Trabajo-R-NOTA10.pdf

Willy_Jojo

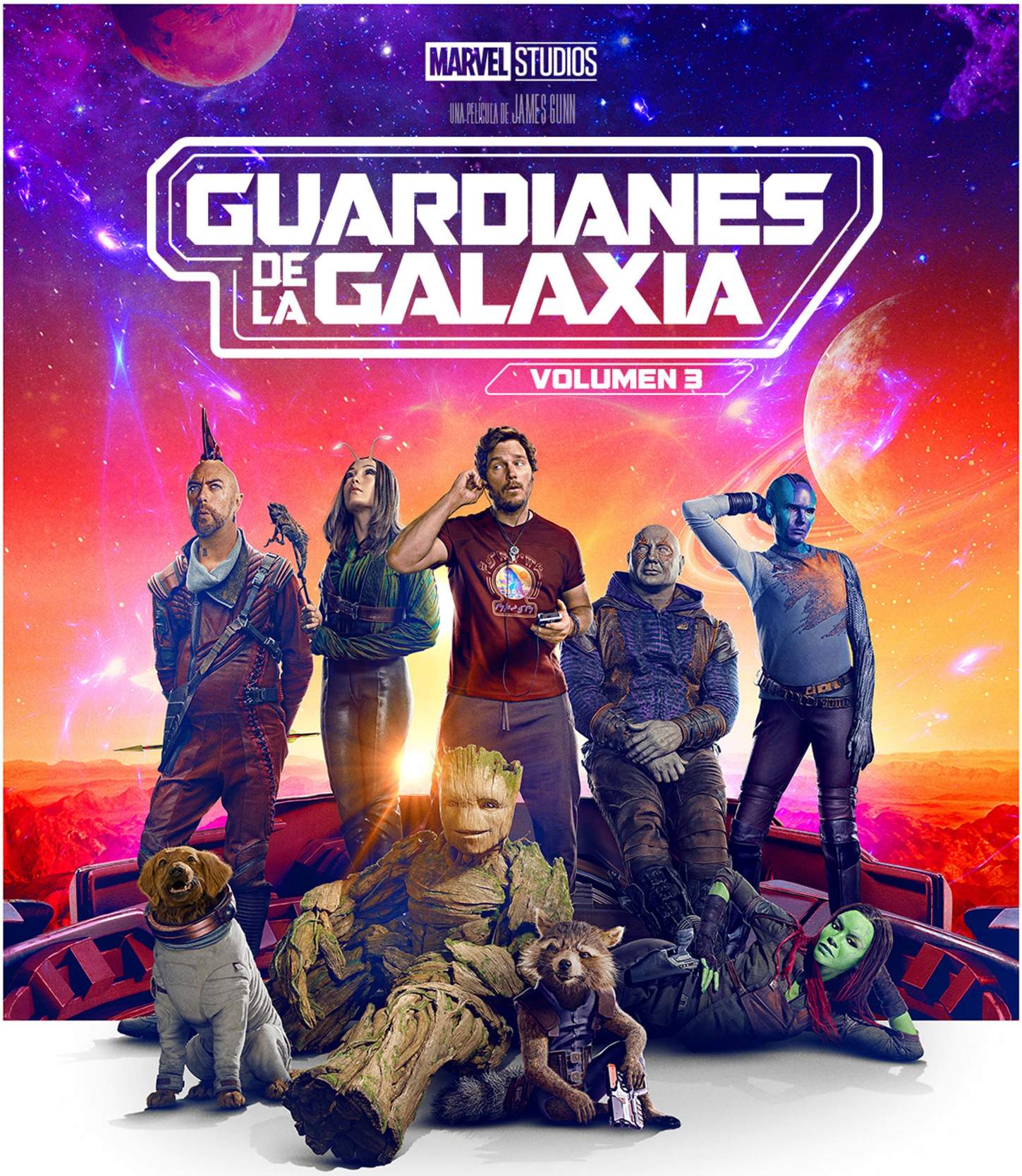
Metodos Estadisticos para la Computacion

1º Grado en Ingeniería del Software

Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

```
#Trabajo R, Métodos Estadísticos para la Computación
#Realizado por: GUillermo Pichaco Panal (75913339E)

#-------------------------------------------------------------------------------

#1.Carga en memoria el fichero CSV como tibble, asegurándote de que las
#variables cualitativas sean leídas como factores.
library(tidyverse)  #Importamos la librería tidyverse, muy útil para muchas funciones

#Leemos y cargamos el csv, además de leer las variables cualitativas como factores

dfTrabajoR <-
read_csv("C:/Users/lacas/OneDrive/Documents/Estadistica/TrabajoR/18447.csv",
                col_types=cols(.default=col_double(), sexo=col_factor(),
                               dietaEsp=col_factor(), nivEstPad=col_factor(),
                               nivEstudios=col_factor(), nivIngresos=col_factor()
                ))

#view(dfTrabajoR)  #Por si queremos echar un primer vistazo

#-------------------------------------------------------------------------------

#2.Construye una nueva columna llamada IMC que sea igual al peso dividido por la
#altura al cuadrado. La variable explicada será IMC, las variables explicatorias
#serán el resto de 12 variables exceptuando peso y altura.

#Creamos la columna IMC con los datos
IMC <- dfTrabajoR$peso/dfTrabajoR$altura^2

#Añadimos la columna IMC a nuestro dataset con el comando mutate
dfTrabajoR <- mutate(dfTrabajoR, IMC)
```

```
#-------------------------------------------------------------------------

#3.Elimina completamente las filas que tengan algún valor NA en una de sus columnas.

#Eliminamos las filas con NA con na.omit(dataset)
dfTrabajoR <- na.omit(dfTrabajoR)

#-------------------------------------------------------------------------

#4.Calcula las medias y desviaciones típicas (no cuasidesviación) de todas las
# variables numéricas.

#Ya que piden solo variables numéricas usamos "keep(is.numeric)" y para usar mean
#(función que calcula la media) en todas ellas, usamos el comando "map_dbl(mean)"
medias <- dfTrabajoR %>% keep(is.numeric) %>% map_dbl(mean)

#Para la desviación típica además habrá que multiplicarlo por el número de
#filas-1 dividido entre el número de filas.
desvtípicas <- dfTrabajoR %>% keep(is.numeric) %>% map_dbl(sd)
desvtípicas <- desvtípicas * sqrt((nrow(dfTrabajoR)-1)/nrow(dfTrabajoR))

#-------------------------------------------------------------------------

#5.Calcula los coeficientes de regresión y el coeficiente de determinación
#para las 12 regresiones lineales unidimensionales.

#Primero guardamos las regresiones lineales unidimensionales en una variable
regUnidim <- names(dfTrabajoR[4:15])

#Definimos la funcion para calcular el coeficiente de regresión usando un ajuste lineal "lm"
```

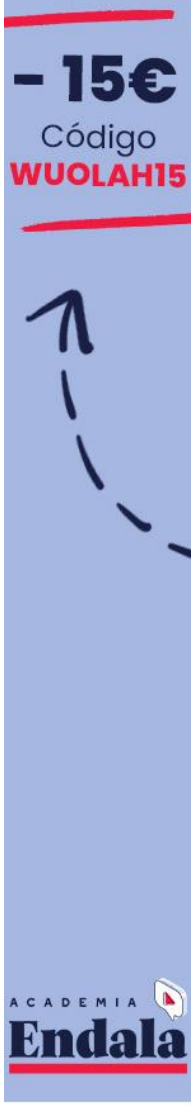


YA EN CINES

COMPRAR ENTRADAS

```
coefReg <- function(df,y,x){
  mod<-lm(str_c(y,"~",x),df)
  summary(mod)$coefficients[2]
}

#Y aquí para calcular el coeficiente de determinación usando un ajuste lineal "lm"
R2 <- function(df,y,x){
  mod<-lm(str_c(y,"~",x),df)
  summary(mod)$r.squared
}

#Calculamos ambos coeficientes utilizando las funciones previamente definidas
#utilizando como parámetro de entrada la variable "regUnidim"
pendientes <- regUnidim%>%map_dbl(coefReg,df=dfTrabajoR,y="IMC")

coefDet <- regUnidim %>% map_dbl(R2, df=dfTrabajoR, y="IMC")

#------------------------------------------------------------------------------

#6.Representa los gráficos de dispersión en el caso de variables numéricas y los
#box-plots en el caso de variables cualitativas. En el caso de las variables
#numéricas (y sólo en ese caso) el gráfico debe tener sobreimpresa la recta de
#regresión simple correspondiente.


#Definimos una función que calcula un ajuste lineal de 2 variables
ajusteLineal <- function(df, y, x){
  list(x=x, y=y, mod=lm(str_c(y, "~", x), df))
}

#Función que crea los gráficos y los almacena en una dirección
```

```
#(he escogido esos colores por gusto, pero se pueden cambiar)
modelDrawer <- function(mod){

jpeg(str_c("C:/Users/lacas/OneDrive/Documents/Estadistica/TrabajoR/graphs/",mod$x,".jpeg"
))
 if(is.numeric(dfTrabajoR[[mod$x]])){
  plot(dfTrabajoR[[mod$x]],
dfTrabajoR[[mod$y]],main="Gráfica",xlab=mod$x,ylab=mod$y,col="grey")
  abline(mod$mod, col="red")
 }else{
  boxplot(formula=dfTrabajoR[[mod$y]]~dfTrabajoR[[mod$x]], xlab=mod$x, ylab="IMC",
col="grey")
 }
 dev.off()
}

#Creamos la función que crea los gráficos de dispersión ayudándonos con el "walk()"
mods<-regUnidim%>%map(~ajusteLineal(dfTrabajoR,"IMC",.))
mods %>% walk(modelDrawer)

#------------------------------------------------------------------------------

#7.Separa el conjunto original de datos en tres conjuntos de entrenamiento,
#test y validación en las proporciones 60%, 20% y 20%.

#Función para separar el set original en tres: entrenamiento, test y validación

separar <- function(df, p1, p2) {
 rDf <- 1:nrow(df)
 rTrain <- sample(rDf, p1 * length(rDf))
 rTemp <- setdiff(rDf, rTrain)
 rTest <- sample(rTemp, p2 * length(rDf))
```



Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

```
  rValid <- setdiff(rTemp, rTest)
  list(train=df[rTrain,], test=df[rTest,], valid=df[rValid,])
}

setSeparados <- separar(dfTrabajoR,.6,.2)

#-------------------------------------------------------------------------------

#8.Selecciona cuál de las 12 variables sería la que mejor explica la variable
#IMC de manera individual, entrenando con el conjunto de entrenamiento y
#testeando con el conjunto de test.



#Creamos la función que calcula el ajuste siendo x un vector
linearAdjust <- function(df, y, x) {
  lm(str_c(y, "~", str_c(x, collapse="+")), df)
}

#Aqui creamos una para calcular el ajuste de R2 ajustado
calcR2ajst <- function(df, mod, y) {
  MSE <- mean((df[[y]] - predict.lm(mod, df)) ^ 2)
  varY <- mean(df[[y]] ^ 2) - mean(df[[y]]) ^ 2
  R2 <- 1 - MSE / varY
  ajR2 <- 1 - (1- R2) * (nrow(df) - 1) / (nrow(df) - mod$rank)
  ajR2
}

#Aquí no devolvemos el modelo, si no el coef. de determinación directamente
calcModR2 <- function(dfTrain, dfTest, y, x) {
  mod <- linearAdjust(dfTrain, y, x)
  calcR2ajst(dfTest, mod, y)
```

```
}

#Calculamos los R2 ajustados de las 12 regresiones
AjstR2 <- regUnidim %>%
map_dbl(calcModR2,dfTrain=setSeparados$train,dfTest=setSeparados$test,y="IMC")

#Aquí calculamos la mejor variable que explique el IMC
x <- which.max(AjstR2)
mejorVar <- regUnidim[x]
#Dicha variable será entonces:
mejorVar
#Con valor:
AjstR2[x]

#------------------------------------------------------------------------------

#9.Selecciona un modelo óptimo lineal de regresión, entrenando en el conjunto de
#entrenamiento, testeando en el conjunto de test el coeficiente de determinación
#ajustado y utilizando una técnica progresiva de ir añadiendo la mejor variable.

#Buscamos el mejor ajuste lineal, añadiendo gradualmente la mejor variable (de ahí
#el repeat_if()break)
findBestAjst <- function(dfTrain, dfTest, varPos) {
  mejorVars <- character(0)
  ajR2 <- 0
  repeat {
    ajR2v <- map_dbl(varPos, ~calcModR2(dfTrain, dfTest, "IMC", c(mejorVars, .)))
    i <- which.max(ajR2v)
    ajR2M <- ajR2v[i]
    if (ajR2M <= ajR2) break
```

```
  cat(sprintf("%1.4f %s\n", ajR2M, varPos[i]))
  ajR2 <- ajR2M
  mejorVars <- c(mejorVars, varPos[i])
  varPos <- varPos[-i]
 }
 mod <- linearAdjust(dfTrain, "IMC", mejorVars)
 list(vars=mejorVars, mod=mod)
}

#Una vez hallamos el mejor R2 ajustado, mostramos en una lista las mejores
#variables junto con sus respectivos coeficientes
ajr2v <- regUnidim %>% map_dbl(calcModR2, dfTrain=setSeparados$train,
dfTest=setSeparados$test, y="IMC")
MejorAjuste <- findBestAjst(setSeparados$train,setSeparados$test,regUnidim)

#-------------------------------------------------------------------------------

#10.Evalúa el resultado en el conjunto de validación.

#Validamos el modelo utilizando el set de validación y la función "calcR2ajst"
calcR2ajst(setSeparados$valid,MejorAjuste$mod,y="IMC")
```