Escuela Nacional de Estudios Superiores Unidad León

Centro de Ciencias Matemáticas

UGA-LANGEBIO CINVESTAV

Análisis estadístico de datos de Microbioma con R

Capítulo 8: Análisis univariado de comunidades

Equipo 4

Andrés Arredondo Cruz (andresabstract@gmail.com)
Adriana Haydé Contreras Peruyero (haydeeperuyero@gmail.com)
David Alberto García Estrada (david.garcia.e@cinvestav.mx)

Morelia

Septiembre de 2022

Índice

1. Introducción

Dividimos el estudio de la composición de las comunidades microbianas en dos grandes componentes:

(a)Evaluación de hipótesis sobre diversidad taxonómica, OTU y Taxones.
(b) Análisis de diferencias entre grupos.

El primer componente pertenece principalmente a análisis univariado de comunidades. El segundo puede ser dividido en varias técnicas multivariadas, como lo son "clustering" y "ordinations", y la evaluación de hipótesis de análisis multivariado de disimilitudes.

2. Comparaciones de diversidades entre dos grupos

En nuestro estudio con ratones Vdr−/−, uno de los propósitos es probar la diferencia de diversidades entre dos grupos (Vdr−/− y ratones de tipo salvaje) en sitios fecales y cecales. Aquí ilustraremos el análisis de la comunidad univariante, y compararemos la diversidad de Shannon (calculada anteriormente en el Cap. 6 en las muestras fecales) usando varios estadísticos de prueba.

2.1. Prueba t de Welch para dos muestras

El estadístico t fue introducido en 1908 por William Sealy Gosset. Una prueba t de dos muestras se utiliza para probar que las medias de dos poblaciones son iguales. Se aplica más comúnmente cuando el estadístico de prueba sigue una distribución normal. Si los dos grupos tienen la misma varianza, el estadístico t se puede calcular de la siguiente manera:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

donde, $s_p = \sqrt{\frac{(n-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$ es un estimador de desviación estándar de las dos muestras. La prueba t de Welch o la prueba t de varianzas desiguales es una adaptación de la prueba t. El estadístico de la prueba t de Welch está dado por:

$$t \;=\; \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{\Delta}}},$$

donde $s_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$; s_1^2 y s_2^2 son los estimadores imparciales de la varianza de las muestras 1 y 2, respectivamente. Cuando las dos muestras tienen varianzas desiguales y tamaños de muestra desiguales, la prueba t de Welch se considera más confiable.

Por lo tanto, aquí usamos la prueba t de Welch para nuestros datos de ratón Vdr−/−.

Primero, cargamos y calculamos la transpuesta del conjunto de datos:

```
# IMPORTANTE
# Ajusta el directorio de trabajo según tu PC
# Mostramos directorio actual
getwd()
```

```
## [1] "D:/Users/hayde/Documents/R_sites/Equipo4/Chapter8"
```

```
# Si es necesario, cambiar la ruta donde están los archivos almacenados.
# "." significa el directorio actual.
workingDir <-  "."
setwd(workingDir)
```

```
abund_table <-  read.csv(paste0(workingDir,"/data/VdrGenusCounts.csv"),
                         row.names=1, check.names=FALSE)
abund_table <- t(abund_table)
```

Para incorporar la información del grupo del conjunto de datos directamente a la comparación, necesitamos administrar los datos. En el conjunto de datos, la información del ID de la muestra y el grupo están en una franja de caracteres. Primero los extraemos de allí.

```
grouping <- data.frame(row.names = rownames(abund_table),
                       t(as.data.frame(strsplit(rownames(abund_table),"_"))))
grouping$Location <- with(grouping, ifelse(X3%in%"drySt-28F", "Fecal", "Cecal"))
grouping$Group <- with(grouping,ifelse(as.factor(X2)%in% c(11,12,13,14,15),
                                       c("Vdr-/-"), c("WT")))
grouping <- grouping[,c(4,5)]
grouping
```

	Location	Group
5_15_drySt-28F	Fecal	Vdr-/-
20_12_CeSt-28F	Cecal	Vdr-/-
1_11_drySt-28F	Fecal	Vdr-/-
2_12_drySt-28F	Fecal	Vdr-/-
3_13_drySt-28F	Fecal	Vdr-/-
4_14_drySt-28F	Fecal	Vdr-/-
7_22_drySt-28F	Fecal	WT
8_23_drySt-28F	Fecal	WT
9_24_drySt-28F	Fecal	WT
19_11_CeSt-28F	Cecal	Vdr-/-
21_13_CeSt-28F	Cecal	Vdr-/-
22_14_CeSt-28F	Cecal	Vdr-/-
23_15_CeSt-28F	Cecal	Vdr-/-
25_22_CeSt-28F	Cecal	WT
26_23_CeSt-28F	Cecal	WT
27_24_CeSt-28F	Cecal	WT

Repetimos el calculo de la diversidad de Shannon para esta tabla, igual que en el capítulo 6.

```
#library(vegan)
H<-diversity(abund_table, "shannon")
```

Luego combinamos los dataframes de la diversidad de Shannon y agrupación para crear un nuevo dataframe.

```
#hacemos un df de la diversidad de shannon
df_H<-data.frame(sample=names(H),value=H,measure=rep("Shannon",length(H)))
#combinamos la diversidad y los dataframes agrupados para crear un nuevo df
df_G <-cbind(df_H, grouping)
rownames(df_G)<-NULL
df_G
```

sample	value	measure	Location	Group
5_15_drySt-28F	2.460729	Shannon	Fecal	Vdr-/-
20_12_CeSt-28F	2.339725	Shannon	Cecal	Vdr-/-
1_11_drySt-28F	2.228023	Shannon	Fecal	Vdr-/-
2_12_drySt-28F	2.734405	Shannon	Fecal	Vdr-/-
3_13_drySt-28F	2.077282	Shannon	Fecal	Vdr-/-
4_14_drySt-28F	2.466830	Shannon	Fecal	Vdr-/-
7_22_drySt-28F	1.777171	Shannon	Fecal	WT
8_23_drySt-28F	1.999559	Shannon	Fecal	WT
9_24_drySt-28F	1.971996	Shannon	Fecal	WT
19_11_CeSt-28F	1.344813	Shannon	Cecal	Vdr-/-
21_13_CeSt-28F	2.016113	Shannon	Cecal	Vdr-/-
22_14_CeSt-28F	1.955432	Shannon	Cecal	Vdr-/-
23_15_CeSt-28F	1.614456	Shannon	Cecal	Vdr-/-
25_22_CeSt-28F	1.958839	Shannon	Cecal	WT
26_23_CeSt-28F	2.270818	Shannon	Cecal	WT
27_24_CeSt-28F	2.002195	Shannon	Cecal	WT

A continuación, creamos subconjuntos de datos fecales del nuevo dataframe.

```
Fecal_G<- subset(df_G, Location=="Fecal")
Fecal_G
```

	sample	value	measure	Location	Group
1	5_15_drySt-28F	2.460729	Shannon	Fecal	Vdr-/-
3	1_11_drySt-28F	2.228023	Shannon	Fecal	Vdr-/-
4	2_12_drySt-28F	2.734405	Shannon	Fecal	Vdr-/-
5	3_13_drySt-28F	2.077282	Shannon	Fecal	Vdr-/-
6	4_14_drySt-28F	2.466830	Shannon	Fecal	Vdr-/-
7	7_22_drySt-28F	1.777171	Shannon	Fecal	WT
8	8_23_drySt-28F	1.999559	Shannon	Fecal	WT
9	9_24_drySt-28F	1.971996	Shannon	Fecal	WT

Ahora, los datos están listos para el análisis estadístico. Antes de realizar la prueba de hipótesis, exploremos la distribución de los valores de diversidad de Shannon usando la función `ggplot()`.

```
#library(ggplot2)

#Ahora dividimos el gráfico en dos paneles usando facet_grid.
p<-ggplot(Fecal_G, aes(x=value))+
  geom_histogram(color="black", fill="black")+
  facet_grid(Group ~ .)
p
```

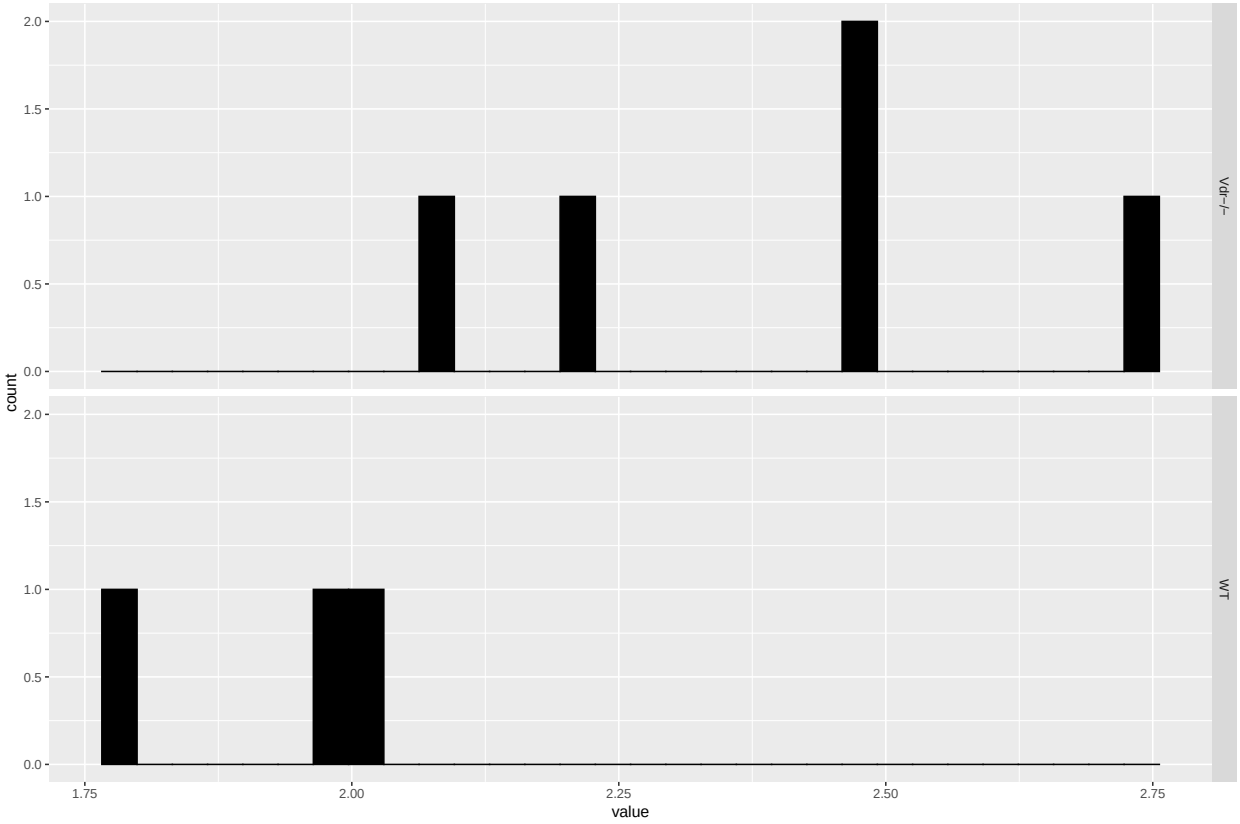

Figura 1: Diversidad de Shannon

El paquete `plyr` se utiliza para calcular los valores promedio de la diversidad de Shannon de cada grupo.

```
#library(plyr)
#la función ddply toma input un df y arroja un df de output
mu <- ddply(Fecal_G, "Group", summarise, grp.mean=mean(value))
head(mu)
```

Group	grp.mean
Vdr-/-	2.393454
WT	1.916242

```
#Agregamos las líneas de la media a la gráfica
p+geom_vline(data=mu, aes(xintercept=grp.mean, color="red"),
             linetype="dashed")
```

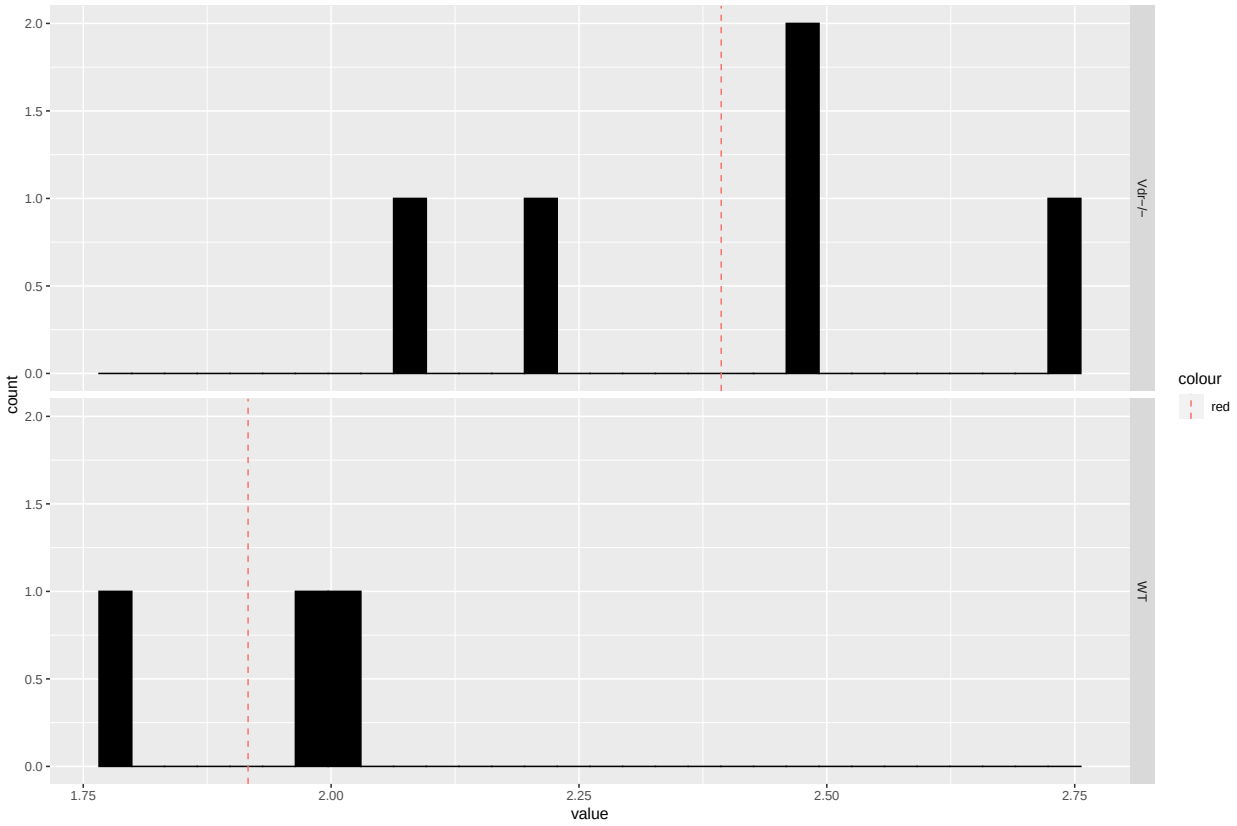

Figura 2: Diversidad de Shannon con líneas de media.

En el histograma anterior (Figura 2) la eliminación de Vdr de este grupo se desplaza hacia la derecha en relación con el grupo WT (hacia valores de diversidad más altos), lo que da como resultado una mayor diversidad.

Hacemos la prueba t de Welch.

```
#Usamos la prueba t de Welch para probar la hipótesis nula
fit_t <- t.test(value ~ Group, data=Fecal_G)
fit_t
```

```
##
##  Welch Two Sample t-test
##
## data:  value by Group
## t = 3.5999, df = 5.9206, p-value = 0.01163
## alternative hypothesis: true difference in means between group Vdr-/- and group WT is not equal to 0
## 95 percent confidence interval:
##  0.1517841 0.8026392
## sample estimates:
## mean in group Vdr-/-     mean in group WT
##             2.393454             1.916242
```

Para probar la hipótesis nula de que no hay diferencia en la diversidad de Shannon, se utilizó una prueba t de Welch que dio como resultado un valor de p = 0.0116305 (t = 3.5998798, df = 5.9206265). Por lo tanto,

rechazamos la hipótesis nula de no diferencia a favor de la alternativa de que las diversidades de Shannon son diferentes en los dos grupos.

2.2. La prueba de la suma de rangos de Wilcoxon

La prueba de la suma de rangos de Wilcoxon es equivalente a la prueba "U de Mann-Whitney" desarrollada por Mann y Whitney en 1947. Es una alternativa no paramétrica a la prueba t para dos muestras que utiliza rangos de datos de dos muestras que son independientes, para probar la hipótesis nula: las dos muestras independientes provienen de poblaciones con la misma distribución (es decir, las dos poblaciones son idénticas). A diferencia de la prueba t, *la prueba de suma de rangos de Wilcoxon* no requiere la suposición de distribuciones normales y es casi tan eficiente como la prueba t. Por lo tanto, es ampliamente utilizado en el estudio del microbioma. Se necesitan tres pasos principales para realizar *la prueba de suma de rangos de Wilcoxon* para encontrar el valor de la estadística de prueba:

Paso 1: Asignar rangos a todas las observaciones, el valor más pequeño obtiene un rango de 1. Cuando los valores estén empatados, asignar la media de los rangos involucrados en el empate.

Paso 2: Sumar los rangos de cualquiera de las dos muestras. La suma de rangos en otra muestra se puede determinar ya que la suma de todos los rangos es igual a $N(N+1)/2$, donde N es el número total de observaciones.

Si las dos poblaciones de prueba tienen la misma distribución, entonces el rango R tiene la media de $\mu_R = \frac{n_1(n_1+n_2+1)}{2}$ y la desviación estándar de $\sigma_R = \sqrt{\frac{n_1 n_2(n_1+n_2+1)}{12}}$. *La prueba de la suma de rangos de Wilcoxon* rechaza la hipótesis de que las dos poblaciones tienen distribuciones idénticas cuando la suma de rangos R está lejos de su media. El valor calculado de la suma de rangos se vuelve aproximadamente normal a medida que los dos tamaños de muestra aumentan. Podemos formar el valor estadístico estandarizando la suma de rangos.

Paso 3: Calcular el valor estadístico de la *prueba* z usando la fórmula siguiente:

$$z = \frac{R - \mu_R}{\sigma_R},$$

donde

- R: la suma de rangos de la muestra con número n_1.
- n_1: el tamaño de la muestra para el que se encuentra la suma de rangos R (como la muestra 1).
- n_2: otro tamaño de muestra (como la muestra 2).

El siguiente código se usa para realizar la prueba de suma de rangos de Wilcoxon.

```
##8.1.2 Wilcoxon Rank Sum Test
fit_w <- wilcox.test(value ~ Group, data=Fecal_G)
fit_w
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  value by Group
## W = 15, p-value = 0.03571
## alternative hypothesis: true location shift is not equal to 0
```

La Figura 2 muestra que las distribuciones están sesgadas con el pequeño tamaño de la muestra. La prueba de la suma de rangos de Wilcoxon puede ser más apropiada; sin embargo, el resultado $p = 0.0357143$ dado por *la prueba de la suma de rangos de Wilcoxon* conduce a la misma conclusión que la prueba t de Welch con $p = 0.0116305$ a un nivel de significancia de $0,05$.

3. Comparaciones de un taxón de interés entre dos grupos

3.1. Comparación de la abundancia relativa utilizando la prueba de la suma de rangos de Wilcoxon

Cuando analizamos los datos de abundancia de microbiomas, no es apropiado sacar inferencias sobre la abundancia total en el ecosistema a partir de la abundancia de OTUs o abundancia de taxones en las muestras. Más bien podemos usar la abundancia relativa en la muestra para inferir la abundancia relativa de un taxón en el ecosistema. La razón subyacente es que existe una restricción de composición: todas las abundancias relativas dentro de una muestra suman a uno, lo que da como resultado datos de composición residiendo en un **simplejo** en lugar de residir en el espacio euclidiano. Por lo tanto, a menudo es necesario estandarizar los datos a una escala común para facilitar la comparación de la abundancia del taxón entre grupos. La forma es dividir el conteo de taxones por el número total de lecturas en 100 para convertir la abundancia en el porcentaje de lecturas en la muestra, escalar los datos a "el número de taxones por 100 lecturas".

Cuando seleccionamos un solo taxón específico para probarlo en grupos, es importante asegurarse de que el taxón especificado se base en una hipótesis o teoría para reducir la posibilidad de inflar la tasa de falsos positivos (es decir, rechazar la hipótesis nula cuando no debería ser rechazada).

Vdr en ratones afecta sustancialmente la diversidad beta e influencia constantemente taxones bacterianos individuales, como los Parabacteroides (Wang et al. 2016). En esta sección, se ilustra la prueba de suma de rangos de Wilcoxon para comparar Bacteroides bacterianos en el conjunto de datos de ratones Vdr utilizando muestras fecales.

Primero, verificamos la abundancia total en cada muestra.

```
apply(abund_table,1, sum)
```

```
## 5_15_drySt-28F 20_12_CeSt-28F 1_11_drySt-28F 2_12_drySt-28F 3_13_drySt-28F
##           1853           3239           6211           5115           6016
## 4_14_drySt-28F 7_22_drySt-28F 8_23_drySt-28F 9_24_drySt-28F 19_11_CeSt-28F
##           2343           2262           7255           5502           5067
## 21_13_CeSt-28F 22_14_CeSt-28F 23_15_CeSt-28F 25_22_CeSt-28F 26_23_CeSt-28F
##           2397           3788           9264           2072           6903
## 27_24_CeSt-28F
##           6327
```

Luego, calculamos la abundancia relativa dividiendo cada valor por la abundancia total de la muestra:

```
#Calculamos la abundancia relativa
relative_abund_table <- decostand(abund_table, method = "total")
```

Comprobamos la abundancia total en cada muestra para comprobar que los cálculos anteriores sean correctos.

```
apply(relative_abund_table, 1, sum)
```

```
## 5_15_drySt-28F 20_12_CeSt-28F 1_11_drySt-28F 2_12_drySt-28F 3_13_drySt-28F
##              1              1              1              1              1
## 4_14_drySt-28F 7_22_drySt-28F 8_23_drySt-28F 9_24_drySt-28F 19_11_CeSt-28F
##              1              1              1              1              1
## 21_13_CeSt-28F 22_14_CeSt-28F 23_15_CeSt-28F 25_22_CeSt-28F 26_23_CeSt-28F
##              1              1              1              1              1
## 27_24_CeSt-28F
##              1
```

Eche un vistazo a los datos transformados.

```
#Visualizamos los datos transformados
relative_abund_table[1:16,1:8]
```

```
##                  Tannerella Lactococcus Lactobacillus Lactobacillus::Lactococcus
## 5_15_drySt-28F  0.256880734  0.17593092    0.05072855               0.0005396654
## 20_12_CeSt-28F  0.020685397  0.22753936    0.18431615               0.0037048472
## 1_11_drySt-28F  0.088391563  0.36982773    0.06987603               0.0040251167
## 2_12_drySt-28F  0.113000978  0.10713587    0.14056696               0.0009775171
## 3_13_drySt-28F  0.165558511  0.39527926    0.05352394               0.0028257979
## 4_14_drySt-28F  0.172428510  0.20102433    0.08749466               0.0004268032
## 7_22_drySt-28F  0.141025641  0.38992042    0.28470380               0.0057471264
## 8_23_drySt-28F  0.072501723  0.27195038    0.32253618               0.0020675396
## 9_24_drySt-28F  0.077062886  0.41948382    0.18175209               0.0025445293
## 19_11_CeSt-28F  0.000000000  0.08328399    0.06512729               0.0013814881
## 21_13_CeSt-28F  0.002503129  0.07217355    0.26658323               0.0000000000
## 22_14_CeSt-28F  0.005279831  0.15311510    0.16710665               0.0007919747
## 23_15_CeSt-28F  0.003993955  0.52536701    0.19635147               0.0026986183
## 25_22_CeSt-28F  0.018339768  0.34121622    0.30164093               0.0043436293
## 26_23_CeSt-28F  0.011734029  0.20338983    0.19716065               0.0014486455
## 27_24_CeSt-28F  0.037142406  0.30235499    0.05768927               0.0020546863
##                 Parasutterella Helicobacter   Prevotella Bacteroides
## 5_15_drySt-28F   0.0005396654  0.048030221 0.0652995143 0.147328656
## 20_12_CeSt-28F   0.0000000000  0.000000000 0.0021611609 0.010497067
## 1_11_drySt-28F   0.0001610047  0.000000000 0.0465303494 0.154242473
## 2_12_drySt-28F   0.0007820137  0.002541544 0.0193548387 0.073704790
## 3_13_drySt-28F   0.0003324468  0.003989362 0.0556848404 0.087433511
## 4_14_drySt-28F   0.0000000000  0.013657704 0.0610328638 0.085360649
## 7_22_drySt-28F   0.0000000000  0.001326260 0.0490716180 0.038019452
## 8_23_drySt-28F   0.0016540317  0.000000000 0.0122674018 0.058442453
## 9_24_drySt-28F   0.0001817521  0.000000000 0.0152671756 0.036713922
## 19_11_CeSt-28F   0.0000000000  0.000000000 0.0000000000 0.000000000
## 21_13_CeSt-28F   0.0000000000  0.000000000 0.0004171882 0.002085941
## 22_14_CeSt-28F   0.0000000000  0.000000000 0.0007919747 0.005279831
## 23_15_CeSt-28F   0.0002158895  0.000000000 0.0010794473 0.003346287
## 25_22_CeSt-28F   0.0000000000  0.000000000 0.0033783784 0.009169884
## 26_23_CeSt-28F   0.0002897291  0.000000000 0.0007243228 0.006663769
## 27_24_CeSt-28F   0.0000000000  0.000000000 0.0039513197 0.019282440
```

Nuestra bacteria de interés **Bacteroides** está en la columna 8, subdividamosla:

```
#subdividimos a los bacterioides
(Bacteroides <-relative_abund_table[,8])
```

```
## 5_15_drySt-28F 20_12_CeSt-28F 1_11_drySt-28F 2_12_drySt-28F 3_13_drySt-28F
##    0.147328656    0.010497067    0.154242473    0.073704790    0.087433511
## 4_14_drySt-28F 7_22_drySt-28F 8_23_drySt-28F 9_24_drySt-28F 19_11_CeSt-28F
##    0.085360649    0.038019452    0.058442453    0.036713922    0.000000000
## 21_13_CeSt-28F 22_14_CeSt-28F 23_15_CeSt-28F 25_22_CeSt-28F 26_23_CeSt-28F
##    0.002085941    0.005279831    0.003346287    0.009169884    0.006663769
## 27_24_CeSt-28F
##    0.019282440
```

Ahora, combinamos Bacteroides y los dataframes agrupados y creamos subconjuntos de las muestras fecales para un uso posterior.

```
#combine Bacterioides y los df agrupados y crea subconjuntos de las muestras
Bacteroides_G <-cbind(Bacteroides, grouping)
rownames(Bacteroides_G)<-NULL
Fecal_Bacteroides_G <- subset(Bacteroides_G, Location=="Fecal")
Fecal_Bacteroides_G
```

	Bacteroides	Location	Group
1	0.1473287	Fecal	Vdr-/-
3	0.1542425	Fecal	Vdr-/-
4	0.0737048	Fecal	Vdr-/-
5	0.0874335	Fecal	Vdr-/-
6	0.0853606	Fecal	Vdr-/-
7	0.0380195	Fecal	WT
8	0.0584425	Fecal	WT
9	0.0367139	Fecal	WT

La función `boxplot()` se usa para generar un diagrama de caja simple de Bacteroides con el grupo.

```
#creamos un boxplot con los  grupos
#Diagrama de caja de Bacteroides con grupos Vdr-/- y WT en muestras fecales
boxplot(Bacteroides ~ Group,data=Fecal_Bacteroides_G, col=rainbow(2),
        main="Bacteroides en ratones Vdr WT/KO")
```

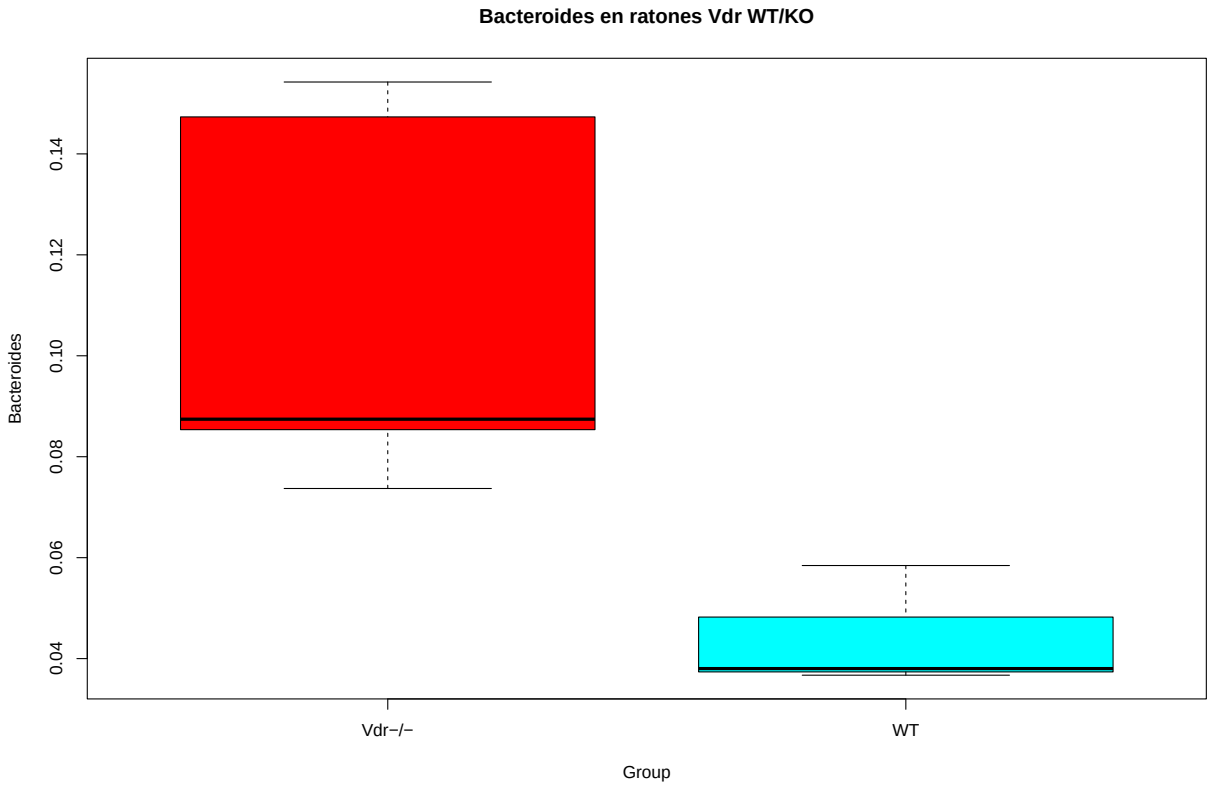

Figura 3: Diagrama de caja de Bacteroides con Vdr-/- y grupos WT en muestras fecales.

```
#con ggplot generamos el boxplot con el siguente codigo
ggplot(Fecal_Bacteroides_G, aes(x=Group, y=Bacteroides,col=factor(Group))) +
  geom_boxplot(notch=FALSE)
```

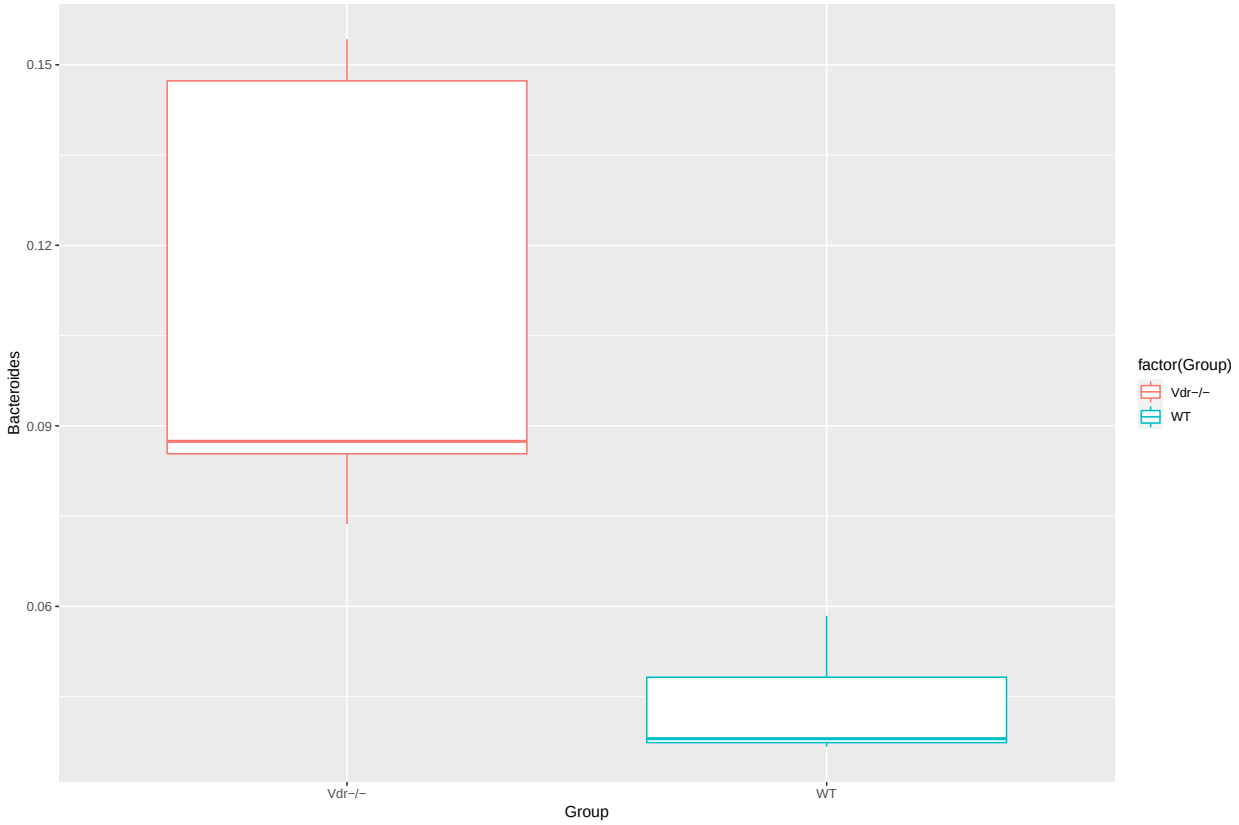

Figura 4: Diagrama de caja de Bacteroides con Vdr-/- y grupos WT en muestras fecales generadas usando ggplot.

```
#Diagrama de caja  bacterias Bacteroides con Vdr-/- y WT en muestras fecales
#generados usando ggplot
ggplot(Fecal_Bacteroides_G, aes(x=Group, y=Bacteroides)) +
  geom_boxplot(outlier.colour="red", outlier.shape=8, outlier.size=4) #+
```

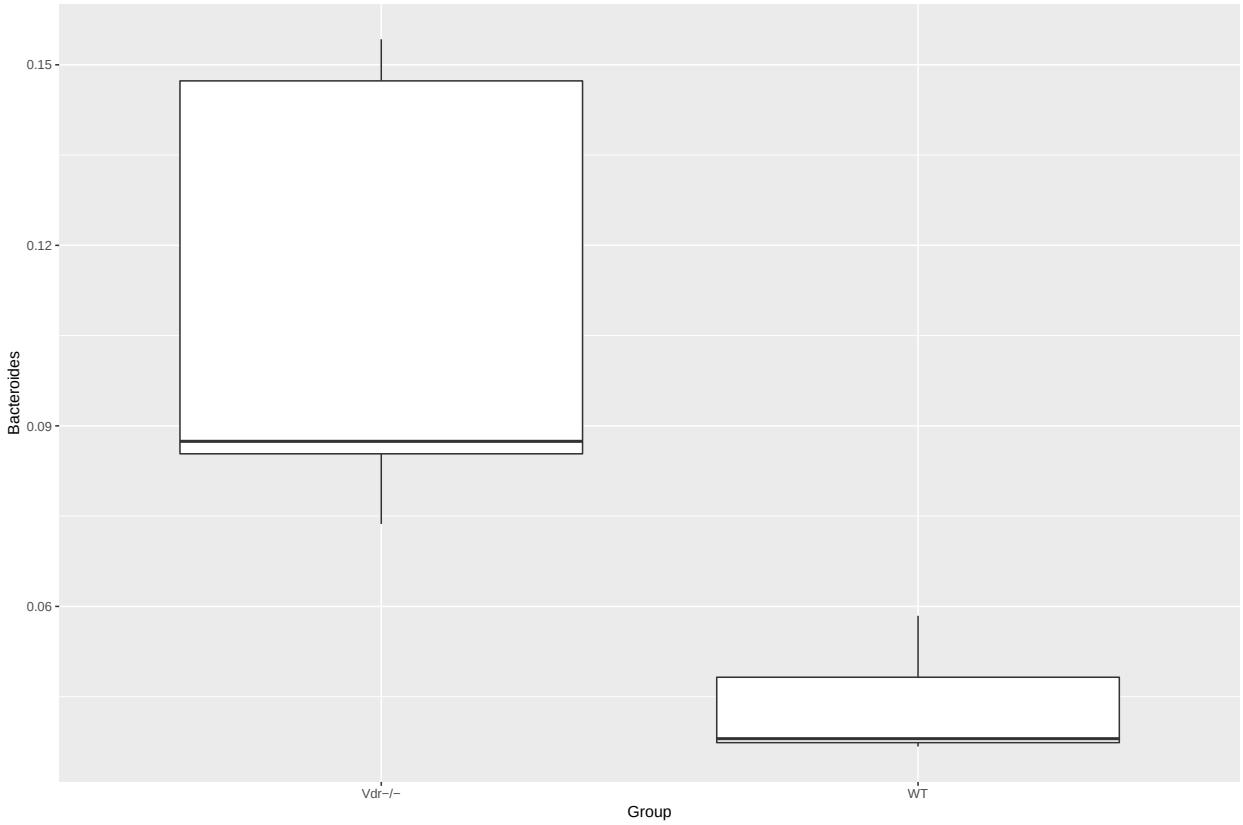

Figura 5: Diagrama de caja de Bacteroides con Vdr-/- y grupos WT en muestras fecales generadas usando ggplot con otro formato.

```
  #layer(stat_params = list(binwidth = 2))
#el argumento bindwith=2 genera un error
```

Los diagramas de caja muestran taxones (**Bacteriodes**) en ratones WT (WT, n = 3) y para ratones knockout de Vdr (KO, n = 5)

```
#Hacemos una prueba de suma de rangos de Wilcoxon
fit_w_b <- wilcox.test(Bacteroides ~ Group,data=Fecal_Bacteroides_G)
fit_w_b
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  Bacteroides by Group
## W = 15, p-value = 0.03571
## alternative hypothesis: true location shift is not equal to 0
```

La prueba de Wilcoxon anterior indica que existe una abundancia relativa estadísticamente significativa de Bacteroides entre los ratones Vdr−/− y WT. Podemos concluir que Vdr knockout enriquece Bacteroides.

3.2. Comparación de taxones presentes o ausentes utilizando la Prueba de chi-cuadrado

Una prueba de chi-cuadrado, también conocida como prueba χ^2, a menudo utilizada como abreviatura de la prueba de chi-cuadrado de Pearson, fué propuesta e investigada por primera vez por Karl Pearson en 1900. La prueba χ^2 se aplica a conjuntos de datos categóricos para probar si la distribución de frecuencia observada difiere de una distribución teórica o propuesta (pruebas de bondad de ajuste) e investigar si la variable fila y la variable columna en una tabla de contingencia son independientes entre sí (pruebas de independencia).

El estadístico de prueba de bondad de ajuste viene dado por:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = N \sum_{i=1}^{n} \frac{(O_i/N - p_i)^2}{p_i},$$

Donde:

X^2	El estadístico de prueba de Pearson, que se acerca asintóticamente a la distribución χ^2
O_i	número de observaciones de categoría i
N	Número total de observaciones
$E_i = Np_i$	Frecuencia esperada(teórica) de la categoría i bajo la hipótesis nula
p_i	probabilidad de categoría i en la población
n	número de celdas en la tabla

El estadístico de prueba de independencia se da de la siguiente forma.

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} = N \sum_{i,j} p_{i.} p_{.j} \left(\frac{(O_{i,j}/N) - p_{i.} p_{.j}}{p_{i.} p_{.j}} \right)^2,$$

Donde

N	Tamaño total de la muestra (la suma de todas las celdas de la tabla)
$E_{ij} = Np_{i.}p_{.j}$	Frecuencia esperada(teórica) bajo la hipótesis nula de independencia
$p_{i.} = \frac{O_{i.}}{N} = \sum_{j=1}^{c} \frac{O_{ij}}{N}$	Probabilidad de observaciones de categoría i ignorando el atributo de columna (probabilidad de totales de fila)
$p_{.j} = \frac{O_{.j}}{N} = \sum_{i=1}^{r} \frac{O_{ij}}{N}$	Probabilidad de observaciones de categoría j ignorando el atributo de fila (probabilidad de totales de columna)

Como regla general, se requiere que todos los recuentos de celdas esperados sean iguales o superiores a 5 para proporcionar una aproximación adecuada a la distribución de la distribución de Chi-cuadrado (Wackerly et al. 2002), aunque Cochran (1952) señaló que este valor podría ser tan bajo como 1 para algunas situaciones.

En esta sección, ilustramos la prueba χ^2 para comparar bacterias Parabacteroides en el conjunto de datos de ratones Vdr utilizando muestras cecales. Para ilustrar la prueba χ^2, transformamos los datos de conteo de abundancia de Parabacteroides en una variable binaria. Los datos de conteo en la tabla de abundancia para el taxón Parabacteroides se transformarían a 0 si el taxón está ausente en la muestra o 1 si el taxón está presente en la muestra. Los datos transformados se resumen en la tabla 6.

Grupo	**Presencia**	**Ausencia**	**Total**
Vdr$-/-$	3 (60 %)	2 (40 %)	5
WT	3 (100 %)	0 (0 %)	3

Cuadro 6: Distribució de la tasa de Parabacteroides en muestras Vdr $-/-$ y WT obtenidas del conjunto de datos de ratones Vdr.

Primero, observe los datos de abundancia para identificar Parabacteroides y subdividirlas.

```
#Distribution of the Parabacteroides rate
#across Vdr-/- and WT cecal samples obtained
abund_table[1:16,1:27]
```

```
##                Tannerella Lactococcus Lactobacillus Lactobacillus::Lactococcus
## 5_15_drySt-28F        476          326            94                           1
## 20_12_CeSt-28F         67          737           597                          12
## 1_11_drySt-28F        549         2297           434                          25
## 2_12_drySt-28F        578          548           719                           5
## 3_13_drySt-28F        996         2378           322                          17
## 4_14_drySt-28F        404          471           205                           1
## 7_22_drySt-28F        319          882           644                          13
## 8_23_drySt-28F        526         1973          2340                          15
## 9_24_drySt-28F        424         2308          1000                          14
## 19_11_CeSt-28F          0          422           330                           7
## 21_13_CeSt-28F          6          173           639                           0
## 22_14_CeSt-28F         20          580           633                           3
## 23_15_CeSt-28F         37         4867          1819                          25
## 25_22_CeSt-28F         38          707           625                           9
## 26_23_CeSt-28F         81         1404          1361                          10
## 27_24_CeSt-28F        235         1913           365                          13
##                Parasutterella Helicobacter Prevotella Bacteroides Barnesiella
## 5_15_drySt-28F              1           89        121         273           9
## 20_12_CeSt-28F              0            0          7          34           1
## 1_11_drySt-28F              1            0        289         958           2
## 2_12_drySt-28F              4           13         99         377           2
## 3_13_drySt-28F              2           24        335         526           1
## 4_14_drySt-28F              0           32        143         200           2
## 7_22_drySt-28F              0            3        111          86           0
## 8_23_drySt-28F             12            0         89         424           3
## 9_24_drySt-28F              1            0         84         202           1
## 19_11_CeSt-28F              0            0          0           0           0
## 21_13_CeSt-28F              0            0          1           5           0
## 22_14_CeSt-28F              0            0          3          20           0
## 23_15_CeSt-28F              2            0         10          31           1
## 25_22_CeSt-28F              0            0          7          19           0
## 26_23_CeSt-28F              2            0          5          46           0
## 27_24_CeSt-28F              0            0         25         122           1
##                Odoribacter Eubacterium Allobaculum Roseburia Clostridium
## 5_15_drySt-28F           1          52           0         1         130
## 20_12_CeSt-28F           0         131         241        18         401
## 1_11_drySt-28F          22         144           0        12         597
## 2_12_drySt-28F           7         238         271        44         815
## 3_13_drySt-28F           2         129          21         1         203
## 4_14_drySt-28F           4          90           3         0         232
## 7_22_drySt-28F           6          20           0         0          43
## 8_23_drySt-28F          35          88        1109         5         114
## 9_24_drySt-28F           8         192         237         1         184
## 19_11_CeSt-28F           0          15           1        34         425
```

```
## 21_13_CeSt-28F           0          77          42         7         234
## 22_14_CeSt-28F           0         191          26         2         753
## 23_15_CeSt-28F           0         514           3        16        1012
## 25_22_CeSt-28F           1          23           0        23         327
## 26_23_CeSt-28F           1         190        1235        20        1037
## 27_24_CeSt-28F           1         359         152        23         718
##                Porphyromonas Butyrivibrio Ruminococcus Acholeplasma Alistipes
## 5_15_drySt-28F             4           32           3            0        13
## 20_12_CeSt-28F             0          450          22            0         2
## 1_11_drySt-28F             6          136          25            1        84
## 2_12_drySt-28F             5          357          34            2       160
## 3_13_drySt-28F             2           89          20            2        28
## 4_14_drySt-28F             2           45          60            2        13
## 7_22_drySt-28F            15            2           1            0        11
## 8_23_drySt-28F             2           26           9            0        19
## 9_24_drySt-28F             7          179           1            0        27
## 19_11_CeSt-28F             0         3213          35            0         0
## 21_13_CeSt-28F             0          582          22            3         0
## 22_14_CeSt-28F             1           91          16            0         1
## 23_15_CeSt-28F             0           97          40            0         1
## 25_22_CeSt-28F             0           68          35            0         2
## 26_23_CeSt-28F             0          307          81            0        10
## 27_24_CeSt-28F             2         2021          43            0         4
##                Clostridium::Coprococcus
## 5_15_drySt-28F                        0
## 20_12_CeSt-28F                        0
## 1_11_drySt-28F                        0
## 2_12_drySt-28F                        0
## 3_13_drySt-28F                        0
## 4_14_drySt-28F                        0
## 7_22_drySt-28F                        0
## 8_23_drySt-28F                        0
## 9_24_drySt-28F                        0
## 19_11_CeSt-28F                        0
## 21_13_CeSt-28F                        0
## 22_14_CeSt-28F                        0
## 23_15_CeSt-28F                        0
## 25_22_CeSt-28F                        0
## 26_23_CeSt-28F                        3
## 27_24_CeSt-28F                        0
##                Eubacterium (Erysipelotrichaceae)::Eubacterium
## 5_15_drySt-28F                                              0
## 20_12_CeSt-28F                                              2
## 1_11_drySt-28F                                              1
## 2_12_drySt-28F                                              2
## 3_13_drySt-28F                                              0
## 4_14_drySt-28F                                              0
## 7_22_drySt-28F                                              0
## 8_23_drySt-28F                                              2
## 9_24_drySt-28F                                              0
## 19_11_CeSt-28F                                              4
## 21_13_CeSt-28F                                              1
## 22_14_CeSt-28F                                              1
## 23_15_CeSt-28F                                              0
```

```
## 25_22_CeSt-28F                                          0
## 26_23_CeSt-28F                                          7
## 27_24_CeSt-28F                                          0
##                 Hydrogenoanaerobacterium Paraprevotella Blautia
## 5_15_drySt-28F                         0             10       5
## 20_12_CeSt-28F                         0              0       6
## 1_11_drySt-28F                         2              3       6
## 2_12_drySt-28F                         0              6       9
## 3_13_drySt-28F                         0              4       0
## 4_14_drySt-28F                         0              6       3
## 7_22_drySt-28F                         1              3       0
## 8_23_drySt-28F                         0             12      35
## 9_24_drySt-28F                         0              8       1
## 19_11_CeSt-28F                         0              0       3
## 21_13_CeSt-28F                         0              0       0
## 22_14_CeSt-28F                         2              0       4
## 23_15_CeSt-28F                         0              0      59
## 25_22_CeSt-28F                         0              0      10
## 26_23_CeSt-28F                         0              0     551
## 27_24_CeSt-28F                         0              0      21
##                 Adlercreutzia::Asaccharobacter Coprococcus Parabacteroides
## 5_15_drySt-28F                               0          13              45
## 20_12_CeSt-28F                               0          13               0
## 1_11_drySt-28F                               0         116              25
## 2_12_drySt-28F                               6         119               4
## 3_13_drySt-28F                               0          10              14
## 4_14_drySt-28F                               0          12              16
## 7_22_drySt-28F                               0           0               5
## 8_23_drySt-28F                               1          22              14
## 9_24_drySt-28F                               1          12               6
## 19_11_CeSt-28F                               1           2               0
## 21_13_CeSt-28F                               1           1               1
## 22_14_CeSt-28F                               0           1               4
## 23_15_CeSt-28F                               5          36              15
## 25_22_CeSt-28F                               0           0               5
## 26_23_CeSt-28F                               4          43               4
## 27_24_CeSt-28F                               1          37               6
```

```
(Parabacteroides <- abund_table[,27])
```

```
## 5_15_drySt-28F 20_12_CeSt-28F 1_11_drySt-28F 2_12_drySt-28F 3_13_drySt-28F
##             45              0             25              4             14
## 4_14_drySt-28F 7_22_drySt-28F 8_23_drySt-28F 9_24_drySt-28F 19_11_CeSt-28F
##             16              5             14              6              0
## 21_13_CeSt-28F 22_14_CeSt-28F 23_15_CeSt-28F 25_22_CeSt-28F 26_23_CeSt-28F
##              1              4             15              5              4
## 27_24_CeSt-28F
##              6
```

Luego, combinamos los datos subdivididos con el marco de datos de agrupación.

```
#combina los datos subdivididos
Parabacteroides_G <-cbind(Parabacteroides, grouping)
rownames(Parabacteroides_G)<-NULL
```

Dado que el dataframe combinado incluye muestras fecales y cecales, subdividamos datos cecales de este dataframe

```
Cecal_Parabacteroides_G <- subset(Parabacteroides_G, Location=="Cecal")
Cecal_Parabacteroides_G
```

	Parabacteroides	Location	Group
2	0	Cecal	Vdr-/-
10	0	Cecal	Vdr-/-
11	1	Cecal	Vdr-/-
12	4	Cecal	Vdr-/-
13	15	Cecal	Vdr-/-
14	5	Cecal	WT
15	4	Cecal	WT
16	6	Cecal	WT

Antes, recodificamos una variable binaria: “Presente”, para la prueba de Chi-cuadrado.

```
Cecal_Parabacteroides_G$Present <- ifelse((
  Cecal_Parabacteroides_G$Parabacteroides > 0), "Present","Absent")
Cecal_Parabacteroides_G
```

	Parabacteroides	Location	Group	Present
2	0	Cecal	Vdr-/-	Absent
10	0	Cecal	Vdr-/-	Absent
11	1	Cecal	Vdr-/-	Present
12	4	Cecal	Vdr-/-	Present
13	15	Cecal	Vdr-/-	Present
14	5	Cecal	WT	Present
15	4	Cecal	WT	Present
16	6	Cecal	WT	Present

El siguiente código es usado para crear una prueba de Chi-cuadrado

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
tbl = table(Cecal_Parabacteroides_G$Group, Cecal_Parabacteroides_G$Present)
tbl
```

```
##
##          Absent Present
##   Vdr-/-      2       3
##   WT          0       3
```

```
chi <- chisq.test(tbl)
chi
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tbl
## X-squared = 0.17778, df = 1, p-value = 0.6733
```

La tabla 6 muestra la distribución de estas tasas: 3(60 %) de 5 muestras cecales Vdr−/− tenían Parabacteroides, mientras que 3(100 %) de 3 muestras de WT sí lo tenían. Para probar la hipótesis nula de que no hay diferencia en las tasas de ocurrencia entre los dos grupos, una prueba de chi-cuadrado dio un $p-$valor de 0.67329 (X-cuadrado = 0.1777778, df = 1), por lo que *no podemos rechazar la hipótesis nula de que no hay diferencia entre los dos grupos* y concluir que no tienen tasas diferentes de ocurrencia. Tenga en cuenta que debido al pequeño tamaño de la muestra, hay un mensaje de advertencia en la salida. Por lo general, si los valores de las celdas son pequeños (como < 5) en la tabla de contingencia, la prueba de chi-cuadrado puede ser incorrecta, entonces se aplica una prueba de exactitud de Fisher ver:

```
#prueba de exactitud de fisher
fisher <- fisher.test(tbl)
fisher
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tbl
## p-value = 0.4643
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1090089       Inf
## sample estimates:
## odds ratio
##        Inf
```

El resultado de la prueba de exactitud de Fisher tampoco es significativo con un $p-$valor de 0.4642857, *lo que es consistente con la prueba de Chi-cuadrado*. Sin embargo, es difícil tener conclusiones sobre la prueba con el intervalo de confianza infinito.

4. Comparación entre más de dos grupos usando ANOVA

4.1. ANOVA de una vía

El análisis de varianza (ANOVA) fue propuesto por Ronal Fisher en 1918 y se hizo bien conocido después de la publicación del libro de Fisher, *"Statitiscal Methods for Research Workers"* en 1925. El ANOVA generaliza

la prueba de t de dos muestras a más de dos grupos. La **Hipótesis nula** del ANOVA es: *Todas las medias de grupos comparados son iguales*. El análisis usando ANOVA se basa en un supuesto de *Normalidad* de los datos subyacentes. Sin embargo, la composición de la mayoría de los datos de comunidades microbiana, especialmente datos multivariados, no están normalmente distribuidos, por lo tanto, ANOVA solo es usaod para comparar análisis univariado de mediciones de alfa diversidad. Para datos de composición comunitaria multivariante, se aplica una versión no paramétrica La formación de la estadística de prueba se realiza mediante la partición tradicional de la suma de cuadrados (división de la variación). La ecuación de definición de la muestra varianza es

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

donde $s^{2} = $ varianza de la muestra. La varianza de la muestra es calculada por la suma de cuadados (SS) dividido por $n-1$ (grados de libertad, DF). El resultado es llamado media cuadrada (MS) y los términos cuadrados son desviaciones de la media simple. La técnica fundamental del ANOVA divide la suma total de cuadrados (SS) de desviaciones en dos componentes: suma de cuadrados relacionados con el tratamiento y suma de cuadrados relacionados con el error:

$$SS_{Total} = SS_{Tratamientos} + SS_{Error}$$

El número de grados de libertad, DF, pueden ser particionados de una manera similar:

$$DF_{Total} = DF_{Tratamientos} + DF_{Error}$$

La prueba F es usada para comparar los factores de la desviación total. El valor F es obtenido por la división de la varianza entre tratamientos por la varianza dentro de tratamientos. La prueba estadística F en un ANOVA de una vía esta dado por:

$$F = \frac{MS_{Tratamientos}}{MS_{Error}} = \frac{SS_{Tratamientos}/(K-1)}{SS_{Error}/(N-1)}$$

donde MS = media cuadrada, K = número de tratamientos y N = número total de muestras. Para ilustrar el ANOVA en el estudio de la composición de la comunidad del microbioma, utilizamos los datos de nuestro ratón $Vdr^{-/-}$. Una hipótesis de este estudio es que el estado de Vdr y la localización intestinal no tienen efectos en la comunidad bacteriana del intestino. Analizamos las medidas de diversidad alfa de Chao 1 utilizando ANOVA para abordar esta hipótesis. Los siguientes códigos crean un marco de datos de riqueza Chao1 y añaden la información de los grupos a este marco de datos

```
CH <- estimateR(abund_table)[2,]
df_CH <- data.frame(sample = names(CH), value = CH,
                    measure = rep("Chao1", length(CH)))
df_CH_G <- cbind(df_CH, grouping)
rownames(df_G) <- NULL
df_CH_G
```

	sample	value	measure	Location	Group
5_15_drySt-28F	5_15_drySt-28F	94.75000	Chao1	Fecal	Vdr-/-
20_12_CeSt-28F	20_12_CeSt-28F	59.80000	Chao1	Cecal	Vdr-/-
1_11_drySt-28F	1_11_drySt-28F	77.00000	Chao1	Fecal	Vdr-/-
2_12_drySt-28F	2_12_drySt-28F	103.27273	Chao1	Fecal	Vdr-/-

	sample	value	measure	Location	Group
3_13_drySt-28F	3_13_drySt-28F	85.66667	Chao1	Fecal	Vdr-/-
4_14_drySt-28F	4_14_drySt-28F	55.14286	Chao1	Fecal	Vdr-/-
7_22_drySt-28F	7_22_drySt-28F	62.75000	Chao1	Fecal	WT
8_23_drySt-28F	8_23_drySt-28F	67.66667	Chao1	Fecal	WT
9_24_drySt-28F	9_24_drySt-28F	80.50000	Chao1	Fecal	WT
19_11_CeSt-28F	19_11_CeSt-28F	52.16667	Chao1	Cecal	Vdr-/-
21_13_CeSt-28F	21_13_CeSt-28F	55.00000	Chao1	Cecal	Vdr-/-
22_14_CeSt-28F	22_14_CeSt-28F	59.00000	Chao1	Cecal	Vdr-/-
23_15_CeSt-28F	23_15_CeSt-28F	60.87500	Chao1	Cecal	Vdr-/-
25_22_CeSt-28F	25_22_CeSt-28F	51.00000	Chao1	Cecal	WT
26_23_CeSt-28F	26_23_CeSt-28F	112.85714	Chao1	Cecal	WT
27_24_CeSt-28F	27_24_CeSt-28F	78.05882	Chao1	Cecal	WT

Los nuevos 4 niveles de grupo son generados usando interacción de Locación y Grupo

```
df_CH_G$Group4 <- with(df_CH_G, interaction(Location,Group))
df_CH_G
```

	sample	value	measure	Location	Group	Group4
5_15_drySt-28F	5_15_drySt-28F	94.75000	Chao1	Fecal	Vdr-/-	Fecal.Vdr-/-
20_12_CeSt-28F	20_12_CeSt-28F	59.80000	Chao1	Cecal	Vdr-/-	Cecal.Vdr-/-
1_11_drySt-28F	1_11_drySt-28F	77.00000	Chao1	Fecal	Vdr-/-	Fecal.Vdr-/-
2_12_drySt-28F	2_12_drySt-28F	103.27273	Chao1	Fecal	Vdr-/-	Fecal.Vdr-/-
3_13_drySt-28F	3_13_drySt-28F	85.66667	Chao1	Fecal	Vdr-/-	Fecal.Vdr-/-
4_14_drySt-28F	4_14_drySt-28F	55.14286	Chao1	Fecal	Vdr-/-	Fecal.Vdr-/-
7_22_drySt-28F	7_22_drySt-28F	62.75000	Chao1	Fecal	WT	Fecal.WT
8_23_drySt-28F	8_23_drySt-28F	67.66667	Chao1	Fecal	WT	Fecal.WT
9_24_drySt-28F	9_24_drySt-28F	80.50000	Chao1	Fecal	WT	Fecal.WT
19_11_CeSt-28F	19_11_CeSt-28F	52.16667	Chao1	Cecal	Vdr-/-	Cecal.Vdr-/-
21_13_CeSt-28F	21_13_CeSt-28F	55.00000	Chao1	Cecal	Vdr-/-	Cecal.Vdr-/-
22_14_CeSt-28F	22_14_CeSt-28F	59.00000	Chao1	Cecal	Vdr-/-	Cecal.Vdr-/-
23_15_CeSt-28F	23_15_CeSt-28F	60.87500	Chao1	Cecal	Vdr-/-	Cecal.Vdr-/-
25_22_CeSt-28F	25_22_CeSt-28F	51.00000	Chao1	Cecal	WT	Cecal.WT
26_23_CeSt-28F	26_23_CeSt-28F	112.85714	Chao1	Cecal	WT	Cecal.WT
27_24_CeSt-28F	27_24_CeSt-28F	78.05882	Chao1	Cecal	WT	Cecal.WT

Exploramos el índice de Chao usando `boxplot()`

```
boxplot(value~Group4, data=df_CH_G, col = rainbow(4), main="Chao1 index")
```

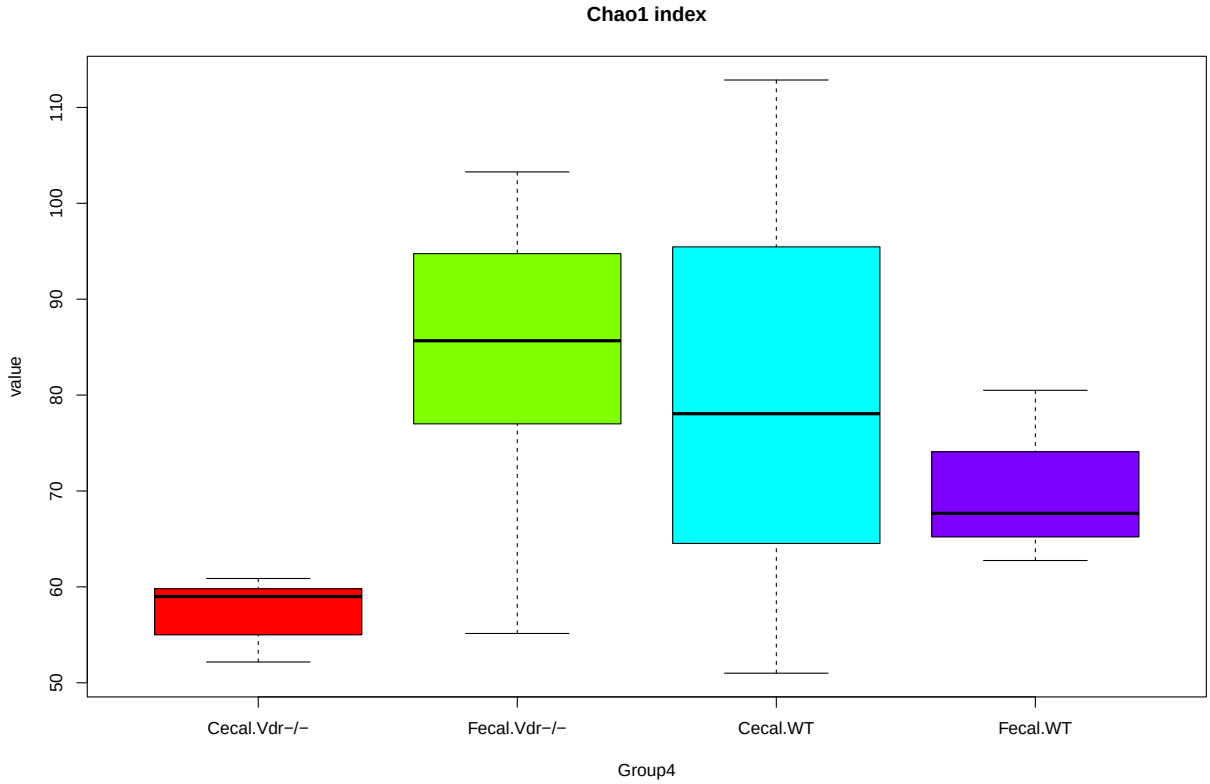

Figura 6: Boxplot del índice Chao 1 con cuatro grupos generados usando la función 'boxplot()'.

El siguiente `ggplot()` genera una alta calidad de boxplot para publicarlo

```
library(ggplot2)
p <- ggplot(df_CH_G, aes(x=Group4, y=value),
            col=rainbow(4), main="Chao1 index") +
  geom_boxplot()
p + coord_flip()
```

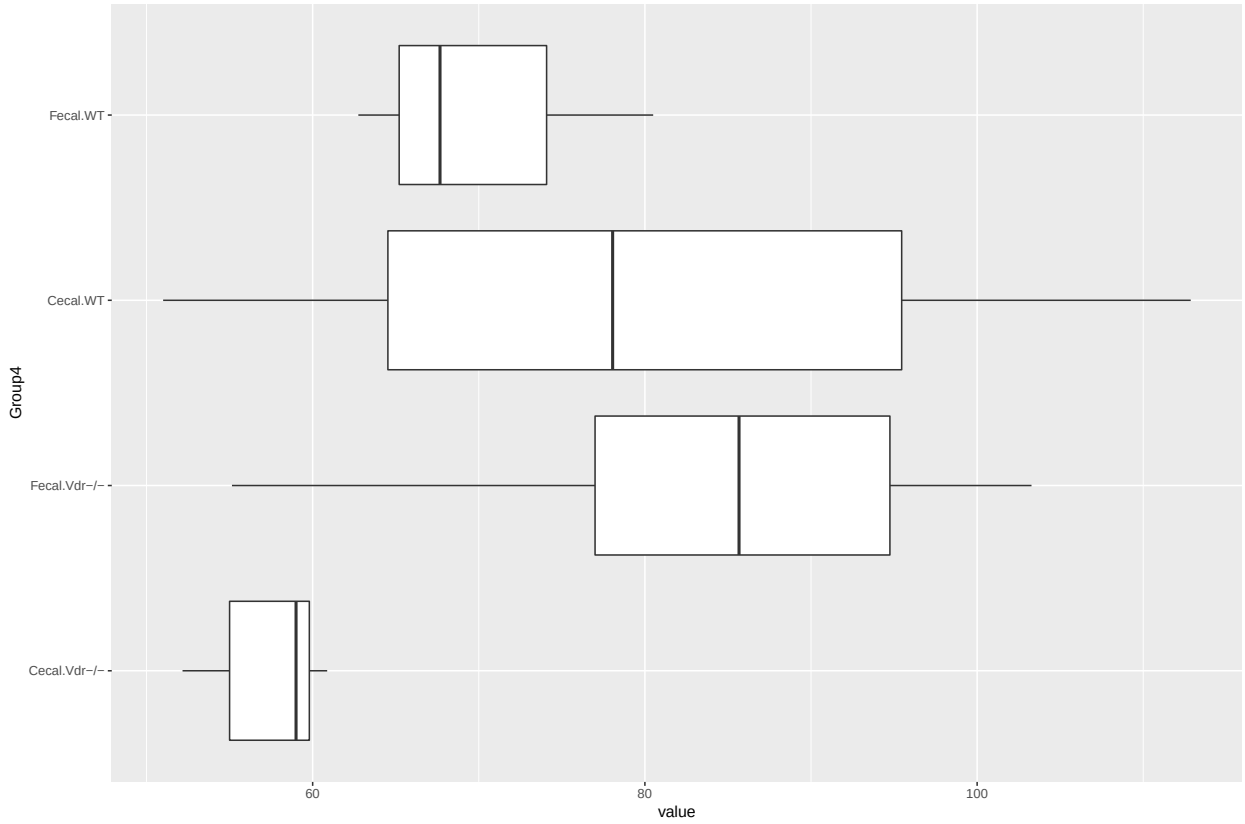

Figura 7: Boxplot del índice Chao 1 con cuatro grupos generados usando la función 'ggplot()'.

```
ggplot(df_CH_G, aes(x = Group4, y = value, col = factor(Group4))) +
  geom_boxplot(notch = FALSE)
```

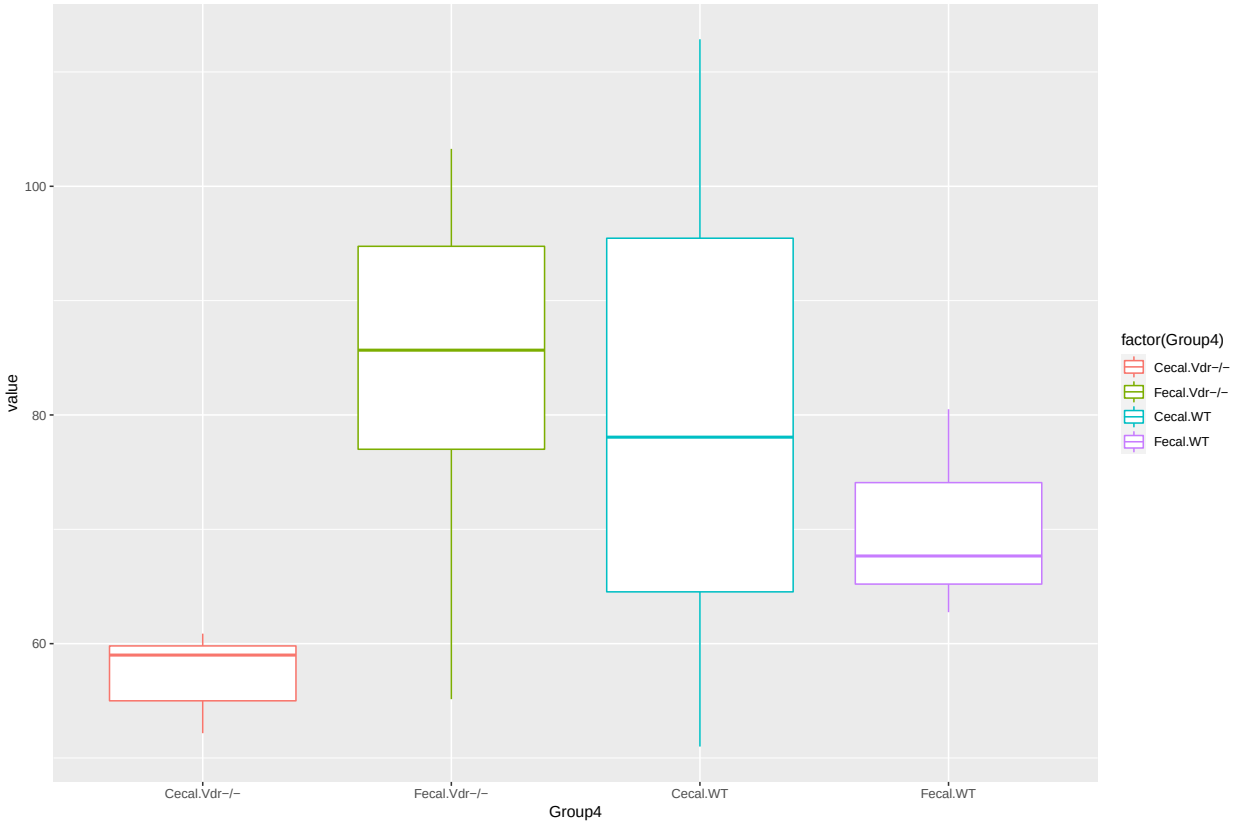

Figura 8: Boxplot del índice Chao 1 con cuatro grupos generados usando la función 'ggplot()'.

Además de la inspección visual de la normalidad de los datos subyacentes, la homogeneidad de varianza puede ser probada. Sokal y Rohlf (2011) describen tres pruebas de este tipo: la prueba de homogeneidad de Bartlett, la prueba F_{max} de Hartley y la prueba log-anova o Scheffé-Box. Para proceder a la comprobación del uso de ANOVA, primero debemos comprobar la homogeneidad de las varianzas. El software R proporciona dos pruebas: la prueba de Bartlett y la prueba de Fligner-Killeen. Para ilustrar la prueba de homogeneidad de varianzas, utilizamos las medidas de riqueza Chao 1 de los datos de los ratones $Vdr^{-/-}$ y WT de las localizaciones fecales y cecales. La hipótesis nula (H_0) de es que todas las varianzas de los cuatro grupos son iguales. Comenzamos con la prueba de Bartlett. Para facilitar el procesamiento de la prueba de Bartlett de Bartlett, utilizamos la función `select()` del paquete dplyr para seleccionar el grupo pertinente y las columnas de valor de Chao 1.

```
library(dplyr)

df_CH_G4 <- dplyr::select(df_CH_G, Group4, value)
df_CH_G4
```

	Group4	value
5_15_drySt-28F	Fecal.Vdr-/-	94.75000
20_12_CeSt-28F	Cecal.Vdr-/-	59.80000
1_11_drySt-28F	Fecal.Vdr-/-	77.00000
2_12_drySt-28F	Fecal.Vdr-/-	103.27273
3_13_drySt-28F	Fecal.Vdr-/-	85.66667
4_14_drySt-28F	Fecal.Vdr-/-	55.14286

	Group4	value
7_22_drySt-28F	Fecal.WT	62.75000
8_23_drySt-28F	Fecal.WT	67.66667
9_24_drySt-28F	Fecal.WT	80.50000
19_11_CeSt-28F	Cecal.Vdr-/-	52.16667
21_13_CeSt-28F	Cecal.Vdr-/-	55.00000
22_14_CeSt-28F	Cecal.Vdr-/-	59.00000
23_15_CeSt-28F	Cecal.Vdr-/-	60.87500
25_22_CeSt-28F	Cecal.WT	51.00000
26_23_CeSt-28F	Cecal.WT	112.85714
27_24_CeSt-28F	Cecal.WT	78.05882

Los códigos R siguientes conducen a la prueba Barlett de homogeneidad de varanzas:

```
# No corre
# bartlett.test(df_CH_G4, Group4)
```

La función nos da el valor K al cuadrado de las pruebas estadísticas y el valor p. Muestra que la hipótesis nula puede rechazarse al nivel del 5 %. Como alternativa, podemos comparar el K-cuadrado de Bartlett con el valor de las tablas de chi-cuadrado, utilizando el mismo nivel de alfa y grados de libertad en la función `qchisq()`. Si Chi-cuadrado > K-cuadrado de Bartlett, aceptamos la hipótesis nula H_0 (homogeneidad de varianzas), de lo contrario rechazamos la hipótesis nula.

```
qchisq(0.95, 1)
```

```
## [1] 3.841459
```

Como la Chi-cuadrado es menor que la K-cuadrado de Bartlett, rechazamos la hipótesis nula H_0 y concluimos que las varianzas no son iguales. Ahora utilizamos la prueba de Fligner-Killeen para comprobar la homocedasticidad. La sintaxis como a continuación es bastante similar.

```
fligner.test(df_CH_G4, Group4)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  df_CH_G4
## Fligner-Killeen:med chi-squared = 20.572, df = 1, p-value = 5.742e-06
```

Las conclusiones son similares a las de la prueba de Bartlett: las varianzas no son iguales. Sin embargo, a efectos de ilustración, procedemos a analizar los datos mediante ANOVA independientemente de los resultados de la prueba de homegeneidad de varianzas. Los siguientes códigos R se ajustan al modelo:

```
fit = lm(formula = value~Group4, data = df_CH_G)
```

Entonces analizamos el modelo de ANOVA

```
anova(fit)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Group4	3	1925.600	641.8668	2.192736	0.1417611
Residuals	12	3512.691	292.7242	NA	NA

O solo usamos el siguiente codigo: la función `aov()` anidade dentro de la función `summary()`.

```
summary(aov(value~Group4, data=df_CH_G))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Group4       3   1926   641.9   2.193  0.142
## Residuals   12   3513   292.7
```

Podemos imprimir el intercepto usando lo siguiente

```
aov_fit <- aov(value~Group4, data = df_CH_G)
summary(aov_fit, intercept=T)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## (Intercept)  1  83450   83450 285.080 9.97e-10 ***
## Group4       3   1926     642   2.193    0.142
## Residuals   12   3513     293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como el valor $p > 0.05$, aceptamos la hipótesis H_0: las cuatro medias no son diferentes. Podemos comprar el valor F con el valor tabulado de F:

```
qf(0.95, 12, 3)
```

```
## [1] 8.744641
```

Debido a que el valor tabulado de F es más grande que el computado valor F, aceptamos la H_0. Los resultados del ANOVA son un poco desordenados. Puede utilizar el paquete broom para obtener las tablas ordenadas y más informativas.

```
#install.packages("mmormt")
library("broom")
tidy(aov_fit)
```

term	df	sumsq	meansq	statistic	p.value
Group4	3	1925.600	641.8668	2.192736	0.1417611
Residuals	12	3512.691	292.7242	NA	NA

```
augment(aov_fit)
```

.rownames	value	Group4	.fitted	.resid	.hat	.sigma	.cooksd	.std.resid
5_15_drySt-28F	94.75000	Fecal.Vdr-/-	83.16645	11.583550	0.2000000	17.43812	0.0358109	0.7569503
20_12_CeSt-28F	59.80000	Cecal.Vdr-/-	57.36833	2.431667	0.2000000	17.85115	0.0015781	0.1589021
1_11_drySt-28F	77.00000	Fecal.Vdr-/-	83.16645	-6.166450	0.2000000	17.74865	0.0101485	-0.4029590
2_12_drySt-28F	103.27273	Fecal.Vdr-/-	83.16645	20.106277	0.2000000	16.53471	0.1078934	1.3138850
3_13_drySt-28F	85.66667	Fecal.Vdr-/-	83.16645	2.500217	0.2000000	17.85007	0.0016683	0.1633817
4_14_drySt-28F	55.14286	Fecal.Vdr-/-	83.16645	-28.023593	0.2000000	15.16886	0.2095941	-1.8312579
7_22_drySt-28F	62.75000	Fecal.WT	70.30556	-7.555556	0.3333333	17.65081	0.0365658	-0.5408573
8_23_drySt-28F	67.66667	Fecal.WT	70.30556	-2.638889	0.3333333	17.84337	0.0044605	-0.1889024
9_24_drySt-28F	80.50000	Fecal.WT	70.30556	10.194444	0.3333333	17.46893	0.0665687	0.7297597
19_11_CeSt-28F	52.16667	Cecal.Vdr-/-	57.36833	-5.201667	0.2000000	17.78372	0.0072213	-0.3399133
21_13_CeSt-28F	55.00000	Cecal.Vdr-/-	57.36833	-2.368333	0.2000000	17.85212	0.0014970	-0.1547635
22_14_CeSt-28F	59.00000	Cecal.Vdr-/-	57.36833	1.631667	0.2000000	17.86149	0.0007105	0.1066245
23_15_CeSt-28F	60.87500	Cecal.Vdr-/-	57.36833	3.506667	0.2000000	17.83082	0.0032819	0.2291502
25_22_CeSt-28F	51.00000	Cecal.WT	80.63866	-29.638655	0.3333333	14.12611	0.5626776	-2.1216552
26_23_CeSt-28F	112.85714	Cecal.WT	80.63866	32.218487	0.3333333	13.33364	0.6648948	2.3063300
27_24_CeSt-28F	78.05882	Cecal.WT	80.63866	-2.579832	0.3333333	17.84455	0.0042631	-0.1846748

```
glance(aov_fit)
```

logLik	AIC	BIC	deviance	nobs	r.squared
-65.83541	141.6708	145.5338	3512.691	16	0.3540819

4.2. Comparaciones múltiples Pareadas y de Tukey

Los resultados del ANOVA dan la prueba global de diferencia de grupos (en este caso, 4 grupos con combinación fecal, cecal, $Vdr^{-/-}$ y WT). Nuestro propósito es probar también cada diferencia de pares asociada a la riqueza de Chao 1. Los siguientes pasos sirven para ilustrar las capacidades de la prueba t por pares y las comparaciones múltiples ad hoc de Tukey en R. Vamos a ejecutar la prueba t por pares no ajustada para los cuatro grupos. La configuración por defecto en R para esta prueba es ajustar los valores p como post hoc usando el método Holm, así que para obtener valores p no ajustados, debe especificar `p.adjust = "none"`. El valor por defecto de R asume la homogeneidad de la varianza, por lo que no es necesario especificar `pool.sd = T`. Si sus datos tienen una varianza desigual, debe utilizar `pool.sd = F`.

```r
# Pruebas de diferencias de medias por pares
pairwise.t.test(df_CH_G$value, df_CH_G$Group4,
                p.adjust = "none", pool.sd = T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df_CH_G$value and df_CH_G$Group4
##
##              Cecal.Vdr-/- Fecal.Vdr-/- Cecal.WT
## Fecal.Vdr-/- 0.035        -            -
## Cecal.WT     0.087        0.843        -
## Fecal.WT     0.321        0.324        0.474
##
## P value adjustment method: none
```

Si no hacemos ningún ajuste a nuestros valores p, hay diferencias estadísticamente entre $Vdr^{-/-}$ fecal, $Vdr^{-/-}$ cecal, y marginalmente diferencias estadísticas entre WT cecal y $Vdr^{-/-}$ cecal. Estas diferencias se visualizan en el `boxplot`. Como observamos, la función `p.adjust()` está anidada dentro de la función `pairwise.t.test()` de pares. Esta es una función básica y muy útil de R. Puede utilizarse para controlar el error de tipo I de la familia. La función `p.adjust()` puede anidarse en otra función, o ser llamada de forma independiente. En una llamada independiente, la sintaxis se da a continuación:

$$p.adjust(p, method = p.adjust.methods, n = length(p))$$

donde, p = vector numérico de valores p, *método* = método de corrección, n = número de comparaciones, debe ser al menos $length(p)$. Los métodos de ajuste incluyen `c("bonferroni", "holm", "hochberg", "hommel", "BH", "BY", "fdr", "none")`. Donde "bonferroni" es la corrección de Bonferroni en la que los valores p se multiplican por el número de comparaciones; "holm", "hochberg", "hommel", "BH", "BY", "fdr" se refieren a Holm (1979), Hochberg (1988), Hommel (1988), Benjamini y Hochberg (1995) y Benjamini y Yekutieli (2001), y "fdr" es un alias de "BH". Son correcciones menos conservadoras.

```r
# Ajuste conservador de Bonferroni
pairwise.t.test(df_CH_G$value, df_CH_G$Group4,
                p.adjust = "bonferroni", pool.sd = T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df_CH_G$value and df_CH_G$Group4
##
##              Cecal.Vdr-/- Fecal.Vdr-/- Cecal.WT
## Fecal.Vdr-/- 0.21         -            -
## Cecal.WT     0.52         1.00         -
## Fecal.WT     1.00         1.00         1.00
##
## P value adjustment method: bonferroni
```

```r
# Método de Holm
pairwise.t.test(df_CH_G$value, df_CH_G$Group4,
                p.adjust = "holm", pool.sd = T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df_CH_G$value and df_CH_G$Group4
##
##              Cecal.Vdr-/- Fecal.Vdr-/- Cecal.WT
## Fecal.Vdr-/- 0.21         -            -
## Cecal.WT     0.44         1.00         -
## Fecal.WT     1.00         1.00         1.00
##
## P value adjustment method: holm
```

```
# Método de Benjamini & Hochberg(BH)
pairwise.t.test(df_CH_G$value, df_CH_G$Group4,
                p.adjust= "BH", pool.sd = T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df_CH_G$value and df_CH_G$Group4
##
##              Cecal.Vdr-/- Fecal.Vdr-/- Cecal.WT
## Fecal.Vdr-/- 0.21         -            -
## Cecal.WT     0.26         0.84         -
## Fecal.WT     0.49         0.49         0.57
##
## P value adjustment method: BH
```

```
# Método de Benjamini & Yekutieli
pairwise.t.test(df_CH_G$value, df_CH_G$Group4,
                p.adjust = "BY", pool.sd = T)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df_CH_G$value and df_CH_G$Group4
##
##              Cecal.Vdr-/- Fecal.Vdr-/- Cecal.WT
## Fecal.Vdr-/- 0.51         -            -
## Cecal.WT     0.64         1.00         -
## Fecal.WT     1.00         1.00         1.00
##
## P value adjustment method: BY
```

Los cuatro ajustes anteriores no ofrecen diferencias significativas en las comparaciones por pares. Los ajustes conservadores de Bonferroni y Benjamini & Yekutieli tienen los mayores valores p. Con el método de Benjamini y Hochberg ninguna de las comparaciones son significativas, pero sus valores p ajustados son menores. El método Benjamini & Hochberg es más potente en este caso. Tanto el método de Benjamini & Hochberg (BH) como el de Benjamini & Yekutieli (BY) son para ajustar la "tasa de falsos descubrimientos" (FDR). En realidad no es un verdadero control de error por familia. Los métodos de la Falsa Tasa de Descubrimiento encuentran los mismos resultados: todas las comparaciones por pares no presentan diferencias significativas. A continuación, vamos a mostrar el uso de la función `TukeyHSD()` para hacer comparaciones múltiples de

Tukey de medias y obtener sus intervalos de confianza. La forma de llamar a esta función es similar a la función `summary()`. Toma la variable del cálculo original de ANOVA original como uno de sus argumentos.

```
# Comparaciones multiple de medias Tukey
TukeyHSD(aov_fit, conf.level=.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Group4, data = df_CH_G)
##
## $Group4
##                                 diff        lwr      upr     p adj
## Fecal.Vdr-/--Cecal.Vdr-/-  25.798117  -6.327765 57.92400 0.1333624
## Cecal.WT-Cecal.Vdr-/-      23.270322 -13.825451 60.36609 0.2935302
## Fecal.WT-Cecal.Vdr-/-      12.937222 -24.158550 50.03299 0.7327590
## Cecal.WT-Fecal.Vdr-/-      -2.527795 -39.623567 34.56798 0.9969042
## Fecal.WT-Fecal.Vdr-/-     -12.860895 -49.956667 24.23488 0.7361596
## Fecal.WT-Cecal.WT         -10.333100 -51.807435 31.14123 0.8792446
```

```
# Gráfico
plot(TukeyHSD(aov(df_CH_G$value~df_CH_G$Group4), conf.level=.95))
```

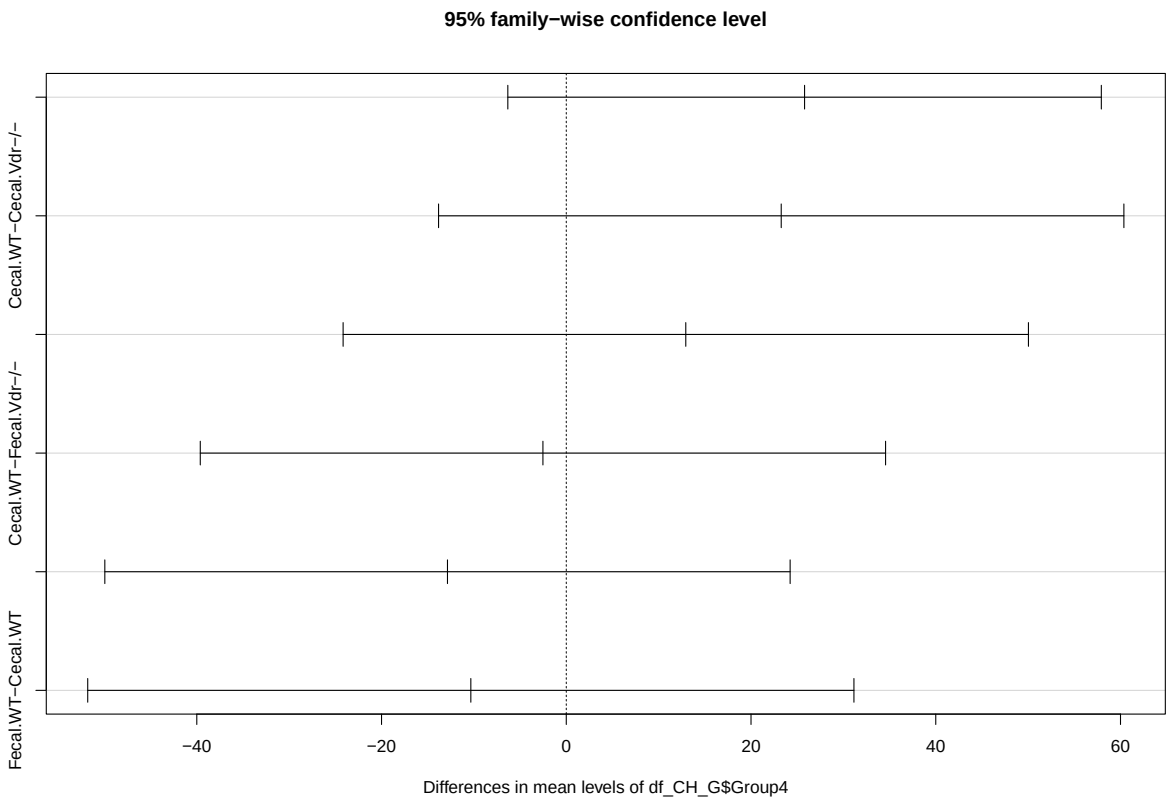

Figura 9: Gráfico de Tukey de comparaciones múltiples de medias y su intervalo de confianza en datos de ratón $Vdr^{-/-}$.

Este gráfico representa todas las pruebas posibles por pares y los valores p, así como los intervalos de confianza del 95 %. El nivel de confianza del 95 % por defecto puede cambiarse según su elección. Dado que todas las líneas de confianza cruzan 0, para este ejemplo, no hay términos significativamente diferentes tras el ajuste mediante comparaciones múltiples de Tukey.

5. Comparaciones entre más de dos grupos usando la prueba de Kruskal-Wallis

5.1. Prueba de Kruskal-Wallis

La prueba de Kruskal-Wallis o ANOVA unidireccional sobre rangos, llamada así por William Kruskal y W.Allen Wallis, es un método no paramétrico para comprobar si las muestras proceden de la misma distribución (Kruskal y Wallis 1952; Daniel 1990). *El equivalente paramétrico de la prueba de Kruskal-Wallis es el ANOVA de una vía.* La ampliación es la prueba U de Mann-Whitney a más de dos grupos. La hipótesis nula de la prueba de Kruskal-Wallis es que los *rangos medios de los grupos son los mismos.* A diferencia del ANOVA unidireccional análogo, la prueba no paramétrica de *Kruskal-Wallis no asume una distribución normal de los datos subyacentes.* Se ha utilizado ampliamente en investigación del microbioma. Por ejemplo, los datos del microbioma posteriores a la secuenciación no están distribuidos normalmente y contienen algunos valores atípicos importantes. Por lo tanto, es conveniente utilizar rangos en lugar de valores reales para evitar que las pruebas se vean afectadas por la presencia de valores atípicos o por una distribución no normal. La estadística de la prueba consta de cuatro pasos principales: + *Paso 1.* Clasificar todos los datos de todos los grupos juntos en una única serie en orden ascendente, es decir, clasificar los datos de 1 a N ignorando la pertenencia a un grupo. + *Paso 2.* Asignar los valores empatados promediando su posición en el ranking. + *Paso 3.* Sumar los diferentes rangos, por ejemplo, R_1 R_2 $R_3 \ldots$ para cada uno de los diferentes grupos. + *Paso 4.* Calcula la estadística de la prueba aplicando la siguiente fórmula:

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(n+1)$$

donde + H Prueba estadística de Kruskal-Wallis + n número total de mediiones en todas las muestras + n_i número de mediciones en la muestra de la población i + k número de poblaciones + R_i rango de sumas para la muestra i.

El estadístico de la prueba de Kruskal-Wallis es aproximadamente una distribución chi-cuadrado, con k - 1 grados de libertad si los valores de n_i son "grandes". La aproximación se acepta generalmente como adecuada cuando cada uno de los valores de n_i es mayor o igual a 5.

5.2. Comparación de diversidades entre grupos

La prueba de Kruskal-Wallis o ANOVA de una vía se realiza para comparar grupos múltiples cuyos datos no siguen una distribución normal. Esta prueba es similar a la prueba de suma de rangos de Wilcoxon para dos muestras. Primero utilizamos los datos de nuestros ratones $Vdr^{-/-}$ para ilustrar esta prueba.

```
library("dplyr")
Data <- mutate(df_CH_G, Group = factor(df_CH_G$Group4,
                                       levels = unique(df_CH_G$Group4)))
```

Estadística descriptiva

```
library("FSA")
Summarize(value ~ Group4, data = df_CH_G)
```

Group4	n	mean	sd	min	Q1	median	Q3	max
Cecal.Vdr-/-	5	57.36833	3.658497	52.16667	55.00000	59.00000	59.80000	60.8750
Fecal.Vdr-/-	5	83.16645	18.493505	55.14286	77.00000	85.66667	94.75000	103.2727
Cecal.WT	3	80.63866	31.009163	51.00000	64.52941	78.05882	95.45798	112.8571
Fecal.WT	3	70.30556	9.164520	62.75000	65.20833	67.66667	74.08333	80.5000

Generamos Histograma por grupo

```
# Gráficos individulaes en un panel de 2X2
library("lattice")
histogram(~ value|Group4, data=df_CH_G,layout=c(2,2))
```

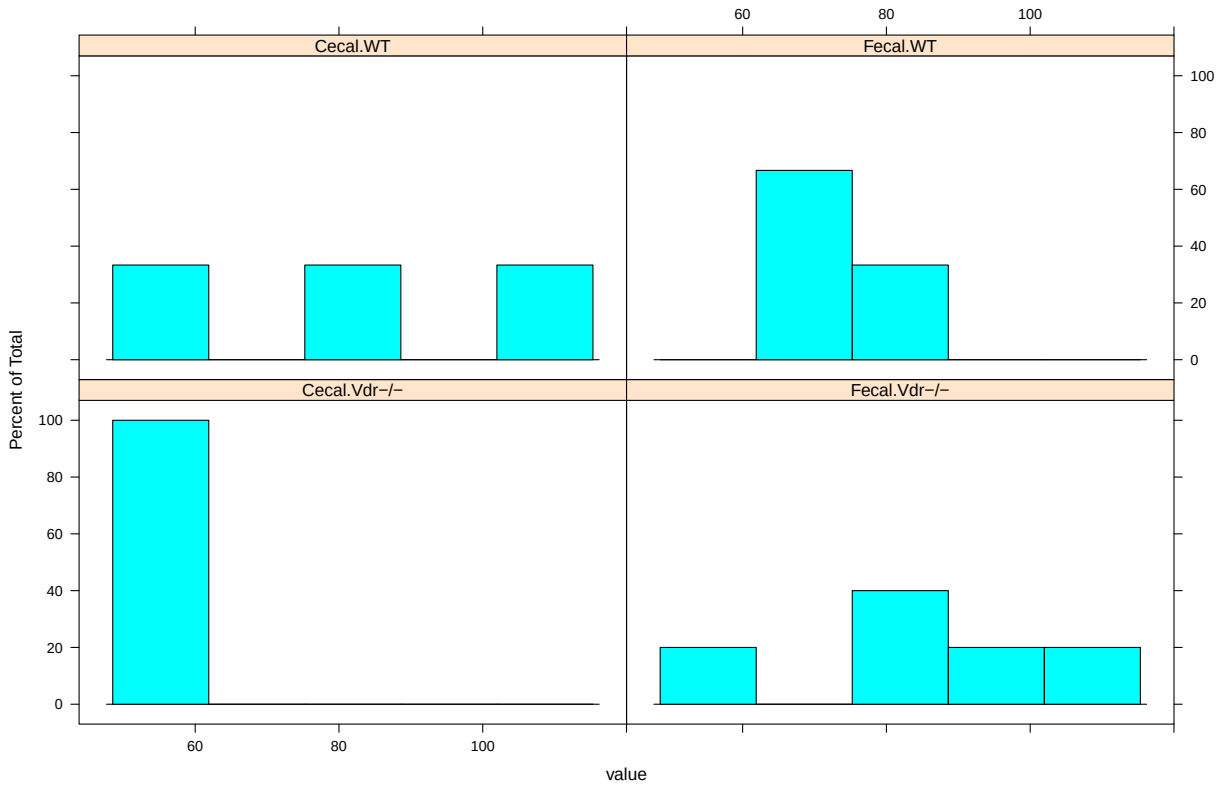

Figura 10: Distribuciones de los valores entre los grupos‘.

El histograma muestra que las distribuciones de los valores entre los grupos son diferentes en este caso. Ahora realizamos la prueba de Kruskal-Wallis para comparar las diferencias de medianas utilizando la función `kruskal.test()`.

```
# Prueba de Kruskal-Wallis de la riqueza de Chao1
kruskal.test(value ~ Group4, data = df_CH_G)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  value by Group4
## Kruskal-Wallis chi-squared = 5.2353, df = 3, p-value = 0.1554
```

El valor de la estadística de la prueba es 5,2 con un valor p superior a 0,05 y también es inferior a la tabulación chi-cuadrado:

```
qchisq(0.950, 3)
```

```
## [1] 7.814728
```

Por tanto, aceptamos la hipótesis nula H_0: las medianas de los 4 grupos son estadísticamente iguales a un nivel significativo del 5 %. Generalmente, se realiza un análisis post hoc para encontrar qué niveles de los grupos son diferentes entre sí si la prueba de Kruskal-Wallis es significativa. En este caso, la prueba de Kruskal-Wallis no es significativa. A modo de ilustración, realizamos dos pruebas post hoc: La prueba de Nemenyi y la prueba de Dunn. De forma similar al ANOVA, podemos elegir un método para ajustar los valores p para controlar la tasa de error familiar o para controlar la tasa de falsos descubrimientos. Al introducir `?p.adjust` en R o RStudio, aparece un enlace al documento "Adjust P-values for Multiple Comparisons". Puede consultar los detalles de los métodos de ajuste desde este enlace.

5.2.1. Prueba de Nemenyi para comparaciones múltiples

La prueba de Nemenyi se realiza mediante la función `NemenyiTest()` del paquete DescTools. Primero cargamos el paquete DescTools y llamamos a la función `NemenyiTest()`. El método para ajustar los valores p debe ser uno de "tukey", "chisq". En este caso elegimos el método Tukey

```
library("DescTools")
# Método de Tukey para ajustar valores p
Test_N <- NemenyiTest(x = df_CH_G$value,
                 g = df_CH_G$Group4,
                 dist="tukey")
Test_N
```

```
##
##  Nemenyi's test of multiple comparisons for independent samples (tukey)
##
##                             mean.rank.diff   pval
## Fecal.Vdr-/--Cecal.Vdr-/-      6.6000000 0.1254
## Cecal.WT-Cecal.Vdr-/-          4.7333333 0.5237
## Fecal.WT-Cecal.Vdr-/-          5.0666667 0.4636
## Cecal.WT-Fecal.Vdr-/-         -1.8666667 0.9501
## Fecal.WT-Fecal.Vdr-/-         -1.5333333 0.9713
## Fecal.WT-Cecal.WT              0.3333333 0.9998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La prueba de Nemenyi muestra que no hay diferencias significativas en el rango medio de la diversidad Chao 1 entre localizaciones y genotipos en las muestras fecales y cecales de vdr knockout utilizando el método de ajuste de Tukey. Sin embargo, cuando los grupos tienen números desiguales de observaciones, la prueba de Nemenyi es inadecuada, y la prueba de Dunn es apropiada (Zar 2010).

5.2.2. Prueba de Dunn para comparaciones múltiples

La prueba post hoc de Kruskal-Wallis más popular es la prueba de Dunn. Podemos realizar la prueba de Dunn utilizando la función `dunnTest()` del paquete FSA. A continuación, llamamos a la función función `dunnTest()` y utilizamos el método de Benjamini y Hochberg para ajustar los valores p.

```
library("FSA")
# "bh" sugiere el método de Benjamini y Hochberg para ajustar los valores p
Test_N <- dunnTest(df_CH_G$value ~ df_CH_G$Group4,
                   data = df_CH_G, method="bh")
Test_N
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Benjamini-Hochberg method.

##                     Comparison           Z    P.unadj     P.adj
## 1     Cecal.Vdr-/- - Cecal.WT -1.36136286 0.17339905 0.3467981
## 2 Cecal.Vdr-/- - Fecal.Vdr-/- -2.19189684 0.02838696 0.1703217
## 3     Cecal.WT - Fecal.Vdr-/- -0.53687549 0.59135362 0.8870304
## 4     Cecal.Vdr-/- - Fecal.WT -1.45723348 0.14505194 0.4351558
## 5         Cecal.WT - Fecal.WT -0.08574929 0.93166572 0.9316657
## 6     Fecal.Vdr-/- - Fecal.WT  0.44100487 0.65920947 0.7910514
```

La prueba de Dunn muestra que hay una diferencia estadísticamente significativa de la diversidad Chao 1 entre las muestras cecales y fecales de $Vdr^{-/-}$. Sin embargo, después de ajustar los valores comparación múltiple con el método Benjamini-Hochberg, no hay términos estadísticamente términos estadísticamente significativos entre las localizaciones y los genotipos de las muestras.

5.3. Encontrar taxones significativos entre los grupos

En esta sección, utilizamos la prueba de Kruskal-Wallis para ilustrar cómo encontrar taxones significativos entre los grupos. Supongamos que queremos saber si existen taxones significativos entre muestras de ratones $Vdr^{-/-}$ y WT de ubicaciones fecales y cecales. Usamos el test de prueba de Kruskal-Wallis para cada uno de los 248 taxones (bacterias) en el conjunto de datos. Primero, normalizamos los datos de abundancia y los convertimos en un marco de datos. Un método de normalización de método de normalización es utilizar la transformación logarítmica.

```
data <- log((abund_table+1)/(rowSums(abund_table)+dim(abund_table)[2]))
df <- as.data.frame(data)
```

Otro método de normalización es convertir las cuentas de abundancia en abudancia relativa

```
df <- as.data.frame(abund_table/rowSums(abund_table))
```

A continuación, utilice la función `kruskal.test()` y una función iterativa de R para realizar 248 pruebas (cada una para una bacteria). La función `kruskal.test()` tiene varios componentes componentes clave: + La prueba es un bucle para todos los taxones (columnas) con los códigos "`for (i in 1:dim(df)[2])`". - Para cada bucle, ejecuta la prueba de Kruskal-Wallis con los códigos "`KW_test <- kruskal.test(df[,i], g=Grupo4)`". - Los resultados se almacenan en un marco de datos con una fila por muestra y una columna por cada valor p de la prueba KW. - Informe el número de pruebas con la función cat "`cat(paste("Kruskal-Wallis test for ", names(df)[i], " ", i, /", dim(df)[2], "; p-value=", KW_test$p.value,"\n")", sep=)`".

```
KW_table <- data.frame()
for (i in 1:dim(df)[2]) {
  # Coremos prueba KW para cada bacteria
  KW_test <- kruskal.test(df[,i], g=df_CH_G$Group4)
  # Almacenamos los resultados en el dataframe
  KW_table <- rbind(KW_table,
                    data.frame(id=names(df)[i],
                    p.value=KW_test$p.value))
  # Reportamos el número de baacteria probada
  cat(paste("Kruskal-Wallis test for ",names(df)[i]," ", i, "/",
            dim(df)[2], "; p-value=", KW_test$p.value,"\n", sep=""))
}
```

```
## Kruskal-Wallis test for Tannerella 1/248; p-value=0.0052887551168933
## Kruskal-Wallis test for Lactococcus 2/248; p-value=0.353414882633892
## Kruskal-Wallis test for Lactobacillus 3/248; p-value=0.0668292640592179
## Kruskal-Wallis test for Lactobacillus::Lactococcus 4/248; p-value=0.476355094306889
## Kruskal-Wallis test for Parasutterella 5/248; p-value=0.231291869427266
## Kruskal-Wallis test for Helicobacter 6/248; p-value=0.0251057855497528
## Kruskal-Wallis test for Prevotella 7/248; p-value=0.00590090852956467
## Kruskal-Wallis test for Bacteroides 8/248; p-value=0.00397577720168958
## Kruskal-Wallis test for Barnesiella 9/248; p-value=0.0401317383052277
## Kruskal-Wallis test for Odoribacter 10/248; p-value=0.00443152726879245
## Kruskal-Wallis test for Eubacterium 11/248; p-value=0.57853361325349
## Kruskal-Wallis test for Allobaculum 12/248; p-value=0.727715895732644
## Kruskal-Wallis test for Roseburia 13/248; p-value=0.111897187856823
## Kruskal-Wallis test for Clostridium 14/248; p-value=0.0329375619788283
## Kruskal-Wallis test for Porphyromonas 15/248; p-value=0.0101839307809772
## Kruskal-Wallis test for Butyrivibrio 16/248; p-value=0.106570927868518
## Kruskal-Wallis test for Ruminococcus 17/248; p-value=0.026320704237654
## Kruskal-Wallis test for Acholeplasma 18/248; p-value=0.0888090977754463
## Kruskal-Wallis test for Alistipes 19/248; p-value=0.00393948262860657
## Kruskal-Wallis test for Clostridium::Coprococcus 20/248; p-value=0.227647113061301
## Kruskal-Wallis test for Eubacterium (Erysipelotrichaceae)::Eubacterium 21/248; p-value=0.37213164260
## Kruskal-Wallis test for Hydrogenoanaerobacterium 22/248; p-value=0.782882943019658
## Kruskal-Wallis test for Paraprevotella 23/248; p-value=0.00475416632504993
## Kruskal-Wallis test for Blautia 24/248; p-value=0.162098601397032
## Kruskal-Wallis test for Adlercreutzia::Asaccharobacter 25/248; p-value=0.725073232872234
## Kruskal-Wallis test for Coprococcus 26/248; p-value=0.125164693212525
## Kruskal-Wallis test for Parabacteroides 27/248; p-value=0.0616475690479629
## Kruskal-Wallis test for Eubacterium (Erysipelotrichaceae) 28/248; p-value=0.899201554861072
## Kruskal-Wallis test for Oscillibacter 29/248; p-value=0.0245787705312523
## Kruskal-Wallis test for Acinetobacter 30/248; p-value=0.531948371210495
## Kruskal-Wallis test for Alkalitalea 31/248; p-value=0.291046112229166
## Kruskal-Wallis test for Sporobacter 32/248; p-value=NaN
```

```
## Kruskal-Wallis test for Oscillospira 33/248; p-value=0.312344080172615
## Kruskal-Wallis test for Blautia::Clostridium 34/248; p-value=0.129409784600271
## Kruskal-Wallis test for Planctomyces 35/248; p-value=0.531948371210495
## Kruskal-Wallis test for Akkermansia 36/248; p-value=0.154574788263912
## Kruskal-Wallis test for Ruminofilibacter 37/248; p-value=0.310813822109325
## Kruskal-Wallis test for Pseudoflavonifractor 38/248; p-value=0.216820677518461
## Kruskal-Wallis test for Butyricimonas 39/248; p-value=0.00537686120875426
## Kruskal-Wallis test for Desulfovibrio 40/248; p-value=0.635967607335924
## Kruskal-Wallis test for Anaerotruncus 41/248; p-value=0.531948371210495
## Kruskal-Wallis test for Alistipes::Bacteroides 42/248; p-value=0.0175659448281556
## Kruskal-Wallis test for Turicibacter 43/248; p-value=0.225788700034188
## Kruskal-Wallis test for Mucispirillum 44/248; p-value=0.531948371210495
## Kruskal-Wallis test for Lachnospira 45/248; p-value=0.137113018301519
## Kruskal-Wallis test for Catabacter 46/248; p-value=0.531948371210495
## Kruskal-Wallis test for Desulfosporosinus 47/248; p-value=0.227647113061301
## Kruskal-Wallis test for Bacillus 48/248; p-value=0.156680145869787
## Kruskal-Wallis test for Sporanaerobacter 49/248; p-value=0.366846653612092
## Kruskal-Wallis test for Streptomyces 50/248; p-value=0.331271241004806
## Kruskal-Wallis test for Butyrivibrio::Clostridium 51/248; p-value=0.610481322254572
## Kruskal-Wallis test for Candidatus Arthromitus 52/248; p-value=0.15737830241587
## Kruskal-Wallis test for Enterorhabdus 53/248; p-value=0.616215837652105
## Kruskal-Wallis test for Kurthia 54/248; p-value=0.531948371210495
## Kruskal-Wallis test for Eggerthella 55/248; p-value=0.061890232897499
## Kruskal-Wallis test for Actinocorallia 56/248; p-value=0.0874598809266384
## Kruskal-Wallis test for Caldanaerocella 57/248; p-value=0.730716747994278
## Kruskal-Wallis test for Dorea 58/248; p-value=0.0692885571632859
## Kruskal-Wallis test for Streptococcus 59/248; p-value=0.982782073009701
## Kruskal-Wallis test for Adlercreutzia 60/248; p-value=0.443430240761201
## Kruskal-Wallis test for Paraeggerthella 61/248; p-value=0.0244868514394503
## Kruskal-Wallis test for Lachnobacterium 62/248; p-value=0.310813822109325
## Kruskal-Wallis test for TM7 (genus) 63/248; p-value=0.0338181771077889
## Kruskal-Wallis test for Clostridium::Anaerostipes 64/248; p-value=0.531948371210495
## Kruskal-Wallis test for Erysipelothrix 65/248; p-value=0.504786911847933
## Kruskal-Wallis test for Formosa 66/248; p-value=NaN
## Kruskal-Wallis test for Rikenella 67/248; p-value=0.0152837097075061
## Kruskal-Wallis test for Dysgonomonas 68/248; p-value=0.155137948838003
## Kruskal-Wallis test for Desulfitobacterium 69/248; p-value=0.381179725807674
## Kruskal-Wallis test for Oribacterium 70/248; p-value=0.730716747994278
## Kruskal-Wallis test for Syntrophococcus 71/248; p-value=NaN
## Kruskal-Wallis test for Ruminococcus::Clostridium 72/248; p-value=0.0635319064439386
## Kruskal-Wallis test for Rhodospirillum 73/248; p-value=0.531948371210495
## Kruskal-Wallis test for Pedobacter 74/248; p-value=0.0102158301994736
## Kruskal-Wallis test for Acetivibrio 75/248; p-value=0.253853716714176
## Kruskal-Wallis test for Clostridium::Eubacterium 76/248; p-value=0.56598353771052
## Kruskal-Wallis test for Limibacter 77/248; p-value=0.0328385403639491
## Kruskal-Wallis test for Clostridium::Ruminococcus 78/248; p-value=0.116642324140909
## Kruskal-Wallis test for Coprobacillus 79/248; p-value=0.723086145004186
## Kruskal-Wallis test for Cytophaga 80/248; p-value=0.0964007544917066
## Kruskal-Wallis test for Denitrobacterium 81/248; p-value=0.183234024503822
## Kruskal-Wallis test for Mycoplasma 82/248; p-value=0.305497320114101
## Kruskal-Wallis test for Roseburia::Clostridium 83/248; p-value=0.461687499237106
## Kruskal-Wallis test for Sporobacterium 84/248; p-value=0.506779636927482
## Kruskal-Wallis test for Eubacterium::Ruminococcus 85/248; p-value=NaN
## Kruskal-Wallis test for Pseudobutyrivibrio 86/248; p-value=0.731238219352963
```

```
## Kruskal-Wallis test for Robinsoniella 87/248; p-value=NaN
## Kruskal-Wallis test for Brevibacterium 88/248; p-value=0.170328740151205
## Kruskal-Wallis test for Blautia::Ruminococcus 89/248; p-value=0.168276852724954
## Kruskal-Wallis test for Pseudomonas 90/248; p-value=0.127405844323988
## Kruskal-Wallis test for Clostridium::Roseburia 91/248; p-value=NaN
## Kruskal-Wallis test for Desulfotomaculum 92/248; p-value=0.0813350148785695
## Kruskal-Wallis test for Clostridium (Erysipelotrichaceae) 93/248; p-value=0.716038214215504
## Kruskal-Wallis test for Olsenella 94/248; p-value=0.316836231291143
## Kruskal-Wallis test for Azospirillum 95/248; p-value=0.331591190693348
## Kruskal-Wallis test for Oxobacter 96/248; p-value=NaN
## Kruskal-Wallis test for Asaccharobacter::Adlercreutzia 97/248; p-value=0.531948371210495
## Kruskal-Wallis test for Desulfocurvus 98/248; p-value=0.531948371210495
## Kruskal-Wallis test for Anaerostipes 99/248; p-value=0.116642324140909
## Kruskal-Wallis test for Clostridium::Ruminococcus::Desulfotomaculum::Escherichia 100/248; p-value=0.1
## Kruskal-Wallis test for Halanaerobacter 101/248; p-value=0.331591190693348
## Kruskal-Wallis test for Slackia 102/248; p-value=0.210359383843896
## Kruskal-Wallis test for Anaplasma 103/248; p-value=0.227647113061301
## Kruskal-Wallis test for Syntrophomonas 104/248; p-value=0.227647113061301
## Kruskal-Wallis test for Clostridium::Blautia 105/248; p-value=0.0262113670860976
## Kruskal-Wallis test for Anaerosporobacter::Clostridium 106/248; p-value=NaN
## Kruskal-Wallis test for Clostridium::Bacteroides 107/248; p-value=0.0836269593937169
## Kruskal-Wallis test for Blautia::Lactonifactor 108/248; p-value=NaN
## Kruskal-Wallis test for Parabacteroides::Bacteroides 109/248; p-value=0.531948371210495
## Kruskal-Wallis test for Enterorhabdus::Adlercreutzia 110/248; p-value=NaN
## Kruskal-Wallis test for Flavobacterium 111/248; p-value=0.0415631075919192
## Kruskal-Wallis test for Clostridium::Desulfotomaculum 112/248; p-value=NaN
## Kruskal-Wallis test for Parasporobacterium 113/248; p-value=0.531948371210495
## Kruskal-Wallis test for Coprococcus::Clostridium 114/248; p-value=NaN
## Kruskal-Wallis test for Fusobacterium 115/248; p-value=0.39162517627109
## Kruskal-Wallis test for Ruminococcus::Escherichia 116/248; p-value=0.461398715042894
## Kruskal-Wallis test for Clostridium::Ruminococcus::Coprococcus 117/248; p-value=NaN
## Kruskal-Wallis test for Lactonifactor 118/248; p-value=NaN
## Kruskal-Wallis test for Haemophilus 119/248; p-value=0.156678124450806
## Kruskal-Wallis test for Sporichthya 120/248; p-value=0.513799266315117
## Kruskal-Wallis test for Cytophaga::Flavobacterium 121/248; p-value=0.461398715042894
## Kruskal-Wallis test for Clostridium::Dorea 122/248; p-value=0.0262113670860976
## Kruskal-Wallis test for Ruminococcus::Blautia 123/248; p-value=0.227647113061301
## Kruskal-Wallis test for Frigoribacterium 124/248; p-value=0.730716747994278
## Kruskal-Wallis test for Paenibacillus 125/248; p-value=0.99052573216016
## Kruskal-Wallis test for Johnsonella 126/248; p-value=0.531948371210495
## Kruskal-Wallis test for Pseudoflavonifractor::Clostridium 127/248; p-value=NaN
## Kruskal-Wallis test for Adlercreutzia::Enterorhabdus::Asaccharobacter 128/248; p-value=0.5319483712
## Kruskal-Wallis test for Nocardiopsis 129/248; p-value=0.327018183274628
## Kruskal-Wallis test for Pedobacter::Pseudomonas 130/248; p-value=NaN
## Kruskal-Wallis test for Flexibacter 131/248; p-value=0.200600894661472
## Kruskal-Wallis test for Catabacter::Ruminococcus 132/248; p-value=NaN
## Kruskal-Wallis test for Butyricimonas::Bacteroides 133/248; p-value=0.531948371210495
## Kruskal-Wallis test for Staphylococcus 134/248; p-value=0.396742675169042
## Kruskal-Wallis test for Alkalitalea::Prevotella 135/248; p-value=NaN
## Kruskal-Wallis test for Lachnospira::Anaerostipes 136/248; p-value=NaN
## Kruskal-Wallis test for Flavobacterium::Cytophaga 137/248; p-value=0.227647113061301
## Kruskal-Wallis test for Gelidibacter 138/248; p-value=0.00633832770736067
## Kruskal-Wallis test for Treponema 139/248; p-value=0.730716747994278
## Kruskal-Wallis test for Pontibacter 140/248; p-value=NaN
```

```
## Kruskal-Wallis test for Desulfuromonas 141/248; p-value=NaN
## Kruskal-Wallis test for Butyricicoccus 142/248; p-value=NaN
## Kruskal-Wallis test for Clostridium::Butyrivibrio 143/248; p-value=0.782882943019658
## Kruskal-Wallis test for Coprobacillus::Clostridium 144/248; p-value=0.531948371210495
## Kruskal-Wallis test for Porphyromonas::Prevotella 145/248; p-value=NaN
## Kruskal-Wallis test for Effluviibacter 146/248; p-value=0.324889697852914
## Kruskal-Wallis test for Caminicella 147/248; p-value=0.531948371210495
## Kruskal-Wallis test for Prevotella::Bacteroides 148/248; p-value=NaN
## Kruskal-Wallis test for Pseudoalteromonas 149/248; p-value=NaN
## Kruskal-Wallis test for Butyrivibrio::Blautia 150/248; p-value=0.227647113061301
## Kruskal-Wallis test for Lachnospira::Clostridium 151/248; p-value=0.143421037011305
## Kruskal-Wallis test for Alicyclobacillus 152/248; p-value=NaN
## Kruskal-Wallis test for Lachnospira::Robinsoniella 153/248; p-value=NaN
## Kruskal-Wallis test for Subdoligranulum 154/248; p-value=0.531948371210495
## Kruskal-Wallis test for Anaerofilum 155/248; p-value=0.227647113061301
## Kruskal-Wallis test for Sporosarcina 156/248; p-value=0.227647113061301
## Kruskal-Wallis test for Alkalitalea::Sphingobacterium 157/248; p-value=0.227647113061301
## Kruskal-Wallis test for Kopriimonas 158/248; p-value=0.531948371210495
## Kruskal-Wallis test for Olivibacter 159/248; p-value=NaN
## Kruskal-Wallis test for Lachnobacterium::Coprococcus 160/248; p-value=0.227647113061301
## Kruskal-Wallis test for Fluviicola 161/248; p-value=0.227647113061301
## Kruskal-Wallis test for Peptococcus 162/248; p-value=NaN
## Kruskal-Wallis test for Ruminococcus::Roseburia 163/248; p-value=0.227647113061301
## Kruskal-Wallis test for Marinilabilia 164/248; p-value=NaN
## Kruskal-Wallis test for Catonella 165/248; p-value=NaN
## Kruskal-Wallis test for Sphingobium 166/248; p-value=0.730716747994278
## Kruskal-Wallis test for Olsenella::Streptomyces 167/248; p-value=NaN
## Kruskal-Wallis test for Terasakiella 168/248; p-value=NaN
## Kruskal-Wallis test for Roseospirillum 169/248; p-value=0.116642324140909
## Kruskal-Wallis test for Clostridium::Ruminococcus::Blautia 170/248; p-value=0.227647113061301
## Kruskal-Wallis test for Lachnospira::Roseburia 171/248; p-value=NaN
## Kruskal-Wallis test for Thalassospira 172/248; p-value=NaN
## Kruskal-Wallis test for Eubacterium::Clostridium 173/248; p-value=0.730716747994278
## Kruskal-Wallis test for Allobaculum::Eubacterium 174/248; p-value=NaN
## Kruskal-Wallis test for Blautia::Lachnospira 175/248; p-value=NaN
## Kruskal-Wallis test for Cryptobacterium 176/248; p-value=NaN
## Kruskal-Wallis test for Atopobium 177/248; p-value=0.0888090977754463
## Kruskal-Wallis test for Eubacterium::Lachnospira 178/248; p-value=NaN
## Kruskal-Wallis test for Bacteroides::Alistipes 179/248; p-value=NaN
## Kruskal-Wallis test for Faecalibacterium 180/248; p-value=0.495049983682098
## Kruskal-Wallis test for Lachnospira::Coprococcus 181/248; p-value=0.227647113061301
## Kruskal-Wallis test for Microbacterium 182/248; p-value=0.531948371210495
## Kruskal-Wallis test for Zhangella::Stella 183/248; p-value=0.531948371210495
## Kruskal-Wallis test for Paludibacter 184/248; p-value=NaN
## Kruskal-Wallis test for Butyrivibrio::Ruminococcus 185/248; p-value=0.324889697852914
## Kruskal-Wallis test for Bacteroides::Lactobacillus 186/248; p-value=NaN
## Kruskal-Wallis test for Prevotella::Flavobacterium 187/248; p-value=0.227647113061301
## Kruskal-Wallis test for Ruminococcus::Dorea 188/248; p-value=NaN
## Kruskal-Wallis test for Slackia::Asaccharobacter::Adlercreutzia 189/248; p-value=0.531948371210495
## Kruskal-Wallis test for Caldicellulosiruptor 190/248; p-value=0.637778389635797
## Kruskal-Wallis test for Kordia 191/248; p-value=NaN
## Kruskal-Wallis test for Bacteroides::Lactobacillus::Lachnospira 192/248; p-value=0.531948371210495
## Kruskal-Wallis test for Sphingobacterium 193/248; p-value=NaN
## Kruskal-Wallis test for Anaeroplasma 194/248; p-value=0.531948371210495
```

```
## Kruskal-Wallis test for Atopostipes 195/248; p-value=0.531948371210495
## Kruskal-Wallis test for Enterococcus 196/248; p-value=NaN
## Kruskal-Wallis test for Insolitispirillum 197/248; p-value=NaN
## Kruskal-Wallis test for Clostridium::Acetivibrio 198/248; p-value=NaN
## Kruskal-Wallis test for Plantactinospora 199/248; p-value=NaN
## Kruskal-Wallis test for Stappia 200/248; p-value=NaN
## Kruskal-Wallis test for Anaplasma::Clostridium 201/248; p-value=0.227647113061301
## Kruskal-Wallis test for Clostridium (Erysipelotrichaceae)::Clostridium 202/248; p-value=NaN
## Kruskal-Wallis test for Algoriphagus 203/248; p-value=NaN
## Kruskal-Wallis test for Dorea::Ruminococcus 204/248; p-value=0.227647113061301
## Kruskal-Wallis test for Roseburia::Lachnospira 205/248; p-value=0.531948371210495
## Kruskal-Wallis test for Rhizobium 206/248; p-value=NaN
## Kruskal-Wallis test for Anaerofustis 207/248; p-value=NaN
## Kruskal-Wallis test for Echinicola 208/248; p-value=NaN
## Kruskal-Wallis test for Anaerostipes::Clostridium 209/248; p-value=NaN
## Kruskal-Wallis test for Lachnospira::Blautia 210/248; p-value=0.531948371210495
## Kruskal-Wallis test for Aestuariimicrobium 211/248; p-value=NaN
## Kruskal-Wallis test for Gelidibacter::Flavobacterium 212/248; p-value=NaN
## Kruskal-Wallis test for Helicobacter::Flexispira 213/248; p-value=0.531948371210495
## Kruskal-Wallis test for Eubacterium (Erysipelotrichaceae)::Paenibacillus::Eubacterium 214/248; p-valu
## Kruskal-Wallis test for Pelospora 215/248; p-value=0.227647113061301
## Kruskal-Wallis test for Stella 216/248; p-value=NaN
## Kruskal-Wallis test for Methylobacterium 217/248; p-value=NaN
## Kruskal-Wallis test for Rickettsia 218/248; p-value=0.531948371210495
## Kruskal-Wallis test for Porphyromonas::Flavobacterium 219/248; p-value=NaN
## Kruskal-Wallis test for Adlercreutzia::Enterorhabdus 220/248; p-value=NaN
## Kruskal-Wallis test for Ruminococcus::Escherichia::Parabacteroides 221/248; p-value=0.53194837121049
## Kruskal-Wallis test for Limibacter::Bacteroides 222/248; p-value=NaN
## Kruskal-Wallis test for Paraprevotella::Prevotella 223/248; p-value=NaN
## Kruskal-Wallis test for Janthinobacterium::Zoogloea::Duganella 224/248; p-value=NaN
## Kruskal-Wallis test for Flavobacterium::Gelidibacter 225/248; p-value=NaN
## Kruskal-Wallis test for Barnesiella::Bacteroides 226/248; p-value=NaN
## Kruskal-Wallis test for Porphyromonas::Gelidibacter 227/248; p-value=NaN
## Kruskal-Wallis test for Bacilloplasma 228/248; p-value=NaN
## Kruskal-Wallis test for Natronincola 229/248; p-value=NaN
## Kruskal-Wallis test for Lumbricincola 230/248; p-value=NaN
## Kruskal-Wallis test for Desulfotomaculum::Clostridium 231/248; p-value=0.227647113061301
## Kruskal-Wallis test for Roseburia::Ruminococcus 232/248; p-value=NaN
## Kruskal-Wallis test for Bifidobacterium 233/248; p-value=NaN
## Kruskal-Wallis test for Streptacidiphilus 234/248; p-value=NaN
## Kruskal-Wallis test for Butyrivibrio::Pseudobutyrivibrio 235/248; p-value=0.531948371210495
## Kruskal-Wallis test for Aeromicrobium 236/248; p-value=NaN
## Kruskal-Wallis test for Proteus 237/248; p-value=NaN
## Kruskal-Wallis test for Sporobacterium::Clostridium 238/248; p-value=NaN
## Kruskal-Wallis test for Butyrivibrio::Roseburia 239/248; p-value=NaN
## Kruskal-Wallis test for Luteococcus 240/248; p-value=NaN
## Kruskal-Wallis test for Clostridium::Blautia::Desulfotomaculum 241/248; p-value=0.531948371210495
## Kruskal-Wallis test for Lachnospira::Ruminococcus 242/248; p-value=0.531948371210495
## Kruskal-Wallis test for Ornithinimicrobium 243/248; p-value=NaN
## Kruskal-Wallis test for Persicivirga 244/248; p-value=NaN
## Kruskal-Wallis test for Lachnospira::Ruminococcus::Escherichia 245/248; p-value=NaN
## Kruskal-Wallis test for Anaerophaga 246/248; p-value=0.227647113061301
## Kruskal-Wallis test for Dysgonomonas::Flavobacterium 247/248; p-value=NaN
## Kruskal-Wallis test for Bizionia 248/248; p-value=0.531948371210495
```

Revisamos la tabla del data.frame para asegurarnos que la función trabaja

```
# Revisamos el data.frame
head(KW_table)
```

id	p.value
Tannerella	0.0052888
Lactococcus	0.3534149
Lactobacillus	0.0668293
Lactobacillus::Lactococcus	0.4763551
Parasutterella	0.2312919
Helicobacter	0.0251058

5.4. Pruebas múltiples y valor E, FWER y FDR

En la literatura existen varios tipos diferentes de correcciones de pruebas múltiples. Entre ellas, la corrección de Bonferroni es la más conservadora. La corrección consiste simplemente en *dividir el alfa por el número de pruebas*. Aquí pesentamos las correcciones generales de las pruebas múltiples: *Valor E*, *Tasa de error por familia (FWER)* y *FDR*.

5.4.1. Valor E

El valor E es el número esperado de falsos positivos por azar cuando se hacen múltiples pruebas. Simplemente se puede multiplicar el valor p por el número de taxones en que se realiza la prueba para obtenerlo: Valor E = valor p epor el número de pruebas. Tenga en cuenta que en el valor E, la corrección de base es utilizar el alfa original, el valor p de las pruebas en lugar del valor p nominal.

```
KW_table$E.value <- KW_table$p.value * dim(KW_table)[1]
KW_table$E.value
```

```
##   [1]   1.3116113  87.6468909  16.5736575 118.1360634  57.3603836   6.2262348
##   [7]   1.4634253   0.9859927   9.9526711   1.0990188 143.4763361 180.4735421
##  [13]  27.7505026   8.1685154   2.5256148  26.4295901   6.5275347  22.0246562
##  [19]   0.9769917  56.4564840  92.2886474 194.1549699   1.1790332  40.2004531
##  [25] 179.8181618  31.0408439  15.2885971 223.0019856   6.0955351 131.9231961
##  [31]  72.1794358         NaN  77.4613319  32.0936266 131.9231961  38.3345475
##  [37]  77.0818279  53.7715280   1.3334616 157.7199666 131.9231961   4.3563543
##  [43]  55.9955976 131.9231961  34.0040285 131.9231961  56.4564840  38.8566762
##  [49]  90.9779701  82.1552678 151.3993679  39.0298190 152.8215277 131.9231961
##  [55]  15.3487778  21.6900505 181.2177535  17.1835622 243.7299541 109.9706997
##  [61]   6.0727392  77.0818279   8.3869079 131.9231961 125.1871541         NaN
##  [67]   3.7903600  38.4742113  94.5325720 181.2177535         NaN  15.7559128
##  [73] 131.9231961   2.5335259  62.9557217 140.3639174   8.1439580  28.9272964
##  [79] 179.3253640  23.9073871  45.4420381  75.7633354 114.4984998 125.6813500
##  [85]         NaN 181.3470784         NaN  42.2415276  41.7326595  31.5966494
##  [91]         NaN  20.1710837 177.5774771  78.5753854  82.2346153         NaN
##  [97] 131.9231961 131.9231961  28.9272964  56.4564840  82.2346153  52.1691272
## [103]  56.4564840  56.4564840   6.5004190         NaN  20.7394859         NaN
## [109] 131.9231961         NaN  10.3076507         NaN 131.9231961         NaN
## [115]  97.1230437 114.4268813         NaN         NaN  38.8561749 127.4222180
```

```
## [121] 114.4268813   6.5004190  56.4564840 181.2177535 245.6503816 131.9231961
## [127]         NaN 131.9231961  81.1005095         NaN  49.7490219         NaN
## [133] 131.9231961  98.3921834         NaN         NaN  56.4564840   1.5719053
## [139] 181.2177535         NaN         NaN         NaN 194.1549699 131.9231961
## [145]         NaN  80.5726451 131.9231961         NaN         NaN  56.4564840
## [151]  35.5684172         NaN         NaN 131.9231961  56.4564840  56.4564840
## [157]  56.4564840 131.9231961         NaN  56.4564840  56.4564840         NaN
## [163]  56.4564840         NaN         NaN 181.2177535         NaN         NaN
## [169]  28.9272964  56.4564840         NaN         NaN 181.2177535         NaN
## [175]         NaN         NaN  22.0246562         NaN         NaN 122.7723960
## [181]  56.4564840 131.9231961 131.9231961         NaN  80.5726451         NaN
## [187]  56.4564840         NaN 131.9231961 158.1690406         NaN 131.9231961
## [193]         NaN 131.9231961 131.9231961         NaN         NaN         NaN
## [199]         NaN         NaN  56.4564840         NaN         NaN  56.4564840
## [205] 131.9231961         NaN         NaN         NaN         NaN 131.9231961
## [211]         NaN         NaN 131.9231961         NaN  56.4564840         NaN
## [217]         NaN 131.9231961         NaN         NaN 131.9231961         NaN
## [223]         NaN         NaN         NaN         NaN         NaN         NaN
## [229]         NaN         NaN  56.4564840         NaN         NaN         NaN
## [235] 131.9231961         NaN         NaN         NaN         NaN         NaN
## [241] 131.9231961 131.9231961         NaN         NaN         NaN  56.4564840
## [247]         NaN 131.9231961
```

Dado que el valor E no es más que multiplicar el valor p por el número de pruebas, puede ser mayor que 1. Si hay muchos taxones en el marco de datos para las pruebas, este método de corrección no es fácil de encontrar los taxones significativos. Los taxones significativos son aquellos para los que el valor E es mucho menor que 1. Los siguientes códigos se utilizan para comprobar si o no los valores E se añaden al marco de datos resultante:

```
# Revisamos el data.frame de resultados
head(KW_table)
```

id	p.value	E.value
Tannerella	0.0052888	1.311611
Lactococcus	0.3534149	87.646891
Lactobacillus	0.0668293	16.573657
Lactobacillus::Lactococcus	0.4763551	118.136063
Parasutterella	0.2312919	57.360384
Helicobacter	0.0251058	6.226235

5.4.2. FWER

La FWER es la probabilidad de obtener al menos un falso positivo (error de tipo I). En En otras palabras, es la probabilidad de no rechazar la hipótesis nula H_0: no hay diferencias entre los grupos mientras se realizan pruebas múltiples. La fórmula es dada por:

$$FWER = 1 - (1 - p - value)^T$$

donde, T = el número de pruebas. Para evitar los errores de redondeo causados por el cálculo directo utilizando la fórmula anterior, en R es mejor calcular la FWER con una prueba de distribución binomial de cola derecha.

```
KW_table$FWER <- pbinom(q=0, p=KW_table$p.value, size=dim(KW_table)[1],
                        lower.tail=FALSE)
```

Revisamos el dataframe para ver si FWER son añadidos al data.frame de resultados

```
head(KW_table)
```

id	p.value	E.value	FWER
Tannerella	0.0052888	1.311611	0.7315504
Lactococcus	0.3534149	87.646891	1.0000000
Lactobacillus	0.0668293	16.573657	1.0000000
Lactobacillus::Lactococcus	0.4763551	118.136063	1.0000000
Parasutterella	0.2312919	57.360384	1.0000000
Helicobacter	0.0251058	6.226235	0.9981742

5.4.3. FDR

Por último, pero no por ello menos importante, está el FDR. Benjamini y Hochberg (1995) definieron la tasa de falsos descubrimientos de la siguiente manera: *FDR = proporción esperada de rechazos erróneos entre todos los rechazos*. En este caso, el FDR es la proporción de falsos positivos entre los taxones aceptados como positivos cuando se hacen múltiples pruebas. La corrección de Benjamini-Hochberg consiste en los siguientes pasos. En primer lugar, se ordenan los valores p de menor a mayor y se hace un rango (1, 2, 3,. . . , k,. . . ,T);

```
# Ordenamos los valores p de menor a mayor
KW_table <- KW_table[order(KW_table$p.value, decreasing=FALSE), ]
head(KW_table)
```

	id	p.value	E.value	FWER
19	Alistipes	0.0039395	0.9769917	0.6242838
8	Bacteroides	0.0039758	0.9859927	0.6276638
10	Odoribacter	0.0044315	1.0990188	0.6676149
23	Paraprevotella	0.0047542	1.1790332	0.6932876
1	Tannerella	0.0052888	1.3116113	0.7315504
39	Butyricimonas	0.0053769	1.3334616	0.7373832

Ahora calculamos el valor q usando la siguiente ecuación

$$q - value = p - value * T/k$$

```
# Calculamos el valor q
KW_table$q.value.factor <- dim(KW_table)[1] / 1:dim(KW_table)[1]
head(KW_table$q.value.factor)
```

```
## [1] 248.00000 124.00000  82.66667  62.00000  49.60000  41.33333
```

```
KW_table$q.value <- KW_table$p.value * KW_table$q.value.factor
head(KW_table$q.value)
```

```
## [1] 0.9769917 0.4929964 0.3663396 0.2947583 0.2623223 0.2222436
```

```
# Revisamos si el valor q es añadido a la tabla de resultados
head(KW_table)
```

	id	p.value	E.value	FWER	q.value.factor	q.value
19	Alistipes	0.0039395	0.9769917	0.6242838	248.00000	0.9769917
8	Bacteroides	0.0039758	0.9859927	0.6276638	124.00000	0.4929964
10	Odoribacter	0.0044315	1.0990188	0.6676149	82.66667	0.3663396
23	Paraprevotella	0.0047542	1.1790332	0.6932876	62.00000	0.2947583
1	Tannerella	0.0052888	1.3116113	0.7315504	49.60000	0.2623223
39	Butyricimonas	0.0053769	1.3334616	0.7373832	41.33333	0.2222436

A continuación, especifique el FDR objetivo e identifique el último elemento de la lista clasificada que tenga un valor q igual o menor que el alfa especificado utilizando los siguientes códigos:

```
# Ajustamos valor alfa
KW_alpha <- 0.05

# Identificamos el último elemento de la lista ranqueada con un valor q  =< alfa
last.significant.item <- max(which(KW_table$q.value <= KW_alpha))
last.significant.item
```

```
## [1] -Inf
```

En nuestro caso, no hay ningún valor q menor o igual al alfa especificado, por lo que el programa devuelve un infinito negativo. Por último, muestra la tabla del marco de resultados y los taxones elegidos:

```
# Mostramos algunos resultados
selected <- 1:5
print(KW_table[selected,])
```

```
##                id     p.value   E.value      FWER q.value.factor   q.value
## 19      Alistipes 0.003939483 0.9769917 0.6242838      248.00000 0.9769917
## 8     Bacteroides 0.003975777 0.9859927 0.6276638      124.00000 0.4929964
## 10    Odoribacter 0.004431527 1.0990188 0.6676149       82.66667 0.3663396
## 23 Paraprevotella 0.004754166 1.1790332 0.6932876       62.00000 0.2947583
## 1      Tannerella 0.005288755 1.3116113 0.7315504       49.60000 0.2623223
```

```
diff.taxa.factor <- KW_table$id[selected]
diff.taxa <- as.vector(diff.taxa.factor)
diff.taxa
```

```
## [1] "Alistipes"      "Bacteroides"    "Odoribacter"    "Paraprevotella"
## [5] "Tannerella"
```

Debido a que en este caso no hay ningún valor q menor o igual al alfa especificado = 0,05, los 5 taxones mostrados arriba no se basan en el FDR. Son los taxones elegidos con valores p más pequeños. La corrección de Benjamini-Hochberg es menos estricta que las otras correcciones de pruebas múltiples presentadas anteriormente y, por lo tanto, tiene una mayor sensibilidad. El FDR se utiliza ampliamente en el microbioma (Le Chatelier et al. 2013; Ballou et al. 2016) y en otros campos de estudio (Jungquist et al. 2010) y en muchas funciones de R.

6. Resumen

En este capítulo, presentamos una variedad de métodos comunes y clásicos en todos los campos de investigación. Algunos de ellos se aplican ampliamente en los estudios del microbioma. Ilustramos estos métodos para analizar los datos del microbioma con una implementación paso a paso en el sistema R. Los lectores pueden utilizar los códigos de R y las explicaciones proporcionadas en este capítulo para analizar sus propios datos del microbioma. Nos centramos en las pruebas de hipótesis para los datos del microbioma comunitario univariante.