NIST_comparacionR_resultados

Valeria Perez

19/11/2021

Problema

El problema a resolver es ajustar un polinomio de grado 10 a los datos llamados filip proporcionados por NIST. De forma matricial, el problema es

$$y = X\beta$$

donde la matrix X es de 82 x 11, la matriz β es de 11 x 1. El problema es un problema de ecuaciones lineales que intenté resolver con 4 métodos diferentes en Julia. Como no encontraba el error a mis algoritmos, decidí ponerlos a prueba. \ En general, son dos pruebas diferentes. La primera es que comparo los resultados de los 4 algoritmos contra el resultado de la función lm en R. La segunda es que hago las estimaciones de los coeficientes β para todos los polinomios, desde el polinomio de grado 1 hasta el polinomio de grado 10. \ Antes que nada, el código para obtener los resultados. En primer lugar, leo y asigno a las variables nuevas los resultados que obtuve de Julia.

```
library(polynom)
```

```
## Warning: package 'polynom' was built under R version 4.0.5
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.5
```

```
#setwd("~/ITAM/Tesis/Julia con R/Code/NIST")
data <- read.csv("filip_data.csv")

#setwd("~/ITAM/Tesis/Julia con R/Code/NIST/Resultados")

temp <- list.files(pattern = "resultados_grado_")
myfiles <- lapply(temp, read.csv)

myfiles[[11]] <- myfiles[[2]]
myfiles <- myfiles[-2]

for (i in 1:10){
  assign(paste("resultados_grado_", i), myfiles[[i]])
}
```

Después, hago el código para el polinomio de grado 1.

```
# Para polinomio de grado = 1
lm_1 <- lm(y ~ x, data = data, x = TRUE)
`resultados_grado_ 1`$R <- lm_1$coefficients
row.names(`resultados_grado_ 1`) <- c("b0", "b1")
X_1 <- lm_1$x
```

Finalmente, el código para los polinomios de grado > 1

```r
# Para polinomios de grado > 1

for (i in 2:10){
  #Hacemos el modelo
  model <- paste("y ~ x", paste("+ I(x^", 2:i, ")", sep='', collapse=''))

  # Lo convertimos en formula
  form <- formula(model)

  #Ejecutamos el modelo
  lm.plus <- lm(form, data = data, x = TRUE)

  # Guardo el df correspondiente a un auxiliar
  resultados_aux <- get(paste("resultados_grado_", i))
  # para unirle los coeficientes
  resultados_aux$R <- lm.plus$coefficients

  nombres <- c("b0")
  # Para el nombre de los renglones
  for (k in 1:i){
    nombres <- c(nombres, paste0("b", k))
  }
  row.names(resultados_aux) <- nombres

  #Finalmente, hago el df final
  assign(paste("resultados_grado_", i), resultados_aux)

  assign(paste("X_", i), lm.plus$x)

}
```

Los resultados estan en las siguientes tablas.

```r
kable(`resultados_grado_ 1`, caption = "Polinomio grado 1")
```

Table 1: Polinomio grado 1

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	1.0592655	1.0592655	1.0592655	1.0592655	1.0592655
b1	0.0340946	0.0340946	0.0340946	0.0340946	0.0340946

```r
kable(`resultados_grado_ 2`, caption = "Polinomio grado 2")
```

Table 2: Polinomio grado 2

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	0.9223409	0.9223409	0.9223409	0.9223409	0.9223409
b1	-0.0140724	-0.0140724	-0.0140724	-0.0140724	-0.0140724
b2	-0.0039770	-0.0039770	-0.0039770	-0.0039770	-0.0039770

```
kable(`resultados_grado_ 3`, caption = "Polinomio grado 3")
```

Table 3: Polinomio grado 3

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	0.3902712	0.3902712	0.3902712	0.3902712	0.3902712
b1	-0.3033645	-0.3033645	-0.3033645	-0.3033645	-0.3033645
b2	-0.0537195	-0.0537195	-0.0537195	-0.0537195	-0.0537195
b3	-0.0027265	-0.0027265	-0.0027265	-0.0027265	-0.0027265

```
kable(`resultados_grado_ 4`, caption = "Polinomio grado 4")
```

Table 4: Polinomio grado 4

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	2.6444057	2.6444057	2.6444057	2.6444057	2.6444057
b1	1.3744058	1.3744058	1.3744058	1.3744058	1.3744058
b2	0.3970969	0.3970969	0.3970969	0.3970969	0.3970969
b3	0.0492439	0.0492439	0.0492439	0.0492439	0.0492439
b4	0.0021749	0.0021749	0.0021749	0.0021749	0.0021749

```
kable(`resultados_grado_ 5`, caption = "Polinomio grado 5")
```

Table 5: Polinomio grado 5

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	4.3006539	4.3006539	4.3006539	4.3006544	4.3006539
b1	2.9237727	2.9237727	2.9237727	2.9237731	2.9237727
b2	0.9589165	0.9589165	0.9589165	0.9589167	0.9589165
b3	0.1481183	0.1481183	0.1481183	0.1481184	0.1481183
b4	0.0106384	0.0106384	0.0106384	0.0106384	0.0106384
b5	0.0002825	0.0002825	0.0002825	0.0002825	0.0002825

```
kable(`resultados_grado_ 6`, caption = "Polinomio grado 6")
```

Table 6: Polinomio grado 6

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	-18.0975496	-18.0975496	-18.0975496	1.9043149	-18.0975496
b1	-22.2966441	-22.2966441	-22.2966441	0.0000000	-22.2966441
b2	-10.5769427	-10.5769427	-10.5769427	-0.4810935	-10.5769427
b3	-2.5981095	-2.5981095	-2.5981095	-0.2185587	-2.5981095
b4	-0.3486584	-0.3486584	-0.3486584	-0.0403353	-0.3486584
b5	-0.0242444	-0.0242444	-0.0242444	-0.0033915	-0.0242444
b6	-0.0006834	-0.0006834	-0.0006834	-0.0001075	-0.0006834

```
kable(`resultados_grado_ 7`, caption = "Polinomio grado 7")
```

Table 7: Polinomio grado 7

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	-8.6609575	-8.6609575	-8.6609588	0.0000000	-8.6609575
b1	-9.8263025	-9.8263025	-9.8263040	0.0000000	-9.8263025
b2	-3.6650346	-3.6650346	-3.6650351	0.0000000	-3.6650346
b3	-0.5141292	-0.5141292	-0.5141295	-0.1742561	-0.5141292
b4	0.0207340	0.0207340	0.0207339	-0.0871662	0.0207340
b5	0.0142807	0.0142807	0.0142807	-0.0170740	0.0142807
b6	0.0015076	0.0015076	0.0015076	-0.0015216	0.0015076
b7	0.0000525	0.0000525	0.0000525	-0.0000515	0.0000525

```
kable(`resultados_grado_ 8`, caption = "Polinomio grado 8")
```

Table 8: Polinomio grado 8

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	175.9750151	175.9750152	175.9750570	0.0000000	175.9750151
b1	269.2657673	269.2657674	269.2657897	0.0000000	269.2657673
b2	177.4702511	177.4702512	177.4702530	0.0000000	177.4702511
b3	65.4365443	65.4365443	65.4365573	0.0000000	65.4365443
b4	14.7617342	14.7617342	14.7617357	0.0554432	14.7617342
b5	2.0867401	2.0867401	2.0867399	0.0285355	2.0867401
b6	0.1806048	0.1806048	0.1806048	0.0056228	0.1806048
b7	0.0087567	0.0087567	0.0087567	0.0004978	0.0087567
b8	0.0001823	0.0001823	0.0001823	0.0000166	0.0001823

```
kable(`resultados_grado_ 9`, caption = "Polinomio grado 9")
```

Table 9: Polinomio grado 9

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	-174.2804420	-174.2804443	-174.2802936	0.0000000	-174.2804456
b1	-326.8822038	-326.8822077	-326.8825148	0.0000000	-326.8822099
b2	-266.0565360	-266.0565388	-266.0571501	0.0000000	-266.0565405
b3	-123.9216126	-123.9216137	-123.9219121	0.0000000	-123.9216144
b4	-36.3816705	-36.3816708	-36.3816855	0.0000000	-36.3816710
b5	-6.9791883	-6.9791883	-6.9791852	-0.0151684	-6.9791884
b6	-0.8746602	-0.8746602	-0.8746601	-0.0078823	-0.8746602
b7	-0.0690601	-0.0690601	-0.0690601	-0.0015521	-0.0690601
b8	-0.0031183	-0.0031183	-0.0031183	-0.0001365	-0.0031183
b9	-0.0000614	-0.0000614	-0.0000614	-0.0000045	-0.0000614

```
kable(`resultados_grado_ 10`, caption = "Polinomio grado 10")
```

Table 10: Polinomio grado 10

	PolFit	QRPivot	MoorePenrose	LinearFit	R
b0	-1467.4896771	9.0134262	9.1661966	0.0000000	-174.2804456
b1	-2772.1797121	1.6525458	2.7223863	0.0000000	-326.8822099
b2	-2316.3711835	-5.7676064	-4.2207903	0.0000000	-266.0565405
b3	-1127.9739915	-3.8636656	-2.7920321	0.0000000	-123.9216144
b4	-354.4782499	-0.6703657	-0.2334151	0.0000000	-36.3816710
b5	-75.1242053	0.1806044	0.2944490	0.0000000	-6.9791884
b6	-10.8753186	0.1055234	0.1251241	0.0036864	-0.8746602
b7	-1.0622150	0.0214449	0.0236685	0.0019173	-0.0690601
b8	-0.0670191	0.0022775	0.0024377	0.0003759	-0.0031183
b9	-0.0024678	0.0001262	0.0001329	0.0000328	-0.0000614
b10	-0.0000403	0.0000029	0.0000030	0.0000011	NA