scriptregresion.R

maru

Thu Nov 16 09:20:50 2017

```r
setwd("~/Dropbox/Estadistica(Q)verano2017/clasespracticasverano2017")
toluca<-read.table("toluca.txt",header=T)  #header=T lee los nombres de las columnas desde el archivo

attach(toluca) #para poder trabajar con las variables de toluca directamente
View(toluca)
####################################################################
# a) grafico de dispersion
####################################################################

plot(tamanio,horas,main="diagrama de dispersion
     de horas vs. tamanio")
# ponemos el tamanio como variable independiente, las horas como variable dependiente

####################################################################
#b) El modelo
####################################################################
# Responder con papel y lapiz. Definir variables y parametros. Cuantos parametros
# tiene este modelo?

####################################################################
#c) Ajuste del modelo
####################################################################
#la instruccion lm (linear model) ajusta la recta de minimos cuadrados a los datos

salida<-lm(horas~tamanio)
#como el aov, el lm produce un objeto (una lista) que contiene muchas cosas
names(salida)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

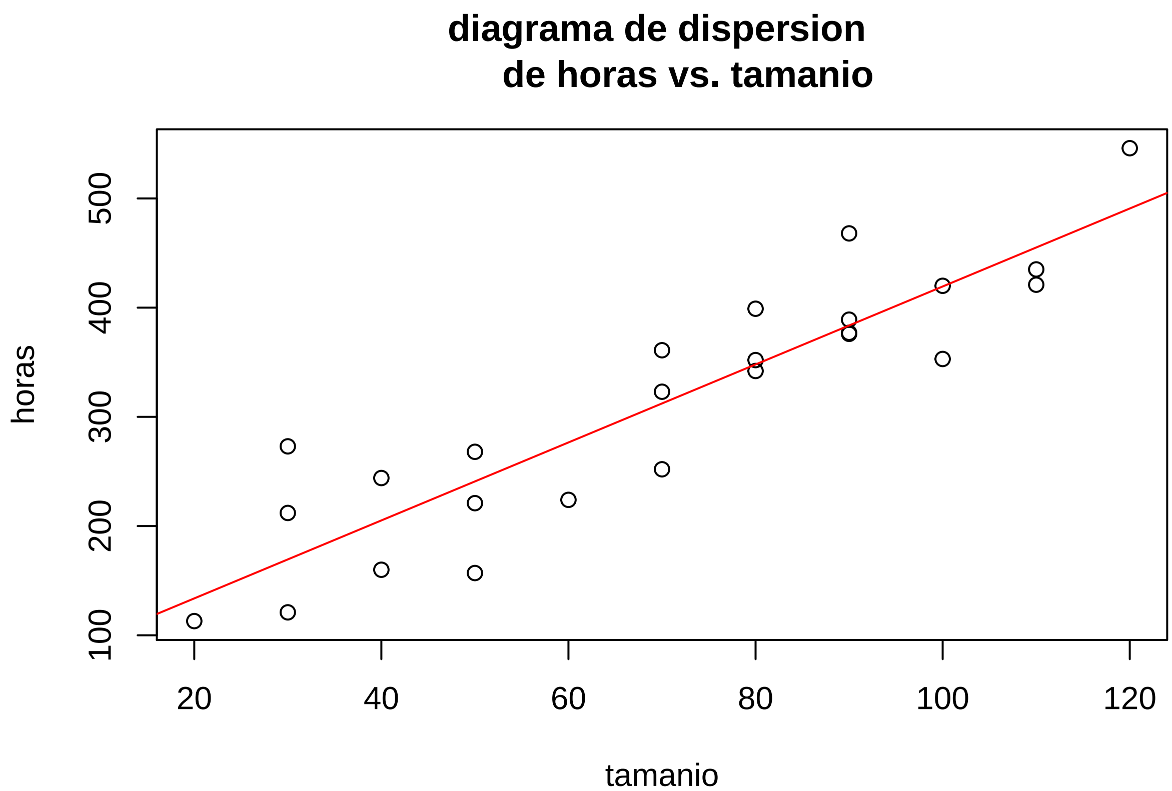
```r
# para obtener los coeficientes, hay dos formas
salida$coefficients
```

```
## (Intercept)     tamanio
##   62.365859    3.570202
```

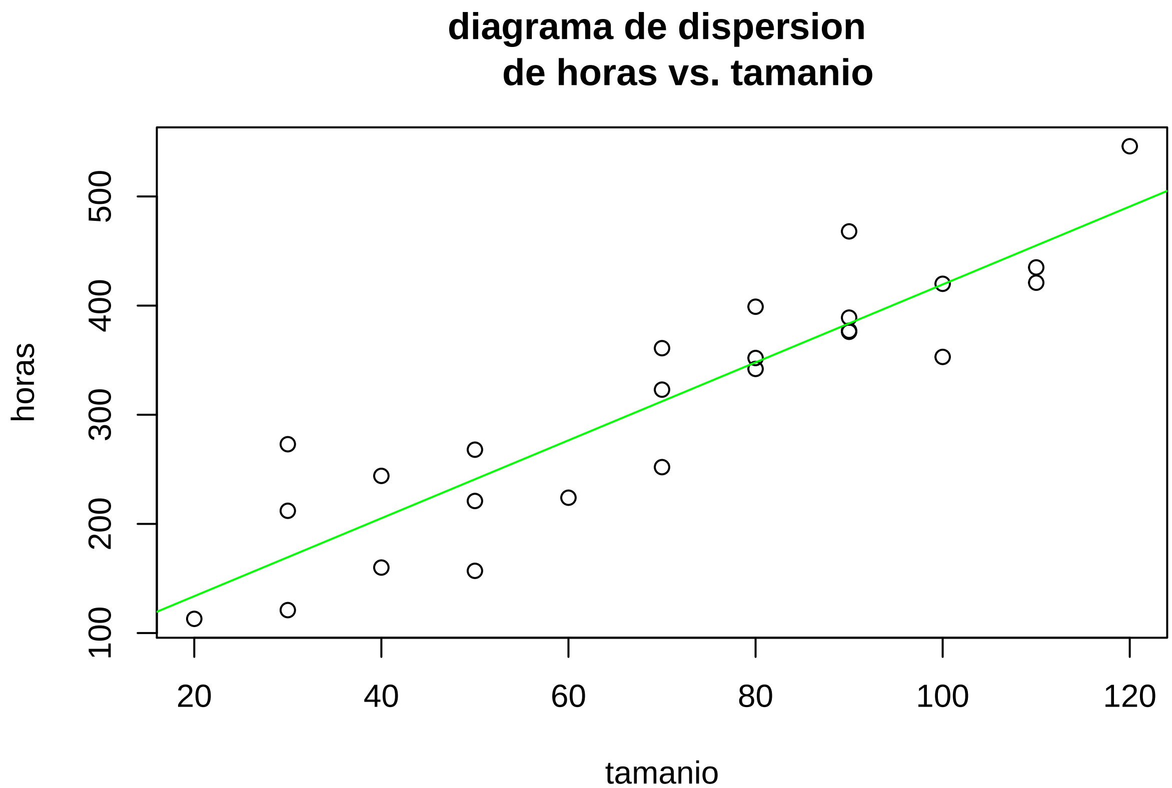
```r
coefficients(salida)
```

```
## (Intercept)     tamanio
##   62.365859    3.570202
```

```
# (Intercept)      tamanio
# 62.365859    3.570202

#superponemos la recta de minimos cuadrados al grafico
plot(tamanio,horas,main="diagrama de dispersion
     de horas vs. tamanio")
abline(salida,col="red")
```




```
# la instruccion abline agrega una o mas lineas rectas al grafico
# se le pueden dar la ordenada al origen y la pendiente, alternativamente
plot(tamanio,horas,main="diagrama de dispersion
     de horas vs. tamanio")
abline(62.3659,3.5702,col="green")
```




```r
?abline

summary(salida)
```

```
##
## Call:
## lm(formula = horas ~ tamanio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -83.876 -34.088  -5.982  38.826 103.528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.366     26.177   2.382   0.0259 *
## tamanio        3.570      0.347  10.290 4.45e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.82 on 23 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8138
## F-statistic: 105.9 on 1 and 23 DF,  p-value: 4.449e-10
```

```r
 names(summary(salida))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
names(salida)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```r
######################################################################
#d) estimamos el valor esperado de las horas cuando tamanio = 90 y 120
######################################################################

coefficients(salida)[1]+coefficients(salida)[2]*90
```

```
## (Intercept)
##     383.684
```

```r
coefficients(salida)[1]+coefficients(salida)[2]*120
```

```
## (Intercept)
##    490.7901
```

```r
# comparar con los fitted.values que calcula directamente el R.
# que observacion corresponde a tamanio=90?
fitted.values(salida)
```

```
##        1        2        3        4        5        6        7        8
## 133.7699 169.4719 169.4719 169.4719 205.1739 205.1739 240.8760 240.8760
##        9       10       11       12       13       14       15       16
## 240.8760 276.5780 312.2800 312.2800 312.2800 347.9820 347.9820 347.9820
##       17       18       19       20       21       22       23       24
## 383.6840 383.6840 383.6840 383.6840 419.3861 419.3861 455.0881 455.0881
##       25
## 490.7901
```

```r
cbind(tamanio,horas,fitted.values(salida),residuals(salida))
```

```
##    tamanio horas
## 1       20   113 133.7699 -20.7698990
## 2       30   121 169.4719 -48.4719192
## 3       30   212 169.4719  42.5280808
## 4       30   273 169.4719 103.5280808
## 5       40   160 205.1739 -45.1739394
## 6       40   244 205.1739  38.8260606
## 7       50   221 240.8760 -19.8759596
## 8       50   157 240.8760 -83.8759596
## 9       50   268 240.8760  27.1240404
## 10      60   224 276.5780 -52.5779798
## 11      70   252 312.2800 -60.2800000
## 12      70   361 312.2800  48.7200000
## 13      70   323 312.2800  10.7200000
## 14      80   399 347.9820  51.0179798
```

```
## 15      80   342 347.9820  -5.9820202
## 16      80   352 347.9820   4.0179798
## 17      90   389 383.6840   5.3159596
## 18      90   468 383.6840  84.3159596
## 19      90   377 383.6840  -6.6840404
## 20      90   376 383.6840  -7.6840404
## 21     100   353 419.3861 -66.3860606
## 22     100   420 419.3861   0.6139394
## 23     110   435 455.0881 -20.0880808
## 24     110   421 455.0881 -34.0880808
## 25     120   546 490.7901  55.2098990
```

```
# cual es el valor ajustado para la vigesima observacion (la numero 20)?
# compararla con los valores ajustados para las observaciones 17 a 19.
# cual es el residuo de la vigesima observacion (la numero 20)?
# compararla con los residuos de las observaciones 17 a 19.

######################################################################
#e) intevalos de confianza para coeficientes
######################################################################

qt(0.975,df=23) #2.068658
```

```
## [1] 2.068658
```

```
#los coeficientes estimados
summary(salida)$coef[,1]
```

```
## (Intercept)     tamanio
##   62.365859    3.570202
```

```
# (Intercept)     tamanio
# 62.365859    3.570202


#el desvio estandar de los coeficientes estimados
summary(salida)$coef[,2]
```

```
## (Intercept)     tamanio
##  26.1774339   0.3469722
```

```
#(Intercept)     tamanio
#26.1774339   0.3469722
# donde se ven en el objeto "salida"?

# que parametro estima la siguiente instruccion?
summary(salida)$sigma
```

```
## [1] 48.82331
```

```r
# 48.82331
# donde figura esta informacion en la tabla que da summary(salida)?
# la podemos calcular a mano

n<-length(horas)
sqrt(sum(salida$res^2)/(n-2))
```

```
## [1] 48.82331
```

```r
#como se calculan los desvios estandares de los coeficientes a partir de el?

#los extremos superiores de los ic
summary(salida)$coef[,1]+ summary(salida)$coef[,2]*qt(0.975,df=23)
```

```
## (Intercept)     tamanio
##  116.518006    4.287969
```

```r
#los extremos inferiores de los ic
summary(salida)$coef[,1]- summary(salida)$coef[,2]*qt(0.975,df=23)
```

```
## (Intercept)     tamanio
##    8.213711    2.852435
```

```r
# el R calcula todo automaticamente
confint(salida)
```

```
##                 2.5 %     97.5 %
## (Intercept) 8.213711 116.518006
## tamanio     2.852435   4.287969
```

```r
####################################################################
#f) Rcuadrado
####################################################################
# en "salida" puede verse. Donde? Que significa? Sino,
names(summary(salida))
```

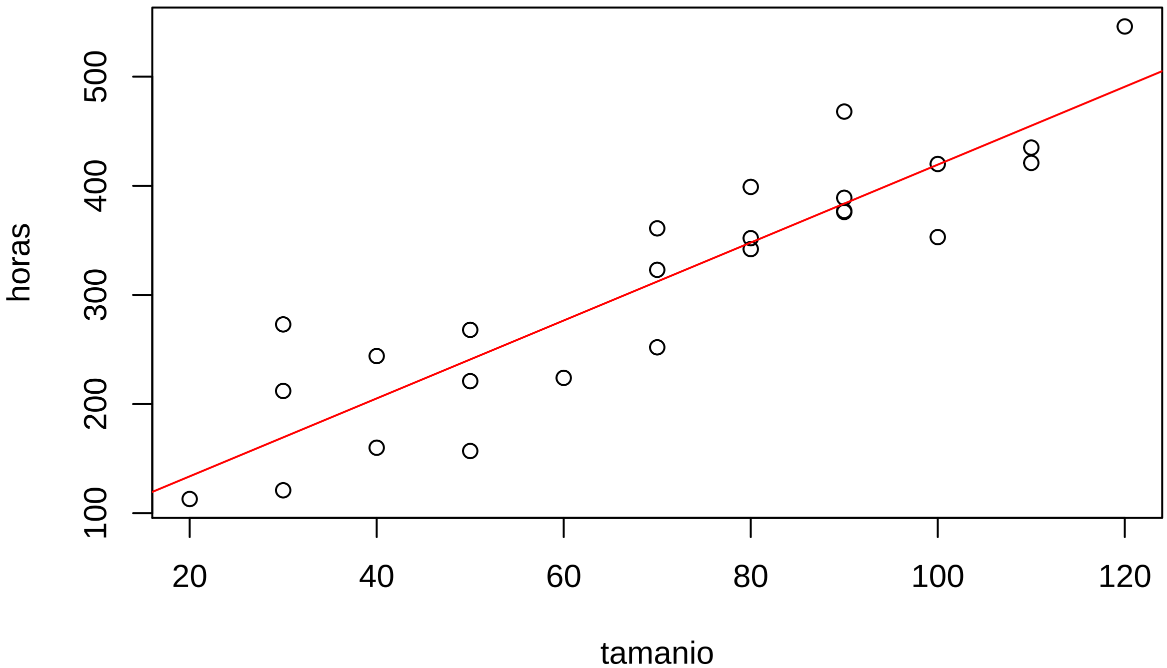
```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
summary(salida)$r.squared
```

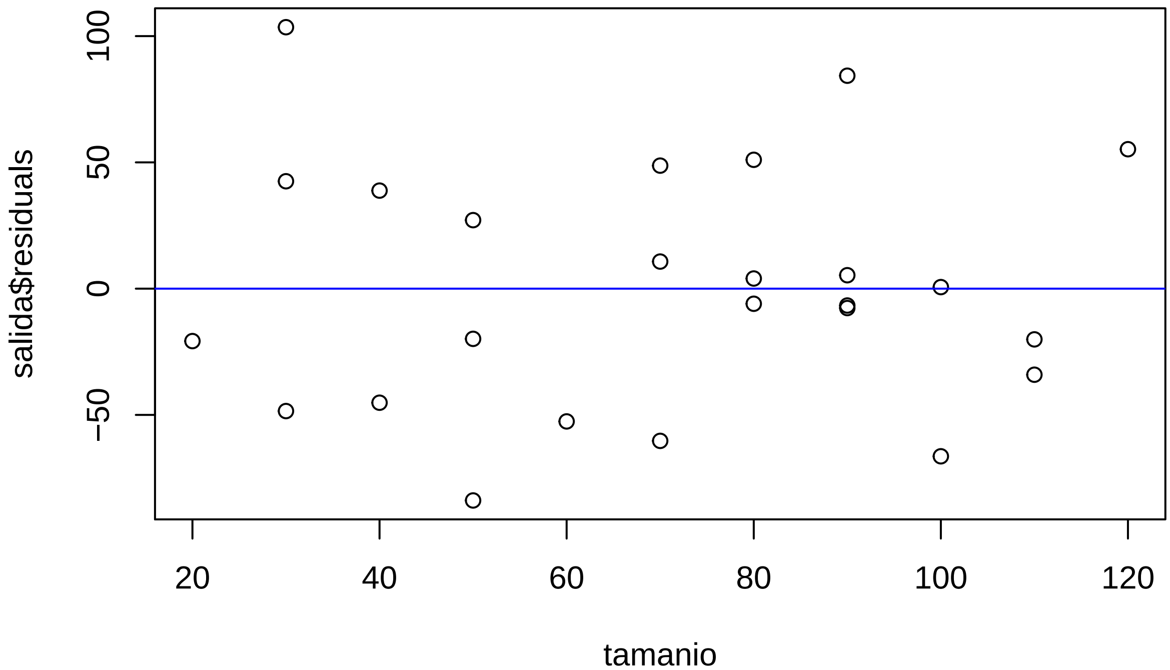
```
## [1] 0.8215335
```

```r
#0.8215335


####################################################################
#g) validacion de supuestos
```

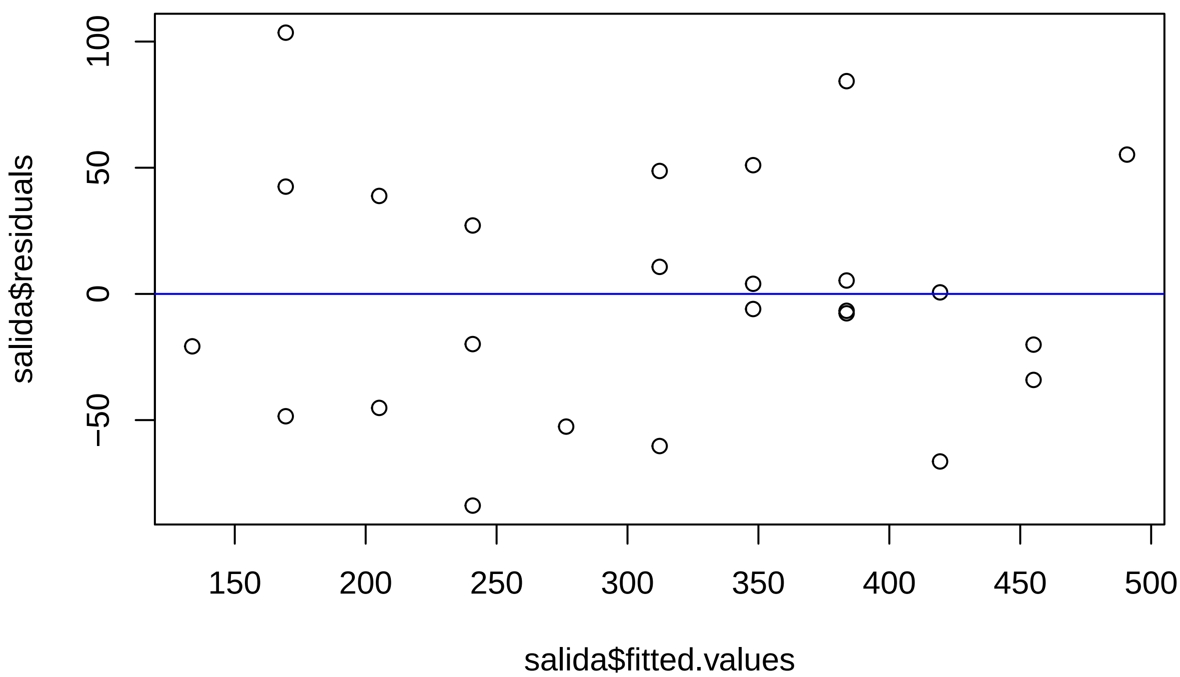
```r
########################################################################
# en funcion de los siguientes graficos, decida si se satisfacen los supuestos de linealidad
#y homoscedasticidad

#1)
plot(tamanio,horas)
abline(salida,col=2)
```




```r
#2)
plot(tamanio,salida$residuals)
abline(a=0,b=0,col=4)
```

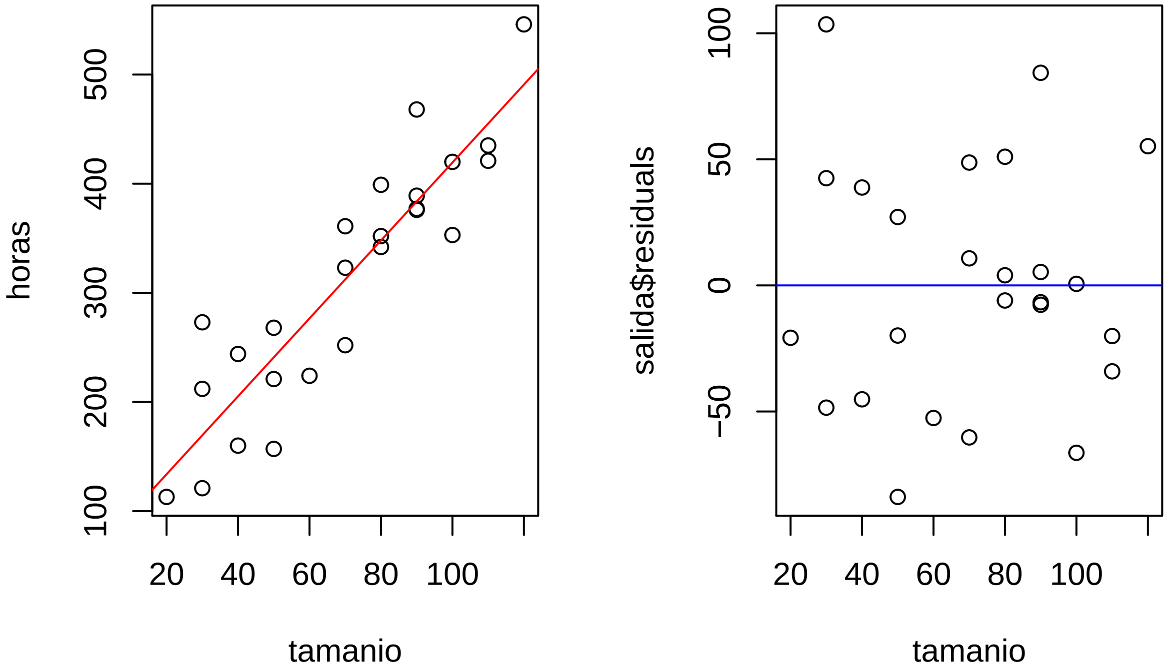



```r
#3)
plot(salida$fitted.values,salida$residuals)
abline(a=0,b=0,col=4)
```

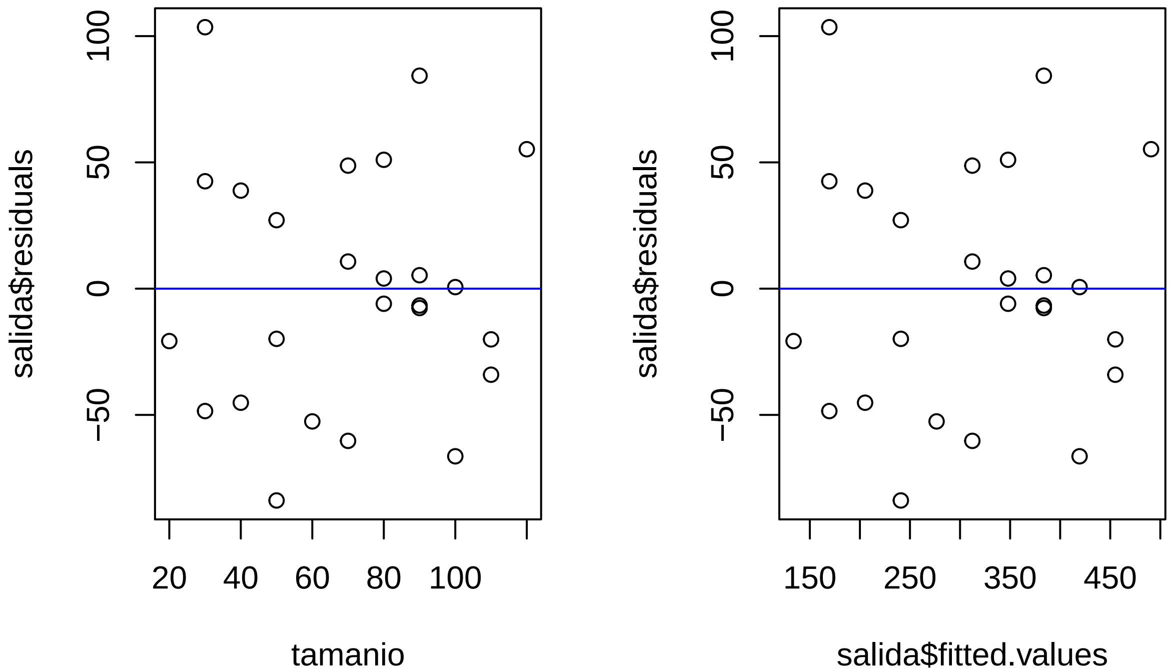



```r
# que se grafica en cada uno de los graficos anteriores?

#compare graficos 1 y 2 mediante las siguientes instrucciones

par(mfrow=c(1,2))
plot(tamanio,horas)
```

```r
abline(salida,col=2)

plot(tamanio,salida$residuals)
abline(a=0,b=0,col=4)
```




```r
#compare graficos 2 y 3 mediante las siguientes instrucciones

par(mfrow=c(1,2))

plot(tamanio,salida$residuals)
abline(a=0,b=0,col=4)

plot(salida$fitted.values,salida$residuals)
abline(a=0,b=0,col=4)
```

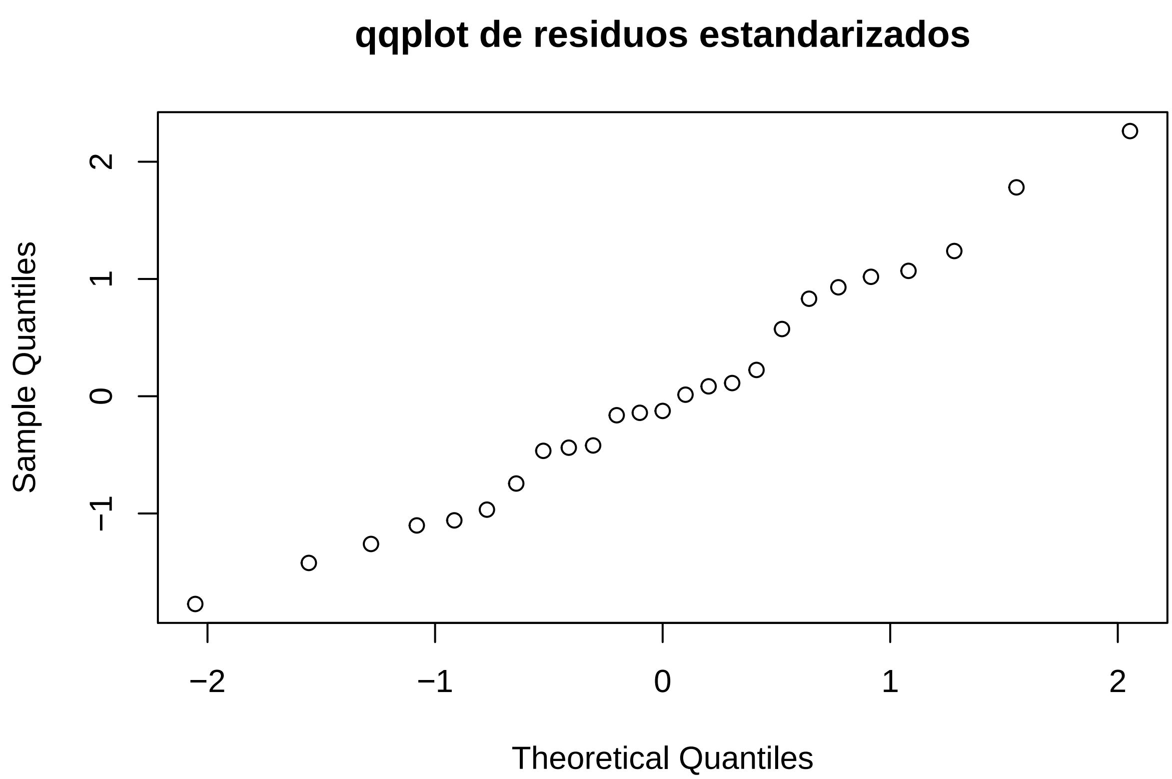



```r
#Rptas.:
#vemos que se satisfacen los supuestos de linealidad y homoscedasticidad ya que en el grafico
#de dispersion vemos que los puntos estan cerca de una recta y la variabilidad de las Y's es
#parecida para los distintos valores de x.
#En los 2 graficos restantes vemos una nube de puntos sin estructura alrededor de la recta y=0


#que hay en cada una de las columnas de la siguiente matriz? relacione sus valores

cbind(horas,salida$fitted.values,salida$residuals,rstandard(salida))
```

```
##    horas
## 1    113 133.7699 -20.7698990 -0.46589980
## 2    121 169.4719 -48.4719192 -1.05881760
## 3    212 169.4719  42.5280808  0.92898076
## 4    273 169.4719 103.5280808  2.26146098
## 5    160 205.1739 -45.1739394 -0.96751587
## 6    244 205.1739  38.8260606  0.83155975
## 7    221 240.8760 -19.8759596 -0.41993651
## 8    157 240.8760 -83.8759596 -1.77211959
## 9    268 240.8760  27.1240404  0.57307295
## 10   224 276.5780 -52.5779798 -1.10201238
## 11   252 312.2800 -60.2800000 -1.26011565
## 12   361 312.2800  48.7200000  1.01846109
## 13   323 312.2800  10.7200000  0.22409489
## 14   399 347.9820  51.0179798  1.06931543
## 15   342 347.9820  -5.9820202 -0.12538063
## 16   352 347.9820   4.0179798  0.08421517
## 17   389 383.6840   5.3159596  0.11231485
## 18   468 383.6840  84.3159596  1.78141585
## 19   377 383.6840  -6.6840404 -0.14121947
## 20   376 383.6840  -7.6840404 -0.16234734
```

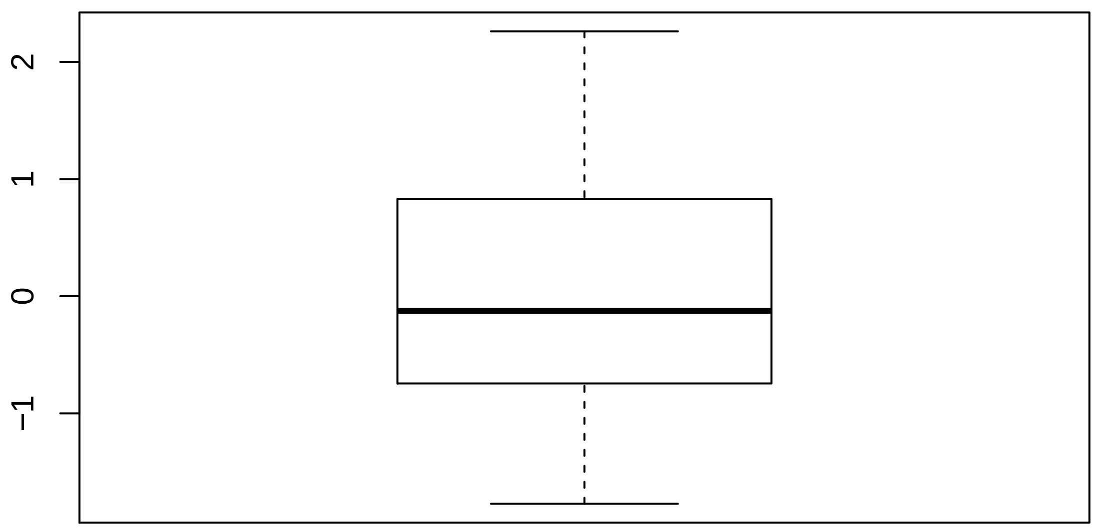
```
## 21   353 419.3861 -66.3860606 -1.42182789
## 22   420 419.3861   0.6139394  0.01314909
## 23   435 455.0881 -20.0880808 -0.43880279
## 24   421 455.0881 -34.0880808 -0.74461792
## 25   546 490.7901  55.2098990  1.23844036
```

```r
#en funcion de los siguientes graficos y salidas, decida si se satisface el supuesto de normalidad

par(mfrow=c(1,1))

qqnorm(rstandard(salida),main="qqplot de residuos estandarizados")
```




```r
boxplot(rstandard(salida),main="boxplot de residuos estandarizados")
```

boxplot de residuos estandarizados




```r
shapiro.test(rstandard(salida))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(salida)
## W = 0.97861, p-value = 0.8563
```

```r
#Rptas.:
# Shapiro-Wilk normality test
#
# data:  rstandard(salida)
# W = 0.9786, p-value = 0.8563

#segun el qqplot y boxplot, los residuos estandarizados son aproximadamente normales y segun
#el test shapiro no rechazamos el supuesto de normalidad a nivel 0.2.
#con lo cual podemos asumir valido el supuesto de normalidad de los errores.


##################################################
#h) estimador e IC para E(Y) cuando X=90 y X=120
##################################################

#para hallar el estimador de E(Y) cuando X=90 tenemos 2 posibilidades
#a mano
salida$coef
```

```
## (Intercept)     tamanio
##   62.365859    3.570202
```

```r
62.365859+3.570202*90
```

```
## [1] 383.684
```

```r
#383.684
coefficients(salida)[1]+coefficients(salida)[2]*90
```

```
## (Intercept)
##     383.684
```

```r
#con los valores ajustados de la salida
cbind(tamanio,salida$fitted.values)
```

```
##    tamanio
## 1       20 133.7699
## 2       30 169.4719
## 3       30 169.4719
## 4       30 169.4719
## 5       40 205.1739
## 6       40 205.1739
## 7       50 240.8760
## 8       50 240.8760
## 9       50 240.8760
## 10      60 276.5780
## 11      70 312.2800
## 12      70 312.2800
## 13      70 312.2800
## 14      80 347.9820
## 15      80 347.9820
## 16      80 347.9820
## 17      90 383.6840
## 18      90 383.6840
## 19      90 383.6840
## 20      90 383.6840
## 21     100 419.3861
## 22     100 419.3861
## 23     110 455.0881
## 24     110 455.0881
## 25     120 490.7901
```

```r
#que componente/s tengo que mirar? corroborar que es igual al calculo hecho a mano

#17 a 20

#383.6840


#para calcular un Ic para E(Y) cuando X=90 tambien tenemos dos posibilidades

#1) con el siguiente comando, que calcula los estimadores e IC 95% correspondientes a todas
# las observaciones

predict(salida,interval="confidence")
```

```
##         fit       lwr      upr
## 1  133.7699  92.58736 174.9524
```

```
## 2  169.4719 134.36734 204.5765
## 3  169.4719 134.36734 204.5765
## 4  169.4719 134.36734 204.5765
## 5  205.1739 175.64938 234.6985
## 6  205.1739 175.64938 234.6985
## 7  240.8760 216.09481 265.6571
## 8  240.8760 216.09481 265.6571
## 9  240.8760 216.09481 265.6571
## 10 276.5780 255.14090 298.0151
## 11 312.2800 292.08026 332.4797
## 12 312.2800 292.08026 332.4797
## 13 312.2800 292.08026 332.4797
## 14 347.9820 326.54494 369.4191
## 15 347.9820 326.54494 369.4191
## 16 347.9820 326.54494 369.4191
## 17 383.6840 358.90290 408.4652
## 18 383.6840 358.90290 408.4652
## 19 383.6840 358.90290 408.4652
## 20 383.6840 358.90290 408.4652
## 21 419.3861 389.86150 448.9106
## 22 419.3861 389.86150 448.9106
## 23 455.0881 419.98350 490.1927
## 24 455.0881 419.98350 490.1927
## 25 490.7901 449.60756 531.9726
```

```
#o tambien...

predict(salida,interval="confidence",level=0.95)
```

```
##         fit       lwr      upr
## 1  133.7699  92.58736 174.9524
## 2  169.4719 134.36734 204.5765
## 3  169.4719 134.36734 204.5765
## 4  169.4719 134.36734 204.5765
## 5  205.1739 175.64938 234.6985
## 6  205.1739 175.64938 234.6985
## 7  240.8760 216.09481 265.6571
## 8  240.8760 216.09481 265.6571
## 9  240.8760 216.09481 265.6571
## 10 276.5780 255.14090 298.0151
## 11 312.2800 292.08026 332.4797
## 12 312.2800 292.08026 332.4797
## 13 312.2800 292.08026 332.4797
## 14 347.9820 326.54494 369.4191
## 15 347.9820 326.54494 369.4191
## 16 347.9820 326.54494 369.4191
## 17 383.6840 358.90290 408.4652
## 18 383.6840 358.90290 408.4652
## 19 383.6840 358.90290 408.4652
## 20 383.6840 358.90290 408.4652
## 21 419.3861 389.86150 448.9106
## 22 419.3861 389.86150 448.9106
## 23 455.0881 419.98350 490.1927
## 24 455.0881 419.98350 490.1927
```

```
## 25 490.7901 449.60756 531.9726
```

```r
cbind(tamanio,predict(salida,interval="confidence",level=0.95))
```

```
##    tamanio      fit       lwr      upr
## 1       20 133.7699  92.58736 174.9524
## 2       30 169.4719 134.36734 204.5765
## 3       30 169.4719 134.36734 204.5765
## 4       30 169.4719 134.36734 204.5765
## 5       40 205.1739 175.64938 234.6985
## 6       40 205.1739 175.64938 234.6985
## 7       50 240.8760 216.09481 265.6571
## 8       50 240.8760 216.09481 265.6571
## 9       50 240.8760 216.09481 265.6571
## 10      60 276.5780 255.14090 298.0151
## 11      70 312.2800 292.08026 332.4797
## 12      70 312.2800 292.08026 332.4797
## 13      70 312.2800 292.08026 332.4797
## 14      80 347.9820 326.54494 369.4191
## 15      80 347.9820 326.54494 369.4191
## 16      80 347.9820 326.54494 369.4191
## 17      90 383.6840 358.90290 408.4652
## 18      90 383.6840 358.90290 408.4652
## 19      90 383.6840 358.90290 408.4652
## 20      90 383.6840 358.90290 408.4652
## 21     100 419.3861 389.86150 448.9106
## 22     100 419.3861 389.86150 448.9106
## 23     110 455.0881 419.98350 490.1927
## 24     110 455.0881 419.98350 490.1927
## 25     120 490.7901 449.60756 531.9726
```

```r
#que fila/s tengo que mirar? ?cual es el IC?

#17 a 20
#[358.90290,408.4652]

#2) con el siguiente comando, que calcula el estimador e IC correspondiente a X=90 unicamente
predict(salida,newdata=data.frame(tamanio=90),interval="confidence")
```

```
##       fit      lwr      upr
## 1 383.684 358.9029 408.4652
```

```r
#    fit      lwr      upr
# 1 383.684 358.9029 408.4652

####################################################
#estimador e IC para E(Y) cuando X=120
####################################################

#resolver utilizando los comandos anteriores que considere adecuados, adaptandolos
#para el caso X=120
```

```r
#estimador de E(Y) cuando X=120

salida$coef
```

```
## (Intercept)     tamanio
##   62.365859    3.570202
```

```r
62.365859+3.570202*120
```

```
## [1] 490.7901
```

```r
#490.7901

#Ic para E(Y) cuando X=120

predict(salida,newdata=data.frame(tamanio=120),interval="confidence")
```

```
##        fit      lwr      upr
## 1 490.7901 449.6076 531.9726
```

```r
#     fit      lwr      upr
# 1 490.7901 449.6076 531.9726

#el IC es [449.6076, 531.9726]

###################################################
#comparacion entre las longitudes de los IC's
###################################################

#que intervalo tiene menor longitud? A que se debe?

IC.90<-predict(salida,newdata=data.frame(tamanio=90),interval="confidence")
IC.120<-predict(salida,newdata=data.frame(tamanio=120),interval="confidence")

long.IC.90<-IC.90[3]-IC.90[2]
long.IC.120<-IC.120[3]-IC.120[2]

long.IC.90 # 49.56229
```

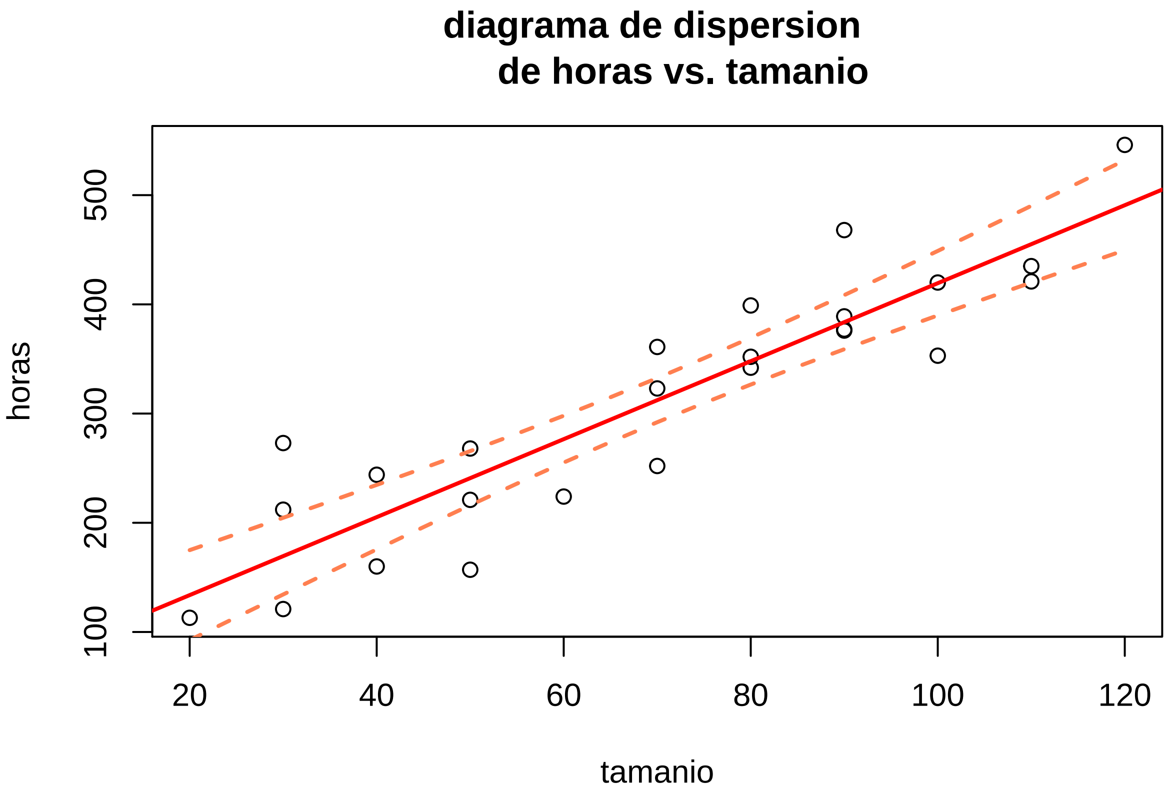
```
## [1] 49.56229
```

```r
long.IC.120 # 82.36508
```

```
## [1] 82.36508
```

```r
#graficamente
plot(tamanio,horas,main="diagrama de dispersion
     de horas vs. tamanio")
abline(salida,col="red",lwd=2)
```

```r
equis<-seq(from=20,to=120,by=5)

IC<-predict(salida,newdata=data.frame(tamanio=equis),interval="confidence")

points(equis,IC[,3],type="l",col="coral",lty=2,lwd=2)
points(equis,IC[,2],type="l",col="coral",lty=2,lwd=2)
```




```r
#lty es una instruccion para que haga la linea punteada, lwd setea el grosor de la linea que dibuja


####################################################
#i) Intervalos de prediccion
####################################################
#para resolver este item se pueden utilizar los mismos comandos que en el item anterior pero
#en la funcion "predict" hay que cambiar el argumento "interval", poniendo "prediction" en lugar
#de #confidence"

#predictor de Y cuando X=90 (idem estimador de E(Y): 383.6840)

#IP para Y cuando X=90, tambien tenemos dos posibilidades
####################################################

#1)el siguiente comando que calcula los predictores e IP correspondientes a todas
#las observaciones

predict(salida,interval="prediction")
```

```
## Warning in predict.lm(salida, interval = "prediction"): predictions on current data refer to _future_
```

```
##         fit       lwr      upr
## 1  133.7699  24.69771 242.8421
## 2  169.4719  62.54638 276.3975
## 3  169.4719  62.54638 276.3975
## 4  169.4719  62.54638 276.3975
## 5  205.1739  99.94828 310.3996
## 6  205.1739  99.94828 310.3996
## 7  240.8760 136.88151 344.8704
## 8  240.8760 136.88151 344.8704
## 9  240.8760 136.88151 344.8704
## 10 276.5780 173.32931 379.8267
## 11 312.2800 209.28112 415.2789
## 12 312.2800 209.28112 415.2789
## 13 312.2800 209.28112 415.2789
## 14 347.9820 244.73335 451.2307
## 15 347.9820 244.73335 451.2307
## 16 347.9820 244.73335 451.2307
## 17 383.6840 279.68959 487.6785
## 18 383.6840 279.68959 487.6785
## 19 383.6840 279.68959 487.6785
## 20 383.6840 279.68959 487.6785
## 21 419.3861 314.16040 524.6117
## 22 419.3861 314.16040 524.6117
## 23 455.0881 348.16254 562.0136
## 24 455.0881 348.16254 562.0136
## 25 490.7901 381.71792 599.8623
```

```
#que fila/s tengo que mirar? ?cual es el IP?

#filas 17 a 20, IP=[279.68959,487.6785]


#2) el siguiente comando calcula el predictor e IP correspondiente a X=90 unicamente
predict(salida,newdata=data.frame(tamanio=90),interval="prediction")
```

```
##       fit      lwr      upr
## 1 383.684 279.6896 487.6785
```

```
###################################################
#predictor e IP para Y cuando X=120
###################################################

#predictor de Y cuando X=120 (idem estimador de E(Y): 490.7901)
#IP para Y cuando X=120, tambien tenemos dos posibilidades

predict(salida,newdata=data.frame(tamanio=120),interval="prediction")
```

```
##        fit      lwr      upr
## 1 490.7901 381.7179 599.8623
```

```
# fit       lwr      upr
# 1 490.7901 381.7179 599.8623
```

```r
##################################################
#comparacion de las longitudes de los IP's entre s?
##################################################

#que intervalo tiene menor longitud? a que se debe?

IP.90<-predict(salida,newdata=data.frame(tamanio=90),interval="prediction")
IP.120<-predict(salida,newdata=data.frame(tamanio=120),interval="prediction")

long.IP.90<-IP.90[3]-IP.90[2]
long.IP.120<-IP.120[3]-IP.120[2]

long.IP.90 # 207.9889
```

```
## [1] 207.9889
```

```r
long.IP.120 # 218.1444
```

```
## [1] 218.1444
```

```r
##################################################
#comparacion de las longitudes de los IP's con los IC's
##################################################

long.IC.90 #  49.56229
```

```
## [1] 49.56229
```

```r
long.IP.90 # 207.9889
```

```
## [1] 207.9889
```

```r
long.IC.120 #  82.36508
```

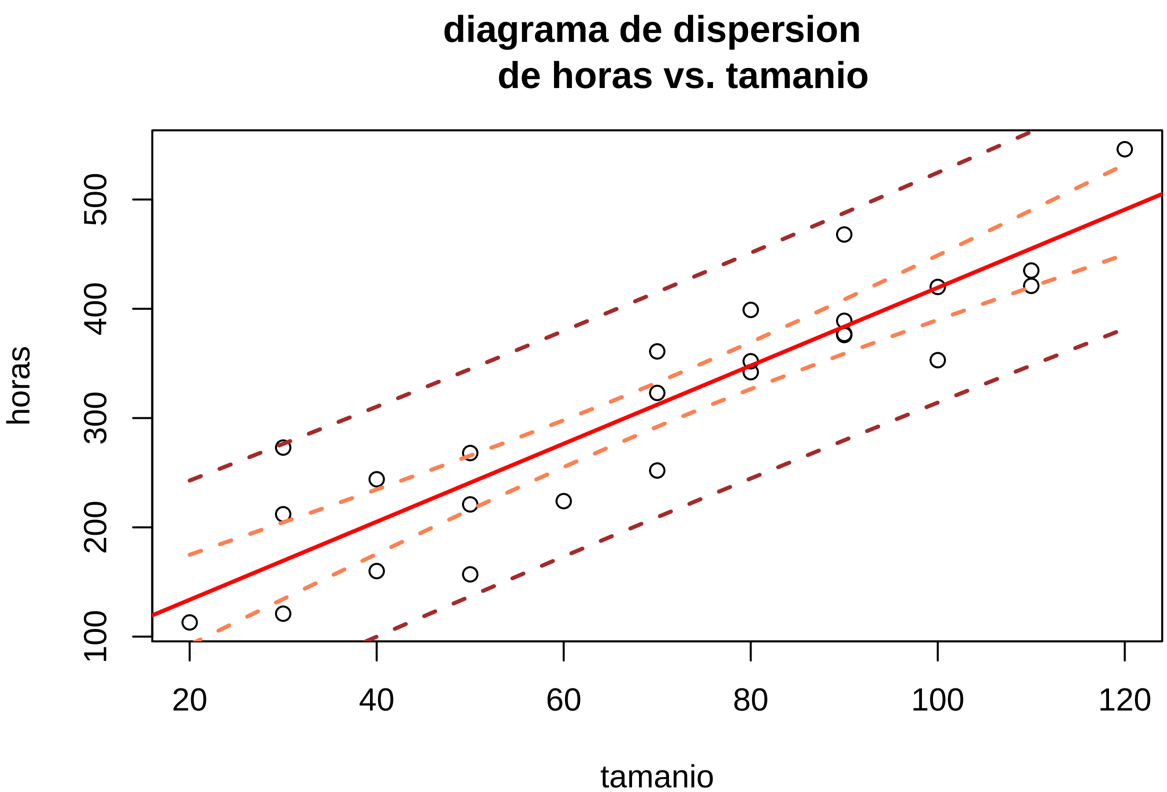
```
## [1] 82.36508
```

```r
long.IP.120 # 218.1444
```

```
## [1] 218.1444
```

```r
plot(tamanio,horas,main="diagrama de dispersion
     de horas vs. tamanio")
abline(salida,col="red",lwd=2)

equis<-seq(from=20,to=120,by=5)
IC<-predict(salida,newdata=data.frame(tamanio=equis),interval="confidence")
IP<-predict(salida,newdata=data.frame(tamanio=equis),interval="prediction")
```

```r
points(equis,IC[,3],type="l",col="coral",lty=2,lwd=2)
points(equis,IC[,2],type="l",col="coral",lty=2,lwd=2)

points(equis,IP[,3],type="l",col="brown",lty=2,lwd=2)
points(equis,IP[,2],type="l",col="brown",lty=2,lwd=2)
```




```r
####################################################
#j) Regresion inversa
####################################################

xpredicho <- (394-coef(salida)[1])/coef(salida)[2]
n<-length(tamanio)
# para el intervalo de confianza
cuantil<-qt(1-0.05/2,df=(n-2))
ese<-summary(salida)$sigma #(es el estimador de sigma)
sumaxx<-sum((tamanio-mean(tamanio))^2)
sumaxx
```

```
## [1] 19800
```

```r
# o tambien
var(tamanio)*(length(tamanio)-1)
```

```
## [1] 19800
```

```r
lio<-ese^2/(coef(salida)[2])^2*(1 + 1/n + (394 - mean(horas))^2/((coef(salida)[2])^2*sumaxx))


#intervalo inverso
xpredicho - cuantil * sqrt(lio)
```

```
## (Intercept)
##    63.67516
```

```r
xpredicho + cuantil * sqrt(lio)
```

```
## (Intercept)
##    122.1038
```