Step by Step Analysis of WIN data

June 29, 2018

1 Step by Step Statistical Analysis of Structural Topology Measures and the IGT in the WIN Data

1.0.1 1. Calculate the Data Distribution (Boxplot statistics, Grubb's outliers, skewness

1.0.2 2. Calculate Generalized Linear Model for both Binary and Weighted

1.0.3 3. Calculate Correlation Matrix for both Binary and Weighted

1.0.4 4. Run Principal Component Regression with Standardized Data

1.0.5 5. Run Principal Component Analysis with Standardized Data

1.0.6 6. Run Singular Value Decomposition with Standardized Data

1.1 Calculate the Data Distribution (Boxplot statistics, Grubb's outliers, skewness

Making mergedWINData Table

```
In [10]: mergedWINData = read.csv(file="~/Desktop/mergedWINData.csv")
         mergedWINData$X = NULL
```

Data Distribution Calculation with summary()

```
In [11]: library(plyr)
         summary(mergedWINData$density_baseline)
         summary(mergedWINData$clustering_coeff_average.binary.)
         summary(mergedWINData$clustering_coeff_average.binary._baseline)
         summary(mergedWINData$transitivity.binary._baseline)
         summary(mergedWINData$network_characteristic_path_length.binary._baseline)
         summary(mergedWINData$small.worldness.binary._baseline)
         summary(mergedWINData$global_efficiency.binary._baseline)
         summary(mergedWINData$diameter_of_graph.binary._baseline)
         summary(mergedWINData$radius_of_graph.binary._baseline)
         summary(mergedWINData$local_efficiency.binary._baseline)
         summary(mergedWINData$assortativity_coefficient.binary._baseline)
         summary(mergedWINData$transitivity.weighted._baseline)
         summary(mergedWINData$network_characteristic_path_length.weighted._baseline)
         summary(mergedWINData$small.worldness.weighted._baseline)
         summary(mergedWINData$global_efficiency.weighted._baseline)
```

```r
summary(mergedWINData$diameter_of_graph.weighted._baseline)
summary(mergedWINData$radius_of_graph.weighted._baseline)
summary(mergedWINData$local_efficiency.weighted._baseline)
summary(mergedWINData$assortativity_coefficient.weighted._baseline)
summary(mergedWINData$baseline_p)
summary(mergedWINData$baseline_q)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.04231 0.05235 0.05500 0.05530 0.05817 0.06980       2 
```

```
Length  Class   Mode 
     0   NULL   NULL 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
 0.1280  0.2256  0.2534  0.2529  0.2773  0.3890       2 
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
0.005455 0.023410 0.035647 0.039178 0.052757 0.127768        2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  2.979   3.331   3.455   3.516   3.682   4.582       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.03603 0.06313 0.07209 0.07240 0.08142 0.10571       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
 0.2957  0.3370  0.3490  0.3481  0.3599  0.3956       2 
```

1	**10**	**11**	**12**	**13**	**14**	**6**	**7**	**8**	**9**	**NAN**
1	12	5	1	1	1	8	32	44	20	1

1	**2**	**3**	**4**	**5**	**6**	**7**	**NAN**
1	34	1	45	40	3	1	1

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  9.367  16.878  18.656  18.670  20.631  28.049       2 
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's 
-0.308120 -0.161616 -0.084556 -0.079472  0.008365  0.191067         2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.03301 0.08003 0.10211 0.10767 0.12607 0.23773       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  2.979   3.331   3.455   3.516   3.682   4.582       2 
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
0.001285 0.004217 0.006244 0.006611 0.007946 0.016948        2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.06073 0.07883 0.08671 0.08749 0.09609 0.11950       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  48.09   90.37  141.67  159.97  195.03  640.17       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  24.74   50.81   74.02   99.23  134.89  384.00       2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  5.144   6.967   7.682   7.782   8.579  13.208       2 
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
-0.18338 -0.07512 -0.02054 -0.02201  0.02934  0.23170        2 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
 -54.00    0.00   22.00   21.59   46.00   70.00       1 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
 -60.00   14.00   38.00   31.35   52.00   84.00       1 
```

Grubb's outlier Calculation with grubbs.test()

In [12]:
```
library(outliers)
grubbs.test(mergedWINData$density_baseline)
grubbs.test(mergedWINData$clustering_coeff_average.binary.)
grubbs.test(mergedWINData$clustering_coeff_average.binary._baseline)
grubbs.test(mergedWINData$transitivity.binary._baseline)
grubbs.test(mergedWINData$network_characteristic_path_length.binary._baseline)
grubbs.test(mergedWINData$small.worldness.binary._baseline)
grubbs.test(mergedWINData$global_efficiency.binary._baseline)
grubbs.test(mergedWINData$diameter_of_graph.binary._baseline)
grubbs.test(mergedWINData$radius_of_graph.binary._baseline)
grubbs.test(mergedWINData$local_efficiency.binary._baseline)
```

```
grubbs.test(mergedWINData$assortativity_coefficient.binary._baseline)
grubbs.test(mergedWINData$transitivity.weighted._baseline)
grubbs.test(mergedWINData$network_characteristic_path_length.weighted._baseline)
grubbs.test(mergedWINData$small.worldness.weighted._baseline)
grubbs.test(mergedWINData$global_efficiency.weighted._baseline)
grubbs.test(mergedWINData$diameter_of_graph.weighted._baseline)
grubbs.test(mergedWINData$radius_of_graph.weighted._baseline)
grubbs.test(mergedWINData$local_efficiency.weighted._baseline)
grubbs.test(mergedWINData$assortativity_coefficient.weighted._baseline)
grubbs.test(mergedWINData$baseline_p)
grubbs.test(mergedWINData$baseline_q)
```

```
Grubbs test for one outlier

data:  mergedWINData$density_baseline
G = 3.26710, U = 0.91251, p-value = 0.05283
alternative hypothesis: highest value 0.0698043 is an outlier
```

```
Error in complete.cases(x): no input has determined the number of cases
Traceback:
```

1. grubbs.test(mergedWINData$clustering_coeff_average.binary.)

2. sort(x[complete.cases(x)])

3. complete.cases(x)

Calculate Skewness for continuous variables with skew()

```
In [13]: library(psych)
         skew(mergedWINData$density_baseline)
         skew(mergedWINData$clustering_coeff_average.binary._baseline)
         skew(mergedWINData$transitivity.binary._baseline)
         skew(mergedWINData$network_characteristic_path_length.binary._baseline)
         skew(mergedWINData$small.worldness.binary._baseline)
         skew(mergedWINData$global_efficiency.binary._baseline)
         skew(mergedWINData$local_efficiency.binary._baseline)
         skew(mergedWINData$assortativity_coefficient.binary._baseline)
         skew(mergedWINData$transitivity.weighted._baseline)
         skew(mergedWINData$network_characteristic_path_length.weighted._baseline)
         skew(mergedWINData$small.worldness.weighted._baseline)
         skew(mergedWINData$global_efficiency.weighted._baseline)
```

```
skew(mergedWINData$diameter_of_graph.weighted._baseline)
skew(mergedWINData$radius_of_graph.weighted._baseline)
skew(mergedWINData$local_efficiency.weighted._baseline)
skew(mergedWINData$assortativity_coefficient.weighted._baseline)
skew(mergedWINData$baseline_p)
skew(mergedWINData$baseline_q)
```

0.0268452834376448
0.112664834807907
1.07356177836757
1.23772661713605
0.0111348690235782
-0.359141232992264
0.0780135478797593
0.150226885923763
0.918417892511746
1.23772661713605
1.24774718146554
0.183816330924891
1.76066191356737
1.7163855691314
0.698468350927476
0.191784899548373
-0.211203145254837
-0.779505008723374

1.2 Calculate Generalized Linear Models

GLM for p and continuous binary variables (diameter and radius are not continuous)

```
In [14]: fit = glm(baseline_p~
                      density_baseline+
                      clustering_coeff_average.binary._baseline+
                      transitivity.binary._baseline+
                      network_characteristic_path_length.binary._baseline+
                      small.worldness.binary._baseline+
                      global_efficiency.binary._baseline+
                      local_efficiency.binary._baseline+
                      assortativity_coefficient.binary._baseline,
                  family = gaussian(identity),
                  data = mergedWINData)
         summary(fit)
```

```
Call:
glm(formula = baseline_p ~ density_baseline + clustering_coeff_average.binary._baseline +
    transitivity.binary._baseline + network_characteristic_path_length.binary._baseline +
    small.worldness.binary._baseline + global_efficiency.binary._baseline +
    local_efficiency.binary._baseline + assortativity_coefficient.binary._baseline,
```

```
    family = gaussian(identity), data = mergedWINData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-74.017  -21.220    0.836   19.488   47.147

Coefficients:
                                                     Estimate Std. Error
(Intercept)                                           35.9763   299.4038
density_baseline                                     673.9390   728.1496
clustering_coeff_average.binary._baseline           -394.6430   761.4294
transitivity.binary._baseline                         63.3034   130.0336
network_characteristic_path_length.binary._baseline    1.9502    45.3704
small.worldness.binary._baseline                    1050.2033  2458.2894
global_efficiency.binary._baseline                  -163.9697   582.4556
local_efficiency.binary._baseline                      0.9435     4.5832
assortativity_coefficient.binary._baseline           -31.1309    26.9734
                                                    t value Pr(>|t|)
(Intercept)                                           0.120    0.905
density_baseline                                      0.926    0.357
clustering_coeff_average.binary._baseline            -0.518    0.605
transitivity.binary._baseline                         0.487    0.627
network_characteristic_path_length.binary._baseline   0.043    0.966
small.worldness.binary._baseline                      0.427    0.670
global_efficiency.binary._baseline                   -0.282    0.779
local_efficiency.binary._baseline                     0.206    0.837
assortativity_coefficient.binary._baseline           -1.154    0.251

(Dispersion parameter for gaussian family taken to be 785.4927)

    Null deviance: 95493  on 123  degrees of freedom
Residual deviance: 90332  on 115  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 1189.2

Number of Fisher Scoring iterations: 2
```

GLM for q and continuous binary variables

```
In [15]: fit = glm(baseline_q~
                    density_baseline+
                    clustering_coeff_average.binary._baseline+
                    transitivity.binary._baseline+
                    network_characteristic_path_length.binary._baseline+
                    small.worldness.binary._baseline+
                    global_efficiency.binary._baseline+
```

```r
                  #diameter_of_graph.binary._baseline+
                  #radius_of_graph.binary._baseline+
                  local_efficiency.binary._baseline+
                  assortativity_coefficient.binary._baseline,
               family = gaussian(identity),
               data = mergedWINData)
     summary(fit)
```

```
Call:
glm(formula = baseline_q ~ density_baseline + clustering_coeff_average.binary._baseline +
    transitivity.binary._baseline + network_characteristic_path_length.binary._baseline +
    small.worldness.binary._baseline + global_efficiency.binary._baseline +
    local_efficiency.binary._baseline + assortativity_coefficient.binary._baseline,
    family = gaussian(identity), data = mergedWINData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-81.386  -19.030    7.608   18.196   56.797

Coefficients:
                                                     Estimate Std. Error
(Intercept)                                          -185.121    308.597
density_baseline                                     1587.651    750.508
clustering_coeff_average.binary._baseline            1075.209    784.810
transitivity.binary._baseline                        -326.595    134.026
network_characteristic_path_length.binary._baseline    -3.316     46.764
small.worldness.binary._baseline                    -2131.562   2533.773
global_efficiency.binary._baseline                    536.444    600.340
local_efficiency.binary._baseline                      -8.188      4.724
assortativity_coefficient.binary._baseline            -21.106     27.802
                                                    t value Pr(>|t|)
(Intercept)                                          -0.600   0.5498
density_baseline                                      2.115   0.0366 *
clustering_coeff_average.binary._baseline             1.370   0.1733
transitivity.binary._baseline                        -2.437   0.0164 *
network_characteristic_path_length.binary._baseline  -0.071   0.9436
small.worldness.binary._baseline                     -0.841   0.4019
global_efficiency.binary._baseline                    0.894   0.3734
local_efficiency.binary._baseline                    -1.733   0.0857 .
assortativity_coefficient.binary._baseline           -0.759   0.4493
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for gaussian family taken to be 834.4714)

    Null deviance: 106737  on 123  degrees of freedom
Residual deviance:  95964  on 115  degrees of freedom
```

```
  (2 observations deleted due to missingness)
AIC: 1196.7

Number of Fisher Scoring iterations: 2
```

GLM for p and continuous weighted variables

```
In [16]: fit = glm(baseline_p~
                      density_baseline+
                      transitivity.weighted._baseline+
                      network_characteristic_path_length.weighted._baseline+
                      small.worldness.weighted._baseline+
                      global_efficiency.weighted._baseline+
                      diameter_of_graph.weighted._baseline+
                      radius_of_graph.weighted._baseline+
                      local_efficiency.weighted._baseline+
                      assortativity_coefficient.weighted._baseline,
                   family = gaussian(identity),
                   data = mergedWINData)
         summary(fit)
```

```
Call:
glm(formula = baseline_p ~ density_baseline + transitivity.weighted._baseline +
    network_characteristic_path_length.weighted._baseline + small.worldness.weighted._baseline
    global_efficiency.weighted._baseline + diameter_of_graph.weighted._baseline +
    radius_of_graph.weighted._baseline + local_efficiency.weighted._baseline +
    assortativity_coefficient.weighted._baseline, family = gaussian(identity),
    data = mergedWINData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-80.257  -19.989   -0.427   19.116   49.560

Coefficients:
                                                        Estimate Std. Error
(Intercept)                                            3.629e+01  7.143e+01
density_baseline                                       5.617e+02  7.311e+02
transitivity.weighted._baseline                        1.622e+02  1.435e+02
network_characteristic_path_length.weighted._baseline -1.700e+01  1.025e+01
small.worldness.weighted._baseline                    -1.547e+03  2.072e+03
global_efficiency.weighted._baseline                   6.256e+01  2.984e+02
diameter_of_graph.weighted._baseline                  -5.164e-02  9.478e-02
radius_of_graph.weighted._baseline                     1.825e-03  1.255e-01
local_efficiency.weighted._baseline                    1.182e+00  2.496e+00
assortativity_coefficient.weighted._baseline          -1.874e+01  3.478e+01
```

```
                                                      t value Pr(>|t|)
(Intercept)                                             0.508    0.612
density_baseline                                        0.768    0.444
transitivity.weighted._baseline                         1.130    0.261
network_characteristic_path_length.weighted._baseline  -1.658    0.100
small.worldness.weighted._baseline                     -0.746    0.457
global_efficiency.weighted._baseline                    0.210    0.834
diameter_of_graph.weighted._baseline                   -0.545    0.587
radius_of_graph.weighted._baseline                      0.015    0.988
local_efficiency.weighted._baseline                     0.474    0.637
assortativity_coefficient.weighted._baseline           -0.539    0.591

(Dispersion parameter for gaussian family taken to be 778.8871)

    Null deviance: 95493  on 123  degrees of freedom
Residual deviance: 88793  on 114  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 1189

Number of Fisher Scoring iterations: 2
```

GLM for q and continuous weighted variables

```
In [17]: fit = glm(baseline_q~
                    density_baseline+
                    transitivity.weighted._baseline+
                    network_characteristic_path_length.weighted._baseline+
                    small.worldness.weighted._baseline+
                    global_efficiency.weighted._baseline+
                    diameter_of_graph.weighted._baseline+
                    radius_of_graph.weighted._baseline+
                    local_efficiency.weighted._baseline+
                    assortativity_coefficient.weighted._baseline,
                 family = gaussian(identity),
                 data = mergedWINData)
         summary(fit)
```

```
Call:
glm(formula = baseline_q ~ density_baseline + transitivity.weighted._baseline +
    network_characteristic_path_length.weighted._baseline + small.worldness.weighted._baseline
    global_efficiency.weighted._baseline + diameter_of_graph.weighted._baseline +
    radius_of_graph.weighted._baseline + local_efficiency.weighted._baseline +
    assortativity_coefficient.weighted._baseline, family = gaussian(identity),
    data = mergedWINData)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-82.266  -12.778    6.836   18.466   50.728

Coefficients:
                                                     Estimate Std. Error
(Intercept)                                         -30.29478   74.76561
density_baseline                                    830.25573  765.21007
transitivity.weighted._baseline                    -231.99757  150.21374
network_characteristic_path_length.weighted._baseline   6.95336   10.72909
small.worldness.weighted._baseline                  884.98725 2168.76135
global_efficiency.weighted._baseline                416.30126  312.36088
diameter_of_graph.weighted._baseline                  0.03036    0.09921
radius_of_graph.weighted._baseline                   -0.05260    0.13140
local_efficiency.weighted._baseline                  -3.33025    2.61276
assortativity_coefficient.weighted._baseline        -11.26503   36.40125
                                                     t value Pr(>|t|)
(Intercept)                                           -0.405    0.686
density_baseline                                       1.085    0.280
transitivity.weighted._baseline                       -1.544    0.125
network_characteristic_path_length.weighted._baseline  0.648    0.518
small.worldness.weighted._baseline                     0.408    0.684
global_efficiency.weighted._baseline                   1.333    0.185
diameter_of_graph.weighted._baseline                   0.306    0.760
radius_of_graph.weighted._baseline                    -0.400    0.690
local_efficiency.weighted._baseline                   -1.275    0.205
assortativity_coefficient.weighted._baseline          -0.309    0.758

(Dispersion parameter for gaussian family taken to be 853.2885)

    Null deviance: 106737  on 123  degrees of freedom
Residual deviance:  97275  on 114  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 1200.4

Number of Fisher Scoring iterations: 2
```

1.3 Calculate Correlation Matrices

Correlation Matrix for Continuous Binary Variables

In [18]:
```
myData <- mergedWINData[1:126, c(2,3,4,6,8,10,16,18)]
head(myData)

corrMatrix <- round(cor(myData, use="complete.obs"), 2)
head(corrMatrix)
```

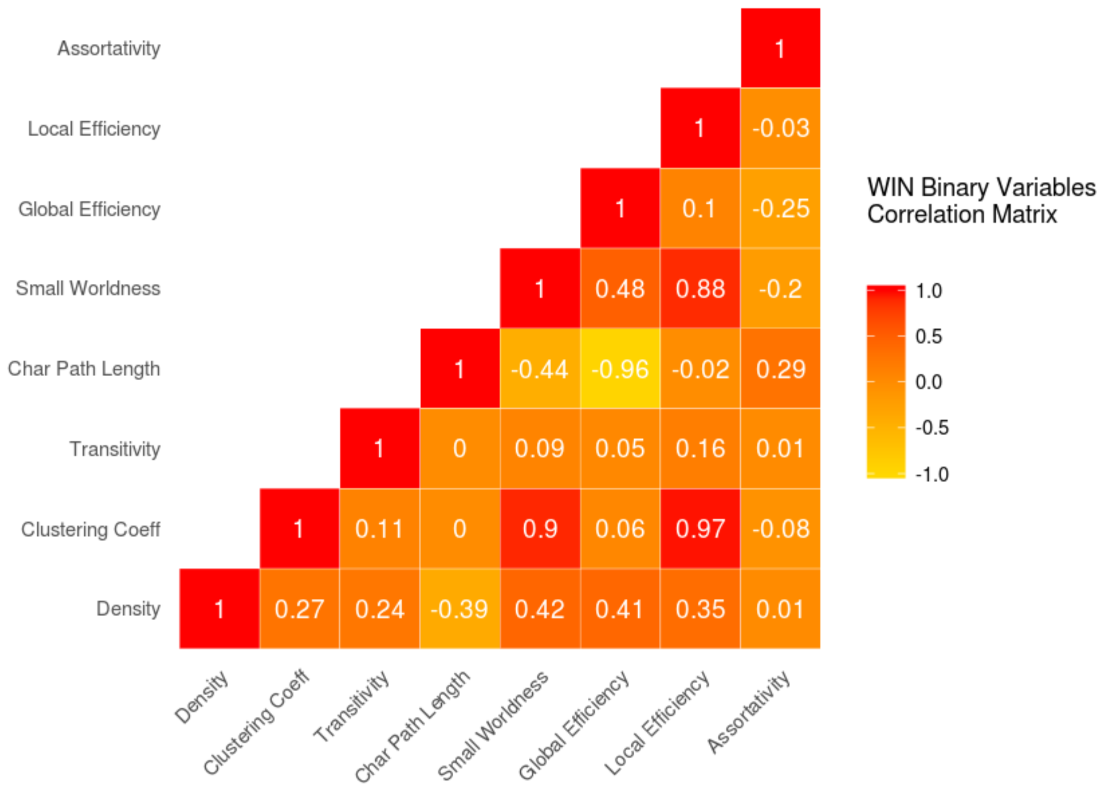
```
library(reshape2)

getLowerTri <- function(corrMatrix){
  corrMatrix[upper.tri(corrMatrix)] <- NA
  return(corrMatrix)
}
lowerTri <- getLowerTri(corrMatrix)
meltedCorrMatrix <- melt(lowerTri, na.rm = TRUE )
head(meltedCorrMatrix)

library(ggplot2)

ggplot(data = meltedCorrMatrix, p.mat = p.mat, aes(x=Var1, y=Var2, fill=value)) +

  geom_tile(color="white") +

  scale_x_discrete(labels = c("Density", "Clustering Coeff", "Transitivity", "Char Pa
                              "Small Worldness", "Global Efficiency", "Local Efficiency
                              "Assortativity")) +

  scale_y_discrete(labels = c("Density", "Clustering Coeff", "Transitivity", "Char Pa
                              "Small Worldness", "Global Efficiency", "Local Efficiency
                              "Assortativity")) +

  scale_fill_gradient(low = "gold", high = "red",
                      limit = c(-1,1), space = "Lab",
                      name = "WIN Binary Variables\nCorrelation Matrix\n\n") +
  theme_minimal() +

  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    axis.text.x = element_text(angle = 45, vjust = 1, size = 9, hjust = 1)) +

  coord_fixed() +

  geom_text(aes(Var1, Var2, label = value), color = "white", size = 4)
```

density_baseline	clustering_coeff_average.binary._baseline	transitivity.binary._baseline	network_chara
0.0502380	0.237317	0.0140703	3.72258
0.0544685	0.292529	0.0132959	3.45408
0.0565838	0.262644	0.0473149	3.45600
0.0528821	0.241766	0.0570605	4.01388
0.0560550	0.271800	0.0203477	3.42474
0.0608144	0.257770	0.0223299	3.60107

	density_baseline	clustering_coeff_average.binary._
density_baseline	1.00	0.27
clustering_coeff_average.binary._baseline	0.27	1.00
transitivity.binary._baseline	0.24	0.11
network_characteristic_path_length.binary._baseline	-0.39	0.00
small.worldness.binary._baseline	0.42	0.90
global_efficiency.binary._baseline	0.41	0.06

Var1	Var2	value
density_baseline	density_baseline	1.00
clustering_coeff_average.binary._baseline	density_baseline	0.27
transitivity.binary._baseline	density_baseline	0.24
network_characteristic_path_length.binary._baseline	density_baseline	-0.39
small.worldness.binary._baseline	density_baseline	0.42
global_efficiency.binary._baseline	density_baseline	0.41

```
Attaching package: ggplot2

The following objects are masked from package:psych:

    %+%, alpha
```



Correlation Matrix for Continuous Weighted Variables

```
In [ ]: myData <- mergedWINData[1:126, c(2,5,7,9,11,17,19)]
        head(myData)

        corrMatrix <- round(cor(myData, use="complete.obs"), 2)
        head(corrMatrix)

        library(reshape2)

        getLowerTri <- function(corrMatrix){
          corrMatrix[upper.tri(corrMatrix)] <- NA
          return(corrMatrix)
```

```
}
lowerTri <- getLowerTri(corrMatrix)
meltedCorrMatrix <- melt(lowerTri, na.rm = TRUE )
head(meltedCorrMatrix)

library(ggplot2)

ggplot(data = meltedCorrMatrix, p.mat = p.mat, aes(x=Var1, y=Var2, fill=value)) +

  geom_tile(color="white") +

  scale_x_discrete(labels = c("Density", "Transitivity", "Char Path Length",
                              "Small Worldness", "Global Efficiency", "Local Efficiency
                              "Assortativity")) +

  scale_y_discrete(labels = c("Density", "Transitivity", "Char Path Length",
                              "Small Worldness", "Global Efficiency", "Local Efficiency
                              "Assortativity")) +

  scale_fill_gradient(low = "yellow", high = "red",
                      limit = c(-1,1), space = "Lab",
                      name = "WIN Weighted Variables\nCorrelation Matrix") +
  theme_minimal() +

  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    axis.text.x = element_text(angle = 45, vjust = 1, size = 9, hjust = 1)) +

  coord_fixed() +

  geom_text(aes(Var1, Var2, label = value), color = "white", size = 4, check_overlap =
```

1.4 Run Principal Component Regression

Standardization of Variables with scale() and adding them to a new Table

In [19]:
```
baseline_p_scaled <- scale(mergedWINData$baseline_p)
baseline_q_scaled <- scale(mergedWINData$baseline_q)

density_baseline_scaled <- scale(mergedWINData$density_baseline)
clustering_coeff_average.binary._baseline_scaled <- scale(mergedWINData$clustering_co
transitivity.binary._baseline_scaled <- scale(mergedWINData$transitivity.binary._basel
network_characteristic_path_length.binary._baseline_scaled <- scale(mergedWINData$netw
```

```
small.worldness.binary._baseline_scaled <- scale(mergedWINData$small.worldness.binary
global_efficiency.binary._baseline_scaled <- scale(mergedWINData$global_efficiency.bi
local_efficiency.binary._baseline_scaled <- scale(mergedWINData$local_efficiency.bina
assortativity_coefficient.binary._baseline_scaled <- scale(mergedWINData$assortativit

transitivity.weighted._baseline_scaled <- scale(mergedWINData$transitivity.weighted._
network_characteristic_path_length.weighted._baseline_scaled <- scale(mergedWINData$n
small.worldness.weighted._baseline_scaled <- scale(mergedWINData$small.worldness.weig
global_efficiency.weighted._baseline_scaled <- scale(mergedWINData$global_efficiency.
local_efficiency.weighted._baseline_scaled <- scale(mergedWINData$local_efficiency.we
assortativity_coefficient.weighted._baseline_scaled <- scale(mergedWINData$assortativ

standardizedVariables <- data.frame(density=density_baseline_scaled,
                                    clust_binary=clustering_coeff_average.binary._bas
                                    trans_binary=transitivity.binary._baseline_scaled
                                    net_binary=network_characteristic_path_length.bin
                                    small_binary=small.worldness.binary._baseline_sca
                                    global_binary=global_efficiency.binary._baseline_
                                    local_binary=local_efficiency.binary._baseline_sc
                                    assort_binary=assortativity_coefficient.binary._b

                                    trans_weighted=transitivity.weighted._baseline_sc
                                    net_weighted=network_characteristic_path_length.w
                                    small_weighted=small.worldness.weighted._baseline
                                    global_weighted=global_efficiency.weighted._basel
                                    local_weighted=local_efficiency.weighted._baselin
                                    assort_weighted=assortativity_coefficient.weighte

                                    Pscore=baseline_p_scaled,
                                    Qscore=baseline_q_scaled)
```

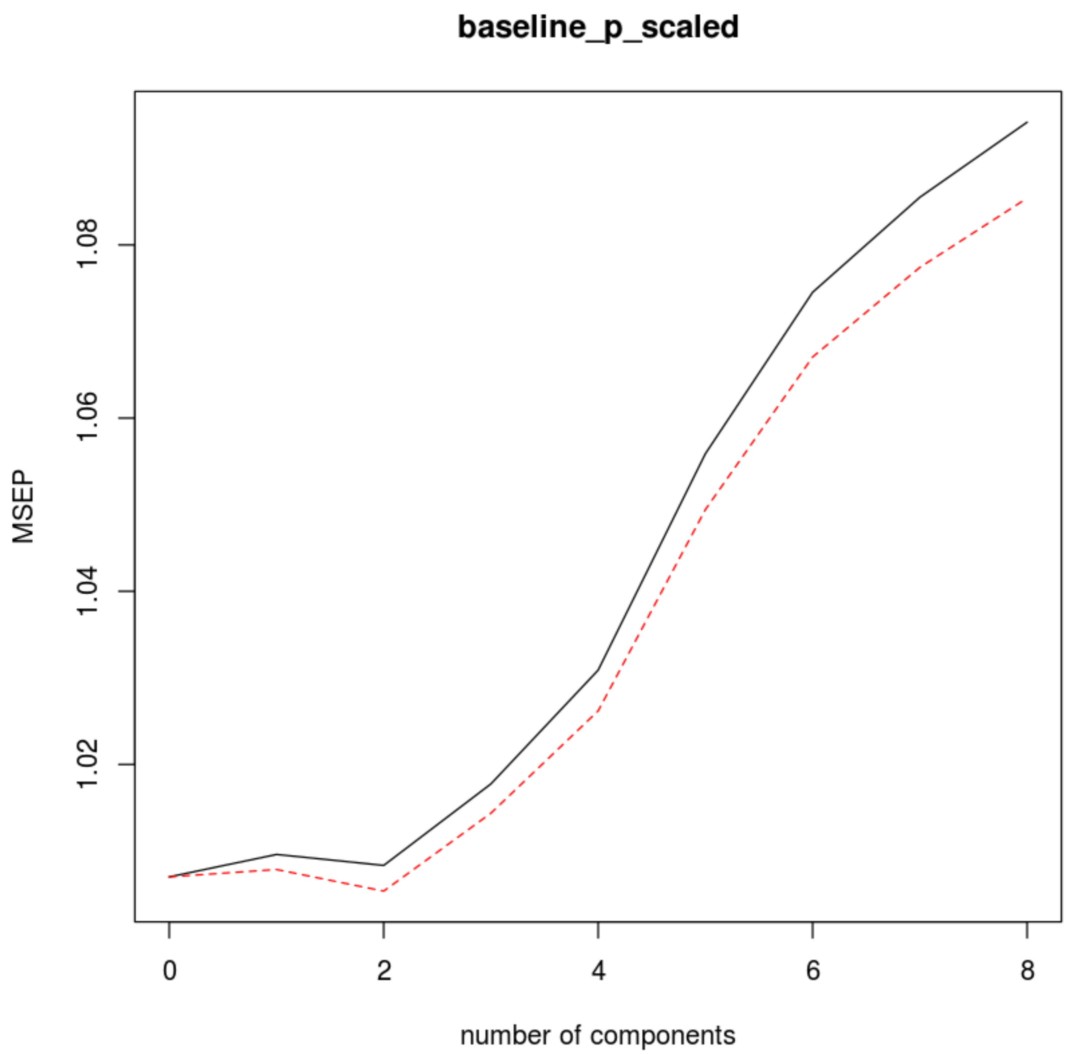
PCR Summary and Validation Curve of Continuous Binary Variables and P

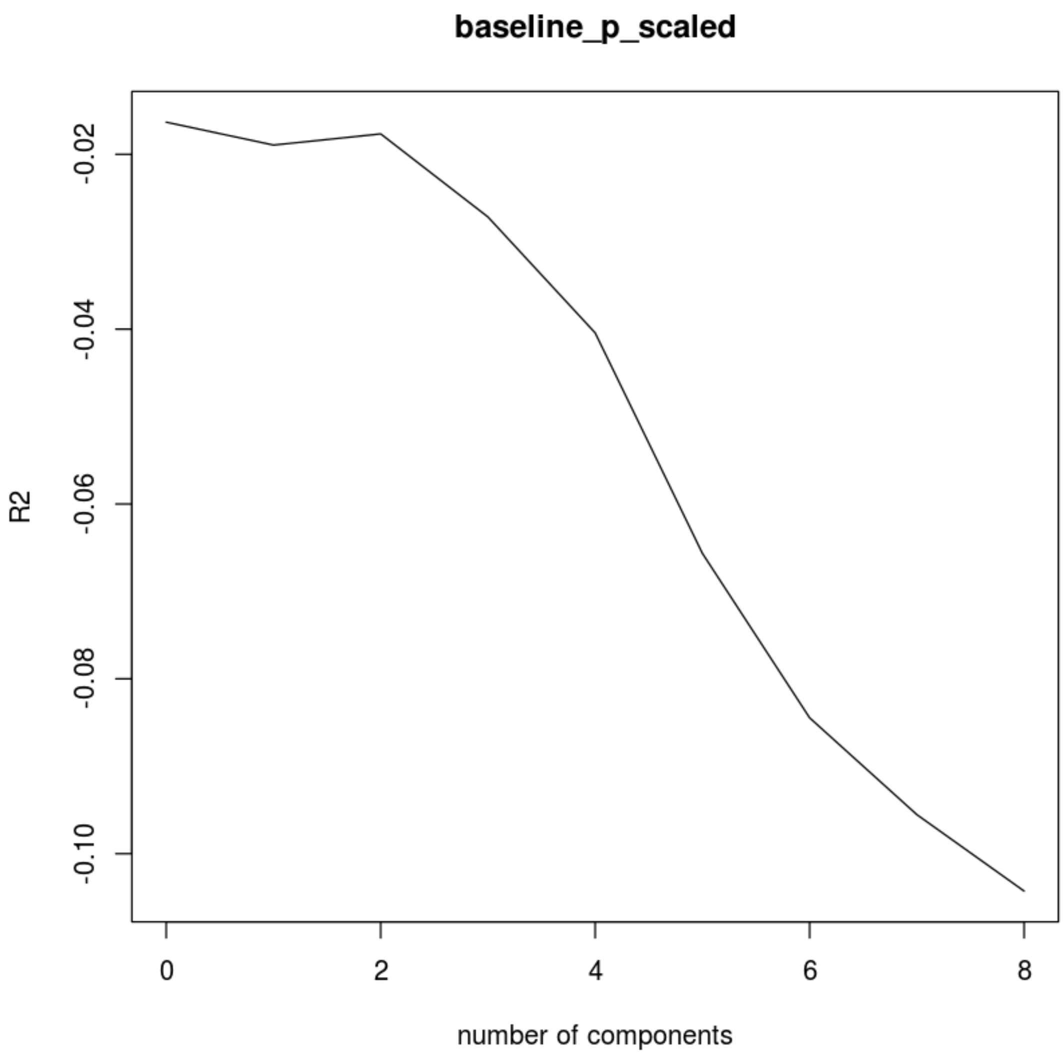
```
In [21]: library(pls)

         pcr.fit = pcr(baseline_p_scaled~
                        density_baseline_scaled+
                        clustering_coeff_average.binary._baseline_scaled+
                        transitivity.binary._baseline_scaled+
                        network_characteristic_path_length.binary._baseline_scaled+
                        small.worldness.binary._baseline_scaled+
                        global_efficiency.binary._baseline_scaled+
                        local_efficiency.binary._baseline_scaled+
                        assortativity_coefficient.binary._baseline_scaled,
                       data = mergedWINData,
                       scale = TRUE,
                       validation = "CV"
                        )
```

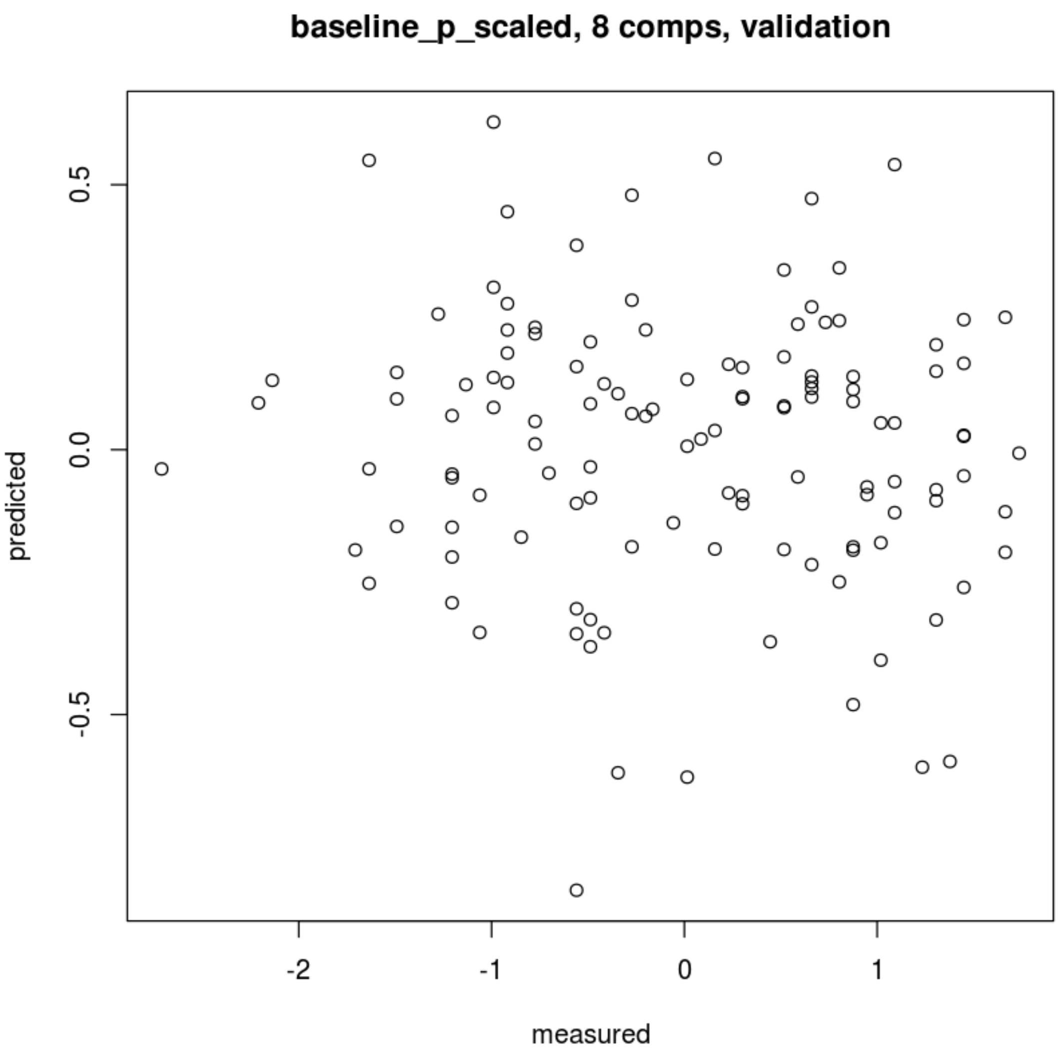
```r
summary(pcr.fit)

validationplot(pcr.fit, val.type = "MSEP")
validationplot(pcr.fit, val.type = "R2")
predplot(pcr.fit)
coefplot(pcr.fit) # plot of regression coefficients
```

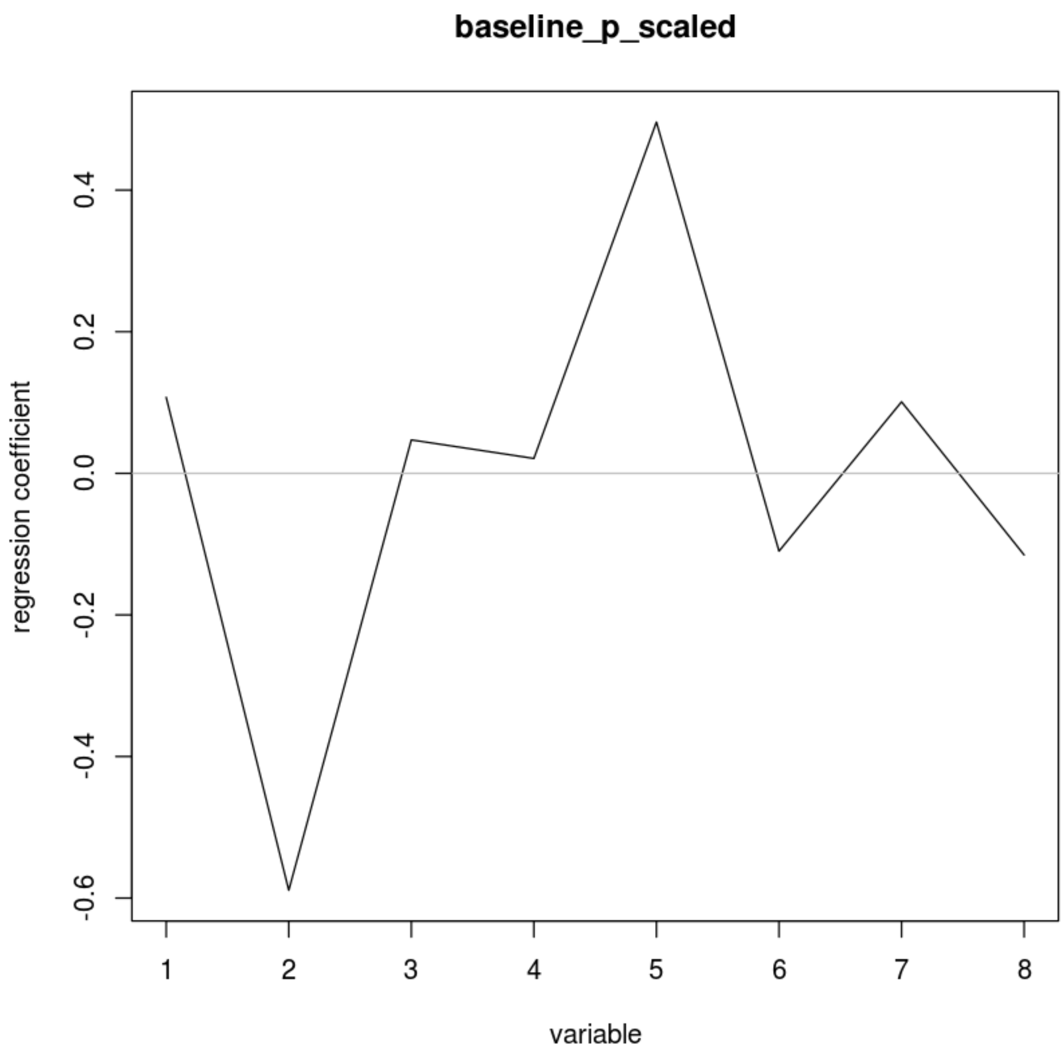
```

Attaching package: pls

The following object is masked from package:outliers:

    scores

The following object is masked from package:stats:

    loadings
```

```
Data:         X dimension: 124 8
        Y dimension: 124 1
Fit method: svdpc
Number of components considered: 8

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.004    1.005    1.004    1.009    1.015    1.028    1.037
adjCV        1.004    1.004    1.003    1.007    1.013    1.024    1.033
       7 comps  8 comps
CV       1.042    1.046
adjCV    1.038    1.042

TRAINING: % variance explained
                   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X                   42.166   67.161   81.110   92.324   99.103   99.726
baseline_p_scaled    1.296    3.735    3.876    4.527    5.189    5.193
                   7 comps  8 comps
X                   99.961  100.000
baseline_p_scaled    5.224    5.405
```


baseline_p_scaled
MSEP
1.02
1.04
1.06
1.08
0
2
4
6
8
number of components


baseline_p_scaled
R2
-0.02
-0.04
-0.06
-0.08
-0.10
0
2
4
6
8
number of components


baseline_p_scaled, 8 comps, validation
predicted
0.5
0.0
-0.5
-2
-1
0
1
measured

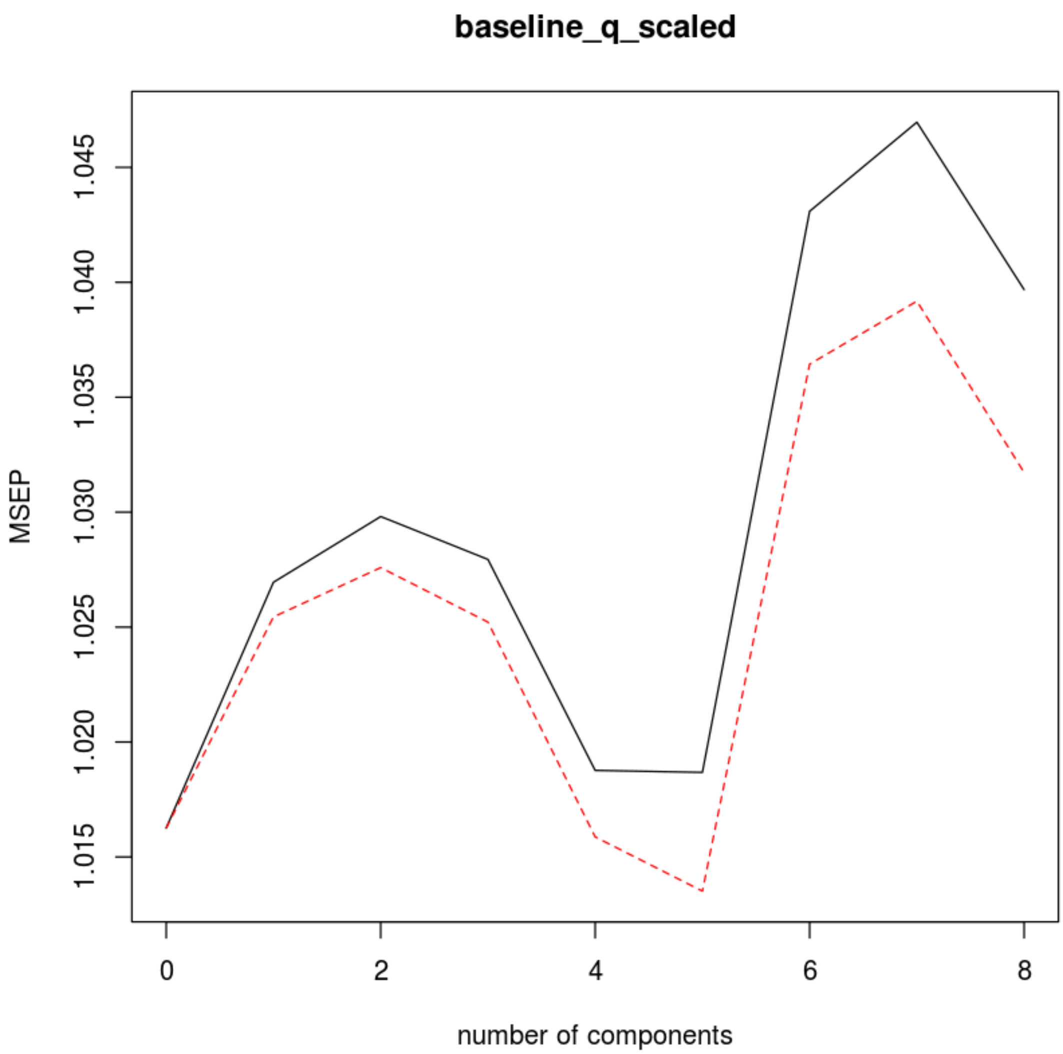


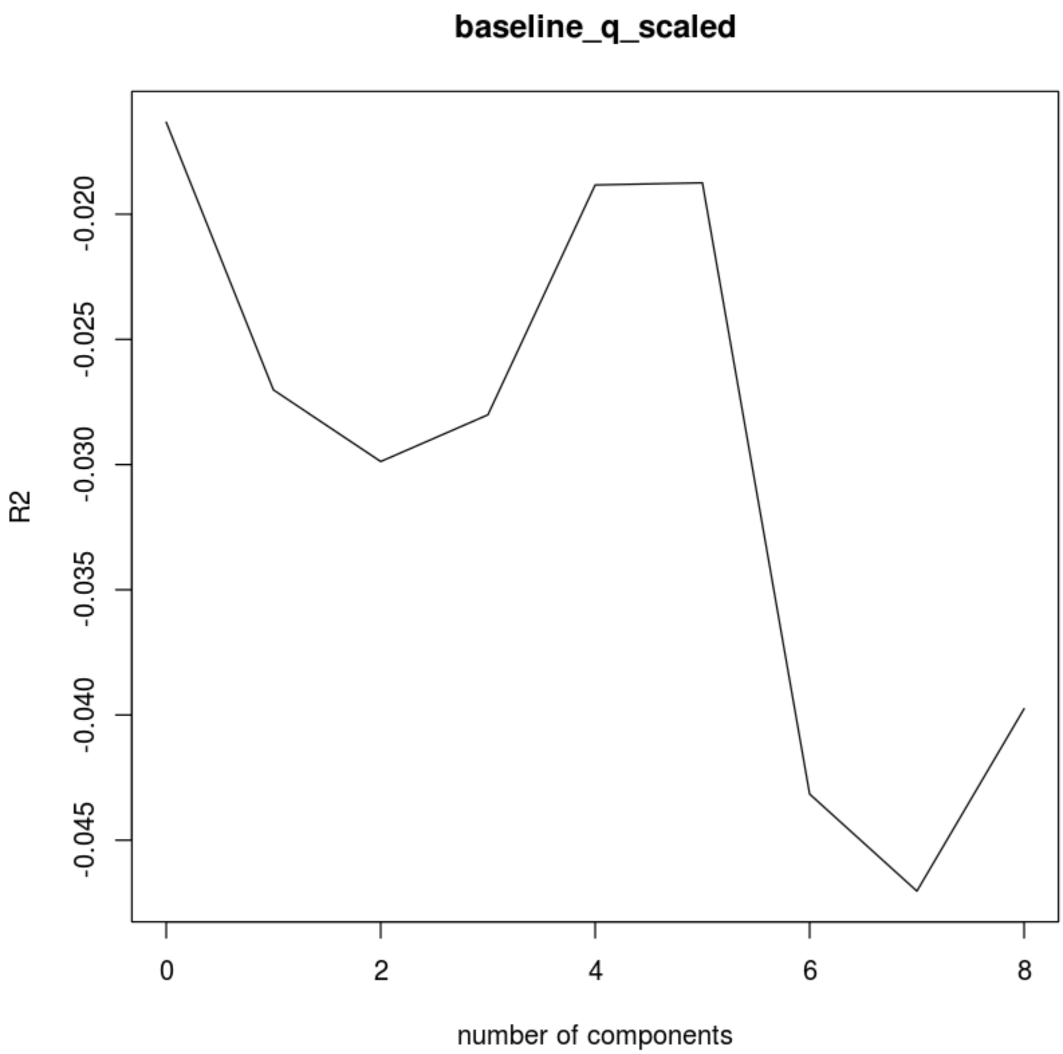
PCR Summary and Validation Curve of Continuous Binary Variables and Q

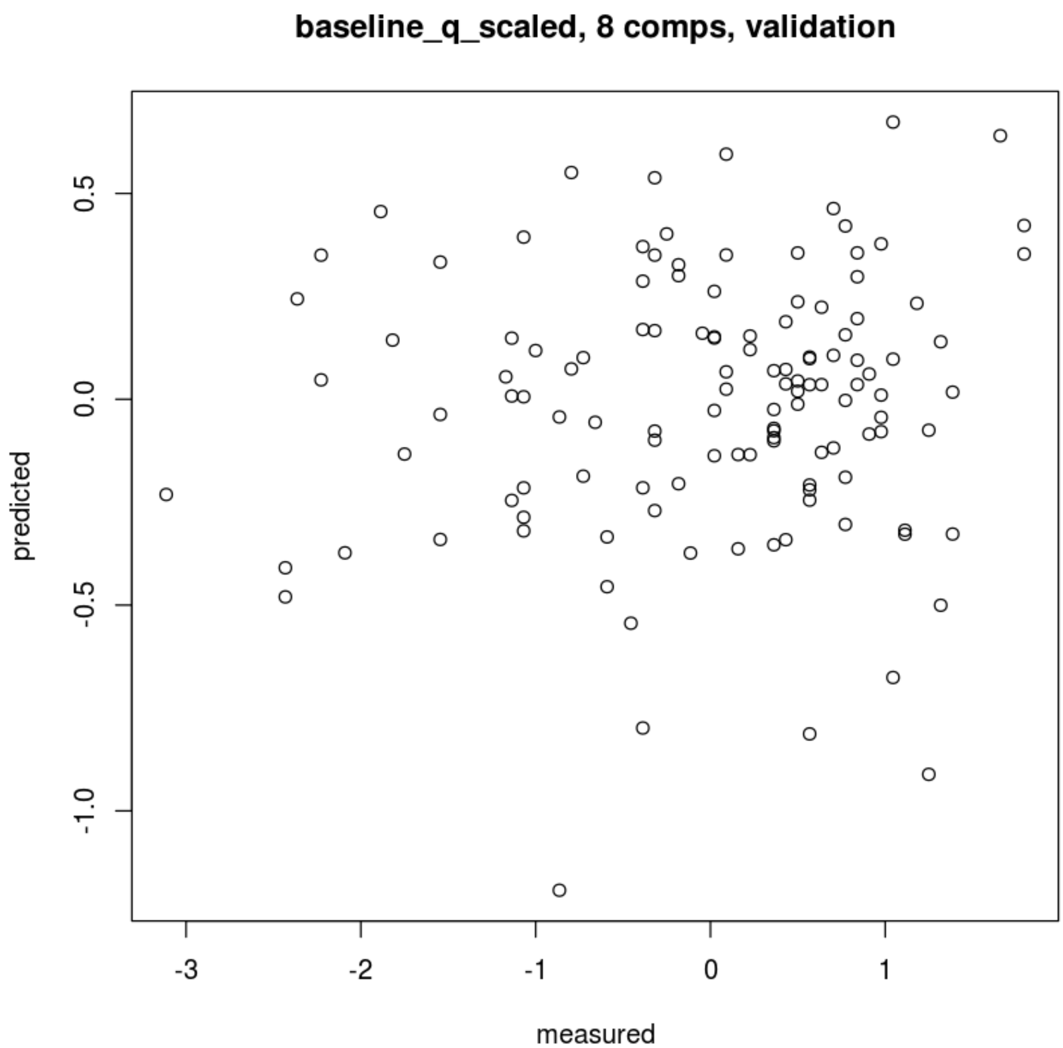
```
In [22]: library(pls)

         pcr.fit = pcr(baseline_q_scaled~
                     density_baseline_scaled+
                     clustering_coeff_average.binary._baseline_scaled+
                     transitivity.binary._baseline_scaled+
                     network_characteristic_path_length.binary._baseline_scaled+
                     small.worldness.binary._baseline_scaled+
                     global_efficiency.binary._baseline_scaled+
                     local_efficiency.binary._baseline_scaled+
                     assortativity_coefficient.binary._baseline_scaled,
```

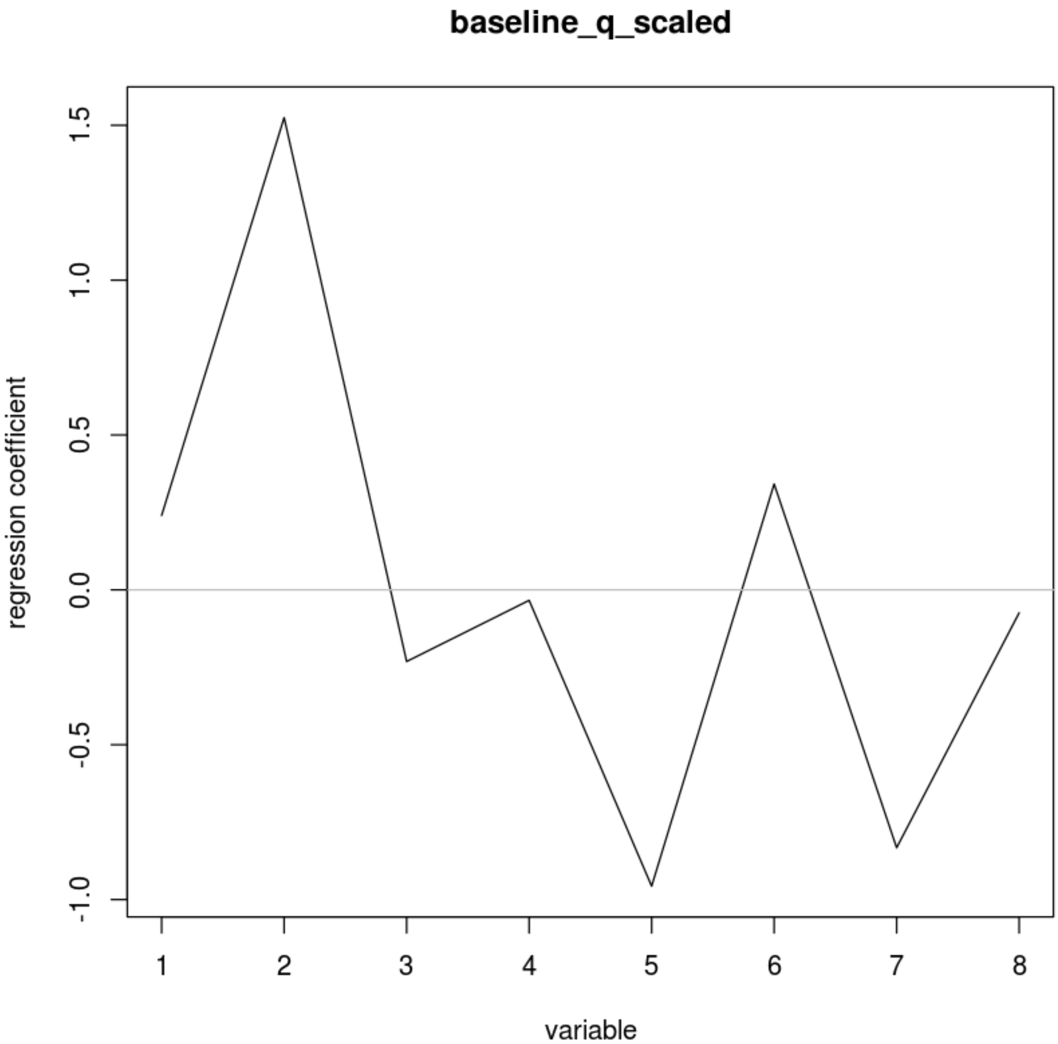
```r
                 data = mergedWINData,
                 scale = TRUE,
                 validation = "CV"
                  )

       summary(pcr.fit)

       validationplot(pcr.fit, val.type = "MSEP")
       validationplot(pcr.fit, val.type = "R2")
       predplot(pcr.fit)
       coefplot(pcr.fit)
```

```
Data:         X dimension: 124 8
        Y dimension: 124 1
Fit method: svdpc
Number of components considered: 8

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.008    1.013    1.015    1.014    1.009    1.009    1.021
adjCV        1.008    1.013    1.014    1.013    1.008    1.007    1.018
       7 comps  8 comps
CV       1.023    1.020
adjCV    1.019    1.016

TRAINING: % variance explained
                   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X                  42.1659   67.161   81.110   92.324   99.103   99.726
baseline_q_scaled   0.2339    1.409    2.983    3.864    7.483    7.688
                   7 comps  8 comps
X                   99.961   100.00
baseline_q_scaled    9.091    10.09
```


baseline_q_scaled
MSEP
1.015
1.020
1.025
1.030
1.035
1.040
1.045
0
2
4
6
8
number of components


baseline_q_scaled
R2
-0.020
-0.025
-0.030
-0.035
-0.040
-0.045
0
2
4
6
8
number of components

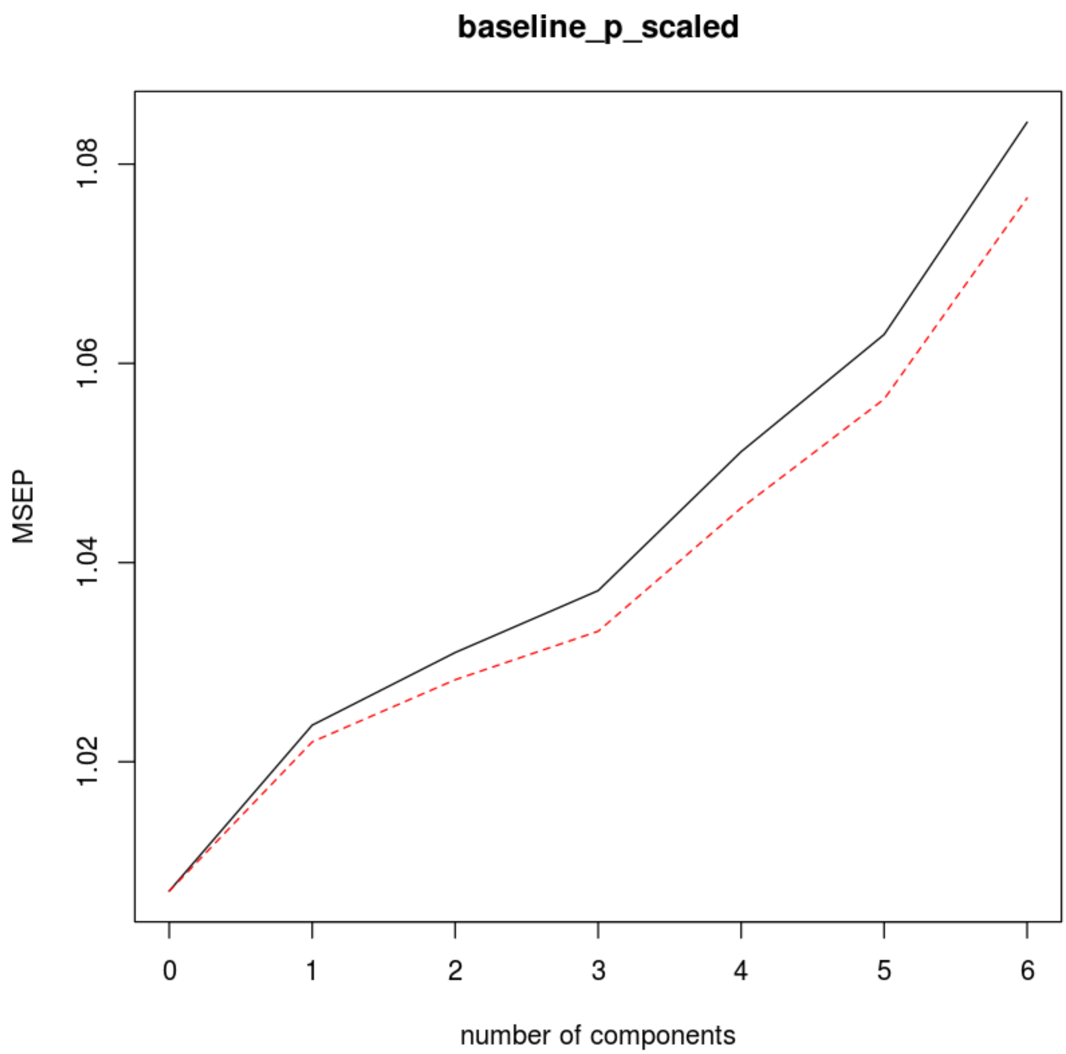

baseline_q_scaled, 8 comps, validation
predicted
measured
0.5
0.0
-0.5
-1.0
-3
-2
-1
0
1



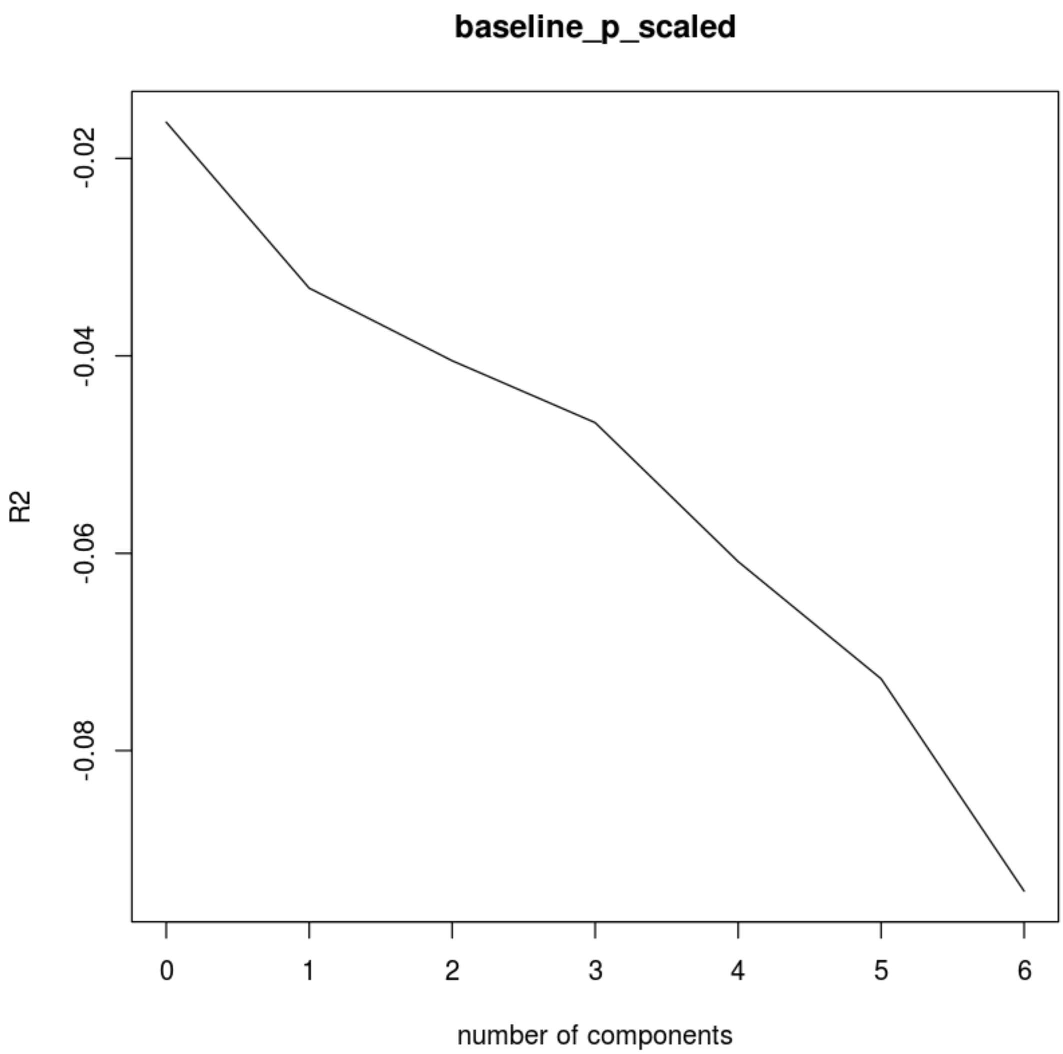
PCR Summary and Validation Curve of Continuous Weighted Variables and P

```
In [23]: library(pls)

         pcr.fit = pcr(baseline_p_scaled~
                    transitivity.weighted._baseline_scaled+
                    network_characteristic_path_length.weighted._baseline_scaled+
                    small.worldness.weighted._baseline_scaled+
                    global_efficiency.weighted._baseline_scaled+
                    local_efficiency.weighted._baseline_scaled+
                    assortativity_coefficient.weighted._baseline_scaled,
                   data = mergedWINData,
                   scale = TRUE,
```

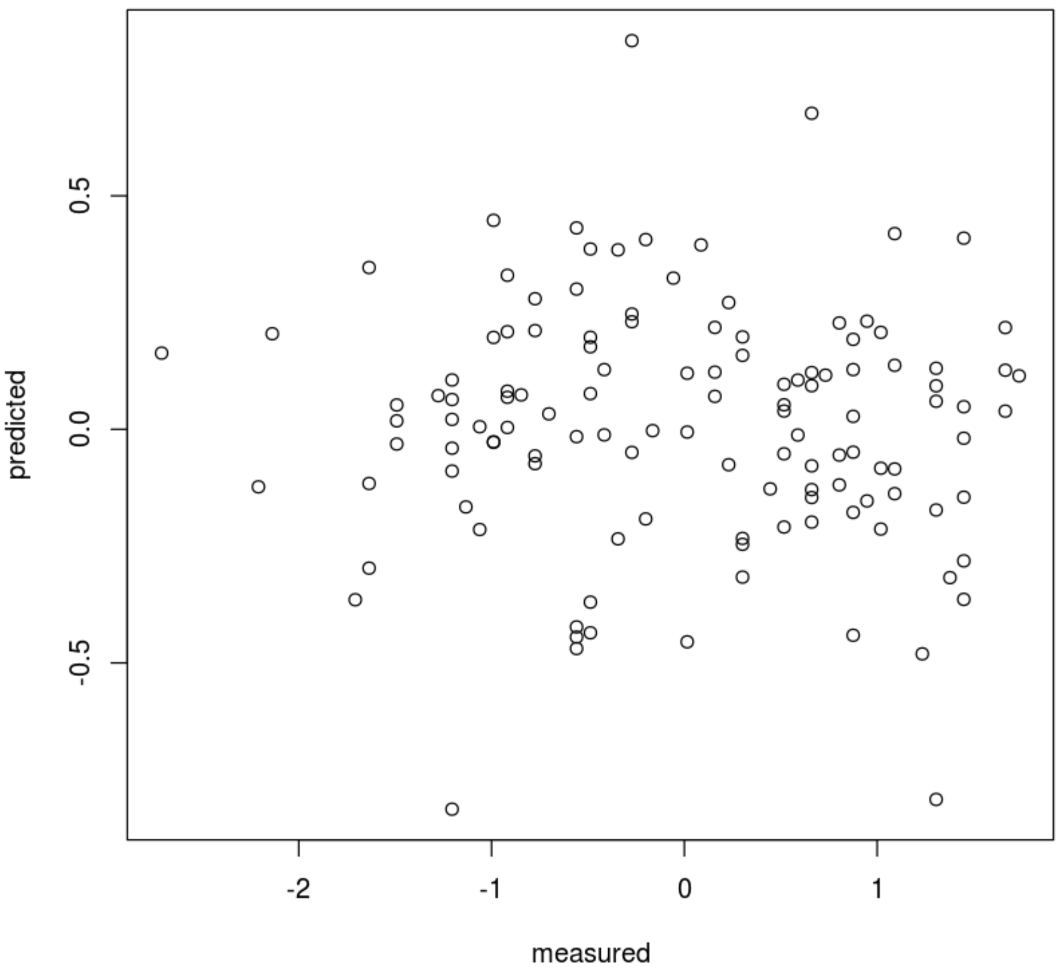
```
                   validation = "CV"
                    )

         summary(pcr.fit)

         validationplot(pcr.fit, val.type = "MSEP")
         validationplot(pcr.fit, val.type = "R2")
         predplot(pcr.fit)
         coefplot(pcr.fit)

Data:         X dimension: 124 6
        Y dimension: 124 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.004    1.012    1.015    1.018    1.025    1.031    1.041
adjCV        1.004    1.011    1.014    1.016    1.022    1.028    1.038

TRAINING: % variance explained
                   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X                  33.3602  59.7049   79.512   91.004   97.856  100.000
baseline_p_scaled   0.0992   0.9743    2.895    4.289    4.697    4.698
```

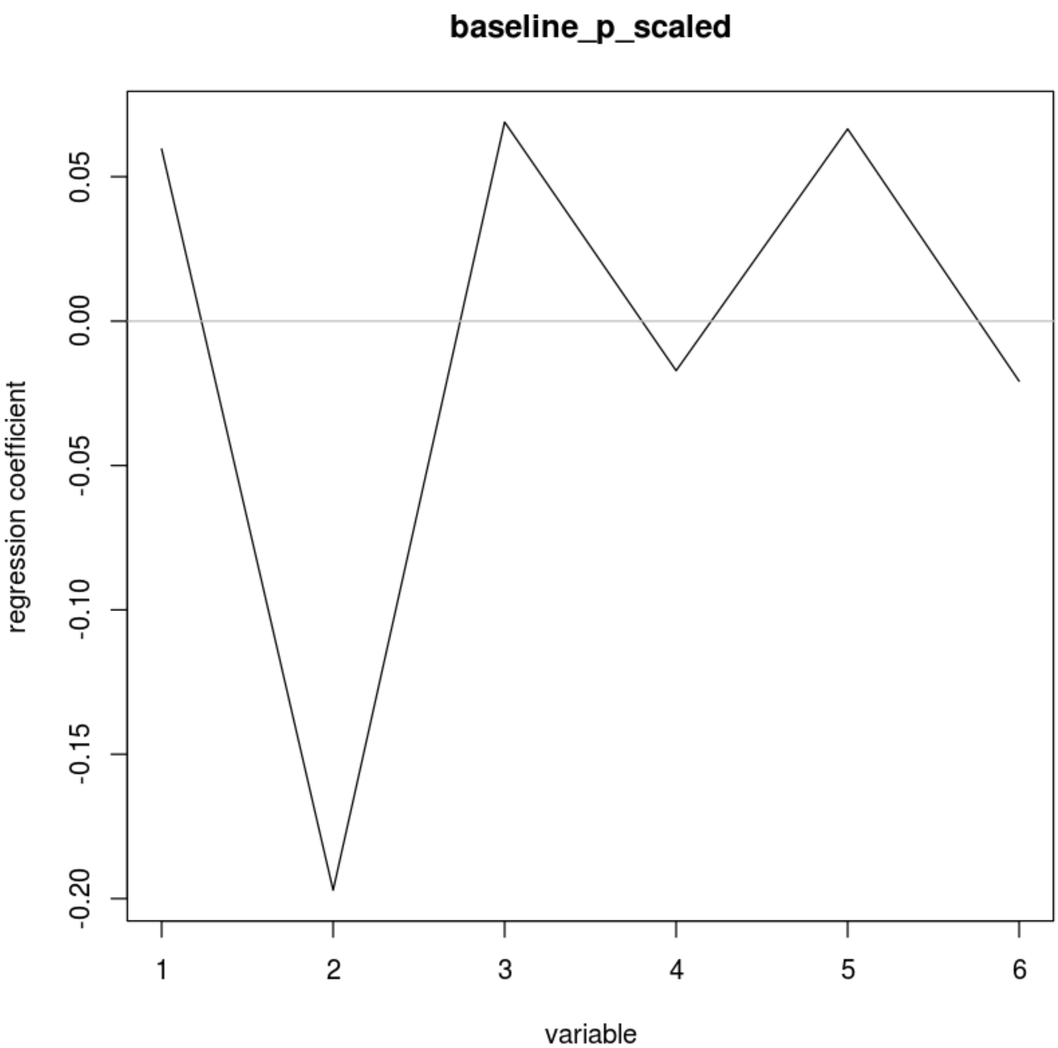


baseline_p_scaled
MSEP
1.08
1.06
1.04
1.02
0
1
2
3
4
5
6
number of components


baseline_p_scaled
-0.02
-0.04
-0.06
-0.08
R2
0
1
2
3
4
5
6
number of components

baseline_p_scaled, 6 comps, validation



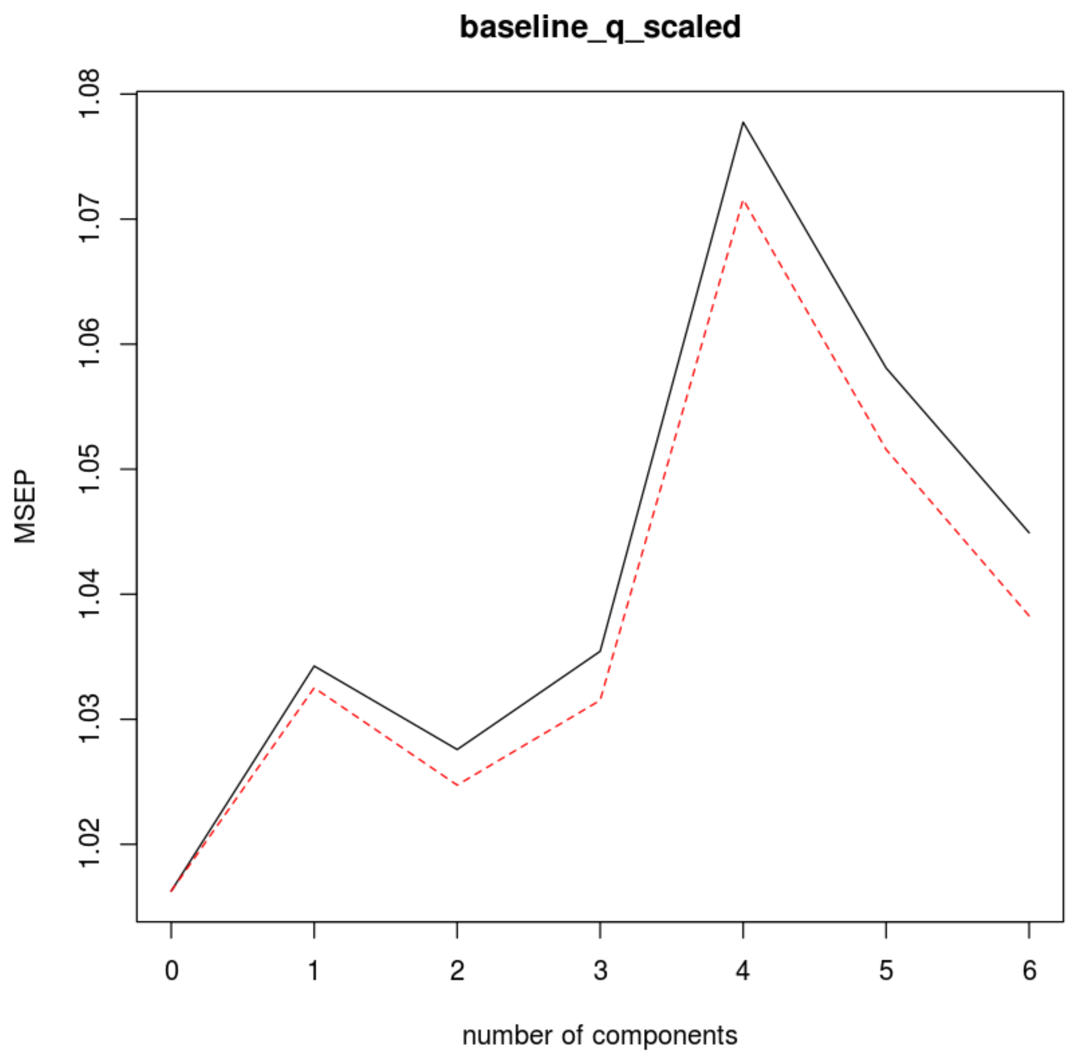


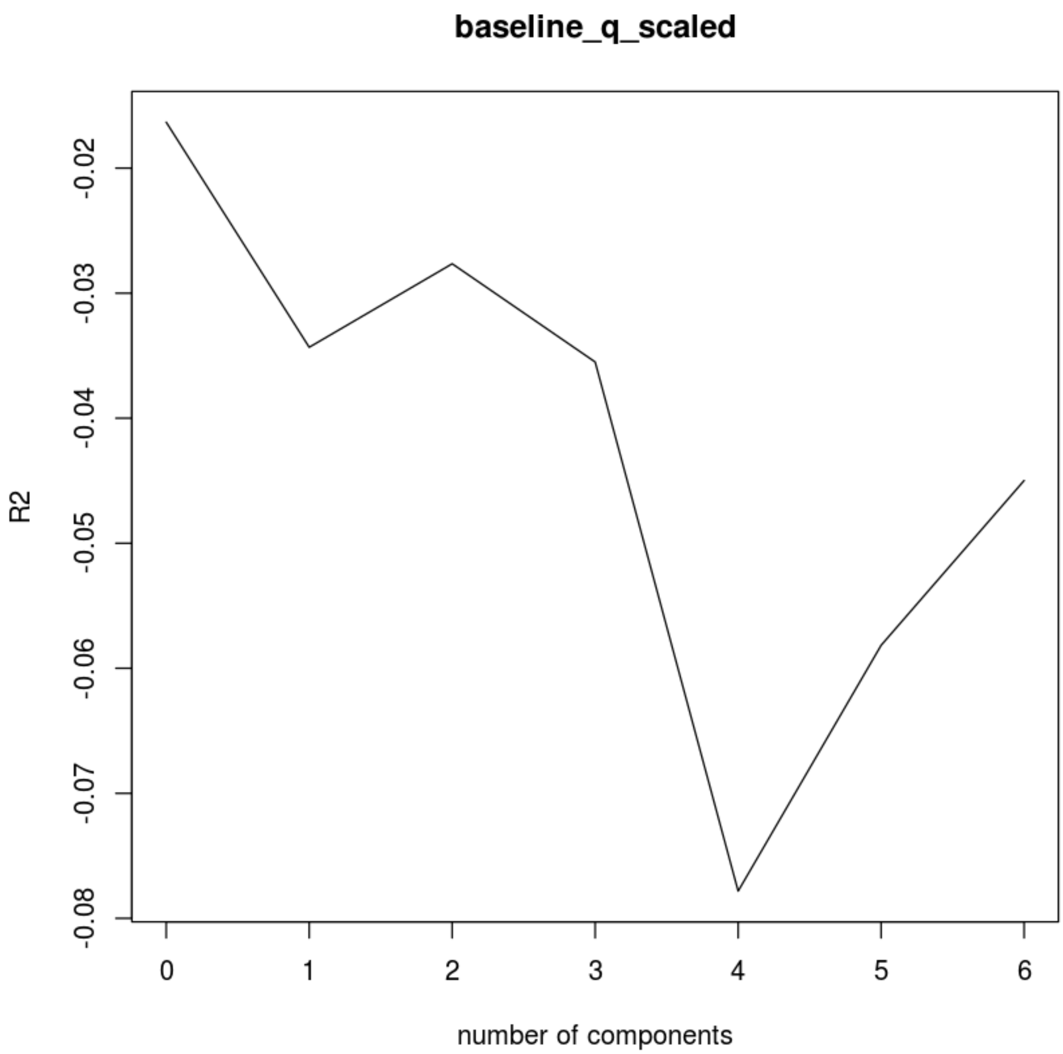
PCR Summary and Validation Curve of Continuous Weighted Variables and Q

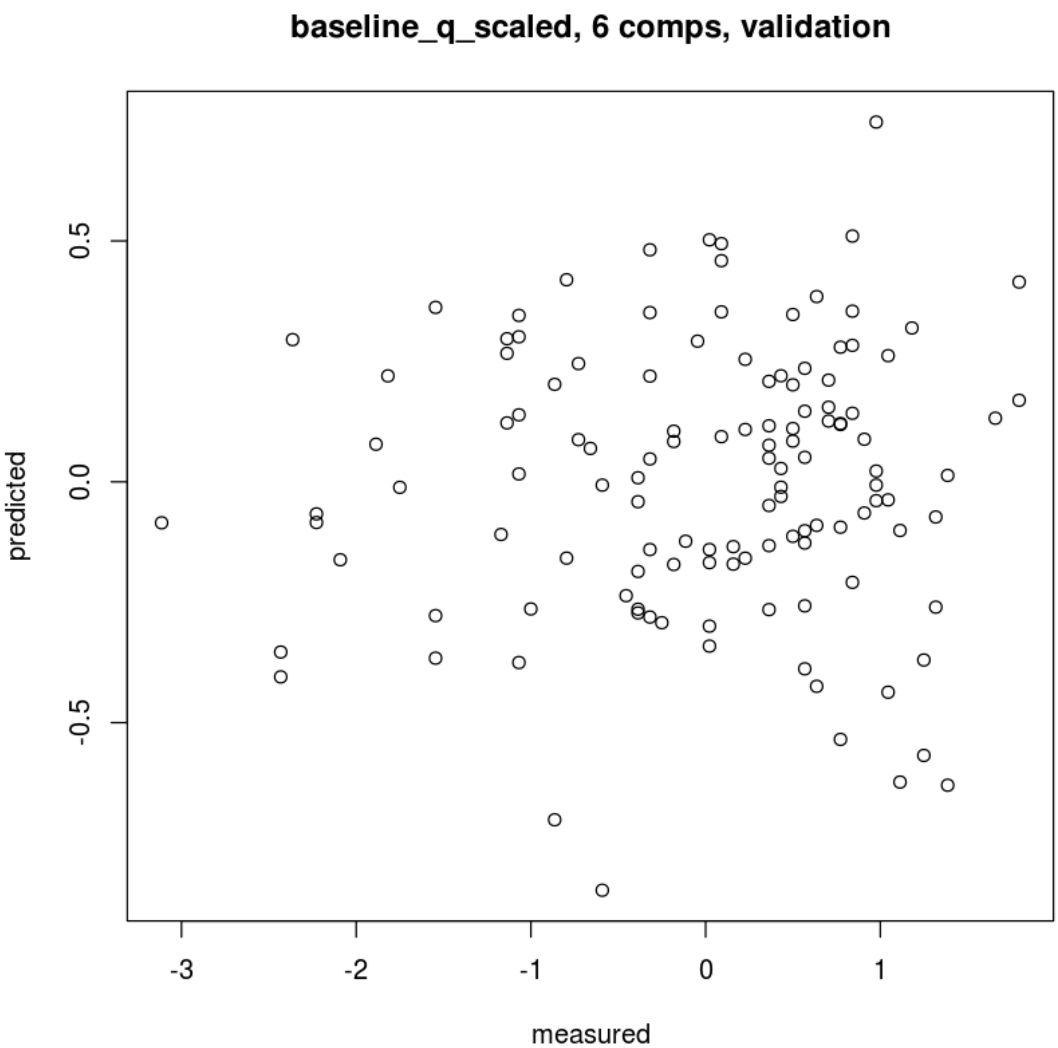
```
In [25]: library(pls)

         pcr.fit = pcr(baseline_q_scaled~
                     transitivity.weighted._baseline_scaled+
                     network_characteristic_path_length.weighted._baseline_scaled+
                     small.worldness.weighted._baseline_scaled+
                     global_efficiency.weighted._baseline_scaled+
                     local_efficiency.weighted._baseline_scaled+
                     assortativity_coefficient.weighted._baseline_scaled,
                    data = mergedWINData,
                    scale = TRUE,
```

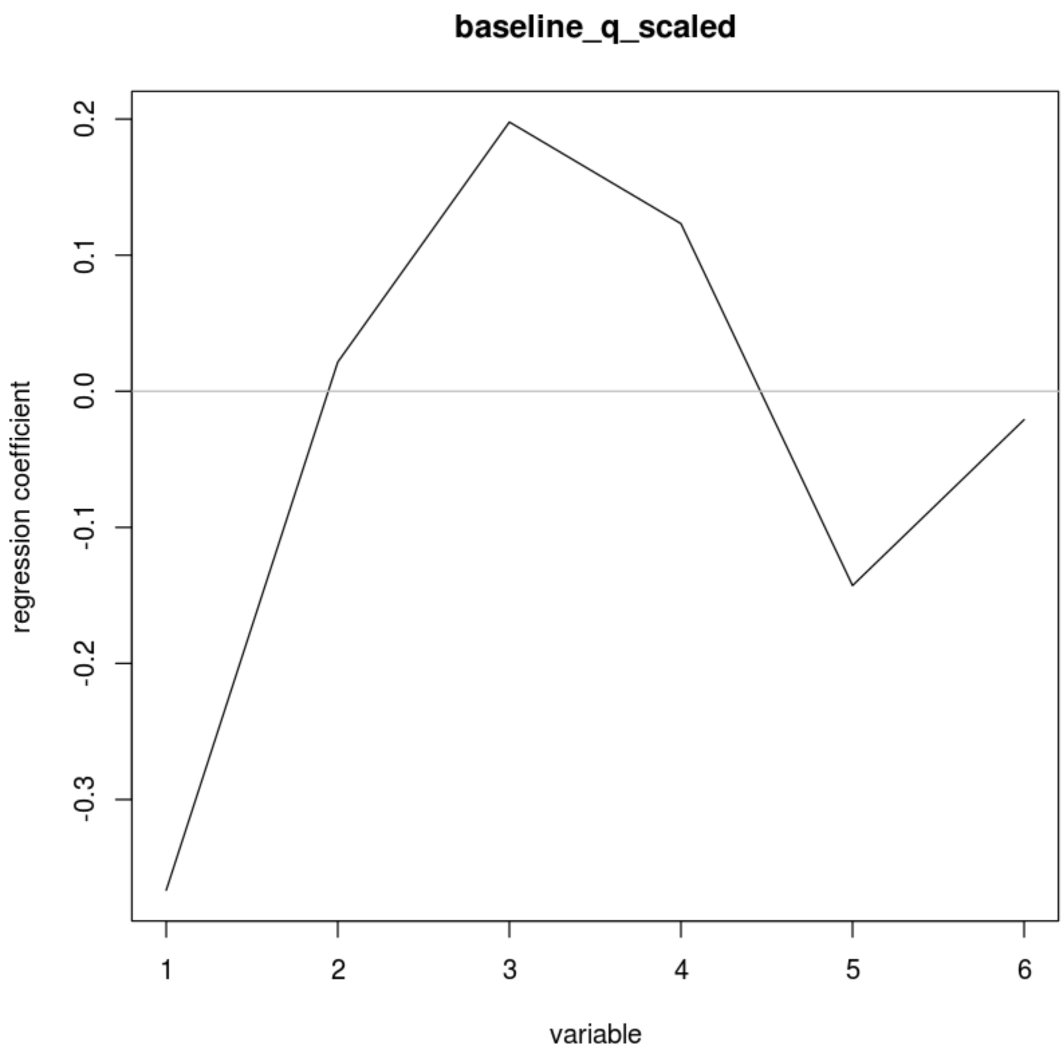
```r
                   validation = "CV"
                    )

         summary(pcr.fit)

         validationplot(pcr.fit, val.type = "MSEP")
         validationplot(pcr.fit, val.type = "R2")
         predplot(pcr.fit)
         coefplot(pcr.fit)
```

```
Data:         X dimension: 124 6
        Y dimension: 124 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.008    1.017    1.014    1.018    1.038    1.029    1.022
adjCV        1.008    1.016    1.012    1.016    1.035    1.025    1.019

TRAINING: % variance explained
                   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X                  33.3602   59.705   79.512   91.004   97.856  100.000
baseline_q_scaled   0.7147    2.833    3.767    3.774    6.292    7.864
```


baseline_q_scaled
MSEP
1.02
1.03
1.04
1.05
1.06
1.07
1.08
0
1
2
3
4
5
6
number of components


baseline_q_scaled
R2
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
-0.08
0
1
2
3
4
5
6
number of components


baseline_q_scaled, 6 comps, validation
predicted
measured
0.5
0.0
-0.5
-3
-2
-1
0
1



1.5 Run Principal Component Analysis

Principal Component Analysis Summary, Screeplot, and Biplot of Binary Variables

```
In [26]: myData <- standardizedVariables[,c(1,2,3,4,5,6,7,8)]

         myPCA = princomp(na.omit(myData),
                          cor = TRUE,
                          scores = TRUE)

         summary(myPCA)

         myPCA$loadings
```

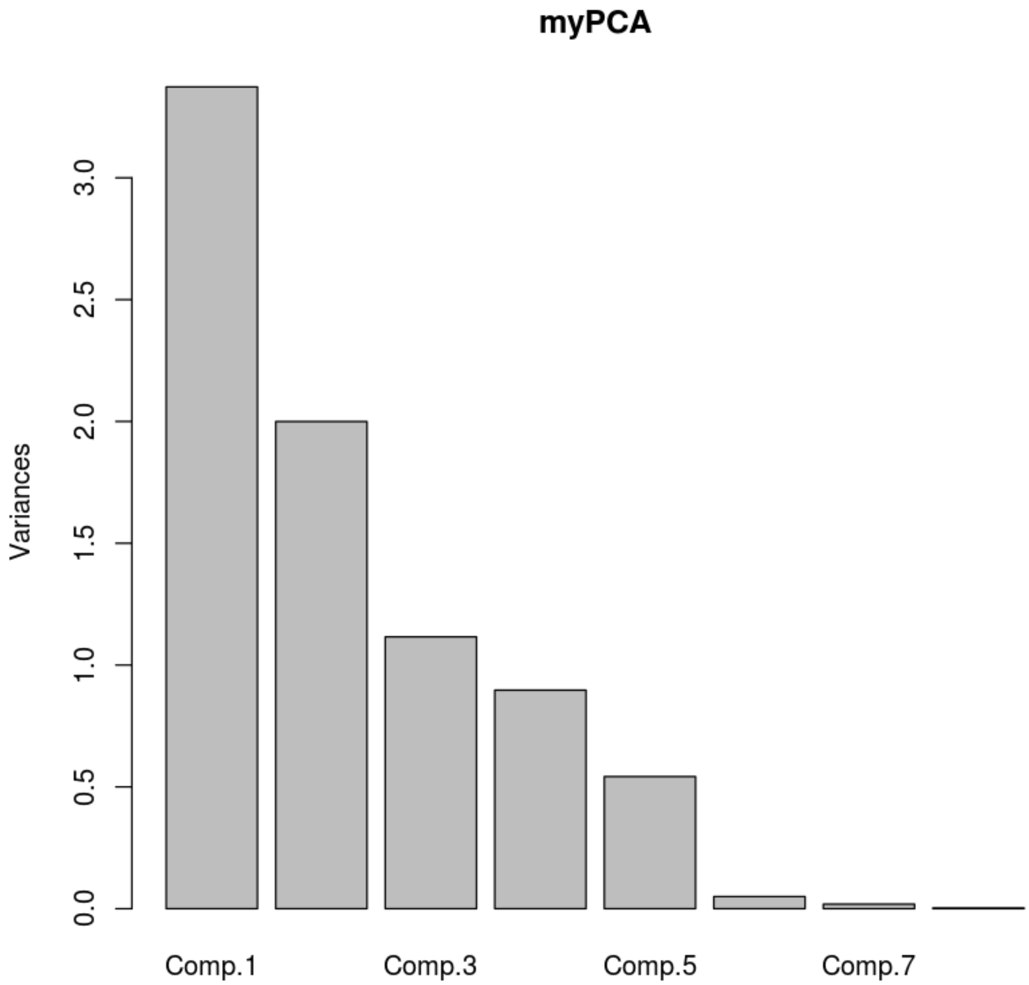
```r
plot(myPCA)
```

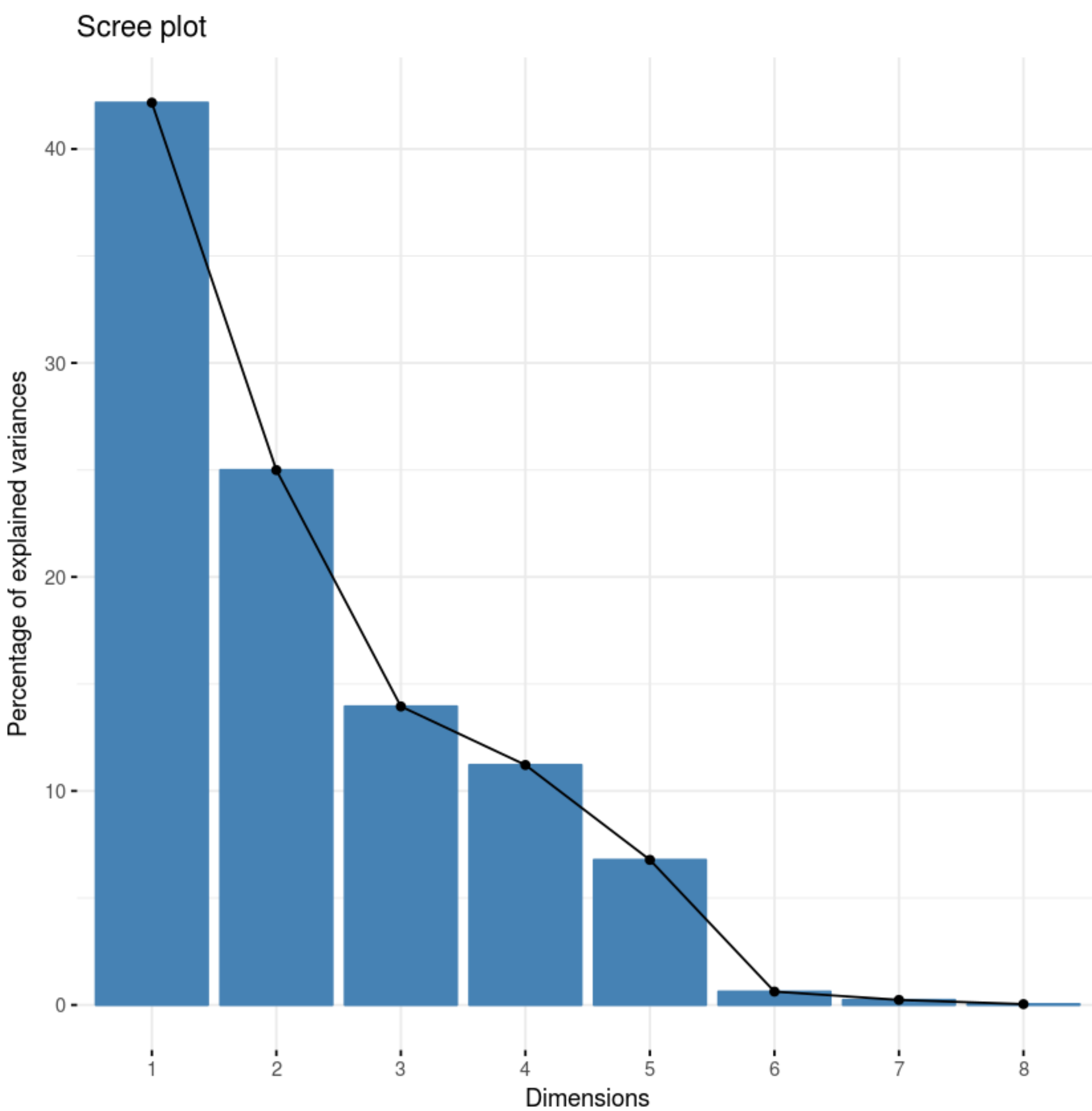
```r
library("factoextra")
fviz_screeplot(myPCA, ncp=10)
fviz_pca_biplot(myPCA) + theme_minimal()
```

```
Importance of components:
                         Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.836647 1.4140841 1.0563707 0.9471558 0.73641242
Proportion of Variance 0.421659 0.2499542 0.1394899 0.1121380 0.06778791
Cumulative Proportion  0.421659 0.6716133 0.8111032 0.9232412 0.99102908
                            Comp.6     Comp.7       Comp.8
Standard deviation     0.223352263 0.13709880 0.0555430344
Proportion of Variance 0.006235779 0.00234951 0.0003856286
Cumulative Proportion  0.997264861 0.99961437 1.0000000000
```

```
Loadings:
              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
density       -0.324  0.129 -0.473 -0.167  0.789
clust_binary  -0.435 -0.405  0.148               -0.212 -0.336  0.680
trans_binary  -0.111        -0.702  0.618 -0.327
net_binary     0.302 -0.565         0.111  0.209  0.527 -0.469 -0.187
small_binary  -0.525 -0.114  0.146        -0.135 -0.285 -0.322 -0.696
global_binary -0.330  0.529        -0.122 -0.269  0.633 -0.333  0.115
local_binary  -0.447 -0.389                       0.436  0.670
assort_binary  0.132 -0.229 -0.487 -0.748 -0.361

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```


myPCA
Variances
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Comp.1
Comp.3
Comp.5
Comp.7


Scree plot
Percentage of explained variances
40
30
20
10
0
1
2
3
4
5
6
7
8
Dimensions

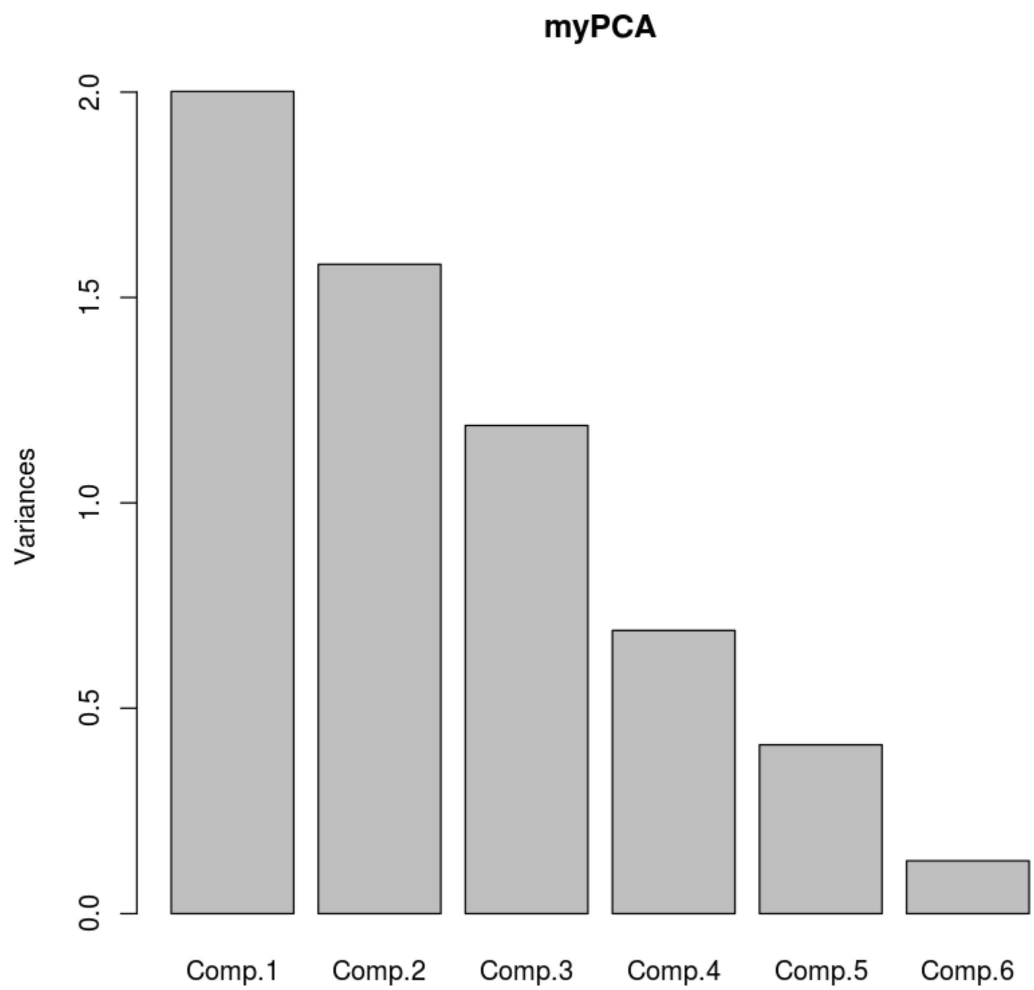
Principal Component Analysis Summary, Screeplot, and Biplot of Weighted Variables

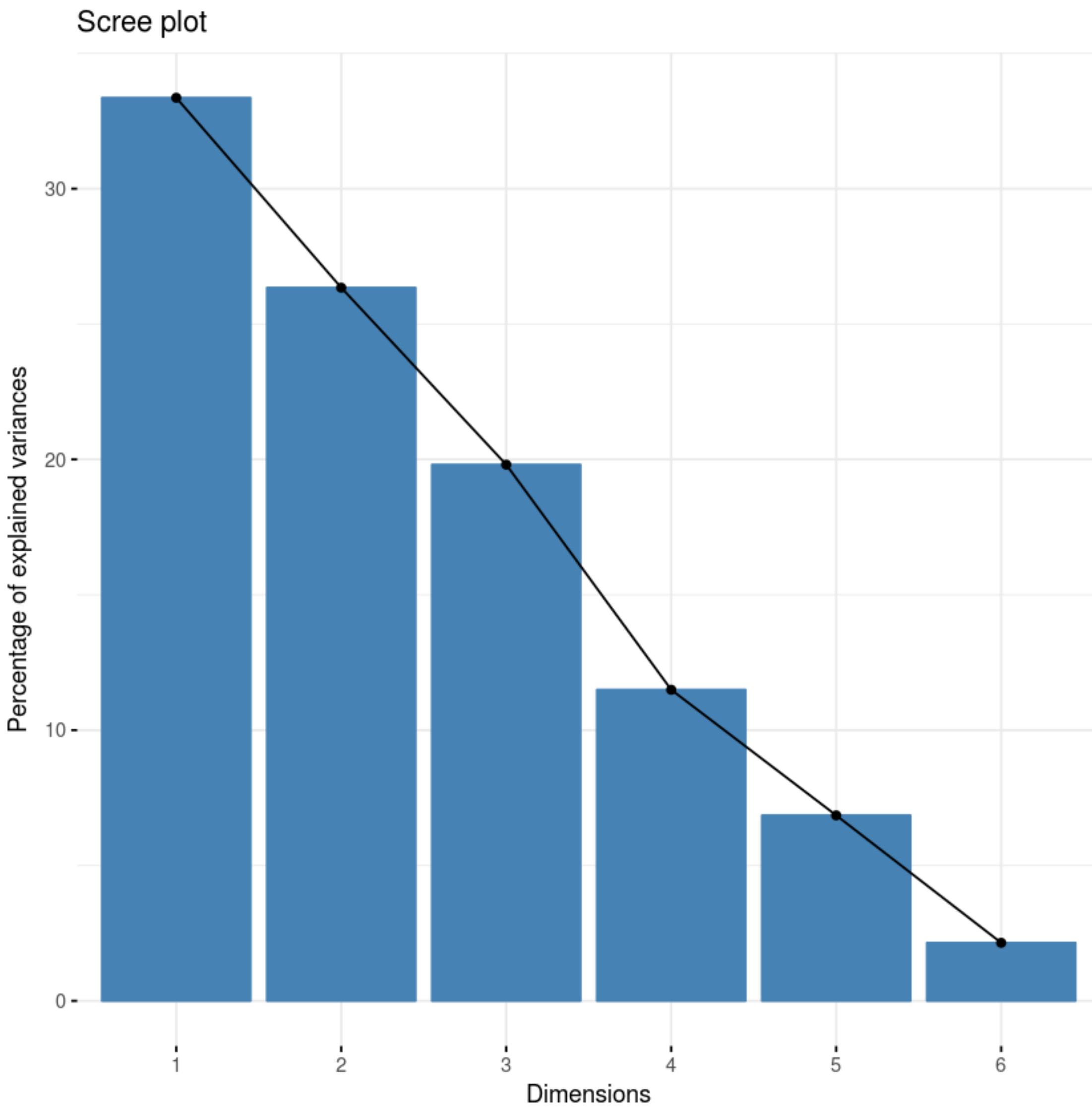
```
In [27]: myData <- standardizedVariables[, c(9,10,11,12,13,14)]

         myPCA = princomp(na.omit(myData),
                          cor = TRUE,
                          scores = TRUE)

         summary(myPCA)
         plot(myPCA)
         myPCA$loadings

         library("factoextra")
```

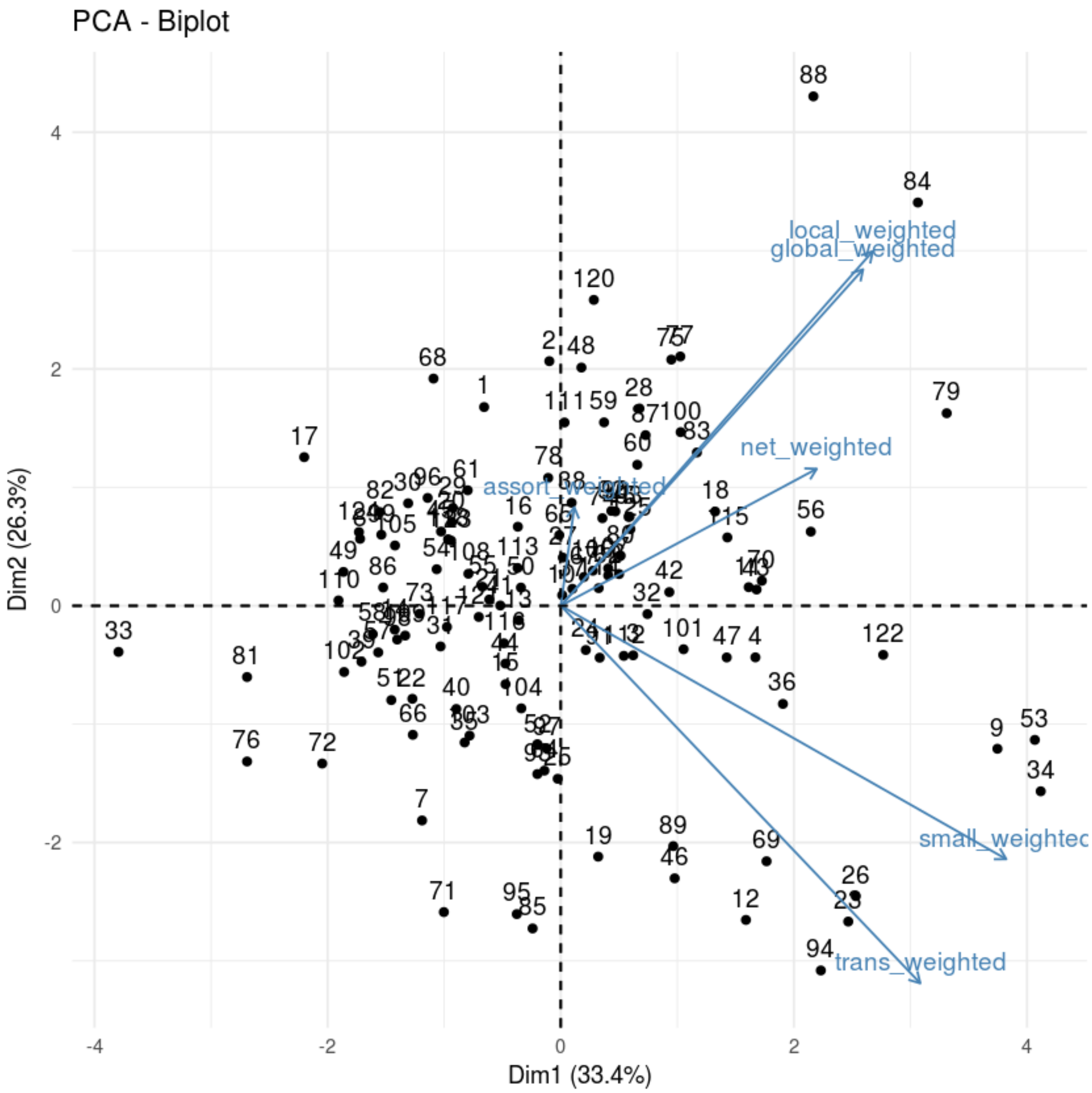
```r
fviz_screeplot(myPCA, ncp=10)
fviz_pca_biplot(myPCA) + theme_minimal()
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.4147835 1.2572524 1.0901543 0.8303591 0.64117730
Proportion of Variance 0.3336021 0.2634473 0.1980727 0.1149160 0.06851806
Cumulative Proportion  0.3336021 0.5970494 0.7951221 0.9100381 0.97855618
                           Comp.6
Standard deviation     0.35869613
Proportion of Variance 0.02144382
Cumulative Proportion  1.00000000
```

```
Loadings:
                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
trans_weighted   0.471 -0.548  0.111         0.123  0.669
net_weighted     0.335  0.199  0.531 -0.724 -0.192
small_weighted   0.584 -0.368 -0.136  0.160        -0.687
global_weighted  0.396  0.489 -0.284  0.249 -0.626  0.265
local_weighted   0.409  0.515 -0.128         0.741
assort_weighted         0.143  0.768  0.620

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.167  0.167  0.167  0.167  0.167  0.167
Cumulative Var  0.167  0.333  0.500  0.667  0.833  1.000
```


myPCA
Variances
0.0
0.5
1.0
1.5
2.0
Comp.1
Comp.2
Comp.3
Comp.4
Comp.5
Comp.6


Scree plot
Percentage of explained variances
30
20
10
0
1
2
3
4
5
6
Dimensions



1.6 Run Singular Value Decomposition

Singular Value Decomposition of Binary Variables

X = U S V

(1 x 8) = (124 x 5) ( ) (5 x 5)

8: Number of variables

124: Number of participants

5: Number of Principle components

```
In [28]: myData <- standardizedVariables[,c(1,2,3,4,5,6,7,8)]

         svd(na.omit(myData), nu = min(124,5), nv = min(5,5)) #SVD
```

$d 1. 20.3694008103758 2. 15.6829516301538 3. 11.7157182715922 4. 10.5044657457933 5. 8.1672088565498 6. 2.47709642513924 7. 1.52049920622528 8. 0.616002050789884

-0.0994089124	0.053465120	0.054860672	-0.109731261	0.041835496
0.0619585805	0.035140230	0.143895307	-0.021701642	-0.037713437
0.0288116952	-0.017043373	0.025681305	0.108547705	-0.060716273
-0.0763112491	0.059148936	0.022064745	0.216492619	-0.088895947
0.0394609040	0.009297452	0.064671853	-0.061150434	-0.029083444
0.0139027273	0.013592753	0.005125472	-0.053222136	-0.173277848
0.0226466923	-0.048451233	-0.011390675	0.064864805	0.042807450
0.0564585643	-0.025457814	-0.030843749	-0.122855129	-0.013171015
-0.0467211155	0.031070577	-0.018689073	0.137957723	-0.123727579
0.0492339221	-0.065008283	0.068875550	-0.021417338	0.009911884
-0.1288014695	0.050874957	0.043002446	0.014714074	0.024758704
0.0958214064	0.084428918	-0.161448777	-0.014440156	0.059942528
0.1873367470	0.044362085	0.002534511	0.081099342	-0.059843321
-0.2490699065	-0.145292226	0.079222131	-0.009065405	-0.005671987
-0.0003538589	-0.032710463	-0.124527800	-0.133898117	0.053481429
-0.0255889240	0.061366630	0.026438251	-0.045301611	-0.048885680
0.0211413308	-0.066226484	0.015428116	-0.136312066	-0.126741359
-0.0594297156	0.060475642	-0.082863798	-0.035051618	-0.009500211
-0.0997745098	0.005392296	-0.224645478	0.081508467	0.040501187
0.0199170831	-0.004436493	-0.055476928	0.036626727	-0.005309503
-0.0373218013	0.274024695	-0.030319128	0.030483863	0.105183618
-0.0411489865	-0.018825414	0.054213481	0.029612186	-0.007649369
0.1351360425	0.007136389	0.010170175	0.080048730	-0.048657846
0.1100330663	0.053234652	-0.034773191	0.061607135	0.027074704
0.0921270263	-0.007189659	-0.087186867	0.047942963	-0.089777893
-0.0387718260	-0.005735568	-0.076688403	0.010814848	-0.054989654
-0.0169740910	0.002568516	0.120424379	-0.021727672	0.042798019
-0.0730190092	0.054108966	0.100980692	-0.064787347	0.292814569
0.0692146329	-0.159993647	0.102912157	0.051137460	-0.155491631
0.0697199142	-0.103244003	0.055198972	-0.091789271	0.369341150

$u

0.0909038990	-0.09232218	-0.0817000363	0.075101880	-0.059226719
-0.1679578602	0.05117818	-0.0104745111	-0.150521143	-0.016780538
0.0473423773	-0.03434864	-0.0919532990	-0.017767264	0.024846207
0.0549960923	0.16677508	-0.0154829107	-0.123223134	0.079655885
-0.0072374657	-0.09280809	-0.0020373026	0.125062252	0.053754431
-0.0132191165	0.09565110	0.1327763748	0.081434866	-0.078140097
0.1338067521	-0.10900720	0.0251646937	0.032217563	0.003243273
-0.0369333847	0.01890233	0.0007851351	0.064257450	0.113906871
-0.0382457895	-0.13345430	0.0711131610	0.041815109	-0.057558444
-0.0109384878	-0.04412667	-0.0884466406	-0.200582324	-0.022223643
-0.0198879908	-0.07707460	0.1310429348	0.037648540	-0.034380482
0.0067205162	0.04541053	0.0936463582	-0.083943694	0.057543662
-0.0681445952	-0.04503583	-0.0117761642	-0.050259511	-0.003622739
0.0931813684	-0.02930038	0.0330080322	-0.018593995	0.018208991
-0.0005510946	-0.02933657	-0.0152402190	0.139472314	0.194181424
0.0952935968	-0.05865948	0.0556138063	-0.032726000	0.031356879
0.0272108152	0.12005347	0.0378228222	-0.079887927	0.083239453
-0.0708578354	-0.08421104	0.0863113050	0.055816883	-0.085647017
0.0713620732	-0.08624675	0.0897521265	0.054748157	0.034302738
-0.0441897734	0.01964209	-0.0973450647	-0.009060939	0.081859466
-0.0728779731	-0.02829310	-0.1398192860	0.043115253	0.107610739
-0.0473601065	-0.12404083	-0.0963380695	0.007931355	-0.047594588

$v

0.3236739	-0.12929184	-0.47334415	-0.1674207696	-0.78939630
0.4354089	0.40521415	0.14751415	0.0008057525	0.06200694
0.1114003	0.06271387	-0.70167400	0.6182138680	0.32669692
-0.3022081	0.56505427	-0.02059100	0.1114816469	-0.20932066
0.5253020	0.11354907	0.14582385	-0.0495238833	0.13548535
0.3295781	-0.52856061	-0.01461109	-0.1224494378	0.26868448
0.4471957	0.38887766	0.05592362	-0.0301801108	0.03743627
-0.1317823	0.22884540	-0.48663021	-0.7476601992	0.36126652

Singular Value Decomposition of Weighted Variables

X = U S(d) V

(124 x 6) = (124 x 5) (1 x 6) (5 x 5)

6: Number of variables

124: Number of participants

5: Number of Principle components

```
In [29]: myData <- standardizedVariables[, c(9,10,11,12,13,14)]

         svd(na.omit(myData), nu = min(124,5), nv = min(5,5)) #SVD
```

$d 1. 15.6907082364576 2. 13.9436041468325 3. 12.0903960417362 4. 9.20912801583384 5. 7.11100029627979 6. 3.97813247178339

-0.0417460930	0.119886907	0.039320892	0.047454504	-0.0002104621
-0.0062245552	0.147585549	-0.019208461	-0.048045314	0.1071645136
0.0394211644	-0.029927606	-0.063222983	0.014865482	0.0466634419
0.1059804923	-0.031140464	0.018640954	0.153808781	-0.0436786286
0.0374625824	0.053780561	-0.039296840	-0.042469125	0.0620552383
0.0317516980	0.019120168	-0.008188557	-0.005797065	-0.1024990122
-0.0755334809	-0.129562684	0.001740033	0.070735129	0.1026766190
-0.1092951463	0.040391558	0.019873935	-0.056318947	0.0156528552
0.2377735841	-0.086299108	-0.036296261	0.004656842	-0.0632306111
0.0007849976	0.006363226	-0.155119925	0.004326619	-0.0039701520
0.1023429652	0.011200446	0.055068100	0.014705024	0.0318153290
0.1008315324	-0.189582729	0.117529916	-0.110515220	0.0153711173
-0.0228204837	-0.008774435	-0.090941064	0.033214843	0.1412227975
-0.0904290256	-0.014428915	0.028905005	0.020718706	-0.2391660309
-0.0300747837	-0.047321506	0.039622021	-0.073603004	0.0085025140
-0.0233689615	0.047677159	0.057139530	0.003882968	-0.1088838010
-0.1397160421	0.089664877	0.069367563	-0.102803541	-0.0411873215
0.0840637227	0.056855435	0.035177083	0.008742479	0.0711587937
0.0203671914	-0.151400715	0.205088856	-0.085395986	-0.0933777846
-0.0592782419	0.050478789	0.080242240	-0.063172647	0.0237924485
-0.0389311149	0.003760294	0.170597523	0.256427336	0.0063567338
-0.0808069478	-0.056156135	-0.051140450	0.133228250	-0.0233942823
0.1565505079	-0.190561345	-0.117735727	-0.033179095	0.0186974156
0.0136229409	-0.026707003	0.002842817	-0.009677177	0.1615123484
-0.0016654668	-0.104416339	0.060898161	-0.087654163	0.0294625217
0.1606414211	-0.175252861	0.029300696	-0.045786471	-0.0718043615
0.0011127673	0.029054594	-0.067287921	0.001883633	-0.1294648006
0.0428406846	0.119018578	0.052936060	0.002057859	0.2079415018
-0.0587628577	0.059036223	-0.175009028	-0.035254016	-0.0423808958
-0.0831072019	0.061827438	0.008543069	-0.197255782	0.1902785050

$u

-0.023947196	-0.186078075	-0.13179157	0.072297530	0.003995924
-0.072483107	0.065061691	0.16736307	0.062629596	-0.075930141
-0.007644200	-0.086048923	-0.01010201	0.022631950	0.099536566
-0.089072229	-0.020400462	0.17404937	0.052267845	0.069181752
-0.097702195	0.042848259	-0.01930230	-0.021984890	0.016543135
0.065286608	0.104760218	-0.07164561	0.116352932	-0.018314804
0.066856576	-0.026249048	-0.20459125	-0.079481823	0.070529393
-0.117982824	-0.039984777	0.08249572	0.070017221	0.050500557
-0.049670667	-0.078359561	-0.04758677	-0.024400310	-0.092170285
-0.021491936	-0.061913622	0.12109558	-0.155652169	0.031185785
-0.090282151	0.036360041	-0.09612682	0.054799248	-0.071669242
0.017494625	0.021155672	0.08531031	-0.094092836	0.045238470
0.025831257	0.022550303	-0.06883408	0.006741146	-0.005601193
-0.050294695	0.019294809	-0.06265033	-0.030674363	0.038414200
-0.084732363	-0.018005290	0.03569743	-0.003482548	0.074132449
-0.121037230	0.003197264	-0.15506281	0.094758136	0.073466779
0.002034083	0.110581173	0.09950431	0.009533698	0.134490179
0.034331352	-0.030244752	-0.05646496	-0.021251191	-0.154491300
-0.023515513	0.022819558	-0.17417535	0.024802342	0.112143278
0.020643508	0.010628624	0.22637780	-0.200919778	0.084624627
0.090702039	0.041211642	-0.06637714	-0.053045301	-0.146729660
-0.030974547	-0.022653621	-0.04467145	-0.035250723	-0.060711524

$v	0.47120963	-0.5483704	0.1113436	-0.05021965	0.123099599
	0.33547851	0.1988936	0.5310660	0.72383302	-0.191663591
	0.58392817	-0.3680808	-0.1360914	-0.16019449	-0.083485569
	0.39624207	0.4886480	-0.2835016	-0.24917913	-0.625652873
	0.40880556	0.5147895	-0.1282597	-0.03891426	0.741382296
	0.01771372	0.1430482	0.7682623	-0.61990073	-0.007245079

```
In [ ]:
```