13. Regresión no lineal

Adrian Pineda Sanchez

2024-09-12

Parte 1: Análisis de normalidad

Accede a los datos de cars en R (data = cars)

```
data(cars)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

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

```
library(nortest)

# 1. Prueba de Shapiro-Wilk
shapiro.test(cars$speed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cars$speed
## W = 0.97765, p-value = 0.4576
```

```
shapiro.test(cars$dist)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cars$dist
## W = 0.95144, p-value = 0.0391
```

```
# 2. Prueba de Anderson-Darling
ad.test(cars$speed)
```

```
##
##  Anderson-Darling normality test
##
## data:  cars$speed
## A = 0.26143, p-value = 0.6927
```

```
ad.test(cars$dist)
```

```
##
##  Anderson-Darling normality test
##
## data:  cars$dist
## A = 0.74067, p-value = 0.05021
```

```
# 3. Prueba de Jarque-Bera
jarque.bera.test(cars$speed)
```

```
##
##  Jarque Bera Test
##
## data:  cars$speed
## X-squared = 0.80217, df = 2, p-value = 0.6696
```

```
jarque.bera.test(cars$dist)
```

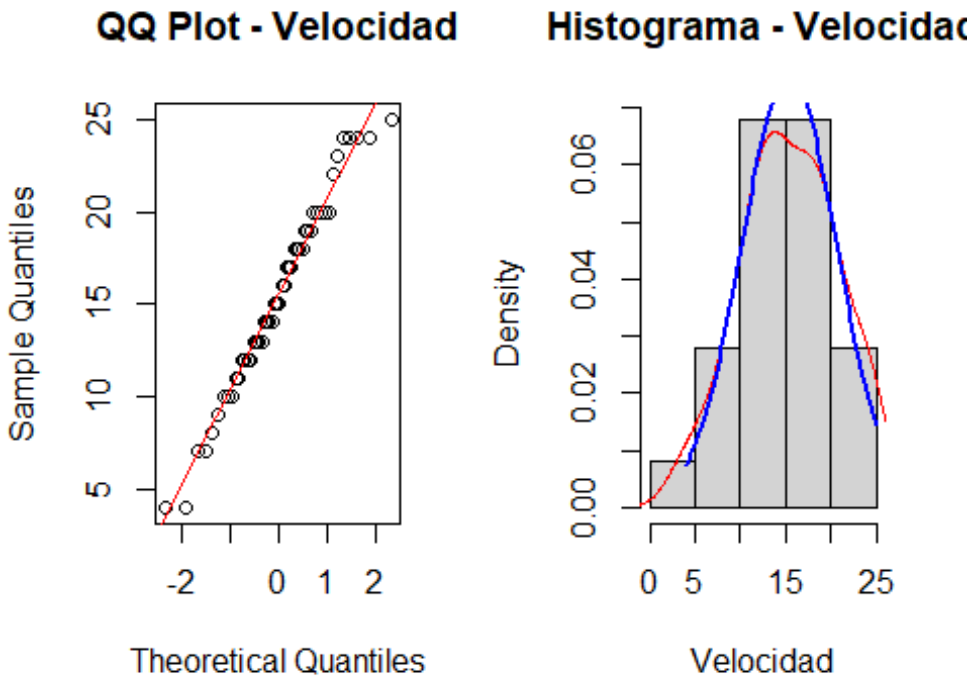
```
##
##  Jarque Bera Test
##
## data:  cars$dist
## X-squared = 5.2305, df = 2, p-value = 0.07315
```

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

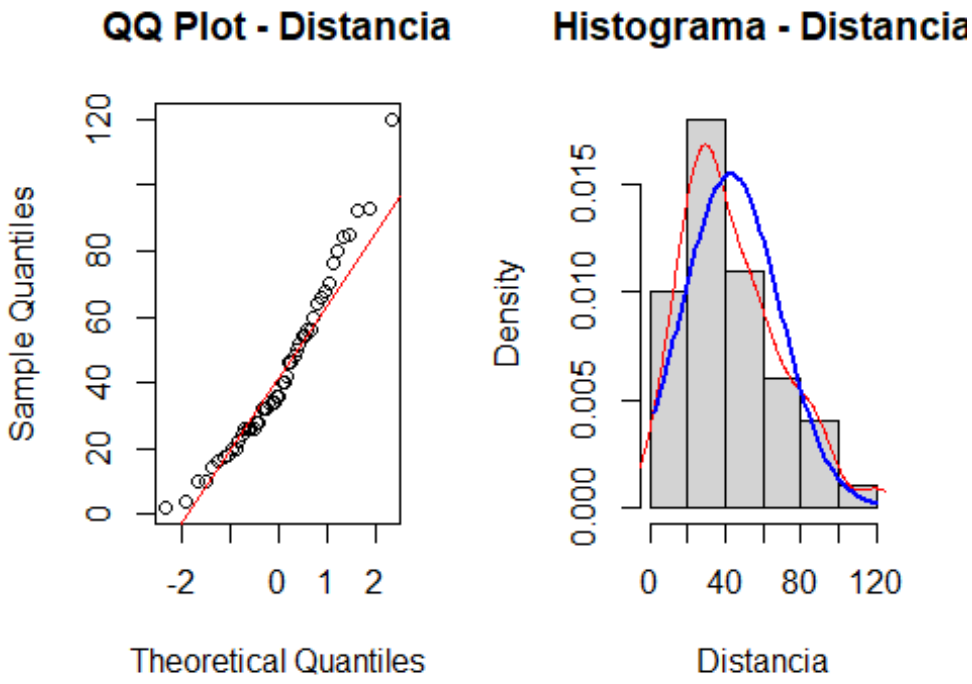
los datos y su respectivo QQPlot: qqnorm(datos) y qqline(datos) para cada variable

Realiza el histograma y su distribución teórica de probabilidad

```
# Configurar el espacio para dos gráficos en una sola ventana
par(mfrow=c(1,2))  # Dos gráficos en una fila

# 1. QQPlot para la velocidad
qqnorm(cars$speed, main = "QQ Plot - Velocidad")
qqline(cars$speed, col = "red")

# 2. Histograma para la velocidad con densidad y curva normal teórica
hist(cars$speed, freq = FALSE, main = "Histograma - Velocidad", xlab =
"Velocidad")
lines(density(cars$speed), col = "red")
curve(dnorm(x, mean=mean(cars$speed), sd=sd(cars$speed)),
      from=min(cars$speed), to=max(cars$speed), add=TRUE, col="blue", lwd=2)
```




```r
# Configurar el espacio para dos gráficos en una sola ventana
par(mfrow=c(1,2))  # Dos gráficos en una fila

# 1. QQPlot para la distancia
qqnorm(cars$dist, main = "QQ Plot - Distancia")
qqline(cars$dist, col = "red")

# 2. Histograma para la distancia con densidad y curva normal teórica
hist(cars$dist, freq = FALSE, main = "Histograma - Distancia", xlab =
"Distancia")
lines(density(cars$dist), col = "red")
curve(dnorm(x, mean=mean(cars$dist), sd=sd(cars$dist)),
      from=min(cars$dist), to=max(cars$dist), add=TRUE, col="blue", lwd=2)
```



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

```
library(e1071)

# Calcular el coeficiente de sesgo y curtosis para velocidad y distancia
sesgo_velocidad <- skewness(cars$speed)
curtosis_velocidad <- kurtosis(cars$speed)

sesgo_distancia <- skewness(cars$dist)
curtosis_distancia <- kurtosis(cars$dist)

# Mostrar resultados
cat("Sesgo de Velocidad:", sesgo_velocidad, "\n")

## Sesgo de Velocidad: -0.1105533

cat("Curtosis de Velocidad:", curtosis_velocidad, "\n")

## Curtosis de Velocidad: -0.6730924

cat("Sesgo de Distancia:", sesgo_distancia, "\n")

## Sesgo de Distancia: 0.7591268

cat("Curtosis de Distancia:", curtosis_distancia, "\n")

## Curtosis de Distancia: 0.1193971
```

2 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.

En la prueba de Shapiro-Wilk, que es adecuada para tamaños de muestra pequeños, los datos de velocidad no mostraron evidencia suficiente para rechazar la hipótesis de normalidad, ya que el p-valor fue de 0.4576 (mayor que 0.05). Sin embargo, los datos de distancia sí presentaron una desviación significativa de la normalidad, con un p-valor de 0.0391, lo que indica que los datos no siguen una distribución normal.

La prueba de Anderson-Darling, que es más sensible a las colas de la distribución, también confirmó que la velocidad parece seguir una distribución normal con un p-valor de 0.6927 > 0.05. Para la distancia, el p-valor fue de 0.05021, muy cercano al umbral de 0.05, lo que sugiere que aunque no se rechaza la hipótesis nula, hay indicios de una leve desviación de la normalidad, particularmente en las colas de la distribución.

Finalmente, la prueba de Jarque-Bera, que evalúa la normalidad basándose en el sesgo y la curtosis, también mostró que los datos de velocidad siguen una distribución normal con un p-valor de 0.6696. Para la distancia, el p-valor fue de 0.07315, lo que sugiere que la hipótesis de normalidad no se puede rechazar con esta prueba, aunque el resultado indica una ligera desviación.

Esto nos hace apreciar que dependiendo las pruebas, la distancia, es muy cercana a no cumplir normalidad en los datos dependiendo del α tomado para la prueba pero al menos con uno de 0.05 esta muy cerca de rechazar la hipotesis nula por Anderson y por Jarque bera, mientras que por Shapiro se rechaza. Mientras que la velocidad pasa facilmente todas.

Los gráficos y los valores de sesgo y curtosis indican que las dos variables, velocidad y distancia, tienen comportamientos diferentes en cuanto a la normalidad. Para velocidad, tanto el QQ plot como el histograma muestran que los datos se ajustan bastante bien a una distribución normal, con un sesgo ligeramente negativo (-0.1105) que sugiere una ligera inclinación hacia la izquierda y una curtosis negativa (-0.6731) que indica colas más delgadas que las de una distribución normal. Por otro lado, la variable distancia muestra signos claros de no normalidad, con el QQ plot exhibiendo una desviación en las colas y el histograma mostrando una asimetría hacia la derecha, confirmada por el sesgo positivo (0.7591). La curtosis para la distancia (0.1194) sugiere colas más gruesas que las de una distribución normal, lo que respalda la conclusión de que los datos de distancia no siguen una distribución normal, particularmente en los extremos.

Los posibles motivos de este alejamiento de la normalidad en la variable distancia pueden incluir la presencia de valores atípicos (outliers) que afectan las colas de la distribución, como se observa en el QQ plot. También es posible que haya una distribución subyacente no normal para la distancia debido a las características propias del fenómeno medido

Parte 2: Regresión lineal

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

```
# 1. Realizar regresión lineal (distancia ~ velocidad)
modelo <- lm(dist ~ speed, data = cars)

# Mostrar el modelo lineal
print(modelo)
```

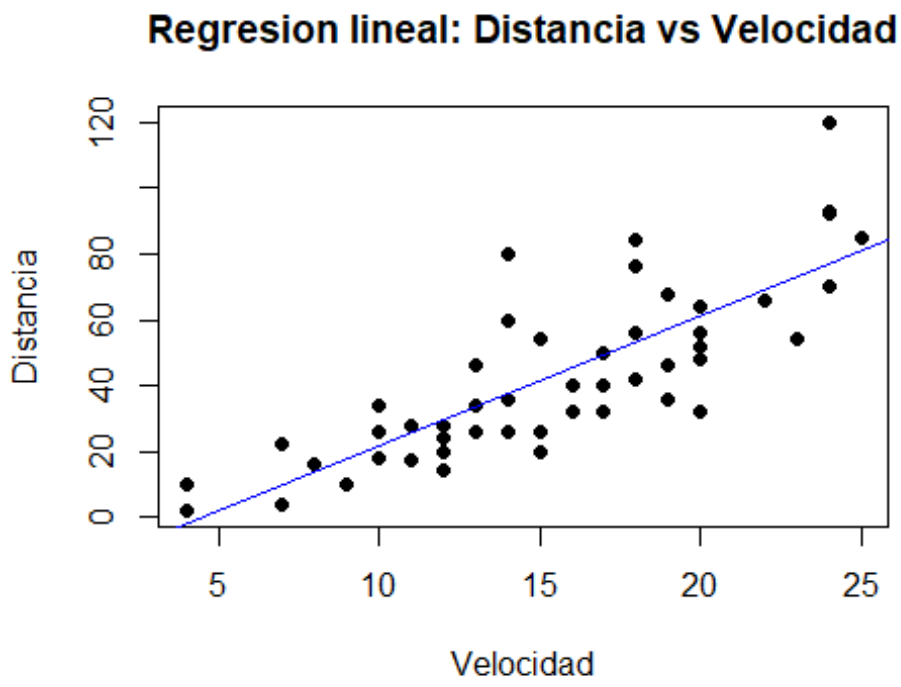
```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

Escribe el modelo lineal obtenido.

$$\widehat{\text{distancia}} = -17.579 + 3.932 \cdot \text{velocidad}$$

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

```
# Graficar los datos y el modelo
plot(cars$speed, cars$dist,
     main = "Regresion lineal: Distancia vs Velocidad",
     xlab = "Velocidad",
     ylab = "Distancia",
     pch = 19)

# Agregar la línea de regresión
abline(modelo, col = "blue")
```



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

```
summary(modelo)
```

```
##
## 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
```

- Coeficientes

- El coeficiente de la velocidad tiene un valor estimado de 3.9324, con un valor p de 1.49e-12, lo cual es mucho menor a 0.05. Esto significa que la velocidad es un predictor significativo de la distancia.
- El intercepto (-17.5791) también es significativo, con un valor p de 0.0123. Esto indica que, en ausencia de velocidad (cuando la velocidad es 0), el valor de la distancia esperada es -17.5791 (aunque este valor negativo no tiene un significado físico relevante).
- Modelo

-La prueba F conjunta tiene un valor de 89.57, con un valor p de 1.49e-12, lo que indica que el modelo en su conjunto es altamente significativo. Es decir, la velocidad en conjunto explica una parte significativa de la variabilidad en la distancia.

- Coeficiente de determinacion:

-El R^2 es de 0.6511, lo que significa que el modelo explica aproximadamente el 65.11% de la variabilidad total en la distancia. Este es un valor razonablemente alto, lo que indica que la velocidad es un buen predictor de la distancia. -El R^2 ajustado es de 0.6438, lo que corrige el coeficiente de determinación por el número de predictores en el modelo (en este caso, solo uno).

3. Analiza validez del modelo.

Residuos con media cero

H_0: Los residuos tienen media 0

H_1: Los residuos no tienen media 0

```
t.test(modelo$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo$residuals
## t = 1.0315e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.326  4.326
## sample estimates:
##    mean of x
## 2.220446e-16
```

Dado que p value = 0.01 aprox, podemos decir que es > 0.05 por lo tanto no rechazos la hipotesis nula, y podria sugerir que los residuos tienen media 0.

Normalidad de los residuos

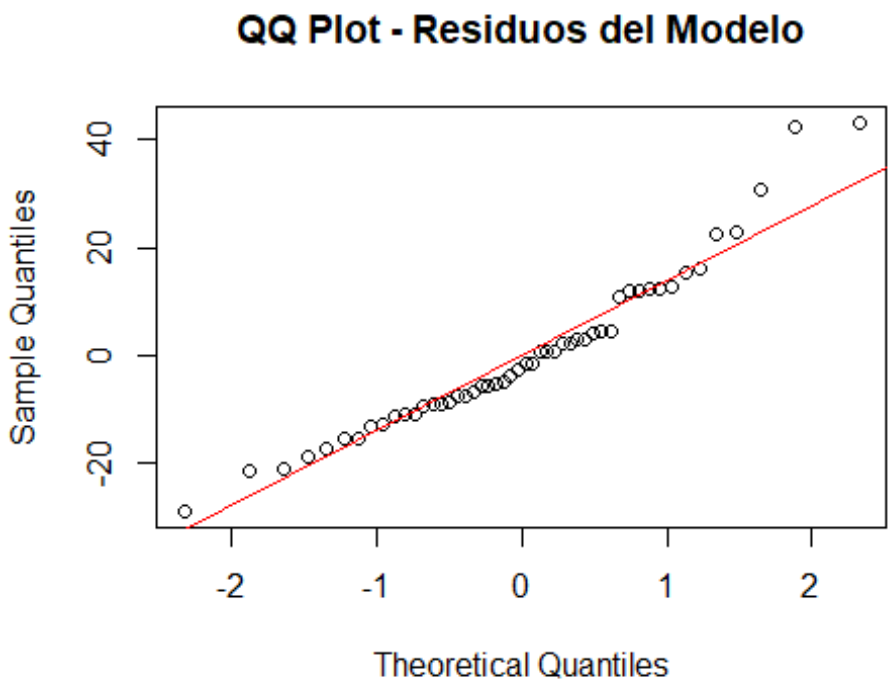
H_0: Los residuos siguen una distribucion Normal

H_1: Los residuos no siguen una distribucion Normal

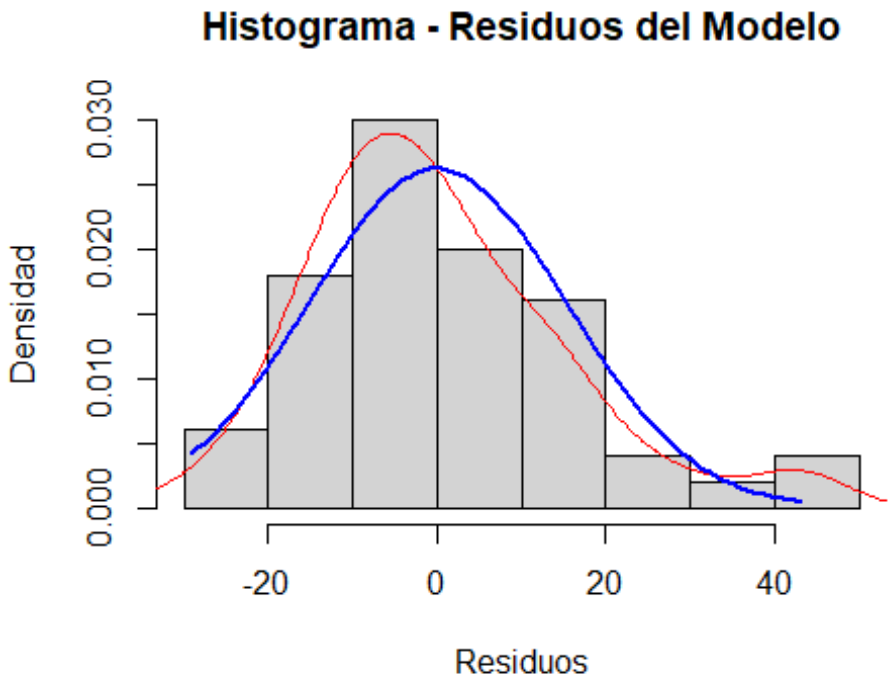
```
library(nortest)
shapiro.test(residuals(modelo))

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

jarque.bera.test(residuals(modelo))

##
##  Jarque Bera Test
##
## data:  residuals(modelo)
## X-squared = 8.1888, df = 2, p-value = 0.01667
```

Dado que incluso con ambas pruebas de normalidad vemos que su p value < 0.05 ya que estan entre .02 y .01 por lo que se rechaza la hipotesis nula de normalidad de los residuos

```
# Graficar la normalidad de los residuos del modelo
# 1. QQ Plot de los residuos
qqnorm(modelo$residuals, main = "QQ Plot - Residuos del Modelo")
qqline(modelo$residuals, col = "red")
```



```r
# 2. Histograma de los residuos
hist(modelo$residuals, freq = FALSE, main = "Histograma - Residuos del
Modelo",
     xlab = "Residuos", ylab = "Densidad")
lines(density(modelo$residuals), col = "red")  # Curva de densidad ajustada

# Agregar la curva de la distribución normal
curve(dnorm(x, mean = mean(modelo$residuals), sd = sd(modelo$residuals)),
      from = min(modelo$residuals), to = max(modelo$residuals),
      add = TRUE, col = "blue", lwd = 2)
```

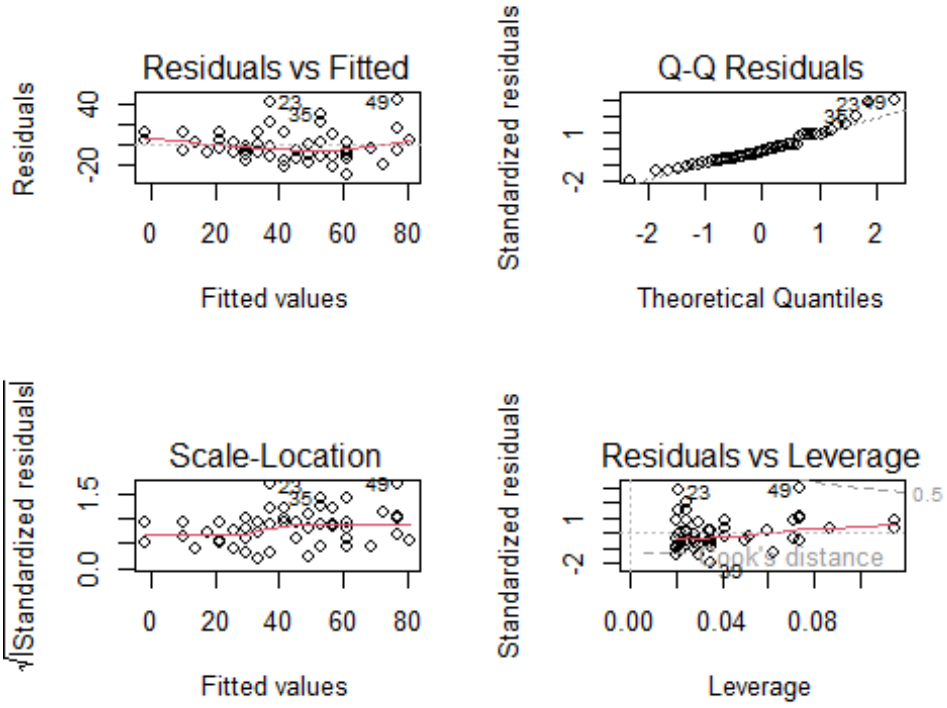


Homocedasticidad, independencia y linealidad.

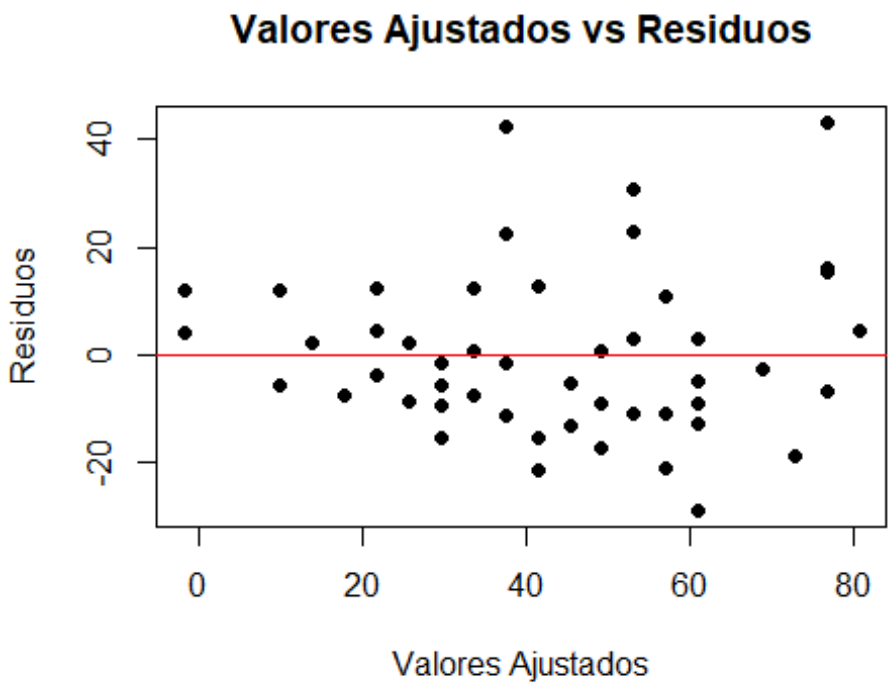
Homocedasticidad

H_0: La varianza de los residuos es constante (homocedasticidad). H_1: La varianza de los residuos no es constante (heterocedasticidad).

```r
# Gráficos de diagnóstico del modelo
par(mfrow = c(2, 2))  # Organizar gráficos
plot(modelo)
```




```r
par(mfrow = c(1, 1))  # Restaurar la configuración de gráficos

# Graficar los valores ajustados vs los residuos
plot(modelo$fitted.values, modelo$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)

# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```




```
library(lmtest)
```

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

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

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

```
bptest(modelo)
```

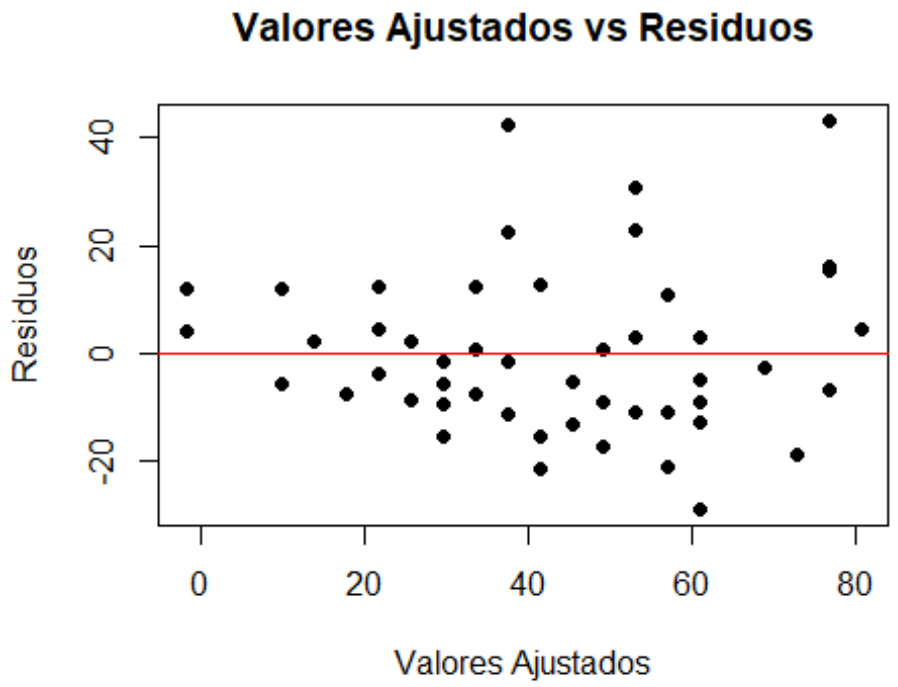
```
##
##  studentized Breusch-Pagan test
##
## data:  modelo
## BP = 3.2149, df = 1, p-value = 0.07297
```

Dado un p value > 0.05 no rechazamos la hipotesis nula por lo que se podria sugerir que se cumple la homocedasticidad aunque lo pasa muy apenas. En el grafico Dado que los residuos no muestran ningún patrón claro de expansión o contracción (es decir, no hay un cambio sistemático en la dispersión de los residuos), se puede concluir que no hay suficiente evidencia de heterocedasticidad en los datos. Por lo tanto, es razonable asumir que el supuesto de homocedasticidad se cumple en este modelo.

Independencia

Hipótesis nula H_0:: Los residuos no están correlacionados (independencia). Hipótesis alternativa H_1: Los residuos están correlacionados.

```
# Graficar los valores ajustados vs los residuos
plot(modelo$fitted.values, modelo$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)

# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```

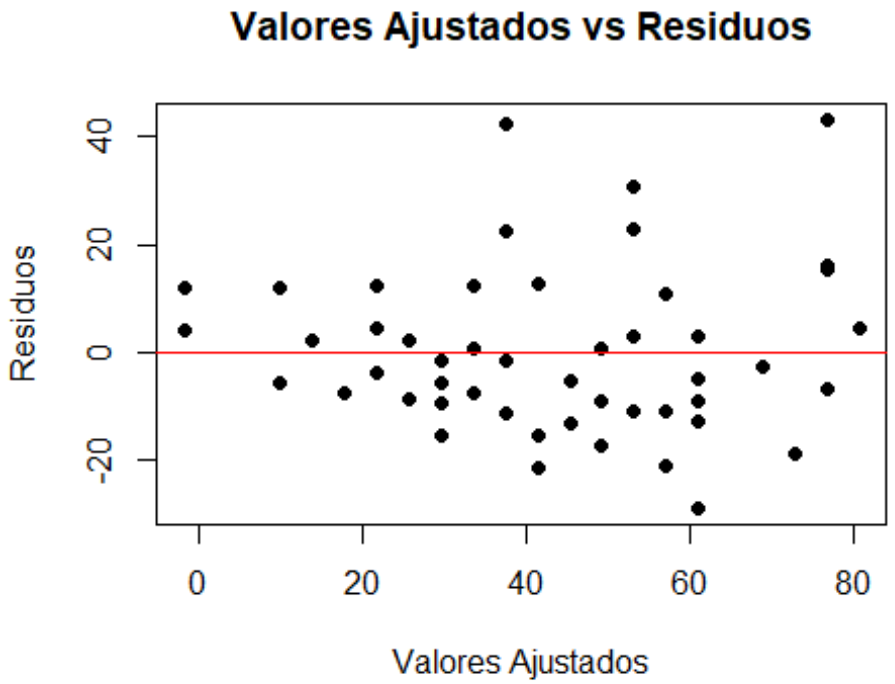



```
dwtest(modelo)

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

Dado p value > 0.05 y no rechazamos hipotesis nula podemos sugerir que los residuos tienen independencia entre si. Es visible a traves del grafico ya que no muestran una tendencia.

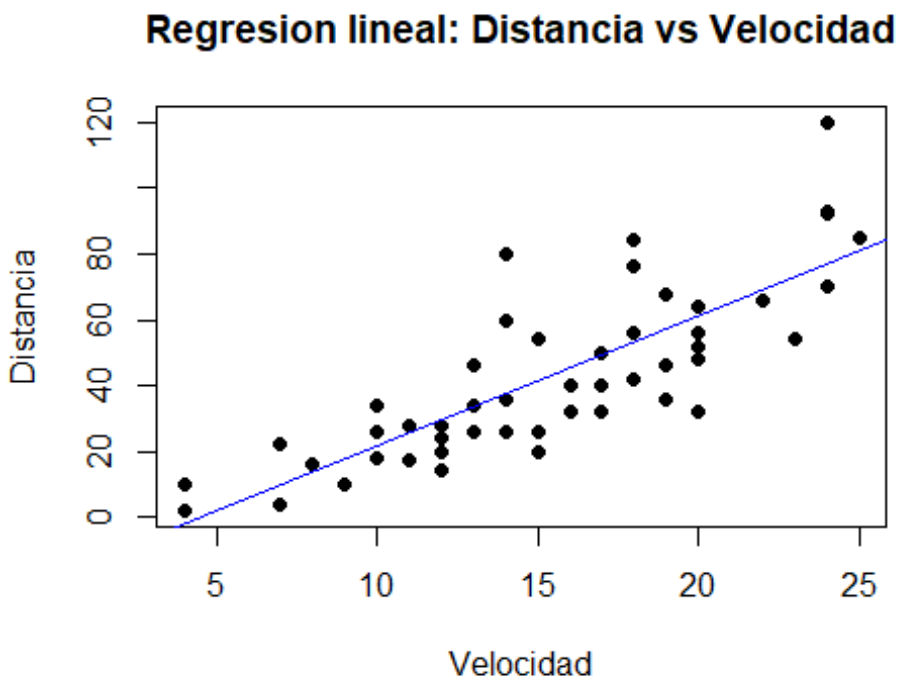
linealidad.

```r
# Graficar los valores ajustados vs los residuos
plot(modelo$fitted.values, modelo$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)

# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```



Hipótesis nula H_0: La relación entre la variable dependiente e independiente es lineal.
Hipótesis alternativa H_1: La relación entre la variable dependiente e independiente no es lineal.

```r
# Graficar los datos y el modelo
plot(cars$speed, cars$dist,
     main = "Regresion lineal: Distancia vs Velocidad",
     xlab = "Velocidad",
     ylab = "Distancia",
     pch = 19)

# Agregar la línea de regresión
abline(modelo, col = "blue")
```



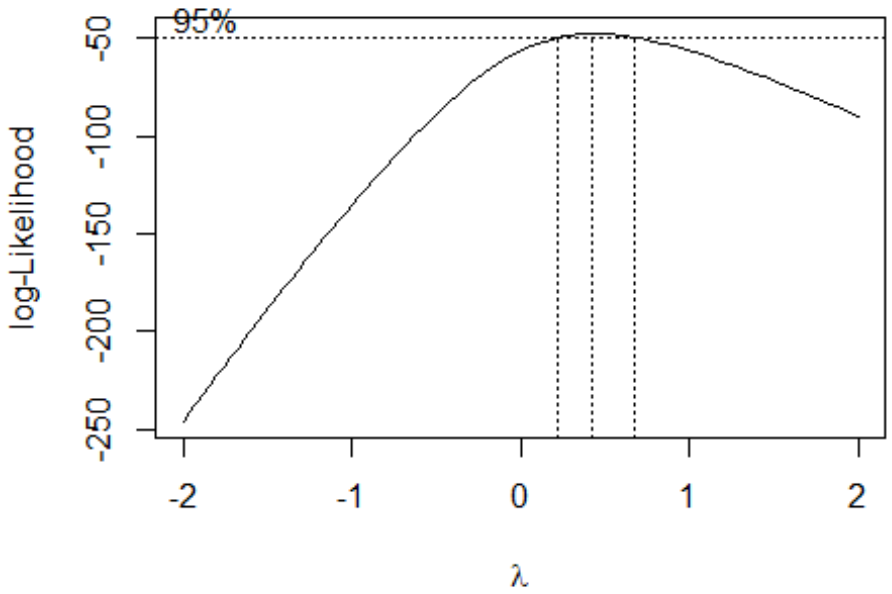
Conclusion

Aun a pesar de los buenos indicadores en los estadisticos obtenidos al inicio del modelo en cuestion de la significancia del modelo y su factores individuales asi como su coeficiente de determicacion , vemos que no cumple todos los supuestos de normalidad en cuestion de la validez del modelo en la parte de la normalidad de los residuos, por lo que podriamos analizar mas alternativas

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).

```
library(MASS)
boxcox_result <- boxcox(modelo, lambda = seq(-2, 2, by = 0.1))
```




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

## Valor óptimo de lambda: 0.4242424
```

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

```r
# Aproximación de la transformación: aplicar raíz cuadrada (sqrt) a la
distancia (cuando lambda es cercano a 0)
dist_transformada_aprox <- sqrt(cars$dist)

# Ajustar un nuevo modelo de regresión con la distancia transformada
(Aproximación sqrt)
modelo_transformado_aprox <- lm(dist_transformada_aprox ~ speed, data = cars)

# Resumen del nuevo modelo transformado (aproximado)
summary(modelo_transformado_aprox)

##
## Call:
## lm(formula = dist_transformada_aprox ~ 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
```

```
# Transformar la variable 'distancia' usando el lambda óptimo encontrado
dist_transformada <- (cars$dist^lambda_optimo - 1) / lambda_optimo

# Ajustar un nuevo modelo de regresión con la distancia transformada
modelo_transformado <- lm(dist_transformada ~ speed, data = cars)

# Resumen del nuevo modelo transformado
summary(modelo_transformado)
```

```
##
## Call:
## lm(formula = dist_transformada ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0926 -1.0444 -0.3055  0.7999  4.7520
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.08227    0.73856   1.465    0.149
## speed        0.49541    0.04541  10.910 1.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 48 degrees of freedom
## Multiple R-squared:  0.7126, Adjusted R-squared:  0.7066
## F-statistic:   119 on 1 and 48 DF,  p-value: 1.354e-14
```

Ecuacion modelo aprox:

$ y = (x)$

Ecuacion modelo exacto:

$ y = (x^)/ $

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

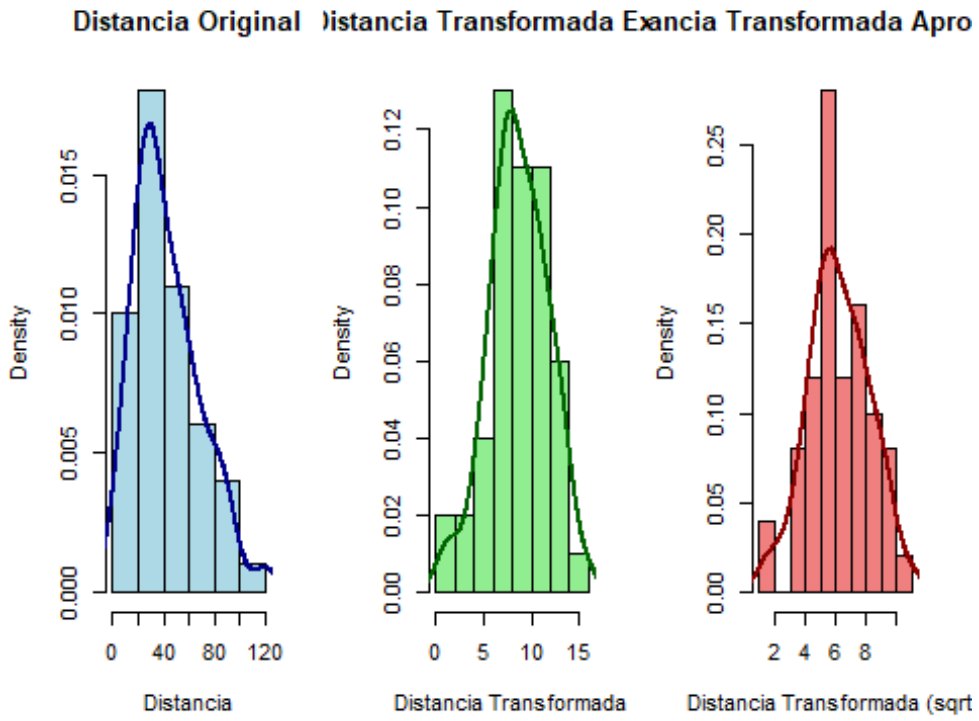
Compara las medidas: sesgo y curtosis.

```
library(e1071)

# Sesgo y curtosis antes de la transformación
sesgo_original <- skewness(cars$dist)
curtosis_original <- kurtosis(cars$dist)

# Sesgo y curtosis después de la transformación aproximada (sqrt)
sesgo_transformado_aprox <- skewness(dist_transformada_aprox)
curtosis_transformada_aprox <- kurtosis(dist_transformada_aprox)

# Sesgo y curtosis después de la transformación
sesgo_transformado <- skewness(dist_transformada)
curtosis_transformada <- kurtosis(dist_transformada)

cat("Sesgo original:", sesgo_original, "\n")

## Sesgo original: 0.7591268

cat("Curtosis original:", curtosis_original, "\n")

## Curtosis original: 0.1193971

cat("Sesgo transformado (sqrt):", sesgo_transformado_aprox, "\n")

## Sesgo transformado (sqrt): -0.01902765

cat("Curtosis transformada (sqrt):", curtosis_transformada_aprox, "\n")

## Curtosis transformada (sqrt): -0.3144682

cat("Sesgo transformado:", sesgo_transformado, "\n")

## Sesgo transformado: -0.1701619

cat("Curtosis transformada:", curtosis_transformada, "\n")

## Curtosis transformada: -0.186884
```

Obten el histograma de los 2 modelos obtenidos (exacto y aproximado) y los datos originales.

```
# Graficar histogramas de las distribuciones originales, transformadas exacta
(Box-Cox) y aproximada (sqrt)
par(mfrow=c(1, 3))  # Tres gráficos en una fila

# Histograma de los datos originales con colores personalizados
hist(cars$dist, freq=FALSE, main="Distancia Original", xlab="Distancia",
col="lightblue", border="black")
lines(density(cars$dist), col="darkblue", lwd=2)
```

```r
# Histograma de los datos transformados (Box-Cox exacto) con colores personalizados
hist(dist_transformada, freq=FALSE, main="Distancia Transformada Exacto", xlab="Distancia Transformada", col="lightgreen", border="black")
lines(density(dist_transformada), col="darkgreen", lwd=2)

# Histograma de los datos transformados (sqrt aproximado) con colores personalizados
hist(dist_transformada_aprox, freq=FALSE, main="Distancia Transformada Aproximado", xlab="Distancia Transformada (sqrt)", col="lightcoral", border="black")
lines(density(dist_transformada_aprox), col="darkred", lwd=2)
```



Realiza algunas pruebas de normalidad para los datos transformados.

H_0: El modelo sigue una distribucion normal

H_1: El modelo no sigue una distribucion normal

```r
# Prueba de normalidad Shapiro-Wilk para los datos originales
shapiro_test_original <- shapiro.test(cars$dist)
cat("Prueba de Shapiro-Wilk para los datos originales: p-valor =", shapiro_test_original$p.value, "\n")
```

```
## Prueba de Shapiro-Wilk para los datos originales: p-valor = 0.03909968
```

```r
# Prueba de normalidad Shapiro-Wilk para los datos transformados (Box-Cox exacto)
shapiro_test_boxcox <- shapiro.test(dist_transformada)
cat("Prueba de Shapiro-Wilk para los datos transformados Exacto: p-valor =", shapiro_test_boxcox$p.value, "\n")
```

```
## Prueba de Shapiro-Wilk para los datos transformados Exacto: p-valor = 0.9772686
```

```r
# Prueba de normalidad Shapiro-Wilk para los datos transformados (sqrt aproximado)
shapiro_test_sqrt <- shapiro.test(dist_transformada_aprox)
cat("Prueba de Shapiro-Wilk para los datos transformados aproximado: p-valor =", shapiro_test_sqrt$p.value, "\n")
```

```
## Prueba de Shapiro-Wilk para los datos transformados aproximado: p-valor = 0.9941205
```

Podemos observar inmediatamente como con el modelo original obtenemos un p value bajisimo < 0.05 por lo que rechazabamos normalidad en el modelo, sin embargo despues de las transformaciones con Boxcox, tanto en el modelo aproximado como el modelo exacto, el p value es altisimo, mejorando significativamente en el modelo la normalidad aparente de los datos en una prueba que anteriormente no habian pasado que era la de shapiro, lo que crea un impacto positivo hacia la normalidad de los datos. Siendo ligeramente mejor a traves de esta prueba posiblemente el modelo aproximado

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 lambda para tus nuevos datos.

Por lo anteriormente descrito, no es necesario la deteccion de anomalias ya que los 2 nuevos modelos generados mediante Boxcox nos han dado una aparente normalidad, siendo la mejor la del metodo aproximado por el p value tan cercano a 1

2. 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.

Ambas transformaciones mejoran mucho la normalidad de los datos casi en manera equivalente, La transformación aproximada (sqrt) ha sido más efectiva en eliminar el sesgo (asimetría) de la distribución y en reducir la curtosis, haciendo la distribución más simétrica y con colas más delgadas. La transformación exacta (Box-Cox) también mejora la simetría y reduce la curtosis, pero no de manera tan significativa como la del modelo aproximado.

Asimismo en las pruebas de normalidad el metodo aproximado mostro mejores resultado al tener un ligero p value aprox > p value exacto, por lo que queda seleccionado como el mejor modelo, debido a que nos ayuda de mejor manera a llegar a normalidad.

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

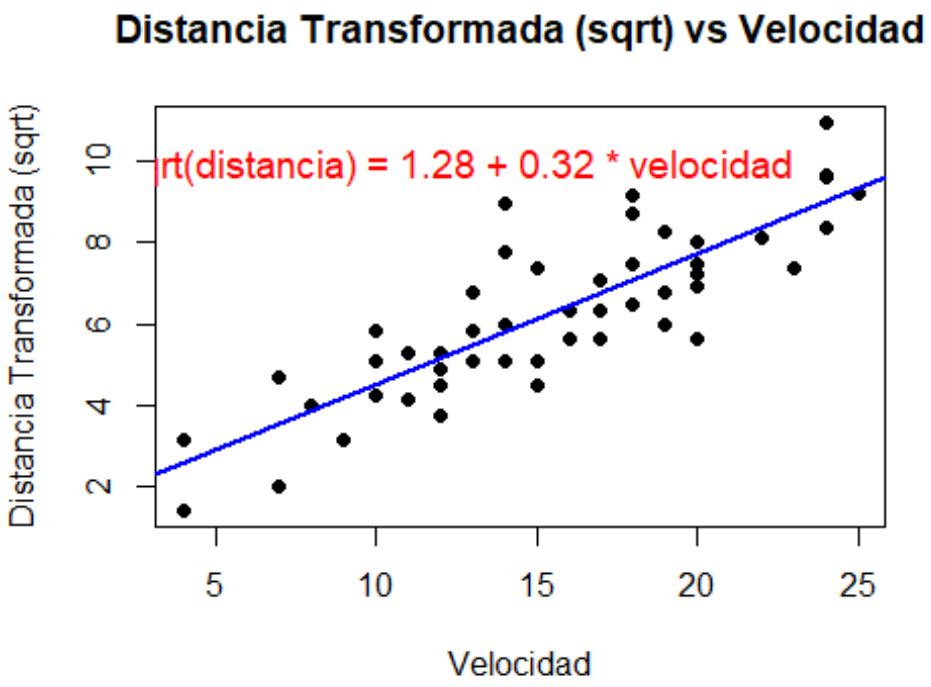
Escribe el modelo lineal para la transformación.

```
# Ajustar el modelo de regresión lineal con la transformación aproximada
(sqrt)
modelo_lambda_aprox <- lm(dist_transformada_aprox ~ speed, data = cars)

# Obtener los coeficientes del modelo
coeficientes <- coef(modelo_lambda_aprox)

# Imprimir la ecuación del modelo
cat("Modelo: sqrt(distancia) = ", (coeficientes[1]), " + ", coeficientes[2],
" * velocidad\n")
```

```
## Modelo: sqrt(distancia) =  1.27705  +  0.3224125  * velocidad
```

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

```
# Graficar la distancia transformada (sqrt) frente a la velocidad
plot(cars$speed, dist_transformada_aprox,
     main = "Distancia Transformada (sqrt) vs Velocidad",
     xlab = "Velocidad",
     ylab = "Distancia Transformada (sqrt)",
     pch = 19)

# Agregar la línea de regresión al gráfico
abline(modelo_lambda_aprox, col = "blue", lwd = 2)

# Agregar la ecuación al gráfico (con sqrt(distancia))
eq <- paste0("sqrt(distancia) = ", round(coeficientes[1], 2), " + ",
round(coeficientes[2], 2), " * velocidad")
text(x = max(cars$speed)*0.5, y = max(dist_transformada_aprox)*0.9, labels =
eq, col = "red", cex = 1.2)
```



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

```
# Resumen del modelo
summary(modelo_lambda_aprox)

##
## Call:
## lm(formula = dist_transformada_aprox ~ 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
```

*Significancia individual (Coeficiente de la velocidad):

-El coeficiente de velocidad es 0.32241, lo que significa que por cada unidad adicional en la velocidad, la transformación sqrt(distancia) aumenta en 0.32241. -El valor p asociado al

coeficiente de velocidad es 1.77e-14, lo que indica que es altamente significativo (muy por debajo de 0.05). Esto significa que la velocidad es un predictor relevante para la distancia transformada.

*Significancia conjunta (F-statistic):

-El valor de la F-statistic es 117.2, con un valor p de 1.773e-14, lo que indica que el modelo completo es significativo. Esto significa que la velocidad, en conjunto, explica una parte considerable de la variabilidad de la transformación de la distancia.

*Coeficiente de determinación (R^2):

-El R^2 es 0.7094, lo que significa que aproximadamente el 70.94% de la variabilidad en la distancia transformada puede ser explicada por la velocidad. Este es un buen ajuste. -El R^2 ajustado es 0.7034, lo que ajusta el valor de R^2 por el número de predictores en el modelo. Dado que solo hay un predictor (velocidad), el ajuste es pequeñ

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.

Residuos con media cero

H_0: Los residuos tienen media 0

H_1: Los residuos no tienen media 0

```
t.test(modelo_lambda_aprox$residuals)
```

```
##
##  One Sample t-test
##
## data:  modelo_lambda_aprox$residuals
## t = 3.9571e-16, df = 49, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.3100858  0.3100858
## sample estimates:
##    mean of x
## 6.105969e-17
```

Dado el p value muy cercano a 1, no rechazamos la hipotesis nula y es probable que la media de los residuos tienda a 0

Normalidad de los residuos

H_0: Los residuos siguen una distribucion Normal

H_1: Los residuos no siguen una distribucion Normal

```r
# Prueba de Shapiro-Wilk para normalidad de los residuos
library(nortest)
shapiro.test(residuals(modelo_lambda_aprox))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(modelo_lambda_aprox)
## W = 0.97332, p-value = 0.3143
```

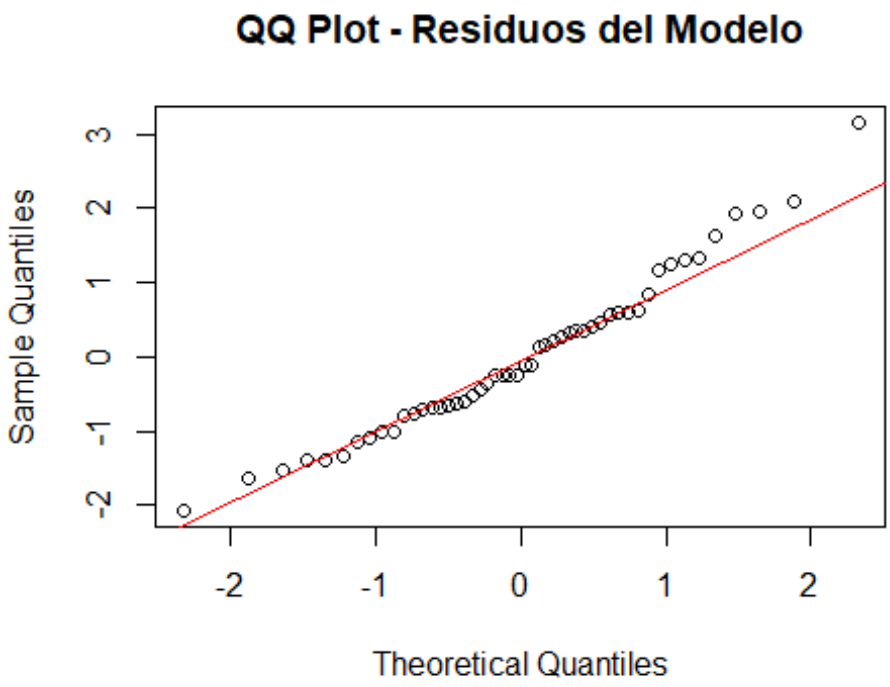
Dado p value > 0.05 no rechazamos la hipotesis nula por lo que ademas es muy alto, por tanto podriamos asumir como verdadero que los residuos podrian seguir una distribucion normal

```r
# Prueba de Jarque-Bera para normalidad de los residuos
library(tseries)
jarque.bera.test(residuals(modelo_lambda_aprox))
```

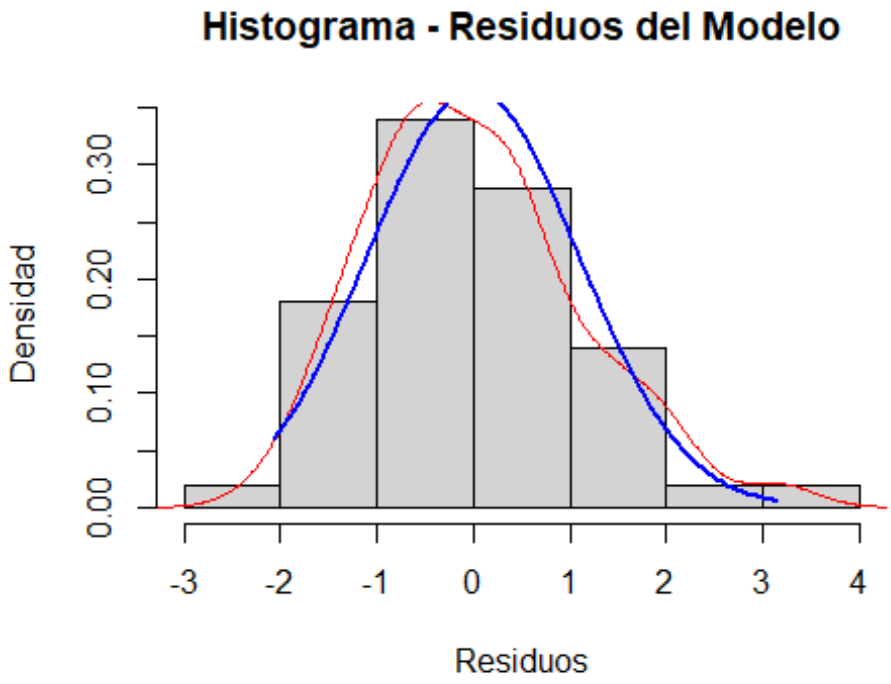
```
##
##  Jarque Bera Test
##
## data:  residuals(modelo_lambda_aprox)
## X-squared = 2.862, df = 2, p-value = 0.2391
```

Mismo Caso, Dado p value > 0.05 no rechazamos la hipotesis nula por lo que ademas es muy alto, por tanto podriamos asumir como verdadero que los residuos podrian seguir una distribucion normal

```r
# Graficar la normalidad de los residuos del modelo
# 1. QQ Plot de los residuos
qqnorm(modelo_lambda_aprox$residuals, main = "QQ Plot - Residuos del Modelo")
qqline(modelo_lambda_aprox$residuals, col = "red")
```




```r
# 2. Histograma de los residuos
hist(modelo_lambda_aprox$residuals, freq = FALSE, main = "Histograma -
Residuos del Modelo",
     xlab = "Residuos", ylab = "Densidad")
lines(density(modelo_lambda_aprox$residuals), col = "red")  # Curva de
densidad ajustada

# Agregar la curva de la distribución normal
curve(dnorm(x, mean = mean(modelo_lambda_aprox$residuals), sd =
sd(modelo_lambda_aprox$residuals)),
      from = min(modelo_lambda_aprox$residuals), to =
max(modelo_lambda_aprox$residuals),
      add = TRUE, col = "blue", lwd = 2)
```

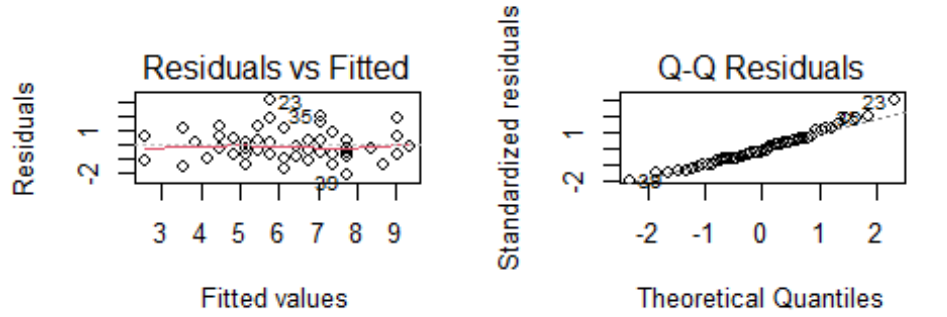


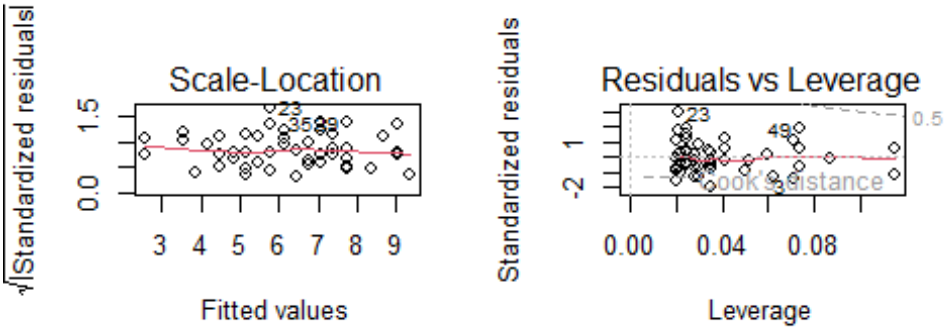
Homocedasticidad, independencia y linealidad.

Homocedasticidad

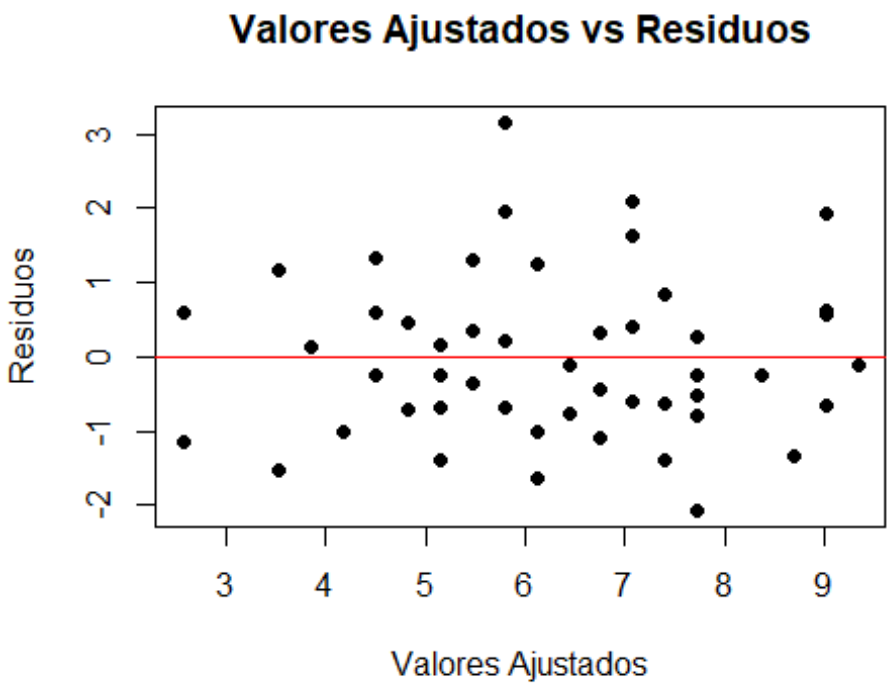
H_0: La varianza de los residuos es constante (homocedasticidad). H_1: La varianza de los residuos no es constante (heterocedasticidad).

```
# Gráficos de diagnóstico del modelo
par(mfrow = c(2, 2))  # Organizar gráficos
plot(modelo_lambda_aprox)
```






```r
par(mfrow = c(1, 1))  # Restaurar la configuración de gráficos

# Graficar los valores ajustados vs los residuos
plot(modelo_lambda_aprox$fitted.values, modelo_lambda_aprox$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)

# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```




```
# Prueba de Breusch-Pagan para homocedasticidad
library(lmtest)
bptest(modelo_lambda_aprox)

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

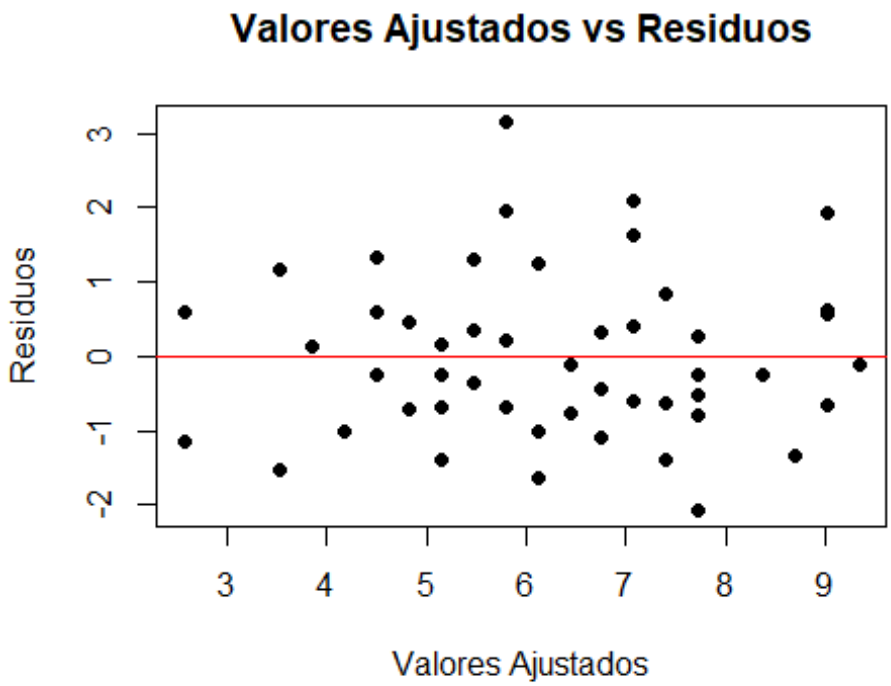
Dado un p value altisimo, > 0,05 podemos dejar en claro que la varianza de los errores pareceria ser constante, donde se cumple la homocedasticidad ya que no rechazamos la hipotesis nula

Independencia

Hipótesis nula H_0:: Los residuos no están correlacionados (independencia). Hipótesis alternativa H_1: Los residuos están correlacionados.

```
# Graficar los valores ajustados vs los residuos
plot(modelo_lambda_aprox$fitted.values,modelo_lambda_aprox$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)
```

```r
# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```

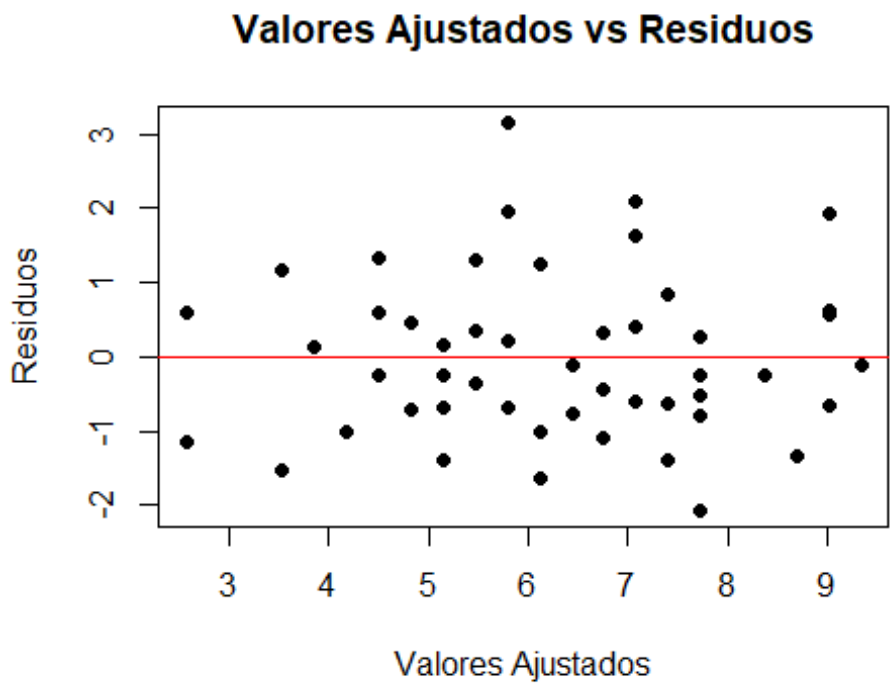



```r
dwtest(modelo_lambda_aprox)

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

De igual medida obtenemos un p value > 0.05 y que debemos recalcar que los p values obtenidos en estas transformaciones son mucho mas grandes que en la parte original, y dado eso no rechazamos hipotesis nula por lo que podemos ver que hay independencia

linealidad.

```r
# Graficar los valores ajustados vs los residuos
plot(modelo_lambda_aprox$fitted.values, modelo_lambda_aprox$residuals,
     main = "Valores Ajustados vs Residuos",
     xlab = "Valores Ajustados",
     ylab = "Residuos",
     pch = 19)

# Agregar una línea horizontal en y = 0
abline(h = 0, col = 'red')
```



Hipótesis nula H_0: La relación entre la variable dependiente e independiente es lineal.
Hipótesis alternativa H_1: La relación entre la variable dependiente e independiente no es lineal.

Observamos que los residuos se reparten sin un patron aparente y parece por el grafico cumplirse una lieanlidad de los mismos

Conclusion

Al considerar todas las conclusiones que hemos hecho tanto en la validez del modelo donde ya paso todas las pruebas asi como que mejoro un poco el coeficiente de determinicacion asi como que a tanto nivel indivual como en conjunto del modelo es altamente significativo podemos afirmar que hemos mejorado fuertemente con Boxcox en comparacion con el modelo simple original.

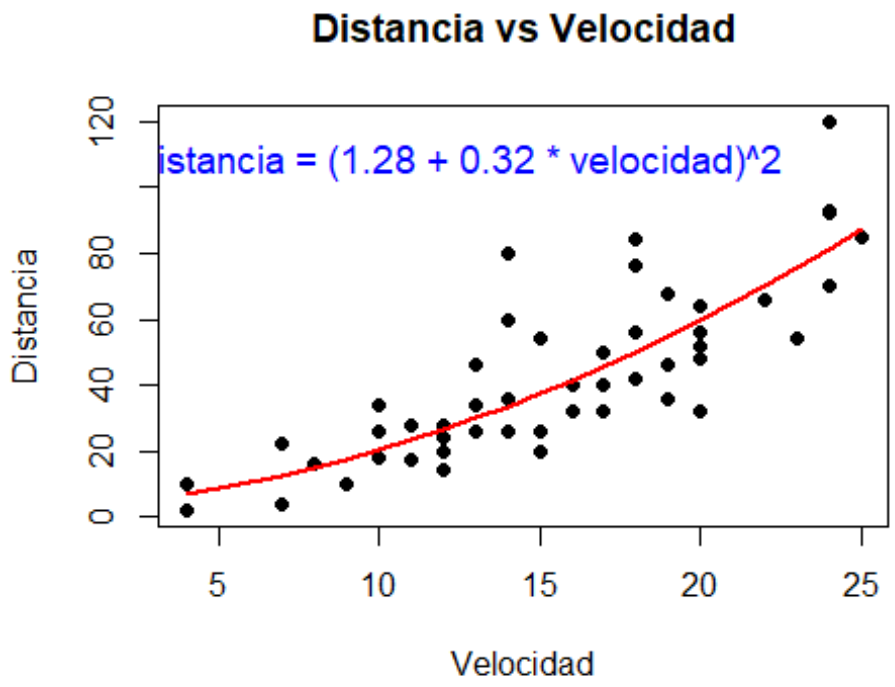
Despeja la distancia del modelo lineal obtenido entre la transformación y la velocidad. Obtendrás el modelo no lineal que relaciona la distancia con la velocidad directamente (y no con su transformación).

Dado que originalmente teniamos $\sqrt{(distancia)}$ en nuestro modelo, al despejarlo directamente obtendremos un cuadratico despejado:

$distancia = (1.27705 + 0.3224125 * velocidad)^2$

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

```r
# Coeficientes del modelo transformado
coeficientes <- coef(modelo_lambda_aprox)

# Despejar la distancia usando el modelo no lineal
vel <- cars$speed
dist_modelo <- (coeficientes[1] + coeficientes[2] * vel)^2

# Graficar los datos originales de distancia vs velocidad
plot(cars$speed, cars$dist,
     main = "Distancia vs Velocidad",
     xlab = "Velocidad",
     ylab = "Distancia",
     pch = 19)

# Agregar la curva del modelo no lineal
lines(vel, dist_modelo, col = "red", lwd = 2)

# Agregar la fórmula en el gráfico
formula <- paste0("distancia = (", round(coeficientes[1], 2), " + ",
round(coeficientes[2], 2), " * velocidad)^2")
text(x = max(cars$speed)*0.5, y = max(cars$dist)*0.9, labels = formula, col =
"blue", cex = 1.2)
```



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

Como lo hemos comentado durante todos estos pasos, si, si nos preguntamos si hubo una mejoria con esta tranformacion no solo en su significancia sino tambien respecto a la validez del modelo, absolutamentem tanto por las pruebas de normalidad pasadas, como la prueba de residuos, la media 0 de residuos, asi como homocedasticidad, linealidad e independencia, podemos ver que el modelo aproximado funciono significativamente mucho mejor que el modelo original, sin mencionar que tuvo algunas ligeras mejoras fuera de la validez sino en la significancia del modelo aumente a un 70.94% el coeficiente de determinacion sin mencionar un mejor desempeno visual en los QQ plots e histograma, por lo tanto fue una buena desicion haberla hecho con este modelo, lo cual seguramente tmb pudo haber ocurrido con el modelo exacto que fue insignificantemente pero que el aproximado.

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.

Si nos referimos como al del punto 1 como el modelo de regresion simple, y al del punto 2 que ya despejado vemos que pasa a un modelo cuadratico que es no lineal, por factores de validez del modelo en cuestion de pruebas de normalidad y de los supeustos de normalidad, el segundo modeloq eu acabamos de desarrollar se desempena significativamente mejor que el modelo del punto 1, sin mencionar que tambien cuenta con mejores estadisticos en cuestion de significancia, por lo tanto el mejor modelo para describir la relacion entre distancia y velocidad es el segundo modelo.

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

Datos atípicos: Es posible que haya valores atípicos en los datos que estén afectando el ajuste del modelo. Los datos atípicos pueden distorsionar tanto los coeficientes del modelo como los residuos, y, aunque el modelo pasa las pruebas de homocedasticidad e independencia, la presencia de estos datos podría influir negativamente en el coeficiente de determinación.

a transformación no lineal de raíz cuadrada, aunque útil para mejorar el ajuste, puede hacer que la interpretación de los coeficientes sea menos intuitiva.

Como podemos ver y aunque pasa las pruebas y supuestos de normalidad y todo, podemos ver que hay una brecha de mejora en cuanto al coeficiente de determinacion del modelo, ya que aunque tiene un 70.94% que si bien no es despreciable en el modelo no lineal, podria ser mejor.