Practica III

Robinson Aldair Cuayal

2023-03-11

Table of Contents

# Punto 1

El ejercicio en cuestión implica la realización de un análisis de regresión para determinar si existe una relación entre el nivel de plomo en el suelo y la categoría del individuo, y si el nivel de plomo en el suelo es un factor significativo para predecir los niveles altos de plomo en la sangre de los niños. Se proporcionan la hipótesis nula y la alternativa, así como los valores de p, la ecuación de regresión y las estadísticas relevantes. El objetivo principal es predecir los niveles altos de plomo en la sangre de los niños. Como primer paso, se va a almacenar la informacion de las tablas en un excel para luego ser cargado y convertido a un dataframe en R.

library(readxl)

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

datos = data.frame(read\_excel("dataset\_ejer1.xlsx"))  
dim(datos)

## [1] 139 2

head(datos)

## Categoria\_del\_Individuo Nivel\_plomo\_suelo  
## 1 1 1290  
## 2 0 90  
## 3 1 894  
## 4 0 193  
## 5 1 1410  
## 6 1 410

Con la carga de los datos, como segundo paso se implementa el modelo de regresion logistica. Cabe resaltar que es importante especificar el tipo de enlace en la modelación logística ya que afecta directamente la interpretación de los coeficientes del modelo y los resultados obtenidos.

# Crear la regresión logística  
modelo\_logistico = glm(  
 Categoria\_del\_Individuo ~ Nivel\_plomo\_suelo,  
 data = datos,  
 family = binomial(link = "logit")  
)  
#modelo general  
summary(modelo\_logistico)

##   
## Call:  
## glm(formula = Categoria\_del\_Individuo ~ Nivel\_plomo\_suelo, family = binomial(link = "logit"),   
## data = datos)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.27743 -0.80117 0.05973 0.85069 1.80313   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.516206 0.339557 -4.465 8.0e-06 \*\*\*  
## Nivel\_plomo\_suelo 0.002743 0.000544 5.042 4.6e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 191.82 on 138 degrees of freedom  
## Residual deviance: 140.24 on 137 degrees of freedom  
## AIC: 144.24  
##   
## Number of Fisher Scoring iterations: 6

# Obtener la razón de grados de probabilidad  
coeficientes = coef(modelo\_logistico)  
razon\_encontrada = exp(coeficientes["Nivel\_plomo\_suelo"])  
razon\_encontrada

## Nivel\_plomo\_suelo   
## 1.002747

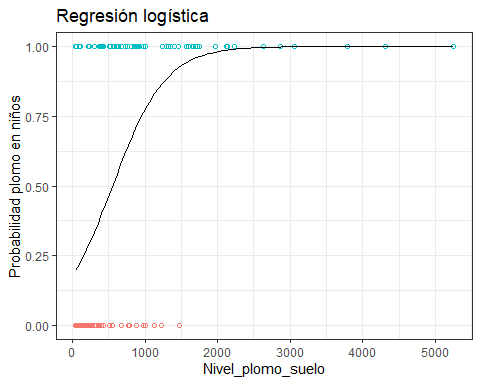
# Compara la razón de grados de probabilidad contra la que obtuvieron los autores (14.25)  
razon\_autores = 14.25  
# Realizar la prueba de significancia  
summary(modelo\_logistico)$coefficients["Nivel\_plomo\_suelo", "Pr(>|z|)"]

## [1] 4.601328e-07

La razón de grados de probabilidad obtenida es de 1.002, lo que indica que por cada diez veces que aumenta el nivel de plomo en la tierra, la proporción relativa de casos con respecto a los grupos de control aumenta en un 0.1. Este resultado difiere del obtenido por los autores del estudio (14.25). El valor de p obtenido es , lo que indica que el nivel de plomo en la tierra es un factor significativo para predecir los niveles altos de plomo en la sangre de los niños. El ecuación del modelo es el siguiente:

Las grafica del modelo es el siguiente:

library(ggplot2)  
ggplot(data = datos, aes(x = Nivel\_plomo\_suelo, y = Categoria\_del\_Individuo)) +  
 geom\_point(aes(color = as.factor(Categoria\_del\_Individuo)), shape = 1) +  
 stat\_function(fun = function(x){predict(modelo\_logistico,  
 newdata = data.frame(Nivel\_plomo\_suelo = x),  
 type = "response")}) +  
 theme\_bw() +  
 labs(title = "Regresión logística",  
 y = "Probabilidad plomo en niños") +  
 theme(legend.position = "none")



## Preguntas

1. La variable dependiente es la “Categoria\_del\_Individuo”.
2. La variable independiente es la “Nivel\_plomo\_suelo”.
3. Hipótesis nula: No hay relación entre el nivel de plomo en el suelo y la categoría del individuo. Hipótesis alternativa: Existe una relación entre el nivel de plomo en el suelo y la categoría del individuo.
4. La hipótesis nula es rechazada debido a que el valor de p obtenido es menor al nivel de significancia del 0.05 establecido. Esto indica que hay suficiente evidencia para concluir que el nivel de plomo en el suelo es un factor significativo para predecir los niveles altos de plomo en la sangre de los niños.
5. El objetivo más relevante es predecir los niveles altos de plomo en la sangre de los niños, ya que esto puede tener graves consecuencias en su salud. Aunque la estimación de parámetros es importante para comprender la relación entre las variables, la predicción tiene una mayor importancia práctica.
6. La población muestreada son niños expuestos al plomo en la ciudad donde se realizó el estudio.
7. La población objetivo son todos los niños expuestos al plomo, independientemente de su ubicación geográfica.
8. La variable “Nivel\_plomo\_suelo” está relacionada directamente con la “Categoria\_del\_Individuo”, ya que aumentos en la primera variable son predictores de una mayor proporción de casos en la segunda variable, al examinar la significación estadística de los coeficientes de la regresión y de las variables incluidas al ser significativo y positivo. Los datos no tienen una alta correlacion segun el Nab pero al tener los resultado del modelo de regresion logistica hay que tener en cuenta el primer enunciado.

r = cor(datos)  
nab = 1-det(r)  
nab

## [1] 0.2290155

1. La ecuación de regresión es: , donde “y” es la “Categoria\_del\_Individuo”, “x” es la “Nivel\_plomo\_suelo”, b0 es el intercepto y b1 es la pendiente. En este caso, la ecuación sería: .
2. Para realizar un analisis estadístico adicional de los valores a los betas del modelo logistico, se obtiene los intervalos de confianza de los coeficientes. Cada fila en la tabla se corresponde con un coeficiente del modelo (intercepto y NivelPlomoSuelo), y cada columna muestra una de las tres medidas de intervalo de confianza: “2.5 %” y “97.5 %” corresponden a la probabilidad de cola baja y alta de la distribución t de Student, respectivamente.

# Obtener los intervalos de confianza para los coeficientes (betas)  
intervalos\_confianza = confint(modelo\_logistico)

## Waiting for profiling to be done...

intervalos\_confianza

## 2.5 % 97.5 %  
## (Intercept) -2.21934712 -0.881028825  
## Nivel\_plomo\_suelo 0.00177278 0.003911804

Se puede utilizar los valores estimados y los límites inferior y superior de los intervalos de confianza para interpretar los resultados del modelo y hacer inferencias sobre la relación entre la variable independiente y la variable dependiente. En general, el intervalo de confianza proporciona una estimación útil de la precisión con la que se ha estimado cada coeficiente en el modelo de regresión logística. Para el caso del modelo encontrado deber ser mayor 95% confianza.

1. La variable “Categoria\_del\_Individuo” es categórica binaria (0 para control y 1 para caso), mientras que la variable “Nivel\_plomo\_suelo” es cuantitativa. La razón de grados de probabilidad (también conocido como odds ratio) es una medida de la asociación entre dos eventos. La variable sexo tambien es categórica binaria. diferentes valores de la variable independiente.
2. Con base en el punto (j) donde se encontro los intervalos de confianza de los betas. Los intervalos de confianza al 95% para los coeficientes del modelo son los siguientes:

* El coeficiente para la variable Intercepto tiene un intervalo de confianza entre -2.219 y -0.881. Esto significa que, en promedio, si el nivel de plomo en el suelo es cero, entonces la probabilidad de que un individuo esté en la categoría afectada es baja, y aumenta significativamente a medida que aumenta el nivel de plomo.
* El coeficiente para la variable Nivel\_plomo\_suelo tiene un intervalo de confianza entre 0.002 y 0.004. Esto significa que cuando el nivel de plomo en el suelo se incrementa en una unidad, la probabilidad de que un individuo esté en la categoría afectada aumenta en promedio entre 0.002 y 0.004.

En general, el modelo nos indica que el nivel de plomo en el suelo es un factor importante en la predicción de si un individuo estará o no en la categoría afectada. Los intervalos de confianza nos ayudan a tener una idea de la precisión de las estimaciones.

# Punto 2

En primer lugar, se realiza la carga de los datos que se almacenaron previamente en una hoja excel, para luego convertir a un dataframe:

library(readxl)  
olmos <- data.frame(read\_excel("dataset\_ejer\_2.xlsx"))  
dim(olmos)

## [1] 30 7

head(olmos)

## X1 X2 X3 X4 X5 X6 X7  
## 1 0.530 0.420 0.135 0.6770 0.2565 0.1415 0.210  
## 2 0.530 0.415 0.150 0.7775 0.2370 0.1415 0.330  
## 3 0.545 0.425 0.125 0.7680 0.2940 0.1495 0.260  
## 4 0.550 0.440 0.150 0.8945 0.3145 0.1510 0.320  
## 5 0.525 0.380 0.140 0.6065 0.1940 0.1475 0.210  
## 6 0.535 0.405 0.145 0.6845 0.2725 0.1710 0.205

## Preguntas

1. Realizar el analisis de componenetes principales (PCA) completo con gráficos de cargas.

Antes de realizar un analisis por PCA, hay que tener en cuenta que debe haber una alta correlacion entre los datos, porque de lo contrario no tiene sentido realizar este analisis.

r = det(cor(olmos))  
nab = 1-r  
nab

## [1] 0.9999999

Como se puede observar hay una alta correlacion de los datos ya que el . Ya con esta prueba es factible hacer este analisis de PCA. Se comienza con la creacion del modelo PCA, donde esta informacion relevante como varianza y correlacion.

# Análisis de componentes principales  
acp = princomp(olmos, cor = TRUE)  
summary(acp)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 2.5273682 0.53058381 0.44469355 0.28934876 0.161625395  
## Proportion of Variance 0.9125128 0.04021703 0.02825034 0.01196039 0.003731824  
## Cumulative Proportion 0.9125128 0.95272986 0.98098020 0.99294059 0.996672411  
## Comp.6 Comp.7  
## Standard deviation 0.136607246 0.0680557282  
## Proportion of Variance 0.002665934 0.0006616546  
## Cumulative Proportion 0.999338345 1.0000000000

#grafico de cargas de las variables  
vector\_propios = acp$loadings  
#cambio de signo  
acp$loadings[,2] = -acp$loadings[,2]  
acp$scores[,2] = -acp$scores[,2]  
#correlacion entre las componentes y las variables  
desviacion\_stand = acp$sdev  
correlacion = sweep(vector\_propios[1:7,1:7],2, desviacion\_stand, "\*" )  
round(correlacion,3)

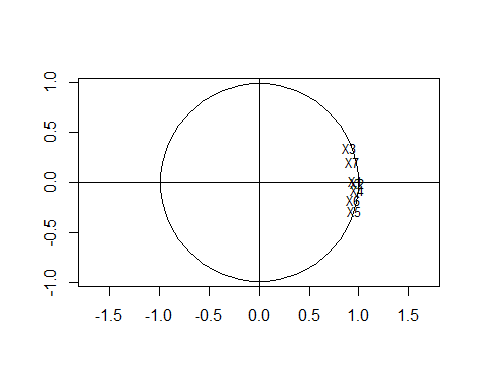
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  
## X1 0.968 0.011 0.050 0.240 0.045 0.037 0.004  
## X2 0.985 -0.006 0.103 0.055 -0.099 -0.084 0.006  
## X3 0.903 0.348 -0.246 -0.026 -0.043 0.032 -0.004  
## X4 0.989 -0.079 0.093 -0.057 0.009 0.012 -0.058  
## X5 0.951 -0.286 -0.008 -0.072 -0.051 0.072 0.024  
## X6 0.948 -0.172 -0.246 -0.031 0.076 -0.062 0.007  
## X7 0.940 0.206 0.235 -0.115 0.063 -0.005 0.023

Se realiza el grafico de cargas de en donde para identificar cuál variable tiene más contribución en el gráfico de cargas circulares, se debe observar la distancia de cada círculo al origen central del gráfico. Por lo cual todas las variables tienen un buen aporte de informacion.

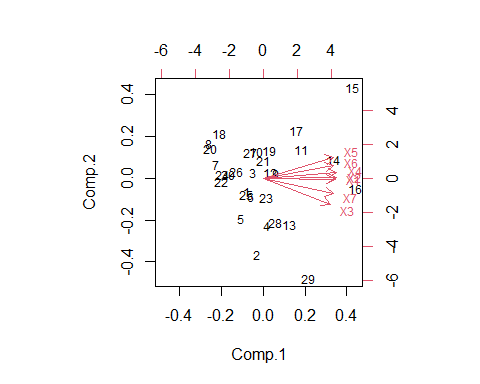
#grafico de cargas de las variables  
library(MASS)

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

eqscplot(correlacion[,1:2], xlim=c(-1,1), ylim=c(-1,1), type = "n")  
abline(h=0,v=0)  
text(correlacion[,1:2],labels=colnames(olmos),cex=0.8)  
symbols(0,0,circles=1, inches=FALSE, add=TRUE)



biplot(acp, cex=0.75)



1. Cuales variables presentan más similitud.

Segun la tabla, se observa que la variable con mayor similitud es X2 (diametro) y X4 (peso total), con un valor de correlación de 97%. Tambien con una correlacion del 96% se tiene relacion las siguientes parejas (X1,X2), (X4,X5)

#Variable con mayor similitud  
cor(olmos)

## X1 X2 X3 X4 X5 X6 X7  
## X1 1.0000000 0.9635989 0.8580583 0.9480612 0.9000948 0.8975352 0.8992660  
## X2 0.9635989 1.0000000 0.8614770 0.9784638 0.9323299 0.9053611 0.9370721  
## X3 0.8580583 0.8614770 1.0000000 0.8441517 0.7671737 0.8522890 0.8631614  
## X4 0.9480612 0.9784638 0.8441517 1.0000000 0.9656858 0.9301455 0.9413077  
## X5 0.9000948 0.9323299 0.7671737 0.9656858 1.0000000 0.9470957 0.8387932  
## X6 0.8975352 0.9053611 0.8522890 0.9301455 0.9470957 1.0000000 0.8076783  
## X7 0.8992660 0.9370721 0.8631614 0.9413077 0.8387932 0.8076783 1.0000000

1. Cuantos componentes explican la mayor variabilidad de los datos.

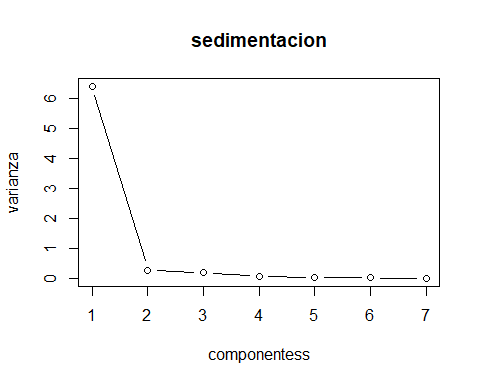
Para conocer cuántos componentes explican la mayor variabilidad de los datos, se usa la función summary() sobre el objeto pca. La salida muestra que el componente 1 explica el 91.25% de la variabilidad. Tambien en el grafico de sedimentacion se observo que era la unica componente al superar el 1, por lo cual solo se escoge esta componente.

summary(acp)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 2.5273682 0.53058381 0.44469355 0.28934876 0.161625395  
## Proportion of Variance 0.9125128 0.04021703 0.02825034 0.01196039 0.003731824  
## Cumulative Proportion 0.9125128 0.95272986 0.98098020 0.99294059 0.996672411  
## Comp.6 Comp.7  
## Standard deviation 0.136607246 0.0680557282  
## Proportion of Variance 0.002665934 0.0006616546  
## Cumulative Proportion 0.999338345 1.0000000000

Se realiza el grafico de sedimentacion en donde se escoge las variables que tengan una varianza mayor a 1.

#calculo de los valores propios  
val\_propios = acp$sdev^2  
plot(1:7, val\_propios, type= "b", xlab="componentess", ylab="varianza",   
main = "sedimentacion")



1. Que grupos de individuos presentan más similitud.

Para conocer los grupos de individuos que presentan mayor similitud se realiza la grafica de dispersion de los individuos y las variables. Segun el grafico se encuentra dos grupos divididos por una linea vertical en el punto 0. El gráfico muestra que las dos observaciones están relativamente cercanas entre sí.

library(psych)

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

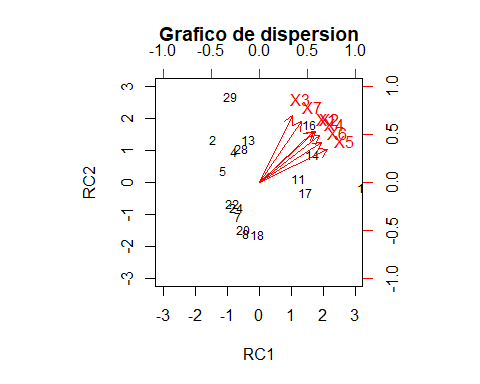
##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

acp\_varimax = principal(olmos, nfactors = 2, rotate = "varimax",scores = TRUE)  
round(cor(olmos, acp\_varimax$scores[,1:2]),3)#tabla de comunidades

## RC1 RC2  
## X1 0.705 0.664  
## X2 0.728 0.663  
## X3 0.428 0.868  
## X4 0.781 0.612  
## X5 0.893 0.434  
## X6 0.814 0.517  
## X7 0.552 0.789

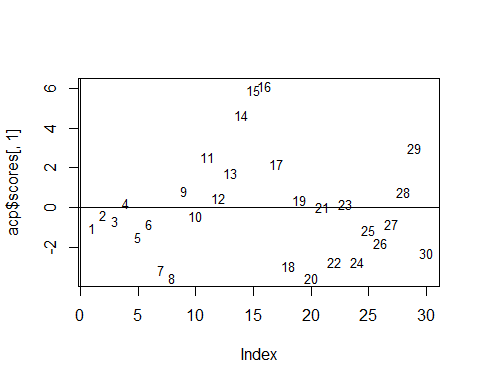
biplot(acp\_varimax, labels = rownames(olmos), cex= 0.75, main = "Grafico de dispersion")



1. Realizar un análisis de regresión lineal del componente 1 y sus respectivas variables de carga.

El siguiente grafico ayuda a ver el aporte que tienen cada individuo al componenetes 1.

plot(acp$scores[,1], type= "n",)  
abline(h=0, v= 0)  
text(acp$scores[,1],   
labels= rownames(olmos), cex=0.8)



Para realizar el análisis de regresión lineal del componente 1 y su respectiva variables de carga, se crea un data frame con las dos columnas.

#para la variable X1  
datos\_regresion = data.frame(acp$scores[,1], olmos$X1)  
names(datos\_regresion) = c('cargasX1', 'datosX1')  
model\_X1 = lm(datos\_regresion$datosX1~datos\_regresion$cargasX1)   
summary(model\_X1)

##   
## Call:  
## lm(formula = datos\_regresion$datosX1 ~ datos\_regresion$cargasX1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.056653 -0.010549 0.002421 0.009731 0.023188   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.541667 0.003133 172.91 <2e-16 \*\*\*  
## datos\_regresion$cargasX1 0.025209 0.001239 20.34 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01716 on 28 degrees of freedom  
## Multiple R-squared: 0.9366, Adjusted R-squared: 0.9343   
## F-statistic: 413.6 on 1 and 28 DF, p-value: < 2.2e-16

#para la varible X2  
datos\_regresion2 = data.frame(acp$scores[,1], olmos$X2)  
names(datos\_regresion2) = c('cargasX1', 'datosX2')  
model\_X2 = lm(datos\_regresion2$datosX2~datos\_regresion2$cargasX1)   
summary(model\_X2)

##   
## Call:  
## lm(formula = datos\_regresion2$datosX2 ~ datos\_regresion2$cargasX1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.015248 -0.006064 -0.002282 0.006445 0.029078   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.4265000 0.0018978 224.7 <2e-16 \*\*\*  
## datos\_regresion2$cargasX1 0.0223745 0.0007509 29.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01039 on 28 degrees of freedom  
## Multiple R-squared: 0.9694, Adjusted R-squared: 0.9683   
## F-statistic: 887.9 on 1 and 28 DF, p-value: < 2.2e-16

#para la varible X3  
datos\_regresion3 = data.frame(acp$scores[,1], olmos$X3)  
names(datos\_regresion3) = c('cargasX1', 'datosX3')  
model\_X3 = lm(datos\_regresion3$datosX3~datos\_regresion3$cargasX1)   
summary(model\_X3)

##   
## Call:  
## lm(formula = datos\_regresion3$datosX3 ~ datos\_regresion3$cargasX1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0202446 -0.0071824 -0.0005052 0.0072554 0.0274423   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.1420000 0.0020455 69.42 < 2e-16 \*\*\*  
## datos\_regresion3$cargasX1 0.0089836 0.0008093 11.10 9.16e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0112 on 28 degrees of freedom  
## Multiple R-squared: 0.8148, Adjusted R-squared: 0.8082   
## F-statistic: 123.2 on 1 and 28 DF, p-value: 9.156e-12

Como se puede ver hay una alta correlacion entre los datos y el nuevo componente 1 PCA, el cual explica el 91% de los datos, por lo cual cuando se realiza regresion lineal con cada una de las variables de los datos da una alta correlacion y con un mayor al 89% a excepcion del X3 que da de un 81%. Solo se hizo modelos lm() hasta la variable 3 los modelos. En conclusión, el análisis de regresión lineal del primer componente principal y sus respectivas variables de carga permite explicar en gran medida la variabilidad de los datos y evidencia la influencia de la variables X1 en la formación del componente. La carga de una variable en un componente principal indica cuánto contribuye esa variable a la varianza explicada por ese componente. Al realizar la regresión lineal, puedes analizar cómo la variable original está relacionada con el componente principal y si existe una relación lineal significativa entre ellas.

# Punto 3

Para la solucion del ejercicio de distancias entre los datos y correlacion, primero se carga los datos a un excel para ser convertido a un dataframe:

library(readxl)  
datos = data.frame(read\_excel("dataset\_ejer\_3.xlsx"))  
datos

## X1 X2 X3 X4 X5 X6 X7  
## 1 8 98 7 2 12 8 2  
## 2 7 107 4 3 9 5 3  
## 3 7 103 4 3 5 6 3  
## 4 10 88 5 2 8 15 4  
## 5 6 91 4 2 8 10 3  
## 6 8 90 5 2 12 12 4  
## 7 9 84 7 4 12 15 5  
## 8 5 72 6 4 21 14 4  
## 9 7 82 5 1 11 11 3  
## 10 8 64 5 2 13 9 4

## Preguntas

1. Calcular la covarianza.

La covarianza es una medida de cómo dos variables se mueven en relación entre sí. Un valor positivo indica que las variables se mueven de manera similar, mientras que un valor negativo indica que las variables se mueven de manera opuesta. En este caso, la matriz de covarianza muestra que la mayoría de las variables están positivamente correlacionadas entre sí.

# Calcular la covarianza  
covarianza = cov(datos)  
covarianza

## X1 X2 X3 X4 X5 X6  
## X1 2.0555556 1.05555556 0.4444444 -0.27777778 -2.277778 1.5000000  
## X2 1.0555556 175.87777778 -4.5333333 0.05555556 -37.433333 -26.2777778  
## X3 0.4444444 -4.53333333 1.2888889 0.33333333 2.755556 2.0000000  
## X4 -0.2777778 0.05555556 0.3333333 0.94444444 1.500000 0.6111111  
## X5 -2.2777778 -37.43333333 2.7555556 1.50000000 18.322222 6.8333333  
## X6 1.5000000 -26.27777778 2.0000000 0.61111111 6.833333 12.7222222  
## X7 0.3888889 -6.16666667 0.2222222 0.38888889 1.166667 2.1666667  
## X7  
## X1 0.3888889  
## X2 -6.1666667  
## X3 0.2222222  
## X4 0.3888889  
## X5 1.1666667  
## X6 2.1666667  
## X7 0.7222222

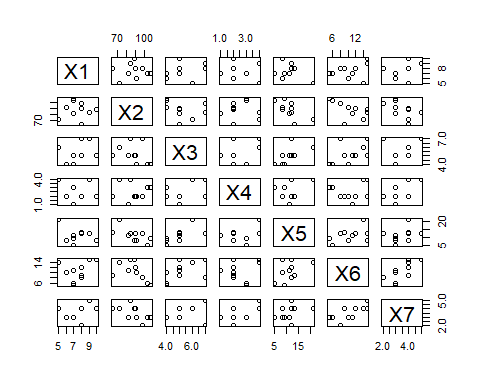
1. Calcular la matriz de correlación.

La matriz de correlación es una medida de la fuerza y dirección de la relación lineal entre dos variables. Los valores van desde -1 (correlación negativa perfecta) a 1 (correlación positiva perfecta), con 0 indicando que no hay relación. Los datos muestran una fuerte correlación positiva entre X6(O3) y X7(HC) y una correlacion alta negativa entre X2(Radiación solar) y X5(N02), mientras que las otras variables no tienen una correlación tan fuerte. En el Dispersograma se puede ver como un especie de linealidad de las variables mencionadas.

# Calcular la matriz de correlación  
correlacion = cor(datos)  
correlacion

## X1 X2 X3 X4 X5 X6  
## X1 1.00000000 0.055515084 0.2730519 -0.199363056 -0.3711570 0.2933224  
## X2 0.05551508 1.000000000 -0.3010956 0.004310561 -0.6594228 -0.5555225  
## X3 0.27305186 -0.301095614 1.0000000 0.302122310 0.5670383 0.4939020  
## X4 -0.19936306 0.004310561 0.3021223 1.000000000 0.3605902 0.1762993  
## X5 -0.37115701 -0.659422776 0.5670383 0.360590248 1.0000000 0.4475711  
## X6 0.29332235 -0.555522511 0.4939020 0.176299258 0.4475711 1.0000000  
## X7 0.31917253 -0.547153983 0.2303267 0.470870956 0.3207172 0.7147846  
## X7  
## X1 0.3191725  
## X2 -0.5471540  
## X3 0.2303267  
## X4 0.4708710  
## X5 0.3207172  
## X6 0.7147846  
## X7 1.0000000

#Dispersograma  
plot(datos)



1. Calcular las distancias euclidianas y mahalanobis

La distancia euclidiana es una medida de la distancia entre dos puntos en un espacio multidimensional. En este caso, muestra la distancia entre las 10 filas de la tabla en un espacio de 7 dimensiones. La distancia mahalanobis es similar, pero tiene en cuenta la covarianza entre las variables. La distancia euclidiana y mahalanobis producen resultados diferentes en este conjunto de datos pero de la misma forma sirven para clasificar un conjunto de datos.

# Media   
datos\_media = colMeans(datos)  
  
# Calcular las distancias euclidianas  
dist\_euclidianas\_indiv = dist(datos, method = "euclidean")  
dist\_euclidianas\_indiv

## 1 2 3 4 5 6 7  
## 2 10.535654   
## 3 9.486833 5.744563   
## 4 13.304135 21.771541 18.083141   
## 5 9.110434 16.852300 13.076697 7.211103   
## 6 9.380832 18.734994 16.062378 5.744563 5.196152   
## 7 16.093477 25.612497 22.561028 6.480741 10.770330 7.416198   
## 8 28.478062 38.209946 35.930488 21.354157 23.579652 20.566964 15.620499  
## 9 16.522712 25.884358 22.516660 8.485281 9.695360 8.306624 6.480741  
## 10 34.146742 43.416587 39.974992 25.317978 27.586228 26.191602 21.142375  
## 8 9  
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9 14.966630   
## 10 12.922848 18.303005

# Calcular las distancias mahalanobis  
dist\_mahalanobis = sqrt(mahalanobis(datos,datos\_media,covarianza))  
dist\_mahalanobis

## [1] 2.773972 2.379065 2.122425 2.839088 2.095724 2.326418 2.749814 2.807176  
## [9] 1.996978 2.801089

Para la distancia con uso de la media:

# Calcular las distancias euclidianas  
dist\_euclidianas\_media = dist(rbind(datos\_media,datos))  
dist\_euclidianas\_media

## 1 2 3 4 5 6 7  
## 2 10.726602   
## 3 20.041457 10.535654   
## 4 16.960542 9.486833 5.744563   
## 5 6.054750 13.304135 21.771541 18.083141   
## 6 4.864155 9.110434 16.852300 13.076697 7.211103   
## 7 2.874022 9.380832 18.734994 16.062378 5.744563 5.196152   
## 8 6.801470 16.093477 25.612497 22.561028 6.480741 10.770330 7.416198  
## 9 19.299223 28.478062 38.209946 35.930488 21.354157 23.579652 20.566964  
## 10 6.153048 16.522712 25.884358 22.516660 8.485281 9.695360 8.306624  
## 11 24.038719 34.146742 43.416587 39.974992 25.317978 27.586228 26.191602  
## 8 9 10  
## 2   
## 3   
## 4   
## 5   
## 6   
## 7   
## 8   
## 9 15.620499   
## 10 6.480741 14.966630   
## 11 21.142375 12.922848 18.303005

# Calcular las distancias mahalanobis  
dist\_mahalanobis\_media = apply(datos, 1, function(xi) {  
 sqrt(mahalanobis(xi, datos\_media, covarianza))  
})  
dist\_mahalanobis\_media

## [1] 2.773972 2.379065 2.122425 2.839088 2.095724 2.326418 2.749814 2.807176  
## [9] 1.996978 2.801089

# Punto 4

Para este ejercicio se pide un hacer una regresion logistica de una tabla que relaciona los estados nutricional con su desempeño escolar, por ello se empieza replicando los datos de la tabla.

# Desempeño\_escolar : deficiente -> 0 satisfactorio -> 1  
# Estado nutricional : deficiente -> 0 bueno -> 1  
  
desemp\_esc = c(rep(0,105), rep(1,80),rep(0,15), rep(1,300))  
estado\_nutric = c(rep(0,105), rep(0,80),rep(1,15), rep(1,300))  
datos = data.frame(desemp\_esc, estado\_nutric)  
table(datos)

## estado\_nutric  
## desemp\_esc 0 1  
## 0 105 15  
## 1 80 300

Se genera el modelo de regresion logistica para los datos de la tabla. se toma como variable dependiente el estado de nutricion y como independiente el desempeño escolar.

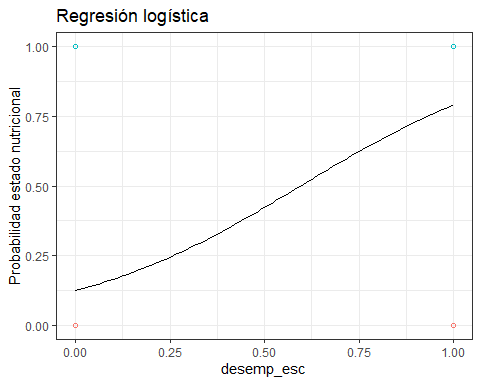
modelo = glm(estado\_nutric~desemp\_esc,  
 data = datos,  
 family = binomial(link = "logit")  
)  
summary(modelo)

##   
## Call:  
## glm(formula = estado\_nutric ~ desemp\_esc, family = binomial(link = "logit"),   
## data = datos)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7653 -0.5168 0.6876 0.6876 2.0393   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.9459 0.2760 -7.05 1.79e-12 \*\*\*  
## desemp\_esc 3.2677 0.3034 10.77 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 658.96 on 499 degrees of freedom  
## Residual deviance: 481.56 on 498 degrees of freedom  
## AIC: 485.56  
##   
## Number of Fisher Scoring iterations: 4

Para interpretar los resultados del modelo, es importante mirar la estimación del coeficiente de regresión. En este caso, el coeficiente de regresión para la variable predictora, desemp\_esc, es positivo (3.2677), lo que sugiere que a medida que aumenta el nivel de desempeño escolar, la probabilidad de estar en el estado de nutrición deseado (estado\_nutric = bueno) aumenta. Además, el valor p asociado (2e-16) con la variable predictora es menor que 0.05, lo que indica que la relación entre la variable predictora y la variable de respuesta es estadísticamente significativa.

El numero de iteraciones fueron 4 y la medidad de calidad relativa AIC es de 485.5 para una comparacion con otros modelos. El grafico del modelo es el siguiente:

library(ggplot2)  
ggplot(data = datos, aes(x = desemp\_esc, y = estado\_nutric)) +  
 geom\_point(aes(color = as.factor(estado\_nutric)), shape = 1) +  
 stat\_function(fun = function(x){predict(modelo,  
 newdata = data.frame(desemp\_esc = x),  
 type = "response")}) +  
 theme\_bw() +  
 labs(title = "Regresión logística",  
 y = "Probabilidad estado nutricional") +  
 theme(legend.position = "none")

 En resumen, el análisis de resultados muestra que la variable predictora, desemp\_esc, parece ser un predictor significativo del estado de nutrición deseado. Sin embargo, como con cualquier modelo estadístico, hay limitaciones y supuestos que se deben tener en cuenta al interpretar los resultados.

# Punto 5

Para resolver este problema, se utiliza la matriz de transición de probabilidad. Primero, se define los estados como 1, 2 y 3 para Tigo, Claro y Movistar, respectivamente. Luego, se crea una matriz de transición de probabilidad que represente las probabilidades de transición de un estado a otro. Para luego crear una matriz de Markov que facilita la transicion de probabilidad de un estado a otro.

## Preguntas

1. Se quiere conocer cuál es la probabilidad de permanencia en la primera y segunda corrida.

# Definición de las probabilidades actuales y estados  
p\_act = c(0.4, 0.25, 0.35)  
estados = c("Tigo", "Claro", "Movistar")  
  
# Definición de la matriz de transición  
trans\_mat = matrix(c(0.6, 0.2, 0.2, 0.3, 0.5, 0.2, 0.3, 0.3, 0.4), nrow = 3, byrow = TRUE)  
  
# Markov  
library(markovchain)

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

## Package: markovchain  
## Version: 0.9.1  
## Date: 2023-01-20  
## BugReport: https://github.com/spedygiorgio/markovchain/issues

mc = new("markovchain", states = estados, transitionMatrix = trans\_mat, name = "Compañías de Telefonía")  
summary(mc)

## Compañías de Telefonía Markov chain that is composed by:   
## Closed classes:   
## Tigo Claro Movistar   
## Recurrent classes:   
## {Tigo,Claro,Movistar}  
## Transient classes:   
## NONE   
## The Markov chain is irreducible   
## The absorbing states are: NONE

# Probabilidad primera corrida  
p\_1 = p\_act \* mc  
p\_1

## Tigo Claro Movistar  
## [1,] 0.42 0.31 0.27

# Probabilidad segunda corrida  
p\_2 = p\_1\*(mc)^2  
p\_2

## Tigo Claro Movistar  
## [1,] 0.4278 0.3214 0.2508

Se puede observar que la probabilidad de permanencia en Claro aumenta ligeramente en la segunda corrida, mientras que la probabilidad de permanencia en Movistar disminuye y la probabilidad de permanencia en Tigo se mantiene relativamente estable. La distribucion estacionaria hacia donde tiende cada iteracion es

# Para hallar la distribucion estacionaria   
dist\_estacion = steadyStates(mc)  
dist\_estacion

## Tigo Claro Movistar  
## [1,] 0.4285714 0.3214286 0.25

1. Realizar el grafico El gráfico generado muestra el grafo de la matriz de transicion con las probabilidades de transición como aristas y el nombre de cada estado como vertice.

plot(mc)

