Análisis de la duración de procesamiento de canciones de acuerdo a su duración para la realización de un experimento de Montecarlo con herramientas adquiridas en los análisis.

Quesada Quesada, Esteban Jesús
Escuela de Computación e Informática
Universidad de Costa Rica
estebanq52@gmail.com

Resumen – En el presente trabajo se estima la probabilidad de que una computadora, trabajando sin descanso y con una demanda continua e inagotable de solicitudes, pueda generar ventas por un millón de dólares en un año utilizando el método de Montecarlo, para la realización del trabajo se utiliza el software denominado RStudio mediante el cual se realizan todos los cálculos necesarios para estimar la probabilidad mencionada. Como resultado se obtiene un estimado que afirma que la computadora puede efectivamente generar incluso más de un millón de dólares en cada una de las simulaciones realizadas en el experimento de Montecarlo, asimismo, se concluye que el procesamiento de las canciones depende en gran medida de la duración de las mismas.

I. INTRODUCCION

El método de Montecarlo proporciona soluciones aproximadas a una gran variedad de problemas matemáticos posibilitando la realización de experimentos con muestreos de números pseudoaleatorios en una computadora. En el presente trabajo se desea realizar un experimento de Montecarlo para estimar la probabilidad de que una computadora trabajando sin descanso y con una demanda continua e inagotable de solicitudes puede generar ventas por un millón de dólares. Se generarán tiempos aleatorios de canciones mediante una distribución gamma, beta o normal, así como un cálculo del tiempo que lleva procesar las canciones debido a su determinada duración mediante el uso de un modelo lineal. Se realizarán N simulaciones que permitirán realizar un estimado del problema mencionado anteriormente.

Como primer objetivo se desea conocer la distribución de la duración de las canciones más populares, analizando la base de datos Million Song Dataset (MSD) seleccionando únicamente las canciones populares cuyos nombres de artistas creadores comiencen con la letra Q. Analizando los datos seleccionados se generará un modelo beta, gamma o normal para poder generar tiempos de canciones posteriormente. Como estrategia se estudiará en primer lugar si la duración de las canciones resulta tener una distribución normal. Luego se procederá a eliminar los valores atípicos y realización de nuevo de la prueba de normalidad. Para finalmente modelar los datos como una distribución gamma en caso de que la prueba de asimetría resulte ser positiva luego de eliminar los valores extremos atípicos.

Un segundo objetivo es realizar estudios sobre el tiempo de procesamiento de canciones de acuerdo a su duración, dichos tiempos de procesamiento ya calculados se inspeccionarán del Cuadro 1: Tiempo de Procesamiento de Polynizer para algunas canciones. El estudio de los tiempos de procesamiento permitirá generar un modelo lineal que será de gran utilidad para el experimento de Montecarlo, utilizándose para averiguar cuál será el tiempo de procesamiento de canciones con tiempos generados mediante el primer modelo mencionado. Una estrategia para lograr este objetivo resulta ser la realización de un diagrama de dispersión para observar la relación entre duración de canción y su tiempo de procesamiento, posteriormente se calculará la regresión lineal y será importante también calcular el coeficiente de determinación R cuadrado. Asimismo, obtener los valores de la pendiente y la intercepción de la regresión será necesario para definir el modelo lineal. Para que por último se realicen pruebas de contraste de hipótesis para demostrar que ni la pendiente ni la intercepción son nulas.

El último objetivo será entonces llevar a cabo una simulación de Montecarlo utilizando algunos recursos de las etapas realizadas anteriormente, esto con el fin de estimar la probabilidad de que la computadora mencionada, trabajando sin descanso y con una demanda continua e inagotable de solicitudes, pueda generar ventas por un millón de dólares en un año. Como estrategia se utilizará el modelo obtenido en la primera parte, es decir la distribución gamma obtenida para la generación de tiempos de canciones, asimismo, se recurre al modelo lineal obtenido en la parte 2 para así generar los tiempos de procesamiento respectivos de cada canción con una duración determinada. Por último, se realizarán cálculos de valores

esperados con los datos obtenidos de la simulación.

II. METODOLOGÍA

Los cálculos realizados para el cumplimiento de los objetivos se llevaron a cabo en una computadora con un procesador Intel Core i5 7 500, y 16 GB de memoria RAM. Donde el sistema operativo de la máquina es Windows. Asimismo, las pruebas se llevaron a cabo en RStudio versión 1.4.1 106, el cual es un entorno de desarrollo integrado para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Asimismo, se utilizó SQLite para poder extraer de la base de datos todas las canciones cuyos nombres de artistas comenzaran con la letra Q, y así producir un csv con tales datos.

Dentro de las librerías que se utilizaron en RStudio se encuentran las siguientes: en primer lugar, se utilizó readr para lidiar con la lectura en archivos planos grandes rápidamente. El paquete proporciona reemplazos para funciones como *read.table() y read.csv()*. Asimismo, se utiliza la librería denominada Nortest, este paquete proporciona cinco pruebas ómnibus para probar la hipótesis compuesta de normalidad. La librería Car, también se incluye, ésta contiene funciones y conjuntos de datos asociados con el libro "An R Companion to Applied Regression", tercera edición, de John Fox y Sanford Weisberg.

La librería Moments se utiliza principalmente para poder utilizar la función denominada skewness, función necesaria para calcular la asimetría. Uno de los paquetes más importantes es fitdistrplus, el cual proporciona funciones para ajustar distribuciones univariadas a diferentes tipos de datos (datos continuos censurados o no censurados y datos discretos) y permite diferentes métodos de estimación (máxima verosimilitud, coincidencia de momentos, coincidencia de cuantiles y estimación de bondad de ajuste máxima). Por último, se utiliza además la librería plotrix que contiene herramientas para el trazado de datos en R, es decir para graficar datos.

I Parte Funciones Utilizadas

Para extraer los datos especificados, es decir los datos de las canciones cuyos artistas comienzan con la letra Q, fue necesario utilizar el comando: *sqlite3 track_metadata.db "SELECT artist_name, title, duration FROM songs WHERE artist_name LIKE 'Q%';"*, asimismo, se redirigió la salida a un csv para posteriormente poder leer el csv en RStudio y comenzar el análisis de los datos de las canciones seleccionadas. La primera función a utilizar se denomina hist(), mediante la misma se logra graficar las distintas duraciones de las canciones para analizar a simple vista cuál distribución parece tener. Se utiliza el parámetro break de la función hist para poder observar los datos de una forma más suave, definiendo *breaks = 300*.

Puesto que el gráfico no indica con completa certeza la distribución de las duraciones de las canciones, se recurre a las pruebas de normalidad, las cuales se llevan a cabo con las funciones *ad.test(duration)* exportada de la librería nortest, y la función *shapiro.test(duration)*, la cuál es una función default ya incluida en R para poder realizar pruebas de normalidad de las duraciones de las canciones, las cuales se pasan con el nombre de "duration". Si con la función se obtiene un p-value menor que *0.05* se dice que la distribución no es normal.

Luego, para continuar comprobando la normalidad se lleva a cabo otra prueba mediante la función *qqPlot(duration)*, permitiendo observar cuan cerca está la distribución de un conjunto de datos a alguna distribución ideal. Posteriormente es necesario observar si la distribución de las duraciones parece tener valores extremos atípicos, por lo que se realizará un diagrama de cajas usando la regla 1.5 IQR para poder mostrar tales valores. En R este diagrama se realizó empleando el siguiente código: *boxplot(duration,range = 1.5)*

El criterio IQR significa que todas las observaciones por encima de *q0.75 +1.5*IQR* o por debajo de *q0.25-1.5*IQR* (donde q0.25 y q0.75 corresponden al primer y tercer cuartil respectivamente, y IQR es la diferencia entre el tercer y primer cuartil) son consideradas como valores atípicos potenciales por R. En otras palabras, todas las observaciones fuera del siguiente intervalo se considerarán como valores atípicos potenciales:
*I = [q0.25-1.5*IQR; q0.75+1.5*IQR]*

Gracias a las funciones utilizadas en el fragmento de código se podrán observar por lo tanto estos valores atípicos extremos, demostrando que quizás sea necesaria la eliminación de los mismos para ver si se cumple cierta normalidad. La eliminación de los valores atípicos se realiza mediante las siguientes funciones:

```
Q <- quantile(duration, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(duration)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
eliminated<- subset(outputData, duration > (Q[1] - 1.5*iqr) &
duration < (Q[2]+1.5*iqr))
```

Mediante la función "subset" se toman los valores que no exceden el rango de IQR, es decir los valores que no son valores extremos atípicos y se almacenan en "eliminated", nombre que hace referencia a que los valores extremos atípicos se han eliminado.

De esta forma quedan eliminados aquellos valores extremos atípicos y se almacenan en "eliminated" para llevar a cabo una nueva prueba de normalidad con las funciones antes utilizadas para llevar a cabo la primera prueba de normalidad, es decir: *ad.test y shapiro.test*. Si al calcular la prueba de normalidad se obtiene un resultado de que la distribución no es normal, se realiza un cálculo de la asimetría, mediante la función *skewness(eliminated$duration)*, esto con el fin de determinar en caso de que dé positivo que se debe realizar un modelado de los datos como una distribución gamma, y de ser en caso contrario, es decir negativo, se deben modelar los datos con una distribución beta. Para modelar los datos con una distribución gamma en R se utiliza el comando:
(fit.gamma <- fitdist(eliminated$duration, distr = "gamma", method = "mle")). Posteriormente, sí seguimos suponiendo que la distribución utilizada fue la gamma obtenemos los parámetros de la distribución gamma mediante la función *summary(fit.gamma)*, donde *fit.gamma* es la distribución gamma antes calculada. Asimismo, si deseamos modelar la distribución gráficamente se utiliza la función *plot(fit.gamma)*.

II Parte Funciones Utilizadas

En primer lugar, para esta parte se realiza un diagrama de dispersión en R mediante el cual se analiza la relación existente entre la duración de las canciones y el tiempo que demora la computadora en calcular su tiempo de procesamiento. Se utiliza la función:
plot(x= proccesTime$Seconds,y =proccesTime$`Processing (s)`) para realizar el diagrama de dispersión, así como la función:
abline(lsfit(proccesTime$Seconds,proccesTime$`Processing (s)`)). De esta forma se refleja de una forma gráfica e incluso más clara la relación entre la duración y el tiempo de procesamiento de las canciones. Posteriormente se debe llevar a cabo una regresión lineal entre los datos, la cual se calcula con la función en R denominada *lmProcessTime = lm(`Processing (s)`~Seconds, data = proccesTime)*, a esta al igual que las funciones anteriores se le da como parámetro la duración del procesamiento de las canciones en segundos y los segundos de duración de las canciones. Asimismo, se realiza el cálculo del coeficiente de determinación R cuadrado para de esta forma asegurarnos si efectivamente el tiempo de la duración de la canción es un buen predictor de la duración del procesamiento de la misma. En este caso al ser positivo el valor mencionado (coeficiente de determinación R cuadrado) se deben calcular los valores estimados de la intercepción y la pendiente mediante la función *summary(lmProcessTime)*, que devuelve un resumen de los valores obtenidos en la regresión lineal. Posteriormente, se debe demostrar por medio de contraste de hipótesis que los valores intercepción y pendiente no son nulos. Por lo tanto, se declara la hipótesis nula que afirma que los valores *M* y *B* son iguales a *0* mientras que también se declara la hipótesis alternativa donde se menciona que los valores no son nulos, es decir son distintos de *0*. Para buscar contrastar la hipótesis nula se recurre al principio matemático que menciona que si *y* es igual para todo *x* elegido en la fórmula *y = M*x + B* entonces *M* definitivamente es nulo, es decir *M = 0*, por lo tanto, podemos generar ciertos *x* y comprobar si la *y* es distinta para cada uno de ellos. Por lo tanto, la forma de negar la hipótesis nula que menciona que *M = 0*, resulta ser tomar dos *x* distintas y calcularlas en la función *y = M*x + B*, si estas dan un resultado distinto quiere decir entonces que *M != 0*. Por otra parte, para demostrar que *B* es distinto de *0*, recordemos que si *y(x = 0)* no es cero, entonces *B* no es *0*, debido a que *B = y(0)*.

III Parte Funciones Utilizadas

Para el experimento de Montecarlo que se va a realizar se desea que el mismo cuente con un error no mayor a *0.05* y probabilidad *0.9*. Se debe iniciar calculando el número de simulaciones, es decir calcular *N* con los parámetros ya brindados, este cálculo se realiza en R de la siguiente forma: *(N <- 0.25*(qnorm(0.95)/0.05)^2)*, basando en la fórmula:

$$N \geq 0.25 \left(\frac{z_{\alpha/2}}{\varepsilon} \right)^2$$

Posteriormente, para llevar a cabo las simulaciones entonces se plantea en R un código con la siguiente estructura:

```
While i < N{
   Tiempo = 0
   While(Tiempo > año en segundos){
         Generar tiempo de canción
         Generar tiempo de procesamiento
         Almacenar dinero total generado
         Demás cálculos…
         Tiempo  += tiempo de procesamiento;
    }
    i++
}
```

Como se puede observar en la estrategia empleada el número de simulaciones serán *N*, y cada simulación no se detendrá hasta que la computadora haya gastado un año entero procesando canciones. En cada iteración se genera una duración de una canción utilizando el modelo obtenido en la parte 1, es decir se utiliza una distribución gamma con parámetros *shape = 5.86283427, rate = 0.02288505*.

La siguiente función de R nos permite calcular la distribución mencionada y generar los tiempos de las canciones: *(duracionCancion <- rgamma(1, shape = 5.86283427, rate = 0.02288505))*. Posterior a esto, una vez generado el tiempo que tarda la canción, se debe calcular cuánto se tarda la computadora procesando tal duración, por lo tanto, se recurre al modelo lineal calculado en la parte 2, el cual tiene parámetros *M<- 0.09772 (pendiente) y B<- -2.77785 (intercepción)*. En R la fórmula usada para calcular el tiempo de procesamiento es:
*tiempoDeProcesamiento= M*duracionCancion+B.*

Por otra parte, para llevar la cuenta de si la computadora puede generar ventas por más de un millón de dólares se establecen variables donde se almacena el precio de acuerdo a las condiciones, es decir si la canción dura menos de *360 segundos*, esta tendrá un valor de *0.99 dólares*, mientras que, si dura más, por cada vez que se pasen los *360 segundos* se cobrarán *0.99 dólares* más. En cuanto a la estrategia para tener un orden con los datos de cada simulación, se crea un vector de N casillas, es decir de tamaño igual al número de simulaciones, donde en cada casilla, es decir cada simulación se almacena el total de dinero generado gracias a las ventas. Asimismo, se crean variables que se encargan de almacenar y llevar un registro de cuantas canciones que exceden los *360 segundos* fueron generadas, así como el dinero generado solo con las canciones que exceden los *360 segundos*.

Luego de las simulaciones de Montecarlo se desean conocer aspectos como el valor esperado de las ventas, este se puede calcular con la ayuda del vector en el que se almacenaron las ventas por cada simulación. Se utiliza la función en R: *mean(ventasN)* arrojando así el valor esperado de las ventas que se realizaron. Por otra parte, se desea conocer el error estándar de las ventas, el cual se puede calcular de la siguiente función: *sd(ventasN)/sqrt(N)*, es decir la desviación estándar de las ventas dividida entre el total de número de ventas que se hicieron. Para calcular la fracción de las de canciones que exceden los *6 minutos*, se utiliza la variable creada

cancionesExceden6, en la cual se almacenaron el número de canciones mayores que *6 minutos*, y esta variable se divide entre el total de las canciones generadas, obteniendo la fracción de canciones que exceden los *6 minutos* con la función en R:(fraccionCancionesEsperada<cancionesExceden6/totalCanciones). Ahora para calcular la fracción de las ventas de canciones de más de *6 minutos*, se utiliza la variable dineroExcedente6, en el cual se almacena el dinero generado con las canciones que exceden los *6 minutos* y se dividen entre el dinero total generado con todas las canciones. Por último, para averiguar el valor esperado de las ganancias después de cobrar las comisiones de las empresas, que resulta ser de *30%* cuando las ganancias exceden el millón de dólares y de *15%* cuando las ganancias son menores al millón de dólares se debe crear un vector donde se guarde el dinero total generado por cada simulación menos las comisiones cobradas de las ventas, los valores que se guardan en este vector se calculan de la siguiente forma:

```
ganancias <- rep(0,N)
for(i in 1:N){
      if(ganancias[i] > 1000000){
            ganancias[i] <- ventasN[i]- (ventasN[i]*0.30)
      }
      if(ganancias[i] < 1000000){
            ganancias[i] <- ventasN[i]- (ventasN[i]*0.15)
      }

}
```

Una vez se ha calculado las ganancias después de descontar las comisiones cobradas, se realiza un mean(ganancias) para lograr obtener el valor esperado de las ganancias.

III. Resultados

Resultados Primera Parte:
Resultado 1:

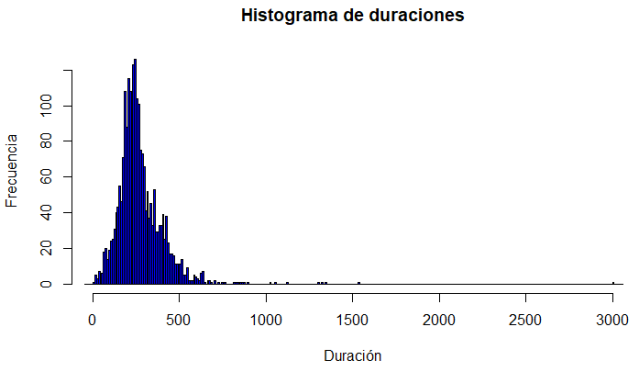

Fig. 1. Distribución de las duraciones de las canciones cuyos artistas poseen un nombre que comienza con la letra Q representadas con un histograma. Datos obtenidos de la base de datos del MSD.

Resultado 2:
Prueba de normalidad a la distribución de duración de las canciones:
Hipótesis:
H0: La muestra proviene de una distribución normal
H1: La muestra no proviene de una distribución normal

El nivel de significancia que se trabajará es de *0.05*. *Alfa = 0.05*
Criterio de Decisión

Si p-value < Alfa Se rechaza H0
Si p-value >= Alfa No se rechaza H0
Según la prueba de normalidad de "Anderson-Darling normality test" *p-value < 2.2e-16*
Según la prueba de normalidad de "Shapiro-Wilk normality test" *p-value < 2.2e-16*

Resultado 3:

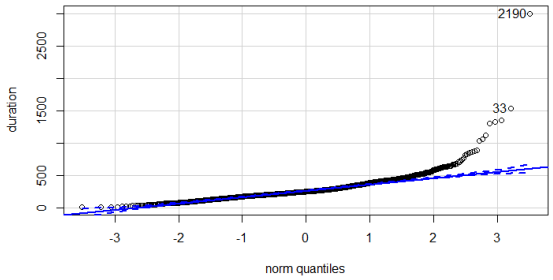

Fig. 2. Gráfico Q-Q sobre las duraciones de las canciones cuyos artistas poseen un nombre que comienza con la letra Q. Datos obtenidos de la base de datos del MSD.

Resultado 4:

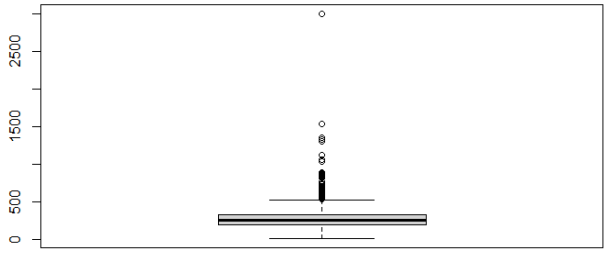

Fig. 3. Diagrama de cajas sobre las duraciones de las canciones cuyos artistas poseen un nombre que comienza con la letra Q. Datos obtenidos de la base de datos del MSD. En la figura se muestran algunos valores extremos atípicos.

Resultado 5:
Una vez eliminados los valores atípicos se vuelve a realizar una prueba de normalidad, la cual da como resultado: *p-value = 2.244e-16*.

Resultado 6:
Dado que los resultados de la prueba de normalidad en el resultado 5 denotaron una distribución no normal, se calcula la asimetría, la cual da como resultado: *0.4014163*, es decir una asimetría positiva.

Resultado 7:
Tabla 1: Parámetros obtenidos de la distribución gamma:

	estimate	Std. Error
shape	*5.86283427*	*0.1685970610*
rate	*0.02288505*	*0.0006857982*

Resultados Segunda Parte:
Resultado 8:

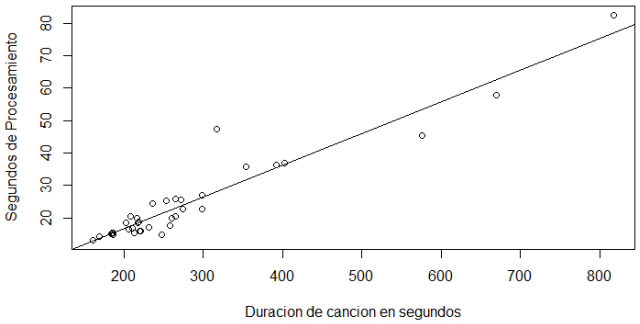

Fig. 4. Diagrama de dispersión de la duración de canciones y su tiempo de procesamiento, donde se muestra una relación aparente de crecimiento de tiempo de duración de procesamiento conforme más duran las canciones.

Resultado 9:
El resultado 9 consiste ser un resumen sobre la regresión lineal realizada a los datos "Tiempo de Procesamiento de Polynizer para algunas canciones", donde los resultados de la función summary, son los siguientes:

Tabla 2: Residuales obtenidos de la regresión lineal

Min	1Q	Median	3Q	Max
-8.225	*-2.727*	*0.056*	*1.534*	*19.134*

Tabla 3: Coeficientes obtenidos de la regresión lineal

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	*-2.77785*	*1.74627*	*-1.591*	*0.121*
Seconds (Pendiente)	*0.09772*	*0.00554*	*17.640*	*<2e-16* ***

Resultado 10:
En primer lugar, se debe de realizar el cálculo de correlación antes de obtener el coeficiente de determinación R cuadrado, el coeficiente de correlación da como resultado: *0.9508482*. Posteriormente una vez se obtiene este resultado se procede a elevar al cuadrado para obtener el coeficiente de determinación R cuadrado: *0.9041124*.

Resultado 11:
Los valores de la intercepción y la pendiente obtenidos de la regresión lineal son *-2.77785 y 0.09772* respectivamente.

Resultado 12: Una vez obtenidos estos valores debemos comprobar si son nulos o no. La hipótesis nula que debemos plantear para esta prueba es *H0: M= B = 0* mientras que la hipótesis alternativa es H1: B y M != 0. Ahora para contradecir esta hipótesis nula y asegurar que los valores de M y B son distintos de 0, podemos basarnos en el siguiente hecho: si para la fórmula y = M*x + B, y es igual para todo x se dice que M vale 0, más si dan valores distintos M no puede ser nulo. Por lo tanto, al realizar la prueba y elegir dos valores distintos obtenemos como resultado lo siguiente:

#con x igual a 302.4766
(y = M*302.4766 + B) = 26.78016
#con x igual a 430.0008
(y = M*430.0008 + B) = 39.24183

#x igual a 0
(y0 <- M*0 + B) = -2.77785

Resultado 13:
Cálculo del número de simulaciones que se debe realizar para el experimento de Montecarlo:
(N <- 0.25*(qnorm (0.95) /0.05) ^2) = 270.5543

Resultado 14:
Valor esperado de las ventas es 1 624 032 dólares.

Resultado 15:
Error estándar de las ventas es 24.62652.

Resultado 16:
Fracción de ventas provenientes de canciones que exceden los 6 minutos: 0.2717685.

Resultado 17:
Fracción esperada de canciones que exceden los 6 minutos: 0.1568977.

Resultado 18:
Fracción esperada de ganancias después del cobro de la comisión del 30% o 15%: 1 380 427 dólares.

IV. CONCLUSIONES

Según el resultado 1 mostrado en la figura 1, las duraciones de las canciones parecen seguir una distribución normal, debido a que se puede suponer una distribución normal si la gráfica tiene aproximadamente una forma de campana y es simétrica con respecto a la media. En este caso, por lo tanto, parece ser una distribución normal, más solo es una suposición basada en la teoría anterior.

De acuerdo al resultado 2, en ambas pruebas de normalidad realizadas, una prueba con el uso de la biblioteca nortest, y la segunda prueba realizada con la prueba de normalidad por defecto que se puede ejecutar en R, se obtiene un p-value muy pequeño, en ambos casos se obtiene un p-value de 2.2e-16, es decir p-value < 0.05, por lo tanto, se rechaza la hipótesis H0. De esta forma podemos afirmar que la muestra NO proviene de una distribución normal, aceptando la hipótesis alternativa.

Analizando el Q-Q plot, es decir, el resultado 3 (figura 2), tenemos que tener claro que diremos se puede asumir una distribución normal si los puntos caen aproximadamente a lo largo de esta línea de referencia. Sin embargo, como se observa en la figura 3, es claro que no podremos asumir tal normalidad debido a que existen muchos puntos que no caen aproximadamente a lo largo de la línea de referencia, podemos notar entre los "norm" cuartiles 1 y 3 una gran cantidad de puntos que desobedecen el seguir la línea de referencia y más bien caen considerablemente lejos de la misma.

Para mostrar los valores extremos atípicos de la distribución de las duraciones de las canciones es necesario recurrir al criterio IQR significa que todas las observaciones por encima de q0.75 +1.5*IQR o por debajo de q0.25-1.5*IQR (donde q0.25 y q0.75 corresponden al primer y tercer cuartil respectivamente, y IQR es la diferencia entre el tercer y primer cuartil) son consideradas como valores atípicos potenciales por R. En otras palabras, todas las observaciones fuera del siguiente intervalo se considerarán como valores atípicos potenciales: I = [q0.25-1.5*IQR; q0.75+1.5*IQR]. De esta forma, realizando un diagrama de cajas notamos como existen bastantes puntos los cuales consideramos valores extremos atípicos. Por lo tanto, podemos suponer que sí existen valores atípicos que deben ser eliminados al observar la figura 3, debido a que existen puntos

que se encuentran fuera del rango I, es decir valores superiores al máximo del rango de IQR.

De acuerdo al resultado 5 se concluye una vez eliminados los valores extremos atípicos que se obtiene un valor de p-value mucho mayor, sin embargo, todavía continúa siendo menor que 0.05 (p-value < 0.05), es decir, no es una distribución normal.

Según el resultado 6 se concluye de una asimetría positiva lo siguiente: existe una distorsión o asimetría que se desvía de la curva de campana simétrica, o distribución normal, en un conjunto de datos, gracias a este hecho volvemos a comprobar que la distribución de los datos no es normal. Asimismo, la asimetría positiva señala que se deben de modelar los datos mediante una distribución gamma.

Analizando el diagrama de dispersión, es decir la figura 4, de izquierda a derecha podemos notar ciertos detalles, que no siempre se cumplen, pero en algunos puntos sí se pueden considerar como ciertos, lo cual es que entre más dure la canción en segundos más tiempo se tarda procesando la misma. Hay que decir que se pueden observar ciertas canciones que duran menos que otras y aun así tardan más tiempo procesándose, por lo que son excepciones al primer punto mencionado. Sin embargo, como se puede notar, la canción que dura más tiempo es la que tarda más tiempo en procesarse.

De acuerdo al resultado 10 se pueden tomar las siguientes conclusiones: la duración de la canción es un buen predictor de la duración del procesamiento de la canción esto debido a que suponemos que entre más dure la canción más tiempo tomará procesarla, una vez en primera instancia al realizar el diagrama de dispersión notamos como la suposición parecía ser verdadera, donde en el gráfico la relación parecía ser positiva, sin embargo, para comprobarlo entonces calculamos el coeficiente de determinación R cuadrado, obteniendo como resultado 0.9041124, esto quiere decir que es un modelo cuyas estimaciones se ajustan muy bien a la variable real. Aunque técnicamente no sea del todo correcto se puede decir que el modelo explica en un 90.4% a la variable real. A la hora de interpretar el coeficiente de determinación R cuadrado de forma más técnica, se puede decir que cuando este arrojaba como resultado -1, se indicaba una fuerte correlación negativa, que significaba que cada vez que la x incrementaba la y decrecía. Por otra parte, si era 0, significaba la no existencia de una asociación entre dos variables "x" y "y". Pero por último y la que más nos interesa es el caso donde 1 indica una fuerte correlación positiva lo cual significa que “y” incrementa con “x”, es decir, en el caso actual, entre más duren las canciones más será el tiempo que se tarde procesándolas.

Como se pueden observar los cálculos en R en el resultado 12, se tomaron distintas *x* mas las *y* dieron distinto por lo tanto se descarta la parte de la hipótesis nula que menciona que la *M* es igual a 0, debido a que la única forma de que sea nula es cuando la *y* da el mismo resultado cuando se toman distintos valores de *x*. Por otra parte, para el caso de la *B*, se dice que *B* no es igual a 0 si *y (x = 0)* no es cero, debido a que *b = y (0)*, y como se puede observar en el resultado 12, *y (x = 0)* es igual a *-2.77785*, por lo tanto, se niega la posibilidad de que la *B* sea nula. Por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alternativa como verdadera, es decir *M* y *B* no son iguales a 0.

Para el experimento de Montecarlo, después de utilizar la fórmula propuesta en la metodología para el cálculo del número de simulaciones que se deben realizar se concluye del resultado que se deben realizar un número de 270 simulaciones.

Una vez se ejecuta el código escrito en R: Experimento de Montecarlo, obtenemos las siguientes conclusiones:
Existe una probabilidad del 100% de que la computadora trabajando sin descanso y con una demanda continua e inagotable de solicitudes, genere ventas por un millón de dólares en un año en cada simulación realizada de las N que se llevaron a cabo. Asimismo, existe una fracción considerable de ingresos generados gracias a canciones que exceden los 6 minutos, sin embargo, el número de canciones que no exceden los 6 minutos es mucho mayor que el número de canciones que sí exceden los 360 segundos, por lo tanto, se termina generando más ingresos por canciones que no exceden los 6 minutos que por las que sí lo hacen. Por otra parte, las empresas que cobran comisiones logran llevarse una gran cantidad de dinero, debido a que cada venta de cada simulación supera el millón de dólares. De esta forma las empresas ganarán un 30% de comisión y no un 15% en caso de que no superen el millón de dólares. Una deducción que se puede mencionar sobre las comisiones, es que en realidad es más rentable generar ingresos cercanos al millón de dólares, pero sin superarlo, debido a que las empresas solo podrán cobrar un 15% de comisión, sin embargo, si superan el millón de dólares por una cifra mínima las empresas podrán darse el derecho de cobrar el 30% en comisiones.

APÉNDICES

CÓDIGOS EN R:

```
library(readr)
library(nortest)
library(car)
library(moments)
library(fitdistrplus)
library(plotrix)

#I Parte Distribución Duracion de Canciones
outputData <- read_csv("C:/Users/Esteban
Quesada/Desktop/ProyectoMetodos/outputData.csv")
#(duration <- outputData$duration)

hist(duration,
    main = "Histograma de duraciones",
    xlab="Duración",
    ylab="Frecuencia",
    col = "blue",
    breaks = 300,
    border = "black")
lines(density(duration))

ad.test(duration)
shapiro.test(duration)

qqPlot(duration)

boxplot(duration,range = 1.5)

Q <- quantile(duration, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(duration)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
qqPlot(iqr)
eliminated<- subset(outputData, duration > (Q[1] - 1.5*iqr)
& duration < (Q[2]+1.5*iqr))

shapiro.test(eliminated$duration)

skewness(eliminated$duration)
(fit.gamma <- fitdist(eliminated$duration, distr = "gamma",
method = "mle"))
fit.gamma
summary(fit.gamma)
plot(fit.gamma)

descdist(eliminated$duration, boot=1000)
```

```
#Experimento de Montecarlo
#Variables del modelo lineal de la parte 2
M<- 0.09772 #pendiente
B<- -2.77785 #intercepcion
(N <- 0.25*(qnorm(0.95)/0.05)^2)

#Locales
ventasN <- rep(0,N)

#Calcular los precios de las canciones:
tiempoDeProcesamientoTotal <- 0
cancionesExceden6 <- 0
totalCanciones <- 0
dineroExcedente6 <- 0
dineroTotalGenerado <- 0

i = 1
while(i < N){
  contadorprimero = i
  ventasPorCadaN = 0
  tiempoDeProcesamientoTotal = 0
  while(tiempoDeProcesamientoTotal < 31536000){
    totalCanciones <- totalCanciones + 1
    (duracionCancion <- rgamma(1, shape = 5.86283427,
rate = 0.02288505))
    (tiempoDeProcesamiento = M*duracionCancion+B  )
    tiempoDeProcesamientoTotal <-
tiempoDeProcesamientoTotal + tiempoDeProcesamiento
    precioCancion <- 0
    if(duracionCancion < 360){
      precioCancion <- 0.99
      ventasPorCadaN = ventasPorCadaN + 0.99
    }else{
      cancionesExceden6 <- cancionesExceden6 + 1
      duracionParcial <- duracionCancion
      precioTotalAcumulado <- 0
      while(duracionParcial > 0){
        duracionParcial <- (duracionParcial-360)
        precioTotalAcumulado <- (precioTotalAcumulado +
0.99)
      }
      precioCancion<- precioTotalAcumulado
      ventasPorCadaN = ventasPorCadaN + precioCancion
      dineroExcedente6 <- dineroExcedente6 +
precioCancion
    }

    dineroTotalGenerado <- dineroTotalGenerado +
precioCancion
  }
  ventasN[i] = ventasPorCadaN
  i <- i+1

}
(ventasN)

#Valor esperado de las ventas:
mean(ventasN)
#Error estandar de las ventas:
std.error(ventasN)
(sd(ventasN)/sqrt(N))

#fraccion de ventas provenientes de ventas de canciones
que exceden los 6 minutos
(fraccionVentasEsperada <-
dineroExcedente6/dineroTotalGenerado)
#Fraccion esperada de canciones que exceden los 6
minutos
(fraccionCancionesEsperada <-
cancionesExceden6/totalCanciones)
```

```
#Fracción esperada de ganancias despues de cobro de
comision del 30%
ganancias <- rep(0,N)
for(i in 1:N){
  if(ganancias[i] > 1000000){
    ganancias[i] <- ventasN[i]- (ventasN[i]*0.30)
  }
  if(ganancias[i] < 1000000){
    ganancias[i] <- ventasN[i]- (ventasN[i]*0.15)
  }

}
mean(ganancias)
```

```
II Parte Procesamiento de Canciones Modelo Lineal
proccesTime <- read_csv("C:/Users/Esteban
Quesada/Desktop/ProyectoMetodos/data2.csv")
plot(x = proccesTime$Seconds,
     y = proccesTime$`Processing (s)`,
     xlab="Duracion de cancion en segundos",
     ylab="Segundos de Procesamiento")
abline(lsfit(proccesTime$Seconds,proccesTime$`Processin
g (s)`))
lmProcessTime = lm(`Processing (s)`~Seconds, data =
proccesTime) #Create the linear regression
summary(lmProcessTime) #Review the results
cor(proccesTime$`Processing (s)`,proccesTime$Seconds)
(R2 = cor(proccesTime$`Processing
(s)`,proccesTime$Seconds)^2)
summary(lmProcessTime) #Review the results
M<- 0.09772 #pendiente
B<- -2.77785 #intercepcion
(y = M*302.4766 + B )
(y = M*430.0008 + B )
(y0 <- M*0 + B)
```

REFERENCES

[1] Baron, M. (2019a). *Probability and Statistics for Computer Scientists* (3rd ed.). CRC Press.

[2] *plotrix Package in R | Tutorial & Programming Examples*. (2020, 30 septiembre). Statistics Globe. https://statisticsglobe.com/plotrix-r-package

[3] Delignette-Muller, M. L. (2015, 20 marzo). *fitdistrplus: An R Package for Fitting Distributions | Delignette-Muller | Journal of Statistical Software*. Journal of Statistical Software. https://www.jstatsoft.org/article/view/v064i04

[4] Chen, J. (2021, 30 abril). *Learn About Skewness*. Investopedia. https://www.investopedia.com/terms/s/skewness.asp

[5] Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. http://yihui.name/knitr/.

[6] Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B. y Lamere, P. (2011). The million song dataset. En Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR).