

# Lesson 3: Aesthetics and Visualizations

## Background for this activity

En esta actividad, revisará un escenario y continuará aplicando sus conocimientos de visualización de datos con ggplot2. Aprenderá más sobre las características estéticas de las visualizaciones y cómo personalizarlas según criterios específicos.

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 Lesson3\_Aesthetics\_Solutions .rmd en la carpeta Soluciones en la Semana 4 para obtener el código completo y correcto.

## The Scenario

En este ejemplo, usted es un analista de datos junior que trabaja para la misma empresa de reservas de hoteles de antes. La última vez, creó algunas visualizaciones simples con “ggplot2” para brindarles a sus partes interesadas información rápida sobre sus datos. Ahora, está interesado en crear visualizaciones que resalten diferentes aspectos de los datos para presentarlos a sus partes interesadas. Ampliará lo que ya ha aprendido sobre “ggplot2” y creará nuevos tipos de visualizaciones, como gráficos de barras.

## 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.

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

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

```
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’ package (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!

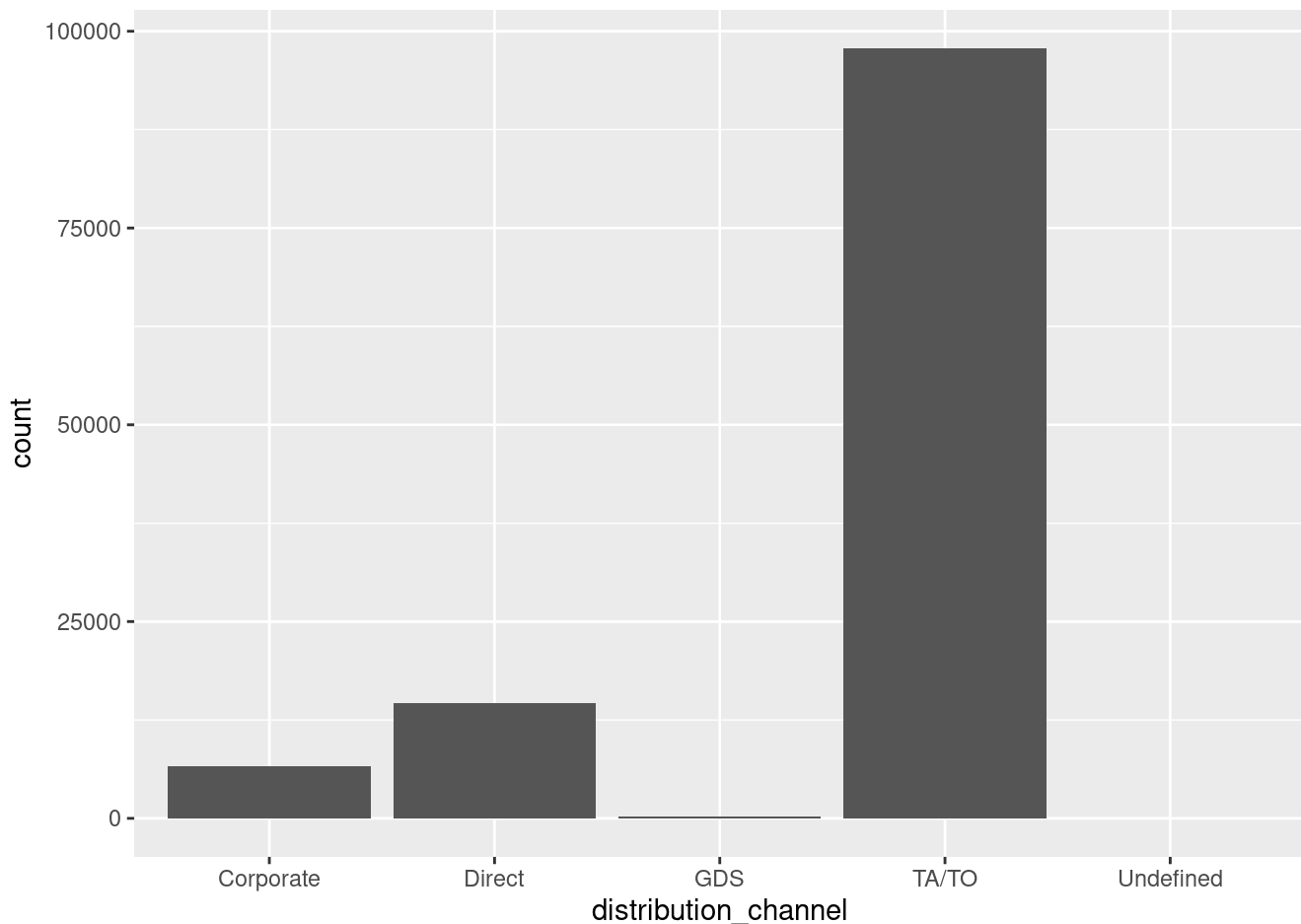
## Step 4: Making a Bar Chart

Su parte interesada está interesada en desarrollar promociones basadas en diferentes distribuciones de reservas, pero primero necesita saber cuántas transacciones se producen para cada tipo de distribución diferente.

Puedes decirle a `ggplot()` qué tipo de gráfico quieres crear usando el argumento `geom_`.

Anteriormente, usaste `geom_point` para hacer un diagrama de dispersión que compara el tiempo de entrega y la cantidad de niños. Ahora, usará `geom_bar` para crear un gráfico de barras en este fragmento de código:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel))
```



Anteriormente, creaste diagramas de dispersión con ggplot2. Este fragmento de código crea un gráfico de barras con 'canal\_distribución' en el eje x y 'recuento' en el eje y. Hay datos para canales de distribución corporativos, directos, GDS, TA/TO y no definidos.

## Practice quiz

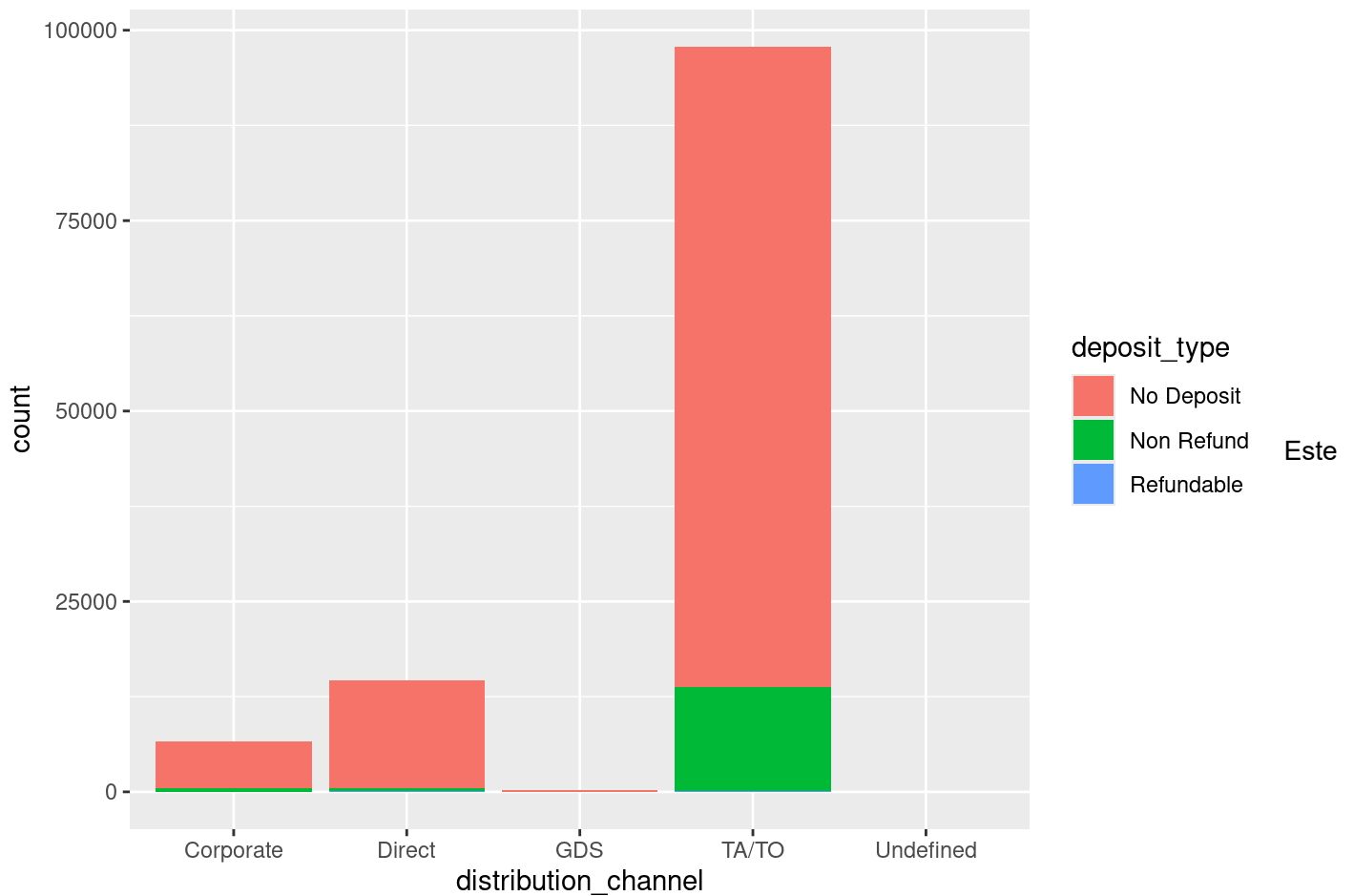
Utilice el gráfico de barras que creó para responder a esta pregunta: ¿qué tipo de distribución tiene la mayor cantidad de reservas? Anota tu respuesta para la pregunta del cuestionario de práctica en Coursera después.

R: TA/A B: directo C: GDS D: Corporativo

## Step 5: Diving deeper into bar charts

Después de explorar su gráfico de barras, su parte interesada tiene más preguntas. Ahora quieren saber si el número de reservas para cada tipo de distribución es diferente en función de si hubo o no depósito o a qué segmento de mercado representan.

Intente modificar el código siguiente para responder la pregunta sobre depósitos agregando 'fill=deposit\_type' después de 'x = Distribution\_channel':



fragmento de código también crea un gráfico de barras con 'canal\_distribución' en el eje x y 'recuento' en el eje y. Pero también incluye datos de la columna 'tipo\_depósito' como secciones codificadas por colores de cada barra. Hay una leyenda que explica qué representa cada color en el lado derecho de la visualización.

Ahora intente agregar 'fill=market\_segment' a este fragmento de código en lugar de 'fill=deposit\_type':

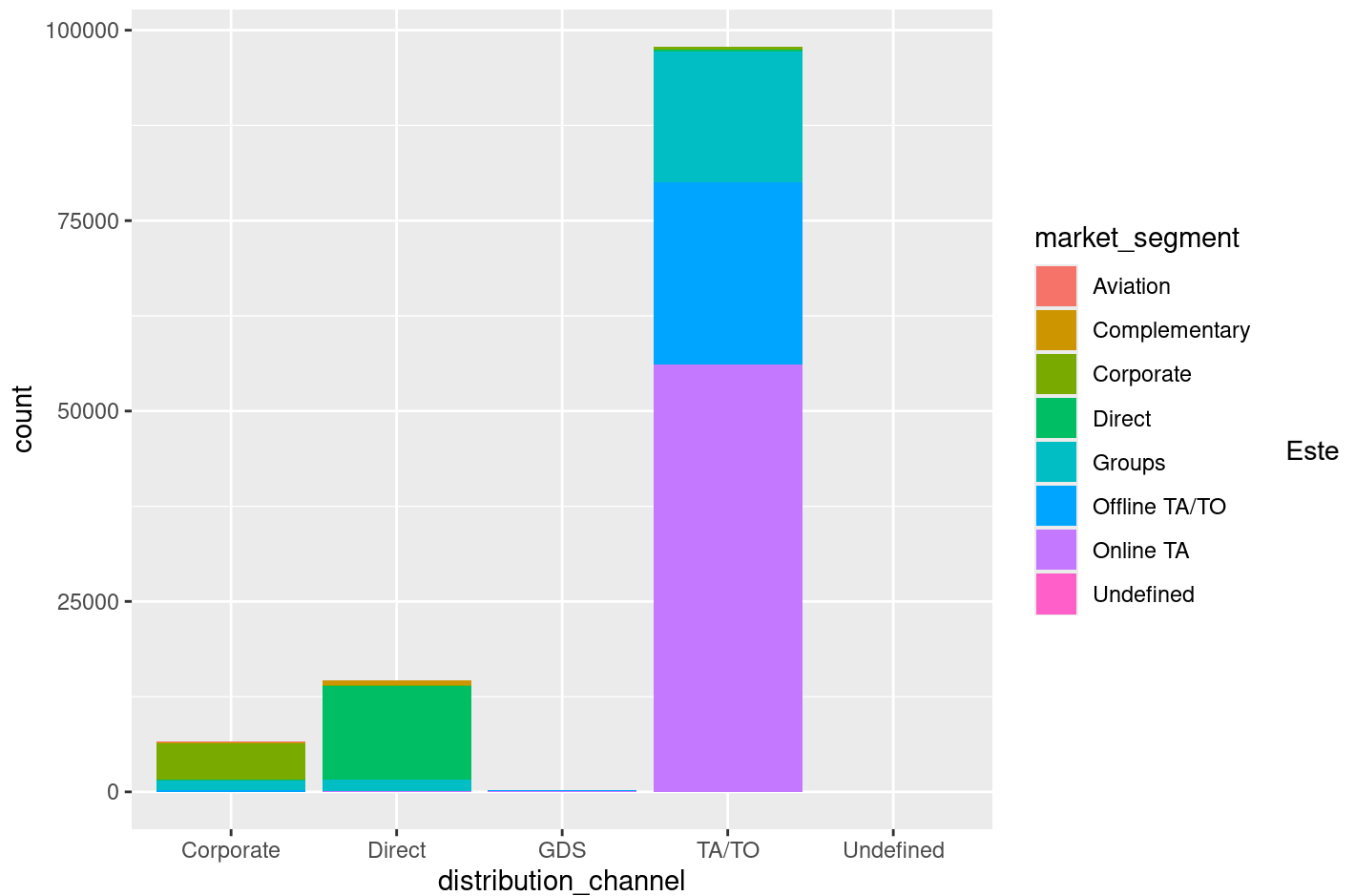


gráfico de barras es similar al gráfico anterior, excepto que los datos del 'segmento\_de\_mercado' se registran en las secciones codificadas por colores de cada barra.

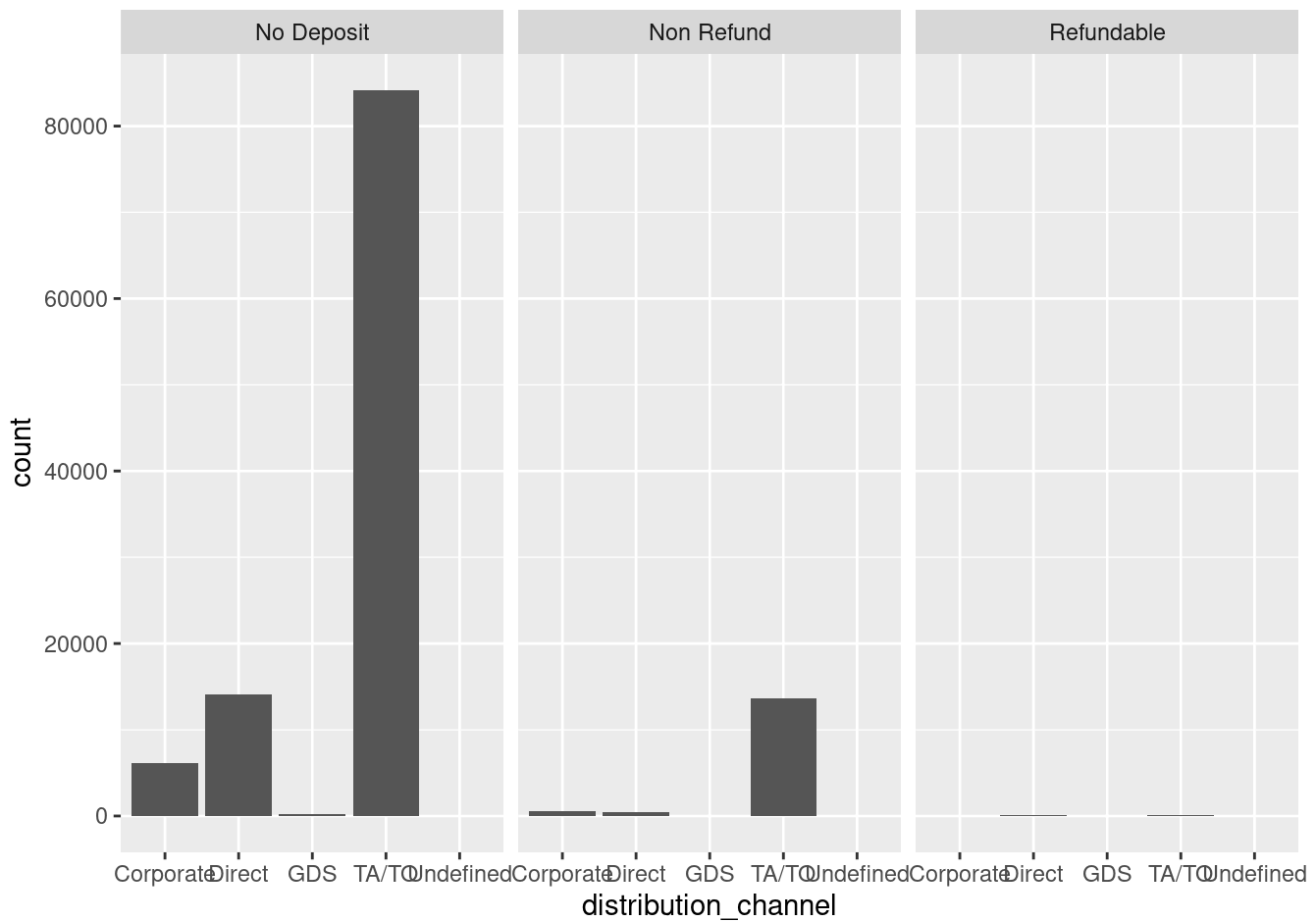
## Step 6: Facets galore

Después de revisar los nuevos gráficos, su parte interesada le pide que cree gráficos separados para cada tipo de depósito y segmento de mercado para ayudarlos a comprender las diferencias con mayor claridad.

Sabes que la función `facet_` puede hacer esto muy rápidamente.

Agregue 'deposit\_type' después del símbolo '~' en el fragmento de código siguiente para crear un gráfico diferente para cada uno.

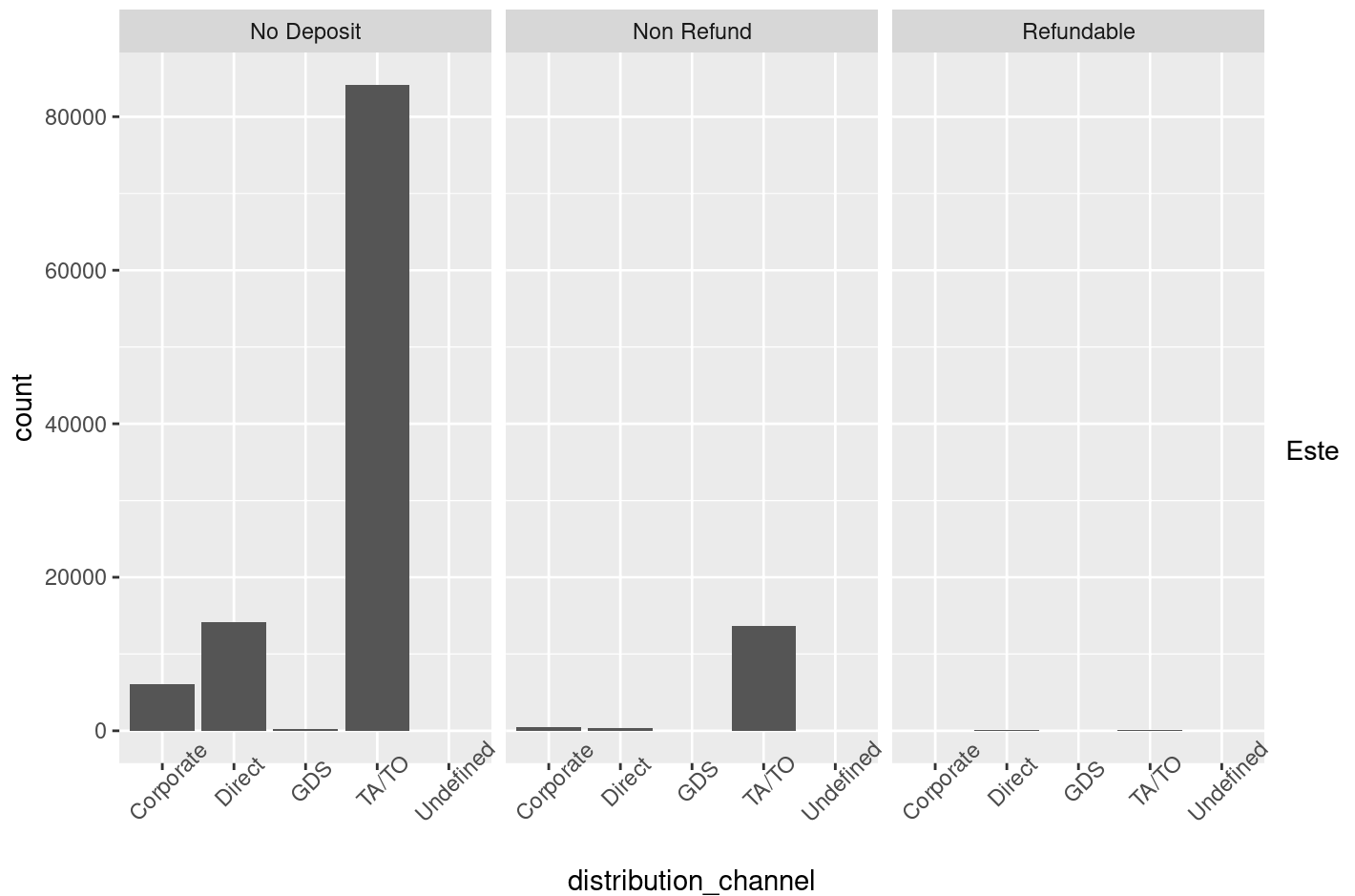
```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~deposit_type )
```



Este fragmento de código crea tres gráficos de barras para los tipos de depósito 'sin depósito', sin reembolso y 'reembolsable'. Observa que aquí es difícil leer las etiquetas del eje x, por lo que agrega un fragmento de código al final que gira el texto a 45 grados para que sea más fácil de leer.

Pruébalo a continuación:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~deposit_type) +
  theme(axis.text.x = element_text(angle = 45))
```

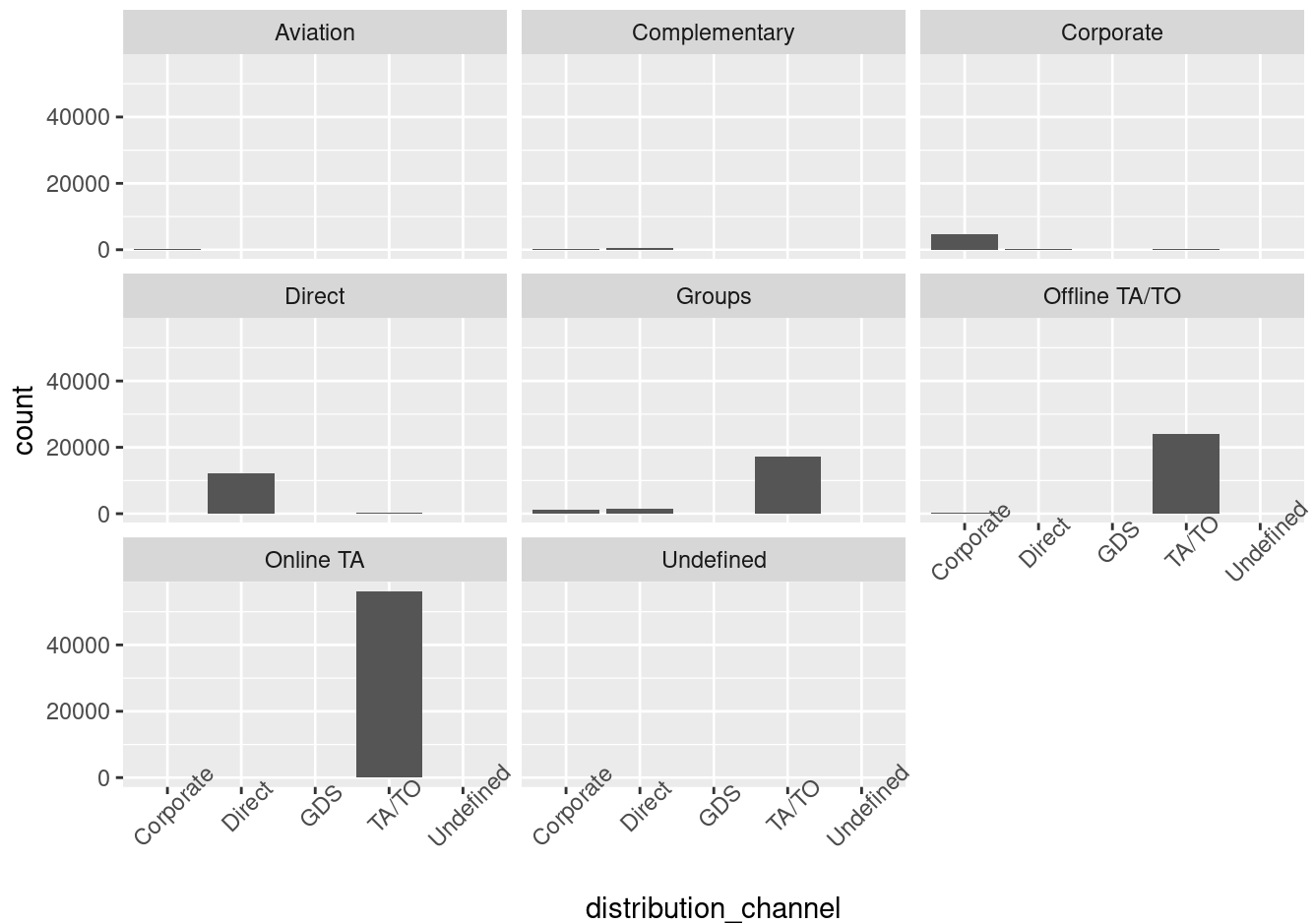


fragmento de código crea un gráfico de barras similar al fragmento anterior, pero ahora las etiquetas en el eje x con los diferentes canales de distribución son más claras.

Puede utilizar la misma sintaxis para crear un gráfico diferente para cada segmento de mercado:

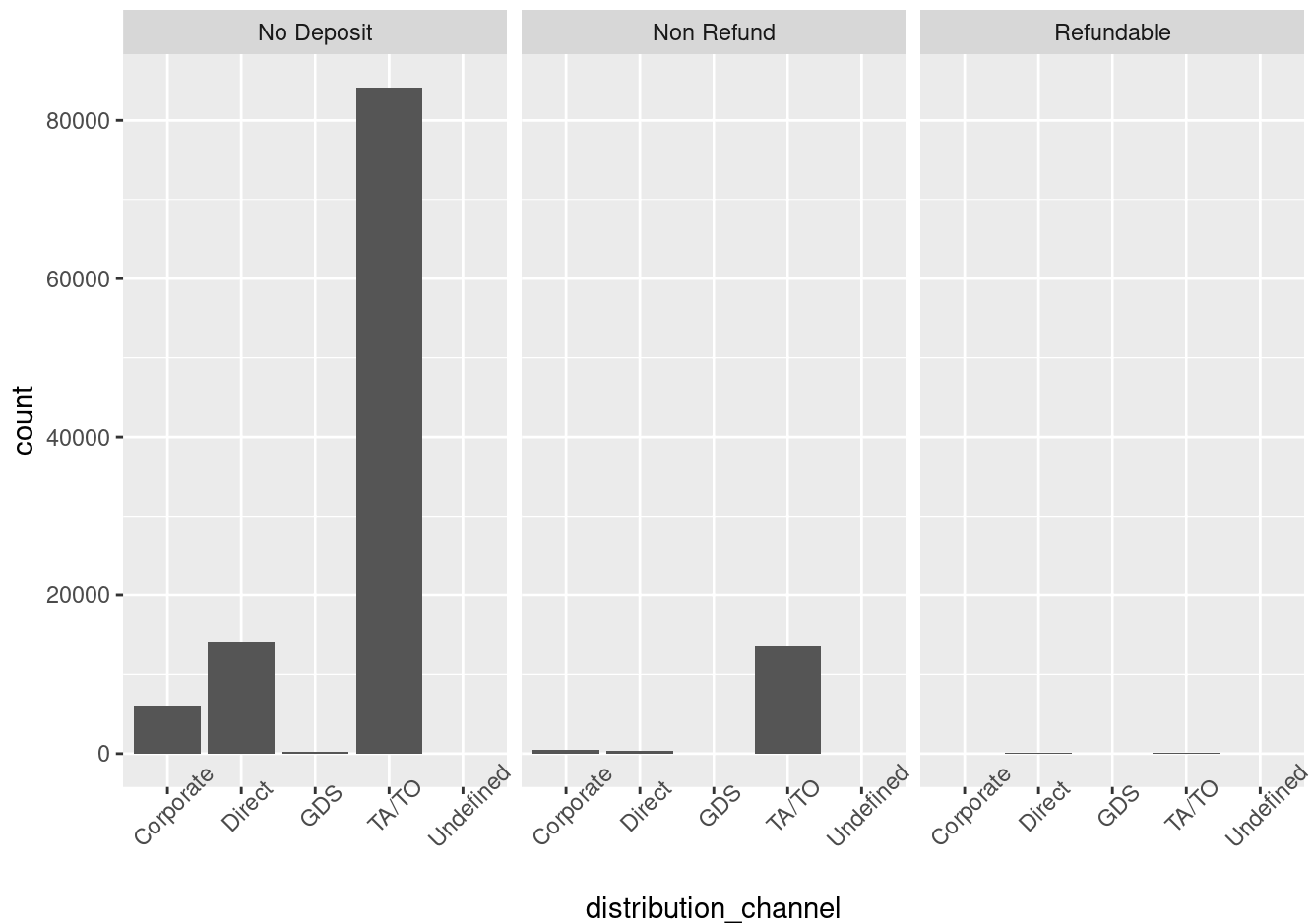
```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~market_segment) +
  theme(axis.text.x = element_text(angle = 45))
```





La función `facet_grid` hace algo similar. La principal diferencia es que `facet_grid` incluirá gráficos incluso si están vacíos. Ejecute el siguiente fragmento de código para comprobarlo:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_grid(~deposit_type) +
  theme(axis.text.x = element_text(angle = 45))
```

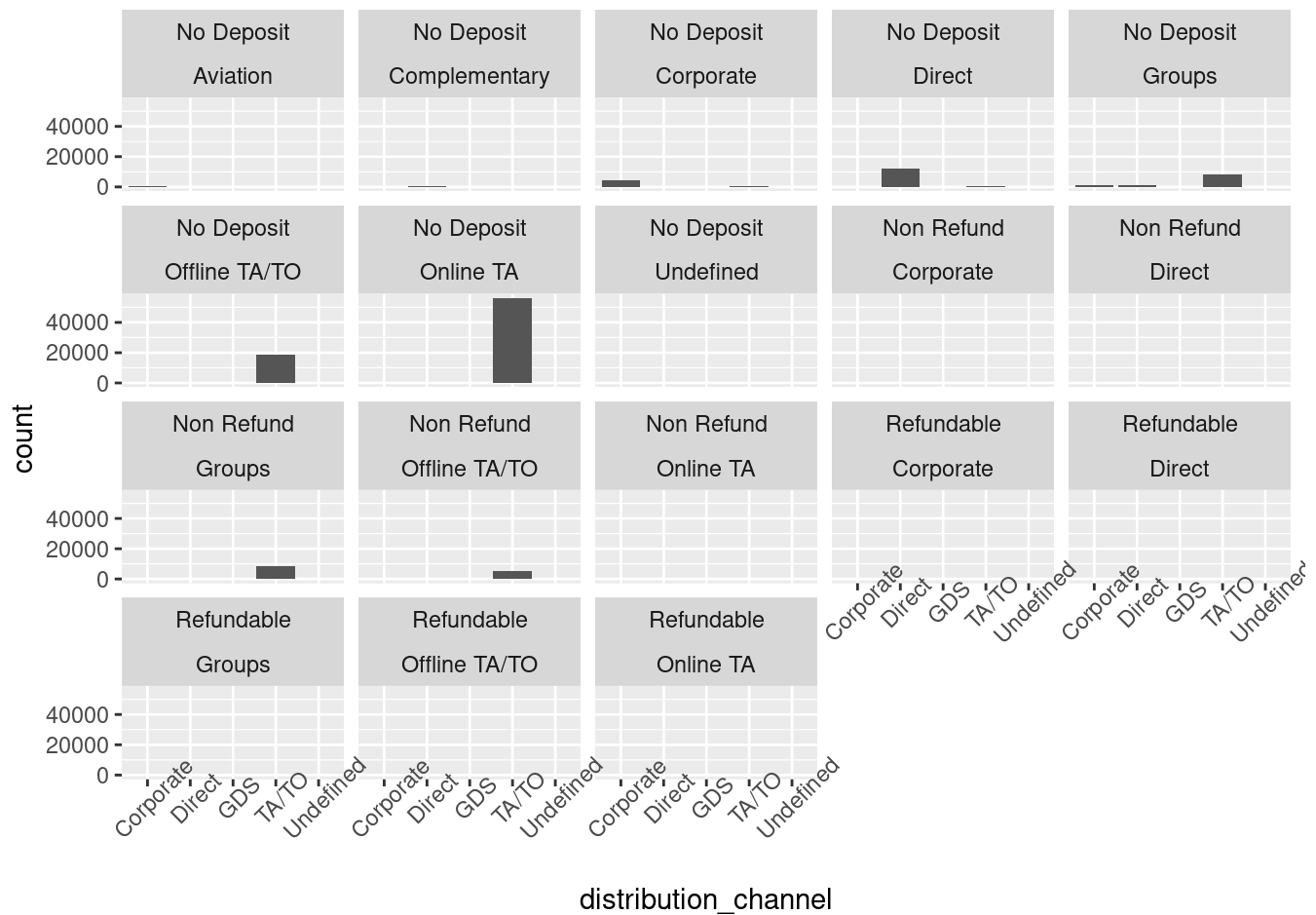


Deberías tener tres gráficos de barras.

Ahora podría poner todo esto en un gráfico y explorar las diferencias por tipo de depósito y segmento de mercado.

Ejecute el fragmento de código siguiente para averiguarlo; observe cómo se usa el carácter ~ antes de las variables por las que se divide el gráfico:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~deposit_type~market_segment) +
  theme(axis.text.x = element_text(angle = 45))
```



Estos gráficos probablemente sean abrumadores y demasiado difíciles de leer, pero pueden resultar útiles si explora sus datos a través de visualizaciones.

## Activity Wrap Up

El paquete `ggplot2` le permite crear una variedad de visualizaciones en `R`, desde simples diagramas de dispersión hasta complicados gráficos de barras multifacéticos. 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. A medida que continúa explorando argumentos estéticos en `ggplot2`, considere cómo podría utilizar las visualizaciones para obtener información y hacer observaciones sobre otros tipos de datos en el futuro.