

# Lesson 4: Annotating and Saving

## Background for this activity

En esta actividad, revisará un escenario y practicará cómo agregar anotaciones a una visualización de datos con ggplot2. También aprenderá cómo guardar imágenes de visualizaciones de ggplot2 para poder agregarlas directamente a las presentaciones.

A lo largo de esta actividad, también tendrá la oportunidad de practicar la escritura de su propio código realizando usted mismo cambios en los fragmentos de código. Si encuentra un error o se queda atascado, siempre puede consultar el archivo Lesson4\_Annotations\_Solutions .rmd en la carpeta Soluciones en la Semana 4 para obtener el código completo y correcto.

## The Scenario

Como analista de datos junior para una empresa de reservas de hoteles, ha estado creando visualizaciones en “R” con el paquete “ggplot2” para compartir información sobre sus datos con las partes interesadas. Después de crear una serie de visualizaciones usando la estética y los filtros ggplot(), ggplot2, la parte interesada le pide que agregue anotaciones a sus visualizaciones para ayudar a explicar sus hallazgos en una presentación. Afortunadamente, ggplot2 tiene funciones de anotación integradas.

## Step 1: Import your data

Si no ha salido de RStudio desde la última vez que importó estos datos, puede omitir estos pasos. Sin embargo, volver a ejecutar estos fragmentos de código no afectará a tu consola si quieres ejecutarlos por si acaso.

Si esta línea causa un error, cópiela en la línea setwd(“/cloud/project/Course 7/Week 4”) anterior.

Ejecute el siguiente código para leer el archivo ‘hotel\_bookings.csv’ en un marco de datos:

```
hotel_bookings <- read.csv("hotel_bookings.csv")
```

## Step 2: Refresh Your Memory

A estas alturas ya estás bastante familiarizado con este conjunto de datos. Pero puedes refrescar tu memoria con las funciones head() y colnames(). Ejecute dos fragmentos de código a continuación para obtener una muestra de los datos y también obtener una vista previa de todos los nombres de las columnas:

```
head(hotel_bookings)
```

```

##      hotel is_canceled lead_time arrival_date_year arrival_date_month
## 1 Resort Hotel      0      342      2015      July
## 2 Resort Hotel      0      737      2015      July
## 3 Resort Hotel      0       7      2015      July
## 4 Resort Hotel      0      13      2015      July
## 5 Resort Hotel      0      14      2015      July
## 6 Resort Hotel      0      14      2015      July
## arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
## 1      27      1      0
## 2      27      1      0
## 3      27      1      0
## 4      27      1      0
## 5      27      1      0
## 6      27      1      0
## stays_in_week_nights adults children babies meal country market_segment
## 1      0      2      0      0 BB PRT Direct
## 2      0      2      0      0 BB PRT Direct
## 3      1      1      0      0 BB GBR Direct
## 4      1      1      0      0 BB GBR Corporate
## 5      2      2      0      0 BB GBR Online TA
## 6      2      2      0      0 BB GBR Online TA
## distribution_channel is_repeated_guest previous_cancellations
## 1      Direct      0      0
## 2      Direct      0      0
## 3      Direct      0      0
## 4      Corporate      0      0
## 5      TA/TO      0      0
## 6      TA/TO      0      0
## previous_bookings_not_canceled reserved_room_type assigned_room_type
## 1      0      C      C
## 2      0      C      C
## 3      0      A      C
## 4      0      A      A
## 5      0      A      A
## 6      0      A      A
## booking_changes deposit_type agent company days_in_waiting_list customer_type
## 1      3 No Deposit NULL NULL 0 Transient
## 2      4 No Deposit NULL NULL 0 Transient
## 3      0 No Deposit NULL NULL 0 Transient
## 4      0 No Deposit 304 NULL 0 Transient
## 5      0 No Deposit 240 NULL 0 Transient
## 6      0 No Deposit 240 NULL 0 Transient
## adr required_car_parking_spaces total_of_special_requests reservation_status
## 1 0 0 0 Check-Out
## 2 0 0 0 Check-Out
## 3 75 0 0 Check-Out
## 4 75 0 0 Check-Out
## 5 98 0 1 Check-Out
## 6 98 0 1 Check-Out
## reservation_status_date
## 1 2015-07-01
## 2 2015-07-01

```

```
## 3      2015-07-02
## 4      2015-07-02
## 5      2015-07-03
## 6      2015-07-03
```

```
colnames(hotel_bookings)
```

```
## [1] "hotel" "is_canceled"
## [3] "lead_time" "arrival_date_year"
## [5] "arrival_date_month" "arrival_date_week_number"
## [7] "arrival_date_day_of_month" "stays_in_weekend_nights"
## [9] "stays_in_week_nights" "adults"
## [11] "children" "babies"
## [13] "meal" "country"
## [15] "market_segment" "distribution_channel"
## [17] "is_repeated_guest" "previous_cancellations"
## [19] "previous_bookings_not_canceled" "reserved_room_type"
## [21] "assigned_room_type" "booking_changes"
## [23] "deposit_type" "agent"
## [25] "company" "days_in_waiting_list"
## [27] "customer_type" "adr"
## [29] "required_car_parking_spaces" "total_of_special_requests"
## [31] "reservation_status" "reservation_status_date"
```

## Step 3: Install and load the ‘ggplot2’ and ‘tidyverse’ packages (optional)

Si aún no ha instalado y cargado el paquete `ggplot2`, deberá hacerlo antes de poder usar la función `ggplot()`. Sólo tienes que hacer esto una vez, no cada vez que llames a `ggplot()`.

También puede omitir este paso si no ha cerrado su cuenta de RStudio desde que realizó la última actividad. Si no está seguro, puede ejecutar el fragmento de código y presionar “cancelar” si aparece un mensaje de advertencia que le indica que ya descargó el paquete “ggplot2”.

Ejecute el siguiente fragmento de código para instalar y cargar `ggplot2`. ¡Esto puede tomar unos pocos minutos!

Si no ha instalado y cargado `tidyverse` en esta sesión de RStudio, puede ejecutar el siguiente fragmento de código. ¡Esto puede tomar unos pocos minutos!

```
install.packages('tidyverse')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.2    ✓ readr      2.1.4
## ✓ forcats   1.0.0    ✓ stringr   1.5.0
## ✓ lubridate 1.9.2    ✓ tibble    3.2.1
## ✓ purrr     1.0.1    ✓ tidyr     1.3.0
```

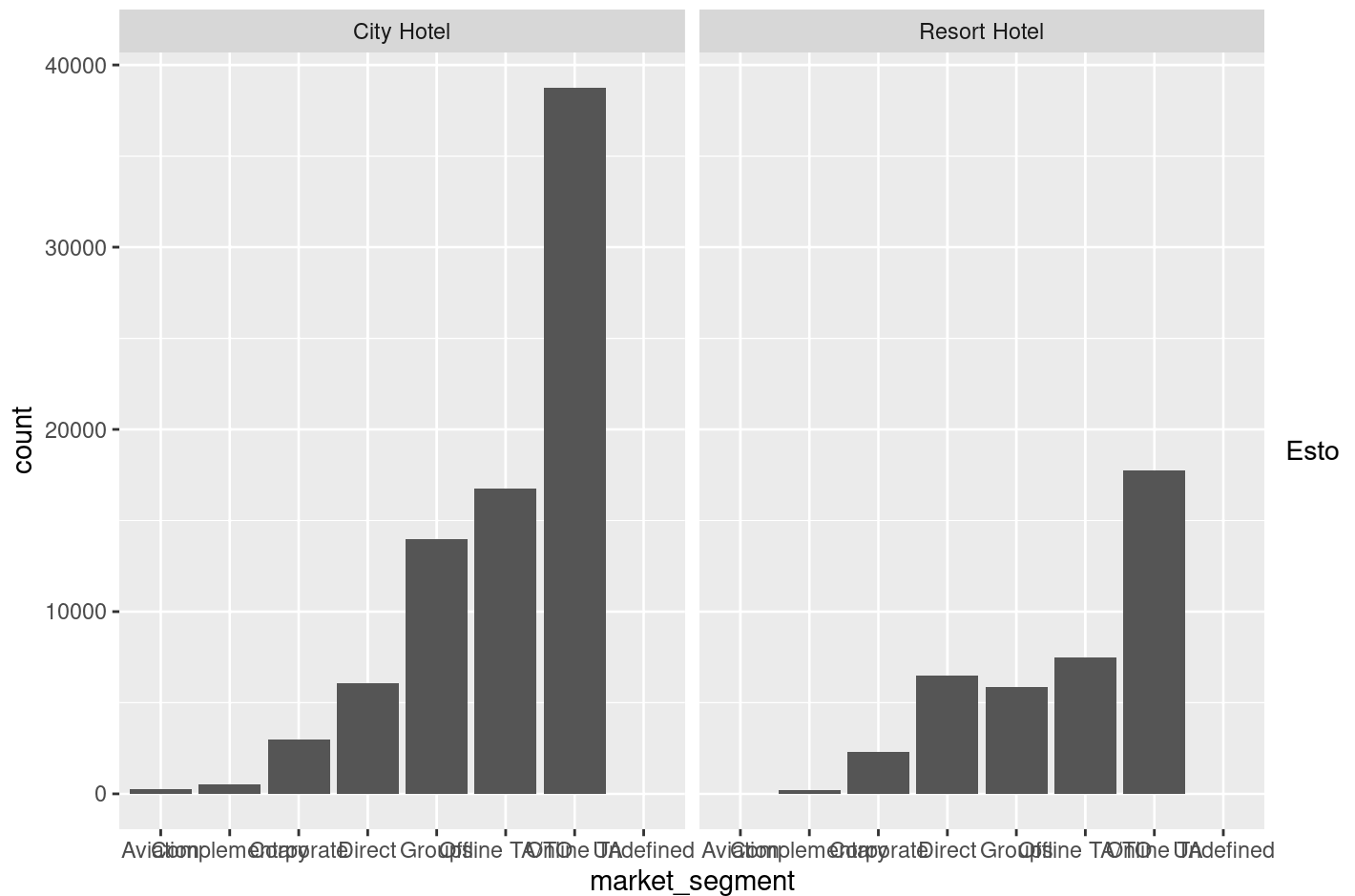
```
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## Step 4: Annotating your chart

Su parte interesada le dice que le gustaría que creara una visualización que compare segmentos de mercado entre hoteles urbanos y hoteles turísticos. Esto ayudará a informar cómo la empresa orienta las promociones en el futuro. Le piden que cree una versión limpia y etiquetada y que la guarde como un archivo .png para poder incluirla en una presentación.

Como repaso, aquí hay un cuadro similar al que creó en una actividad anterior:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel)
```

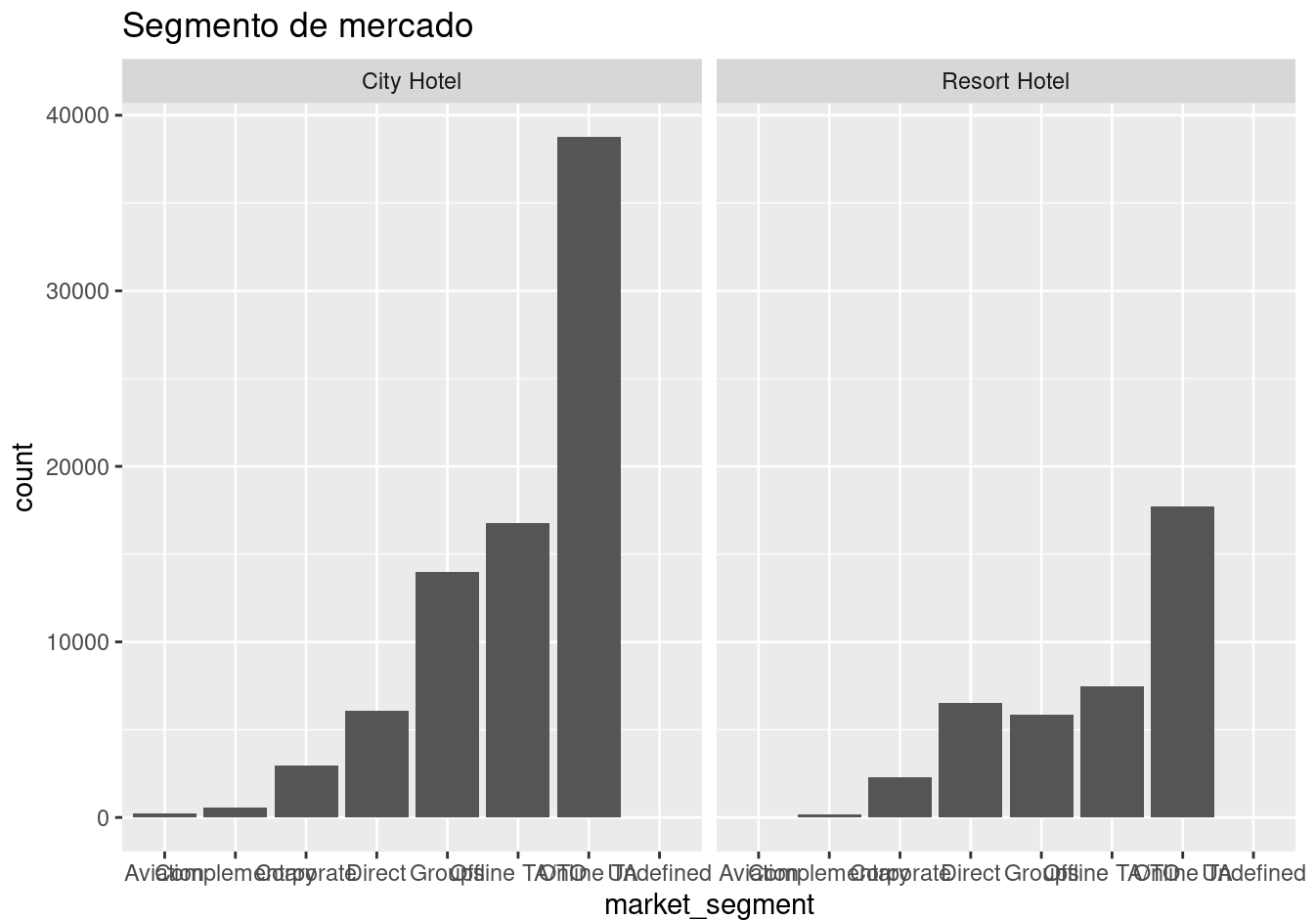


crea dos gráficos de barras: uno para los datos de 'city\_hotel' y otro para los datos de 'resort\_hotel'. El eje x es 'market\_segment' y el eje y es 'count' para ambos gráficos.

En esta visualización no está claro de dónde provienen los datos, cuál es la principal conclusión o incluso qué muestran los datos. Para explicar todo eso, puedes aprovechar las anotaciones en `ggplot2`.

El primer paso será agregar un título; Esto suele ser lo primero a lo que las personas prestarán atención cuando se encuentren con una visualización de datos por primera vez. Para agregar un título, agregará `labs()` al final de su comando `ggplot()` y luego ingresará un título allí. Agregue un título descriptivo al siguiente fragmento de código:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Segmento de mercado")
```



Este fragmento de código generará el mismo gráfico que antes, pero ahora incluye un título para explicar la visualización de datos más claramente a su audiencia.

También desea agregar otro detalle sobre el período de tiempo que cubren estos datos. Para hacer esto, necesita saber de dónde provienen los datos.

Te das cuenta de que puedes usar la función `min()` en la columna de año en los datos:

```
min(hotel_bookings$arrival_date_year)
```

```
## [1] 2015
```

And the `max()` function:

```
max(hotel_bookings$arrival_date_year)
```

```
## [1] 2017
```

Pero necesitarás guardarlas como variables para poder usarlas fácilmente en tu etiquetado; el siguiente fragmento de código crea dos de esas variables:

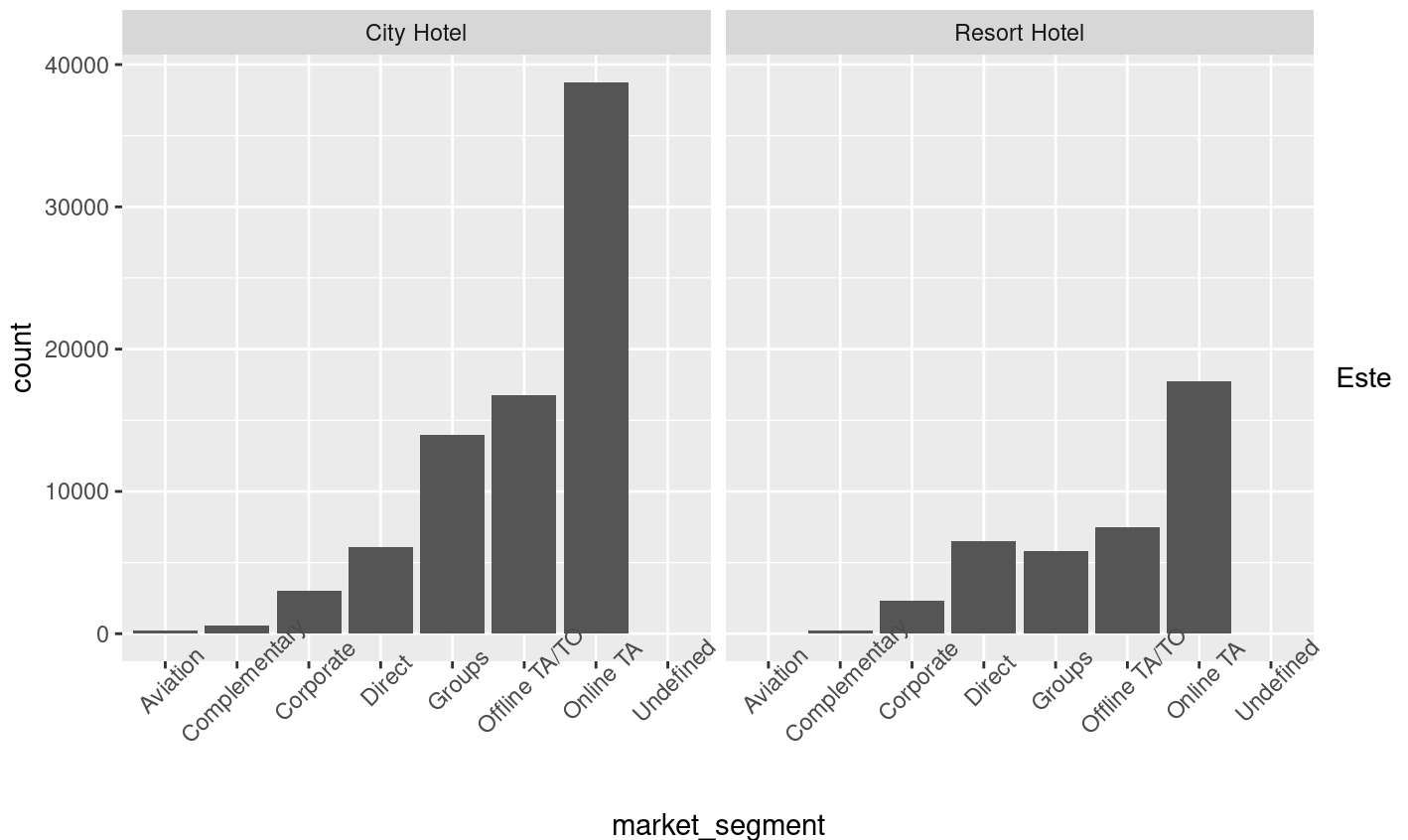
```
mindate <- min(hotel_bookings$arrival_date_year)
maxdate <- max(hotel_bookings$arrival_date_year)
```

Ahora, agregará un subtítulo usando `subtitle=` en la función `labs()`. Luego, puede usar la función `paste0()` para usar las variables recién creadas en sus etiquetas. Esto es realmente útil, porque si los datos se actualizan y se agregan datos más recientes, no es necesario cambiar el código siguiente porque las variables son dinámicas:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       subtitle=paste0("Data from: ", mindate, " to ", maxdate))
```

## Comparison of market segments by hotel type for hotel bookings

Data from: 2015 to 2017

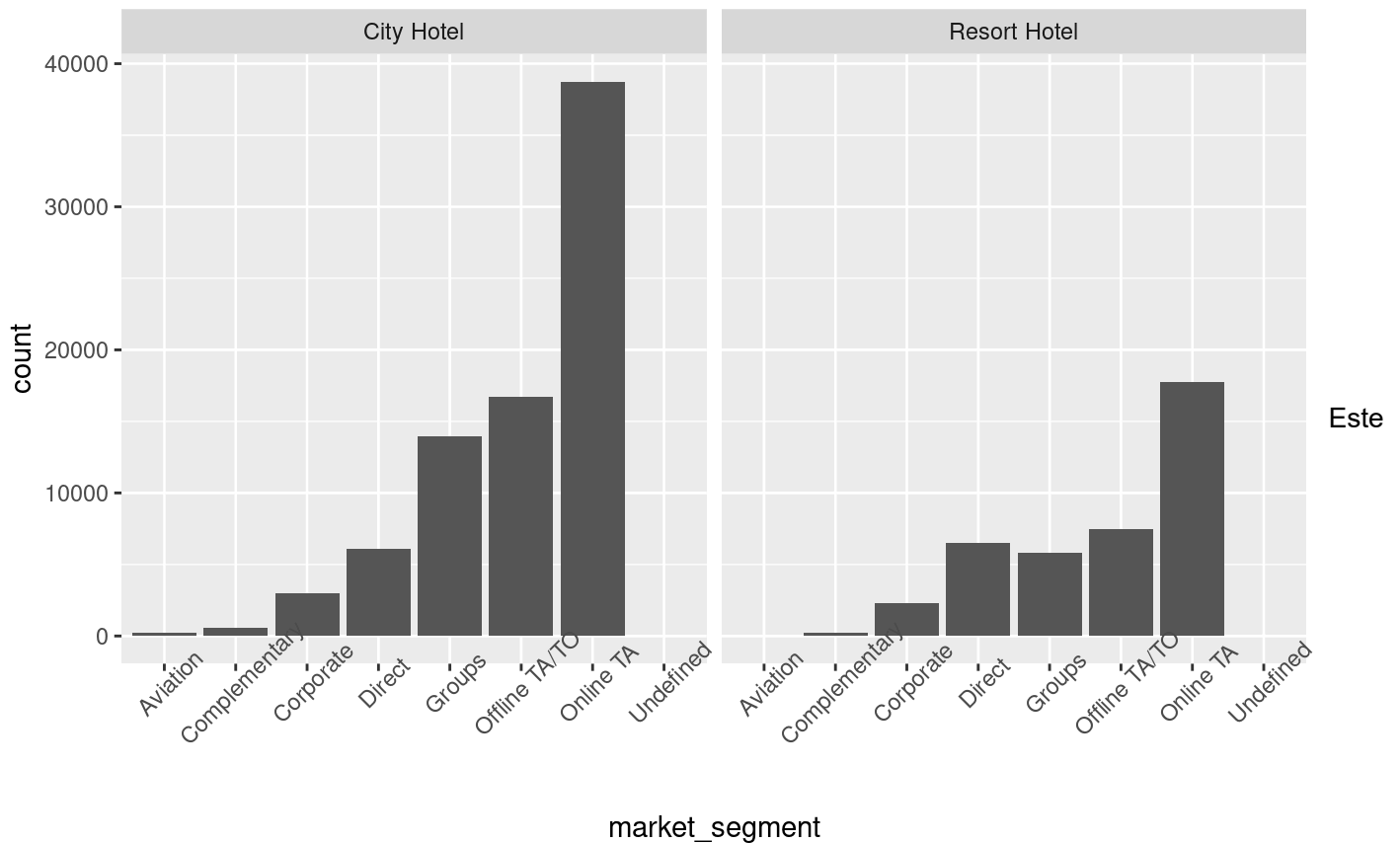


fragmento de código agregará el subtítulo “Datos de: 2015 a 2017” debajo del título que agregó anteriormente al gráfico.

Te das cuenta de que este gráfico muestra los detalles técnicos de forma demasiado destacada. No querrás que eso sea lo segundo que la gente note durante la presentación. Decide cambiar el “subtítulo” por un “título” que aparecerá en la esquina inferior derecha.

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       caption=paste0("Data from: ", mindate, " to ", maxdate))
```

## Comparison of market segments by hotel type for hotel bookings



Data from: 2015 to 2017

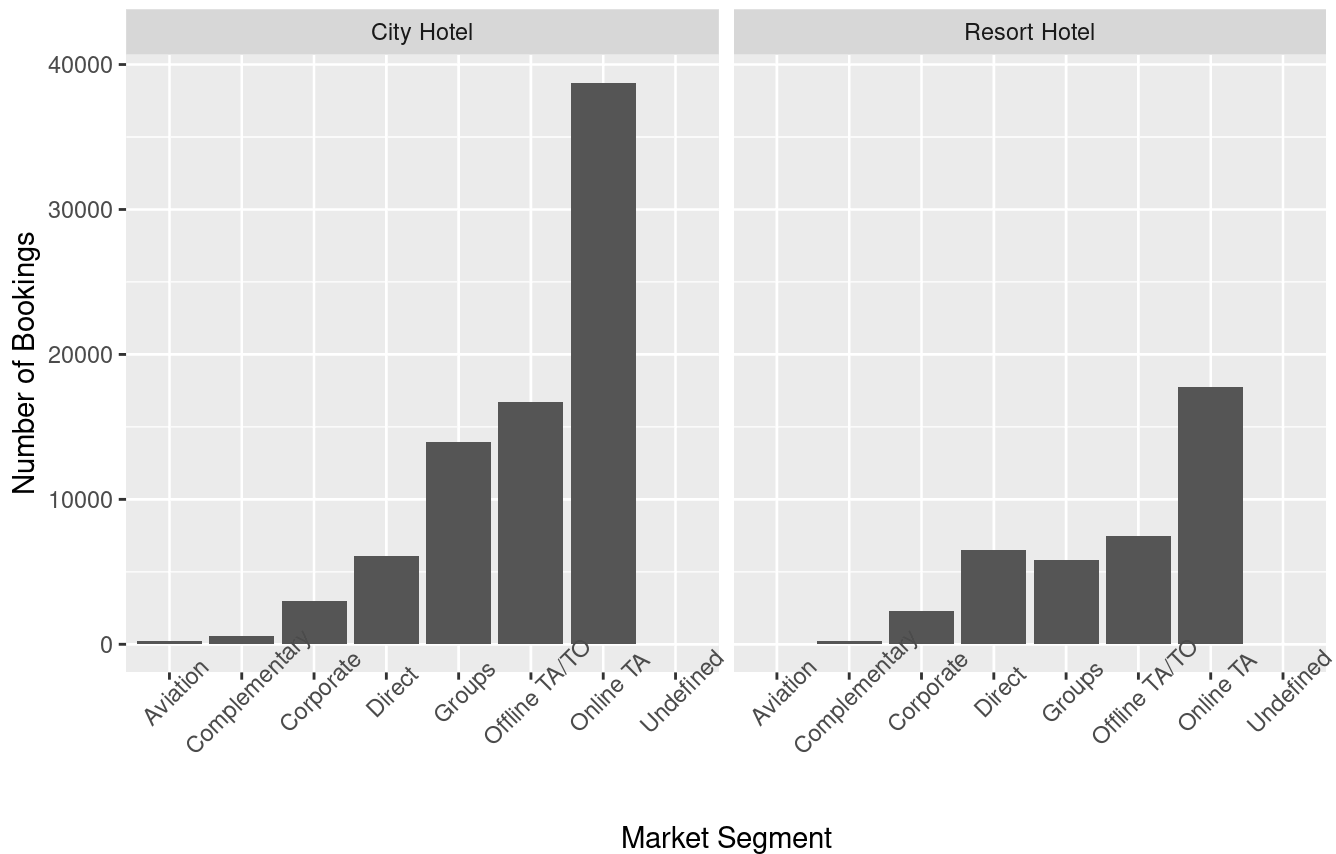
fragmento de código realiza un ligero cambio en la visualización que creó en el último fragmento; ahora el subtítulo “datos de: 2015 a 2017” está en la esquina inferior derecha.

Ahora desea limpiar las etiquetas de los ejes x y para asegurarse de que estén realmente claras. Para hacer eso, puede agregar a la función `labs()` y usar `x=` e `y=`. Siéntete libre de cambiar el texto de la etiqueta y jugar con él:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       caption=paste0("Data from: ", mindate, " to ", maxdate),
       x="Market Segment",
       y="Number of Bookings")
```



## Comparison of market segments by hotel type for hotel bookings



Data from: 2015 to 2017

Ahora tiene la visualización de datos anterior, pero ahora las etiquetas de los ejes x e y se han cambiado de 'segmento\_mercado' y 'recuento' a 'Segmento de mercado' y 'Número de reservas' para que el gráfico sea más claro.

## Step 5: Saving your chart

Ahora es el momento de guardar lo que acaba de crear para poder compartirlo fácilmente con las partes interesadas.

¡Puedes usar la función `ggsave()` para hacer justamente eso! Guardará su imagen como 7x7 en la ruta del archivo que ingrese de forma predeterminada, lo que simplifica la exportación de sus gráficos desde R.

La función `ggsave()` en el fragmento de código siguiente guardará el último gráfico que se generó, por lo que si ejecutó algo después de ejecutar el fragmento de código anterior, ejecute ese fragmento de código nuevamente.

Luego, ejecute el siguiente fragmento de código para guardar ese gráfico como un archivo .png llamado `hotel_booking_chart`, que deja claro a las partes interesadas qué contiene el archivo .png. Ahora debería poder encontrar este archivo en la pestaña 'Archivos' en la parte inferior derecha de la pantalla. ¡Échale un vistazo!

```
ggsave('hotel_booking_chart.png', width=7,
      height=7)
```

# Practice quiz

¿Con qué dimensiones guardaste esta imagen usando `ggsave()` ? Anota tu respuesta y responde en la plataforma Coursera.

R: 5x5 B: 10x10 C: 7x7 Tamaño: 25x25

Si desea hacer su gráfico más grande y más rectangular para que se ajuste a la presentación de diapositivas, puede especificar la altura y el ancho de su `.png` en el comando `ggsave()` . Edite el siguiente fragmento de código para crear una imagen `.png` de 16x8:

```
ggsave('hotel_booking_chart.png',  
      width=9,  
      height=5)
```

## Activity Wrap Up

Ahora que ha terminado de crear y exportar una visualización de datos con anotaciones en `ggplot2` , puede compartir lo que creó con las partes interesadas clave para brindarles información sobre los hallazgos de sus datos. Estas habilidades le permitirán crear, anotar y compartir sus visualizaciones de datos directamente desde el espacio de su consola “R”. Puede practicar estas habilidades modificando los fragmentos de código en el archivo `rmd` o utilizar este código como punto de partida en su propia consola de proyecto. Continuará aprendiendo más sobre las funciones de `ggplot2` en este curso, pero con las habilidades que ya ha estado practicando, podrá generar gráficos, utilizar funciones estéticas, aplicar filtros y crear anotaciones para explicar sus datos.

Cuando complete esta actividad, guárdela haciendo clic en la pestaña Archivo en la parte superior izquierda de la consola RStudio y luego seleccionando Guardar. Utilizará este documento (o cualquier otra actividad de R que haya completado y guardado en este curso) para una próxima actividad.