Tarea 3

Estudiante

John Daniel hoyos Arias
Ivan Santiago Rojas Martinez
Genaro Alfonso Aristizabal Echeverri

Docente

Juan Carlos Salazar Uribe

Asignatura

Analitica de datos

Sede Medellín
15 de Noviembre del 2022

Índice

Índice de figuras

1. Ejercicio1

```
attach(Boston)
```

```
fit1 <- lm(nox ~ poly(dis,4), data = Boston)
coef(summary(fit1))
```

```
##                   Estimate  Std. Error     t value      Pr(>|t|)
## (Intercept)    0.55469506 0.002761339 200.8790240  0.000000e+00
## poly(dis, 4)1 -2.00309590 0.062114782 -32.2482963 2.540459e-124
## poly(dis, 4)2  0.85632995 0.062114782  13.7862506  6.924872e-37
## poly(dis, 4)3 -0.31804899 0.062114782  -5.1203430  4.356581e-07
## poly(dis, 4)4  0.03354668 0.062114782   0.5400757  5.893848e-01
```

Al ajustar la regresión correspondiente, se pudo determinar que es suficiente considerar un polinomio de grado 3. Mientras que con un polinomio de grado 4, se comienzan a presentar problemas de sobreparametrización y, por lo tanto, algunos de los parámetros dejan de ser significativos para el modelo.

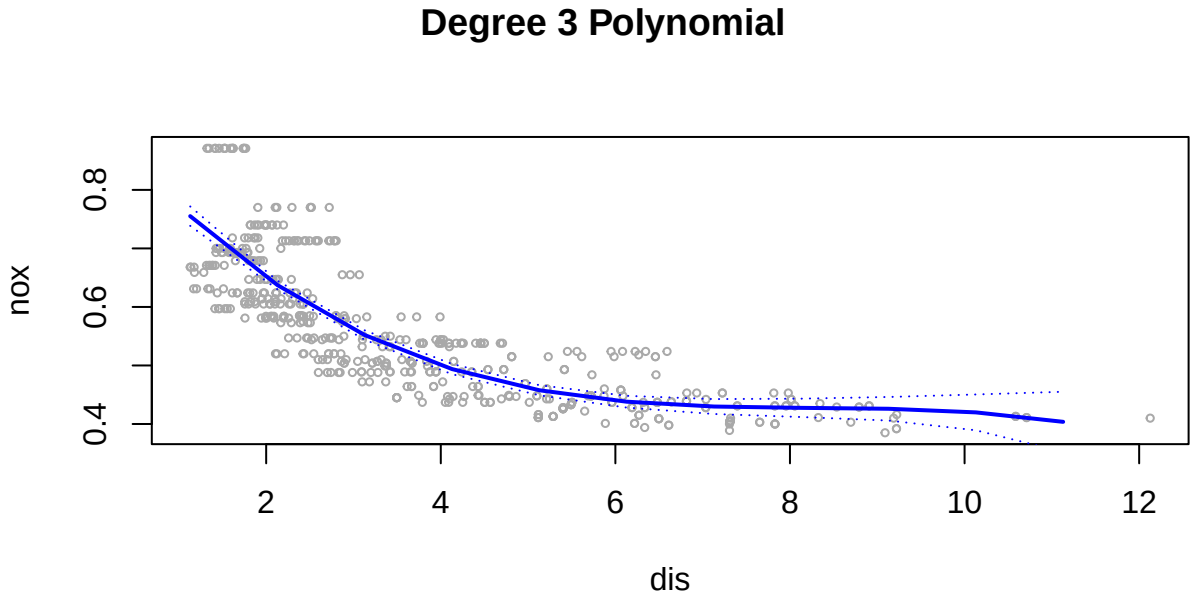
```
fit <- lm(nox ~ poly(dis,3), data = Boston)
(summary(fit))
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Todas las variables en este modelo de regresión cúbica parecen tener valores p que son estadísticamente significativos. El error estándar residual es bajo pero los grados de libertad son altos. Los valores de R-Squared son relativamente altos en 0.71 para explicar la varianza en el modelo.

```
dislims <- range(dis)
dis.grid <- seq(from = dislims[1], to = dislims [2])
preds <- predict(fit , newdata = list(dis = dis.grid),se = TRUE)
se.bands <- cbind(preds$fit + 2 * preds$se.fit ,preds$fit - 2 * preds$se.fit)
```

```
par(mfrow = c(1, 1), mar = c(4.5 , 4.5, 1, 1),
oma = c(0, 0, 4, 0))
plot(dis , nox , xlim = dislims , cex = .5, col = "darkgrey")
title("Degree 3 Polynomial", outer = T)
lines(dis.grid, preds$fit , lwd = 2, col = "blue")
matlines(dis.grid , se.bands, lwd = 1, col = "blue", lty = 3)
```



Graficamente, observamos un ajuste bastante bueno al considerar un polinomio de grado 3.

b) Grafique los ajustes polinómicos para un rango de polinomios de diferentes grados (digamos, de 1 a 10), y reporte la suma de cuadrados de los residuales asociada.

Se ajustarán modelos que van desde uno lineal hasta un polinomio de grado 5, para determinar el modelo más simple que sea suficiente para explicar la relación entre nox y dis.

```
fit.1 <- lm(nox ~ dis , data = Boston)
fit.2 <- lm(nox ~ poly(dis , 2), data = Boston)
fit.3 <- lm(nox ~ poly(dis , 3), data = Boston)
fit.4 <- lm(nox ~ poly(dis , 4), data = Boston)
fit.5 <- lm(nox ~ poly(dis , 5), data = Boston)
anova(fit.1, fit.2, fit.3, fit.4, fit.5)
```

```
## Analysis of Variance Table
##
## Model 1: nox ~ dis
## Model 2: nox ~ poly(dis, 2)
## Model 3: nox ~ poly(dis, 3)
## Model 4: nox ~ poly(dis, 4)
## Model 5: nox ~ poly(dis, 5)
##   Res.Df    RSS Df Sum of Sq        F    Pr(>F)
## 1    504 2.7686
## 2    503 2.0353  1   0.73330 191.4334 < 2.2e-16 ***
## 3    502 1.9341  1   0.10116  26.4073 3.972e-07 ***
## 4    501 1.9330  1   0.00113   0.2938   0.58804
## 5    500 1.9153  1   0.01769   4.6185   0.03211 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coef(summary(fit.5))
```

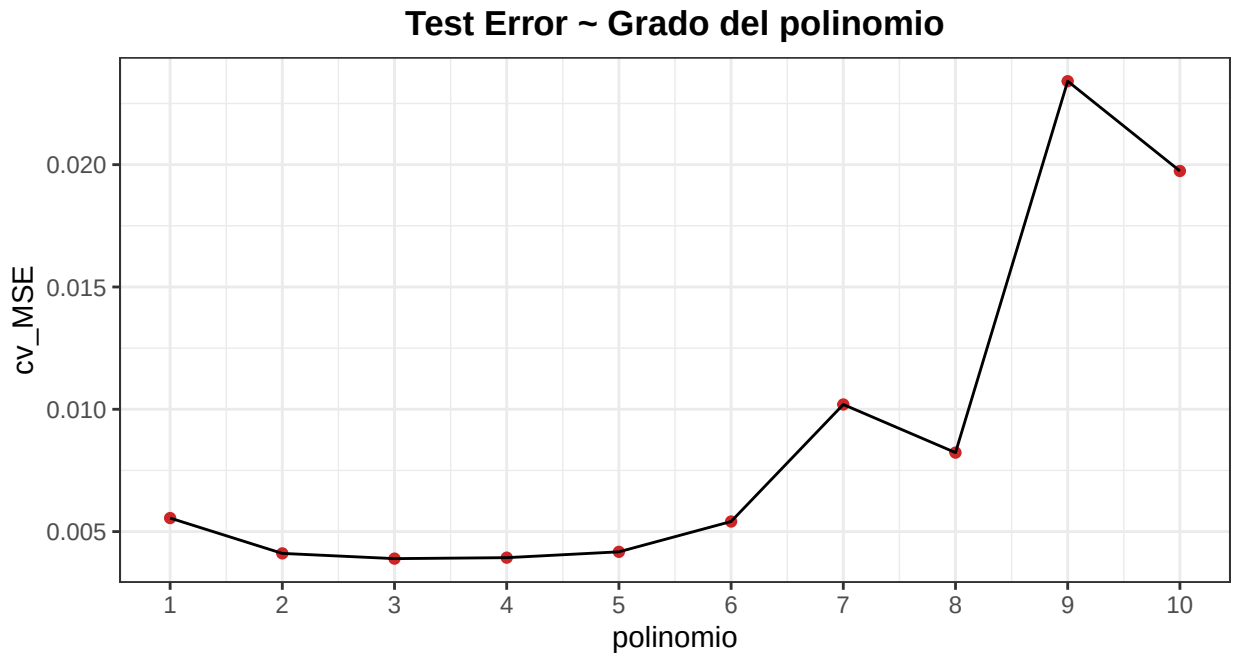
```
##                     Estimate  Std. Error     t value      Pr(>|t|)
## (Intercept)    0.55469506 0.002751421 201.6031400  0.000000e+00
## poly(dis, 5)1 -2.00309590 0.061891679 -32.3645429 9.417911e-125
## poly(dis, 5)2  0.85632995 0.061891679  13.8359464  4.303120e-37
## poly(dis, 5)3 -0.31804899 0.061891679  -5.1388005  3.971902e-07
## poly(dis, 5)4  0.03354668 0.061891679   0.5420225  5.880445e-01
## poly(dis, 5)5  0.13300890 0.061891679   2.1490594  3.210905e-02
```

Parece ser más apropiado utilizar el modelo cúbico donde El p-valor que compara el polinomio de grado 4, Model3 y Model4, es aproximadamente 59 % mientras que el polinomio de grado 5 Model5 parece innecesario por qué su p-valor es 0.03. Por lo tanto, un modelo cuadrático parece proporcionar un ajuste razonable a los datos en comparación a modelos de grados inferiores y superiores.

c) Realice una validación cruzada o alg´un otro enfoque para seleccionar el óptimo grado para el polinomio y explique sus resultados.

K-fold Cross-Validation

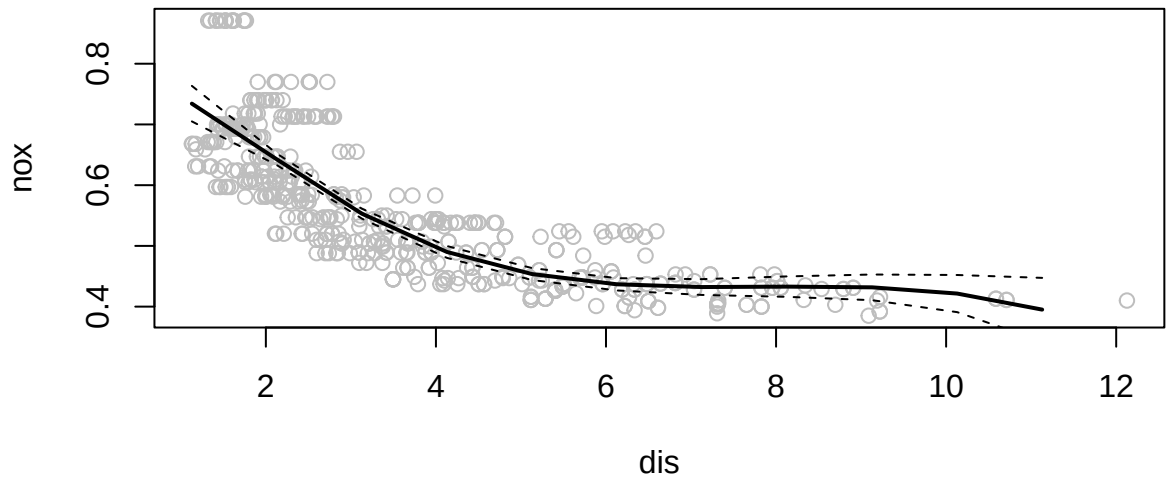
```r
#library(boot)
#library(ggplot2)
cv_MSE_k10 <- rep(NA,10)

for (i in 1:10) {
  modelo <- glm(nox ~ poly(dis, i), data = Boston)
  set.seed(17)
  cv_MSE_k10[i] <- cv.glm(data = Boston, glmfit = modelo, K = 10)$delta[1]
}
ggplot(data = data.frame(polinomio = 1:10, cv_MSE = cv_MSE_k10),
       aes(x = polinomio, y = cv_MSE)) +
geom_point(colour = c("firebrick3")) +
geom_path() +
scale_x_continuous(breaks = c(0:10)) +
theme_bw() +
labs(title  =  'Test Error ~ Grado del polinomio') +
theme(plot.title = element_text(hjust = 0.5, face = 'bold'))
```



Tras el proceso de validación cruzada, se determina que el mejor grado del polinomio es efectivamente el de grado 3, dado que es donde se presenta un menor MSE.

d) Use la función bs() para ajustar una spline de regresión para predecir nox usando dis. Reporte la salida para el ajuste usando cuatro grados de libertad. ¿Cómo ubicó los nodos?. Grafique el ajuste resultante.

```r
#library(splines)
fit <- lm(nox ~ bs(dis , knots = c(3.2)), data = Boston)
pred <- predict(fit , newdata = list(dis = dis.grid), se = T)
plot(dis , nox , col = "gray")
lines(dis.grid, pred$fit , lwd = 2)
lines(dis.grid , pred$fit + 2 * pred$se, lty = "dashed")
lines(dis.grid , pred$fit - 2 * pred$se, lty = "dashed")
```




```r
#library(ggplot2)

attr(bs(dis , df = 4), "knots")
```

```
##     50%
## 3.20745
```

Para ajustar una spline de regresión con 4 grados de libertad, se utilizó la función attr la cual nos permite determinar la posición de los nodos según los grados de libertad, en este caso, como acabamos de observar, para 4 grados de libertad, solo permite un nodo que representa la mediana en 3.2.

e) Ahora ajuste una spline de regresión para un rango de grados de libertad, y grafique los ajustes resultantes e informe el RSS resultante. Describa los resultados obtenidos.

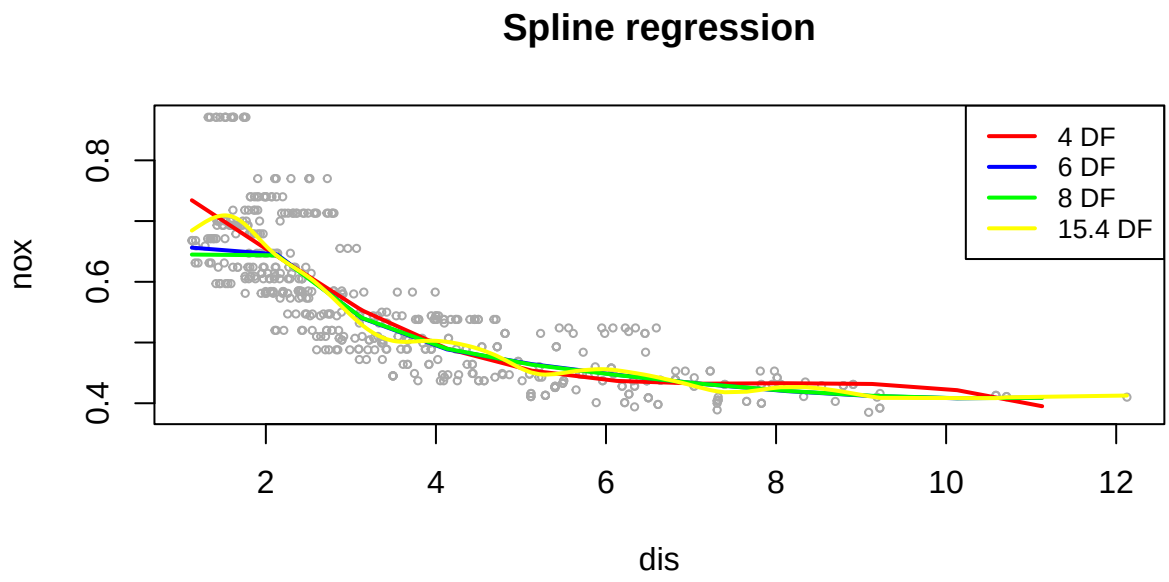
```
plot.new()
plot(dis , nox , xlim = dislims , cex = .5, col = "darkgrey")
title("Spline regression")
# 4grados de libertad
fit.1 <- lm(nox ~ bs(dis , knots = c(3.2)), data = Boston)
# 6grados de libertad
attr(bs(dis , df = 6), "knots")
```

```
##      25%      50%      75%
## 2.100175 3.207450 5.188425
```

```
fit.2 <- lm(nox ~ bs(dis , knots = c(2.1,3.2,5.18)), data = Boston)
# 7grados de libertad
attr(bs(dis , df = 7), "knots")
```

```
##    20%    40%    60%    80%
## 1.9512 2.6403 3.8750 5.6150
```

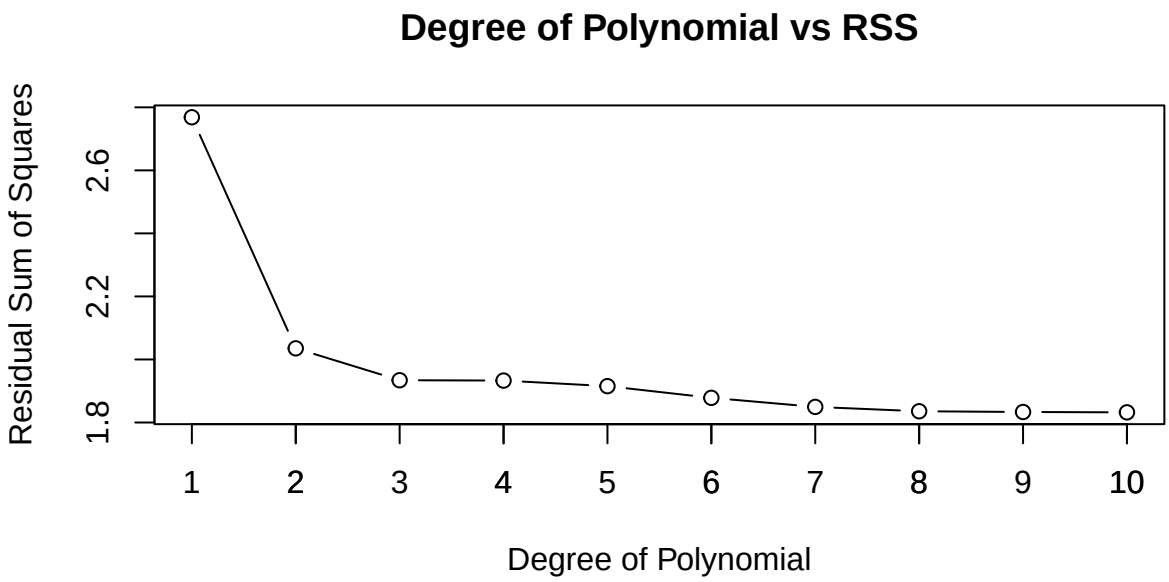
```
fit.3 <- lm(nox ~ bs(dis , knots = c(1.95,2.6,3.87,5.61)), data = Boston,cv=T)
#spline suave, con grados de libertad obtenidos a partir de cv
fit.4 <- smooth.spline(dis , nox , cv = TRUE)

pred1 <- predict(fit.1 , newdata = list(dis = dis.grid), se = T)
pred2 <- predict(fit.2, newdata = list(dis = dis.grid), se = T)
pred3<- predict(fit.3, newdata = list(dis = dis.grid), se = T)

lines(dis.grid, pred1$fit ,col='red', lwd = 2)
lines(dis.grid, pred2$fit ,col='blue', lwd = 2)
lines(dis.grid, pred3$fit ,col='green', lwd = 2)
lines(fit.4 , col = "yellow", lwd = 2)

legend("topright", legend = c("4 DF", "6 DF",'8 DF','15.4 DF'),col = c("red", "blue",'gr
```



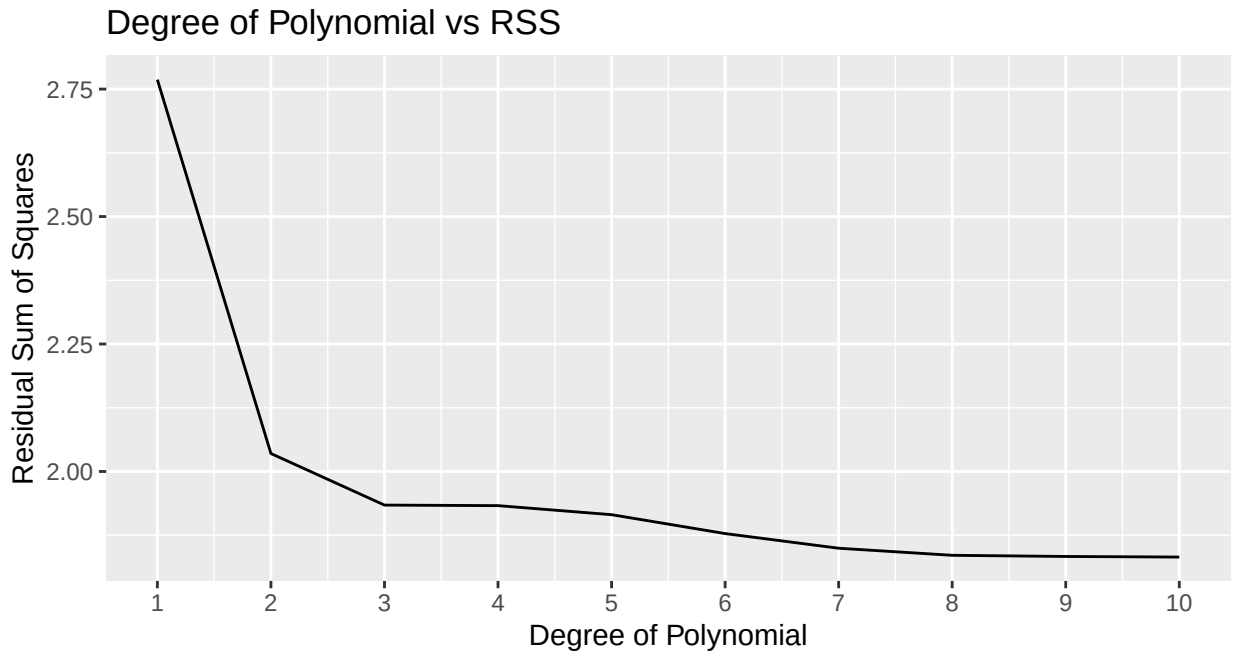
Se observan splines de regresión con 4, 6, 7 y un spline suave con 15.4 gl, claramente el mejor ajuste se observa para un spline cúbico con 4 grados de libertad, a medida que se aumentan los grados de libertad, la tendencia del ajuste se deja llevar por el ruido y esto genera problemas de predicción.

Analisis de RSS

```
rss <- rep(NA, 10)
for (i in 1:10) {
  poly.fit <- lm(nox ~ poly(dis, i), data=Boston)
  rss[i] <- sum(poly.fit$residuals^2)
}

plot(1:10, rss, xlab='Degree of Polynomial', ylab='Residual Sum of Squares', type='b', m
axis(1, at = seq(1,10, by=1))
```




```
#library(data.table)
rss <- data.table(seq(1:10), rss, keep.rownames = TRUE)
ggplot(rss, aes(V1, rss)) +
  geom_line() +
  scale_x_continuous(breaks=c(1:10)) +
  labs(x='Degree of Polynomial', y='Residual Sum of Squares', title='Degree of Polynomia
```



```
rss[c(2,3),]
```

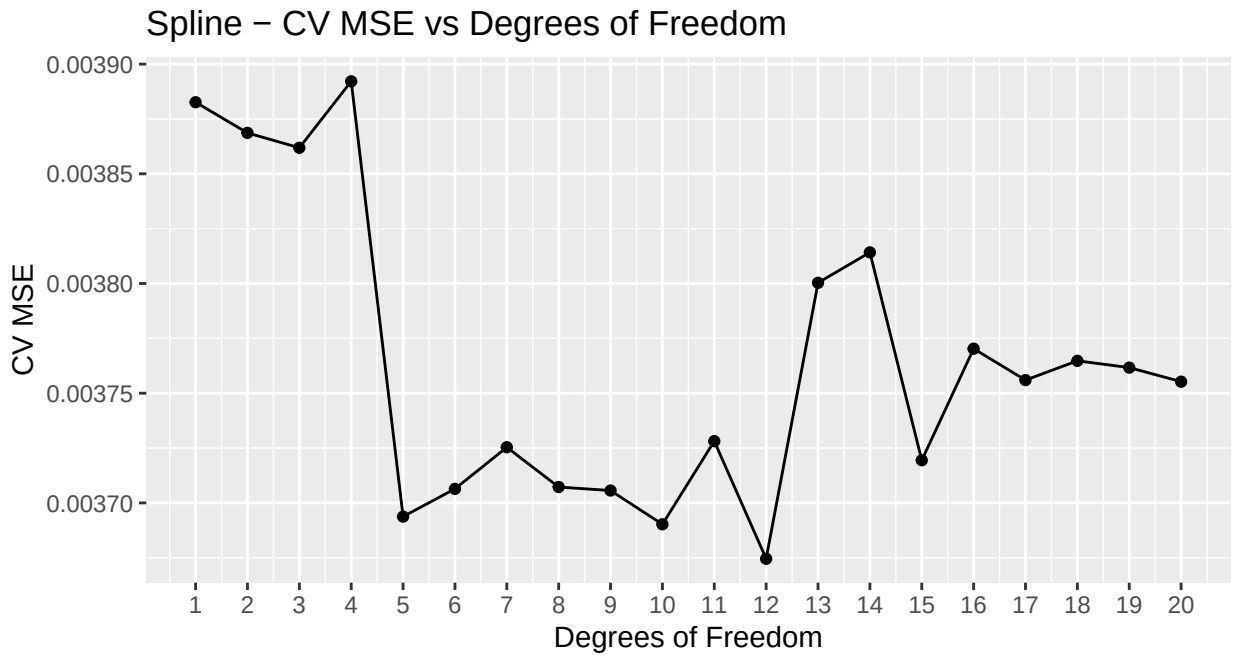
```
##    V1      rss
## 1:  2 2.035262
## 2:  3 1.934107
```

el menor Rss se obtiene con 10 grados de libertad.

f) Realice una validación cruzada o algún otro enfoque para seleccionar los mejores grados de libertad para una spline de regresión sobre estos datos. Describa sus resultados.

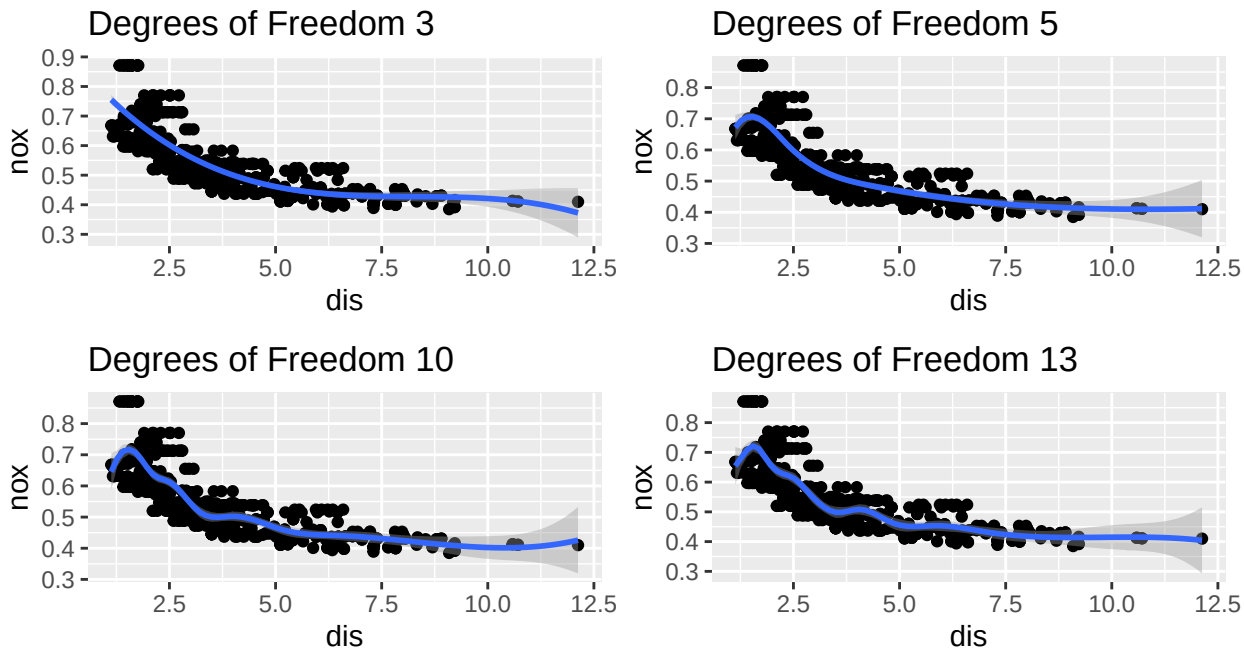
```
#library(boot)
#library(data.table)
#library(ggplot2)
cv.spline.fun <- function(i) {
  fit <- glm(nox ~ bs(dis, df=i), data=Boston)
  cv.error <- cv.glm(Boston, fit, K=10)$delta[2]
}

cv.err <- sapply(1:20, cv.spline.fun)

df <- seq(1:20)

dt <- data.table(df, cv.err)

plot.new()
```

```
ggplot(dt, aes(df, cv.err)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks=seq(1:20)) +
  labs(x='Degrees of Freedom', y='CV MSE', title='Spline - CV MSE vs Degrees of Freedom'
```




```
#library(gridExtra)

p1 <- ggplot(Boston, aes(x=dis, y=nox)) +
    geom_point(colour='black') +
    stat_smooth(method='lm', formula= y ~ bs(x,df=3)) +
  labs(x='dis',y='nox', title='Degrees of Freedom 3')

p2 <- ggplot(Boston, aes(x=dis, y=nox)) +
    geom_point(colour='black') +
    stat_smooth(method='lm', formula= y ~ bs(x,df=5)) +
  labs(x='dis',y='nox', title='Degrees of Freedom 5')


p3 <- ggplot(Boston, aes(x=dis, y=nox)) +
    geom_point(colour='black') +
    stat_smooth(method='lm', formula= y ~ bs(x,df=10)) +
  labs(x='dis',y='nox', title='Degrees of Freedom 10')
```

```
p4 <- ggplot(Boston, aes(x=dis, y=nox)) +
    geom_point(colour='black') +
    stat_smooth(method='lm', formula= y ~ bs(x,df=13)) +
  labs(x='dis',y='nox', title='Degrees of Freedom 13')


grid.arrange(p1,p2,p3,p4, ncol=2, nrow=2)
```



Se observa que el ajuste puede darse entre 5 y 10 grados de libertad, aunque según el RSS el que mejor ajuste tiene es el de 10 grados de libertad.

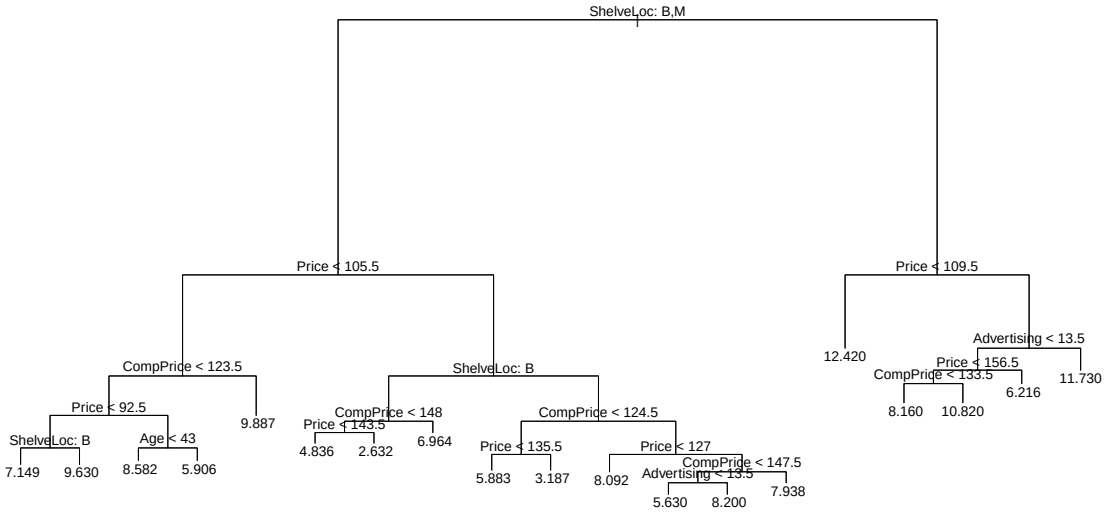
2. Ejercicio2

2.1. a)

2.2. b)

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Age"         "Advertising"
## Number of terminal nodes:  19
```

```
## Residual mean deviance:  2.359 = 662.9 / 281
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.1290 -0.9993  0.0563  0.0000  0.9134  5.3040
```




```
## NULL
```