Trabajo N°1 Punto 2: Ajuste de un modelo de R.L.M

Universidad Nacional de Colombia
Analisís de Regresión 2022-1S
Medellín, Colombia
2022

Daniel Villa 1005087556 Juan Pablo Vanegas 1000640165

Contents

1 Introducción

Consideremos el modelo de regresió $Y_i = f(x_i; \beta) + \varepsilon_i$ $(i = 1, ..., n)$ donde f es una expectativa conocida como función (llamada curva de calibración) que es monótona en el rango de interés y $\varepsilon_i \sim N(0, \sigma^2)$.

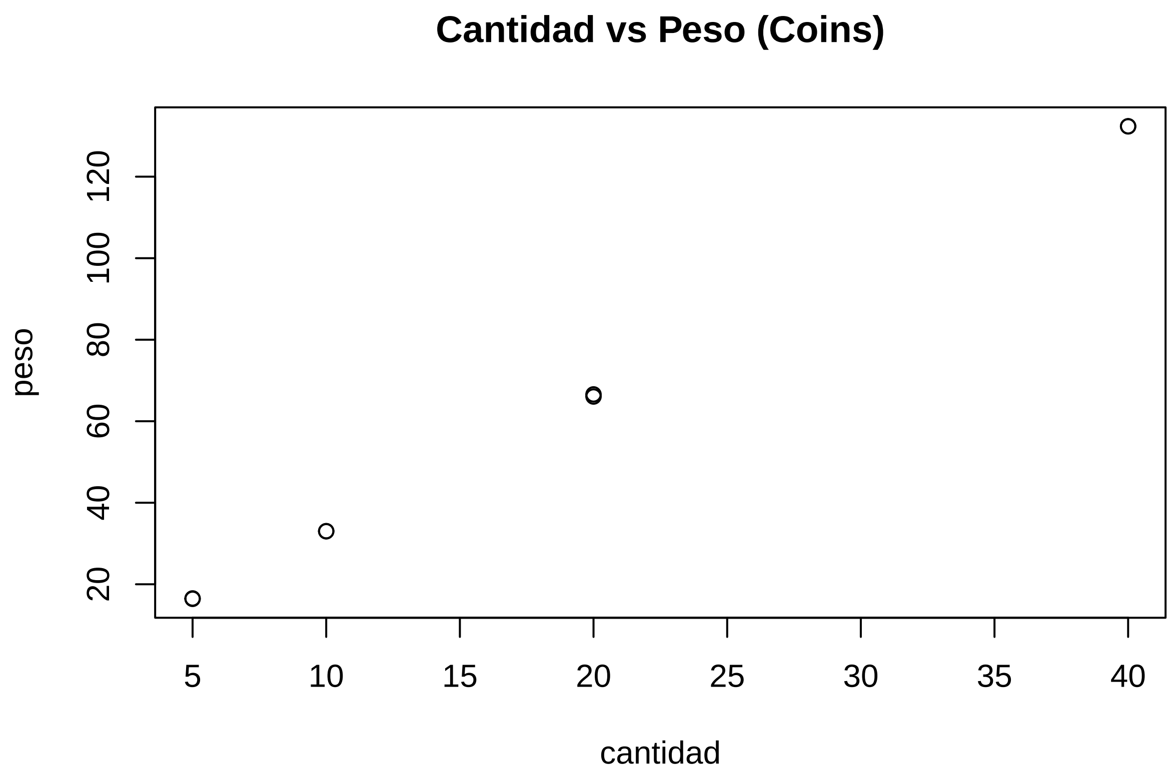
Un problema común en regresiones predecir una respuesta futura y_0 a partir de un valor conocido de la variable explicativa x_0. sin embargo, a menudo es necesario hacer lo contrario; es decir, dado un valor observa de la respuesta $Y = y_0$ estimar el valor conocido de la variable explicativa x_0. esto se conoce como el problema de calibración, aunque nos referimos a él de forma más general como estimación inversa.

En este documento trataremos de ajustar un modelo de regresión lineal simple por medio del problema de calibración.

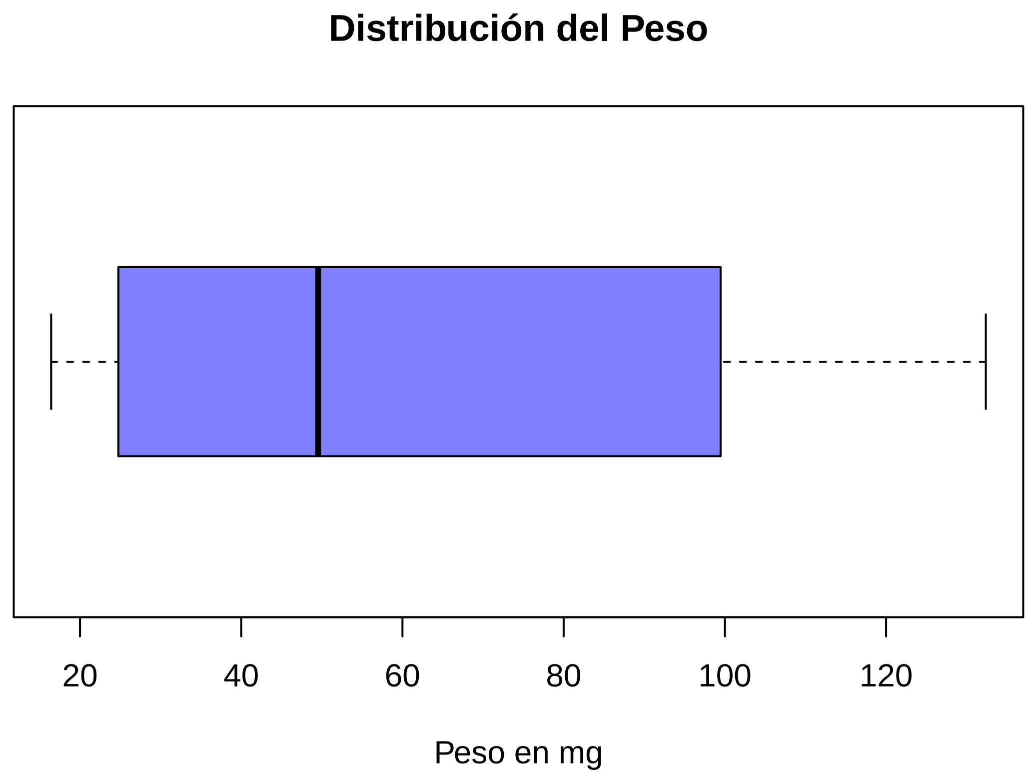
Nota: las modelas a utlizar son denominación

2 Exploración de los datos:

cantidad	peso
5	16.53
5	16.42
10	33.04
10	32.99
20	66.59
20	66.06
40	132.37
40	132.34



Vamos a ver como estos pocos datos se comportan.



Como podemos ver nuestros datos estan centralizados en $50mg$ sin datos atipicos a a la muestra y con un rango muy amplio.

3 Modelo

Crearemos un modelo para la explicación de la cantidad de monedas x_0 através del peso y_0

Eliminamos el intercepto.

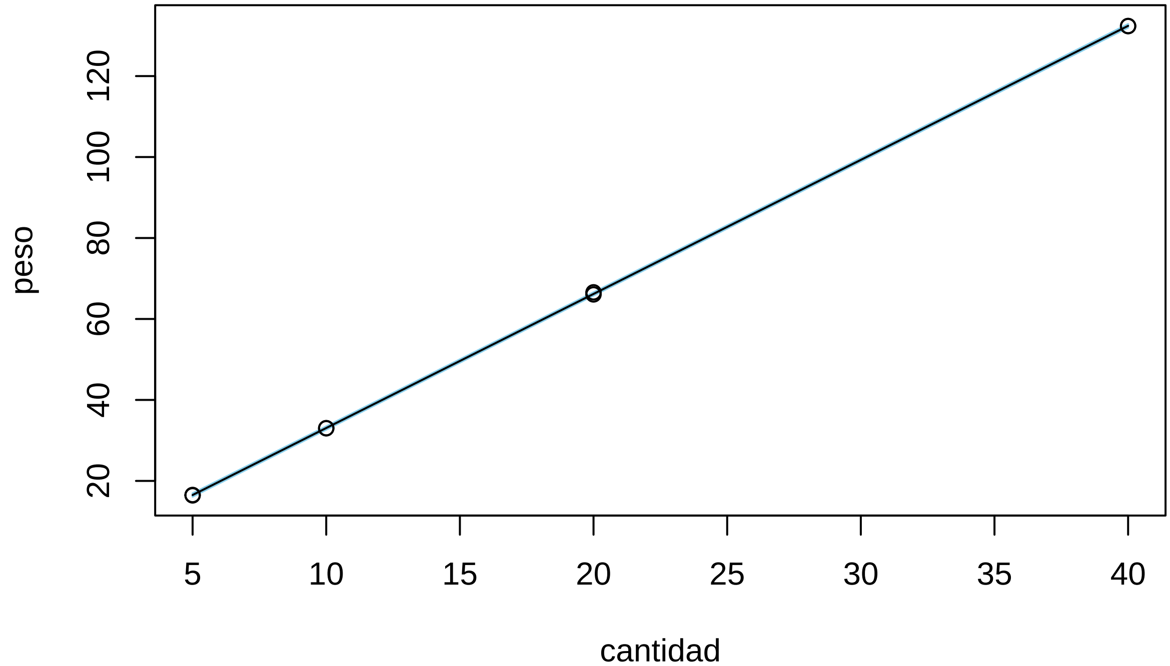
```
##
## Call:
## lm(formula = peso ~ cantidad, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12170 -0.08143 -0.05383 -0.01643  0.40830
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.045435   0.111917  -0.406    0.699
## cantidad     3.311357   0.004856 681.959 6.71e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1841 on 6 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.651e+05 on 1 and 6 DF,  p-value: 6.71e-16
```

ahora sin porbar supuestos ya que este solo es el inicio del modelo como tal, utilizamos la libreria `investr` para crear el modelo inverso:

COn esto calibramos el modelo de forma inversa para que hay una respuesta desde la variable Y, esto se hace con la función `calibrate`, con un intervalo de "inversión" por el cual se puede decir que se le saco la inversa a la función, pero en este caso diremos de forma estdística o probabilistica.

```
res <- calibrate(model.1,y0=0,interval = "inversion", level=0.9)

plotFit(model.1, interval = "prediction",
        level = 0.9, shade = TRUE, col.pred = "skyblue")
```




```
res
```

```
##   estimate      lower     upper
##  0.0137209 -0.1128453  0.1399828
```

de los resultados podemos sacar el valor estimado del parametreo que acompañara a y_i, en este caso se encuentra sin intercepto ya que desde el inicio se elimino el valor por cuestiones del $Pr(>|t|)$

Vemos ahora la regresión sobre los datos explicando a *cantidad* y no a *peso* como pasaba en `model.1`

por cuestiones de mejoras y para presentar el SE de los datos, pasaremos a cambiar el intervalo de inversion default al de **Wald**

```
res <- calibrate(model.1,y0=0,interval = "Wald", level=0.9)
res
```

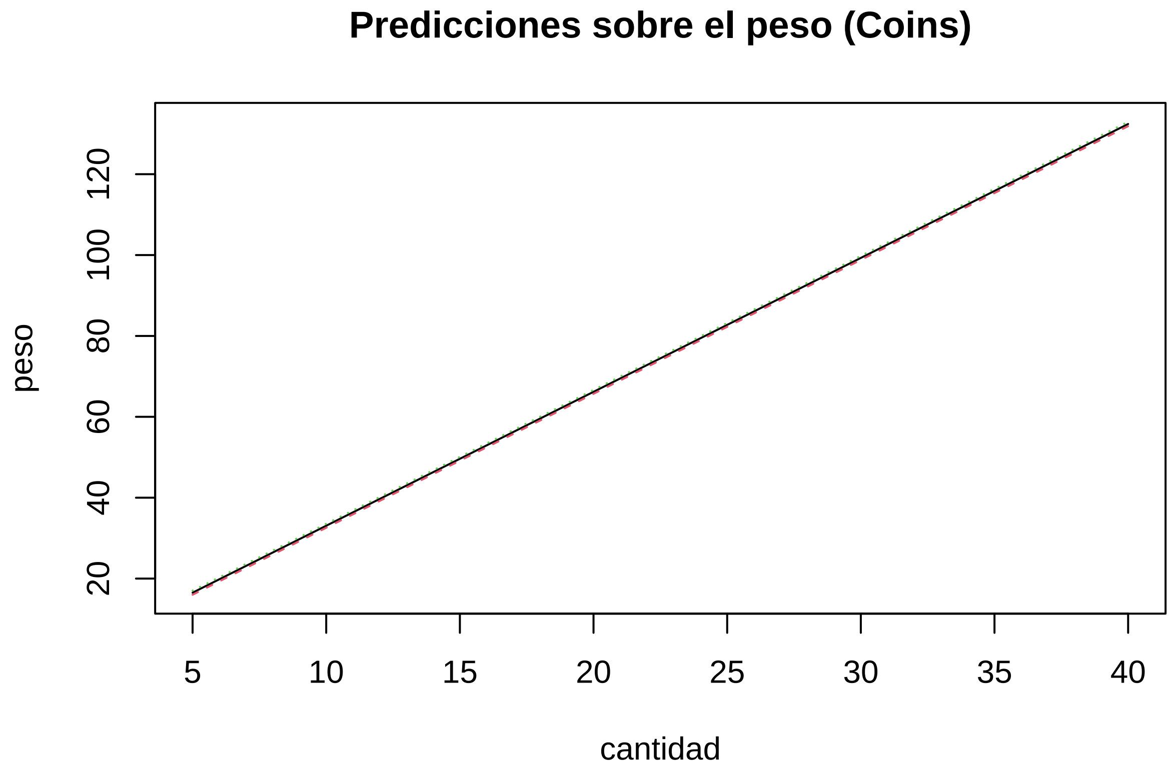
```
##   estimate      lower     upper         se
##  0.0137209 -0.1126926  0.1401344  0.0650549
```

aqui presenta el error cuadratico.

Predicción de nuevas observaciones

```
##        fit      lwr      upr
## 1 16.51135 16.00639 17.01631
## 2 24.78974 24.29359 25.28588
## 3 33.06813 32.57915 33.55711
## 4 41.34652 40.86299 41.83006
## 5 49.62491 49.14504 50.10479
```

```
## 6 57.90330 57.42527 58.38134
```

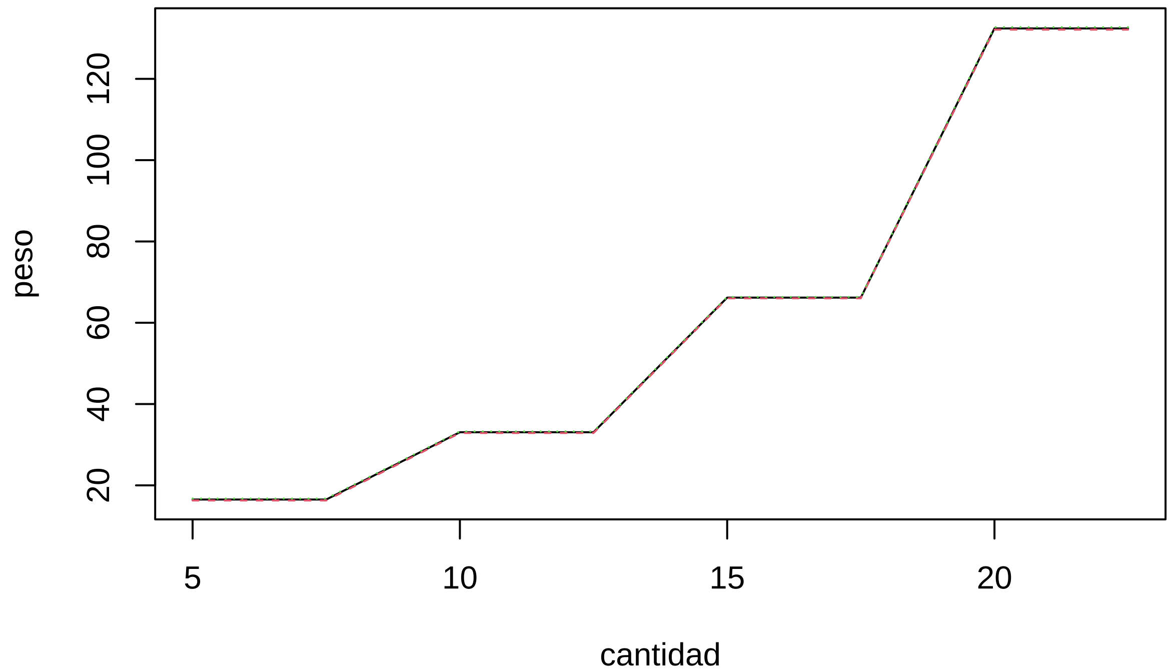


Ahora porcedemos a caulcular las bandas de confiaza y predicción

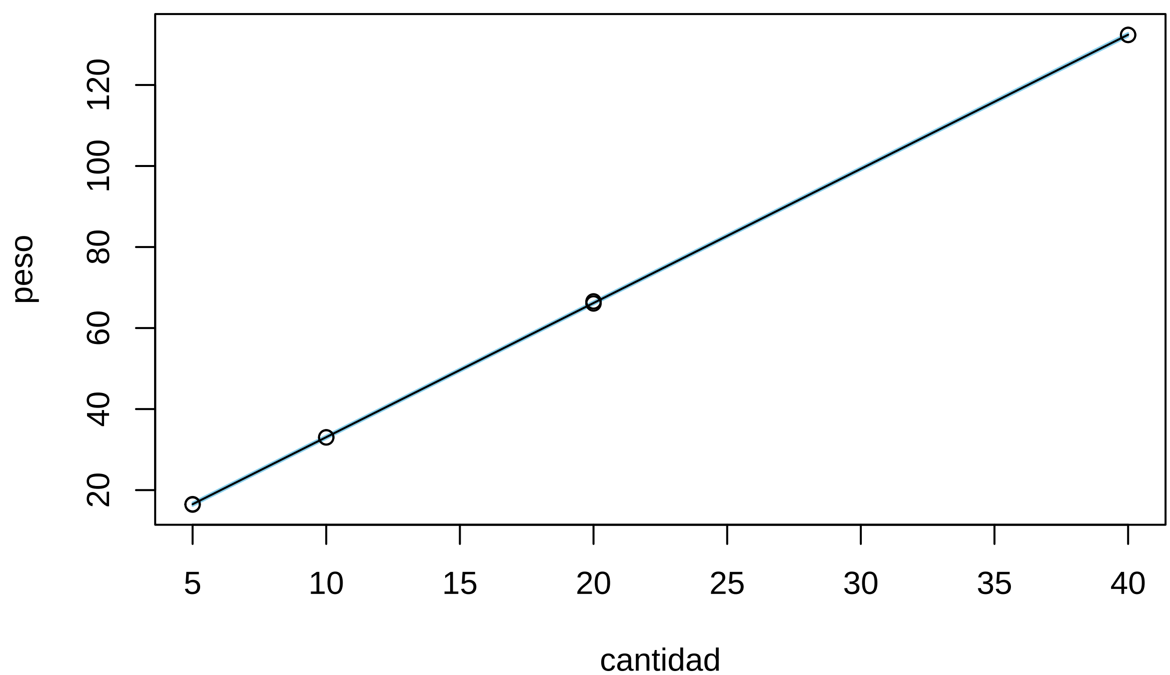
Nota: llegado a este punto ya se pueden hacer predicciones con `model.1`

```
##        fit      lwr      upr
## 1 16.51135 16.28319 16.73950
## 2 16.51135 16.28319 16.73950
## 3 33.06813 32.87794 33.25832
## 4 33.06813 32.87794 33.25832
## 5 66.18170 66.02174 66.34165
## 6 66.18170 66.02174 66.34165
```

Dibujamos las bandas de confianza, que además reflejan la incertidumbre sobre futuras observaciones:



Por último podemos hacer un gráfico con la nube de puntos y las dos bandas, la de confianza y la de predicción (Ferrari & Head, 2010).



Como podemos ver nuestros IC y de predicción sonmuy angostos (no necesariamente bueno) por lo que aun ajustando el modelo no podremos probar los supuestos, más bien nos quedamos con la interpretación de `res` calibración del modelo anteior

esto nos dice que por cada unidad de peso en el cambio influye en la cantidad de minedas en 0.01372 esto os dice que muy poco a vanzara el aumento del peso de monedas, dado que la industria o mejor dicho el banco de la republica controla hasta el peso de sus monedas, podemos hacer predicciones más no inferencia.

Nuestro modelo queda de la forma: $\hat{x}_0 = 0.01372 * \hat{y}_0 + \varepsilon_i \sim N(0,1)$

sin intercepto dado que nada se empieza con peso sino hay un cuerpo para sustentarlo.

nuestro modelo ahora si puede cumplir a cabalidad las predicciones, más los supuestos no.

4 Codigo

```
## ----message=FALSE, warning=FALSE, include=FALSE--------------------------------------------
# Paqueteria
library(readr)
library(tidyverse)
library(kableExtra)
library(magrittr)
library(janitor)
library(tidystats)
library(car)
library(faraway)
library(lmtest)
library(caret)
library(data.table)
library(MLmetrics)
library(performance)
library(mctest)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------
cantidad <-  c(5,5,10,10,20,20,40,40)
peso <-  c(16.53,16.42,33.04,32.99,66.59,66.06,132.37,132.34)
df <- data.frame(cantidad,peso)
kable(df)
plot(cantidad, peso)
title(main="Cantidad vs Peso (Coins)")


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------
boxplot(df$peso, col=rgb(0,0,1, alpha=0.5), horizontal = T)
title(main="Distribución del Peso", xlab="Peso en mg")


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------
model.1 <- lm(peso~cantidad, df)
summary(model.1)


## ----message=FALSE, warning=FALSE, include=FALSE--------------------------------------------
library(investr)


## ---------------------------------------------------------------------------------------------
res <- calibrate(model.1,y0=0,interval = "inversion", level=0.9)

plotFit(model.1, interval = "prediction",
        level = 0.9, shade = TRUE, col.pred = "skyblue")

res
```

```
## ------------------------------------------------------------------------------------------------
res <- calibrate(model.1,y0=0,interval = "Wald", level=0.9)
res


## ----echo=FALSE, message=FALSE, warning=FALSE----------------------------------------------------
x0 <- seq(min(df$cantidad), max(df$cantidad), length = 15)
dfp <- data.frame(cantidad = x0)
pred.ip <- predict(model.1, dfp, interval = "prediction",
                     se.fit =TRUE, data = datos)
head(pred.ip$fit)
matplot(x0, pred.ip$fit, type = "l", xlab = "cantidad", ylab = "peso")
title(main="Predicciones sobre el peso (Coins)")


## ----echo=FALSE, message=FALSE, warning=FALSE----------------------------------------------------
pred.ic <- predict(model.1, df, interval = "confidence", se.fit = TRUE, data =
                     df)
head(pred.ic$fit)


## ----echo=FALSE, message=FALSE, warning=FALSE----------------------------------------------------
matplot(x0[1:8], pred.ic$fit, type = "l", xlab = "cantidad", ylab = "peso")


## ----echo=FALSE, message=FALSE, warning=FALSE----------------------------------------------------
plotFit(model.1, interval = "prediction",
        level = 0.9, shade = TRUE, col.pred = "skyblue")
```