

Proyecto 1: Galton Board

Esteban Villalobos-Gómez

2022-03-24

Proyecto 1: Galton Board

Instalación de bibliotecas

```
## Loading required package: psych
## Loading required package: FSA
## ## FSA v0.9.3. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
##
## Attaching package: 'FSA'
## The following object is masked from 'package:psych':
##
##     headtail
## Loading required package: Rmisc
## Loading required package: lattice
## Loading required package: plyr
##
## Attaching package: 'plyr'
## The following object is masked from 'package:FSA':
##
##     mapvalues
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
## Loading required package: car
## Loading required package: carData
## Registered S3 methods overwritten by 'car':
##   method      from
##   hist.boot    FSA
##   confint.boot FSA
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:FSA':
##
##      bootCase

## The following object is masked from 'package:psych':
##
##      logit

## Loading required package: multcompView
## Loading required package: lsmeans
## Loading required package: emmeans

## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.

## Loading required package: lsr
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##      recode

## The following objects are masked from 'package:plyr':
##
##      arrange, count, desc, failwith, id, mutate, rename, summarise,
##      summarize

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

## Loading required package: tidyr
## Loading required package: splitstackshape
```

Exploración de los Datos

Carga de datos

```
library(readr)
Data_raw <- read_csv("datos.csv",
                     col_types = list(col_integer(),col_character(),col_integer(),
                                      col_integer(),col_integer(),col_integer(),
                                      col_integer(),col_integer(),col_integer(),
                                      col_integer()),
                     show_col_types = FALSE)
```

Conversión de columna Distribución a factor.

```
Data_raw$Distribucion = factor(
  Data_raw$Distribucion,
  levels = unique(Data_raw$Distribucion))
```

Verificación

```
library(psych)
headTail(Data_raw)
```

```
##   Repeticion Distribucion  C1  C2  C3  C4  C5  C6  C7 Indefinido
## 1          1        Normal   4   7  14  13  15  10   7         <NA>
## 2          2        Normal   8   8  13  10  16   5  10         <NA>
## 3          3        Normal   7   8  11  10  19   6   9         <NA>
## 4          4        Normal   5   4  11  12  12  14  11           1
## 5          ...          <NA> ... ... ... ... ... ... ...         ...
## 6         67        Normal   4   8  15  11  15  11   6           1
## 7         68        Normal   6   9  11  13  15   7  10         <NA>
## 8         69        Normal   7   6  11  11  16   9  11         <NA>
## 9         70        Normal  11   9  10   9  13   9  10         <NA>
```

```
str(Data_raw)
```

```
## spec_tbl_df [70 x 10] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ Repeticion : int [1:70] 1 2 3 4 5 6 7 8 9 10 ...
## $ Distribucion: Factor w/ 2 levels "Normal","Uniforme": 1 1 1 1 1 1 1 1 1 1 ...
## $ C1          : int [1:70] 4 8 7 5 10 5 2 6 4 7 ...
## $ C2          : int [1:70] 7 8 8 4 3 5 8 6 6 6 ...
## $ C3          : int [1:70] 14 13 11 11 13 13 17 13 14 17 ...
## $ C4          : int [1:70] 13 10 10 12 10 13 9 18 16 12 ...
## $ C5          : int [1:70] 15 16 19 12 16 18 11 12 13 16 ...
## $ C6          : int [1:70] 10 5 6 14 8 11 13 9 8 6 ...
## $ C7          : int [1:70] 7 10 9 11 10 5 10 6 9 6 ...
## $ Indefinido  : int [1:70] NA NA NA 1 1 NA NA NA NA NA ...
## - attr(*, "spec")=
## .. cols(
## ..   Repeticion = col_integer(),
## ..   Distribucion = col_character(),
## ..   C1 = col_integer(),
## ..   C2 = col_integer(),
## ..   C3 = col_integer(),
## ..   C4 = col_integer(),
## ..   C5 = col_integer(),
## ..   C6 = col_integer(),
## ..   C7 = col_integer(),
## ..   Indefinido = col_integer()
## .. )
## - attr(*, "problems")=<externalptr>
```

Experimento 1: Distribución Normal

```
library(dplyr)
```

```
normal = Data_raw %>%
  filter(Distribucion == "Normal") %>%
  dplyr::select(C1:C7)

summary(normal)
```

```
##           C1           C2           C3           C4
## Min.      : 1.000   Min.      : 3.000   Min.      : 9.00   Min.      : 6.0
## 1st Qu.: 5.000   1st Qu.: 6.000   1st Qu.:11.00   1st Qu.:10.0
## Median : 6.000   Median : 8.000   Median :13.00   Median :11.0
## Mean      : 6.314   Mean      : 8.171   Mean      :12.86   Mean      :11.6
## 3rd Qu.: 8.000   3rd Qu.:10.000   3rd Qu.:14.00   3rd Qu.:13.0
## Max.      :11.000   Max.      :12.000   Max.      :17.00   Max.      :18.0
##           C5           C6           C7
## Min.      : 8.00   Min.      : 5.000   Min.      : 3.000
## 1st Qu.:11.50   1st Qu.: 7.000   1st Qu.: 7.000
## Median :13.00   Median : 9.000   Median : 9.000
## Mean      :13.43   Mean      : 8.686   Mean      : 8.971
## 3rd Qu.:15.50   3rd Qu.:10.000   3rd Qu.:10.000
## Max.      :19.00   Max.      :14.000   Max.      :14.000
```

Preprocesamiento

Cada fila se divide en múltiples filas, una por contenedor

```
library(tidyr)
d1 = gather(normal, Contenedor, "n", 1:7, factor_key=TRUE)
```

Visualización de los datos

1##### Histograma de promedios

```
Summarize(n ~ Contenedor,
  data = d1, digits = 4)
```

```
##   Contenedor  n    mean    sd min  Q1 median  Q3 max
## 1          C1 35  6.3143 2.3611   1  5.0      6  8.0 11
## 2          C2 35  8.1714 2.2555   3  6.0      8 10.0 12
## 3          C3 35 12.8571 2.2511   9 11.0     13 14.0 17
## 4          C4 35 11.6000 2.4758   6 10.0     11 13.0 18
## 5          C5 35 13.4286 2.5814   8 11.5     13 15.5 19
## 6          C6 35  8.6857 2.3735   5  7.0      9 10.0 14
## 7          C7 35  8.9714 2.7811   3  7.0      9 10.0 14
```

```
d1_summary = d1 %>%
  group_by(Contenedor) %>%
  summarise(avg = mean(n))
```

```
barplot(height=d1_summary$avg, names=d1_summary$Contenedor, main = "Medias por contenedor. Exp. 1", xlab = "Contenedor")
```

Medias por contenedor. Exp. 1

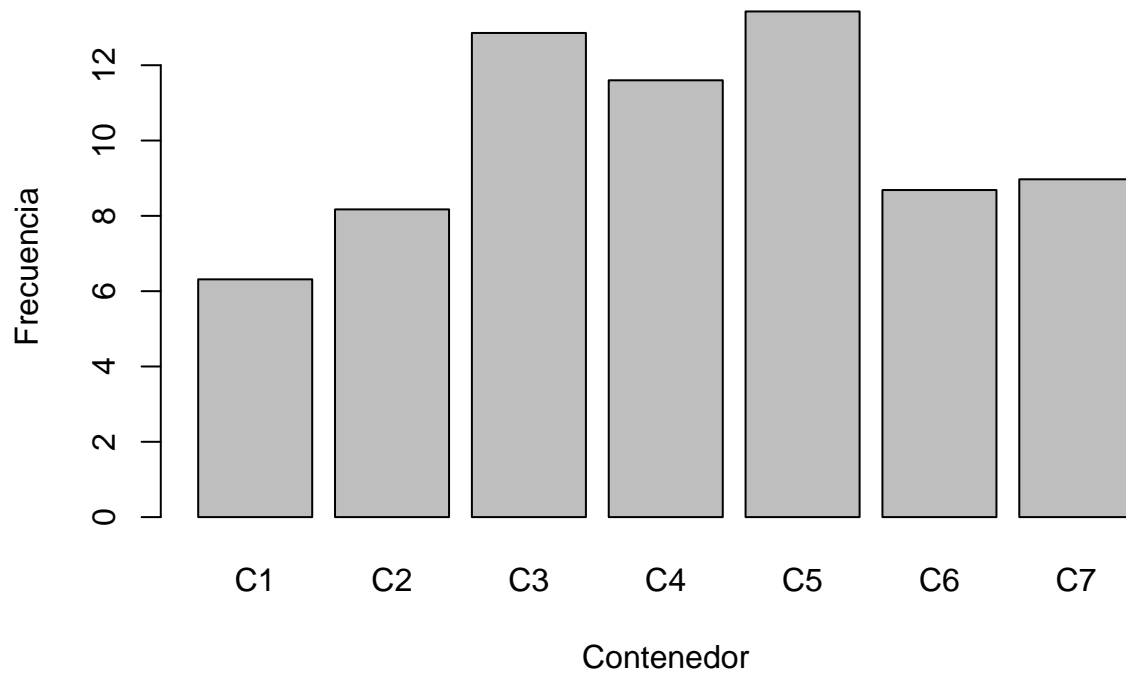


Diagrama de cajas

En este gráfico los bigotes representan el rango:

```
M1 = tapply(d1$n,  
            INDEX = d1$Contenedor,  
            FUN = mean)  
boxplot(n ~ Contenedor, ylab= "Frecuencia",  
        data = d1)  
points(M1, col="red", pch = "+",  
       cex = 2)
```

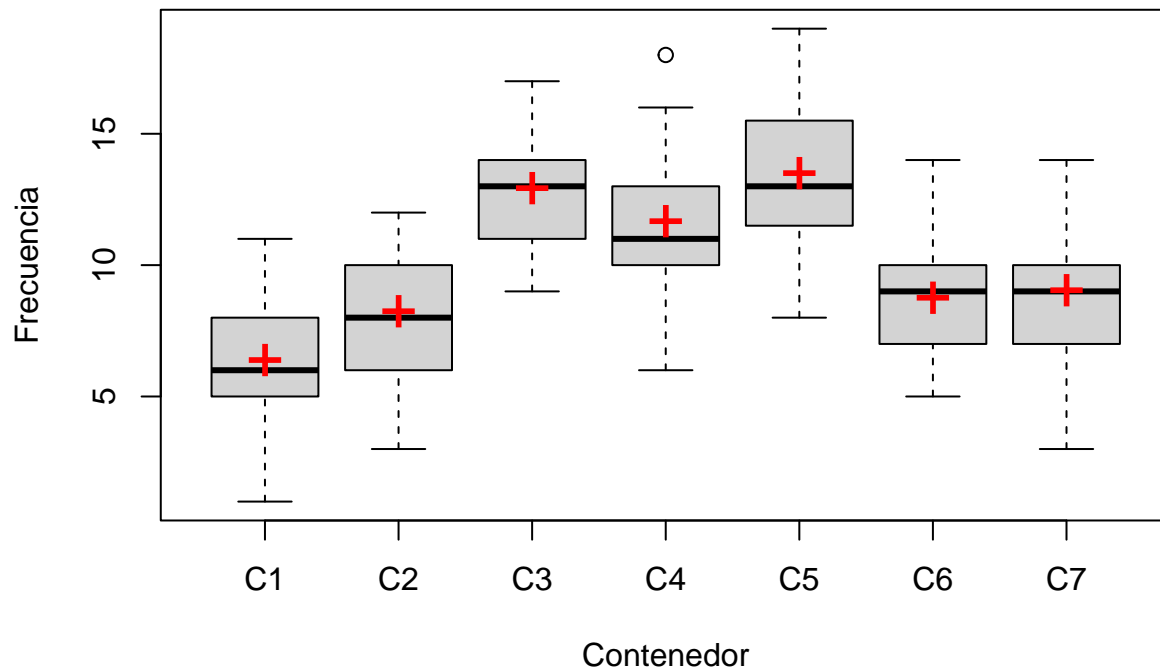


Gráfico de promedios e intervalos de confianza

Cálculo de los intervalos:

```
library(rcompanion)
```

```
##
```

```
## Attaching package: 'rcompanion'
```

```
## The following object is masked from 'package:psych':
```

```
##
```

```
## phi
```

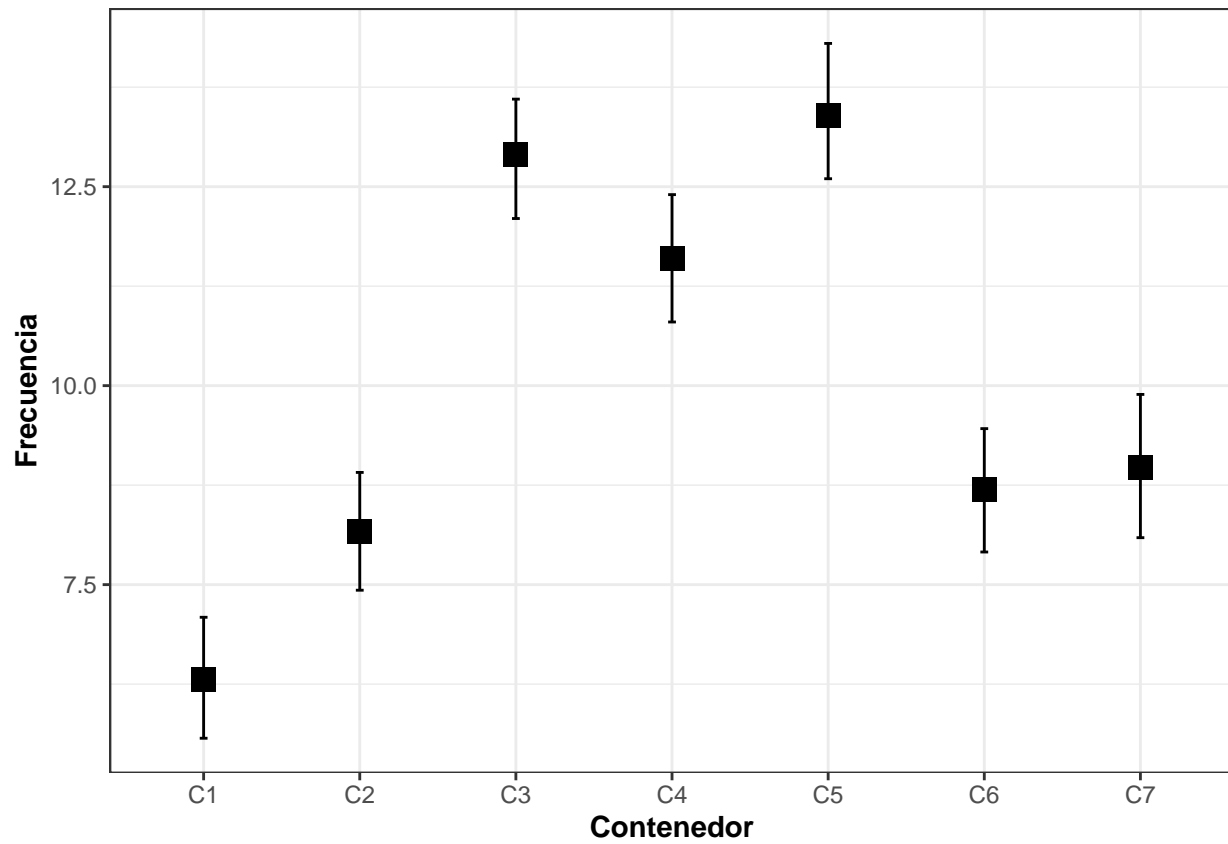
```
Sum1= groupwiseMean(n ~ Contenedor, data = d1, conf = 0.95, digits = 3, traditional = FALSE, percentile
Sum1
```

```
##   Contenedor  n  Mean Conf.level Percentile.lower Percentile.upper
## 1          C1 35  6.31      0.95           5.57           7.09
## 2          C2 35  8.17      0.95           7.43           8.91
## 3          C3 35 12.90      0.95          12.10          13.60
## 4          C4 35 11.60      0.95          10.80          12.40
## 5          C5 35 13.40      0.95          12.60          14.30
## 6          C6 35  8.69      0.95           7.91           9.46
## 7          C7 35  8.97      0.95           8.09           9.89
```

Gráfico de I.C. (los bigotes representan el intervalo de confianza):

```
ggplot(Sum1,aes(x = Contenedor, 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() +
```

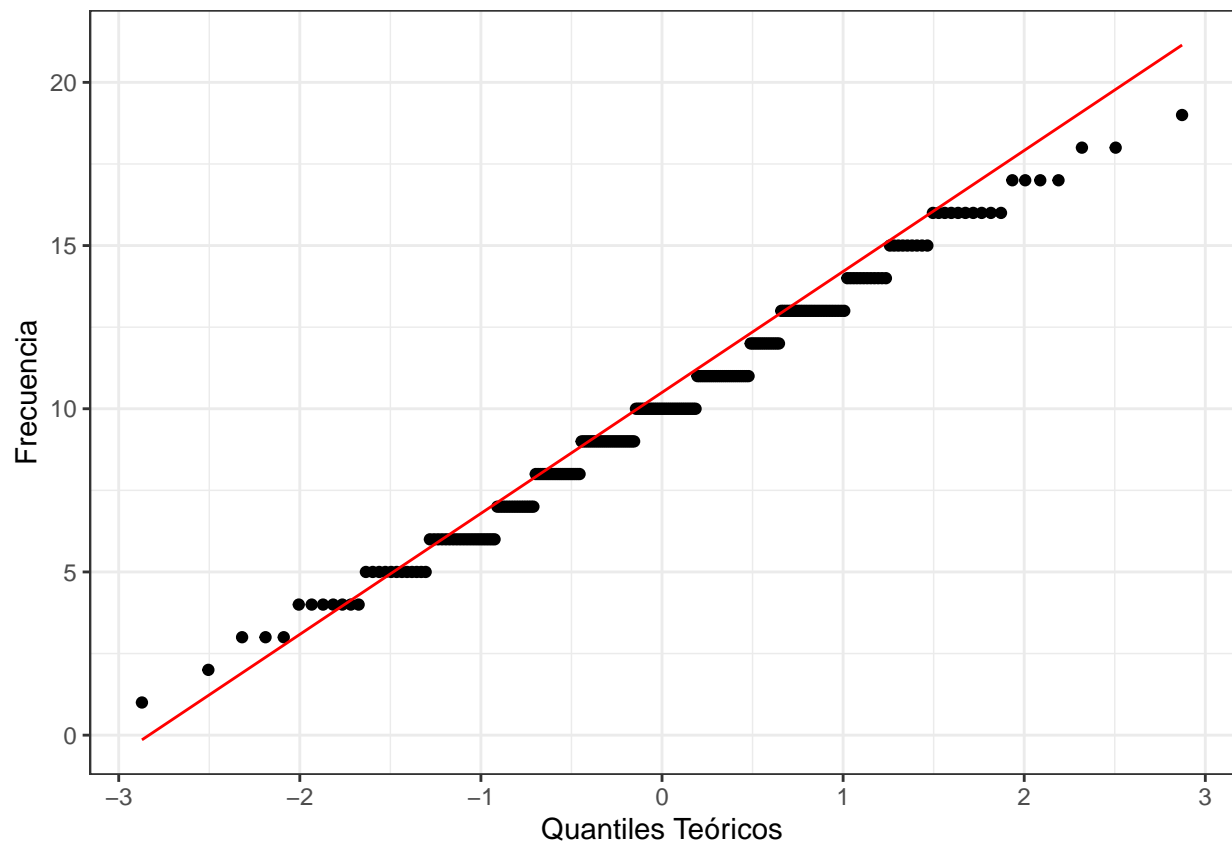
```
theme(axis.title = element_text(face = "bold")) +
ylab("Frecuencia")
```



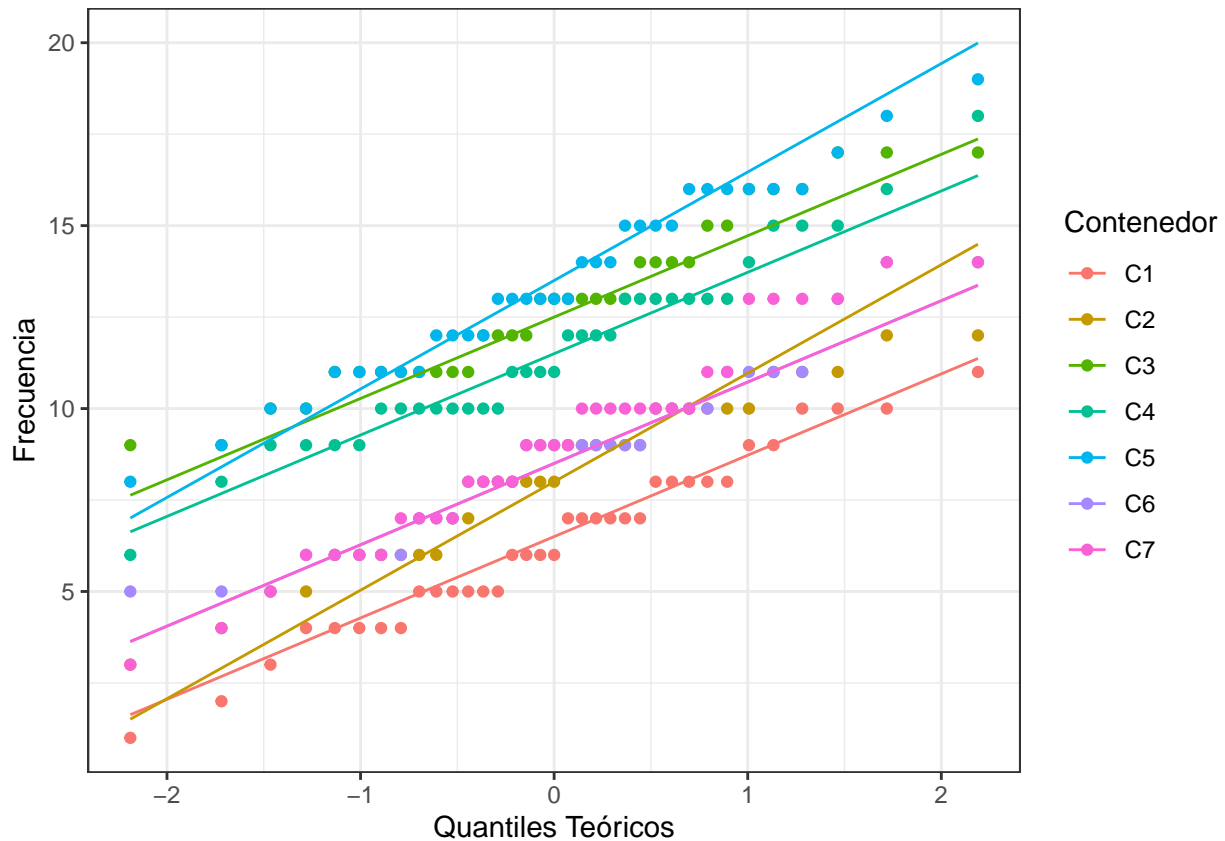
Gráficos Q-Q

```
library(ggplot2)

ggplot(d1, aes(sample=n)) +
  stat_qq() +
  stat_qq_line(col="red", distribution = stats::qnorm) +
  xlab("Quantiles Teóricos") + ylab("Frecuencia") +
  theme_bw()
```



```
p<-qqplot(sample = n, data = d1, color=Contenedor)+theme_bw()+stat_qq_line()+
  xlab("Quantiles Teóricos") + ylab("Frecuencia")
p
```

```
shapiro.test(d1$n)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d1$n
## W = 0.9882, p-value = 0.04209
```

Modelo lineal

```
model = lm(n ~ Contenedor, data = d1)
summary(model)
```

```
##
## Call:
## lm(formula = n ~ Contenedor, data = d1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.9714 -1.8571  0.0286  1.5714  6.4000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.3143     0.4135   15.270 < 2e-16 ***
## ContenedorC2   1.8571     0.5848    3.176  0.00169 **
## ContenedorC3   6.5429     0.5848   11.189 < 2e-16 ***
## ContenedorC4   5.2857     0.5848    9.039 < 2e-16 ***
```

```
## ContenedorC5    7.1143      0.5848  12.166 < 2e-16 ***
## ContenedorC6    2.3714      0.5848   4.055 6.79e-05 ***
## ContenedorC7    2.6571      0.5848   4.544 8.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.446 on 238 degrees of freedom
## Multiple R-squared:  0.509, Adjusted R-squared:  0.4967
## F-statistic: 41.13 on 6 and 238 DF, p-value: < 2.2e-16
```

Para las observaciones del experimento de dist. normal, el p-value resulta ser significativo, pero el R^2 es relativamente bajo, lo que quiere decir que hay mucha variabilidad pero la cantidad suficiente de datos explica que si hay una relación significativa.

Anova

```
library(car)

Anova(model, # Tipo 3 es el por defecto
      type = "II") # Suma de cuadrados

## Anova Table (Type II tests)
##
## Response: n
##           Sum Sq Df F value    Pr(>F)
## Contenedor 1476.7  6  41.127 < 2.2e-16 ***
## Residuals  1424.3 238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

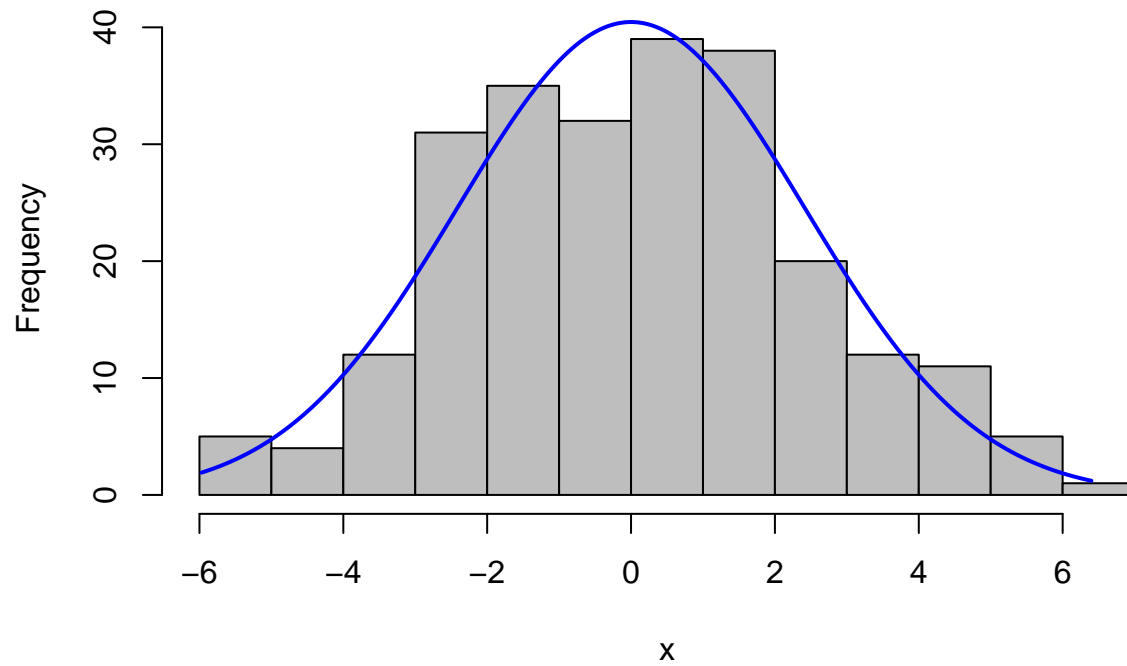
Este p-value tan pequeño indica que sí hay diferencia entre los contenedores, con un alto grado de significancia. Si la distribución fuera uniforme, esperaríamos tener la misma probabilidad para cada contenedor.

Supuesto de Normalidad

```
x = residuals(model)

library(rcompanion)

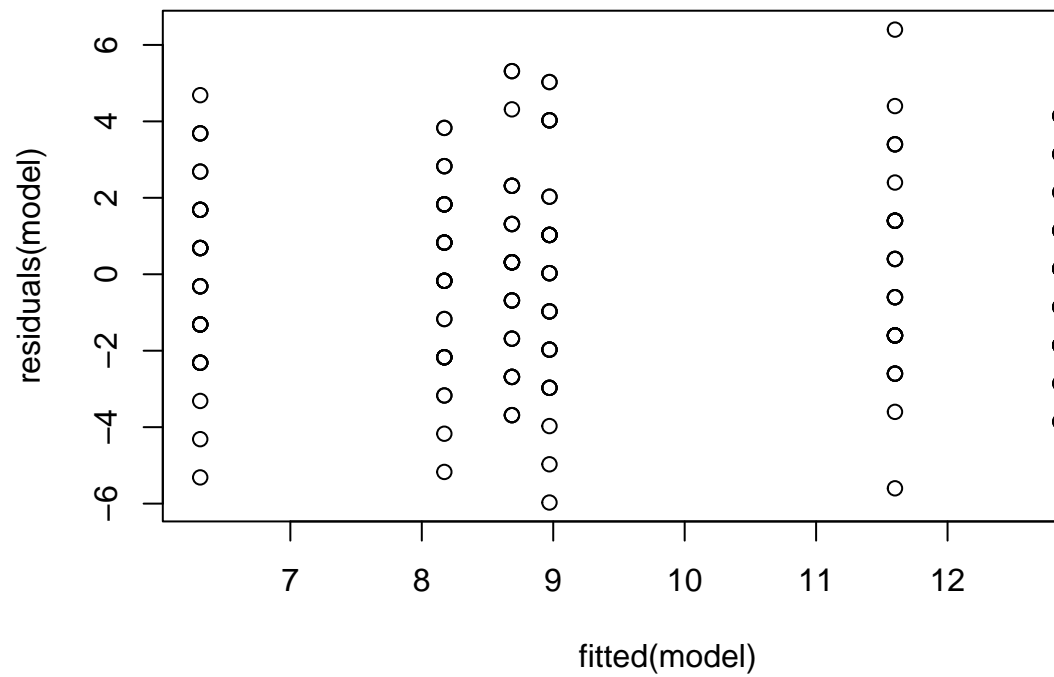
plotNormalHistogram(x)
```



Histograma de residuos

La distribución de residuos parece ser normal.

```
plot(fitted(model), residuals(model))
```



Homogeneidad de la varianza

Con confianza podemos decir que la varianza es similar.

Análisis post-hoc

Mínimos cuadrados para múltiples comparaciones

```
library(multcompView)
library(lsmeans)

marginal = lsmeans(model, ~ Contenedor)

# Ejecuta todas las comparaciones de pares para la variable
# Algoritmo.
pairs(marginal, adjust="tukey")
```

```
## contrast estimate SE df t.ratio p.value
## C1 - C2 -1.857 0.585 238 -3.176 0.0278
## C1 - C3 -6.543 0.585 238 -11.189 <.0001
## C1 - C4 -5.286 0.585 238 -9.039 <.0001
## C1 - C5 -7.114 0.585 238 -12.166 <.0001
## C1 - C6 -2.371 0.585 238 -4.055 0.0013
## C1 - C7 -2.657 0.585 238 -4.544 0.0002
## C2 - C3 -4.686 0.585 238 -8.013 <.0001
## C2 - C4 -3.429 0.585 238 -5.863 <.0001
## C2 - C5 -5.257 0.585 238 -8.990 <.0001
## C2 - C6 -0.514 0.585 238 -0.879 0.9753
## C2 - C7 -0.800 0.585 238 -1.368 0.8182
## C3 - C4 1.257 0.585 238 2.150 0.3272
## C3 - C5 -0.571 0.585 238 -0.977 0.9585
## C3 - C6 4.171 0.585 238 7.133 <.0001
## C3 - C7 3.886 0.585 238 6.645 <.0001
## C4 - C5 -1.829 0.585 238 -3.127 0.0322
## C4 - C6 2.914 0.585 238 4.984 <.0001
## C4 - C7 2.629 0.585 238 4.495 0.0002
## C5 - C6 4.743 0.585 238 8.111 <.0001
## C5 - C7 4.457 0.585 238 7.622 <.0001
## C6 - C7 -0.286 0.585 238 -0.489 0.9990
##
## P value adjustment: tukey method for comparing a family of 7 estimates
```

De esta tabla se pueden leer las comparaciones entre grupos, para revisar si son significativamente similares o diferentes. Sin embargo si la cantidad de factores es alta, hacer la comparación con la función “cld” es mas útil:

```
CLD = cld(marginal,
          alpha = 0.05,
          Letters = letters,
          adjust = "tukey")
```

```
## Note: adjust = "tukey" was changed to "sidak"
## because "tukey" is only appropriate for one set of pairwise comparisons
```

```
CLD # Nos muestra despliegue por letras
```

```
## Contenedor lsmean SE df lower.CL upper.CL .group
## C1 6.31 0.414 238 5.20 7.43 a
## C2 8.17 0.414 238 7.05 9.29 b
## C6 8.69 0.414 238 7.57 9.80 b
## C7 8.97 0.414 238 7.85 10.09 b
```

```
## C4      11.60 0.414 238    10.48    12.72    c
## C3      12.86 0.414 238    11.74    13.98    cd
## C5      13.43 0.414 238    12.31    14.55    d
##
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 7 estimates
## P value adjustment: tukey method for comparing a family of 7 estimates
## significance level used: alpha = 0.05
## NOTE: Compact letter displays can be misleading
##       because they show NON-findings rather than findings.
##       Consider using 'pairs()', 'pwpp()', or 'pwpm()' instead.
```

En este caso, CLD asigna una letra por cada tipo de agrupación de factores similares, en este caso el grupo “a” contiene solamente al Algoritmo C, pero el grupo B, contiene a los algoritmos A y B.

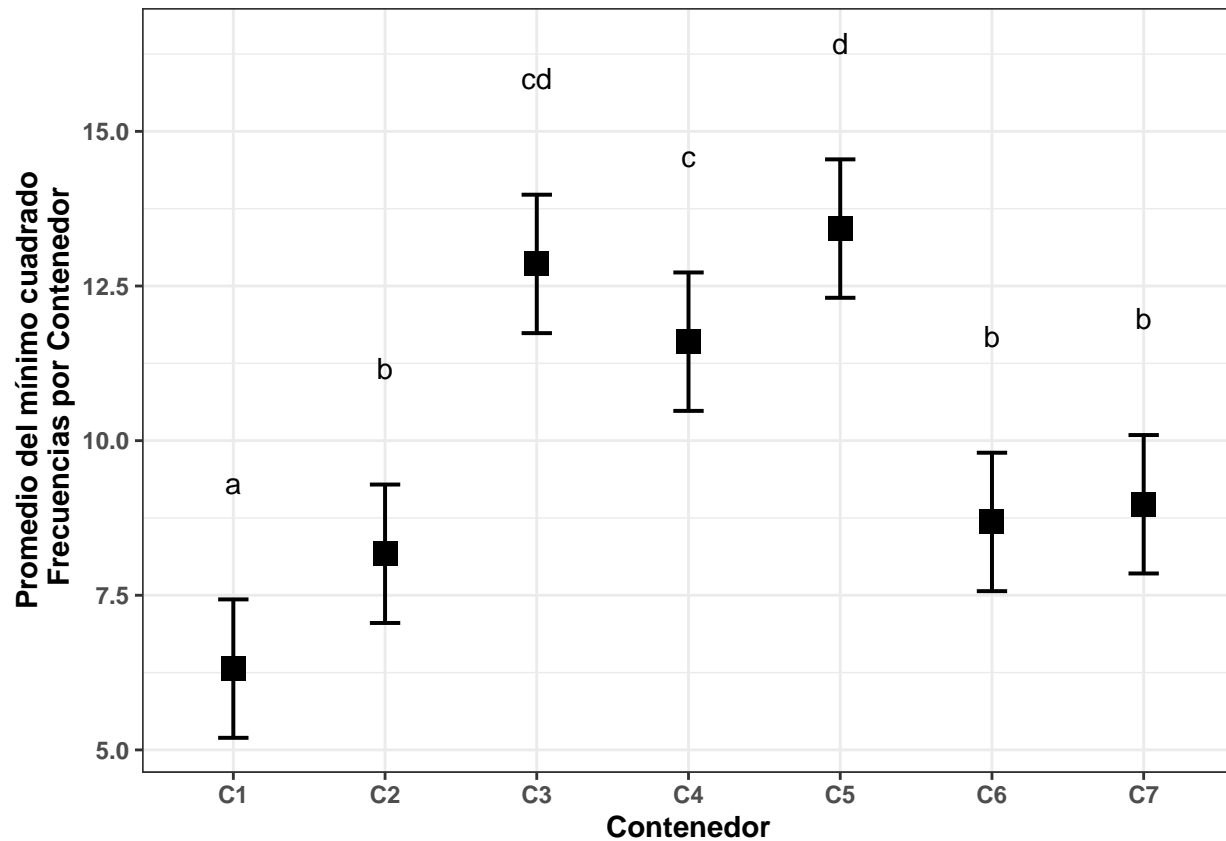
Graficación de promedios y separación entre grupos

```
CLD$Contenedor = factor(CLD$Contenedor,
                        levels = c("C1", "C2", "C3",
                                   "C4", "C5", "C6",
                                   "C7"))

# Removemos los espacios en blanco en CLD
CLD$.group=gsub(" ", "", CLD$.group)

library(ggplot2)

ggplot(
  CLD, aes(x = Contenedor, y = lsmean, label = .group)) +
  geom_point(shape = 15, size = 4) +
  geom_errorbar(
    aes(ymin = lower.CL, ymax = upper.CL),
    width = 0.2, size = 0.7) +
  theme_bw() +
  theme(axis.title = element_text(face = "bold"),
        axis.text = element_text(face = "bold"),
        plot.caption = element_text(hjust = 0)) +
  ylab("Promedio del mínimo cuadrado\n Frecuencias por Contenedor") +
  geom_text(nudge_x = c(0, 0, 0, 0, 0, 0, 0),
            nudge_y = c(3, 3, 3, 3, 3, 3, 3),
            color = "black")
```



- Los bigotes representan los intervalos de confianza para las diferencias de mínimos cuadrados.

Conclusión

Experimento 2: Distribución Uniforme

```
uniforme = Data_raw %>%
  filter(Distribucion == "Uniforme") %>%
  dplyr::select(C1, C2, C3, C4, C5, C6, C7)

summary(uniforme)
```

```
##           C1           C2           C3           C4
##  Min.    : 3.000   Min.    : 4.00   Min.    : 5.000   Min.    : 3.0
## 1st Qu.: 8.500   1st Qu.: 9.50   1st Qu.: 7.000   1st Qu.: 7.0
## Median :10.000   Median :10.00   Median : 9.000   Median : 9.0
## Mean    : 9.343   Mean    :10.26   Mean    : 9.029   Mean    : 8.6
## 3rd Qu.:10.500   3rd Qu.:11.50   3rd Qu.:11.000   3rd Qu.:10.0
## Max.    :15.000   Max.    :14.00   Max.    :15.000   Max.    :14.0
##           C5           C6           C7
##  Min.    : 3.0    Min.    : 7.00   Min.    : 9.00
## 1st Qu.: 7.5     1st Qu.: 9.00   1st Qu.:11.00
## Median :10.0     Median :11.00   Median :13.00
## Mean    : 9.8     Mean    :10.77   Mean    :12.77
## 3rd Qu.:12.0     3rd Qu.:12.00   3rd Qu.:14.00
## Max.    :19.0     Max.    :14.00   Max.    :20.00
```

Preprocesamiento

Cada fila se divide en múltiples filas, una por contenedor

```
library(tidyr)
d2 = gather(uniforme, "Contenedor", "n", 1:7, factor_key=TRUE,)
```

Visualización de los datos

Histograma de promedios

```
Summarize(n ~ Contenedor,
           data = d2, digits = 4)
```

##	Contenedor	n	mean	sd	min	Q1	median	Q3	max
## 1	C1	35	9.3429	2.6451	3	8.5	10	10.5	15
## 2	C2	35	10.2571	2.1052	4	9.5	10	11.5	14
## 3	C3	35	9.0286	2.4792	5	7.0	9	11.0	15
## 4	C4	35	8.6000	2.3414	3	7.0	9	10.0	14
## 5	C5	35	9.8000	3.4283	3	7.5	10	12.0	19
## 6	C6	35	10.7714	1.9416	7	9.0	11	12.0	14
## 7	C7	35	12.7714	2.3020	9	11.0	13	14.0	20

```
d2_summary = d2 %>%
  group_by(Contenedor) %>%
  summarise(avg = mean(n))
```

```
barplot(height=d2_summary$avg, names=d2_summary$Contenedor, main = "Medias por contenedor. Exp. 2", xlab = "Contenedor")
```

Medias por contenedor. Exp. 2

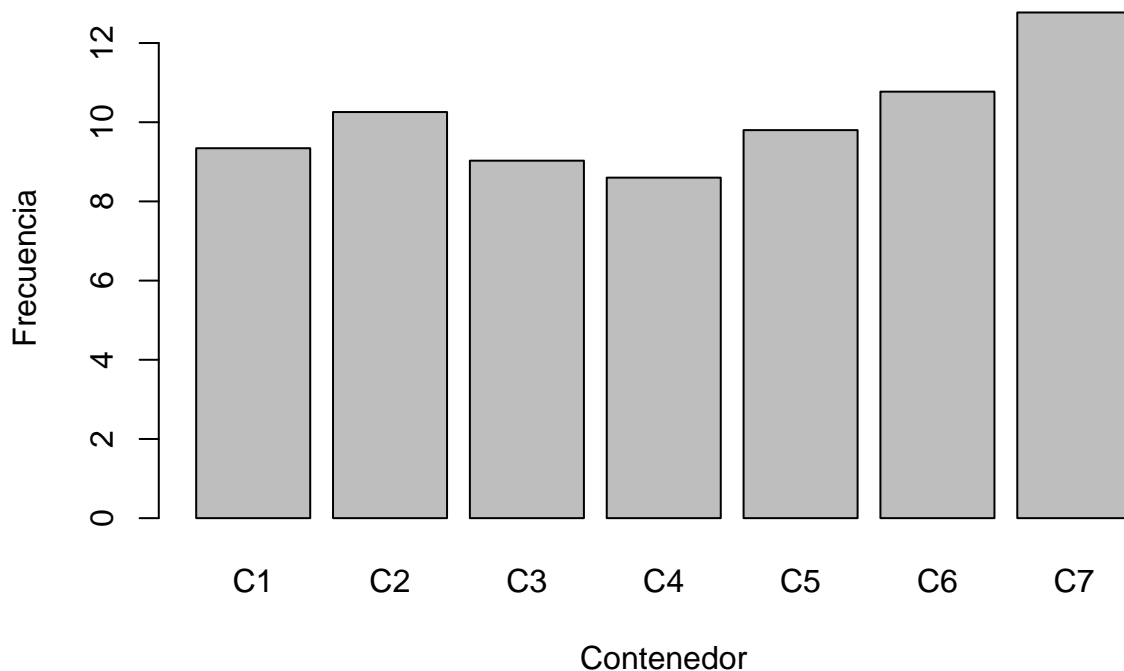


Diagrama de cajas

En este gráfico los bigotes representan el rango:

```
M = tapply(d2$n,
           INDEX = d2$Contenedor,
           FUN = mean)
boxplot(n ~ Contenedor, ylab= "Frecuencia",
        data = d2)
points(M, col="red", pch = "+",
       cex = 2)
```

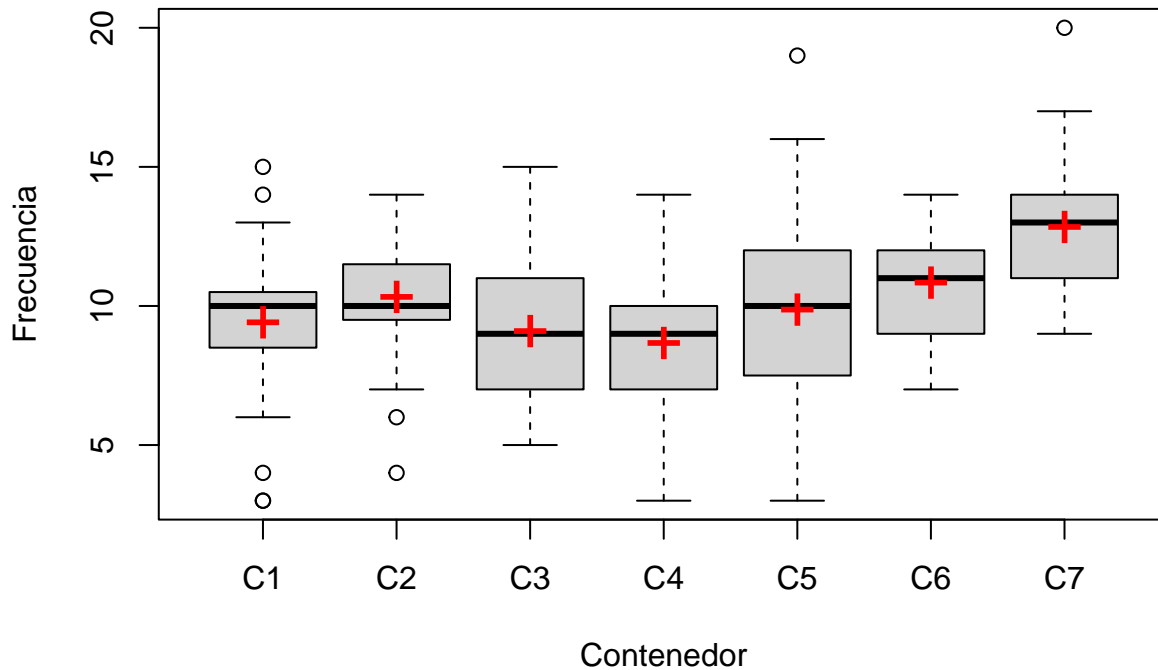


Gráfico de promedios e intervalos de confianza

Cálculo de los intervalos:

```
Sum2= groupwiseMean(n ~ Contenedor, data = d2, conf = 0.95, digits = 3,
                    traditional = FALSE, percentile = TRUE)
Sum2
```

##	Contenedor	n	Mean	Conf.level	Percentile.lower	Percentile.upper
## 1	C1	35	9.34	0.95	8.46	10.20
## 2	C2	35	10.30	0.95	9.54	10.90
## 3	C3	35	9.03	0.95	8.26	9.86
## 4	C4	35	8.60	0.95	7.83	9.37
## 5	C5	35	9.80	0.95	8.71	10.90
## 6	C6	35	10.80	0.95	10.10	11.40
## 7	C7	35	12.80	0.95	12.10	13.60

Gráfico de I.C. (los bigotes representan el intervalo de confianza):

```
ggplot(Sum2, aes(x = Contenedor, 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("Frecuencia")
```

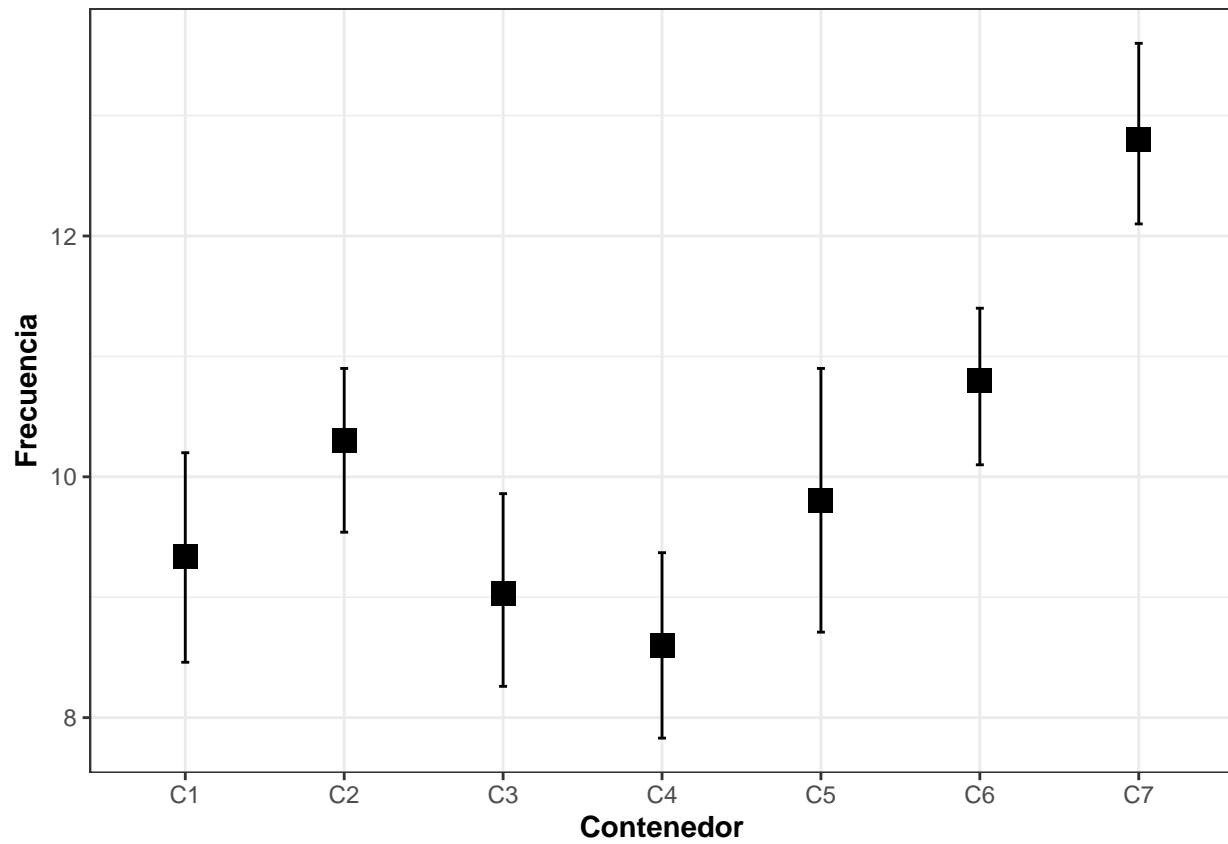
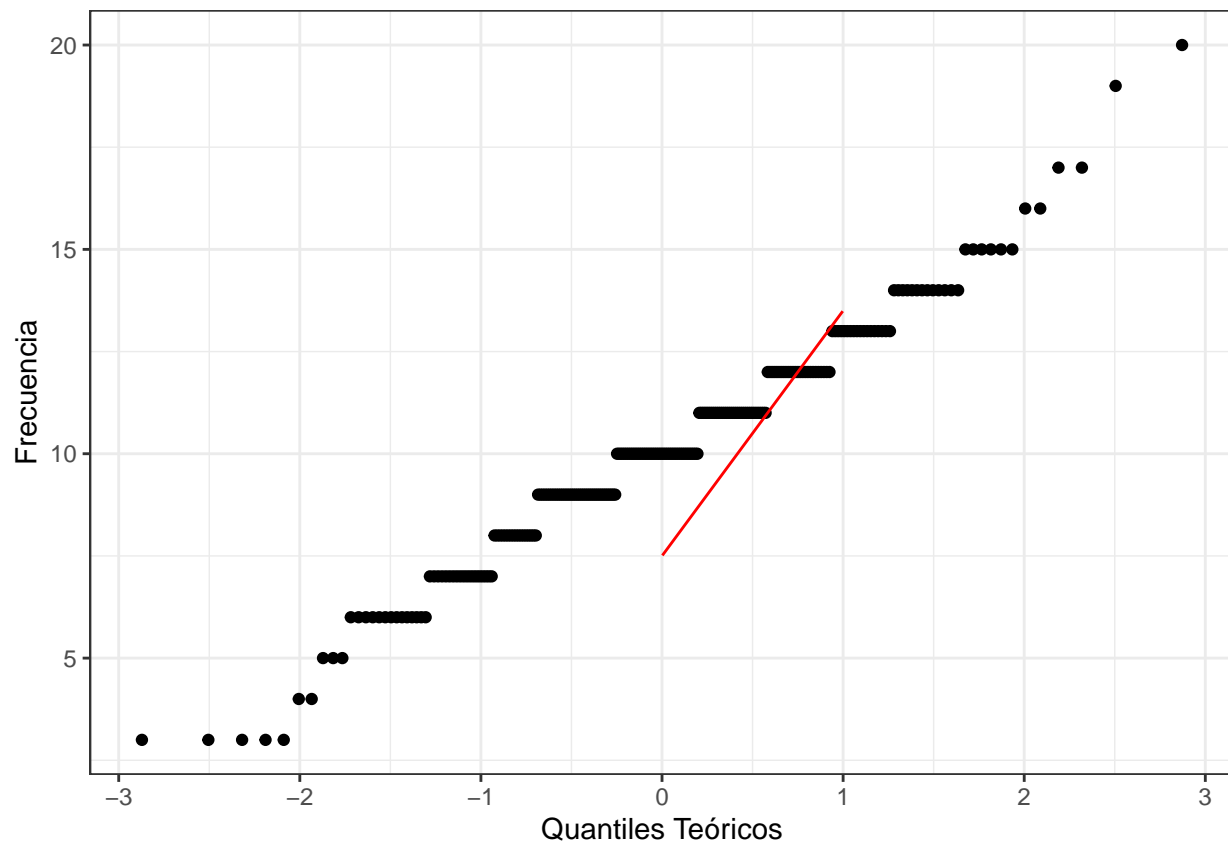


Gráfico Q-Q

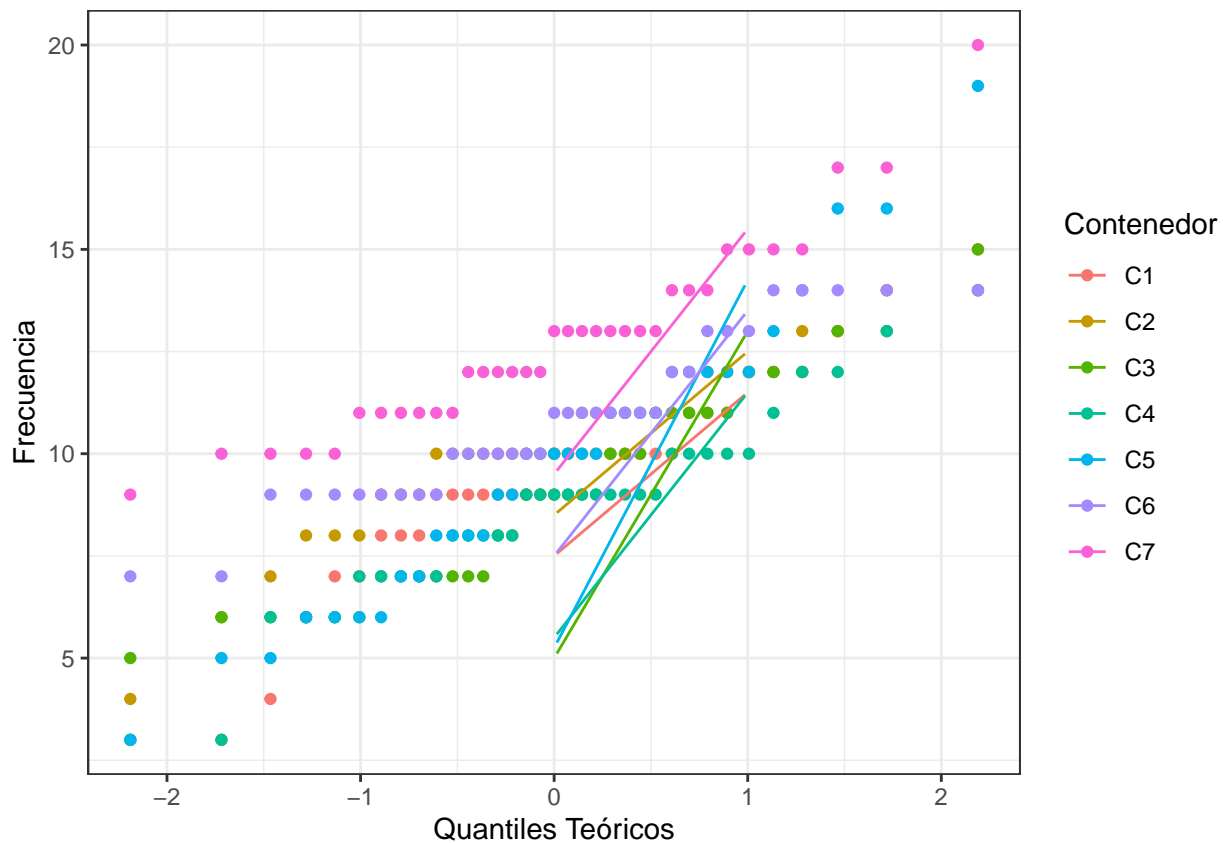
```
library(ggplot2)

ggplot(d2, aes(sample=n)) +
  stat_qq() +
  stat_qq_line(col="red", distribution = stats::qunif) +
  xlab("Quantiles Teóricos") + ylab("Frecuencia") +
  theme_bw()
```



Filtered w/o outliers

```
p1<-qplot(sample = n, data = d2, color=Contenedor)+
  theme_bw()+
  stat_qq_line( distribution = stats::qunif) +
  xlab("Quantiles Teóricos") + ylab("Frecuencia")
p1
```



Prueba de Pearson- χ^2 para determinar uniformidad

Dado: 1. H_0 (Nula): los datos presentan una distribución es uniforme. 2. $H - 1$ (Alternativa): los datos no presentan una distribución no es uniforme.

Se realiza la prueba de Pearson- χ^2 , con un alfa de 0.025, para un 95 de confianza, y 6 grados de confianza (dados los 7 contenedores del tablero):

```
d2_sum = d2 %>%
  group_by(Contenedor) %>%
  summarise(sum_observaciones = sum(n))

chisq.test(d2_sum$sum_observaciones)

##
## Chi-squared test for given probabilities
##
## data:  d2_sum$sum_observaciones
## X-squared = 40.517, df = 6, p-value = 3.604e-07
##Conclusión
```

Dado que el p-value de $3.604e^{-07}$ es menor que el alfa, se rechaza la H_0 , por lo que los datos no siguen una distribución predominantemente uniforme.

Esto se evidencia a la hora de ver el histograma, donde hay una inclinación predominante hacia la derecha.

Conlusiones finales

Parece que existe una degradación de la planta entre mas se utiliza, esto debido a la calidad de los materiales utilizados, y el estrés que las bolinchas de vidrio ejercen con cada repetición de los experimentos.

Para apoyar esta teoría, se realizo el test del Pearson- χ^2 en los dos datasets recolectados en las dos sesiones de experimentos, por aparte, donde para el primer set de pruebas:

```
Data_raw <- read_csv("datos_1.csv",
                     col_types = list(col_integer(),col_character(),col_integer(),
                                     col_integer(),col_integer(),col_integer(),
                                     col_integer(),col_integer(),col_integer(),
                                     col_integer()),
                     show_col_types = FALSE)

Data_raw$Distribucion = factor(
  Data_raw$Distribucion,
  levels = unique(Data_raw$Distribucion))

d2_sum = Data_raw %>%
  filter(Distribucion == "Uniforme") %>%
  dplyr::select(C1:C7) %>%
  gather("Contenedor", "n", 1:7, factor_key=TRUE) %>%
  group_by(Contenedor) %>%
  summarise(sum_observaciones = sum(n))

chisq.test(d2_sum$sum_observaciones)
```

```
##
## Chi-squared test for given probabilities
##
## data:  d2_sum$sum_observaciones
## X-squared = 25.853, df = 6, p-value = 0.0002372
```

Y para el segundo dataset recolectado:

```
Data_raw <- read_csv("datos_2.csv",
                     col_types = list(col_integer(),col_character(),col_integer(),
                                     col_integer(),col_integer(),col_integer(),
                                     col_integer(),col_integer(),col_integer(),
                                     col_integer()),
                     show_col_types = FALSE)

Data_raw$Distribucion = factor(
  Data_raw$Distribucion,
  levels = unique(Data_raw$Distribucion))

d2_sum = Data_raw %>%
  filter(Distribucion == "Uniforme") %>%
  dplyr::select(C1:C7) %>%
  gather("Contenedor", "n", 1:7, factor_key=TRUE) %>%
  group_by(Contenedor) %>%
  summarise(sum_observaciones = sum(n))

chisq.test(d2_sum$sum_observaciones)
```

```
##
```

```
## Chi-squared test for given probabilities
##
## data: d2_sum$sum_observaciones
## X-squared = 37.172, df = 6, p-value = 1.63e-06
```

De aquí se puede apreciar que el p-value del dataset de la primera sesión es mayor al p-value del dataset recolectado el segundo día, lo cual se podría interpretar como una consecuencia a la degradación de la planta.