

Tarea 12 - Regresión Lineal con Análisis de Errores

Héctor Hibrán Tapia Fernández - A01661114

2024-09-04

El Validez del Modelo

```
df = read.csv("./Estatura-peso_HyM.csv")

df_hombres = subset(df, df$Sexo == "H")
df_mujeres = subset(df, df$Sexo == "M")

df_medidas = data.frame(
  "H-Estatura" = df_hombres$Estatura,
  "H-Peso" = df_hombres$Peso,
  "M-Estatura" = df_mujeres$Estatura,
  "M-Peso" = df_mujeres$Peso
)
```

Modelo 1.

```
Modelo1H = lm(Peso ~ Estatura, data = df_hombres)
Modelo1H
```

```
##
## Call:
## lm(formula = Peso ~ Estatura, data = df_hombres)
##
## Coefficients:
## (Intercept)      Estatura
##      -83.68         94.66
```

Modelo 2.

```
Modelo1M = lm(Peso ~ Estatura, data = df_mujeres)
Modelo1M
```

```
##
## Call:
## lm(formula = Peso ~ Estatura, data = df_mujeres)
##
## Coefficients:
## (Intercept)      Estatura
##      -72.56       81.15
```

Modelo 3.

```
ModeloInter = lm(Peso ~ Estatura * Sexo, data = df)
ModeloInter
```

```
##
## Call:
## lm(formula = Peso ~ Estatura * Sexo, data = df)
##
## Coefficients:
## (Intercept)      Estatura      SexoM Estatura:SexoM
##      -83.68       94.66       11.12      -13.51
```

Modelo 4.

```
ModelosinInter = lm(Peso ~ Estatura + Sexo, data = df)
ModelosinInter
```

```
##
## Call:
## lm(formula = Peso ~ Estatura + Sexo, data = df)
##
## Coefficients:
## (Intercept)      Estatura      SexoM
##      -74.75       89.26      -10.56
```

1. Analiza si el (los) modelo(s) obtenidos anteriormente son apropiados para el conjunto de

datos.

Realiza el análisis de los residuos:

1.1. Normalidad de los residuos

```
library(nortest)
ad.test(Modelo1H$residuals)
```

```
##
## Anderson-Darling normality test
##
## data:  Modelo1H$residuals
## A = 0.3009, p-value = 0.5771
```

```
ad.test(Modelo1M$residuals)
```

```
##
## Anderson-Darling normality test
##
## data:  Modelo1M$residuals
## A = 0.24899, p-value = 0.7451
```

```
ad.test(ModeloInter$residuals)
```

```
##
## Anderson-Darling normality test
##
## data:  ModeloInter$residuals
## A = 0.8138, p-value = 0.03516
```

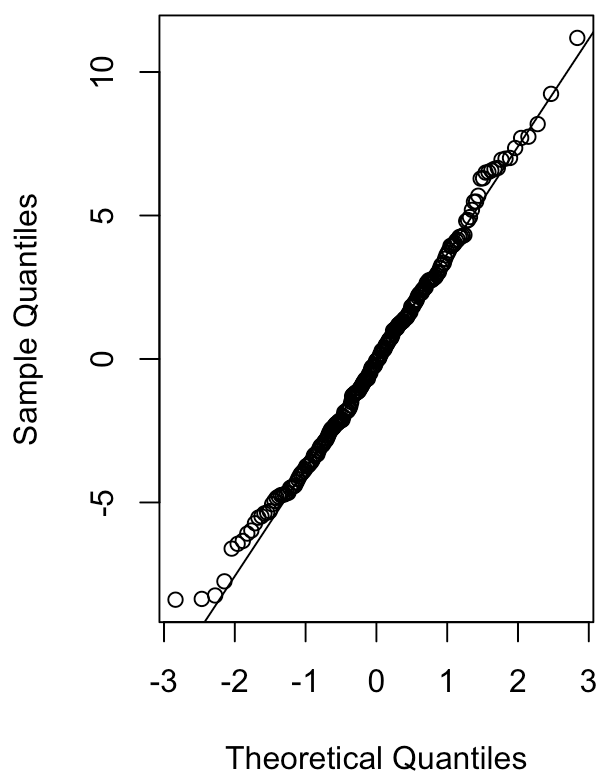
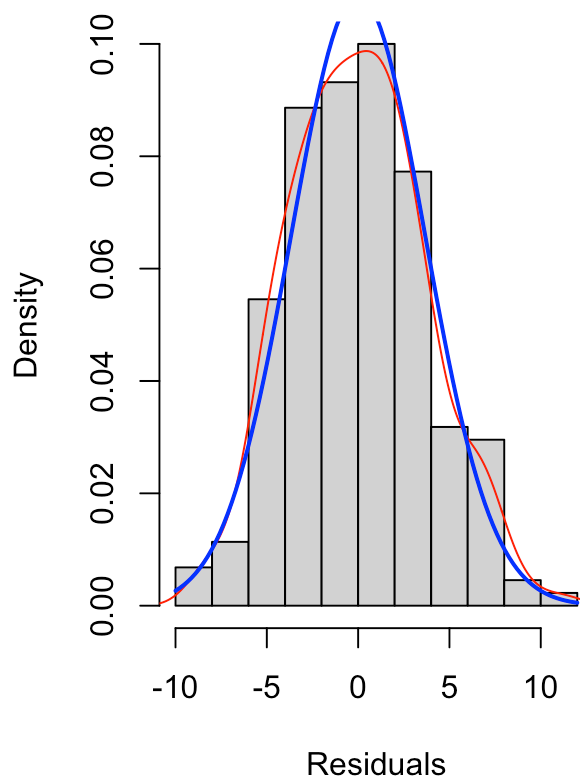
```
ad.test(ModelosinInter$residuals)
```

```
##
## Anderson-Darling normality test
##
## data:  ModelosinInter$residuals
## A = 0.79651, p-value = 0.03879
```

```

par(mfrow = c(1,2))
qqnorm(Modelo1H$residuals, main = "Q-Q Plot - Modelo1H")
qqline(Modelo1H$residuals)
hist(Modelo1H$residuals, freq = FALSE, main = "Modelo1H", xlab = "Residuals", ylab = "Density")
lines(density(Modelo1H$residuals), col = "red")
curve(dnorm(x, mean = mean(Modelo1H$residuals), sd=sd(Modelo1H$residuals)), add = TRUE,
col = "blue", lwd = 2)

```

Q-Q Plot - Modelo1H**Modelo1H**

```

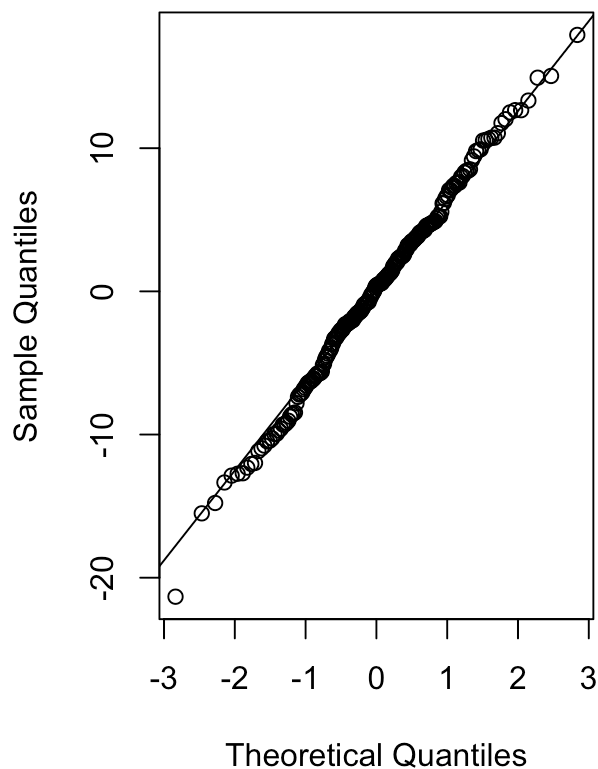
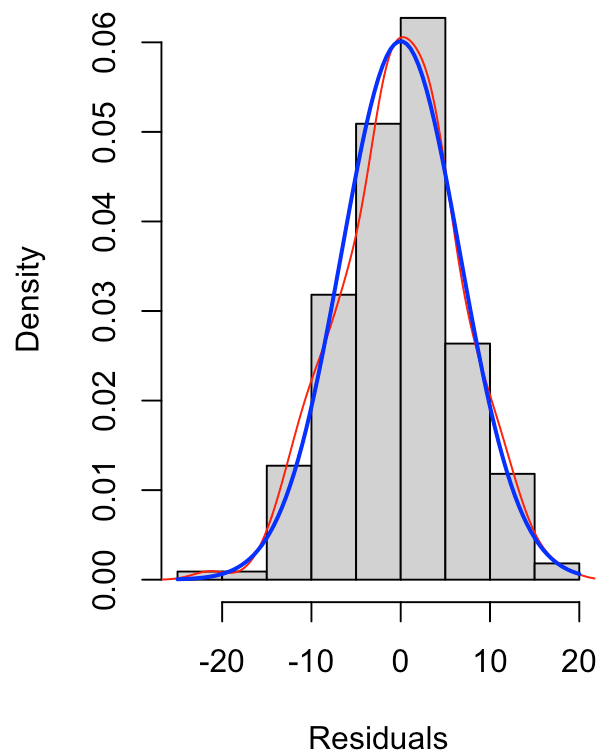
par(mfrow = c(1,1))

```

```

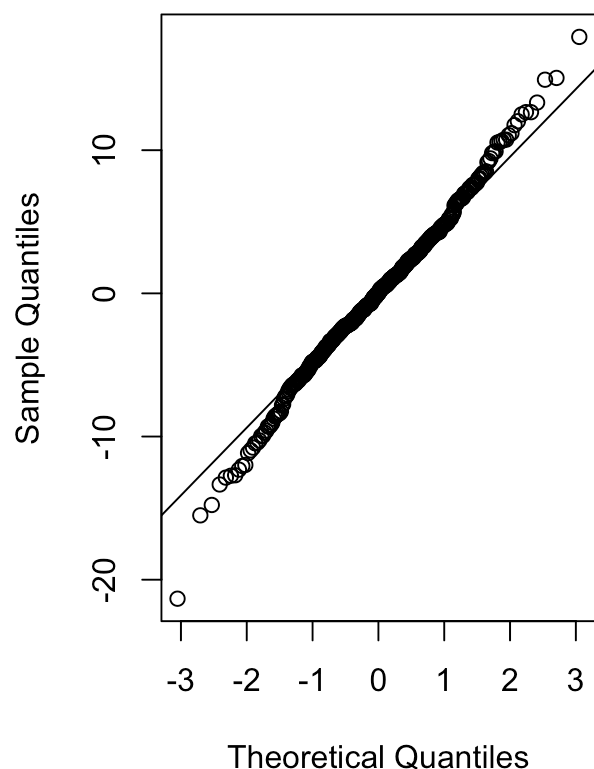
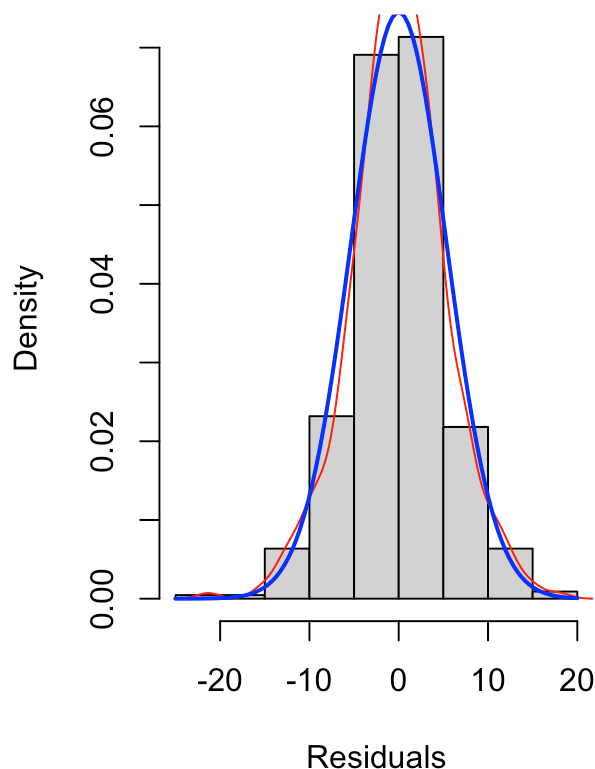
par(mfrow = c(1,2))
qqnorm(Modelo1M$residuals, main = "Q-Q Plot - Modelo1M")
qqline(Modelo1M$residuals)
hist(Modelo1M$residuals, freq = FALSE, main = "Modelo1M", xlab = "Residuals", ylab = "Density")
lines(density(Modelo1M$residuals), col = "red")
curve(dnorm(x, mean = mean(Modelo1H$residuals), sd = sd(Modelo1M$residuals)), add = TRUE,
col = "blue", lwd = 2)

```

Q-Q Plot - Modelo1M**Modelo1M**

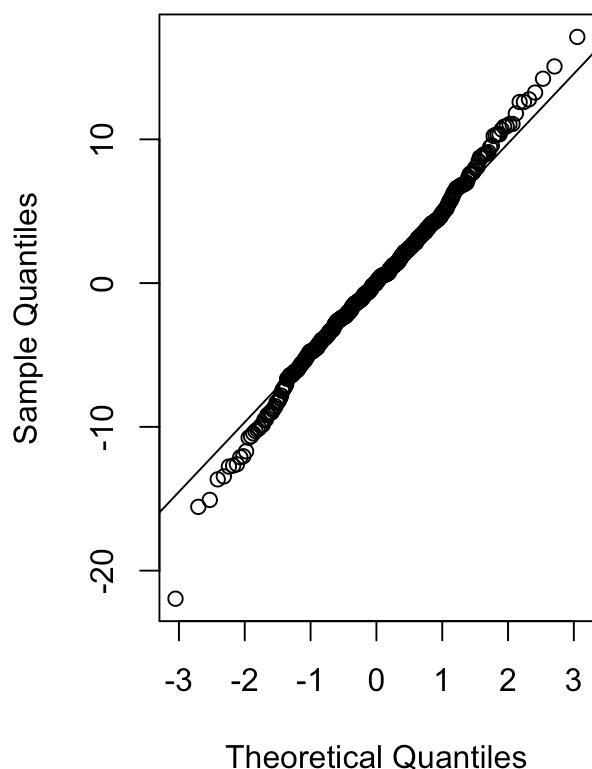
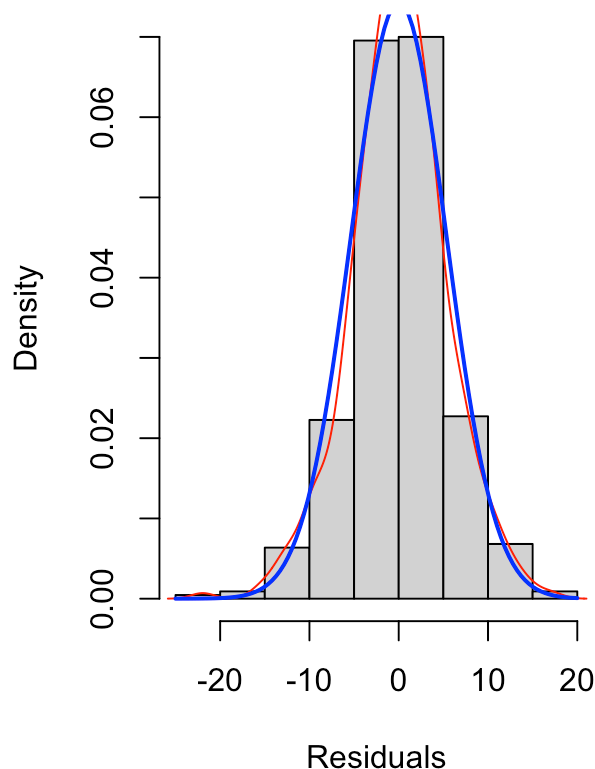
```
par(mfrow = c(1,1))
```

```
par(mfrow = c(1,2))
qqnorm(ModeloInter$residuals, main = "Q-Q Plot - ModeloInter")
qqline(ModeloInter$residuals)
hist(ModeloInter$residuals, freq = FALSE, main = "ModeloInter", xlab = "Residuals", ylab = "Density")
lines(density(ModeloInter$residuals), col = "red")
curve(dnorm(x, mean = mean(ModeloInter$residuals), sd = sd(ModeloInter$residuals)), add = TRUE, col = "blue", lwd = 2)
```

Q-Q Plot - ModeloInter**ModeloInter**

```
par(mfrow = c(1,1))
```

```
par(mfrow = c(1,2))
qqnorm(ModelosinInter$residuals, main = "Q-Q Plot - ModelosinInter")
qqline(ModelosinInter$residuals)
hist(ModelosinInter$residuals, freq = FALSE, main = "ModelosinInter", xlab = "Residuals", ylab = "Density")
lines(density(ModelosinInter$residuals), col = "red")
curve(dnorm(x, mean = mean(ModelosinInter$residuals), sd = sd(ModelosinInter$residuals)), add = TRUE, col = "blue", lwd = 2)
```

Q-Q Plot - ModelosinInter**ModelosinInter**

```
par(mfrow = c(1,1))
```

1.2. Verificación de media cero

```
t.test(Modelo1H$residuals)
```

```
##
## One Sample t-test
##
## data:  Modelo1H$residuals
## t = 1.4929e-15, df = 219, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.4876507  0.4876507
## sample estimates:
## mean of x
## 3.694015e-16
```

```
t.test(Modelo1M$residuals)
```

```
##
## One Sample t-test
##
## data: Modelo1M$residuals
## t = 9.3862e-16, df = 219, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.881609 0.881609
## sample estimates:
## mean of x
## 4.198662e-16
```

```
t.test(ModeloInter$residuals)
```

```
##
## One Sample t-test
##
## data: ModeloInter$residuals
## t = 6.3252e-17, df = 439, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.5017741 0.5017741
## sample estimates:
## mean of x
## 1.61487e-17
```

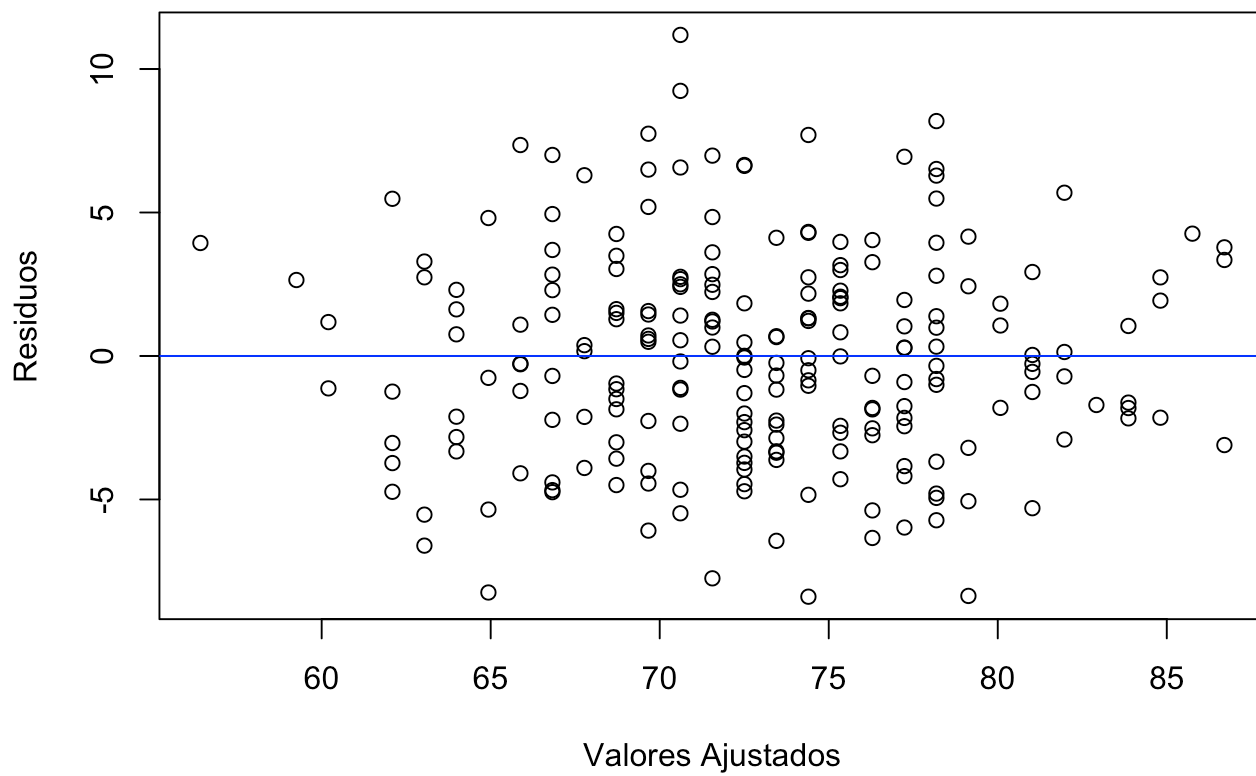
```
t.test(ModelosinInter$residuals)
```

```
##
## One Sample t-test
##
## data: ModelosinInter$residuals
## t = -5.048e-16, df = 439, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -0.5029859 0.5029859
## sample estimates:
## mean of x
## -1.291896e-16
```

1.3. Homocedasticidad e independencia

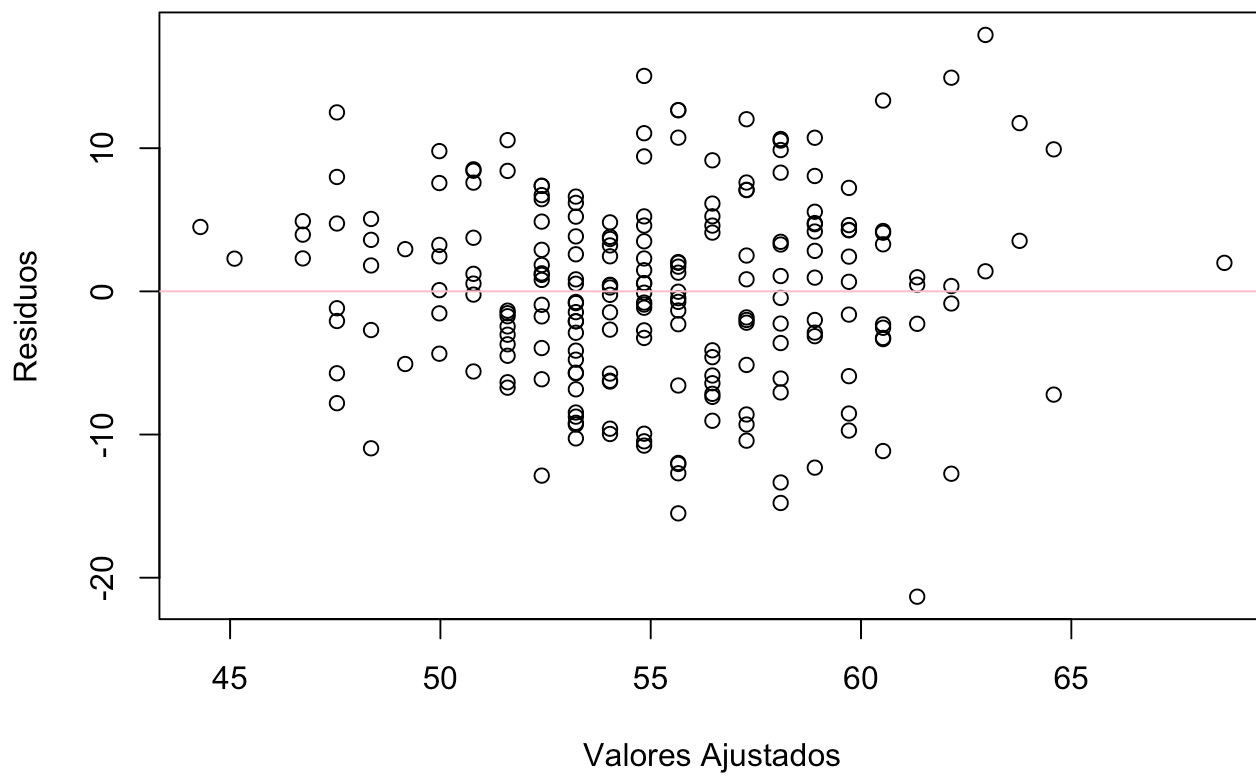
```
plot(Modelo1H$fitted.values, Modelo1H$residuals, main = "Modelo1H", xlab = "Valores Ajustados", ylab = "Residuos")
abline(h = 0, col = "blue")
```


Modelo1H



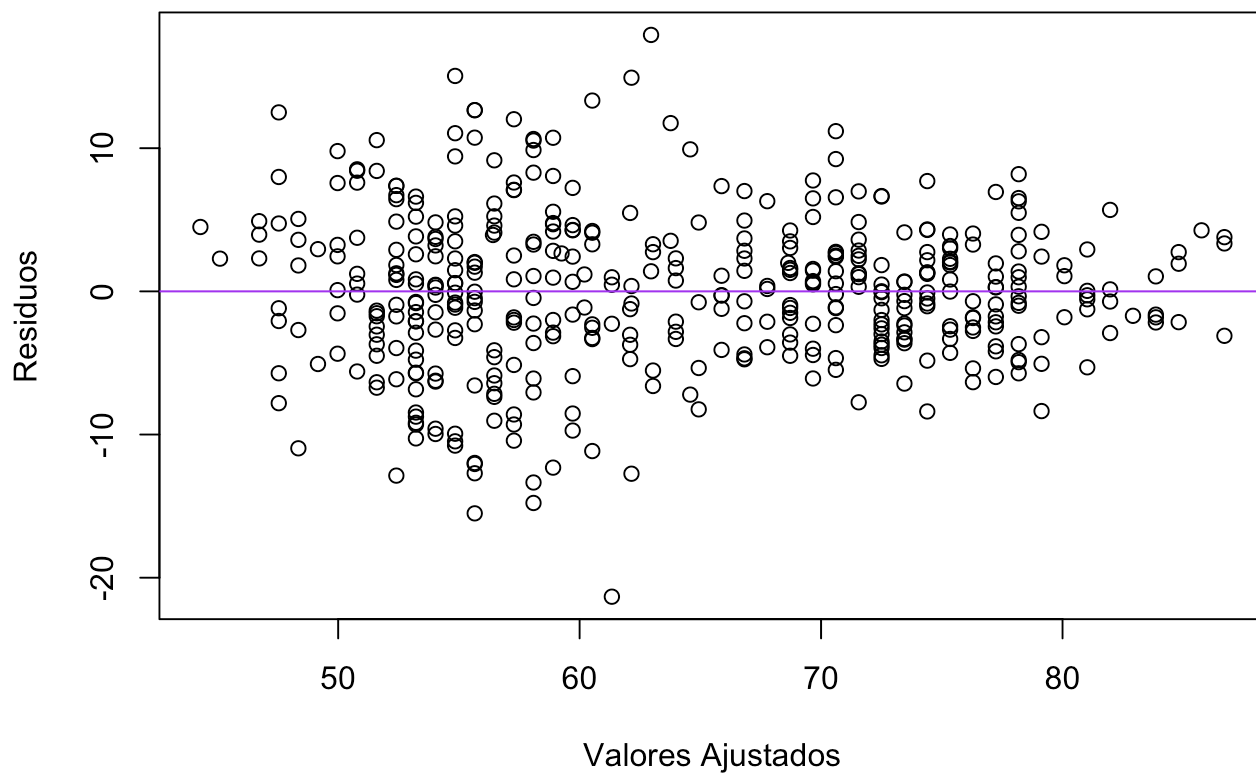
```
plot(Modelo1M$fitted.values, Modelo1M$residuals, main = "Modelo1M", xlab = "Valores Ajustados", ylab = "Residuos")  
abline(h = 0, col = "pink")
```

Modelo1M



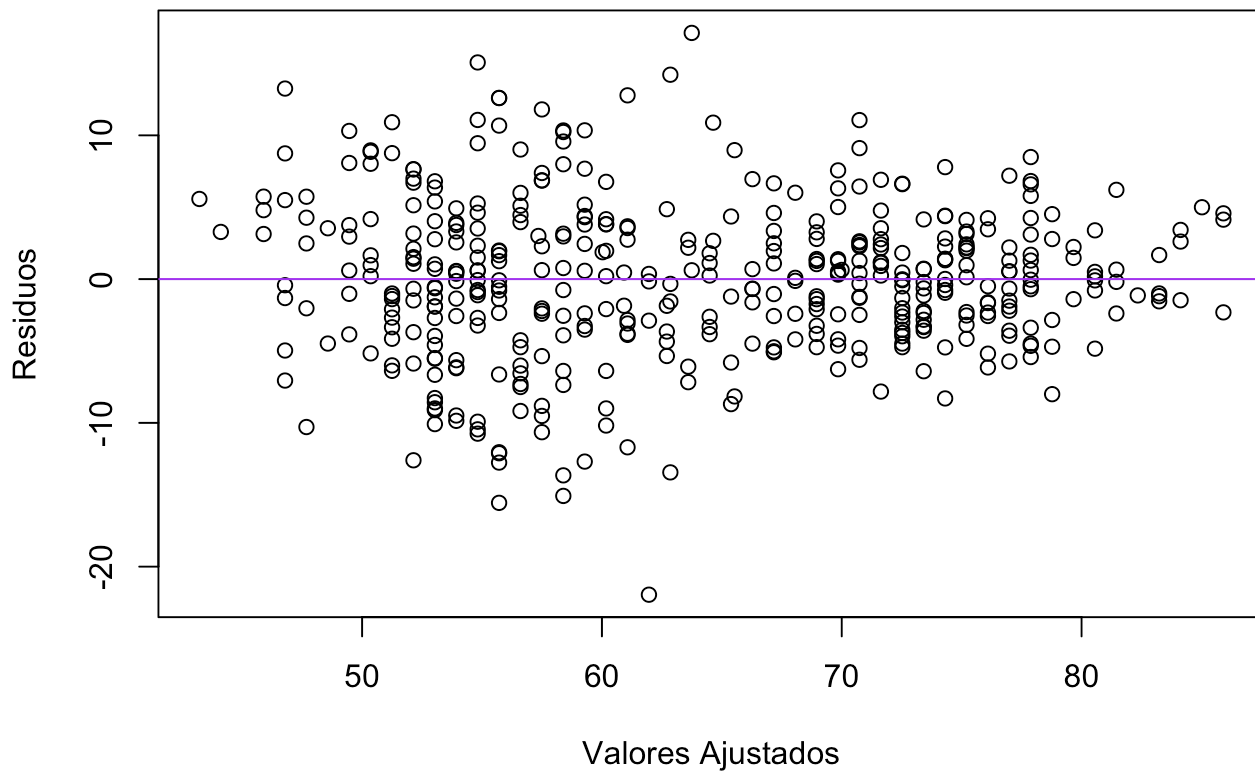
```
plot(ModeloInter$fitted.values, ModeloInter$residuals, main = "ModeloInter", xlab = "Valores Ajustados", ylab = "Residuos")
abline(h = 0, col = "purple")
```

ModeloInter



```
plot(ModelosinInter$fitted.values, ModelosinInter$residuals, main = "ModelosinInter", xlab = "Valores Ajustados", ylab = "Residuos")  
abline(h = 0, col = "purple")
```

ModelosinInter



2. No te olvides de incluir las hipótesis en la pruebas de hipótesis que realices.

Test de Durbin-Watson y Prueba de Breusch-Godfrey

- Hipótesis Nula (H_0): Los errores no están autocorrelacionados.
- Hipótesis Alternativa (H_1): Los errores están autocorrelacionados.

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
dw_test_hombres = dwtest(Modelo1H)
bg_test_hombres = bgtest(Modelo1H)
print("Resultados para Modelo1H:")
```

```
## [1] "Resultados para Modelo1H:"
```

```
print(dw_test_hombres)
```

```
##
## Durbin-Watson test
##
## data:  Modelo1H
## DW = 2.0556, p-value = 0.6599
## alternative hypothesis: true autocorrelation is greater than 0
```

```
print(bg_test_hombres)
```

```
##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  Modelo1H
## LM test = 0.20778, df = 1, p-value = 0.6485
```

```
dw_test_mujeres = dwtest(Modelo1M)
bg_test_mujeres = bgtest(Modelo1M)
print("Resultados para Modelo1M:")
```

```
## [1] "Resultados para Modelo1M:"
```

```
print(dw_test_mujeres)
```

```
##
## Durbin-Watson test
##
## data:  Modelo1M
## DW = 1.8062, p-value = 0.07532
## alternative hypothesis: true autocorrelation is greater than 0
```

```
print(bg_test_mujeres)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: Modelo1M  
## LM test = 1.4655, df = 1, p-value = 0.2261
```

```
dw_test_inter = dwtest(ModeloInter)  
bg_test_inter = bgtest(ModeloInter)  
print("Resultados para ModeloInter:")
```

```
## [1] "Resultados para ModeloInter:"
```

```
print(dw_test_inter)
```

```
##  
## Durbin-Watson test  
##  
## data: ModeloInter  
## DW = 1.8646, p-value = 0.07113  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
print(bg_test_inter)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: ModeloInter  
## LM test = 1.3453, df = 1, p-value = 0.2461
```

```
dw_test_inter = dwtest(ModelosinInter)  
bg_test_inter = bgtest(ModelosinInter)  
print("Resultados para ModelosinInter:")
```

```
## [1] "Resultados para ModelosinInter:"
```

```
print(dw_test_inter)
```

```
##  
## Durbin-Watson test  
##  
## data: ModelosinInter  
## DW = 1.8663, p-value = 0.07325  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
print(bg_test_inter)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: ModelosinInter  
## LM test = 1.3595, df = 1, p-value = 0.2436
```

Prueba de Breusch-Pagan y White

- H_0 : La varianza de los errores es constante (homocedasticidad)
- H_1 : La varianza de los errores no es constante (heterocedasticidad)

```
bp_test_hombres = bptest(Modelo1H)  
white_test_hombres = qqtest(Modelo1H)  
print("Resultados para Modelo1H:")
```

```
## [1] "Resultados para Modelo1H:"
```

```
print(bp_test_hombres)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: Modelo1H  
## BP = 0.93324, df = 1, p-value = 0.334
```

```
print(white_test_hombres)
```

```
##  
## Goldfeld-Quandt test  
##  
## data: Modelo1H  
## GQ = 0.84148, df1 = 108, df2 = 108, p-value = 0.8144  
## alternative hypothesis: variance increases from segment 1 to 2
```

```
bp_test_mujeres = bptest(Modelo1M)  
white_test_mujeres = qqtest(Modelo1M)  
print("Resultados para Modelo1M:")
```

```
## [1] "Resultados para Modelo1M:"
```

```
print(bp_test_mujeres)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: Modelo1M  
## BP = 8.4976, df = 1, p-value = 0.003556
```

```
print(white_test_mujeres)
```

```
##  
## Goldfeld-Quandt test  
##  
## data: Modelo1M  
## GQ = 1.4265, df1 = 108, df2 = 108, p-value = 0.03313  
## alternative hypothesis: variance increases from segment 1 to 2
```

```
bp_test_inter = bptest(ModeloInter)  
white_test_inter = gqtest(ModeloInter)  
print("Resultados para ModeloInter:")
```

```
## [1] "Resultados para ModeloInter:"
```

```
print(bp_test_inter)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: ModeloInter  
## BP = 59.211, df = 3, p-value = 8.667e-13
```

```
print(white_test_inter)
```

```
##  
## Goldfeld-Quandt test  
##  
## data: ModeloInter  
## GQ = 3.2684, df1 = 216, df2 = 216, p-value < 2.2e-16  
## alternative hypothesis: variance increases from segment 1 to 2
```

```
bp_test_inter = bptest(ModelosinInter)  
white_test_inter = gqtest(ModelosinInter)  
print("Resultados para ModelosinInter:")
```

```
## [1] "Resultados para ModelosinInter:"
```



```
print(bp_test_inter)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: ModelosinInter  
## BP = 48.202, df = 2, p-value = 3.413e-11
```

```
print(white_test_inter)
```

```
##  
## Goldfeld-Quandt test  
##  
## data: ModelosinInter  
## GQ = 3.2684, df1 = 217, df2 = 217, p-value < 2.2e-16  
## alternative hypothesis: variance increases from segment 1 to 2
```

Intervalos de Confianza

1. Con los datos de las estaturas y pesos de los hombres y las mujeres construye la gráfica de los intervalos de confianza y predicción para la estimación y predicción de Y para el mejor modelo seleccionado.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
# Volvemos a traer nuestras variables para evitar errores de procesamiento.
```

```
df_hombres = subset(df, df$Sexo == "H")  
df_mujeres = subset(df, df$Sexo == "M")
```

```
Modelo1H = lm(Peso ~ Estatura, data = df_hombres)  
Modelo1M = lm(Peso ~ Estatura, data = df_mujeres)
```

```
prediccion_hombres = predict(Modelo1H, interval = "prediction", level = 0.97)
```

```
## Warning in predict.lm(Modelo1H, interval = "prediction", level = 0.97): predictions on current data refer to _future_ responses
```

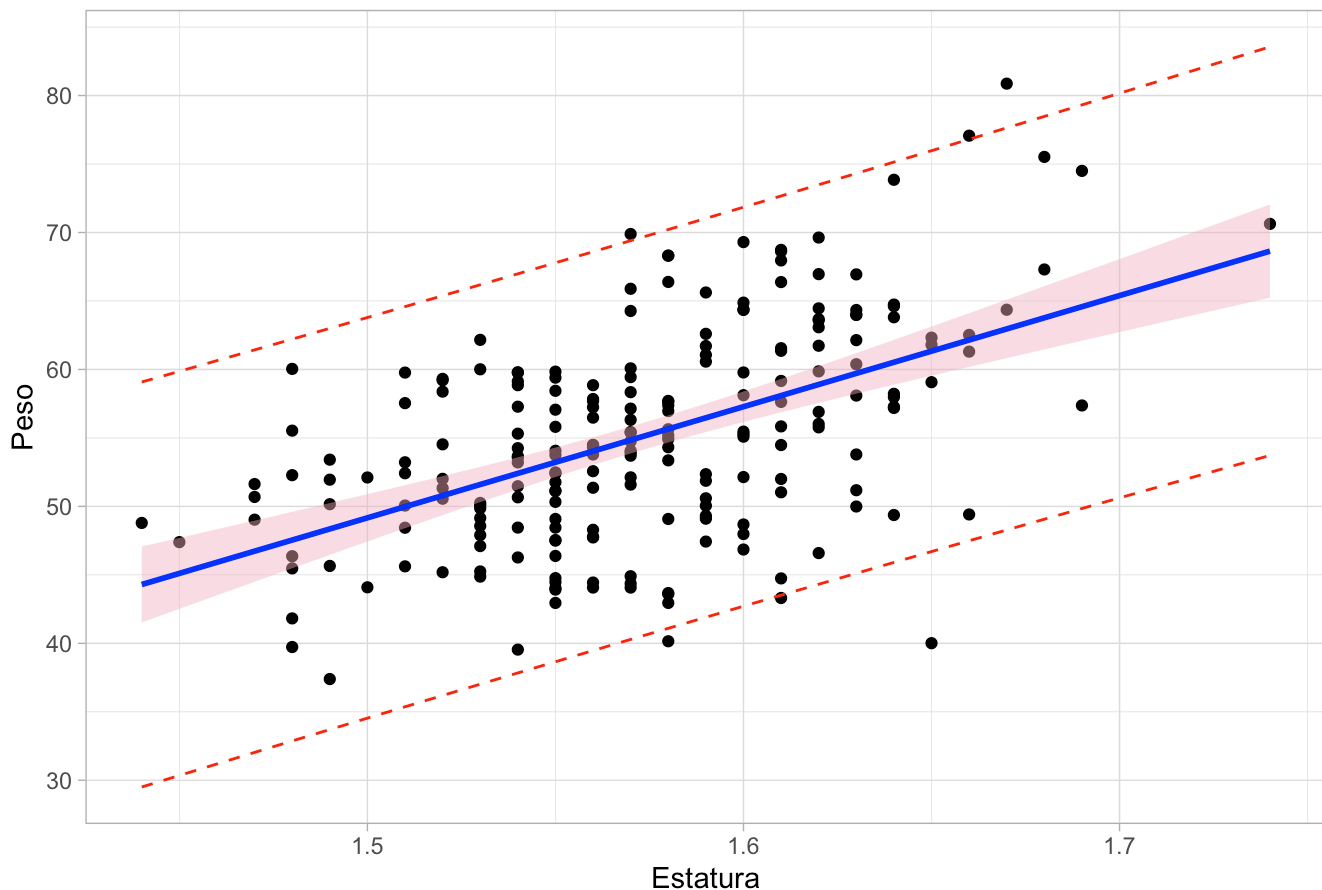
```
prediccion_mujeres = predict(Modelo1M, interval = "prediction", level = 0.97)
```

```
## Warning in predict.lm(Modelo1M, interval = "prediction", level = 0.97): predictions on current data refer to _future_ responses
```

```
df_hombres = cbind(df_hombres, prediccion_hombres)
df_mujeres = cbind(df_mujeres, prediccion_mujeres)
```

```
ggplot(df_mujeres, aes(x = Estatura, y = Peso)) + geom_point() + geom_line(aes(y = lwr), color = "red", linetype = "dashed") + geom_line(aes(y = upr), color = "red", linetype = "dashed") + geom_smooth(method = "lm", formula = y ~ x, se = TRUE, level = 0.97, col = "blue", fill = "pink2") + theme_light() + labs(title = "Relación Peso y en Estatura Mujeres", x = "Estatura", y = "Peso")
```

Relación Peso y en Estatura Mujeres



```
ggplot(df_hombres, aes(x = Estatura, y = Peso)) + geom_point() + geom_line(aes(y = lwr), color = "red", linetype = "dashed") + geom_line(aes(y = upr), color = "red", linetype = "dashed") + geom_smooth(method = "lm", formula = y ~ x, se = TRUE, level = 0.97, col = "blue", fill = "pink2") + theme_light() + labs(title = "Relación Peso y Estatura en Hombres", x = "Estatura", y = "Peso")
```

Relación Peso y Estatura en Hombres

