Parcial

Maestría en Ciencia de Datos, ITAM

12/10/2020

Contents

Equipo:

- HM, LUZ AURORA
- SC, ITA-ANDEHUI
- ZC, JOSE LUIS ROBERTO

"Fundamentos de Estadística con Remuestreo"

Prof. Teresa Ortiz Mancera

Entrega: 12 de octubre antes de las 15:00 horas, por correo electrónico con el título fundamentos-parcial, un solo documento por equipo. Cada día de retraso se penaliza con un punto.

Instrucciones:

- Tus respuestas deben ser claras y debes explicar los resultados, incluye también tus procedimientos/código de manera ordenada, y el código comentado.
- Se evaluará la presentación de resultados (calidad de las gráficas, tablas, ...), revisa la sección de visualización en las notas.

- Se puede realizar individual o en grupos de 2 ó 3.
- Si tienes preguntas puedes escribirlas en el canal de slack #examen-parcial, será el único medio para resolver dudas del examen.

Análisis exporatorio

1. NBER TH

Considera la tabla de datos dada en tabla_nber_th.csv. Es la tabla de frecuencias de 4353 pilotos de la segunda guerra mundial, donde los individuos están clasificados según:

- Tipo de ocupación en 1969. SE significa self-employed.
- Resultados de estudios de aptitud de 1943 (A5 es el nivel más alto, y A1 el más bajo).
- Nivel de educación en 1969 (esta incluye años de escuela en 1943 más estudios posteriores a la guerra). E4 es el nivel más alto y E1 es el nivel más bajo.

1. ¿Qué relación existe entre aptitud (1943) y el nivel de educación (1969)? Describe esta relación usando tablas de porcentajes y de índices (o perfiles).

```
## Import Data
data_pilotos <- read_csv('tabla_nber_th.csv')
```

```
## Parsed with column specification:
## cols(
##   Ocup_group = col_character(),
##   Education = col_character(),
##   Aptitude = col_character(),
##   Freq = col_double()
## )
```

```
## Create % table
tblprcnt_pilots <- data_pilotos %>% dplyr::select(-Ocup_group) %>%
  group_by(Aptitude,Education) %>% tally(Freq) %>%
  group_by(Aptitude) %>% mutate(prcnt = 100*n/sum(n)) %>%
  dplyr::select(-n) %>% spread(key = Education, value = prcnt) %>%
  mutate(across(where(is.numeric), round, 3))

## Format table
kbl(tblprcnt_pilots, caption = "Distr Porcentual, Nivel Aptitud vs Nivel Educación.",
    align = 'c') %>%
  kable_styling(bootstrap_options = c("striped", "scale_down"), full_width = F)
```

Si bien esta tabla aún no nos permite visualizar de forma precisa la relación que existe entre estas dos variables, ya se percibe que existe una relación directa entre nivel de aptitud baja y nivel de educación bajo. Ahora construimos una tabla de perfiles que nos permitan establecer una apreciación de proporcionalidad.

De la table anterior y de los datos, sabemos que el número de categorías en Aptitude son 4, por lo que omitimos declarar esta variable. Realizamos los cálculos y generemos la tabla comparativa de `Aptitude vs. Education`, a través de un % relativo a la distancia de cada componente a al promedio del peso de su categoría.

Table 1: Distr Porcentual, Nivel Aptitud vs Nivel Educación.

Aptitude	E1	E2	E3	E4
A1	33.385	32.298	21.429	12.888
A2	26.839	27.221	27.125	18.816
A3	23.411	24.292	28.068	24.229
A4	17.582	24.176	32.692	25.549
A5	12.754	15.362	37.971	33.913

Table 2: Tabla de Porcentajes Relativos, Nivel Aptitud vs Nivel Educación.

Aptitude	aptitude_distrib	E1	E2	E3	E4
A1	14.8	39.7	26.9	-24.6	-42.0
A2	24.0	12.5	7.1	-4.4	-15.2
A3	36.6	-2.3	-4.8	-1.5	8.6
A4	16.7	-26.1	-4.7	15.5	15.3
A5	7.9	-46.6	-39.6	33.7	52.6

```
## Create relative % distrib
tblprfl_pilots <- data_pilotos %>% dplyr::select(-Ocup_group) %>%
  group_by(Education, Aptitude) %>% tally(Freq) %>%
  group_by(Education) %>% mutate(prcnt = n/sum(n)) %>%
  group_by(Aptitude) %>%
  mutate(avg_prcnt = sum(prcnt)*100/4) %>%
  mutate(perf_val =  (100*(prcnt/(avg_prcnt/100)-1))) %>%
  dplyr::select(Education, Aptitude, perf_val, aptitude_distrib = avg_prcnt) %>%
  spread(key = Education, value = perf_val) %>%
  mutate(across(where(is.numeric), round, 1))

## Format table
kbl(tblprfl_pilots, caption = "Tabla de Porcentajes Relativos, Nivel Aptitud vs Nivel Educación.",
    align = 'c') %>%
  kable_styling(bootstrap_options = c("striped", "scale_down"), full_width = F)
```

La columna `aptitude _distrib` nos muestra el tamaño de cada nivel (siendo 5 el más alto). La mayor parte de la población de pilotos muestra que tiene un nivel medio de aptitudes, casi el 40% de los pilotos tenían un nivel bajo de aptitud y casi el 25% tenía un nivel medio-alto y alto de aptitude. Ahora anlizaremos los niveles bajo, medio y alto de aptitud (1,3,5) y su relación con el nivel de educación:

- De los pilotos con un nivel bajo de aptitud, se tiene que una proporción cercana al 40% tiene un nivel bajo de educación en comparación con el promedio en esta categoría.
- Para el nivel medio de aptitud, se observa que no existen variaciones significativas en los niveles de educación en relación al promedio de esta categoría, lo que implica una cierta uniformidad en la distribución para cada nivel; sólo el nivel alto de educación es ligeramente mayor.
- Justo se observa el efecto contario que en el nivel bajo de aptitude. Los pilotos con un nivel alto de aptitude presentan un nivel alto de educación a una proporción casi 50% por encima que el promedio de los niveles de edución de esta categoría (alta aptitude).

```r
## Se transforma la tabla a un formato largo y se direcciona a
## ggplot para graficar el resultado de los hallazgos.

graph_tblprfl_pilots <- tblprfl_pilots %>% dplyr::select(-aptitude_distrib) %>%
  gather(2:5, key = "Education", value = "val_perf") %>%
  ggplot(aes(x = Aptitude, xend = Aptitude, y = val_perf,
                          yend = 0, group = Education)) +
  geom_point(col="steelblue", size = 3) + geom_segment(col="black") +
  facet_wrap(~Education) + geom_hline(yintercept = 0 , colour = "gray") +
  coord_flip() +
  labs(title ="Aptitud vs Nivel educativo", x = "Aptitud", y = "Perfil") +
  theme_igray()

graph_tblprfl_pilots
```

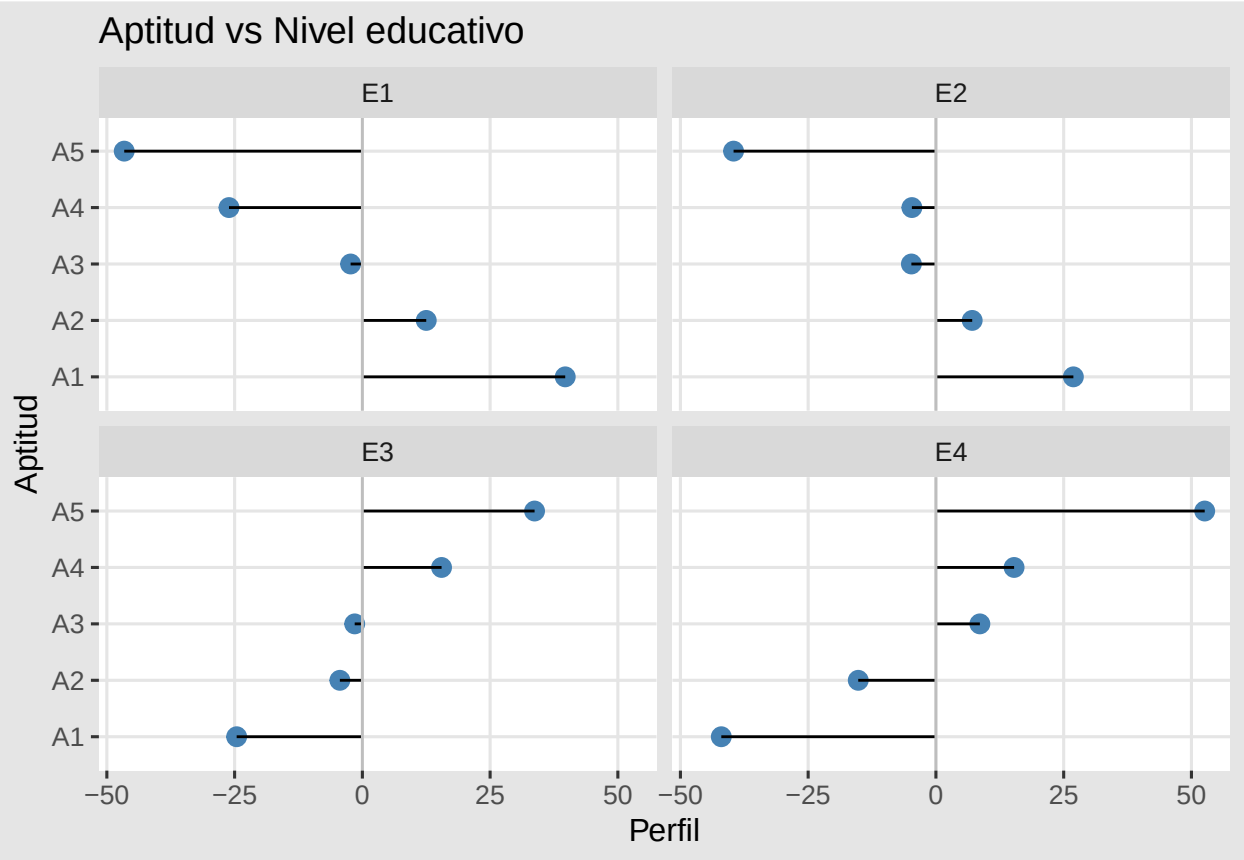


Gráfico de Porcentajes, Nivel de Aptitud vs Nivel de Educación.
Este gráfico nos presenta una síntesis muy clara y rápidamente abstraíble de las conclusiones del análisis anterior:

- El nivel de aptitude medio (`A3`) se observa muy cercano a la barra central (promedio), sin variaciones importantes.

- En el caso de un nivel de aptitude alto, observemos como sus barras indicadoras son grandes y contrarias: negativa para un nivel bajo de educación E1 y positiva para un nivel alto `E5`.

2. Describe la relación entre nivel de educación y ocupación ¿Qué concluyes de esta relación? ¿Cuáles dirías

que son ocupaciones asociadas a educación alta y cuáles a educación baja?

- Utilizamos el gráfico de perfiles para realizar nuestro análisis. Podemos observar que los niveles de educación superiores están relacionados con los pilotos que eran maestros y profesionistas.
- Los pilotos que eran empleados con salario o tenían un negocio se observa que tienen proporciones muy similares entre los distintos niveles de educación, excepto los pilotos con nivel medio bajo de educación (`E2`) que se observa una inclicación por tener un negocio.
- No se observa claramente una definición de ocupación asociada a un nivel de eduación bajo (`E1`, `E2`).

```
## Create Table
tbl2prfl_pilots <- data_pilotos %>% dplyr::select(-Aptitude) %>%
  group_by(Education, Ocup_group) %>% tally(Freq) %>%
  group_by(Education) %>% mutate(prcnt = n/sum(n)) %>%
  group_by(Ocup_group) %>%
  mutate(avg_prcnt = sum(prcnt)*100/4) %>%
  mutate(perf_val =  (100*(prcnt/(avg_prcnt/100)-1))) %>%
  dplyr::select(Education,Ocup_group,perf_val,Ocup_distib = avg_prcnt) %>%
  spread(key = Education, value = perf_val) %>%
  mutate(across(where(is.numeric), round, 1))

## Graphing
graph_tbl2prfl_pilots <- tbl2prfl_pilots %>%
  dplyr::select(-Ocup_distib) %>%
  gather(2:5, key = "Education", value = "val_perf") %>%
  ggplot(aes(x = Ocup_group, xend = Ocup_group, y = val_perf,
                         yend = 0, group = Education)) +
  geom_point(col="darkred", size = 3) + geom_segment(col="black") +
  facet_wrap(~Education) + geom_hline(yintercept = 0 , colour = "gray") +
  coord_flip() +
  labs(title ="Ocupación vs Nivel educativo", x = "Ocupación", y = "Perfil") +
  theme_igray()

## Output
graph_tbl2prfl_pilots
```

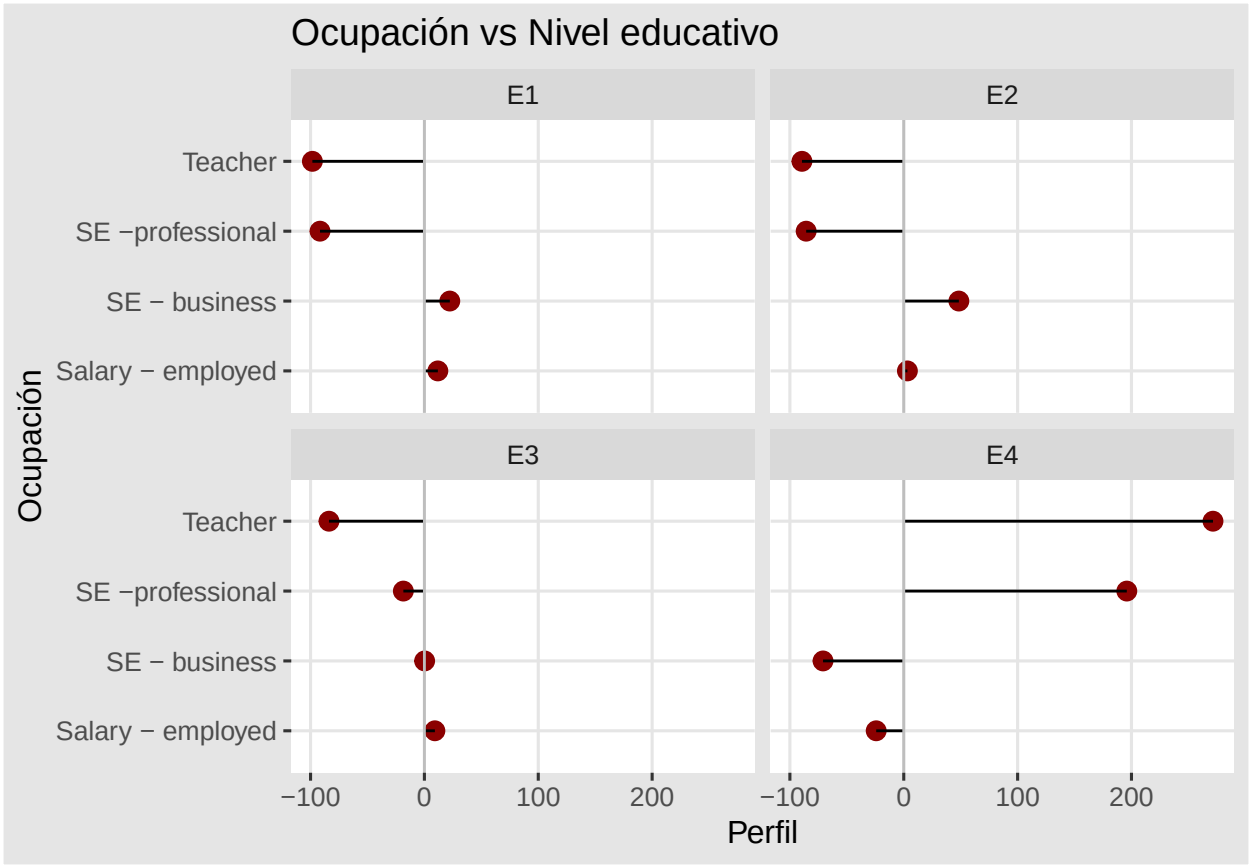


Gráfico de Porcentajes, Nivel de Aptitud vs Nivel de Educación.

2. Cereales

Usa el conjunto de datos UScereal (que está en R, en el paquete MASS, ver ?UScereal) para contestar las siguientes preguntas:

1. Describe la distribución del contenido de potasio y de fibra de los cereales. ¿Existe o no dispersión suficiente en estos datos para que la elección de cereal pueda tener algún efecto nutricional (busca una tabla de requerimientos mínimos, por ejemplo)?

- Con base en la información actualizada de la FDA de EUA, se tiene que:
  - Dietary Fiber 28g
  - Potassium 4700mg, aprox. 4.7g

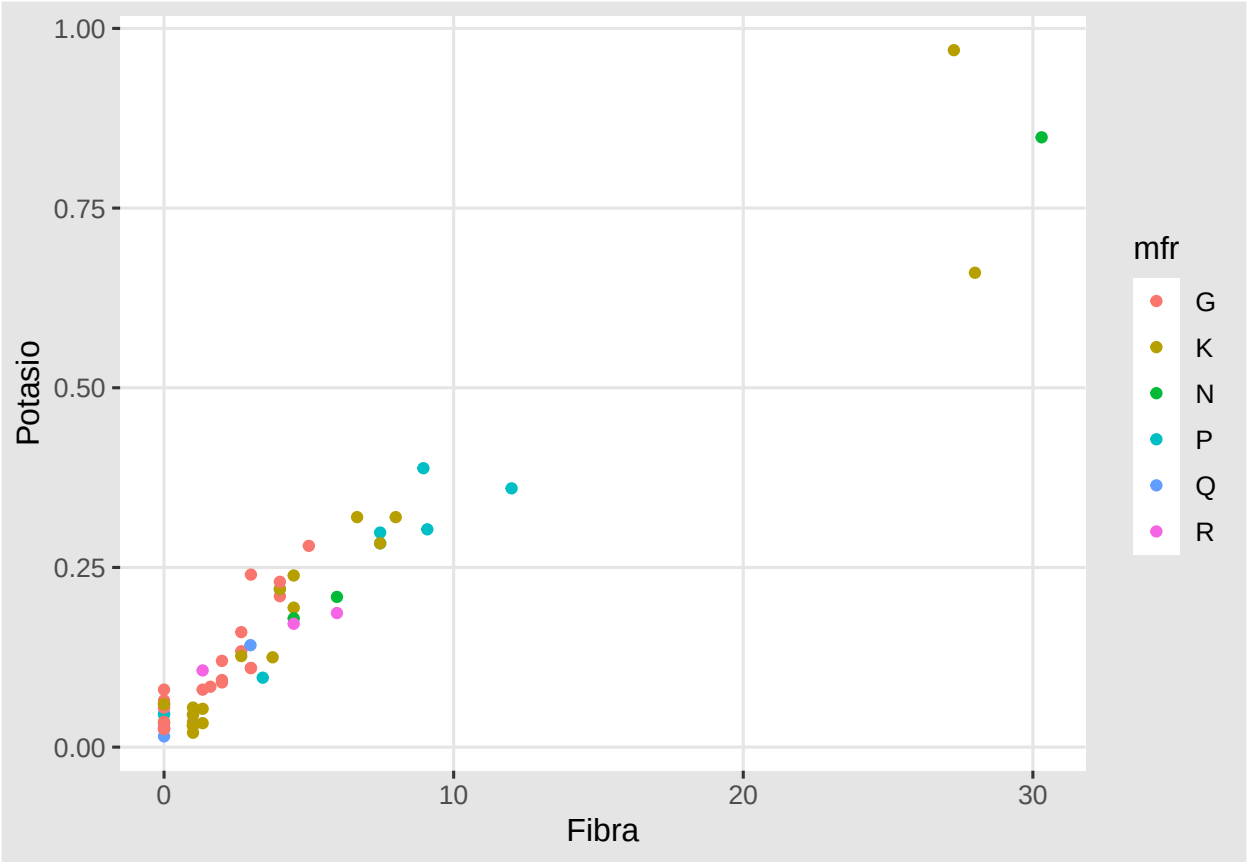
fuente: (FDA_Daily Value)[https://www.fda.gov/food/new-nutrition-facts-label/daily-value-new-nutrition-and-supplement-facts-labels] Existe una relación lineal creciente entre los diversos cereales analizados, indicando que a mayor cantidad de fibra se tiene una mayor cantidad de aportación de potasio. No obstante, de los 62 cereales analizados, 59 presentan un nivel bajo (por medida homologada de comparación, 1 cup us) de aportación de acuerdo a los niveles mínimos diarios indicados por la FDA (ver liga), de forma aproximada, inferior a un tercio de la fibra requerida e inferior (1/18) de la ingesta de potasio recomendada.

Si bien es cierto que la elección de estos cereales podría tener una mejor concentración de nutrientes, sólo tres poseen una cantidad aceptable de los requerimientos mínimos recomendados: aproximadamente un cuarto

de la ingesta de potasio requerida y prácticamente aporta toda el nivel de ingesta de fibra recomendada, manufactura de Kellogs y Nabisco con sus productos especializados de fibra.

Cabe mencionar que sería importante analizar otro tipo de variables como la relación de carbohidratos que puden afectar severamente a la salud.

```
cereales <- MASS::UScereal

grph_uscer <- ggplot(data = cereales ,
                     aes(x = fibre, y = (potassium/1000), colour = mfr)) +
  geom_point() +  theme_igray() +
  xlab("Fibra") + ylab("Potasio")

grph_uscer
```

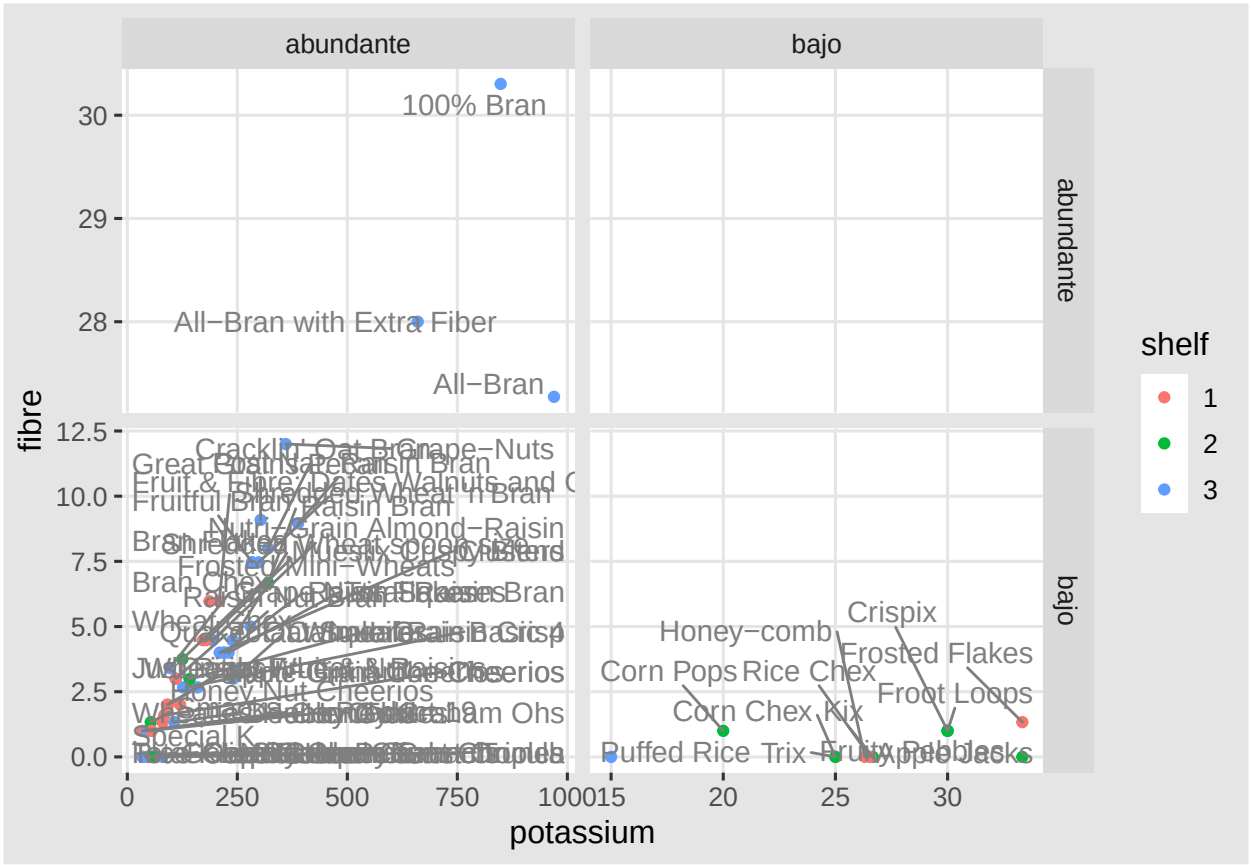



```
glimpse(cereales)
```

```
## Rows: 65
## Columns: 11
## $ mfr       <fct> N, K, K, G, K, G, R, P, Q, G, G, G, G, R, K, K, G, K, K, ...
## $ calories  <dbl> 212.1212, 212.1212, 100.0000, 146.6667, 110.0000, 173.333...
## $ protein   <dbl> 12.121212, 12.121212, 8.000000, 2.666667, 2.000000, 4.000...
## $ fat       <dbl> 3.030303, 3.030303, 0.000000, 2.666667, 0.000000, 2.66666...
## $ sodium    <dbl> 393.9394, 787.8788, 280.0000, 240.0000, 125.0000, 280.000...
## $ fibre     <dbl> 30.303030, 27.272727, 28.000000, 2.000000, 1.000000, 2.66...
```

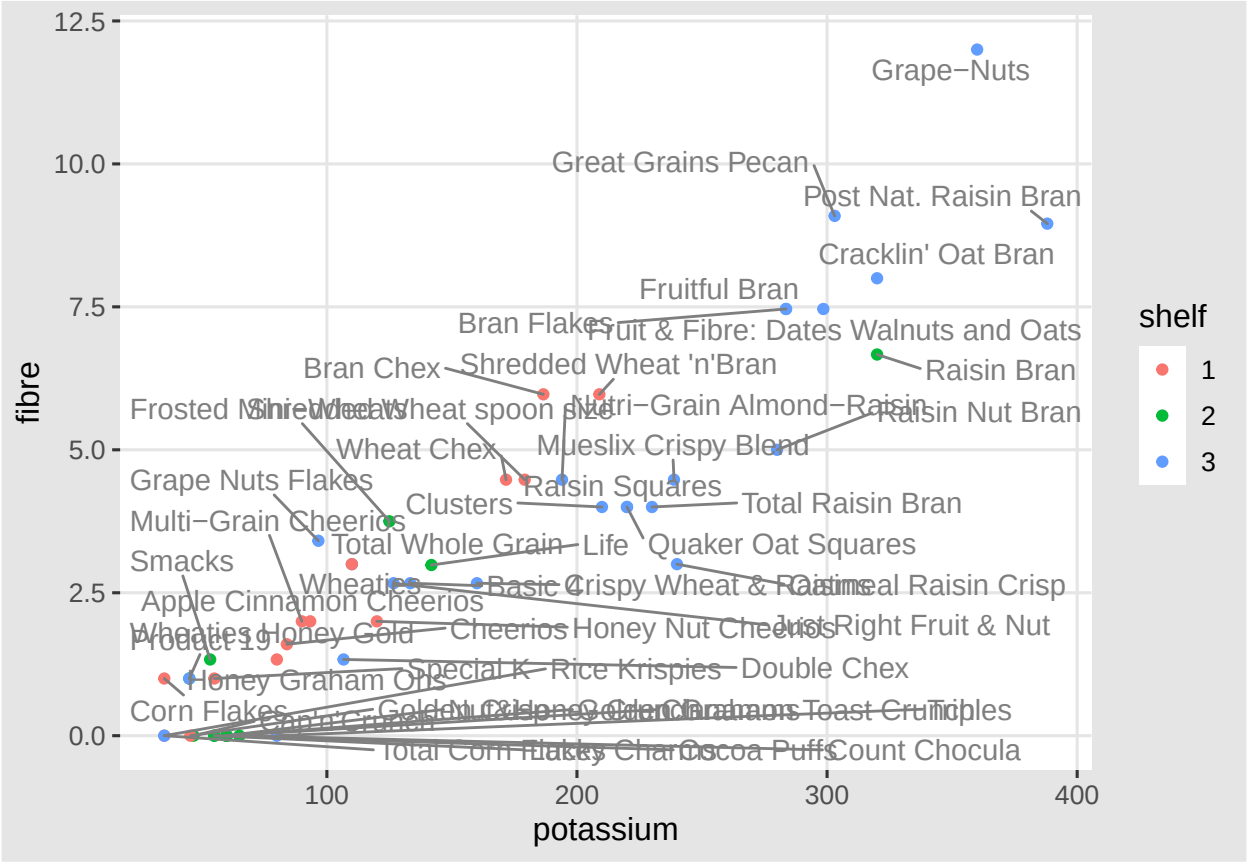
```
## $ carbo     <dbl> 15.15152, 21.21212, 16.00000, 14.00000, 11.00000, 24.0000...
## $ sugars    <dbl> 18.181818, 15.151515, 0.000000, 13.333333, 14.000000, 10....
## $ shelf     <int> 3, 3, 3, 1, 2, 3, 1, 3, 2, 1, 2, 3, 2, 1, 1, 2, 2, 3, 3, ...
## $ potassium <dbl> 848.48485, 969.69697, 660.00000, 93.33333, 30.00000, 133....
## $ vitamins  <fct> enriched, enriched, enriched, enriched, enriched, enriche...
```

```r
cereales$shelf = as.factor(cereales$shelf)

cereal<-cereales %>%
  mutate(marca = rownames(cereales))

cereal<-cereal %>%
  mutate(min_pot = ifelse(potassium < 35,"bajo", "abundante"),
         min_fib = ifelse(fibre < 20,"bajo", "abundante"))

ggplot(cereal, aes(potassium,fibre, label = marca)) +
  geom_point(aes(colour = shelf)) +
  geom_text_repel(colour = "gray50") +
  facet_grid(rows = vars(min_fib),
             cols = vars(min_pot),
             scales = "free",
             margins = "conservation") +
  theme_igray()
```

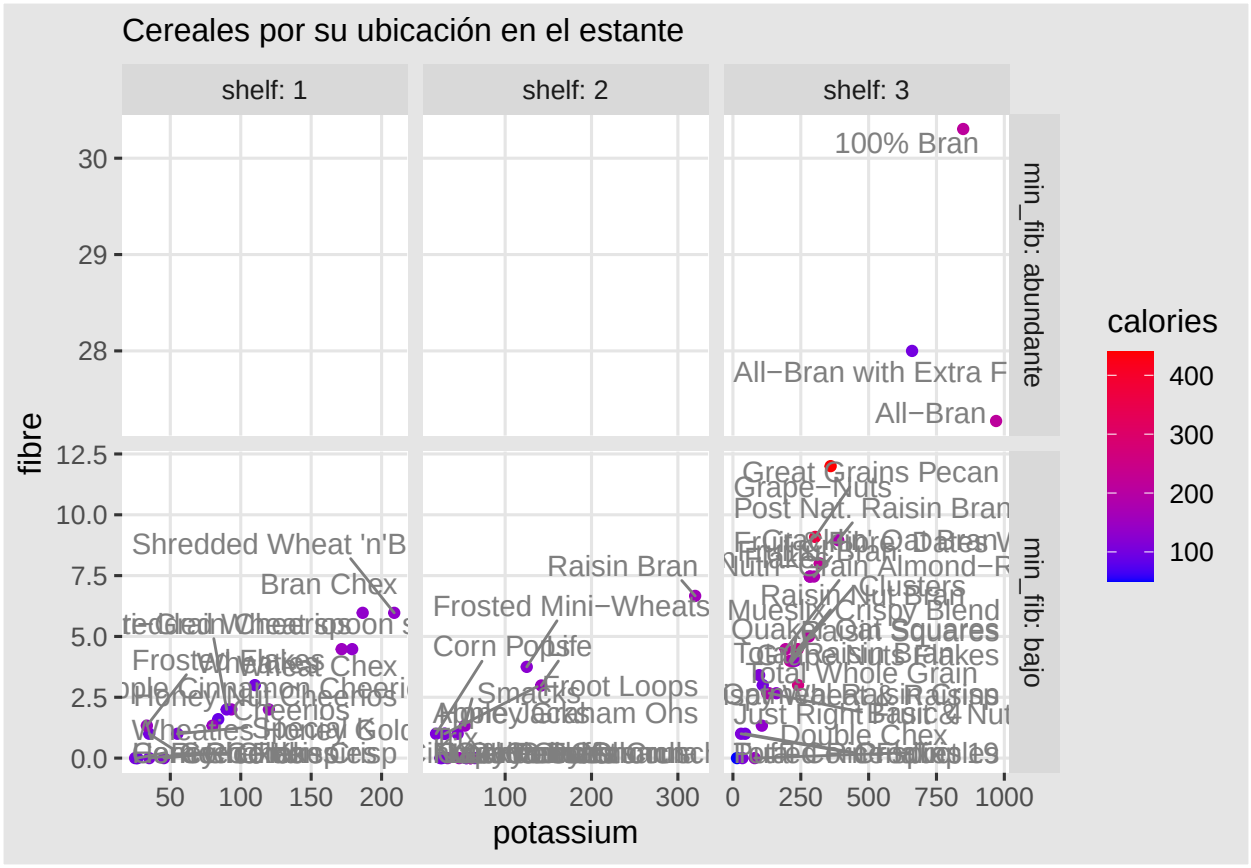


Conservemos los cereales donde están concentrados la mayoría para ver si hay algo

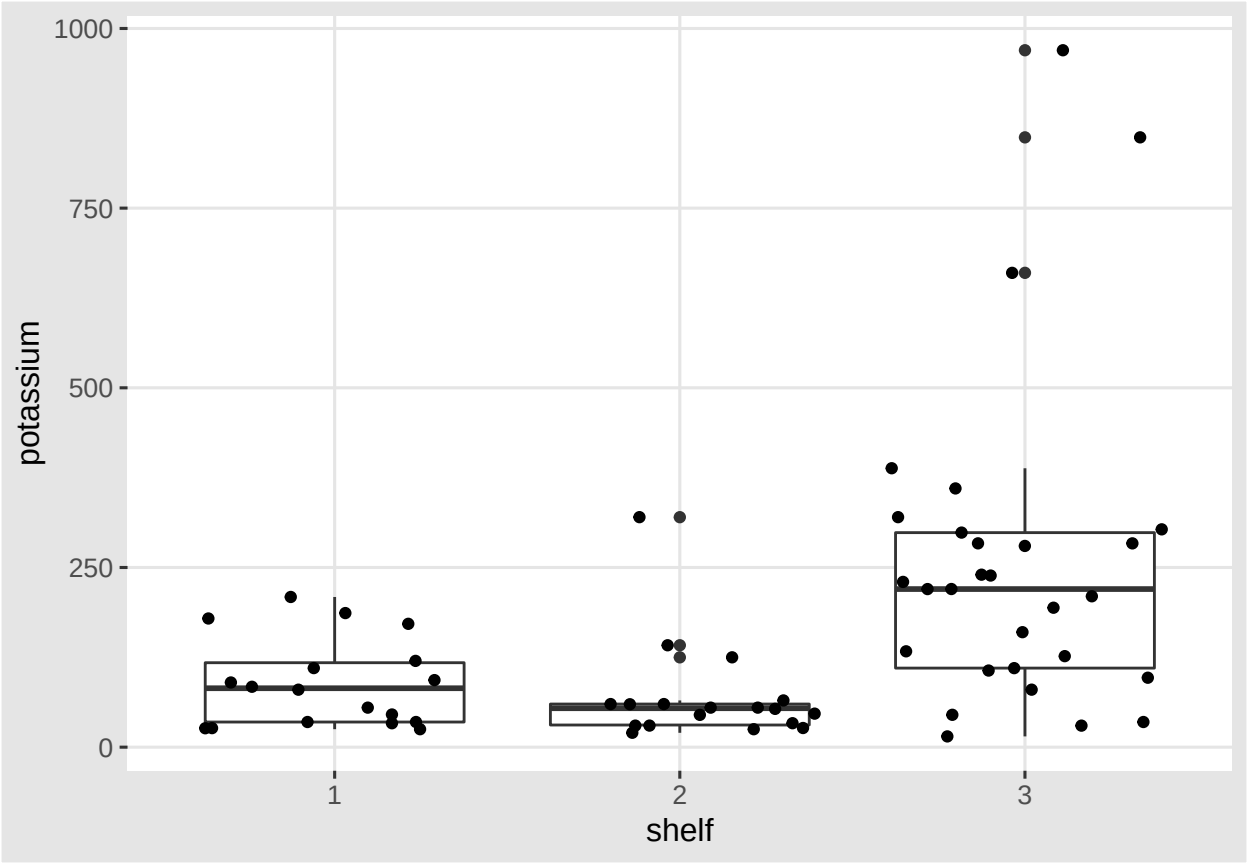
```
cereales1 <- cereal[cereal$min_fib == "bajo" & cereal$min_pot == "abundante",]
ggplot(cereales1,aes(potassium, fibre, label = marca)) +
  geom_point(aes(colour = shelf)) +
  geom_text_repel(colour = "gray50") +
  theme_igray()
```



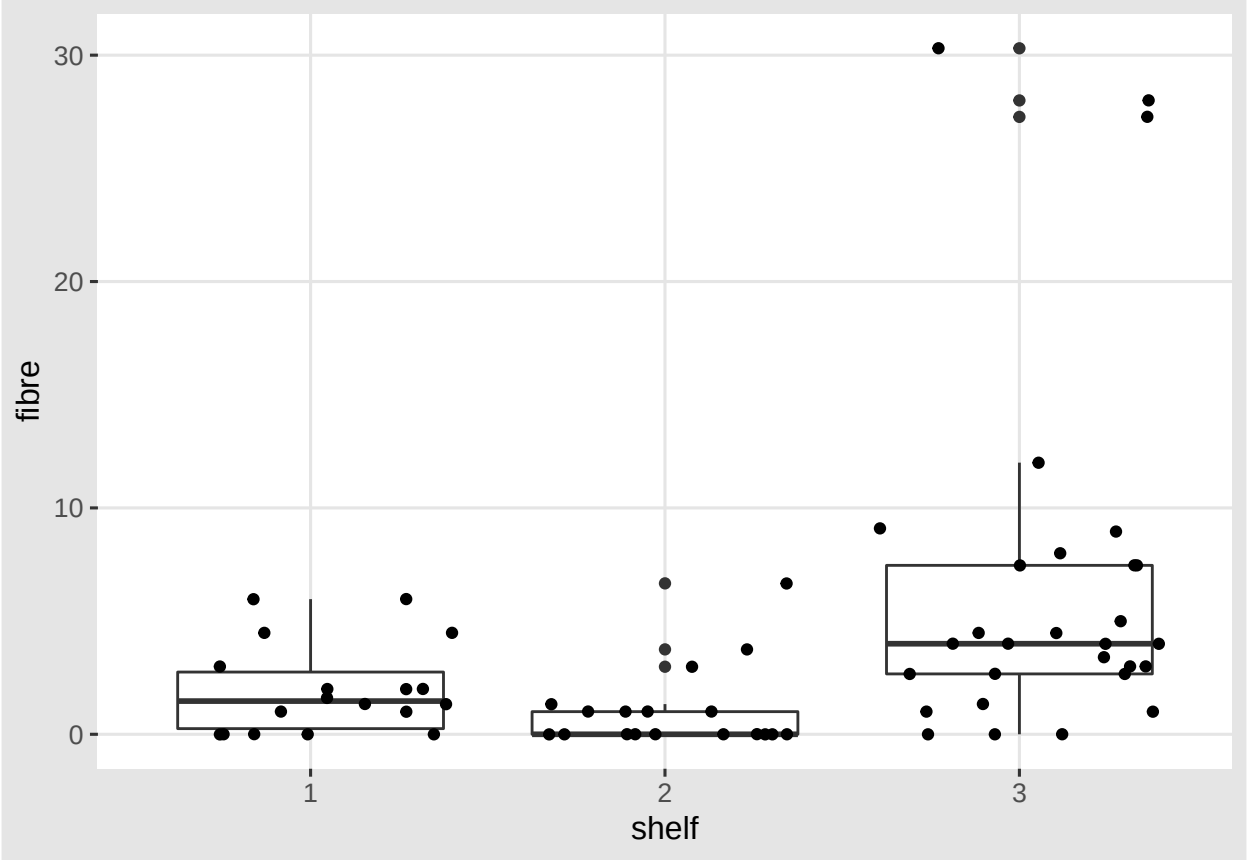

```
p <- ggplot(cereal,aes(potassium, fibre, label = marca)) +
  geom_point(aes(colour = calories)) +
  geom_text_repel(colour = "gray50") +
  facet_wrap(~ shelf, scales = "free_x") +
  labs("Cantidad de potasio y fibra por ubicación del estante de
       distintas marcas de cereal") +
  scale_color_gradient(low = "blue", high = "red")

ggplot(cereal,aes(potassium, fibre, label = marca)) +
  geom_point(aes(colour = calories))+
  geom_text_repel(colour = "gray50") +
  facet_grid(cols = vars(shelf), rows = vars(min_fib), scales = "free",
             labeller = label_both) +
  labs(subtitle = "Cereales por su ubicación en el estante") +
  scale_color_gradient(low = "blue", high = "red") +
  theme_igray()
```




```
ggplot(cereal,aes(shelf, potassium)) +
  geom_boxplot() + geom_jitter() +
  theme_igray()
```



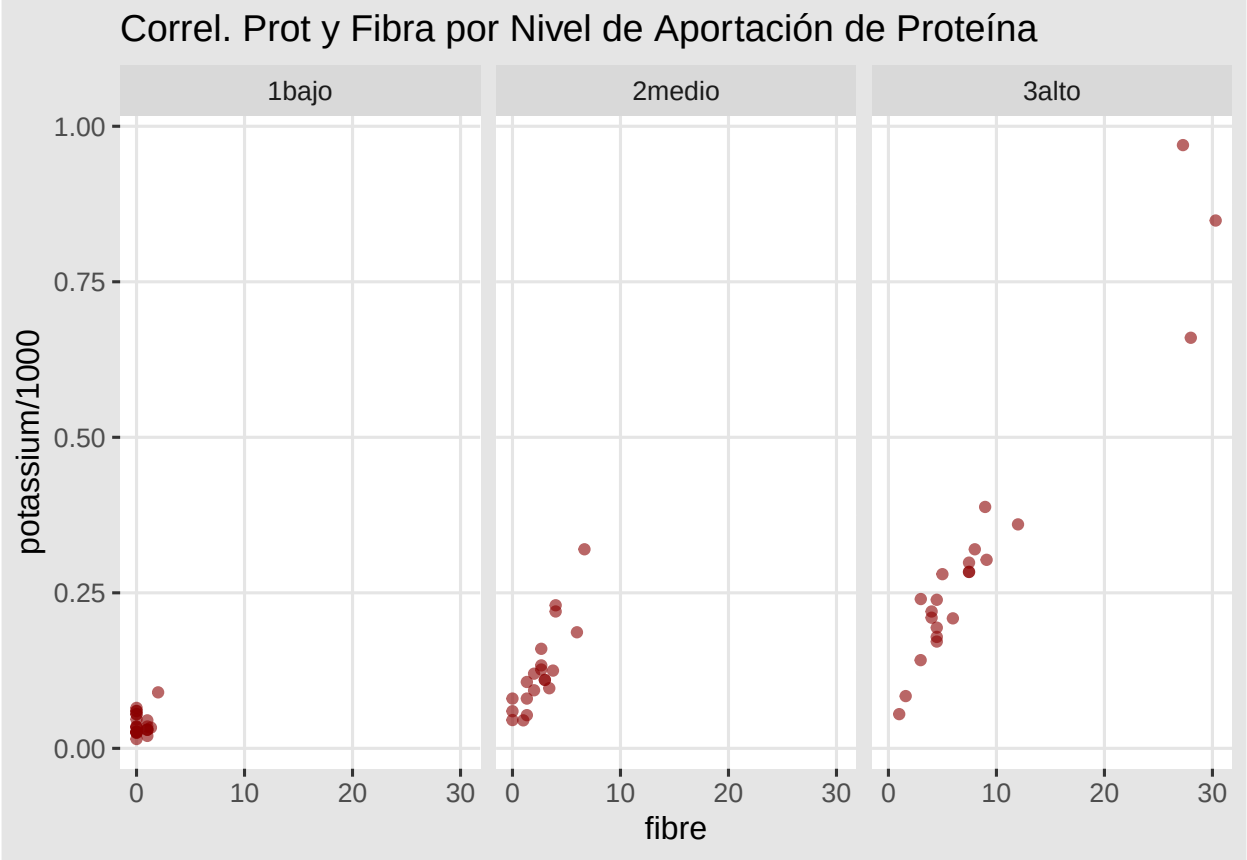

```
ggplot(cereal, aes(shelf, fibre)) +
  geom_boxplot() + geom_jitter() +
  theme_igray()
```



Conclusiones Considerando la ubicación en los estantes notamos que los cereales menos nutritivos se encuentran en el estante no. 1, probablemente para estar más al alcance de los niños, mientras que en el estante no. 3 se encuentran los que contienen más fibra y potasio.

2. Divide los cereales en tres grupos, según los cuantiles 1/3 y 2/3 del contenido de proteína. Grafica pequeños múltiplos para describir la relación entre potasio y fibra para cada uno de los tres grupos. ¿Se trata de la misma relación en cada grupo? ¿En qué son diferentes? ¿Cómo describirías los cereales del grupo con menos contenido de proteína (ve qué cereales son)?

```
prot_q3_q6 <- cereales %>% pull(protein) %>% quantile(., c((1/3),(2/3))) %>%
  round(3)

cereales2 <- cereales %>% rownames_to_column(var = 'brand') %>%
  mutate(aportac_prot = case_when(protein <= prot_q3_q6[1] ~ '1bajo',
                                  protein >= prot_q3_q6[2] ~ '3alto',
                                  TRUE ~ '2medio')) %>%
  dplyr::select(brand,fibre,potassium,protein,aportac_prot)

grph_prot <- ggplot(data = cereales2, aes(x = fibre, y = potassium/1000)) +
  geom_point(col = "darkred", alpha = .6) + facet_wrap(~aportac_prot,nrow = 1)+
  labs(title="Correl. Prot y Fibra por Nivel de Aportación de Proteína") +  theme_igray()

grph_prot
```



En el siguiente gráfico, nos permite confirmar nuestras observaciones anteriores.

Detectamos un clúster de cereales con niveles de aportación de fibra y poteína prácticamente nulos (se transforma la variable proteína (1/1000) para una mejor comprensión de los datos) y consistente con esto, pertenecen al grupo de bajo nivel de aportación de proteína. Consideramos que este grupo de cereales deberían ser considerados "vacíos" o sin aportación de nutrientes.

En el siguiente clúster (nivel de aportación de proteína medio) observamos que los niveles de aportación de fibra y potasio no cubren al menos un tercio de la aportación diraria requerida para el primero y es nulo para el segundo. En el tercer nivel de aportación de proteína, tenemos dos subgrupos:

- 1 Este subgrupo contiene un nivel ligeramente superior en los niveles de aportación de proteína y fibra en comparación con los que tienen aportación de proteína bajo y medio.
- 2 Un pequño subgrupo de tres cereales que tiene niveles muy altos, de fibra el 100% de la ingesta recomendada y en potasio un nivel cercano al 25%.

```
tab_cer2 <- cereales2 %>% filter(aportac_prot == "1bajo") %>%
  mutate(across(where(is.numeric),round,2))

## Format table
kbl(tab_cer2, caption = "low nutrition cereals.",
    align = 'c') %>%
  kable_styling(bootstrap_options = c("striped"), full_width = F)
```

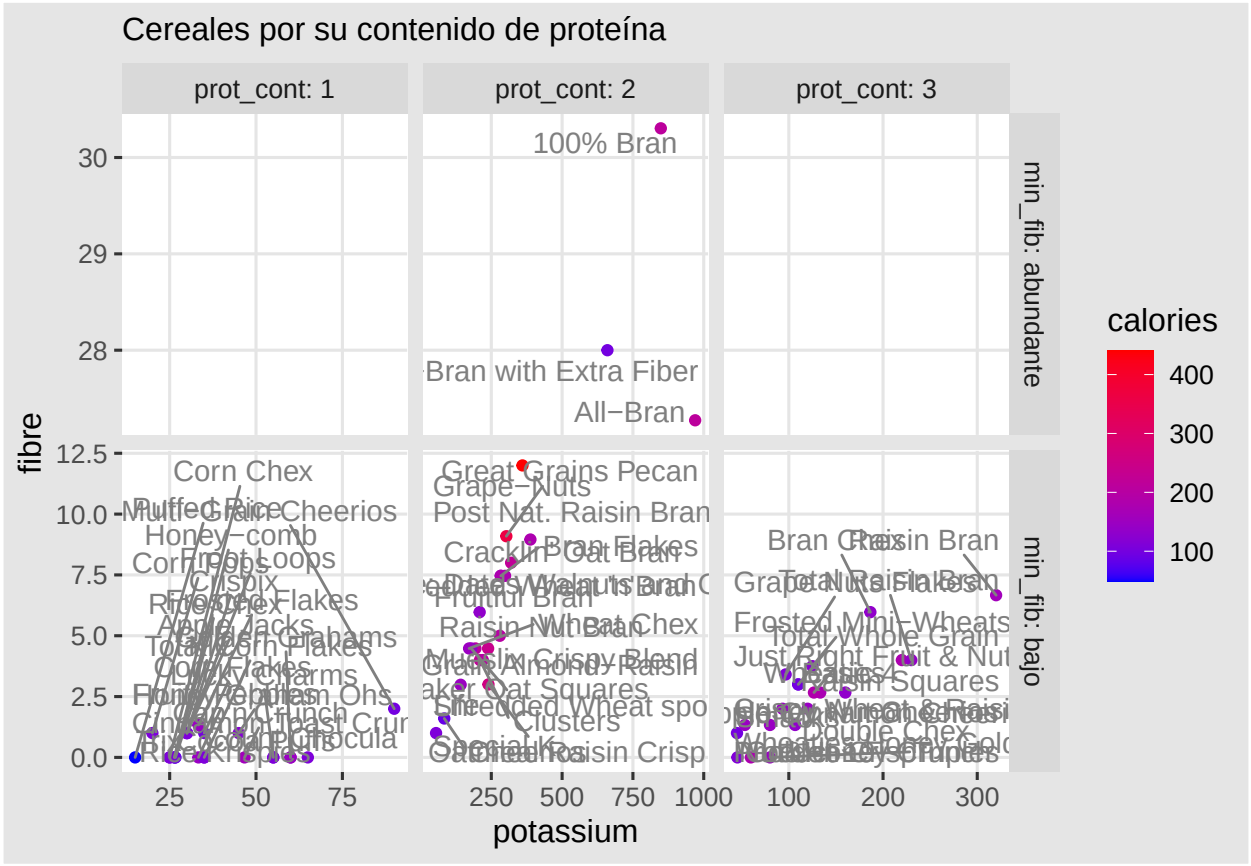
Table 3: low nutrition cereals.

brand	fibre	potassium	protein	aportac_prot
Apple Jacks	1.00	30.00	2.00	1bajo
Cap'n'Crunch	0.00	46.67	1.33	1bajo
Cinnamon Toast Crunch	0.00	60.00	1.33	1bajo
Cocoa Puffs	0.00	55.00	1.00	1bajo
Corn Chex	0.00	25.00	2.00	1bajo
Corn Flakes	1.00	35.00	2.00	1bajo
Corn Pops	1.00	20.00	1.00	1bajo
Count Chocula	0.00	65.00	1.00	1bajo
Crispix	1.00	30.00	2.00	1bajo
Froot Loops	1.00	30.00	2.00	1bajo
Frosted Flakes	1.33	33.33	1.33	1bajo
Fruity Pebbles	0.00	33.33	1.33	1bajo
Golden Grahams	0.00	60.00	1.33	1bajo
Honey Graham Ohs	1.00	45.00	1.00	1bajo
Honey-comb	0.00	26.32	0.75	1bajo
Kix	0.00	26.67	1.33	1bajo
Lucky Charms	0.00	55.00	2.00	1bajo
Multi-Grain Cheerios	2.00	90.00	2.00	1bajo
Puffed Rice	0.00	15.00	1.00	1bajo
Rice Chex	0.00	26.55	0.88	1bajo
Rice Krispies	0.00	35.00	2.00	1bajo
Total Corn Flakes	0.00	35.00	2.00	1bajo
Trix	0.00	25.00	1.00	1bajo

```
cuantil(cereal$protein,probs = c(1/3,2/3,1))
```

```
## # A tibble: 3 x 2
##   cuantil valor
##     <dbl> <dbl>
## 1   0.333  2
## 2   0.667  4.32
## 3   1     12.1
```

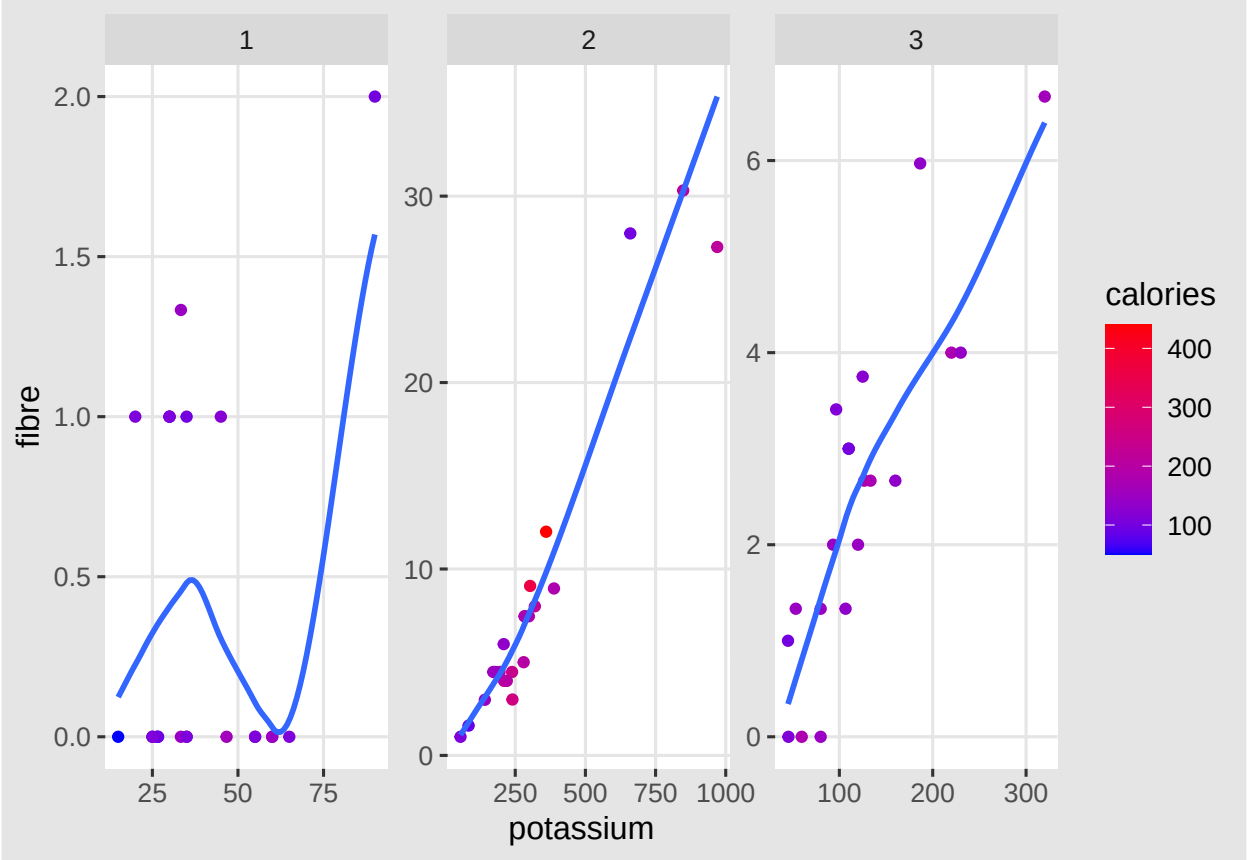
```
cereales2 <- cereal%>%
  mutate(prot_cont = ifelse(protein <= 2, "1",
                            ifelse(protein > 2 & protein >= 4.32, "2", "3")))

ggplot(cereales2, aes(potassium, fibre, label = marca)) +
  geom_point(aes(colour = calories)) +
  geom_text_repel(colour = "gray50") +
  facet_grid(cols = vars(prot_cont),rows = vars(min_fib), scales = "free",
             labeller = label_both) +
  labs(subtitle = "Cereales por su contenido de proteína") +
  scale_color_gradient(low = "blue", high = "red") +
  theme_igray()
```



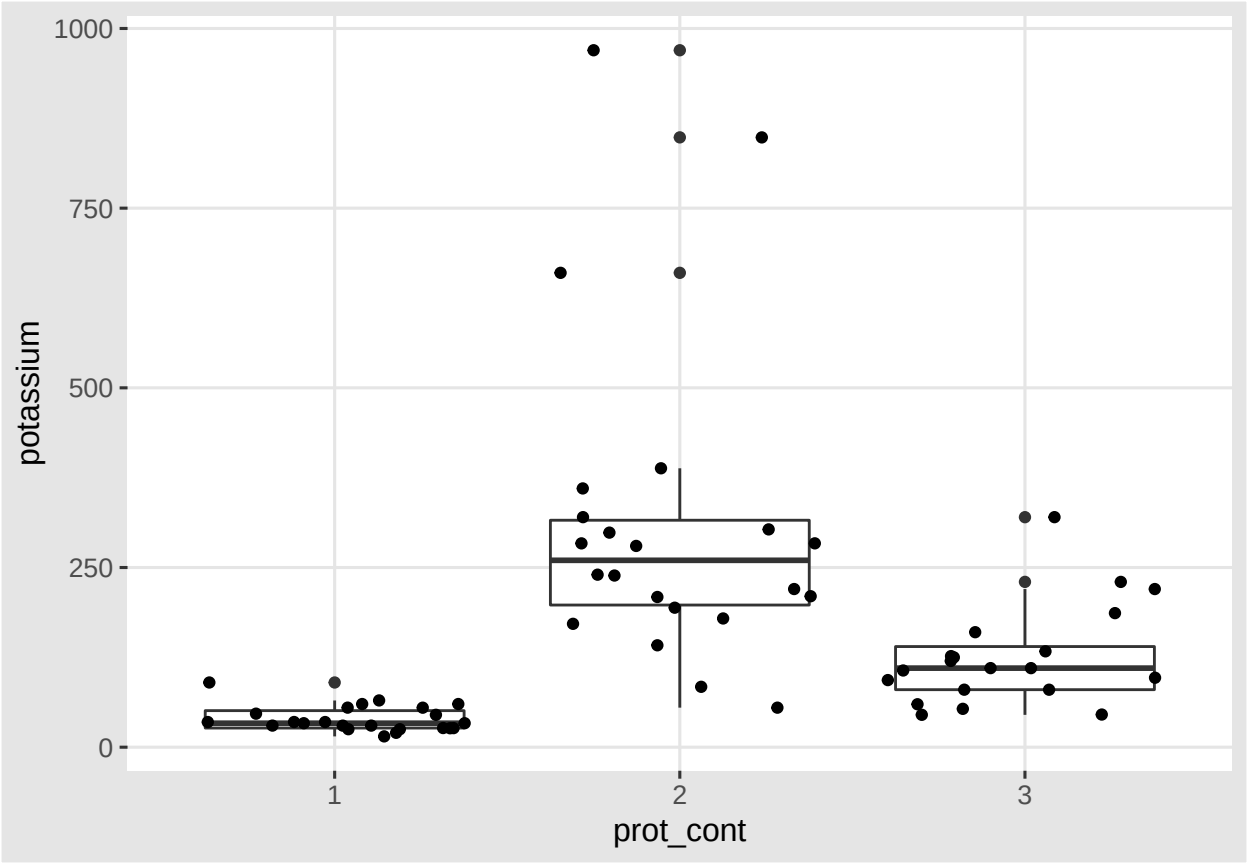

```
ggplot(cereales2,aes(potassium,fibre)) +
  geom_point(aes(colour=calories)) +
```

```
  geom_smooth(method = "loess", span = 0.8, se = FALSE,
              method.args = list(degree = 1, family = "symmetric"))+
  facet_wrap(vars(prot_cont),scales = "free")+
  scale_color_gradient(low = "blue",high = "red") +
  theme_igray()
```

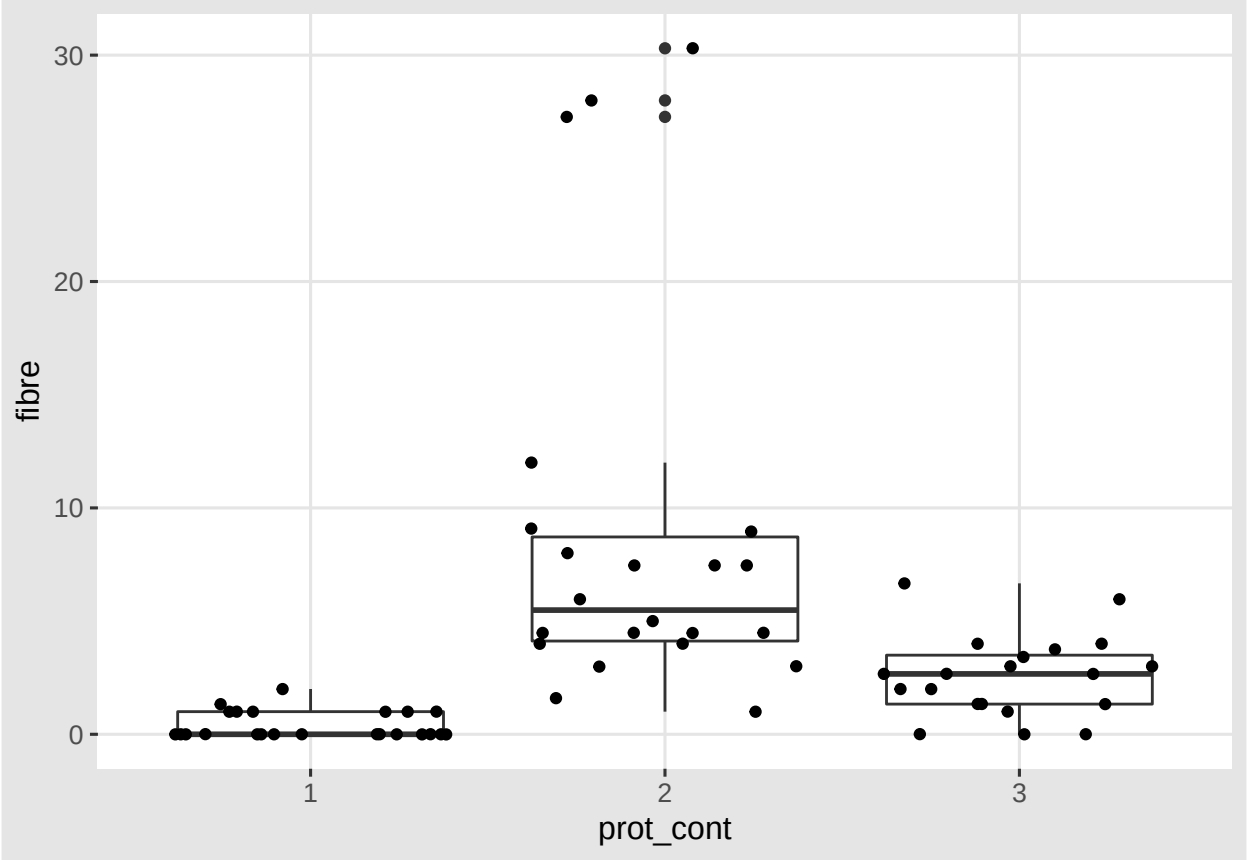
```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(cereales2,aes(prot_cont,potassium)) +
  geom_boxplot() + geom_jitter() +
  theme_igray()
```




```
ggplot(cereales2,aes(prot_cont,fibre)) +
  geom_boxplot() + geom_jitter() +
  theme_igray()
```



Pruebas de hipótesis

1. Ascorbato

Pacientes con cancer terminal avanzado en el estómago y mama se trataron con ascorbato para prolongar la supervivencia. Los datos `ascorbate` muestran la supervivencia en días. Trabaja con los datos en escala logarítimica.

```
asco <- read.csv("ascorbate.csv")
asco <- asco%>%
  mutate(ID = seq(1:13)) %>%
  mutate(log_stomach = log(stomach), log_breast = log(breast))

# Alargamos la tabla
asco_larga <- asco %>%
  dplyr::select(ID,log_breast,log_stomach)%>%
  pivot_longer(cols = c("log_stomach","log_breast"))
```

1. Realiza una prueba de hipótesis visual para comparar las mediciones de los dos grupos. Describe tus conclusiones.

Prueba de permutación

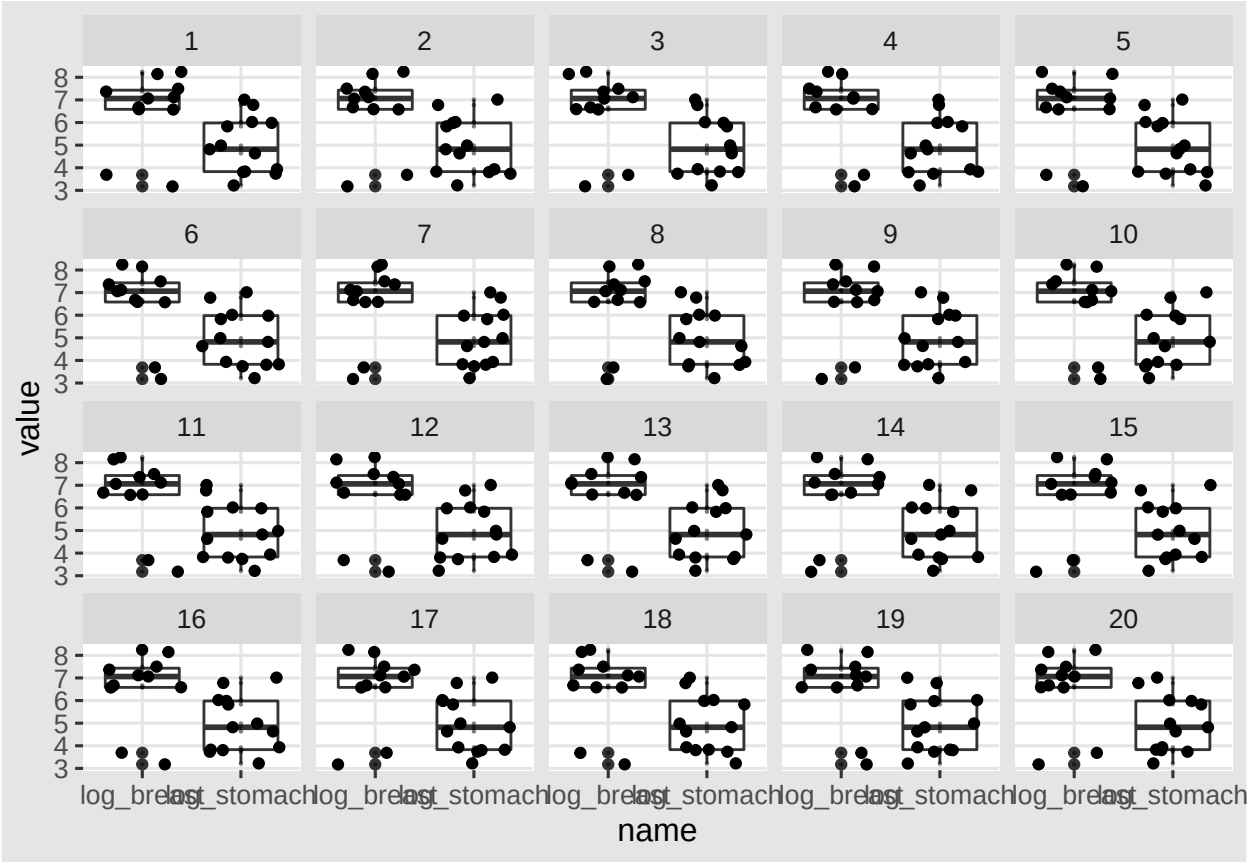
- H_0: los grupos tienen distribuciones similares

- H_1: los grupos se distribuyen distinto

```
set.seed(123)
reps <- lineup(null_permute("log_stomach"), asco_larga, 20)
```

```
## decrypt("Iw5d XVqV Hz k2AHqH2z tF")
```

```
ggplot(reps, aes(name, value)) +
  geom_boxplot() + geom_point(alpha = 0.5, size = 0.1) +
  geom_jitter() + facet_wrap(~.sample) +
  theme_igray()
```




```
decrypt("Iw5d XVqV Hz k2AHqH2z tF")
```

```
## [1] "True data in position  15"
```

Conclusión No se logra escoger la gráfica, así que no tenemos evidencia suficiente en contra de que la supervivencia de ambos tipos de cancer se distribuya de forma similar. Es posible que las diferencia que observamos se deban a variación muestral.

2. Usa una prueba de permutación para examinar la hipótesis de que no hay diferencia en la media de los tiempos de supervivencia. Escribe la hipóteisis nula, hipótesis alterna, y reporta el valor p de la prueba.

Hacemos una prueba de diferencia de medias

Prueba de permutación

- H_0: La media de los tiempos de supervivencia de ambos grupos es similar.
- H_1: La media de los tiempos de supervivencia de ambos grupos es diferente.
- Calculamos la diferencia observada

```
dif_obs <- asco %>%
  summarise(med_est = mean(log_stomach,na.rm = TRUE),
            med_br = mean(log_breast, na.rm = TRUE)) %>%
  mutate(dif_obs = med_est - med_br) %>%
  pull(dif_obs)

dif_obs
```

```
## [1] -1.590684
```

- Hacemos una permutación

```
asco_perm <- asco_larga %>%
  group_by(ID) %>%
  mutate(name = sample(c("log_stomach", "log_breast"))) %>%
  ungroup()
```

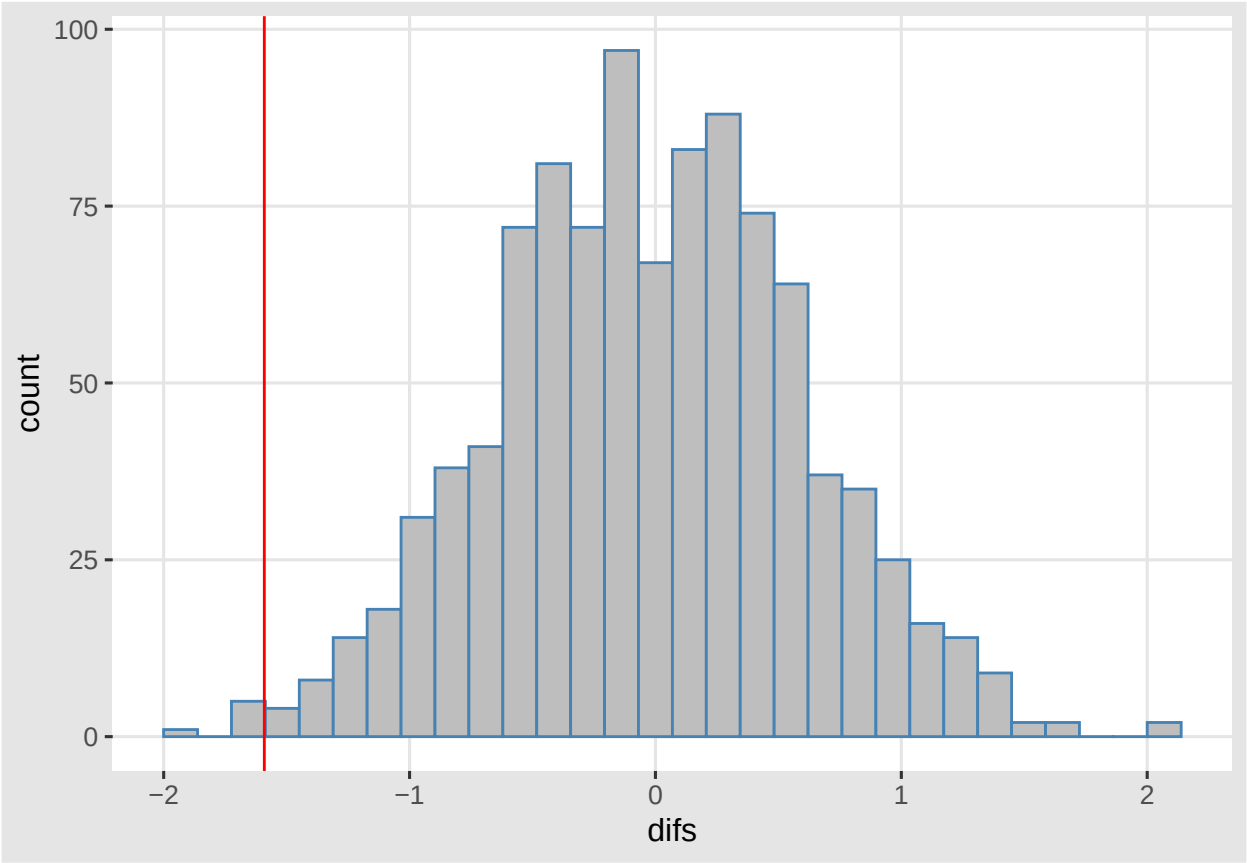
- Calculamos la estadística en esta permutación

```
asco_perm %>%
  pivot_wider(names_from = "name", values_from = "value") %>%
  summarise(med_est = mean(log_stomach,na.rm = TRUE),med_br = mean(log_breast,na.rm = TRUE))%>%
  mutate(media_dif = med_est-med_br)%>%
  pull(media_dif)
```

```
## [1] 0.761039
```

```
#Lo hacemos función
calcula_est<-function(){
  asco_perm <- asco_larga %>%
    group_by(ID) %>%
    mutate(name = sample(c("log_stomach", "log_breast"))) %>%
    ungroup()

  asco_perm %>%
    pivot_wider(names_from = "name", values_from = "value") %>%
    summarise(med_est = mean(log_stomach,na.rm=TRUE),med_br = mean(log_breast,na.rm = TRUE))%>%
    mutate(media_dif = med_est - med_br)%>%
    pull(media_dif)

}

difs <- rerun(1000, calcula_est()) %>% flatten_dbl()

perms <- tibble(sims = 1:1000, difs = difs)
```

```
ggplot(perms, aes(x = difs)) +
  geom_histogram(colour = "steelblue", fill = "gray", bins = 30) +
  geom_vline(xintercept = dif_obs, color = "red") +
  theme_igray()
```




```
dist_perm<-ecdf(perms$difs)
```

- Calculamos el percentil del valor observado

```
percentil_obs <- dist_perm(dif_obs)
percentil_obs
```

```
## [1] 0.006
```

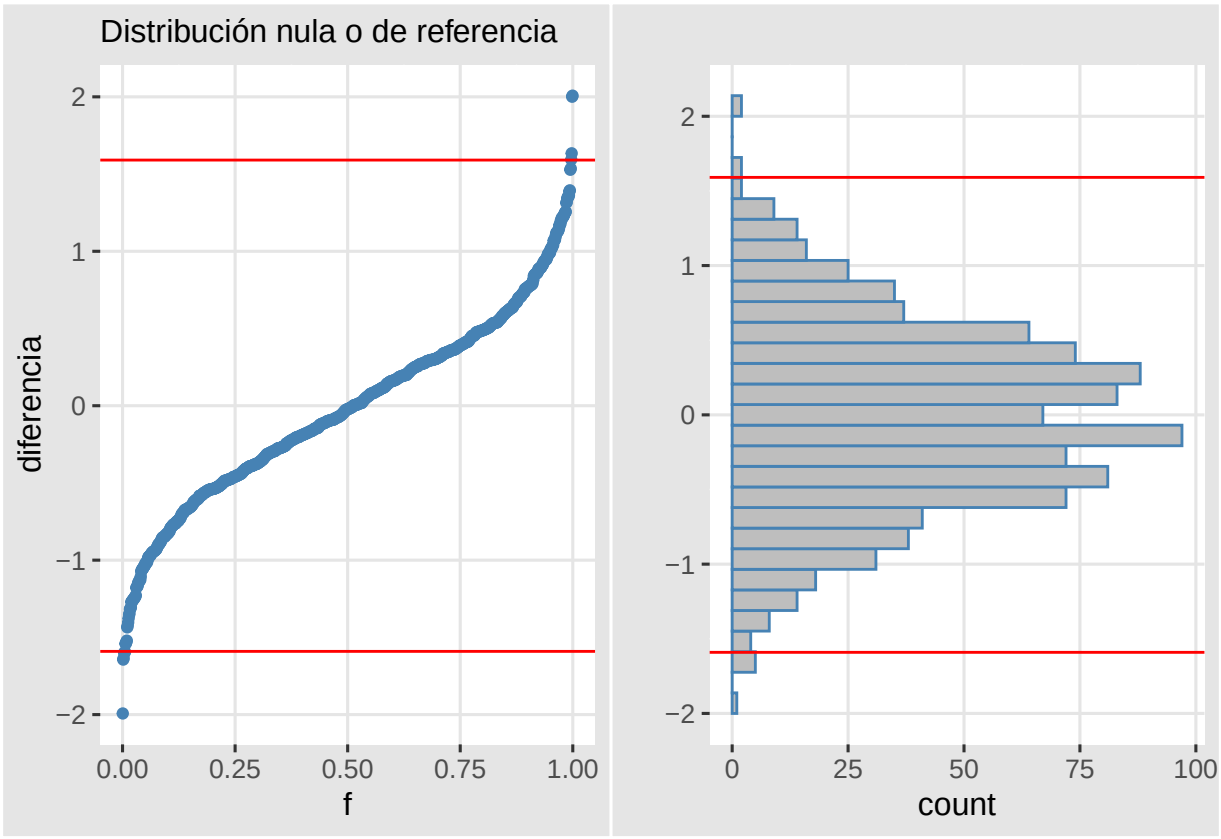
Calculamos el p-value de dos colas

```
p_valor <- 2*min(dist_perm(dif_obs),(1 - dist_perm(dif_obs)))
p_valor
```

```
## [1] 0.012
```

No aporta mucha evidencia en contra de que los grupos tienen distribuciones similares

```
g1 <- ggplot(perms, aes(sample = difs)) +
  geom_qq(distribution = stats::qunif, color = "steelblue") +
  geom_hline(yintercept = dif_obs, color = "red") +
  geom_hline(yintercept = - dif_obs, color = "red") +
  xlab("f") + ylab("diferencia") + labs(subtitle = "Distribución nula o de referencia") +
  theme_igray()

g2 <- ggplot(perms, aes(x = difs)) +
  geom_histogram(colour = "steelblue", fill = "gray", bins = 30) +
  geom_vline(xintercept = dif_obs, color = "red") +
  geom_vline(xintercept = - dif_obs, color = "red") +
  coord_flip() + xlab("") + labs(subtitle = " ") +
  theme_igray()

g1 + g2
```



2. Prueba visual

1. La base de datos `places` (Boyer y Savageau 1984) contiene *ratings* de varios aspectos de ciudades de EUA. El objetivo de este ejercicio es investigar si las variables en estos datos están asociadas, en particular se considera clima (Climate) y costo de vivienda (HousingCost). Valores bajos en clima implican temperaturas inconvenientes, puede ser mucho calor o mucho frío, mientras que valores altos corresponden a temperaturas más moderadas. Por su parte, valores altos en vivienda indican costos altos para una casa familiar simple.

- ¿Qué relación esperarías entre las variables? Escribe la hipótesis nula.

- Describe un método gráfico para probar tu hipótesis e implementalo. Genera 9 conjuntos de datos nulos y graficalos junto con los datos reales, escribe el nivel de significancia de la prueba y tus conclusiones.

```
places <- read.csv("places.csv", row.names=1, sep="")
```

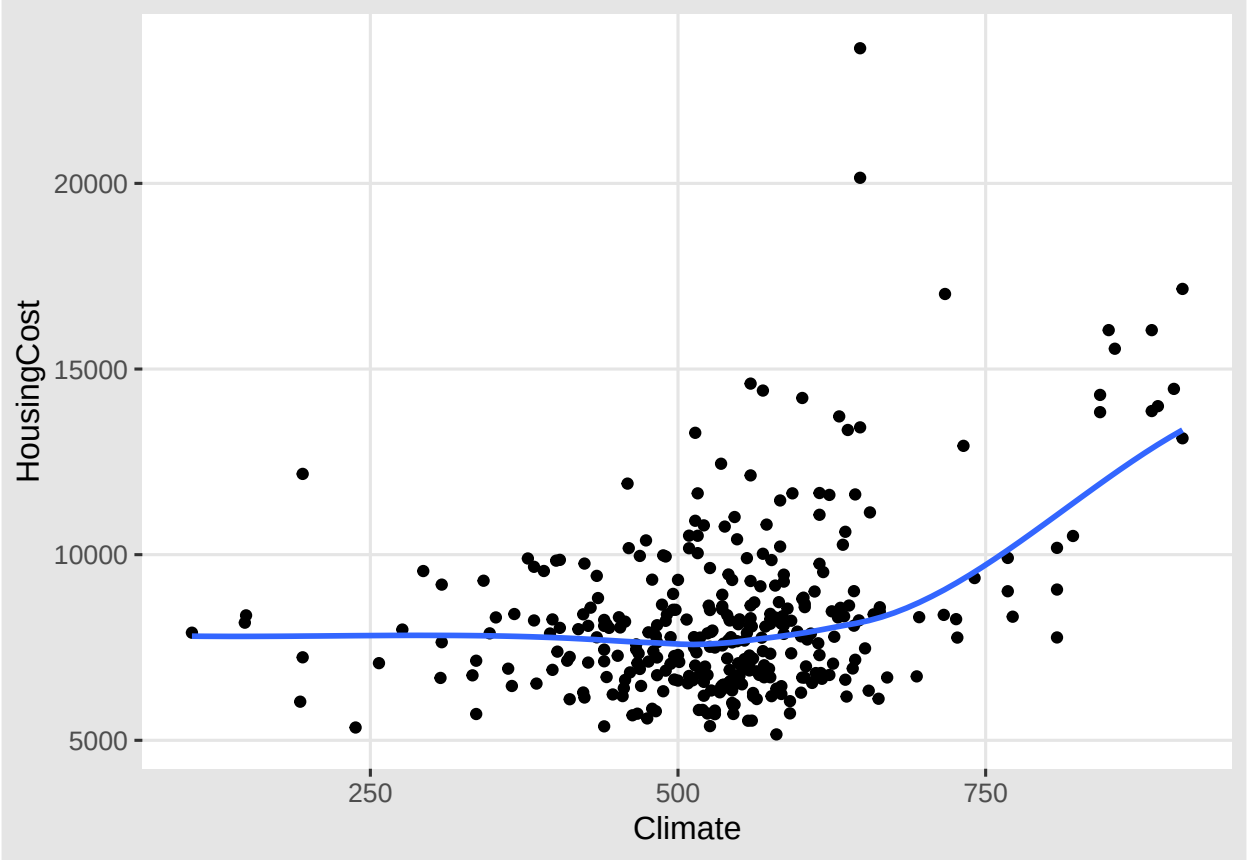
```
glimpse(places)
```

```
## Rows: 329
## Columns: 14
## $ Climate     <int> 521, 575, 468, 476, 659, 520, 559, 537, 561, 609, 885, ...
## $ HousingCost <int> 6200, 8138, 7339, 7908, 8393, 5819, 8288, 6487, 6191, 6...
## $ HlthCare    <int> 237, 1656, 618, 1431, 1853, 640, 621, 965, 432, 669, 20...
## $ Crime       <int> 923, 886, 970, 610, 1483, 727, 514, 706, 399, 1073, 983...
## $ Transp      <int> 4031, 4883, 2531, 6883, 6558, 2444, 2881, 4975, 4246, 4...
## $ Educ        <int> 2757, 2438, 2560, 3399, 3026, 2972, 3144, 2945, 2778, 2...
## $ Arts        <int> 996, 5564, 237, 4655, 4496, 334, 2333, 1487, 256, 1235,...
## $ Recreat     <int> 1405, 2632, 859, 1617, 2612, 1018, 1117, 1280, 1210, 11...
## $ Econ        <int> 7633, 4350, 5250, 5864, 5727, 5254, 5097, 5795, 4230, 6...
## $ CaseNum     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Long        <dbl> -99.6890, -81.5180, -84.1580, -73.7983, -106.6500, -92....
## $ Lat         <dbl> 32.55900, 41.08500, 31.57500, 42.73270, 35.08300, 31.30...
## $ Pop         <int> 110932, 660328, 112402, 835880, 419700, 135282, 635481,...
## $ StNum       <int> 44, 36, 11, 35, 33, 19, 39, 15, 39, 44, 5, 1, 16, 41, 2...
```

Ya que queremos saber si las variables están correlacionadas vamos a verlas con una relación loess

```
ggplot(places,aes(Climate, HousingCost)) +
  geom_point() +
  geom_smooth(method = "loess", span = 0.8, se = FALSE,
              method.args = list(degree = 1, family = "symmetric")) +
  theme_igray()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Un diagrama de dispersión muestra la relación entre dos variables continuas y responde a la pregunta: ¿existe una relación entre x y y ?

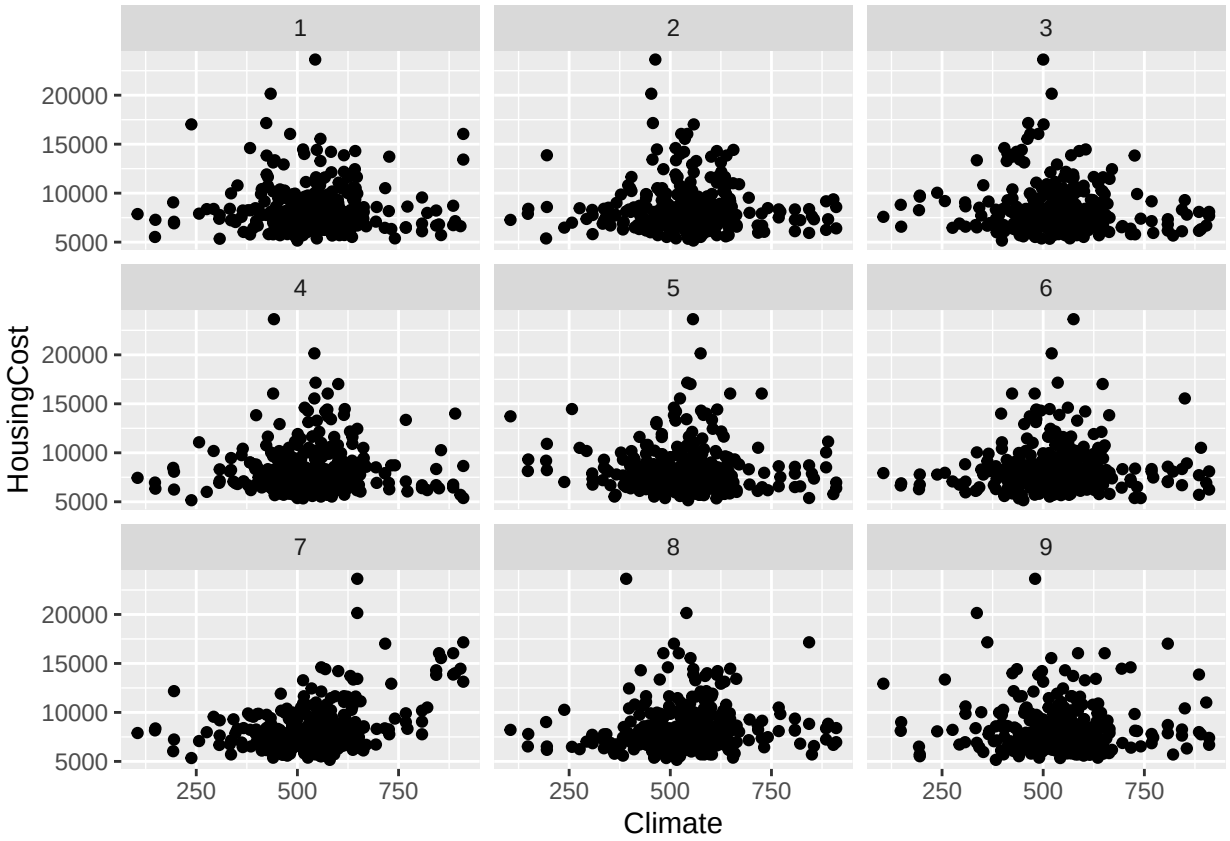
Una posible hipótesis nula es que no hay relación entre las variables. Vamos usar pruebas visuales para esta hipótesis

Graficamos 9 conjuntos de datos e intentamos encontrar nuestros datos, si lo identificamos estaríamos rechazando la hipótesis nula con una significancia de 1/9.

```
set.seed(43)
reps <- lineup(null_permute("Climate"), places, 9)
```

```
## decrypt("Iw5d XVqV Hz k2AHqH2z 9m")
```

```
ggplot(reps, aes(Climate,HousingCost)) +
  geom_point()+
  facet_wrap(~.sample)
```




```
decrypt("Iw5d XVqV Hz k2AHqH2z 9m")
```

```
## [1] "True data in position  7"
```

Conclusiones Sí identificamos el conjunto más o menos claramente, así que tenemos evidencia en contra de que no hay relación entre las variables con un nivel de significancia de 1/9.

Bootstrap

Elige uno de los siguientes dos ejercicios (bioequivalencia o tráfico):

1. Bioequivalencia

La bioequivalencia es un término utilizado en farmacocinética para evaluar la equivalencia terapéutica entre dos formulaciones de un medicamento que contiene el mismo principio activo o fármaco. Cuando una farmacéutica cambia una fórmula o desarrolla una nueva presentación de un medicamento ya disponible al público, requiere aprobación de la FDA para poder distribuirlo, para lograr esta aprobación debe demostrar que el nuevo medicamento es bioequivalente al medicamento ya aprobado, es así que se diseñan ensayos clínicos donde se compara la respuesta de los participantes al recibir las distintas formulaciones del medicamento.

En este ejercicio analizarás la bioequivalencia de una nueva tableta de Cimetidine de 800 mg en relación a tabletas aprobadas de 400 mg:

- En el ensayo clínico participaron 24 voluntarios saludables, cada uno se asignó de manera aleatoria para recibir la formulación de 800 mg (formulación a prueba) o dos tabletas de 400 mg (formulación referencia).
- Se recolectaron muestras de sangre antes de la dosis y durante las siguientes 24 horas.

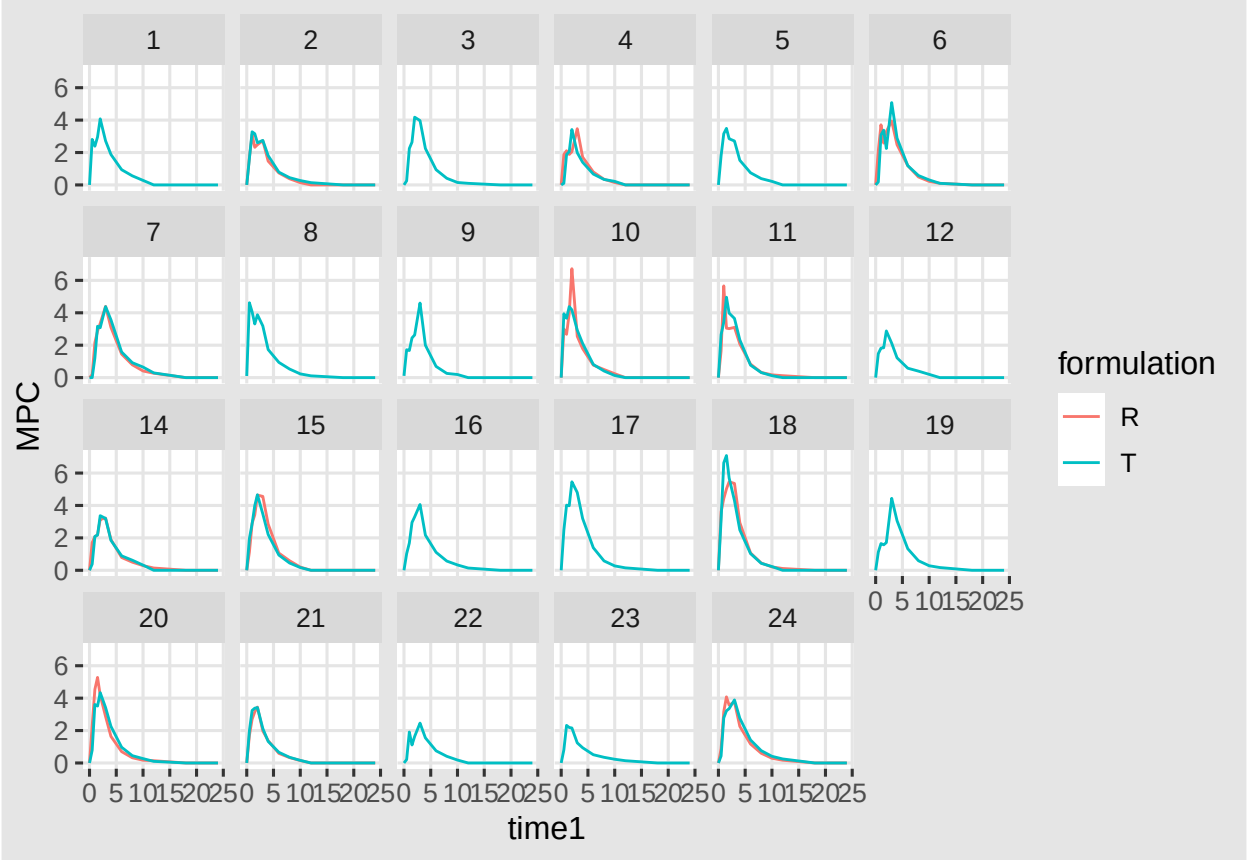
Realiza lo siguiente:

1. Lee los datos cimetidine_raw, las variables son: + `formulation` indica si la observación corresponde a formulación de refernecia o de prueba, + `subj` identifica al sujeto, + `seq` toma dos valores 1 indica que el sujeto se eavluó tomando la formulación de tratamiento primero y después la de referencia, 2 indica el caso contrario, + `prd` indica el periodo (1 o 2), + las variables `HRXX` indican la medición (concentración de Cimeditine en mCG/ml) para la hora XX (HR00 corresponde a la hora cero, HR05 a media hora, HR10 a una hora,..., HR240 a 24 horas). Desafortunadamente estos datos crudos están truncados y para algunos sujetos (1, 3, 5, 8, 9, 12, 16, 17, 19, 22, 23) no tenemos las mediciones del tratamiento de referencia, sin embargo, podemos usar la información disponible para entender como se calculan las métricas.

```
cime <- read_delim("cimeditine-raw.txt",
    "\t", escape_double = FALSE, trim_ws = TRUE)

cime_auc <- read_csv("cimetidine-auc.csv")
cime_boot <- read.csv(url("https://raw.githubusercontent.com/tereom/est-computacional-2018/master/data/
```

2. Grafica la concentración del medicamento por hora. Debes graficar en el eje horizontal la hora, en el eje vertical la concentración para cada persona, bajo cada tratamiento. Un ejemplo de lo que debes hacer es esta gráfica, con la diferencia que la curva de Wikipedia es el promedio sobre todos los individuos y tu graficarás una para cada uno. ¿Qué puedes ver en las gráficas?

```
cime_larga <- cime %>%
  pivot_longer(cols = starts_with("HR"), names_to = "time", values_to = "MPC") %>%
  arrange(subj) %>%
  mutate(time1 = rep(c(0,0.5,1,1.5,2,3,4,6,8,10,12,18,24),35))

cime_larga$subj = as.factor(cime_larga$subj)

ggplot(cime_larga, aes(time1, MPC, color = formulation)) +
  geom_line() + facet_wrap(~subj, nrow = 4, ncol = 6) +
  theme_igray()
```



Podemos ver que para todos los sujetos las gráficas son similares en cuanto a la concentración del tratamiento por hora, teniendo todos la máxima concentración entre las 0 y las 5 hrs. En el caso de los sujetos para los que tenemos ambas mediciones podemos notar que estas son similares entre ambos tratamientos.

Para comprobar bioequivalencia la FDA solicita que el medicamento de prueba tenga un comportamiento similar al del medicamento de referencia. Para ello se compara la máxima concentración de medicamento (CMAX), el tiempo que tarda el individuo en alcanzar las concentración máxima en sangre (TMAX), y el área bajo la curva de concentración contra tiempo (AUC, que acabas de graficar). En particular para que la FDA concluya bioequivalencia se requiere que para cada medición (CMAX, TMAX y AUC) un intervalo de 90% de confianza para el cociente μ_T/μ_R de la media del tratamiento de prueba μ_T y la media de la referencia μ_R esté contenido en el intervalo $(80\%, 125\%)$. En este ejercicio usarás bootstrap para calcular un intervalo de confianza para el cociente de las medias de AUC.

3. Calcula el AUC para cada individuo en cada tratamiento disponible, usa la fórmula de área trapezoidal:

$$AUC = \sum_{\tau=1}^{k} (c_\tau + c_{\tau-1}) \times (t_\tau - t_{\tau-1})/2$$

donde c es la concentración y t son las horas. Tip: Si usas las funciones de dplyr puede resultar útil usar `lag()`. Presenta una tabla con tus resultados.

```
AUC<-cime_larga %>%
  group_by(formulation, subj) %>%
  summarise(auc_i = sum((MPC + lag(MPC,default = 0))*
                        (time1 - lag(time1,default = 0)))/2)
```

```
## `summarise()` regrouping output by 'formulation' (override with `.groups` argument)
```

```
head(AUC)
```

```
## # A tibble: 6 x 3
## # Groups:   formulation [1]
##   formulation subj  auc_i
##   <chr>       <fct> <dbl>
## 1 R           2      12.9
## 2 R           4      13.1
## 3 R           6      18.5
## 4 R           7      20.5
## 5 R           10     18.1
## 6 R           11     16.8
```

```
tail(AUC)
```

```
## # A tibble: 6 x 3
## # Groups:   formulation [1]
##   formulation subj  auc_i
##   <chr>       <fct> <dbl>
## 1 T           19    17.6
## 2 T           20    17.8
## 3 T           21    13.3
## 4 T           22    10.3
## 5 T           23     9.67
## 6 T           24    19.9
```

En estos últimos dos incisos usa los datos cimeditine_boot.csv, para el i-ésimo individuo `subj` tienes mediciones de AUC bajo tratamiento y referencia, denotemos a este par $x_i = (auc_{Ti}, auc_{Ri})$, suponiendo que las x_i se obuvieron de manera aleatoria de una distribución bivariada P, entonces la cantidad poblacional de interés es el parámetro $\theta = \mu_T/\mu_R$. Calcula el estimador *plug-in* de θ.

4. Usa bootstrap para generar un intervalo del 90% de confianza para θ, ¿la nueva tableta cumple el criterio de bioequivalencia de la FDA?

```
cime_ancha<-cime_auc%>%
  dplyr::select(Formulation,Subject,AUC)%>%
  pivot_wider(names_from = Formulation,values_from=AUC)
```

El estimador de interés es

```
theta <- cime_ancha %>%
  summarise(mu_t = mean(T),mu_r = mean(R)) %>%
  summarise(theta = mu_t/mu_r) %>%
  pull(theta)

estimador_razon <- function(split,...){
  muestra <- analysis(split)
  muestra %>%
    summarise(mu_t = mean(T), mu_r = mean(R)) %>%
    summarise(estimate = mu_t/mu_r) %>%
    mutate(term = "estimador de razón")
}
```

```r
dist_boot <- bootstraps(cime_ancha, 1000)%>%
  mutate(res_boot = map(splits, estimador_razon))

dist_boot %>% int_pctl(res_boot, 0.10)
```

```
## # A tibble: 1 x 6
##   term                .lower .estimate .upper .alpha .method
##   <chr>                <dbl>     <dbl>  <dbl>  <dbl> <chr>
## 1 estimador de razón  0.937     0.992   1.05    0.1 percentile
```

Con una probabilidad del 90 por ciento el tratamiento está contenido en el intervalo (94% y 104%). Sí está contenido en el intervalo de la FDA

2. Tráfico

La base de datos *amis* (publicada por G. Amis) contiene información de velocidades de coches en millas por hora, las mediciones se realizaron en caminos de Cambridgeshire, y para cada ubicación se realizan mediciones en dos sitios, en uno de estos sitios se situó una señal de alerta (de dismunición de velocidad). Mas aún, las mediciones se realizaron en dos ocasiones, antes y después de que se instalara la señal de alerta. La cantidad de interés es el cambio medio relativo de velocidad en el cuantil 0.85. Se eligió esta cantidad porque el objetivo de la señal de alerta es disminuir la velocidad de los conductores más veloces.

Variables: + speed: velocidad de los autos en mph, + period: periodo en que se hicieron las mediciones 1 indica antes de la señal, 2 cuando ya había señal, + pair: carretera en que se hizo la medición (1,2,5,7,8,9,10,11,12,13,14), + warning: si se colocó señal de alerta en el sitio 1 indica que si había señal, 2 que no había.

a) ¿Las observaciones conforman una muestra aleatoria? Explica tu respuesta y en caso de ser negativa explica la estructura de los datos.

Respuesta: La muestra es aleatoria: en las ubicaciones en las que se realizaron las observaciones pasan un número indeterminado de autos, por lo que, al hacer los registros en un tiempo determinado se considera como una muestra aleatoria.

```r
amis<-read.csv("amis.csv",header=FALSE,skip=1)
colnames(amis)<-c("n","speed","period","warning","pair")
```

b) El estadístico de interés se puede escribir como

$$\eta = \frac{1}{m}\sum[(\eta_{a1} - \eta_{b1}) - (\eta_{a0} - \eta_{b0})]$$

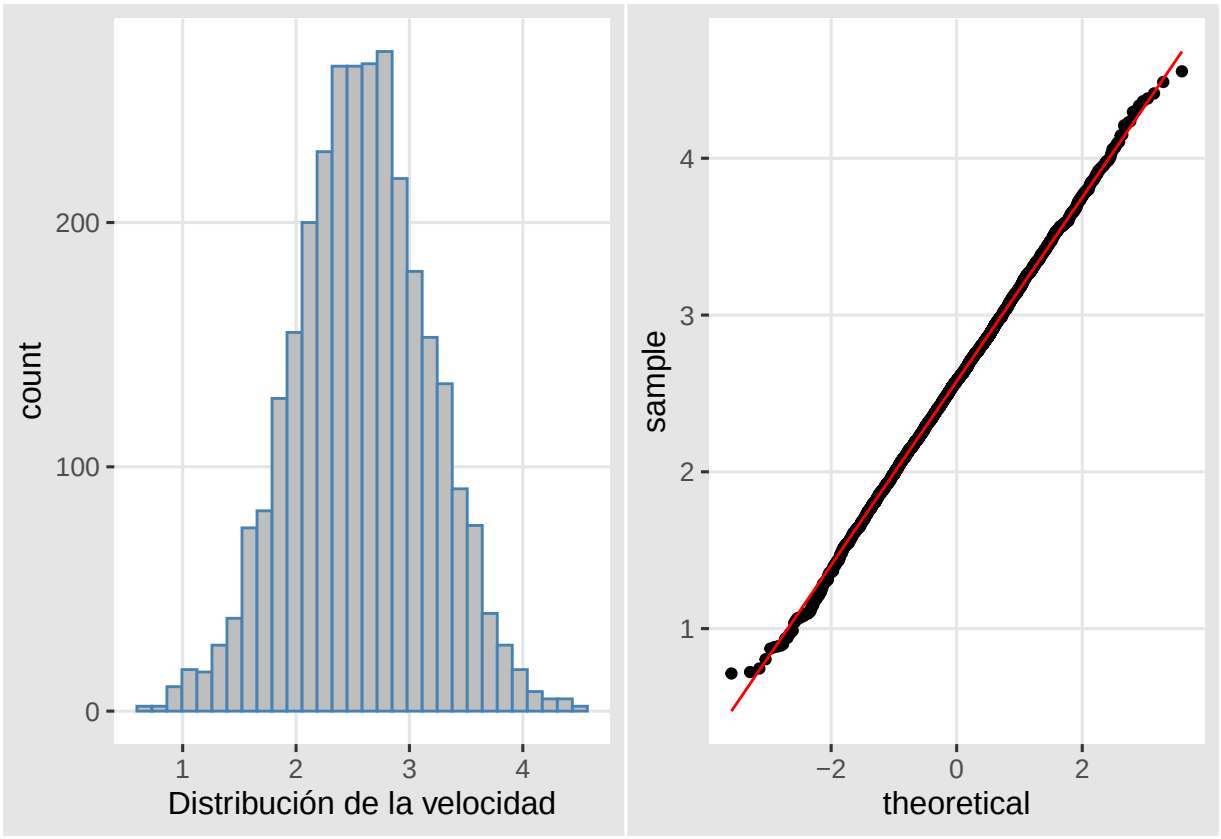
donde η_{a1}, η_{b1} corresponden a los cuartiles 0.85 de la distribución de velocidad en los sitios en los que se colocó la señal de alerta, (a corresponde a las mediciones antes de la señal y b después) y η_{a0}, η_{b0} son los correspondientes para los sitios sin alerta, m denota el número de carreteras. Calcula el estimador *plug-in* de η.

```r
#El estimador plug-in
eta <- amis %>%
  group_by(pair, period, warning) %>%
  summarise(eta_ab = quantile(speed,probs=0.85)) %>% #cuartiles
  group_by(pair) %>%
```

```
  group_by(warning) %>%
  mutate(resta = eta_ab-lead(eta_ab)) %>% # sacamos las diferencias
  ungroup() %>%
  filter(period == 1) %>%
  mutate(resta2 = resta-lead(resta)) %>%
  ungroup() %>%
  filter(warning == 1) %>%
  summarise(eta = mean(resta2)) %>% # promedio de las diferencias
  pull(eta)
```

c) Genera $B = 3000$ replicaciones bootstrap de η y realiza un histograma. Generamos 3000 replicaciones, esto en una función

```
# vamos a estimar eta en una función
estimador_eta <- function(split,...){
  muestra <- analysis(split)
  muestra %>%
    group_by(pair,period,warning) %>%
    summarise(eta_ab = quantile(speed,probs = 0.85), .groups = "drop") %>%
    group_by(pair) %>%
    group_by(warning) %>%
    mutate(resta = eta_ab - lead(eta_ab)) %>%
    ungroup() %>%
    filter(period == 1) %>%
    mutate(resta2 = resta - lead(resta)) %>%
    ungroup() %>%
    filter(warning == 1) %>%
    summarise(estimate = mean(resta2), .groups="drop") %>%
    mutate(term = "eta estimado")
}

#distribuciones bootstrap
dist_boot <- bootstraps(amis, 3000) %>%
  mutate(res_boot = map(splits, estimador_eta))


g_1 <- ggplot(dist_boot %>% unnest(res_boot), aes(x = estimate)) +
  geom_histogram(colour = "steelblue", fill = "gray", bins = 30) +
    xlab("Distribución de la velocidad") +
  theme_igray()

g_2 <- ggplot(dist_boot %>% unnest(res_boot), aes(sample = estimate)) +
  geom_qq() + geom_qq_line(colour = 'red') +
  theme_igray()

g_1+g_2
```



Se ven normales, hacemos los intervalos normales

d) Genera intervalos de confianza usando la aproximación normal y percentiles. Comparalos y en caso de encontrar diferencias explica a que se deben.

```
# d intervalos normales

eta_IC <- dist_boot %>%
  unnest(res_boot) %>%
  summarise(sd = sd(estimate)) %>%
  mutate(izq = eta - 2 * sd, der = eta + 2 * sd)
eta_IC
```

```
## # A tibble: 1 x 3
##      sd   izq   der
##   <dbl> <dbl> <dbl>
## 1 0.594  1.20  3.58
```

```
# d ahora los intervalos de percentiles

dist_boot %>% int_pctl(res_boot, alpha = 0.05) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  dplyr::select(term, .lower, .upper)
```

```
## # A tibble: 1 x 3
##   term         .lower .upper
```

```
##   <chr>          <dbl>  <dbl>
## 1 eta estimado     1.4   3.74
```

Consideramos que los intervalos son similares, por lo tanto, considerando el intervalo normal con una probabilidad del 95%, el verdadero valor de η se encuentra entre 1.2 y 3.6.

3. Cobertura de intervalos

En este problema realizarás un ejercicio de simulación para comparar la exactitud de distintos intervalos de confianza. Simularás muestras de
una distribución Poisson con parámetro $\lambda = 2.5$ y el estadístico de interés
es $\theta = exp(-2\lambda)$.

Sigue el siguiente proceso:

i) Genera una muestra aleatoria de tamaño $n = 60$ con distribución $Poisson(\lambda)$, parámetro $\lambda = 2.5$ (en R usa la función `rpois()`).

ii) Genera $5,000$ muestras bootstrap y calcula intervalos de confianza del 95% para $\hat{\theta}$ usando 1) el método normal y 2) percentiles.

iii) Revisa si el intervalo de confianza contiene el verdadero valor del parámetro ($\theta = exp(-2 \cdot 2.5)$), en caso de que no lo contenga registra si falló por la izquierda (el límite inferior mayor $exp(-2.5 * \lambda)$) o falló por la derecha (el límite superior menor $exp(-2.5 * \lambda)$).

```
theta <- exp(-2*2.5)
fx_get_interv <- function(iter = 1, tam_muest = 60){

  ## --> Random sample, bootstrap sample
  muest_pois <- tibble(pois_val = rpois(tam_muest, lambda = 2.5))
  estim_theta <- exp(-2*mean(muest_pois$pois_val))
  ## --> Estimator function
  estad_theta <- function(data){
    valor <- exp(-2*mean(data$pois_val))
    valor
    }
  ## --> Re-sample with replacement
  remues_boot <- function(muest_pois){
    sample_n(tbl = muest_pois, size = tam_muest, replace = T)
    }
  ## --> Estimate process
  ## --> Estimate process
  dist_muestral<- map_dbl(seq(1,5000),~estad_theta(remues_boot(muest_pois))) %>%
    tibble(boot_val = .) %>%
    summarise(hat_ee_boot = sd(boot_val),
              hat_ee_q025 = round(quantile(boot_val, probs = 0.025),6),
hat_ee_q975 = round(quantile(boot_val, probs = 0.975),6))
resultado <- tibble(iter = c(iter, iter)) %>%
mutate(method = c('Normal', 'Percentiles'),
l_inf = c(estim_theta-2*dist_muestral$hat_ee_boot,
dist_muestral$hat_ee_q025),
l_sup = c(estim_theta+2*dist_muestral$hat_ee_boot,
dist_muestral$hat_ee_q975),
f_lft = ifelse(theta < l_inf,1,0),
f_rgt = ifelse(theta > l_sup,1,0),
```

Table 5: Intervalos de Confianza al 95Simul. Bootstrap con muestra de tamaño 60

method	prcnt_fallo_izq	prcnt_fallo_der	Cobertura	LongProm
Normal	0.001	0.064	0.935	0.012786
Percentiles	0.032	0.031	0.937	0.012283

```
cover = ifelse(f_lft + f_rgt == 0,1,0),
lngth = c(4*dist_muestral$hat_ee_boot, dist_muestral$hat_ee_q975 -
dist_muestral$hat_ee_q025)

 )
resultado

}

conf_int_s60 <- map_df(1:1000, ~ fx_get_interv(iter = .x , tam_muest = 60))
conf_int_s300 <- map_df(1:1000, ~ fx_get_interv(iter = .x , tam_muest = 300))
```

a) Repite el proceso descrito 1000 veces y llena la siguiente tabla:

Método	% fallo izquierda	% fallo derecha	Cobertura	Longitud promedio
Normal				
Percentiles				

La columna cobertura es una estimación de la cobertura del intervalo basada en las simulaciones, para calcularla simplemente escribe el porcentaje de los intervalos que incluyeron el verdadero valor del parámetro. La longitud promedio es la longitud promedio de los intervalos de confianza bajo cada método.

```
tbl_int_s60 <- conf_int_s60 %>% group_by(method) %>%
  dplyr::select(method, f_lft, f_rgt, cover, lngth) %>%
  summarise(prcnt_fallo_izq = mean(f_lft), prcnt_fallo_der =  mean(f_rgt),
            Cobertura = mean(cover), LongProm = mean(lngth)) %>%
  mutate(across(where(is.numeric),round,6))
```

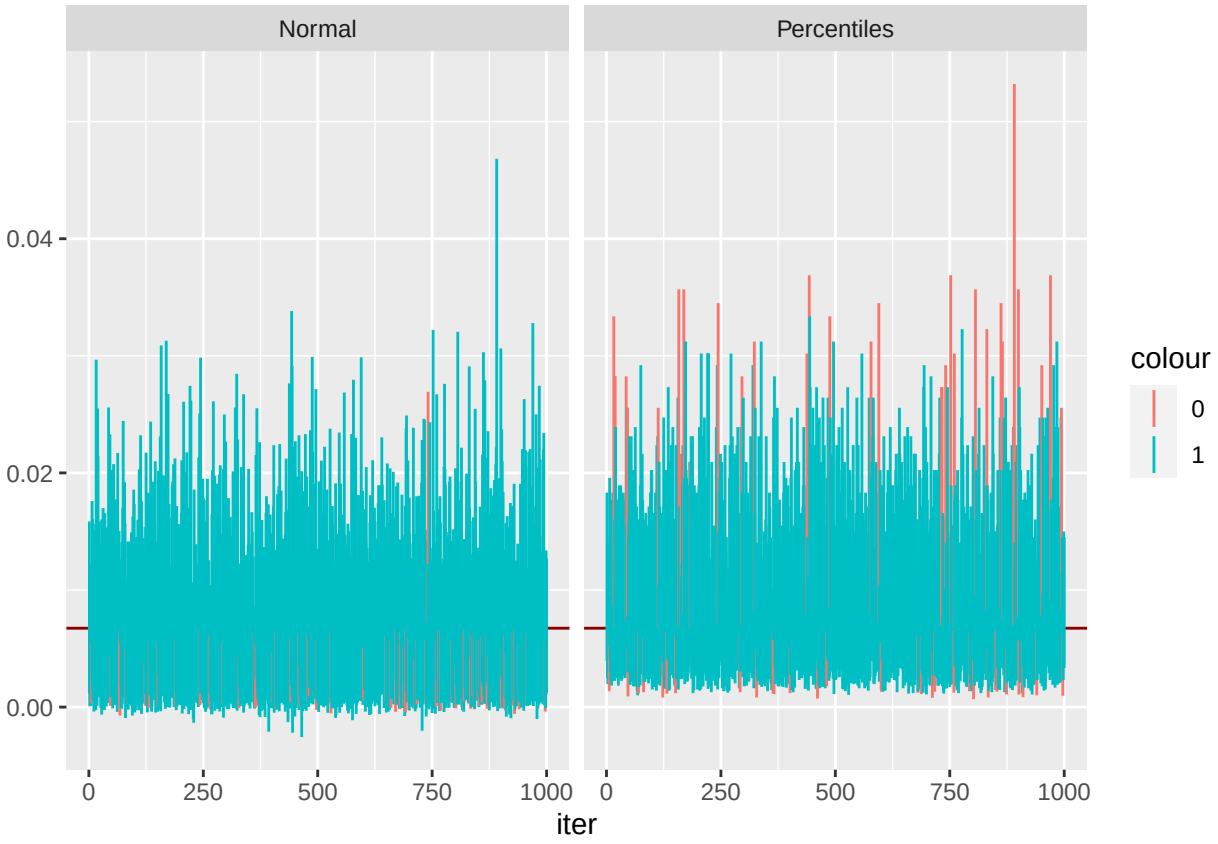
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## Format table
kbl(tbl_int_s60, caption = "Intervalos de Confianza al 95%,
     Simul. Bootstrap con muestra de tamaño 60",
    align = 'c') %>%
  kable_styling(bootstrap_options = c("striped"), full_width = F)
```

b) Realiza una gráfica de páneles, en cada panel mostrarás los resultados de uno de los métodos (normal, percentiles), en el vertical graficarás los límites de los intervalos.

```
conf_int_s60 <- conf_int_s60 %>%
  mutate(cobertura = factor(cover))

ggplot(data = conf_int_s60, aes(x = iter, col = 'red')) +
```

```
  geom_hline(yintercept = theta, col = "darkred") +
  geom_linerange(aes(ymin = l_inf, ymax = l_sup, colour = cobertura)) +
  facet_wrap(~method)
```



c) Repite los incisos a) y b) seleccionando muestras de tamaño 300.

Intervalos de Confianza, tabla resultado de simulación con muestra de tamaño 300

```
tbl_int_s300 <- conf_int_s300 %>% group_by(method) %>%
  dplyr::select(method, f_lft, f_rgt, cover, lngth) %>%
  summarise(prcnt_fallo_izq = mean(f_lft), prcnt_fallo_der =  mean(f_rgt),
            Cobertura = mean(cover), LongProm = mean(lngth)) %>%
  mutate(across(where(is.numeric),round,6))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

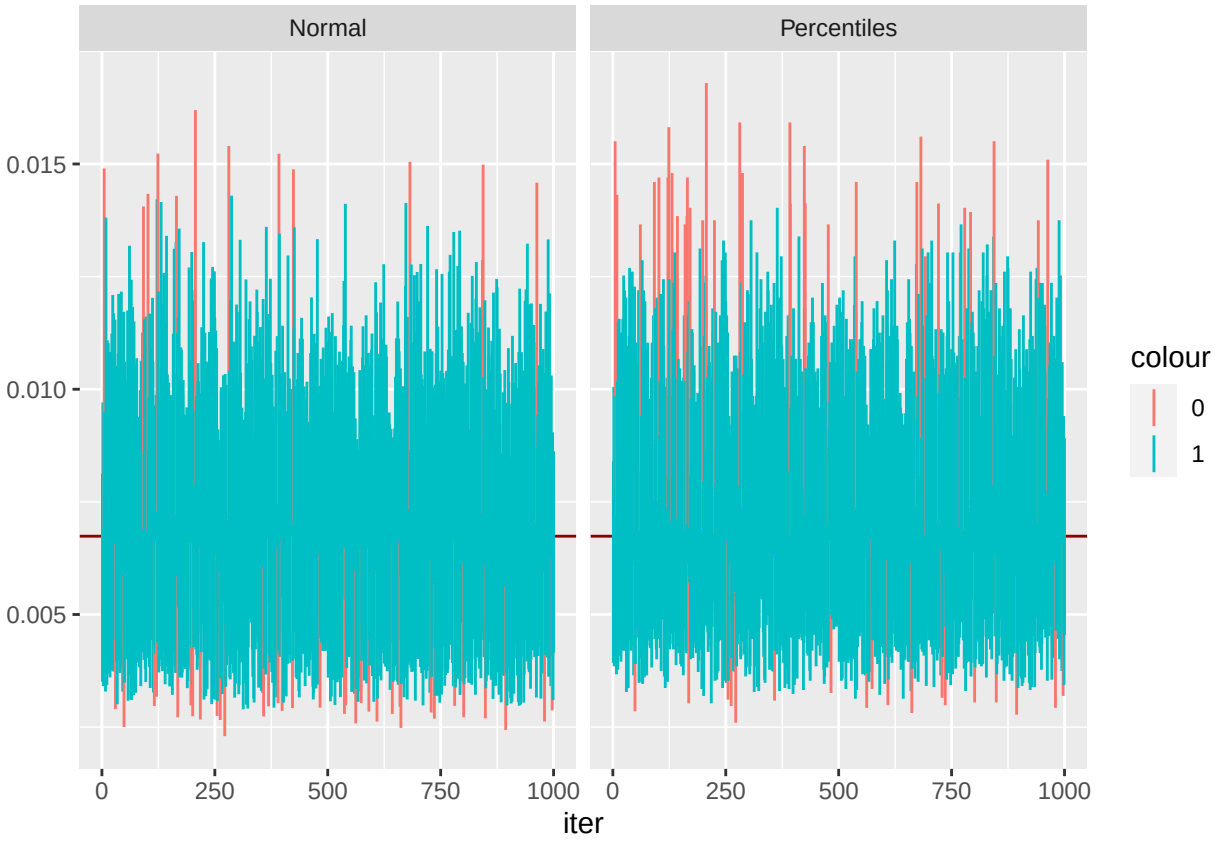
```
## Format table

kbl(tbl_int_s300, caption = "Intervalos de Confianza al 95%,
     Simul. Bootstrap con muestra de tamaño 300",
    align = 'c') %>%
  kable_styling(bootstrap_options = c("striped"), full_width = F)
```

Intervalos de Confianza, paneles simulación con muestra de tamaño 300

Table 6: Intervalos de Confianza al 95Simul. Bootstrap con muestra de tamaño 300

method	prcnt_fallo_izq	prcnt_fallo_der	Cobertura	LongProm
Normal	0.012	0.034	0.954	0.005156
Percentiles	0.034	0.018	0.948	0.005029

```
conf_int_s300 <- conf_int_s300 %>%
  mutate(cobertura = factor(cover))

ggplot(data = conf_int_s300, aes(x = iter, col = 'red')) +
  geom_hline(yintercept = theta, col = "darkred") +
  geom_linerange(aes(ymin = l_inf, ymax = l_sup, colour = cobertura)) +
  facet_wrap(~method)
```



Al iniciar a resolver el ejercicio (3), nos causó confusión el hecho de que al graficar la distribución de muestreo, ésta mostraba una forma sesgada a la derecha que iba un poco en diferencia de todos los ejercicios que habíamos realizado en clase de muestreo y remuestreo de estadísticos que presentaban un comportamiento "normal". Después de analizarlo, disipamos la duda partiendo de que un estadístico es una métrica (característica numérica) de los datos, de la muestra que se busca analizar estimar un parámetro.

Con esto mente, se construye una función que, de forma simultánea, realice un muestreo bootstrap a la cual sea posible indicarle el tamaño de la muestra y el número de veces a repetir el proceso y calcule los intervalos de confianza. Se realizaron dos ejercicios, con muestras de la población de tamaño 60 y tamaño 300.

Concluimos lo siguiente:

- La longitud promedio de los intervalos para la muestra y remuestras de tamaño 60 es más amplio que la de 300.
- En ambos casos, los niveles de cobertura son muy similares tanto para 60 como para 300. Sin embargo, cuando observamos los valores por método, se puede observar que el método `Normal` tiende a fallar más por la derecha que por la izquierda y que el método de percentiles está mejor centrado. Esto es consistente con lo que esperábamos al principio dado que el estadístico naturalmente está sesgado a la derecha.

Nota: Un ejemplo en donde la cantidad $P(X = 0)^2 = e^{-\lambda}$ es de interés es como sigue, las llamadas telefónicas a un conmutador se modelan con un proceso Poisson y λ es el número promedio de llamadas por minuto, entonces $e^{-\lambda}$ es la probabilidad de que no se reciban llamadas en 1 minuto.