Tarea sobre los Temas 3 a 5

Grado en Estadística. Universidad de Sevilla

Marta Venegas Pardo

Contents

1 Pregunta 1 (datasets::precip)

En el dataset datasets::precip se recoge la cantidad media de precipitaciones de 70 ciudades de Estados Unidos (unidad = inches).

Extraigo los datos

```
precip<-datasets::precip
# View(precip)
```

```
#dotchart(precip[order(precip)], main = "precip data")
#title(sub = "Average annual precipitation (in.)")
```

```
precip_data<-as.data.frame(precip)
names<-names(precip)
precipdf<-cbind.data.frame(names,precip_data)
colnames(precipdf)<-c("City","Precip")



cbind.data.frame(precipdf[1:35,],precipdf[36:70,])%>%
  kable(booktabs=TRUE,longtable=TRUE) %>%
```

```
  kable_styling(latex_options = c("striped","scale_down")) %>%
  footnote(general = "Annual Precipitation in US Cities")
```

1.1 Realizar una estimación no paramétrica de la función de densidad por el método del núcleo.

```
x=precipdf[,2]
hist(precipdf[,2], br=10,prob=TRUE, main="Precipitación",
     col="gray",
     xlab="Precipitaciones (litros)", ylab="Densidad estimada",
     ylim = c(0,0.05),xlim = c(0,70))
lines(density(x, bw="SJ"),lwd=2,col="red")
rug(x)
```

Hemos elegidio el método SJ, que implementa la propuesta de Sheather y Jones, basado en la estimación del núcleo de f''

1.2 Realizar una estimación no paramétrica de la función de densidad por el método de los logsplines.

Estima el logaritmo de la función de densidad mediante un spline cúbico.

```
library(logspline)
ajuste <- logspline(precip_data)
ajuste # 7nudos, criterio BIC
```

Vamos a comprobar los valores AIC

```
resul<- ajuste$logl
nudos<- resul[,1]
logL<- resul[,3]
AIC<- -2*logL+log(length(precip_data))*(nudos-1)
AIC
```

Lo dibujamos

```
plot(nudos,AIC,type="l")
```

Ahora el histograma

```
hist(x,br=30, prob=TRUE,
     main="Histograma y estimac. de la densidad",
     ylab = expression(hat(f)(x)),
     xlab="x")

lines(density(precipdf[,2],bw="SJ"),col="blue",lwd=2)

plot(ajuste,col="red",lwd=2,add=TRUE)

legend("topright",
       col=c("blue","red"),
       lwd=2,
       legend=c("KERNEL","Logspline"))
```

1.3 Estimar P[precip>42] y el cuantil 0.90.

```
1-plogspline(42, ajuste)
```

```
qlogspline(0.9, ajuste)
```

2 Pregunta 2

Cuando cierta empresa recibe una invitación para optar a un contrato, la oferta no se puede completar hasta que sea revisada por cuatro departamentos: Ingeniería, Personal, Legal y Contabilidad. Los departamentos empiezan a trabajar al mismo tiempo, pero lo hacen de forma independiente. El tiempo en semanas que emplean en completar la revisión es una variable aleatoria con las siguientes distribuciones:

- Ingeniería: Exponencial con media 3 semanas
- Personal: Normal con media 4 y desviación típica 1;
- Legal: 2 o 4 semanas, siendo ambos valores equiprobables
- Contabilidad: Uniforme continua en el intervalo (1,5).

Se trata de simular el tiempo W que tarda la empresa en preparar una oferta. Para ello se pueden implementar los siguientes pasos:

2.1 Fijar M (número de ofertas, se recomienda al menos 1000).

```
M=1000 # número de ofertas
```

2.2 Definir una matriz Mx4 donde se irán almacenando los valores generados.

```
matriz=matrix(NA,M,4) # Posteriormente almacenaremos los datos
dim(matriz)
```

2.3 Repetir M veces:

2.3.1 Generar de forma independiente los cuatro tiempos según las cuatro distribuciones.

```
W = rep(NA,M)
for (i in 1:M) { # Guardar esos cuatro tiempos en una fila de la matriz
  vIng = rexp(1,1/3)
  VPer = rnorm(1,4,1)
  x=c(2,4)
  vLeg = sample(x,1,replace = T)
  vCon = runif(1,1,5)


# Para cada i genero un valor de W
  W[i] = max(vIng,VPer,vLeg,vCon) #Calcular W como el máximo de los cuatro tiempos.
}
# Al final tendrás una m.a.s. de valores W de tamaño M: W_1, ..., W_M
```

Creo la matriz

```
m= matrix(W,ncol = 4)
colnames(m)<- c("Ingeniería","Personal","Legal","Contabilidad")
head(m)
```

Se pide:

- Estudiar gráficamente la distribución de la variable aleatoria W “tiempo que transcurre hasta completar la oferta”.

```
hist(W,
     br=30,
     prob=TRUE,
```

```
     main="Histograma y estimación de la densidad",
     ylab=expression(hat(f)(x)),xlab="x")
lines(density(W,bw="SJ"),col="blue",lwd=2)
```

Estimar su media y su mediana.

```
apply(X=m,FUN=median,MARGIN = 2) # Mediana por columnas
```

```
apply(X=m,FUN=mean,MARGIN = 2) # Media por columnas
```

Para los datos totales, sin hacerlo por departamento:

Estimación del tiempo medio

```
mean(W)
```

Estimación de la mediana:

```
median(W)
```

2.3.2 Estimar la probabilidad de que W supere las 6 semanas.

$$P[W > 6]$$

```
ajuste<- logspline(m)
ajuste
```

Luego, la probabilidad será:

```
1-plogspline(6, ajuste)
```

2.3.3 ¿Cuál es el departamento que suele tardar más en completar la revisión?

```
m %>%
  colSums()/M
```

El tiempo medio de cada departamento, por tanto, vemos que el departamento de personal es el que es más lento a la hora de hacer una revisión completa.

2.3.4 ¿Cuál es la ordenación más frecuente de los cuatro tiempos?

```
library(modeest)

apply(X=m,FUN=mfv,MARGIN = 2)
```

```
mfv(W)
```

4 unidades de tiempo (horas)

3 Pregunta 3

Generar aleatoriamente un conjunto de datos donde tengo sentido construir un modelo de clasificación o de predicción. Ajustar el modelo y estimar su capacidad de generalización mediante Jackknife y mediante Validación Cruzada (K=10).

3.1 Generamos los datos con la función sample(aleatoriamente).

Generaremos un dataset con 120 filas y 3 variables, por ejemplo.

```
V=c(rep("Z1",70),rep("Z2",50))
V1=runif(120,5,13)
V2=rnorm(120,80,25)
V3=rnorm(120,30,5)


dataSet=cbind.data.frame(V,V1,V2,V3)

head(dataSet)
```

```
summary(dataSet)
```

```
dataSet$V = as.factor(dataSet$V)
summary(dataSet)
```

3.2 Modelo

Elegimos como variable respuesta la variable V

```
modelo = glm(V~., data = dataSet, family = "binomial")
summary(modelo)
```

Intervalo de confianza para las estimaciones de las variables, al 95% de nivel de significación:

```
confint(modelo)
```

```
exp(coef(modelo))
```

3.3 Predicciones

```
predicciones = predict.glm(modelo,newdata = dataSet)
head(predicciones)
```

```
probabilidad = exp(predicciones)/(1+exp(predicciones))
head(probabilidad)
```

Definimos las funciones:

```
abinario=function(x)
{if (x>0.5) {return(1)}
  else{return(0)}
}
elfactor=function(x)
{if (x==0) {return(("Z1"))}
  if (x==1){return(("Z2"))}}
```

```
estimacionbi=lapply(probabilidad,abinario)
estimacion= lapply(estimacionbi, elfactor)
```

Vamos a ver la tabla con las predicciones, donde podremos ver si nos hemos equivocado o no al predecir.

```
Prediccion=t(as.data.frame(estimacion))
Tabla_Completa=cbind.data.frame(dataSet,Prediccion)
Tabla_Completa
```

```
length(Tabla_Completa$V)
length(Tabla_Completa$Prediccion)

#si=length(which(Tabla_Completa$V==Tabla_Completa$Prediccion))
#si/nrow(Tabla_Completa)*100
```

3.4 Estimaciones JACKKNIFE

```
n=nrow(dataSet)
prediccionJackknife = numeric(n)
probJack=numeric(n)
for(i in 1:n){
  modelo_i = glm(V~.,data=dataSet[-i,],family = "binomial")
  prJack<-predict.glm(modelo_i,newdata=dataSet[i,])
  probJack[i] = exp(prJack)/(1+exp(prJack))
  binJack= abinario(probJack[i])
  prediccionJackknife[i]=elfactor(binJack)
}

length(which(prediccionJackknife==dataSet$V))*100/n

length(which(prediccionJackknife==Tabla_Completa$Prediccion))*100/n
```

3.5 Validación cruzada

```
library(caret)
modelocaret=train(V~.,data=dataSet, method="glm",
                  trControl = trainControl(method="CV",number=10))
modelocaret$results
```

```
modelocaret$results["Accuracy"]*100
```

4 Pregunta 4

Bootstrap. Implementar una función que calcule el estadístico de Fisher de comparación de coeficientes de correlación lineal:

$$T = \frac{Z_1 - Z_2}{\sqrt{\frac{1}{n_1 - 3} + \frac{1}{n_2 - 3}}}$$

Donde:

- Z1 y Z2 representan la transformación de Fisher de los respectivos coeficientes de correlación lineal para dos grupos
- n1 y n2 son las frecuencias absolutas respectivas de los grupos.

Sobre un conjunto de datos apropiado (que puede ser generado), utilizando el anterior estadístico, realizar e interpretar un test bootstrap bilateral de comparación de los coeficientes de correlación lineal (B=1999).