

# Análisis de Felicidad

Jaime Riquelme

2024-11-28

## Contruccion de los modelos

```
# Modelo 1: Solo con 'Edad'
modelo_edad <- lm(Felicidad ~ Edad, data = muestra)
cat("\n\n")

cat("Modelo con solo 'Edad' --> 'Felicidad'\n")

## Modelo con solo 'Edad' --> 'Felicidad'

cat("-----\n")

## -----

print(summary(modelo_edad))

##
## Call:
## lm(formula = Felicidad ~ Edad, data = muestra)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.8764 -1.0600 -0.0528  0.9327  4.2854
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.5981     2.4577   3.498  0.00083 ***
## Edad        -0.1618     0.0964  -1.679  0.09781 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.644 on 68 degrees of freedom
## Multiple R-squared:  0.03979,    Adjusted R-squared:  0.02567
## F-statistic: 2.818 on 1 and 68 DF,  p-value: 0.09781

# Modelo 2: Con 'Edad' + 'N_amigos'
modelo_amigos <- lm(Felicidad ~ Edad + N_amigos, data = muestra)
cat("\n\n")
```

```
cat("Modelo con 'Edad' + 'N_amigos' --> 'Felicidad'\n")
```

```
## Modelo con 'Edad' + 'N_amigos' --> 'Felicidad'
```

```
cat("-----\n")
```

```
## -----
```

```
print(summary(modelo_amigos))
```

```
##
## Call:
## lm(formula = Felicidad ~ Edad + N_amigos, data = muestra)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.5002 -1.0367 -0.0324  0.9314  3.7960
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.34625    2.53306   2.505  0.0147 *
## Edad        -0.12762    0.09387  -1.360  0.1785
## N_amigos     0.19207    0.07676   2.502  0.0148 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.584 on 67 degrees of freedom
## Multiple R-squared:  0.1219, Adjusted R-squared:  0.09564
## F-statistic: 4.649 on 2 and 67 DF,  p-value: 0.01287
```

```
# Modelo 3: Con 'Edad' + 'N_amigos' + 'N_mascotas'
modelo_completo <- lm(Felicidad ~ Edad + N_amigos + N_mascotas, data = muestra)
cat("\n\n")
```

```
cat("Modelo con 'Edad' + 'N_amigos' + 'N_mascotas' --> 'Felicidad'\n")
```

```
## Modelo con 'Edad' + 'N_amigos' + 'N_mascotas' --> 'Felicidad'
```

```
cat("-----\n")
```

```
## -----
```

```
print(summary(modelo_completo))
```

```
##
## Call:
## lm(formula = Felicidad ~ Edad + N_amigos + N_mascotas, data = muestra)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -3.0185 -1.1831 -0.0803  0.9836  3.5281
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.82036    2.41755   2.408  0.01886 *
## Edad        -0.11626    0.08941  -1.300  0.19803
## N_amigos     0.15554    0.07417   2.097  0.03983 *
## N_mascotas   0.44321    0.15676   2.827  0.00621 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.507 on 66 degrees of freedom
## Multiple R-squared:  0.2167, Adjusted R-squared:  0.1811
## F-statistic: 6.087 on 3 and 66 DF,  p-value: 0.001012
```

```
# Comparar los modelos para ver cuál tiene el mejor ajuste
cat("\n\n")
```

```
cat("Comparación de modelos jerárquicos\n")
```

```
## Comparación de modelos jerárquicos
```

```
cat("-----\n")
```

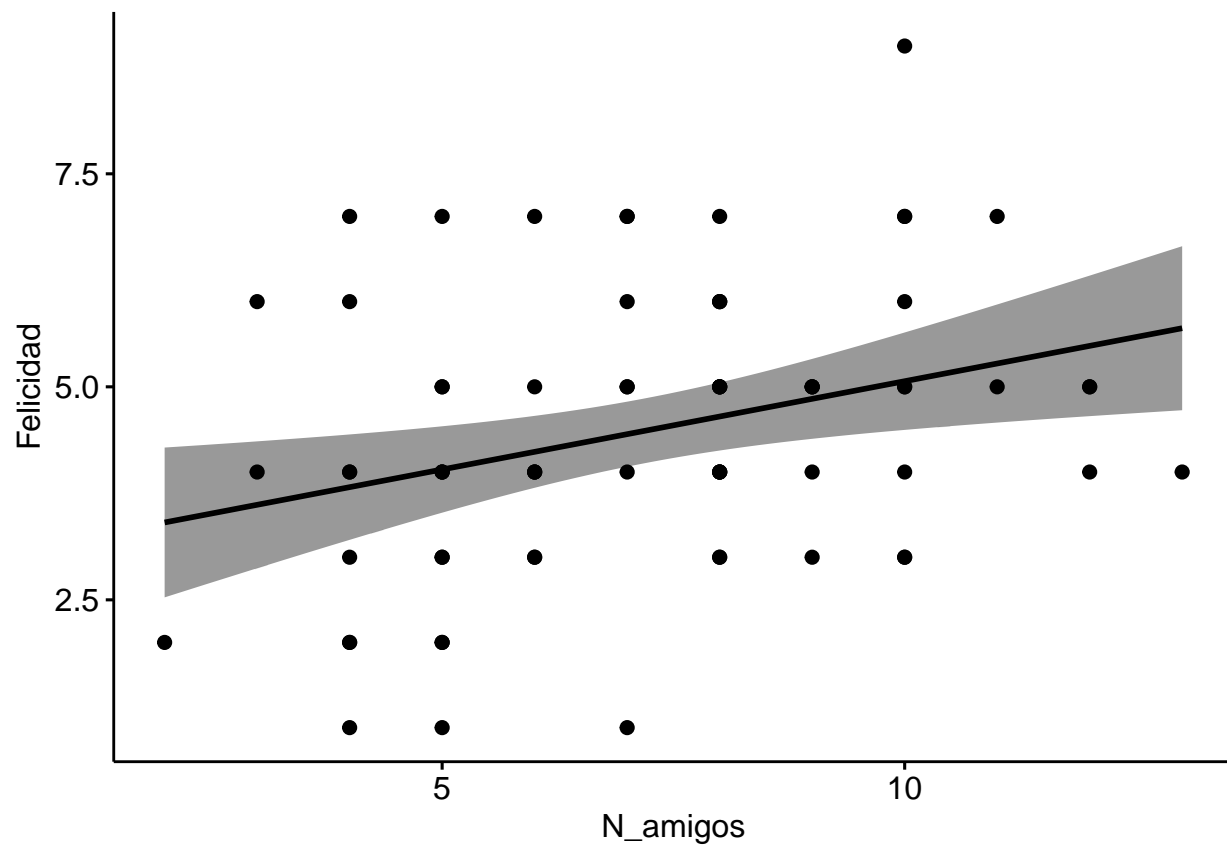
```
## -----
```

```
print(anova(modelo_edad, modelo_amigos, modelo_completo))
```

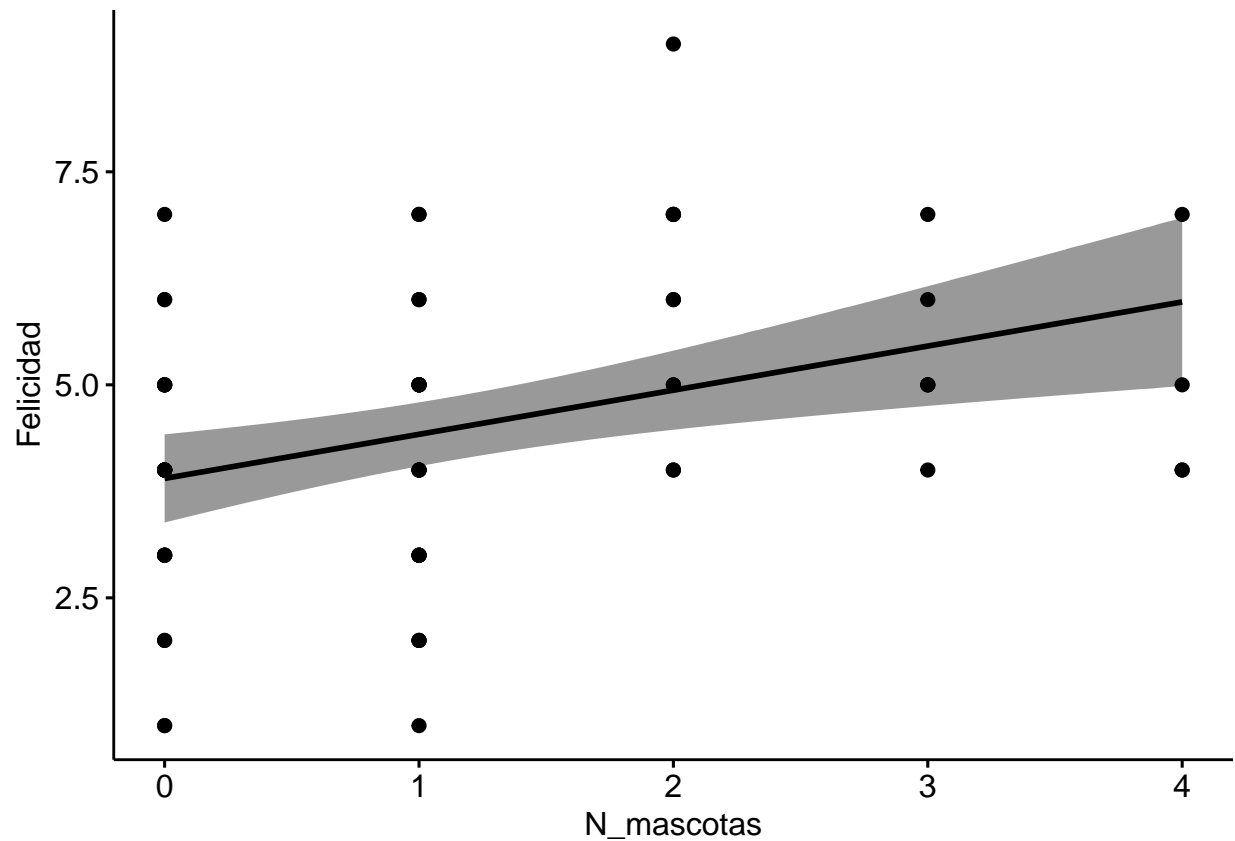
```
## Analysis of Variance Table
##
## Model 1: Felicidad ~ Edad
## Model 2: Felicidad ~ Edad + N_amigos
## Model 3: Felicidad ~ Edad + N_amigos + N_mascotas
##   Res.Df    RSS Df Sum of Sq    F  Pr(>F)
## 1      68 183.87
## 2      67 168.15  1    15.715 6.9152 0.010624 *
## 3      66 149.99  1    18.166 7.9938 0.006207 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Evaluación de los modelos

```
# Gráficos de dispersión y líneas de regresión
g_felicidad_amigos <- ggscatter(muestra, x = "N_amigos", y = "Felicidad",
                               add = "reg.line", conf.int = TRUE)
print(g_felicidad_amigos)
```



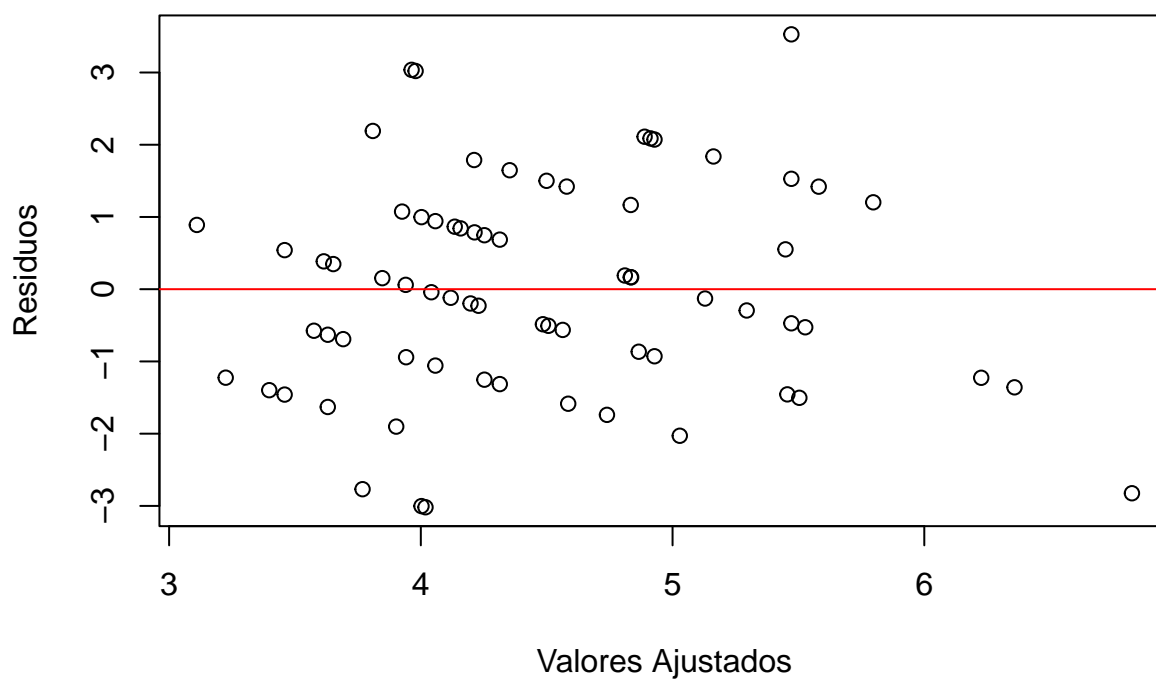
```
g_felicidad_mascotas <- ggscatter(muestra, x = "N_mascotas", y = "Felicidad",  
                                  add = "reg.line", conf.int = TRUE)  
print(g_felicidad_mascotas)
```



## Condicion de homocedasticidad

```
# Gráfico de residuos vs. valores ajustados
plot(modelo_completo$fitted.values, modelo_completo$residuals,
     xlab = "Valores Ajustados", ylab = "Residuos", main = "Residuos vs. Valores Ajustados")
abline(h = 0, col = "red")
```

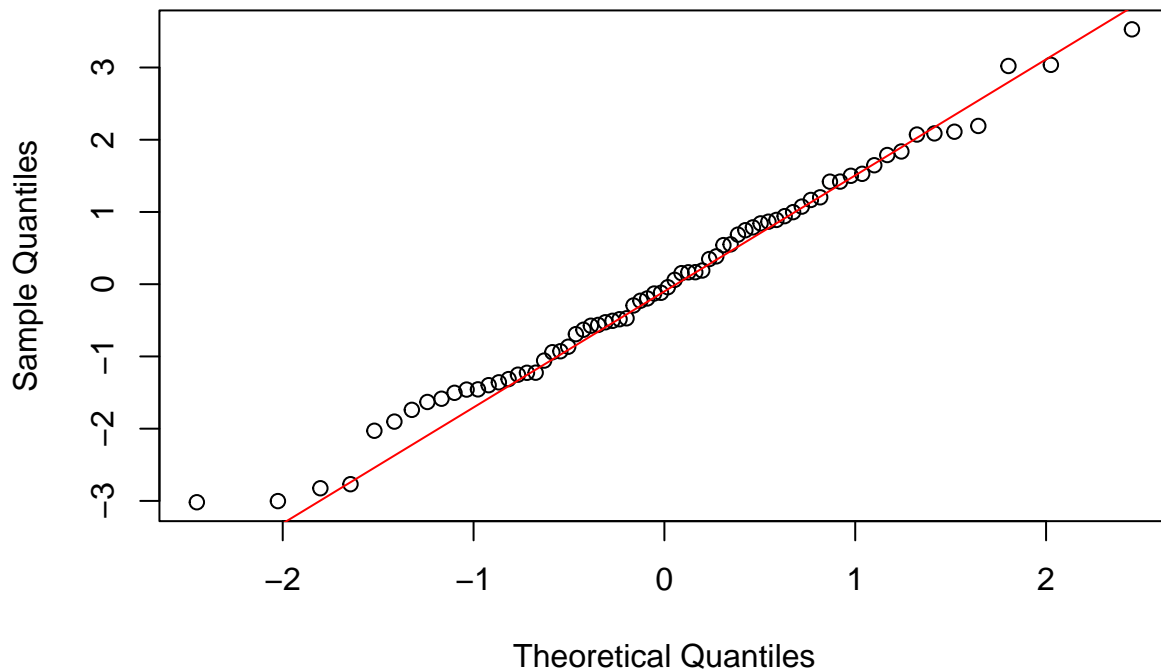
## Residuos vs. Valores Ajustados



## Condicion de normalidad de los errores

```
# Gráfico Q-Q de los residuos  
qqnorm(modelo_completo$residuals)  
qqline(modelo_completo$residuals, col = "red")
```

## Normal Q-Q Plot



## Evaluacion del poder predictivo del modelo

```
# Calcular el error cuadrático medio (RMSE) y  $R^2$ 
rmse <- sqrt(mean(modelo_completo$residuals^2))
cat("\nRMSE del modelo completo:", rmse, "\n")
```

```
##
## RMSE del modelo completo: 1.463781
```

```
#  $R^2$  (coeficiente de determinación)
r2 <- summary(modelo_completo)$r.squared
cat("R2 del modelo completo:", r2, "\n")
```

```
## R2 del modelo completo: 0.2167259
```

```
# Comparar las predicciones del modelo con los valores reales
predicciones <- predict(modelo_completo, muestra)
comparacion <- data.frame(Real = muestra$Felicidad, Predicho = predicciones)
print(comparacion)
```

```
##      Real Predicho
```

## 74	3 3.575235
## 50	5 5.129048
## 46	7 4.889003
## 13	2 3.630388
## 76	4 3.614511
## 10	4 5.503632
## 59	5 4.134712
## 100	4 4.229141
## 29	3 3.630388
## 60	2 3.224884
## 51	4 3.847035
## 68	6 4.499375
## 48	2 3.397864
## 49	2 3.902188
## 67	7 4.912401
## 26	7 5.471878
## 87	3 4.585448
## 84	5 5.471878
## 81	5 4.213264
## 12	3 4.313648
## 35	1 4.018450
## 33	5 5.294507
## 5	3 4.739421
## 41	5 4.158111
## 25	5 5.527031
## 64	3 5.028663
## 1	5 6.226168
## 65	3 4.057726
## 45	5 3.925587
## 11	7 3.979174
## 9	4 4.485063
## 99	7 5.797265
## 98	5 4.002573
## 86	4 3.110187
## 21	4 4.865604
## 91	3 4.252539
## 47	1 3.768483
## 66	7 5.162368
## 89	3 3.691497
## 6	5 4.313648
## 32	1 4.002573
## 94	4 3.458973
## 36	4 4.563615
## 57	6 4.579492
## 27	5 6.358308
## 17	9 5.471878
## 42	5 4.057726
## 14	7 5.580619
## 40	6 4.833849
## 88	2 3.458973
## 16	7 4.928278
## 90	4 4.041849
## 58	4 5.456000
## 97	6 4.352924



## 72	4 4.928278
## 2	5 4.252539
## 19	4 6.824921
## 63	7 3.963297
## 18	3 3.941464
## 52	4 3.652221
## 56	4 4.506897
## 71	6 3.809325
## 37	4 3.939899
## 55	4 4.197386
## 93	5 4.810451
## 82	5 4.835415
## 34	5 4.833849
## 85	6 4.211698
## 20	6 5.448479
## 31	4 4.118835