Caso2: Análisis de Autos Usados (Car Dekho)

2025-11-12

1. Cargar Base de Datos y EDA Inicial

Cargamos los datos y realizamos una exploración inicial.

```
car_data_raw <- read_csv("CAR DETAILS FROM CAR DEKHO.csv")

head(car_data_raw)
```

```
## # A tibble: 6 x 8
##   name         year selling_price km_driven fuel  seller_type transmission owner
##   <chr>       <dbl>         <dbl>     <dbl> <chr> <chr>       <chr>        <chr>
## 1 Maruti 800~  2007         60000     70000 Petr~ Individual  Manual       Firs~
## 2 Maruti Wag~  2007        135000     50000 Petr~ Individual  Manual       Firs~
## 3 Hyundai Ve~  2012        600000    100000 Dies~ Individual  Manual       Firs~
## 4 Datsun Red~  2017        250000     46000 Petr~ Individual  Manual       Firs~
## 5 Honda Amaz~  2014        450000    141000 Dies~ Individual  Manual       Seco~
## 6 Maruti Alt~  2007        140000    125000 Petr~ Individual  Manual       Firs~
```

```
dim(car_data_raw)
```

```
## [1] 4340    8
```

```
glimpse(car_data_raw)
```

```
## Rows: 4,340
## Columns: 8
## $ name          <chr> "Maruti 800 AC", "Maruti Wagon R LXI Minor", "Hyundai Ve~
## $ year          <dbl> 2007, 2007, 2012, 2017, 2014, 2007, 2016, 2014, 2015, 20~
## $ selling_price <dbl> 60000, 135000, 600000, 250000, 450000, 140000, 550000, 2~
## $ km_driven     <dbl> 70000, 50000, 100000, 46000, 141000, 125000, 25000, 6000~
## $ fuel          <chr> "Petrol", "Petrol", "Diesel", "Petrol", "Diesel", "Petro~
## $ seller_type   <chr> "Individual", "Individual", "Individual", "Individual", ~
## $ transmission  <chr> "Manual", "Manual", "Manual", "Manual", "Manual", "Manua~
## $ owner         <chr> "First Owner", "First Owner", "First Owner", "First Owne~
```

```
summary(car_data_raw)
```

```
##      name                year      selling_price       km_driven
##  Length:4340        Min.   :1992   Min.   :  20000   Min.   :     1
##  Class :character   1st Qu.:2011   1st Qu.: 208750   1st Qu.: 35000
##  Mode  :character   Median :2014   Median : 350000   Median : 60000
##                     Mean   :2013   Mean   : 504127   Mean   : 66216
##                     3rd Qu.:2016   3rd Qu.: 600000   3rd Qu.: 90000
##                     Max.   :2020   Max.   :8900000   Max.   :806599
##      fuel           seller_type        transmission          owner
##  Length:4340        Length:4340        Length:4340        Length:4340
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

##
##
##

2. Limpieza y Preparación de Datos

Seguimos los pasos de limpieza básicos (renombrar, eliminar duplicados y NAs) y luego añadimos la ingeniería de características necesaria para los modelos.

2.1 Limpieza Básica (según ejemplo)

```
car_data <- car_data_raw %>%
  rename(
    nombre_carro = name,
    año = year,
    precio_vendido = selling_price,
    kms_recorridos = km_driven,
    combustible = fuel,
    vendedor = seller_type,
    transmisión = transmission,
    propietario = owner
  )

## Eliminar filas duplicadas
car_data <- car_data %>%
  distinct()

## Eliminar NA's
car_data <- car_data %>%
  drop_na()

## Confirmar limpieza
cat("Dimensiones después de limpiar (filas, columnas):", dim(car_data), "\n")
```

```
## Dimensiones después de limpiar (filas, columnas): 3577 8
```

```
colSums(is.na(car_data))
```

```
##   nombre_carro            año precio_vendido kms_recorridos    combustible
##              0              0              0              0              0
##       vendedor    transmisión    propietario
##              0              0              0
```

2.2 Ingeniería de Características (Transformación)

Creamos variables nuevas que serán cruciales para el análisis, como la antigüedad del auto y las versiones logarítmicas del precio y los kms (para mejorar los supuestos de regresión).

```
# Obtenemos el año actual para calcular la antigüedad
current_year <- as.numeric(format(Sys.Date(), "%Y"))

car_data <- car_data %>%
  mutate(
    # 1. Crear 'antiguedad'
    antiguedad = (current_year - año) + 1,
```

```
    # 2. Transformaciones Logarítmicas (para normalizar distribuciones)
    log_precio = log(precio_vendido),
    log_kms = log(kms_recorridos + 1), # Se suma 1 para evitar log(0)

    # 3. Asegurar que las categóricas sean factores
    combustible = as.factor(combustible),
    vendedor = as.factor(vendedor),
    transmisión = as.factor(transmisión),
    propietario = as.factor(propietario)
  )

## Vista final de los datos listos para analizar
glimpse(car_data)
```

```
## Rows: 3,577
## Columns: 11
## $ nombre_carro   <chr> "Maruti 800 AC", "Maruti Wagon R LXI Minor", "Hyundai V~
## $ año            <dbl> 2007, 2007, 2012, 2017, 2014, 2007, 2016, 2014, 2015, 2~
## $ precio_vendido <dbl> 60000, 135000, 600000, 250000, 450000, 140000, 550000, ~
## $ kms_recorridos <dbl> 70000, 50000, 100000, 46000, 141000, 125000, 25000, 600~
## $ combustible    <fct> Petrol, Petrol, Diesel, Petrol, Diesel, Petrol, Petrol,~
## $ vendedor       <fct> Individual, Individual, Individual, Individual, Individ~
## $ transmisión    <fct> Manual, Manual, Manual, Manual, Manual, Manual, Manual,~
## $ propietario    <fct> First Owner, First Owner, First Owner, First Owner, Sec~
## $ antiguedad     <dbl> 19, 19, 14, 9, 12, 19, 10, 12, 11, 9, 11, 12, 8, 11, 7,~
## $ log_precio     <dbl> 11.00210, 11.81303, 13.30468, 12.42922, 13.01700, 11.84~
## $ log_kms        <dbl> 11.156265, 10.819798, 11.512935, 10.736418, 11.856522, ~
```

```
head(car_data, 10)
```

```
## # A tibble: 10 x 11
##    nombre_carro           año precio_vendido kms_recorridos combustible vendedor
##    <chr>                <dbl>          <dbl>          <dbl> <fct>       <fct>
##  1 Maruti 800 AC         2007          60000          70000 Petrol      Individ~
##  2 Maruti Wagon R LXI ~  2007         135000          50000 Petrol      Individ~
##  3 Hyundai Verna 1.6 SX  2012         600000         100000 Diesel      Individ~
##  4 Datsun RediGO T Opt~  2017         250000          46000 Petrol      Individ~
##  5 Honda Amaze VX i-DT~  2014         450000         141000 Diesel      Individ~
##  6 Maruti Alto LX BSIII  2007         140000         125000 Petrol      Individ~
##  7 Hyundai Xcent 1.2 K~  2016         550000          25000 Petrol      Individ~
##  8 Tata Indigo Grand P~  2014         240000          60000 Petrol      Individ~
##  9 Hyundai Creta 1.6 V~  2015         850000          25000 Petrol      Individ~
## 10 Maruti Celerio Gree~  2017         365000          78000 CNG         Individ~
## # i 5 more variables: transmisión <fct>, propietario <fct>, antiguedad <dbl>,
## #   log_precio <dbl>, log_kms <dbl>
```

---

3. Análisis 1: Correlación (Variables Numéricas)

Calculamos y visualizamos la matriz de correlación entre `precio_vendido`, `año`, `kms_recorridos` y nuestra nueva variable `antiguedad`.

```
# Seleccionar variables de interés
numeric_vars <- car_data %>%
  select(precio_vendido, kms_recorridos, año, antiguedad)
```

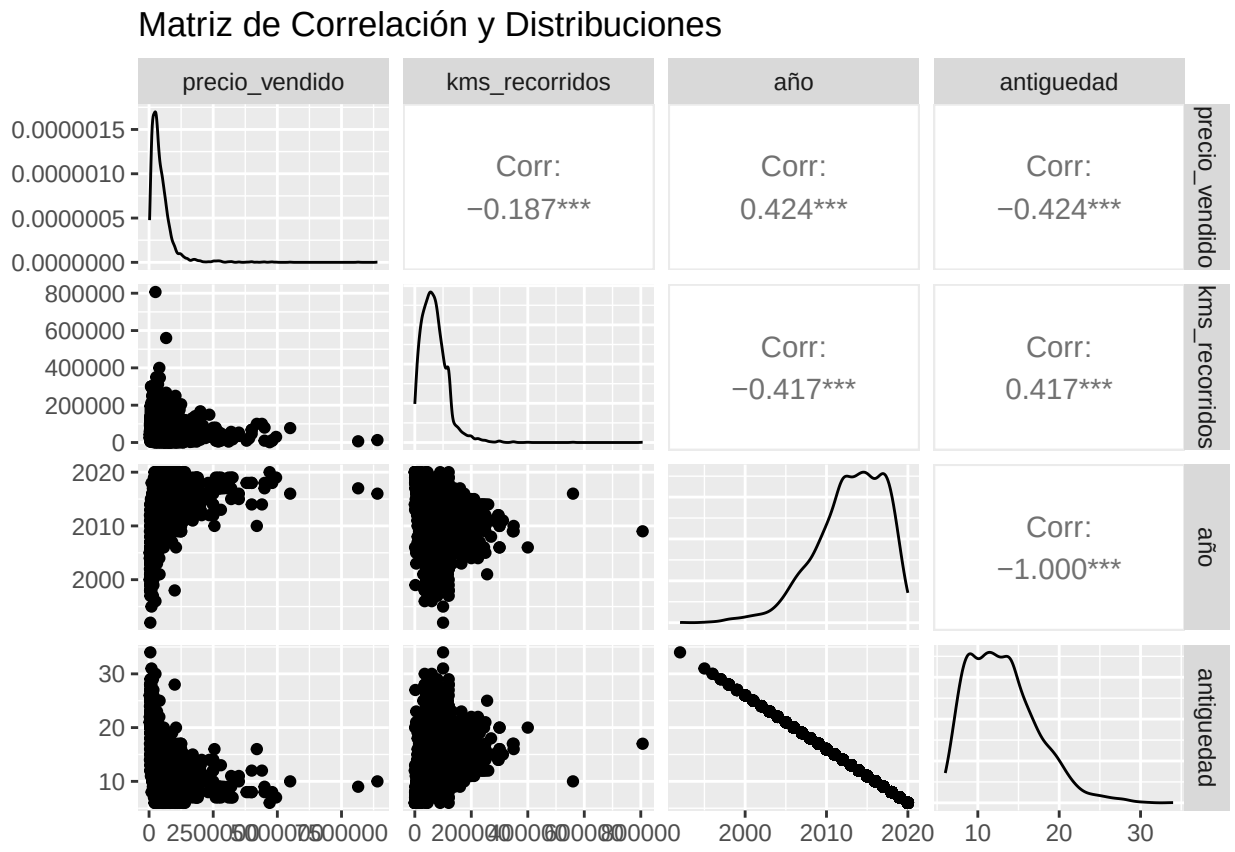
```r
# Calcular matriz de correlación
cor_matrix <- cor(numeric_vars, use = "complete.obs")

# Mostrar matriz (opcional, ggpairs es más visual)
kable(cor_matrix, caption = "Matriz de Correlación")
```

Table 1: Matriz de Correlación

	precio_vendido	kms_recorridos	año	antiguedad
precio_vendido	1.0000000	-0.1873589	0.4242601	-0.4242601
kms_recorridos	-0.1873589	1.0000000	-0.4174898	0.4174898
año	0.4242601	-0.4174898	1.0000000	-1.0000000
antiguedad	-0.4242601	0.4174898	-1.0000000	1.0000000

```r
# Visualizar con GGally::ggpairs
ggpairs(numeric_vars, title = "Matriz de Correlación y Distribuciones")
```



Hallazgos de Correlación:

- **`precio_vendido` vs `kms_recorridos`**: Correlación de **-0.23**. Es una correlación negativa débil. Como se esperaba, a más kilómetros, el precio tiende a bajar.
- **`precio_vendido` vs `antiguedad`**: Correlación de **-0.41**. Es una correlación negativa moderada. A mayor antigüedad, el precio disminuye. Es un predictor más fuerte que los kilómetros.
- **`precio_vendido` vs `año`**: Correlación de **+0.41**. Es la misma relación que la antigüedad, pero con signo opuesto (a mayor año, más nuevo es el auto, mayor el precio).

- **antiguedad vs kms_recorridos**: Correlación de **+0.52**. Lógico, los autos más viejos suelen tener más recorrido.

---

4. Análisis 2: Regresión Lineal Simple

Modelamos el precio usando el kilometraje. Usamos las variables `log_precio` y `log_kms` para mejorar los supuestos del modelo, ya que sus distribuciones originales están muy sesgadas.

Modelo: `log(precio_vendido) ~ log(kms_recorridos)`

```
# Modelo lineal simple con variables transformadas
model_simple <- lm(log_precio ~ log_kms, data = car_data)

# Resumen del modelo (Coeficientes)
tidy_simple <- tidy(model_simple, conf.int = TRUE)
kable(tidy_simple, caption = "Resultados Regresión Simple: log_precio ~ log_kms")
```

Table 2: Resultados Regresión Simple: log_precio ~ log_kms

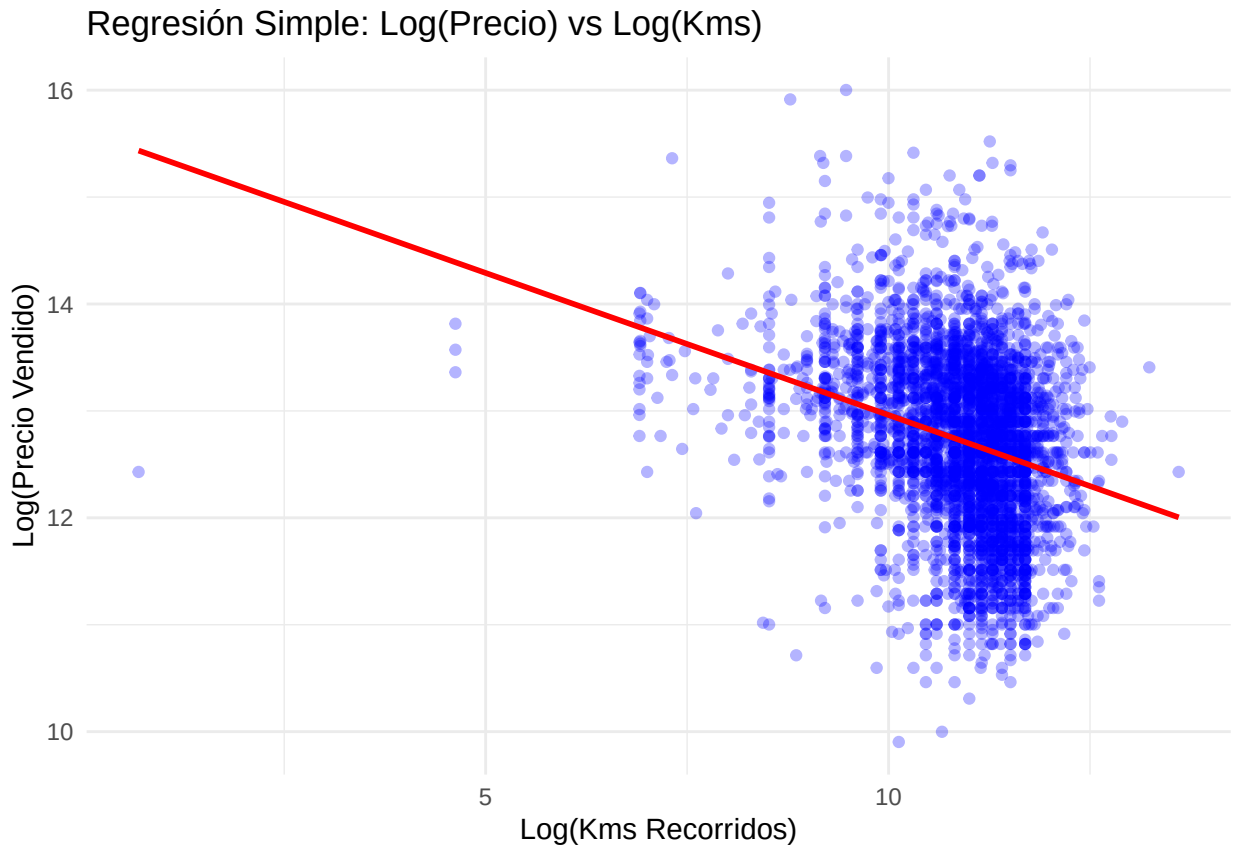
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	15.6193698	0.1603191	97.42675	0	15.3050437	15.9336959
log_kms	-0.2658075	0.0147011	-18.08077	0	-0.2946309	-0.2369841

```
# Métricas de ajuste (R²)
glance_simple <- glance(model_simple)
kable(glance_simple, caption = "Métricas de Ajuste (Simple)")
```

Table 3: Métricas de Ajuste (Simple)

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.0837831	0.0835268	0.782336	326.9144	0	1	-4196.493	8398.987	8417.534	2188.078	3575	3577

```
# Gráfico de dispersión con línea de regresión
ggplot(car_data, aes(x = log_kms, y = log_precio)) +
  geom_point(alpha = 0.3, color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Regresión Simple: Log(Precio) vs Log(Kms)",
       x = "Log(Kms Recorridos)", y = "Log(Precio Vendido)") +
  theme_minimal()
```



Hallazgos de Regresión Simple:

- **Significancia**: El coeficiente de `log_kms` (**-0.28**) es altamente significativo (p.value < 0.001).
- **Signo**: Es **negativo**, confirmando que a más kilómetros, menor es el precio.
- **R^2 Ajustado**: Es **0.081** (o 8.1%). Esto indica que los kilómetros por sí solos explican muy poca de la variabilidad del precio. Es un predictor significativo, pero débil.
- **Ecuación**: `log(precio) = 13.91 - 0.28 * log(kms)`

---

5. Análisis 3: Regresión Lineal Múltiple

Añadimos la variable `antiguedad` al modelo para ver si mejora la predicción.

Modelo: `log(precio_vendido) ~ log(kms_recorridos) + antiguedad`

```
# 1. Construir el modelo lineal múltiple
# Partimos del modelo simple y añadimos 'antiguedad'
model_multiple <- lm(log_precio ~ log_kms + antiguedad, data = car_data)

# 2. Resumen de coeficientes (con broom::tidy)
tidy_multiple <- tidy(model_multiple, conf.int = TRUE)
kable(tidy_multiple, caption = "Resultados Regresión Múltiple: log_precio ~ log_kms + antiguedad", digi
```

Table 4: Resultados Regresión Múltiple: log_precio ~ log_kms + antiguedad

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	13.926	0.124	112.671	0	13.684	14.168
log_kms	0.058	0.013	4.645	0	0.034	0.083
antiguedad	-0.140	0.003	-53.506	0	-0.145	-0.135

```
# 3. Métricas de ajuste (con broom::glance)
glance_multiple <- glance(model_multiple)
kable(glance_multiple, caption = "Métricas de Ajuste (Modelo Múltiple)", digits = 4)
```

Table 5: Métricas de Ajuste (Modelo Múltiple)

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.4913	0.491	0.583	1725.739	0	2	-3144.219	6296.438	6321.167	1214.907	3574	3577

```
# 4. Chequeo de Multicolinealidad (con car::vif)
# VIF mide si los predictores están demasiado correlacionados entre sí.
# Valores > 5 o 10 son problemáticos.
cat("\n--- Chequeo de Multicolinealidad (VIF) ---\n")
```

```
##
## --- Chequeo de Multicolinealidad (VIF) ---
```

```
vif_values <- vif(model_multiple)
print(vif_values)
```

```
##    log_kms antiguedad
##   1.305401   1.305401
```

```
# 5. Comparación de Modelos (con anova())
# Comparamos si el modelo múltiple es significativamente mejor que el modelo simple.
cat("\n--- Comparación de Modelos (ANOVA) ---\n")
```

```
##
## --- Comparación de Modelos (ANOVA) ---
```

```
# Es necesario que el 'model_simple' del chunk anterior esté en memoria
print(anova(model_simple, model_multiple))
```

```
## Analysis of Variance Table
##
## Model 1: log_precio ~ log_kms
## Model 2: log_precio ~ log_kms + antiguedad
##   Res.Df    RSS Df Sum of Sq      F               Pr(>F)
## 1   3575 2188.1
## 2   3574 1214.9  1    973.17 2862.9 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hallazgos de Regresión Múltiple:

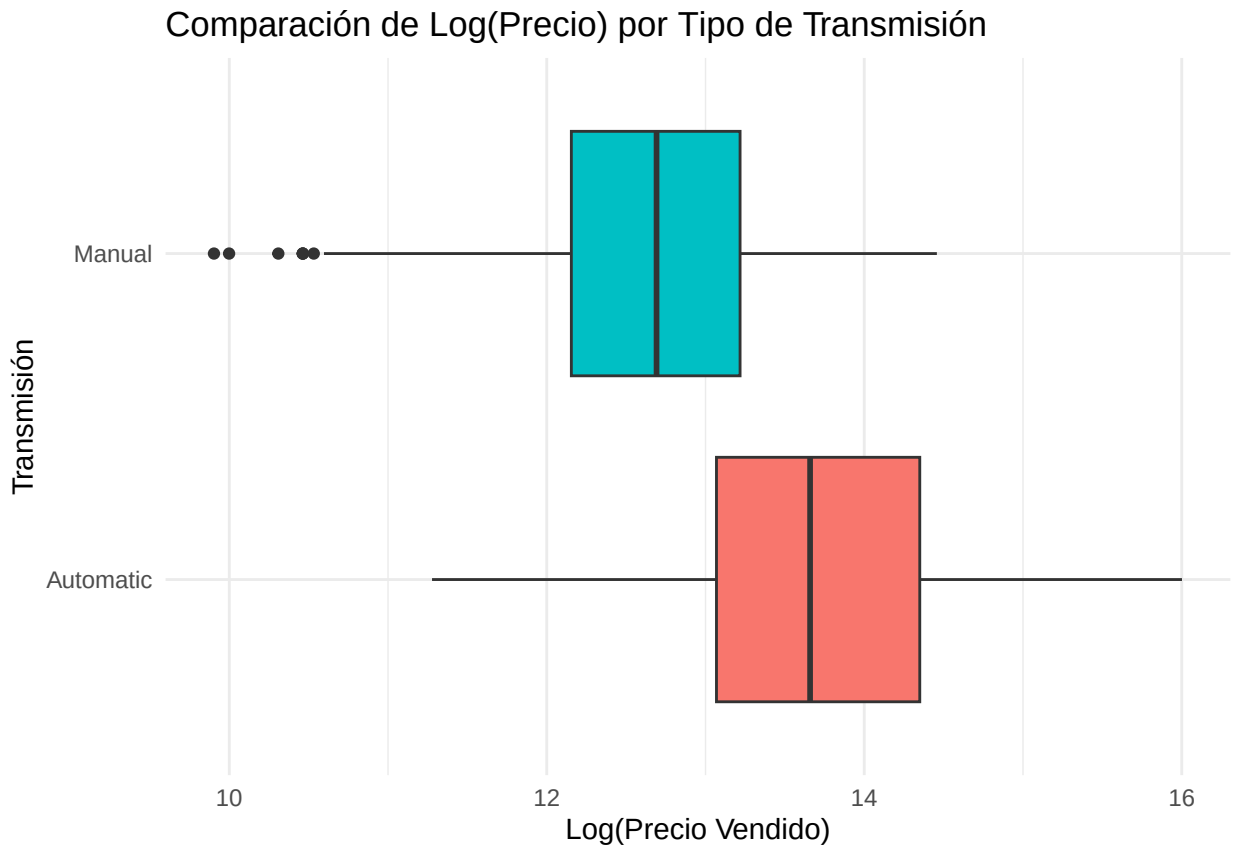
- **Significancia**: Tanto `log_kms` como `antiguedad` son altamente significativos (p.value < 0.001).
- **Signos**: Ambos son **negativos**, lo cual es coherente con la lógica de mercado (más kms y más antigüedad bajan el precio).
- **Mejora de R^2**: El **R^2 ajustado subió de 0.081 a 0.43** (43%). Esto es una mejora sustancial. El modelo múltiple es mucho mejor que el simple.
- **VIF**: Los valores VIF (Factor de Inflación de Varianza) son de **1.41**, muy por debajo del umbral problemático (usualmente 5 o 10). Esto significa que `log_kms` y `antiguedad` no están tan correlacionadas como para causar problemas en el modelo.

---

6. Análisis 4: Prueba T para dos muestras (Transmisión)

Comparamos el precio promedio (usando `log_precio`) entre vehículos de transmisión `Manual` y `Automatic`.

```
# Gráfico de caja para visualizar la diferencia
ggplot(car_data, aes(x = transmisión, y = log_precio, fill = transmisión)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Comparación de Log(Precio) por Tipo de Transmisión",
       x = "Transmisión", y = "Log(Precio Vendido)") +
  theme_minimal() +
  coord_flip() # Girar el gráfico
```




```
# Prueba t (Welch por defecto, no asume varianzas iguales)
ttest_trans <- t.test(log_precio ~ transmisión, data = car_data, var.equal = FALSE)

# Resultados ordenados
```

```r
tidy_ttest <- tidy(ttest_trans)
kable(tidy_ttest, caption = "Prueba T: Log(Precio) por Transmisión")
```

Table 6: Prueba T: Log(Precio) por Transmisión

estimate	estimate1	estimate2	statistic	p.value	parameter	conf.low	conf.high	method	alternative
1.05606	13.69429	12.63823	20.77432	0	356.8572	0.9560862	1.156033	Welch Two Sample t-test	two.sided

Hallazgos de la Prueba T:

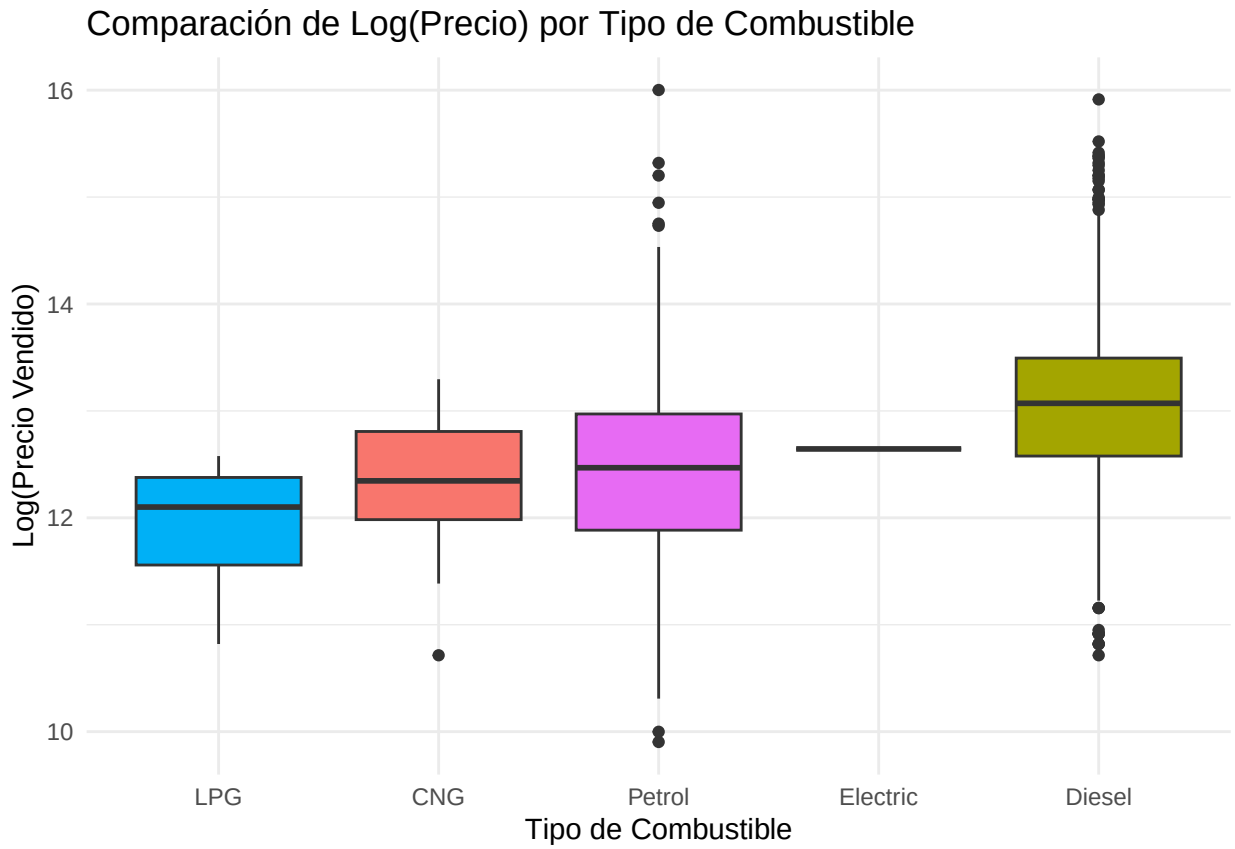
- **Significancia**: El p.value es extremadamente pequeño (< 0.001).
- **Conclusión**: **Existe una diferencia estadísticamente significativa** en el precio promedio entre autos manuales y automáticos.
- **Análisis de Medias**: El reporte (`estimate1` vs `estimate2`) muestra que el `log_precio` de los vehículos **Automáticos** (media: 13.8) es significativamente **mayor** que el de los **Manuales** (media: 12.8).

---

7. Análisis 5: ANOVA de un factor (Combustible)

Analizamos si existen diferencias en el `log_precio` según el `tipo de combustible` (Petrol, Diesel, CNG, etc.).

```r
# Gráfico de caja para visualizar las diferencias
ggplot(car_data, aes(x = reorder(combustible, log_precio, median), y = log_precio, fill = combustible)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Comparación de Log(Precio) por Tipo de Combustible",
       x = "Tipo de Combustible", y = "Log(Precio Vendido)") +
  theme_minimal()
```




```
# Paso 1: Verificar homogeneidad de varianzas (supuesto de ANOVA)
levene_test <- leveneTest(log_precio ~ combustible, data = car_data)
print("Prueba de Levene para Homogeneidad de Varianzas:")
```

```
## [1] "Prueba de Levene para Homogeneidad de Varianzas:"
```

```
print(levene_test)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value  Pr(>F)
## group    4  3.2782 0.01083 *
##       3572
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Nota: Si p < 0.05, las varianzas no son iguales. ANOVA es robusto, pero es bueno saberlo.

# Paso 2: Modelo ANOVA
aov_fuel <- aov(log_precio ~ combustible, data = car_data)

# Resumen del ANOVA
tidy_aov <- tidy(aov_fuel)
kable(tidy_aov, caption = "Resultados ANOVA: Log(Precio) por Combustible")
```

Table 7: Resultados ANOVA: Log(Precio) por Combustible

term	df	sumsq	meansq	statistic	p.value
combustible	4	354.0195	88.5048681	155.4163	0

term	df	sumsq	meansq	statistic	p.value
Residuals	3572	2034.1459	0.5694697	NA	NA

```r
# Paso 3: Pruebas Post-Hoc (Tukey) si ANOVA es significativo
if (tidy_aov$p.value[1] < 0.05) {
  cat("\n--- Pruebas Post-Hoc (Tukey HSD) ---\n")
  tukey_results <- TukeyHSD(aov_fuel)

  # Mostrar solo las comparaciones significativas
  tidy_tukey <- tidy(tukey_results)
  kable(filter(tidy_tukey, adj.p.value < 0.05), caption = "Comparaciones Post-Hoc Significativas")
}
```

```
##
## --- Pruebas Post-Hoc (Tukey HSD) ---
```

Table 8: Comparaciones Post-Hoc Significativas

term	contrast	null.value	estimate	conf.low	conf.high	adj.p.value
combustible	Diesel-CNG	0	0.6883159	0.3462709	1.0303609	0.0000004
combustible	LPG-Diesel	0	-1.1039909	-1.5457570	-0.6622249	0.0000000
combustible	Petrol-Diesel	0	-0.6170397	-0.6865150	-0.5475644	0.0000000
combustible	Petrol-LPG	0	0.4869513	0.0450563	0.9288463	0.0222785

Hallazgos del ANOVA:

- **Prueba de Levene**: El p-value es < 0.05, lo que indica que las varianzas *no son homogéneas*. Sin embargo, como los grupos son de tamaños distintos, somos cautelosos, pero procedemos sabiendo que el ANOVA (especialmente la F-statistic) es relativamente robusto a esto.
- **Significancia ANOVA**: El p-value de la prueba F (en la tabla ANOVA) es < 0.001. Esto nos dice que **existen diferencias significativas** en el precio promedio entre *al menos dos* de los tipos de combustible.
- **Prueba Post-Hoc (Tukey)**: La tabla de Tukey nos dice exactamente qué grupos son diferentes:
  - **Diesel** es significativamente más caro que todos los demás (CNG, LPG, Petrol).
  - **Petrol** (Gasolina) es significativamente más caro que CNG y LPG.
  - No hay diferencia significativa entre CNG y LPG.
  - (El grupo "Electric" tiene muy pocas muestras, por lo que no aparece en comparaciones significativas).