Actividad 1.3 Normalidad univariada. Transformaciones para normalidad

Franco Mendoza Muraira A01383399

2023-11-04

```
## Warning: package 'lmtest' was built under R version 4.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.3

##
## Attaching package: 'zoo'

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

## Warning: package 'e1071' was built under R version 4.2.3

## Warning: package 'MASS' was built under R version 4.2.3

## Warning: package 'VGAM' was built under R version 4.2.3

## Loading required package: stats4

## Loading required package: splines

##
## Attaching package: 'VGAM'

## The following object is masked from 'package:lmtest':
##
##     lrtest

## Warning: package 'tseries' was built under R version 4.2.3

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

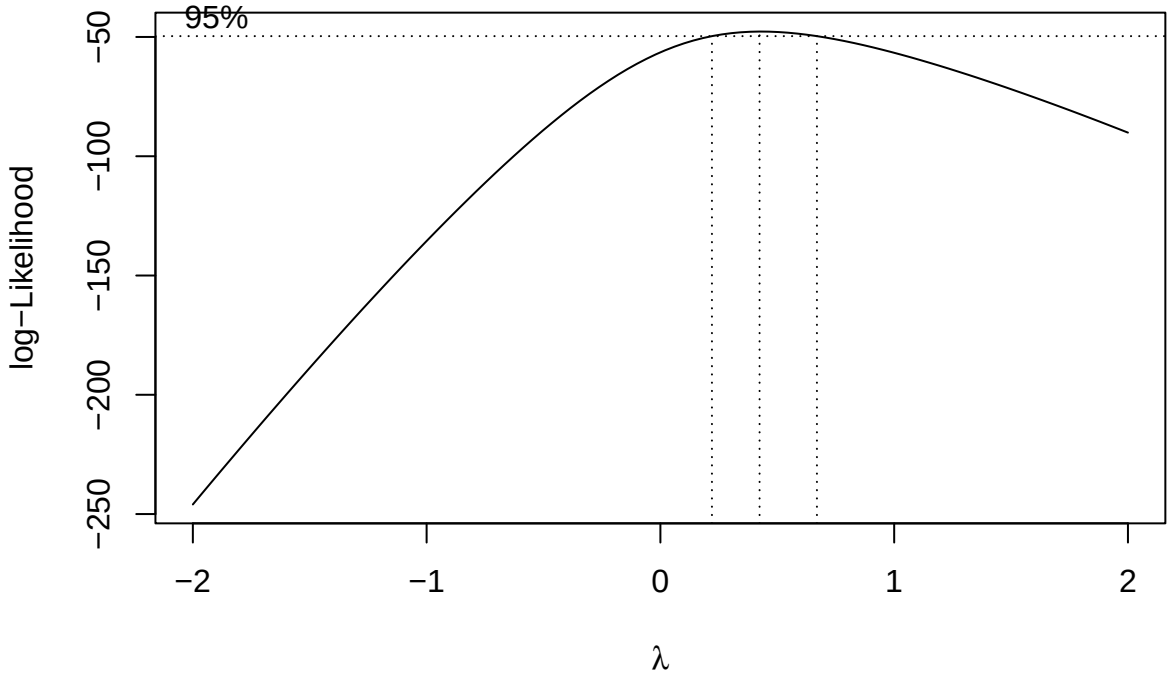
Parte 2

```r
y = cars$dist
x= cars$speed
```

1. Normalidad

Valor de lambda (Box-Cox)

```r
bc =boxcox(lm(y~x))
```




```r
l = bc$x[which.max(bc$y)]
l
```

```
## [1] 0.4242424
```

La transformacion aproximada es $d_1 = \sqrt{d}$, o el valor exacto es: $d_2 = \frac{d^{0.42}-1}{0.42}$.

Comparación de medidas

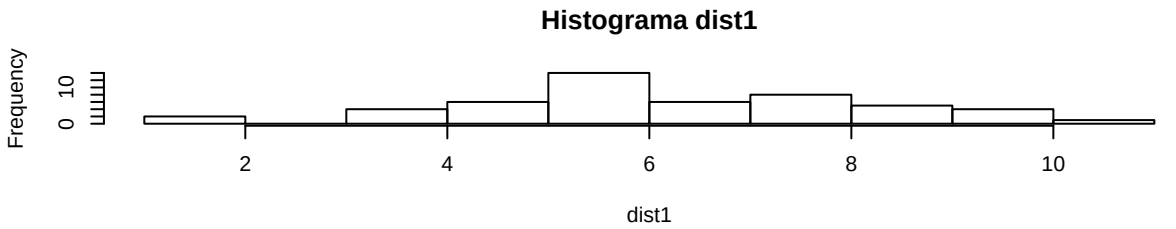
```
dist1= sqrt(y)
dist2 = (((y)^l)-1)/l
m0=round(c(as.numeric(summary(y)),kurtosis(y),skewness(y)),3)
m1=round(c(as.numeric(summary(dist1)),kurtosis(dist1),skewness(dist1)),3)
m2=round(c(as.numeric(summary(dist2)),kurtosis(dist2),skewness(dist2)),3)
m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)=c("Minimo","Q1","Mediana","Media","Q3","Máximo","Curtosis","Sesgo")
m
```

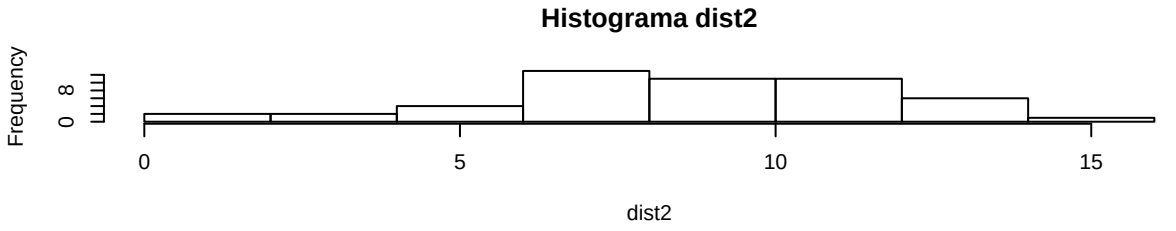
```
##                Minimo     Q1 Mediana  Media     Q3  Máximo Curtosis  Sesgo
## Original        2.000 26.000  36.000 42.980 56.000 120.000    0.119  0.759
## Primer modelo   1.414  5.099   6.000  6.242  7.483  10.954   -0.314 -0.019
## Segundo Modelo  0.806  7.033   8.423  8.712 10.646  15.609   -0.187 -0.170
```

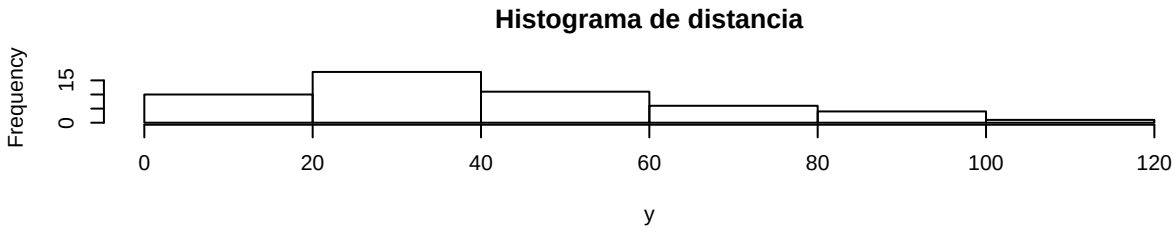
En las medidas de los 3 vectores de datos se pueden ver grandes diferencias, las más notables son la conversión a datos mucho más chicos en las transformaciones, para lo que también al buscar una distribución normal se acercaron más las medias y medianas de los 2 modelos nuevos por lo que se pueden esperar mejores resultados. También se puede ver una disminución grande en el sesgo de los datos comparado a los datos originales.

Histogramas de las transformaciones obtenidas, comparando con el original

```
par(mfrow=c(3,1))
hist(dist1,col=0,main="Histograma dist1")
hist(dist2,col=0,main="Histograma dist2")
hist(y,col=0,main="Histograma de distancia")
```







Pruebas de normalidad

H_0: Los datos tienen una distribución normal. H_1: Los datos no tienen una distribución normal.

α: 0.05

```
D0 =ad.test(y)
D1 = ad.test(dist1)
D2 = ad.test(dist2)
```

```
m0=round(as.numeric(D0$p.value),3)
m1=round(as.numeric(D1$p.value),3)
m2=round(as.numeric(D2$p.value),3)
m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)="Valor p"
m
```

```
##                Valor p
## Original         0.050
## Primer modelo    0.973
## Segundo Modelo   0.972
```

Con la prueba de Anderson-Darling, como se puede ver en la tabla se consiguieron los valores p de 0.05 en los datos originales, 0.973 en el primer modelo con la transformación aproximada, y de 0.972 con la transformaión exacta. Se ve con esta prueba que la transformacón aproximada nos da un valor p más alto.

Pero con estas 2 transformaciones no tenemos suficiente evidencia para rechazar H_0, por lo que se concluye que tienen una distribución normal.

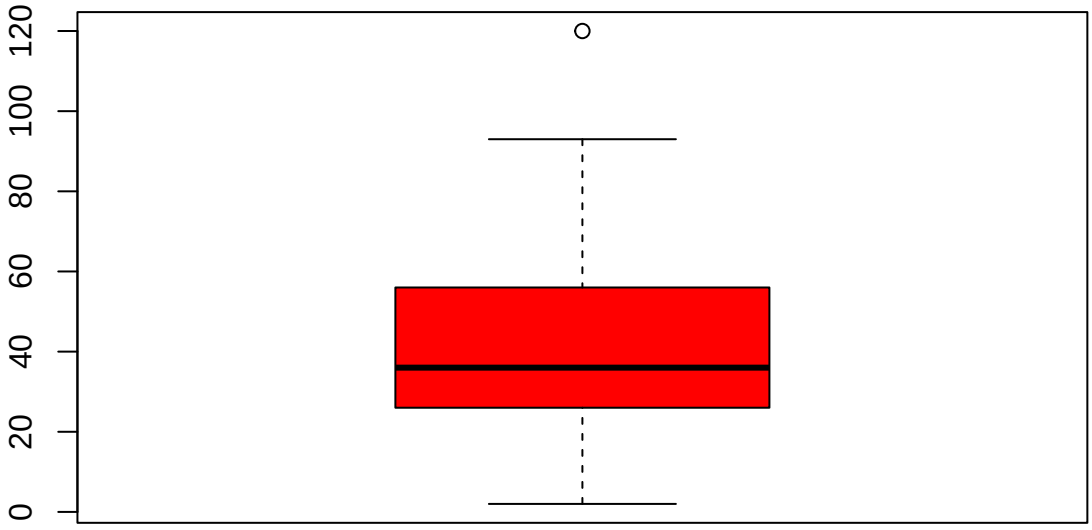
```
J0 =jarque.bera.test(y)
J1 = jarque.bera.test(dist1)
J2 = jarque.bera.test(dist2)
```

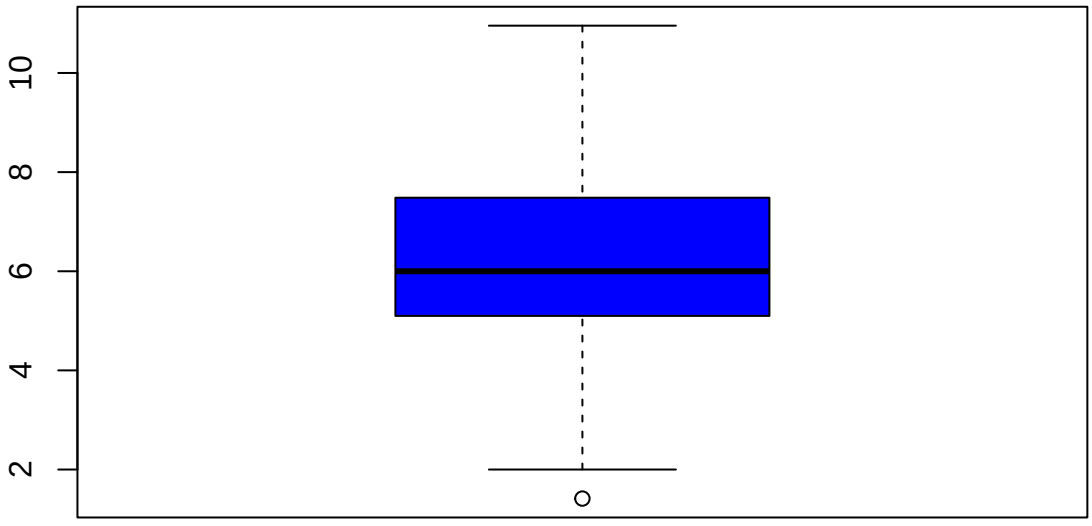
```
m0=round(as.numeric(J0$p.value),3)
m1=round(as.numeric(J1$p.value),3)
m2=round(as.numeric(J2$p.value),3)
m<-as.data.frame(rbind(m0,m1,m2))
row.names(m)=c("Original","Primer modelo","Segundo Modelo")
names(m)="Valor p"
m
```

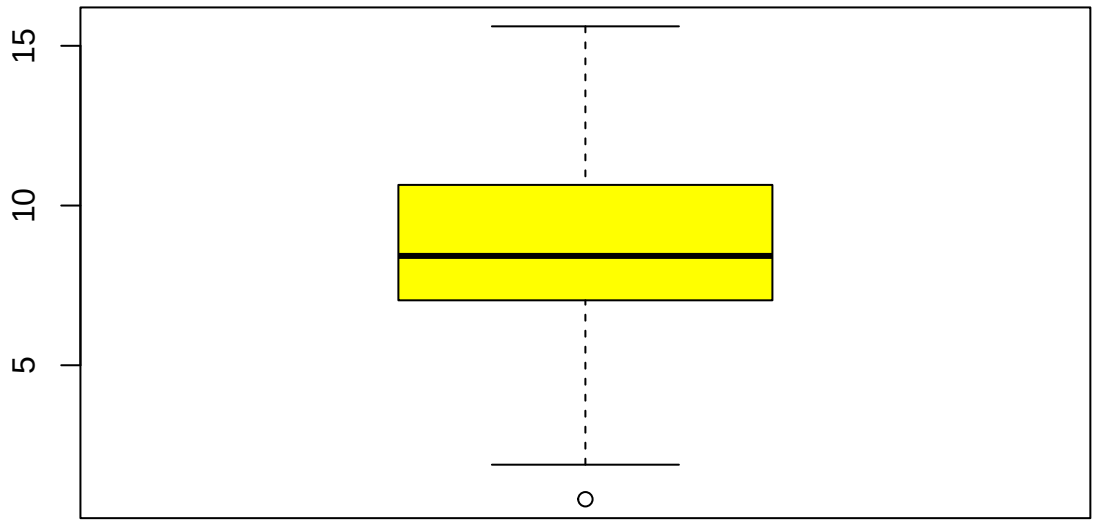
```
##                Valor p
## Original         0.073
## Primer modelo    0.956
## Segundo Modelo   0.875
```

Al igual que la prueba anterior, con la prueba de Jarque Bera, se pueden ver las diferencias del pvalue en las 3 bases de datos, las 2 transformaciones siendo las que no tienen evidencia para rechazar H_0, y las que presentan una distribución normal. También como la prueba pasada se ve que el primer modelo, el cuál es el aproximado, salió con el mejor pvalue.

Anomalías





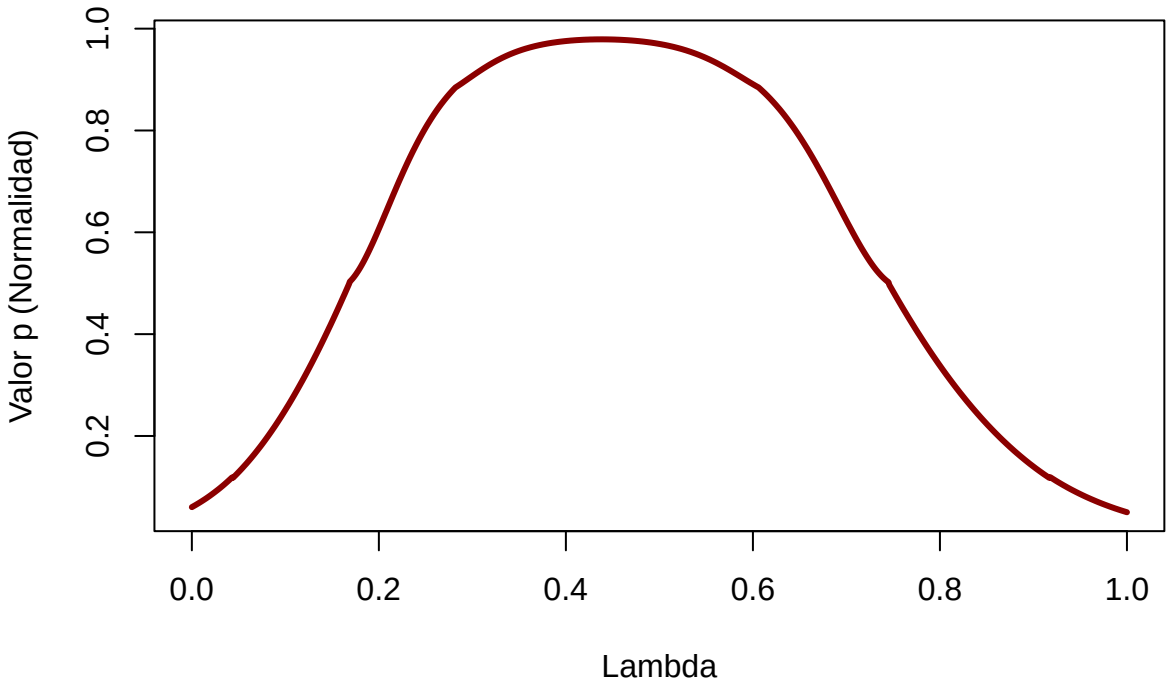


En los boxplots se puede ver solo un punto de anomalía en cada uno.

2. Yeo Johnson

Maximizacion del valor p

```
lp<- seq(0,1,0.001)
nlp <- length(lp)
n = length(y)
D <- matrix(as.numeric(NA),ncol = 2,nrow=nlp)
d<- NA
for (i in 1:nlp){
  d= yeo.johnson(y, lambda = lp[i])
  p=ad.test(d)
  D[i,]=c(lp[i],p$p.value)}
N=as.data.frame(D)
plot(N$V1,N$V2,
type="l",col="darkred",lwd=3,
xlab="Lambda",
ylab="Valor p (Normalidad)")
```



3. Mejor Transformación

```
G=data.frame(subset(N,N$V2==max(N$V2)))
l=G[,"V1"]
p = G[,"V2"]
cat("Lambda con el valor p más alto:",l)
```

```
## Lambda con el valor p más alto: 0.438
```

```
cat("\nSu valor p:",p)
```

```
##
## Su valor p: 0.9789807
```

```
dist3 = yeo.johnson(y,lambda=l)
```

Se encontró la lambda de 0.438, que tuvo el valor p más alto en la prueba de Anderson-Darling, con un valor de 0.979, esto se hizo iterando sobre todos los valores posibles de lambda entre 0 y 1. La ecuación de la transformación de Yeo Johnson es $d_3 = \frac{(d+1)^{0.438}-1}{0.438}$

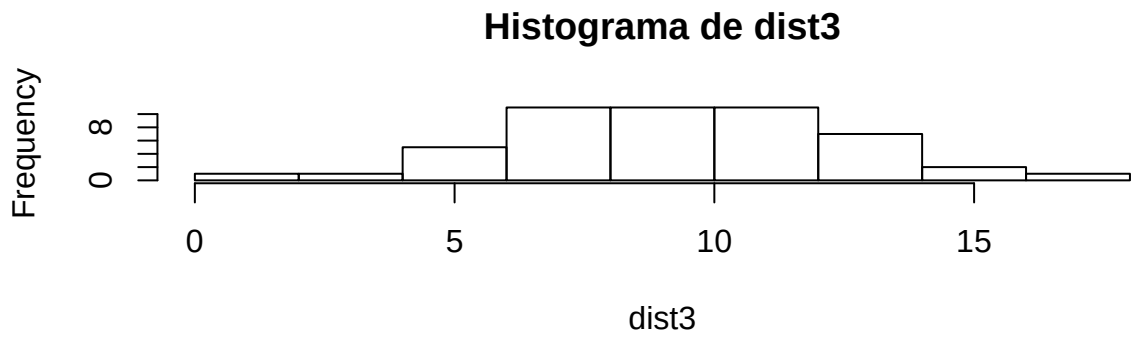
```r
ad.test(dist3)
```

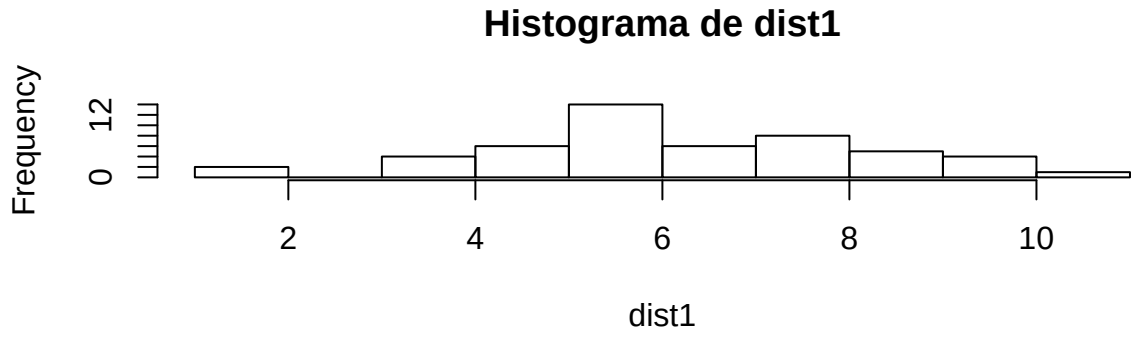
```
##
##  Anderson-Darling normality test
##
## data:  dist3
## A = 0.13282, p-value = 0.979
```

```r
jarque.bera.test(dist3)
```

```
##
##  Jarque Bera Test
##
## data:  dist3
## X-squared = 0.13843, df = 2, p-value = 0.9331
```

```r
par(mfrow=c(2,1))
hist(dist3,col=0,main="Histograma de dist3")
hist(dist1,col=0,main="Histograma de dist1")
```





Con la transformación de Yeo Johnson, pudimos observar que tuvo pvalues muy altas, y al igual que las otras transformaciones, debido a que rechazamos H_0 se concluye una distribución normal, pero a diferencia de las otras transformaciones debido a que tiene el pvalue más alto, podemos asumir que esta es la mejor transformación, buscando una distribución normal.

4. Regresión Lineal Simple

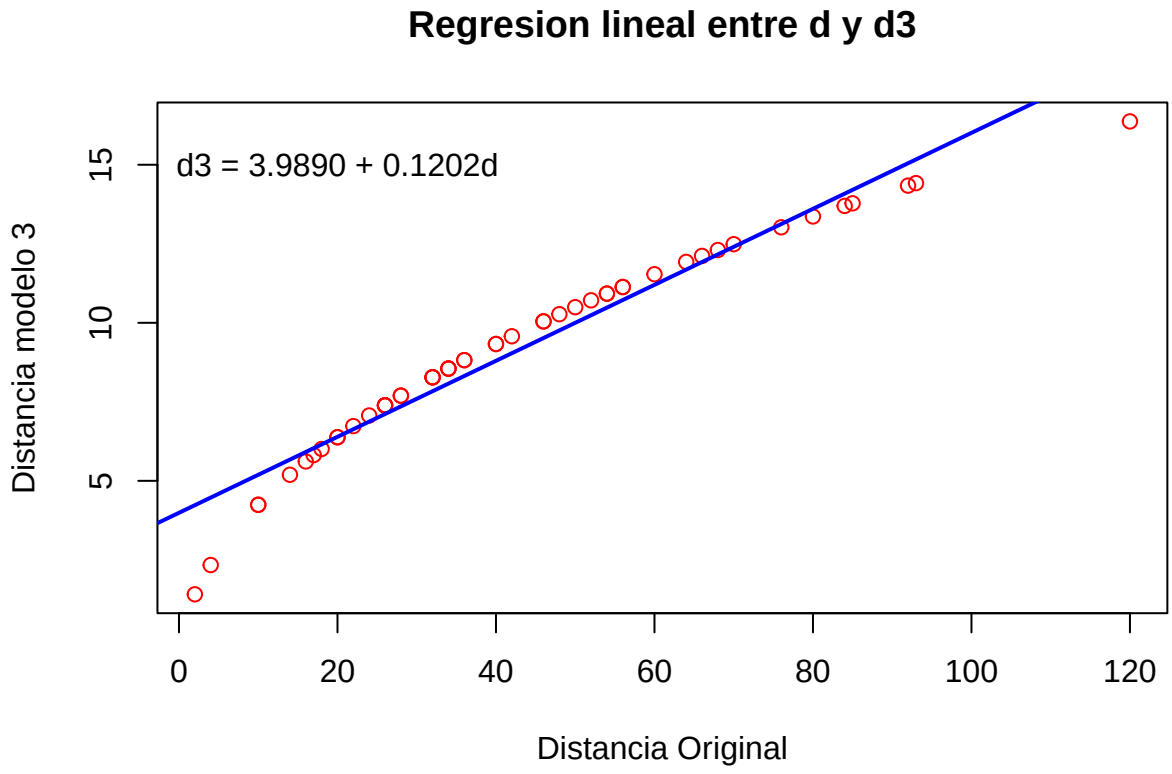
Modelo de Regresión Lineal transformación vs datos originales

```
A =lm(dist3~y)
A
```

```
##
## Call:
## lm(formula = dist3 ~ y)
##
## Coefficients:
## (Intercept)            y
##      3.9890       0.1202
```

El modelo de regresión lineal simple con la mejor transformación es $d_3 = 3.9890 + 0.1202d$

```
plot( y, dist3 ,col ="red",ylab ="Distancia modelo 3",xlab = "Distancia Original" ,main = "Regresion li
abline(A, col="blue" , lwd="2")
text( 20 , 15, "d3 = 3.9890 + 0.1202d")
```



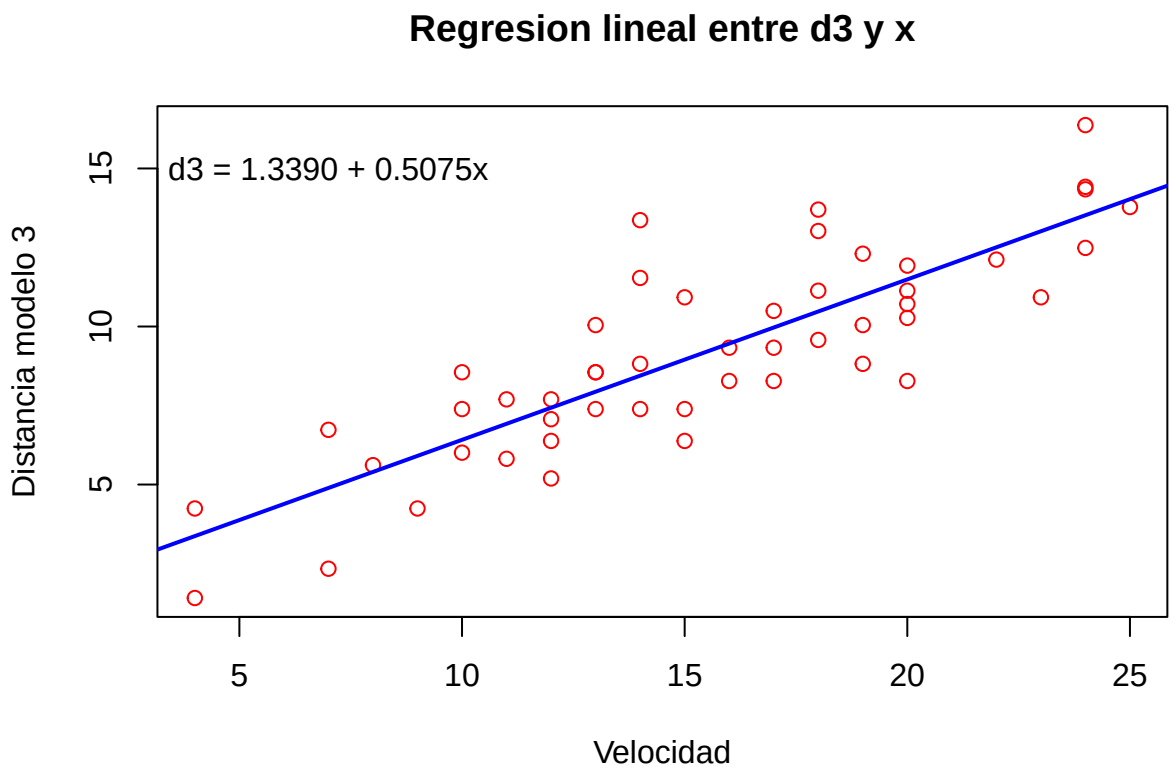
Modelo de Regresión Lineal transformación vs velocidad

```
B=lm(dist3~x)
B
```

```
##
## Call:
## lm(formula = dist3 ~ x)
##
## Coefficients:
## (Intercept)            x
##      1.3390       0.5075
```

El modelo de regresión lineal simple para la transformación d_3 en función de la velocidad x es $d_3 = 1.3390 + 0.5075x$, este modelo se puede ver en la siguiente gráfica

```
plot( x, dist3 ,col ="red",ylab ="Distancia modelo 3",xlab = "Velocidad" ,main = "Regresion lineal entre
abline(B, col="blue" , lwd="2")
text( 7 , 15, "d3 = 1.3390 + 0.5075x")
```




```
summary(B)
```

```
##
```

```
## Call:
## lm(formula = dist3 ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2137 -1.0952 -0.3026  0.8604  4.9196
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.33902    0.75845   1.765   0.0838 .
## x            0.50755    0.04663  10.885 1.47e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.726 on 48 degrees of freedom
## Multiple R-squared:  0.7117, Adjusted R-squared:  0.7057
## F-statistic: 118.5 on 1 and 48 DF,  p-value: 1.469e-14
```

El modelo de regresión es significativo en su conjunto, ya que el valor F y el bajo pvalue de 1.469e-14 indican que la variable predictora de la velocidad tiene un efecto significativo en la variable de respuesta que es la distancia transformada.

El coeficiente de determinación es 0.7117, lo que significa que alrededor del 71% de la variabilidad en la variable de respuesta es explicada por la variable predictora.

En resumen, el modelo es significativo, explica bien los datos y la variable de velocidad es una influencia significativa en la variable de respuesta.

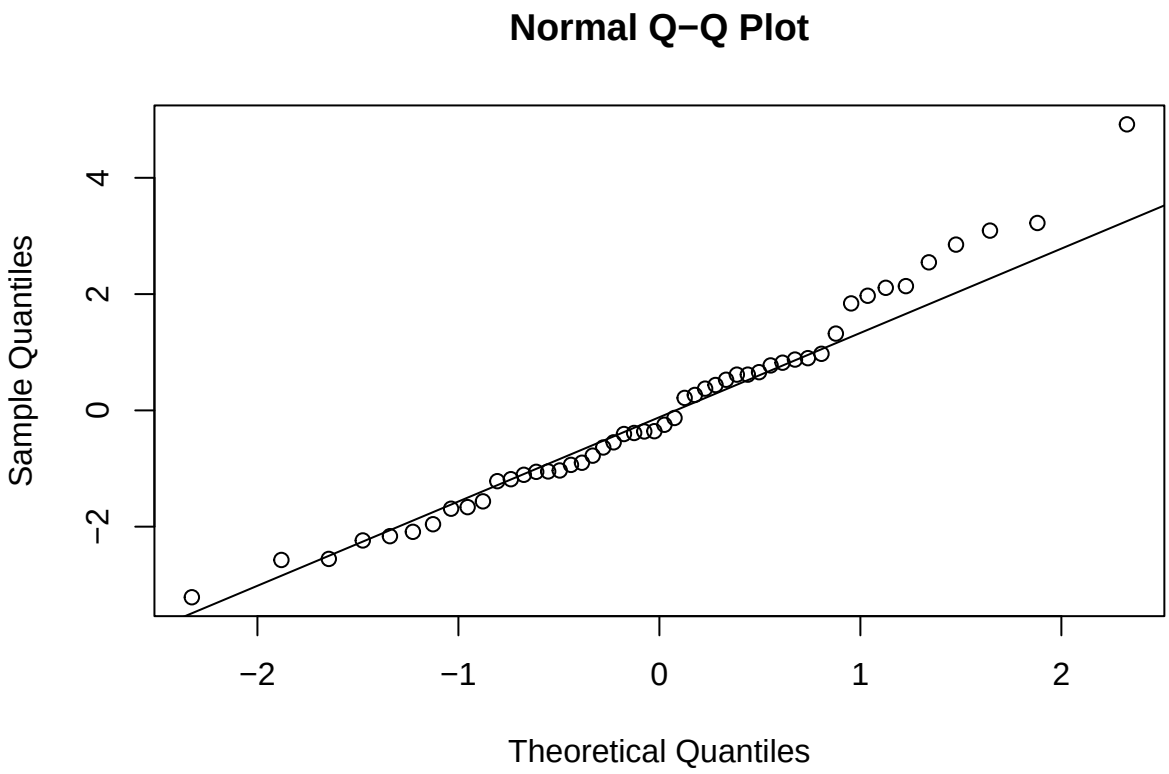
Validez del modelo

Normalidad H_0 : Los residuos siguen una distribución normal. H_1 : Los residuos no siguen una distribución normal.

```
ad.test(residuals(B))
```

```
##
##  Anderson-Darling normality test
##
## data:  residuals(B)
## A = 0.36914, p-value = 0.4143
```

```
qqnorm(residuals(B))
qqline(residuals(B))
```



Al hacer la prueba de Anderson-Darling para verificar normalidad, se obtuvo un pvalue de 0.4143, el cual es mayor que el α por lo que no se rechaza H_0 y se concluye que los residuos del modelo sí cumplen una distribución normal.

Homocedasticidad H_0 : La varianza de los errores es constante. H_1 : La varianza de los errores no es constante.

```
bptest(B)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  B
## BP = 0.011121, df = 1, p-value = 0.916
```

Debido a que el pvalue en la prueba Breusch-Pagan es muy alto con 0.916, no se tiene suficiente evidencia para rechazar la hipótesis nula, y se concluye que la varianza de los errores no es constante, es variable.

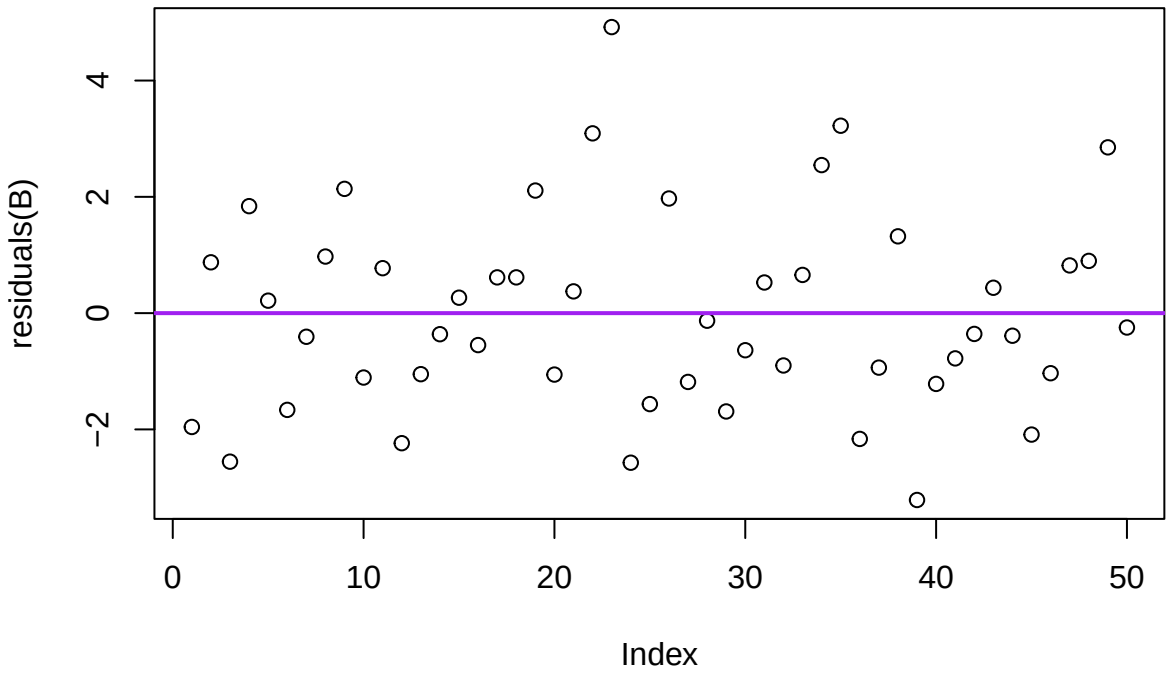
Independencia H_0 : autocorrelacion en los residuos $= 0$ H_1 : autocorrelacion en los residuos $\neq 0$

```
dwtest(B)
```

```
##
##  Durbin-Watson test
```

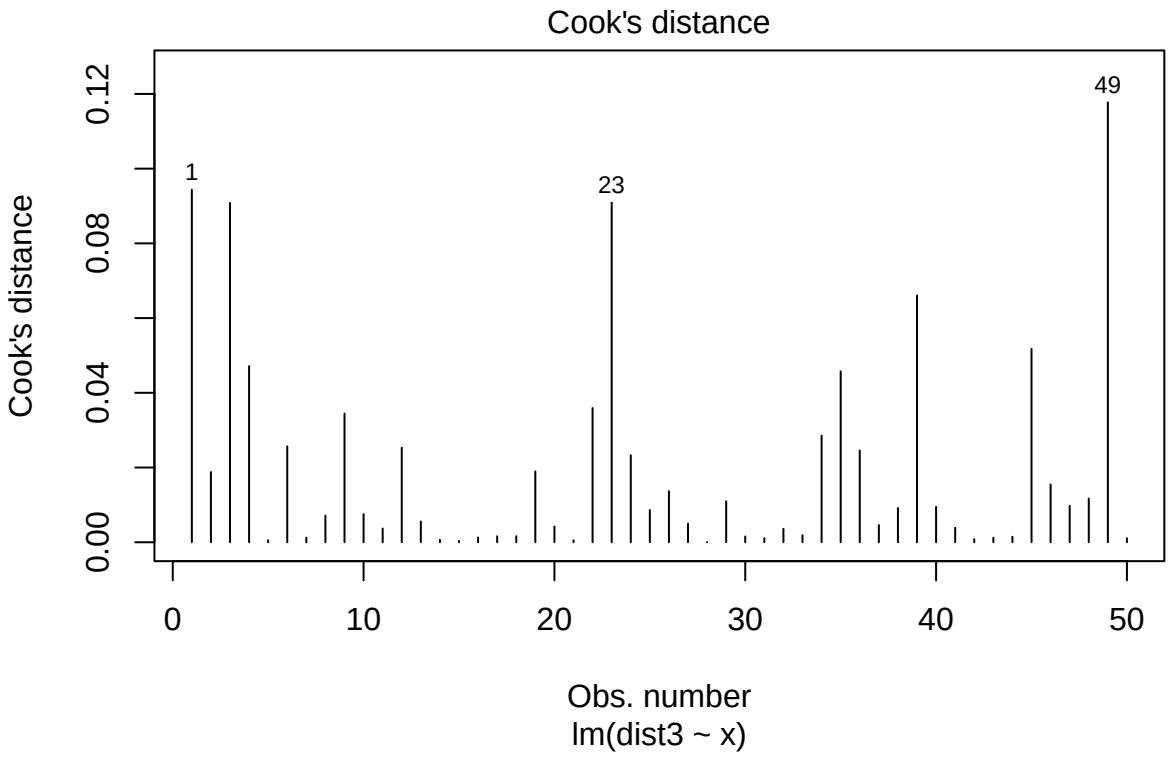
```
##
## data:  B
## DW = 1.9539, p-value = 0.3772
## alternative hypothesis: true autocorrelation is greater than 0
```

```
plot(residuals(B))
abline(h=0,col="purple",lwd=2)
```



El pvalue de nuestra prueba Durbin-Watson es alto, con un valor de 0.3772, sabiendo esto no podemos rechazar la hipótesis nula y se concluye que si hay autocorrelación en los residuos de nuestro modelo. Esto significa que los residuos tienen independencia.

```
z <- abs(scale(residuals(B)))
at <- which(z >2)
plot(B, which = 4)
```



Outliers

```
## [1] 23
```

En el gráfico de valores de Cook se puede ver que las observaciones influyentes son las observaciones 1, 23, y la 49, y según el z score, el único valor atípico que se encontró fue el de la observación 23. Esto significa que estas observaciones pueden tener un impacto significativo en el modelo ya que los valores atípicos no están dentro de la tendencia general de los datos , y las influyentes tienen una proporción más grande de influencia sobre el modelo alterandolo más que las demás observaciones. Estas observaciones se tienen que considerar al analizar el modelo.

Modelo NO Lineal

d: distancia

x: speed

Función de la mejor transformación (Yeo Johnson): $d_3 = \frac{(d+1)^{0.438}-1}{0.438}$

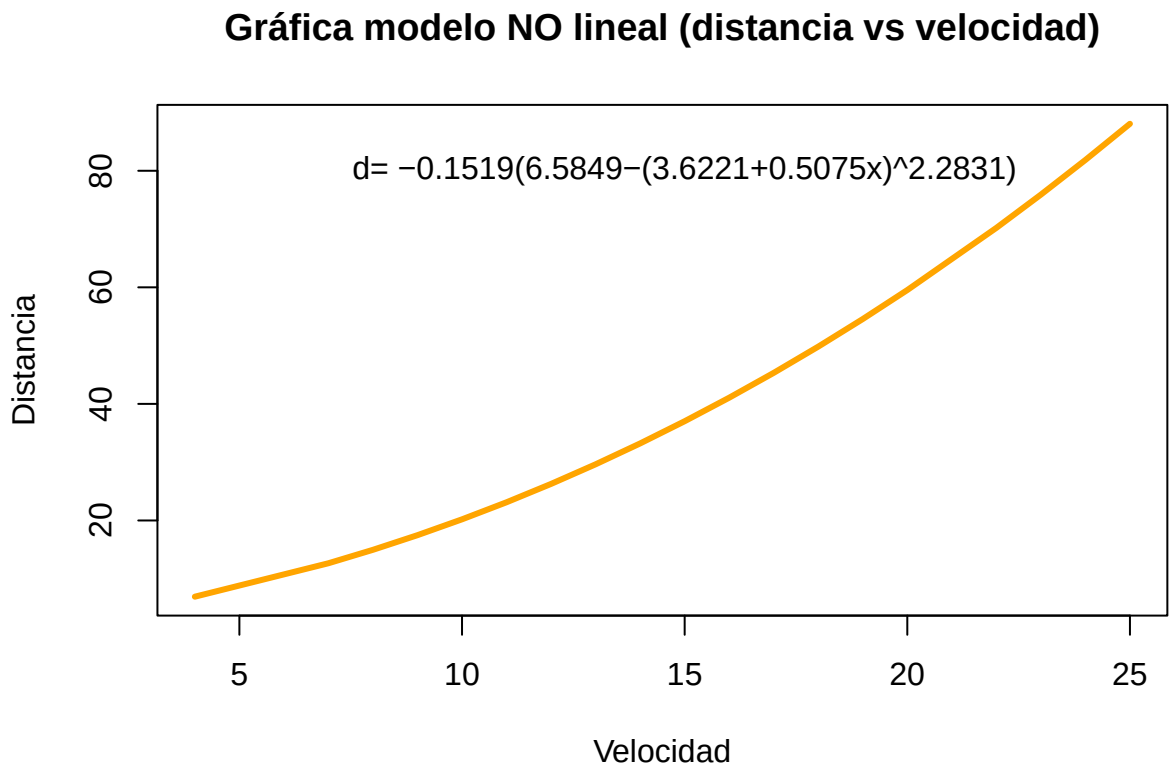
Función modelo lineal d3 vs speed: $d_3 = 1.3390 + 0.5075x$

Al igualar ambas funciones, tenemos lo siguiente: $\frac{(d+1)^{0.438}-1}{0.438} = 1.3390 + 0.5075x$, esto nos da el siguiente modelo no lineal:

$d = -0.1519(6.5849 - (3.6221 + 0.5075x)^{2.2831})$

```
func_d = function(x){return(-0.1519 *(6.5849 - (3.6221 + 0.5075*x)^2.2831))}
val_y= sapply(x,func_d)
```

```r
plot(x,val_y,xlab="Velocidad",ylab="Distancia",main="Gráfica modelo NO lineal (distancia vs velocidad)"
text( 15 , 80, "d= -0.1519(6.5849-(3.6221+0.5075x)^2.2831)")
```



5. Conclusiones

En esta actividad se hicieron transformaciones con 2 diferentes métodos, usando las transformaciones de Box-Cox y consiguiendo su lambda con el pvalue de normalidad más alto, y también se usaron las transformaciones de Yeo Johnson, de la cual también sacamos su mejor lambda. De estos 2 métodos terminamos encontrando que el que daba mejores resultados era el de Yeo Johnson debido a su pvalue tan alta.

Probamos esta transformación con un modelo de regresión lineal, revisando diferentes elementos, como sus medidas generales, la independencia, normalidad y homocedasticidad de sus residuos para poder hacer un análisis completo del modelo creado.

En resumen se pudo aprender sobre las transformaciones, y lo importante que son para poder usar datos de manera más fácil, y con estas crear modelos válidos. Pudimos comparar diferentes transformaciones, y sus métodos aproximados y exactos para tener una imagen completa del tema, al igual que aprender sobre hacer modelos de calidad..