Análisis Multivariado

Práctica 2

Multicolinealidad, componentes principales, normal multivariado, proporción de varianzas, Biplot y regresión lineal múltiple.

Enríquez Hernández Leobardo

14 de junio de 2024

Índice

Utilizando la base de datos Student-Scores y con las columnas de las calificaciones obtenidas por los alumnos realizaremos algunos análisis de multicolinealidad de covariables, componentes principales, pruebas de distribución normal multivariado, diagrama de dispersión y proporción de varianza, el Biplot y agrupaciones, verificación de variables importantes, y regresión lineal múltiple.

1. Análisis exploratorio de datos.

Antes de comenzar cargamos la base de datos que vamos a utilizar y hacemos una inspección rápida del tipo de datos para cada variable y de la estadística descriptiva de los datos numéricos.

```r
#Cargamos la base de datos
student_scores <- read.csv("DataSets/student-scores.csv")
summary(student_scores) #Estadística descriptiva de los datos numéricos
```

```
##        id          first_name         last_name           email          
##  Min.   :   1.0   Length:2000        Length:2000        Length:2000       
##  1st Qu.: 500.8   Class :character   Class :character   Class :character  
##  Median :1000.5   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :1000.5                                                           
##  3rd Qu.:1500.2                                                           
##  Max.   :2000.0                                                           
##     gender          part_time_job       absence_days   
##  Length:2000        Length:2000        Min.   : 0.000  
##  Class :character   Class :character   1st Qu.: 2.000  
##  Mode  :character   Mode  :character   Median : 3.000  
##                                        Mean   : 3.666  
##                                        3rd Qu.: 5.000  
##                                        Max.   :10.000  
##  extracurricular_activities weekly_self_study_hours career_aspiration 
##  Length:2000                Min.   : 0.00           Length:2000       
##  Class :character           1st Qu.: 5.00           Class :character  
##  Mode  :character           Median :18.00           Mode  :character  
##                             Mean   :17.76                             
##                             3rd Qu.:28.00                             
##                             Max.   :50.00                             
##    math_score     history_score    physics_score    chemistry_score
##  Min.   : 40.00   Min.   : 50.00   Min.   : 50.00   Min.   : 50    
##  1st Qu.: 77.00   1st Qu.: 69.75   1st Qu.: 71.00   1st Qu.: 69    
##  Median : 87.00   Median : 82.00   Median : 83.00   Median : 81    
##  Mean   : 83.45   Mean   : 80.33   Mean   : 81.34   Mean   : 80    
##  3rd Qu.: 93.00   3rd Qu.: 91.00   3rd Qu.: 92.00   3rd Qu.: 91    
##  Max.   :100.00   Max.   :100.00   Max.   :100.00   Max.   :100    
##  biology_score    english_score   geography_score 
##  Min.   : 30.00   Min.   :50.00   Min.   : 60.00  
##  1st Qu.: 69.00   1st Qu.:72.00   1st Qu.: 71.00  
```

```
##  Median : 81.00   Median :83.00   Median : 81.00
##  Mean   : 79.58   Mean   :81.28   Mean   : 80.89
##  3rd Qu.: 91.00   3rd Qu.:91.00   3rd Qu.: 91.00
##  Max.   :100.00   Max.   :99.00   Max.   :100.00
```

Luego clasificaremos estas variables por escalas, agruparemos las numéricas y enterios por un lado, luego las nominales y ordinales por otro lado.

```r
tipoDatos <- sapply(student_scores, class) # Saber los tipos de datos
continuas <-  which(tipoDatos == "numeric") # continuas
enteras <- which(tipoDatos == "integer") # enteras
numericas <- c(continuas,enteras) #unirlas

nominales <- which( tipoDatos == "factor") # categoricas
ordinales <- which( sapply(student_scores, is.ordered) )  # ordinales
categoricas <- c(nominales, ordinales) #unirlas
```

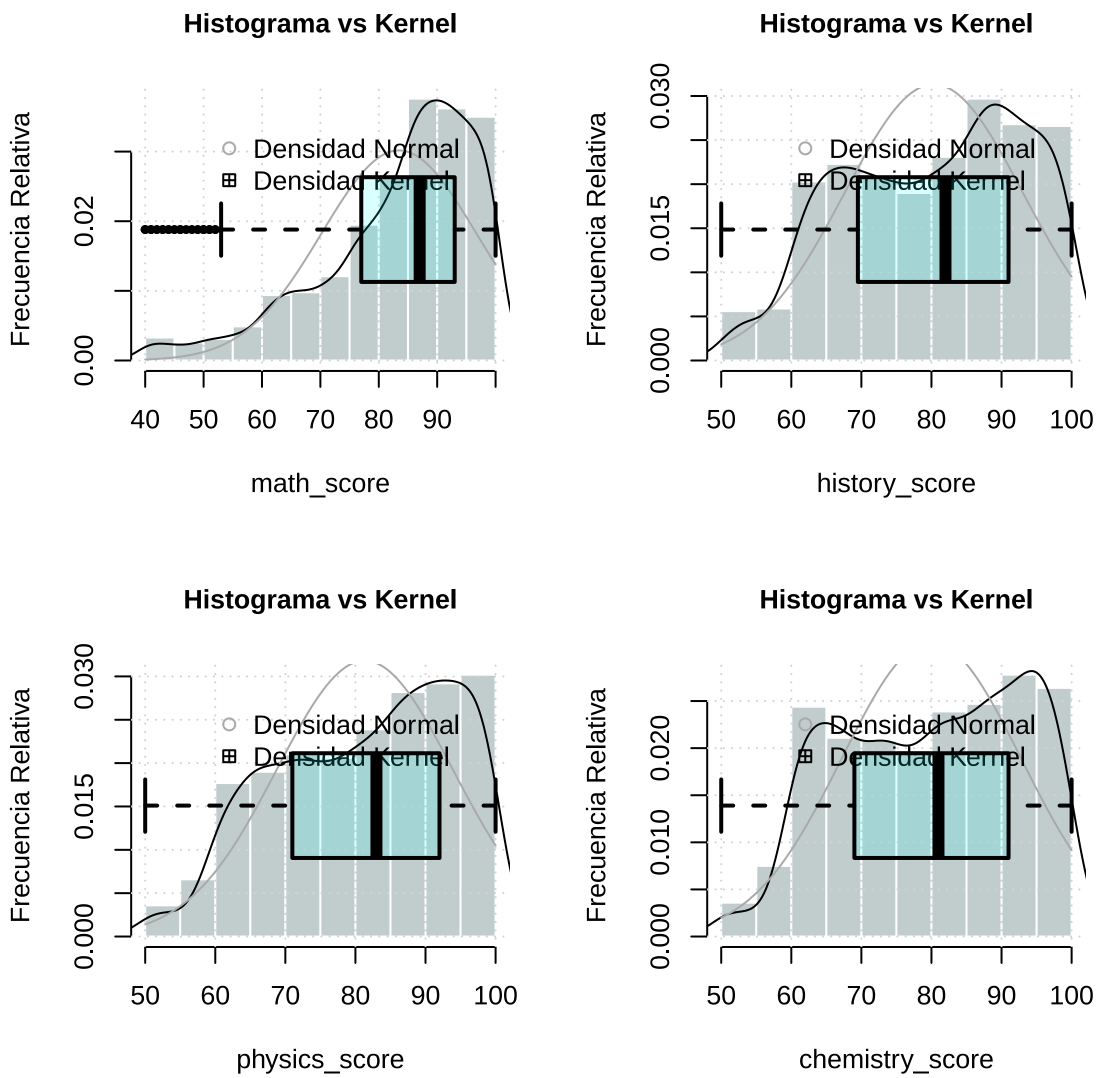
Ya que realizamos una inspección básica a las columnas de la base de datos, el tipo de dato, así como sus estadísticos principales lo que sigue será definir otras 2 variables donde tomaremos las columnas de las calificaciones de los alumnos y otra con la variable dependiente para hacer el modelo de regresión.

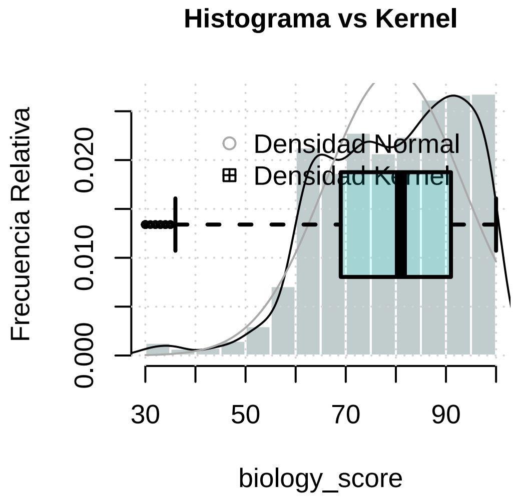
```r
vars_predict <- c('math_score','history_score','physics_score',"chemistry_score",
                  "biology_score","english_score","geography_score")  #covariables
dependiente <- "weekly_self_study_hours" #variable explicada o dependiente
```

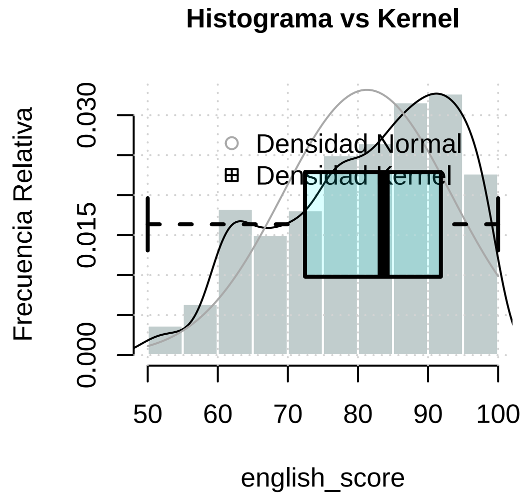
Con las nuevas variables definidas haremos un análisis exploratorio para ver si no tendremos problemas de escalas con los datos, outliers, NA´s, etc. Algo que debemos mencionar es que con la variable vars_predict donde guardamos las calificaciones de los alumnos no tendremos problemas de escala ya que las calificaciones por alumno se encuentran definidas con el mismo rango escalar de 0 a 100.

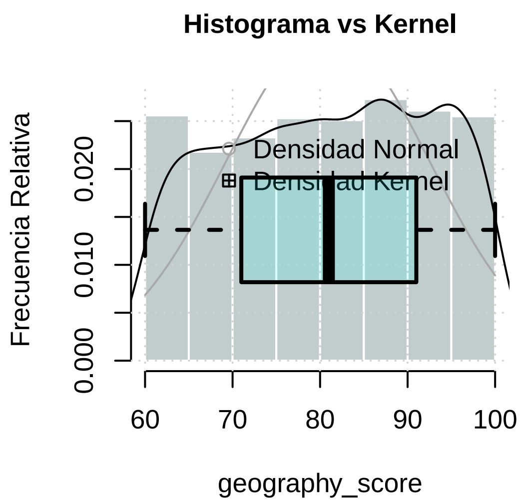
En los siguientes histogramas con BoxPlot y densidad combinados, podemos observar algunos comportamientos de las variables explicativas. Por ejemplo math_score tiene una asimetría hacia la izquierda (o negativa), mostrando varios outliers por debajo del primer cuantil. Por otro lado, geography_score muestra un comportamiento más uniforme, con mayor simetría y sin outliers.

```r
par(mfrow = c(2, 2))
multi.hist(student_scores[, vars_predict]) # Histogramas con formato de funciones.R
```


Histograma vs Kernel
Frecuencia Relativa
Densidad Normal
Densidad Kernel
math_score
0.00
0.02
40
50
60
70
80
90
Histograma vs Kernel
Frecuencia Relativa
Densidad Normal
Densidad Kernel
history_score
0.000
0.015
0.030
50
60
70
80
90
100
Histograma vs Kernel
Frecuencia Relativa
Densidad Normal
Densidad Kernel
physics_score
0.000
0.015
0.030
50
60
70
80
90
100
Histograma vs Kernel
Frecuencia Relativa
Densidad Normal
Densidad Kernel
chemistry_score
0.000
0.010
0.020
50
60
70
80
90
100

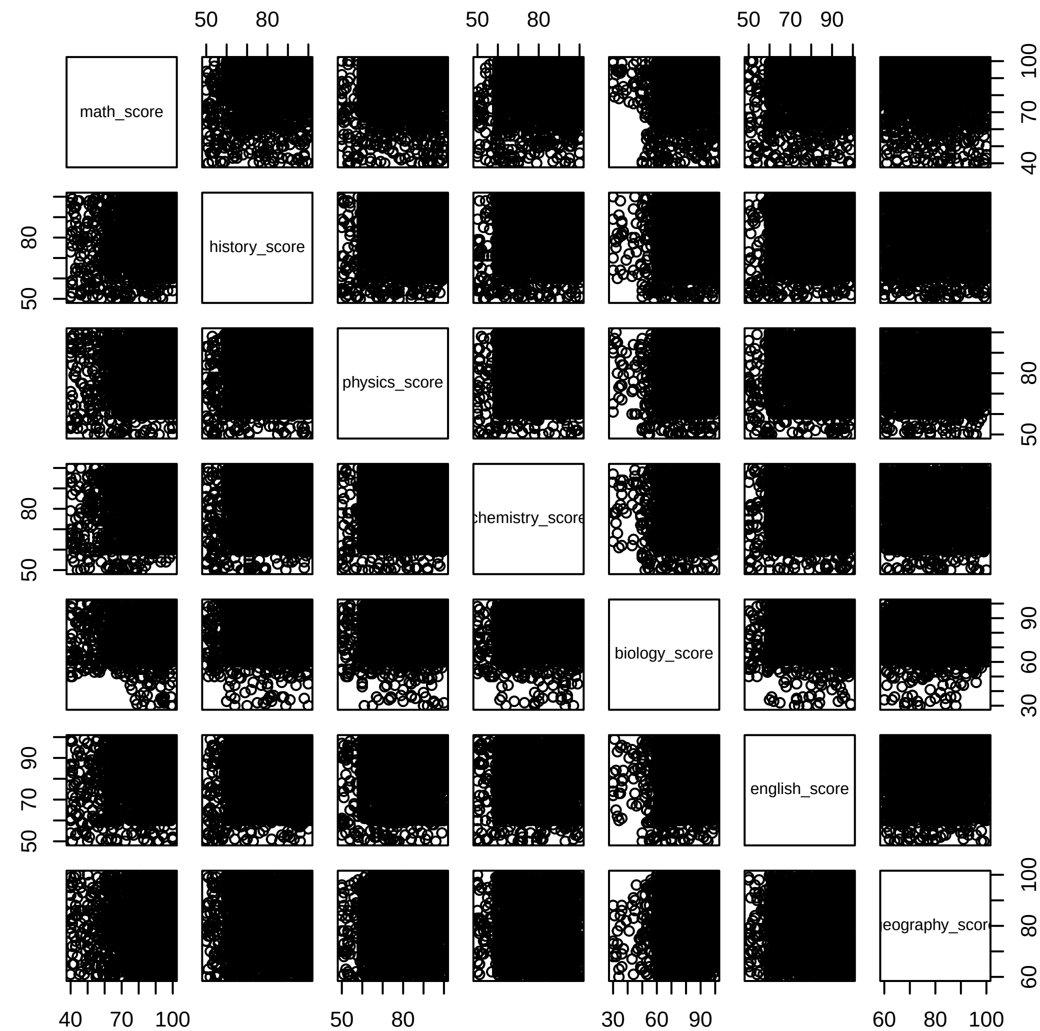






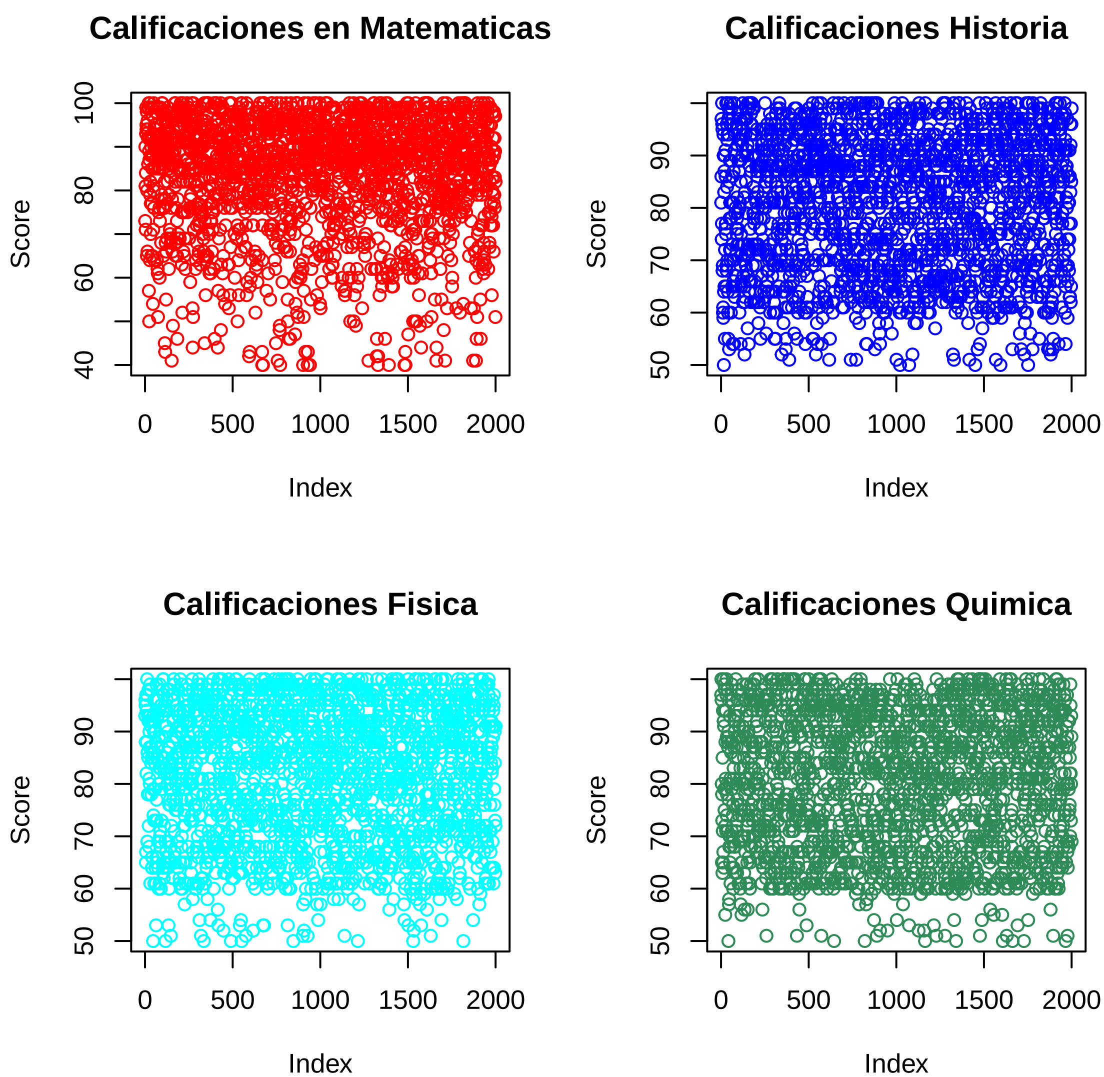
En las siguientes gráficas de dispersión entre las covariables, podemos observar que no hay relaciones lineales evidentes entre éstas.

```
pairs(student_scores[,vars_predict])
```

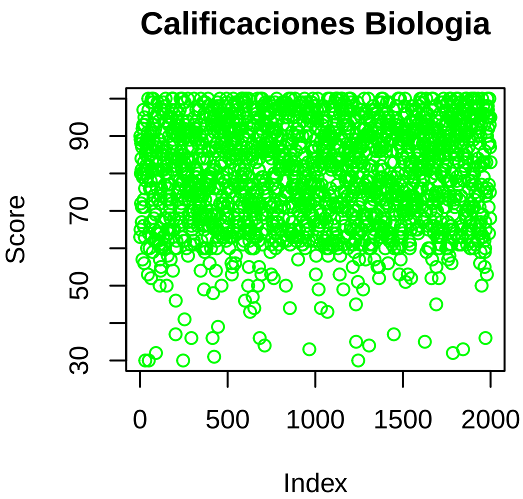


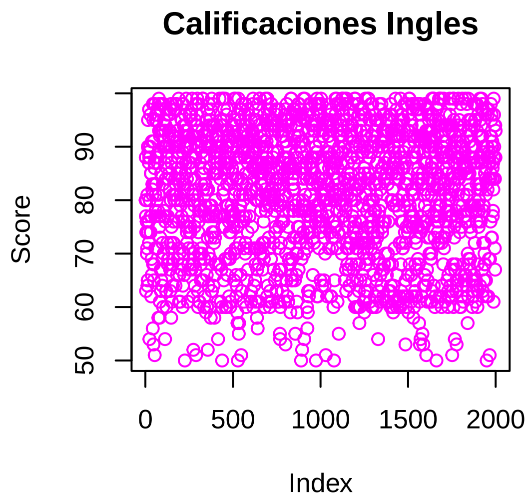
En las siguientes gráficas de dispersión individuales, no se muestra algún comportamiento, tendencia o patrón definido.

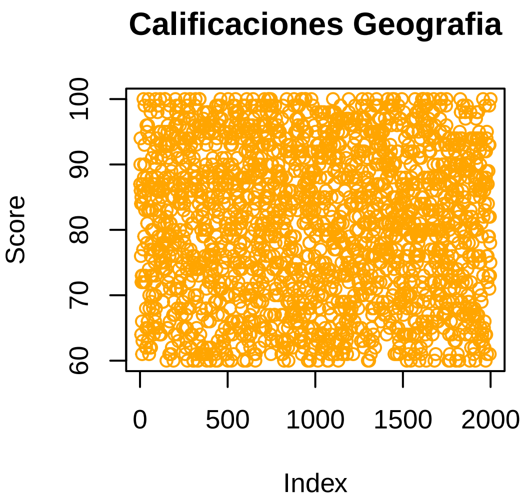
```
par(mfrow = c(2, 2))
plot(student_scores$math_score,col="red",main = "Calificaciones en Matematicas",ylab = "Score")
plot(student_scores$history_score,col="blue",main = "Calificaciones Historia",ylab = "Score")
plot(student_scores$physics_score,col="cyan",main = "Calificaciones Fisica",ylab = "Score")
plot(student_scores$chemistry_score,col="seagreen",main = "Calificaciones Quimica",ylab = "Score")
```




```r
plot(student_scores$biology_score,col="green",main = "Calificaciones Biologia",ylab = "Score")
plot(student_scores$english_score,col="magenta",main = "Calificaciones Ingles",ylab = "Score")
plot(student_scores$geography_score,col="orange",main = "Calificaciones Geografia",ylab = "Score")
```

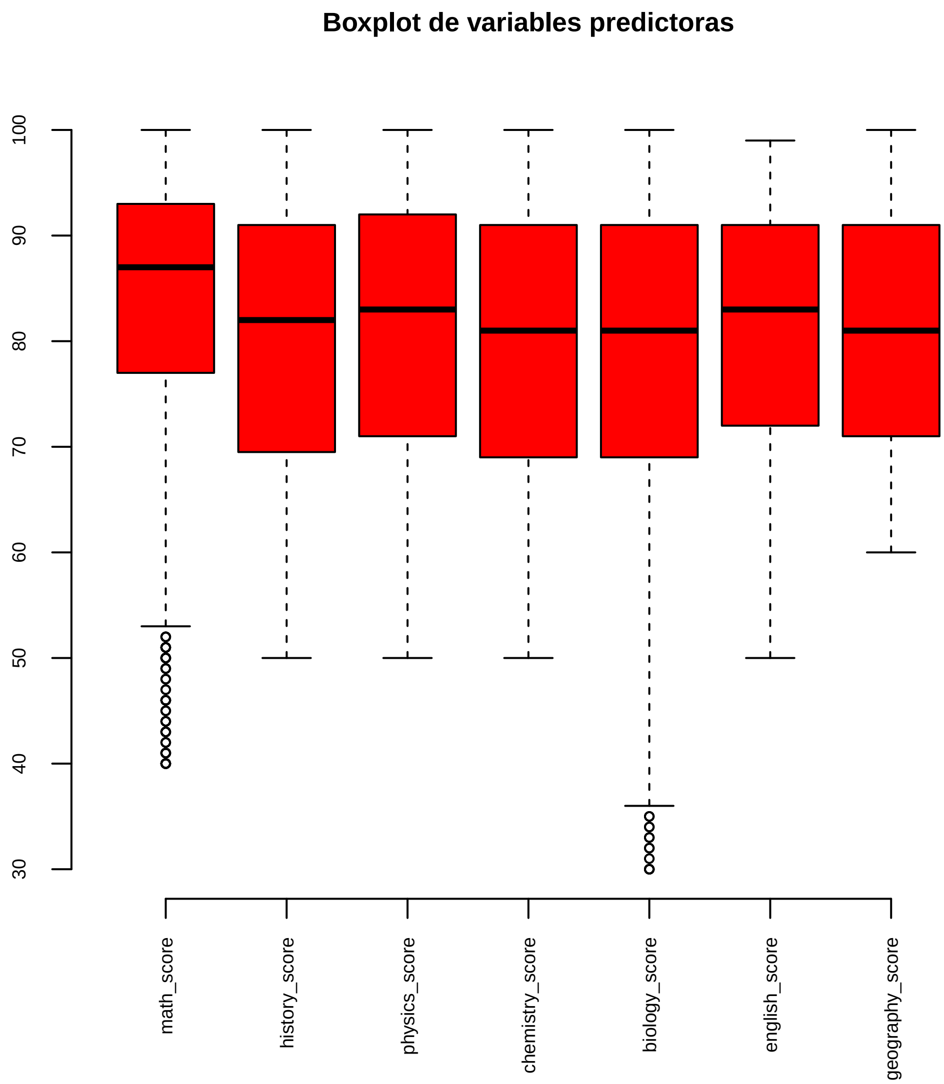






En los siguientes BoxPlot parece no haber problemas de escala, y algunos aoutliers en las variables math_score y biology_score.

```
par(bty = "n")
boxplot(student_scores[,vars_predict],main = "Boxplot de variables predictoras",las = 3,cex=0.6,cex.mai
```



Del análisis exploratorio que hicimos podemos ver que en efecto no tenemos problemas de escala con las calificaciones, tenemos algunos valores outliers o atípicos pero no es algo que nos genere problemas.

2. Determinar si existe multicolinealidad usando VIF o KMO.

```
KMO(student_scores[,vars_predict])
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = student_scores[, vars_predict])
## Overall MSA =  0.66
## MSA for each item =
##      math_score   history_score   physics_score chemistry_score   biology_score
##            0.66            0.65            0.65            0.67            0.67
##   english_score geography_score
##            0.65            0.67
```

De los resultados obtenidos de la prueba KMO tenemos que los valores son mediocres o regulares para nuestras variables, pues están fuera del rango 0.8 a 1, donde la muestra sería adecuada para el análisis factorial. Sin embargo no son menores a 0.6, que indicaría que la muestra no es adecuada.

Por otro lado, podemos observar el factor de inflación de varianza y parece no haber multicolinealidad entre las covariables.

```
# definimos los datos que usara el modelo
#variables originales
datos <- student_scores[c(vars_predict, dependiente)]
fit_model <- lm(weekly_self_study_hours ~., data = datos)
colinearidad <- data.frame(Variance_Inflation_Factor = vif(fit_model));colinearidad
```

```
##                   Variance_Inflation_Factor
## math_score                         1.059224
## history_score                      1.055562
## physics_score                      1.047626
## chemistry_score                    1.051409
## biology_score                      1.046167
## english_score                      1.043667
## geography_score                    1.027365
```

3. Componentes principales

Ahora obtendremos las componentes principales de nuestras variables predictoras. Se observa que la proporción de varianza no se acumula rápidamente a 1, el primer componente tiene 0.23, y los primeros tres acumulan 0.52, es decir, la mitad. Hasta el quinto componente acumula 77 %.

```
comps <- princomp(student_scores[,vars_predict])
summary(comps)
```

```
## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     16.1894537 13.2037936 12.3394732 12.0545319 11.7906155
## Proportion of Variance  0.2328853  0.1549084  0.1352916  0.1291155  0.1235238
## Cumulative Proportion   0.2328853  0.3877937  0.5230853  0.6522008  0.7757246
##                           Comp.6     Comp.7
## Standard deviation     11.3260534 11.1413200
## Proportion of Variance  0.1139816  0.1102938
## Cumulative Proportion   0.8897062  1.0000000
```

La matriz Gamma o de vectores propios es la siguiente.

```
#### loadings:  Matriz Gamma
comps$loadings
```

```
## 
## Loadings:
##                 Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## math_score       0.455  0.490  0.218  0.547  0.404         0.207
## history_score    0.380  0.368 -0.500 -0.330 -0.207  0.563       
## physics_score    0.348 -0.195  0.589  0.137 -0.509  0.332 -0.327
## chemistry_score  0.401         0.368 -0.727  0.320 -0.268       
## biology_score    0.489 -0.712 -0.362  0.180  0.293              
## english_score    0.285  0.250 -0.299  0.107 -0.327 -0.644 -0.484
## geography_score  0.217 -0.127               -0.490 -0.291  0.782
## 
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
```

```
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

La muestra aleatoria de las componentes principales es la siguiente, se muestran las primeras 10.

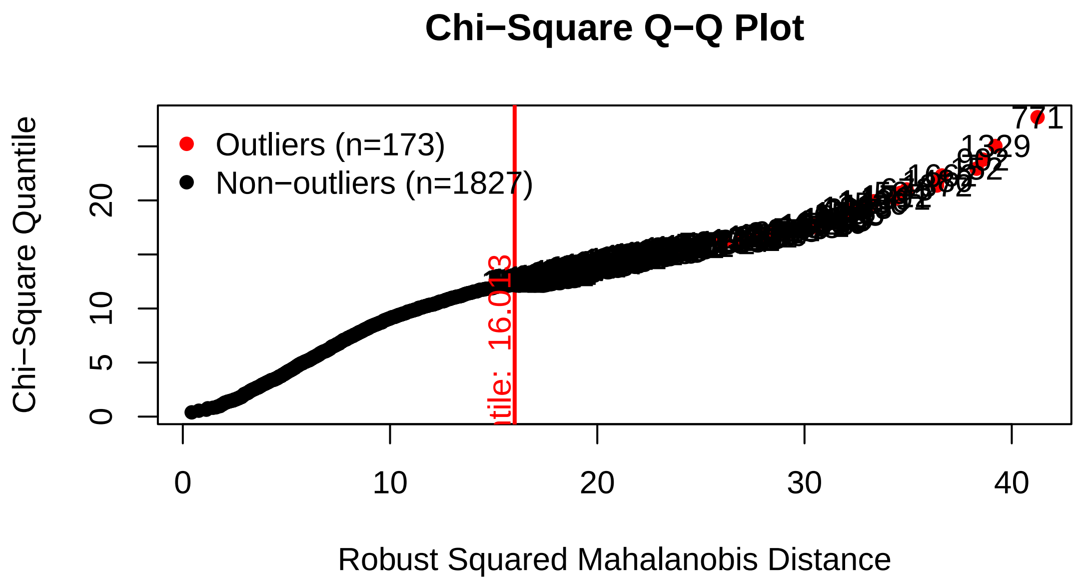
```r
#### scores: La muestra aleatoria de las ccomponentes ppales
head(comps$scores, 10)
```

```
##           Comp.1      Comp.2      Comp.3     Comp.4     Comp.5    Comp.6
##  [1,]  -0.7618592   3.9668266  16.9228253 -19.763332 -12.299268 -2.028698
##  [2,]  27.2381257  -3.9896721   8.8395286  -8.133651  -3.200276 -3.822412
##  [3,]  10.8878736  10.2934527  11.6784544 -19.563597 -15.583157  8.038332
##  [4,]  -5.2530396 -21.6573752   6.4468393  -4.023467  -0.870790  8.953877
##  [5,] -15.6437646   0.3758993 -11.3431384   9.306846   9.325239  2.860983
##  [6,]   2.6354894  19.3433696 -13.7773673  -3.262716   3.103894  6.729581
##  [7,]  14.6028988   4.9981457   0.6833324  11.341741   5.301996 24.818274
##  [8,]   4.0685963   4.0334681  -8.8920743  13.544673  -1.177618 19.489528
##  [9,]   3.2070066  -3.4125735  19.3950023   7.330242   9.122398  3.460563
## [10,]  -9.7758235   8.8484338  17.0498812  15.711055   5.489164  7.068756
##           Comp.7
##  [1,]   0.3967555
##  [2,]   0.0857024
##  [3,]   8.8575581
##  [4,]   7.2897149
##  [5,]   4.9865151
##  [6,]  10.9458277
##  [7,] -12.4114245
##  [8,]  11.1719512
##  [9,]  -5.9750084
## [10,]  -0.1306289
```

4. Distribucion Normal multivariado.

Con las componentes obtenidas veremos si estas se distribuyen de forma normal multivariada con el Henze-Zirkler's test y el método de detección de outliers quan que es un método cuantil basado en la distancia Mahalanobis. A continuación se muestra el Chi-Square Q-Q Plot generado.

```r
#fit <- MVN::mvn(data = comps$scores, mvnTest = "hz", multivariateOutlierMethod = "quan")
scores_matrix<-as.matrix(comps$scores)
scores_matrix<-as.data.frame(scores_matrix)
names(scores_matrix)<-c("comp1", "comp2","comp3", "comp4" , "comp5","comp6", "comp7" )
ajuste <- MVN::mvn(data = scores_matrix, mvnTest="hz", multivariateOutlierMethod="quan")
```



Observemos que el test de Henze-Zirkler rechaza que sea normal multivariado.

```
ajuste$multivariateNormality
```

```
##           Test       HZ p value MVN
## 1 Henze-Zirkler 3.050838       0  NO
```

Finalmente, el test de Anderson-Darling muestra que para 5 de los 7 componentes no forman una normal multivariada, particularmente las primeras dos componentes que nos interesan.

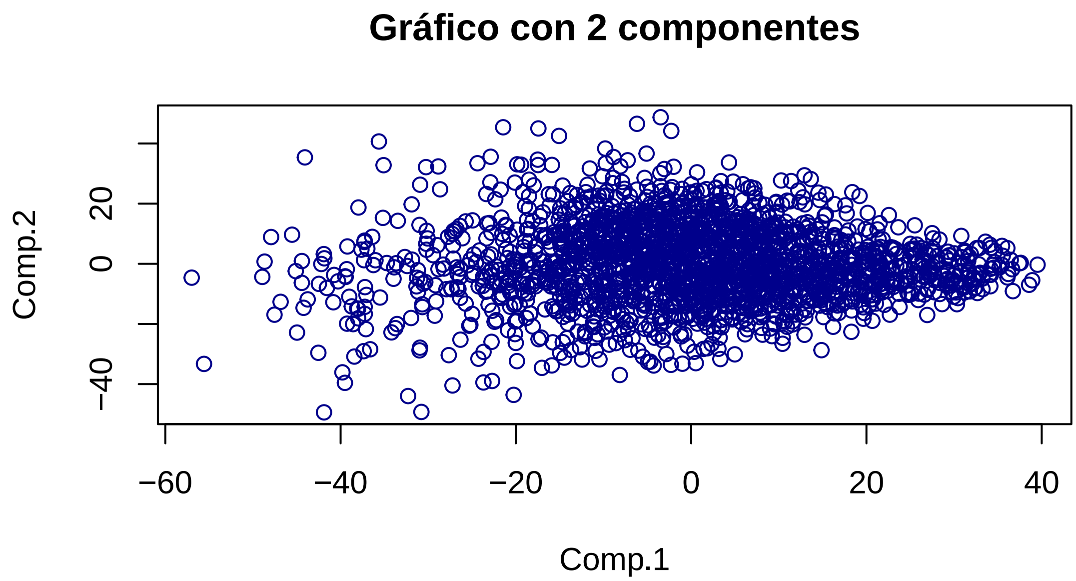
```
ajuste$univariateNormality
```

```
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling   comp1      2.3329  <0.001      NO
## 2 Anderson-Darling   comp2      1.4887   8e-04      NO
## 3 Anderson-Darling   comp3      0.2034  0.8765      YES
## 4 Anderson-Darling   comp4      1.2723  0.0026      NO
## 5 Anderson-Darling   comp5      0.5219   0.184      YES
## 6 Anderson-Darling   comp6      1.1058  0.0067      NO
## 7 Anderson-Darling   comp7      0.8082  0.0365      NO
```

5. Diagrama de Dispersion y proporción de varianza.

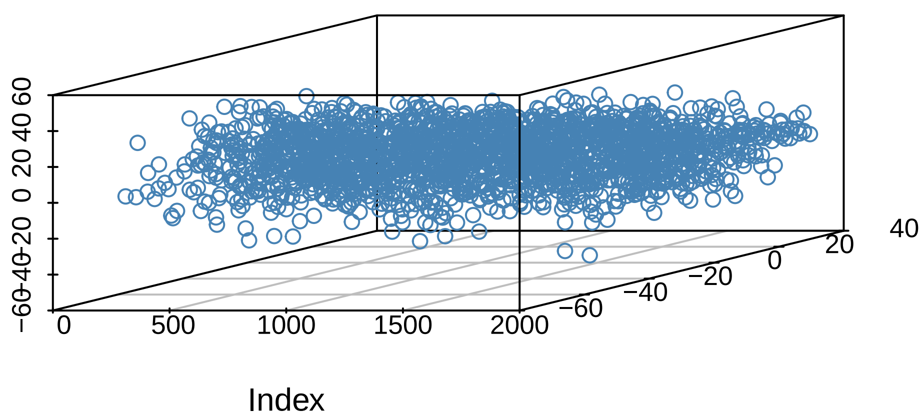
Con la variable de scores obtenida del calculo de las componentes principales graficaremos un diagrama de dispersión tomando únicamente las primeras 2 componentes. Lo que podemos observar es que las primeras 2 componentes no están conservando mucha variabilidad de los datos originales.

```
plot(comps$scores[,c(1,2)], col = "blue4", main = "Gráfico con 2 componentes")
```



Si graficamos los primeros tres componentes tenemos lo siguiente.

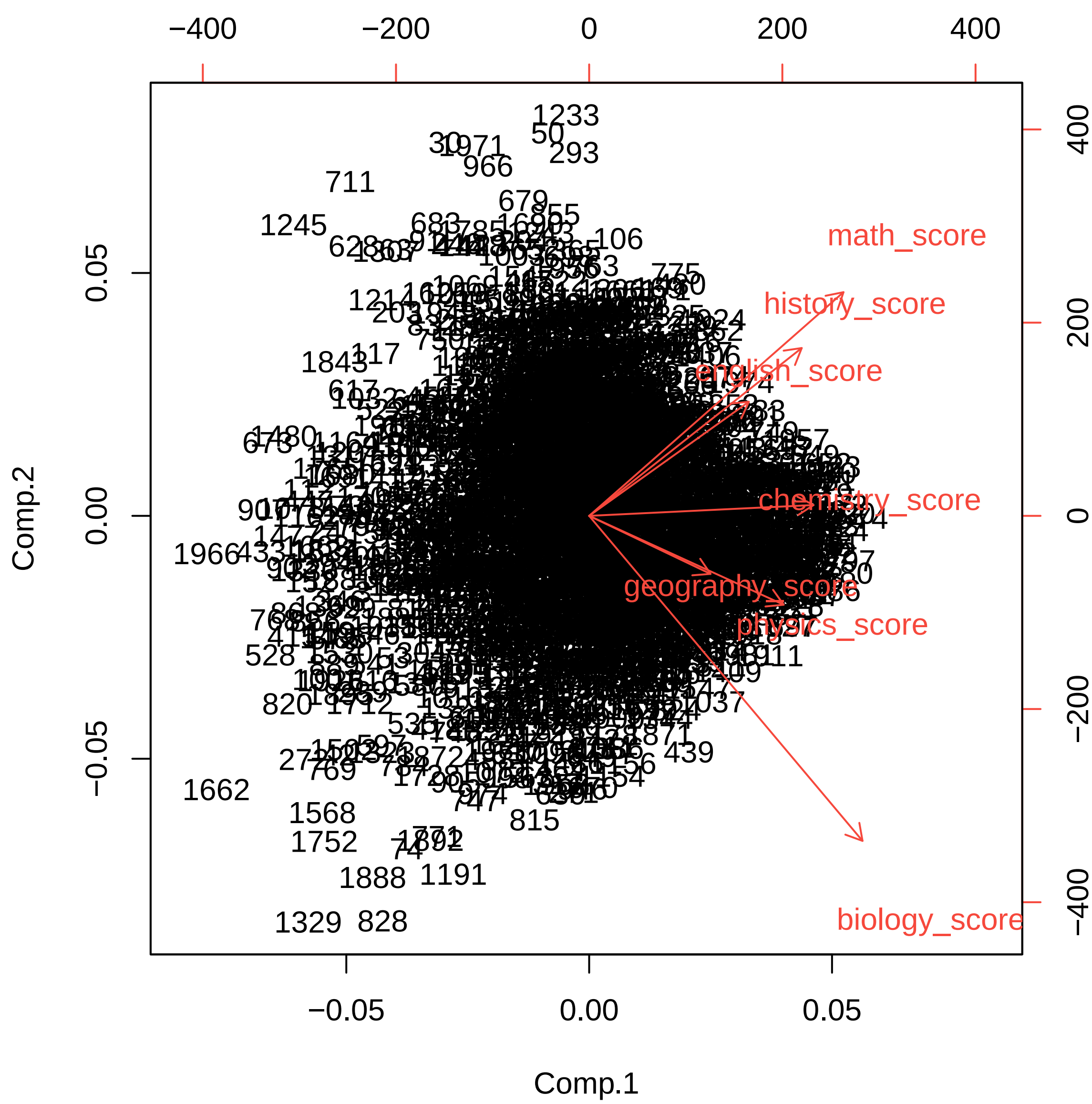
```
scatterplot3d(comps$scores[,c(1,3)], color="steelblue")
```



6. Biplot, agrupaciones y variables importantes.

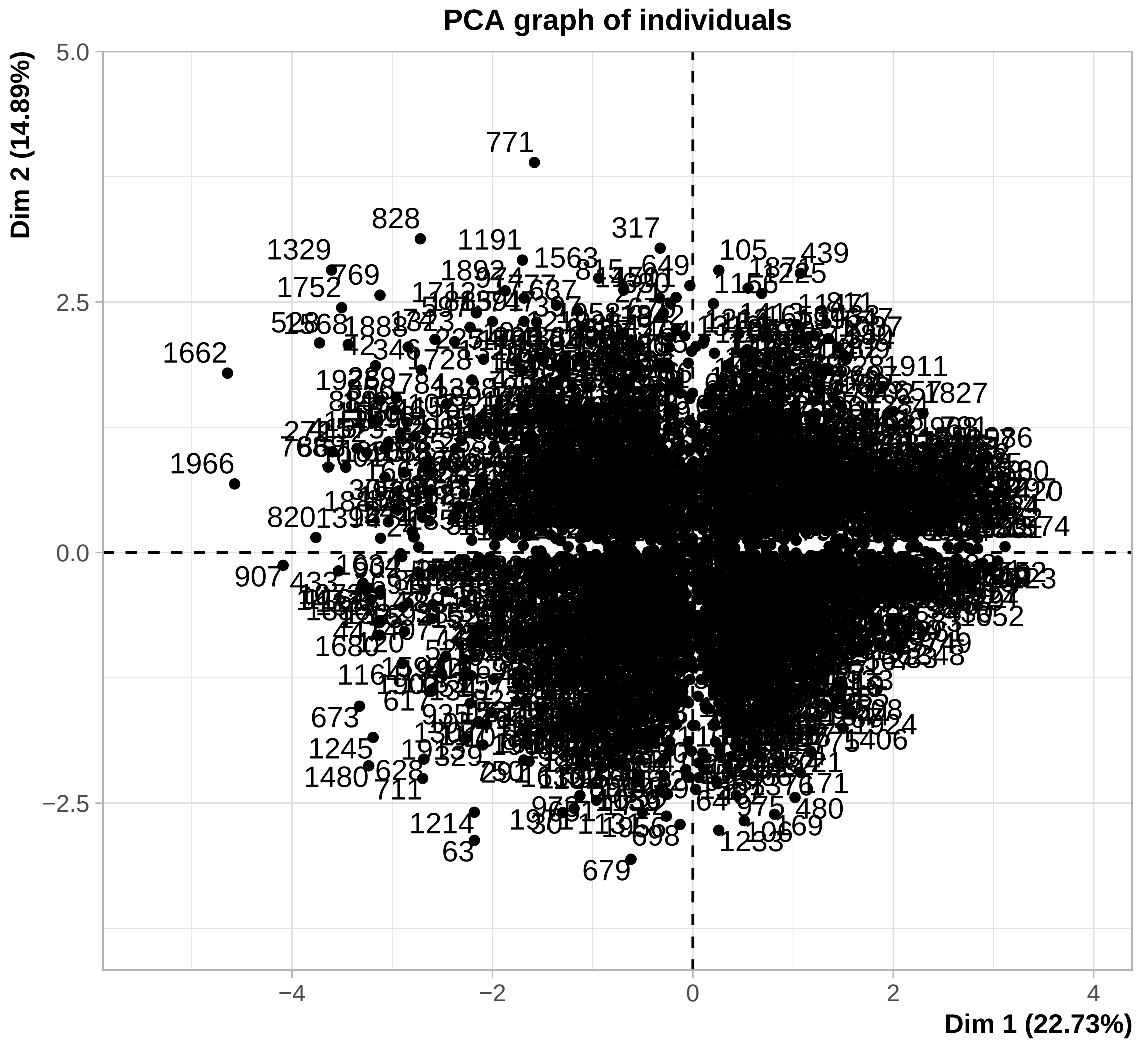
En la siguiente gráfica Biplot de los primeros dos componentes vemos las flechas que son las variables y los puntos numéricos que son las observaciones. El largo de las flechas indica la varianza, entre más largas mayor varianza. El coseno del ángulo entre las flechas aproxima la correlación entre las variables, entre más cercano a 90 o 270 grados menor correlación entre las variables, un ángulo de 0 o de 180 grados refleja correlación de 1 y -1 respectivamente. Observamos aquí las correlaciones entre las variables dentro de los componentes principales.

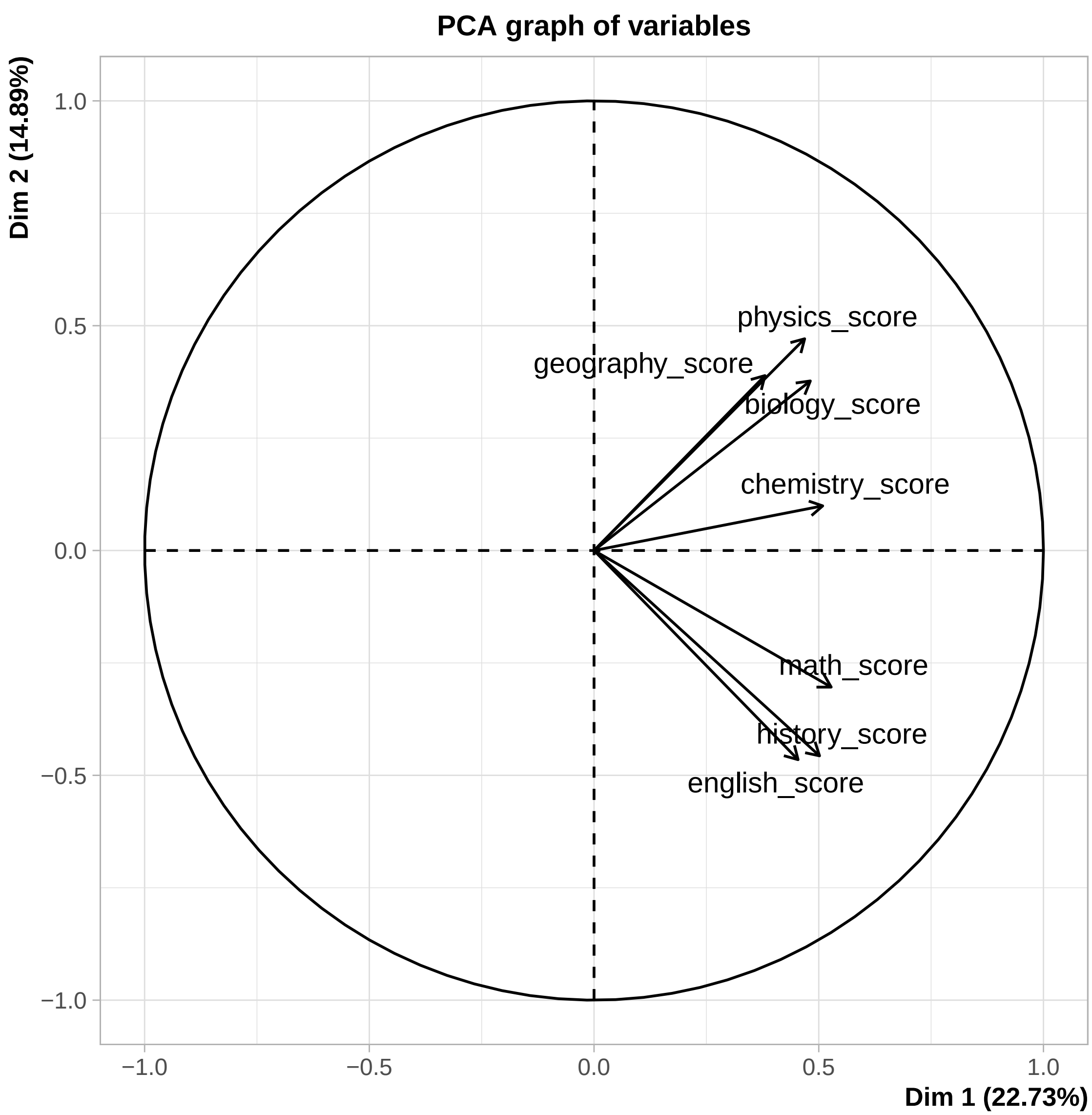
```r
biplot(comps)
```



Por otro lado, podemos usar otro método, en el que en ncp indicamos el número de componentes principales que queremos, y con ello obtenemos un PCA graph, por una parte de observaciones y por otro de la variable, por separado.

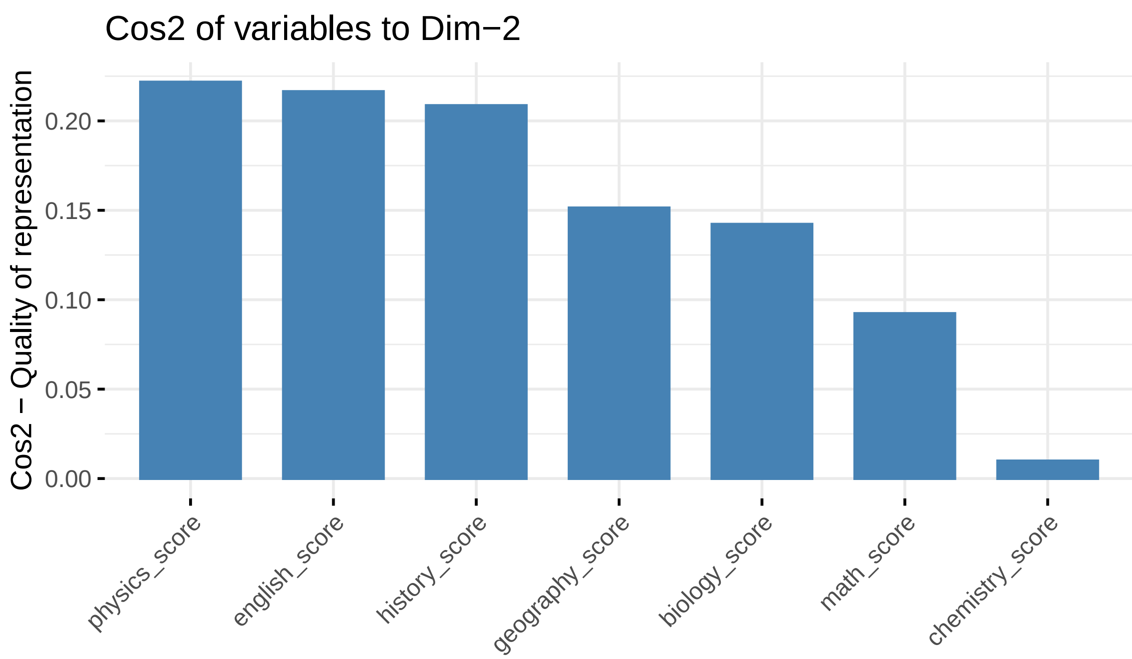
```r
par(mfrow = c(1, 2))
pca_comps <- PCA(student_scores[,vars_predict],scale.unit = T, ncp = 3, graph = T) #FactoMineR package
```


PCA graph of individuals
Dim 2 (14.89%)
Dim 1 (22.73%)
5.0
2.5
0.0
−2.5
−4
−2
0
2
4



Para determinar la importancia de cada componente con las variables originales usaremos la función cos2 la cual nos dice que un valor bajo significa que la variable no está perfectamente representada por esa componente, mientras que un valor alto significa que es una buena representación de esa varíable con esa componente. con Coseno 2 podemos observar la importancia de las variables en las primeras componentes principales. En este caso tenemos las correlaciones entre las variables y las componentes principales.

```
fviz_cos2(pca_comps,choice = "var",axes = 2)#factoextra package
```



Observando las dimensiones tenemos lo siguiente.

```
pca_comps$desc <- dimdesc(pca_comps, axes = c(1,2), proba = 0.05)
pca_comps$desc$Dim.1
```

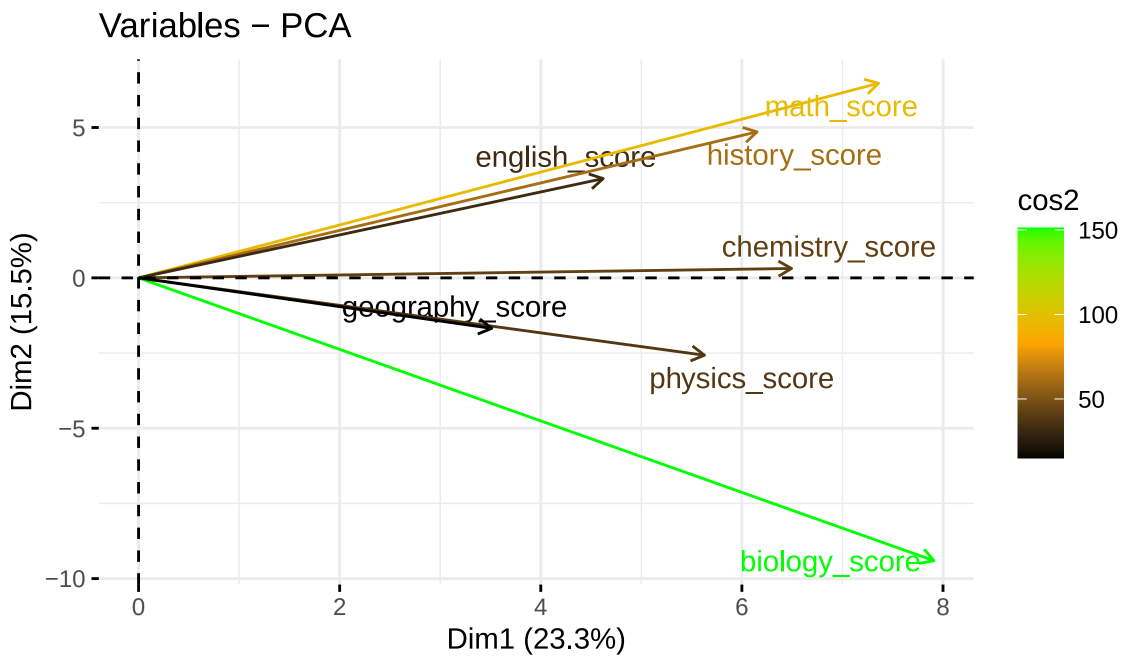
```
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##                 correlation       p.value
## math_score        0.5277368 7.999187e-144
## chemistry_score   0.5088131 3.603752e-132
## history_score     0.5017574 5.212873e-128
## biology_score     0.4812949 1.757672e-116
## physics_score     0.4687030 9.099105e-110
## english_score     0.4540056 2.806852e-102
## geography_score   0.3803661  7.369194e-70
```

```
pca_comps$desc$Dim.2
```

```
##
## Link between the variable and the continuous variables (R-square)
## =================================================================================
##                 correlation       p.value
## physics_score    0.47086177 6.726618e-111
## geography_score  0.38903172  2.900618e-73
## biology_score    0.37711742  1.310961e-68
## chemistry_score  0.09926032  8.696948e-06
## math_score      -0.30379176  5.706193e-44
## history_score   -0.45669936 1.267281e-103
## english_score   -0.46520160 5.978037e-108
```

Otra forma de visualizar esto es combinar el biplot y la importancia de las componentes en el que los atributos con puntuaciones cos2 similares tendrán colores similares.

```r
fviz_pca_var(comps, col.var = "cos2",   #replicar esta grafica con pca_comps en lugar
                                                  #del parametro comps.
             gradient.cols = c("black", "orange", "green"),
             repel = TRUE)
```



7. Regresión lineal múltiple. Modelo predictivo.

Como ultimo paso ajustaremos un modelo de regresión usando las variables que definimos al principio donde tomamos las calificaciones y la variable que nos interesa modelar.

Se muestra a continuación el factor de inflación de varianza para un modelo de regresión con datos originales, todos son cercanos a 1 por lo que no parece haber problemas de multicolinealidad.

```r
# definimos los datos que usara el modelo
#variables originales
datos <- student_scores[c(vars_predict, dependiente)]
fit_model <- lm(weekly_self_study_hours ~., data = datos)
colinearidad <- data.frame(Variance_Inflation_Factor = vif(fit_model));colinearidad
```

```
##                   Variance_Inflation_Factor
## math_score                         1.059224
## history_score                      1.055562
## physics_score                      1.047626
## chemistry_score                    1.051409
## biology_score                      1.046167
## english_score                      1.043667
## geography_score                    1.027365
```

A continuación se muestra el summary del modelo ajustado con los datos originales.

```r
summary(fit_model)
```

```
##
## Call:
```

```
## lm(formula = weekly_self_study_hours ~ ., data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8797  -7.5630   0.3167   8.0106  24.8864
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -62.86002    3.12556 -20.112  < 2e-16 ***
## math_score        0.28307    0.01790  15.815  < 2e-16 ***
## history_score     0.16762    0.01855   9.034  < 2e-16 ***
## physics_score     0.11099    0.01877   5.912 3.96e-09 ***
## chemistry_score   0.09286    0.01846   5.031 5.32e-07 ***
## biology_score     0.09042    0.01714   5.274 1.48e-07 ***
## english_score     0.15483    0.01954   7.925 3.76e-15 ***
## geography_score   0.09015    0.02003   4.501 7.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.28 on 1992 degrees of freedom
## Multiple R-squared:  0.2838, Adjusted R-squared:  0.2813
## F-statistic: 112.8 on 7 and 1992 DF,  p-value: < 2.2e-16
```

A continuación mostramos el factor de inflación de varianza para un modelo de regresión con componentes principales, todos son 1 por lo que no hay problemas de multicolinealidad.

```
#componentes principales
data2 <- cbind(student_scores[,dependiente],comps$scores)
colnames(data2) <- c('weekly_self_study_hours','comp1','comp2','comp3','comp4','comp5','comp6','comp7')
data2<-as.data.frame(data2)
fit_model2 <- lm(weekly_self_study_hours ~., data = data2)
colinearidad_check <- data.frame(Variance_Inflation_Factor = vif(fit_model2)); colinearidad_check
```

```
##       Variance_Inflation_Factor
## comp1                         1
## comp2                         1
## comp3                         1
## comp4                         1
## comp5                         1
## comp6                         1
## comp7                         1
```

A continuación se presenta el summary del modelo ajustando con los componentes principales.

```
summary(fit_model2)
```

```
##
## Call:
## lm(formula = weekly_self_study_hours ~ ., data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8797  -7.5630   0.3167   8.0106  24.8864
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 17.755500   0.229940  77.218  < 2e-16 ***
## comp1        0.376119   0.014203  26.481  < 2e-16 ***
## comp2        0.143781   0.017415   8.256 2.70e-16 ***
## comp3       -0.001283   0.018635  -0.069    0.945
## comp4        0.081004   0.019075   4.247 2.27e-05 ***
## comp5       -0.015468   0.019502  -0.793    0.428
## comp6       -0.010707   0.020302  -0.527    0.598
## comp7        0.019595   0.020639   0.949    0.343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.28 on 1992 degrees of freedom
## Multiple R-squared:  0.2838, Adjusted R-squared:  0.2813
## F-statistic: 112.8 on 7 and 1992 DF,  p-value: < 2.2e-16
```