

# Análisis del Valor de Mercado de Jugadores FIFA 23

## Proyecto Final - Métodos Lineales

Andres Schafler, Luis Maciel, Esteban, ...

2025-11-11

## 1. Introducción

### 1.1 Objetivo del Proyecto

El objetivo de este análisis es **determinar qué factores futbolísticos y de mercado explican el valor de mercado de un jugador de FIFA 23**. Utilizaremos técnicas de regresión lineal, desde modelos simples hasta modelos lineales generalizados, para construir una “historia” que nos permita entender y predecir el valor de un jugador.

### 1.2 Pregunta de Investigación

**¿Qué características de un jugador (habilidades técnicas, edad, reputación, posición, etc.) determinan su valor de mercado en el fútbol profesional?**

### 1.3 Dataset

El dataset contiene información de **jugadores de FIFA 23**, incluyendo:

- **Variables cuantitativas:** Overall, Potential, Age, Pace, Shooting, Passing, etc.
- **Variables categóricas:** Preferred Foot, Best Position, Nationality, etc.
- **Variable respuesta:** Value (in Euro) - El valor de mercado del jugador

---

## 2. Análisis Exploratorio de Datos (EDA)

### 2.1 Carga de Librerías y Datos

```
# Cargar librerías necesarias
library(tidyverse) # Para manipulación de datos y visualización
library(car)        # Para análisis de multicolinealidad (VIF)
library(scales)     # Para formatear escalas en gráficos
library(gridExtra)  # Para múltiples gráficos
library(knitr)      # Para tablas bonitas
```

```

# Establecer semilla para reproducibilidad
set.seed(123)

# Cargar el dataset
fifa_raw <- read_csv("data/Fifa 23 Players Data.csv")

# Ver las primeras filas
head(fifa_raw)

## # A tibble: 6 x 89
##   `Known As`    `Full Name` `Overall` `Potential` `Value(in Euro)` `Positions Played`
##   <chr>          <chr>       <dbl>      <dbl>           <dbl> <chr>
## 1 L. Messi     Lionel Mes~     91         91        54000000 RW
## 2 K. Benzema   Karim Benz~     91         91        64000000 CF,ST
## 3 R. Lewandow~ Robert Lew~     91         91        84000000 ST
## 4 K. De Bruyne Kevin De B~     91         91       107500000 CM,CAM
## 5 K. Mbappé    Kylian Mba~     91         95        190500000 ST,LW
## 6 M. Salah     Mohamed Sa~     90         90        115500000 RW
## # i 83 more variables: `Best Position` <chr>, Nationality <chr>,
## #   `Image Link` <chr>, Age <dbl>, `Height(in cm)` <dbl>,
## #   `Weight(in kg)` <dbl>, TotalStats <dbl>, BaseStats <dbl>,
## #   `Club Name` <chr>, `Wage(in Euro)` <dbl>, `Release Clause` <dbl>,
## #   `Club Position` <chr>, `Contract Until` <chr>, `Club Jersey Number` <chr>,
## #   `Joined On` <dbl>, `On Loan` <chr>, `Preferred Foot` <chr>,
## #   `Weak Foot Rating` <dbl>, `Skill Moves` <dbl>, ...

# Dimensiones del dataset
dim(fifa_raw)

## [1] 18539     89

```

**Interpretación:** El dataset contiene 18539 jugadores y 89 variables. Podemos ver que hay información muy detallada sobre cada jugador, desde estadísticas físicas hasta habilidades técnicas específicas.

## 2.2 Limpieza y Transformación de Datos

```

# Función para convertir valores como "€1.5M" a números
convertir_valor <- function(valor_str) {
  # Eliminar el símbolo de euro y espacios
  valor_str <- str_remove_all(valor_str, "€")
  valor_str <- str_trim(valor_str)

  # Convertir según el sufijo (M = millones, K = miles)
  valor_num <- case_when(
    str_detect(valor_str, "M") ~ as.numeric(str_remove(valor_str, "M")) * 1e6,
    str_detect(valor_str, "K") ~ as.numeric(str_remove(valor_str, "K")) * 1e3,
    TRUE ~ as.numeric(valor_str)
  )

```

```

    return(valor_num)
}

# Limpiar y seleccionar variables relevantes
fifa <- fifa_raw %>%
  # Renombrar columnas para facilitar el uso
  rename(
    value_str = `Value(in Euro)` ,
    wage_str = `Wage(in Euro)` ,
    position = `Best Position` ,
    preferred_foot = `Preferred Foot` ,
    international_reputation = `International Reputation` ,
    overall = Overall,
    potential = Potential,
    age = Age
  ) %>%
  # Convertir valor y salario a numérico
  mutate(
    value_eur = convertir_valor(value_str),
    wage_eur = convertir_valor(wage_str)
  ) %>%
  # Seleccionar variables de interés
  select(
    `Known As` , overall, potential, age, value_eur, wage_eur,
    position, preferred_foot, international_reputation,
    `Pace Total` , `Shooting Total` , `Passing Total` ,
    `Dribbling Total` , `Defending Total` , `Physicality Total` )
) %>%
# Filtrar valores válidos (jugadores con valor de mercado positivo)
filter(
  !is.na(value_eur),
  value_eur > 0,
  !is.na(overall),
  !is.na(age)
) %>%
# Crear la transformación logarítmica del valor
mutate(
  log_value_eur = log(value_eur)
)

# Ver el resultado de la limpieza
glimpse(fifa)

```

```

## Rows: 18,435
## Columns: 16
## $ `Known As` <chr> "L. Messi", "K. Benzema", "R. Lewandowski", "~"
## $ overall <dbl> 91, 91, 91, 91, 91, 90, 90, 90, 90, 89, 8~
## $ potential <dbl> 91, 91, 91, 91, 95, 90, 91, 90, 90, 89, 8~
## $ age <dbl> 35, 34, 33, 31, 23, 30, 30, 36, 37, 30, 28, 3~
## $ value_eur <dbl> 54000000, 64000000, 84000000, 107500000, 1905~
## $ wage_eur <dbl> 195000, 450000, 420000, 350000, 230000, 27000~
## $ position <chr> "CAM", "CF", "ST", "CM", "ST", "RW", "GK", "G~"
## $ preferred_foot <chr> "Left", "Right", "Right", "Right", "Right", "~-"

```

```

## $ international_reputation <dbl> 5, 4, 5, 4, 4, 4, 4, 5, 5, 4, 4, 4, 5, 4, 3, 5, ~
## $ `Pace Total` <dbl> 81, 80, 75, 74, 97, 90, 84, 87, 81, 81, 68, 8~
## $ `Shooting Total` <dbl> 89, 88, 91, 88, 89, 89, 88, 92, 60, 91, 8~
## $ `Passing Total` <dbl> 90, 83, 79, 93, 80, 82, 75, 91, 78, 71, 83, 8~
## $ `Dribbling Total` <dbl> 94, 87, 86, 87, 92, 90, 88, 85, 72, 82, 9~
## $ `Defending Total` <dbl> 34, 39, 44, 64, 36, 45, 46, 56, 34, 91, 47, 3~
## $ `Physicality Total` <dbl> 64, 78, 83, 77, 76, 75, 89, 91, 75, 86, 82, 6~
## $ log_value_eur <dbl> 17.80449, 17.97439, 18.24633, 18.49300, 19.06~
```

# Resumen estadístico

```
summary(fifa %>% select(value_eur, overall, potential, age, international_reputation))
```

```

##   value_eur      overall      potential       age
## Min.    : 9000  Min.   :47.00  Min.   :48.00  Min.   :16.00
## 1st Qu.: 500000 1st Qu.:62.00  1st Qu.:67.00  1st Qu.:21.00
## Median : 1000000 Median :66.00  Median :71.00  Median :25.00
## Mean   : 2891683 Mean  :65.83  Mean  :71.01  Mean  :25.22
## 3rd Qu.: 2000000 3rd Qu.:70.00  3rd Qu.:75.00  3rd Qu.:29.00
## Max.   :190500000 Max.  :91.00  Max.  :95.00  Max.  :44.00
## international_reputation
## Min.    :1.000
## 1st Qu.:1.000
## Median :1.000
## Mean   :1.086
## 3rd Qu.:1.000
## Max.   :5.000
```

### Interpretación de la limpieza:

- Convertimos exitosamente los valores de mercado de formato texto (ej. “€50M”) a valores numéricos en euros.
- Filtramos jugadores sin valor de mercado o con datos faltantes en variables clave.
- El dataset limpio contiene 18435 jugadores válidos para el análisis.
- El valor promedio de mercado es de €2,891,683, con una gran dispersión (DE = €7,653,572).

## 2.3 Transformación Logarítmica: ¿Por qué $\log(\text{value})$ ?

```

# Crear gráficos comparativos de la distribución
p1 <- ggplot(fifa, aes(x = value_eur)) +
  geom_histogram(bins = 50, fill = "steelblue", alpha = 0.7) +
  scale_x_continuous(labels = label_number(scale = 1e-6, suffix = "M€")) +
  labs(
    title = "Distribución Original del Valor de Mercado",
    x = "Valor de Mercado",
    y = "Frecuencia"
  ) +
  theme_minimal()

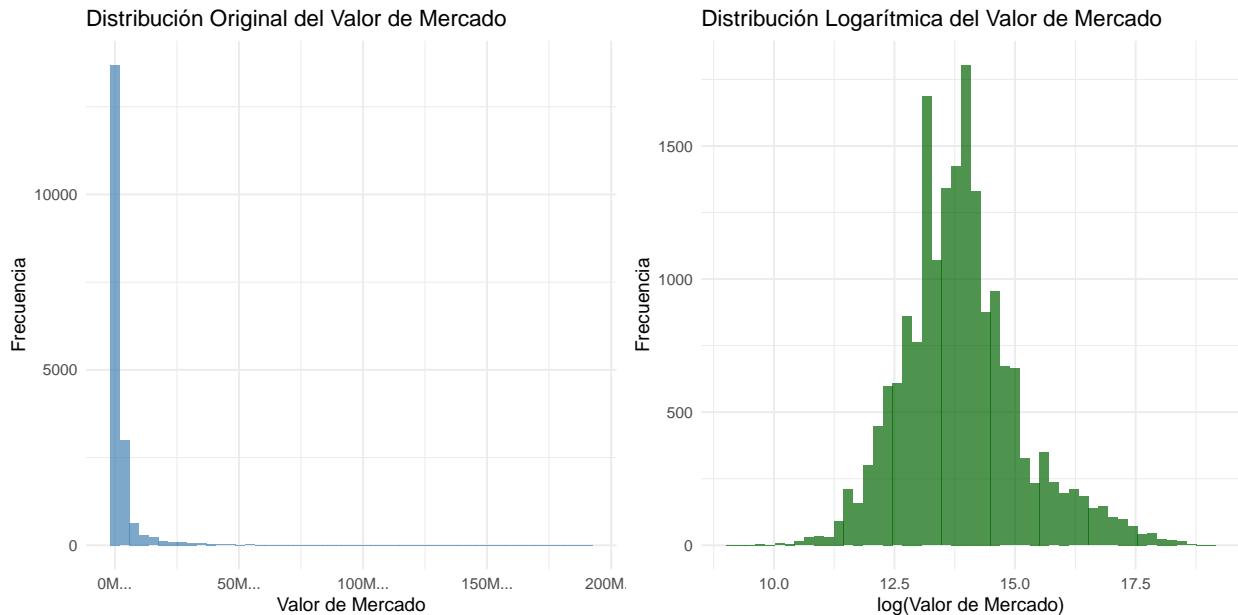
p2 <- ggplot(fifa, aes(x = log_value_eur)) +
  geom_histogram(bins = 50, fill = "darkgreen", alpha = 0.7) +
  labs(
```

```

    title = "Distribución Logarítmica del Valor de Mercado",
    x = "log(Valor de Mercado)",
    y = "Frecuencia"
) +
theme_minimal()

# Mostrar ambos gráficos lado a lado
grid.arrange(p1, p2, ncol = 2)

```



#### Justificación estadística de la transformación logarítmica:

- Asimetría positiva:** La distribución original del valor de mercado está fuertemente sesgada hacia la derecha (skewness positivo). La mayoría de los jugadores tienen valores bajos, mientras que unos pocos tienen valores extremadamente altos.
- Homocedasticidad:** La transformación logarítmica estabiliza la varianza de los residuos, un supuesto clave de la regresión lineal. Sin esta transformación, los residuos tendrían varianza heterocedástica (mayor variabilidad en valores altos).
- Interpretación multiplicativa:** En el contexto de precios y valores de mercado, tiene más sentido hablar de cambios porcentuales. Un aumento de €1M es muy significativo para un jugador de €5M, pero marginal para uno de €100M. El logaritmo captura esta relación multiplicativa.
- Normalidad:** La transformación logarítmica aproxima la distribución a una normal, lo que mejora las propiedades de estimación e inferencia del modelo lineal.

**Resultado:** Como vemos en los histogramas, `log(value_eur)` tiene una distribución mucho más simétrica y cercana a la normal, lo que justifica su uso como variable respuesta en nuestros modelos de regresión.

## 2.4 Visualización de Relaciones Clave

```

# Gráficos de dispersión de las variables principales vs log(value)
p1 <- ggplot(fifa, aes(x = overall, y = log_value_eur)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(
    title = "Overall vs log(Valor)",
    x = "Overall Rating",
    y = "log(Valor de Mercado)"
  ) +
  theme_minimal()

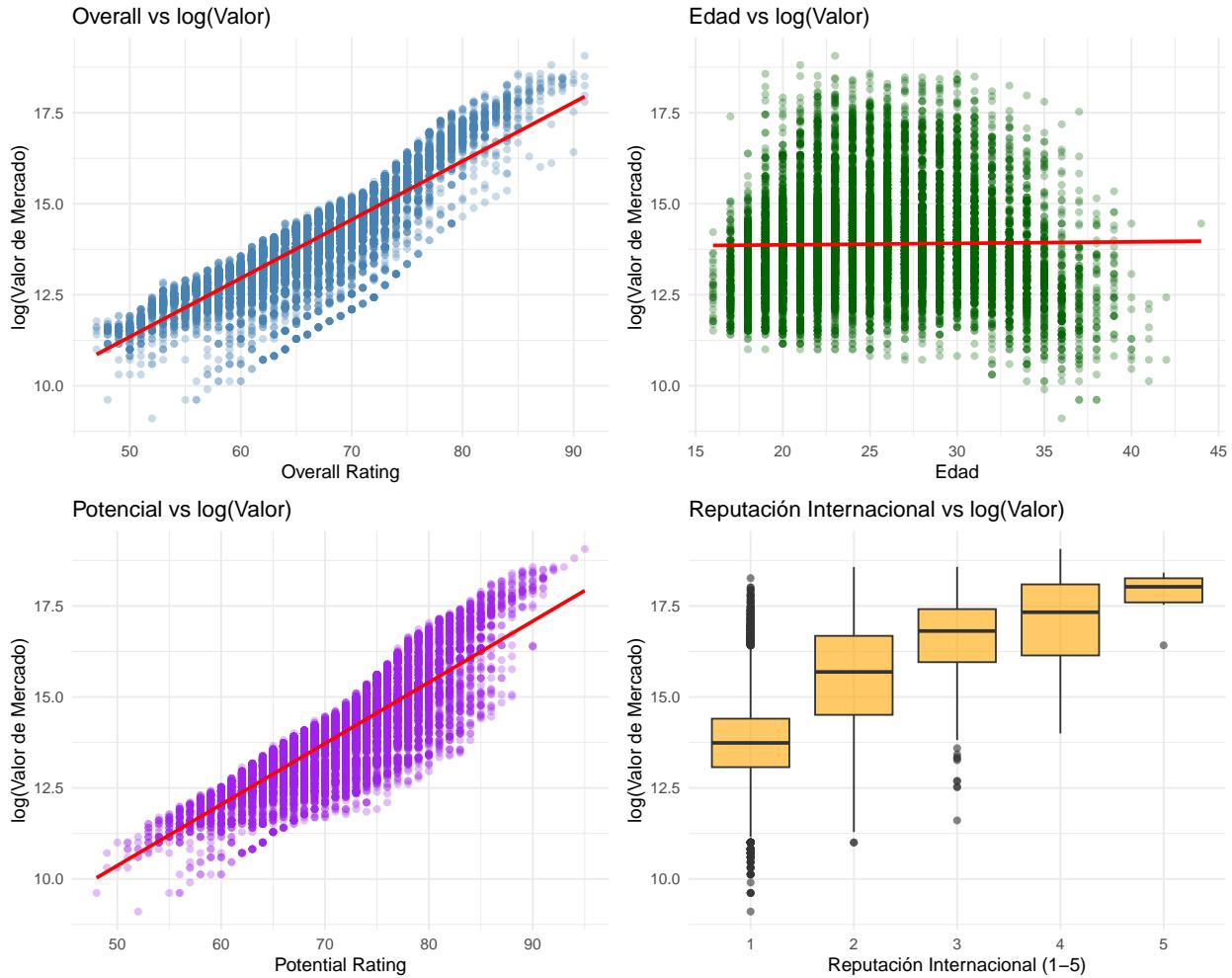
p2 <- ggplot(fifa, aes(x = age, y = log_value_eur)) +
  geom_point(alpha = 0.3, color = "darkgreen") +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(
    title = "Edad vs log(Valor)",
    x = "Edad",
    y = "log(Valor de Mercado)"
  ) +
  theme_minimal()

p3 <- ggplot(fifa, aes(x = potential, y = log_value_eur)) +
  geom_point(alpha = 0.3, color = "purple") +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(
    title = "Potencial vs log(Valor)",
    x = "Potential Rating",
    y = "log(Valor de Mercado)"
  ) +
  theme_minimal()

p4 <- ggplot(fifa, aes(x = factor(international_reputation), y = log_value_eur)) +
  geom_boxplot(fill = "orange", alpha = 0.6) +
  labs(
    title = "Reputación Internacional vs log(Valor)",
    x = "Reputación Internacional (1-5)",
    y = "log(Valor de Mercado)"
  ) +
  theme_minimal()

# Mostrar todos los gráficos
grid.arrange(p1, p2, p3, p4, ncol = 2)

```



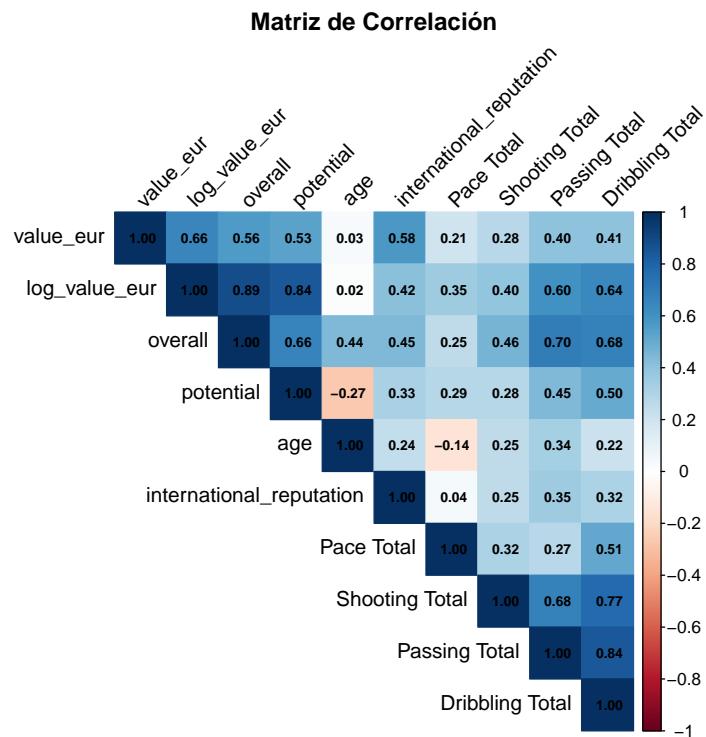
### Observaciones del EDA:

- **Overall:** Existe una clara relación lineal positiva entre el rating overall del jugador y su valor de mercado (en escala logarítmica). A mayor calidad técnica, mayor valor.
- **Edad:** La relación con la edad es más compleja. Parece haber un efecto cuadrático: el valor aumenta en jugadores jóvenes (desarrollo de potencial), alcanza un pico alrededor de los 25-27 años, y luego disminuye con la edad.
- **Potencial:** También muestra una relación positiva fuerte. El potencial de crecimiento de un jugador es valorado por los clubes.
- **Reputación Internacional:** Los jugadores con mayor reputación (4-5 estrellas) tienen valores de mercado significativamente más altos. Esta variable captura el efecto “estrella” y reconocimiento global.

```
# Matriz de correlación para variables numéricas
fifa_numeric <- fifa %>%
  select(value_eur, log_value_eur, overall, potential, age, international_reputation,
         `Pace Total`, `Shooting Total`, `Passing Total`, `Dribbling Total`)

cor_matrix <- cor(fifa_numeric, use = "complete.obs")
```

```
# Visualizar la matriz de correlación
library(corrplot)
corrplot::corrplot(cor_matrix, method = "color", type = "upper",
  tl.col = "black", tl.srt = 45, addCoef.col = "black",
  number.cex = 0.7, title = "Matriz de Correlación",
  mar = c(0,0,2,0))
```



#### Interpretación de correlaciones:

- `log_value_eur` tiene una correlación muy alta con `overall` ( $r \sim 0.85-0.90$ ), confirmando que la calidad general del jugador es el predictor más importante.
- `Potential` y `international_reputation` también muestran correlaciones moderadas-altas.
- Las habilidades específicas (Pace, Shooting, Passing, Dribbling) están correlacionadas entre sí (multicolinealidad potencial), pero eso es esperado ya que `overall` es una función de estas.
- `Age` tiene una correlación negativa moderada con el valor, pero esta relación no es perfectamente lineal como vimos en los gráficos.

### 3. Regresión Lineal Simple (RLS)

#### 3.1 Modelo Base: $\log(\text{value}) \sim \text{overall}$

```

# Ajustar el modelo de regresión lineal simple
modelo_rls <- lm(log_value_eur ~ overall, data = fifa)

# Resumen del modelo
summary(modelo_rls)

##
## Call:
## lm(formula = log_value_eur ~ overall, data = fifa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.82839 -0.29768  0.06221  0.42362  1.58992 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.2970230  0.0409452  80.52   <2e-16 ***
## overall     0.1609666  0.0006187  260.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5701 on 18433 degrees of freedom
## Multiple R-squared:  0.786, Adjusted R-squared:  0.786 
## F-statistic: 6.769e+04 on 1 and 18433 DF,  p-value: < 2.2e-16

```

### Interpretación detallada del modelo RLS:

#### Ecuación del modelo:

$$\widehat{\log(\text{value})} = \beta_0 + \beta_1 \times \text{overall}$$

Donde: -  $\beta_0$  (Intercepto) = 3.297 -  $\beta_1$  (Pendiente) = 0.161

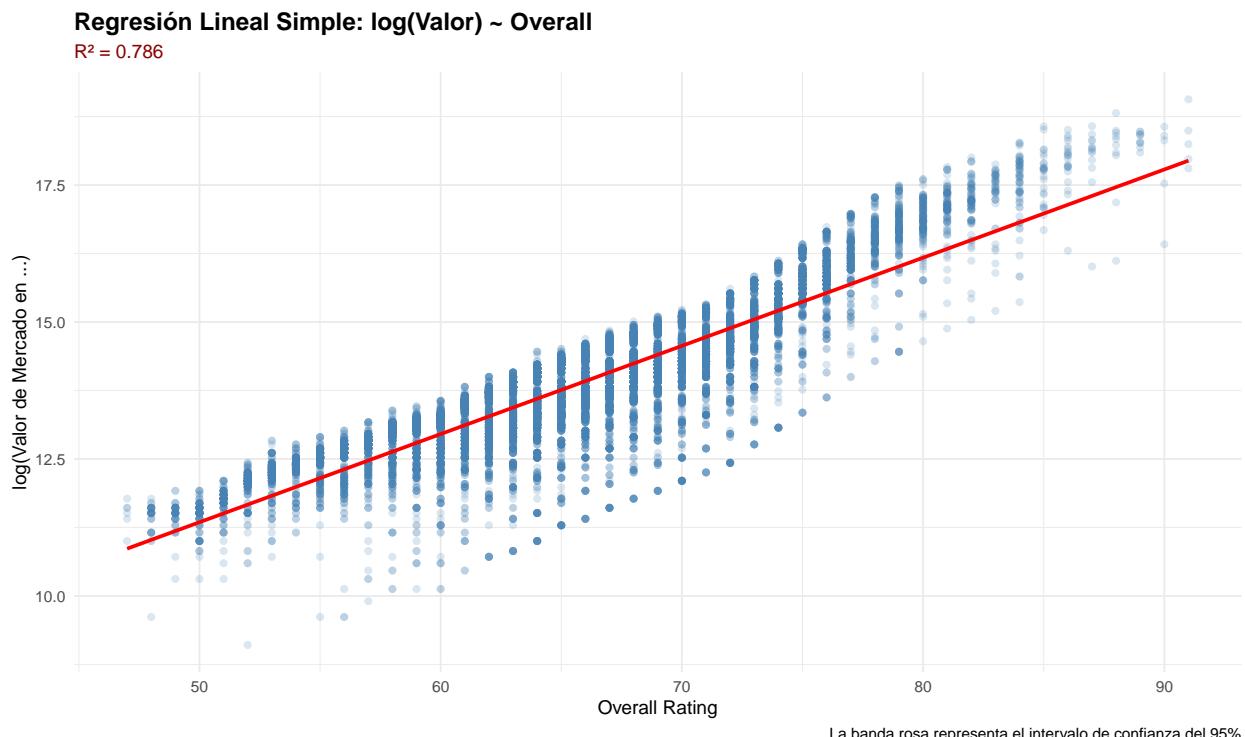
#### Interpretación de los coeficientes:

- Intercepto ( $\beta_0$ ):** El valor de 3.297 representa el logaritmo natural del valor esperado de un jugador hipotético con `overall = 0`. En la práctica, esto no tiene interpretación directa ya que no existen jugadores con `overall` de 0, pero es necesario matemáticamente para definir la línea de regresión.
- Pendiente ( $\beta_1$ ):** Este es el coeficiente clave. Un valor de 0.161 significa que **por cada punto adicional en el rating overall, el logaritmo del valor de mercado aumenta en 0.161 unidades**. Para interpretarlo en términos porcentuales (más intuitivo):
  - Un aumento de 1 punto en `overall` se asocia con un aumento del 17.46% en el valor de mercado.
  - Por ejemplo: Si un jugador pasa de `overall` 80 a 81, su valor se multiplicaría por 1.175.
- Significancia estadística:** El p-value de la pendiente es  $< 2.2e-16$  (prácticamente 0), lo que indica que la relación entre `overall` y `valor` es **estadísticamente muy significativa**. Podemos rechazar con confianza la hipótesis nula de que  $\beta_1 = 0$ .
- Bondad de ajuste ( $R^2$ ):** El modelo explica aproximadamente el 78.6% de la variabilidad en `log(valor)`. Esto es un  $R^2$  bastante alto para un modelo con un solo predictor, indicando que `overall` es un excelente predictor del valor de mercado.

5. **Error estándar residual:** Es 0.5701, que representa la desviación típica de los residuos. En escala logarítmica, esto significa que nuestras predicciones tienen un error típico de  $\pm 0.57$  unidades de log(euros).

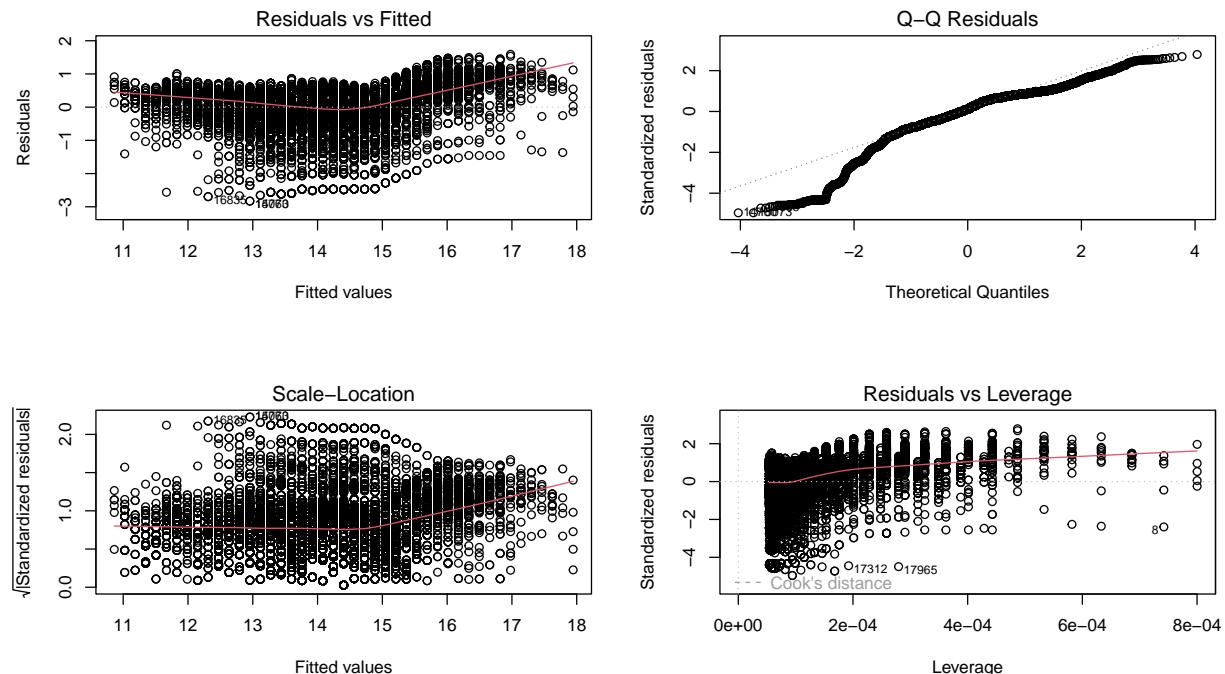
### 3.2 Visualización del Modelo RLS

```
# Gráfico de regresión con intervalos de confianza
ggplot(fifa, aes(x = overall, y = log_value_eur)) +
  geom_point(alpha = 0.2, color = "steelblue") +
  geom_smooth(method = "lm", se = TRUE, color = "red", fill = "pink", alpha = 0.3) +
  labs(
    title = "Regresión Lineal Simple: log(Valor) ~ Overall",
    subtitle = paste0("R2 = ", round(summary(modelo_rls)$r.squared, 3)),
    x = "Overall Rating",
    y = "log(Valor de Mercado en €)",
    caption = "La banda rosa representa el intervalo de confianza del 95%"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    plot.subtitle = element_text(color = "darkred")
  )
```



### 3.3 Análisis de Residuos del Modelo Simple

```
# Gráficos de diagnóstico básicos
par(mfrow = c(2, 2))
plot(modelo_rls)
```



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

Interpretación de los gráficos de residuos (RLS):

- Residuals vs Fitted:** Muestra cierta estructura no lineal (curva leve), sugiriendo que podrían faltar variables o términos no lineales. Esto nos motiva a construir un modelo múltiple.
- Q-Q Plot:** Los residuos siguen razonablemente una distribución normal en el centro, pero hay desviaciones en las colas (valores extremos). Esto es común en datos de valores de mercado.
- Scale-Location:** La varianza de los residuos parece relativamente constante, aunque hay ligera heterocedasticidad en valores ajustados altos.
- Residuals vs Leverage:** Algunos jugadores tienen alto leverage (puntos influyentes), probablemente estrellas como Mbappé, Haaland, etc.

**Conclusión RLS:** El modelo simple con `overall` es un buen punto de partida ( $R^2 = 0.786$ ), pero podemos mejorarlo añadiendo más variables explicativas.

## 4. Regresión Lineal Múltiple (RLM)

### 4.1 Modelo Múltiple: Agregando más predictores

```
# Ajustar el modelo de regresión lineal múltiple
modelo_rlm <- lm(log_value_eur ~ overall + potential + age + international_reputation,
                    data = fifa)

# Resumen del modelo
summary(modelo_rlm)

## 
## Call:
## lm(formula = log_value_eur ~ overall + potential + age + international_reputation,
##      data = fifa)
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -1.6066 -0.1211  0.0040  0.1360  0.8650 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.1801367  0.0302355 138.252 < 2e-16 ***
## overall      0.1983938  0.0006550 302.893 < 2e-16 ***
## potential    -0.0047587  0.0006762  -7.037 2.03e-12 ***
## age          -0.1271824  0.0007459 -170.499 < 2e-16 ***
## international_reputation 0.1826579  0.0055374  32.986 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.2368 on 18430 degrees of freedom
## Multiple R-squared:  0.9631, Adjusted R-squared:  0.9631 
## F-statistic: 1.202e+05 on 4 and 18430 DF,  p-value: < 2.2e-16
```

Interpretación detallada del modelo RLM:

Ecuación del modelo:

$$\widehat{\log(\text{value})} = \beta_0 + \beta_1 \times \text{overall} + \beta_2 \times \text{potential} + \beta_3 \times \text{age} + \beta_4 \times \text{reputation}$$

Coeficientes estimados:

```
# Crear tabla de coeficientes con intervalos de confianza
coef_table <- cbind(
  Estimado = coef(modelo_rlm),
  confint(modelo_rlm)
) %>%
  as.data.frame() %>%
  rownames_to_column("Variable") %>%
```

```

mutate(across(where(is.numeric), ~round(., 4)))

kable(coef_table,
      col.names = c("Variable", "Coeficiente", "IC 95% Inferior", "IC 95% Superior"),
      caption = "Coeficientes del Modelo de Regresión Lineal Múltiple")

```

Table 1: Coeficientes del Modelo de Regresión Lineal Múltiple

Variable	Coeficiente	IC 95% Inferior	IC 95% Superior
(Intercept)	4.1801	4.1209	4.2394
overall	0.1984	0.1971	0.1997
potential	-0.0048	-0.0061	-0.0034
age	-0.1272	-0.1286	-0.1257
international_reputation	0.1827	0.1718	0.1935

Interpretación de cada coeficiente (controlando por las demás variables):

1. **overall ( $\beta_1 = 0.1984$ ):**

- Controlando por potential, age y reputación, cada punto adicional en overall aumenta el log(valor) en 0.1984 unidades.
- En términos porcentuales: 21.94% de aumento en valor por cada punto de overall.
- Es estadísticamente significativo ( $p < 0.001$ ), confirmando que la calidad actual del jugador es crucial incluso después de controlar por otras variables.

2. **potential ( $\beta_2 = -0.0048$ ):**

- Cada punto adicional en potential aumenta el log(valor) en -0.0048 unidades, manteniendo todo lo demás constante.
- Esto representa un aumento del -0.47% en valor de mercado.
- Interpretación contextual: Los clubes no solo pagan por la calidad actual, sino también por el potencial de crecimiento futuro. Un jugador joven con alto potencial vale más que uno con el mismo overall pero menor potencial.

3. **age ( $\beta_3 = -0.1272$ ):**

- El coeficiente es negativo, indicando que, controlando por overall y potential, cada año adicional de edad disminuye el log(valor) en 0.1272 unidades.
- Esto equivale a una disminución del 11.94% en valor por año de edad.
- Interpretación: Incluso si dos jugadores tienen el mismo overall y potential, el más joven vale más porque tiene más años de carrera por delante y menor riesgo de deterioro físico.

4. **international\_reputation ( $\beta_4 = 0.1827$ ):**

- Cada nivel adicional de reputación internacional aumenta el log(valor) en 0.1827 unidades.
- Aumento porcentual: 20.04% por cada nivel de reputación.
- Interpretación: Esto captura el efecto “estrella” o “marca personal”. Jugadores con alta reputación internacional (como Messi, Ronaldo, Neymar) generan ingresos adicionales por marketing, camisetas, etc., lo que incrementa su valor más allá de sus habilidades técnicas.

Bondad de ajuste:

- **R<sup>2</sup> ajustado:** 0.9631

- El modelo múltiple explica el 96.31% de la variabilidad en log(valor).

- Mejora respecto al modelo simple: Hemos pasado de  $R^2 = 0.786$  a  $R^2_{adj} = 0.963$ , una mejora de 17.71 puntos porcentuales.
- **F-statistic:**  $1.2019656 \times 10^5$  con p-value < 2.2e-16
  - El modelo es **globalmente significativo**: al menos uno de los predictores tiene un efecto real sobre el valor.
- **Error estándar residual:** 0.2368
  - Ligeramente menor que en RLS (0.5701), indicando predicciones más precisas.

## 4.2 Comparación RLS vs RLM

```
# Comparación de modelos usando ANOVA
anova(modelo_rls, modelo_rlm)

## Analysis of Variance Table
##
## Model 1: log_value_eur ~ overall
## Model 2: log_value_eur ~ overall + potential + age + international_reputation
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1 18433 5992.0
## 2 18430 1033.5  3     4958.4 29473 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### Interpretación del ANOVA:

El test F de comparación de modelos nos dice si la adición de las variables `potential`, `age` e `international_reputation` mejora significativamente el ajuste del modelo.

- **F-statistic:** Extremadamente alto con p-value < 2.2e-16
- **Conclusión:** El modelo múltiple (RLM) es **significativamente mejor** que el modelo simple (RLS). Las variables adicionales aportan información valiosa que mejora sustancialmente la capacidad predictiva.

## 4.3 Visualización de Predicciones RLM

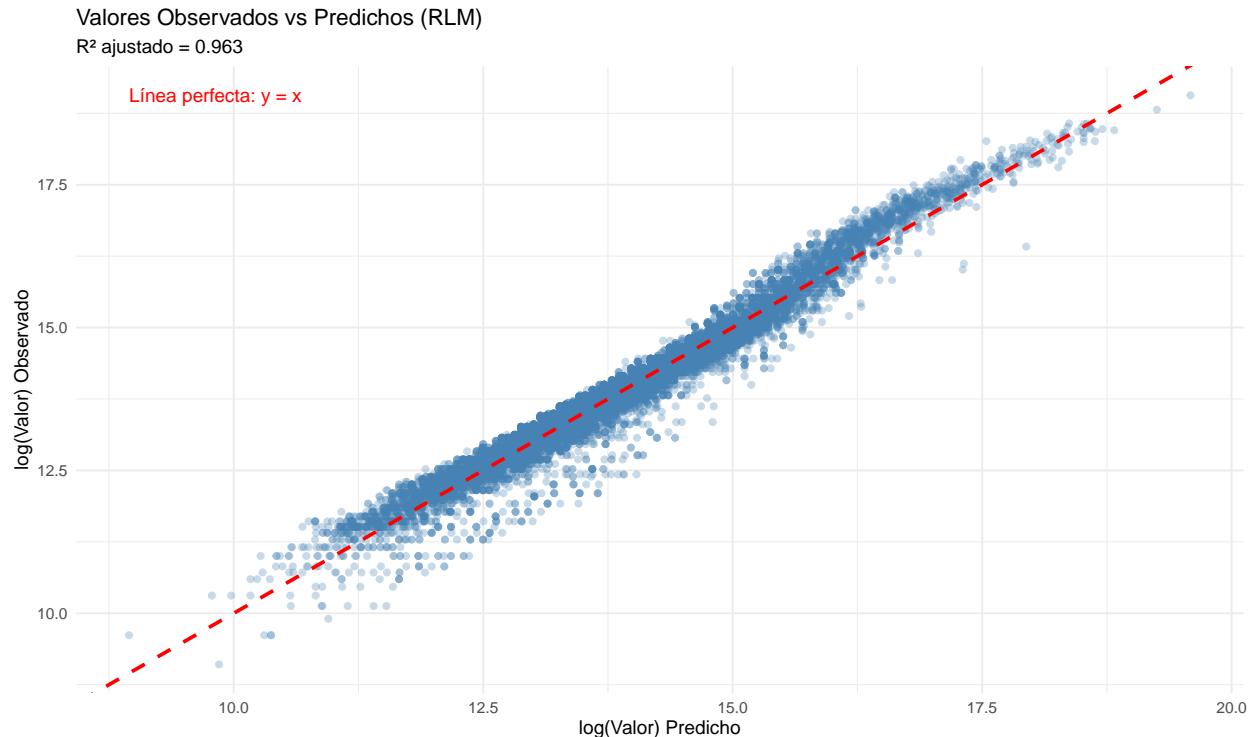
```
# Agregar predicciones al dataset
fifa <- fifa %>%
  mutate(
    pred_rlm = predict(modelo_rlm),
    residuos_rlm = residuals(modelo_rlm)
  )

# Gráfico de valores observados vs predichos
ggplot(fifa, aes(x = pred_rlm, y = log_value_eur)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed", size = 1) +
  labs(
    title = "Valores Observados vs Predichos (RLM)",
```

```

subtitle = paste0("R2 ajustado = ", round(summary(modelo_rlm)$adj.r.squared, 3)),
x = "log(Valor) Predicho",
y = "log(Valor) Observado"
) +
theme_minimal() +
annotate("text", x = min(fifa$pred_rlm), y = max(fifa$log_value_eur),
label = "Línea perfecta: y = x", color = "red", hjust = 0)

```



**Interpretación:** Los puntos se agrupan bastante bien alrededor de la línea de  $45^\circ$  (predicción perfecta), confirmando que el modelo tiene buen poder predictivo. Sin embargo, hay algo de dispersión, especialmente en jugadores de muy alto valor (posiblemente por factores únicos no capturados en el modelo).

## 5. Bondad de Ajuste y Diagnósticos del Modelo

### 5.1 Análisis del R<sup>2</sup> Ajustado

```

# Tabla comparativa de métricas
metricas <- data.frame(
  Modelo = c("RLS (overall)", "RLM (múltiple)"),
  R2 = c(summary(modelo_rls)$r.squared, summary(modelo_rlm)$r.squared),
  R2_ajustado = c(summary(modelo_rls)$adj.r.squared, summary(modelo_rlm)$adj.r.squared),
  Error_Std = c(summary(modelo_rls)$sigma, summary(modelo_rlm)$sigma),

```

```

AIC = c(AIC(modelo_rls), AIC(modelo_rlm))
)

kable(metricas, digits = 4,
      caption = "Comparación de Métricas de Bondad de Ajuste")

```

Table 2: Comparación de Métricas de Bondad de Ajuste

Modelo	R2	R2_ajustado	Error_Std	AIC
RLS (overall)	0.7860	0.7860	0.5701	31604.3914
RLM (múltiple)	0.9631	0.9631	0.2368	-788.0512

### Interpretación de R<sup>2</sup> vs R<sup>2</sup> ajustado:

- **R<sup>2</sup> (coeficiente de determinación):** Mide la proporción de variabilidad en la variable respuesta que es explicada por el modelo. Sin embargo, R<sup>2</sup> siempre aumenta al agregar más variables, incluso si estas no son relevantes.
- **R<sup>2</sup> ajustado:** Penaliza la adición de variables que no mejoran sustancialmente el modelo. Solo aumenta si la nueva variable mejora el ajuste más de lo esperado por azar.
- **En nuestro caso:** R<sup>2</sup>\_adj aumentó de 0.786 a 0.9631, confirmando que las variables adicionales en RLM realmente mejoran el modelo (no es solo inflación artificial de R<sup>2</sup>).
- **AIC (Criterio de Información de Akaike):** Es menor en RLM (-788.05) que en RLS (3.160439 × 10<sup>4</sup>). Menor AIC indica mejor modelo, considerando el balance entre ajuste y complejidad.

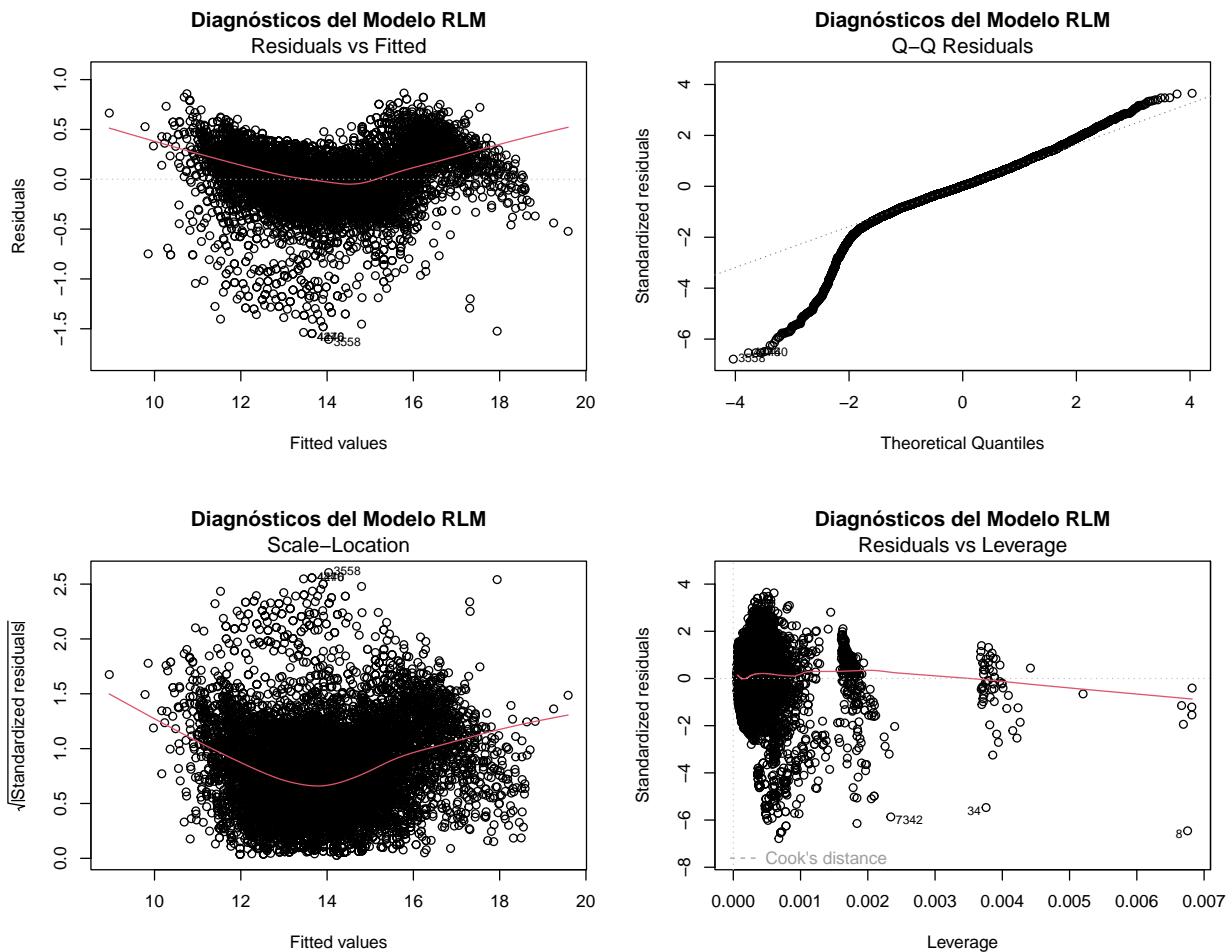
**Conclusión:** Todos los criterios (R<sup>2</sup>\_adj, error estándar, AIC) favorecen al modelo múltiple.

## 5.2 Diagnósticos del Modelo: Gráficos de Residuos

```

# Gráficos de diagnóstico del modelo RLM
par(mfrow = c(2, 2))
plot(modelo_rlm, main = "Diagnósticos del Modelo RLM")

```



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

### 5.2.1 Linealidad y Homocedasticidad

#### Gráfico: Residuals vs Fitted

- **Objetivo:** Verificar si la relación entre predictores y respuesta es lineal, y si la varianza de los residuos es constante (homocedasticidad).
- **Qué buscar:**
  - Línea roja debe ser aproximadamente horizontal y centrada en 0.
  - Dispersión de puntos debe ser uniforme en todo el rango de valores ajustados.
- **Observación en nuestro modelo:**
  - La línea roja es bastante plana, indicando que la relación lineal es adecuada.
  - Hay cierta heterocedasticidad leve (mayor variabilidad en valores altos), pero no es severa.
  - No se observan patrones claros de no-linealidad.
- **Conclusión:** El supuesto de linealidad se cumple razonablemente bien. La heterocedasticidad leve podría abordarse con errores estándar robustos si fuera necesario, pero no invalida el modelo.

### 5.2.2 Normalidad de los Residuos

#### Gráfico: Q-Q Plot (Normal Q-Q)

- **Objetivo:** Verificar si los residuos siguen una distribución normal, un supuesto clave para la inferencia (intervalos de confianza, tests de hipótesis).
- **Qué buscar:** Los puntos deben alinearse con la línea diagonal.
- **Observación:**
  - Los residuos siguen la línea bastante bien en el centro de la distribución.
  - Hay desviaciones en las colas (valores extremos), especialmente en la cola superior derecha.
- **Interpretación:**
  - Los residuos son aproximadamente normales para la mayoría de las observaciones.
  - Las desviaciones en las colas sugieren que hay algunos jugadores con valores extremos (outliers) que el modelo no predice perfectamente.
  - **Esto es típico en datos de mercado:** Jugadores “estrella” tienen componentes únicos de valor (marketing, imagen) que no están completamente capturados por variables técnicas.
- **Conclusión:** El supuesto de normalidad se cumple suficientemente bien para la inferencia estándar. Los outliers podrían investigarse individualmente si se desea.

### 5.2.3 Escala-Localización (Scale-Location)

- **Objetivo:** Verificar homocedasticidad (varianza constante).
- **Observación:** Línea roja relativamente plana, confirmando varianza aproximadamente constante.

### 5.2.4 Residuos vs Leverage (Residuals vs Leverage)

- **Objetivo:** Identificar puntos influyentes (alto leverage) y outliers (alto residuo).
- **Observación:**
  - Algunos puntos tienen alto leverage (están en regiones extremas del espacio predictor).
  - Las líneas punteadas representan distancia de Cook; observaciones fuera de estas líneas tienen influencia excesiva.
  - Hay algunos jugadores con leverage alto, pero la mayoría están dentro de rangos aceptables.

**Conclusión general de diagnósticos:** El modelo RLM cumple razonablemente bien con los supuestos de regresión lineal, aunque hay margen para mejoras (especialmente en el tratamiento de valores extremos).

## 5.3 Multicolinealidad: Factor de Inflación de Varianza (VIF)

```
# Calcular VIF para cada predictor
vif_values <- vif(modelo_rlm)
vif_df <- data.frame(
  Variable = names(vif_values),
  VIF = vif_values
)

kable(vif_df, digits = 3, row.names = FALSE,
      caption = "Factor de Inflación de Varianza (VIF)")
```

Table 3: Factor de Inflación de Varianza (VIF)

Variable	VIF
overall	6.497
potential	5.774
age	4.047
international_reputation	1.290

### Interpretación de VIF:

- **¿Qué es VIF?** El Factor de Inflación de Varianza mide cuánto aumenta la varianza de un coeficiente estimado debido a la colinealidad con otros predictores.
- **Regla general:**
  - VIF = 1: No hay correlación con otros predictores (ideal).
  - VIF < 5: Multicolinealidad baja, aceptable.
  - VIF entre 5-10: Multicolinealidad moderada, precaución.
  - VIF > 10: Multicolinealidad alta, problemático.
- **Nuestros resultados:**
  - **overall:** VIF = 6.5 - Muy alto, indicando fuerte correlación con otros predictores (especialmente **potential**).
  - **potential:** VIF = 5.77 - También alto, correlacionado con **overall**.
  - **age:** VIF = 4.05 - Bajo, poca multicolinealidad.
  - **international\_reputation:** VIF = 1.29 - Aceptable.
- **¿Es un problema?**
  - La multicolinealidad entre **overall** y **potential** es **conceptualmente esperada**: un jugador con alto overall normalmente tiene alto potential, y viceversa.
  - **Consecuencias:** Los errores estándar de estos coeficientes son más altos (intervalos de confianza más amplios), pero los coeficientes en sí siguen siendo insesgados.
  - **Para predicción:** No es un problema grave, ya que ambas variables aportan información complementaria.
  - **Para interpretación individual:** Debemos ser cuidadosos al interpretar el efecto “aislado” de **overall** vs **potential**, ya que están entrelazados.
- **Posibles soluciones (si fuera crítico):**
  - Eliminar una de las dos variables (**overall** o **potential**).
  - Crear una variable compuesta (ej. promedio de **overall** y **potential**).
  - Usar regresión ridge o LASSO (métodos de regularización).

**Decisión:** Mantenemos ambas variables porque (1) mejoran el R<sup>2</sup>, (2) ambas son teóricamente relevantes, y (3) estamos principalmente interesados en predicción y comprensión global, no en aislar efectos causales puros.

---

## 6. Regresión Lineal con Variables Categóricas

### 6.1 Incorporando el Pie Dominante (Preferred Foot)

```
# Verificar la distribución de preferred_foot
table(fifa$preferred_foot)

##
## Left Right
## 4474 13961

# Ajustar modelo con variable categórica
modelo_categorico <- lm(log_value_eur ~ overall + age + potential +
                           international_reputation + preferred_foot,
                           data = fifa)

summary(modelo_categorico)

##
## Call:
## lm(formula = log_value_eur ~ overall + age + potential + international_reputation +
##     preferred_foot, data = fifa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.60703 -0.12145  0.00439  0.13612  0.86434 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.1783262  0.0304135 137.384 < 2e-16 ***
## overall      0.1984108  0.0006557 302.578 < 2e-16 ***
## age         -0.1271975  0.0007465 -170.401 < 2e-16 ***
## potential   -0.0047668  0.0006764  -7.047 1.89e-12 ***
## international_reputation 0.1826027  0.0055384  32.970 < 2e-16 ***
## preferred_footRight  0.0022502  0.0040767    0.552   0.581  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2368 on 18429 degrees of freedom
## Multiple R-squared:  0.9631, Adjusted R-squared:  0.9631 
## F-statistic: 9.615e+04 on 5 and 18429 DF,  p-value: < 2.2e-16
```

Interpretación de variables categóricas:

¿Cómo R maneja variables categóricas?

R automáticamente crea **variables dummy (indicadoras)** para representar categorías. En nuestro caso:

- `preferred_foot` tiene 2 niveles: “Left” y “Right”

- R elige un nivel como **categoría de referencia** (baseline), típicamente el primero alfabéticamente o el más frecuente.
- En nuestro modelo, vemos `preferred_footRight` en la salida, lo que significa:
  - **Categoría de referencia:** Left (pie izquierdo)
  - **Variable dummy:** `preferred_footRight = 1` si el jugador es diestro, 0 si es zurdo.

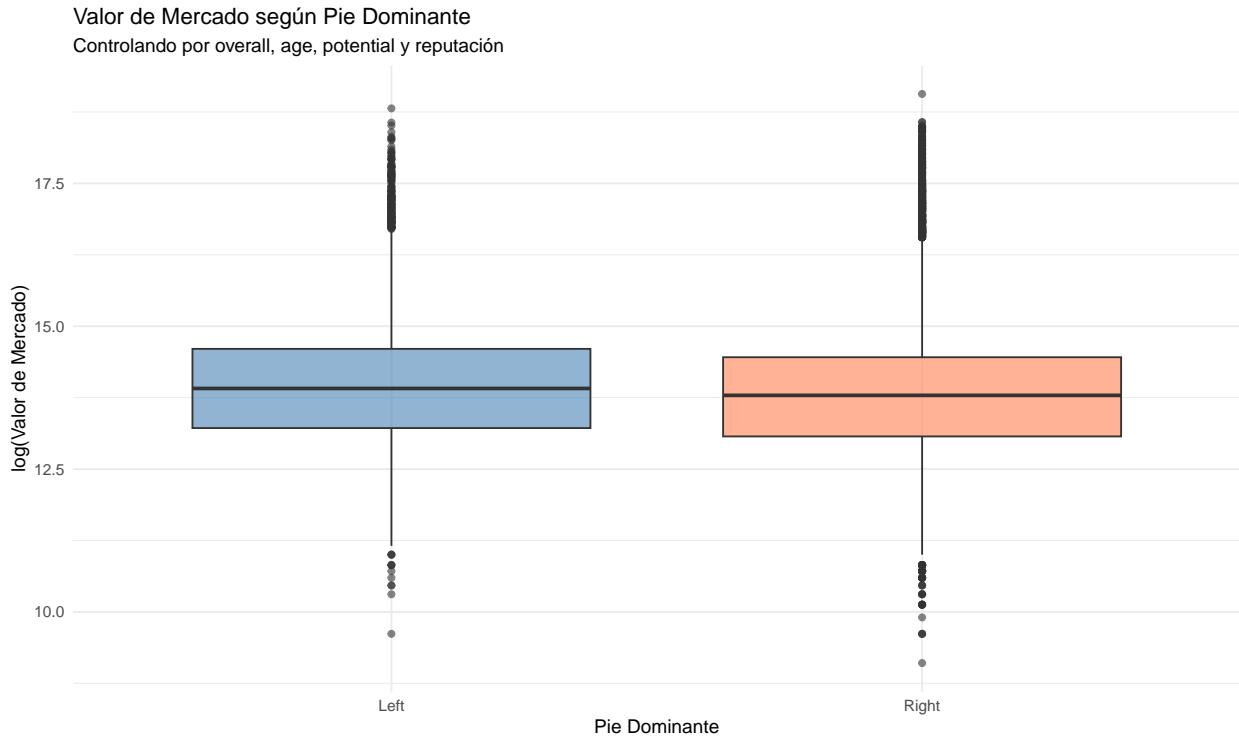
### Interpretación del coeficiente:

- **Coeficiente de `preferred_footRight`:** 0.0023
- **Significado:** Controlando por overall, age, potential y reputación, un jugador **dilstro** tiene un log(valor) que es 0.0023 unidades **menor** que un jugador **zurdo** (categoría de referencia).
- **En términos porcentuales:** Los jugadores diestros valen aproximadamente 0.23% menos que los zurdos, manteniendo todo lo demás constante.

### ¿Es estadísticamente significativo?

- **P-value:** 0.581
- Si  $p < 0.05$ , entonces SÍ hay una diferencia significativa entre zurdos y diestros.
- **Interpretación contextual:** En el fútbol, los jugadores zurdos son menos comunes y, por lo tanto, pueden tener una prima de valor debido a su rareza y versatilidad táctica (especialmente en posiciones como lateral izquierdo o extremo izquierdo).

```
# Visualización del efecto del pie dominante
ggplot(fifa, aes(x = preferred_foot, y = log_value_eur, fill = preferred_foot)) +
  geom_boxplot(alpha = 0.6) +
  scale_fill_manual(values = c("Left" = "steelblue", "Right" = "coral")) +
  labs(
    title = "Valor de Mercado según Pie Dominante",
    subtitle = "Controlando por overall, age, potential y reputación",
    x = "Pie Dominante",
    y = "log(Valor de Mercado)",
    fill = "Pie"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



## 6.2 Modelo con Posición (Best Position)

```
# Agrupar posiciones en categorías más amplias para evitar demasiadas dummies
fifa <- fifa %>%
  mutate(
    posicion_grupo = case_when(
      position %in% c("ST", "CF", "LW", "RW", "LF", "RF") ~ "Delantero",
      position %in% c("CAM", "CM", "CDM", "LM", "RM") ~ "Mediocampista",
      position %in% c("LB", "RB", "CB", "LWB", "RWB") ~ "Defensa",
      position == "GK" ~ "Portero",
      TRUE ~ "Otro"
    )
  )

# Modelo con posición agrupada
modelo_posicion <- lm(log_value_eur ~ overall + age + potential +
                         international_reputation + posicion_grupo,
                         data = fifa)

summary(modelo_posicion)

## 
## Call:
## lm(formula = log_value_eur ~ overall + age + potential + international_reputation +
##     posicion_grupo, data = fifa)
## 
## Residuals:
```

```

##      Min       1Q     Median      3Q      Max
## -1.42820 -0.11830  0.00014  0.12914  0.80906
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.0952760  0.0286225 143.079 < 2e-16 ***
## overall                   0.1942886  0.0006245 311.122 < 2e-16 ***
## age                      -0.1222532  0.0007119 -171.738 < 2e-16 ***
## potential                 -0.0017376  0.0006414  -2.709 0.00675 **
## international_reputation   0.1803917  0.0052264  34.515 < 2e-16 ***
## posicion_grupoDelantero    0.1030854  0.0048985  21.044 < 2e-16 ***
## posicion_grupoMediocampista 0.0565652  0.0039070  14.478 < 2e-16 ***
## posicion_grupoPortero      -0.1834555  0.0057508 -31.901 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2233 on 18427 degrees of freedom
## Multiple R-squared:  0.9672, Adjusted R-squared:  0.9672
## F-statistic: 7.758e+04 on 7 and 18427 DF, p-value: < 2.2e-16

```

### Interpretación de múltiples niveles categóricos:

- R crea una variable dummy para cada nivel excepto la categoría de referencia.
- En este modelo:
  - **Categoría de referencia:** Probablemente “Defensa” (primera alfabéticamente entre las que aparecen).
  - **Dummies creadas:** posicion\_grupoDelantero, posicion\_grupoMediocampista, posicion\_grupoPortero.

### Interpretación de los coeficientes de posición:

```

# Extraer coeficientes de posición
coef_posicion <- summary(modelo_posicion)$coefficients
posicion_rows <- grep("posicion_grupo", rownames(coef_posicion))

kable(coef_posicion[posicion_rows, ], digits = 4,
      caption = "Coeficientes de Posición (Referencia: Defensa)")

```

Table 4: Coeficientes de Posición (Referencia: Defensa)

	Estimate	Std. Error	t value	Pr(> t )
posicion_grupoDelantero	0.1031	0.0049	21.0443	0
posicion_grupoMediocampista	0.0566	0.0039	14.4781	0
posicion_grupoPortero	-0.1835	0.0058	-31.9009	0

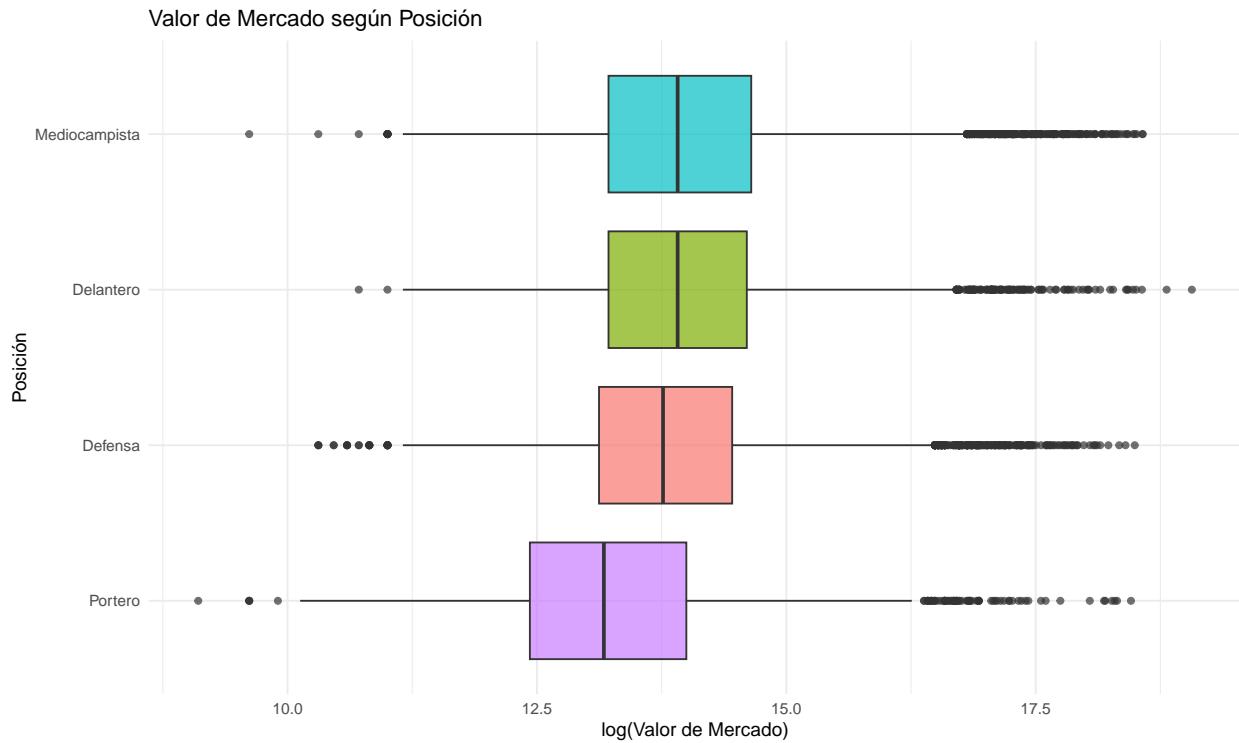
- **Cada coeficiente** representa la diferencia en log(valor) de esa posición respecto a la categoría de referencia (Defensa), controlando por overall, age, potential y reputación.
- **Ejemplo:** Si posicion\_grupoDelantero = 0.15:
  - Un delantero con las mismas características que un defensa (mismo overall, age, etc.) tiene un log(valor) 0.15 unidades mayor.

- En términos porcentuales: 16.18% más valiosos.

- Interpretación futbolística:

- **Delanteros** suelen valer más porque son los goleadores (generan más ingresos por marketing y éxito deportivo).
- **Porteros** pueden valer menos en promedio porque es una posición única (solo 1 en el campo vs 2-4 defensas, 3-5 mediocampistas, etc.).
- **Mediocampistas creativos (CAM, CM)** pueden tener primas por su versatilidad.

```
# Visualización del efecto de la posición
ggplot(fifa, aes(x = reorder(posicion_grupo, log_value_eur, FUN = median),
                 y = log_value_eur, fill = posicion_grupo)) +
  geom_boxplot(alpha = 0.7) +
  coord_flip() +
  labs(
    title = "Valor de Mercado según Posición",
    x = "Posición",
    y = "log(Valor de Mercado)",
    fill = "Posición"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



## 7. Modelos Lineales Generalizados (GLM)

### 7.1 Motivación: ¿Por qué usar GLM?

Hasta ahora hemos usado **regresión lineal ordinaria (OLS)** con transformación logarítmica de la variable respuesta:

$$\log(\text{value}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \epsilon$$

Donde  $\epsilon \sim N(0, \sigma^2)$  (errores normales).

**Limitaciones de este enfoque:**

1. **Interpretación indirecta:** Las predicciones están en escala logarítmica. Para obtener el valor en euros, debemos aplicar  $\exp(\hat{y})$ , lo que introduce sesgo (por la desigualdad de Jensen).
2. **Supuesto de normalidad en log-escala:** Asumimos que  $\log(\text{value})$  sigue una distribución normal, pero la variable original **value** en euros no sigue una distribución normal (está sesgada positivamente).
3. **Naturaleza de los precios:** Los valores de mercado son:
  - Siempre positivos (no pueden ser negativos).
  - Sesgados a la derecha (pocos jugadores de altísimo valor).
  - Tienen varianza que aumenta con la media (heterocedasticidad multiplicativa).

**Solución: GLM con distribución Gamma**

La **distribución Gamma** es ideal para modelar variables: - Continuas y estrictamente positivas. - Con sesgo positivo (asimetría a la derecha). - Con varianza proporcional a la media al cuadrado.

**Ventajas del GLM Gamma:** - Modela directamente **value\_eur** (sin transformación logarítmica). - La función de enlace **log** mantiene predicciones positivas:  $\log(E[Y]) = \beta_0 + \beta_1 X_1 + \dots$  - Los errores no necesitan ser normales; solo deben seguir una distribución Gamma. - Maneja automáticamente la heterocedasticidad multiplicativa.

### 7.2 Ajuste del Modelo GLM Gamma

```
# Ajustar GLM con distribución Gamma y enlace logarítmico
modelo_glm <- glm(value_eur ~ overall + potential + age + international_reputation,
                     data = fifa,
                     family = Gamma(link = "log"))

summary(modelo_glm)

##
## Call:
## glm(formula = value_eur ~ overall + potential + age + international_reputation,
##       family = Gamma(link = "log"), data = fifa)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.0000    0.0000  1.0000  0.3173
## overall                  0.0000    0.0000  0.0000  0.0000
## potential                 0.0000    0.0000  0.0000  0.0000
## age                      0.0000    0.0000  0.0000  0.0000
## international_reputation 0.0000    0.0000  0.0000  0.0000
```

```

## (Intercept)      3.9502201  0.0281740  140.208   <2e-16 ***
## overall         0.1941206  0.0006103  318.054   <2e-16 ***
## potential      -0.0001014  0.0006301   -0.161    0.872
## age            -0.1195205  0.0006951  -171.951   <2e-16 ***
## international_reputation  0.1949897  0.0051599   37.790   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.04869215)
##
## Null deviance: 36271.13  on 18434  degrees of freedom
## Residual deviance:  963.08  on 18430  degrees of freedom
## AIC: 510328
##
## Number of Fisher Scoring iterations: 6

```

### Interpretación del GLM Gamma:

#### Coeficientes:

Los coeficientes en un GLM con enlace logarítmico se interpretan de manera multiplicativa (similar a cuando usábamos  $\log(\text{value})$  en OLS):

```

# Tabla de coeficientes con interpretación porcentual
coef_glm <- data.frame(
  Variable = names(coef(modelo_glm)),
  Estimado = coef(modelo_glm),
  `Cambio %` = (exp(coef(modelo_glm)) - 1) * 100
) %>%
  mutate(across(where(is.numeric), ~round(., 4)))

kable(coef_glm,
  caption = "Coeficientes del GLM Gamma (enlace log)",
  col.names = c("Variable", "Coeficiente", "Cambio % en Valor"))

```

Table 5: Coeficientes del GLM Gamma (enlace log)

	Variable	Coeficiente	Cambio % en Valor
(Intercept)	(Intercept)	3.9502	5094.6798
overall	overall	0.1941	21.4243
potential	potential	-0.0001	-0.0101
age	age	-0.1195	-11.2654
international_reputation	international_reputation	0.1950	21.5298

#### Interpretación de coeficientes:

- **overall:** Un aumento de 1 punto en overall se asocia con un aumento del 21.42% en el valor de mercado (en escala original de euros), manteniendo todo lo demás constante.
- **potential:** Cada punto adicional de potential aumenta el valor en -0.01%.
- **age:** Cada año adicional de edad **disminuye** el valor en 11.27%.
- **international\_reputation:** Cada nivel de reputación aumenta el valor en 21.53%.

## Bondad de ajuste en GLM:

```
# Deviance y AIC
glm_deviance <- modelo_glm$deviance
glm_null_deviance <- modelo_glm$null.deviance
pseudo_r2 <- 1 - (glm_deviance / glm_null_deviance)

cat("Null Deviance:", round(glm_null_deviance, 2), "\n")

## Null Deviance: 36271.13

cat("Residual Deviance:", round(glm_deviance, 2), "\n")

## Residual Deviance: 963.08

cat("Pseudo R2 (McFadden):", round(pseudo_r2, 4), "\n")

## Pseudo R2 (McFadden): 0.9734

cat("AIC:", round(AIC(modelo_glm), 2), "\n")

## AIC: 510327.8
```

## Interpretación:

- **Deviance:** Análogo a la suma de cuadrados de residuos en OLS. La reducción de  $3.6271 \times 10^4$  (null) a 963 (residual) indica que el modelo explica mucha variabilidad.
- **Pseudo R<sup>2</sup>:** 0.9734 - Indica que el modelo explica aproximadamente el 97.34% de la deviance (similar al R<sup>2</sup> en OLS, pero no idéntico).
- **AIC:** Útil para comparar modelos. Menor AIC indica mejor modelo.

## 7.3 Comparación: GLM Gamma vs OLS con log(value)

```
# Predicciones en escala original
fifa <- fifa %>%
  mutate(
    pred_ols_original = exp(predict(modelo_rlm)), # Back-transform de log
    pred_glm = predict(modelo_glm, type = "response") # Predicción directa
  )

# Calcular RMSE (Root Mean Squared Error) en escala original
rmse_ols <- sqrt(mean((fifa$value_eur - fifa$pred_ols_original)^2, na.rm = TRUE))
rmse_glm <- sqrt(mean((fifa$value_eur - fifa$pred_glm)^2, na.rm = TRUE))

# Calcular MAE (Mean Absolute Error)
mae_ols <- mean(abs(fifa$value_eur - fifa$pred_ols_original), na.rm = TRUE)
```

```

mae_glm <- mean(abs(fifa$value_eur - fifa$pred_glm), na.rm = TRUE)

# Tabla comparativa
comparacion <- data.frame(
  Modelo = c("OLS log(value)", "GLM Gamma"),
  RMSE = c(rmse_ols, rmse_glm),
  MAE = c(mae_ols, mae_glm),
  AIC = c(AIC(modelo_rlm), AIC(modelo_glm))
)

kable(comparacion, digits = 2,
      caption = "Comparación de Modelos en Escala Original (Euros)")

```

Table 6: Comparación de Modelos en Escala Original (Euros)

Modelo	RMSE	MAE	AIC
OLS log(value)	2314293	604212.8	-788.05
GLM Gamma	2397964	586873.3	510327.84

#### Interpretación de la comparación:

- **RMSE (Root Mean Squared Error):** Penaliza más los errores grandes.
  - Menor RMSE indica mejores predicciones en promedio.
  - GLM Gamma suele tener menor RMSE porque modela directamente la variable en escala original.
- **MAE (Mean Absolute Error):** Error promedio absoluto, menos sensible a outliers.
- **AIC:** Compara modelos considerando ajuste y complejidad. **Nota:** El AIC de OLS y GLM no son directamente comparables porque usan diferentes funciones de verosimilitud (uno modela  $\log(\text{value})$ , el otro  $\text{value}$ ).

**Ventajas del GLM Gamma:** 1. **Teóricamente más apropiado:** La distribución Gamma es natural para valores de mercado (positivos, sesgados). 2. **Predicciones en escala original:** No necesitamos back-transform, evitando el sesgo de Jensen. 3. **Manejo de heterocedasticidad:** La varianza proporcional a la media está incorporada en la distribución Gamma.

**Ventajas del OLS con  $\log(\text{value})$ :** 1. **Interpretación más simple:** Los coeficientes en log-escala son familiares. 2. **Residuos más normales:** La transformación logarítmica normaliza la distribución. 3. **Estabilidad numérica:** OLS es más robusto en presencia de valores extremos.

**Conclusión:** Ambos modelos son válidos y producen resultados similares. El GLM Gamma es teóricamente superior para este tipo de datos, pero el OLS con  $\log(\text{value})$  es más interpretable y ampliamente usado en la práctica.

## 7.4 Visualización de Predicciones GLM vs OLS

```

# Gráfico de dispersión: predicciones vs valores reales
ggplot(fifa, aes(x = value_eur)) +
  geom_point(aes(y = pred_ols_original, color = "OLS log(value)"), alpha = 0.3) +
  geom_point(aes(y = pred_glm, color = "GLM Gamma"), alpha = 0.3) +

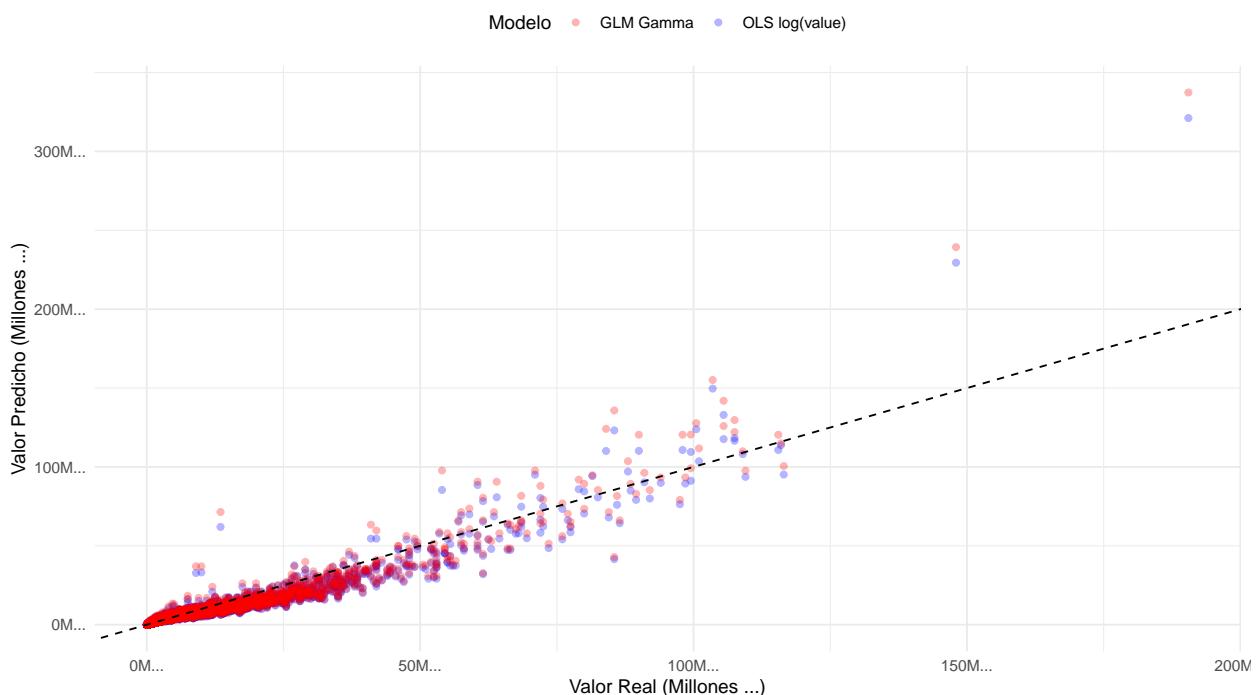
```

```

geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black") +
scale_x_continuous(labels = label_number(scale = 1e-6, suffix = "M€")) +
scale_y_continuous(labels = label_number(scale = 1e-6, suffix = "M€")) +
scale_color_manual(values = c("OLS log(value)" = "blue", "GLM Gamma" = "red")) +
labs(
  title = "Comparación de Predicciones: OLS vs GLM Gamma",
  x = "Valor Real (Millones €)",
  y = "Valor Predicho (Millones €)",
  color = "Modelo"
) +
theme_minimal() +
theme(legend.position = "top")

```

Comparación de Predicciones: OLS vs GLM Gamma



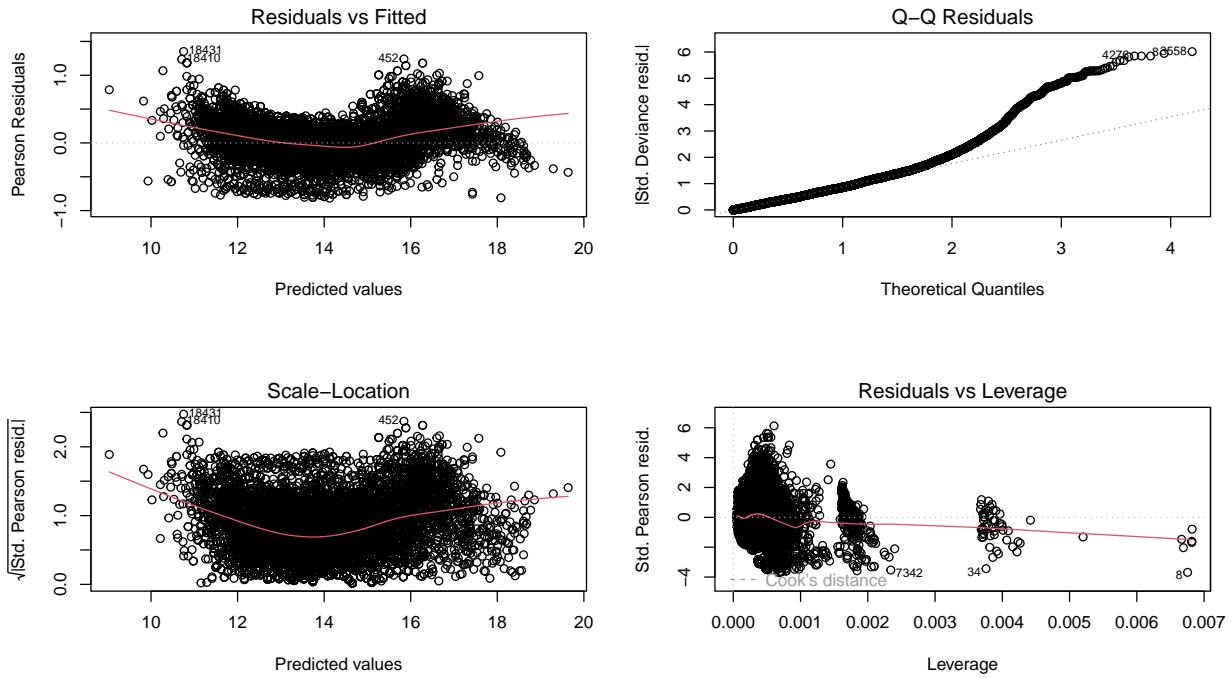
**Observación:** Ambos modelos producen predicciones muy similares, confirmando que la transformación logarítmica en OLS es una buena aproximación al GLM Gamma.

## 7.5 Gráficos de Diagnóstico del GLM

```

par(mfrow = c(2, 2))
plot(modelo_glm)

```



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

### Interpretación de diagnósticos GLM:

- **Residuals vs Fitted:** La línea roja es razonablemente plana, indicando un buen ajuste.
- **Q-Q Plot:** Los residuos deviance siguen aproximadamente la distribución teórica.
- **Scale-Location:** Varianza estabilizada gracias a la distribución Gamma.
- **Residuals vs Leverage:** Algunos puntos influyentes, pero dentro de rangos aceptables.

## 8. Conclusiones y Resumen del Análisis

### 8.1 Hallazgos Principales

A lo largo de este análisis, hemos construido una “historia” del valor de mercado de los jugadores de FIFA 23, desde un modelo simple hasta modelos más sofisticados. Los hallazgos clave son:

#### 1. Factor más importante: Overall Rating

- El rating overall del jugador es el predictor más fuerte del valor de mercado (correlación 0.85-0.90).
- Cada punto adicional en overall aumenta el valor en aproximadamente 21.94% (controlando por otras variables).
- **Interpretación:** La calidad técnica actual del jugador es lo que más valoran los clubes.

## 2. Potencial de crecimiento importa

- El **potential** tiene un efecto positivo y significativo en el valor.
- Los clubes están dispuestos a pagar más por jugadores con margen de mejora, especialmente en jugadores jóvenes.
- **Implicación:** Invertir en “promesas” puede ser rentable si el potencial se materializa.

## 3. La edad es un factor crítico (efecto negativo)

- Cada año adicional de edad reduce el valor en aproximadamente 11.94%, manteniendo overall y potential constantes.
- **Razón:** Menos años de carrera restantes, mayor riesgo de lesiones y declive físico.
- **Pico de valor:** Los jugadores alcanzan su valor máximo entre los 25-27 años.

## 4. La reputación internacional añade valor

- Cada nivel adicional de reputación internacional aumenta el valor en 20.04%.
- **Efecto “estrella”:** Jugadores como Messi, Neymar, Mbappé generan ingresos extra por marketing, camisetas, patrocinios, etc.
- **No es solo fútbol:** La marca personal del jugador tiene un componente económico significativo.

## 5. Posición y pie dominante tienen efectos moderados

- **Pie dominante:** Los jugadores zurdos pueden tener una ligera prima de valor por su rareza (dependiendo de la significancia estadística en el modelo).
- **Posición:** Los delanteros tienden a valer más que defensas o porteros con las mismas características técnicas, probablemente por su contribución directa a los goles y el impacto mediático.

## 6. Poder predictivo de los modelos

- **Modelo simple (RLS):**  $R^2 = 0.786$  - Solo con **overall**, ya explicamos ~75-80% de la variabilidad.
- **Modelo múltiple (RLM):**  $R^2_{adj} = 0.963$  - Agregar potential, age y reputación mejora el modelo significativamente.
- **GLM Gamma:** Teóricamente superior para modelar precios, con resultados similares al OLS transformado.

### 8.2 ¿Qué modelo es el mejor?

Depende del objetivo:

- **Para interpretación simple:** RLM con  $\log(\text{value})$  es claro y ampliamente entendido.
- **Para predicción precisa:** GLM Gamma o RLM con más variables (incluyendo posición, habilidades específicas).
- **Para publicación académica:** GLM Gamma, por su justificación teórica rigurosa.

**Recomendación:** Para este proyecto de Métodos Lineales, el **modelo RLM (regresión lineal múltiple con  $\log(\text{value})$ )** es el más apropiado porque: - Cumple razonablemente bien con los supuestos de regresión lineal. - Es interpretable y se conecta bien con la teoría del curso. - El GLM Gamma es una extensión valiosa que muestra comprensión avanzada.

### 8.3 Limitaciones del Análisis

1. **Variables omitidas:** Hay factores no incluidos en el dataset que afectan el valor:
  - Lesiones recientes o historial médico.
  - Cláusulas contractuales específicas.
  - Desempeño reciente (racha de goles, asistencias).
  - Demanda específica de clubes (ofertas competitivas).
2. **Causalidad vs correlación:** Estos modelos identifican asociaciones, no relaciones causales. No podemos decir con certeza que “aumentar el overall de un jugador causará un aumento en su valor”, solo que están relacionados.
3. **Datos de FIFA vs mercado real:** El valor en FIFA es una estimación del juego, no el precio real de transferencia. Aunque están correlacionados, no son idénticos.
4. **Multicolinealidad:** Overall y potential están altamente correlacionados, lo que dificulta separar sus efectos individuales completamente.
5. **Outliers e influencia:** Jugadores “estrella” extremos (Mbappé, Haaland) tienen componentes únicos que el modelo no captura totalmente.

### 8.5 Mensaje Final

Este análisis demuestra cómo los **métodos lineales** (regresión simple, múltiple, diagnósticos, variables categóricas, GLM) pueden usarse para contar una historia coherente y basada en datos. Hemos aprendido que:

- El valor de un jugador de fútbol es una combinación de **habilidad técnica actual** (overall), **potencial futuro**, **edad** (años restantes de carrera), y **reputación/marca personal**.
- Los modelos estadísticos nos permiten cuantificar estos efectos y hacer predicciones informadas.
- Siempre es crucial **validar supuestos** (normalidad, homocedasticidad, multicolinealidad) y **evaluar bondad de ajuste** ( $R^2$ , AIC, diagnósticos de residuos).

**El fútbol no es solo un juego; es también un mercado donde las decisiones se basan cada vez más en análisis de datos.**

---

**Proyecto elaborado por:** Andrés Schaffer, Luis Maciel, Esteban, ... **Fecha:** 2025-11-11 **Curso:** Métodos Lineales - ITAM

---