

# Entrega 3 Problemas y Talleres MATIII Estadística grado informática 2019-2020

Ricardo Alberich

13-05-2020

## Contenidos

<b>1 Entregas 3 Problemas: Estadística Inferencial 1</b>	<b>1</b>
1.1 Problema 1: Contraste de parámetros de dos muestras. . . . .	1
1.2 Problema 2 : Contraste dos muestras . . . . .	6
1.3 Problema 3 : Bondad de ajuste. La ley de Benford . . . . .	10
1.4 Problema 4 : Homegeneidad e independencia . . . . .	17
1.5 Problema 5 : ANOVA notas numéricas de tres grupos. . . . .	20
1.6 Problema 6 : ANOVA Comparación de las tasas de interés para la compra de coches entre seis ciudades. . . . .	24
1.7 Problema 7: Cuestiones cortas . . . . .	29

## 1 Entregas 3 Problemas: Estadística Inferencial 1

Contestad cada GRUPO de 3 a los siguientes problemas y cuestiones en un fichero Rmd y su salida en html o pdf.

Cambien podéis incluir capturas de problemas hechos en papel. Cada pregunta vale lo mismo y se reparte la nota entre sus apartados.

### 1.1 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  $\sigma_1$  y  $\sigma_2$ . Disponemos de dos muestras independientes de cuestionarios realizados por estudiantes de cada grado,  $n_1 = n_2 = 50$ .

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

```
G1=read.table("http://bioinfo.uib.es/~recerca/MATIIGMAT/NotasTestGrado/grado1.txt",
              header=TRUE)$x
G2=read.table("http://bioinfo.uib.es/~recerca/MATIIGMAT/NotasTestGrado/grado2.txt",
              header=TRUE)$x
```

```
n1=length(na.omit(G1))
n2=length(na.omit(G2))
media.muestra1=mean(G1,na.rm=TRUE)
media.muestra2=mean(G2,na.rm=TRUE)
desv.tip.muestra1=sd(G1,na.rm=TRUE)
desv.tip.muestra2=sd(G2,na.rm=TRUE)
```

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

$$\begin{array}{rcl} n_1 & = & 50, \\ \bar{x}_1 & = & 9.7592926, \\ \tilde{s}_1 & = & 1.1501225, \end{array} \quad \begin{array}{rcl} n_2 & = & 50 \\ \bar{x}_2 & = & 11.4660825 \\ \tilde{s}_2 & = & 1.5642932 \end{array}$$

Se pide:

1. Comentad brevemente el código de R explicando que hace cada instrucción.
2. Contrastad si hay evidencia de que las notas medias son distintas entre los dos grupos. En dos casos considerando las varianzas desconocidas pero iguales o desconocidas pero distintas. Tenéis que hacer el contraste de forma manual y con funciones de R y resolver el contraste con el  $p$ -valor.
3. Calculad e interpretad los intervalos de confianza para la diferencia de medias asociados a los dos test anteriores.
4. Comprobad con el test de Fisher y el de Levene si las varianzas de las dos muestras son iguales contra que son distintas. Tenéis que resolver el test de Fisher con R y de forma manual y el test de Levene con R y decidir utilizando el  $p$ -valor.

### 1.1.1 Solución

**Apartado 1.** El código R carga en las variables G1 y G2 las variables  $x$  de dos data frames de un servidor bioinfo.uib.es.

Luego calcula los estadísticos básicos para realizar las siguientes preguntas. Para los tamaños muestrales  $n_1$  y  $n_2$  se omiten los valores NA antes de asignar la `length` de los arrays. También se calculan las medias y las desviaciones típicas muestrales omitiendo (si es que hay) los valores no disponibles.

**Apartado 2.** Denotemos por  $\mu_1$  y  $\mu_2$  las medias de las notas de los grupos 1 y 2 respectivamente. El contraste que se pide es

$$\begin{cases} H_0 : \mu_1 = \mu_2 \\ H_1 : \mu_1 \neq \mu_2 \end{cases}$$

Estamos en un diseño de comparación de medias de dos grupos con dos muestras independientes de tamaño 50 grande. Tenemos dos casos varianzas desconocidas pero iguales y varianzas desconocidas pero distintas. Las funciones de R de este contraste para estos casos son:

#### Varianzas iguales

```
# test para varianzas iguales
t.test(G1,G2,var.equal = TRUE,alternative = "two.sided")
```

```
##
## Two Sample t-test
##
```

```
## data: G1 and G2
## t = -6.2159, df = 98, p-value = 0.00000001248
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -2.251691 -1.161889
## sample estimates:
## mean of x mean of y
## 9.759293 11.466083
```

### Varianzas distintas

```
# test para varianzas distintas
t.test(G1,G2,var.equal = FALSE,alternative = "two.sided")
```

```
##
## Welch Two Sample t-test
##
## data: G1 and G2
## t = -6.2159, df = 89.996, p-value = 0.00000001562
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -2.252298 -1.161282
## sample estimates:
## mean of x mean of y
## 9.759293 11.466083
```

El  $p$ -valor en ambos casos es muy pequeño así que la muestra no aporta evidencias rechazar la hipótesis nula las medias son iguales contra que son distintas.

Veamos en cálculo manual

### Varianzas desconocidas pero iguales, $n_1$ y $n_2$ grande

Si suponemos que  $\sigma_1 = \sigma_2$ , el estadístico de contraste es

$$t_0 = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2}\right) \cdot \frac{((n_1-1)\tilde{S}_1^2 + (n_2-1)\tilde{S}_2^2)}{(n_1+n_2-2)}}} = \frac{9.7592926 - 11.4660825}{\sqrt{\left(\frac{1}{50} + \frac{1}{50}\right) \cdot \frac{((50-1)1.1501225^2 + (50-1)1.5642932^2)}{(50+50-2)}}$$

operando obtenemos que  $t_0 = -6.2159314$ . y sabemos que sigue una distribución  $t$ -Student  $t_{n_1+n_2-2} = t_{98}$ . Para este hipótesis alternativa el  $p$ -valor es

$2 \cdot P(t_{98} > |-6.2159314|)$ , lo calculamos con R

### Varianzas desconocidas pero distintas, $n_1$ y $n_2$ grande

Si suponemos que  $\sigma_1 \neq \sigma_2$ , el estadístico de contraste es  $t_0 = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\tilde{S}_1^2}{n_1} + \frac{\tilde{S}_2^2}{n_2}}} \sim t_f$ , que, cuando  $\mu_1 = \mu_2$ , tiene distribución (aproximadamente, en caso de muestras grandes)  $t_f$  con

$$f = \left\lfloor \frac{\left(\frac{\tilde{S}_1^2}{n_1} + \frac{\tilde{S}_2^2}{n_2}\right)^2}{\frac{1}{n_1-1} \left(\frac{\tilde{S}_1^2}{n_1}\right)^2 + \frac{1}{n_2-1} \left(\frac{\tilde{S}_2^2}{n_2}\right)^2} \right\rfloor - 2.$$

Calculamos el estadístico y los grados de libertad con R

```
t0=(media.muestra1-media.muestra2)/sqrt(desv.tip.muestra1^2/n1+desv.tip.muestra2^2/n2)
#calculo el valor dentro del floor que es el que utiliza R

f1=(desv.tip.muestra1^2/n1+desv.tip.muestra2^2/n2)^2/(
  (1/(n1-1))*(desv.tip.muestra1^2/n1)^2+(1/(n2-1))*(desv.tip.muestra2^2/n2)^2)
f1
```

```
## [1] 89.99613
```

```
# calculo el propuesto en las transparencias que es anticuado
f=floor(f1)-2
f
```

```
## [1] 87
```

El  $p$  valor es

```
# el pvañor de la función de R
2*(pt(abs(t0),f1,lower.tail = FALSE))
```

```
## [1] 0.00000001562353
```

```
# el pvalor propuesto en teoría
2*(pt(abs(t0),f,lower.tail = FALSE))
```

```
## [1] 0.00000001713797
```

### Apartado 3

Los intervalos de confianza al nivel del 95% los podemos obtener así

```
t.test(G1,G2,var.equal = TRUE,alternative = "two.sided",conf.level = 0.95)$conf.int
```

```
## [1] -2.251691 -1.161889
## attr(,"conf.level")
## [1] 0.95
```

```
t.test(G1,G2,var.equal = FALSE,alternative = "two.sided",conf.level = 0.95)$conf.int
```

```
## [1] -2.252298 -1.161282
## attr(,"conf.level")
## [1] 0.95
```

Son similares, podemos asegurar que la diferencia de medias se encuentra  $-2.25 < \mu_1 - \mu_2 < -1.16$  al nivel del 95 el grupo 2 tiene una media entre 2.25 y 1.16 puntos mayor que el grupo 2 aproximadamente.

**Apartado 4** El test que nos piden es el de igualdad de varianzas

$$\begin{cases} H_0 : \sigma_1^2 = \sigma_2^2 \\ H_0 : \sigma_1^2 \neq \sigma_2^2 \end{cases}$$

El test de Fisher de igualdad de varianzas

```
var.test(G1,G2,alternative ="two.sided" )
```

```
##
## F test to compare two variances
##
## data: G1 and G2
## F = 0.54057, num df = 49, denom df = 49, p-value = 0.03354
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3067606 0.9525862
## sample estimates:
## ratio of variances
## 0.54057
```

Obtenemos un  $p$ -valor alto no podemos rechazar la igualdad de varianzas.

De forma manual el estadístico de este test sabemos que es

$$f_0 = \frac{\tilde{S}_1^2}{\tilde{S}_2^2} = \frac{1.3227818}{2.4470131} = 0.54057.$$

Que sigue una ley e Fisher y el  $p$ \_valor es  $\min\{2 \cdot P(F_{n_1-1, n_2-1} \leq f_0), 2 \cdot P(F_{n_1-1, n_2-1} \geq f_0)\}$ .  
que con R es

```
n1
```

```
## [1] 50
```

```
n2
```

```
## [1] 50
```

```
f0=desv.tip.muestra1^2/desv.tip.muestra2^2
f0
```

```
## [1] 0.54057
```

```
pvalor=min(2*pf(f0,n1-1,n2-2),2*pf(f0,n1-1,n2-2,lower.tail = FALSE))
pvalor
```

```
## [1] 0.03420609
```

Obtenemos los mismos resultados que con la función `var.test`.

El test de Levene con R tiene las mismas hipótesis que el anterior

```
library(car,quietly = TRUE)# pogo quietly para que quite avisos
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##      recode
```

```
## The following object is masked from 'package:purrr':
##
##      some
```

```
notas=c(G1,G2)
grupo=as.factor(c(rep(1,length(G1)),rep(2,length(G1))))
leveneTest(notas~grupo)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  1  1.8029 0.1825
##      98
```

El  $p$ -valor obtenido es alto así que el test de levene no aporta evidencias contra la igualdad de varianzas entre las notas de los dos grupos.

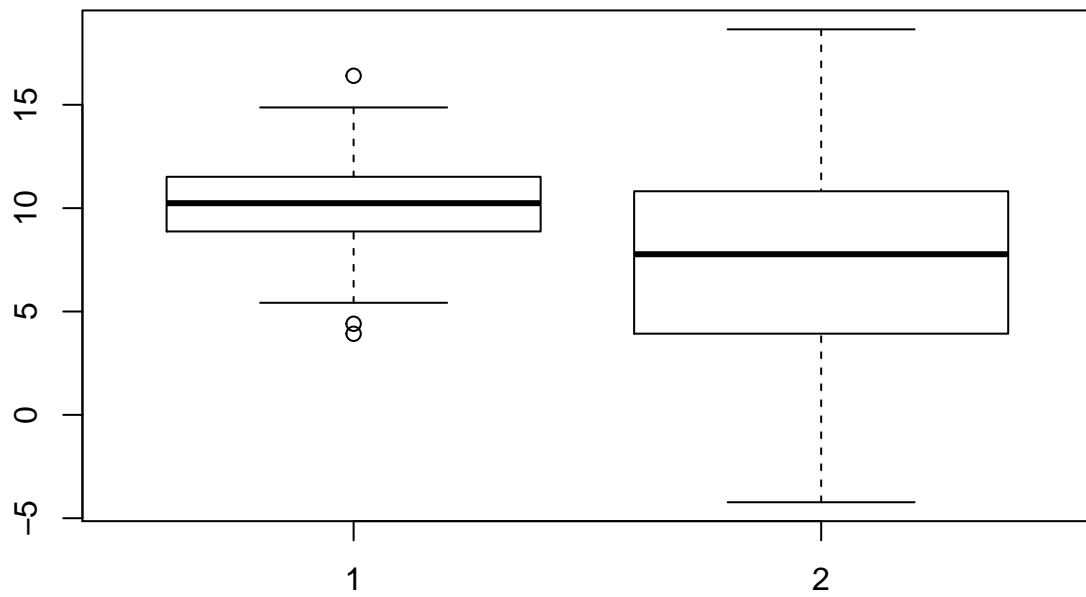
## 1.2 Problema 2 : Contraste dos muestras

Simulamos dos muestras con las funciones siguientes

```
set.seed(2020)
x1=rnorm(100,mean = 10,sd=2)
x2=rnorm(100,mean = 8,sd=4)
```

Dibujamos estos gráficos

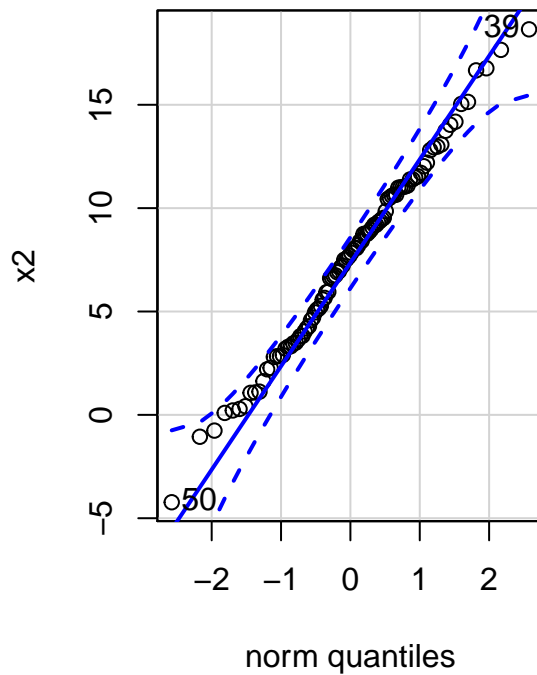
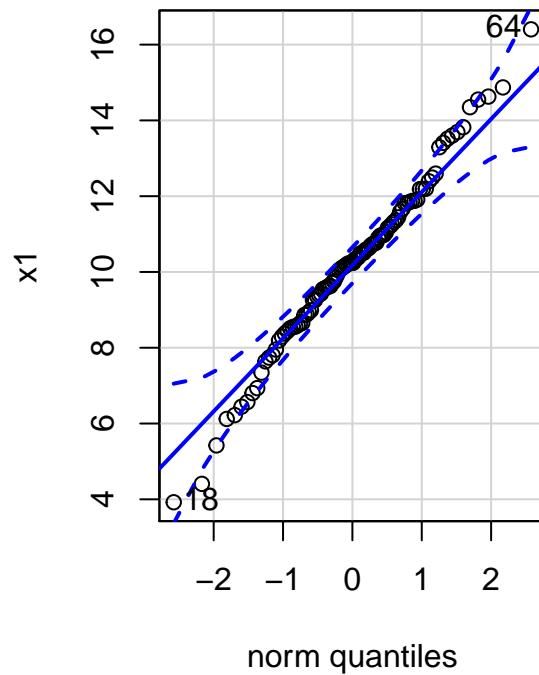
```
boxplot(x1,x2)
```



```
library(car)
par(mfrow=c(1,2))
qqPlot(x1)
```

```
## [1] 18 64
```

```
qqPlot(x2)
```



```
## [1] 50 39
```

```
par(mfrow=c(1,1))
```

Realizamos algunos contrastes de hipótesis de igual de medias entre ambas muestras

```
t.test(x1,x2,var.equal = TRUE,alternative = "greater")
```

```
##
## Two Sample t-test
##
## data: x1 and x2
## t = 5.3009, df = 198, p-value = 0.0000001531
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.844757      Inf
## sample estimates:
## mean of x mean of y
## 10.217784  7.537402
```

```
t.test(x1,x2,var.equal = FALSE,alternative = "two.sided")
```

```
##
```



```
## Welch Two Sample t-test
##
## data: x1 and x2
## t = 5.3009, df = 144.56, p-value = 0.0000004221
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 1.680966 3.679797
## sample estimates:
## mean of x mean of y
## 10.217784 7.537402
```

```
t.test(x1,x2,var.equal = TRUE)
```

```
##
## Two Sample t-test
##
## data: x1 and x2
## t = 5.3009, df = 198, p-value = 0.0000003061
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 1.683238 3.677526
## sample estimates:
## mean of x mean of y
## 10.217784 7.537402
```

Se pide

1. ¿Cuál es la distribución y los parámetros de las muestras generadas?
2. ¿Qué muestran y cuál es la interpretación de los gráficos?
3. ¿Qué test contrasta si hay evidencia a favor de que las medias poblacionales de las notas en cada grupo sean distintas? Di qué código de los anteriores resuelve este test.
4. Para el test del apartado anterior dad las hipótesis nula y alternativa y redactar la conclusión del contraste.

### 1.2.1 Solución

#### Apartado 1

Se generan dos muestras de poblaciones normales de medias 10 y 8 y desviaciones típicas 2 y 4.

**Apartado 2** El primer gráficos es un diagrama de caja (*boxplot*) que compara las distribuciones de los datos. Vemos que efectivamente la muestra 1 tiene una caja y unos bigotes más comprimidos que la muestra 2 así que la primera tiene menos varianza. Vemos que los valores medianos de la muestra 1 son más grandes que los de la muestra 2. Recordemos que la distribución normal es simétrica por lo que la media y la mediana coinciden. La muestra 1 tiene valores atípicos en la parte superior 1 y en la inferior parece que 2.

El segundo gráfico es un gráfico cuantil-cuantil o qqplot de normalidad. Compara los cuantiles muestrales con los teóricos de una normal y nos da un intervalo de confianza para esas observaciones.

Vemos que los cuantiles teóricos no difieren excesivamente de los muestrales en cada una de las muestras y que muy pocos valores se escapan de los intervalos de confianza esperados en el caso de normalidad. Así que no hay motivo para pensar que las distribuciones de ambas muestras proceden de poblaciones normales.

#### Apartado 3

El código es

```
t.test(x1,x2,var.equal = TRUE,alternative = "greater")
t.test(x1,x2,var.equal = FALSE,alternative = "two.sided")
t.test(x1,x2,var.equal = TRUE)
```

El primer test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales  $H_0; \mu_1 = \mu_2$  contra  $H_1 : \mu_1 > \mu_2$ . **Así que este TEST NO ES**

El segundo test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales  $H_0; \mu_1 = \mu_2$  contra  $H_1 : \mu_1 > \mu_2$ . **Así que este TEST NO ES** pues no contrasta contra medias distintas.

El segundo test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales  $H_0; \mu_1 = \mu_2$  contra  $H_1 : \mu_1 \neq \mu_2$ . **Así que este TEST SI PUEDE SER** contrasta contra medias distintas para el caso de varianzas distintas.

El segundo test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales  $H_0; \mu_1 = \mu_2$  contra  $H_1 : \mu_1 > \mu_2$  pues la opción por defecto de la función. **Así que este TEST SI PUEDE SER** contra medias distintas para el caso de varianzas iguales.

**Apartado 4** El contrastes es

$$\begin{cases} H_0 : & \mu_1 = \mu_2 \\ H_1 : & \mu_1 \neq \mu_2 \end{cases}$$

En los dos últimos test los  $p$ -valores son muy muy pequeños así que hay evidencias en contra de la igualdad de medias entre las dos muestras. Además claramente los intervalos de confianza no contienen al cero.

### 1.3 Problema 3 : Bondad de ajuste. La ley de Benford

La ley de Benford es una distribución discreta que siguen las frecuencias de los primeros 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 lo podemos hacer así

```
prob=log10(1+1/c(1:9))
prob
```

```
## [1] 0.30103000 0.17609126 0.12493874 0.09691001 0.07918125 0.06694679 0.05799195
## [8] 0.05115252 0.04575749
```

```
MM=rbind(c(1:9),prob)
df=data.frame(rbind(prob))
# Y hacemos una bonita tabla
colnames(df)=paste("Díg.",c(1:9),sep = " ")
knitr::kable(df,format = 'markdown')
```

	Díg. 1	Díg. 2	Díg. 3	Díg. 4	Díg. 5	Díg. 6	Díg. 7	Díg. 8	Díg. 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 suficiente para extraer los tres primeros dígitos.
muestra_pot_2_3digitos=str_sub(as.character(2^c(1:1000)),1,3)
head(muestra_pot_2_3digitos)
```

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

```
tail(muestra_pot_2_3digitos)
```

```
## [1] "334" "669" "133" "267" "535" "107"
```

```
#Construimos un data frame con tres columnas que nos dan el primer,
#segundo y tercer dígito respectivamente.
df_digitos=data.frame(muestra_pot_2_3digitos,
                      primer_digito=as.integer(
                        substring(muestra_pot_2_3digitos, 1, 1)),
                      segundo_digito=as.integer(
                        substring(muestra_pot_2_3digitos, 2, 2)),
                      tercer_digito=as.integer(
                        substring(muestra_pot_2_3digitos, 3, 3)))
head(df_digitos)
```

```
## muestra_pot_2_3digitos primer_digito segundo_digito tercer_digito
## 1                2                2                NA                NA
## 2                4                4                NA                NA
## 3                8                8                NA                NA
## 4               16                1                6                NA
## 5               32                3                2                NA
## 6               64                6                4                NA
```

Notad que los NA en el segundo y el tercer dígito corresponden a número con uno o dos dígitos.

Se pide:

1. Contrastad con un test  $\chi^2$  que el primer dígito sigue una ley de Benford. Notad que el primer dígito no puede ser 0. Resolved manualmente y con una función de R.
2. Contrastad con un test  $\chi^2$  que el segundo dígito sigue una ley de uniforme discreta. Notad que ahora si puede ser 0. Resolved con funciones de R.

3. Contrastad con un test  $\chi^2$  que el tercer dígito sigue una ley de uniforme discreta. Notad que ahora si puede ser 0. Resolved con manualmente calculado las frecuencias esperadas y observadas, el estadístico de contraste y el  $p$ -valor utilizando R. Comprobad que vuestros resultados coinciden con los de la función de R que calcula este contraste.
4. Dibujad con R para los apartados 1 y 2 los diagramas de frecuencias esperados y observados. Comentad estos gráficos

### 1.3.1 Solución

**Apartado 1** El contraste que nos piden es

$$\begin{cases} H_0 : & \text{El primer dígito de las 1000 primeras potencias de 2 sigue una ley Benford,} \\ H_1 : & \text{sigue cualquier otra distribución.} \end{cases}$$

El siguiente código resuelve el tes de forma manual calculando el estadístico y el  $p$ -valor

```
prob=log10(1+1/(1:9))
prob_benford=prob
n=1000
frec_esp_benford=n*prob_benford
frec_esp_benford
```

```
## [1] 301.03000 176.09126 124.93874 96.91001 79.18125 66.94679 57.99195
## [8] 51.15252 45.75749
```

```
frec_obs_primer=table(df_digitos$primer_digito)
frec_obs_primer
```

```
##
##  1  2  3  4  5  6  7  8  9
## 301 176 125 97 79 69 56 52 45
```

```
chi2_est=sum((frec_obs_primer-frec_esp_benford)^2/frec_esp_benford)
chi2_est
```

```
## [1] 0.1585506
```

```
pchisq(chi2_est,9-1,lower.tail = FALSE)
```

```
## [1] 0.9999985
```

La función de R que resuelve el test es

```
chisq.test(frec_obs_primer,p=prob_benford)
```

```
##
## Chi-squared test for given probabilities
##
## data:  frec_obs_primer
## X-squared = 0.15855, df = 8, p-value = 1
```

Obtenemos los mismos resultados. El  $p$  valor es muy alto. No podemos rechazar que el primer dígito de las 1000 primeras potencias enteras de 2 siga una ley de distribución de Benford.

## Apartado 2

El contraste que nos piden es

$$\begin{cases} H_0 : & \text{El segundo dígito de las 1000 primeras potencias de 2 sigue una ley uniforme,} \\ H_1 : & \text{sigue cualquier otra distribución.} \end{cases}$$

Procedemos de forma similar al caso anterior. De forma manual el cálculo es

```
prob_unif=rep(1/10,10)
prob_unif

## [1] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1

segundo_digito=na.omit(df_digitos$segundo_digito)
n=length(segundo_digito)
n

## [1] 997

frec_esp_uniforme=n*prob_unif
frec_esp_uniforme

## [1] 99.7 99.7 99.7 99.7 99.7 99.7 99.7 99.7 99.7 99.7

frec_obs_segundo=table(segundo_digito)
frec_obs_segundo

## segundo_digito
##  0  1  2  3  4  5  6  7  8  9
## 121 112 109 108 98 95 94 91 83 86

chi2_est=sum((frec_obs_segundo-frec_esp_uniforme)^2/frec_esp_uniforme)
chi2_est

## [1] 13.64193

1-pchisq(chi2_est,10-1)

## [1] 0.1356449

pchisq(chi2_est,10-1,lower.tail = FALSE)

## [1] 0.1356449
```

Con la función `chisq.test` obtenemos los mismo resultados

```
chisq.test(frec_obs_segundo,p=prob_unif)
```

```
##  
## Chi-squared test for given probabilities  
##  
## data:  frec_obs_segundo  
## X-squared = 13.642, df = 9, p-value = 0.1356
```

Para el segundo dígito con un  $p$ -valor de 0.1356 no podemos rechazar que el segundo dígito siga una ley uniforme.

### Apartado 3

El contraste que nos piden es

$$\begin{cases} H_0 : & \text{El tercer dígito de las 1000 primeras potencias de 2 sigue una ley uniforme,} \\ H_1 : & \text{sigue cualquier otra distribución.} \end{cases}$$

Procedemos de forma similar al caso anterior. De forma manual el cálculo es

```
prob_unif=rep(1/10,10)  
prob_unif
```

```
## [1] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
tercer_digito=na.omit(df_digitos$tercer_digito)  
n=length(tercer_digito)  
n
```

```
## [1] 994
```

```
frec_esp_uniforme=n*prob_unif  
frec_esp_uniforme
```

```
## [1] 99.4 99.4 99.4 99.4 99.4 99.4 99.4 99.4 99.4 99.4
```

```
frec_obs_tercer=table(tercer_digito)  
frec_obs_tercer
```

```
## tercer_digito  
##  0  1  2  3  4  5  6  7  8  9  
## 96 101 88 116 97 90 110 92 95 109
```

```
chi2_est=sum((frec_obs_tercer-frec_esp_uniforme)^2/frec_esp_uniforme)  
chi2_est
```

```
## [1] 7.971831
```

```
1-pchisq(chi2_est,10-1)
```

```
## [1] 0.5369875
```

```
pchisq(chi2_est,10-1,lower.tail = FALSE)
```

```
## [1] 0.5369875
```

Con la función `chisq.test` obtenemos los mismo resultados

```
chisq.test(frec_obs_tercer,p=prob_unif)
```

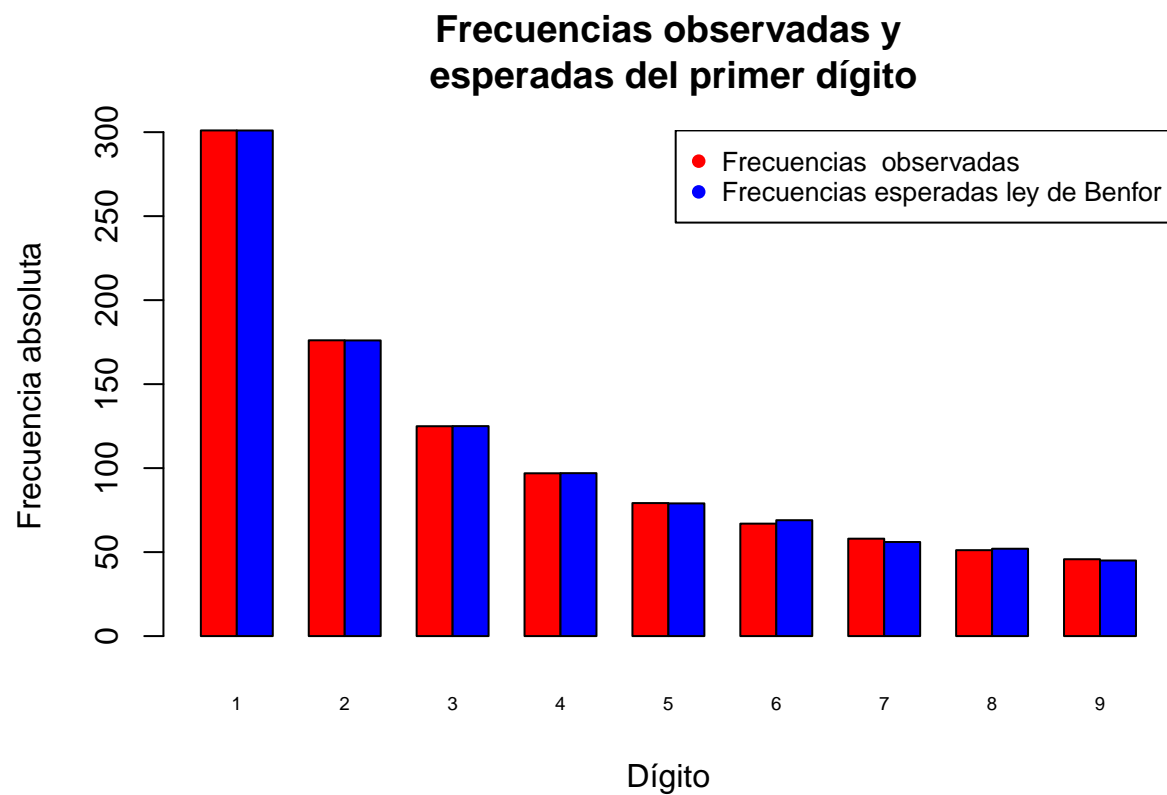
```
##  
## Chi-squared test for given probabilities  
##  
## data:  frec_obs_tercer  
## X-squared = 7.9718, df = 9, p-value = 0.537
```

Para el segundo dígito con un *p*-valor alto 0.537 no podemos rechazar que el tercer dígito siga una ley uniforme.

#### Apartado 4

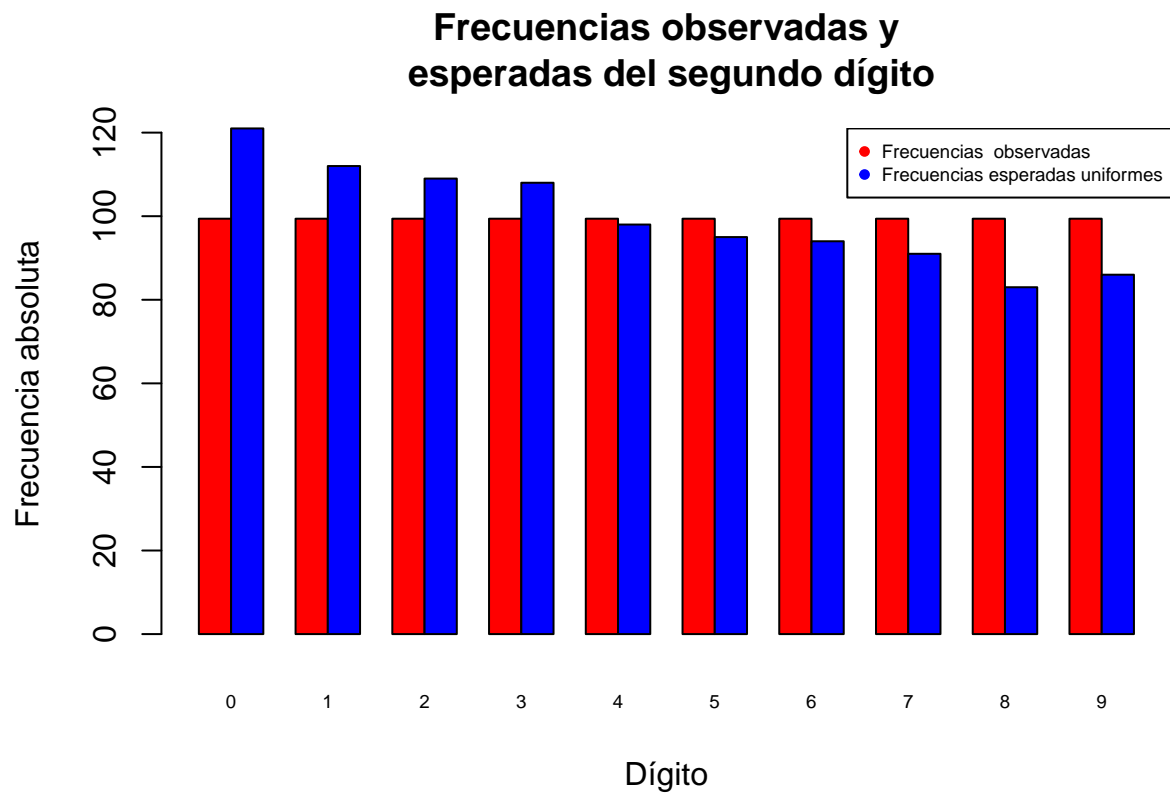
Una gráfica comparando las frecuencias

```
barplot(rbind(frec_esp_benford,frec_obs_primer),  
        beside=TRUE,col=c("red","blue"),  
        main="Frecuencias observadas y\n esperadas del primer dígito",  
        cex.names =0.6,xlab="Dígito",ylab="Frecuencia absoluta")  
legend("topright",legen=c("Frecuencias observadas",  
                           "Frecuencias esperadas ley de Benfor"),pch=19,col=c("red","blue"),  
       cex=0.8)
```



```
barplot(rbind(frec_esp_uniforme,frec_obs_segundo),
        beside=TRUE,col=c("red","blue"),
        main="Frecuencias observadas y\n esperadas del segundo dígito",
        cex.names =0.6,xlab="Dígito",ylab="Frecuencia absoluta")
legend("topright",legen=c("Frecuencias observadas",
                           "Frecuencias esperadas uniformes"),pch=19,col=c("red","blue"),
        cex=0.6)
```





## 1.4 Problema 4 : Homogeneidad e independencia

Queremos analizar los resultados de aprendizaje con tres tecnologías. Para ello se seleccionan 3 muestras de 50 estudiantes y se les somete a evaluación después de un curso.

```
set.seed(2020)
nota=factor(sample(c(1,2,3,4),p=c(0.1,0.4,0.3,0.2),replace=TRUE,size=150),
            labels=c("S","A","N","E"))
tecnologia=rep(c("Mathematica","R","Python"),each=50)
frec=table(nota,tecnologia)
frec
```

```
##      tecnologia
## nota Mathematica Python  R
##   S             7      6  2
##   A            18     15 22
##   N            15     20 18
##   E            10      9  8
```

```
col_frec=colSums(frec)
col_frec
```

```
## Mathematica      Python      R
##           50           50           50
```

```
row_frec=rowSums(frec)
row_frec
```

```
## S A N E
## 15 55 53 27
```

```
N=sum(frec)
teoricas=row_frec%*%t(col_frec)/N
teoricas
```

```
##      Mathematica  Python      R
## [1,]      5.00000  5.00000  5.00000
## [2,]     18.33333 18.33333 18.33333
## [3,]     17.66667 17.66667 17.66667
## [4,]      9.00000  9.00000  9.00000
```

```
dim(frec)
```

```
## [1] 4 3
```

```
dim(teoricas)
```

```
## [1] 4 3
```

```
sum((frec-teoricas)^2/teoricas)
```

```
## [1] 5.084658
```

```
chisq.test(table(nota,tecnologia))
```

```
##
## Pearson's Chi-squared test
##
## data:  table(nota, tecnologia)
## X-squared = 5.0847, df = 6, p-value = 0.533
```

Se pide

1. Discutid si hacemos un contraste de independencia o de homogeneidad de las distribuciones de las notas por tecnología. Escribid las hipótesis del contraste.
2. Interpretad la función `chisq.test` y resolved el contraste.
3. Interpretad `teoricas=row_frec%*%t(col_frec)/N` reproducid manualmente el segundo resultado de la primera fila.

### 1.4.1 Solución

#### Apartado 1

Podemos plantear dos hipótesis distintas. La primera es que la calificación obtenidas es independiente de la tecnología, la segunda es las distribuciones en las 4 clases de notas son las mismas para cada tecnología. En el primer caso es un contraste de independencia y en el segundo es de homogeneidad.

Podemos optar en este caso por cualquiera de los dos. Yo he optado por el de independencia

así que la hipótesis  $H_0$ : la nota obtenida es independientes de la tecnología del curso, contra que no lo es.

**Apartado 2** Para el contraste de independencia (o el de homogeneidad) hay que pasar una tabla de contingencia a la función `chisq.test` eso es lo que hacemos

```
table(nota,tecnologia)
```

```
##      tecnologia
## nota Mathematica Python  R
##   S              7      6  2
##   A             18     15 22
##   N             15     20 18
##   E             10      9  8
```

```
chisq.test(table(nota,tecnologia))
```

```
##
## Pearson's Chi-squared test
##
## data:  table(nota, tecnologia)
## X-squared = 5.0847, df = 6, p-value = 0.533
```

El  $p$ -valor obtenido es grande así que no podemos rechazar la hipótesis nula las calificaciones obtenidas no dependen del programa utilizado en el curso.

#### Apartado 3

La expresión  $teoricas=row\_frec\%*\%t(col\_frec)/N$

```
row_frec
```

```
## S  A  N  E
## 15 55 53 27
```

```
col_frec
```

```
## Mathematica      Python      R
##           50           50      50
```

```
row_frec%*%t(col_frec)
```

```
##      Mathematica Python      R
## [1,]           750      750  750
## [2,]          2750     2750 2750
## [3,]          2650     2650 2650
## [4,]          1350     1350 1350
```

```
row_frec%*%t(col_frec)/N
```

```
##      Mathematica  Python      R
## [1,]      5.00000  5.00000  5.00000
## [2,]     18.33333 18.33333 18.33333
## [3,]     17.66667 17.66667 17.66667
## [4,]      9.00000  9.00000  9.00000
```

Así que `row_frec%*%t(col_frec)` calcula el producto de las frecuencia marginal por filas y columnas.

Por ejemplo la segunda fila es  $2750 = 50 \cdot 55$  en los tres casos

Por último al dividir por  $N = 150$  el tamaño total de las muestra obtenemos las frecuencias esperadas en el caso de independencia  $2750/150 = 18.33333$ .

## 1.5 Problema 5 : ANOVA notas numéricas de tres grupos.

El siguiente código nos da las notas numéricas (variable `nota`) de los mismos ejercicios para tres tecnologías en tres muestra independientes de estudiantes de estas tres tecnologías diferentes

```
head(nota)
```

```
## [1] 79.424303 77.538709 42.549421 41.739852 0.086642 88.014337
```

```
library(nortest)
lillie.test(nota[tecnologia=="Mathematica"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  nota[tecnologia == "Mathematica"]
## D = 0.08739, p-value = 0.4436
```

```
lillie.test(nota[tecnologia=="R"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  nota[tecnologia == "R"]
## D = 0.082139, p-value = 0.5449
```

```
lillie.test(nota[tecnologia=="Python"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  nota[tecnologia == "Python"]
## D = 0.089681, p-value = 0.4019
```

```
lillie.test(nota)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  nota
## D = 0.056381, p-value = 0.2885
```

```
bartlett.test(nota~tecnologia)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  nota by tecnologia
## Bartlett's K-squared = 0.50309, df = 2, p-value = 0.7776
```

```
library(car)
leveneTest(nota~as.factor(tecnologia))
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group    2  0.3881  0.679
##           147
```

```
sol_aov=aov(nota~as.factor(tecnologia))
sol_aov
```

```
## Call:
##  aov(formula = nota ~ as.factor(tecnologia))
##
## Terms:
##              as.factor(tecnologia) Residuals
## Sum of Squares              837.39 123445.06
## Deg. of Freedom              2         147
##
## Residual standard error: 28.97865
## Estimated effects may be unbalanced
```

Del summary(sol\_aov) os damos la salida a falta de algunos de los valores

```
> summary(sol_aov)
              Df Sum Sq Mean Sq F value Pr(>F)
as.factor(tecnologia) ---    837    418.7  ---  ---
Residuals         --- 123445    839.8
```

```
pairwise.t.test(nota,as.factor(tecnologia),p.adjust.method = "none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
```

```
##
## data:  nota and as.factor(tecnologia)
##
##      Mathematica Python
## Python 0.35      -
## R      0.89      0.43
##
## P value adjustment method: none
```

```
pairwise.t.test(nota,as.factor(tecnologia),p.adjust.method = "bonferroni")
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  nota and as.factor(tecnologia)
##
##      Mathematica Python
## Python 1      -
## R      1      1
##
## P value adjustment method: bonferroni
```

Se pide

1. ¿Podemos asegurar que las muestras son normales en cada grupo? ¿y son homocedásticas? Sea cual sea la respuesta justificad que parte del código la confirma.
2. La función `aov` que `test` calcula. Escribid formalmente la hipótesis nula y la alternativa.
3. Calcula la tabla de ANOVA y resuelve el test.
4. ¿Qué contrastes realiza la función `pairwise.t.test`? Utilizando los resultados anteriores aplicad e interpretad los contrastes del apartado anterior utilizando el ajuste de Holm.

### 1.5.1 Solución

**Apartado 1** Este código es el que pasa un test de normalidad con la función `lillie.test` para la distribución de notas en cada una de las tecnologías

```
head(nota)
```

```
## [1] 79.424303 77.538709 42.549421 41.739852 0.086642 88.014337
```

```
library(nortest)
lillie.test(nota[tecnologia=="Mathematica"])
```

```
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  nota[tecnologia == "Mathematica"]
## D = 0.08739, p-value = 0.4436
```

```
lillie.test(nota[tecnologia=="R"])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data: nota[tecnologia == "R"]  
## D = 0.082139, p-value = 0.5449
```

```
lillie.test(nota[tecnologia=="Python"])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data: nota[tecnologia == "Python"]  
## D = 0.089681, p-value = 0.4019
```

Como se ve los  $p$ -valores son altos en el pero de los caso es mayor que 0.4 no podemos rechazar de las notas numéricas en cada uno de los tres casos.

Para la homogeneidad de las varianzas en las tres poblaciones hacemos un `bartlett.testo` un LeveneTest.

```
bartlett.test(nota~tecnologia)
```

```
##  
## Bartlett test of homogeneity of variances  
##  
## data: nota by tecnologia  
## Bartlett's K-squared = 0.50309, df = 2, p-value = 0.7776
```

```
library(car)  
leveneTest(nota~as.factor(tecnologia))
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group  2  0.3881  0.679  
##      147
```

Nos salen  $p$ -valores del orden de 0.7 o 0.6 por lo que no podemos rechazar la homocedasticidad de las tres varianzas.

## Apartado 2

La función `aov` calcula un test de igualdad de medias de las notas numéricas en los tres cursos.

$$\begin{cases} H_0 : & \mu_{\text{Mathematica}} = \mu_{\text{Python}} = \mu_R \\ H_1 : & \text{no todas las medias son iguales.} \end{cases}$$

## Apartado 3

```
sol_aov=aov(nota~as.factor(tecnologia))  
sol_aov
```

```
## Call:
## aov(formula = nota ~ as.factor(tecnologia))
##
## Terms:
##          as.factor(tecnologia) Residuals
## Sum of Squares          837.39 123445.06
## Deg. of Freedom           2       147
##
## Residual standard error: 28.97865
## Estimated effects may be unbalanced
```

```
summary(sol_aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(tecnologia)  2    837   418.7   0.499  0.608
## Residuals          147 123445   839.8
```

EL  $p$ -valor es muy grade 0.608 no podemos rechazar la igualdad de medias.

#### Apartado 4

La función `pairwise.t.test` compara las medias dos a dos en este caso hay 4 comparaciones por pares. Para ajustar el nivel de significación tenemos que recurrir a alguno de los métodos de ajuste del  $p$ -valor que se pasan con el parámetro `p.adjust.method` en este caso nos piden que utilicemos el método de Holm que tenemos que calcular pues no figura en el código del enunciado.

```
pairwise.t.test(nota,as.factor(tecnologia),p.adjust.method = "holm")
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  nota and as.factor(tecnologia)
##
##      Mathematica Python
## Python 1      -
## R      1      1
##
## P value adjustment method: holm
```

Los tres  $p$ -valores son prácticamente son 1 con el ajuste del método de Holm; o podemos rechazar las igual de medias dos a dos.

## 1.6 Problema 6 : ANOVA Comparación de las tasas de interés para la compra de coches entre seis ciudades.

Consideremos el data set `newcar` accesible desde <https://www.itl.nist.gov/div898/education/anova/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 dos columnas:

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

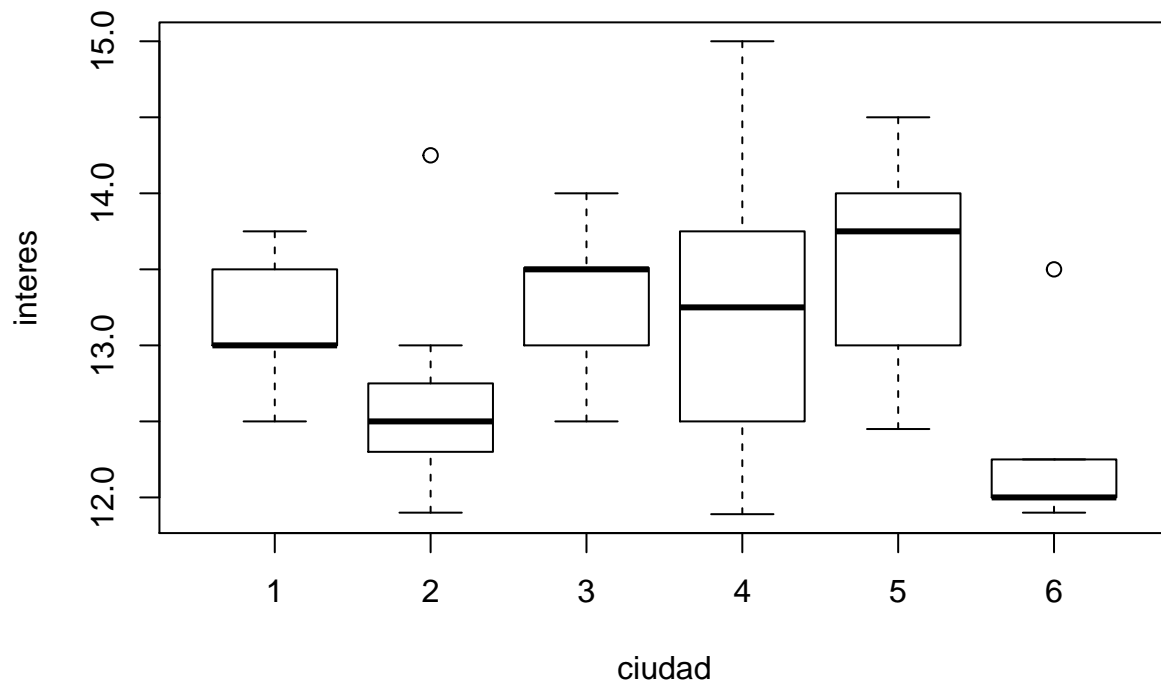


- City (ciudad) : la ciudad en la que se observó la tasa de interés para distintos concesionarios (codificada a enteros). Tenemos observaciones de 6 ciudades.

```
datos_interes=read.table(
  "https://www.itl.nist.gov/div898/education/anova/newcar.dat",
  skip=25)
# salto las 25 primeras líneas del fichero,son un preámbulo que explica los datos.
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(interres~ciudad,data=datos_interes)
```



Se pide:

1. Comentad el código y el diagrama de caja.
2. Se trata de contrastar si hay evidencia de que las 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.
3. Comprobad las condiciones del ANOVA con un test KS y un test de Levene (con código de R). Justificad las conclusiones.

4. Realizad el contraste de ANOVA (se cumplan las condiciones o no) y redactar adecuadamente la conclusión. Tenéis que hacerlo con funciones de R.
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 significación  $\alpha = 0.1$ . Redactad las conclusiones que se obtienen de las mismas.

### 1.6.1 Solución

#### Apartado 1

El código del enunciado nos carga los datos de una web, tenemos que pasar el parámetro `skip=25` para que se salte las 25 primeras líneas del fichero de texto que son un preámbulo que explica los datos.

En el diagrama de caja vemos que las medias las distribuciones de la **Rate** por ciudad son muy distintas, no parecen tener ni medias ni varianzas iguales

#### Apartado 2

Las condiciones para realizar un ANOVA son:

- Muestras independientes y aleatorias
- Distribución normal de la Rate  $N(\mu_i, \sigma_i)$  para las seis ciudades  $i = 1, 2, 3, 4, 5, 6$ .
- homocedasticidad; igualdad de varianzas  $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = \sigma_5 = \sigma_6$ .

El ANOVA que se pide es

$$\begin{cases} H_0 : & \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 \\ H_1 : & \text{no todas las medias son iguales.} \end{cases}$$

#### Apartado 3

El siguiente código realiza un test KS con corrección de Lillie para la normalidad de la variable Rate en cada una de la seis ciudades

```
library(nortest)
lillie.test(datos_interes$interes[datos_interes$ciudad==1])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos_interes$interes[datos_interes$ciudad == 1]
## D = 0.22384, p-value = 0.2163
```

```
lillie.test(datos_interes$interes[datos_interes$ciudad==2])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  datos_interes$interes[datos_interes$ciudad == 2]
## D = 0.22884, p-value = 0.1903
```

```
lillie.test(datos_interes$interes[datos_interes$ciudad==3])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  datos_interes$interes[datos_interes$ciudad == 3]  
## D = 0.19145, p-value = 0.4459
```

```
lillie.test(datos_interes$interes[datos_interes$ciudad==4])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  datos_interes$interes[datos_interes$ciudad == 4]  
## D = 0.11264, p-value = 0.9852
```

```
lillie.test(datos_interes$interes[datos_interes$ciudad==5])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  datos_interes$interes[datos_interes$ciudad == 5]  
## D = 0.20021, p-value = 0.3743
```

```
lillie.test(datos_interes$interes[datos_interes$ciudad==6])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  datos_interes$interes[datos_interes$ciudad == 6]  
## D = 0.3494, p-value = 0.002236
```

No podemos rechazar la normalidad con el `lillie.test` en las 5 primeras ciudades, pero parece que la última está lejos de ser normal.

Ahora comprobemos que

$$\begin{cases} H_0 : & \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2 = \sigma_5^2 = \sigma_6^2 \\ H_1 : & \text{no todas las varianzas son iguales.} \end{cases}$$

con el test de Levene (o el de Bartlett)

```
library(car)  
print(leveneTest(datos_interes$interes~as.factor(datos_interes$ciudad)))
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 5  1.2797 0.2882  
##      48
```

El test de levene nos da un  $p$ -valor superior a 0.28 aceptamos la igualdad de varianzas

#### Apartado 4

Resolvemos el ANOVA con el código siguiente

```
summary(aov(datos_interes$interes~as.factor(datos_interes$ciudad)))
```

```
##                               Df Sum Sq Mean Sq F value   Pr(>F)
## as.factor(datos_interes$ciudad)  5  10.95   2.1891    4.829 0.00117 **
## Residuals                      48   21.76   0.4533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El  $p$ -valor es muy bajo 0.00117 rechazamos la igualdad de las seis medias, al menos hay dos distintas.

Comprobamos por gusto el  $p$ -valor a partir de los datos del summary

```
Fest=2.1891/0.4533
```

```
Fest
```

```
## [1] 4.829252
```

```
1-pf(Fest,5,48)
```

```
## [1] 0.001174782
```

```
pf(Fest,5,48,lower.tail = FALSE)
```

```
## [1] 0.001174782
```

#### Apartado 5

Comparemos las medias dos a dos son 15 comparaciones

```
pairwise.t.test(datos_interes$interes,as.factor(datos_interes$ciudad),p.adjust.method = "holm")
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  datos_interes$interes and as.factor(datos_interes$ciudad)
##
##    1      2      3      4      5
## 2 0.5781 -      -      -      -
## 3 1.0000 0.3330 -      -      -
## 4 1.0000 0.4651 1.0000 -      -
## 5 1.0000 0.0926 1.0000 1.0000 -
## 6 0.0353 1.0000 0.0148 0.0244 0.0028
##
## P value adjustment method: holm
```

Nos piden que decidamos con  $\alpha = 0.1$ , así que rechazaremos la igualdad de medias de todas las comparaciones con  $p$ -valor inferior a 0.1.

Tenemos que rechazar la igualdad de medias entre la ciudad 2 con la 5 y la de la ciudad 6 con las ciudades 1, 3, 4 y 5.

## 1.7 Problema 7: Cuestiones cortas

- Cuestión 1: Supongamos que conocemos el  $p$ -valor de un contraste. Para que valores de nivel de significación  $\alpha$  RECHAZAMOS la hipótesis nula.
- Cuestión 2: Hemos realizado un ANOVA de un factor con 3 niveles, y hemos obtenido un  $p$ -valor de 0.001. Suponiendo que las poblaciones satisfacen las condiciones para que el ANOVA tenga sentido, ¿podemos afirmar con un nivel de significación  $\alpha = 0.05$  que las medias de los tres niveles son diferentes dos a dos? Justificad la respuesta.
- Cuestión 3: Lanzamos 300 veces un dado de 6 caras de parchís, queremos contrastar que los resultados son equiprobables. ¿Cuáles serían las frecuencias esperadas o teóricas del contraste?
- Cuestión 4: En un ANOVA de una vía, queremos contrastar si los 6 niveles de un factor definen poblaciones con la misma media. Sabemos que estas seis poblaciones son normales con la misma varianza  $\sigma = 2$ . Estudiamos a 11 individuos de cada nivel y obtenemos que  $SS_{Total} = 256.6$  y  $SS_{Tr} = 60.3$ . ¿Qué vale  $SS_E$ . ¿Qué valor estimamos que tiene  $\sigma^2$ ?
- Cuestión 5: Calculad la correlación entre los vectores de datos  $x = (1, 3, 4, 4)$ ,  $y = (2, 4, 12, 6)$ .
- Cuestión 6: De estas cuatro matrices, indicad cuáles pueden ser matrices de correlaciones, y explicad por qué.

$$A = \begin{pmatrix} 1 & 0.8 \\ -0.8 & 1 \end{pmatrix}, B = \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & 0.8 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, D = \begin{pmatrix} 1 & 1.2 \\ 1.2 & 1 \end{pmatrix}.$$

### 1.7.1 Solución

**Question 1:** Rechazamos la hipótesis nula para todos los niveles de significación  $\alpha$  menores que el  $p$ -valor.

**Question 2:** No, no podemos afirmar eso. Lo que sabemos después de rechazar un ANOVA es que hay al menos dos medias distintas.

**Question 3:** pues son

```
probs=c(1/6,1/6,1/6,1/6,1/6,1/6)
frec.esp=300*probs
frec.esp
```

```
## [1] 50 50 50 50 50 50
```

**Question 4**  $SS_E = SS_{Total} - SS_{Tr} = 256.6 - 60.3 = 196.3$ .

El número de observaciones totales es  $N = 11 \cdot 6 = 66$  y el número de niveles del factor es  $k = 6$ .

El estimador de la varianza conjunto  $\sigma^2$  es  $MS_E = \frac{SS_E}{N-k} = \frac{196.3}{66-6} = 17.8454545$ .

**\*Cuestión 5**

```
x=c(1,3,4,4)
y=c(2,4,12,6)
cor(x,y)
```

```
## [1] 0.7637626
```

**Cuestión 6**

Son matrices de correlaciones 2x2 sabemos que la diagonal tiene que ser 1 pues es la correlación de una variable consigo misma. También sabemos que  $\rho_{xy} = \rho_{yx}$  así que la matriz debe ser simétrica. Además  $-1 \leq \rho_{xy} \leq 1$ .

$A$  no es simétrica,  $B$  no tiene la diagonal de unos y  $D$  tienen valores mayores que 1 luego estas tres matrices no son de correlaciones. La única matriz que cumple todas las condiciones es la  $C$ .