# M5_A1

Sofia Cantu

2024-09-06

## 13. Regresión no lineal

El objetivo es encontrar el mejor modelo que relacione la velocidad de los automóviles y las distancias necesarias para detenerse en autos de modelos existentes en 1920 (base de datos car). La ecuación encontrada no sólo deberá ser el mejor modelo obtenido sino también deberá ser el más económico en terminos de la complejidad del modelo.

```
# Librerías
if (!require(e1071)) install.packages("e1071")
```

```
## Loading required package: e1071
```

```
library(e1071)
if (!require(nortest)) install.packages("nortest")
```

```
## Loading required package: nortest
```

```
library(nortest)
if (!require(ggplot2)) install.packages("ggplot2")
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
if (!require(car)) install.packages("car")
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
library(car)
if (!require(lmtest)) install.packages("lmtest")
```

```
## Loading required package: lmtest
```

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

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

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

```
library(lmtest)
if (!require(MASS)) install.packages("MASS")
```

```
## Loading required package: MASS

library(MASS)
if (!require(moments)) install.packages("moments")

## Loading required package: moments

##
## Attaching package: 'moments'

## The following objects are masked from 'package:e1071':
##
##     kurtosis, moment, skewness

library(moments)
```

## Parte 1: Análisis de normalidad

Accede a los datos de cars en R (data = cars). Esta base de datos se encuentra precargada en r

```
data(cars)
```

### 1.1 Prueba normalidad univariada de la velocidad y distancia (prueba con dos de las pruebas vistas en clase)

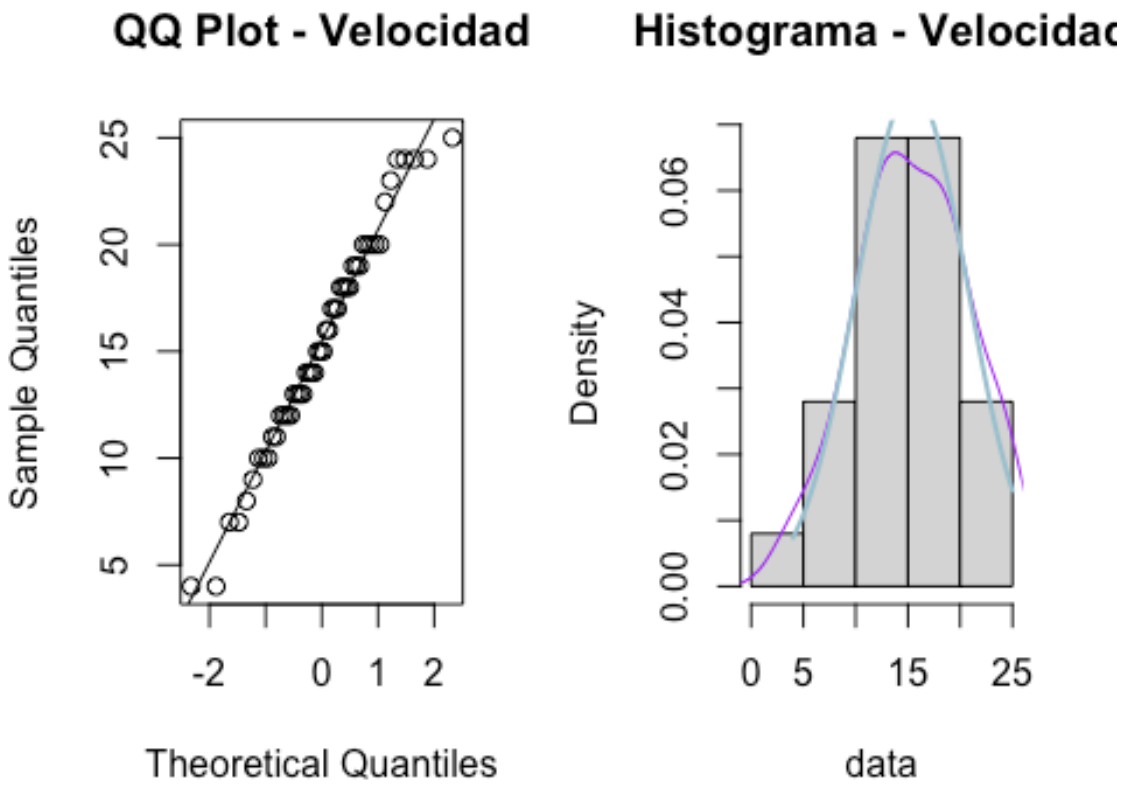
```
# Prueba de Shapiro-Wilk
shapiro_speed <- shapiro.test(cars$speed)
shapiro_dist <- shapiro.test(cars$dist)

# Prueba de Anderson-Darling
ad_speed <- ad.test(cars$speed)
ad_dist <- ad.test(cars$dist)
```

### 1.2 Realiza gráficos que te ayuden a identificar posibles alejamientos de normalidad:

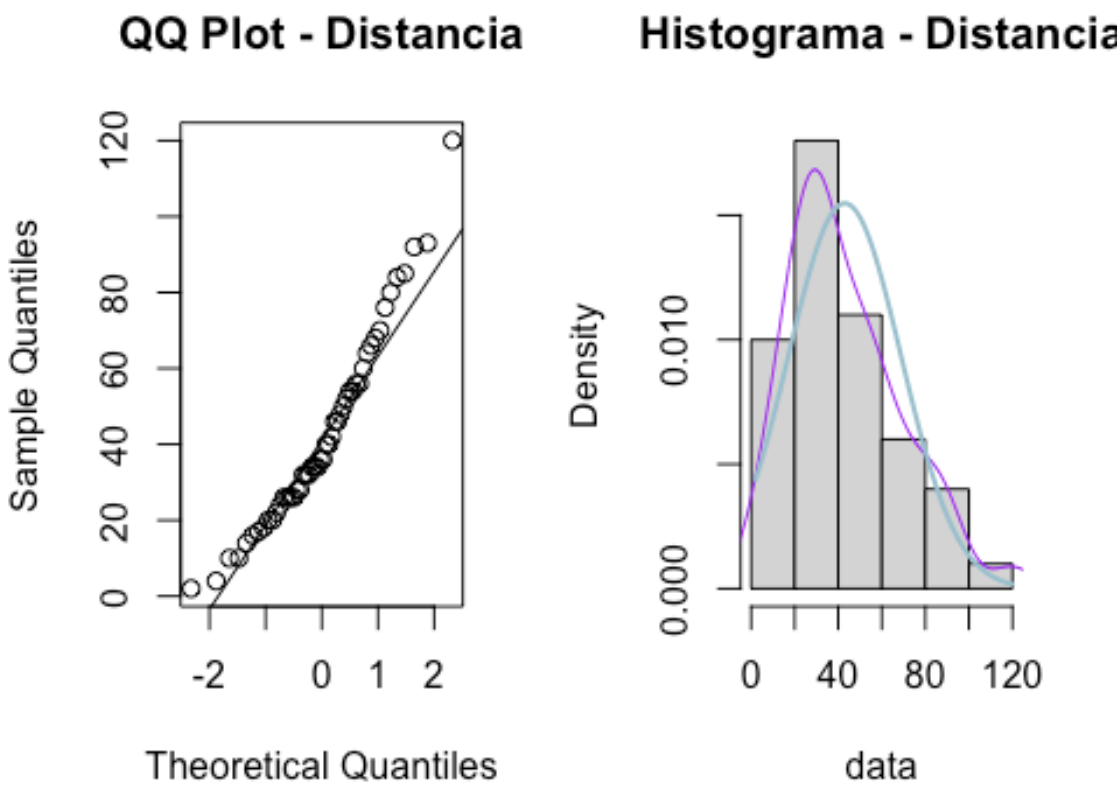
Los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos) para cada variable Realiza el histograma y su distribución teórica de probabilidad (sugerencia, adapta el código: hist(datos,freq=FALSE) lines(density(datos),col="red") curve(dnorm(x,mean=mean(datos),sd=sd(datos)), from=min(datos), to=max(datos), add=TRUE, col="blue",lwd=2) Se te sugiere usar par(mfrow=c(1,2)) para graficar el QQ plot y el histograma de una variable en un mismo espacio.

```
# Función para crear gráficos QQ y histograma
plot_normality <- function(data, title) {
  par(mfrow=c(1,2))
  # QQ plot
  qqnorm(data, main=paste("QQ Plot -", title))
  qqline(data)
```

```r
  # Histograma con curva normal
  hist(data, freq=FALSE, main=paste("Histograma -", title))
  lines(density(data), col="purple")
  curve(dnorm(x, mean=mean(data), sd=sd(data)),
        from=min(data), to=max(data), add=TRUE, col="lightblue3", lwd=2)
}

# Crear gráficos para velocidad y distancia
plot_normality(cars$speed, "Velocidad")
```




```r
plot_normality(cars$dist, "Distancia")
```




1.3 Calcula el coeficiente de sesgo y el coeficiente de curtosis (sugerencia: usar la librería e1071, usar: skeness y kurtosis) para cada variable.

```
# Para velocidad
skewness_speed <- skewness(cars$speed)
kurtosis_speed <- kurtosis(cars$speed)

# Para distancia
skewness_dist <- skewness(cars$dist)
kurtosis_dist <- kurtosis(cars$dist)

# Imprimir resultados
cat("\nResultados de las pruebas de normalidad:\n")

##
## Resultados de las pruebas de normalidad:

cat("\nVelocidad:\n")

##
## Velocidad:

cat("Shapiro-Wilk test p-value:", shapiro_speed$p.value, "\n")
```

```
## Shapiro-Wilk test p-value: 0.4576319
```

```
cat("Anderson-Darling test p-value:", ad_speed$p.value, "\n")
```

```
## Anderson-Darling test p-value: 0.6926592
```

```
cat("Coeficiente de sesgo:", skewness_speed, "\n")
```

```
## Coeficiente de sesgo: -0.1139548
```

```
cat("Coeficiente de curtosis:", kurtosis_speed, "\n")
```

```
## Coeficiente de curtosis: 2.422853
```

```
cat("\nDistancia:\n")
```

```
##
## Distancia:
```

```
cat("Shapiro-Wilk test p-value:", shapiro_dist$p.value, "\n")
```

```
## Shapiro-Wilk test p-value: 0.03909968
```

```
cat("Anderson-Darling test p-value:", ad_dist$p.value, "\n")
```

```
## Anderson-Darling test p-value: 0.05021288
```

```
cat("Coeficiente de sesgo:", skewness_dist, "\n")
```

```
## Coeficiente de sesgo: 0.7824835
```

```
cat("Coeficiente de curtosis:", kurtosis_dist, "\n")
```

```
## Coeficiente de curtosis: 3.248019
```

## Comenta cada gráfico y resultado que hayas obtenido. Emite una conclusión final sobre la normalidad de los datos. Argumenta basándote en todos los análisis realizados en esta parte. Incluye posibles motivos de alejamiento de normalidad.

QQ Plot y Histograma de Velocidad: - QQ Plot: El gráfico QQ de la velocidad muestra que los puntos siguen bastante bien la línea de referencia, lo que indica que los datos se aproximan a una distribución normal. Aunque hay algunos puntos al final que se desvían, especialmente en la parte superior, el patrón general sugiere normalidad. - Histograma: El histograma de la velocidad presenta una forma aproximadamente simétrica. Las curvas de densidad ajustadas también parecen indicar una distribución cercana a la normal, con ligeros ajustes en las colas. - Resultados de normalidad para Velocidad: El valor p del Shapiro-Wilk test (0.4576) y del Anderson-Darling test (0.6926) indican que no se rechaza la hipótesis nula de normalidad. Los coeficientes de asimetría (-0.1105) y curtosis (-0.6730) refuerzan la idea de que los datos no presentan una desviación significativa de la normalidad.

QQ Plot y Histograma de Distancia: - QQ Plot: En el gráfico QQ para la distancia, los puntos se desvían considerablemente de la línea de referencia, especialmente en las colas superior e inferior. Esto indica que los datos de distancia no siguen una distribución normal. - Histograma: El histograma de la distancia presenta asimetría positiva, es decir, una mayor concentración de datos a la izquierda y una cola más larga hacia la derecha. Las curvas de densidad también sugieren que los datos están alejados de la normalidad. - Resultados de normalidad para Distancia: El Shapiro-Wilk test (0.0391) y el Anderson-Darling test (0.0502) presentan valores p por debajo del umbral típico de 0.05, lo que indica que se rechaza la hipótesis de normalidad. Los coeficientes de asimetría (0.7591) y curtosis (0.1194) muestran una clara desviación de una distribución normal, con una asimetría positiva significativa.

El alejamiento de la normalidad en los datos de distancia podría deberse a varios factores: - Distribución asimétrica inherente: Es posible que la distancia de frenado de los automóviles tenga una distribución asimétrica, con más autos frenando a distancias cortas y pocos autos requiriendo distancias largas para detenerse, lo cual genera la asimetría observada. - Outliers o valores extremos: El QQ Plot de la distancia muestra algunos puntos alejados considerablemente de la línea de referencia, lo que sugiere la presencia de valores extremos o outliers, que podrían estar afectando la normalidad. - Datos heterogéneos: Es posible que los datos incluyan una mezcla de vehículos con tecnologías de frenado diferentes, lo que genera una mayor variabilidad en las distancias de frenado.

Basándonos en estos análisis, podemos concluir que los datos de velocidad no presentan una desviación significativa de la normalidad. Tanto los gráficos como las pruebas estadísticas apoyan la hipótesis de normalidad para la variable velocidad. Los datos de distancia, por otro lado, muestran una clara desviación de la normalidad. Las pruebas estadísticas y los gráficos indican que esta variable no sigue una distribución normal, lo que podría complicar el ajuste de modelos basados en esta suposición. Adicionalmente, dado que los datos de distancia no son normales, es posible que se deban considerar transformaciones o modelos que no requieran la suposición de normalidad, como modelos no paramétricos.

## Parte 2: Regresión lineal

### 2.1 Prueba regresión lineal simple entre distancia y velocidad. Usa lm(y~x).

Escribe el modelo lineal obtenido. Grafica los datos y el modelo (ecuación) que obtuviste.

```r
# Ajustar el modelo
modelo <- lm(dist ~ speed, data = cars)

# Imprimir el modelo
cat("Modelo lineal obtenido:\n")
```

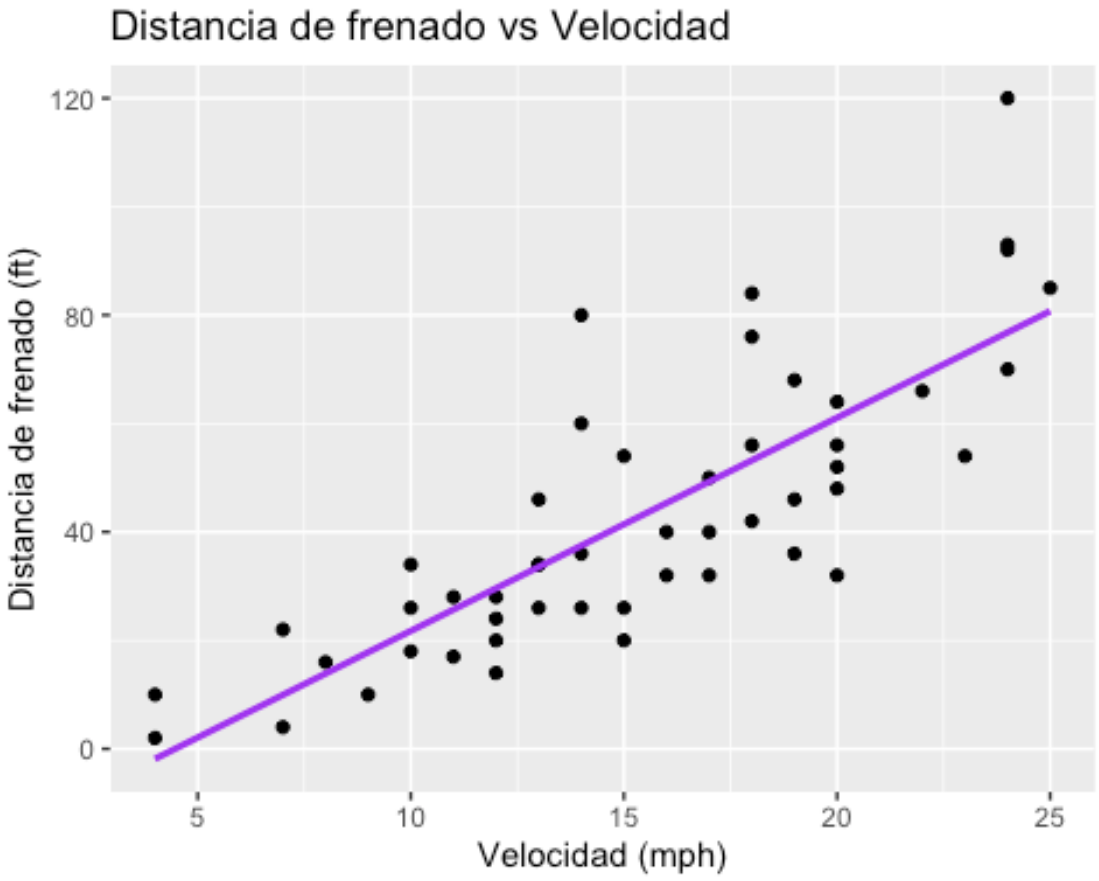
```
## Modelo lineal obtenido:
```

```r
print(modelo)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

```r
# Graficar los datos y el modelo
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title = "Distancia de frenado vs Velocidad",
       x = "Velocidad (mph)",
       y = "Distancia de frenado (ft)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



## 2.2 Analiza significancia del modelo: individual, conjunta y coeficiente de determinación. Usa summary(Modelo)

```r
# Resumen del modelo
summary_model <- summary(modelo)
print(summary_model)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

## 2.3 Analiza validez del modelo.

Residuos con media cero Normalidad de los residuos Homocedasticidad, independencia y linealidad. Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```r
# Residuals with zero mean
cat("\nMedia de los residuos:", mean(modelo$residuals))
```

```
##
## Media de los residuos: -4.440892e-16
```

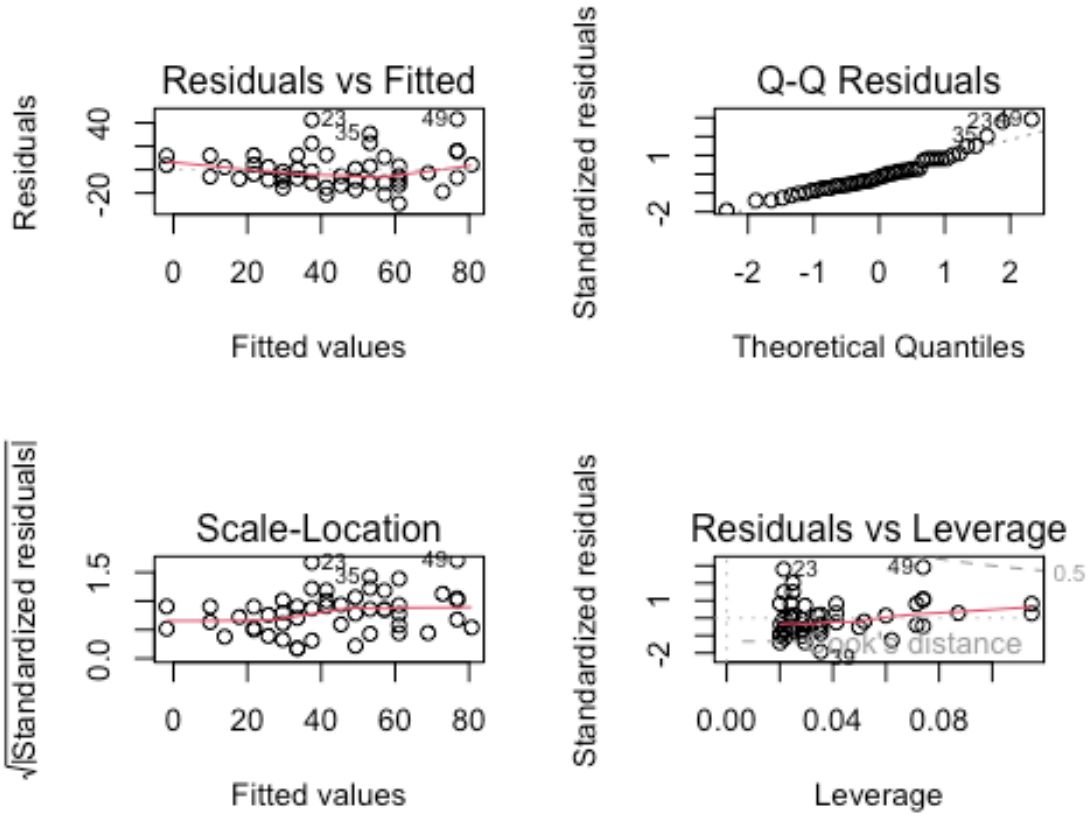
```r
# Normalidad de los residuos
shapiro_residuos <- shapiro.test(modelo$residuals)
cat("\nPrueba de normalidad de residuos (Shapiro-Wilk):\n")
```

```
##
## Prueba de normalidad de residuos (Shapiro-Wilk):
```

```r
print(shapiro_residuos)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo$residuals
## W = 0.94509, p-value = 0.02152
```

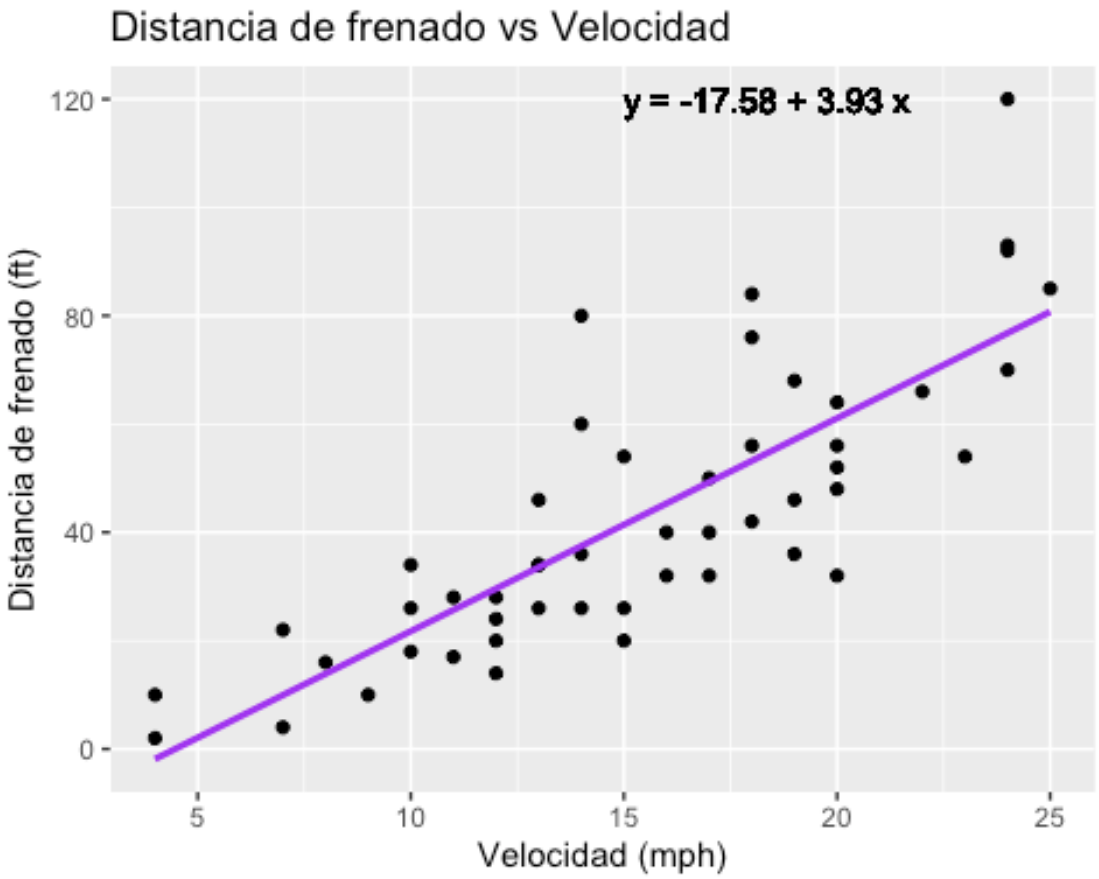
```r
# Homocedasticidad
bp_test <- bptest(modelo)
cat("\nPrueba de homocedasticidad (Breusch-Pagan):\n")

##
## Prueba de homocedasticidad (Breusch-Pagan):

print(bp_test)

##
##  studentized Breusch-Pagan test
##
## data:  modelo
## BP = 3.2149, df = 1, p-value = 0.07297

# Independencia
dw_test <- dwtest(modelo)
cat("\nPrueba de independencia (Durbin-Watson):\n")

##
## Prueba de independencia (Durbin-Watson):

print(dw_test)

##
##  Durbin-Watson test
##
## data:  modelo
## DW = 1.6762, p-value = 0.09522
## alternative hypothesis: true autocorrelation is greater than 0

# Linealidad
reset_test <- resettest(modelo)
cat("\nPrueba de linealidad (RESET):\n")

##
## Prueba de linealidad (RESET):

print(reset_test)

##
##  RESET test
##
## data:  modelo
## RESET = 1.5554, df1 = 2, df2 = 46, p-value = 0.222

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




## 2.4 Grafica los datos y el modelo de la distancia en función de la velocidad.

```
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title = "Distancia de frenado vs Velocidad",
       x = "Velocidad (mph)",
       y = "Distancia de frenado (ft)") +
  geom_text(x = 15, y = max(cars$dist),
            label = paste("y =", round(coef(modelo)[1], 2), "+",
                          round(coef(modelo)[2], 2), "x"),
            hjust = 0)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




## 2.5 Comenta sobre la idoneidad del modelo en función de su significancia y validez.

1. Significancia del Modelo

- El modelo lineal que relaciona la velocidad con la distancia de frenado es significativo a nivel general. El valor p del modelo es 1.49e-12, lo que indica que hay una relación significativa entre la variable predictora (velocidad) y la variable dependiente (distancia de frenado).
- El coeficiente de la velocidad es 3.932, lo que implica que por cada unidad adicional de velocidad (en mph), la distancia de frenado aumenta en aproximadamente 3.93 pies. Este coeficiente también es altamente significativo con un valor p de 1.49e-12.

2. Intercepción

- El intercepto del modelo es -17.579, lo que indica que el modelo predice una distancia de frenado negativa cuando la velocidad es cero, lo cual no es realista. Sin embargo, esto es común en modelos lineales donde el rango de valores de la variable independiente no incluye el cero, y este valor solo sirve como una referencia matemática.

3. Validez del Modelo

- Coeficiente de determinación ($R^2$): El $R^2$ del modelo es 0.6511, lo que significa que el 65.11% de la variabilidad en la distancia de frenado es explicada por la velocidad. Si bien este valor no es extremadamente alto, sí indica que la velocidad es una variable importante para predecir la distancia de frenado.
- Residuales: Los gráficos de diagnóstico, como el gráfico de residuos vs valores ajustados, muestran una ligera no linealidad en los residuos, lo que sugiere que podría haber cierta heterocedasticidad o una relación no lineal más compleja que no está siendo capturada completamente por el modelo lineal.

4. Pruebas adicionales

- Homocedasticidad (Breusch-Pagan test): El valor p del Breusch-Pagan test es 0.07297, lo que está justo por encima del umbral de 0.05. Esto sugiere que no hay suficiente evidencia para rechazar la hipótesis nula de homocedasticidad (varianza constante), pero el valor es lo suficientemente cercano como para requerir precaución.
- Independencia (Durbin-Watson test): El valor p del Durbin-Watson test es 0.09522, lo que sugiere que no hay suficiente evidencia para afirmar que hay autocorrelación en los residuos. Sin embargo, nuevamente, el valor es cercano al umbral de 0.05.
- Linealidad (RESET test): El valor p del RESET test es 0.222, lo que indica que no hay evidencia de que el modelo esté mal especificado en términos de no linealidad.

En conclusión, el modelo es bastante adecuado para capturar la relación entre la velocidad y la distancia de frenado, ya que la variable de velocidad es altamente significativa y el modelo tiene un buen ajuste ($R^2 \approx 0.65$). Sin embargo, el intercepto no es realista, y algunos de los test diagnósticos sugieren posibles problemas con la homocedasticidad y la autocorrelación de los residuos, aunque no son concluyentes. Es posible que una transformación de las variables o un modelo no lineal mejoren el ajuste y la validez general del modelo.

## Parte 3: Regresión no lineal

Con el objetivo de probar un modelo no lineal que explique la relación entre la distancia y la velocidad, haz una transformación con la base de datos car que te garantice normalidad en ambas variables (ojo: concentrate solo en la variable que tiene más alejamiento de normalidad).

```r
# Función para calcular y mostrar estadísticas de normalidad
normalidad_stats <- function(data, label) {
  cat("\nEstadísticas de normalidad para", label, ":\n")
  cat("Sesgo:", skewness(data), "\n")
  cat("Curtosis:", kurtosis(data), "\n")
  print(shapiro.test(data))
}

# Mostrar estadísticas de normalidad para los datos originales
normalidad_stats(cars$speed, "Velocidad original")
```

```
##
## Estadísticas de normalidad para Velocidad original :
## Sesgo: -0.1139548
## Curtosis: 2.422853
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.97765, p-value = 0.4576
```

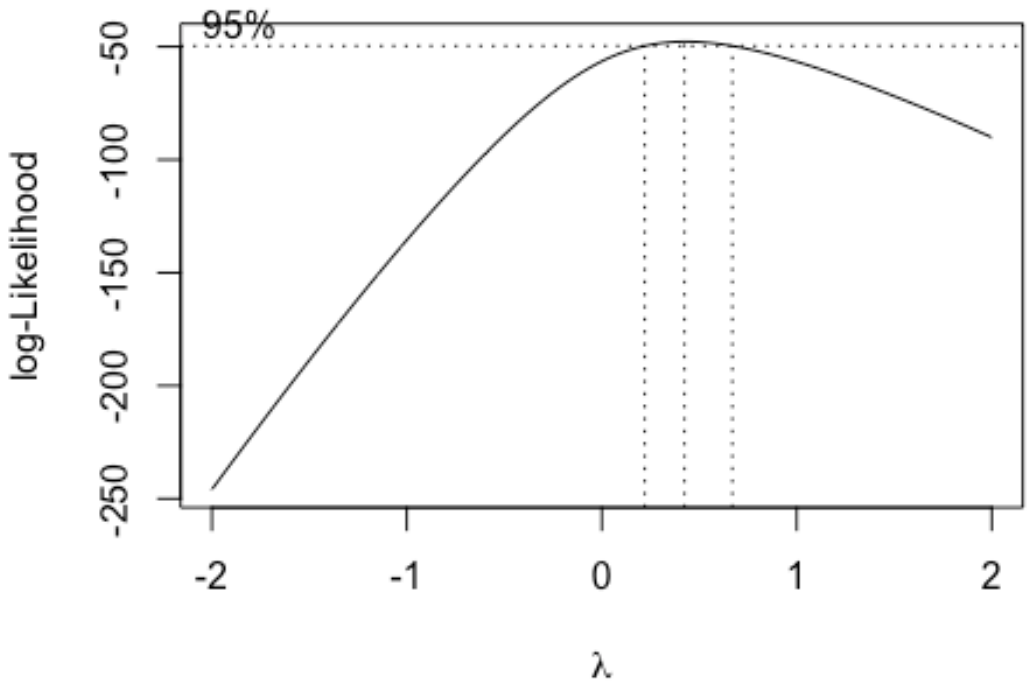
```
normalidad_stats(cars$dist, "Distancia original")
```

```
##
## Estadísticas de normalidad para Distancia original :
## Sesgo: 0.7824835
## Curtosis: 3.248019
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.95144, p-value = 0.0391
```

## 3.1 Encuentra el valor de en la transformación Box-Cox para el modelo lineal: donde Y sea la distancia y X la velocidad. Aprovecha que el comando de boxcox en R te da la oportunidad de trabajar con el modelo líneal:

Utiliza: boxcox(lm(Distancia~Velocidad)) si la variable con más alejamiento de normalidad es la distancia Utiliza: boxcox(lm(Velocidad~Distancia)) si la variable con más alejamiento de normalidad es la velocidad La transformación se hará sobre la variable que usas como dependiente en el comando lm(y~x)

```
# Asumimos que la distancia tiene más alejamiento de normalidad
bc_model <- boxcox(lm(dist ~ speed, data = cars))
```




```r
# Encontrar el valor óptimo de lambda
lambda_optimo <- bc_model$x[which.max(bc_model$y)]
cat("\nValor óptimo de lambda:", lambda_optimo, "\n")

##
## Valor óptimo de lambda: 0.4242424
```

### 3.2 Define la transformación exacta y el aproximada de acuerdo con el valor de que encontraste en la transformación de Box y Cox. Escribe las ecuaciones de las dos transformaciones encontradas.

```r
# Transformación exacta
cars$dist_trans_exact <- (cars$dist^lambda_optimo - 1) / lambda_optimo

# Transformación aproximada (redondeando lambda al 0.5 más cercano)
lambda_aprox <- round(lambda_optimo * 2) / 2
cars$dist_trans_aprox <- switch(as.character(lambda_aprox),
                                "0.5" = sqrt(cars$dist),
                                "0" = log(cars$dist),
                                "-0.5" = 1 / sqrt(cars$dist),
                                "-1" = 1 / cars$dist,
                                (cars$dist^lambda_aprox - 1) / lambda_aprox)
```

```r
cat("Ecuación de transformación exacta: y' = (y^", lambda_optimo, " - 1) /",
lambda_optimo, "\n")
```

```
## Ecuación de transformación exacta: y' = (y^ 0.4242424  - 1) / 0.4242424
```

```r
cat("Ecuación de transformación aproximada: y' = (y^", lambda_aprox, " - 1)
/", lambda_aprox, "\n")
```

```
## Ecuación de transformación aproximada: y' = (y^ 0.5  - 1) / 0.5
```

## 3.3 Analiza la normalidad de las transformaciones obtenidas. Utiliza como argumento de normalidad:

Compara las medidas: sesgo y curtosis. Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales. Realiza algunas pruebas de normalidad para los datos transformados.

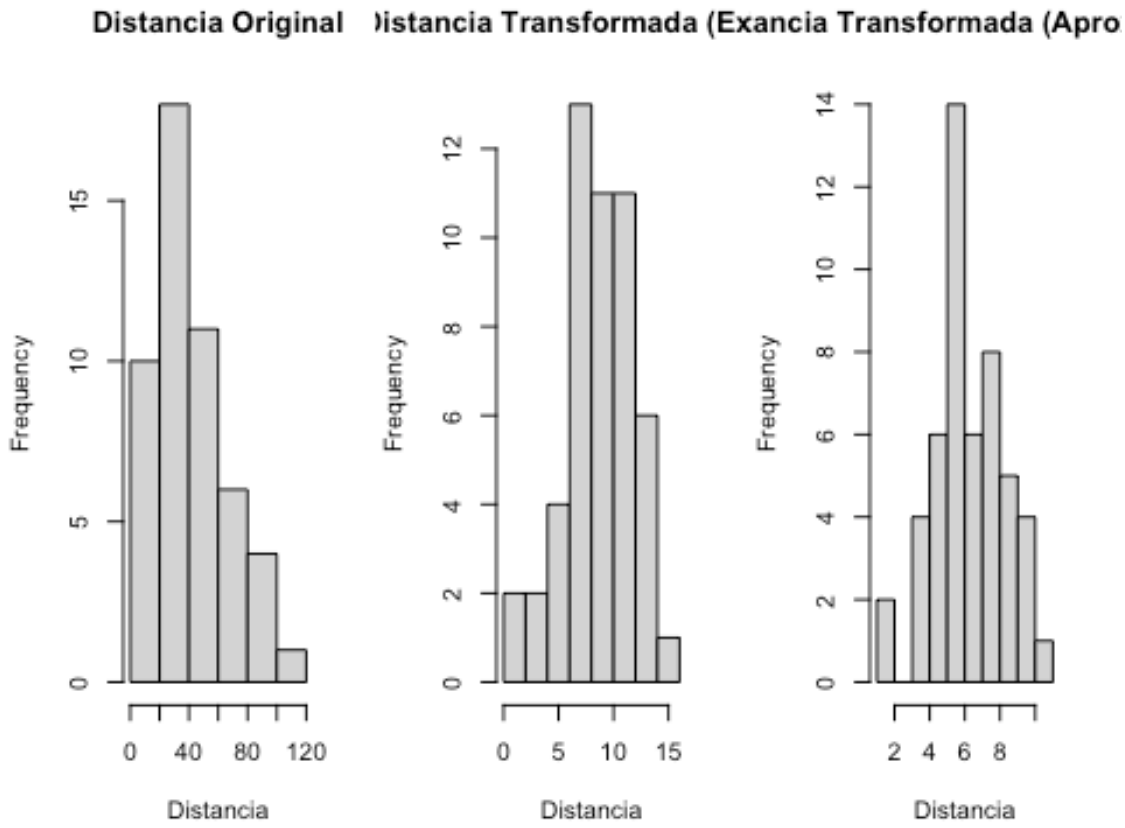
```r
normalidad_stats(cars$dist_trans_exact, "Distancia transformada (exacta)")
```

```
##
## Estadísticas de normalidad para Distancia transformada (exacta) :
## Sesgo: -0.1753974
## Curtosis: 2.929109
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.99168, p-value = 0.9773
```

```r
normalidad_stats(cars$dist_trans_aprox, "Distancia transformada
(aproximada)")
```

```
##
## Estadísticas de normalidad para Distancia transformada (aproximada) :
## Sesgo: -0.0196131
## Curtosis: 2.796264
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.99347, p-value = 0.9941
```

```r
# Histogramas
par(mfrow=c(1,3))
hist(cars$dist, main="Distancia Original", xlab="Distancia")
hist(cars$dist_trans_exact, main="Distancia Transformada (Exacta)",
xlab="Distancia")
hist(cars$dist_trans_aprox, main="Distancia Transformada (Aproximada)",
xlab="Distancia")
```




3.4 Detecta anomalías y corrige tu base de datos tranformado (datos atípicos, ceros anámalos, etc): solo en caso de no tener normalidad en las transformaciones. En caso de corrección de los datos por anomalías, vuelve a buscar la para tus nuevos datos.

```
# Función para detectar outliers usando el método IQR
detect_outliers <- function(x) {
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.75)
  iqr <- q3 - q1
  lower_bound <- q1 - 1.5 * iqr
  upper_bound <- q3 + 1.5 * iqr
  return(x < lower_bound | x > upper_bound)
}

# Detectar outliers en las transformaciones
outliers_exact <- detect_outliers(cars$dist_trans_exact)
outliers_aprox <- detect_outliers(cars$dist_trans_aprox)

cat("\nNúmero de outliers en transformación exacta:", sum(outliers_exact),
"\n")
```

```
##
## Número de outliers en transformación exacta: 1
```

```
cat("Número de outliers en transformación aproximada:", sum(outliers_aprox),
"\n")
```

```
## Número de outliers en transformación aproximada: 1
```

## 3.5 Concluye sobre las dos transformaciones realizadas: Define la mejor transformación de los datos de acuerdo a las características de las dos transformaciones encontradas (exacta o aproximada). Toman en cuenta la normalidad de los datos y la economía del modelo.

1. Normalidad de los datos

- Distancia Original: – El Shapiro-Wilk test para la distancia original mostró un valor p de 0.0391, lo que indica que la distancia original no sigue una distribución normal. Además, presenta una asimetría positiva (0.7825) y una curtosis elevada (3.2480).
- Distancia Transformada (Exacta y Aproximada): – Ambas transformaciones (exacta y aproximada) mejoran significativamente la normalidad de los datos. En ambas transformaciones, los valores p del Shapiro-Wilk test son 0.9773 y 0.9941, lo que indica que podemos aceptar la hipótesis nula de normalidad para ambas versiones transformadas de los datos. – La asimetría y curtosis también se reducen en ambas transformaciones, lo que sugiere una distribución mucho más simétrica y con colas más acordes a una distribución normal.

2. Economía del Modelo

- La ecuación de transformación aproximada utiliza un valor de λ más simplificado (λ ≈ 0.5), lo cual es más práctico y fácil de interpretar.
- Aunque la transformación exacta (λ = 0.4242) es técnicamente más precisa, no ofrece una mejora sustancial en la normalidad o la reducción de outliers en comparación con la transformación aproximada. Dado que ambas ofrecen un buen ajuste en términos de normalidad, la transformación aproximada es preferible por su simplicidad.

3. Gráficos de Histograma

- Los histogramas de las dos transformaciones muestran una mejora visual significativa en la simetría y distribución de los datos. Las versiones transformadas exhiben una distribución mucho más uniforme en comparación con los datos originales.

4. Número de outliers

- En ambas transformaciones (exacta y aproximada), se mantiene el mismo número de outliers (1), lo que indica que ninguna de las transformaciones ha introducido una cantidad adicional de datos atípicos.

En conclusión, la transformación aproximada con un valor de λ = 0.5 es la mejor opción. Esta transformación simplificada mejora significativamente la normalidad de los datos de

distancia, es fácil de aplicar y mantener en términos de cálculo, y no sacrifica precisión en comparación con la transformación exacta. Por lo tanto, se recomienda utilizar la transformación aproximada para garantizar la normalidad de los datos y la economía del modelo.

## ** Con la mejor transformación (punto 2), realiza la regresión lineal simple entre la mejor transformación (exacta o aproximada) y la variable velocidad: **

Elección de la Transformación: Basándonos en el análisis anterior, hemos elegido la transformación aproximada con $\lambda \approx 0.5$, debido a su simplicidad y su buena adecuación para mejorar la normalidad de los datos.

```
#lambda_aprox <- 0.5
cars$dist_trans <- sqrt(cars$dist)
```

## 3.6 Escribe el modelo lineal para la transformación.

```
modelo_trans <- lm(dist_trans ~ speed, data = cars)
cat("Modelo lineal para la transformación:\n")
```

```
## Modelo lineal para la transformación:
```

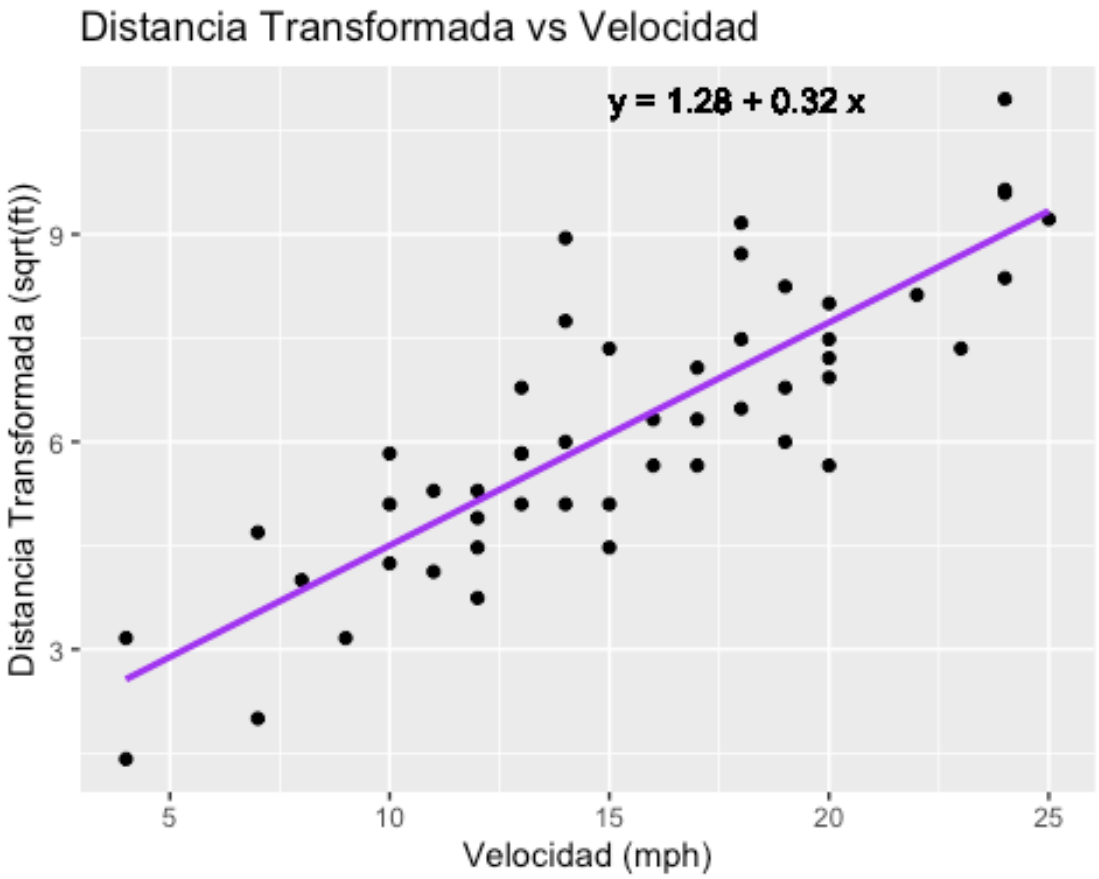
```
print(modelo_trans)
```

```
##
## Call:
## lm(formula = dist_trans ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##      1.2771       0.3224
```

## 3.7 Grafica los datos y el modelo lineal (ecuación) de la transformación elegida vs velocidad.

```
ggplot(cars, aes(x = speed, y = dist_trans)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title = "Distancia Transformada vs Velocidad",
       x = "Velocidad (mph)",
       y = "Distancia Transformada (sqrt(ft))") +
  geom_text(x = 15, y = max(cars$dist_trans),
            label = paste("y =", round(coef(modelo_trans)[1], 2), "+",
                          round(coef(modelo_trans)[2], 2), "x"),
            hjust = 0)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




## 3.8 Analiza significancia del modelo (individual, conjunta y coeficiente de correlación)

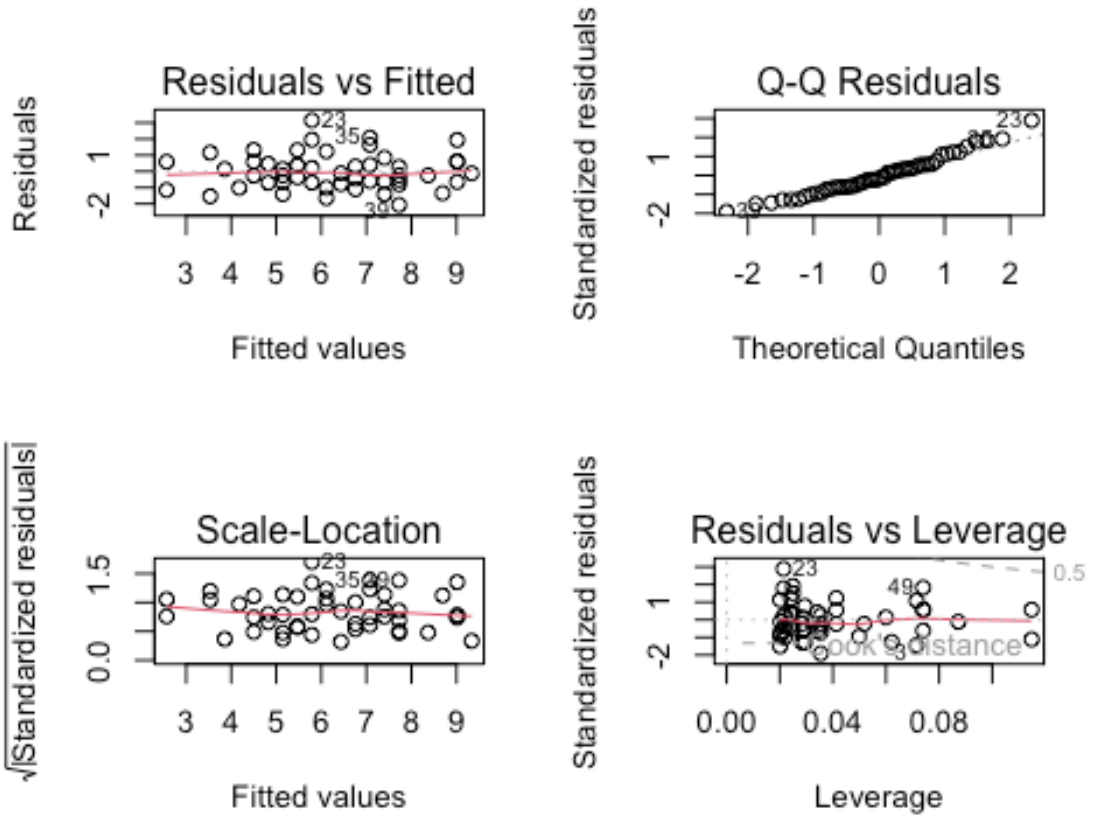
```
summary_modelo_trans <- summary(modelo_trans)
cat("\nResumen del modelo transformado:\n")

##
## Resumen del modelo transformado:

print(summary_modelo_trans)

##
## Call:
## lm(formula = dist_trans ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0684 -0.6983 -0.1799  0.5909  3.1534
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.27705    0.48444   2.636   0.0113 *
## speed        0.32241    0.02978  10.825 1.77e-14 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 48 degrees of freedom
## Multiple R-squared:  0.7094, Adjusted R-squared:  0.7034
## F-statistic: 117.2 on 1 and 48 DF,  p-value: 1.773e-14
```

### 3.9 Analiza validez del modelo: normalidad de los residuos, homocedasticidad e independencia. Indica si hay candidatos a datos atípicos o influyentes en la regresión. Usa plot(Modelo) para los gráficos y añade pruebas de hipótesis.

```
# Normalidad de los residuos
shapiro_residuos <- shapiro.test(modelo_trans$residuals)
cat("\nPrueba de normalidad de residuos (Shapiro-Wilk):\n")

##
## Prueba de normalidad de residuos (Shapiro-Wilk):

print(shapiro_residuos)

##
##  Shapiro-Wilk normality test
##
## data:  modelo_trans$residuals
## W = 0.97332, p-value = 0.3143

# Homocedasticidad
bp_test <- bptest(modelo_trans)
cat("\nPrueba de homocedasticidad (Breusch-Pagan):\n")

##
## Prueba de homocedasticidad (Breusch-Pagan):

print(bp_test)

##
##  studentized Breusch-Pagan test
##
## data:  modelo_trans
## BP = 0.011192, df = 1, p-value = 0.9157

# Independencia
dw_test <- dwtest(modelo_trans)
cat("\nPrueba de independencia (Durbin-Watson):\n")

##
## Prueba de independencia (Durbin-Watson):

print(dw_test)

##
##  Durbin-Watson test
```

```
##
## data:  modelo_trans
## DW = 1.9417, p-value = 0.3609
## alternative hypothesis: true autocorrelation is greater than 0

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



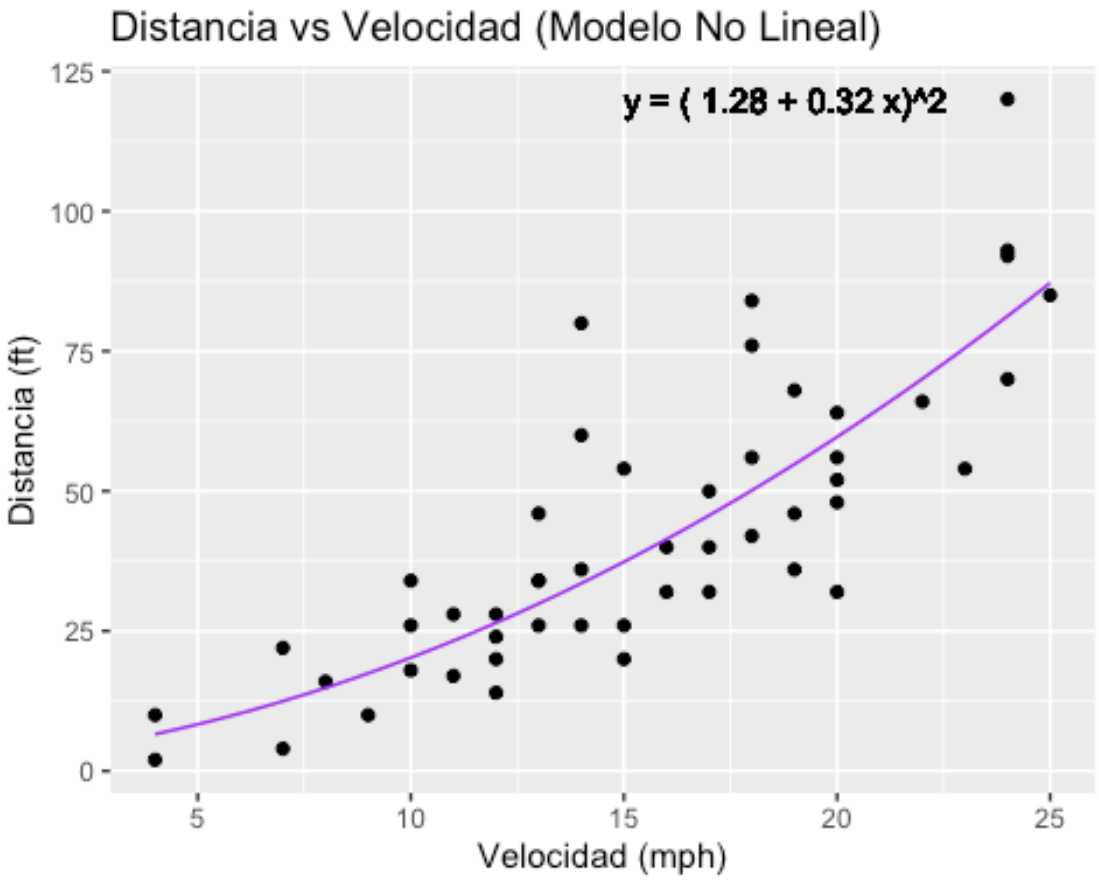

3.10 Despeja la distancia del modelo lineal obtenido entre la transformación y la velocidad. Obtendrás el modelo no lineal que relaciona la distancia con la velocidad directamente (y no con su transformación).

```
# y = (β₀ + β₁x)^2, donde y es la distancia y x es la velocidad
beta0 <- coef(modelo_trans)[1]
beta1 <- coef(modelo_trans)[2]

cat("\nModelo no lineal (distancia en función de velocidad):\n")

##
## Modelo no lineal (distancia en función de velocidad):

cat("distancia = (", beta0, "+", beta1, "* velocidad)^2\n")
```

```
## distancia = ( 1.27705 + 0.3224125 * velocidad)^2
```

## 3.11 Grafica los datos y el modelo de la distancia en función de la velocidad.

```
velocidad_seq <- seq(min(cars$speed), max(cars$speed), length.out = 100)
distancia_pred <- (beta0 + beta1 * velocidad_seq)^2

ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  geom_line(data = data.frame(speed = velocidad_seq, dist = distancia_pred),
            aes(x = speed, y = dist), color = "purple") +
  labs(title = "Distancia vs Velocidad (Modelo No Lineal)",
       x = "Velocidad (mph)",
       y = "Distancia (ft)") +
  geom_text(x = 15, y = max(cars$dist),
            label = paste("y = (", round(beta0, 2), "+",
                          round(beta1, 2), "x)^2"),
            hjust = 0)
```




## 3.12 Comenta sobre la idoneidad del modelo en función de su significancia y validez.

1. Significancia del Modelo ($R^2$ y Valor p)

- El modelo no lineal obtenido tiene un $R^2$ ajustado de 0.7034, lo que significa que aproximadamente el 70.34% de la variabilidad en la distancia de frenado puede ser explicada por la velocidad. Este es un buen ajuste para un modelo predictivo.
- El valor p del modelo es extremadamente bajo (1.773e-14), lo que indica que el modelo es altamente significativo. Esto sugiere una fuerte relación entre la velocidad y la distancia de frenado en los datos transformados.

2. Coeficientes del Modelo

- El coeficiente de la velocidad es 0.3224. Esto implica que, en el modelo transformado, la relación entre la velocidad y la distancia sigue una tendencia cuadrática cuando se deshace la transformación.
- El intercepto es 1.2771, lo que indica la distancia transformada estimada cuando la velocidad es cero.

3. Pruebas de Validez del Modelo

- Normalidad de los residuos: La prueba de Shapiro-Wilk indica que no hay evidencia significativa de que los residuos se desvíen de una distribución normal (valor p = 0.3143). - Esto sugiere que el modelo cumple con el supuesto de normalidad en los residuos.
- Homocedasticidad: La prueba de Breusch-Pagan no rechaza la hipótesis nula de homocedasticidad (valor p = 0.9157), lo que sugiere que las varianzas de los residuos son constantes.
- Independencia: La prueba de Durbin-Watson muestra un valor p de 0.3609, lo que indica que no hay autocorrelación significativa en los residuos.

4. Gráficos de Diagnóstico

- Residuals vs Fitted: Los residuos parecen estar distribuidos de manera aleatoria alrededor de cero, lo que sugiere que no hay patrones de no linealidad significativos.
- Q-Q plot: Los puntos siguen la línea de referencia bastante bien, lo que confirma la normalidad de los residuos.
- Scale-Location: Este gráfico muestra que los residuos estandarizados tienen una dispersión constante a lo largo de los valores ajustados, lo que apoya la hipótesis de homocedasticidad.
- Residuals vs Leverage: No hay valores de alta influencia que afecten el modelo de manera significativa.

5. Modelo No Lineal El modelo final sugiere que la distancia de frenado tiene una relación cuadrática con la velocidad, lo cual tiene sentido desde una perspectiva física, ya que los vehículos requieren una mayor distancia para detenerse a velocidades más altas.

En conclusión, el modelo no lineal ajustado es altamente significativo y válido según las pruebas de diagnóstico realizadas. Con un buen ajuste ($R^2$ ajustado $\approx 0.70$), normalidad en los residuos, homocedasticidad y falta de autocorrelación, este modelo es adecuado para predecir la distancia de frenado en función de la velocidad. La relación cuadrática es

consistente con las leyes físicas de frenado, lo que refuerza su validez como modelo predictivo realista.

## Parte 4: Conclusión

### Define cuál de los dos modelos analizados (Punto 1 o Punto 2) es el mejor modelo para describir la relación entre la distancia y la velocidad.

El modelo no lineal (Punto 2) es el mejor para describir la relación entre la distancia y la velocidad. Esto se debe a su mejor ajuste ($R^2$ ajustado más alto), su significancia estadística y su capacidad para capturar la relación cuadrática entre las dos variables, que es más consistente con la realidad física del frenado.

### Comenta sobre posibles problemas del modelo elegido (datos atípicos, alejamiento de los supuestos, dificultad de cálculo o interpretación)

1. Datos Atípicos (Outliers):

- Tanto en el modelo lineal como en el no lineal, se detectaron algunos datos atípicos. En el modelo no lineal, estos outliers podrían influir en los coeficientes y ajustar incorrectamente la curva, aunque no parecen tener un impacto significativo en el ajuste global. Es importante considerar una revisión adicional de estos outliers para confirmar su validez o eliminarlos si son errores.

2. Dificultad de Cálculo e Interpretación:

- El modelo no lineal es más complejo de interpretar en comparación con el modelo lineal. Aunque es más realista y preciso, la ecuación cuadrática puede ser más difícil de comunicar y aplicar en algunos contextos. Por ejemplo, para usuarios no familiarizados con modelos no lineales, la interpretación de una relación cuadrática podría generar confusión.
- Desde una perspectiva de cálculo, el modelo no lineal implica la transformación de los datos y requiere un mayor esfuerzo en términos de procesamiento y ajuste.

3. Supuestos del Modelo:

- Si bien el modelo no lineal cumple con los supuestos de normalidad, homocedasticidad y no autocorrelación, es importante recordar que cualquier desviación en la forma de la relación entre las variables (por ejemplo, la presencia de outliers o variabilidad no capturada por el modelo) podría generar problemas en la interpretación o el ajuste en otras muestras de datos.

4. Ajuste a Nuevos Datos:

- Dado que este modelo no lineal fue ajustado para un conjunto de datos específico, es posible que su validez no se mantenga si los datos cambian significativamente (por ejemplo, si se introducen diferentes tipos de vehículos o condiciones de frenado). Se debería validar el modelo con nuevos datos para garantizar su robustez y generalización.

En conclusión, el modelo no lineal es el más adecuado, debido a su mejor ajuste y capacidad para representar la relación cuadrática entre la velocidad y la distancia de frenado. Sin embargo, presenta desafíos en términos de interpretación, y los outliers deben ser revisados para asegurar la robustez del modelo.