Escuela de Ingeniería en computación

Programa Maestría en Computación

MC6104 - Diseño de Experimentos

Apuntes clase 02/03/2023

Aarón Sibaja Villalobos
2018319668

San José, Costa Rica

marzo, 2023

Índice de contenidos

1. Inferencia Estadística

Utiliza pruebas estadísticas para genera una conclusión. Se utiliza ampliamente en los campos de la medicina.

Es importante resaltar que para asegurar que las muestras son representativas a la población, esta debe de ser selecionada de manera completamente aleatoria.

Ejemplo: Inferir el tiempo de espera de una persona realizando encuestas a 10 unidades de emergencia.

2. Pruebas de Hipótesis

2.1. Hipótesis Nula

Se supone, en principio, lo contrario de lo que se desea probar, hasta que los datos y pruebas obtenidas demuestran que el punto de partida era falso, por lo tanto se rechaza.

Ejemplo: Dos grupos no son iguales, no hay correlación entre dos variables. Etc

2.2. Hipótesis Alternativa

Contraria a la hipótesis nula, los grupos son iguales y hay una correlación entre dos variables.

2.3. P-value

Asumiendo que la hipótesis nula es verdadera, es la probabilidad de obtener un resultado igual o más extremo a lo obtenido por los datos.

2.4. Visualización del P-value

Problema: se va tirar una moneda 100 veces

Hipótesis Nula:

La moneda es justa, existe la misma probabilidad de caer en escudo o corona, o cara o cruz

Hipótesis Alternativa:

La moneda no es justa.

El resultado que tenemos es que la moneda cayó en escudo 95 de las veces.

$$95\,\% \textbf{ corresponde a un } \alpha = 0,05 \textbf{ entonces, p-value} = 0,05$$

3. Errores de Inferencia

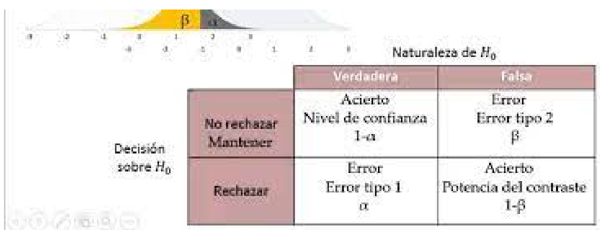
Puede darse el caso de que los resultado que se obtuvieron, aunque improbables, sean el resultado de la casualidad.

Error Tipo I:

El error de tipo I se comete cuando la hipótesis nula es verdadera y, pera esta es rechazada. La probabilidad de cometer el error de tipo I es el nivel de significación de α.

Error Tipo II:

El error de tipo II se comete cuando la hipótesis nula es falsa y, pero esta se acepta.



4. T-test

También conocida como prueba t de Student, corresponde a una herramienta para evaluar las medias de uno o dos grupos mediante yu cómo estan relacionados mediante pruebas de hipótesis.

Ejemplo: Se crea un algoritmo nuevo que permite mejorar los tiempos de entrenamiento para una red neuronal y se quiere comparar contra la mejor solución conocida hasta el momento:

En este caso, al ”Método de Entrenamiento” se le llama factor y los niveles de nuestro factor son ”Algoritmo A(viejo)” y ”Algoritmo B(Nuevo)”.

Cuadro 1: Componentes del T-test

	Factor
	Método de Entrenamiento
Niveles	Algoritmo 1
	Algoritmo 2

La tabla 1 puede contener más factores y niveles, si embargo se requiere otro tipo de prueba.

Para aplicar el T-test se deben de cumplir algunos requerimientos:

- Los datos de ambas poblaciones deben de estar normalmente distribuidas.
- Ambas poblaciones tienen la misma varianza.
- En R, se puede utilizar la Welch's T-test cuando no tengamos la misma varianza

Student T-tes

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p\sqrt{\frac{2}{n}}} \tag{1}$$

es matemáticamente igual a un Anova mono factorial con dos niveles

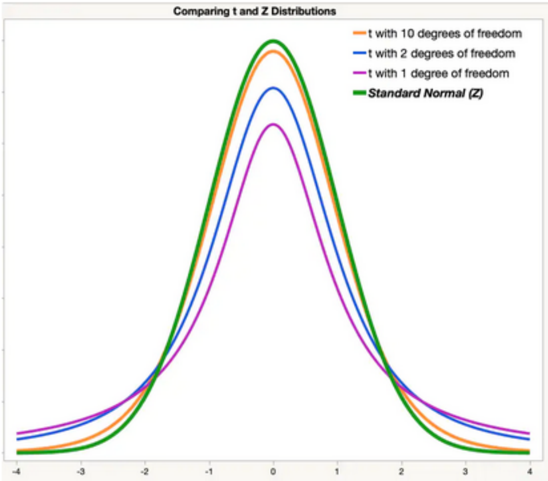
$$s_p\sqrt{\frac{s_{x1}^2 - s_{x2}^2}{2}} \tag{2}$$

- La Diferencia de promedios se calcula en el numerador. Entre más grande la diferencia más grande el numerador.
- El denominador es el error estándar de la diferencia de promedios. Se vuelve más pequeño si el tamaño de la muestra aumenta. También se le conoce como desviación estándar combinada.
- El valor de t se hace grande si la diferencia entre promedios es grande, si las varianzas disminuye, o si el tamaño de la muestra aumenta.
- Se calcula la probabilidad de observar t bajo la hipótesis nula utilizando la distribución t

4.1. Distribución T

Corresponde a la distribución de la ubicación del promedio de la muestra relativa al promedio verdadero dividido entra le desviación estandar de la muestra multiplicado por $\sqrt[2]{n}$

Se puede utilizar para construir un intervalo de confianza sobre el promedio verdadero y veridicar la diferencia entre el promedio de dos poblaciones.



- Cuando hacemos un t-test verificamos si la prueba estadística es un valor más extremo que el esperado de la distribución t
- Rechazamos la hipótesis nula si estamos fuera del valor
- Los grados de libertad están asociados al tamaño de la muestra.
- Para tamaños grandes podemos utilizar la distribución Z

Distribución t de Student::

Es una distribución de probabilidad que surge del problema de estimar la media de una población normalmente distribuida cuando el tamaño de la muestra es pequeña y la desviación estándar poblacional es desconocida. La desviación de la población se estima a partir de la muestra.

4.2. Ejemplo R T-test

Es necesario instalar las librerías necesarias

```r
install.packages("mnormt")
if(!require(psych)){install.packages("psych", dependencies = TRUE)}
if(!require(FSA)){install.packages("FSA")}
if(!require(lattice)){install.packages("lattice")}
if(!require(lsr)){install.packages("lsr")}
if(!require(lsr)){install.packages("rcompanion")}
```

Se deben de cargar los datos en formato texto y para luego parsear el texto en un objeto DataFrame

```r

Datos = ("
```

```
           Algoritmo      Ejecucion  Tiempo
'Algorimo A'    '1'        12070
'Algorimo A'    '2'        14040
'Algorimo A'    '3'        13580
'Algorimo A'    '4'         9540
'Algorimo A'    '5'        14070
'Algorimo A'    '6'        11520
'Algorimo A'    '7'        13030
'Algorimo A'    '8'        13245
'Algorimo A'    '9'        14215
'Algorimo A'    '10'       15070
'Algorimo A'    '11'       12580
'Algorimo A'    '12'       11540
'Algorimo A'    '13'        9580
'Algorimo A'    '14'       11510
'Algorimo A'    '15'       16070
'Algorimo A'    '16'       13010
'Algorimo A'    '17'       10530
'Algorimo A'    '18'       13030
'Algorimo A'    '19'       17080
'Algorimo A'    '20'       13020
'Algorimo B'    '1'        11070
'Algorimo B'    '2'        12010
'Algorimo B'    '3'        12550
'Algorimo B'    '4'        10500
'Algorimo B'    '5'        12000
'Algorimo B'    '6'        12520
'Algorimo B'    '7'        13520
'Algorimo B'    '8'        13540
'Algorimo B'    '9'        13255
'Algorimo B'    '10'       15235
'Algorimo B'    '11'       12235
'Algorimo B'    '12'       11285
'Algorimo B'    '13'       10040
'Algorimo B'    '14'       11295
'Algorimo B'    '15'       14080
'Algorimo B'    '16'       12080
'Algorimo B'    '17'       11580
'Algorimo B'    '18'       14070
'Algorimo B'    '19'       15050
'Algorimo B'    '20'       12050
           ")

Data = read.table(textConnection(Datos), header = T)

rm(Datos)
```

Seguidamente se orden los datos

```
library(psych)
headTail(Data)
```

con el siguiente Output:

```
     Algoritmo Ejecucion Tiempo
1   Algorimo A         1  12070
2   Algorimo A         2  14040
3   Algorimo A         3  13580
4   Algorimo A         4   9540
...       <NA>       ...    ...
37  Algorimo B        17  11580
38  Algorimo B        18  14070
39  Algorimo B        19  15050
40  Algorimo B        20  12050
```

Se debe de realizar un resumen de los datos que permita analizar la metadata y los tipos de datos. La siguiente función muestra y examina algunas medidas de tendencia central

```r
# Resumen Datos
str(Data)
summary(Data)
```

Esto tiene el siguiente output:

```
> str(Data)
'data.frame':	40 obs. of  3 variables:
 $ Algoritmo: chr  "Algorimo A" "Algorimo A" "Algorimo A" "Algorimo A" ...
 $ Ejecucion: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Tiempo   : int  12070 14040 13580 9540 14070 11520 13030 13245 14215 15070 ...
> summary(Data)
  Algoritmo           Ejecucion         Tiempo     
 Length:40          Min.   : 1.00   Min.   : 9540  
 Class :character   1st Qu.: 5.75   1st Qu.:11535  
 Mode  :character   Median :10.50   Median :12565  
                    Mean   :10.50   Mean   :12707  
                    3rd Qu.:15.25   3rd Qu.:13695  
                    Max.   :20.00   Max.   :17080  
```

En la imagen **??** hay dos errores. El primero es que no tiene mucho sentido sacar medidas de posición central para una variable discreta cómo Ejecución. Por otra parte,no esta considerando que hay dos clases de algoritmos y lo está tomando cómo un todo.

Ahora resumiendo los datos por grupo se tiene los siguiente:

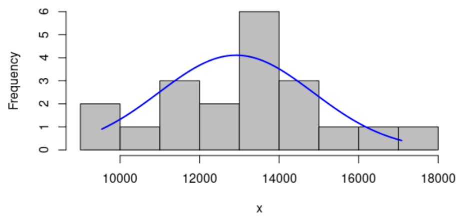
```r
library(FSA)
Summarize(Tiempo ~ Algoritmo, data=Data, digits=4)
```

Esto nos da como resultado un resumen de las medidas de posición central para cada Algoritmo A y B

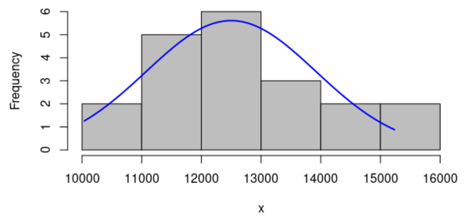
```
   Algoritmo  n     mean       sd   min       Q1  median      Q3   max
1 Algorimo A 20 12916.50 1942.198  9540 11535.00 13025.0 14047.5 17080
2 Algorimo B 20 12498.25 1422.549 10040 11508.75 12157.5 13525.0 15235
```

Luego se separa los datos en los grupos A y B, y seguidamente se grafica el histograma para cada uno.

```r

A = Data$Tiempo[Data$Algoritmo == 'Algorimo A']
B = Data$Tiempo[Data$Algoritmo == 'Algorimo B']

library(rcompanion)

plotNormalHistogram(A)
plotNormalHistogram(B)
```



(a) Algoritmo A



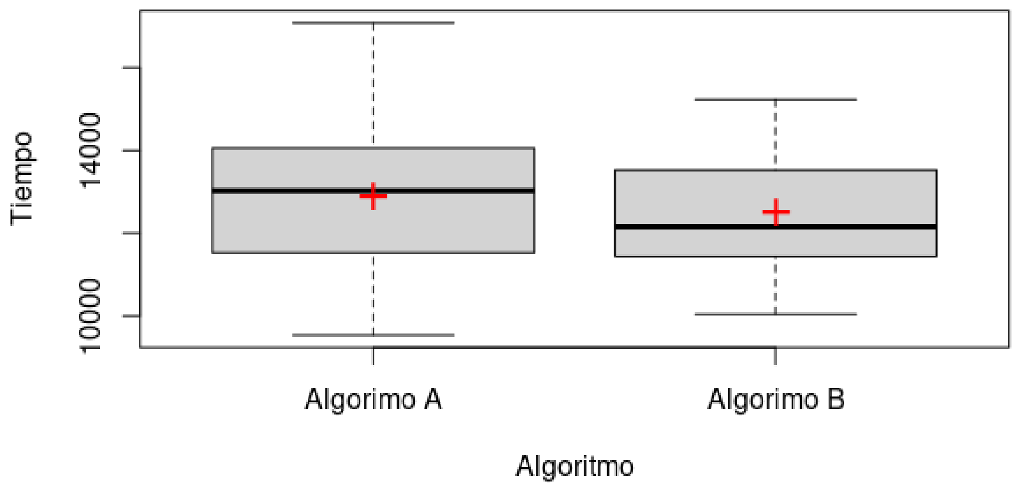
(b) Algoritmo B

Figura 1: Histogramas de los dos tipos de Algoritmos

Cómo no tenemos la información completa, es necesario realizar diagramas de caja para cada grupo (Algoritmo A y B)

```
M = tapply(Data$Tiempo, Data$Algoritmo, mean)

boxplot(Tiempo ~ Algoritmo, data=Data)

points(M, col="red", pch="+", cex=2)
```

Esto nos da cómo resultado lo siguiente:



En la imagen **??**, la cruz roja corresponde a la media, la linea negra corresponde a la mediana o también el segundo cuartil. Además, la primera linea horizontal de abajo hacia arriba corresponde al primer cuartil y la segunda corresponde al tercero. El primer cuartil denotar que el 25 % de los datos por debajo de ese valor. Los vigotes se puden utilizar como el rango o valor máximo y mínimos de los datos.

Ahora aplicando la prueba T en R, se tiene que:

```
t.test(Tiempo ~ Algoritmo, data= Data)
```

Esto nos da cómo resultado lo siguiente:

```
	Welch Two Sample t-test

data:  Tiempo by Algoritmo
t = 0.77695, df = 34.83, p-value = 0.4424
alternative hypothesis: true difference in means between group Algorimo A and group Algorimo B is not equ
al to 0
95 percent confidence interval:
 -674.7892 1511.2892
sample estimates:
mean in group Algorimo A mean in group Algorimo B 
                12916.50                 12498.25 
```

De la imagen anterior podemos determinar que no podemos rechazar H_0, la hipótesis nula la definimos como que ambos grupos no son distintos. Ya que el $p-value = 0,4424$, no rechazamos la hipótesis nula, para rechazarla se ocupa un valor de $p - value = 0,044$

4.3. Conclusiones

El promedio de tiempo de entrenamiento de nuestro algoritmo (nuevo) no es suficiente ni menor con respecto al otro algoritmo(viejo) comparado

- Nuestro resultado no es estadísticamente significativo o menor al otro.
- Este análisis es fácilmente replicable en un sin fin de experimentos donde se comparen dos sistemas, dos poblaciones, etc.

5. Distribución Normal

- Generalmente es requerida para las pruebas paramétricas
- Para la prueba se requiere que los residuos sean normales

Residuos:

Diferencia entra las observaciones y el valor predicho por el modelo.

La t-test es robusta a violaciones de normalidad para distribuciones simétricas (se pueden tolerar pequeñas violaciones). Por otro lado, las violaciones evidentemente grandes invalidan la prueba.

5.1. Pruebas de normalidad

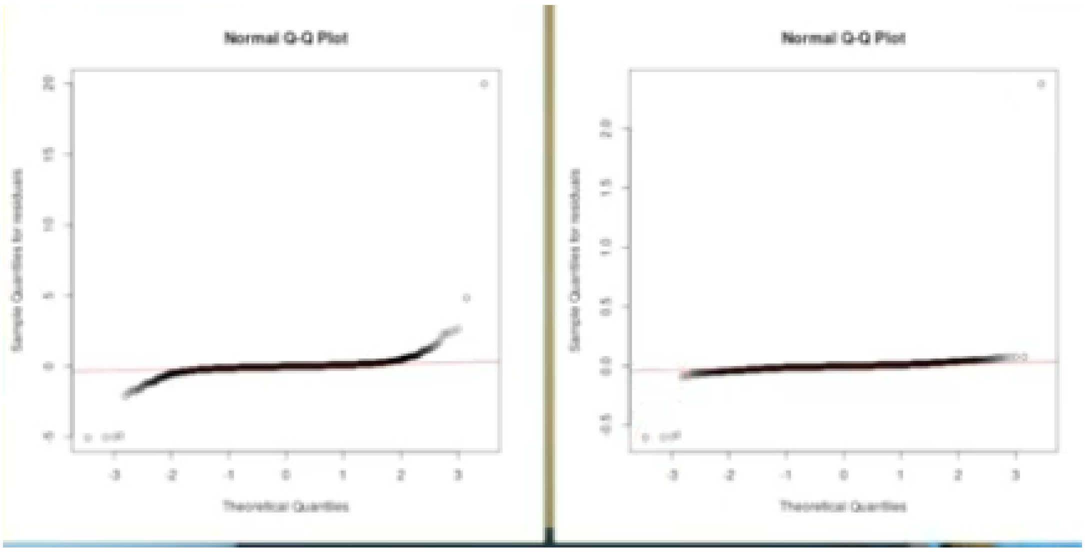
- Shapire - Wilk
- Anderson - Darling
- Kolmologov - Smirnov
- D'Agustino - Pearson

Lamentablemente las pruebas anteriores dependen de un tamaño de muestra grande. Por eso, con una muestra más pequeña una evaluación visual es lo suficientemente bueno cómo para asumir normalidad para el conjunto de datos.

6. Gráficos Q-Q

Corresponde a un diagrama de puntos que grafica las funciones de distribución acumuladas.En caso de que provengan de una misma distribución los puntos aparecen alineados en el gráfico. Es utilizada para determinar si la muestra sigue una distribución teórica, esta puede ser una distribución normal por ejemplo.

Este tipo de gráficos permite determinar outliers, es más fácil de evaluar visualmente que un histograma.



7. Anova monofactorial

Se utiliza cuando se quieren comparar mas de dos grupos. Por ejemplo utilizando la variable dependiente tiempo queremos comparar los tres algoritmos A, B y C.

7.1. Supuestos

1. Los residuos son normalmente distribuidos, aunque se puede tolerar una desviación moderada de la normalidad
2. Se requiere que los grupos tengan homocedasticidad, la misma varianza. El valor de una observación no se ve afectado por alguna otra observación.
3. Las observaciones entre grupos son independientes.

7.2. Ejemplo R Anova

Se deben de instalar las siguientes librerias en el R-base

```r
if(!require(psych)){install.packages("psych", dependencies = TRUE)}
if(!require(FSA)){install.packages("FSA")}
if(!require(Rmisc)){install.packages("Rmisc")}
if(!require(ggplot)){install.packages("ggplot")}
if(!require(car)){install.packages("car", dependencies = TRUE)}
if(!require(multcompView)){install.packages("multcompView", dependencies = TRUE)}
if(!require(multcompView)){install.packages("multcomp", dependencies = TRUE)}
if(!require(ismeans)){install.packages("ismeans", dependencies = TRUE)}
if(!require(rcompanion)){install.packages("rcompanion", dependencies = TRUE)}
```

Se carga la Data en formato texto, muy similar al ejemplo anterior. Se hace uso de la función *factor*, para modificar la columna Algoritmo como los factores a utilizar

```r
Datos = ("
Algoritmo       Ejecucion  Tiempo
'Algoritmo A'      '1'    12060
'Algoritmo A'      '2'    14089
'Algoritmo A'      '3'    13502
'Algoritmo A'      '4'     9574
'Algoritmo A'      '5'    14056
'Algoritmo A'      '6'    11569
'Algoritmo A'      '7'    13047
'Algoritmo A'      '8'    13275
'Algoritmo A'      '9'    14257
'Algoritmo A'      '10'   15075
'Algoritmo A'      '11'   12506
'Algoritmo A'      '12'   11557
'Algoritmo A'      '13'    9548
'Algoritmo A'      '14'   11514
'Algoritmo A'      '15'   16015
'Algoritmo A'      '16'   13004
'Algoritmo A'      '17'   10510
'Algoritmo A'      '18'   13040
'Algoritmo A'      '19'   17098
'Algoritmo A'      '20'   13080
'Algoritmo B'      '1'    11080
'Algoritmo B'      '2'    12089
'Algoritmo B'      '3'    12538
'Algoritmo B'      '4'    10571
'Algoritmo B'      '5'    12010
'Algoritmo B'      '6'    12598
'Algoritmo B'      '7'    13543
'Algoritmo B'      '8'    13547
'Algoritmo B'      '9'    13217
'Algoritmo B'      '10'   15297
'Algoritmo B'      '11'   12210
'Algoritmo B'      '12'   11299
'Algoritmo B'      '13'   10067
'Algoritmo B'      '14'   11279
'Algoritmo B'      '15'   14006
'Algoritmo B'      '16'   12099
'Algoritmo B'      '17'   11581
'Algoritmo B'      '18'   14012
'Algoritmo B'      '19'   15069
```

```
'Algoritmo B'      '20'   12000
'Algoritmo C'      '1'     9081
'Algoritmo C'      '2'    11012
'Algoritmo C'      '3'    11529
'Algoritmo C'      '4'     9569
'Algoritmo C'      '5'    11092
'Algoritmo C'      '6'    11524
'Algoritmo C'      '7'    12522
'Algoritmo C'      '8'    12588
'Algoritmo C'      '9'    12241
'Algoritmo C'      '10'   13257
'Algoritmo C'      '11'   11294
'Algoritmo C'      '12'   10226
'Algoritmo C'      '13'    9591
'Algoritmo C'      '14'    9224
'Algoritmo C'      '15'   12033
'Algoritmo C'      '16'   11063
'Algoritmo C'      '17'    9537
'Algoritmo C'      '18'   13014
'Algoritmo C'      '19'   14033
'Algoritmo C'      '20'   11093
         ")

# Load Data and RM String text
Data = read.table(textConnection(Datos), header = T)

rm(Datos)


Data$Algoritmo = factor(Data$Algoritmo, levels = unique(Data$Algoritmo))
```

Se hace un resumen sobre los tipos y nombres de las columnas. Además de un resumen de las medidas de posicion central por grupos.

```
library(psych)
headtail(Data)
str(Data)
summary(Data)
```

Es importante considerar hacer un resumen de las medidas de posición central para cada grupo o tipo de algoritmo

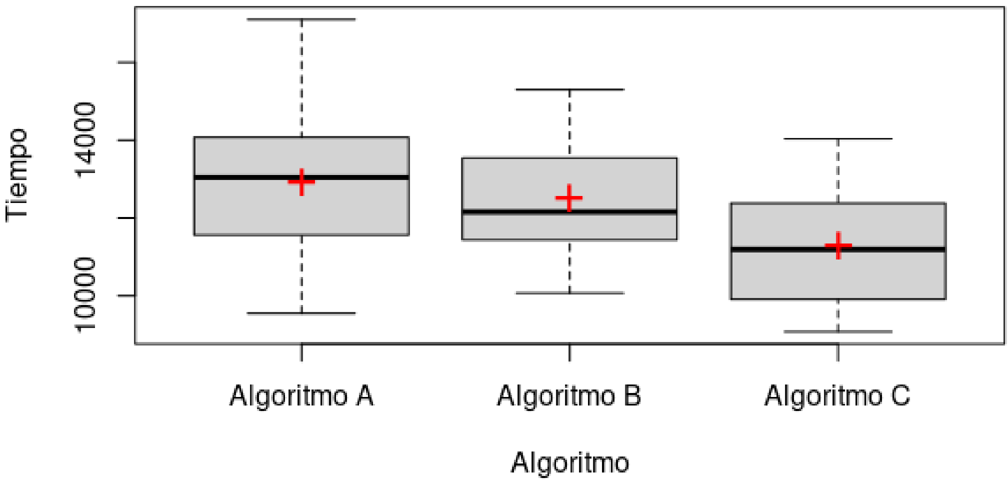
```
Summarize(Tiempo ~ Algoritmo, data = Data, digits = 4)
```

El código anterior tiene cómo output:

	Algoritmo	n	mean	sd	min	Q1	median	Q3	max
1	Algoritmo A	20	12918.80	1941.191	9548	11566.00	13043.5	14064.25	17098
2	Algoritmo B	20	12505.60	1414.667	10067	11510.50	12154.5	13544.00	15297
3	Algoritmo C	20	11276.15	1424.242	9081	10067.25	11193.5	12311.25	14033

El siguiente código permite visualizar un box plot para los tres grupos A, B y C

```
#Box Plots

M = tapply(Data$Tiempo, Data$Algoritmo, mean)

boxplot(Tiempo ~ Algoritmo, data=Data)

points(M, col="red", pch="+", cex=2)
```



El el grafico **??** se puede ver que la media del Algoritmo A, osea casi el 50 % de los datos está por encima de C. Además de que el tercer quartil, o sea el 75 % de los datos esta por debajo de el 50 % de A

7.3. Gráficos de Intervalos de Confianza

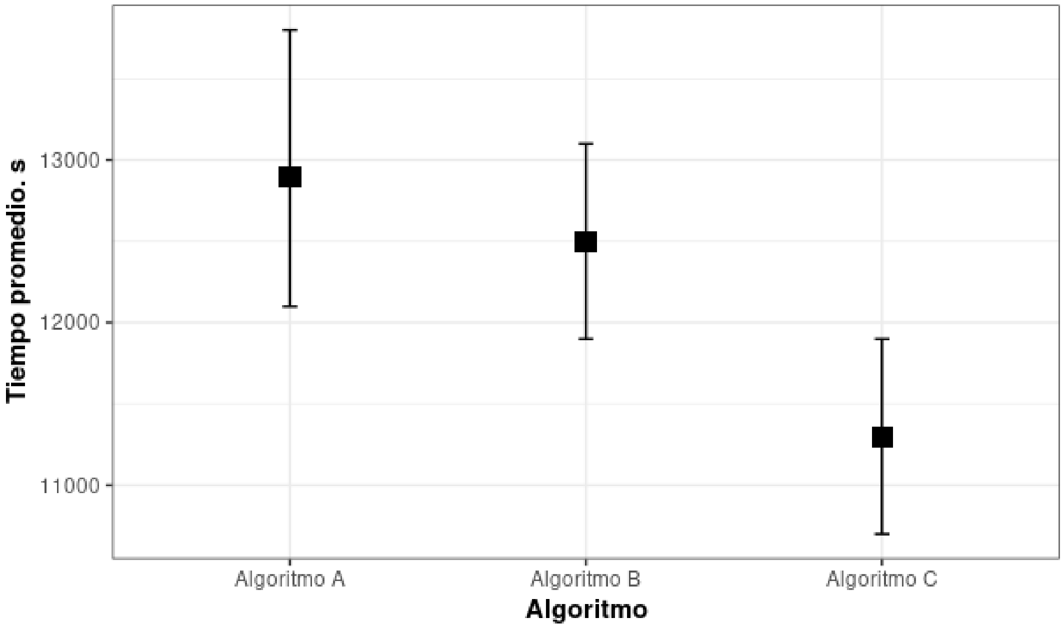
El el siguiente código se crea un intervalo de confianza al 95 %

```

library(rcompanion)

sum = groupwiseMean(
  Tiempo ~ Algoritmo,
  data=Data,
  conf=0.95,
  digit=3,
  traditional = FALSE,
  percentile = TRUE
)
```

```
    Algoritmo  n  Mean Conf.level Percentile.lower Percentile.upper
1 Algoritmo A 20 12900       0.95            12100            13800
2 Algoritmo B 20 12500       0.95            11900            13100
3 Algoritmo C 20 11300       0.95            10700            11900
```

Graficando los intervalos de confianza se tiene lo siguiente:

```

library(ggplot2)

ggplot(sum, aes(x= Algoritmo, y=Mean))+
geom_errorbar(aes(ymin=Percentile.lower, ymax=Percentile.upper), width=0.05,size=0.5) +
geom_point(shape=15, size=4)+
theme_bw()+
theme(axis.title = element_text(face = "bold")) +
ylab("Tiempo promedio. s")
```



Como se puede ver en la imagen **??**, el 95 % de los datos no se traslapan con A ni con B.

Cuando los gráficos no se traslapan podemos afirmar que no son estadísticamente diferentes. Sin embargo si se traslapan hay que ejecutar la prueba

Los intervalos de confianza presentan las siguientes características.

- Funcionan para indicar que tan precisa puee ser una estadística calcula
- Esta asociada al error de muestreo(diferencia entre el promedio de la población y la muestra).
- El intervalo nos dice que tan precisa puede ser nuestra inferencia
- El rango del intervalo de confianza expresa donde puede estar el promedio de la población.

Nivel de Confianza:

Si los intervalos de confianza de dos grupos no se traslapan pordemos asegurar que son estadísticamente diferentes. No hace falta realizar una prueba para demostrarlo.

Nivel de Confianza:

Es la probabilidad de que la estimación de la ubicación para un parámetro estadístico (por ejemplo un promedio) de una muestra sea igual al de la población.

7.4. Modelo Lineal

Modelo Lineal

Una variable dependiente es predictiva por un conjunto de variables independientes a través de una regresión lineal.

Todos los modelos lineales tiene supuestos sobre los datos.

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{3}$$

R-cuadrado

Es una medida que describe que tan bien el modelo explica los datos. Es una estadística de Goodness-fit

Ejemplo: un R-cuadrado de 0.6 indica que el 60 % de la variabilidad de la variable dependientes es explicada por el modelo.

P-value indica si hay una relación significativa descrita por el modelo. R-cuadrado mide el grado sobre el cúal los datos son explicados por el modelo.

```r
# Modelo
model = lm(Tiempo ~ Algoritmo, data = Data)
summary(model)
```

```
Call:
lm(formula = Tiempo ~ Algoritmo, data = Data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3370.8 -1211.6    25.1  1065.4  4179.2 

Coefficients:
                     Estimate Std. Error t value             Pr(>|t|)    
(Intercept)           12918.8      360.5  35.835 < 0.0000000000000002 ***
AlgoritmoAlgoritmo B   -413.2      509.8  -0.810              0.42105    
AlgoritmoAlgoritmo C  -1642.7      509.8  -3.222              0.00211 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1612 on 57 degrees of freedom
Multiple R-squared:  0.1647,	Adjusted R-squared:  0.1353 
F-statistic: 5.618 on 2 and 57 DF,  p-value: 0.005932
```

7.5. Ejecución del Anova

```r
library(car)
Anova(model, type= "II" ) #Suma de cuadrados
```

```
Anova Table (Type II tests)

Response: Tiempo
             Sum Sq Df F value   Pr(>F)
Algoritmo  29203870  2  5.6176 0.005932 **
Residuals 148161499 57
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Es posible graficar los histogramas de los residuos.

```
X = residuals(model)
library(rcompanion)
plotNormalHistogram(X)
plot(fitted(model),residuals(model))
plot(model)
```