

Proyecto 3 – World Happiness

DESCRIPCIÓN DEL PROBLEMA



En este problema se nos proporciona un informe sobre la felicidad en el mundo. Se proporciona un ranking de distintos países cuya puntuación esta basada en función de 6 **factores que contribuyen a la felicidad**. Estos son:

- PIB per cápita
- Apoyo social
- Esperanza de vida saludable
- Libertad de elección de vida
- Generosidad
- Percepciones de corrupción

Nuestra tarea consistirá en:

- Hacer 7 modelos de regresión línea. 6 de ellos serán de los factores que contribuyen a la felicidad en relación a la puntuación. Otro modelo consistirá en un modelo de todos los factores en relación a la puntuación.
- El modelo estará dividido en un 80% de ejemplos de entrenamiento y un 20% para hacer test.

ANALISIS DE LOS RESULTADOS

Diferencia entre el modelo y el valor real

```
> #imprime valor
> print(paste0("- Mean average error Todos: ", mean_avg_error_Todos))
[1] "- Mean average error Todos: 0.131494138484699"
> print(paste0("- Mean average error GPD: ", mean_avg_error_GPD))
[1] "- Mean average error GPD: 0.5445638746733"
> print(paste0("- Mean average error Social: ", mean_avg_error_Social))
[1] "- Mean average error Social: 0.559918528271534"
> print(paste0("- Mean average error Healthy: ", mean_avg_error_Healthy))
[1] "- Mean average error Healthy: 0.596164073330331"
> print(paste0("- Mean average error Freedom: ", mean_avg_error_Freedom))
[1] "- Mean average error Freedom: 0.723194067536709"
> print(paste0("- Mean average error Generosity: ", mean_avg_error_Generosity))
[1] "- Mean average error Generosity: 0.880158896696669"
> print(paste0("- Mean average error Corruption: ", mean_avg_error_Corruption))
[1] "- Mean average error Corruption: 0.769818812325635"
```

Resumen y graficos de los modelos

- **Modelo de todos los atributos:**

Call:

```
lm(formula = Score ~ ., data = training_data)
```

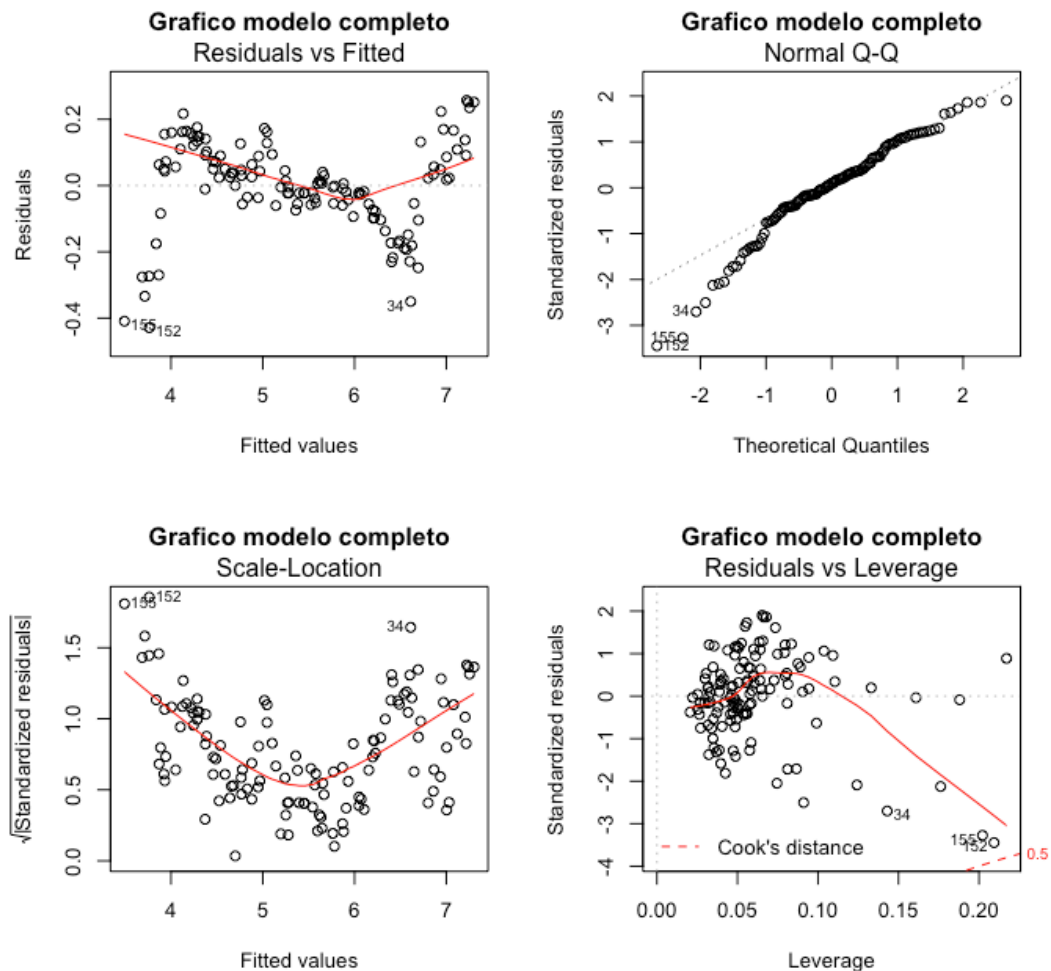
Residuals:

	Min	1Q	Median	3Q	Max
	-0.56672	-0.08756	0.02440	0.10620	0.36926

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.8345631	0.1619789	42.194	< 2e-16 ***
Overall.rank	-0.0231567	0.0006611	-35.030	< 2e-16 ***
GDP.per.capita	-0.0592458	0.0735803	-0.805	0.42231
Social.support	0.2486544	0.0783980	3.172	0.00192 **
Healthy.life.expectancy	-0.0170870	0.1131975	-0.151	0.88027
Freedom.to.make.life.choices	0.1579593	0.1305039	1.210	0.22851
Generosity	0.1853941	0.1595075	1.162	0.24742
Perceptions.of.corruption	0.4260617	0.1743176	2.444	0.01597 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



- **Modelo PIB per cápita:**

Residual standard error: 0.1586 on 120 degrees of freedom
Multiple R-squared: 0.9817, Adjusted R-squared: 0.9807
F-statistic: 921.1 on 7 and 120 DF, p-value: < 2.2e-16

> summary(model_GPD) #Devuelve valores de la regresion lineal para determinar la calidad

Call:

lm(formula = Score ~ GDP.per.capita, data = training_data)

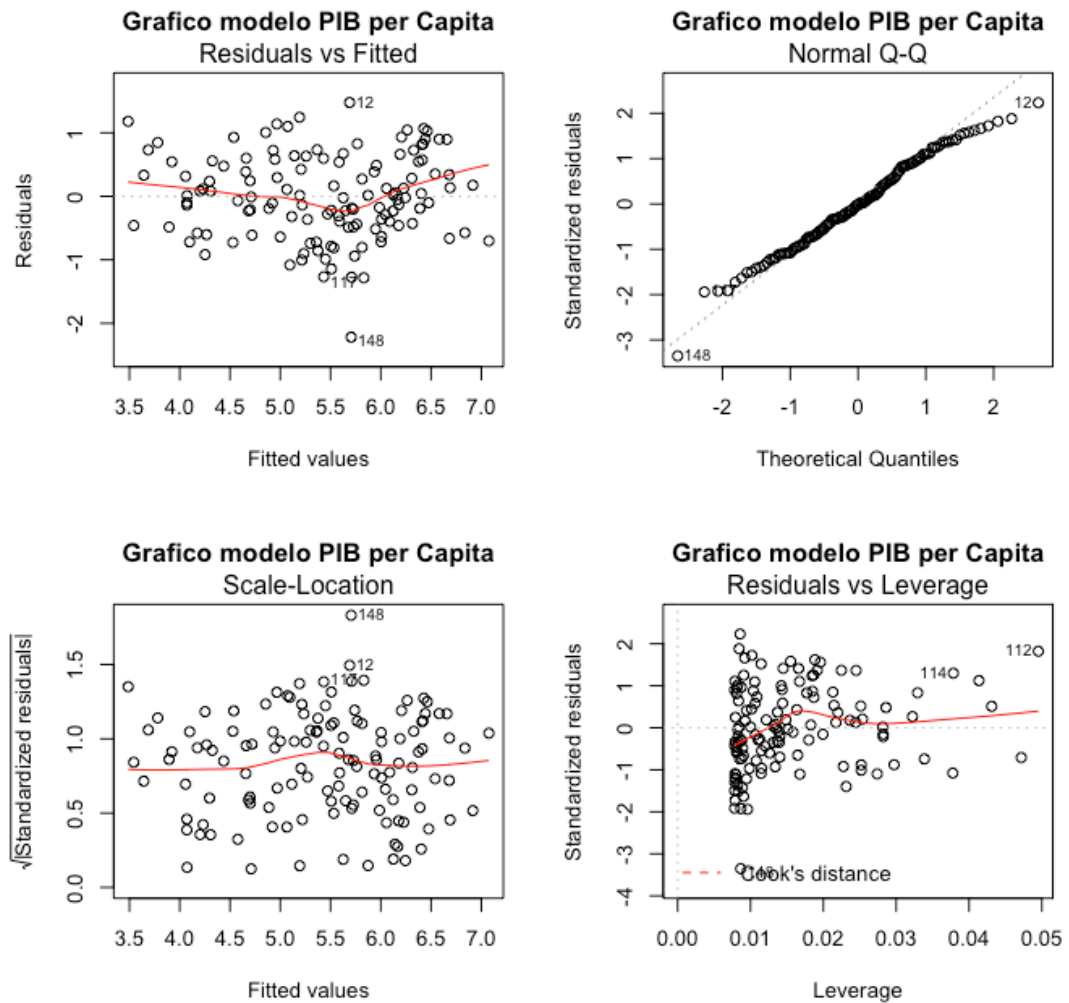
Residuals:

	Min	1Q	Median	3Q	Max
	-2.21607	-0.48622	0.00239	0.49052	1.39043

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.3557	0.1454	23.07	<2e-16 ***
GDP.per.capita	2.2559	0.1475	15.29	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



- **Modelo Apoyo Social:**

```
[1] "----- Summary Social -----"
```

```
> summary(model_Social) #Devuelve valores de la regresion lineal para determinar la calidad
```

Call:

```
lm(formula = Score ~ Social.support, data = training_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-1.86574	-0.48711	-0.02689	0.54614	1.72354

Coefficients:

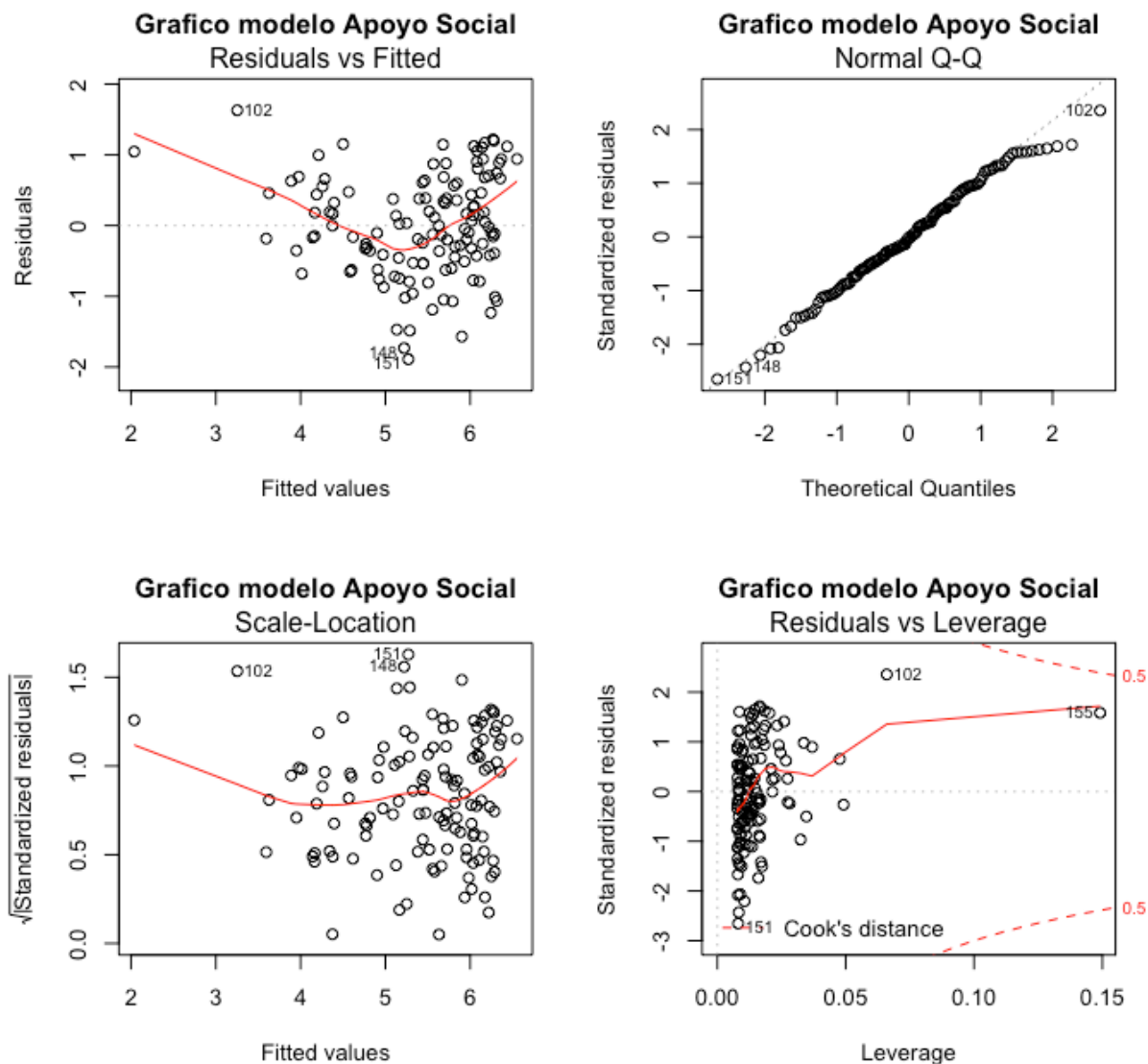
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.9037	0.2554	7.453	1.28e-11 ***
Social.support	2.8737	0.2044	14.057	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7144 on 126 degrees of freedom

Multiple R-squared: 0.6106, Adjusted R-squared: 0.6076

F-statistic: 197.6 on 1 and 126 DF, p-value: < 2.2e-16



- **Modelo Esperanza de vida saludable:**

```
[1] "----- Summary Healthy -----"
```

```
> summary(model_Healthy) #Devuelve valores de la regresion lineal para determinar la calidad
```

Call:

```
lm(formula = Score ~ Healthy.life.expectancy, data = training_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-1.64462	-0.52933	0.08379	0.48241	1.66245

Coefficients:

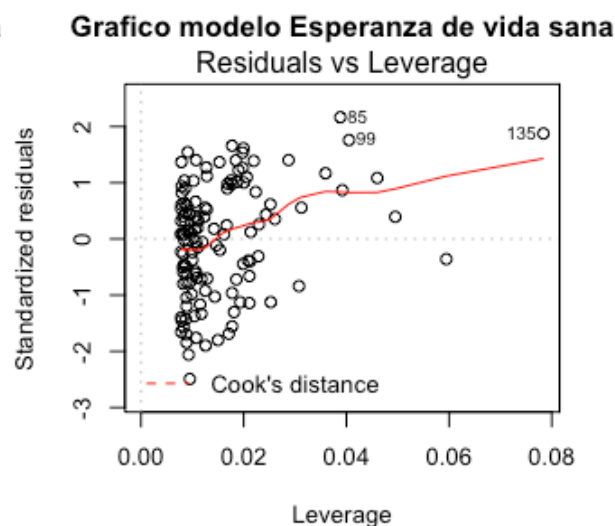
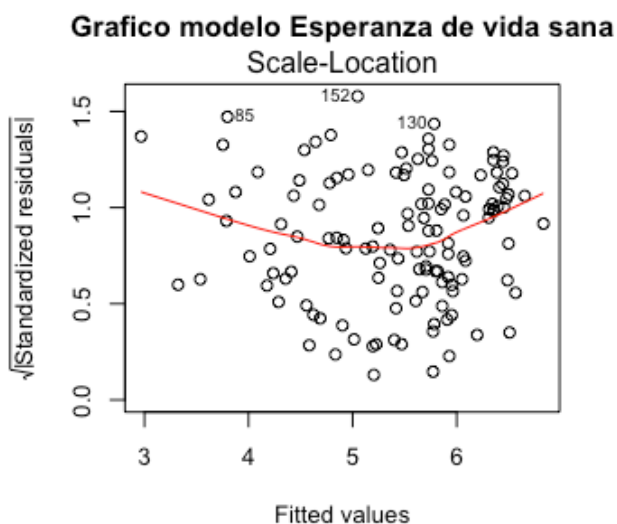
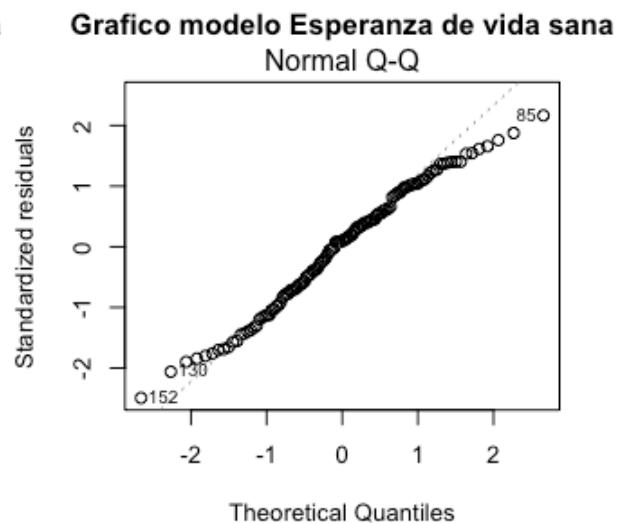
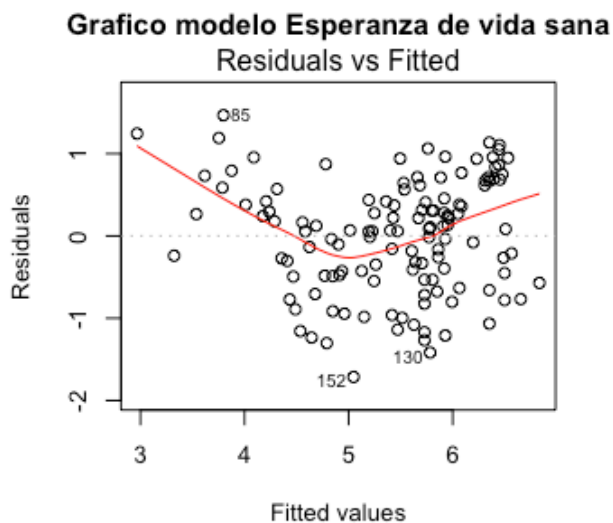
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.6889	0.1931	13.93	<2e-16 ***
Healthy.life.expectancy	3.7292	0.2534	14.71	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6944 on 126 degrees of freedom

Multiple R-squared: 0.6321, Adjusted R-squared: 0.6292

F-statistic: 216.5 on 1 and 126 DF, p-value: < 2.2e-16



- **Modelo Libertad de elección de vida**

[1] "----- Summary Freedom -----"

> summary(model_Freedom) #Devuelve valores de la regresion lineal para determinar la calidad

Call:

lm(formula = Score ~ Freedom.to.make.life.choices, data = training_data)

Residuals:

	Min	1Q	Median	3Q	Max
	-2.7280	-0.6060	0.0279	0.6879	1.8367

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.7704	0.2665	14.147	< 2e-16 ***
Freedom.to.make.life.choices	4.1289	0.6310	6.544	1.36e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.93 on 126 degrees of freedom

Multiple R-squared: 0.2536, Adjusted R-squared: 0.2477

F-statistic: 42.82 on 1 and 126 DF, p-value: 1.364e-09

Grafico modelo Libertad de eleccion de vic

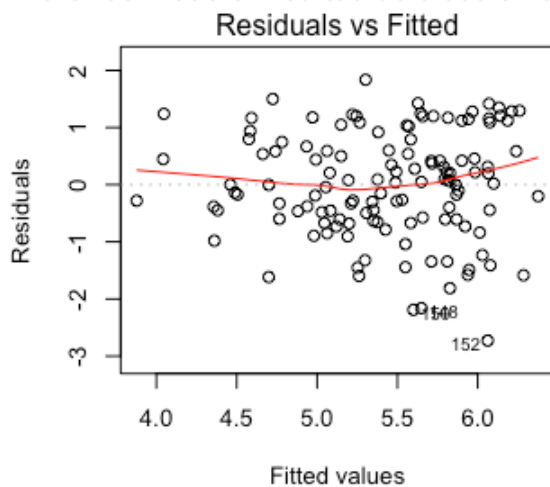


Grafico modelo Libertad de eleccion de vic

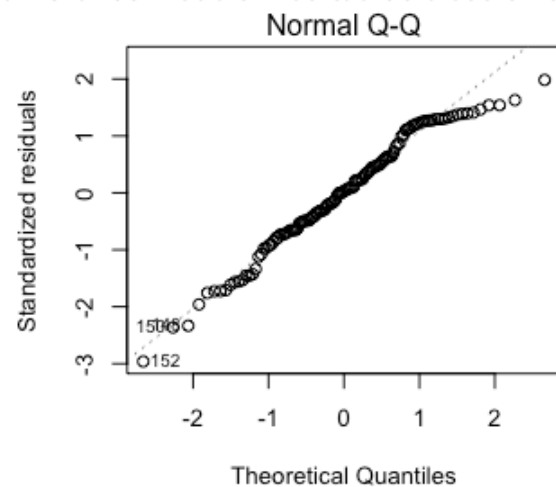


Grafico modelo Libertad de eleccion de vic

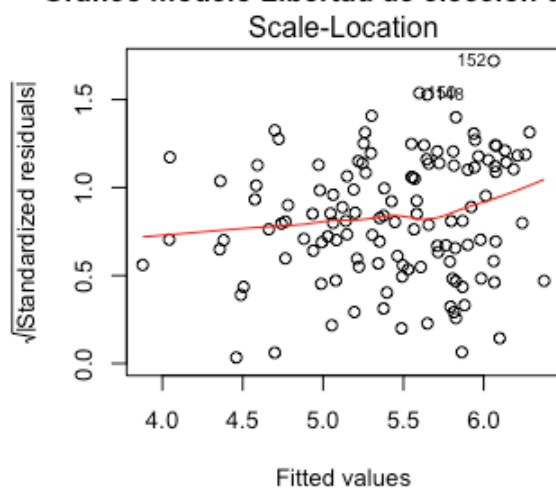
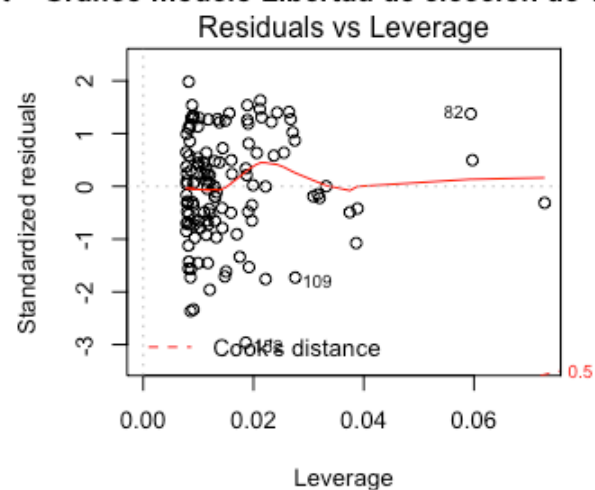


Grafico modelo Libertad de eleccion de vic



• Modelo Generosidad

```
> print(paste0("----- Summary Generosity -----"))
[1] "----- Summary Generosity -----"
> summary(model_Generosity) #Devuelve valores de la regresion lineal para determinar la cali
```

Call:
lm(formula = Score ~ Generosity, data = training_data)

Residuals:

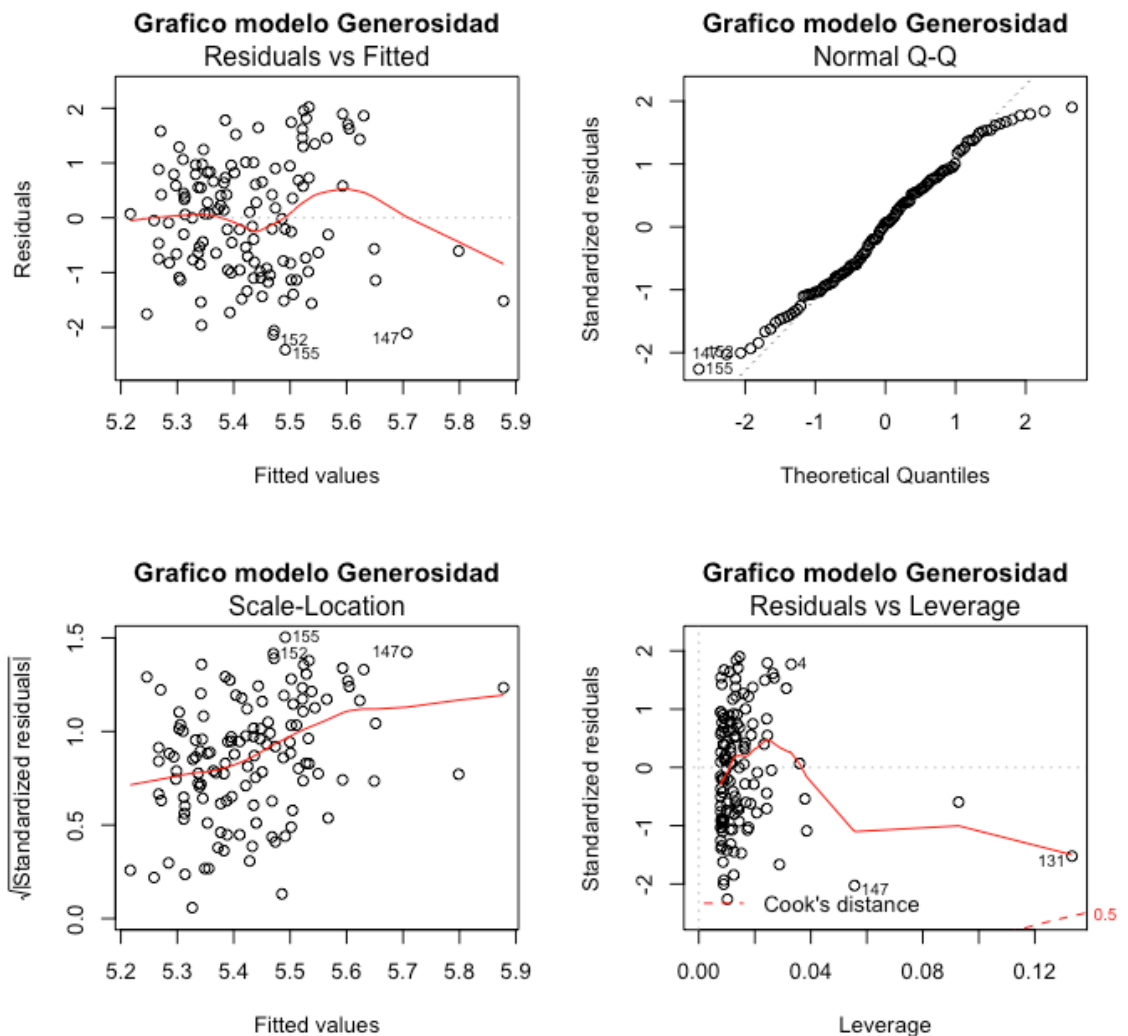
Min	1Q	Median	3Q	Max
-2.40831	-0.82671	0.06472	0.79982	2.02063

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.2168	0.2031	25.681	<2e-16 ***
Generosity	1.1681	0.9876	1.183	0.239

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.071 on 126 degrees of freedom
Multiple R-squared: 0.01098, Adjusted R-squared: 0.003133
F-statistic: 1.399 on 1 and 126 DF, p-value: 0.2391



- **Modelo Percepción de la corrupción**

```
> summary(model_Corruption) #Devuelve valores de la regresion lineal para determinar la cali
```

Call:

```
lm(formula = Score ~ Perceptions.of.corruption, data = training_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-3.4502	-0.7676	0.1447	0.8205	2.0198

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.9466	0.1369	36.138	< 2e-16 ***
Perceptions.of.corruption	4.4712	0.9706	4.606	9.88e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9959 on 126 degrees of freedom

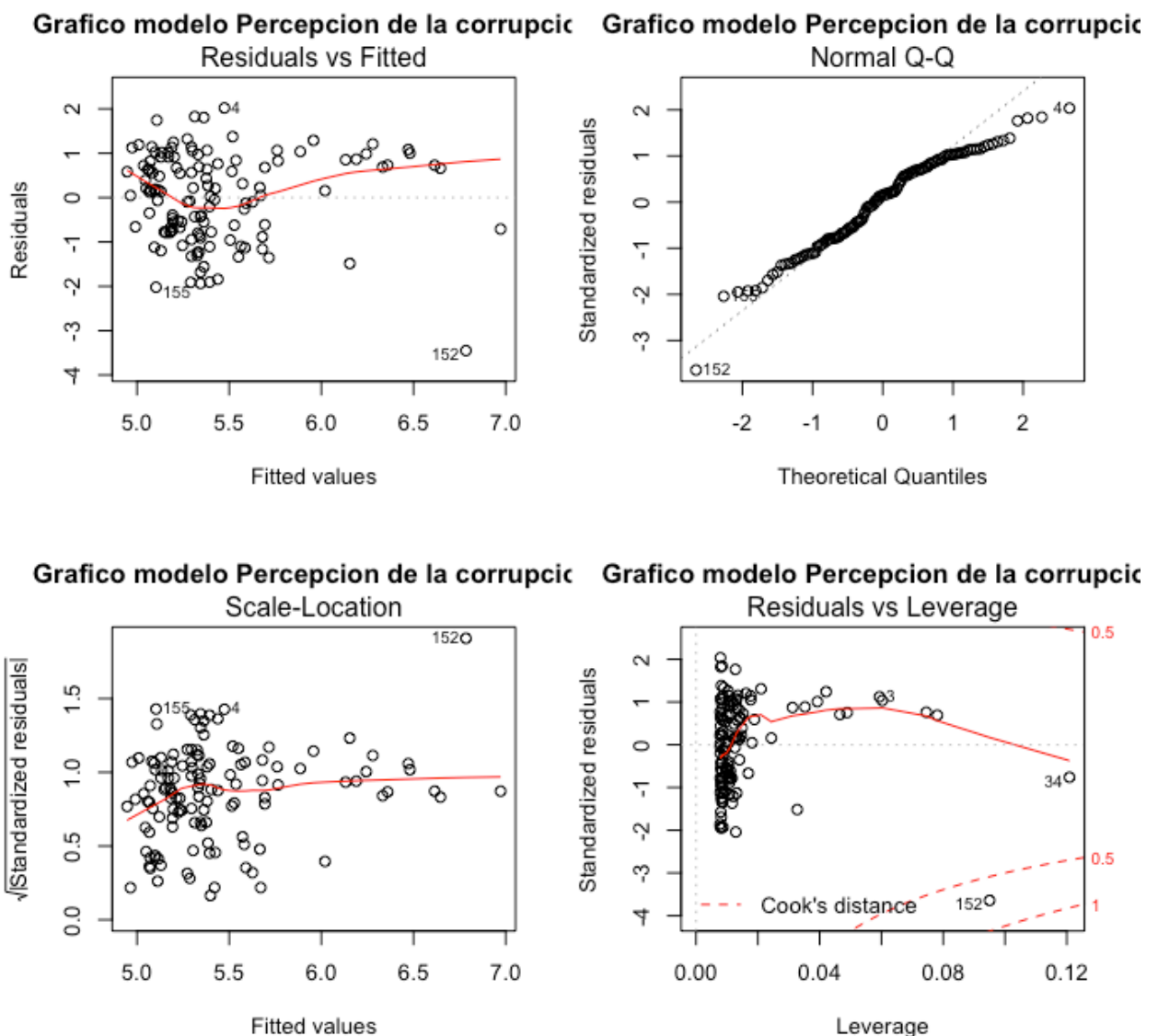
Multiple R-squared: 0.1441, Adjusted R-squared: 0.1373

F-statistic: 21.22 on 1 and 126 DF, p-value: 9.88e-06

```
> # Plot model: Crea graficos
```

```
> par(mfrow = c(2,2))
```

```
plot(model_Corruption)
```

CONCLUSIONES

Como podemos observar en la diferencia entre el modelo y el valor, el modelo cuyo error es mejor es el que utiliza todos los factores para determinar el resultado del ranking. Esto se debe a que al poseer mas factores puede proporcionar unos datos mas precisos.

En cuanto al resumen y generación de gráficos, genera la formula para determinar la calidad del modelo. En este caso, si miramos los coeficientes nos proporciona los valores para realizar la formula de predicción de felicidad.

Si miramos el modelo de los factores de forma independiente, el modelo con menos error es el modelo de PIB per Cápita y el que mas error tiene es el de la generosidad, por lo que sería el factor que menos se debería tener en cuenta a la hora de determinar cual es la felicidad de un pais.