Regresion pruebas e intervalos

February 28, 2019

0.0.1 Tarea1 de Regresion Lineal Simple, parte aplicada

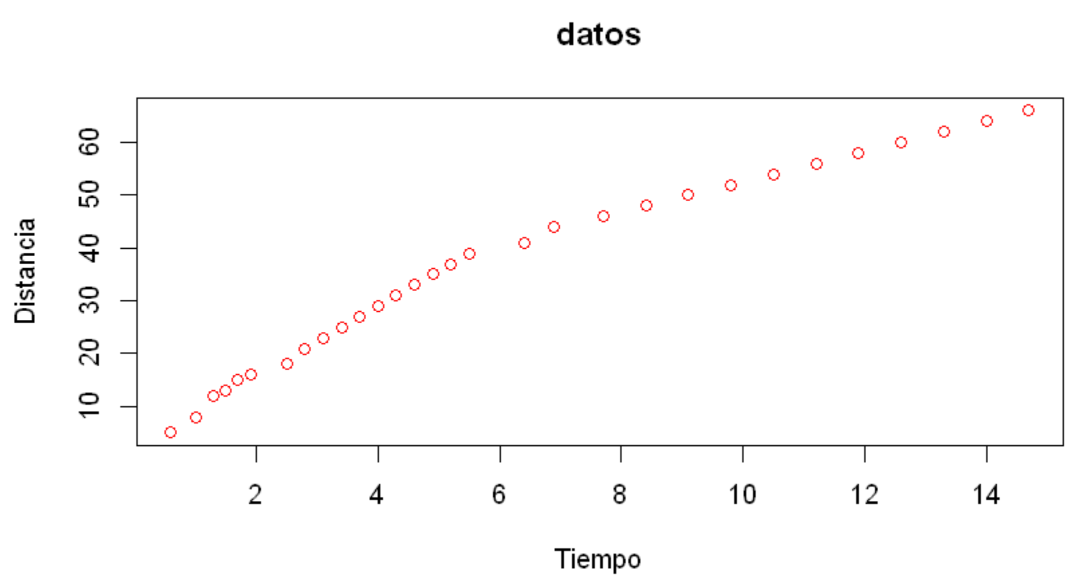
Considere los siguientes datos, que representan la distancia recorrida de un movil(x) en función del tiempo(t) a una velocidad constante v:

$$x = x_0 + vt$$

Supongamos que se puede ajustar un modelo lineal simple

```
In [214]: data=read.csv('datosML.csv',sep = ',', header=TRUE)
```

```
In [202]: options(repr.plot.width=7, repr.plot.height=4)
          plot(data[,1],data[,2],xlab="", ylab="",col='red')
          title(main = 'datos',xlab = "Tiempo", ylab = "Distancia")
```




```
In [159]: x=data$X
          y=data$Y
```

```
In [164]: modelo=lm(y~x)
          #ajustamos un modelo lineal
          summary(modelo)
```

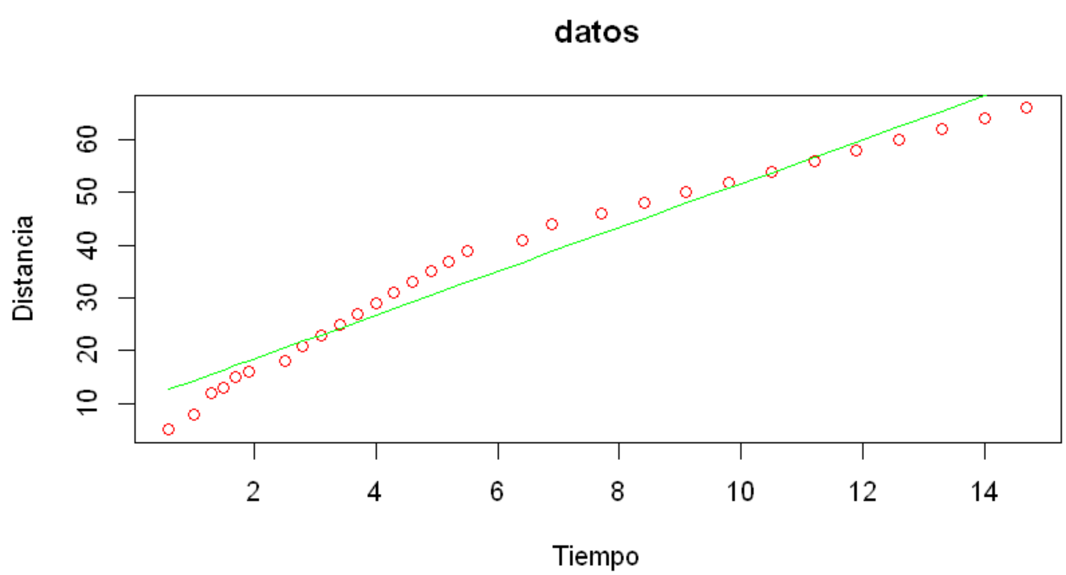
```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6629 -2.5406  0.0875  2.9634  5.9866

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   10.171      1.226   8.293 5.04e-09 ***
x              4.153      0.162  25.640  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.749 on 28 degrees of freedom
Multiple R-squared:  0.9591,Adjusted R-squared:  0.9577
F-statistic: 657.4 on 1 and 28 DF,  p-value: < 2.2e-16
```

```
In [195]: #ajuste
          options(repr.plot.width=7, repr.plot.height=4)
          plot(data[,1],data[,2],xlab="", ylab="",col='red')
          lines(data[,1],predict(modelo),col="green")
          title(main = 'datos',xlab = "Tiempo", ylab = "Distancia")
```



```
In [162]: #tamaño de la muestra
          n=length(x)
```

```
In [186]: #obtenemos las valores ajustados del modelo
          yp=predict(modelo)
          #el error estandar estimado con el estimador insesgado
          sigma_e2=sum((y-yp)^2)/(n-2)
          sigma_e2
          #media muestral
          x_barra=mean(x)
          #calculamos S_xx
          S_xx=sum((x-x_barra)^2)
```

14.052299794579

0.0.2 Prueba de hipotesis para:

$$H_0 : \beta_1 = 0 \quad vs \quad H_1 : \beta_1 \neq 0$$

```
In [196]: #obtenemos el valor de $\beta_1$ estimado
          beta_1=coef(modelo)[2]
```

```
In [106]: funcion que devuleve el valor del errror estimado
          se_beta1=function(){
              x_barra=mean(x)
              S_xx=sum((x-x_barra)^2)
              yp=predict(modelo)
              sigma_e2=sum((y-yp)^2)/(n-2)
              return(sqrt(sigma_e2/S_xx))
          }
          se_beta1()
```

0.161976773355465

```
In [190]: prueba_hipotesis=function(estimador,b_1){
          T=(beta_1-b_1)/sqrt(sigma_e2/S_xx)
          q_0975=qt(0.975, df=n-2)
          if(abs(T)>=q_0975){
          print('Se rechaza H_0')}
          else{'No se rechaza H_0'} }
          prueba_hipotesis(beta_1,0)
```

```
[1] "Se rechaza H_0"
```

1. Ahora haga la prueba de hipótesis para

$$H_0 : \beta_0 = 0 \quad vs \quad H_1 : \beta_0 \neq 0$$

, use la t de Gosset

2. Vuelva a hacer la prueba

$$H_0 : \beta_1 = 0 \quad vs \quad H_1 : \beta_1 \neq 0$$

calculando el estadístico F_0 y rechace H_0 si $F_0 > F_{1-\alpha,1,n-2}$, donde F es la F de Snedecor con 1,n-2 grados de libertad.

In [201]:
```
F_095=qf(0.95,2,n-2) #Cuantil de una F con 2,n-2 grados de libertad
F_095
```

3.34038555823776

0.0.3 Intervalo de confianza al 95% para β_1

In [217]:
```
intervalo = function(estimador,t_n_2){
    se=se_beta1()
    lim_inferior=estimador-se*t_n_2
    lim_superior=estimador+se*t_n_2
    inter=c(lim_inferior,lim_superior)
    lista=list('lim_inf'=lim_inferior,'lim_sup'=lim_superior)
    return(lista)
}
intervalo(beta_1,q_0975)
```

$lim_inf x: 3.82135426034553

$lim_sup x: 4.4849430190381

3.- Ahora debera construir un intervalo para β_0 y para σ^2 al 95%
4.- Construya un intervalo de confianza para una nueva observación donde $x_{nueva} = 8.0$