Taller de problemas resueltos inferencia

Ricardo Alberich

Table of Contents

# Taller Problemas resueltos: Estadística Inferencial

Se trata de resolver los siguientes problemas y cuestiones en un fichero Rmd y su salida en un informe en html, word o pdf.

## 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 y . Disponemos de dos muestras independientes de cuestionarios realizados por estudiantes de cada grado, .

Los datos están en <https://github.com/joanby/estadistica-inferencial/>, en la carpeta datasets en dos ficheros grado1.txt y grado2.txt.

Para bajarlos utilizad la dirección del los ficheros raw que se muestran en el siguiente código

G1=read.csv(  
 "https://raw.githubusercontent.com/joanby/estadistica-inferencial/master/datasets/grado1.txt",  
 header=TRUE)$x  
G2=read.csv(  
 "https://raw.githubusercontent.com/joanby/estadistica-inferencial/master/datasets/grado2.txt",  
 header=TRUE)$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:

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 contrate con el -valor.
3. Calculad e interpretar 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 varianza 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 -valor.

### Solución

**Apartado 1.** El cogido R carga en las variables G1 y G2 la variables x de dos data frames de un servidor en github y por lo tanto hemos tenido que pasar la url del fichero original o *raw*.

Luego calcula los estadísticos básicos para realizar las siguientes preguntas. Para los tamaños muestrales y 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 y las medias de las notas de los grupos 1 y 2 respectivamente. El contraste que se pide es

$$
\left\{\begin{array}{ll}
H\_0:\mu\_{1} = \mu\_{2}\\
H\_1: \mu\_{1} \not= \mu\_{2}
\end{array}\right.
$$

Estamos en un diseño de comparación de medias de dos grupos con dos muestras independientes de tamaño 50 que es grande. Tenemos dos casos varianzas desconocidas pero iguales y varianzas desconocidas pero distintas. Las funciones de R del 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 -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 el cálculo manual.

**Varianzas desconocidas pero iguales, y grande**

Si suponemos que , el estadístico de contraste es

t0=(media.muestra1-media.muestra2)/  
 sqrt((1/n1+1/n2)\*   
((n1-1) \*desv.tip.muestra1^2+(n2-1)\*desv.tip.muestra2^2)/(n1+n2-2))  
t0

## [1] -6.215931

operando obtenemos que y sabemos que sigue una distribución -Student . Para este hipótesis alternativa el -valor es

, lo calculamos con R

t0=(media.muestra1-media.muestra2)/  
 sqrt((1/n1+1/n2)\*   
((n1-1)\*desv.tip.muestra1^2+(n2-1)\*desv.tip.muestra2^2)/(n1+n2-2))  
t0

## [1] -6.215931

n1

## [1] 50

n2

## [1] 50

2\*(1-pt(abs(t0),df=n1+n2-2)) # calculo la probabilidad del complementario

## [1] 0.00000001247958

2\*pt(abs(t0),df=n1+n2-2,lower.tail = FALSE)# calcula el área la cola superior

## [1] 0.00000001247958

**Varianzas desconocidas pero distintas, y grande**

Si suponemos que , el estadístico de contraste es que, cuando , tiene distribución (aproximadamente, en caso de muestras grandes) con

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 la función t.test de R para este caso.  
t0

## [1] -6.215931

f=(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)  
f

## [1] 89.99613

El valor es

# el p-valor de la función t.test de R  
2\*(pt(abs(t0),f,lower.tail = FALSE))

## [1] 0.00000001562353

**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 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

$$\left\{\begin{array}{ll}H\_0: & \sigma\_1^2=\sigma\_2^2\\
H\_1: & \sigma\_1^2\not=\sigma\_2^2\end{array}\right..$$

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 -valor alto no podemos rechazar la igualdad de varianzas.

De forma manual el estadístico de este test sabemos que es

Que sigue una ley de distribución de Fisher y el \_valor es

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)# pongo 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 -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.

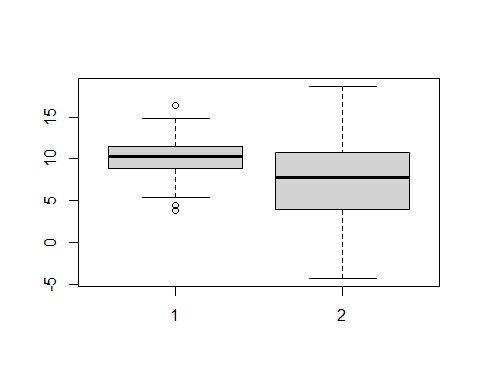
## Problema 2 : Contraste dos muestras

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

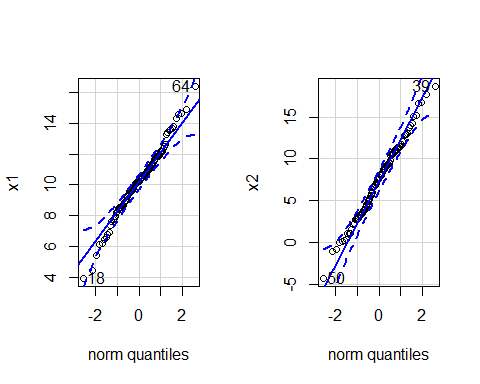
boxplot(x1,x2)



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.

### 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 contra . **Así que este TEST NO ES**

El segundo test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales contra $H\_1:\mu\_1\not=\mu\_2$. **Así que este TEST SÍ PUEDE SER** contrasta contra medias distintas para el caso de varianzas distintas.

El tercer test contrasta para muestras independientes supuestas varianzas desconocidas pero iguales contra pues la opción por defecto de la función. **Así que este TEST SÍ PUEDE SER** contra medias distintas para el caso de varianzas iguales.

**Apartado 4** El contrastes es

$$\left\{\begin{array}{ll}H\_0: & \mu\_1=\mu\_2\\ H\_1: & \mu\_1\not=\mu\_2\end{array}\right.
$$

En los dos últimos test los -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.

## 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 diremos que sigue una ley de Benford si

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

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 puede según qué versión pasar 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 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 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 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 -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

### Solución

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

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

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

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 -valor de no podemos rechazar que el segundo dígito siga una ley uniforme.

**Apartado 3**

El contraste que nos piden es

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)

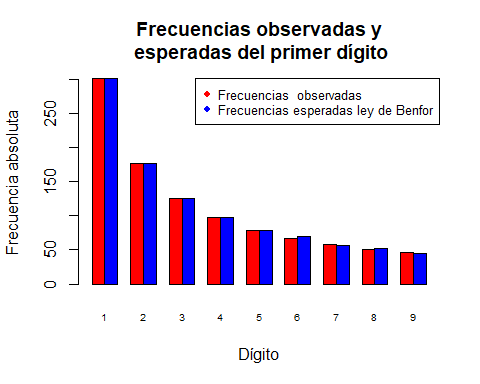
##   
## Chi-squared test for given probabilities  
##   
## data: frec\_obs\_tercer  
## X-squared = 7.9718, df = 9, p-value = 0.537

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

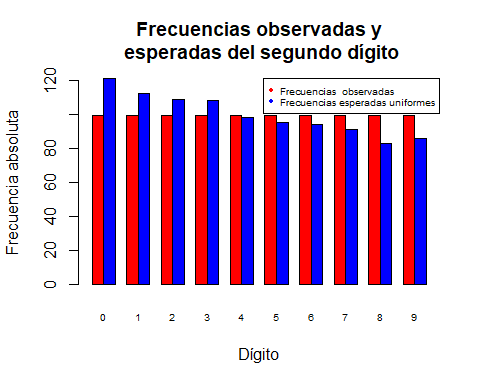
**Apartado 4**

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)



## Problema 4 : Homegeneidad e independencia

Queremos analiza 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.

### 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 : la nota obtenida es independientes ce 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 -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 en los tres casos

Por último al dividir por el tamaño total de las muestra obtenemos las frecuencias esperadas en el caso de independencia .

## 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 la 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 contrates realiza la función pairwise.t.test? Utilizando los resultados anteriores aplicad e interpretad los contrates del apartado anterior utilizando el ajuste de Holm.

### 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 -valores son altos en el pero de los caso es mayor que 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 -valores del orden de o 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.

**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 -valor es muy grade 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 -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 -valores son prácticamente son 1 con el ajuste del método de Holm; o podemos rechazar las igual de medias dos a dos.

## 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.*

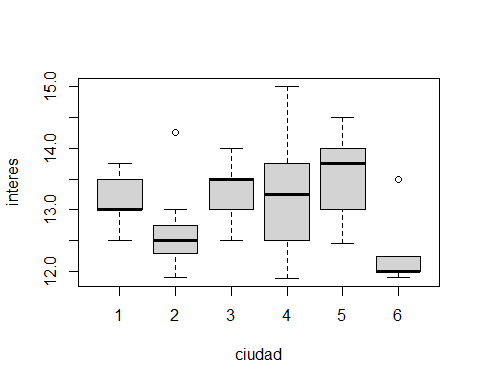
Este data set contiene dos columnas:

* 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(interes~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 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.
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 hacedlo con funciones de R.
5. Se acepte o no la igualdad de medias realizar las comparaciones dos a dos con ajustando los -valor tanto por Bonferroni como por Holm al nivel de significación . Redactad las conclusiones que se obtienen de las mismas.

### 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 lineas 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 para las seis ciudades .
* homocedasticidad; igualdad de varianzas

El ANOVA que se pide es

**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

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

Comprobamos por gusto el -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 , así que rechazaremos la igualdad de medias de todas las comparaciones con -valor inferior a .

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.

## Problema 7: Cuestiones cortas

* Cuestión 1: Supongamos que conocemos el -valor de un contraste. Para que valores de nivel de significación RECHAZAMOS la hipótesis nula.
* Cuestión 2: Hemos realizado un ANOVA de un factor con 3 niveles, y hemos obtenido un -valor de 0.001. Suponiendo que las poblaciones satisfacen las condiciones para que el ANOVA tenga sentido, ¿podemos afirmar con un nivel de significación 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 serian 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 . Estudiamos a 11 individuos de cada nivel y obtenemos que y . ¿Qué vale . ¿Qué valor estimamos que tiene ?
* Cuestión 5: Calculad la correlación entre los vectores de datos , .
* Cuestión 6: De estas cuatro matrices, indicad cuáles pueden ser matrices de correlaciones, y explicad por qué.

, , , .

### Solución

**Cuestion 1**: Rechazamos la hipótesis nula para todos los niveles de significación menores que el -valor.

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

**Cuestion 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

**Cuestion 4** $SS\_E=SS\_{Total}-SS-{Tr}=256.6-60.3=196.3.

El número de observaciones totales es y el número de niveles del factor es .

El estimador de la varianza conjunto es

\***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 así que la matriz debe ser simétrica. Además .

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

## Problema 8: Contraste de proporciones de dos muestras independientes.

Queremos comparar las proporciones de aciertos de dos redes neuronales que detectan si una foto con un móvil de una avispa es una [avispa velutina o asiática](https://es.wikipedia.org/wiki/Vespa_velutina) o si es una avispa común. Esta avispa en una especie invasora y peligrosa por el veneno de su picadura. Para ello disponemos de una muestra de 1000 imágenes de insectos etiquetadas como avispa velutina y no velutina.

[Aquí tenéis el acceso a los datos](http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina). Cada uno está en fichero selecciona 500 fotos de de forma independiente para el algoritmo 1 y el 2. Los aciertos están codificados con 1 y los fallos con 0.

Se pide:

1. Cargad los datos desde el servidos y calcular el tamaño de las muestras y la proporción de aciertos de cada muestra.
2. Contrastad si hay evidencia de que las las proporciones de aciertos del algoritmo 1 son mayores que las del algoritmo 2. Definid bien las hipótesis y las condiciones del contraste. Tenéis que hacer el contraste con funciones de R y resolver el contrate con el -valor.
3. Calculad e interpretar los intervalos de confianza para la diferencia de proporciones asociados al test anterior, con funciones de R.

### Solución

algoritmo1=read.table("http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina/algoritmo1.csv")  
algoritmo2=read.table("http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina/algoritmo2.csv")

Proporción aciertos de cada algoritmo

n1=dim(algoritmo1)[1]  
n1

## [1] 500

n1=length(algoritmo1$V1)  
n1

## [1] 500

n2=length(algoritmo2$V1)  
n2

## [1] 500

aciertos\_absolutos\_algoritmo1=table(algoritmo1)["1"]  
aciertos\_absolutos\_algoritmo1

## 1   
## 396

p1=prop.table(table(algoritmo1))["1"]  
p1

## 1   
## 0.792

aciertos\_absolutos\_algoritmo2=table(algoritmo2)["1"]  
aciertos\_absolutos\_algoritmo2

## 1   
## 437

p2=prop.table(table(algoritmo2))["1"]  
p2

## 1   
## 0.874

Después de los cálculos preliminares si denotamos las proporciones poblacionales de aciertos de cada algoritmo por y respectivamente, el contraste que nos piden es

estamos ante un diseño de comparación de proporciones con muestras independientes. Con R lo podemos resolver con el fisher.test o con el prop.test

x=matrix(c(aciertos\_absolutos\_algoritmo1,  
 n1-aciertos\_absolutos\_algoritmo1,  
 aciertos\_absolutos\_algoritmo2,  
 n2-aciertos\_absolutos\_algoritmo2),  
 ncol=2,byrow=FALSE)  
x

## [,1] [,2]  
## [1,] 396 437  
## [2,] 104 63

fisher.test(x,alternative="greater",conf.level=0.95)

##   
## Fisher's Exact Test for Count Data  
##   
## data: x  
## p-value = 0.9998  
## alternative hypothesis: true odds ratio is greater than 1  
## 95 percent confidence interval:  
## 0.4056457 Inf  
## sample estimates:  
## odds ratio   
## 0.5492712

c(aciertos\_absolutos\_algoritmo1,  
 aciertos\_absolutos\_algoritmo2)

## 1 1   
## 396 437

c(n1,n2)

## [1] 500 500

prop.test(c(aciertos\_absolutos\_algoritmo1,  
 aciertos\_absolutos\_algoritmo2), c(n1,n2),  
 alternative="greater",conf.level=0.95)

##   
## 2-sample test for equality of proportions with continuity correction  
##   
## data: c(aciertos\_absolutos\_algoritmo1, aciertos\_absolutos\_algoritmo2) out of c(n1, n2)  
## X-squared = 11.502, df = 1, p-value = 0.9997  
## alternative hypothesis: greater  
## 95 percent confidence interval:  
## -0.1225654 1.0000000  
## sample estimates:  
## prop 1 prop 2   
## 0.792 0.874

Con ambos test obtenemos valores altos (el más pequeño es el de Fisher y es mayor que , así que no podemos rechazar que las proporciones de aciertos de los dos algoritmos sean iguales contra que la proporción de aciertos del algoritmo 1 es mejor que la del 2.

El intervalo de confianza asociado a este test es

prop.test(c(aciertos\_absolutos\_algoritmo1,  
 aciertos\_absolutos\_algoritmo2),  
 c(n1,n2),  
 alternative="greater",  
 conf.level=0.95)$conf.int

## [1] -0.1225654 1.0000000  
## attr(,"conf.level")  
## [1] 0.95

luego con una probabilidad del 95% la contiene el 0 y no podemos despreciar que sean iguales contra que

## Problema 9 : Contraste de proporciones de dos muestras emparejadas.

En el problema anterior hemos decidido quedarnos con el mejor de los algoritmos y mejorarlo. Pasamos las mismas 1000 imágenes a la version\_beta del algoritmo y a la version\_alpha. [Aquí tenéis el acceso a los datos en el mismo orden para las 1000 imágenes](http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina2). Cada uno está en fichero los aciertos están codificados con 1 y los fallos con 0.

1. Cargad los datos desde el servidos y calcular el tamaño de las muestras y la proporción de aciertos de cada muestra.
2. Contrastad si hay evidencia de que las las proporciones de aciertos del algoritmo alfa son iguales que las del algoritmo beta. Definid bien las hipótesis y las condiciones del contraste. Tenéis que hacer el contraste con funciones de R y resolver el contrate con el -valor.

### Solución

Cargamos los datos y hacemos los cálculos preliminares

algoritmoalfa=read.table("http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina2/algoritmo\_alpha.csv")  
algoritmobeta=read.table("http://bioinfo.uib.es/~recerca/MATIIIGINF/velutina2/algoritmo\_beta.csv")

El test que nos piden es

$$
\left\{
\begin{array}{ll}
H\_0: & p\_{\alpha}=p\_{\beta}\\
H\_1: & p\_{\alpha}\not=p\_{\beta}\\
\end{array}
\right.
$$

Es un diseño de muestras emparejadas y tenemos que utilizar el mcnear.test:

X=table(algoritmoalfa$V1,algoritmobeta$V1)  
X

##   
## 0 1  
## 0 15 110  
## 1 88 787

mcnemar.test(X)

##   
## McNemar's Chi-squared test with continuity correction  
##   
## data: X  
## McNemar's chi-squared = 2.2273, df = 1, p-value = 0.1356

El -valor es 0.1356 no podemos rechazar la igualdad de la proporción de aciertos.

## Problema 10 : ANOVA comparación media puntuaciones según fabricante.

Una vez mejorado nuestro algoritmo queremos saber su comportamiento bajo distintos tipos de móviles.

Seleccionamos 6 móviles de la misma gama de calidad de 6 fabricantes distintos. A los fabricantes los denotamos por F1, F2, F3, F4, F5 y F6.

Vamos a jugar no con la clasificación sino con el score que produce el algoritmo. Para ello seleccionamos 4 muestra aleatorias de fotos de insectos enviadas por los usuarios y la puntuación (*score*) que nos da el algoritmo que es una variable aleatoria continua de con rango de 0 a 100.

La idea es comprobar si la media de las puntuaciones del algoritmo es la misma para cada uno de los fabricantes.

Los datos los podéis descargar de esta dirección del [servidor bioinfo.uib.es](http://bioinfo.uib.es/~recerca/MATIIIGINF/anova_score/).

Antes de descargarlo, visualizar el fichero desde el navegador, para saber cómo descargarlo.

1. ¿Podemos asegurar que la muestras son normales en cada grupo? ¿y son homocedásticas? Justificar la respuesta con el correspondiente código en R comentado.
2. Escribid formalmente la hipótesis nula y la alternativa. Calcular la tabla de ANOVA y resuelve el test de forma manual.
3. Calcular la tabla de ANOVA y resuelve el test con la función aov de R.
4. Haced una comparación de pares con la función adecuada de R para la corrección del holm al nivel de significación . Interpreta el resultado.
5. Comparar por grupos con el test de Duncan del paquete agricolae. Interpreta el resultado.

### Solución

df=read.table("http://bioinfo.uib.es/~recerca/MATIIIGINF/anova\_score/score\_manufacturer.csv")  
head(df)

## score manufacturer  
## 1 69.32030 F1  
## 2 66.93433 F1  
## 3 67.70541 F1  
## 4 63.47195 F1  
## 5 65.58738 F1  
## 6 65.47437 F1

df$manufacturer=as.factor(df$manufacturer)

table(df$manufacturer)

##   
## F1 F2 F3 F4 F5 F6   
## 100 100 100 100 100 100

Tenemos que comprobar la normalidad de la distribución de la muestra para cada nivel del factor ; el test es

library(nortest)  
# El test KS\_Lillie para en nivel "F1"  
#lillie.test(df$score[df$manufacturer=="F1"])  
sapply(levels(df$manufacturer), FUN=function(x) {print(lillie.test(df$score[df$manufacturer==x]))})

##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.091505, p-value = 0.03825  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.067758, p-value = 0.3121  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.069567, p-value = 0.2744  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.069567, p-value = 0.2744  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.069567, p-value = 0.2744  
##   
##   
## Lilliefors (Kolmogorov-Smirnov) normality test  
##   
## data: df$score[df$manufacturer == x]  
## D = 0.10632, p-value = 0.007255

## F1   
## statistic 0.09150477   
## p.value 0.03825059   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"   
## F2   
## statistic 0.06775771   
## p.value 0.3121052   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"   
## F3   
## statistic 0.06956719   
## p.value 0.2743622   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"   
## F4   
## statistic 0.06956719   
## p.value 0.2743622   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"   
## F5   
## statistic 0.06956719   
## p.value 0.2743622   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"   
## F6   
## statistic 0.1063201   
## p.value 0.007255259   
## method "Lilliefors (Kolmogorov-Smirnov) normality test"  
## data.name "df$score[df$manufacturer == x]"

# También podemos hacer un bucle clásico  
#for(Fabricante in levels(df$manufacturer)){  
#print(lillie.test(df$score[df$manufacturer==Fabricante]))  
#}

El nivel “F1” y “F6” dan valores pequeños no podemos asegurar la normalidad en estos casos.

Nos aseguramos con el ómnibus test de D’Agostino

library(fBasics)

## Loading required package: timeDate

## Loading required package: timeSeries

##   
## Attaching package: 'fBasics'

## The following object is masked from 'package:car':  
##   
## densityPlot

dagoTest(df$score[df$manufacturer=="F1"])

##   
## Title:  
## D'Agostino Normality Test  
##   
## Test Results:  
## STATISTIC:  
## Chi2 | Omnibus: 5.8827  
## Z3 | Skewness: 2.1113  
## Z4 | Kurtosis: 1.1939  
## P VALUE:  
## Omnibus Test: 0.05279   
## Skewness Test: 0.03475   
## Kurtosis Test: 0.2325   
##   
## Description:  
## Wed May 26 10:24:49 2021 by user: t169

dagoTest(df$score[df$manufacturer=="F6"])

##   
## Title:  
## D'Agostino Normality Test  
##   
## Test Results:  
## STATISTIC:  
## Chi2 | Omnibus: 1.7625  
## Z3 | Skewness: -1.2831  
## Z4 | Kurtosis: 0.3408  
## P VALUE:  
## Omnibus Test: 0.4143   
## Skewness Test: 0.1995   
## Kurtosis Test: 0.7333   
##   
## Description:  
## Wed May 26 10:24:49 2021 by user: t169

Parece que no podemos rechazar la normalidad para los casos dudosos.

Ahora realizamos el test de comparación de medias para son las medias para cada nivel del factor.

summary(aov(df$score~df$manufacturer))

## Df Sum Sq Mean Sq F value Pr(>F)   
## df$manufacturer 5 9143 1828.5 17.54 0.000000000000000317 \*\*\*  
## Residuals 594 61910 104.2   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Como el -valor es muy pequeño NO podemos aceptar que las 6 medias sean iguales

Ahora tenemos que contrastar que pares de medias dos a dos son iguales y ajustar los -valores con el ajuste del -valor por el método de Holm

pairwise.t.test(df$score,df$manufacturer,p.adjust.method = "holm")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: df$score and df$manufacturer   
##   
## F1 F2 F3 F4 F5   
## F2 1.00000 - - - -   
## F3 0.07729 0.44250 - - -   
## F4 0.00011 0.00000297115 0.00000000017 - -   
## F5 0.00011 0.00000297115 0.00000000017 1.00000 -   
## F6 0.00724 0.00040 0.00000014687 1.00000 1.00000  
##   
## P value adjustment method: holm

Comparamos los -valores con y aceptamos que las medias de los niveles , y son iguales dos a dos, también son iguales la media del con el , y la media del nivel con el . EL resto de comparaciones tienen -valores bajos así que no podemos aceptar la igualdad de medias.

Ahora comparamos las medias por grupos de igualdades con el test de Duncan

library(agricolae)

##   
## Attaching package: 'agricolae'

## The following objects are masked from 'package:timeDate':  
##   
## kurtosis, skewness

resultado.anova=aov(df$score~df$manufacturer)  
duncan.test(resultado.anova,"df$manufacturer",group=TRUE)$group

## df$score groups  
## F3 74.07499 a  
## F2 71.61166 ab  
## F1 70.47337 b  
## F6 65.71299 c  
## F4 64.07499 c  
## F5 64.07499 c

Obtenemos tres grupos el **a** dice la media de el **b** dice que y el **c** dice que . Obtenemos conclusiones similares al test de comparación de medias.

## Problema 11: Regresión lineal simple.

Consideremos los siguientes datos

x=c(-2,-1,2,0,1,2)  
y=c(-7, -5, 5, -3, 3.0, 4)  
summary(lm(y~x))

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## 1 2 3 4 5 6   
## 0.675 -0.400 0.375 -1.475 1.450 -0.625   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.5250 0.4872 -3.130 0.035176 \*   
## x 3.0750 0.3189 9.642 0.000647 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.165 on 4 degrees of freedom  
## Multiple R-squared: 0.9587, Adjusted R-squared: 0.9484   
## F-statistic: 92.96 on 1 and 4 DF, p-value: 0.0006472

1. Calcular manualmente los coeficiente de la regresión lineal de y sobre x
2. Calcular los valores para los valores de la muestra y el error cometido.
3. Calcular la estimación de la varianza del error.
4. Resolver manualmente el contraste $\left\{\begin{array}{ll} H\_0: & \beta\_1=0 \\ H\_1: & \beta\_1\not=0\end{array}\right. ,$ calculando el -valor.
5. Calcular , y .
6. Calcular el coeficiente de regresión lineal y el coeficiente de determinación . Interpretad el resultado en términos de la cantidad de varianza explicada por el modelo
7. Comprobar que los resultados son los mismos que los obtenidos con la función summary(lm(y~x)).

### Solución

Faltan añadir los NECESARIOS COMENTARIOS.

x=c(-2,-1,2,0,1,2)  
y=c(-7, -5, 5, -3, 3.0, 4)  
sol\_lm=lm(y~x)  
summary(sol\_lm)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## 1 2 3 4 5 6   
## 0.675 -0.400 0.375 -1.475 1.450 -0.625   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.5250 0.4872 -3.130 0.035176 \*   
## x 3.0750 0.3189 9.642 0.000647 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.165 on 4 degrees of freedom  
## Multiple R-squared: 0.9587, Adjusted R-squared: 0.9484   
## F-statistic: 92.96 on 1 and 4 DF, p-value: 0.0006472

mediay=mean(y)  
mediax=mean(x)  
sdx=sd(x)  
sdy=sd(y)  
sxy=cov(x,y)  
b1=sxy/sdx^2  
b1

## [1] 3.075

b0=mediay-b1\*mediax  
b0

## [1] -1.525

sol\_lm$coefficients

## (Intercept) x   
## -1.525 3.075

c(b0,b1)==sol\_lm$coefficients# dan distintos errores de redondeo

## (Intercept) x   
## FALSE TRUE

near(c(b0,b1),sol\_lm$coefficients)# opcional

## (Intercept) x   
## TRUE TRUE

sol\_lm$fitted.values

## 1 2 3 4 5 6   
## -7.675 -4.600 4.625 -1.525 1.550 4.625

recta=function(x) b0+b1\*x  
y\_est=recta(x)  
y\_est

## [1] -7.675 -4.600 4.625 -1.525 1.550 4.625

predict(sol\_lm,newdata = data.frame(x=x))

## 1 2 3 4 5 6   
## -7.675 -4.600 4.625 -1.525 1.550 4.625

y

## [1] -7 -5 5 -3 3 4

y\_est

## [1] -7.675 -4.600 4.625 -1.525 1.550 4.625

e=y-y\_est  
e

## [1] 0.675 -0.400 0.375 -1.475 1.450 -0.625

sol\_lm$residuals

## 1 2 3 4 5 6   
## 0.675 -0.400 0.375 -1.475 1.450 -0.625

mean(e) # es cero, pero por error de redondeo no da exacto.

## [1] -0.0000000000000002220446

SSE=sum(e^2)  
SSE

## [1] 5.425

n=length(x)  
n

## [1] 6

S2=SSE/(n-2)#estimacion\_var\_error  
S2

## [1] 1.35625

S=sqrt(S2)# Residual estándar error: 1.165  
S

## [1] 1.164581

round(S,3) # con los mismos decimales da lo mismo

## [1] 1.165

# contraste beta1=0  
t0=b1/(S/(sdx\*sqrt(n-1)))  
t0

## [1] 9.6415

2\*pt(abs(t0),n-2,lower.tail = FALSE)

## [1] 0.0006472191

2\*(1-pt(abs(t0),n-2,lower.tail = TRUE))

## [1] 0.0006472191

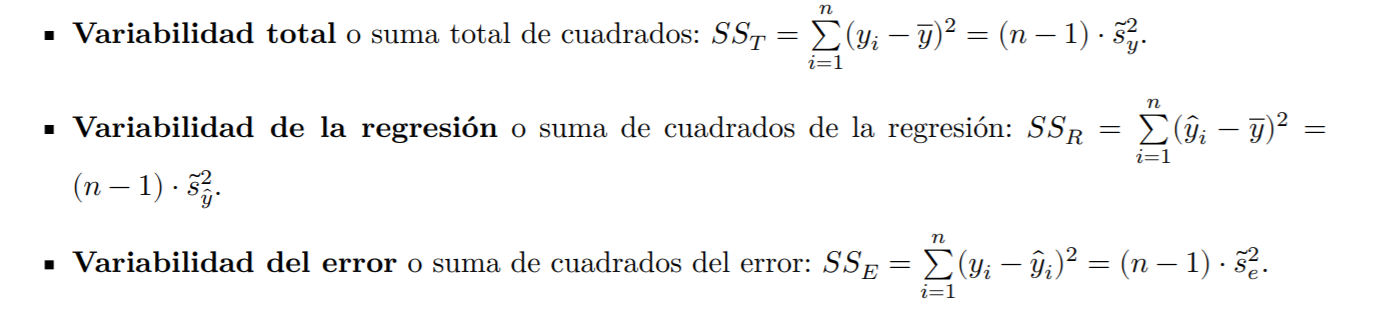
2\*(1-pt(abs(t0),n-2))

## [1] 0.0006472191

comparar con

Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -1.5250 0.4872 -3.130 0.035176 \*   
x 3.0750 0.3189 9.642 0.000647 \*\*\*

knitr::include\_graphics("formulas\_regre.PNG")



SST=sum((y-mean(y))^2)  
SST

## [1] 131.5

mean(y\_est)

## [1] -0.5

mean(y)#media estimados regresión igual a media variable y

## [1] -0.5

SSR=sum((y\_est-mean(y))^2)  
SSR

## [1] 126.075

SSE# ya lo había a calculado

## [1] 5.425

SST-SSR# da lo mismo pues SST=SSR+SSE

## [1] 5.425

R2=SSR/SST  
R2

## [1] 0.9587452

cor(x,y)

## [1] 0.9791554

cor(x,y)^2# en el caso regresión simple R2=cor(xy)^2

## [1] 0.9587452

## Problema 12: Distribución de los grados de un grafo de contactos.

En el artículo de A. Broder et al., [Graph structure in the Web. Computer Networks 33, 309 (2000)](http://snap.stanford.edu/class/cs224w-readings/broder00bowtie.pdf).

Se recopiló el número de enlaces a sitios web encontrados en un rastreo web de 1997 de aproximadamente 200 millones de páginas web,

Con el se construyó una [tabla](http://tuvalu.santafe.edu/~aaronc/powerlaws/data/weblinks.hist) con la frecuencia de sitios por número de enlaces. El código siguiente carga del enlace que han puesto los autores del artículo

data\_links=read.table("http://tuvalu.santafe.edu/~aaronc/powerlaws/data/weblinks.hist",header=TRUE)  
head(data\_links)

## degree frequency  
## 1 0 35159835  
## 2 1 106649769  
## 3 2 40711748  
## 4 3 22648832  
## 5 4 12617832  
## 6 5 8188854

str(data\_links)

## 'data.frame': 14480 obs. of 2 variables:  
## $ degree : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ frequency: int 35159835 106649769 40711748 22648832 12617832 8188854 6438634 4690068 4954649 3731928 ...

# eliminamos la páginas con menos de 8 enlaces enlaces y las de más de 1000 enlaces  
data\_links\_central=data\_links[data\_links$degree>8&data\_links$degree<10^3,]  
head(data\_links\_central)

## degree frequency  
## 10 9 3731928  
## 11 10 3036333  
## 12 11 2496648  
## 13 12 2119312  
## 14 13 1790068  
## 15 14 1546579

tail(data\_links\_central)

## degree frequency  
## 995 994 213  
## 996 995 193  
## 997 996 157  
## 998 997 137  
## 999 998 178  
## 1000 999 153

El siguiente código calcula las regresiones exponencial, potencial y lineal (en algún orden) de las frecuencias (frequency) contra los enlaces (degree).

sol1=lm(frequency~ degree,data=data\_links\_central)  
summary(sol1)

##   
## Call:  
## lm(formula = frequency ~ degree, data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -96861 -69548 -25033 22374 3598744   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 134974.49 13778.17 9.796 <0.0000000000000002 \*\*\*  
## degree -198.98 23.77 -8.369 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 214100 on 989 degrees of freedom  
## Multiple R-squared: 0.06614, Adjusted R-squared: 0.06519   
## F-statistic: 70.04 on 1 and 989 DF, p-value: < 0.00000000000000022

sol2=lm(log10(frequency)~ degree,data=data\_links\_central)  
summary(sol2)

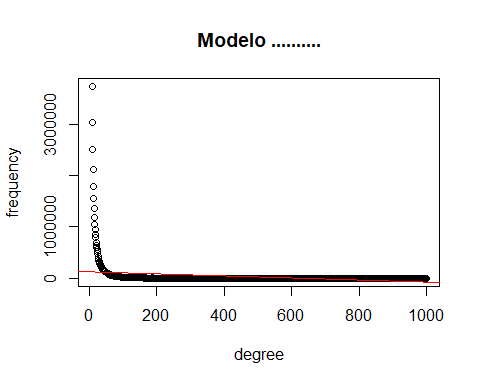
##   
## Call:  
## lm(formula = log10(frequency) ~ degree, data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.43758 -0.26558 -0.07671 0.16681 2.13097   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.46504979 0.02381018 187.53 <0.0000000000000002 \*\*\*  
## degree -0.00267658 0.00004109 -65.15 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.37 on 989 degrees of freedom  
## Multiple R-squared: 0.811, Adjusted R-squared: 0.8108   
## F-statistic: 4244 on 1 and 989 DF, p-value: < 0.00000000000000022

sol3=lm(log10(frequency)~ log10(degree),data=data\_links\_central)  
summary(sol3)

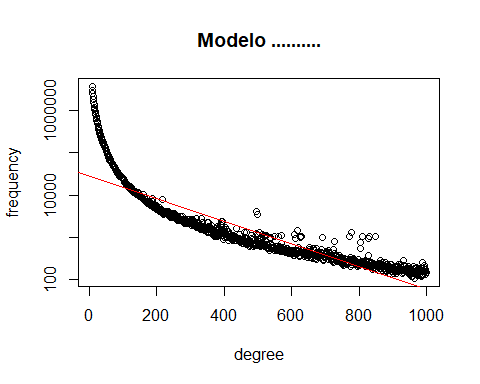
##   
## Call:  
## lm(formula = log10(frequency) ~ log10(degree), data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.21376 -0.04747 -0.01555 0.01958 0.73976   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.722036 0.020623 422.9 <0.0000000000000002 \*\*\*  
## log10(degree) -2.170129 0.007894 -274.9 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09674 on 989 degrees of freedom  
## Multiple R-squared: 0.9871, Adjusted R-squared: 0.9871   
## F-statistic: 7.557e+04 on 1 and 989 DF, p-value: < 0.00000000000000022

Ahora dibujamos los gráficos adecuados a cada modelo

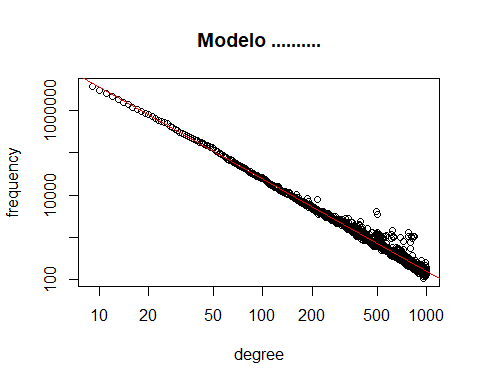
plot(data\_links\_central,main="Modelo ..........")  
abline(sol1,col="red")



plot(data\_links\_central,main="Modelo ..........",log="y")  
abline(sol2,col="red")



plot(data\_links\_central,main="Modelo ..........",log="xy")  
abline(sol3,col="red")



Se pide:

1. Explicad el modelo de regresión que calcula cada función lm
2. ¿Qué modelo y en función de qué parámetros es el mejor?
3. Para el mejor modelo calcular los coeficientes en las unidades originales y escribir la ecuación del modelos.

### Solución

**Apartado 1**

Los modelos son

sol1=lm(frequency~ degree,data=data\_links\_central)  
summary(sol1)

##   
## Call:  
## lm(formula = frequency ~ degree, data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -96861 -69548 -25033 22374 3598744   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 134974.49 13778.17 9.796 <0.0000000000000002 \*\*\*  
## degree -198.98 23.77 -8.369 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 214100 on 989 degrees of freedom  
## Multiple R-squared: 0.06614, Adjusted R-squared: 0.06519   
## F-statistic: 70.04 on 1 and 989 DF, p-value: < 0.00000000000000022

sol2=lm(log10(frequency)~ degree,data=data\_links\_central)  
summary(sol2)

##   
## Call:  
## lm(formula = log10(frequency) ~ degree, data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.43758 -0.26558 -0.07671 0.16681 2.13097   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.46504979 0.02381018 187.53 <0.0000000000000002 \*\*\*  
## degree -0.00267658 0.00004109 -65.15 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.37 on 989 degrees of freedom  
## Multiple R-squared: 0.811, Adjusted R-squared: 0.8108   
## F-statistic: 4244 on 1 and 989 DF, p-value: < 0.00000000000000022

sol3=lm(log10(frequency)~ log10(degree),data=data\_links\_central)  
summary(sol3)

##   
## Call:  
## lm(formula = log10(frequency) ~ log10(degree), data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.21376 -0.04747 -0.01555 0.01958 0.73976   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.722036 0.020623 422.9 <0.0000000000000002 \*\*\*  
## log10(degree) -2.170129 0.007894 -274.9 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09674 on 989 degrees of freedom  
## Multiple R-squared: 0.9871, Adjusted R-squared: 0.9871   
## F-statistic: 7.557e+04 on 1 and 989 DF, p-value: < 0.00000000000000022

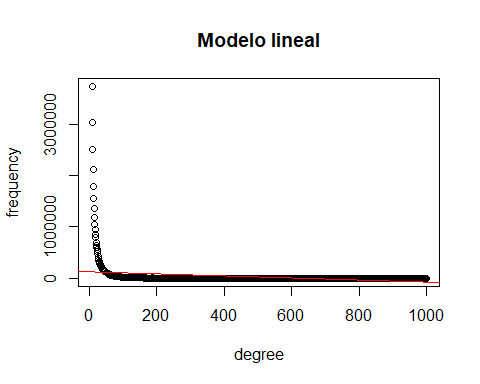
Así que

* el primero es que es el modelo LINEAL que resuelve la función lm.
* el segundo es que operando adecuadamente es

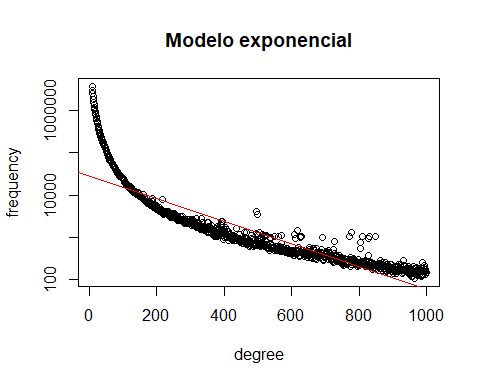
operando obtenemos que el modelo final es un modelo EXPONENCIAL . \* el tercer modelo es que despejando es operando obtenemos que el modelo final es un modelo POTENCIAL

Los dibujos con el título adecuado son

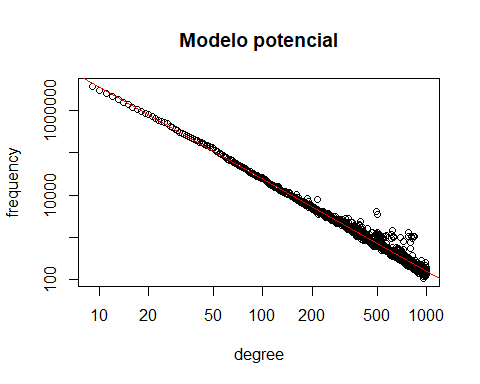
plot(data\_links\_central,main="Modelo lineal")  
abline(sol1,col="red")



plot(data\_links\_central,main="Modelo exponencial",log="y")  
abline(sol2,col="red")



plot(data\_links\_central,main="Modelo potencial",log="xy")  
abline(sol3,col="red")



Con lo que hemos visto el modelo con mejor es el mejor

summary(sol1)$r.squared

## [1] 0.06613879

summary(sol2)$r.squared

## [1] 0.8110118

summary(sol3)$r.squared

## [1] 0.9870815

Así que el mejor modelo es el tercero el potencial pues su es muy alto

**Apartado 3**

Calcularemos la ecuación del modelo para las unidades originales sin transformaciones logarítmicas.

Recordemos el resultado de la sol3

summary(sol3)

##   
## Call:  
## lm(formula = log10(frequency) ~ log10(degree), data = data\_links\_central)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.21376 -0.04747 -0.01555 0.01958 0.73976   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.722036 0.020623 422.9 <0.0000000000000002 \*\*\*  
## log10(degree) -2.170129 0.007894 -274.9 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09674 on 989 degrees of freedom  
## Multiple R-squared: 0.9871, Adjusted R-squared: 0.9871   
## F-statistic: 7.557e+04 on 1 and 989 DF, p-value: < 0.00000000000000022

Los estimadores son y el modelo es . Sustituyendo los valores el modelo obtenemos , finalmente operando

No se pedía en el ejercicio pero hacemos el dibujo de los datos y la ecuación en las unidades originales

frequency=data\_links\_central$frequency  
degree=data\_links\_central$degree  
potencial=function(x) 10^(8.722036)\*x^(-2.170129)  
plot(degree,frequency,main="Modelo potencial",xlab="degree",ylab="frequency")  
curve(potencial,xlim=c(0,1000),ylim=c(0,3731928),col="red",add=TRUE)

