Black Friday

05/01/2019

Contents

AUTORES:

- Reinier Mujica
- Jorge Lazo Rosado
- Miquel Martorell

PREGUNTAS

Trabajo final de tecnología en R (formato markdown):

(Restar 1 Millón y los datos estarán en el rango de 0 y 10 K.)

1.- ¿Qué variables pueden ser factores?

2.- ¿Se venden igual las mismas categorías?

3.- ¿Los usuarios compran el mismo item?

4.- Sistemas de recomendación

5.- ¿Gastan más los hombres o las mujeres?

6.- Gráfico bipartito entre productos y consumidores

7.- Otros: Clustering, proyección de ventas, estimar el gasto futuro, deducir con una serie de datos si el consumidor es hombre o mujer.

8.- Dividir los datos en conjunto de entrenamiento y prueba.

Esta es la lluvia de ideas que se hizo en clase... el profe valora mucho la innovación y que hagas cosas originales que nos guste a nosotros.

Introducción.

Según el autor de esta recopilación de datos (Mehdi Dagdoug), tenemos un conjunto de datos de 550.000 observaciones sobre el Black Friday en una tienda minorista, el cual contiene diferentes tipos de variables numéricas o categóricas.

La tienda quiere conocer mejor el comportamiento de compra de los clientes a la hora de decantarse por un producto u otro. Procederemos a analizar las diferentes variables de este conjunto de datos y, finalmente, predecir mediante una regresión lineal la variable dependiente (el monto de la compra) con la ayuda de la información contenida en las variables explicativas.

Además, este conjunto de datos también es particularmente conveniente para la agrupación en clústeres, tal vez para encontrar diferentes grupos de consumidores dentro de él.

Para empezar este análisis procedemos a cargar el conjunto de datos BlackFriday.csv:

```
rm(list=ls())
cat("\014") ## limpia la pantalla del R
```

```
#getwd()
```

```
datos = read.csv("/Users/jlazoros/BlackFriday.csv")
```

Librerias Utilizadas.

El paquete "tidyverse" es el que usaremos para visualizar y explorar nuestro conjunto de datos. Es conocido por su sintaxis fácil de leer y su gran cantidad de funciones útiles. El paquete "scales" se utilizará principalmente para personalizar el eje del gráfico. *Por último, el paquete "arules"" se utilizará en la parte final del análisis, Association Rule Learning y Apriori.*

Análisis Descriptivo

Análisis Preliminar

Comencemos con una rápida visión general de todo el conjunto de datos:

```
summary(datos)
```

```
##     User_ID            Product_ID     Gender        Age
##  Min.   :1000001   P00265242:  1858   F:132197   0-17 : 14707
##  1st Qu.:1001495   P00110742:  1591   M:405380   18-25: 97634
##  Median :1003031   P00025442:  1586              26-35:214690
##  Mean   :1002992   P00112142:  1539              36-45:107499
##  3rd Qu.:1004417   P00057642:  1430              46-50: 44526
##  Max.   :1006040   P00184942:  1424              51-55: 37618
##                    (Other)  :528149              55+  : 20903
##    Occupation     City_Category Stay_In_Current_City_Years
##  Min.   : 0.000   A:144638      0 : 72725
```

```
##  1st Qu.: 2.000   B:226493      1 :189192
##  Median : 7.000   C:166446      2 : 99459
##  Mean   : 8.083                 3 : 93312
##  3rd Qu.:14.000                 4+: 82889
##  Max.   :20.000
##
##  Marital_Status   Product_Category_1 Product_Category_2 Product_Category_3
##  Min.   :0.0000   Min.   : 1.000     Min.   : 2.00      Min.   : 3.0
##  1st Qu.:0.0000   1st Qu.: 1.000     1st Qu.: 5.00      1st Qu.: 9.0
##  Median :0.0000   Median : 5.000     Median : 9.00      Median :14.0
##  Mean   :0.4088   Mean   : 5.296     Mean   : 9.84      Mean   :12.7
##  3rd Qu.:1.0000   3rd Qu.: 8.000     3rd Qu.:15.00      3rd Qu.:16.0
##  Max.   :1.0000   Max.   :18.000     Max.   :18.00      Max.   :18.0
##                                      NA's   :166986     NA's   :373299
##     Purchase
##  Min.   :  185
##  1st Qu.: 5866
##  Median : 8062
##  Mean   : 9334
##  3rd Qu.:12073
##  Max.   :23961
##
```

```
glimpse(datos)
```

```
## Observations: 537,577
## Variables: 12
## $ User_ID                   <int> 1000001, 1000001, 1000001, 1000001,...
## $ Product_ID                <fct> P00069042, P00248942, P00087842, P0...
## $ Gender                    <fct> F, F, F, F, M, M, M, M, M, M, M, M,...
## $ Age                       <fct> 0-17, 0-17, 0-17, 0-17, 55+, 26-35,...
## $ Occupation                <int> 10, 10, 10, 10, 16, 15, 7, 7, 7, 20...
## $ City_Category             <fct> A, A, A, A, C, A, B, B, B, A, A, A,...
## $ Stay_In_Current_City_Years <fct> 2, 2, 2, 2, 4+, 3, 2, 2, 2, 1, 1, 1...
## $ Marital_Status            <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,...
## $ Product_Category_1        <int> 3, 1, 12, 12, 8, 1, 1, 1, 1, 8, 5, ...
## $ Product_Category_2        <int> NA, 6, NA, 14, NA, 2, 8, 15, 16, NA...
## $ Product_Category_3        <int> NA, 14, NA, NA, NA, NA, 17, NA, NA,...
## $ Purchase                  <int> 8370, 15200, 1422, 1057, 7969, 1522...
```

Vamos a restar 1.000.000 a la columna User_ID para que estos datos estén en el rango [0-10.000].

```
datos[["User_ID"]] = datos[["User_ID"]]-1000000
glimpse(datos)
```

```
## Observations: 537,577
## Variables: 12
## $ User_ID                   <dbl> 1, 1, 1, 1, 2, 3, 4, 4, 4, 5, 5, 5,...
## $ Product_ID                <fct> P00069042, P00248942, P00087842, P0...
## $ Gender                    <fct> F, F, F, F, M, M, M, M, M, M, M, M,...
## $ Age                       <fct> 0-17, 0-17, 0-17, 0-17, 55+, 26-35,...
## $ Occupation                <int> 10, 10, 10, 10, 16, 15, 7, 7, 7, 20...
## $ City_Category             <fct> A, A, A, A, C, A, B, B, B, A, A, A,...
## $ Stay_In_Current_City_Years <fct> 2, 2, 2, 2, 4+, 3, 2, 2, 2, 1, 1, 1...
## $ Marital_Status            <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,...
## $ Product_Category_1        <int> 3, 1, 12, 12, 8, 1, 1, 1, 1, 8, 5, ...
```

```
## $ Product_Category_2         <int> NA, 6, NA, 14, NA, 2, 8, 15, 16, NA...
## $ Product_Category_3         <int> NA, 14, NA, NA, NA, NA, 17, NA, NA,...
## $ Purchase                   <int> 8370, 15200, 1422, 1057, 7969, 1522...
```

Tenemos 12 columnas diferentes, cada una de las cuales representa una variable, las cuales procedemos a describir a continuación:

- User_ID: Identificador único del comprador.
- Product_ID: Identificador único del producto.
- Gender: Sexo del comprador.
- Age: Edad del comprador.
- Occupation: Ocupación del comprador.
- City_Category: Lugar de residencia del comprador.
- Stay_In_Current_City_Years: Número de años de permanencia en la ciudad actual.
- Marital_Status: Estado civil del comprador.
- Product_Category_1: Categoría 1 de producto de la compra.
- Product_Category_2: Categoría 2 de producto de la compra.
- Product_Category_3: Categoría 3 de producto de la compra.
- Purchase: Importe de la compra en dólares.

Si observamos las primeras filas de nuestro conjunto de datos, podemos ver que cada fila representa una transacción diferente, o un artículo comprado por un cliente en concreto. Más adelante, cuando agrupemos todas las transacciones por un usuario en concreto, obtendremos una suma de todas las compras realizadas por un solo cliente.

Debemos recalcar que en este conjunto de datos no hay una clave dada con respecto a los diferentes Product_IDs y al artículo/producto que representan. Por ejemplo, no podemos atribuir P00265242 a un producto reconocible. En realidad, deberíamos tener otro conjunto de datos el cual nos proporcionase el nombre de un prodcuto y su Product_ID para así unirlo todo en nuestro conjunto de datos existente. *Esto no afectará necesariamente a nuestro análisis, pero sería más útil durante nuestra implementación del algoritmo Apriori y podría hacer que algunas partes de la EDA sean más claras de interpretar.*

Análisis Variable Género Clientes.

Para empezar nuestro verdadero análisis, vamos a examinar la variable género de los compradores en esta tienda. Dado que cada fila representa una transacción individual, primero debemos agrupar los datos por User_ID para eliminar los duplicados:

```
datos_gender = datos %>%
               select(User_ID, Gender) %>%
               group_by(User_ID) %>%
               distinct()
head(datos_gender)
```

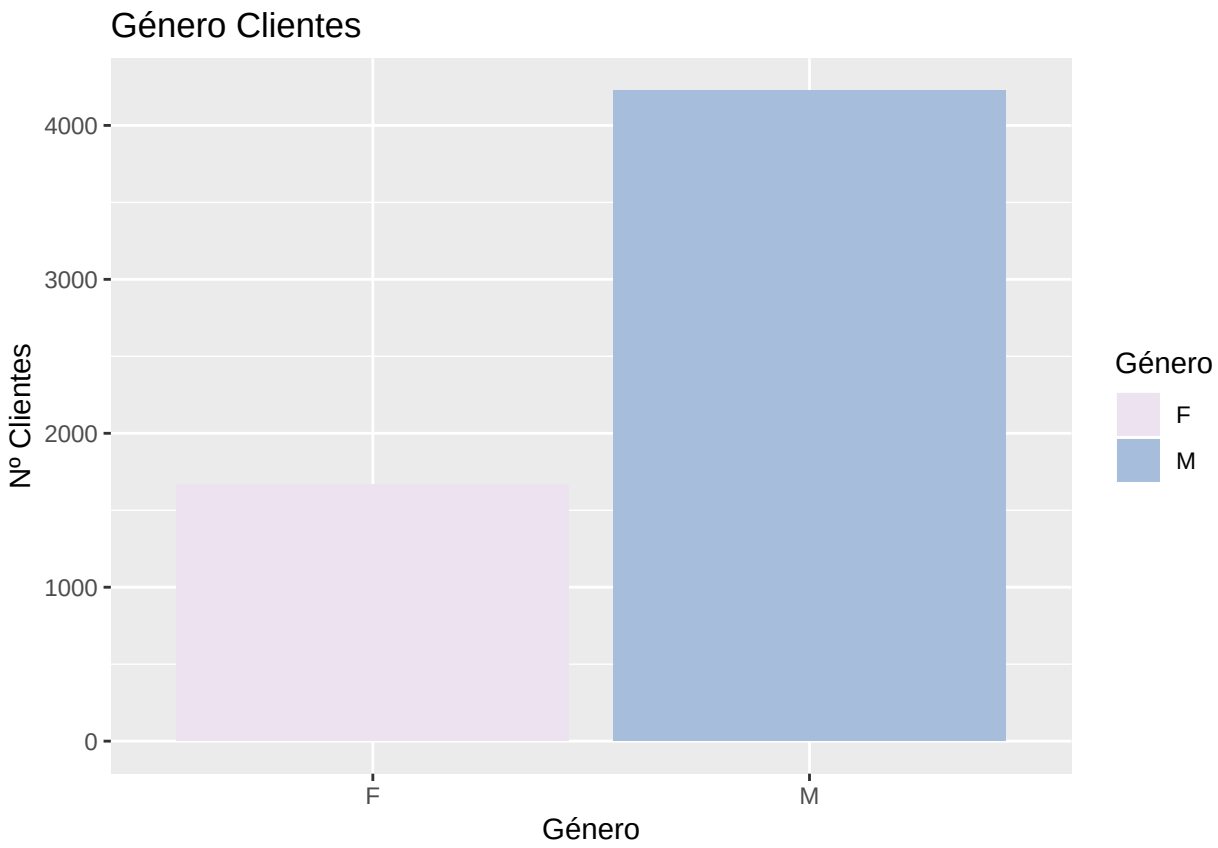
```
## # A tibble: 6 x 2
## # Groups:   User_ID [6]
##   User_ID Gender
##     <dbl> <fct>
## 1       1 F
## 2       2 M
## 3       3 M
## 4       4 M
## 5       5 M
## 6       6 F
```

```r
summary(datos_gender$Gender)
```

```
##    F    M
## 1666 4225
```

Ahora tenemos el conjunto de datos correctamente filtrado para ver el género de cada User_ID y sus totales como referencia. Procedemos a realizar un gráfico el cual representa la distribución de género a través de nuestro conjunto de datos:

```r
options(scipen=10000)   # Eliminamos la numeración científica.
gender_dist  = ggplot(data = datos_gender) +
               geom_bar(mapping = aes(x = Gender, y = ..count.., fill = Gender)) +
               labs(title = 'Género Clientes', x = 'Género', y = 'Nº Clientes', fill = 'Género') +
               scale_fill_brewer(palette = 'PuBuGn')
print(gender_dist)
```



Como podemos ver, hay bastantes más hombres que mujeres comprando en esta tienda durante el Black Friday. Esta sencilla división de género podría ser útil para los minoristas porque algunos podrían querer modificar el diseño de su tienda, la selección de productos y otras variables de manera diferente dependiendo del porcentaje de género de sus compradores.

Para realizar un análisis más profundo, calculemos el importe promedio de gasto en relación al género. Para facilitar la interpretación y el seguimiento, crearemos tablas separadas y luego las uniremos:

```r
total_purchase_user = datos %>%
                      select(User_ID, Gender, Purchase) %>%
                      group_by(User_ID) %>%
                      arrange(User_ID) %>%
                      dplyr::summarize(Total_Purchase = sum(Purchase))
```

```r
user_gender = datos %>%
              select(User_ID, Gender) %>%
              group_by(User_ID) %>%
              arrange(User_ID) %>%
              distinct()
head(user_gender)
```

```
## # A tibble: 6 x 2
## # Groups:   User_ID [6]
##   User_ID Gender
##     <dbl> <fct>
## 1       1 F
## 2       2 M
## 3       3 M
## 4       4 M
## 5       5 M
## 6       6 F
```

```r
head(total_purchase_user)
```

```
## # A tibble: 6 x 2
##   User_ID Total_Purchase
##     <dbl>          <int>
## 1       1         333481
## 2       2         810353
## 3       3         341635
## 4       4         205987
## 5       5         821001
## 6       6         379450
```

```r
user_purchase_gender = dplyr::full_join(total_purchase_user, user_gender, by = "User_ID")
head(user_purchase_gender)
```

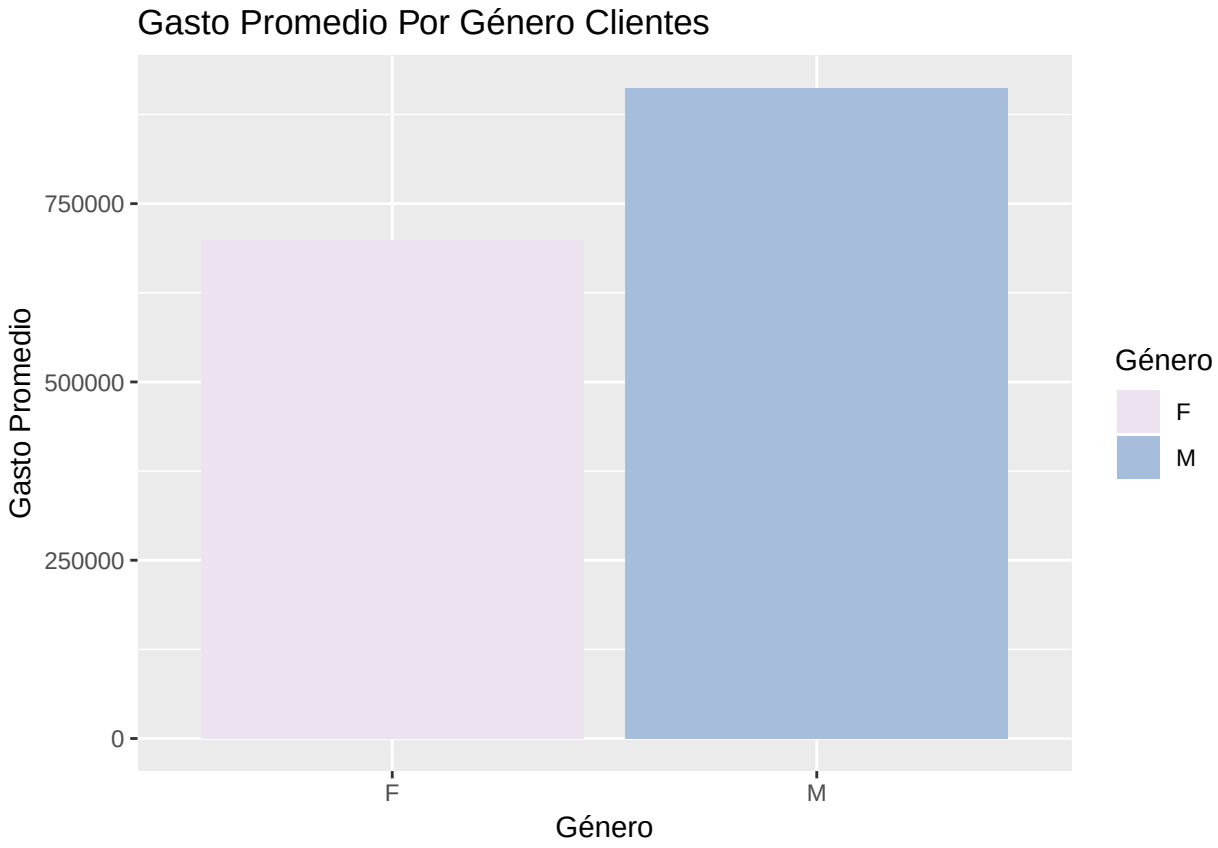
```
## # A tibble: 6 x 3
##   User_ID Total_Purchase Gender
##     <dbl>          <int> <fct>
## 1       1         333481 F
## 2       2         810353 M
## 3       3         341635 M
## 4       4         205987 M
## 5       5         821001 M
## 6       6         379450 F
```

```r
average_spending_gender = user_purchase_gender %>%
                          group_by(Gender) %>%
                          dplyr::summarize(Purchase = sum(as.numeric(Total_Purchase)),
                                    Count = n(),
                                    Average = Purchase/Count)
head(average_spending_gender)
```

```
## # A tibble: 2 x 4
##   Gender   Purchase Count Average
##   <fct>       <dbl> <int>   <dbl>
## 1 F      1164624021  1666 699054.
## 2 M      3853044357  4225 911963.
```

Podemos ver que el promedio de transacciones para las mujeres fue de 699.054,00 y el promedio de transacciones para los hombres fue de 911.963,20. Visualicemos nuestros resultados.

```
gender_average  = ggplot(data = average_spending_gender) +
                         geom_bar(mapping = aes(x = Gender, y = Average, fill = Gender), stat = 'identi
                         labs(title = 'Gasto Promedio Por Género Clientes', x = 'Género', y = 'Gasto Pr
                         scale_fill_brewer(palette = 'PuBuGn')
print(gender_average)
```



Aquí podemos ver una observación interesante. Aunque las mujeres hacen menos compras que los hombres, parecen estar comprando casi tanto en promedio como los hombres. Dicho esto, hay que tener en cuenta la escala, ya que las mujeres siguen gastando en promedio unos 250.000 dólares menos que los hombres.

Análisis Variable Top Sellers (Productos Más Vendidos).

Ahora vamos a examinar los productos más vendidos en esta tienda durante el Black Friday. En esta situación, no agruparemos por ID de producto ya que queremos ver duplicados, por si acaso la gente está comprando 2 o más cantidades del mismo producto.

```
top_sellers = datos %>%
              count(Product_ID, sort = TRUE)
top_5 = head(top_sellers, 5)
top_5
```

```
## # A tibble: 5 x 2
##   Product_ID     n
##   <fct>      <int>
## 1 P00265242   1858
```

```
## 2 P00110742   1591
## 3 P00025442   1586
## 4 P00112142   1539
## 5 P00057642   1430
```

Estos son las 5 referencias de productos que más se venden en orden descendente. Por lo que el primero es el lider. Los 5 productos más vendidos son (por ID del producto):

- P00265242 = 1858
- P00110742 = 1591
- P00025442 = 1586
- P00112142 = 1539
- P00057642 = 1430

Ahora que hemos identificado nuestros 5 productos más vendidos, vamos a examinar detenidamente el producto más vendido, el producto P00265242.

```
best_seller = datos[datos$Product_ID == 'P00265242', ]
head(best_seller)
```

```
##      User_ID Product_ID Gender   Age Occupation City_Category
## 400       66  P00265242      M 26-35         18             C
## 1192     196  P00265242      F 36-45          9             C
## 1373     222  P00265242      M 26-35          1             A
## 1846     301  P00265242      M 18-25          4             B
## 2210     345  P00265242      M 26-35         12             A
## 2405     383  P00265242      F 26-35          7             A
##      Stay_In_Current_City_Years Marital_Status Product_Category_1
## 400                           2              0                  5
## 1192                         4+              0                  5
## 1373                          1              0                  5
## 1846                         4+              0                  5
## 2210                          2              1                  5
## 2405                         4+              1                  5
##      Product_Category_2 Product_Category_3 Purchase
## 400                   8                 NA     8652
## 1192                  8                 NA     8767
## 1373                  8                 NA     6944
## 1846                  8                 NA     8628
## 2210                  8                 NA     8593
## 2405                  8                 NA     6998
```

Podemos ver que este producto encaja en Product_Category_1 = 5 y Product_Category_2 = 8. Como se mencionó en la introducción, sería útil tener una clave para hacer referencia al nombre del artículo con el fin de determinar exactamente qué producto es.

Sería deseable contar con una clave para identificar el nombre de la referencia y determinar de qué producto se trata

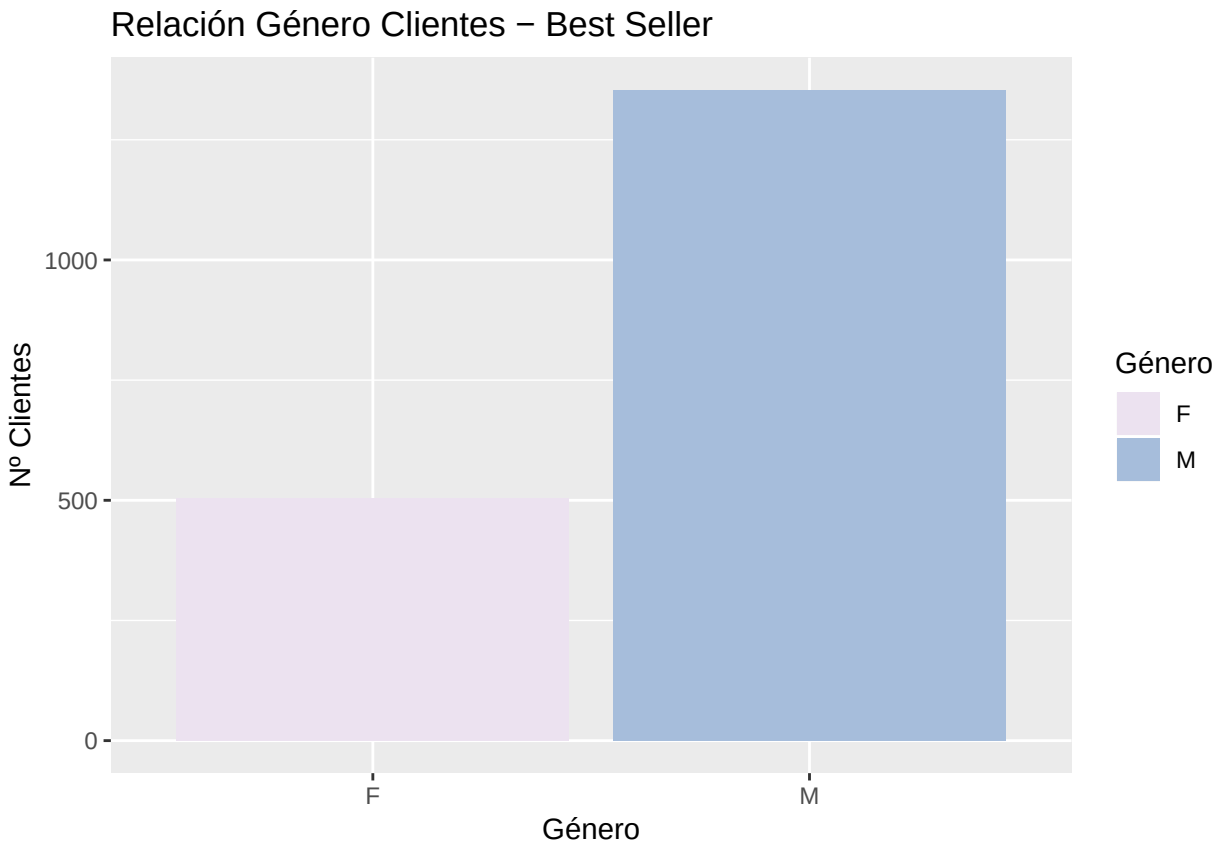
Otro observación interesante es que, aunque la gente está comprando el mismo producto, están pagando precios diferentes. Esto puede deberse a varias promociones, descuentos o códigos de cupones del Black Friday. De lo contrario, habría que investigar la razón de que los precios de compra del mismo producto difieran entre los clientes.

Otra característica que podemos observar es que a pesar de tratarse de un mismo producto, la gente paga precios diferentes. Esto se puede deber a una diferenciación del precio mediante descuentos, promociones, cupones durante le Black Friday.

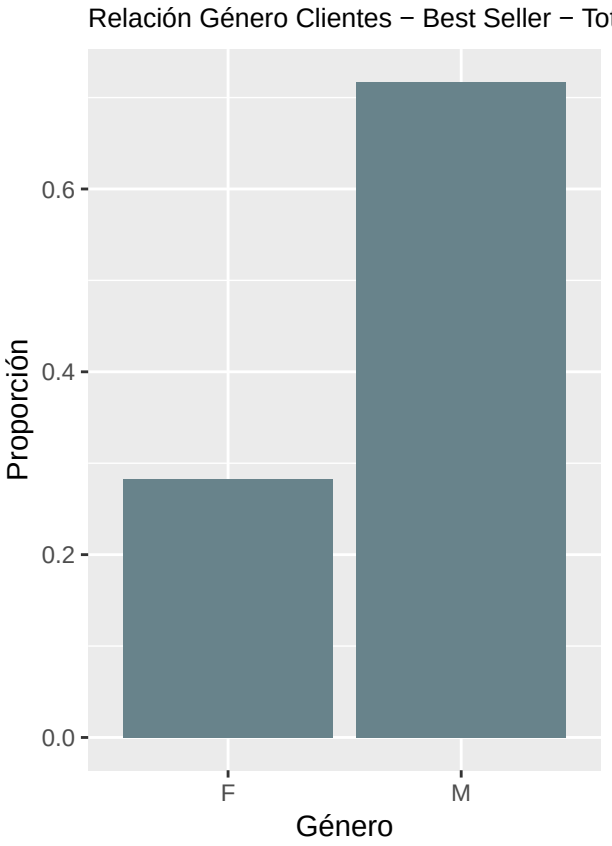
En todo caso, habría que profundizar sobre las razones de esta diferenciación de precio

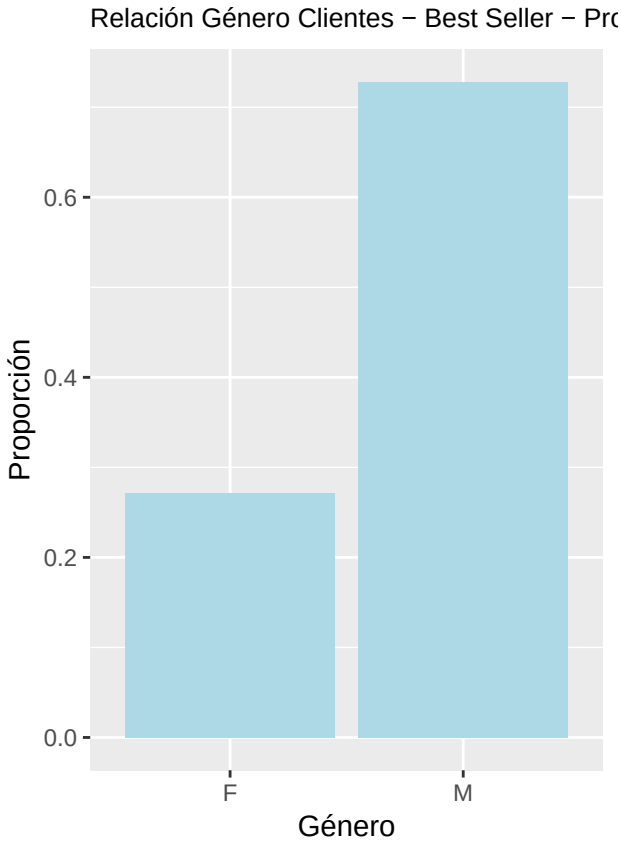
Continuemos analizando nuestro best seller para ver si existe alguna relación con el género.

```
gender_dist_bs  = ggplot(data = best_seller) +
                  geom_bar(mapping = aes(x = Gender, y = ..count.., fill = Gender)) +
                  labs(title = 'Relación Género Clientes - Best Seller', x = 'Género', y = 'Nº Clientes'
                  scale_fill_brewer(palette = 'PuBuGn')
print(gender_dist_bs)
```




```
gender_dist_bs_prop = ggplot(data = best_seller) +
                          geom_bar(fill = 'lightblue', mapping = aes(x = Gender, y = ..prop.., group =
                          labs(title = 'Relación Género Clientes - Best Seller - Proporción', x = 'Gé
                          theme(plot.title = element_text(size=9.5))
gender_dist_prop = ggplot(data = datos_gender) +
                       geom_bar(fill = "lightblue4", mapping = aes(x = Gender, y = ..prop.., group = 
                       labs(title = 'Relación Género Clientes - Best Seller - Total', x = 'Género', y
                       theme(plot.title = element_text(size=9.5))
grid.arrange(gender_dist_prop, gender_dist_bs_prop, ncol=2)
```





Podemos ver que entre el conjunto de observaciones generales, tanto los compradores de los productos más vendidos como los compradores de todos los productos son aproximadamente ~25% mujeres y ~75% hombres. Existe una ligera diferencia, pero parece que en general podemos concluir que nuestro best seller no atiende a un género específico.

Ahora, sigamos adelante y examinemos la variable Edad.

Análisis Variable Edad.

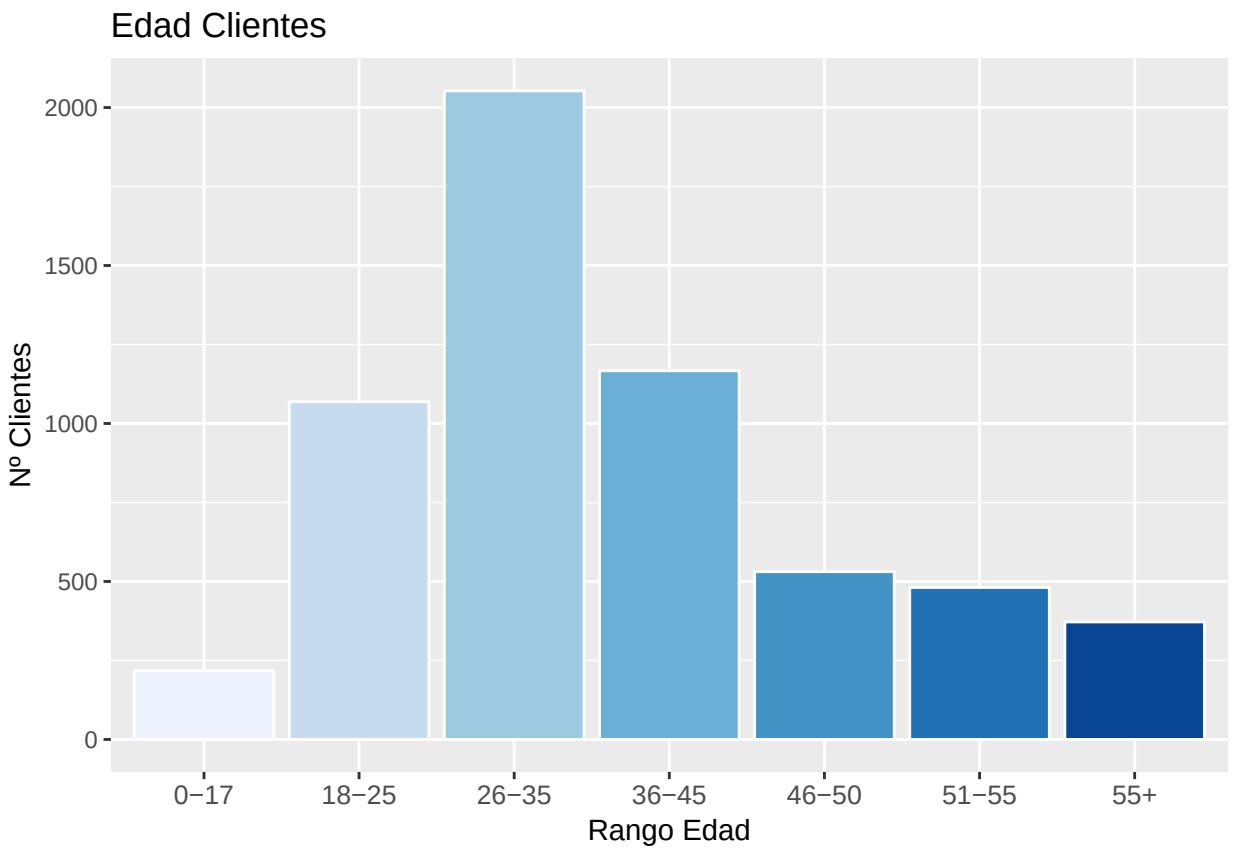
Comencemos a examinar la edad creando una tabla de cada grupo de edad individual y sus respectivos recuentos.

```
customers_age = datos %>%
                select(User_ID, Age) %>%
                distinct() %>%
                count(Age)
customers_age
```

```
## # A tibble: 7 x 2
##   Age       n
##   <fct> <int>
## 1 0-17    218
## 2 18-25  1069
## 3 26-35  2053
## 4 36-45  1167
## 5 46-50   531
## 6 51-55   481
## 7 55+     372
```

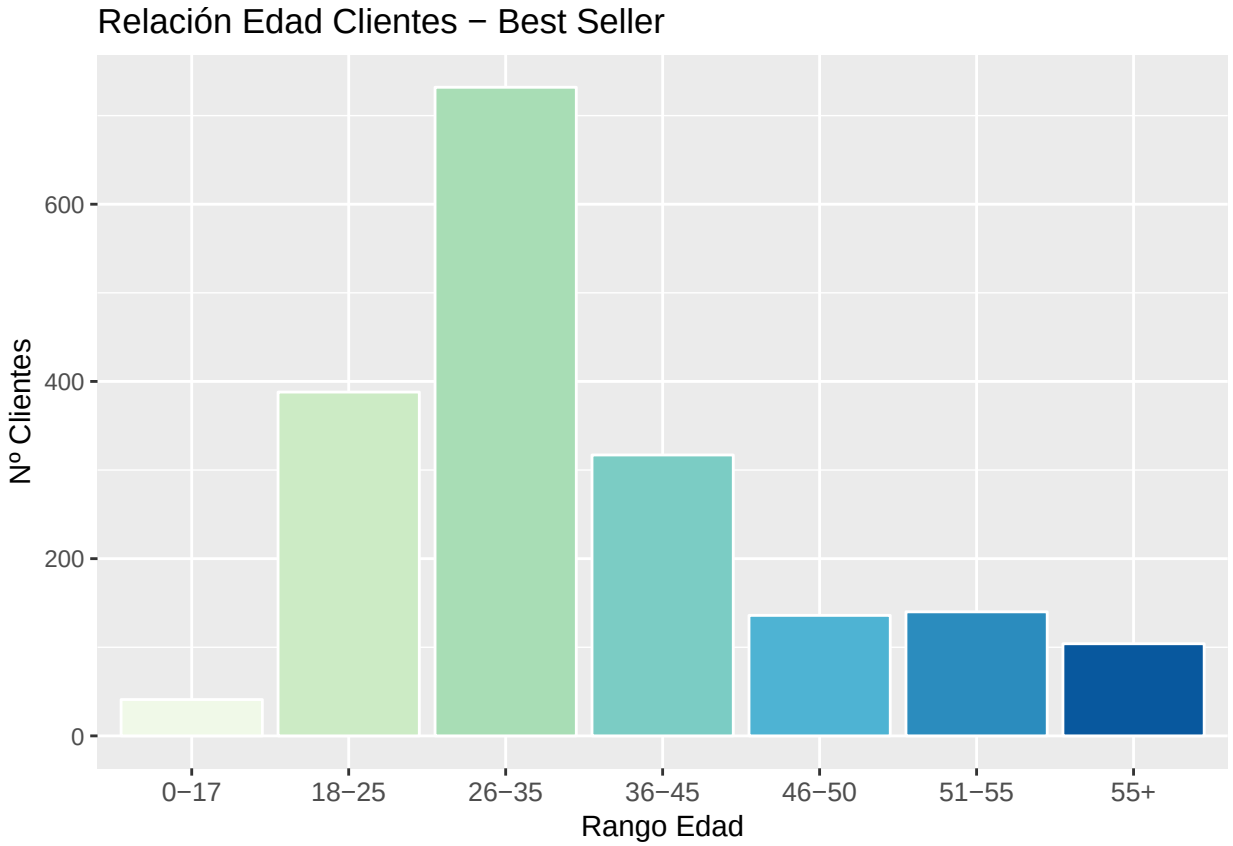
Aquí podemos ver un dato que muestra el recuento de cada categoría de edad de los clientes. Vamos a visualizar esta tabla.

```
customers_age_vis = ggplot(data = customers_age) +
                           geom_bar(color = 'white', stat = 'identity', mapping = aes(x = Age, y = n, fi
                           labs(title = 'Edad Clientes', x = 'Rango Edad', y = 'Nº Clientes') +
                           theme(axis.text.x = element_text(size = 10)) +
                           scale_fill_brewer(palette = 'Blues') +
                           theme(legend.position="none")
print(customers_age_vis)
```



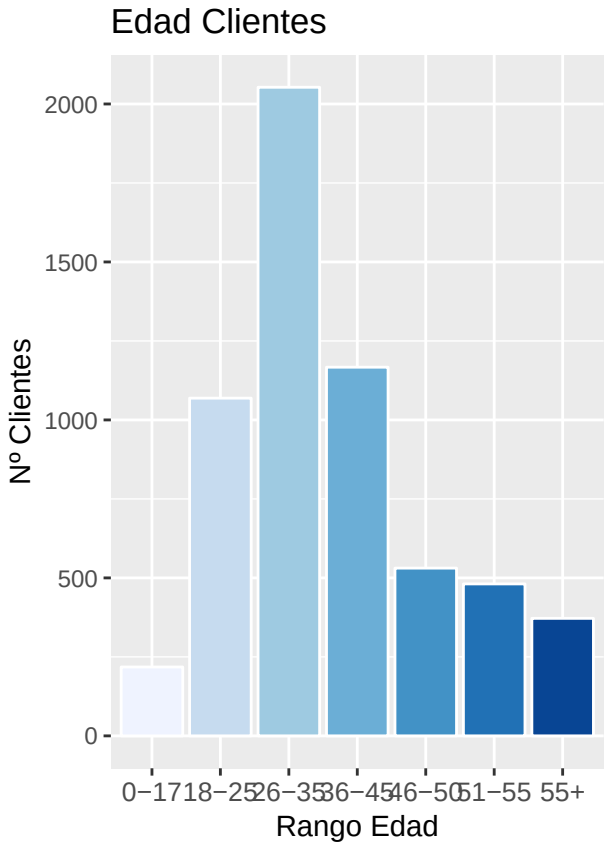
También podemos trazar un gráfico similar que represente la distribución de la edad dentro de nuestra categoría de "best seller". Esto nos mostrará si hay una categoría de edad específica que compró el producto más vendido.

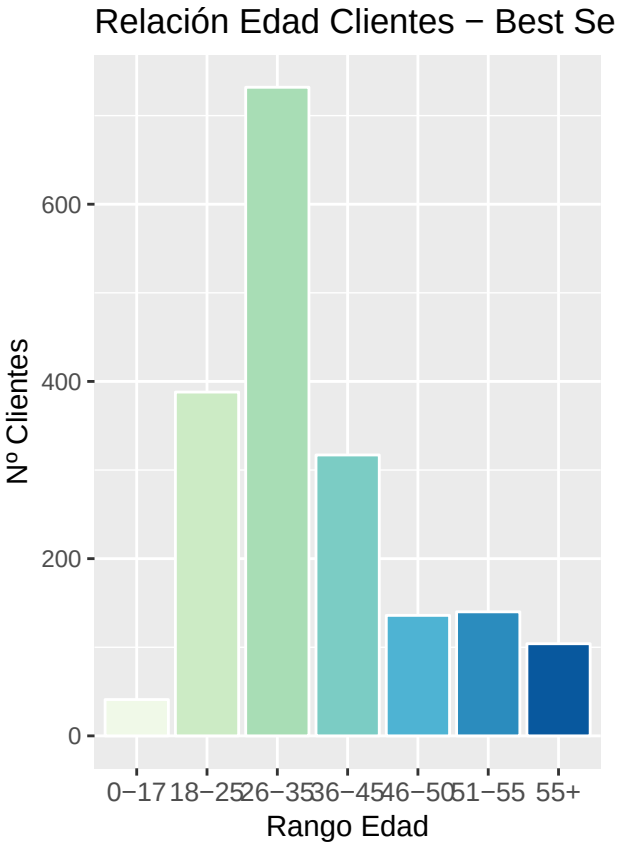
```
ageDist_bs  = ggplot(data = best_seller) +
                     geom_bar(color = 'white', mapping = aes(x = Age, y = ..count.., fill = Age)) +
                     labs(title = 'Relación Edad Clientes - Best Seller', x = 'Rango Edad', y = 'Nº Cli
                     theme(axis.text.x = element_text(size = 10)) +
                     scale_fill_brewer(palette = 'GnBu') +
                     theme(legend.position="none")
print(ageDist_bs)
```



Parece que los jóvenes (18-25 años y 26-35 años) son los que más compran el producto más vendido. Comparemos esta observación con los datos generales.

```
grid.arrange(customers_age_vis, ageDist_bs, ncol=2)
```





Podemos ver que hay alguna desviación con la proporción de clientes agrupados por edad al comparar el producto más vendido con los datos globales. Parece que los clientes mayores de 45 años compran el top seller un poco menos que otros productos incluidos en los datos generales.

Ahora que hemos examinado la edad, vamos a pasar a otra variable, la variable ciudad.

Análisis Variable Ciudad.

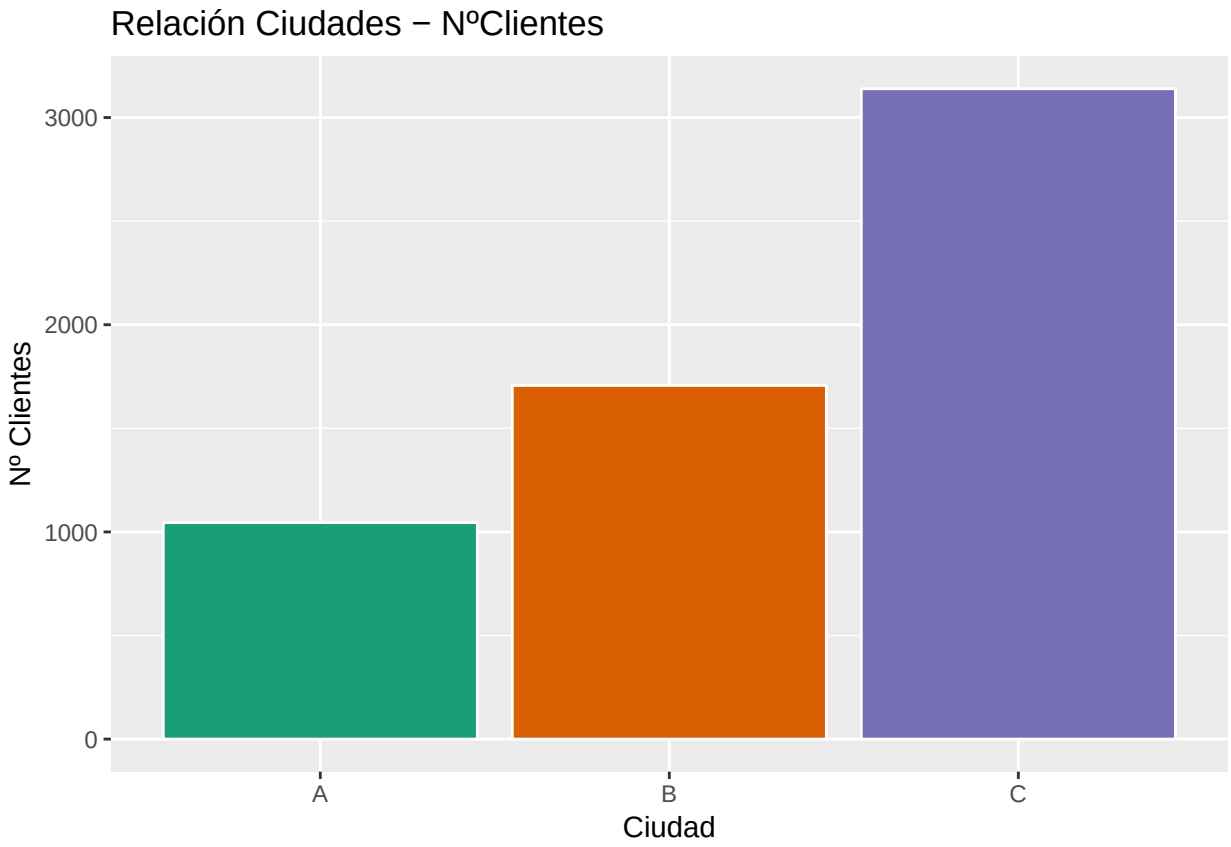
Vamos a crear una tabla de cada User_ID y su correspondiente City_Category.

```
customers_location =  datos %>%
                      select(User_ID, City_Category) %>%
                      distinct()
head(customers_location)
```

```
##   User_ID City_Category
## 1       1             A
## 2       2             C
## 3       3             A
## 4       4             B
## 5       5             A
## 6       6             A
```

```
customers_location_vis = ggplot(data = customers_location) +
                                geom_bar(color = 'white', mapping = aes(x = City_Category, y = ..count.
                                labs(title = 'Relación Ciudades - NºClientes', x = 'Ciudad', y = 'Nº Cl
                                scale_fill_brewer(palette = "Dark2") +
```

```
                                theme(legend.position="none")
print(customers_location_vis)
```



Podemos ver que la mayoría de nuestros clientes viven en la Ciudad C. Ahora, podemos calcular la cantidad total de compra por Ciudad para ver qué clientes de la ciudad gastaron más.

```
purchases_city = datos %>%
                 group_by(City_Category) %>%
                 dplyr::summarize(Purchases = sum(Purchase))
purchases_city_1000s = purchases_city %>%
                       mutate(purchasesThousands = purchases_city$Purchases / 1000)
purchases_city_1000s
```

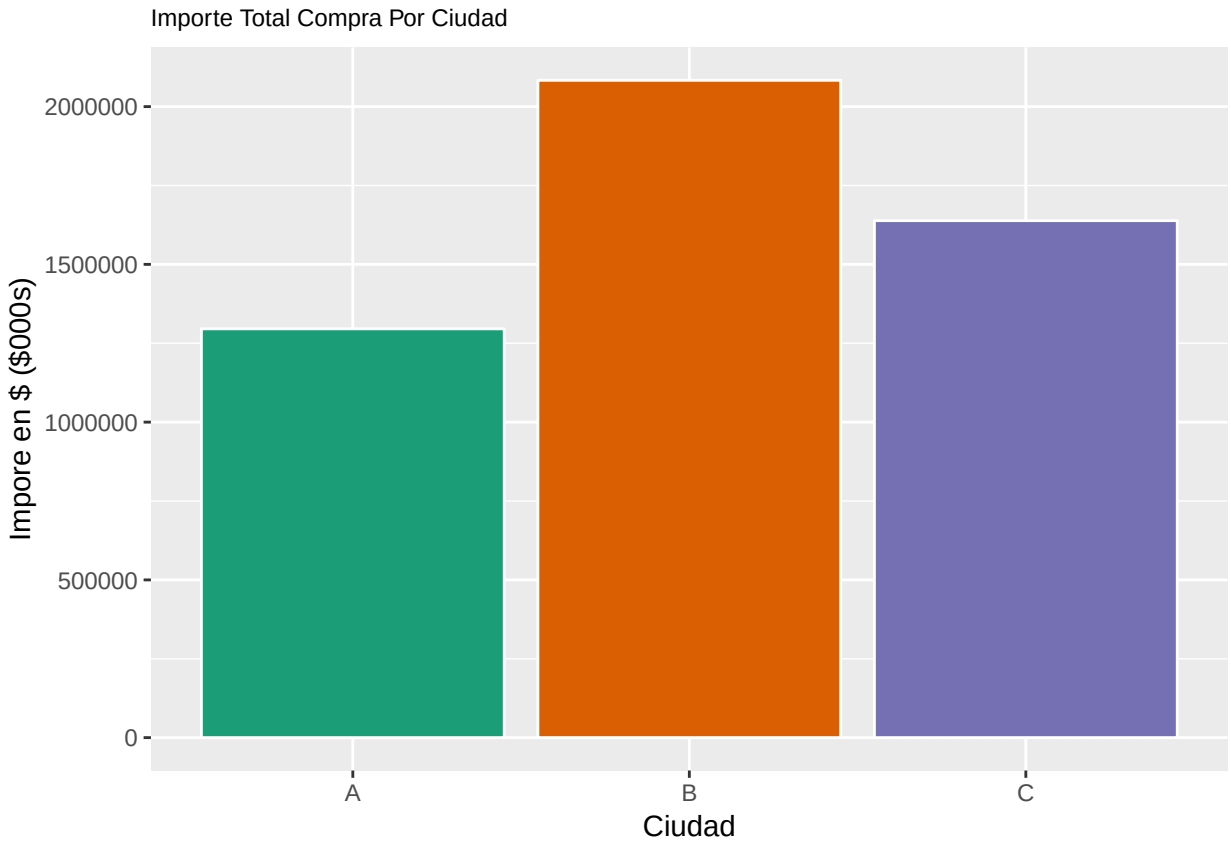
```
## # A tibble: 3 x 3
##   City_Category  Purchases purchasesThousands
##   <fct>              <int>              <dbl>
## 1 A             1295668797           1295669.
## 2 B             2083431612           2083432.
## 3 C             1638567969           1638568.
```

Para trabajar con números grandes, dividimos la columna Compras entre 1000. Esta es una práctica común dentro del mundo de los negocios y de la contabilidad, y hace que los grandes números sean más fáciles de leer y graficar.

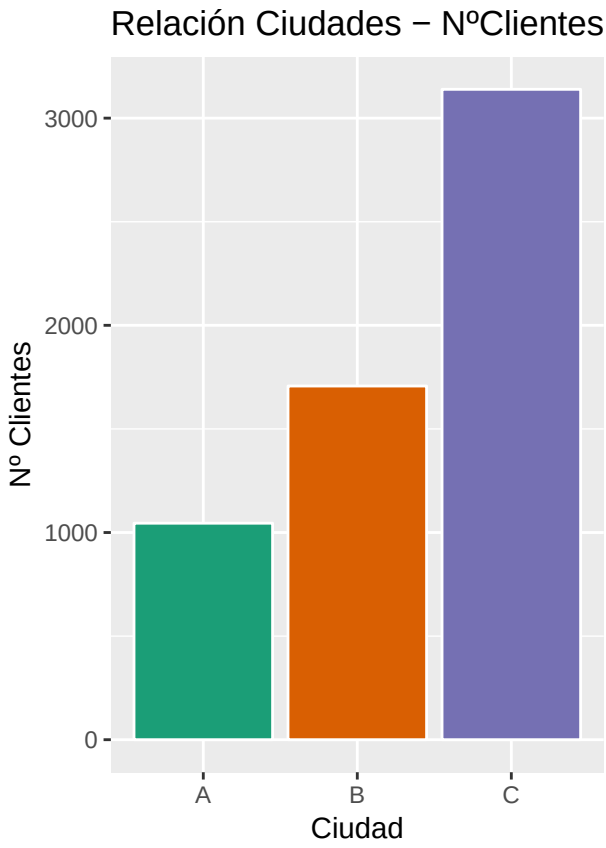
Ahora que tenemos nuestra tabla, visualicemos nuestros resultados.

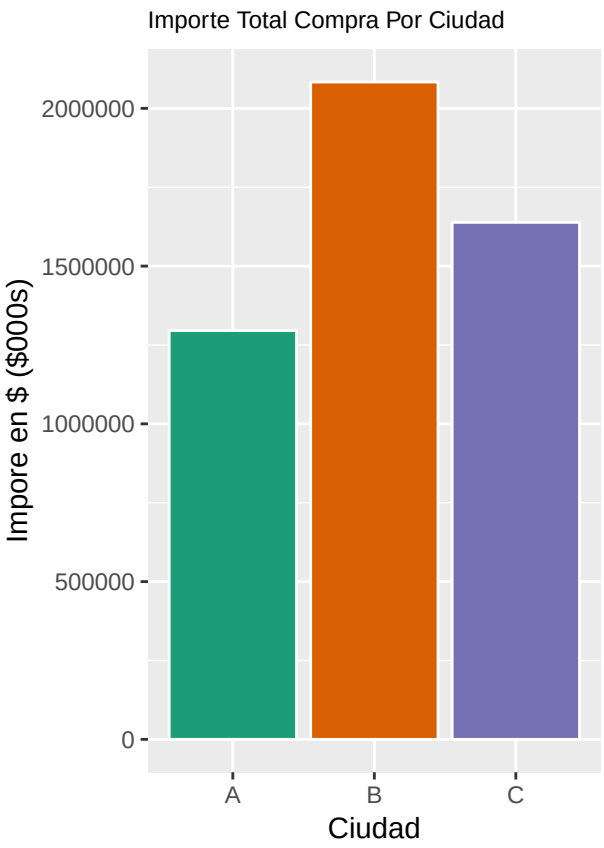
```
purchaseCity_vis = ggplot(data = purchases_city_1000s, aes(x = City_Category, y = purchasesThousands, f
                          geom_bar(color = 'white', stat = 'identity') +
                          labs(title = 'Importe Total Compra Por Ciudad', y = 'Impore en $ ($000s)', x =
```

```
                        scale_fill_brewer(palette = "Dark2") +
                        theme(legend.position="none", plot.title = element_text(size = 9))
print(purchaseCity_vis)
```




```
grid.arrange(customers_location_vis, purchaseCity_vis, ncol=2)
```





Aquí podemos ver que los clientes de la Ciudad C fueron los compradores más frecuentes durante el Black Friday, pero los clientes de la Ciudad B tuvieron la mayor cantidad de compras totales.

Continuemos investigando e intentemos determinar la razón de esta observación.

Averigüemos cuántas compras fueron hechas por los clientes de cada ciudad. En primer lugar, obtendremos el número total de compras para cada ID de usuario correspondiente.

```
customers = datos %>%
            group_by(User_ID) %>%
            count(User_ID)
head(customers)
```

```
## # A tibble: 6 x 2
## # Groups:   User_ID [6]
##   User_ID     n
##     <dbl> <int>
## 1       1    34
## 2       2    76
## 3       3    29
## 4       4    13
## 5       5   106
## 6       6    46
```

Esto nos dice cuántas veces un determinado usuario hizo una compra. Para profundizar un poco más, calcularemos el importe total de la compra de cada usuario y luego lo uniremos a la otra tabla.

```
customers_City =  datos %>%
                  select(User_ID, City_Category) %>%
                  group_by(User_ID) %>%
```

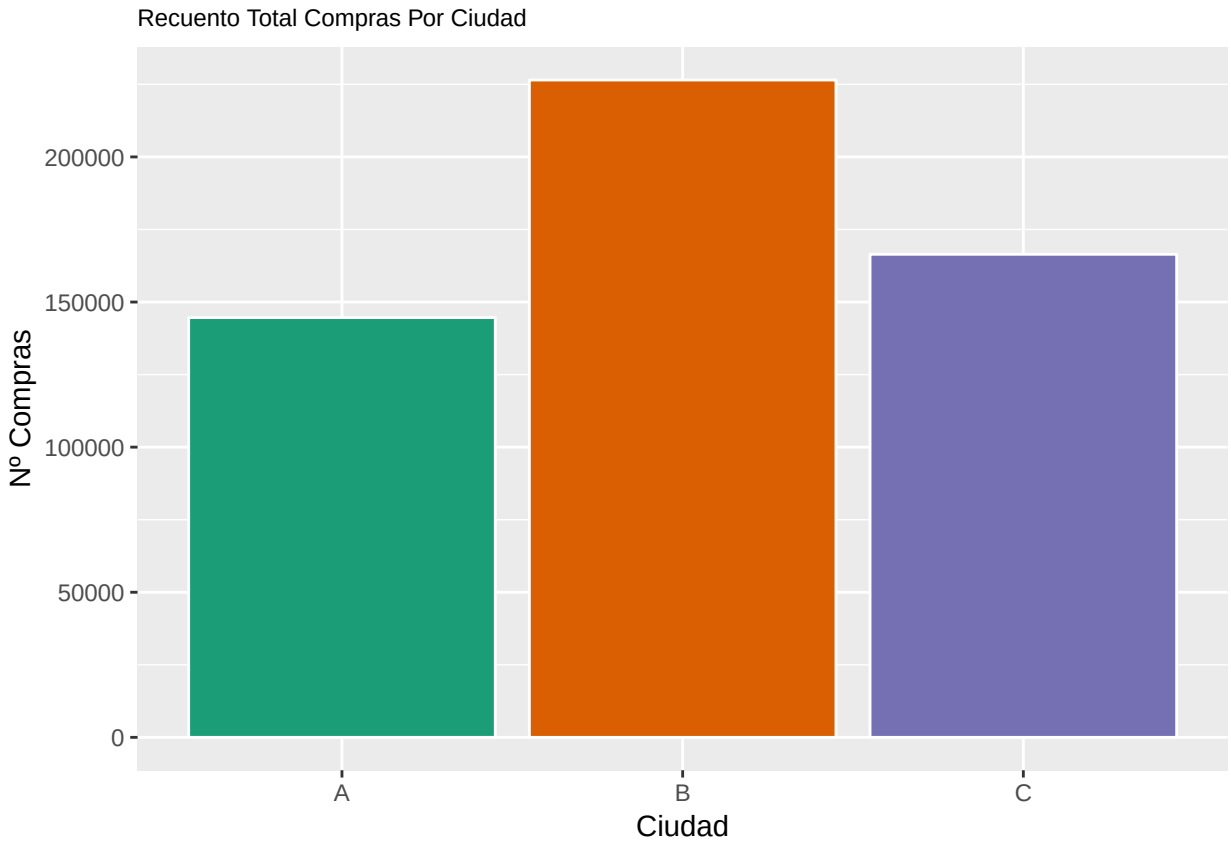
```
                  distinct() %>%
                  ungroup() %>%
                  left_join(customers, customers_City, by = 'User_ID')
head(customers_City)
```

```
## # A tibble: 6 x 3
##   User_ID City_Category     n
##     <dbl> <fct>         <int>
## 1       1 A                34
## 2       2 C                76
## 3       3 A                29
## 4       4 B                13
## 5       5 A               106
## 6       6 A                46
```

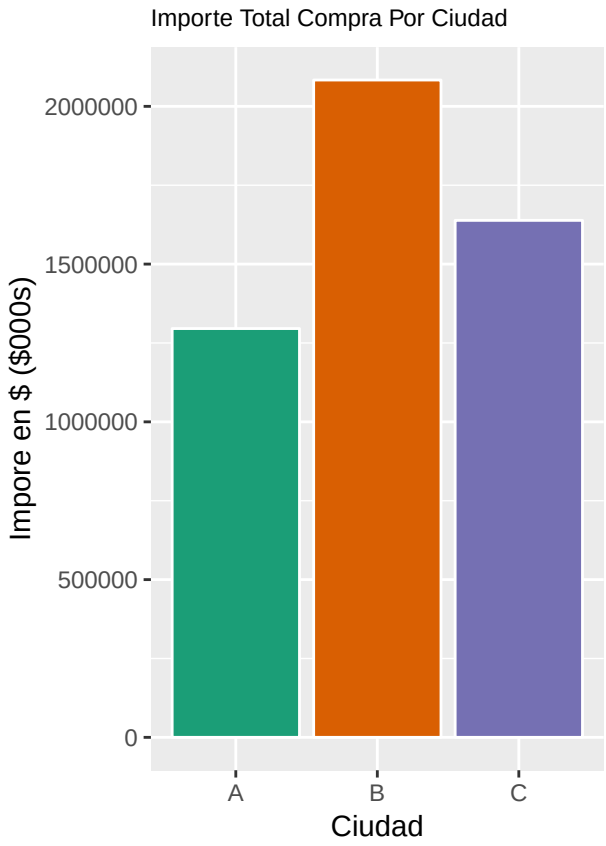
```
city_purchases_count = customers_City %>%
                       select(City_Category, n) %>%
                       group_by(City_Category) %>%
                       summarise(CountOfPurchases = sum(n))
city_purchases_count
```

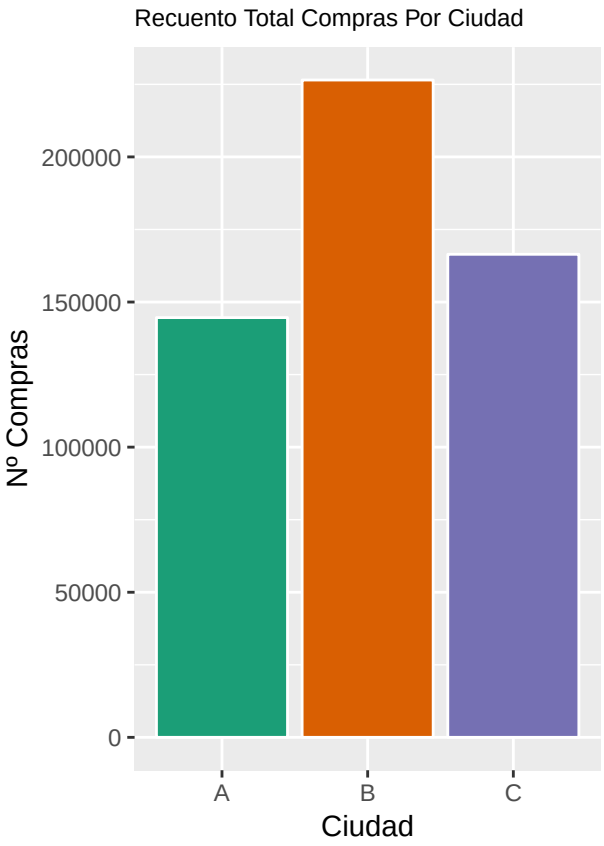
```
## # A tibble: 3 x 2
##   City_Category CountOfPurchases
##   <fct>                    <int>
## 1 A                       144638
## 2 B                       226493
## 3 C                       166446
```

```
city_count_purchases_vis = ggplot(data = city_purchases_count, aes(x = City_Category, y = CountOfPurchas
                           geom_bar(color = 'white', stat = 'identity') +
                           labs(title = 'Recuento Total Compras Por Ciudad', y = 'Nº Compras', x = 'Ciu
                           scale_fill_brewer(palette = "Dark2") +
                           theme(legend.position="none", plot.title = element_text(size = 9))
print(city_count_purchases_vis)
```




```
grid.arrange(purchaseCity_vis, city_count_purchases_vis, ncol = 2)
```





Una afirmación que podemos realizar de estos gráficos es que los clientes de la Ciudad B simplemente están haciendo más compras que los clientes de la Ciudad A + clientes de la Ciudad C, y no necesariamente comprando productos más caros.

Podemos hacer esta suposición debido al hecho de que el gráfico del "Total de Compras" tiene un aspecto muy similar al gráfico del "Total de Compras de Clientes". Si fuera el otro caso, lo más probable es que los clientes de la Ciudad B tuvieran un recuento más bajo de las compras totales, lo que corresponde a una mayor cantidad total de compras.

Ahora, ya que hemos identificado que las cuentas de compra a través de City_Category siguen una distribución similar a la cantidad total de compra, vamos a examinar la distribución de nuestro producto más vendido (P00265242) dentro de cada City_Category.

```
head(best_seller)
```

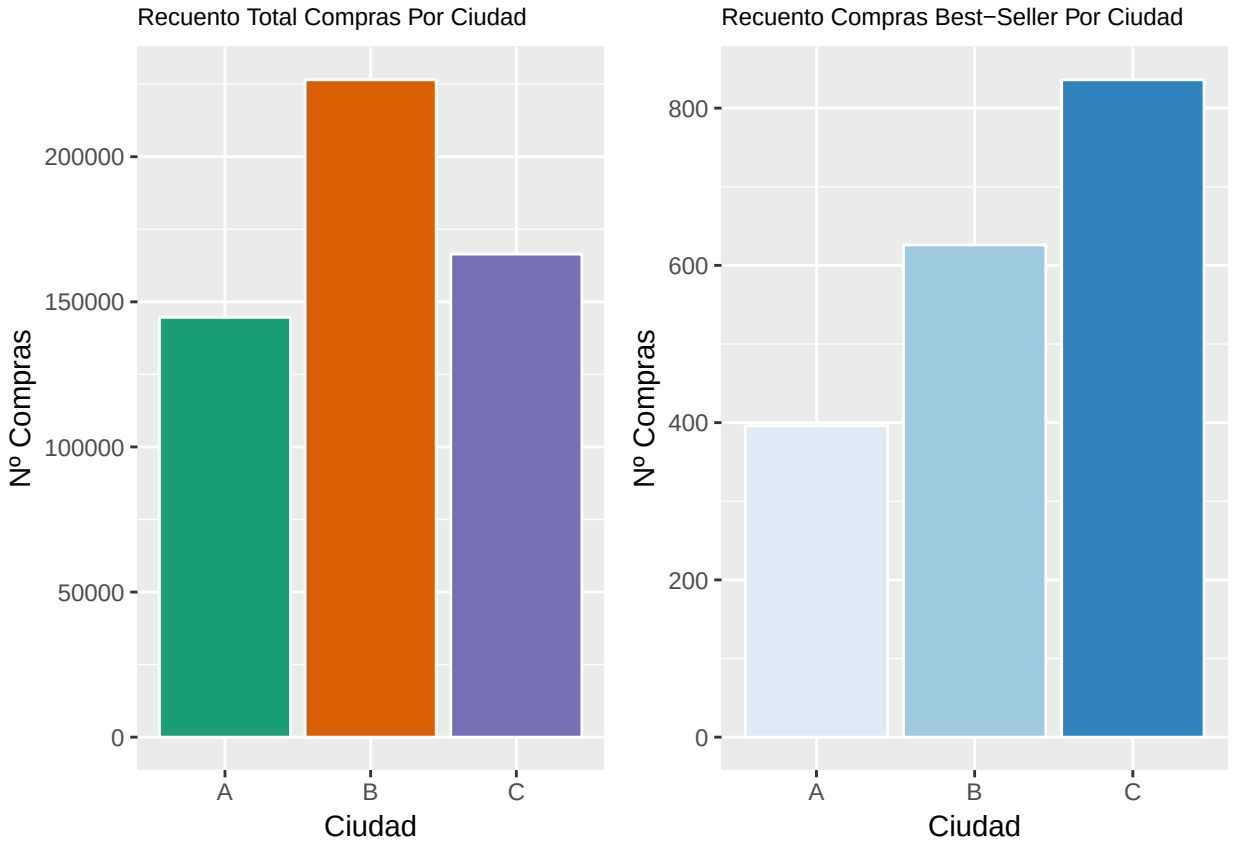
```
##      User_ID Product_ID Gender   Age Occupation City_Category
## 400       66  P00265242      M 26-35         18             C
## 1192     196  P00265242      F 36-45          9             C
## 1373     222  P00265242      M 26-35          1             A
## 1846     301  P00265242      M 18-25          4             B
## 2210     345  P00265242      M 26-35         12             A
## 2405     383  P00265242      F 26-35          7             A
##      Stay_In_Current_City_Years Marital_Status Product_Category_1
## 400                           2              0                 5
## 1192                         4+              0                 5
## 1373                          1              0                 5
## 1846                         4+              0                 5
## 2210                          2              1                 5
## 2405                         4+              1                 5
```

```
##      Product_Category_2 Product_Category_3 Purchase
## 400                   8                 NA     8652
## 1192                  8                 NA     8767
## 1373                  8                 NA     6944
## 1846                  8                 NA     8628
## 2210                  8                 NA     8593
## 2405                  8                 NA     6998
```

```
best_seller_city = best_seller %>%
                   select(User_ID, City_Category) %>%
                   distinct() %>%
                   count(City_Category)
best_seller_city
```

```
## # A tibble: 3 x 2
##   City_Category     n
##   <fct>         <int>
## 1 A               396
## 2 B               626
## 3 C               836
```

```
best_seller_city_vis = ggplot(data = best_seller_city, aes(x = City_Category, y = n, fill = City_Categor
                              geom_bar(color = 'white', stat = 'identity') +
                              labs(title = 'Recuento Compras Best-Seller Por Ciudad', y = 'Nº Compras',
                              scale_fill_brewer(palette = "Blues") +
                              theme(legend.position="none", plot.title = element_text(size = 9))
grid.arrange(city_count_purchases_vis,best_seller_city_vis, ncol = 2)
```



Aunque los clientes que residen en la Ciudad C compran más de nuestros "best seller" que la Ciudad A + B,

los residentes de la Ciudad C se quedan atrás de la Ciudad B en el número total de compras.

Análisis Variable Estancia en la Ciudad Actual (Años).

Examinemos ahora la distribución de los clientes que han vivido más tiempo en su ciudad.

```
customers_stay = datos %>%
                 select(User_ID, City_Category, Stay_In_Current_City_Years) %>%
                 group_by(User_ID) %>%
                 distinct()
head(customers_stay)
```

```
## # A tibble: 6 x 3
## # Groups:   User_ID [6]
##   User_ID City_Category Stay_In_Current_City_Years
##     <dbl> <fct>         <fct>
## 1       1 A             2
## 2       2 C             4+
## 3       3 A             3
## 4       4 B             2
## 5       5 A             1
## 6       6 A             1
```

Ahora que tenemos los datos en orden, podemos trazar y explorar.

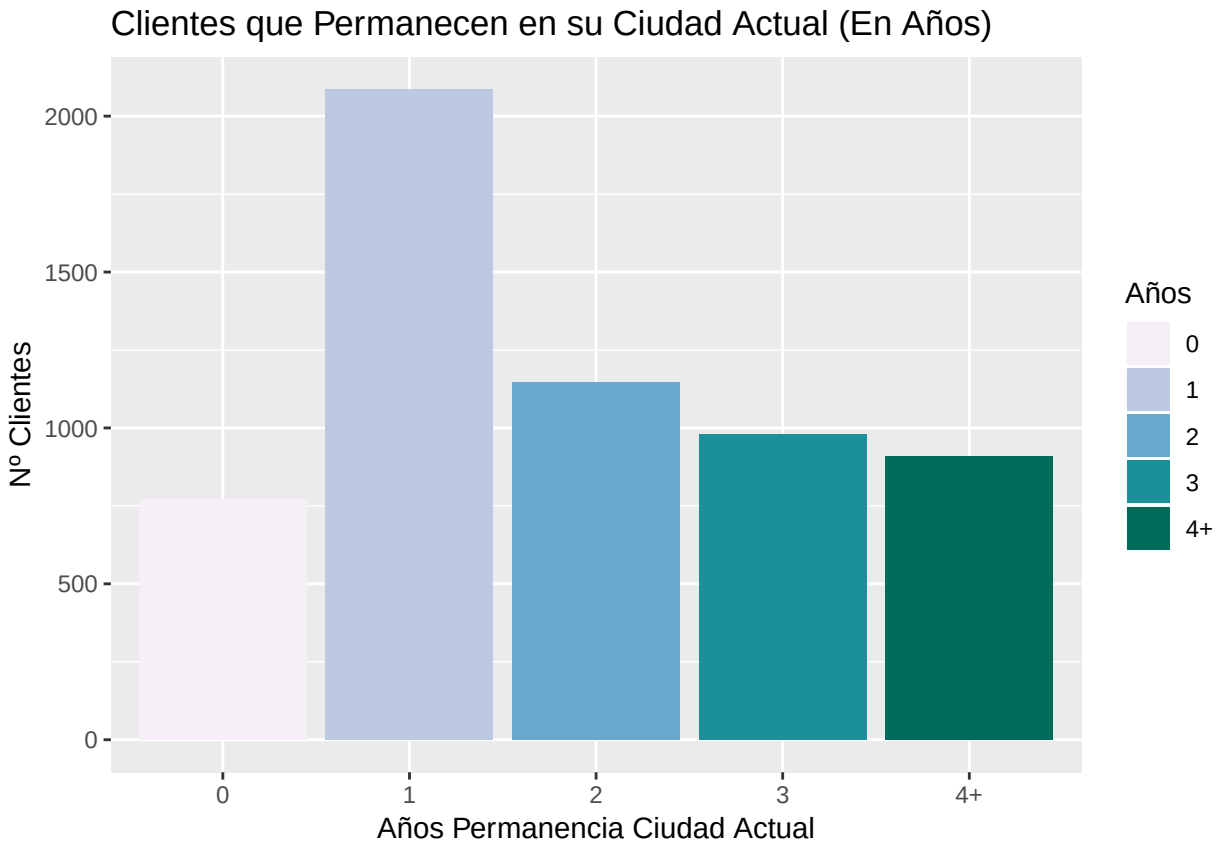
Veamos dónde viven la mayoría de nuestros clientes.

```
residence = customers_stay %>%
            group_by(City_Category) %>%
            tally()
head(residence)
```

```
## # A tibble: 3 x 2
##   City_Category     n
##   <fct>         <int>
## 1 A              1045
## 2 B              1707
## 3 C              3139
```

Parece que la mayoría de nuestros clientes viven en la Ciudad C.

```
customers_stay_vis = ggplot(data = customers_stay, aes(x = Stay_In_Current_City_Years, y = ..count.., fi
                            geom_bar(stat = 'count') +
                            scale_fill_brewer(palette = 10) +
                            labs(title = 'Clientes que Permanecen en su Ciudad Actual (En Años)', y = '
print(customers_stay_vis)
```

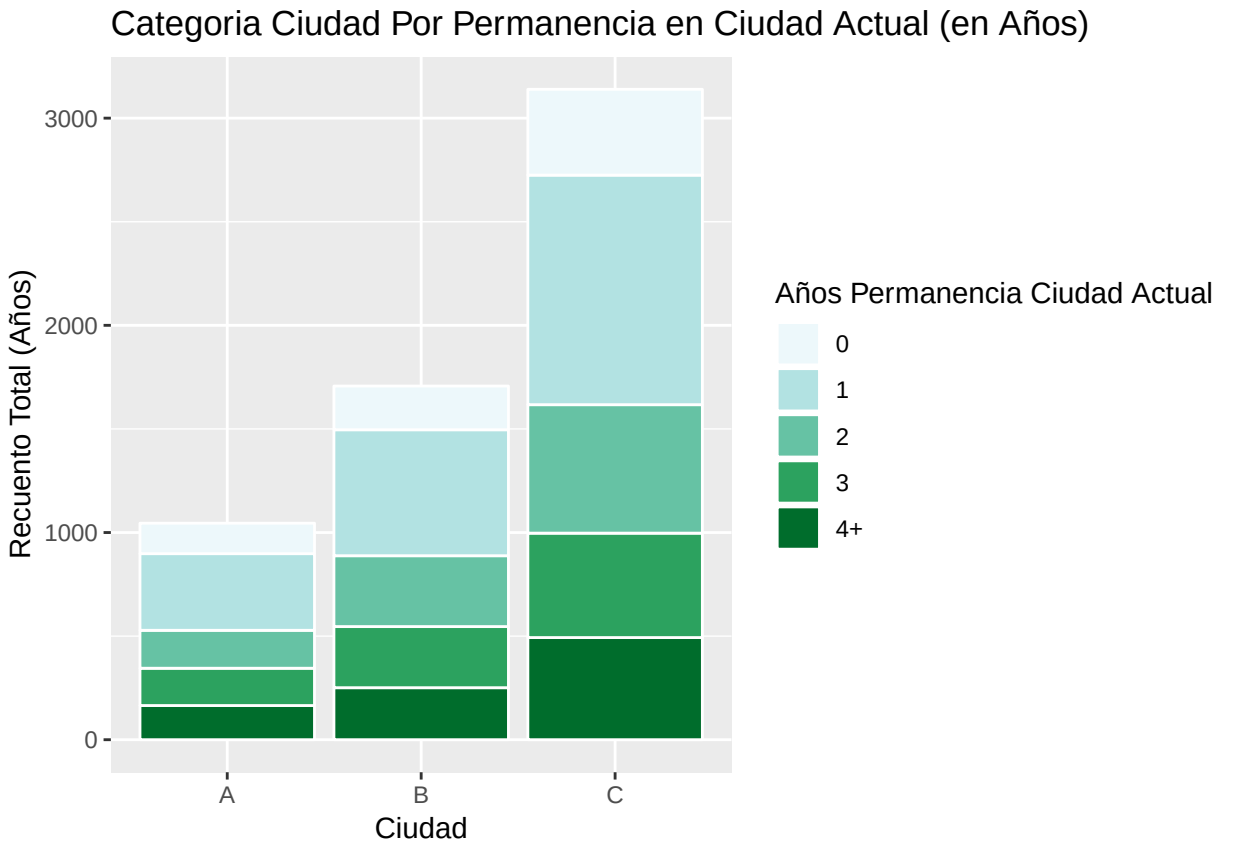


Parece que la mayoría de nuestros clientes sólo han vivido en sus respectivas ciudades durante un año. Para ver una mejor distribución, hagamos un gráfico de barras apiladas de acuerdo a cada Ciudad_Categoría.

```
stay_cities = customers_stay %>%
              group_by(City_Category, Stay_In_Current_City_Years) %>%
              tally() %>%
              mutate(Percentage = (n/sum(n))*100)
head(stay_cities)
```

```
## # A tibble: 6 x 4
## # Groups:   City_Category [2]
##   City_Category Stay_In_Current_City_Years     n Percentage
##   <fct>         <fct>                      <int>      <dbl>
## 1 A             0                            147       14.1
## 2 A             1                            370       35.4
## 3 A             2                            183       17.5
## 4 A             3                            180       17.2
## 5 A             4+                           165       15.8
## 6 B             0                            211       12.4
```

```
ggplot(data = stay_cities, aes(x = City_Category, y = n, fill = Stay_In_Current_City_Years)) +
       geom_bar(stat = "identity", color = 'white') +
       scale_fill_brewer(palette = 2) +
       labs(title = "Categoria Ciudad Por Permanencia en Ciudad Actual (en Años)",
            y = "Recuento Total (Años)",
            x = "Ciudad",
            fill = "Años Permanencia Ciudad Actual")
```



En este gráfico podemos ver la distribución de la base total de clientes y sus respectivas ciudades, dividida por la cantidad de tiempo que han vivido allí. Aquí, podemos notar que en cada Ciudad_Categoría, la duración de la estancia más común parece ser de 1 año.

Análisis Variable Importe Total Compras.

Vamos a hacer algunas investigaciones con respecto a los clientes y sus compras. Comenzaremos por calcular el monto total de la compra por identificación de usuario.

```
customers_total_purchase_amount = datos %>%
                                  group_by(User_ID) %>%
                                  summarise(Purchase_Amount = sum(Purchase))
head(customers_total_purchase_amount)
```

```
## # A tibble: 6 x 2
##   User_ID Purchase_Amount
##     <dbl>           <int>
## 1       1          333481
## 2       2          810353
## 3       3          341635
## 4       4          205987
## 5       5          821001
## 6       6          379450
```

Ahora que hemos agrupado nuestras compras y agrupado por ID de usuario, ordenaremos y encontraremos a los que más gasten.

```
customers_total_purchase_amount = arrange(customers_total_purchase_amount, desc((Purchase_Amount)))
head(customers_total_purchase_amount)
```

```
## # A tibble: 6 x 2
##   User_ID Purchase_Amount
##     <dbl>           <int>
## 1    4277        10536783
## 2    1680         8699232
## 3    2909         7577505
## 4    1941         6817493
## 5     424         6573609
## 6    4448         6565878
```

Parece que el ID de usuario 1004277 es el más gastador. Utilicemos summary() para ver otras facetas de nuestros datos de gasto total de clientes.

```
summary(customers_total_purchase_amount)
```

```
##     User_ID     Purchase_Amount
##  Min.   :   1   Min.   :   44108
##  1st Qu.:1518   1st Qu.:  234914
##  Median :3026   Median :  512612
##  Mean   :3025   Mean   :  851752
##  3rd Qu.:4532   3rd Qu.: 1099005
##  Max.   :6040   Max.   :10536783
```

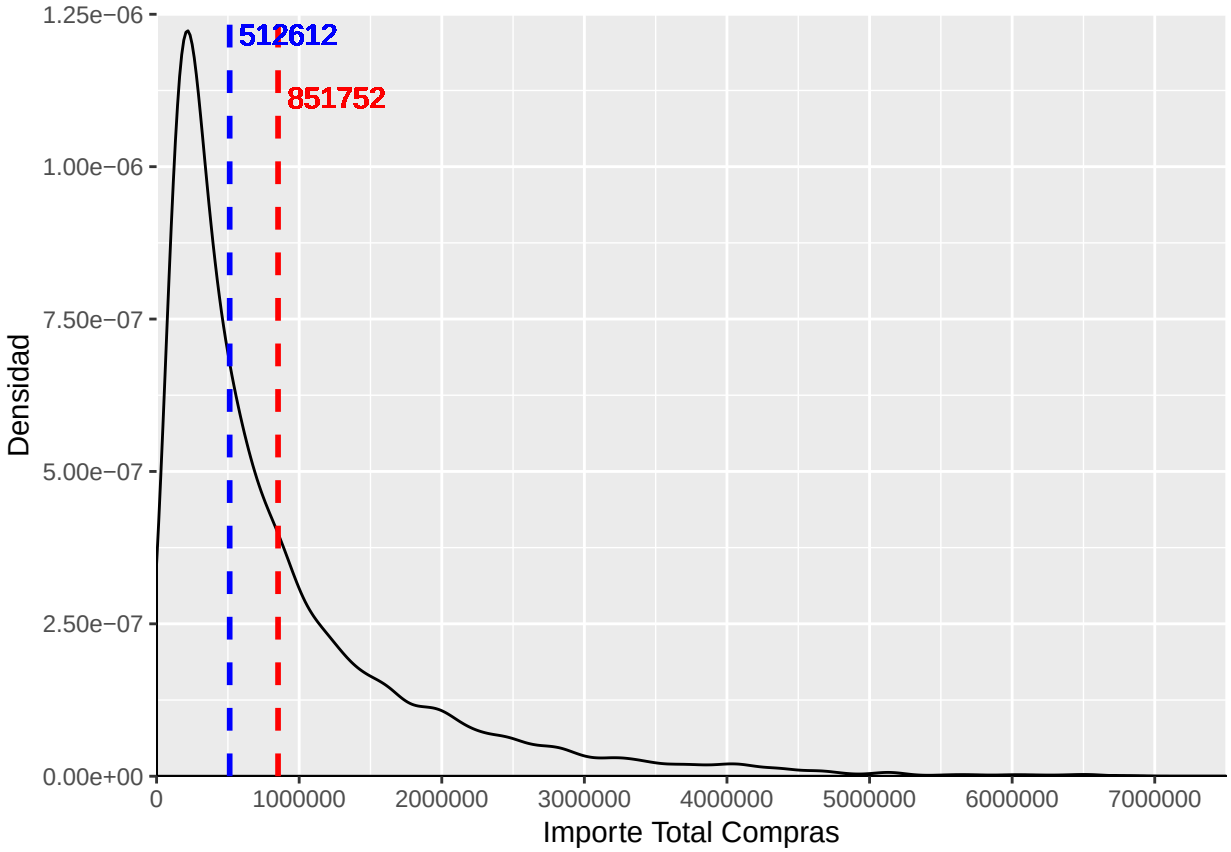
Podemos ver una cantidad total de compra promedio de 851752, una cantidad total de compra máxima de 10536783, una cantidad total de compra mínima de 44108 y una cantidad de compra media de 512612.

Vamos a trazar un gráfico que muestre la distribución de los montos de compra para ver si las compras se distribuyen normalmente o si contienen alguna asimetría. Un diagrama de densidad nos mostrará dónde se encuentra el mayor número de cantidades de compra similares de acuerdo con toda la base de clientes. Es importante notar que las tablas de densidad representan la probabilidad esperada de los valores, dados los datos como entrada, y luego trazan una línea alrededor de esos valores (estimación).

```
ggplot(customers_total_purchase_amount, aes(Purchase_Amount)) +
       geom_density(adjust = 1) +
       geom_vline(aes(xintercept=median(Purchase_Amount)),
                  color="blue", linetype="dashed", size=1) +
       geom_vline(aes(xintercept=mean(Purchase_Amount)),
                  color="red", linetype="dashed", size=1) +
       geom_text(aes(x=mean(Purchase_Amount), label=round(mean(Purchase_Amount)), y=1.2e-06), color = '
       geom_text(aes(x=median(Purchase_Amount), label=round(median(Purchase_Amount)), y=1.2e-06), color
       scale_x_continuous(name="Importe Total Compras", limits=c(0, 7500000), breaks = seq(0,7500000, by
       scale_y_continuous(name="Densidad", limits=c(0, .00000125), labels = scientific, expand = c(0,0))
```

```
## Warning: Ignoring unknown parameters: msize
```

```
## Warning: Removed 3 rows containing non-finite values (stat_density).
```



Aquí estamos viendo un diagrama de densidad sesgado muy a la derecha (positivo) con una larga cola. Esto significa que hay bastantes valores que se sitúan por encima de la media y que la mayor densidad de valores no es una serie distribuida de forma estándar. Vemos que la mayor densidad de compras se sitúa en torno a los 2.500.000 dólares.

Análisis Estado Civil.

Examinemos ahora el estado civil de los clientes de la tienda.

```
datos_maritalStatus = datos %>%
                        select(User_ID, Marital_Status) %>%
                        group_by(User_ID) %>%
                        distinct()
head(datos_maritalStatus)
```

```
## # A tibble: 6 x 2
## # Groups:   User_ID [6]
##   User_ID Marital_Status
##     <dbl>          <int>
## 1       1              0
## 2       2              0
## 3       3              0
## 4       4              1
## 5       5              1
## 6       6              0
```

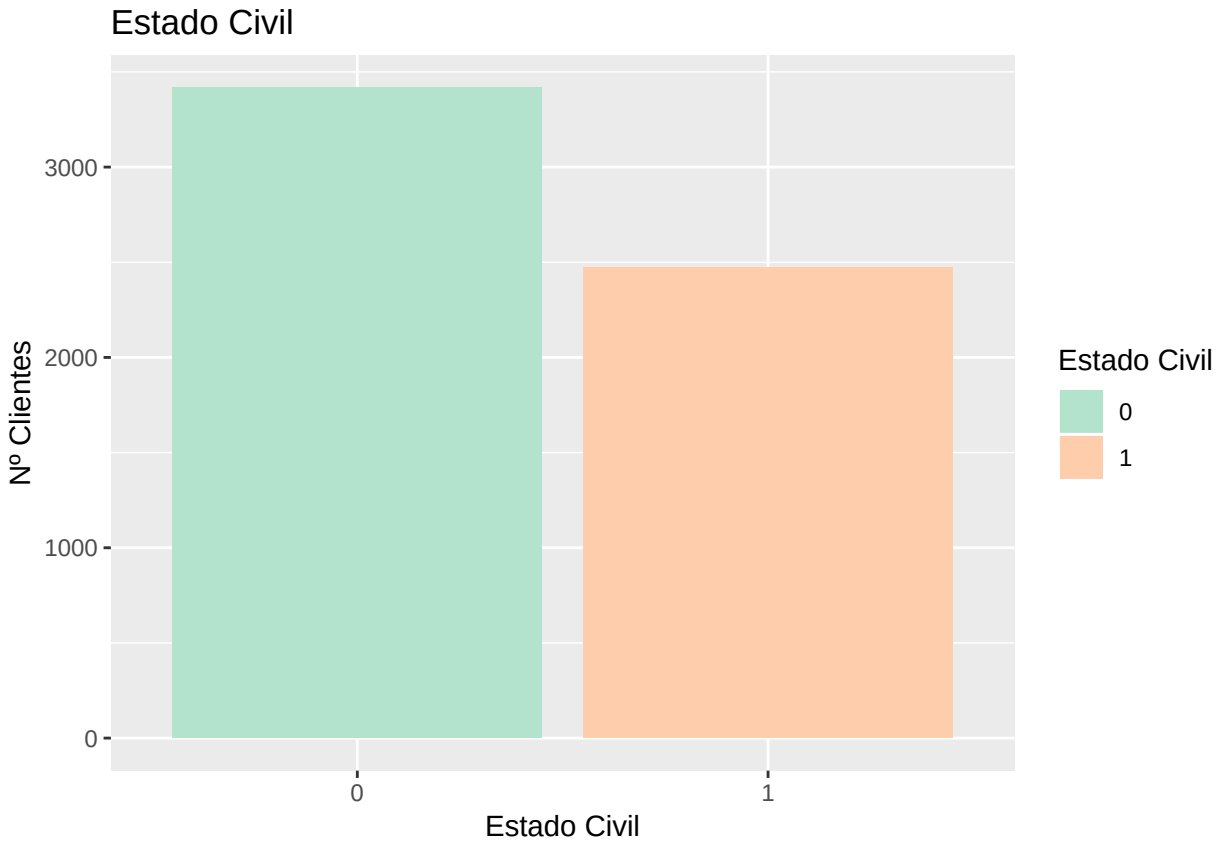
Para empezar necesitamos cambiar el tipo de la variable (numérica) a un tipo categórica.

```
datos_maritalStatus$Marital_Status = as.character(datos_maritalStatus$Marital_Status)
typeof(datos_maritalStatus$Marital_Status)
```

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

Si echamos la vista atrás a las descripciones variables de los datos, no tenemos una guía clara del estado civil. En este caso, asumiremos que 1 = casado y 0 = soltero.

```
marital_vis = ggplot(data = datos_maritalStatus) +
                    geom_bar(mapping = aes(x = Marital_Status, y = ..count.., fill = Marital_Status)) +
                    labs(title = 'Estado Civil', x = 'Estado Civil', y = 'Nº Clientes', fill = 'Estado
                    scale_fill_brewer(palette = 'Pastel2')
print(marital_vis)
```



Parece que la mayoría de nuestros compradores son solteros. De manera similar a nuestra investigación de los grupos de edad, podemos ver la composición del Estado_Matrimonial en cada Categoría_Ciudad.

```
datos_maritalStatus = datos_maritalStatus %>%
                        full_join(customers_stay, by = 'User_ID')
head(datos_maritalStatus)
```

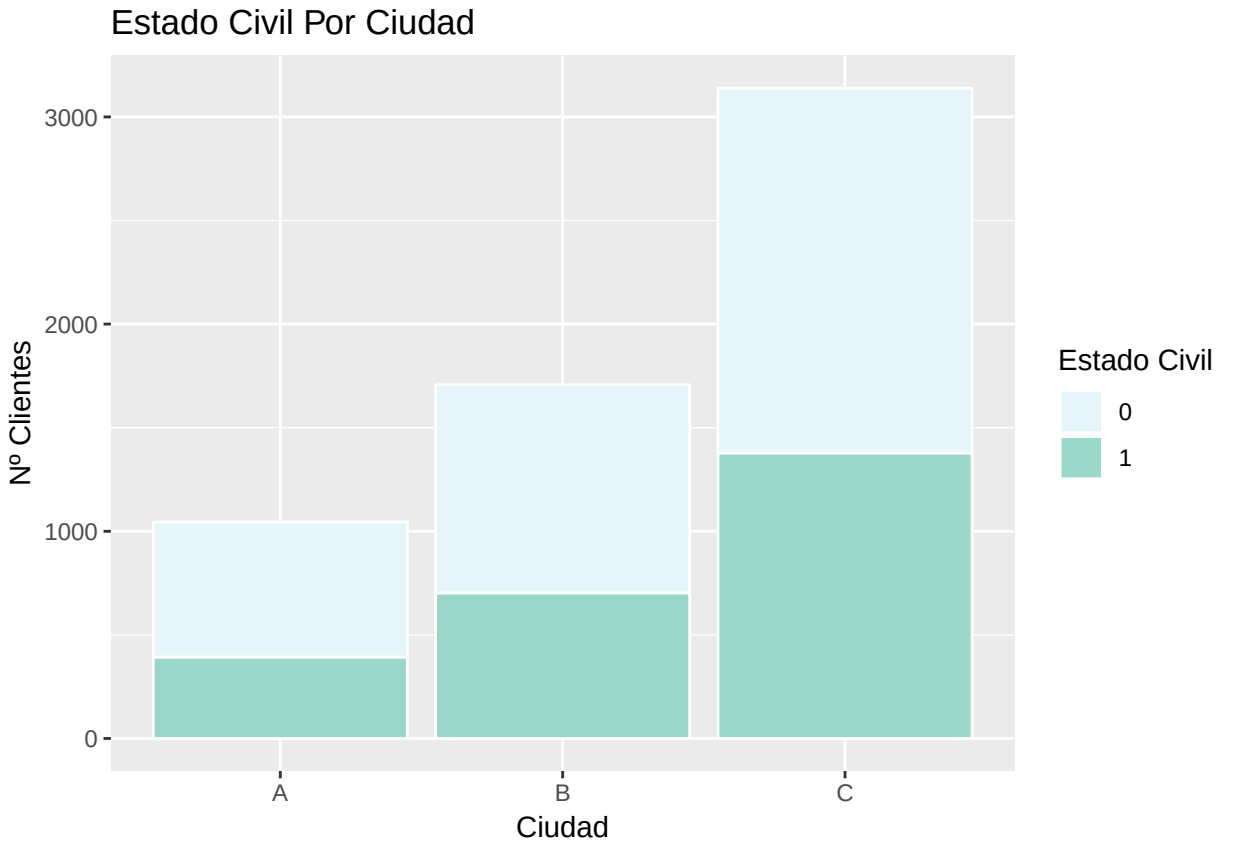
```
## # A tibble: 6 x 4
## # Groups:   User_ID [6]
##   User_ID Marital_Status City_Category Stay_In_Current_City_Years
##     <dbl> <chr>          <fct>         <fct>
## 1       1 0              A             2
## 2       2 0              C             4+
## 3       3 0              A             3
## 4       4 1              B             2
```

```
## 5       5 1              A             1
## 6       6 0              A             1
```

```
maritalStatus_cities = datos_maritalStatus %>%
                       group_by(City_Category, Marital_Status) %>%
                       tally()
head(maritalStatus_cities)
```

```
## # A tibble: 6 x 3
## # Groups:   City_Category [3]
##   City_Category Marital_Status     n
##   <fct>         <chr>          <int>
## 1 A             0                652
## 2 A             1                393
## 3 B             0               1004
## 4 B             1                703
## 5 C             0               1761
## 6 C             1               1378
```

```
ggplot(data = maritalStatus_cities, aes(x = City_Category, y = n, fill = Marital_Status)) +
       geom_bar(stat = "identity", color = 'white') +
       scale_fill_brewer(palette = 2) +
       labs(title = "Estado Civil Por Ciudad",
            y = "Nº Clientes",
            x = "Ciudad",
            fill = "Estado Civil")
```



Aquí, podemos ver que fuera de todas las ciudades, la mayor proporción de compradores individuales parece estar en la Ciudad A. Ahora, investiguemos la distribución de Stay_in_Current_City dentro de cada

Ciudad_Categoría.

```
Users_Age = datos %>%
            select(User_ID, Age) %>%
            distinct()
head(Users_Age)
```

```
##   User_ID   Age
## 1       1  0-17
## 2       2   55+
## 3       3 26-35
## 4       4 46-50
## 5       5 26-35
## 6       6 51-55
```

```
datos_maritalStatus = datos_maritalStatus %>%
                      full_join(Users_Age, by = 'User_ID')
head(datos_maritalStatus)
```

```
## # A tibble: 6 x 5
## # Groups:   User_ID [6]
##   User_ID Marital_Status City_Category Stay_In_Current_City_Years Age
##     <dbl> <chr>          <fct>         <fct>                      <fct>
## 1       1 0              A             2                          0-17
## 2       2 0              C             4+                         55+
## 3       3 0              A             3                          26-35
## 4       4 1              B             2                          46-50
## 5       5 1              A             1                          26-35
## 6       6 0              A             1                          51-55
```

```
City_A = datos_maritalStatus %>%
         filter(City_Category == 'A')
City_B = datos_maritalStatus %>%
         filter(City_Category == 'B')
City_C = datos_maritalStatus %>%
         filter(City_Category == 'C')
head(City_A)
```

```
## # A tibble: 6 x 5
## # Groups:   User_ID [6]
##   User_ID Marital_Status City_Category Stay_In_Current_City_Years Age
##     <dbl> <chr>          <fct>         <fct>                      <fct>
## 1       1 0              A             2                          0-17
## 2       3 0              A             3                          26-35
## 3       5 1              A             1                          26-35
## 4       6 0              A             1                          51-55
## 5      15 0              A             1                          26-35
## 6      19 0              A             3                          0-17
```

```
head(City_B)
```

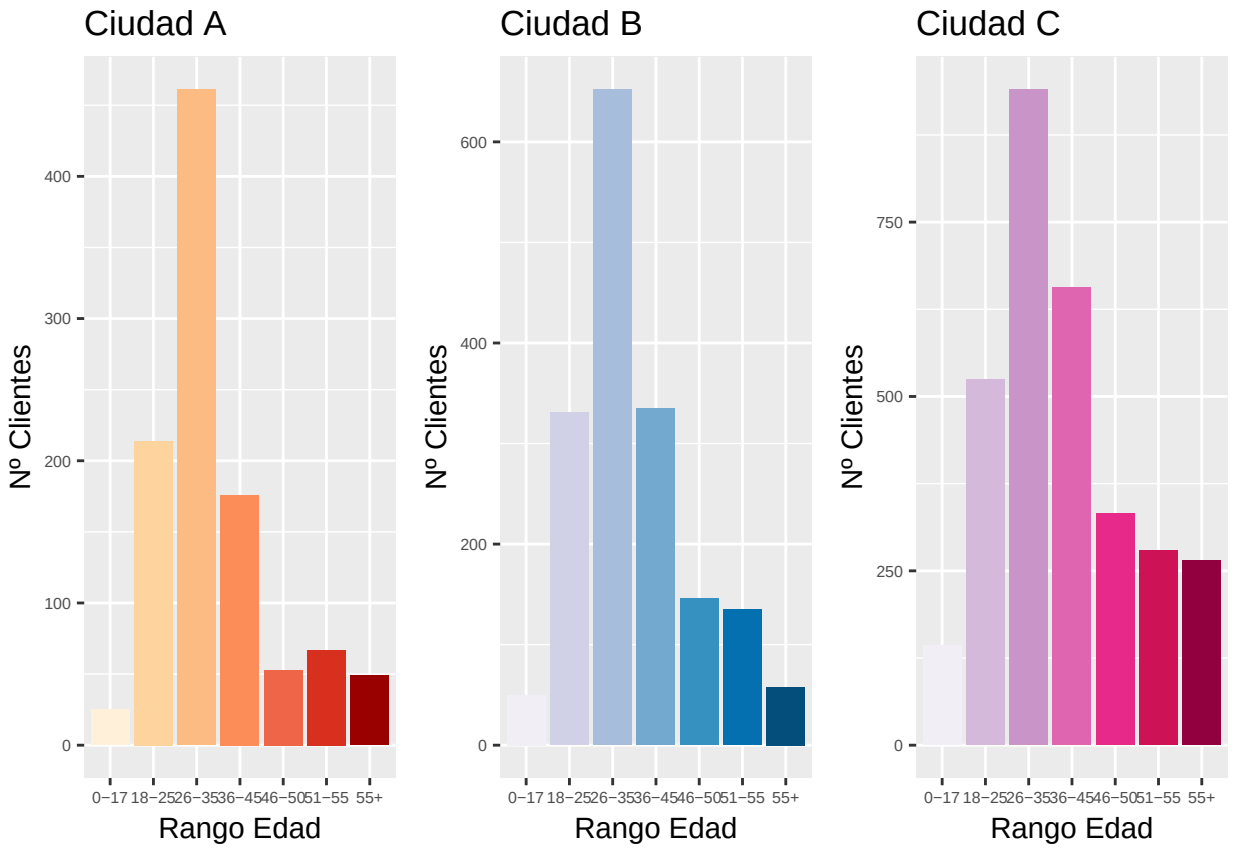
```
## # A tibble: 6 x 5
## # Groups:   User_ID [6]
##   User_ID Marital_Status City_Category Stay_In_Current_City_Years Age
##     <dbl> <chr>          <fct>         <fct>                      <fct>
## 1       4 1              B             2                          46-50
## 2       7 1              B             1                          36-45
```

```
## 3      10 1              B             4+                          36-45
## 4      18 0              B             3                           18-25
## 5      21 0              B             0                           18-25
## 6      23 1              B             3                           36-45
```

```
head(City_C)
```

```
## # A tibble: 6 x 5
## # Groups:   User_ID [6]
##   User_ID Marital_Status City_Category Stay_In_Current_City_Years Age
##     <dbl> <chr>          <fct>         <fct>                       <fct>
## 1       2 0              C             4+                          55+
## 2       8 1              C             4+                          26-35
## 3       9 0              C             0                           26-35
## 4      11 0              C             1                           26-35
## 5      12 0              C             2                           26-35
## 6      13 1              C             3                           46-50
```

```
City_A_stay_vis = ggplot(data = City_A, aes(x = Age, y = ..count.., fill = Age)) +
                         geom_bar(stat = 'count') +
                         scale_fill_brewer(palette = 8) +
                         theme(legend.position="none", axis.text = element_text(size = 6)) +
                         labs(title = 'Ciudad A', y = 'Nº Clientes', x = 'Rango Edad', fill = 'Rango Eda
City_B_stay_vis = ggplot(data = City_B, aes(x = Age, y = ..count.., fill = Age)) +
                         geom_bar(stat = 'count') +
                         scale_fill_brewer(palette = 9) +
                         theme(legend.position="none", axis.text = element_text(size = 6)) +
                         labs(title = 'Ciudad B', y = 'Nº Clientes', x = 'Rango Edad', fill = 'Rango Eda
City_C_stay_vis = ggplot(data = City_C, aes(x = Age, y = ..count.., fill = Age)) +
                         geom_bar(stat = 'count') +
                         scale_fill_brewer(palette = 11) +
                         theme(legend.position="none", axis.text = element_text(size = 6)) +
                         labs(title = 'Ciudad C', y = 'Nº Clientes', x = 'Rango Edad', fill = 'Rango Eda
grid.arrange(City_A_stay_vis, City_B_stay_vis, City_C_stay_vis, ncol = 3)
```



Parece que la ciudad A tiene menos compradores mayores de 45 años que las otras ciudades. Esto podría ser un factor en los niveles resultantes de Estado_Matrimonial dentro de cada ciudad.

Análisis Variable TOP Compradores.

Ahora, investigaremos quiénes fueron nuestros principales compradores durante el Black Friday.

```
top_shoppers = datos %>%
               count(User_ID, sort = TRUE)
head(top_shoppers)
```

```
## # A tibble: 6 x 2
##   User_ID     n
##     <dbl> <int>
## 1    1680  1025
## 2    4277   978
## 3    1941   898
## 4    1181   861
## 5     889   822
## 6    3618   766
```

Parece que User_ID 1001680 es el que más aparece. Cada línea individual representa una transacción/producto diferente, por lo que este usuario hizo más de 1.000 transacciones totales. Podemos unir estos datos de los mejores compradores con los datos de compras totales de clientes para verlos combinados conjuntamente.

```
top_shoppers =  top_shoppers %>%
                select(User_ID, n) %>%
```

```
                left_join(customers_total_purchase_amount, Purchase_Amount, by = 'User_ID')
head(top_shoppers)
```

```
## # A tibble: 6 x 3
##   User_ID     n Purchase_Amount
##     <dbl> <int>           <int>
## 1    1680  1025         8699232
## 2    4277   978        10536783
## 3    1941   898         6817493
## 4    1181   861         6387899
## 5     889   822         5499812
## 6    3618   766         5961987
```

Ahora que hemos unido las dos tablas, podemos ver que aunque User_ID 1001680 tiene el mayor número de compras totales, User_ID 1004277 tiene la mayor cantidad de compra identificada en nuestros gráficos anteriores también. A partir de aquí, también podemos calcular el promedio de la cantidad_compra para cada usuario.

```
top_shoppers = mutate(top_shoppers,
                Average_Purchase_Amount = Purchase_Amount/n)
head(top_shoppers)
```

```
## # A tibble: 6 x 4
##   User_ID     n Purchase_Amount Average_Purchase_Amount
##     <dbl> <int>           <int>                   <dbl>
## 1    1680  1025         8699232                   8487.
## 2    4277   978        10536783                  10774.
## 3    1941   898         6817493                   7592.
## 4    1181   861         6387899                   7419.
## 5     889   822         5499812                   6691.
## 6    3618   766         5961987                   7783.
```

Clasificamos de acuerdo a la Cantidad_de_Compra_Promedio para ver qué clientes, en promedio, están gastando más.

```
top_shoppers_averagePurchase = top_shoppers %>%
                               arrange(desc(Average_Purchase_Amount))
head(top_shoppers_averagePurchase)
```

```
## # A tibble: 6 x 4
##   User_ID     n Purchase_Amount Average_Purchase_Amount
##     <dbl> <int>           <int>                   <dbl>
## 1    5069    16          308454                  19278.
## 2    3902    93         1746284                  18777.
## 3    5999    18          330227                  18346.
## 4    1349    23          417743                  18163.
## 5     101    65         1138239                  17511.
## 6    3461    20          350174                  17509.
```

Parece que User_ID 1005069 tiene la cantidad de compra promedio más alta y una cantidad total de compra de 308454. User_ID 1003902 está justo detrás de User_ID 1005069 en Average_Purchase_Amount, pero tiene una cantidad total de compra mucho mayor de 1746284.

Análisis Variable Empleo Clientes.

Lo último que analizaremos es la ocupación de los clientes en nuestros datos.

```
customers_Occupation =  datos %>%
                        select(User_ID, Occupation) %>%
                        group_by(User_ID) %>%
                        distinct() %>%
                        left_join(customers_total_purchase_amount, Occupation, by = 'User_ID')
head(customers_Occupation)
```

```
## # A tibble: 6 x 3
## # Groups:   User_ID [6]
##   User_ID Occupation Purchase_Amount
##     <dbl>      <int>           <int>
## 1       1         10          333481
## 2       2         16          810353
## 3       3         15          341635
## 4       4          7          205987
## 5       5         20          821001
## 6       6          9          379450
```

Ahora que tenemos los datos necesarios, podemos agrupar el importe total de la compra para cada identificador de ocupación. A continuación, convertiremos la Ocupación en un tipo de datos carácter.

```
totalPurchases_Occupation = customers_Occupation %>%
                            group_by(Occupation) %>%
                            summarise(Purchase_Amount = sum(Purchase_Amount)) %>%
                            arrange(desc(Purchase_Amount))
totalPurchases_Occupation$Occupation = as.character(totalPurchases_Occupation$Occupation)
typeof(totalPurchases_Occupation$Occupation)
```

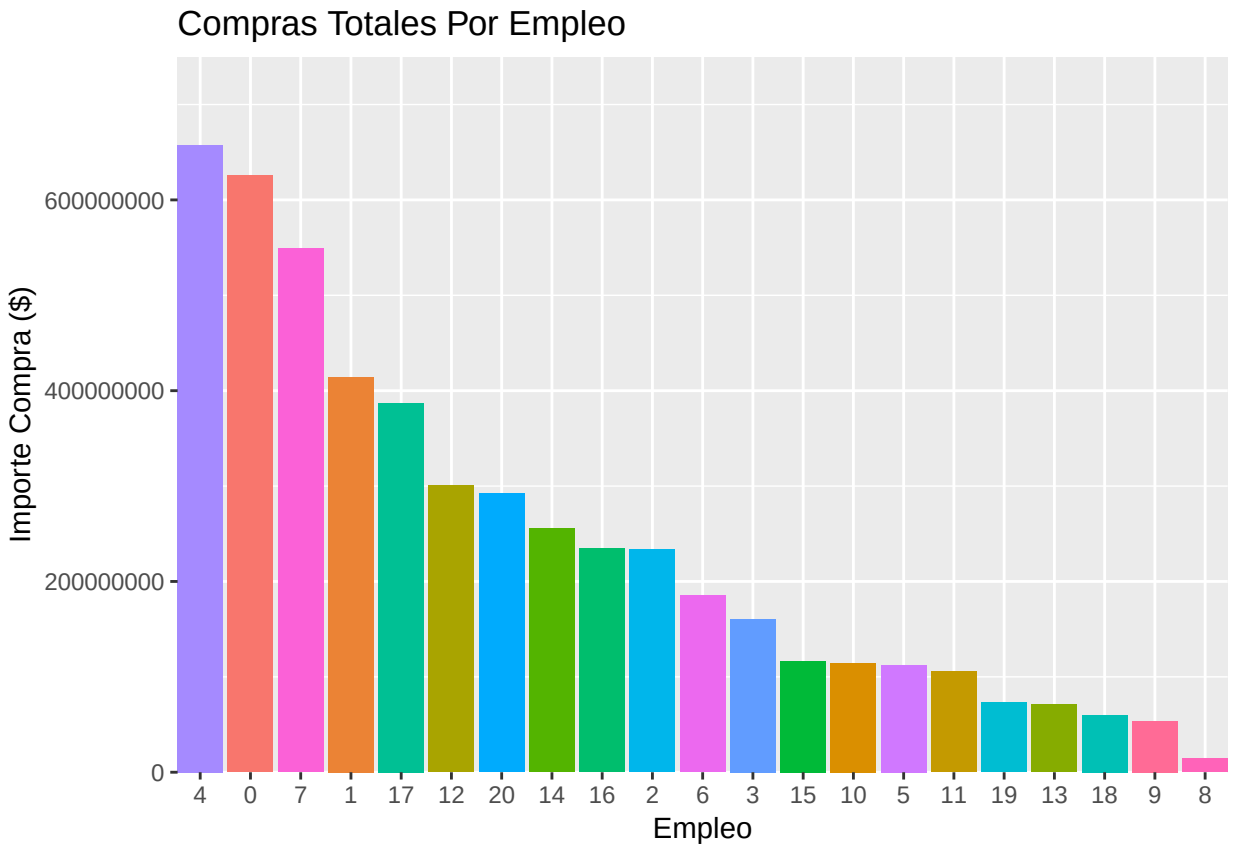
```
## [1] "character"
```

```
head(totalPurchases_Occupation)
```

```
## # A tibble: 6 x 2
##   Occupation Purchase_Amount
##   <chr>                <int>
## 1 4                657530393
## 2 0                625814811
## 3 7                549282744
## 4 1                414552829
## 5 17               387240355
## 6 12               300672105
```

Ahora, vamos a trazar cada ocupación y su total Purchase_Amount.

```
occupation = ggplot(data = totalPurchases_Occupation) +
                    geom_bar(mapping = aes(x = reorder(Occupation, -Purchase_Amount), y = Purchase_Amou
                    scale_x_discrete(name="Empleo", breaks = seq(0,20, by = 1), expand = c(0,0)) +
                    scale_y_continuous(name="Importe Compra ($)", expand = c(0,0), limits = c(0, 750000
                    labs(title = 'Compras Totales Por Empleo') +
                    theme(legend.position="none")
print(occupation)
```



Parece que los clientes etiquetados como Empleo 4, Empleo 0 y Empleo 7 pasaron la mayor parte del tiempo en el Black Friday, con los clientes de Ocupación 1 ya muy por detrás.

Aprendizaje No Supervisado

```
datos1 <- datos
```

Tratamiento de la muestra

Verificamos valores no determinados (NA)

```
na_Product_ID <- sum(is.na(datos1$Product_ID))
na_Gender <- sum(is.na(datos1$Gender))
na_Age <- sum(is.na(datos1$Age))
na_Occupation <- sum(is.na(datos1$Occupation))
na_City_Category <- sum(is.na(datos1$City_Category))
na_Stay_In_Current_City_Years <- sum(is.na(datos1$Stay_In_Current_City_Years))
na_Marital_Status <- sum(is.na(datos1$Product_ID))
na_Product_Category_1 <- sum(is.na(datos1$Product_ID))
na_Product_Category_2 <- sum(is.na(datos1$Product_ID))
na_Product_Category_3 <- sum(is.na(datos1$Product_ID))
na_Purchase <- sum(is.na(datos1$Product_ID))
```

Eliminamos valores no determinados (NA)

```
datos2 <- datos1[complete.cases(datos1), ]
dim(datos2)
```

```
## [1] 164278     12
```

Conversión de variables

```
#datos2$Gender <- revalue(datos2$Gender, c("M"=0, "F"=1))
#datos2$Age <- revalue(datos2$Age, c("0-17"=1, "18-25"=2, "26-35"=3, "36-45"=4, "46-50"=5, "51-55"=6, "
#datos2$Stay_In_Current_City_Years <- revalue(datos2$Stay_In_Current_City_Years, c("4+"=4))
```

```
set.seed(10000)

inTraining <- createDataPartition(datos2$Purchase, p=0.6, list=FALSE)
training.set <- datos2[inTraining,]

Totalvalidation.set <- datos2[-inTraining,]
```

```
set.seed(10000)
inValidation <- createDataPartition(Totalvalidation.set$Purchase, p=0.5, list=FALSE)
testing.set <- Totalvalidation.set[inValidation,]
validation.set <- Totalvalidation.set[-inValidation,]
```

Por tanto tenemos 3 matrices de datos para hacer la validación cruzada

- training.set
- validation.set
- testing.set

```
head(training.set)
```

```
##    User_ID Product_ID Gender   Age Occupation City_Category
## 7        4  P00184942      M 46-50          7             B
## 15       6  P00231342      F 51-55          9             A
## 20       8  P00249542      M 26-35         12             C
## 25       8  P00303442      M 26-35         12             C
## 29       9  P00078742      M 26-35         17             C
## 30      10  P00085942      F 36-45          1             B
##    Stay_In_Current_City_Years Marital_Status Product_Category_1
## 7                           2              1                  1
## 15                          1              0                  5
## 20                         4+              1                  1
## 25                         4+              1                  1
## 29                          0              0                  5
## 30                         4+              1                  2
##    Product_Category_2 Product_Category_3 Purchase
## 7                   8                 17    19215
## 15                  8                 14     5378
## 20                  5                 15    19614
## 25                  8                 14    11927
## 29                  8                 14     5391
## 30                  4                  8    16352
```

```
head(validation.set)
```

```
##    User_ID Product_ID Gender   Age Occupation City_Category
## 14       5  P00145042      M 26-35         20             A
## 37      10  P00182642      F 36-45          1             B
## 42      10  P00111142      F 36-45          1             B
## 44      10   P0094542      F 36-45          1             B
## 70      17  P00073842      M 51-55          1             C
## 72      18  P00190742      F 18-25          3             B
##    Stay_In_Current_City_Years Marital_Status Product_Category_1
## 14                          1              1                  1
## 37                         4+              1                  2
## 42                         4+              1                  1
## 44                         4+              1                  2
## 70                          0              0                  1
## 72                          3              0                  3
##    Product_Category_2 Product_Category_3 Purchase
## 14                  2                  5    15665
## 37                  4                  9    12909
## 42                 15                 16    18963
## 44                  4                  9    16406
## 70                 15                 17    15172
## 72                  4                  9    10754
```

```
head(testing.set)
```

```
##    User_ID Product_ID Gender   Age Occupation City_Category
## 2        1  P00248942      F  0-17         10             A
## 17       6   P0096642      F 51-55          9             A
## 19       7  P00036842      M 36-45          1             B
## 45      10  P00148642      F 36-45          1             B
## 47      10  P00113242      F 36-45          1             B
## 77      18  P00222242      F 18-25          3             B
##    Stay_In_Current_City_Years Marital_Status Product_Category_1
## 2                           2              0                  1
## 17                          1              0                  2
## 19                          1              1                  1
## 45                         4+              1                  6
## 47                         4+              1                  1
## 77                          3              0                  1
##    Product_Category_2 Product_Category_3 Purchase
## 2                   6                 14    15200
## 17                  3                  4    13055
## 19                 14                 16    11788
## 45                 10                 13    12642
## 47                  6                  8    11562
## 77                  2                 13    15182
```

Análisis Cluster

Con muestra de entrenamiento en función del valor de compra

Determinar el número de grupos (clusters) por el Método de k-means

Cálculo de los centroides (k)

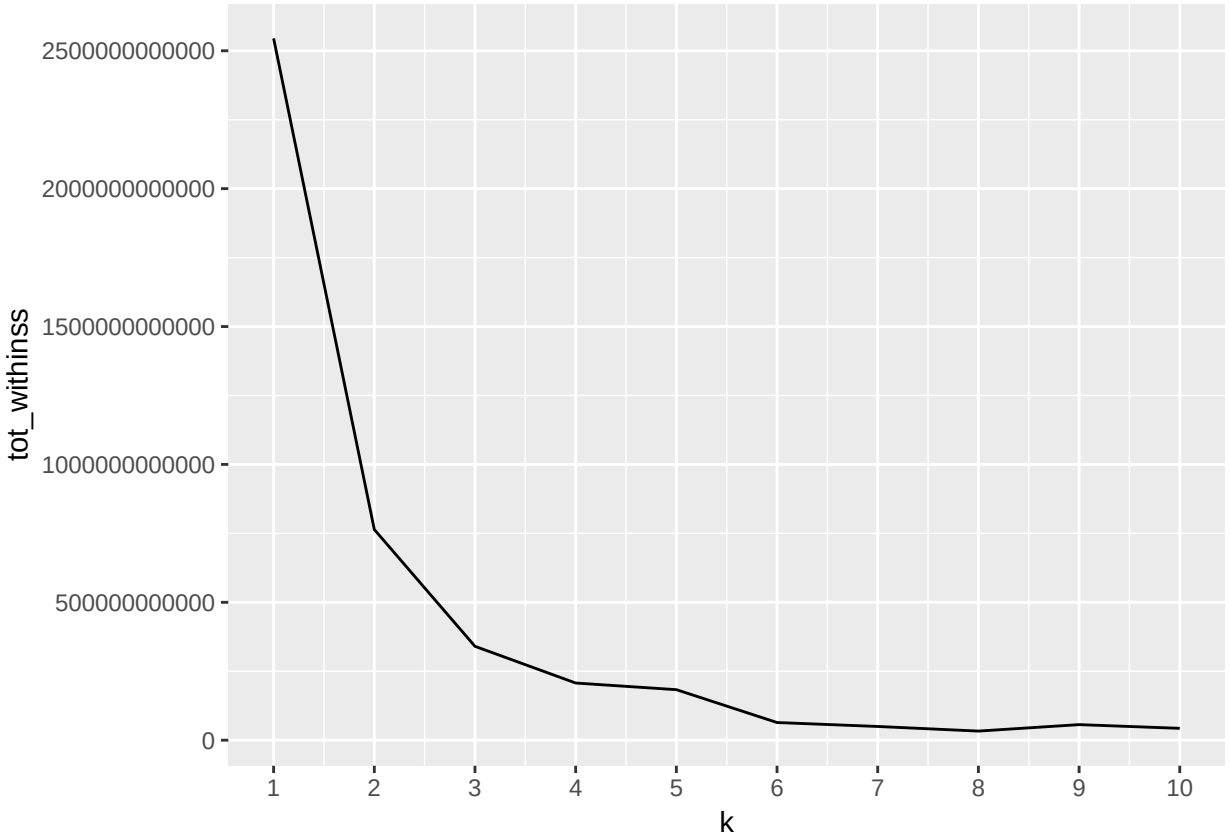
```
tot_withinss <- map_dbl(1:10,  function(k){
  model <- kmeans(x = BF_Cluster, centers = k)
  model$tot.withinss
})
```

Estimación de la cantidad de grupos posibles por el método del codo

```
elbow_df <- data.frame(
  k = 1:10,
  tot_withinss = tot_withinss
)
```

Graficamos el codo

```
ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
  geom_line() +
  scale_x_continuous(breaks = 1:10)
```



Podemos visualizar que podemos considerar hasta 3 clusters

Generamos un análisis cluster por el método de k-means

```
meto_km <- kmeans(BF_Cluster, centers = 3)
```

Vector de asignación de clusters con el método k-means

```
clust_km <- meto_km$cluster
```

Crear una base de datos nuevas, añadiendo las asignaciones de cluster

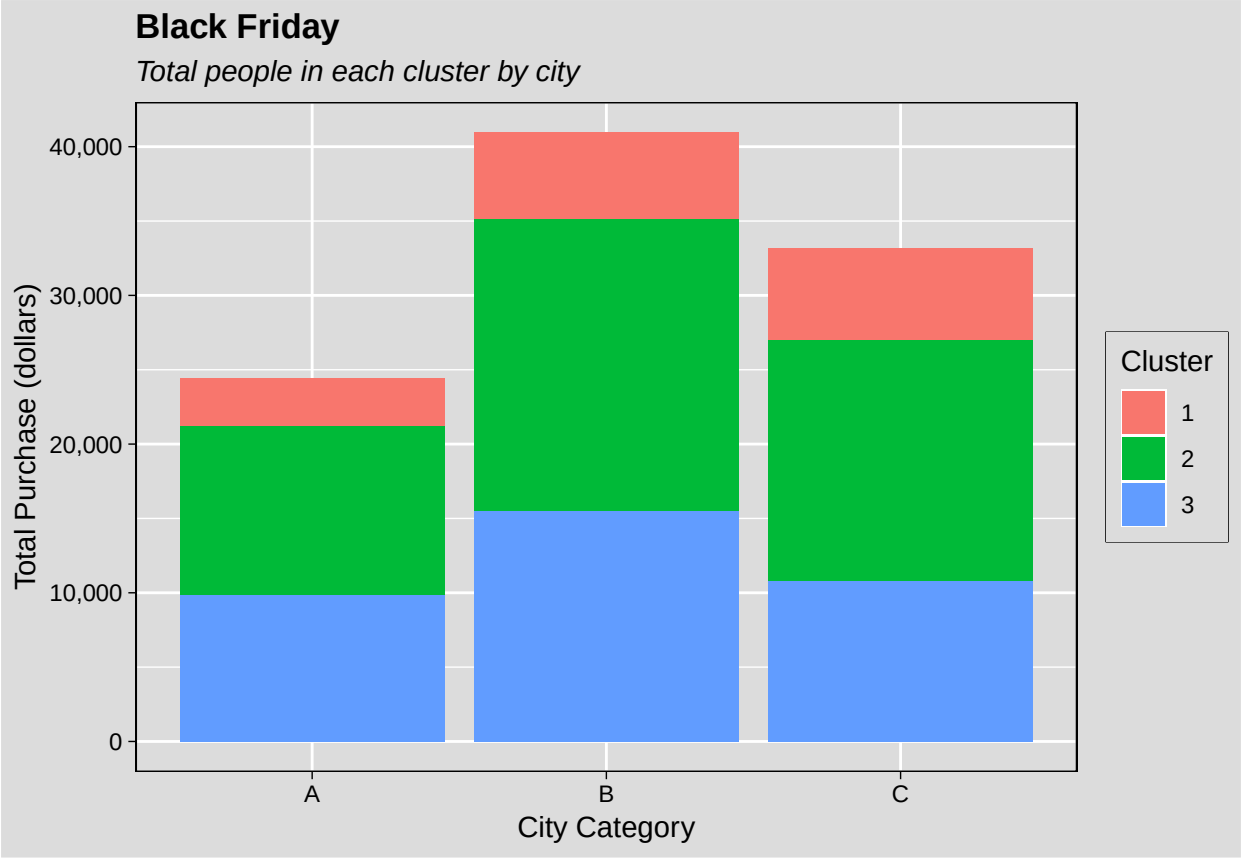
```
BF_Cluster <- mutate(training.set, cluster = clust_km)
```

Resumen de los parámetros del custer

```
BF_Clust_Res <- BF_Cluster %>%
                          group_by(cluster) %>%
                          dplyr::summarize(min_purchase = min(Purchase),
                                   max_purchase = max(Purchase),
                                   avg_purchase = round(mean(Purchase),0))
```

Clasificación de los compradores en función de las ciudades

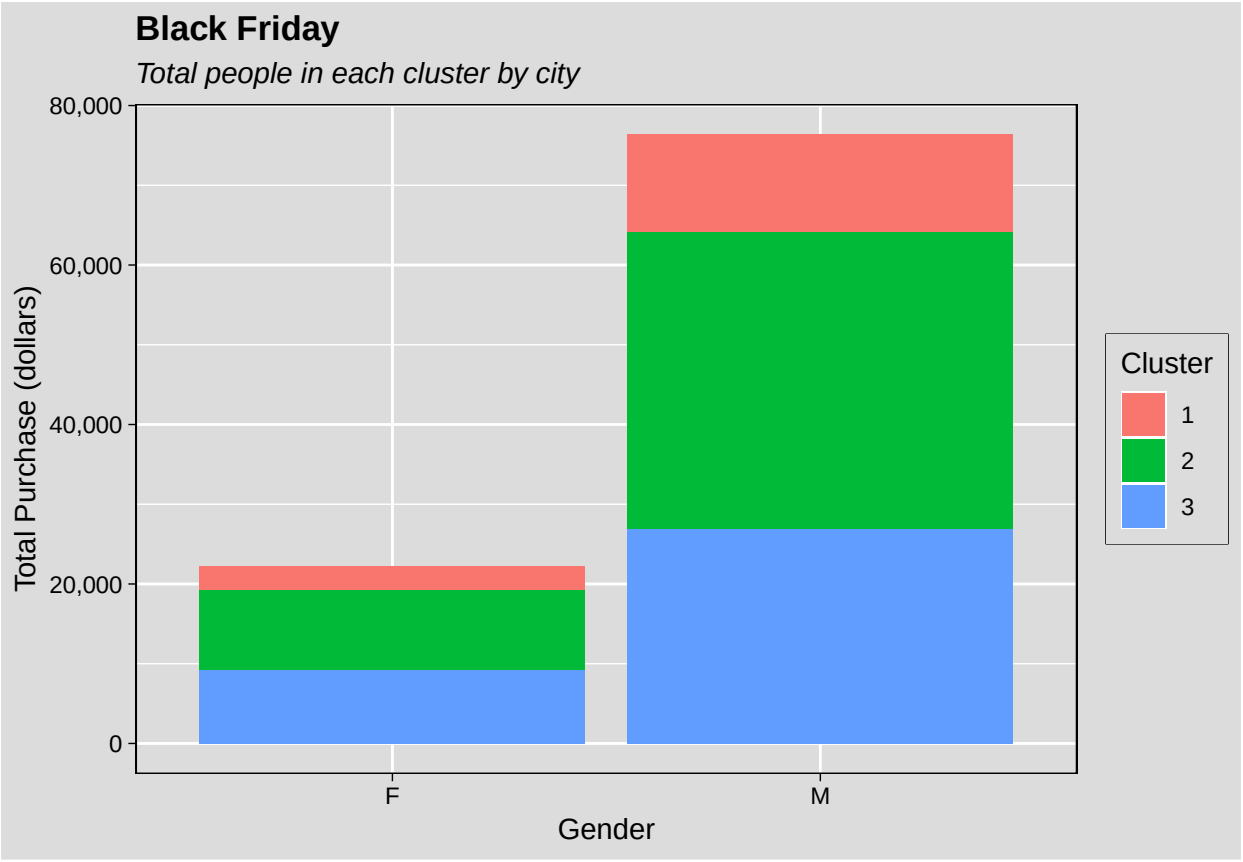
```
BF_Cluster %>%
group_by(City_Category, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=City_Category, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'City Category', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```



Vemos que el cluster 2 es el que más agrupaciones presenta en las 3 ciudades, seguido por el cluster 3.

Clasificación de los compradores en función del género

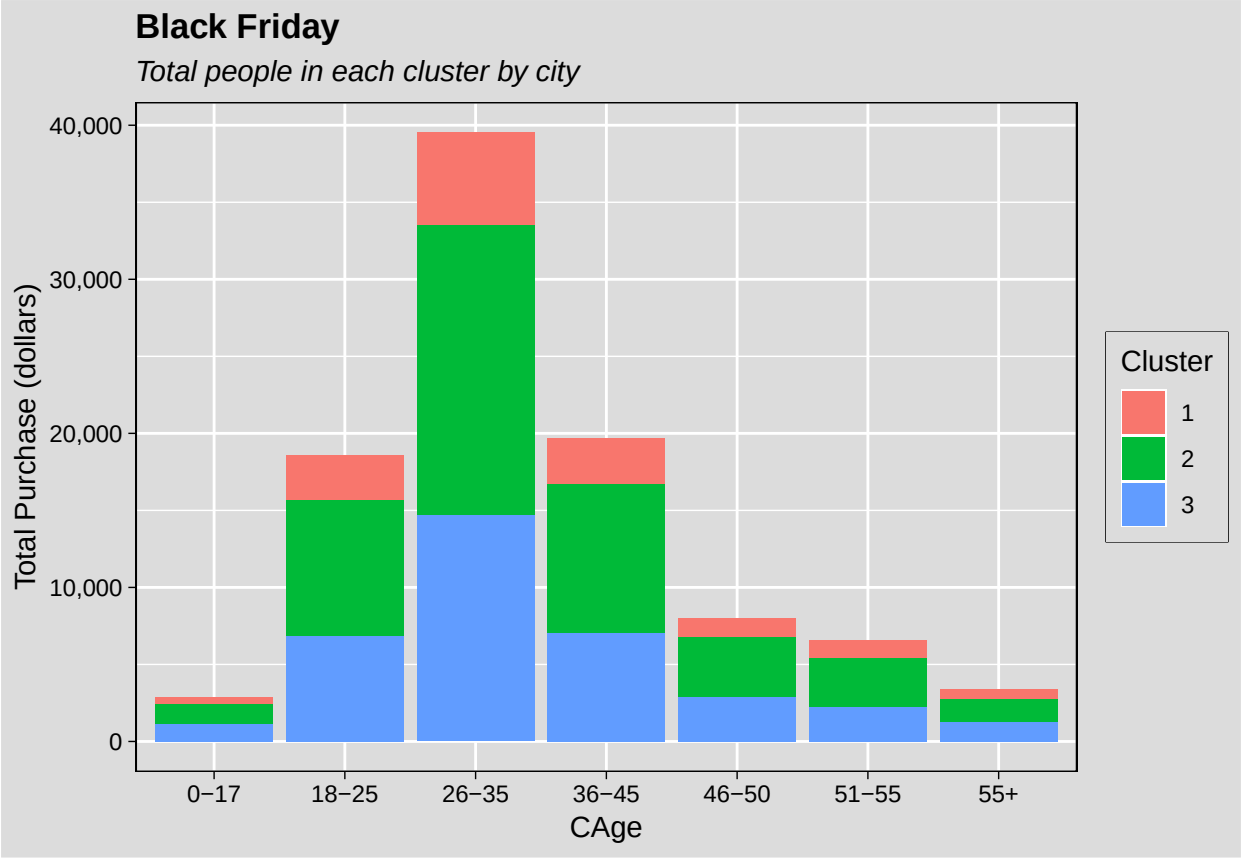
```
BF_Cluster %>%
group_by(Gender, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=Gender, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'Gender', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```



Vemos que el cluster 2 y 3 son los más representativos en ambos géneros

Clasificación de los compradores en función de los tramos de edad

```
BF_Cluster %>%
group_by(Age, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=Age, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'CAge', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```



Vemos que el segundo y tercer cluster son los más representativos en todos los tramos de genero. Por tanto no se aprecia una diferenciación en cuanto a los tramos de edad

Con muestra de prueba

Determinar el número de grupos (clusters) k-means

Cálculo de los centroides (k)

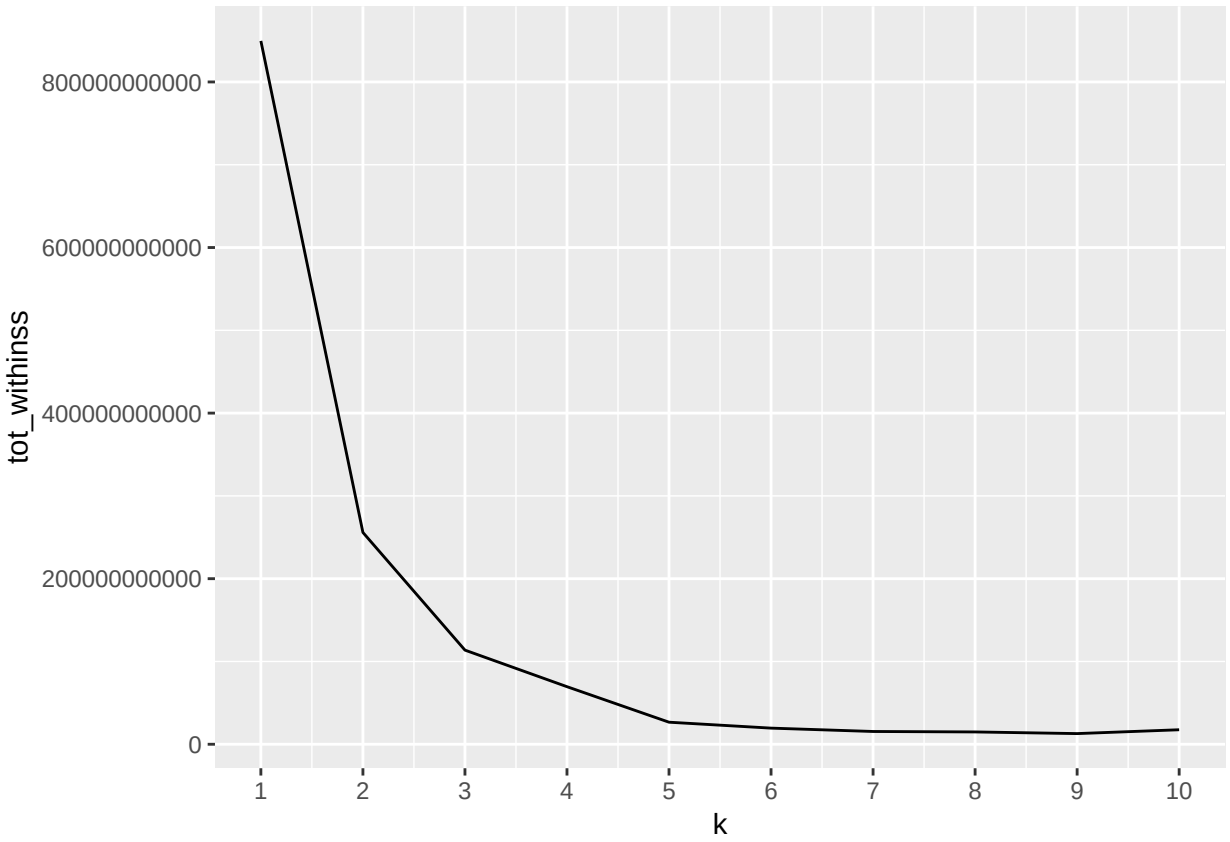
```
tot_withinss <- map_dbl(1:10,  function(k){
  model <- kmeans(x = BF_Cluster, centers = k)
  model$tot.withinss
})
```

Estimación de la cantidad de grupos posibles por el método del codo

```
elbow_df <- data.frame(
  k = 1:10,
  tot_withinss = tot_withinss
)
```

Graficamos el codo

```
ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
  geom_line() +
  scale_x_continuous(breaks = 1:10)
```



Podemos visualizar que podemos considerar hasta 3 clusters

Generamos un análisis cluster por el método de k-means

```
meto_km <- kmeans(BF_Cluster, centers = 3)
```

Vector de asignación de clusters con el método k-means

```
clust_km <- meto_km$cluster
```

Crear una base de datos nuevas, añadiendo las asignaciones de cluster

```
BF_Cluster <- mutate(testing.set, cluster = clust_km)
```

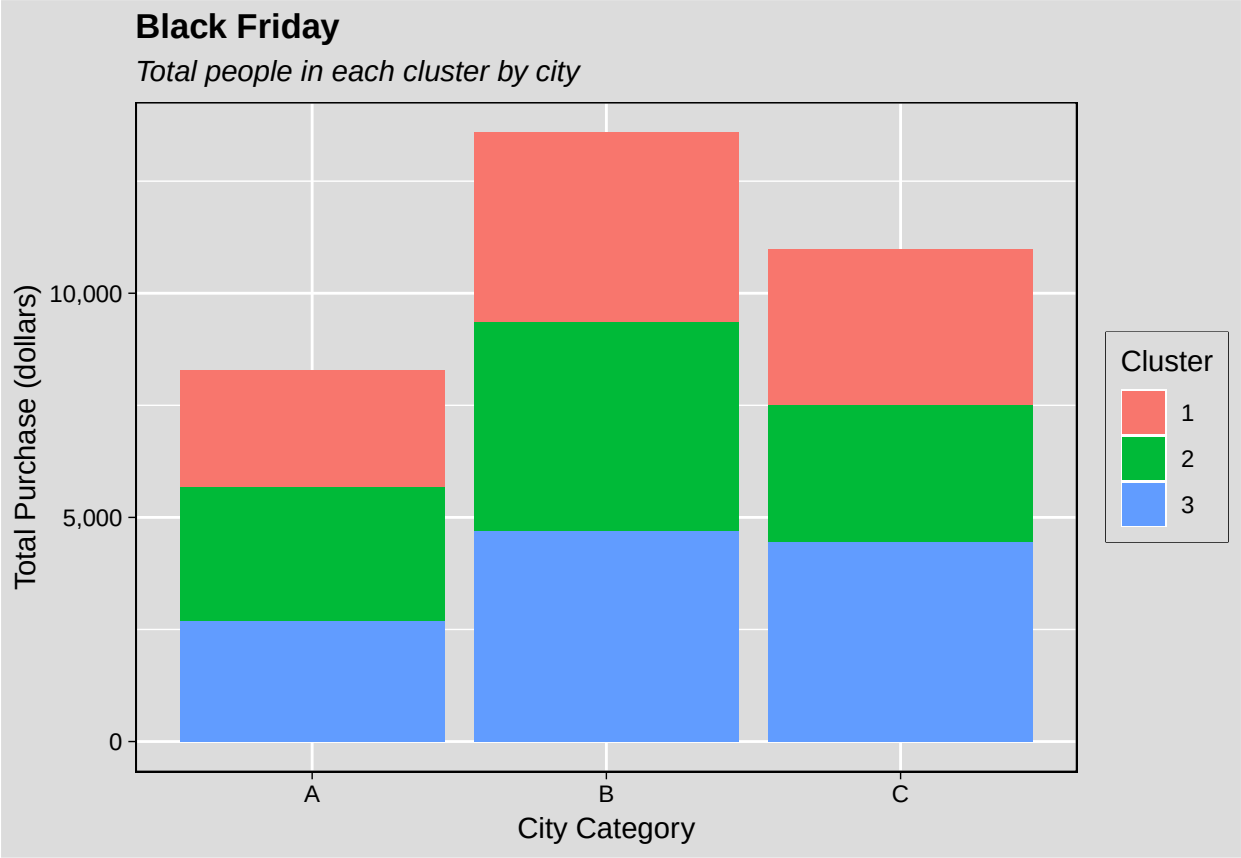
Resumen de los parámetros del cluster

```
BF_Clust_Res <- BF_Cluster %>%
                          group_by(cluster) %>%
                          dplyr::summarize(min_purchase = min(Purchase),
                                   max_purchase = max(Purchase),
                                   avg_purchase = round(mean(Purchase),0))
```

Clasificación de los compradores en función de las ciudades

```
BF_Cluster %>%
group_by(City_Category, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=City_Category, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
```

```
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'City Category', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```

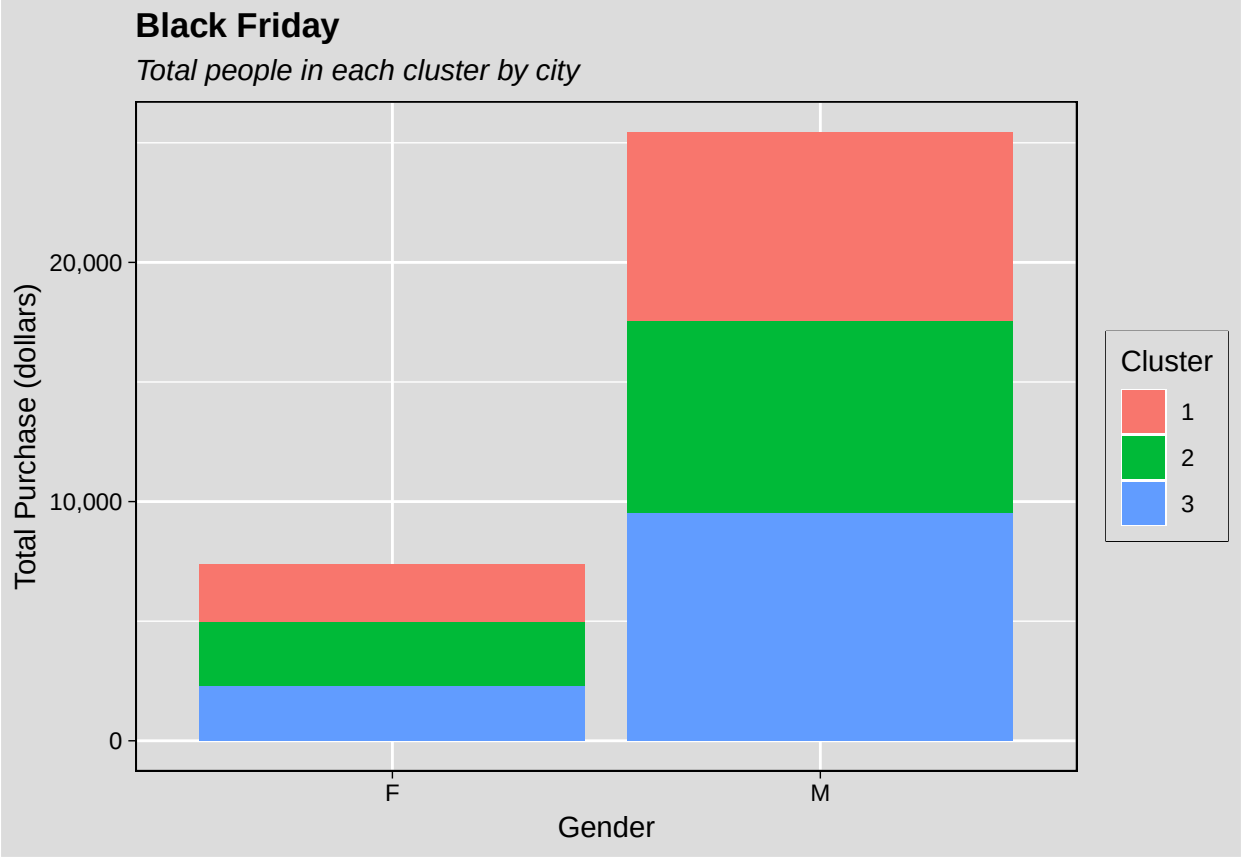


A diferencia del anterior modelo, vemos que los 3 clusters se reparten proporcionalment entre las 3 ciudades. Además se observa esta proporción de reparto similar entre las ciudades

Clasificación de los compradores en función del género

```
BF_Cluster %>%
group_by(Gender, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=Gender, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
```

```
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'Gender', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```

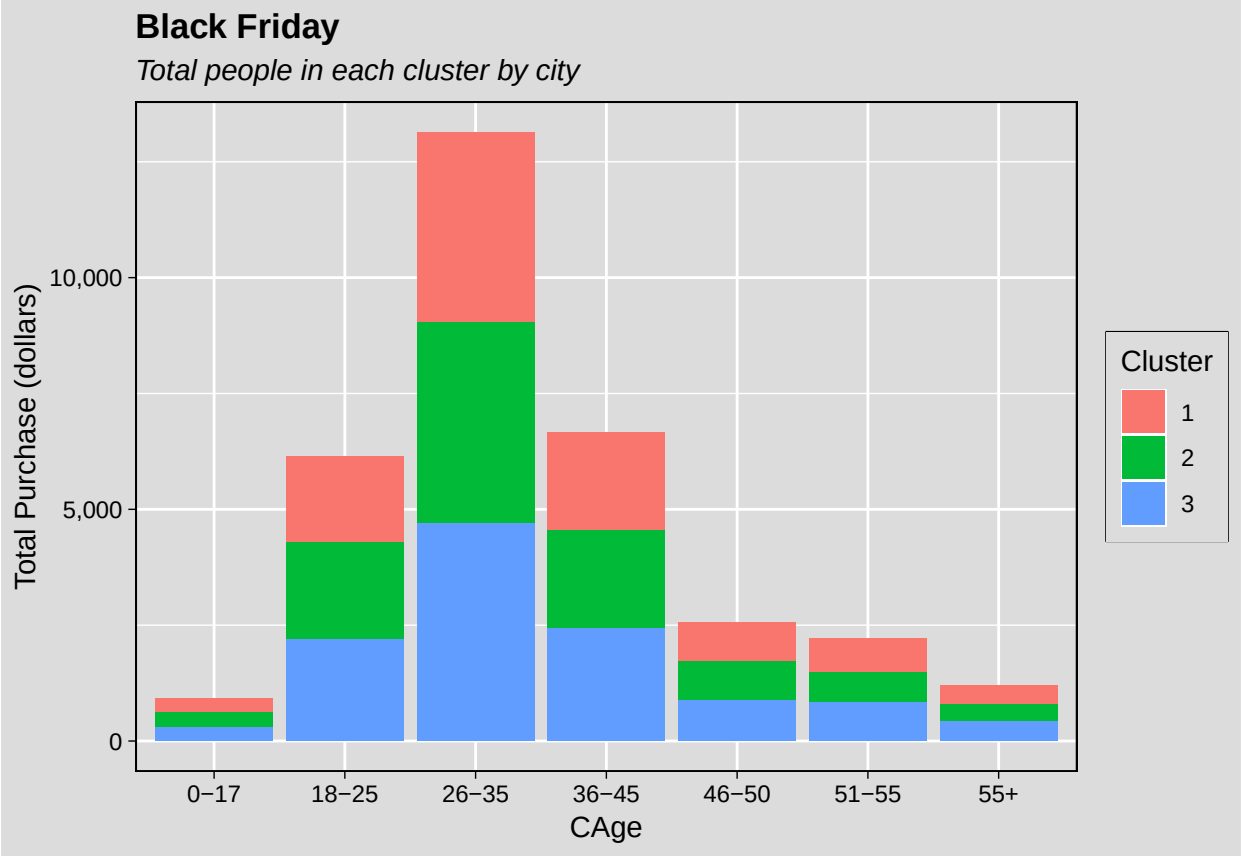


Vemos que los 3 clusters son proporcionales en ambos géneros y no se ve una diferenciación en cuanto a la proporción en cada género

Clasificación de los compradores en función de los tramos de edad

```
BF_Cluster %>%
group_by(Age, cluster) %>%
dplyr::summarize(n = n()) %>%
ggplot(aes(x=Age, y = n)) +
geom_col(aes(fill = as.factor(cluster))) +
theme_linedraw() +
theme(legend.box.background = element_rect(colour = "black"),
      legend.background = element_rect(fill = "gainsboro"),
      panel.background = element_rect(fill = "gainsboro", colour = "white", size = 0.5, linetype = "soli
```

```
      plot.background = element_rect(fill = "gainsboro"), #theme panel settings
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"), #theme panel s
      panel.grid.minor = element_line(size = 0.25, linetype = 'solid', colour = "white"), #theme panel
      plot.title = element_text(hjust = 0, face = 'bold',color = 'black'), #title settings
      plot.subtitle = element_text(face = "italic")) + #subtitle settings
labs(x = 'CAge', y = 'Total Purchase (dollars)', title = "Black Friday", #name title and axis
     subtitle = "Total people in each cluster by city") + #name subtitle
guides(fill=guide_legend(title = "Cluster")) + #remove color legend
scale_y_continuous(labels = scales::comma) #prevent scientific number in x-axis
```



Vemos que los 3 clusters son proporcionales en todos los tramos de edad y no se ve una diferencia en cuanto a esta proporción entre dichos tramos.

Aprendizaje Supervisado

Regresión lineal (MCO):

Modelo con muestra de entrenamiento

Predicción del valor de compra en función de las siguientes variables explicativas

```
lm.fit1 = lm(Purchase~Gender+Age+Occupation+Stay_In_Current_City_Years+Marital_Status+Product_Category_
```

```
summary(lm.fit1)
```

```
##
```

```
## Call:
## lm(formula = Purchase ~ Gender + Age + Occupation + Stay_In_Current_City_Years +
##     Marital_Status + Product_Category_1 + Product_Category_2 +
##     Product_Category_3, data = training.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10849.7  -2858.2   -463.3   2729.5  19631.4
##
## Coefficients:
##                               Estimate Std. Error  t value
## (Intercept)                  11912.487    107.632  110.678
## GenderM                        308.351     35.814    8.610
## Age18-25                       347.965     93.837    3.708
## Age26-35                       432.666     91.139    4.747
## Age36-45                       636.877     93.968    6.778
## Age46-50                       688.579    104.146    6.612
## Age51-55                      1062.229    106.910    9.936
## Age55+                        1018.211    119.740    8.504
## Occupation                       8.371      2.312    3.621
## Stay_In_Current_City_Years1    133.659     47.599    2.808
## Stay_In_Current_City_Years2    190.649     52.823    3.609
## Stay_In_Current_City_Years3    106.925     53.716    1.991
## Stay_In_Current_City_Years4+   183.839     55.404    3.318
## Marital_Status                 -13.561     32.167   -0.422
## Product_Category_1            -836.531      6.660 -125.602
## Product_Category_2              27.570      4.399    6.267
## Product_Category_3              70.342      4.275   16.455
##                                        Pr(>|t|)
## (Intercept)                  < 0.0000000000000002 ***
## GenderM                      < 0.0000000000000002 ***
## Age18-25                                0.000209 ***
## Age26-35                     0.0000020640825 ***
## Age36-45                     0.0000000000123 ***
## Age46-50                     0.0000000000382 ***
## Age51-55                     < 0.0000000000000002 ***
## Age55+                       < 0.0000000000000002 ***
## Occupation                              0.000293 ***
## Stay_In_Current_City_Years1             0.004986 **
## Stay_In_Current_City_Years2             0.000307 ***
## Stay_In_Current_City_Years3             0.046531 *
## Stay_In_Current_City_Years4+            0.000907 ***
## Marital_Status                          0.673344
## Product_Category_1           < 0.0000000000000002 ***
## Product_Category_2                0.0000000003689 ***
## Product_Category_3           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4639 on 98551 degrees of freedom
## Multiple R-squared:  0.1665, Adjusted R-squared:  0.1664
## F-statistic:  1231 on 16 and 98551 DF,  p-value: < 0.00000000000000022
```

Probamos la hipótesis conjunta con el valor de F Ho: ß1 = ß2 = ….. = ßk = 0 Ha: ß1 = ß2 = ….. = ßk

<> 0 para n = 98568, k = 16 df = n-k-1 = 98568 - 16 - 1 = 98551 Dado que el valor F = 1231 y el p-value, podemos concluir que las variables son conjuntamente muy significativas y por tanto se puede rechazar la Ho. Asimismo, se desprende por el coeficiente R2, que el conjunto de las variables dependientes explican un 16,65 % de la variación de la variable dependiente (return) Sin embargo, a nivel individual, casi todas las variables son significativas y la estancia de 3 años en la misma ciudad en menor medida. Sin embargo, el estado civil no es significativo.

Predicción con muestra de validación

El modelo final lo ejecutamos con toda la data

```
# The final model will actually use all data, except test
mix.set <- rbind(training.set, validation.set)
```

```
dim(training.set)
```

```
## [1] 98568    12
```

```
dim(validation.set)
```

```
## [1] 32854    12
```

```
dim(testing.set)
```

```
## [1] 32856    12
```

```
dim(mix.set)
```

```
## [1] 131422     12
```

```
lm.pred1 = predict(lm.fit1, newdata = validation.set)
```

Modelo con muestra de validación

Utilizamos toda la muestra excluyendo la de prueba

```
lm.fit2 = lm(Purchase~Gender+Age+Occupation+Stay_In_Current_City_Years+Marital_Status+Product_Category_
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = Purchase ~ Gender + Age + Occupation + Stay_In_Current_City_Years +
##     Marital_Status + Product_Category_1 + Product_Category_2 +
##     Product_Category_3, data = mix.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10836.6  -2860.4   -477.8   2728.1  19627.7
##
## Coefficients:
##                               Estimate Std. Error  t value
## (Intercept)                  12008.994     92.744  129.486
## GenderM                        320.752     31.048   10.331
## Age18-25                       288.656     80.673    3.578
## Age26-35                       367.156     78.289    4.690
## Age36-45                       554.610     80.788    6.865
## Age46-50                       638.807     89.699    7.122
## Age51-55                      1006.299     92.028   10.935
```

```
## Age55+                          971.507    102.982    9.434
## Occupation                        7.424      2.003    3.707
## Stay_In_Current_City_Years1      99.890     41.223    2.423
## Stay_In_Current_City_Years2     194.607     45.739    4.255
## Stay_In_Current_City_Years3      68.630     46.469    1.477
## Stay_In_Current_City_Years4+    147.968     48.025    3.081
## Marital_Status                  -16.432     27.861   -0.590
## Product_Category_1             -835.377      5.770 -144.772
## Product_Category_2               27.416      3.808    7.200
## Product_Category_3               69.289      3.704   18.708
##                                          Pr(>|t|)
## (Intercept)                   < 0.0000000000000002 ***
## GenderM                       < 0.0000000000000002 ***
## Age18-25                                  0.000346 ***
## Age26-35                        0.000002738122998 ***
## Age36-45                        0.000000000006679 ***
## Age46-50                        0.000000000001072 ***
## Age51-55                      < 0.0000000000000002 ***
## Age55+                        < 0.0000000000000002 ***
## Occupation                                0.000210 ***
## Stay_In_Current_City_Years1               0.015386 *
## Stay_In_Current_City_Years2     0.000020943024670 ***
## Stay_In_Current_City_Years3               0.139700
## Stay_In_Current_City_Years4+              0.002063 **
## Marital_Status                            0.555333
## Product_Category_1            < 0.0000000000000002 ***
## Product_Category_2              0.000000000000607 ***
## Product_Category_3            < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4639 on 131405 degrees of freedom
## Multiple R-squared:  0.1664, Adjusted R-squared:  0.1663
## F-statistic:  1639 on 16 and 131405 DF,  p-value: < 0.00000000000000022
```

Probamos la hipótesis conjunta con el valor de F Ho: ß1 = ß2 = ….. = ßk = 0 Ha: ß1 = ß2 = ….. = ßk <> 0 para n = 131422, k = 16 df = n-k-1 = 131422 - 16 - 1 = 131405 Dado que el valor F = 1231 y el p-value, podemos concluir que las variables son conjuntamente muy significativas y por tanto se puede rechazar la Ho. Asimismo, se desprende por el coeficiente R2, que el conjunto de las variables dependientes explican un 16,64 % de la variación de la variable dependiente (return) Sin embargo, a nivel individual, casi todas las variables son significativas. Pero la estancia de 1 año en la misma ciudad en menor medida. Sin embargo, tanto el estado civil como la estancia de 3 año en la misma ciudad es significativo.

Predicción con la muestra de prueba

```
lm.pred2 = predict(lm.fit2, newdata = testing.set)
```

Estimamos ECM de la muestras de validación y de prueba

```
error.val1 <- mean((validation.set[, "Purchase"] - lm.pred1)^2)

error.test1 <- mean((testing.set[, "Purchase"] - lm.pred2)^2)
```

- el errror de validación es 21521011.6592281

- el errror de prueba es 21652790.6669036

Vemos que la diferencia entre el error de validación y el de prueba es mínimo.