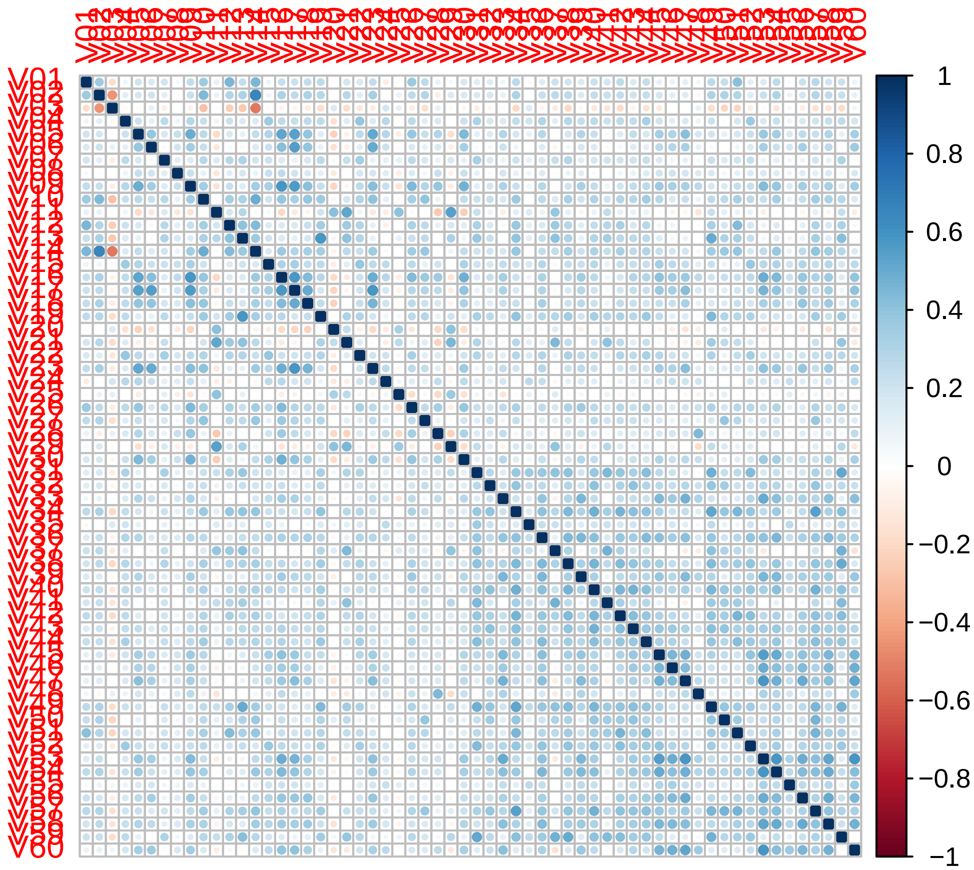
```
## V11
## V12
## V13
## V14
## V15
## V16
## V17
## V18
## V19
## V20 1
## V21 .   1
## V22         1
## V23 .       .   1
## V24         .   .   1
## V25 *   .               1
## V26         *   .       .   1
## V27             .           .   1
## V28 .   .   .       .   .   *       1
## V29 *   *               *           .   1
## V30 .       .   *   .   .   *   .   *   .   1
## V31     .   .               .       .   .       1
## V32         .               .   .   .           .   1
## V33             .   .       .       .       .       .   1
## V34     .   .   .           *   .       .       *   *   .   1
## V35         .       .                   .       *   .       .   1
## V36         .   .   .       .               .   *   .   *   *   .   1
## V37     *                   .           *       *           .   .       1
## V38     .   *   .           *   .   .       .   *   .   .   *   .   *   *   1
## V39         .   .   .       *       .       .   .   .   *   *       *       *
## V40     .   .   .           .   .           .   *   *   .   *   .   *   .   *
## V41     *                   .   .       .       *   .       *   .   .   *   .
## V42     .   .   .           .   .           .   *   *   .   *   .   *   *   *
## V43         .   .           .   .           .   *   *   .   *   .   *   .   *
## V44     .   *   .           .               .   *   .   .   *   .   *   .   *
## V45         .   *   .       .   .   .       .   .   .   *   *       *       *
## V46         .   *   .               .       .   .   .   *   .   .   *       .
## V47         .   *   .       .       .       .       .   *   .       *       .
## V48                 .       .       *   .   .       .   *           .
## V49     *   *               *   .       .       *   *   .   *   .   *   *   *
## V50     .                   .   *       .       .   *       *       .   *   *
## V51     .   .   .           .   .               .   .       *       .   .   *
## V52         *   .   .       .               .   *   .   .   *   .   *   .   *
## V53         .   *   .       .   .   .       *   .   .   *   .       *       .
## V54         .   *           .   .   .       *   .   .   *   *       *       *
## V55             .   .                       .           .       .   .       .
## V56         .   *                           .   .   .   *   .       *       .
## V57     .   .   .           .   *           .   *   *   .   *       *   .   *
## V58 .       .   *   .       .   .           .   .   .   *   *       *       *
## V59     *   .               .   .       .       *   .       *   .   *   *   *
## V60         .   *   .       .       .       *       .   *       .   *   .   .
##     V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57
## V01
## V02
## V03
```

```
## V04
## V05
## V06
## V07
## V08
## V09
## V10
## V11
## V12
## V13
## V14
## V15
## V16
## V17
## V18
## V19
## V20
## V21
## V22
## V23
## V24
## V25
## V26
## V27
## V28
## V29
## V30
## V31
## V32
## V33
## V34
## V35
## V36
## V37
## V38
## V39 1
## V40 *   1
## V41     .   1
## V42 .   *   *   1
## V43 .   *   .   *   1
## V44 .   *   .   *   *   1
## V45 *   .       *   *   *   1
## V46 .   .       .   *   *   *   1
## V47 *   *       .   *   .   *   *   1
## V48 .   .           .       .   .   .   1
## V49 .   *   *   *   *   *   *   .   .       1
## V50     *   *   *   *   .   .   .   .       *   1
## V51 .   *   *   *   *   *   .   .   .       *   *   1
## V52 .   *   .   *   *   *   *   .   .       *   .   .   1
## V53 *   *       .   *   *   *   *   *   *   .   .   .   .   1
## V54 *   *       *   *   *   *   *   *   .   *   .   *   *   *   1
## V55 .   .           .       .   *   .   .   .               *   .   1
## V56 *   .       *   *   *   *   *   *       .   .   .   .   *   *   .   1
## V57 *   *   .   *   *   *   *   *   *   .   *   *   *   *   *   *   .   *   1
```

```
## V58 *   *       *   *   *   *   *   *   .   .   .   *   .   *   *   .   *   *
## V59     *   *   *   *   *   .               *   .   *   *       .       .   *
## V60 *   .       .   .   .   *   *   *   *           .   .   *   *   *   *   *
##     V58 V59 V6
## V01
## V02
## V03
## V04
## V05
## V06
## V07
## V08
## V09
## V10
## V11
## V12
## V13
## V14
## V15
## V16
## V17
## V18
## V19
## V20
## V21
## V22
## V23
## V24
## V25
## V26
## V27
## V28
## V29
## V30
## V31
## V32
## V33
## V34
## V35
## V36
## V37
## V38
## V39
## V40
## V41
## V42
## V43
## V44
## V45
## V46
## V47
## V48
## V49
## V50
```

```
## V51
## V52
## V53
## V54
## V55
## V56
## V57
## V58 1
## V59 .    1
## V60 *        1
## attr(,"legend")
## [1] 0 ' ' 0.15 '.' 0.3 '*' 0.6 '**' 0.9 '***' 1
```

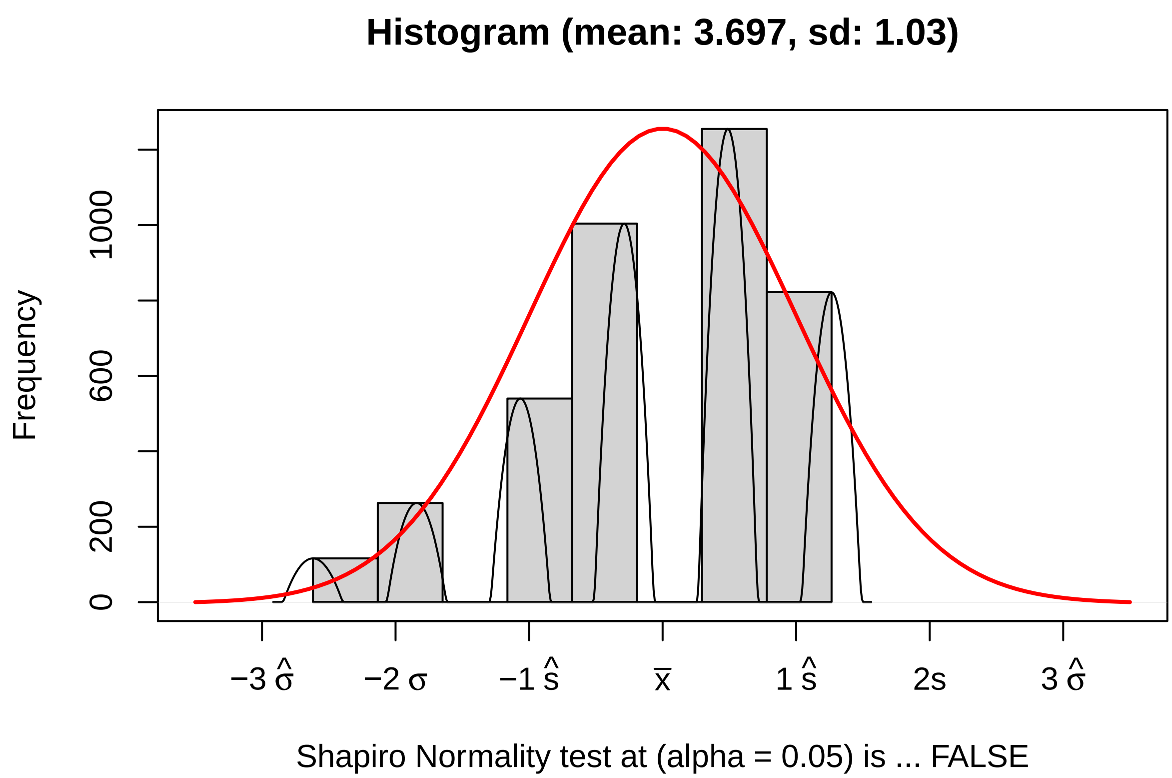
```
    corrplot::corrplot( (cor(personality.Vs)) );
```



```
## let's look at all the data ...
Vs = sample( as.vector(unlist(personality.Vs)) );
head(Vs);
```

```
## [1] 3.4 2.6 3.4 2.6 1.0 4.2
```

```
normalDiagnosticPlot(Vs[1:4000]);  # shapiro.test breaks if too large ...
```




We have too many observations (n=678) and too many variables (m=60) to be able to effectively analyze this data. We need to begin performing multivariate analysis.

In general, we treat the rows as observations and the columns as features, factors, or variables. We could transpose the dataframe and reverse rows/cols. As we proceed, remember this. We will talk about general forms of data manipulation called data reduction.

In general, we want to reduce the number of factors to consider. Or we may want to classify the subjects observed (the rows) into like groups. For the next two weeks we will consider "exploratory data analysis of multivariate data".

### Correlation does not imply causation

Does zero correlation imply independence?

```
X = rnorm(100,0,1);
  Y = ( X - mean(X) ) / sd(X);
cor(X,Y);  ## ARGH!
```

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

```
  Y = X^2;
cor(X,Y);  ## Getting closer
```

```
## [1] -0.03895873
```

```
  Y = ( sample(X) - mean(X) ) / sd(X);
cor(X,Y);  ## Getting closer
```

```
## [1] -0.09396706
```

```
# build a Y that is a function of X such that correlation is 0.
# it can be close to zero with isClose function
# ?isClose

# maybe review the formula for correlation
# \url{https://en.wikipedia.org/wiki/Correlation_and_dependence#Sample_correlation_coefficient}

# \url{https://math.stackexchange.com/questions/444408/why-does-zero-correlation-not-imply-independence
# \url{https://stats.stackexchange.com/questions/413326/why-does-independence-imply-zero-correlation}

# if need be, you can change the form of X, but it should be of length 100...
```