Modelos Actuariales II - Tarea 1

Oscar Andrei Zempoalteca Ramírez 164880

Lee la base de datos pizza_delivery_tarea en R en un data frame. La base de datos está integrada por todos los pedidos de una pizzería que tiene 3 sucursales en un mes. Los pedidos son recibidos en una central telefónica y son asignadas a la sucursal más cercana a la dirección de entrega. La pizzería tiene la política de que si el pedido tarda más de 40 min en entregarse al cliente se le regala una botella de vino (aunque hay ocasiones en la que la botella no es entregada). Tiene las siguientes variables:

- Día de la semana
- Fecha
- Tiempo que tardó en entregarse la pizza
- Nombre de quien recibió el pedido
- Sucursal
- Repartidor
- Temperatura a la que fue entregada la pizza
- Valor de la cuenta
- Número de pizzas ordenadas
- Vino de regalo (se regala si la pizza tarda más de 40 min en llegar)
- Indicadora si se entregó la botella de vino
- Indicadora si el cliente usó un cupón de descuento

```
pizza_delivery_tarea <- read.csv("C:/Users/HP/Downloads/pizza_delivery_tarea (1).csv")
dato<-pizza_delivery_tarea
names(dato)
```

```
##  [1] "X"                "day"              "date"
##  [4] "time"             "operator"         "branch"
##  [7] "driver"           "temperature"      "bill"
## [10] "pizzas"           "free_wine"        "got_wine"
## [13] "discount_customer"
```

En la base de datos encuentras 4 “NA”, corrige la base de datos con datos que tengan sentido en el contexto de la base de datos. Explica la corrección de cada uno.

```
dato[is.na(dato$X),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$day),]
```

```
##     X  day      date     time operator branch driver temperature bill pizzas
## 10 10 <NA> 01-May-14 37.95479  Melissa Centre  Bruno    60.00738 44.8      5
##    free_wine got_wine discount_customer
## 10         0    FALSE                 0
```

El día faltante puede ser facilmente corregido pues la fecha viene en la columna de la derecha

```
dato[10,2]<-"Thursday"
```

```
dato[is.na(dato$date),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$time),]
```

```
##   X      day      date time operator branch driver temperature bill pizzas
## 6 6 Thursday 01-May-14   NA  Melissa Centre  Bruno     60.7595 61.8      4
##   free_wine got_wine discount_customer
## 6         1     TRUE                 0
```

Para corregir el NA en “time” notemos que para esa observación “free wine” es 1 y para que sea más preciso,también filtremos aquellos tiempos que provienen del “Branch” Center así que saquemos el promedio de todos los tiempos que cumplen estas condiciones y este valor será el que reemplace el NA.

```
dato[6,4]<-mean(dato[dato$branch=="Centre"&dato$free_wine==1&!is.na(dato$time),"time"])
dato[6,]
```

```
##   X      day      date     time operator branch driver temperature bill pizzas
## 6 6 Thursday 01-May-14 43.67683  Melissa Centre  Bruno     60.7595 61.8      4
##   free_wine got_wine discount_customer
## 6         1     TRUE                 0
```

```
dato[is.na(dato$time),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$operator),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$branch),]
```

```
##     X      day      date     time operator branch driver temperature bill
## 15 15 Thursday 01-May-14 23.78428    Laura   <NA>  Bruno    64.62646 25.9
##    pizzas free_wine got_wine discount_customer
## 15      1         0    FALSE                 0
```

Para corregir este NA, saquemos una muestra tamaño 1 de todos los branch en los cuales Bruno fue conductor

```
dato[15,6]<-sample(dato[dato$driver=="Bruno"&!is.na(dato$branch),"branch"],size = 1)
dato[15,]
```

```
##     X      day      date     time operator branch driver temperature bill
## 15 15 Thursday 01-May-14 23.78428    Laura Centre  Bruno    64.62646 25.9
##    pizzas free_wine got_wine discount_customer
## 15      1         0    FALSE                 0
```

```
dato[is.na(dato$driver),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$temperature),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$bill),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$pizzas),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$free_wine),]
```

```
##   X      day      date     time operator branch driver temperature bill pizzas
## 7 7 Thursday 01-May-14 48.72861    Laura   West  Bruno     58.2587 57.9      3
##   free_wine got_wine discount_customer
## 7        NA     TRUE                 0
```

Notemos que “got_wine” tiene el valor de TRUE, con esta información podemos conocer el valor faltante de “free_wine”

```
dato[7,11]<-1
dato[7,]
```

```
##   X      day      date     time operator branch driver temperature bill pizzas
## 7 7 Thursday 01-May-14 48.72861    Laura   West  Bruno     58.2587 57.9      3
##   free_wine got_wine discount_customer
## 7         1     TRUE                 0
```

```
dato[is.na(dato$free_wine),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$got_wine),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

```
dato[is.na(dato$discount_customer),]
```

```
##  [1] X                day              date             time
##  [5] operator         branch           driver           temperature
##  [9] bill             pizzas           free_wine        got_wine
## [13] discount_customer
## <0 rows> (or 0-length row.names)
```

1. ¿Tipo de cada una de las variables? Utiliza class() en cada variable

```
class(dato$day)
```

```
## [1] "character"
```

```
class(dato$date)
```

```
## [1] "character"
```

```
class(dato$time)
```

```
## [1] "numeric"
```

```
class(dato$operator)
```

```
## [1] "character"
```

```
class(dato$branch)
```

```
## [1] "character"
```

```
class(dato$branch)
```

```
## [1] "character"
```

```
class(dato$driver)
```

```
## [1] "character"
```

```
class(dato$temperature)
```

```
## [1] "numeric"
```

```
class(dato$bill)
```

```
## [1] "numeric"
```

```
class(dato$pizzas)
```

```
## [1] "integer"
```

```
class(dato$free_wine)
```

```
## [1] "numeric"
```

```
class(dato$got_wine)
```

```
## [1] "logical"
```

```
class(dato$discount_customer)
```

```
## [1] "integer"
```

```
sapply(dato,class)
```

```
##                 X               day              date              time
##         "integer"       "character"       "character"         "numeric"
##          operator            branch            driver       temperature
##       "character"       "character"       "character"         "numeric"
##              bill            pizzas         free_wine          got_wine
##         "numeric"         "integer"         "numeric"         "logical"
## discount_customer
##         "integer"
```

2. Transforma la variable date en tipo date (si no está en ese tipo).

Primero llamemos los paquetes que utilizaremos más adelante:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
library(moments)
```

```
## Warning: package 'moments' was built under R version 4.0.3
```

```
library(infotheo)
```

```
## Warning: package 'infotheo' was built under R version 4.0.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.5     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## Warning: package 'purrr' was built under R version 4.0.3
```

```
## Warning: package 'stringr' was built under R version 4.0.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
Sys.setlocale(locale="Spanish")
```

```
## [1] "LC_COLLATE=Spanish_Spain.1252;LC_CTYPE=Spanish_Spain.1252;LC_MONETARY=Spanish_Spain.125
2;LC_NUMERIC=C;LC_TIME=Spanish_Spain.1252"
```

```
class(dato$date)
```

```
## [1] "character"
```

```
head(dato)
```

```
##   X      day      date     time operator branch    driver temperature bill
## 1 1 Thursday 01-May-14 35.12837    Laura   East     Bruno    68.28772 58.4
## 2 2 Thursday 01-May-14 25.20307  Melissa   East Salvatore    70.99779 26.4
## 3 3 Thursday 01-May-14 45.64340  Melissa   West Salvatore    53.39415 58.1
## 4 4 Thursday 01-May-14 29.37430  Melissa   East Salvatore    70.30660 35.2
## 5 5 Thursday 01-May-14 29.99461  Melissa   West Salvatore    71.50169 38.4
## 6 6 Thursday 01-May-14 43.67683  Melissa Centre     Bruno    60.75950 61.8
##   pizzas free_wine got_wine discount_customer
## 1      4         0    FALSE                 1
## 2      2         0    FALSE                 0
## 3      3         1    FALSE                 0
## 4      3         0    FALSE                 0
## 5      2         0    FALSE                 0
## 6      4         1     TRUE                 0
```

```
d5<-mutate(dato, date= as.Date(date, format="%d-%b-%y"))
class(d5$date)
```

```
## [1] "Date"
```

```
head(d5)
```

```
##   X      day       date     time operator branch    driver temperature bill
## 1 1 Thursday 2014-05-01 35.12837    Laura   East     Bruno    68.28772 58.4
## 2 2 Thursday 2014-05-01 25.20307  Melissa   East Salvatore    70.99779 26.4
## 3 3 Thursday 2014-05-01 45.64340  Melissa   West Salvatore    53.39415 58.1
## 4 4 Thursday 2014-05-01 29.37430  Melissa   East Salvatore    70.30660 35.2
## 5 5 Thursday 2014-05-01 29.99461  Melissa   West Salvatore    71.50169 38.4
## 6 6 Thursday 2014-05-01 43.67683  Melissa Centre     Bruno    60.75950 61.8
##   pizzas free_wine got_wine discount_customer
## 1      4         0    FALSE                 1
## 2      2         0    FALSE                 0
## 3      3         1    FALSE                 0
## 4      3         0    FALSE                 0
## 5      2         0    FALSE                 0
## 6      4         1     TRUE                 0
```

Para transformar toda la variable usamos la función “mutate” y esto lo guardamos en el nuevo dataframe “d5”.

3. Si tienes alguna variable lógica, transfórmala a numérica y verifica que tenga sentido.

Ya habíamos visto que la única lógica es “got_wine”, asignaremos un 1 a TRUE y 0 a FALSE

```
class(d5$got_wine)
```

```
## [1] "logical"
```

```
d5$got_wine<-as.numeric(d5$got_wine)
class(d5$got_wine)
```

```
## [1] "numeric"
```

```
d5$got_wine
```

```
##    [1] 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
##   [38] 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0
##   [75] 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##  [112] 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
##  [149] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
##  [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
##  [223] 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1
##  [260] 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1
##  [297] 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0
##  [334] 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
##  [371] 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
##  [408] 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [445] 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [482] 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
##  [519] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [556] 0 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
##  [593] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [630] 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
##  [667] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
##  [704] 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1
##  [741] 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##  [778] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
##  [815] 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0
##  [852] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
##  [889] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1
##  [926] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [963] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0
## [1000] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
## [1037] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
## [1074] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [1111] 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 0 0 0
## [1148] 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0
## [1185] 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
## [1222] 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
## [1259] 0 0 1 0 0 0 0 0
```

4. Crea una nueva variable que se llame quejas: variable indicadora de si hubo una queja o no. Se presenta una queja cuando la temperatura de la pizza en la entrega es menor a los 60ªC o si tardó más de 40min y no se entregó la botella.

```
head(d5)
```

```
##   X      day       date     time operator branch    driver temperature bill
## 1 1 Thursday 2014-05-01 35.12837    Laura   East     Bruno    68.28772 58.4
## 2 2 Thursday 2014-05-01 25.20307  Melissa   East Salvatore    70.99779 26.4
## 3 3 Thursday 2014-05-01 45.64340  Melissa   West Salvatore    53.39415 58.1
## 4 4 Thursday 2014-05-01 29.37430  Melissa   East Salvatore    70.30660 35.2
## 5 5 Thursday 2014-05-01 29.99461  Melissa   West Salvatore    71.50169 38.4
## 6 6 Thursday 2014-05-01 43.67683  Melissa Centre     Bruno    60.75950 61.8
##   pizzas free_wine got_wine discount_customer
## 1      4         0        0                 1
## 2      2         0        0                 0
## 3      3         1        0                 0
## 4      3         0        0                 0
## 5      2         0        0                 0
## 6      4         1        1                 0
```

```
d6 <- d5 %>%
  mutate(quejas = if_else(temperature<60 |  (time>40 & got_wine==0), 1, 0))
head(d6)
```

```
##   X      day       date     time operator branch    driver temperature bill
## 1 1 Thursday 2014-05-01 35.12837    Laura   East     Bruno    68.28772 58.4
## 2 2 Thursday 2014-05-01 25.20307  Melissa   East Salvatore    70.99779 26.4
## 3 3 Thursday 2014-05-01 45.64340  Melissa   West Salvatore    53.39415 58.1
## 4 4 Thursday 2014-05-01 29.37430  Melissa   East Salvatore    70.30660 35.2
## 5 5 Thursday 2014-05-01 29.99461  Melissa   West Salvatore    71.50169 38.4
## 6 6 Thursday 2014-05-01 43.67683  Melissa Centre     Bruno    60.75950 61.8
##   pizzas free_wine got_wine discount_customer quejas
## 1      4         0        0                 1      0
## 2      2         0        0                 0      0
## 3      3         1        0                 0      1
## 4      3         0        0                 0      0
## 5      2         0        0                 0      0
## 6      4         1        1                 0      0
```

El resultado lo guardamos en un nuevo dataframe (d6) con la nueva variable “quejas”

5. ¿Existen diferencias significativas en el tiempo de entrega dependiendo del operador que tomó el pedido?

```
d6 %>% group_by(operator) %>% summarise(meantime=mean(time))
```

```
## # A tibble: 2 x 2
##   operator meantime
## * <chr>       <dbl>
## 1 Laura        34.1
## 2 Melissa      34.4
```

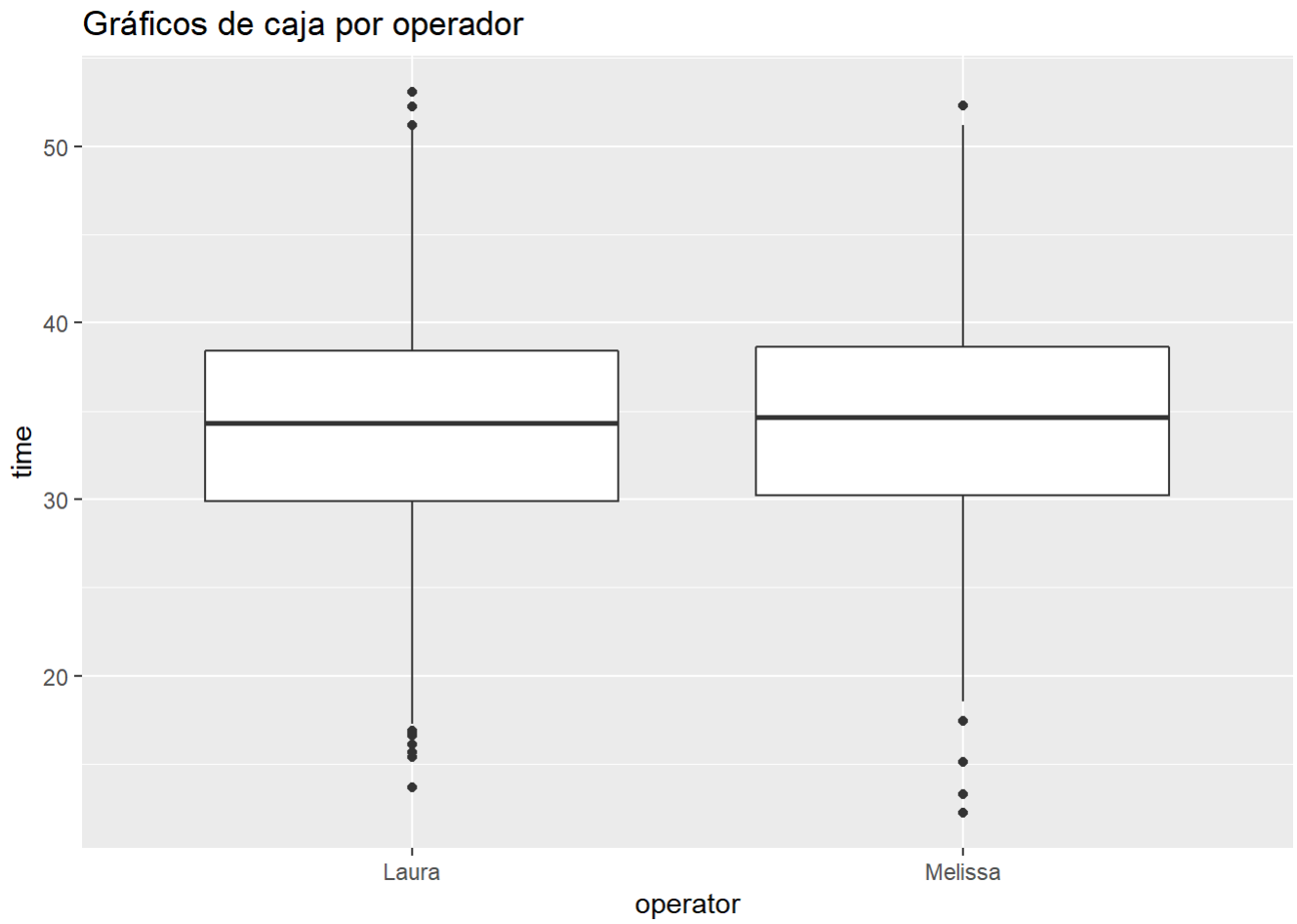
```
d6 %>% group_by(operator) %>% summarise(mintime=min(time),maxtime=max(time))
```

```
## # A tibble: 2 x 3
##   operator mintime maxtime
## * <chr>      <dbl>   <dbl>
## 1 Laura       13.7    53.1
## 2 Melissa     12.3    52.3
```

```
d6 %>% group_by(operator) %>% summarise(desvest=sd(time))
```

```
## # A tibble: 2 x 2
##   operator desvest
## * <chr>      <dbl>
## 1 Laura       6.59
## 2 Melissa     6.34
```

```
ggplot(d6,aes(x=operator,y=time))+geom_boxplot()+ggtitle("Gráficos de caja por operador")
```



Después de analizar lo anterior, no encontramos evidencia de la existencia de diferencias significativas entre operadores.

6. ¿Existen diferencias significativas en la temperatura de entrega dependiendo del repartidor? Justifica

```
d6 %>% group_by(driver) %>% summarise(meantemperature=mean(temperature))
```

```
## # A tibble: 5 x 2
##   driver    meantemperature
## * <chr>               <dbl>
## 1 Bruno                61.9
## 2 Domenico             68.4
## 3 Luigi                63.6
## 4 Mario                62.8
## 5 Salvatore            62.1
```

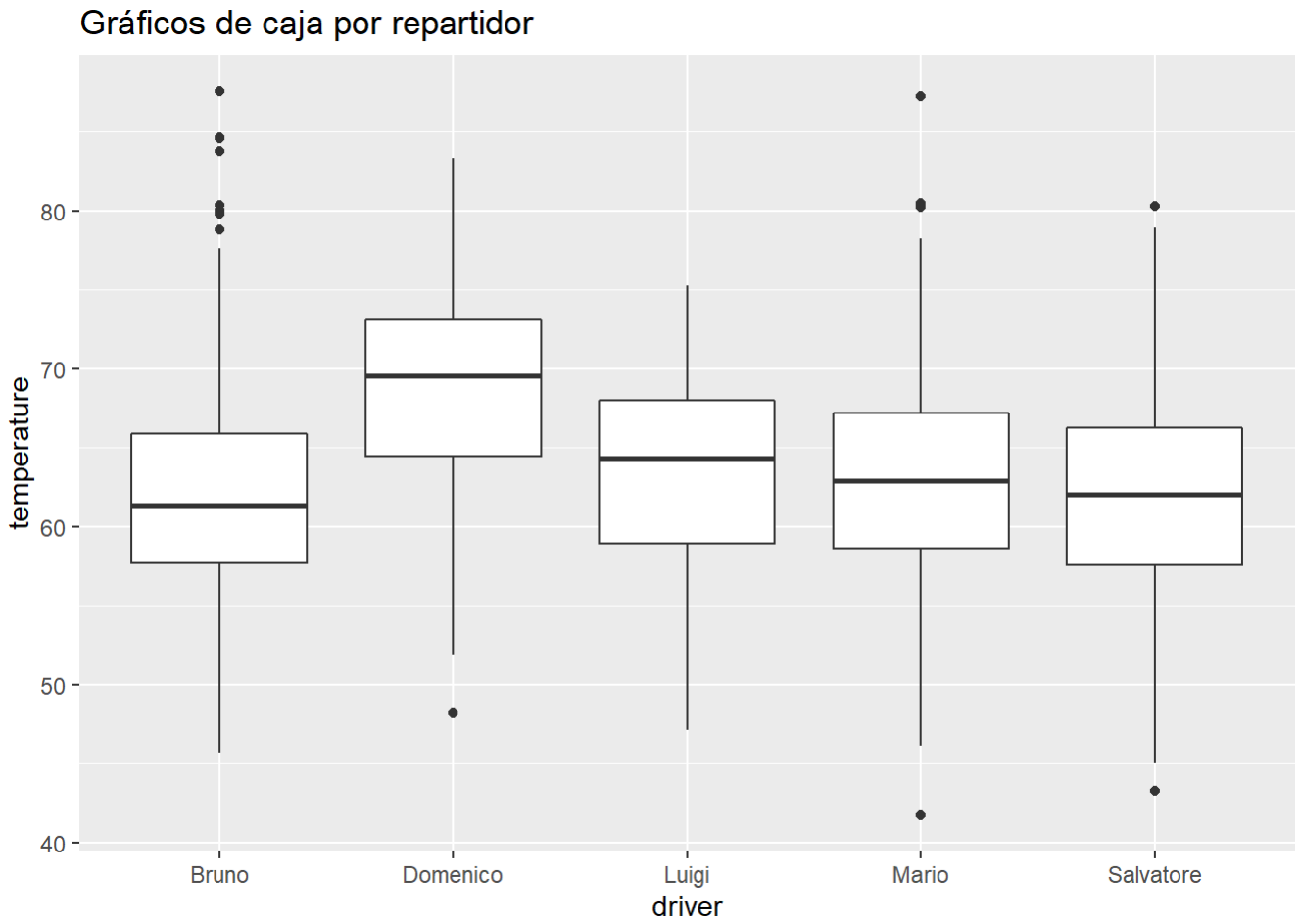
```
d6 %>% group_by(driver) %>% summarise(mintemperature=min(temperature),maxtemperature=max(tempera
ture))
```

```
## # A tibble: 5 x 3
##   driver    mintemperature maxtemperature
## * <chr>              <dbl>          <dbl>
## 1 Bruno               45.7           87.6
## 2 Domenico            48.2           83.4
## 3 Luigi               47.2           75.3
## 4 Mario               41.8           87.3
## 5 Salvatore           43.3           80.3
```

```
d6 %>% group_by(driver) %>% summarise(stdev=sd(temperature))
```

```
## # A tibble: 5 x 2
##   driver    stdev
## * <chr>     <dbl>
## 1 Bruno      6.75
## 2 Domenico   7.30
## 3 Luigi      6.74
## 4 Mario      6.65
## 5 Salvatore  6.57
```

```
ggplot(d6,aes(x=driver,y=temperature))+geom_boxplot()+ggtitle("Gráficos de caja por repartidor")
```



Sí encontramos diferencias significativas entre repartidores. Domenico mantuvo temperaturas en promedio superiores al resto y su temperatura mínima de entrega fue por lo menos 1 grado superior a sus compañeros. Los otros 3 repartidores no parecen mostrar diferencias significativas entre ellos.

7. ¿Existen diferencias significativas en el tiempo de entrega dependiendo de la sucursal?

```
d6 %>% group_by(branch) %>% summarise(meantime=mean(time))
```

```
## # A tibble: 3 x 2
##   branch meantime
## * <chr>     <dbl>
## 1 Centre     36.3
## 2 East       31.1
## 3 West       35.2
```

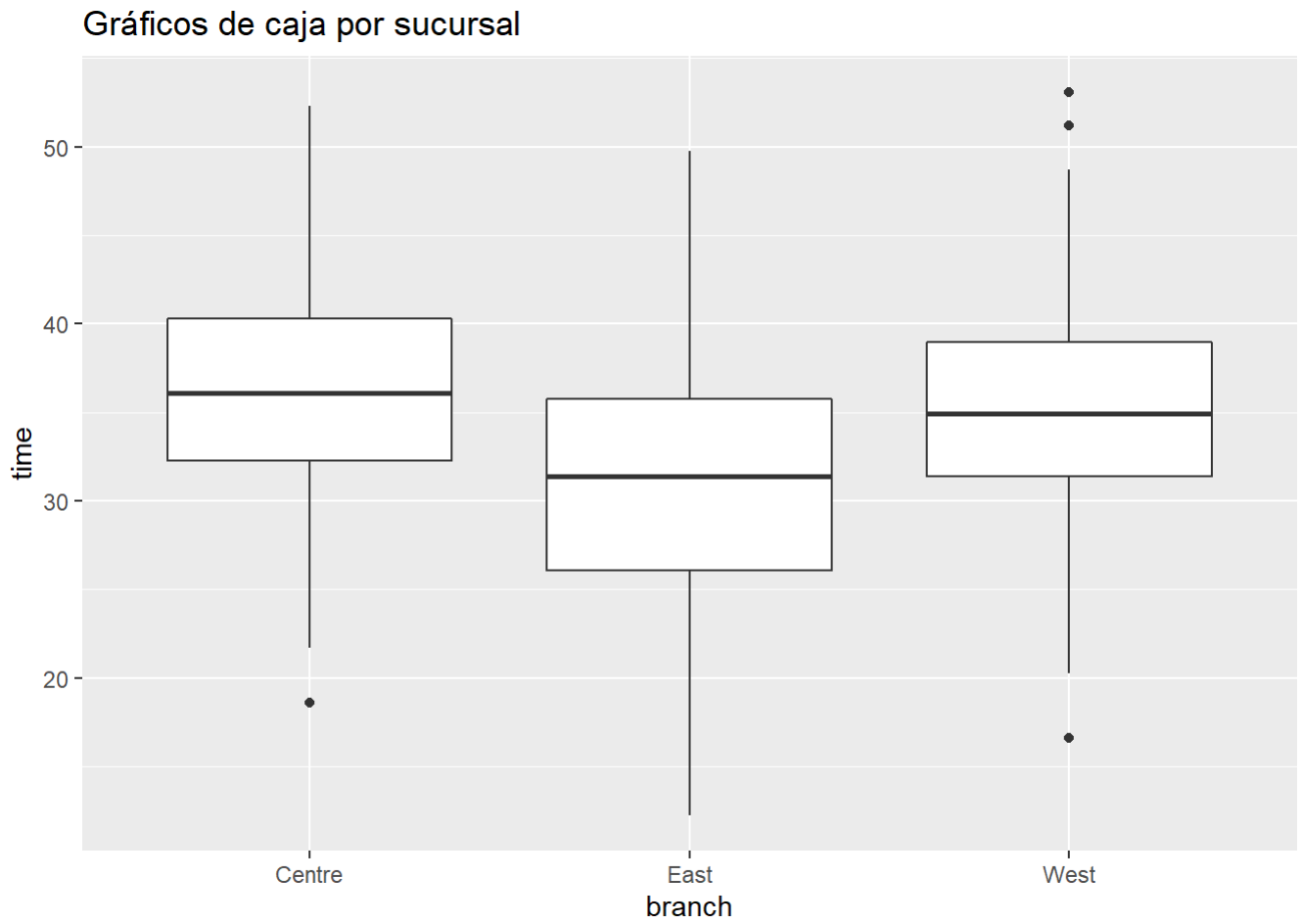
```
d6 %>% group_by(branch) %>% summarise(mintime=min(time),maxtime=max(time))
```

```
## # A tibble: 3 x 3
##   branch mintime maxtime
## * <chr>    <dbl>   <dbl>
## 1 Centre    18.6    52.3
## 2 East      12.3    49.7
## 3 West      16.6    53.1
```

```
d6 %>% group_by(branch) %>% summarise(stdev=sd(time))
```

```
## # A tibble: 3 x 2
##   branch stdev
## * <chr>  <dbl>
## 1 Centre  5.80
## 2 East    6.68
## 3 West    5.71
```

```
ggplot(d6,aes(x=branch,y=time))+geom_boxplot()+ggtitle("Gráficos de caja por sucursal")
```



Podemos encontrar diferencias significativas. East presenta tiempos promedio menores al resto además de generar el tiempo mínimo entre grupos y su tiempo máximo también está por debajo de los máximos de Centre y West. Además de East, las dos sucursales restantes presentan diferencias menos notorias aunque en general Centre presentó los tiempos de entregamás altos.

8. ¿Existen diferencias significativas en la temperatura de entrega dependiendo de la sucursal? Justifica

```
d6 %>% group_by(branch) %>% summarise(meantime=mean(temperature))
```

```
## # A tibble: 3 x 2
##   branch meantime
## * <chr>     <dbl>
## 1 Centre     60.3
## 2 East       66.6
## 3 West       61.8
```

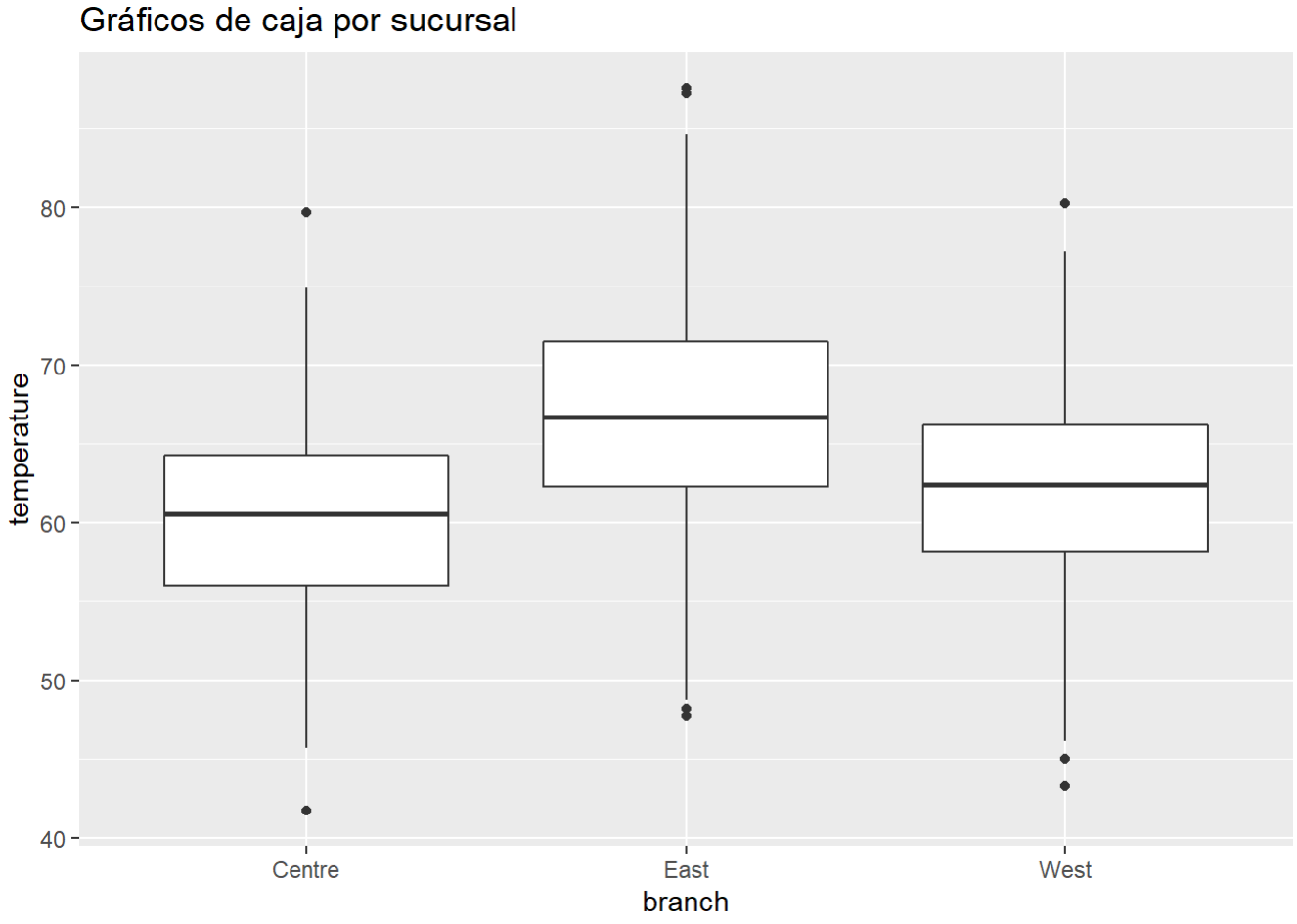
```
d6 %>% group_by(branch) %>% summarise(mintime=min(temperature),maxtime=max(temperature))
```

```
## # A tibble: 3 x 3
##   branch mintime maxtime
## * <chr>    <dbl>   <dbl>
## 1 Centre    41.8    79.7
## 2 East      47.7    87.6
## 3 West      43.3    80.2
```

```
d6 %>% group_by(branch) %>% summarise(stdev=sd(temperature))
```

```
## # A tibble: 3 x 2
##   branch stdev
## * <chr>  <dbl>
## 1 Centre  5.90
## 2 East    7.06
## 3 West    6.11
```

```
ggplot(d6,aes(x=branch,y=temperature))+geom_boxplot()+ggtitle("Gráficos de caja por sucursal")
```



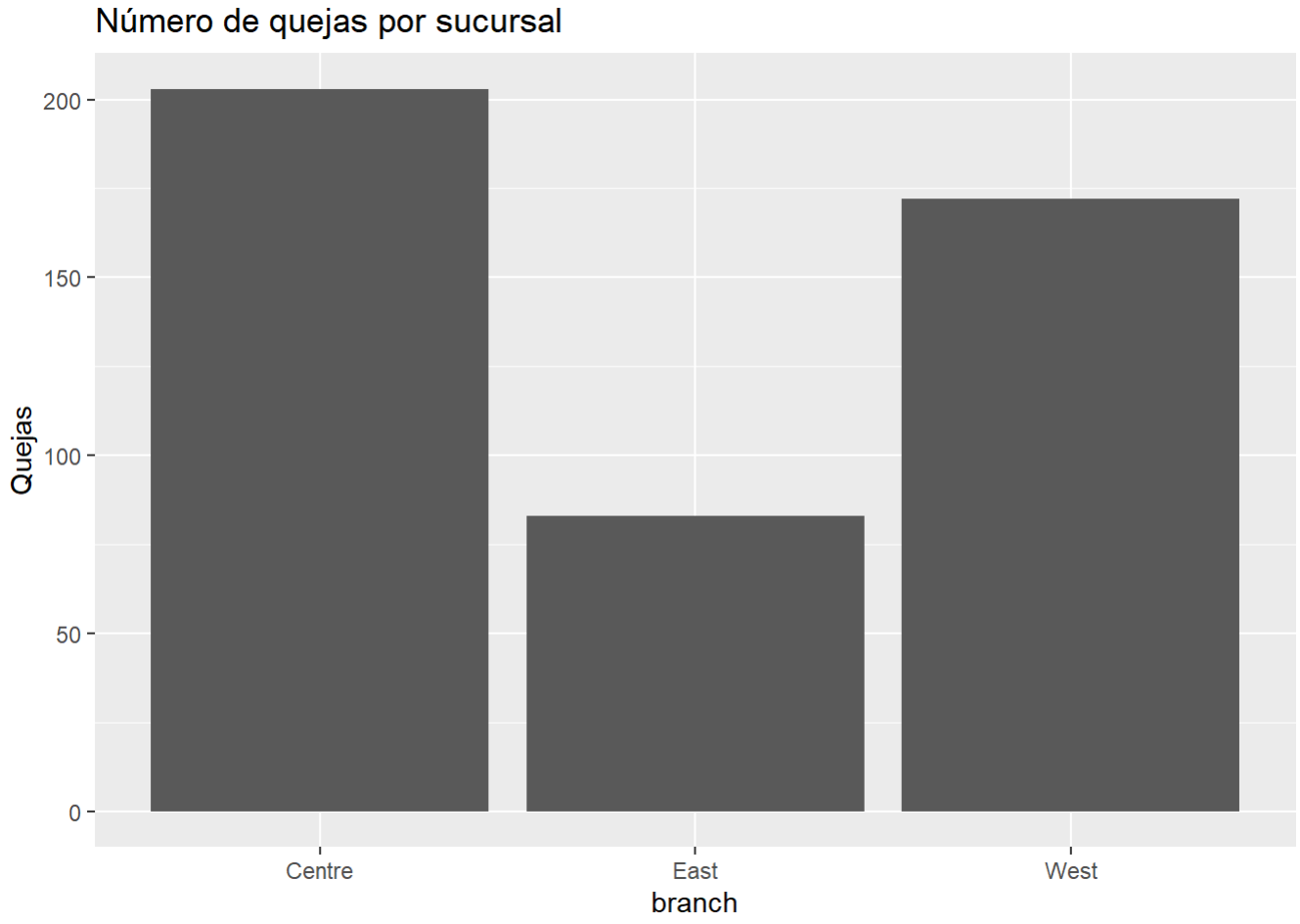
Sí encontramos diferencias significativas en la temperatura por sucursal. East está claramente sobre el resto al presentar temperaturas promedio más altas y en general sus valores registrados son más altos que los de Centre y West. Centre es el que registra temperaturas menores al resto.

9. Analiza el número de quejas por sucursal.

```
resumen<-d6 %>% group_by(branch) %>% summarise(Quejas=sum(quejas))
resumen
```

```
## # A tibble: 3 x 2
##   branch Quejas
## * <chr>   <dbl>
## 1 Centre    203
## 2 East       83
## 3 West      172
```

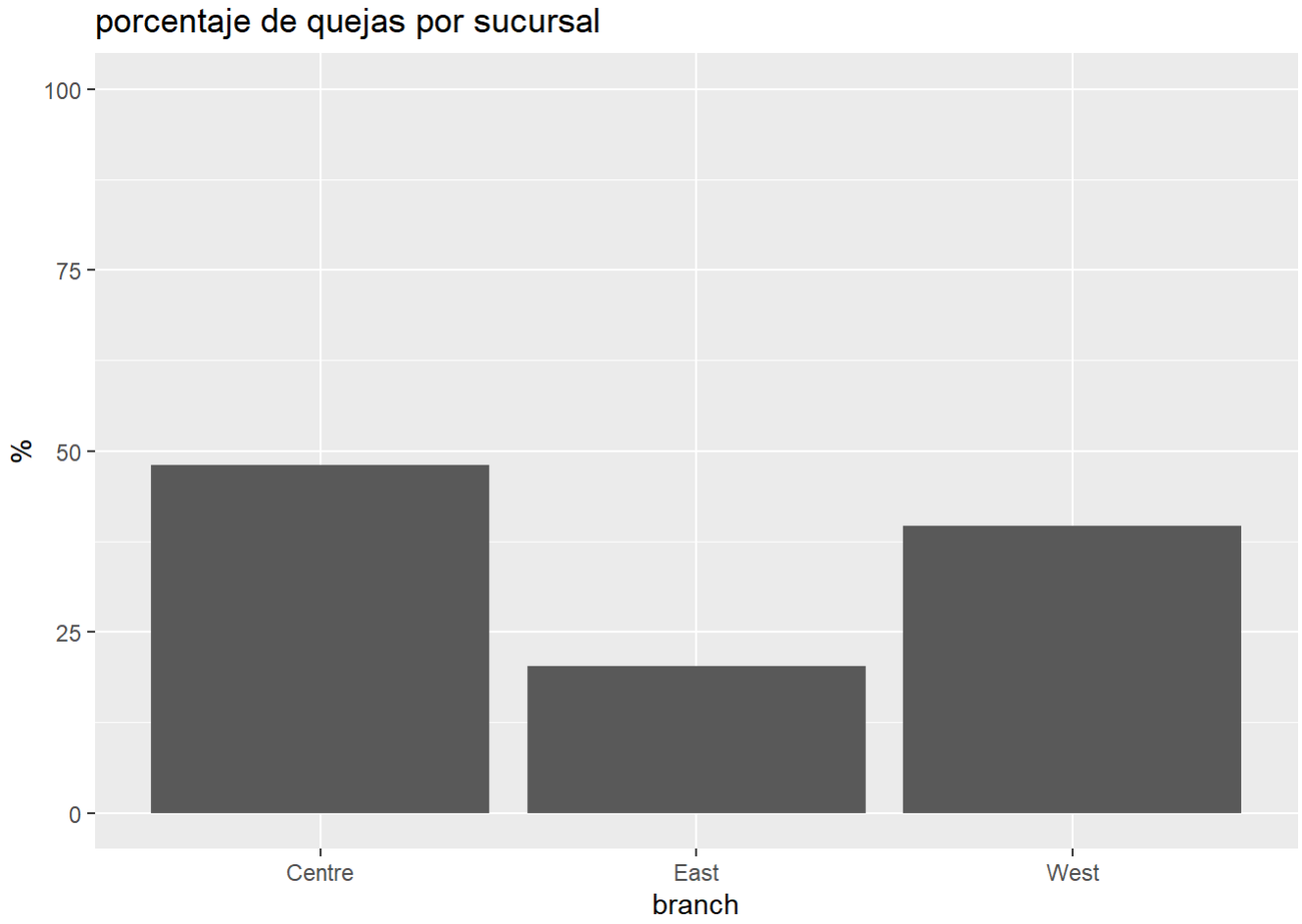
```
ggplot(resumen,aes(x=branch,y=Quejas))+geom_bar(stat="identity")+ggtitle("Número de quejas por s
ucursal")
```




```
resumen1<-d6 %>% group_by(branch) %>% summarise(porcentajedequejas=sum(quejas)*100/n())
resumen1
```

```
## # A tibble: 3 x 2
##   branch porcentajedequejas
## * <chr>               <dbl>
## 1 Centre               48.1
## 2 East                 20.2
## 3 West                 39.6
```

```
ggplot(resumen1,aes(x=branch,y=porcentajedequejas))+geom_bar(stat="identity")+
  ylim(0,100)+ggtitle("porcentaje de quejas por sucursal")+ylab("%")
```



Claramente la sucursal “Centre” tiene problema de quejas puesto que casi 1 de cada 2 pedidos presenta una queja y además también tiene más quejas que el resto de sucursales. “East” es la sucursal que presenta menos quejas tanto individualmente como por cada pedido. “West” se acerca más a “Centre” en en número de quejas y también se le debe poner atención.

10. Analiza la distribución de la variable Valor de la cuenta tanto gráfica como numéricamente

```
summary(d6$bill)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    9.10   35.50   42.90   42.76   50.50   75.00
```

```
sd(d6$bill)
```

```
## [1] 11.22292
```

```
IQR(d6$bill)
```

```
## [1] 15
```

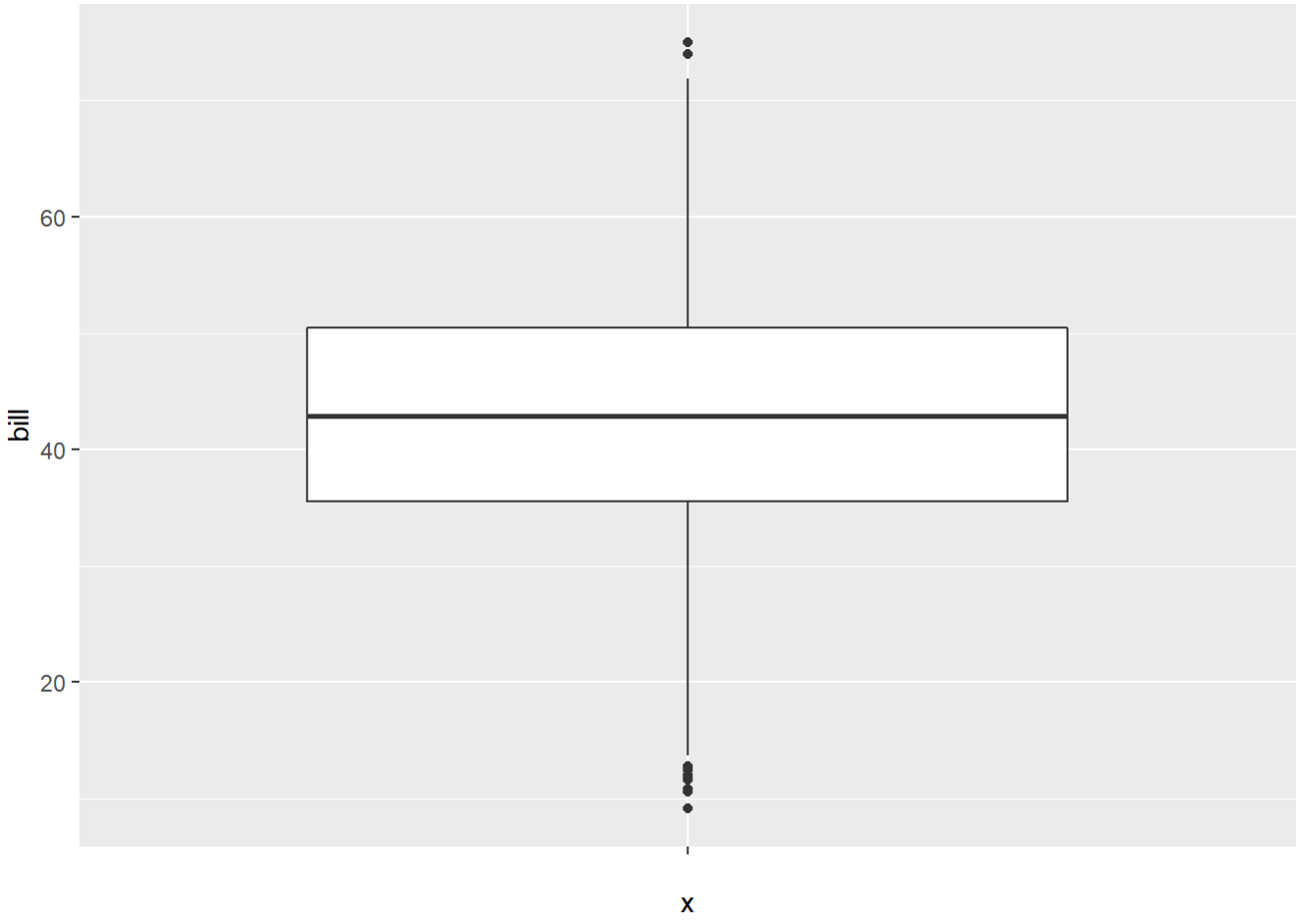
```
skewness(d6$bill)
```

```
## [1] -0.1283392
```

```
kurtosis(d6$bill)
```

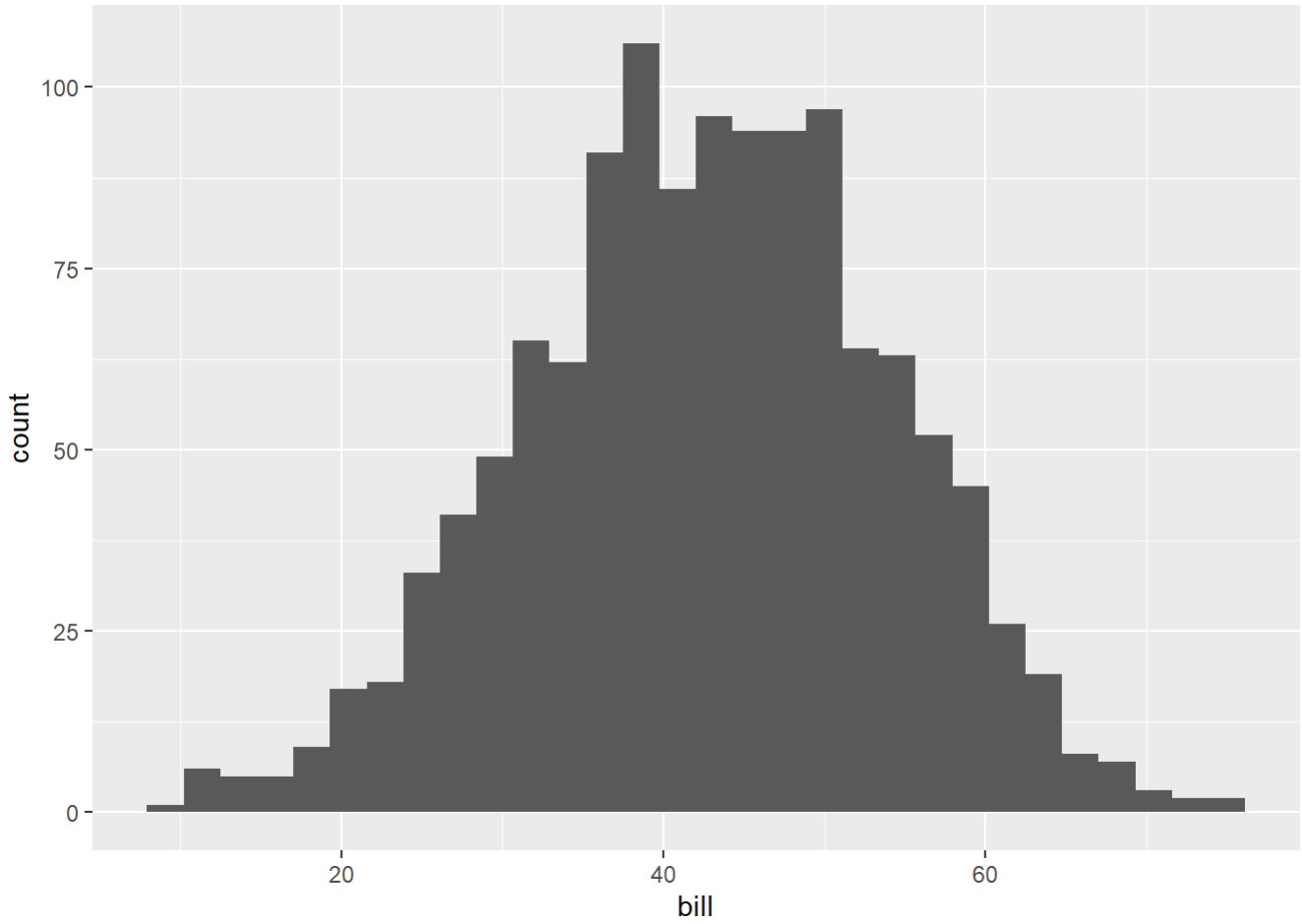
```
## [1] 2.808882
```

```
ggplot(d6,aes(x="",y=bill))+geom_boxplot()
```




```
ggplot(d6,aes(x=bill))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Esta variable muestra un sesgo a la izquierda y está más concentrada que la normal indicando colas más ligeras y sus valores están alrededor de la media 42.76. La cuenta nunca superó el valor de 75 aunque presentó valores de 9.1.

11. Analiza la distribución de la variable número de pizzas ordenadas. Tanto gráfica como numéricamente.

```
summary(d6$pizzas)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.013   4.000  11.000
```

```
sd(d6$pizzas)
```

```
## [1] 1.467102
```

```
IQR(d6$pizzas)
```

```
## [1] 2
```

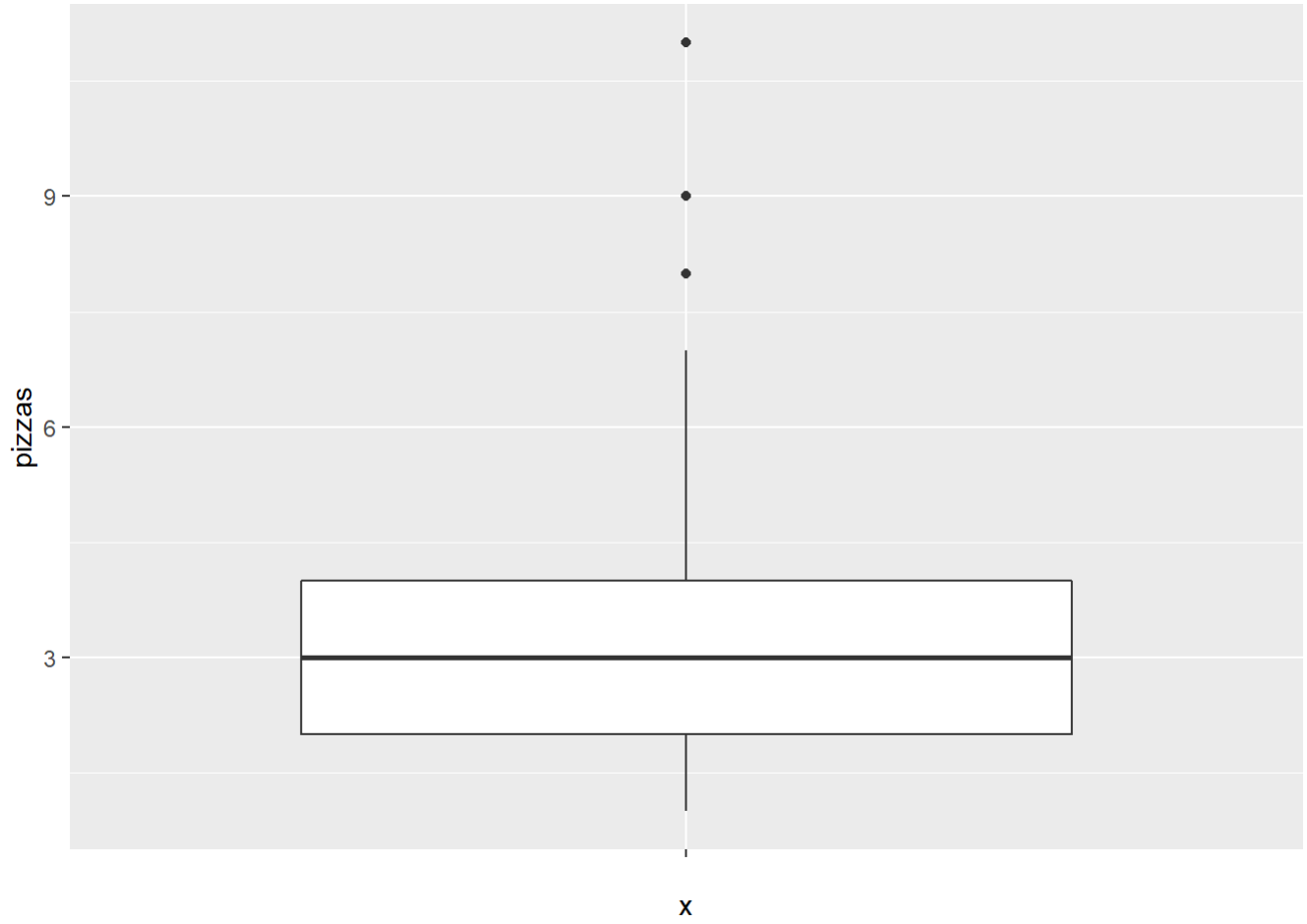
```
skewness(d6$pizzas)
```

```
## [1] 0.8963952
```

```
kurtosis(d6$pizzas)
```

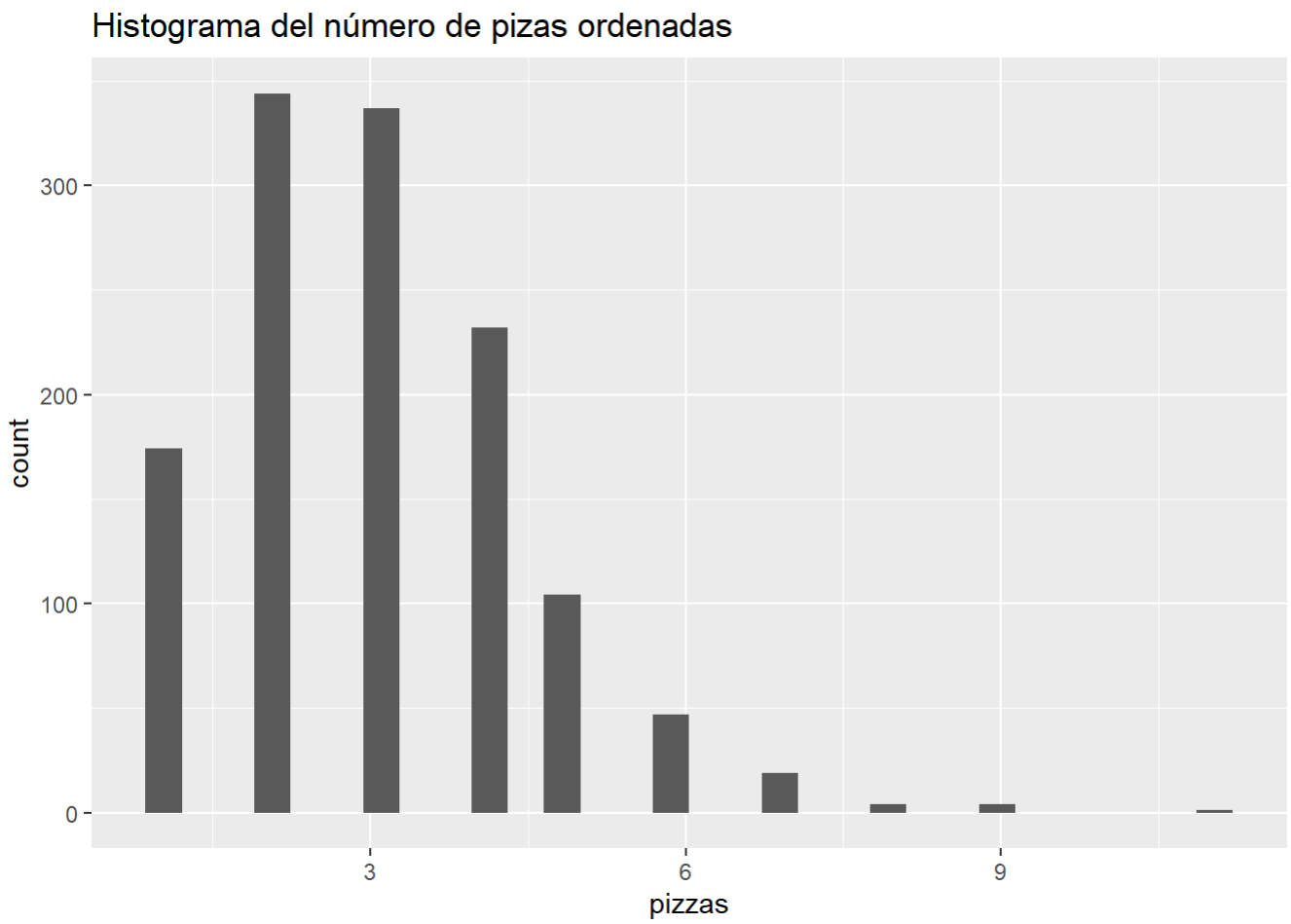
```
## [1] 4.317028
```

```
ggplot(d6,aes(x="",y=pizzas))+geom_boxplot()
```




```
ggplot(d6,aes(x=pizzas))+geom_histogram()+ggtitle("Histograma del número de pizas ordenadas")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



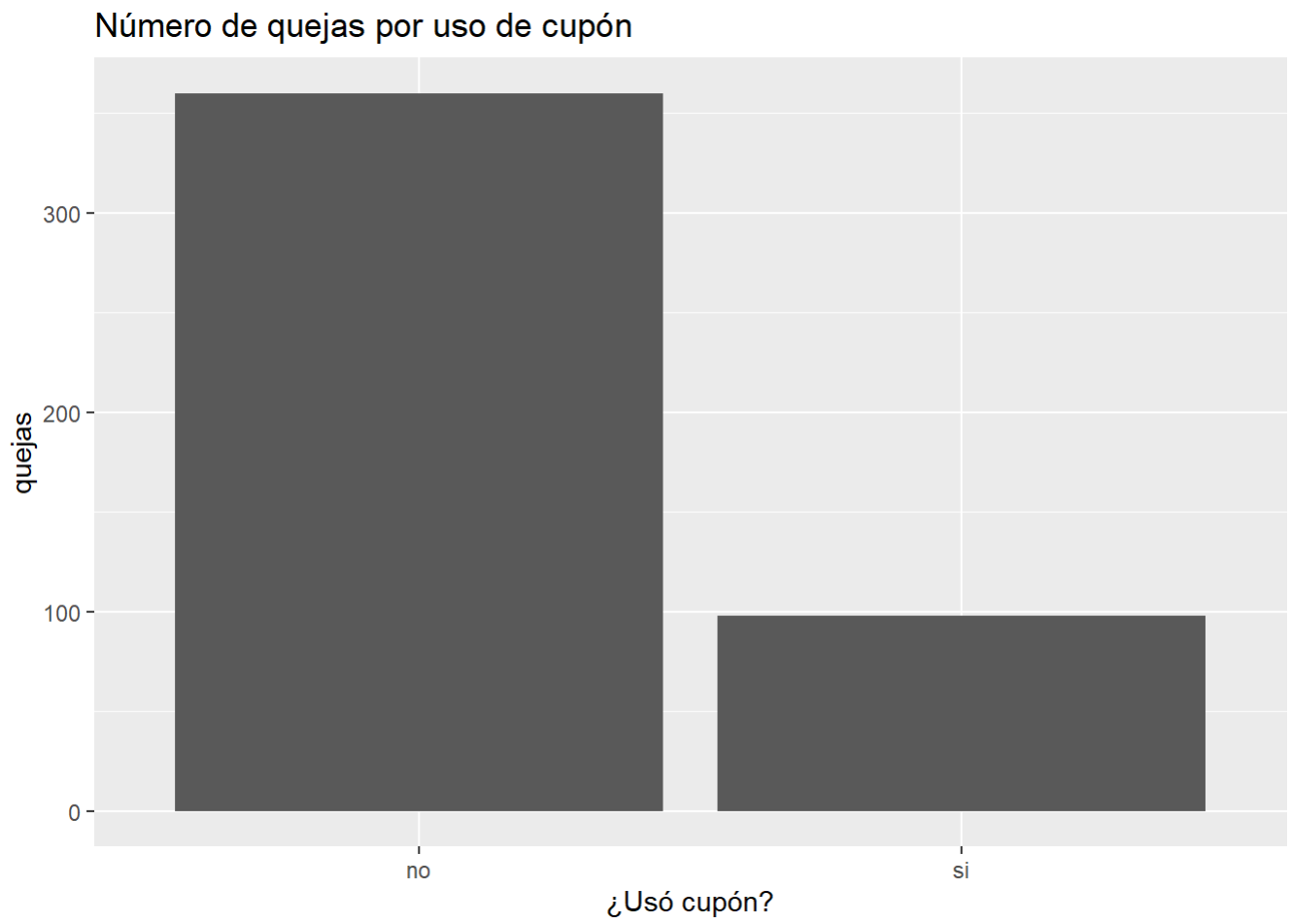
El número de pizzas odenadas jamás superó las 11 unidades, su distribución mostró un sesgo a la derecha y está más concentrada que la Normal indicando colas más pesadas. En promedio se ordenaron 3 pizzas aunque en general, un mayor número de personas ordenó 2 pizzas por orden

12. ¿Existe alguna relación entre la variable quejas y si el cliente presentó un cupón de descuento o no? Tanto gráfica como numéricamente

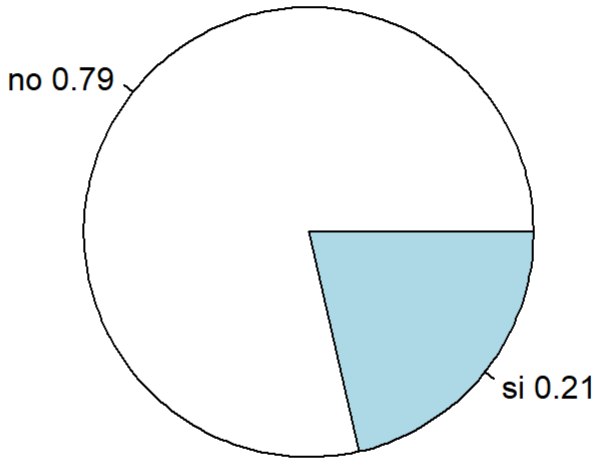
```
e<-d6 %>% group_by(discount_customer) %>% summarise(numerodequejas=sum(quejas))
usocupon<-c("no","si")
grafico<-data.frame(usocupon,e[,2])
grafico
```

```
##   usocupon numerodequejas
## 1       no            360
## 2       si             98
```

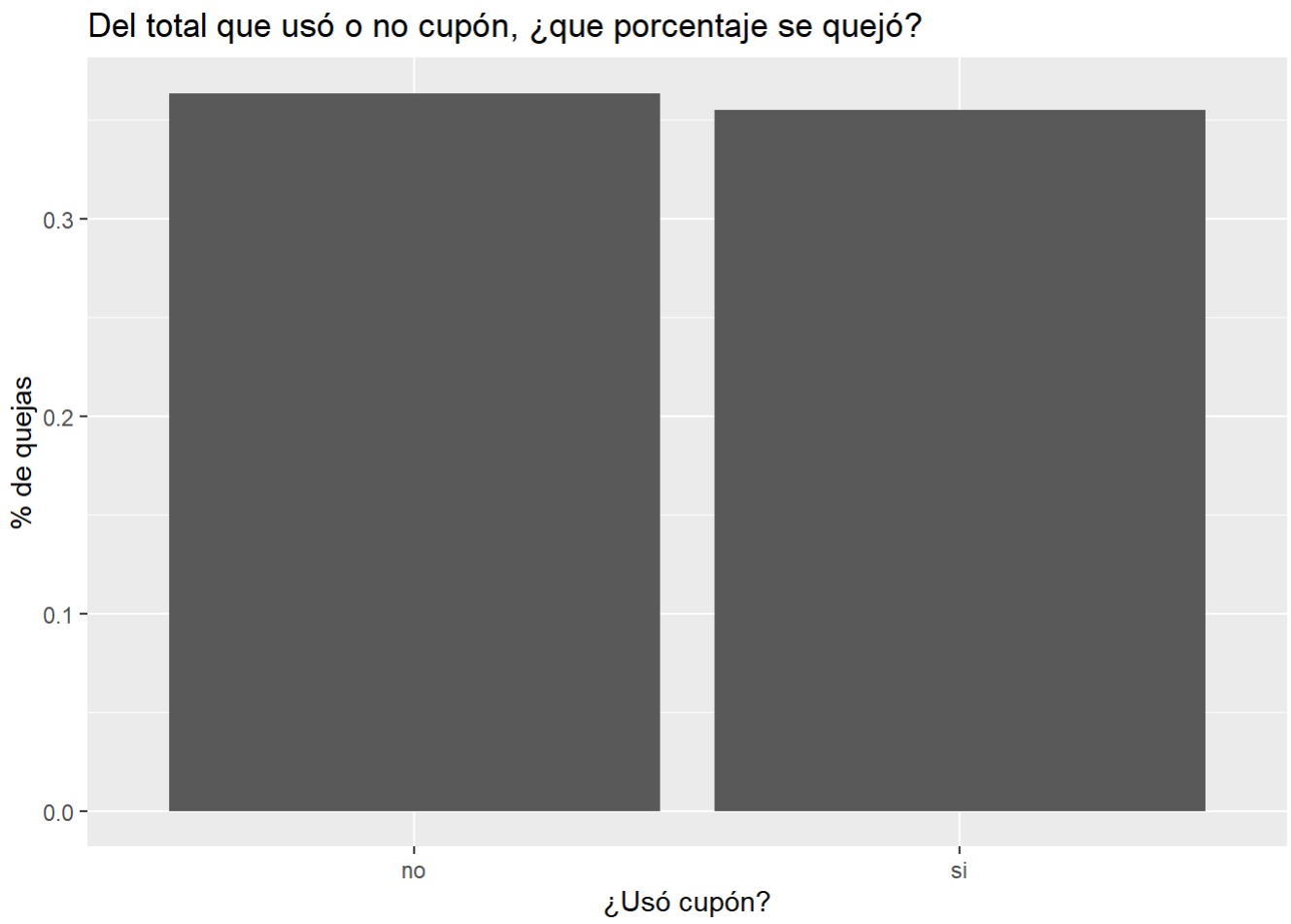
```
ggplot(grafico,aes(x=usocupon,y=numerodequejas))+geom_bar(stat="identity")+ggtitle("Número de qu
ejas por uso de cupón")+
  xlab("¿Usó cupón?")+ylab("quejas")
```




```
f<-d6 %>% group_by(discount_customer) %>% summarise(numdequejas=sum(quejas)/
                                                    sum(d6$quejas))
grafico1<-data.frame(usocupon,f[,2])
pie(grafico1[,2],labels = paste(usocupon,round(grafico1[,2],2)),main = "Del total de quejas, ¿qu
é porcentaje usó o no cupón?")
```




```
tt<-table(d6$discount_customer)
t<-c(as.integer(e[1,2])/tt[1],as.integer(e[2,2])/tt[2])
xx<-data.frame(usocupon,t)#del total que no usó cupón, el 36% se quejó
ggplot(xx,aes(x=usocupon,y=t))+geom_bar(stat="identity")+ggtitle("Del total que usó o no cupón,
 ¿que porcentaje se quejó?")+
  xlab("¿Usó cupón?")+ylab("% de quejas")
```




```
tab<-table(d6$discount_customer,d6$quejas)
chisq.test(tab)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab
## X-squared = 0.036482, df = 1, p-value = 0.8485
```

```
mutinformation(d6$discount_customer,d6$quejas)
```

```
## [1] 2.712889e-05
```

Graficamente, si sólo comparamos el número de quejas por uso de cupón parecería que ambas variables están relacionadas entre ellas pero estaríamos equivocados puesto que debemos tomar en cuenta el número de personas que usaron y no usaron cupón para ver las proporciones de quejas en ambos casos. Al ver este gráfico nos damos cuenta que es similar el porcentaje de personas que se queja en ambos casos y no existen diferencias significativas entre usar o no usar cupones al momento de quejarse. Numericamente también comprobamos esto al calcular la medida de información mutua y la chi-cuadrada de Pearson, el primero nos arroja un valor cercano a cero y el segundo un valor p cercano a 1 por lo que a partir de este análisis concluimos que no existe asociación entre variables.

13. Elabora la tabla de frecuencias de la variable día de la semana y número de pizzas ordenadas por pedido.Determina la distribución del número de pizzas ordenadas los viernes y compárala con la distribución marginal del número de pizzas ordenadas.

Tabla de frecuencias:

```
tfreq<-table(data.frame(d6$day,d6$pizzas))
tfreq1<-addmargins(tfreq)
tfreq1
```

```
##            d6.pizzas
## d6.day          1    2    3    4    5    6    7    8    9   11  Sum
##   Friday       30   65   76   42   18    9    3    2    1    0  246
##   Monday       19   35   37   27    9    8    1    0    0    0  136
##   Saturday     36   70   62   39   23   10    5    1    0    0  246
##   Sunday       25   49   45   30   20    4    3    1    0    0  177
##   Thursday     28   60   45   39   12    8    3    0    1    0  196
##   Tuesday      20   34   35   27   11    4    1    0    1    0  133
##   Wednesday    16   31   37   28   11    4    3    0    1    1  132
##   Sum         174  344  337  232  104   47   19    4    4    1 1266
```

La conjunta y sus marginales:

```
per<-prop.table(tfreq)
addmargins(per)
```

```
##           d6.pizzas
## d6.day                1           2           3           4           5
##   Friday    0.0236966825 0.0513428120 0.0600315956 0.0331753555 0.0142180095
##   Monday    0.0150078989 0.0276461295 0.0292259084 0.0213270142 0.0071090047
##   Saturday  0.0284360190 0.0552922591 0.0489731438 0.0308056872 0.0181674566
##   Sunday    0.0197472354 0.0387045814 0.0355450237 0.0236966825 0.0157977883
##   Thursday  0.0221169036 0.0473933649 0.0355450237 0.0308056872 0.0094786730
##   Tuesday   0.0157977883 0.0268562401 0.0276461295 0.0213270142 0.0086887836
##   Wednesday 0.0126382306 0.0244865719 0.0292259084 0.0221169036 0.0086887836
##   Sum       0.1374407583 0.2717219589 0.2661927330 0.1832543444 0.0821484992
##           d6.pizzas
## d6.day                6           7           8           9          11
##   Friday    0.0071090047 0.0023696682 0.0015797788 0.0007898894 0.0000000000
##   Monday    0.0063191153 0.0007898894 0.0000000000 0.0000000000 0.0000000000
##   Saturday  0.0078988942 0.0039494471 0.0007898894 0.0000000000 0.0000000000
##   Sunday    0.0031595577 0.0023696682 0.0007898894 0.0000000000 0.0000000000
##   Thursday  0.0063191153 0.0023696682 0.0000000000 0.0007898894 0.0000000000
##   Tuesday   0.0031595577 0.0007898894 0.0000000000 0.0007898894 0.0000000000
##   Wednesday 0.0031595577 0.0023696682 0.0000000000 0.0007898894 0.0007898894
##   Sum       0.0371248025 0.0150078989 0.0031595577 0.0031595577 0.0007898894
##           d6.pizzas
## d6.day              Sum
##   Friday    0.1943127962
##   Monday    0.1074249605
##   Saturday  0.1943127962
##   Sunday    0.1398104265
##   Thursday  0.1548183254
##   Tuesday   0.1050552923
##   Wednesday 0.1042654028
##   Sum       1.0000000000
```

La distribución del número de pizzas ordenadas los viernes:

```
viernes=data.frame(prop.table(tfreq[1,]))
viernes
```

```
##    prop.table.tfreq.1....
## 1              0.121951220
## 2              0.264227642
## 3              0.308943089
## 4              0.170731707
## 5              0.073170732
## 6              0.036585366
## 7              0.012195122
## 8              0.008130081
## 9              0.004065041
## 11             0.000000000
```

La distribución del número de pizzas ordenadas por días:

```
pizzas=data.frame(margin.table(per,2))
pizzas
```

```
##    d6.pizzas         Freq
## 1          1 0.1374407583
## 2          2 0.2717219589
## 3          3 0.2661927330
## 4          4 0.1832543444
## 5          5 0.0821484992
## 6          6 0.0371248025
## 7          7 0.0150078989
## 8          8 0.0031595577
## 9          9 0.0031595577
## 10        11 0.0007898894
```

Comparaciones:

```
valorespizzas=as.numeric(pizzas[,2])
valorespizzas
```

```
##  [1] 0.1374407583 0.2717219589 0.2661927330 0.1832543444 0.0821484992
##  [6] 0.0371248025 0.0150078989 0.0031595577 0.0031595577 0.0007898894
```

```
numero=as.numeric(as.character(pizzas[,1]))
numero
```

```
##  [1]  1  2  3  4  5  6  7  8  9 11
```

```
medianumeropizzas=sum(numero*valorespizzas)
medianumeropizzas
```

```
## [1] 3.013428
```

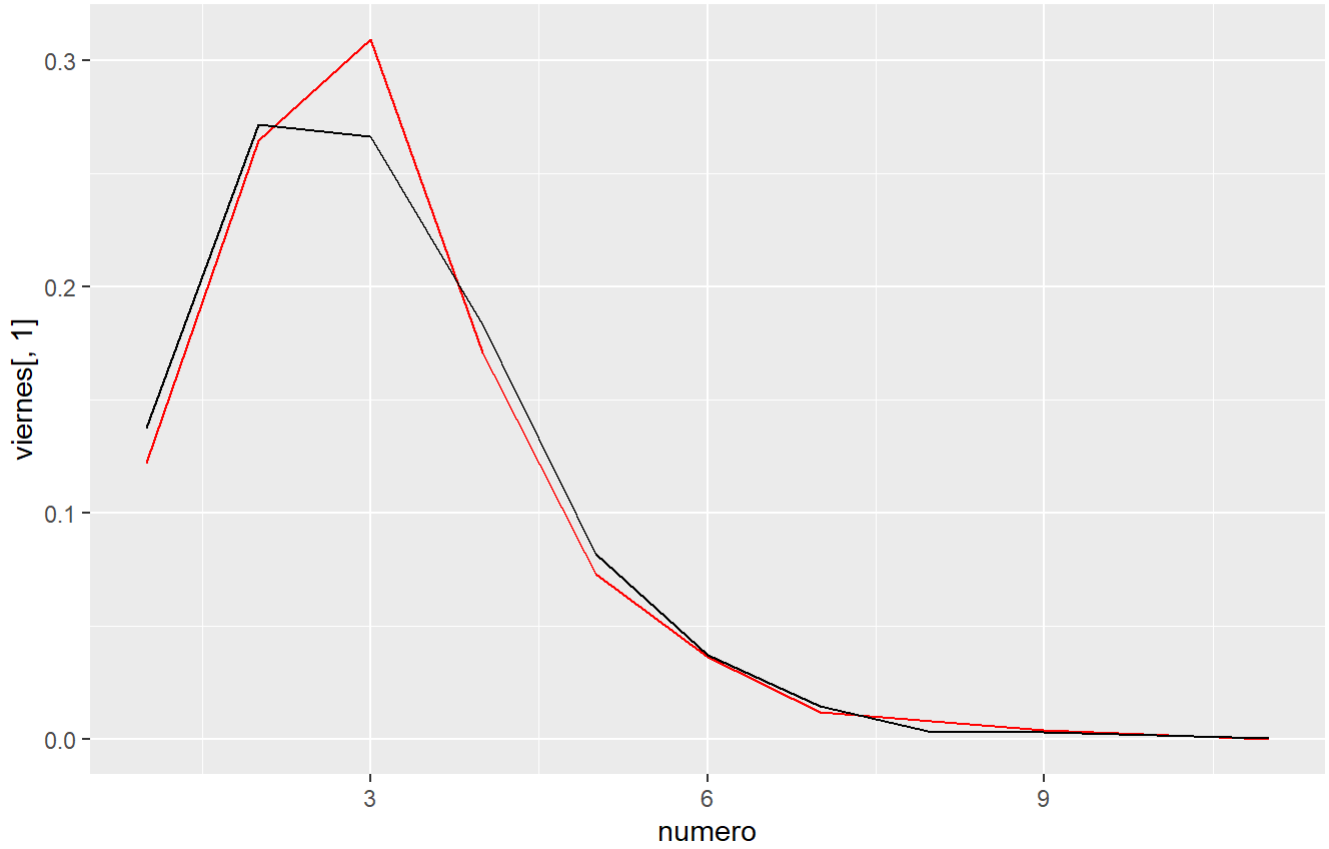
```
valoresviernes=as.numeric(viernes[,1])
mediaviernes=sum(numero*valoresviernes)
mediaviernes
```

```
## [1] 3.03252
```

```
ggplot(pizzas,aes(x=numero))+geom_line(aes(y=viernes[,1]),color="red")+geom_line(y=pizzas[,2])+g
gtitle("Marginales",subtitle = "número de pizzas ordenada por días (negra) y número de pizzas lo
s viernes (rojo)")
```

Marginales

número de pizzas ordenada por días (negra) y número de pizzas los viernes (rojo)



14. Construye una nueva base de datos que contenga dia de la semana, sucursal,número total de pedidos por día de la semana, valor de la cuenta promedio por día de la semana, número de pizzas promedio por día de la semana, número total de quejas. ¿Qué conclusiones puedes sacar de esta base de datos?

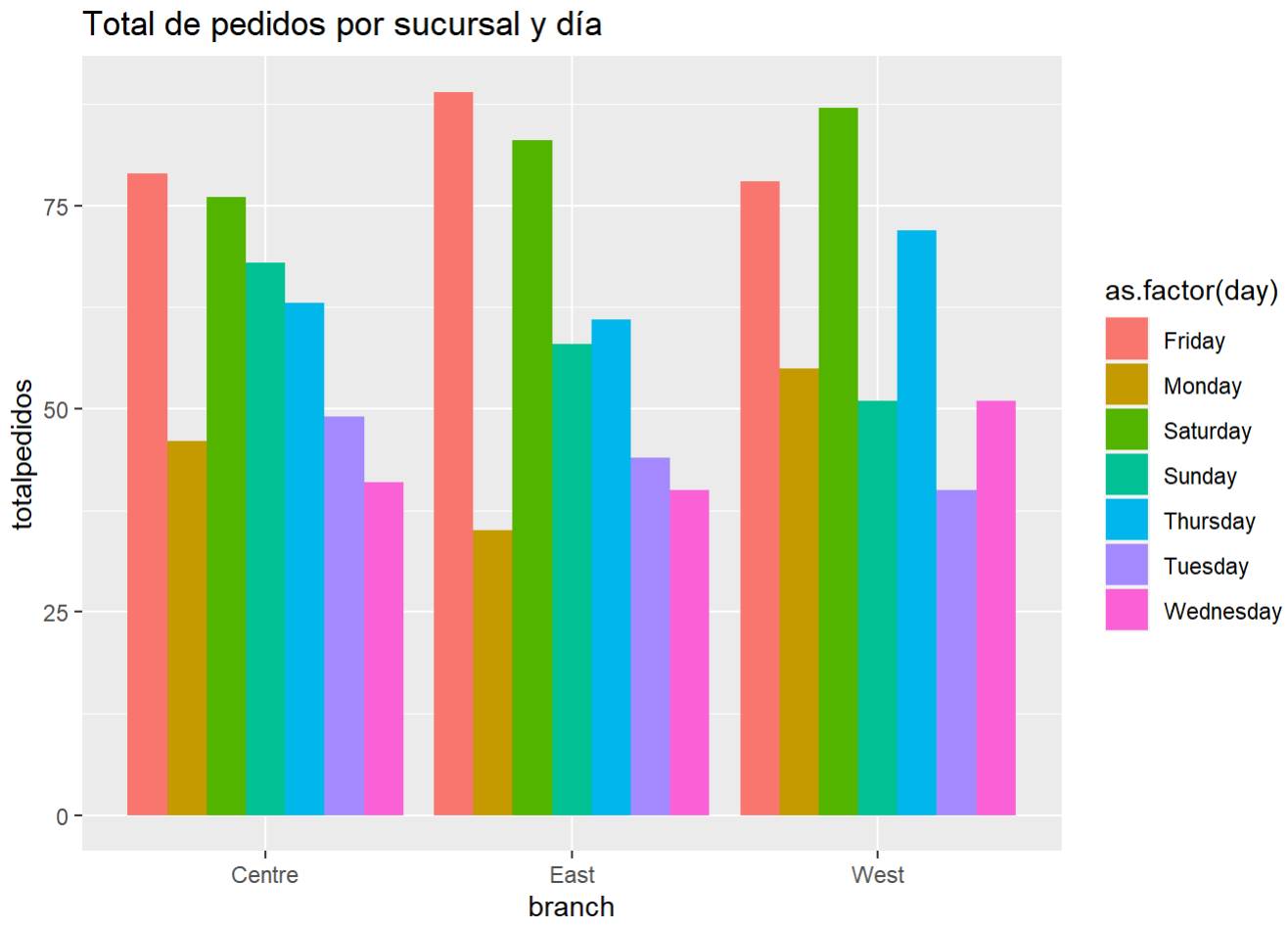
```
nuevo<-d6 %>% group_by(day,branch) %>% summarise(totalpedidos=n(),
                                    meancuenta=mean(bill),meanpizzas=mean(pizzas),
                                    totalquejas=sum(quejas))
```

```
## `summarise()` has grouped output by 'day'. You can override using the `.groups` argument.
```

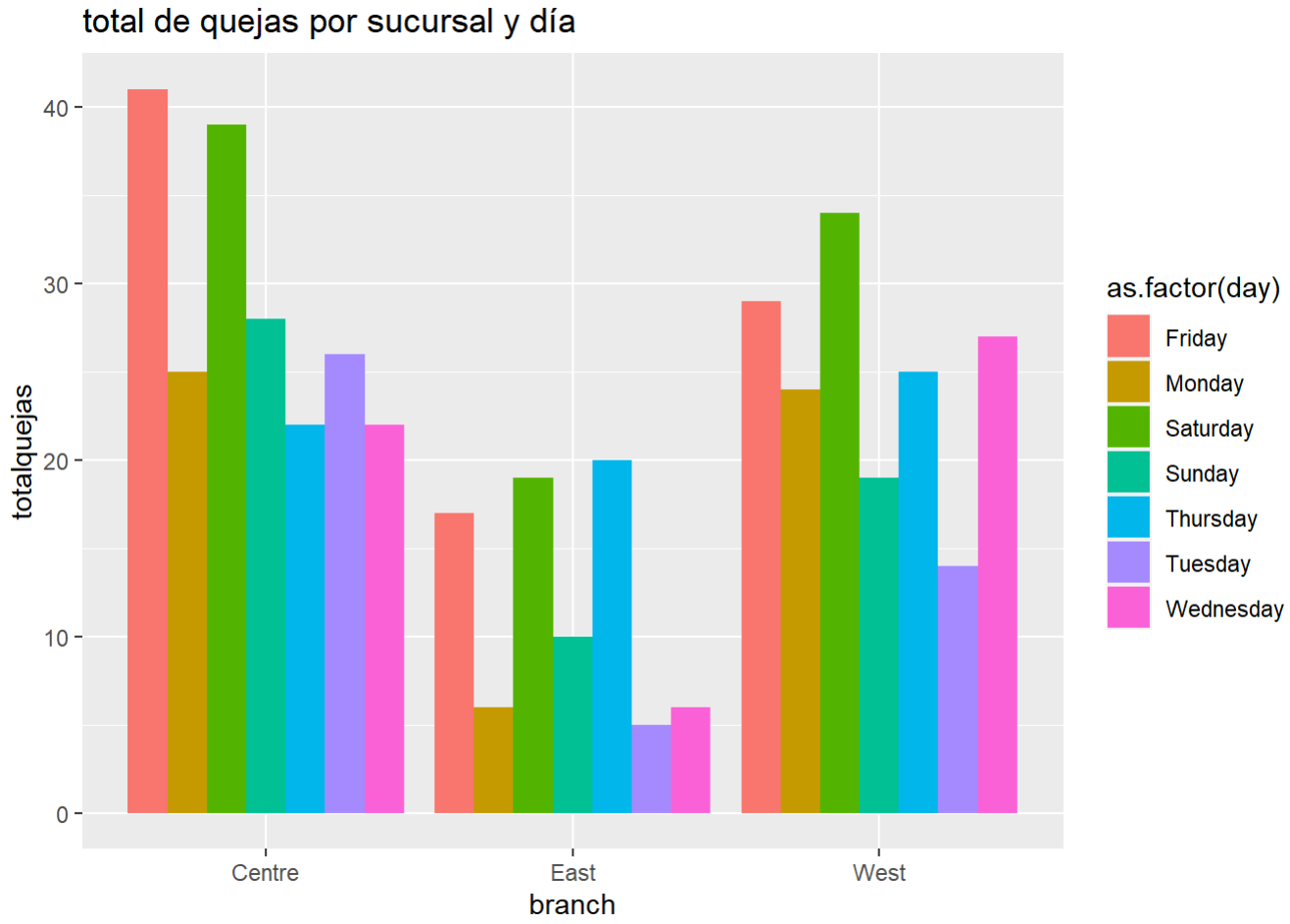
```
nuevo<-data.frame(nuevo)
nuevo
```

```
##          day branch totalpedidos meancuenta meanpizzas totalquejas
## 1     Friday Centre           79   46.80633   3.291139          41
## 2     Friday   East           89   37.50562   2.404494          17
## 3     Friday   West           78   44.12436   3.487179          29
## 4     Monday Centre           46   47.69130   3.695652          25
## 5     Monday   East           35   34.74000   2.200000           6
## 6     Monday   West           55   45.78364   2.927273          24
## 7   Saturday Centre           76   46.65789   3.355263          39
## 8   Saturday   East           83   35.88554   2.433735          19
## 9   Saturday   West           87   43.60230   3.206897          34
## 10    Sunday Centre           68   46.20588   3.500000          28
## 11    Sunday   East           58   37.04828   2.310345          10
## 12    Sunday   West           51   41.93137   3.117647          19
## 13  Thursday Centre           63   47.06984   3.317460          22
## 14  Thursday   East           61   37.18361   2.475410          20
## 15  Thursday   West           72   44.90556   3.013889          25
## 16   Tuesday Centre           49   47.29184   3.428571          26
## 17   Tuesday   East           44   35.53409   2.431818           5
## 18   Tuesday   West           40   43.00000   3.025000          14
## 19 Wednesday Centre           41   48.66098   4.121951          22
## 20 Wednesday   East           40   38.78250   2.300000           6
## 21 Wednesday   West           51   46.32549   3.137255          27
```

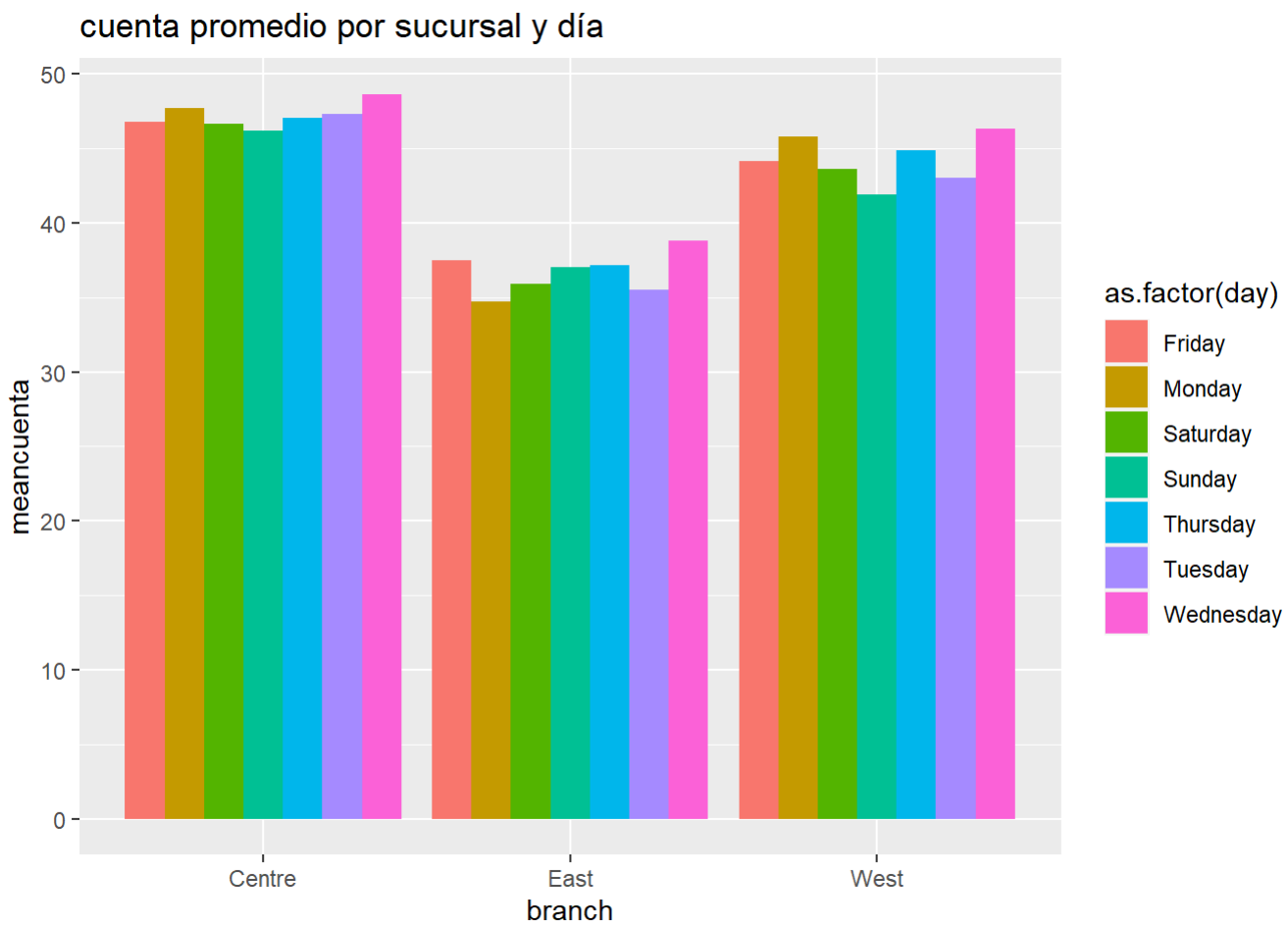
```
ggplot(nuevo,aes(x=branch,y=totalpedidos,fill=as.factor(day)))+geom_bar(stat = "identity",positi
on = "dodge")+ggtitle("Total de pedidos por sucursal y día")
```



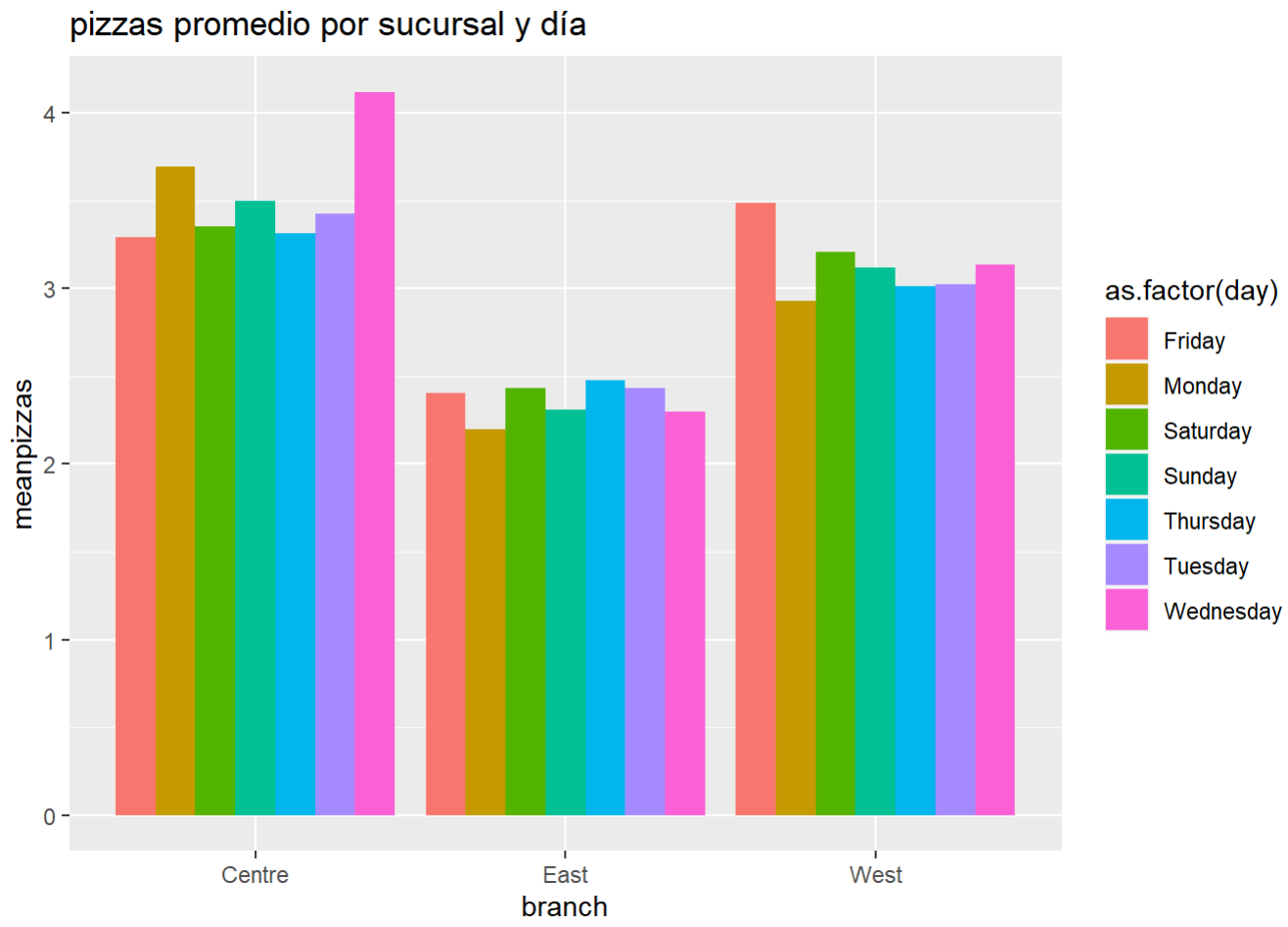
```
ggplot(nuevo,aes(x=branch,y=totalquejas,fill=as.factor(day)))+geom_bar(stat = "identity",positio
n = "dodge")+ggtitle("total de quejas por sucursal y día")
```




```
ggplot(nuevo,aes(x=branch,y=meancuenta,fill=as.factor(day)))+geom_bar(stat = "identity",position
= "dodge")+ggtitle("cuenta promedio por sucursal y día")
```




```
ggplot(nuevo,aes(x=branch,y=meanpizzas,fill=as.factor(day)))+geom_bar(stat = "identity",position
= "dodge")+ggtitle("pizzas promedio por sucursal y día")
```



A partir de lo observado en las gráficas anteriores podemos concluir que la sucursal “East” tiene un comportamiento diferente al resto. Por un lado aunque el número de pedidos es similar en las tres sucursales, “East” presenta menos quejas sin importar el día pero que tenga menor número de quejas no significa que su cuenta promedio sea superior, sino sucede todo lo contrario pues la cuenta promedio es inferior en esta sucursal sin importar el día de la semana y además también el número promedio de pizzas es menor al resto de sucursales sin importar el día. Esto podría suponer que a mayor número de pizzas promedio, mayor número de quejas y mayor cuenta promedio. Por último también podemos ver que viernes y sábado son los días con mayor número de pedidos en todas las sucursales y esto sólo se refleja en el número de quejas más no en la cuenta promedio.

15. Elabora un reporte sobre el ingreso diario total por sucursal. Debe ser presentado en una tabla que contenga la fecha en formato número de dia de nombre de mes completo de año con 4 dígitos, y otras 3 columnas con el ingreso total de cada sucursal en ese día.

```
nuevo1<-d6 %>% group_by(date,branch) %>% summarise(total=sum(bill))
```

```
## `summarise()` has grouped output by 'date'. You can override using the `.groups` argument.
```

```
nuevo1<-data.frame(nuevo1)
nuevo1$date<- format(nuevo1$date, "%d de %B de %Y")
reporte<-data.frame(pivot_wider(nuevo1,names_from = branch,values_from = total))
names(reporte)<-c("fecha","I.total Centre","I.total East","I.total West")
reporte
```

```
##                fecha I.total Centre I.total East I.total West
## 1  01 de mayo de 2014          768.0        477.6        879.6
## 2  02 de mayo de 2014          639.5        618.1        915.4
## 3  03 de mayo de 2014          714.8        562.0        834.2
## 4  04 de mayo de 2014         1108.2        372.0        530.9
## 5  05 de mayo de 2014          568.2        383.5        449.2
## 6  06 de mayo de 2014          464.7        396.9        524.2
## 7  07 de mayo de 2014          631.1        344.3        519.3
## 8  08 de mayo de 2014          444.7        464.6        591.2
## 9  09 de mayo de 2014          840.8        515.8        650.0
## 10 10 de mayo de 2014          955.2        373.4        572.6
## 11 11 de mayo de 2014          540.3        711.1        570.7
## 12 12 de mayo de 2014          632.5        343.9        546.4
## 13 13 de mayo de 2014          568.4        297.2        393.5
## 14 14 de mayo de 2014          510.6        366.9        491.7
## 15 15 de mayo de 2014          649.9        301.0        535.9
## 16 16 de mayo de 2014          872.9        820.4        809.9
## 17 17 de mayo de 2014          641.1        686.6        926.0
## 18 18 de mayo de 2014          579.4        613.8        622.2
## 19 19 de mayo de 2014          521.8        278.4        869.5
## 20 20 de mayo de 2014          753.6        453.1        365.0
## 21 21 de mayo de 2014          458.4        458.8        540.6
## 22 22 de mayo de 2014          471.5        529.7        660.6
## 23 23 de mayo de 2014          716.9        586.5        646.3
## 24 24 de mayo de 2014          455.4        840.2        719.0
## 25 25 de mayo de 2014          914.1        451.9        414.7
## 26 26 de mayo de 2014          471.3        210.1        653.0
## 27 27 de mayo de 2014          530.6        416.3        437.3
## 28 28 de mayo de 2014          395.0        381.3        811.0
## 29 29 de mayo de 2014          631.3        495.3        565.9
## 30 30 de mayo de 2014          627.6        797.2        420.1
## 31 31 de mayo de 2014          779.5        516.3        741.6
```

¿Hay alguna o algunas variables que explican el número de quejas? Usando los puntos anteriores y el análisis bivariado adecuado para las parejas de variables, ¿cuáles son tus conclusiones?

```
chi1<-table(d6$day,d6$quejas)
chisq.test(chi1)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi1
## X-squared = 4.8867, df = 6, p-value = 0.5584
```

```
chi2<-table(d6$branch,d6$quejas)
chisq.test(chi2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi2
## X-squared = 73.323, df = 2, p-value < 2.2e-16
```

```
chi3<-table(d6$driver,d6$quejas)
chisq.test(chi3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi3
## X-squared = 22.645, df = 4, p-value = 0.000149
```

```
chi4<-table(d6$operator,d6$quejas)
chisq.test(chi4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  chi4
## X-squared = 1.5642, df = 1, p-value = 0.2111
```

```
chi5<-table(d6$pizzas,d6$quejas)
chisq.test(chi5)
```

```
## Warning in chisq.test(chi5): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi5
## X-squared = 116.93, df = 9, p-value < 2.2e-16
```

```
chi7<-table(discretize(d6$temperature)[,1],d6$quejas)
chisq.test(chi7)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi7
## X-squared = 1042.6, df = 9, p-value < 2.2e-16
```

```
chi8<-table(discretize(d6$time)[,1],d6$quejas)
chisq.test(chi8)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi8
## X-squared = 140.55, df = 9, p-value < 2.2e-16
```

```
chi9<-table(discretize(d6$bill)[,1],d6$quejas)
chisq.test(chi9)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chi9
## X-squared = 112.39, df = 9, p-value < 2.2e-16
```

```
mutinformation(discretize(d6))
```

```
##                          X         day        date        time     operator
## X                2.302478999 0.657890832 1.896080453 0.030353030 9.516709e-03
## day              0.657890832 1.912670175 0.786185143 0.021047942 1.516266e-01
## date             1.896080453 0.786185143 2.290630948 0.037396861 1.282668e-02
## time             0.030353030 0.021047942 0.037396861 2.302478999 3.844978e-03
## operator         0.009516709 0.151626554 0.012826681 0.003844978 6.931160e-01
## branch           0.005698138 0.004551185 0.006311687 0.075713646 7.871071e-05
## driver           0.092190144 0.307167713 0.101620556 0.042603468 8.254701e-03
## temperature      0.045715107 0.019807478 0.043627246 0.155232980 4.708625e-03
## bill             0.023361454 0.025460170 0.024908093 0.145145578 2.272120e-03
## pizzas           0.018292969 0.005621046 0.022660234 0.122652952 9.283267e-04
## free_wine        0.004791746 0.003018696 0.005467696 0.419041181 3.455340e-04
## got_wine         0.004316796 0.002074825 0.006484312 0.306981111 2.996473e-05
## discount_customer 0.006178947 0.002027152 0.005267693 0.010571651 8.697494e-06
## quejas           0.009732768 0.001920912 0.007816141 0.063252588 6.771876e-04
##                        branch      driver temperature        bill       pizzas
## X                5.698138e-03 0.092190144 0.045715107 0.023361454 0.0182929689
## day              4.551185e-03 0.307167713 0.019807478 0.025460170 0.0056210462
## date             6.311687e-03 0.101620556 0.043627246 0.024908093 0.0226602341
## time             7.571365e-02 0.042603468 0.155232980 0.145145578 0.1226529524
## operator         7.871071e-05 0.008254701 0.004708625 0.002272120 0.0009283267
## branch           1.098343e+00 0.021868272 0.091702943 0.090513511 0.0643779942
## driver           2.186827e-02 1.498614653 0.048973544 0.052589897 0.0354816324
## temperature      9.170294e-02 0.048973544 2.302478999 0.145667655 0.1087637296
## bill             9.051351e-02 0.052589897 0.145667655 2.302410185 0.1321757522
## pizzas           6.437799e-02 0.035481632 0.108763730 0.132175752 1.6628153658
## free_wine        1.906930e-02 0.008328760 0.021565796 0.031528041 0.0280962779
## got_wine         1.795802e-02 0.007759087 0.019791391 0.023400727 0.0249227270
## discount_customer 1.893356e-03 0.002458069 0.003519469 0.002550435 0.0022188759
## quejas           3.027901e-02 0.009601626 0.512082760 0.049205895 0.0484760860
##                     free_wine     got_wine discount_customer       quejas
## X                0.0047917462 4.316796e-03      6.178947e-03 9.732768e-03
## day              0.0030186959 2.074825e-03      2.027152e-03 1.920912e-03
## date             0.0054676955 6.484312e-03      5.267693e-03 7.816141e-03
## time             0.4190411809 3.069811e-01      1.057165e-02 6.325259e-02
## operator         0.0003455340 2.996473e-05      8.697494e-06 6.771876e-04
## branch           0.0190692966 1.795802e-02      1.893356e-03 3.027901e-02
## driver           0.0083287601 7.759087e-03      2.458069e-03 9.601626e-03
## temperature      0.0215657958 1.979139e-02      3.519469e-03 5.120828e-01
## bill             0.0315280407 2.340073e-02      2.550435e-03 4.920590e-02
## pizzas           0.0280962779 2.492273e-02      2.218876e-03 4.847609e-02
## free_wine        0.4727323154 3.350932e-01      1.931202e-04 2.767989e-02
## got_wine         0.3350931713 4.200970e-01      5.763078e-05 9.275518e-03
## discount_customer 0.0001931202 5.763078e-05      5.243771e-01 2.712889e-05
## quejas           0.0276798903 9.275518e-03      2.712889e-05 6.544294e-01
```

```
da <- d6[, c("quejas","discount_customer","free_wine","got_wine","pizzas","bill",
             "temperature","time")]
cor(da)
```

```
##                       quejas discount_customer   free_wine    got_wine
## quejas             1.000000000      -0.007358771  0.23981436  0.13862618
## discount_customer -0.007358771       1.000000000 -0.01950304 -0.01068329
## free_wine          0.239814365      -0.019503041  1.00000000  0.88867135
## got_wine           0.138626175      -0.010683285  0.88867135  1.00000000
## pizzas             0.263246882      -0.046578153  0.19019686  0.17490927
## bill               0.258550742      -0.012249834  0.20841663  0.17802413
## temperature       -0.748029522       0.036756510 -0.17543971 -0.16668784
## time               0.323301314      -0.044609516  0.67085369  0.60625693
##                         pizzas        bill temperature        time
## quejas             0.26324688  0.25855074 -0.74802952  0.32330131
## discount_customer -0.04657815 -0.01224983  0.03675651 -0.04460952
## free_wine          0.19019686  0.20841663 -0.17543971  0.67085369
## got_wine           0.17490927  0.17802413 -0.16668784  0.60625693
## pizzas             1.00000000  0.38675248 -0.36858039  0.35328792
## bill               0.38675248  1.00000000 -0.41515557  0.44910053
## temperature       -0.36858039 -0.41515557  1.00000000 -0.43384522
## time               0.35328792  0.44910053 -0.43384522  1.00000000
```

```
cor(da,method = "spearman")
```

```
##                       quejas discount_customer   free_wine    got_wine
## quejas             1.000000000      -0.007358771  0.23981436  0.13862618
## discount_customer -0.007358771       1.000000000 -0.01950304 -0.01068329
## free_wine          0.239814365      -0.019503041  1.00000000  0.88867135
## got_wine           0.138626175      -0.010683285  0.88867135  1.00000000
## pizzas             0.289363759      -0.057582614  0.21252815  0.19940654
## bill               0.254387803      -0.013610092  0.21160273  0.18108030
## temperature       -0.797111569       0.037003508 -0.18666233 -0.17907336
## time               0.316276275      -0.045111930  0.66670571  0.59537545
##                         pizzas        bill temperature        time
## quejas             0.28936376  0.25438780 -0.79711157  0.31627627
## discount_customer -0.05758261 -0.01361009  0.03700351 -0.04511193
## free_wine          0.21252815  0.21160273 -0.18666233  0.66670571
## got_wine           0.19940654  0.18108030 -0.17907336  0.59537545
## pizzas             1.00000000  0.40855696 -0.39206267  0.38136747
## bill               0.40855696  1.00000000 -0.37078110  0.40535078
## temperature       -0.39206267 -0.37078110  1.00000000 -0.39130532
## time               0.38136747  0.40535078 -0.39130532  1.00000000
```

```
cor(da,method = "kendall")
```

```
##                         quejas discount_customer   free_wine    got_wine
## quejas             1.000000000      -0.007358771  0.23981436  0.13862618
## discount_customer -0.007358771       1.000000000 -0.01950304 -0.01068329
## free_wine          0.239814365      -0.019503041  1.00000000  0.88867135
## got_wine           0.138626175      -0.010683285  0.88867135  1.00000000
## pizzas             0.258592082      -0.051459132  0.18992737  0.17820114
## bill               0.208044821      -0.011130680  0.17305410  0.14809208
## temperature       -0.651095866       0.030225168 -0.15246933 -0.14627052
## time               0.258340467      -0.036848281  0.54457789  0.48631397
##                         pizzas        bill temperature        time
## quejas             0.25859208  0.20804482 -0.65109587  0.25834047
## discount_customer -0.05145913 -0.01113068  0.03022517 -0.03684828
## free_wine          0.18992737  0.17305410 -0.15246933  0.54457789
## got_wine           0.17820114  0.14809208 -0.14627052  0.48631397
## pizzas             1.00000000  0.30693940 -0.29157665  0.28443667
## bill               0.30693940  1.00000000 -0.25527962  0.27962460
## temperature       -0.29157665 -0.25527962  1.00000000 -0.26941286
## time               0.28443667  0.27962460 -0.26941286  1.00000000
```

Por el test de la chi-cuadrada encontramos que la variable quejas se relaciona con “branch”, “driver”, “temperature”, “time” y “bill”. Por información mutua vemos que sólo se relaciona con “temperature”. Por correlación encontramos que se relaciona con “temperature” de forma fuerte en los 3 tipos de correlación y de forma media con “time”. Dados estos resultados y el análisis anterior concluimos que “temperature” y “time” son las variables que mejor explican el número de quejas.

Parte 2

```
library(ggplot2)
library(cluster.datasets)
```

```
## Warning: package 'cluster.datasets' was built under R version 4.0.3
```

```
data("all.mammals.milk.1956")
dat<-all.mammals.milk.1956
```

1. Realiza un análisis de componentes principales para ver si podemos resumir la información de la base de datos con un número menor de variables

```
dato<-as.data.frame(dat[,2:5])
head(dato)
```

```
##   water protein fat lactose
## 1  90.1     2.6 1.0     6.9
## 2  88.5     1.4 3.5     6.0
## 3  88.4     2.2 2.7     6.4
## 4  90.3     1.7 1.4     6.2
## 5  90.4     0.6 4.5     4.4
## 6  87.7     3.5 3.4     4.8
```

```
dat[,1]
```

```
##  [1] "Horse"      "Orangutan"  "Monkey"     "Donkey"     "Hippo"
##  [6] "Camel"      "Bison"      "Buffalo"    "Guinea Pig" "Cat"
## [11] "Fox"        "Llama"      "Mule"       "Pig"        "Zebra"
## [16] "Sheep"      "Dog"        "Elephant"   "Rabbit"     "Rat"
## [21] "Deer"       "Reindeer"   "Whale"      "Seal"       "Dolphin"
```

```
row.names(dato)<-dat[,1]
head(dato)
```

```
##           water protein fat lactose
## Horse      90.1     2.6 1.0     6.9
## Orangutan  88.5     1.4 3.5     6.0
## Monkey     88.4     2.2 2.7     6.4
## Donkey     90.3     1.7 1.4     6.2
## Hippo      90.4     0.6 4.5     4.4
## Camel      87.7     3.5 3.4     4.8
```

```
summary(dato)
```

```
##      water          protein            fat           lactose
##  Min.   :44.90   Min.   : 0.600   Min.   : 1.00   Min.   :0.000
##  1st Qu.:71.30   1st Qu.: 3.000   1st Qu.: 3.40   1st Qu.:2.700
##  Median :82.00   Median : 5.900   Median : 6.30   Median :4.700
##  Mean   :78.18   Mean   : 6.212   Mean   :10.31   Mean   :4.132
##  3rd Qu.:87.70   3rd Qu.: 9.700   3rd Qu.:13.10   3rd Qu.:5.600
##  Max.   :90.40   Max.   :12.300   Max.   :42.00   Max.   :6.900
```

Claramente debemos centrar y escalar para poder hacer este análisis

```
res<-prcomp(dato,center = TRUE,scale = TRUE)
res$rotation
```

```
##                PC1        PC2         PC3       PC4
## water   -0.5194905 -0.3366579 -0.33663482 0.7095548
## protein  0.4656017 -0.7473877  0.43336935 0.1918794
## fat      0.5028931  0.5394823  0.09399801 0.6687464
## lactose -0.5103363  0.1924369  0.83068188 0.1117701
```

```
res$sdev
```

```
## [1] 1.86454412 0.63388021 0.33965244 0.07941879
```

```
res$scale
```

```
##     water   protein       fat   lactose
## 12.817913  3.652547 10.517997  1.831830
```

```
res$center
```

```
##   water protein     fat lactose
##  78.184   6.212  10.308   4.132
```

```
res$x
```

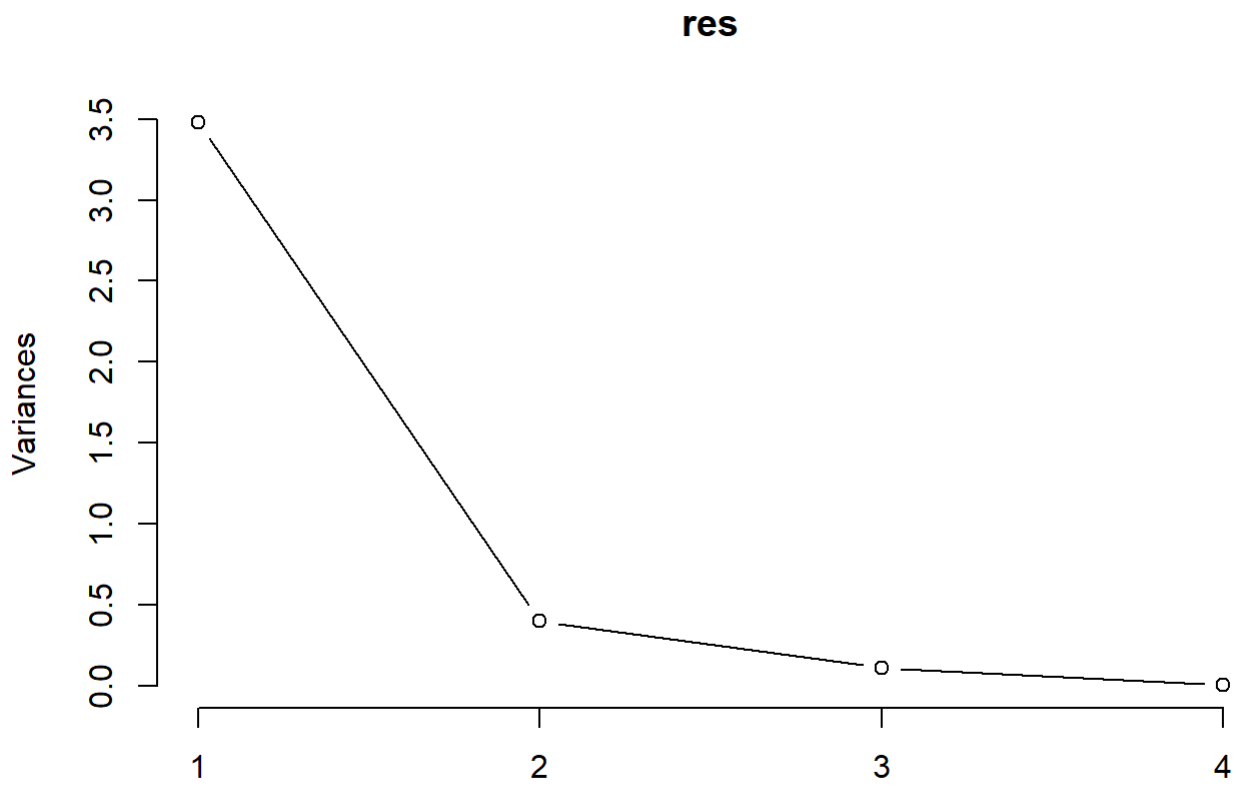
```
##                    PC1         PC2          PC3           PC4
## Horse      -2.15955785  0.23948480  0.430517153  0.0469564017
## Orangutan  -1.87741415  0.56073517 -0.055622373 -0.0006147531
## Monkey     -1.92107037  0.40065249  0.216161690  0.0094173220
## Donkey     -2.06824874  0.36537104  0.004625864 -0.0065304500
## Hippo      -1.56283420  0.55773765 -0.917057530  0.0284924516
## Camel      -1.24776651  0.02085162 -0.330509225 -0.0141573440
## Bison      -1.38164444 -0.21679262  0.237675472 -0.0433235657
## Buffalo    -0.47185521 -0.10285077  0.096187982  0.0819391650
## Guinea Pig  0.25117884 -0.65053675 -0.633784722 -0.0168704718
## Cat         0.09087423 -1.06270816  0.457302902  0.1548651616
## Fox        -0.51370377 -0.31452547  0.265193854 -0.0239251532
## Llama      -1.38058058  0.04430395  0.109455126 -0.0234757261
## Mule       -1.80370758  0.25884445 -0.265755765 -0.0246559087
## Pig        -0.20253900 -0.61544872 -0.258312144 -0.0553144065
## Zebra      -1.32306955  0.28689284 -0.111192041 -0.0039372300
## Sheep      -0.57776326 -0.11577505  0.049814396 -0.0347281221
## Dog         0.74672828 -0.74274894 -0.104684452 -0.0625127957
## Elephant   -0.08997063  1.25926704  0.617504437  0.0017002301
## Rabbit      1.81036875 -1.15619667 -0.084070146 -0.0199211432
## Rat         0.95263023 -0.43196248  0.146995138 -0.0627145701
## Deer        1.90757130 -0.21352863  0.208730141  0.0436874621
## Reindeer    2.04694152 -0.22575418  0.233228837  0.0306023587
## Whale       2.39169651 -0.35598693 -0.119392580  0.0539246206
## Seal        4.39921084  1.31252912 -0.341931084  0.1866818338
## Dolphin     3.98452536  0.89814520  0.148919070 -0.2455853671
```

```
summary(res)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.8645 0.6339 0.33965 0.07942
## Proportion of Variance 0.8691 0.1004 0.02884 0.00158
## Cumulative Proportion  0.8691 0.9696 0.99842 1.00000
```

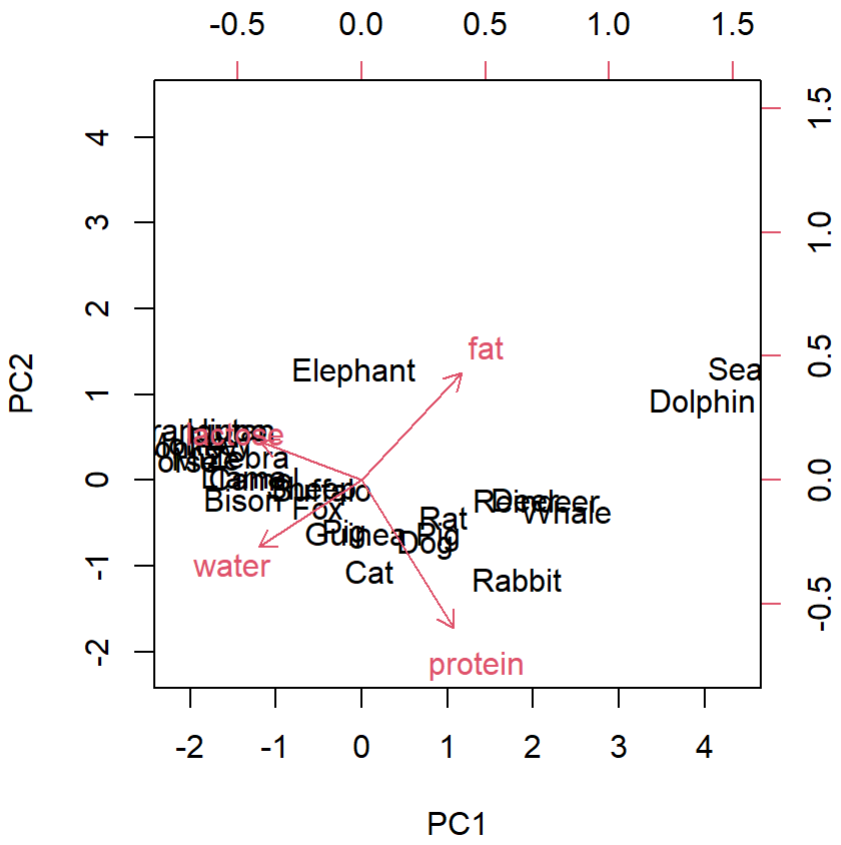
Nos quedaremos con el primer componente pricipal pues ya tenemos acumulado más del 86% y la desviación estándar es mayor a 1. Veamos la screeplot para ver si confirma esto.

```
screeplot(res,type = "l")
```

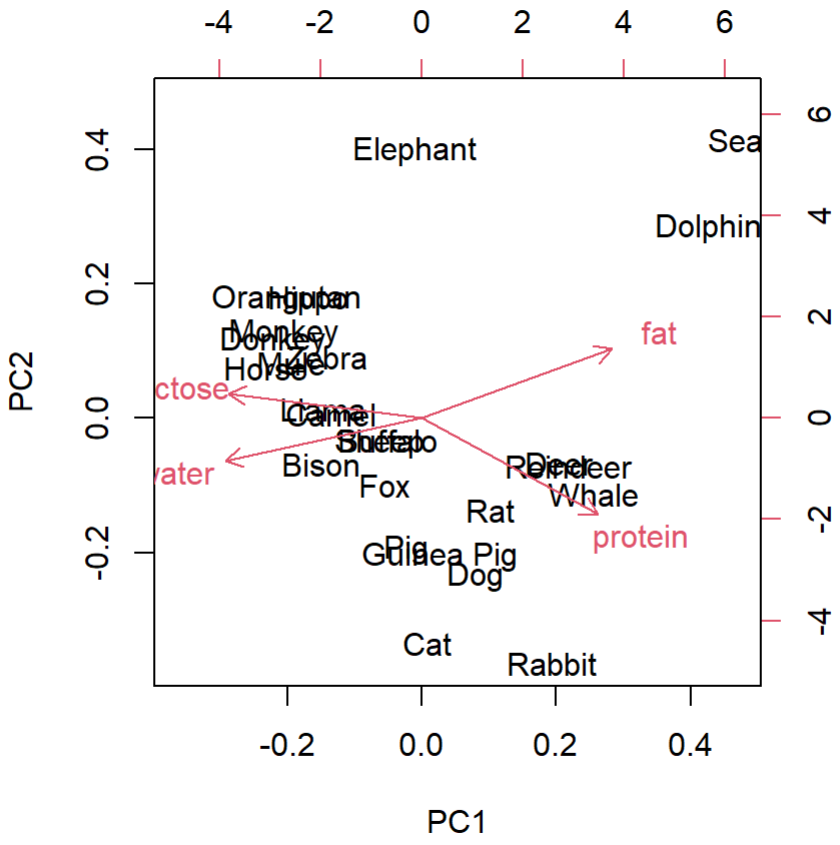


La screeplot nos indica que elijamos 2 componentes principales, sin embargo, los otros 2 métodos anteriormente mencionados nos indican que elijamos sólo 1. Entonces nos quedaremos con 1 componente principal para reducir el número de variables de 4 a sólo 1.

```
biplot(res,scale = 0)
```




```
biplot(res,scale = 1)
```




```
res$x[,1]
```

```
##       Horse   Orangutan      Monkey      Donkey       Hippo       Camel
## -2.15955785 -1.87741415 -1.92107037 -2.06824874 -1.56283420 -1.24776651
##       Bison     Buffalo  Guinea Pig         Cat         Fox       Llama
## -1.38164444 -0.47185521  0.25117884  0.09087423 -0.51370377 -1.38058058
##        Mule         Pig       Zebra       Sheep         Dog    Elephant
## -1.80370758 -0.20253900 -1.32306955 -0.57776326  0.74672828 -0.08997063
##      Rabbit         Rat        Deer    Reindeer       Whale        Seal
##  1.81036875  0.95263023  1.90757130  2.04694152  2.39169651  4.39921084
##     Dolphin
##  3.98452536
```

2. Realiza un análisis de clustering para poder determinar qué tipos de leche son parecidos entre sí tomando en cuenta las características de cada tipo de leche.

Por k-medias:

```
datoclu<-scale(dato)
kme<-kmeans(datoclu,2)
kme$cluster
```

```
##      Horse  Orangutan     Monkey     Donkey      Hippo      Camel      Bison
##          2          2          2          2          2          2          2
##    Buffalo Guinea Pig        Cat        Fox      Llama       Mule        Pig
##          2          2          2          2          2          2          2
##      Zebra      Sheep        Dog   Elephant     Rabbit        Rat       Deer
##          2          2          1          2          1          1          1
##   Reindeer      Whale       Seal    Dolphin
##          1          1          1          1
```

```
kme$centers
```

```
##        water    protein        fat    lactose
## 1 -1.1563115  1.1500194  1.0795306 -1.1775112
## 2  0.5441466 -0.5411856 -0.5080144  0.5541229
```

```
kme$totss
```

```
## [1] 96
```

```
kme$withinss
```

```
## [1] 17.05673 17.63127
```

```
kme$tot.withinss
```

```
## [1] 34.688
```

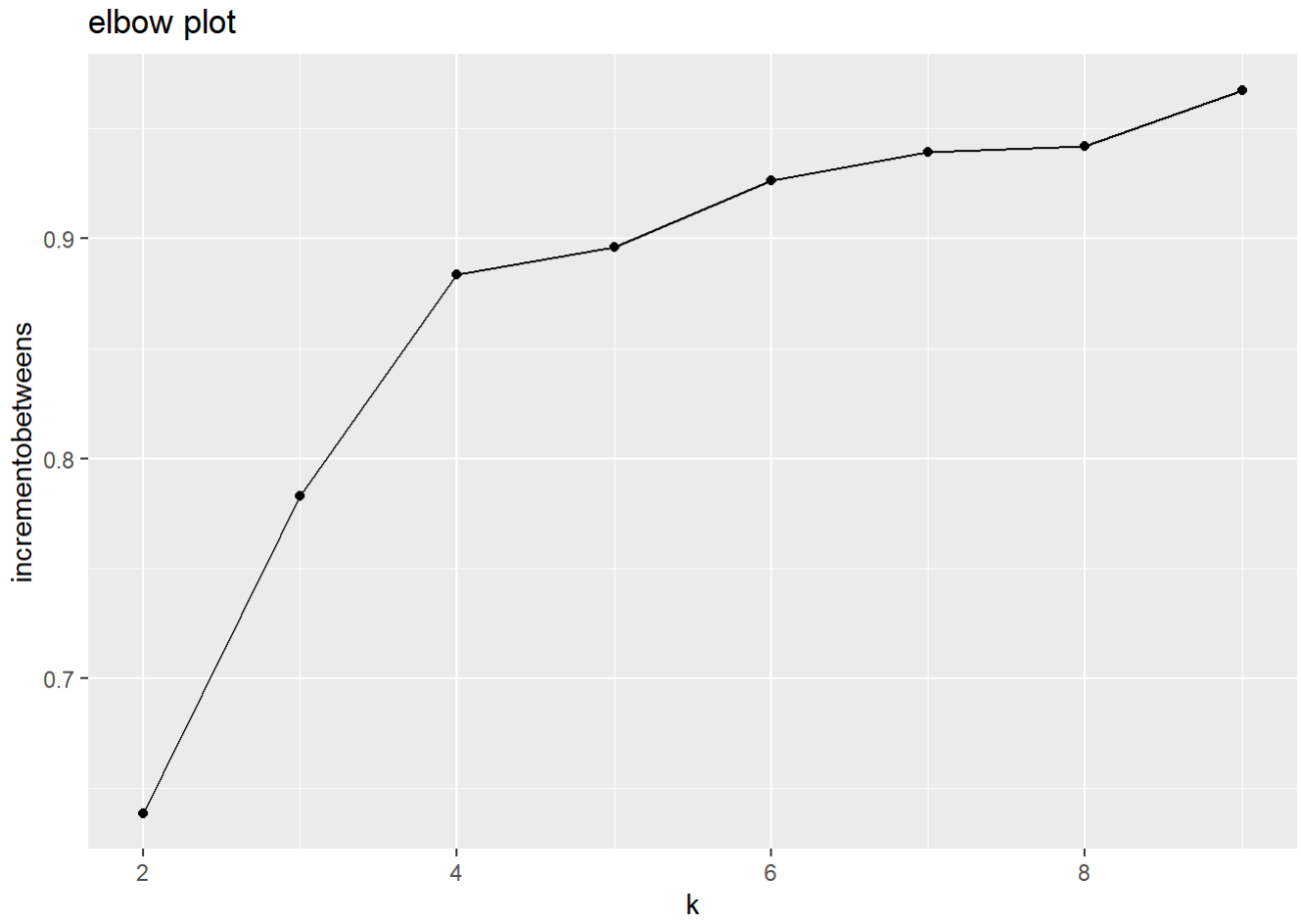
```
kme$betweenss
```

```
## [1] 61.312
```

```
kme$size
```

```
## [1]  8 17
```

```
kme1<-kmeans(datoclu,3)
kme2<-kmeans(datoclu,4)
kme3<-kmeans(datoclu,5)
kme4<-kmeans(datoclu,6)
kme5<-kmeans(datoclu,7)
kme6<-kmeans(datoclu,8)
kme7<-kmeans(datoclu,9)
elbow<-data.frame(k=c(2:9),incrementobetweens=c(kme$betweenss/kme$totss,
                                                kme1$betweenss/kme1$totss,
                                                kme2$betweenss/kme2$totss,
                                                kme3$betweenss/kme3$totss,
                                                kme4$betweenss/kme4$totss,
                                                kme5$betweenss/kme5$totss,
                                                kme6$betweenss/kme6$totss,
                                                kme7$betweenss/kme7$totss))
ggplot(elbow,aes(x=k,y=incrementobetweens))+geom_point()+geom_line()+
  ggtitle("elbow plot")
```



Mediante el método de k-medias, al observar la elbow-plot concluimos que usar k=4 clusters para agruparlos datos es la mejor forma de hacerlo puesto que después de k=4 la Suma de Cuadrados entre k clusters como proporción de la suma de cuadrados de todos los datos, crece de forma más lenta que antes.

```
kme2$cluster
```

```
##      Horse  Orangutan     Monkey     Donkey      Hippo      Camel      Bison
##          2          2          2          2          2          2          2
##    Buffalo Guinea Pig        Cat        Fox      Llama       Mule        Pig
##          1          1          1          1          2          2          1
##      Zebra      Sheep        Dog   Elephant     Rabbit        Rat       Deer
##          2          1          3          1          3          3          3
##   Reindeer      Whale       Seal    Dolphin
##          3          3          4          4
```

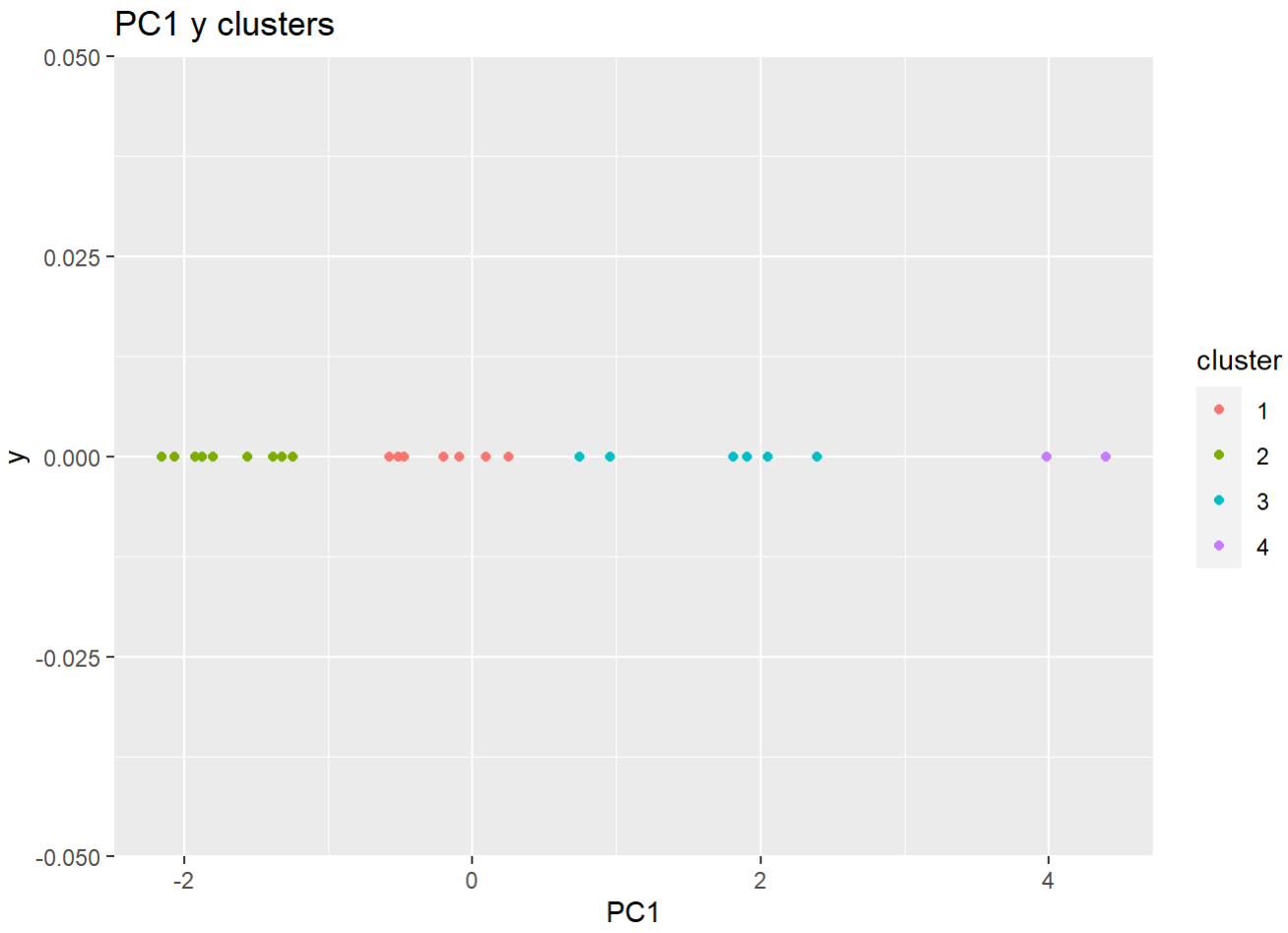
```
grafico<-data.frame(res$x[,1],as.factor(kme2$cluster))
names(grafico)<-c("PC1","cluster")
```

Así quedarían agrupados los tipos de leche en su respectivo cluster y tomando en cuenta la elección del primer componente principal:

```
grafico
```

```
##                   PC1 cluster
## Horse      -2.15955785       2
## Orangutan  -1.87741415       2
## Monkey     -1.92107037       2
## Donkey     -2.06824874       2
## Hippo      -1.56283420       2
## Camel      -1.24776651       2
## Bison      -1.38164444       2
## Buffalo    -0.47185521       1
## Guinea Pig  0.25117884       1
## Cat         0.09087423       1
## Fox        -0.51370377       1
## Llama      -1.38058058       2
## Mule       -1.80370758       2
## Pig        -0.20253900       1
## Zebra      -1.32306955       2
## Sheep      -0.57776326       1
## Dog         0.74672828       3
## Elephant   -0.08997063       1
## Rabbit      1.81036875       3
## Rat         0.95263023       3
## Deer        1.90757130       3
## Reindeer    2.04694152       3
## Whale       2.39169651       3
## Seal        4.39921084       4
## Dolphin     3.98452536       4
```

```
ggplot(grafico,aes(x=PC1,y=0,col=cluster))+geom_point()+ggtitle("PC1 y clusters")
```



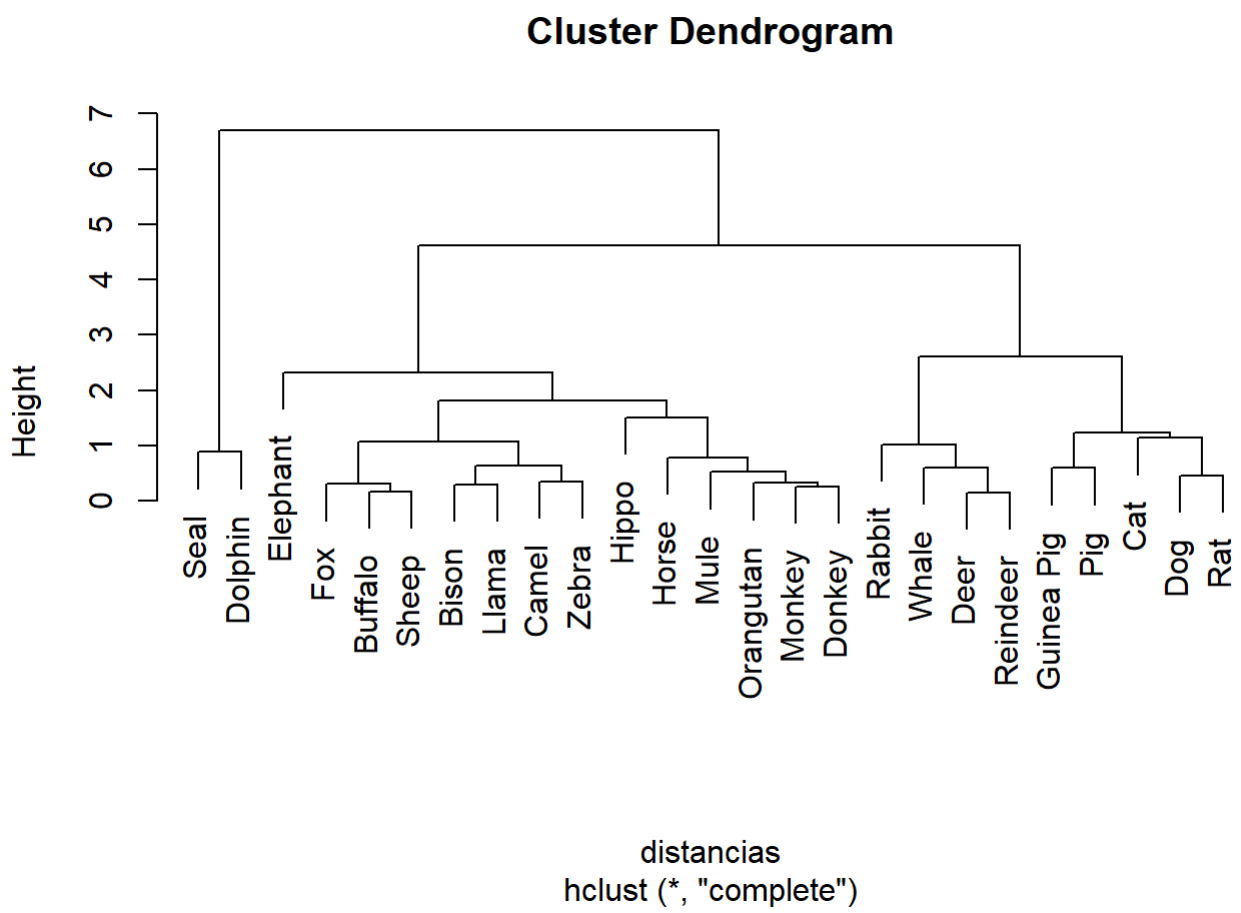
Veamos que resultados nos arroja clustering jerárquico

```
distancias<-dist(datoclu)
distancias #nos da la matriz de proximidad
```

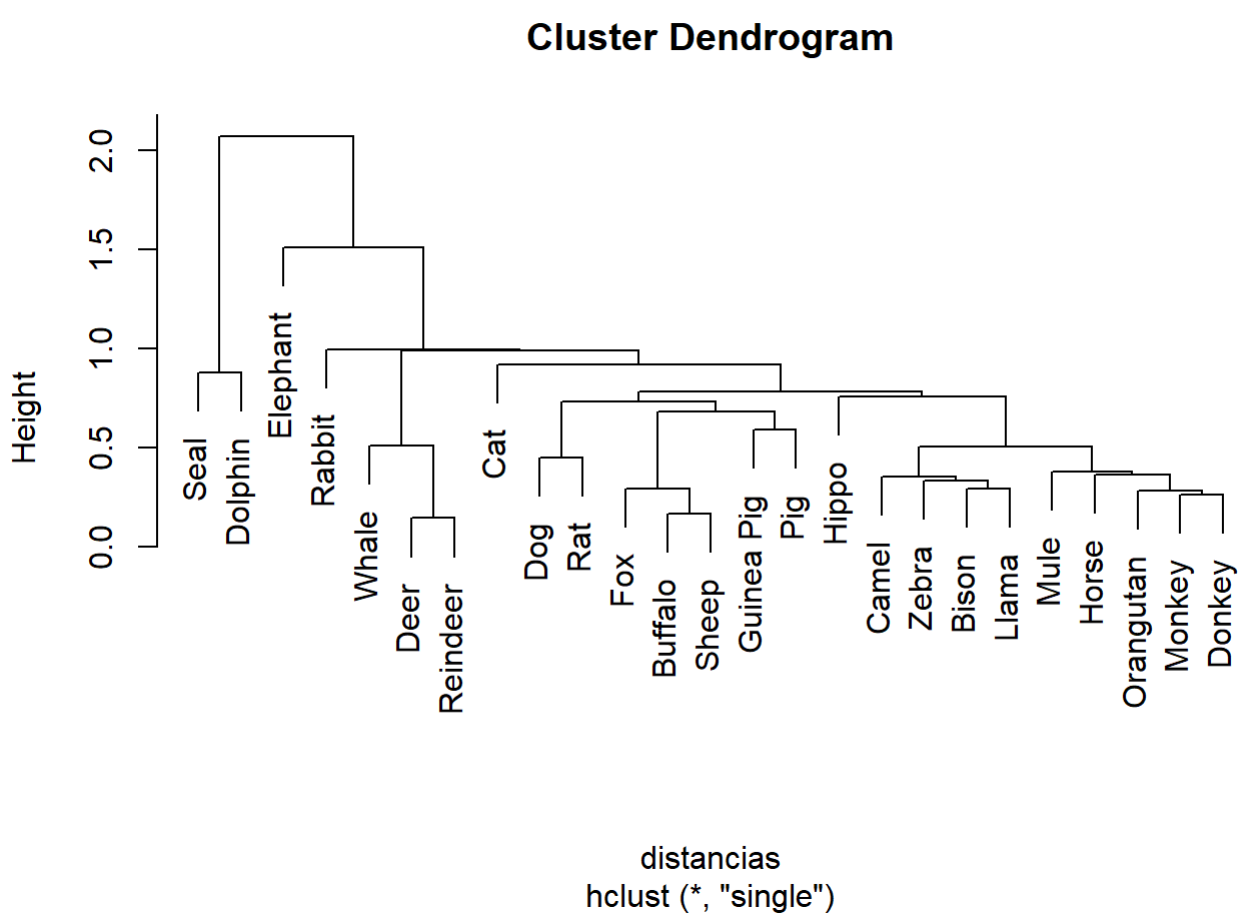
```
##               Horse Orangutan    Monkey    Donkey     Hippo     Camel
## Orangutan  0.6491545
## Monkey     0.3608445 0.3185899
## Donkey     0.4565402 0.2797317 0.2605916
## Hippo      1.5078669 0.9175442 1.1989823 1.0691939
## Camel      1.2091567 0.8738859 0.9470964 0.9509250 0.8563715
## Bison      0.9266481 0.9685994 0.8218657 0.9305943 1.4040273 0.6309379
## Buffalo    1.7545751 1.5639066 1.5405821 1.6685067 1.6297654 0.8992473
## Guinea Pig 2.7821938 2.5164705 2.5586648 2.6114149 2.1983793 1.6702045
## Cat        2.6024052 2.6071045 2.5037271 2.6329047 2.6954467 1.9013883
## Fox        1.7458863 1.6520453 1.5797709 1.7167025 1.8060962 1.0031368
## Llama      0.8677227 0.7357432 0.6569488 0.7663210 1.1632968 0.4602663
## Mule       0.7854480 0.3759051 0.5170000 0.3934014 0.7578759 0.6082880
## Pig        2.2462812 2.0573504 2.0530775 2.1247109 1.9151340 1.2264930
## Zebra      0.9989991 0.6207951 0.6912923 0.7582027 0.8839202 0.3530623
## Sheep      1.6673004 1.4693669 1.4494108 1.5671266 1.5371474 0.7827135
## Dog        3.1160402 2.9311146 2.9210670 3.0277233 2.7737297 2.1481225
## Elephant   2.3152018 2.0337173 2.0618634 2.2557310 2.2398814 1.9424586
## Rabbit     4.2399902 4.0680191 4.0544318 4.1673599 3.8745685 3.2860918
## Rat        3.1982759 3.0065778 2.9935581 3.1280787 2.9064746 2.2972058
## Deer       4.0982876 3.8726534 3.8777500 4.0232388 3.7291011 3.2101738
## Reindeer   4.2367764 4.0129219 4.0172433 4.1638754 3.8687858 3.3519725
## Whale      4.6228727 4.3672333 4.3917011 4.5200115 4.1364383 3.6656383
## Seal       6.6921648 6.3307399 6.4125162 6.5484788 6.0391626 5.7963135
## Dolphin    6.1926142 5.8803085 5.9323779 6.0825891 5.6657303 5.3319727
##                Bison   Buffalo Guinea Pig       Cat       Fox     Llama
## Orangutan
## Monkey
## Donkey
## Hippo
## Camel
## Bison
## Buffalo    0.9361670
## Guinea Pig 1.9011545 1.1684869
## Cat        1.7237747 1.1720550  1.1897681
## Fox        0.8740745 0.2938158  1.2272576 0.9970776
## Llama      0.2915595 0.9266740  1.9229902 1.8824077 0.9510461
## Mule       0.8112702 1.4307411  2.2770679 2.4271348 1.5082344 0.6048543
## Pig        1.3399119 0.6926736  0.5912264 0.9178304 0.6800163 1.3997578
## Zebra      0.6167576 0.9627290  1.9053353 2.0418287 1.0764956 0.3335023
## Sheep      0.8317423 0.1647584  1.2003090 1.2432823 0.3001839 0.8208682
## Dog        2.2190491 1.3984389  0.7321903 0.9463622 1.3821589 2.2786564
## Elephant   1.9983670 1.5097715  2.3086366 2.3395294 1.6676786 1.8440600
## Rabbit     3.3429771 2.5220928  1.7288611 1.8135701 2.4963423 3.4147949
## Rat        2.3460041 1.4700270  1.0730926 1.1331793 1.4762799 2.3819428
## Deer       3.2904953 2.3849631  1.9100028 2.0237695 2.4249810 3.3004222
## Reindeer   3.4293974 2.5260360  2.0394043 2.1429750 2.5629631 3.4407969
## Whale      3.7939995 2.8829267  2.2222031 2.4770960 2.9320705 3.8011405
## Seal       6.0121509 5.0924949  4.6028697 4.9843057 5.2150736 5.9382068
## Dolphin    5.4852213 4.5794512  4.1232545 4.3887272 4.6655427 5.4373060
##                Mule       Pig     Zebra     Sheep       Dog  Elephant
## Orangutan
## Monkey
```

```
## Donkey
## Hippo
## Camel
## Bison
## Buffalo
## Guinea Pig
## Cat
## Fox
## Llama
## Mule
## Pig        1.8245889
## Zebra      0.5060818 1.4470981
## Sheep      1.3202141 0.6970171 0.8628407
## Dog        2.7450482 0.9700345 2.3125092 1.4737761
## Elephant   2.1722299 2.0730500 1.7312062 1.5659762 2.2877529
## Rabbit     3.8854757 2.0918458 3.4499180 2.6084087 1.1421511 3.1525274
## Rat        2.8716588 1.2379063 2.4011826 1.5659841 0.4498072 2.0427438
## Deer       3.7718079 2.2004625 3.2851302 2.4935611 1.3180053 2.5155835
## Reindeer   3.9133583 2.3368823 3.4263100 2.6342133 1.4424611 2.6306244
## Whale      4.2434691 2.6131609 3.7704372 2.9852778 1.6938947 3.0517965
## Seal       6.2957855 4.9958737 5.8211682 5.1973850 4.2051374 4.5945970
## Dolphin    5.8423549 4.4748759 5.3544603 4.6794011 3.6433081 4.1246393
##               Rabbit       Rat      Deer  Reindeer     Whale      Seal
## Orangutan
## Monkey
## Donkey
## Hippo
## Camel
## Bison
## Buffalo
## Guinea Pig
## Cat
## Fox
## Llama
## Mule
## Pig
## Zebra
## Sheep
## Dog
## Elephant
## Rabbit
## Rat        1.1469320
## Deer       0.9939061 0.9872985
## Reindeer   1.0123839 1.1207959 0.1426357
## Whale      0.9924608 1.4701194 0.6020307 0.5105902
## Seal       3.5924765 3.9017197 2.9767088 2.8730917 2.6232056
## Dolphin    3.0087349 3.3158739 2.3741973 2.2584882 2.0667967 0.8783319
```

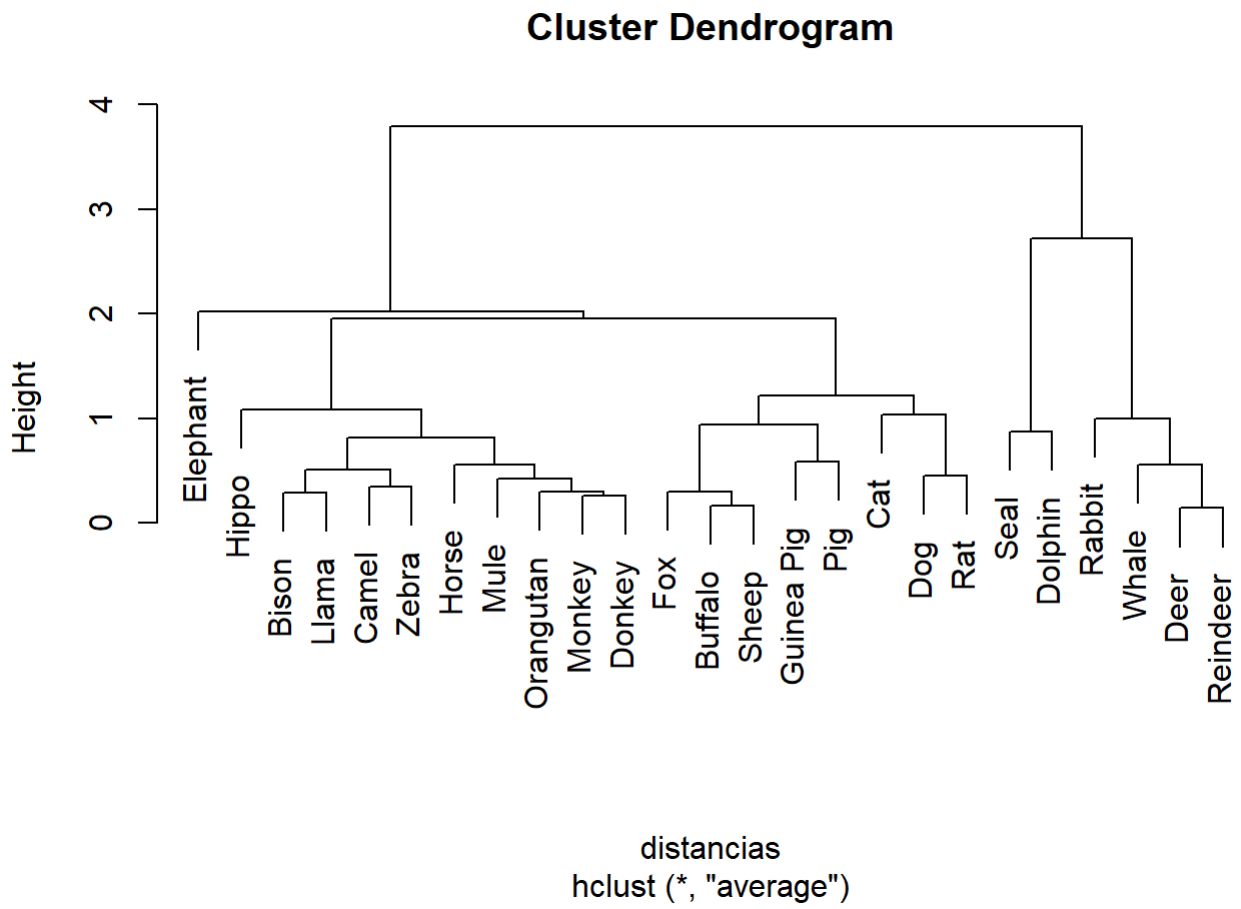
```
dendo<-hclust(distancias) #por default es complete
dendo1<-hclust(distancias,method = "single")
dendo2<-hclust(distancias,method = "average")
dendo3<-hclust(distancias,method = "centroid")
plot(dendo)
```


Cluster Dendrogram
Height
0
1
2
3
4
5
6
7
Seal
Dolphin
Elephant
Fox
Buffalo
Sheep
Bison
Llama
Camel
Zebra
Hippo
Horse
Mule
Orangutan
Monkey
Donkey
Rabbit
Whale
Deer
Reindeer
Guinea Pig
Pig
Cat
Dog
Rat
distancias
hclust (*, "complete")

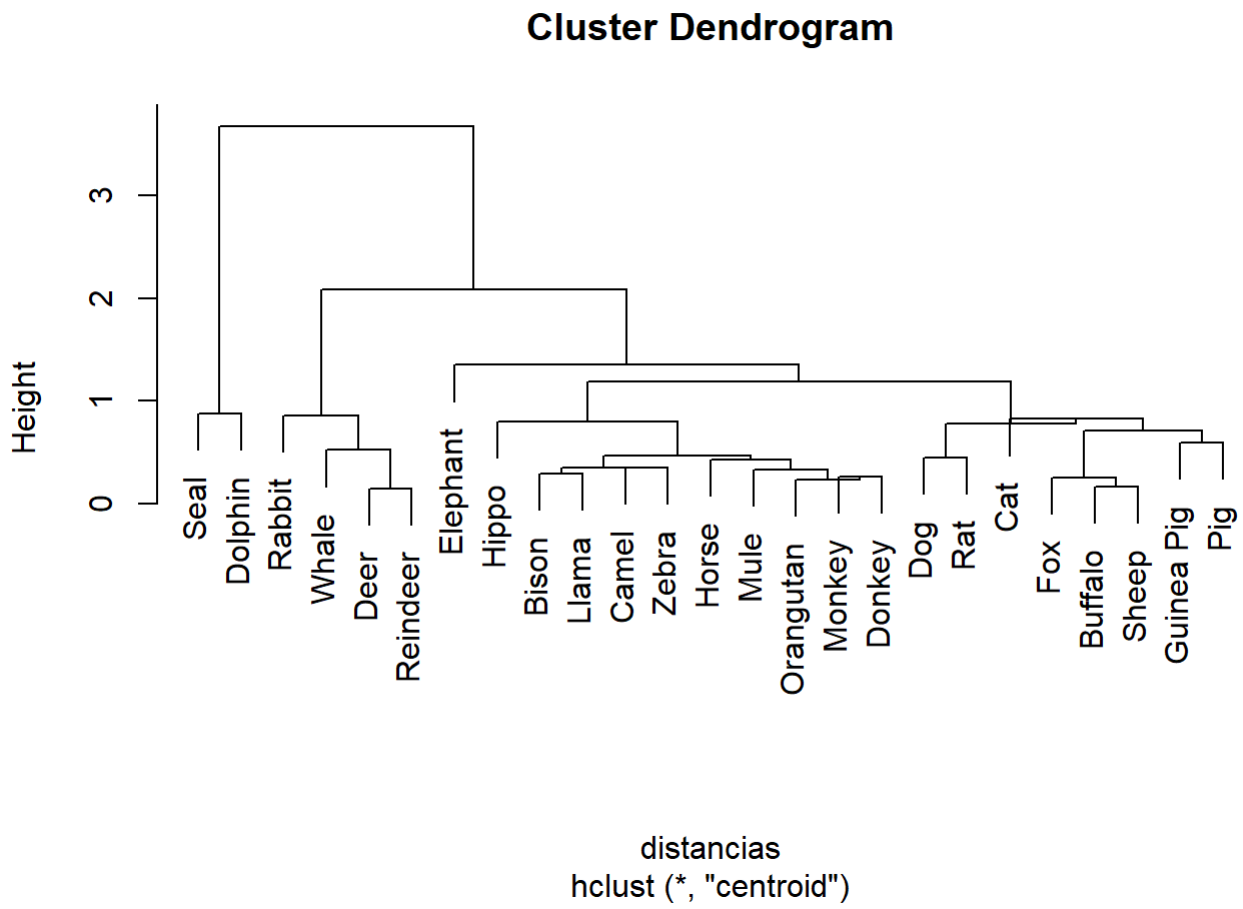
```
plot(dendo1)
```


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
Seal
Dolphin
Elephant
Rabbit
Whale
Deer
Reindeer
Cat
Dog
Rat
Fox
Buffalo
Sheep
Guinea Pig
Pig
Hippo
Camel
Zebra
Bison
Llama
Mule
Horse
Orangutan
Monkey
Donkey
distancias
hclust (*, "single")

```
plot(dendo2)
```


Cluster Dendrogram
Height
0
1
2
3
4
Elephant
Hippo
Bison
Llama
Camel
Zebra
Horse
Mule
Orangutan
Monkey
Donkey
Fox
Buffalo
Sheep
Guinea Pig
Pig
Cat
Dog
Rat
Seal
Dolphin
Rabbit
Whale
Deer
Reindeer
distancias
hclust (*, "average")

```
plot(dendo3)
```

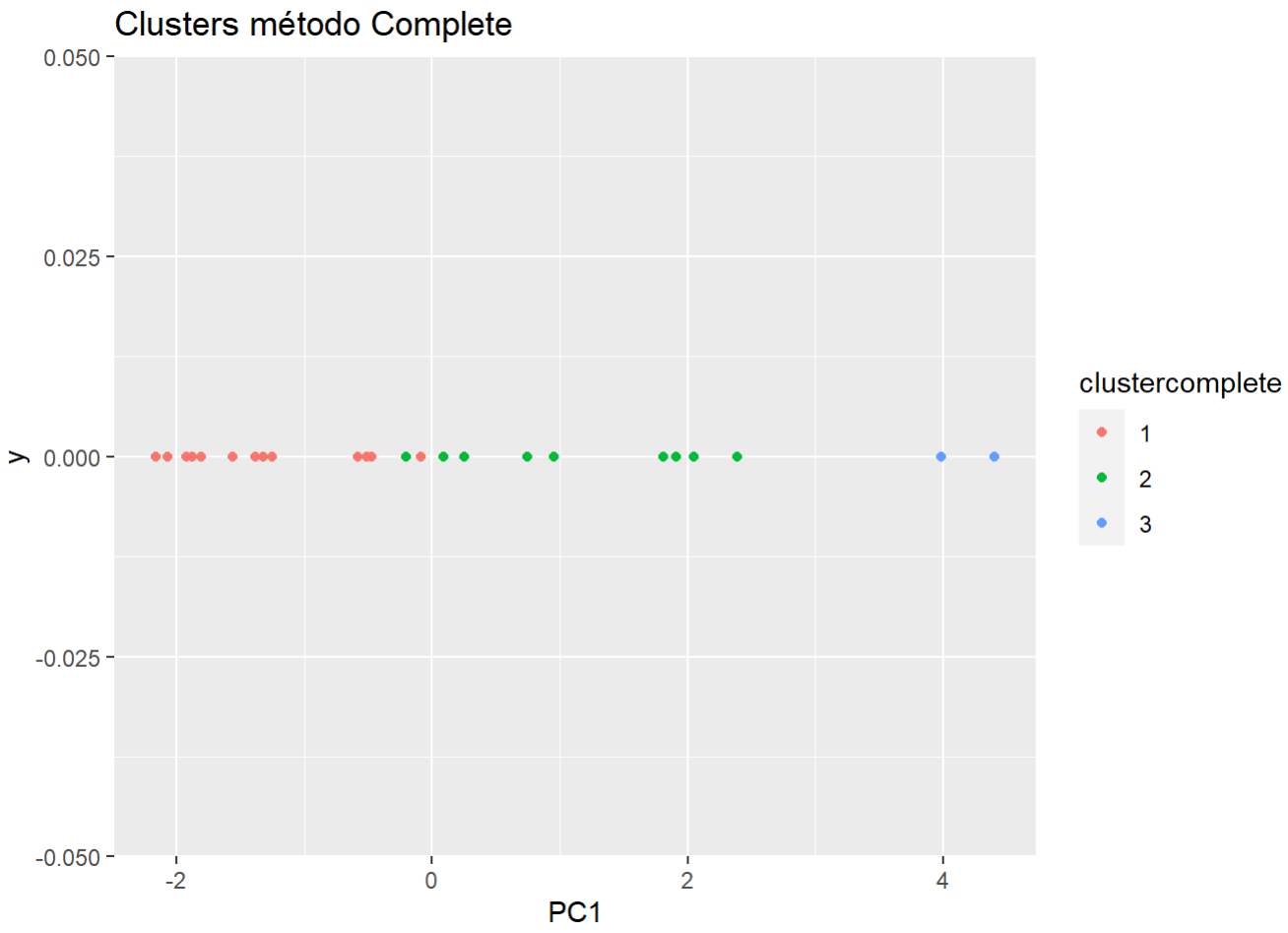


Nos basaremos en los dendogramas calculados con los método complete y average

```
grupos1<-cutree(dendo,3)
grafo1<-data.frame(res$x[,1],as.factor(grupos1))
names(grafo1)<-c("PC1","clustercomplete")
grafo1
```

```
##                    PC1 clustercomplete
## Horse      -2.15955785               1
## Orangutan  -1.87741415               1
## Monkey     -1.92107037               1
## Donkey     -2.06824874               1
## Hippo      -1.56283420               1
## Camel      -1.24776651               1
## Bison      -1.38164444               1
## Buffalo    -0.47185521               1
## Guinea Pig  0.25117884               2
## Cat         0.09087423               2
## Fox        -0.51370377               1
## Llama      -1.38058058               1
## Mule       -1.80370758               1
## Pig        -0.20253900               2
## Zebra      -1.32306955               1
## Sheep      -0.57776326               1
## Dog         0.74672828               2
## Elephant   -0.08997063               1
## Rabbit      1.81036875               2
## Rat         0.95263023               2
## Deer        1.90757130               2
## Reindeer    2.04694152               2
## Whale       2.39169651               2
## Seal        4.39921084               3
## Dolphin     3.98452536               3
```

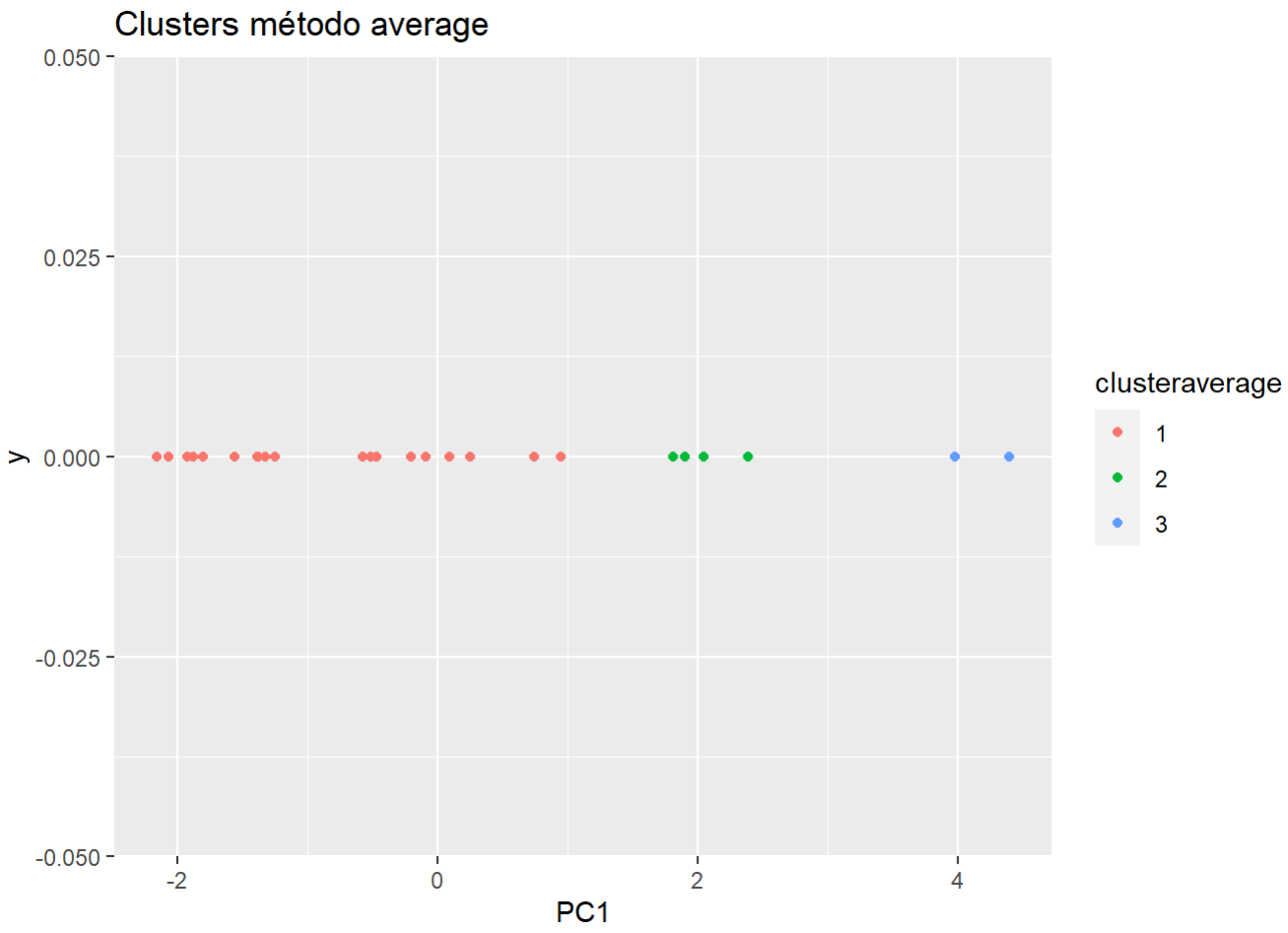
```
ggplot(grafo1,aes(x=PC1,y=0,col=clustercomplete))+geom_point()+ggtitle("Clusters método Complet
e")
```




```
grupos<-cutree(dendo2,3)
grafo<-data.frame(res$x[,1],as.factor(grupos))
names(grafo)<-c("PC1","clusteraverage")
grafo
```

```
##                    PC1 clusteraverage
## Horse      -2.15955785              1
## Orangutan  -1.87741415              1
## Monkey     -1.92107037              1
## Donkey     -2.06824874              1
## Hippo      -1.56283420              1
## Camel      -1.24776651              1
## Bison      -1.38164444              1
## Buffalo    -0.47185521              1
## Guinea Pig  0.25117884              1
## Cat         0.09087423              1
## Fox        -0.51370377              1
## Llama      -1.38058058              1
## Mule       -1.80370758              1
## Pig        -0.20253900              1
## Zebra      -1.32306955              1
## Sheep      -0.57776326              1
## Dog         0.74672828              1
## Elephant   -0.08997063              1
## Rabbit      1.81036875              2
## Rat         0.95263023              1
## Deer        1.90757130              2
## Reindeer    2.04694152              2
## Whale       2.39169651              2
## Seal        4.39921084              3
## Dolphin     3.98452536              3
```

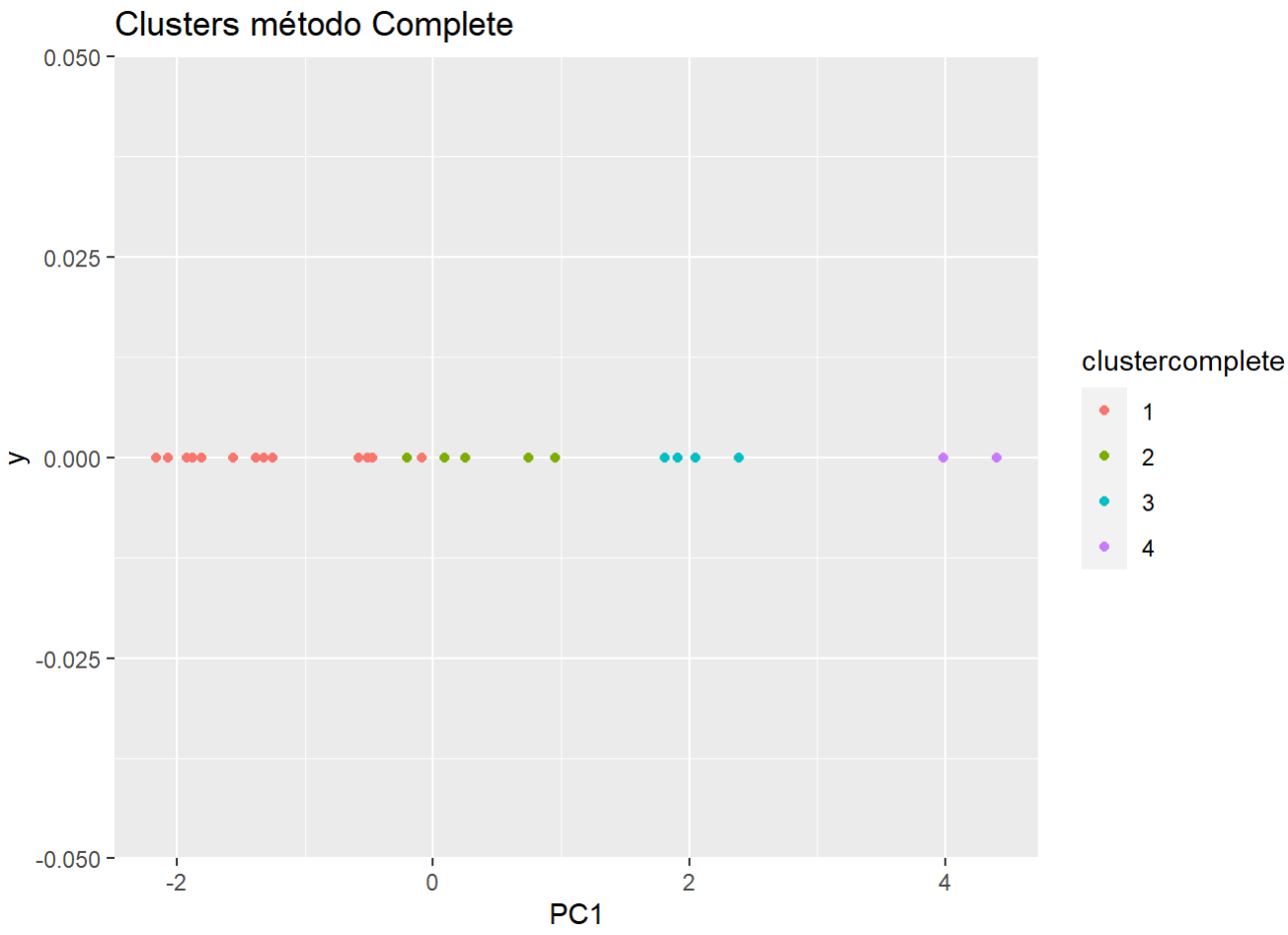
```
ggplot(grafo,aes(x=PC1,y=0,col=clusteraverage))+geom_point()+ggtitle("Clusters método average")
```




```
grupos1<-cutree(dendo,4)
grafo1<-data.frame(res$x[,1],as.factor(grupos1))
names(grafo1)<-c("PC1","clustercomplete")
grafo1
```

```
##                    PC1 clustercomplete
## Horse      -2.15955785               1
## Orangutan  -1.87741415               1
## Monkey     -1.92107037               1
## Donkey     -2.06824874               1
## Hippo      -1.56283420               1
## Camel      -1.24776651               1
## Bison      -1.38164444               1
## Buffalo    -0.47185521               1
## Guinea Pig  0.25117884               2
## Cat         0.09087423               2
## Fox        -0.51370377               1
## Llama      -1.38058058               1
## Mule       -1.80370758               1
## Pig        -0.20253900               2
## Zebra      -1.32306955               1
## Sheep      -0.57776326               1
## Dog         0.74672828               2
## Elephant   -0.08997063               1
## Rabbit      1.81036875               3
## Rat         0.95263023               2
## Deer        1.90757130               3
## Reindeer    2.04694152               3
## Whale       2.39169651               3
## Seal        4.39921084               4
## Dolphin     3.98452536               4
```

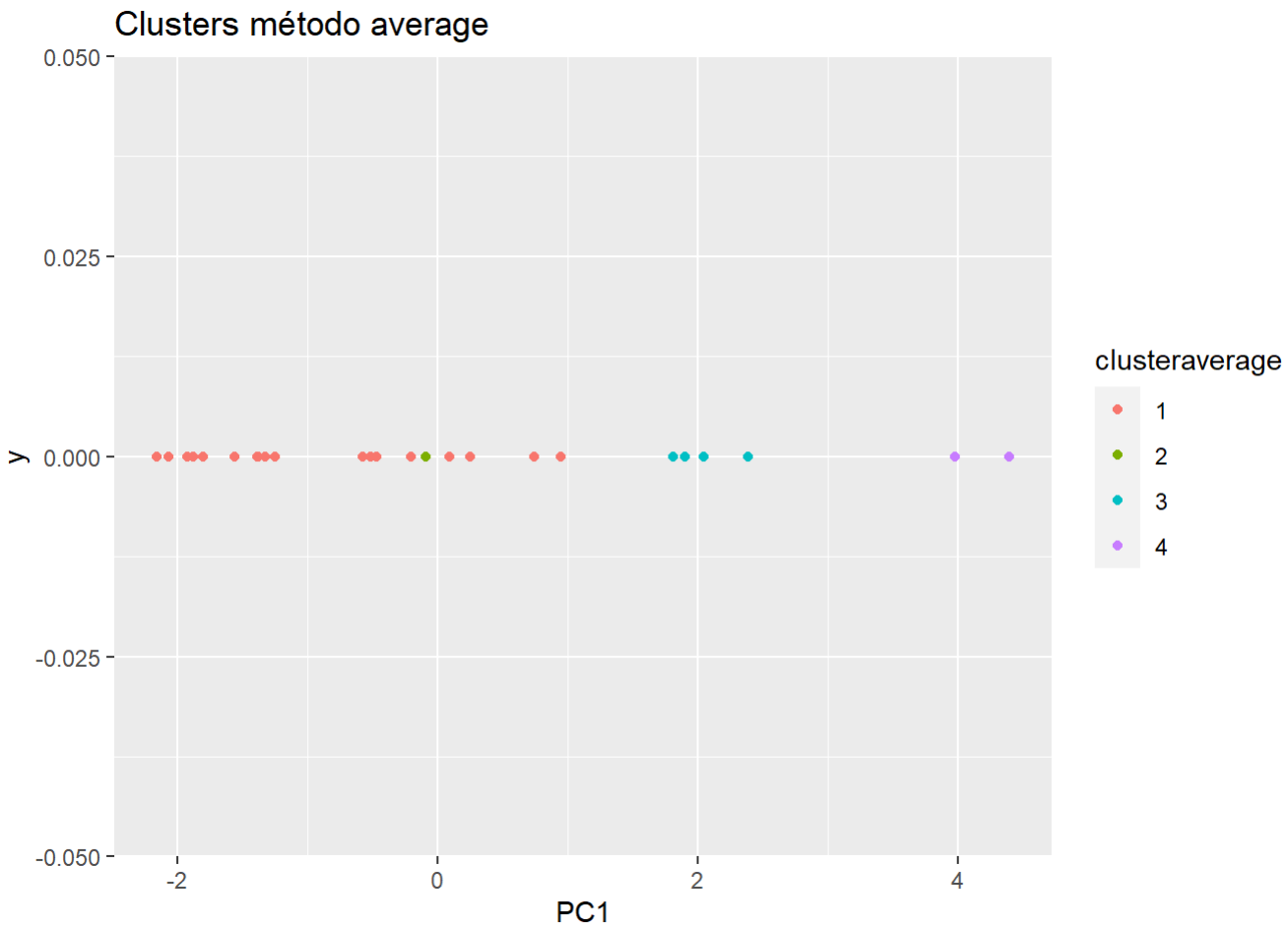
```
ggplot(grafo1,aes(x=PC1,y=0,col=clustercomplete))+geom_point()+ggtitle("Clusters método Complet
e")
```




```
grupos<-cutree(dendo2,4)
grafo<-data.frame(res$x[,1],as.factor(grupos))
names(grafo)<-c("PC1","clusteraverage")
grafo
```

```
##                    PC1 clusteraverage
## Horse      -2.15955785              1
## Orangutan  -1.87741415              1
## Monkey     -1.92107037              1
## Donkey     -2.06824874              1
## Hippo      -1.56283420              1
## Camel      -1.24776651              1
## Bison      -1.38164444              1
## Buffalo    -0.47185521              1
## Guinea Pig  0.25117884              1
## Cat         0.09087423              1
## Fox        -0.51370377              1
## Llama      -1.38058058              1
## Mule       -1.80370758              1
## Pig        -0.20253900              1
## Zebra      -1.32306955              1
## Sheep      -0.57776326              1
## Dog         0.74672828              1
## Elephant   -0.08997063              2
## Rabbit      1.81036875              3
## Rat         0.95263023              1
## Deer        1.90757130              3
## Reindeer    2.04694152              3
## Whale       2.39169651              3
## Seal        4.39921084              4
## Dolphin     3.98452536              4
```

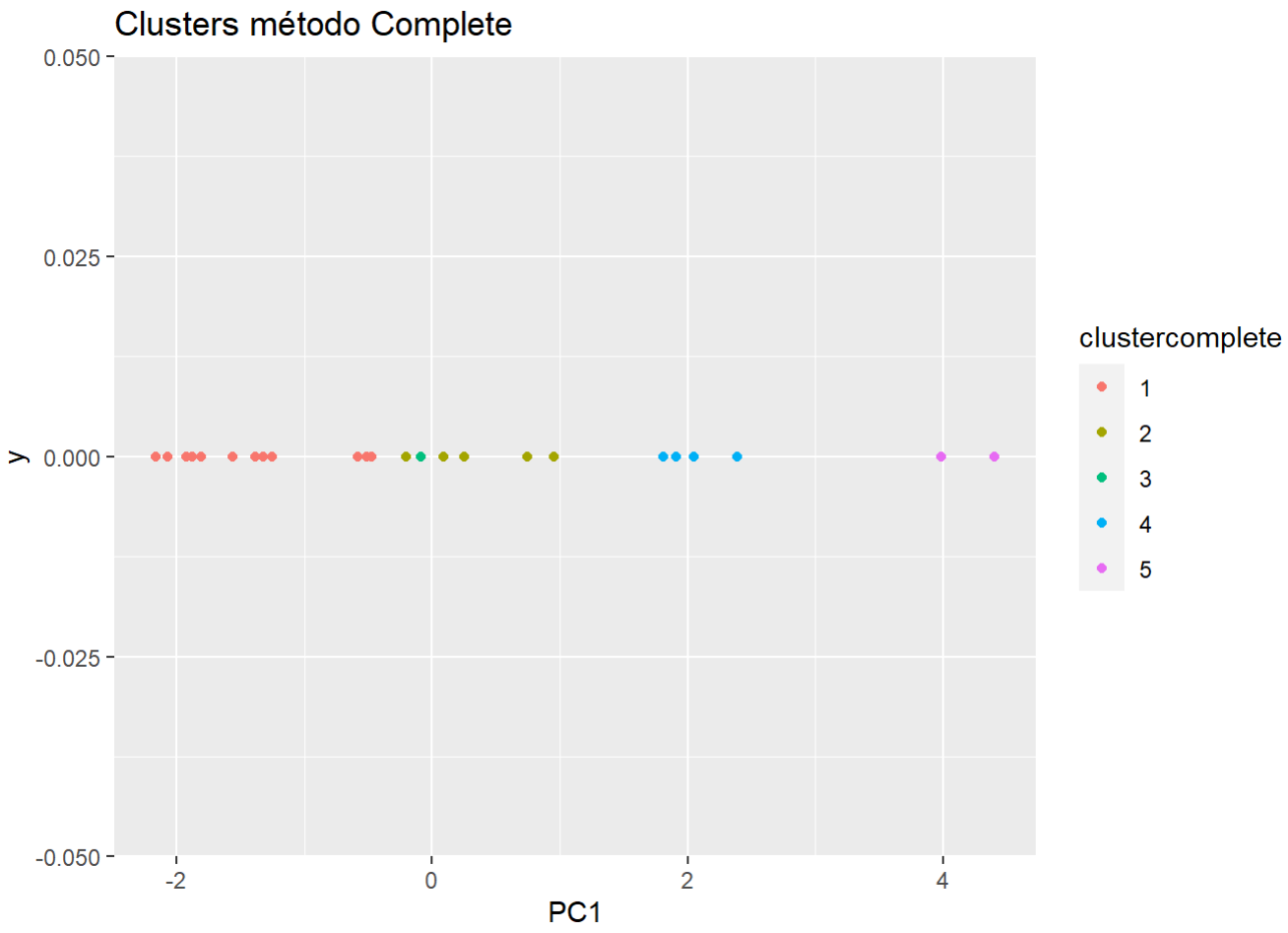
```
ggplot(grafo,aes(x=PC1,y=0,col=clusteraverage))+geom_point()+ggtitle("Clusters método average")
```




```
grupos1<-cutree(dendo,5)
grafo1<-data.frame(res$x[,1],as.factor(grupos1))
names(grafo1)<-c("PC1","clustercomplete")
grafo1
```

```
##                    PC1 clustercomplete
## Horse      -2.15955785               1
## Orangutan  -1.87741415               1
## Monkey     -1.92107037               1
## Donkey     -2.06824874               1
## Hippo      -1.56283420               1
## Camel      -1.24776651               1
## Bison      -1.38164444               1
## Buffalo    -0.47185521               1
## Guinea Pig  0.25117884               2
## Cat         0.09087423               2
## Fox        -0.51370377               1
## Llama      -1.38058058               1
## Mule       -1.80370758               1
## Pig        -0.20253900               2
## Zebra      -1.32306955               1
## Sheep      -0.57776326               1
## Dog         0.74672828               2
## Elephant   -0.08997063               3
## Rabbit      1.81036875               4
## Rat         0.95263023               2
## Deer        1.90757130               4
## Reindeer    2.04694152               4
## Whale       2.39169651               4
## Seal        4.39921084               5
## Dolphin     3.98452536               5
```

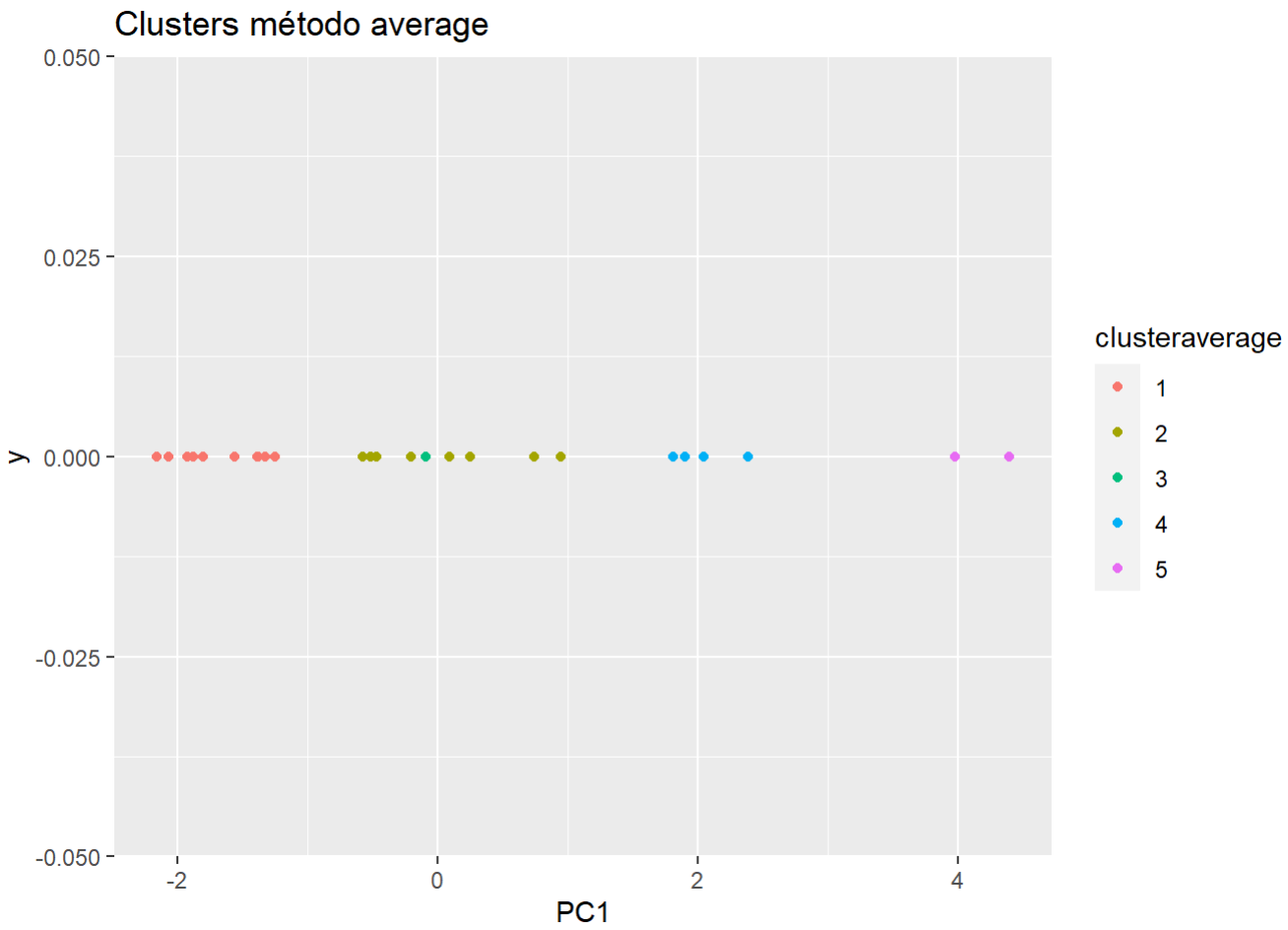
```
ggplot(grafo1,aes(x=PC1,y=0,col=clustercomplete))+geom_point()+ggtitle("Clusters método Complet
e")
```




```
grupos<-cutree(dendo2,5)
grafo<-data.frame(res$x[,1],as.factor(grupos))
names(grafo)<-c("PC1","clusteraverage")
grafo
```

```
##                    PC1 clusteraverage
## Horse      -2.15955785              1
## Orangutan  -1.87741415              1
## Monkey     -1.92107037              1
## Donkey     -2.06824874              1
## Hippo      -1.56283420              1
## Camel      -1.24776651              1
## Bison      -1.38164444              1
## Buffalo    -0.47185521              2
## Guinea Pig  0.25117884              2
## Cat         0.09087423              2
## Fox        -0.51370377              2
## Llama      -1.38058058              1
## Mule       -1.80370758              1
## Pig        -0.20253900              2
## Zebra      -1.32306955              1
## Sheep      -0.57776326              2
## Dog         0.74672828              2
## Elephant   -0.08997063              3
## Rabbit      1.81036875              4
## Rat         0.95263023              2
## Deer        1.90757130              4
## Reindeer    2.04694152              4
## Whale       2.39169651              4
## Seal        4.39921084              5
## Dolphin     3.98452536              5
```

```
ggplot(grafo,aes(x=PC1,y=0,col=clusteraverage))+geom_point()+ggtitle("Clusters método average")
```



Después de observar los dendogramas con los métodos average y complete, si graficamos y observamos como es la agrupación con 3,4 y 5 clusters, podemos concluir que la mejor agrupación se alcanza con el método average usando 3 clusters. Si comparamos este resultado con lo visto en el método de k-medias, el número de clusters escogidos que agrupa de mejor forma los tipos de leche es de 4 que fue el obtenido por k-medias. El resultado se muestra una vez más a continuación.

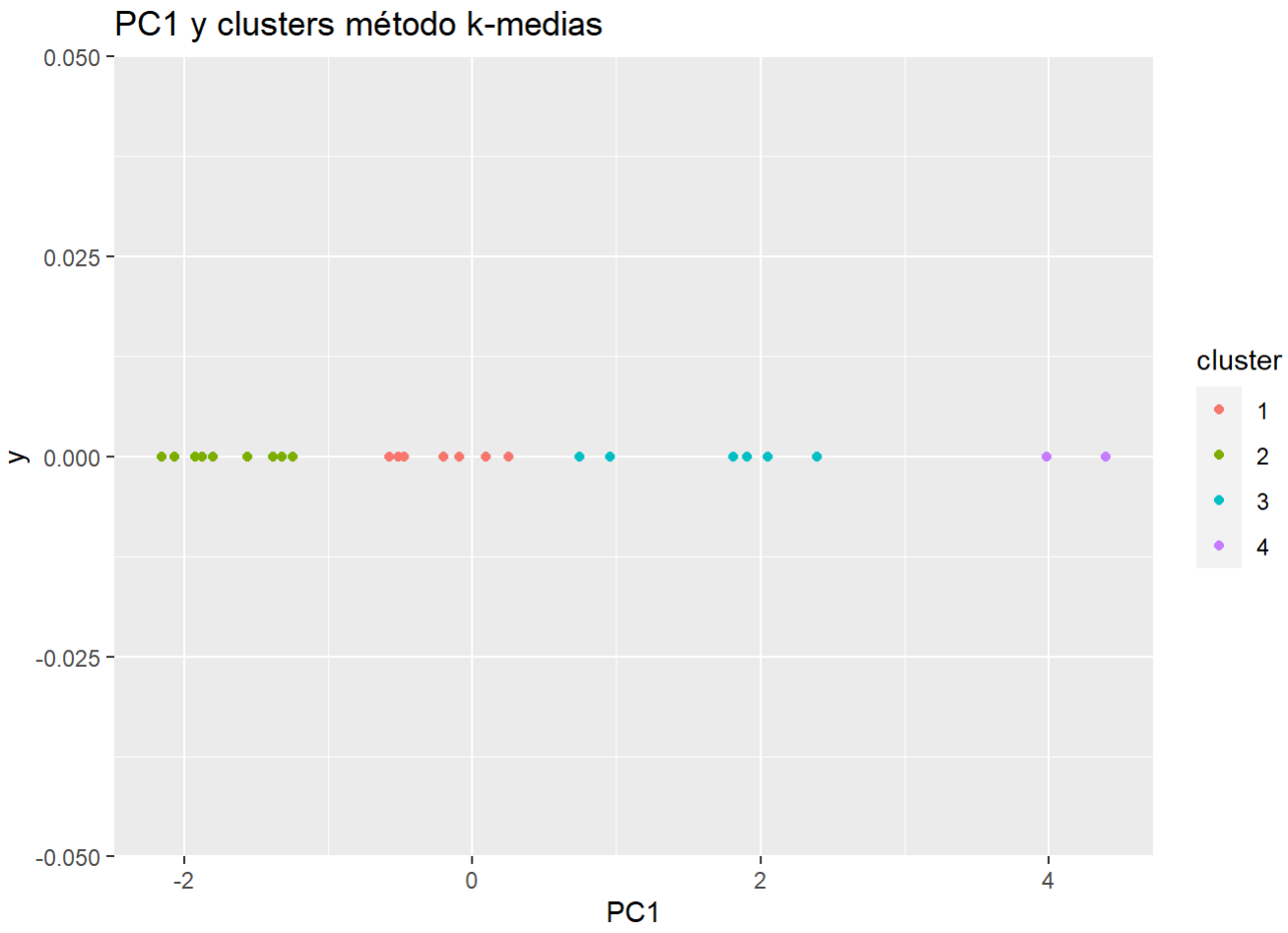
```
kme2$cluster
```

```
##      Horse  Orangutan     Monkey     Donkey      Hippo      Camel      Bison
##          2          2          2          2          2          2          2
##    Buffalo Guinea Pig        Cat        Fox      Llama       Mule        Pig
##          1          1          1          1          2          2          1
##      Zebra      Sheep        Dog   Elephant     Rabbit        Rat       Deer
##          2          1          3          1          3          3          3
##   Reindeer      Whale       Seal    Dolphin
##          3          3          4          4
```

```
grafico<-data.frame(res$x[,1],as.factor(kme2$cluster))
names(grafico)<-c("PC1","cluster")
grafico
```

```
##                    PC1 cluster
## Horse      -2.15955785       2
## Orangutan  -1.87741415       2
## Monkey     -1.92107037       2
## Donkey     -2.06824874       2
## Hippo      -1.56283420       2
## Camel      -1.24776651       2
## Bison      -1.38164444       2
## Buffalo    -0.47185521       1
## Guinea Pig  0.25117884       1
## Cat         0.09087423       1
## Fox        -0.51370377       1
## Llama      -1.38058058       2
## Mule       -1.80370758       2
## Pig        -0.20253900       1
## Zebra      -1.32306955       2
## Sheep      -0.57776326       1
## Dog         0.74672828       3
## Elephant   -0.08997063       1
## Rabbit      1.81036875       3
## Rat         0.95263023       3
## Deer        1.90757130       3
## Reindeer    2.04694152       3
## Whale       2.39169651       3
## Seal        4.39921084       4
## Dolphin     3.98452536       4
```

```
ggplot(grafico,aes(x=PC1,y=0,col=cluster))+geom_point(show.legend = TRUE)+ggtitle("PC1 y cluster
s método k-medias")
```


PC1 y clusters método k-medias
0.050
0.025
0.000
-0.025
-0.050
y
-2
0
2
4
PC1
cluster
1
2
3
4