Taller 4. Curso 2019-2020

Pon tu nombre aquí

Contents

Enunciados

Problema 1: Contraste de parámetros de dos muestras.

Queremos comparar los tiempos de realización de un test entre estudiantes de dos grados G1 y G2, y determinar si es verdad que los estudiantes de G1 emplean menos tiempo que los de G2. No conocemos σ_1 y σ_2. Disponemos de dos muestras independientes de cuestionarios realizados por estudiantes de cada grado, $n_1 = n_2 = 40$.

Los datos están en http://bioinfo.uib.es/~recerca/MAT2/NotasTestGrado/, en dos ficheros `grado1.txt` y `grado2.txt`.

Calculamos las medias y las desviaciones típicas muestrales de los tiempos empleados para cada muestra. Los datos obtenidos se resumen en la siguiente tabla:

$$\begin{array}{llll} n_1 & = 40, & n_2 & = 40 \\ \overline{x}_1 & = 9.7892076, & \overline{x}_2 & = 11.3848297 \\ \tilde{s}_1 & = 1.2011633, & \tilde{s}_1 & = 1.5787924 \end{array}$$

Se pide:

1. Contrastad si hay evidencia de que las notas medias son distintas entre los dos grupos. Tenéis que hacer el contraste adecuado con funciones de `R` y resolver el contrate con el p-valor, **justificando la elección del mismo**.(*1.5 punto*)
2. Calculad e interpretar los intervalos de confianza para la diferencia e medias asociados al test anterior.(*1 punto*)

Problema 2: Compararción de las tasas de interés para la compra de choches entre seis ciudades.

Consideremos el `data set` newcar.dat de *Hoaglin, D., Mosteller, F., and Tukey, J. (1991). Fundamentals of Exploratory Analysis of Variance. Wiley, New York, page 71.*

Este data set contiene el dos columnas:

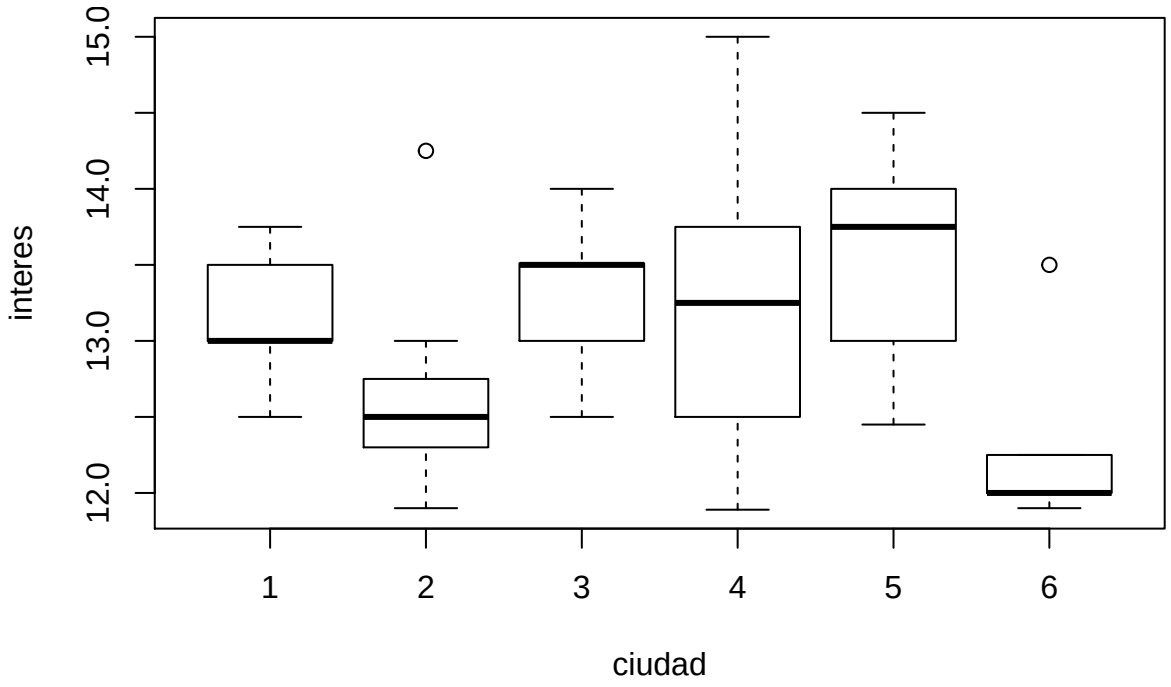
- Rate (interes): tasa de interés en la compra de coches a crédito

- City (ciudad) : la ciudad en la que se observó la tasa de interés para distintos concesionarios (codificada a enteros). Tenemos observaciones de 6 ciudades.

```
datos_interes=read.table("https://www.itl.nist.gov/div898/education/anova/newcar.dat",
                         skip=25)
names(datos_interes)=c("interes","ciudad")
str(datos_interes)
```

```
## 'data.frame':    54 obs. of  2 variables:
##  $ interes: num  13.8 13.8 13.5 13.5 13 ...
##  $ ciudad : int  1 1 1 1 1 1 1 1 1 2 ...
```

```
boxplot(interes~ciudad,data=datos_interes)
```



Se pide:

1. Comentad las líneas de código anteriores así como el resultado que se observa en el diagrama de caja. (*0.5 puntos*)
2. Se trata de contrastar si hay evidencia de que la tasas medias de interés por ciudades son distintas. Definid el ANOVA que contrasta esta hipótesis y especificar qué condiciones deben cumplir las muestras para poder aplicar el ANOVA. (*0.5 punto*)
3. Comprobad si se cumplen o no las condiciones del ANOVA. Justificad las conclusiones. (*0.5 punto*)
4. Realizad el contraste de ANOVA (se cumplan las condiciones o no) y redactar adecuadamente la conclusión. Tenéis que hacerlo con funciones de `R`. (*0.5 punto*)
5. Se acepte o no la igualdad de medias realizar las comparaciones dos a dos con ajustando los p-valor tanto por Bonferroni como por Holm al nivel de siginificación $\alpha = 0.1$. Redactad las conclusiones que se obtienen de las mismas. (*0.5 punto*)

Problema 3: Bondad de ajuste. La ley de Benford

La ley de Benford es una distribución discreta que siguen las frecuencias de los primero dígitos significativos (de 1 a 9) de algunas series de datos curiosas.

Sea una v.a. X con dominio $D_X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ diremos que sigue una ley de Benford si

$$P(X = x) = \log_{10}\left(1 + \frac{1}{x}\right) \text{ para } x \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\,.$$

Concretamente

	Dígito 1	Dígito 2	Dígito 3	Dígito 4	Dígito 5	Dígito 6	Dígito 7	Dígito 8	Dígito 9
prob	0.30103	0.1760913	0.1249387	0.09691	0.0791812	0.0669468	0.0579919	0.0511525	0.0457575

En general esta distribución se suele encontrar en tablas de datos de resultados de observaciones de funciones científicas, contabilidades, cocientes de algunas distribuciones ...

Por ejemplo se dice que las potencias de números enteros siguen esa distribución. Probemos con las potencias de 2. El siguiente código calcula las potencias de 2 de 1 a 1000 y extrae los tres primeros dígitos.

```
# R pasa los enteros  muy grande a reales. Para nuestros propósitos
# es sufiente para extraer los tres primeros dígitos.
muestra_pot_2=as.character(2^c(1:1000))
head(muestra_pot_2)
```

```
## [1] "2"  "4"  "8"  "16" "32" "64"
```

```
tail(muestra_pot_2)
```

```
## [1] "3.34846439745709e+299" "6.69692879491417e+299" "1.33938575898283e+300"
## [4] "2.67877151796567e+300" "5.35754303593134e+300" "1.07150860718627e+301"
```

```
#los pasamos a character
muestra_pot_2=as.character(muestra_pot_2)
head(muestra_pot_2)
```

```
## [1] "2"  "4"  "8"  "16" "32" "64"
```

```
tail(muestra_pot_2)
```

```
## [1] "3.34846439745709e+299" "6.69692879491417e+299" "1.33938575898283e+300"
## [4] "2.67877151796567e+300" "5.35754303593134e+300" "1.07150860718627e+301"
```

```
#sustituimos los . por nada y extraemos los ters primeros dígitos
aux=gsub("[.]","",muestra_pot_2)
#Construimos un data frame con tres columnas que nos dan el primer,
#segudo y tercer dígito respectivamente.
df_digitos=data.frame(primer_digito=as.integer(substring(aux, 1, 1)),
                      segundo_digito=as.integer(substring(aux, 2, 2)),
                      tercer_digito=as.integer(substring(aux, 3, 3)))
head(df_digitos)
```

```
##   primer_digito segundo_digito tercer_digito
## 1             2             NA            NA
## 2             4             NA            NA
## 3             8             NA            NA
## 4             1              6            NA
## 5             3              2            NA
## 6             6              4            NA
```

Se pide

1. Contrastad con un test χ^2 que el primer dígito sigue una ley de Benford. Notad que el primer dígito no puede ser 0. (*0.5 punto*)
2. Contrastad con un test χ^2 que el segundo dígito sigue una ley de uniforme discreta. Notad que ahora si puede ser 0. (*0.5 punto*)
3. Contrastad con un test χ^2 que el tercer dígito sigue una ley de uniforme discreta. Notad que ahora si puede ser 0. (*0.5 punto*)
4. Dibujad con `R` para los apartados 1 y 2 los diagramas de frecuencias esperados y observados. Comentad estos gráficos (*1 punto*)

Problema 4: Regresión puntuacions heptatlón

El `dataset` heptatlón del paquete `HSAUR` contiene los resultados de las siete pruebas olímpicas de heptatlón y la puntuación ponderada de las pruebas en la variable `score` Los resultados de las pruebas se miden en tiempo o en distancia dependiendo del tipo de prueba.

```
list.of.packages <- c("HSAUR")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
library("HSAUR")
```

```
## Loading required package: tools
```

```
data(heptathlon)
str(heptathlon)
```

```
## 'data.frame':    25 obs. of  8 variables:
##  $ hurdles : num  12.7 12.8 13.2 13.6 13.5 ...
##  $ highjump: num  1.86 1.8 1.83 1.8 1.74 1.83 1.8 1.8 1.83 1.77 ...
##  $ shot    : num  15.8 16.2 14.2 15.2 14.8 ...
##  $ run200m : num  22.6 23.6 23.1 23.9 23.9 ...
##  $ longjump: num  7.27 6.71 6.68 6.25 6.32 6.33 6.37 6.47 6.11 6.28 ...
##  $ javelin : num  45.7 42.6 44.5 42.8 47.5 ...
##  $ run800m : num  129 126 124 132 128 ...
##  $ score   : int  7291 6897 6858 6540 6540 6411 6351 6297 6252 6252 ...
```

```
head(heptathlon)
```

```
##                     hurdles highjump  shot run200m longjump javelin run800m
## Joyner-Kersee (USA)   12.69     1.86 15.80   22.56     7.27   45.66  128.51
## John (GDR)            12.85     1.80 16.23   23.65     6.71   42.56  126.12
## Behmer (GDR)          13.20     1.83 14.20   23.10     6.68   44.54  124.20
## Sablovskaite (URS)    13.61     1.80 15.23   23.92     6.25   42.78  132.24
## Choubenkova (URS)     13.51     1.74 14.76   23.93     6.32   47.46  127.90
## Schulz (GDR)          13.75     1.83 13.50   24.65     6.33   42.82  125.79
##                     score
## Joyner-Kersee (USA)  7291
## John (GDR)           6897
## Behmer (GDR)         6858
## Sablovskaite (URS)   6540
## Choubenkova (URS)    6540
## Schulz (GDR)         6411
```

```
names(heptathlon)
```

```
## [1] "hurdles"  "highjump" "shot"     "run200m"  "longjump" "javelin"  "run800m"
## [8] "score"
```

1. Calculad el modelo de regresión lineal múltiple que predice el `score` final en base al resultado obtenido en las siete pruebas del heptatlón. Dad explicitamente la ecuación del modelo de regresión lineal múltiple. Dad una explicación a los coeficientes negativos y positivos del modelo. (*1 punto*)
2. Aplica la función `step` para obtener un modelo más sencillo de regresión. Comparar el modelo más sencillo obtenido con el original. (*1 punto*)
3. Comprobar que se verifican las hipótesis para poder llevar a cabo la regresión lineal múltiple (indicar cuales son y por qué se verifican o no utilizando los test de R oportunos) (*0.5 puntos*)