Lesson 4: Annotating and Saving

Background for this activity

In this activity, you'll review a scenario, and practice adding annotations to a data visualization with ggplot2. You will also learn how to save images from ggplot2 visualizations so that you can add them directly to presentations.

Throughout this activity, you will also have the opportunity to practice writing your own code by making changes to the code chunks yourself. If you encounter an error or get stuck, you can always check the Lesson4_Annotations_Solutions .rmd file in the Solutions folder under Week 4 for the complete, correct code.

The Scenario

As a junior data analyst for a hotel booking company, you have been creating visualizations in `R` with the `ggplot2` package to share insights about your data with stakeholders. After creating a series of visualizations using `ggplot()`, `ggplot2` aesthetics, and filters, your stakeholder asks you to add annotations to your visualizations to help explain your findings in a presentation. Luckily, `ggplot2` has annotation functions built in.

Step 1: Import your data

If you haven't exited out of RStudio since importing this data last time, you can skip these steps. Rerunning these code chunks won't affect your console if you want to run them just in case, though.

If this line causes an error, copy in the line setwd("projects/Course 7/Week 4") before it.

Run the code below to read in the file 'hotel_bookings.csv' into a data frame:

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

Step 2: Refresh Your Memory

By now, you are pretty familiar with this data set. But you can refresh your memory with the `head()` and `colnames()` functions. Run two code chunks below to get at a sample of the data and also preview all the column names:

```
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)

If you haven't already installed and loaded the `ggplot2` package, you will need to do that before you can use the `ggplot()` function. You only have to do this once though, not every time you call `ggplot()`.

You can also skip this step if you haven't closed your RStudio account since doing the last activity. If you aren't sure, you can run the code chunk and hit 'cancel' if the warning message pops up telling you that have already downloaded the `ggplot2` package.

Run the code chunk below to install and load `ggplot2`. This may take a few minutes!

If you haven't installed and loaded tidyverse in this RStudio session, you can run the code chunk below. This may take a few minutes!

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

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

```r
library(tidyverse)
```

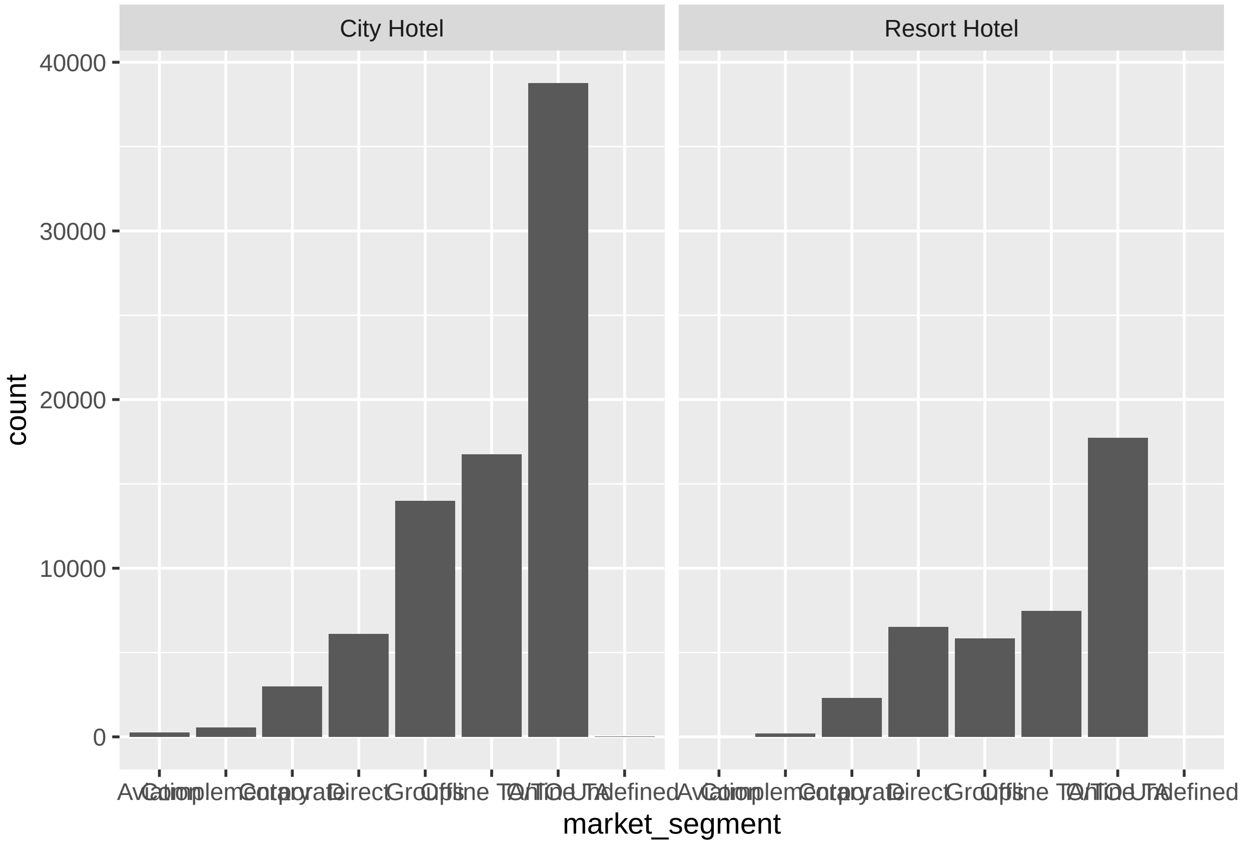
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0

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

Step 4: Annotating your chart

Your stakeholder tells you that they would like you to create a visualization that compares market segments between city hotels and resort hotels. This will help inform how the company targets promotions in the future. They ask you to create a cleaned and labeled version and save it as a .png file so they can include it in a presentation.

As a refresher, here a chart similar to what you created in a previous activity:

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

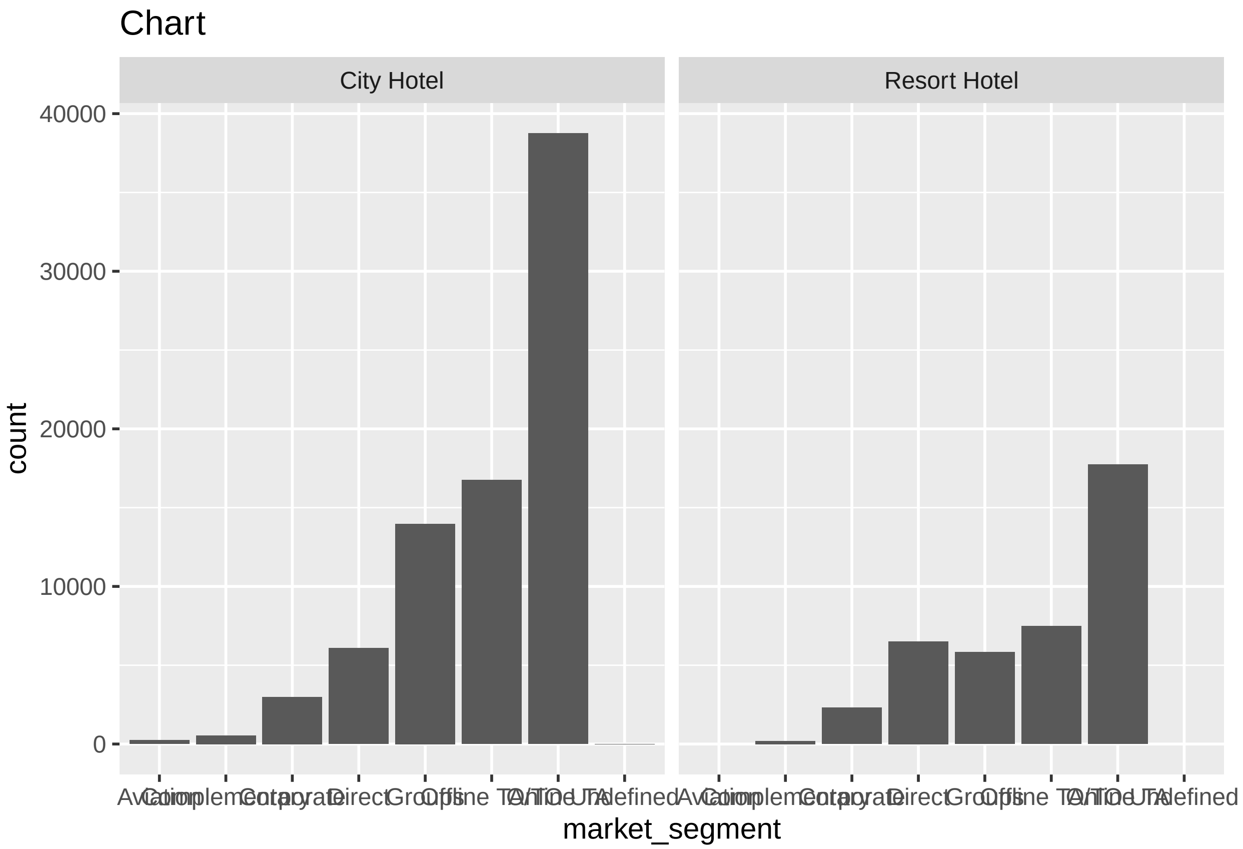


This creates two bar graphs: one for 'city_hotel' data and one for 'resort_hotel' data. The x axis is 'market_segment' and the y axis is 'count' for both charts.

In this visualization it is unclear where the data is from, what the main takeaway is, or even what the data is showing. To explain all of that, you can leverage annotations in `ggplot2`.

The first step will be adding a title; that is often the first thing people will pay attention to when they encounter a data visualization for the first time. To add a title, you will add `labs()` at the end of your `ggplot()` command and then input a title there. Add a descriptive title to the code chunk below:

```r
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Chart")
```



This code chunk will generate the same chart as before, but now it includes a title to explain the data visualization more clearly to your audience.

You also want to add another detail about what time period this data covers. To do this, you need to find out when the data is from.

You realize you can use the `min()` function on the year column in the data:

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

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

And the `max()` function:

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

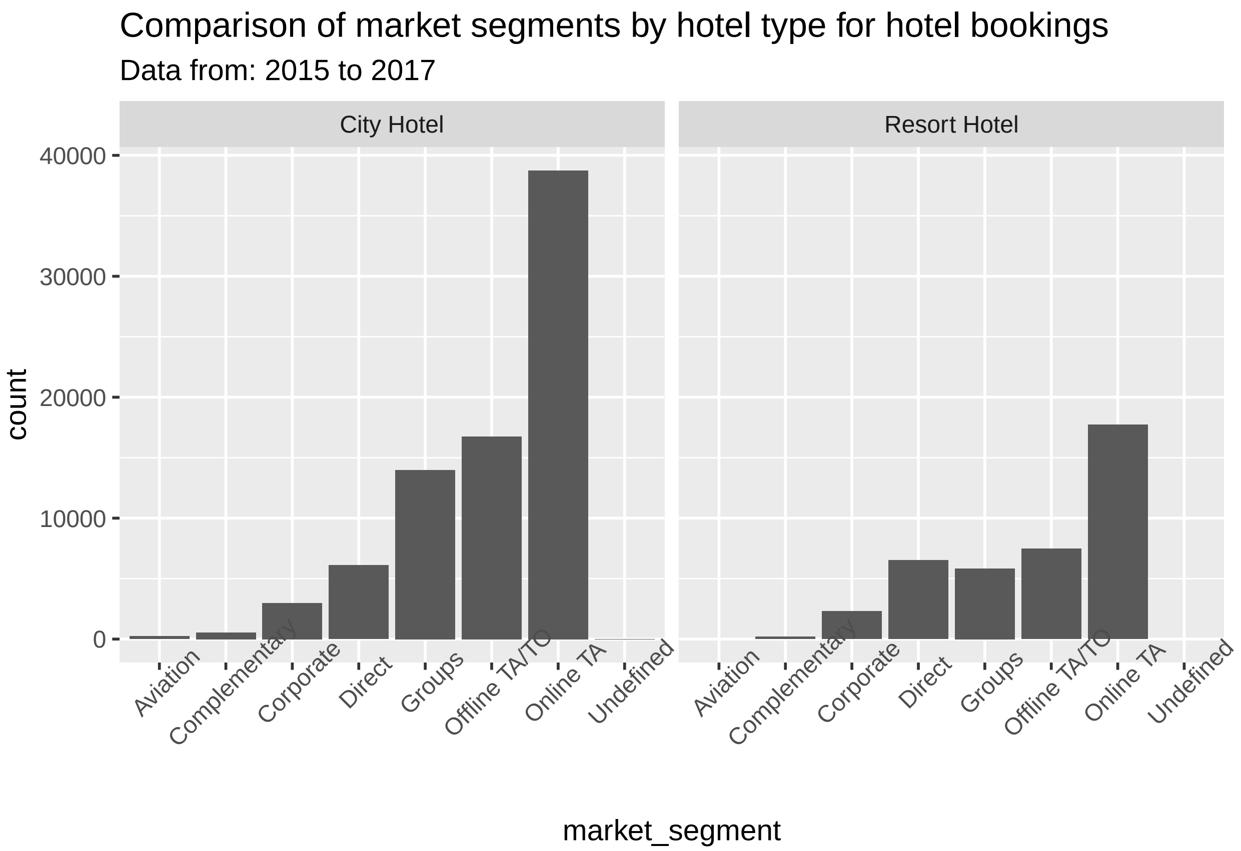
```
## [1] 2017
```

But you will need to save them as variables in order to easily use them in your labeling; the following code chunk creates two of those variables:

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

Now, you will add in a subtitle using `subtitle=` in the `labs()` function. Then, you can use the `paste0()` function to use your newly-created variables in your labels. This is really handy, because if the data gets updated and there is more recent data added, you don't have to change the code below because the variables are dynamic:

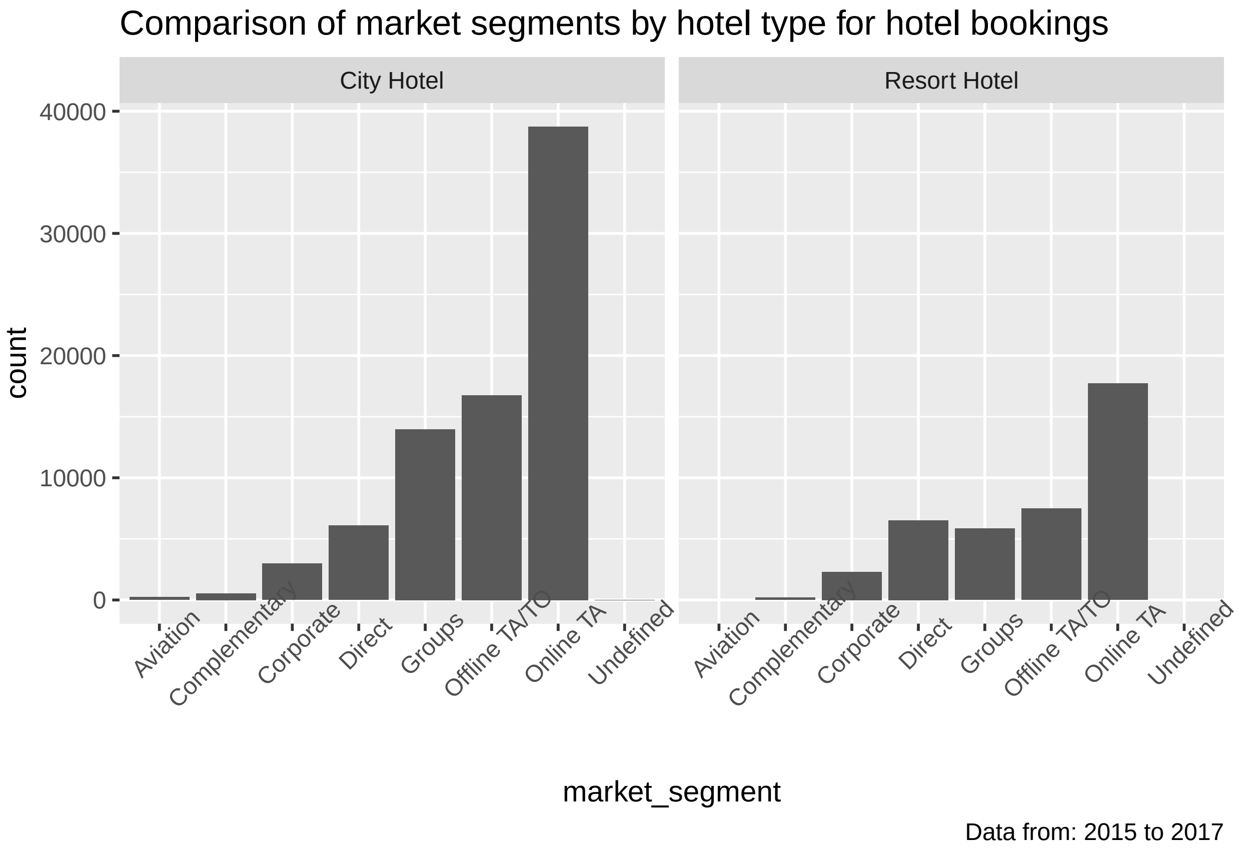
```r
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       subtitle=paste0("Data from: ", mindate, " to ", maxdate))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



This code chunk will add the subtitle 'Data from: 2015 to 2017' underneath the title you added earlier to the chart.

You realize that this chart is displaying the technical details a little too prominently. You don't want that to be the second thing people notice during the presentation. You decide to switch the `subtitle` to a `caption` which will appear in the bottom right corner instead.

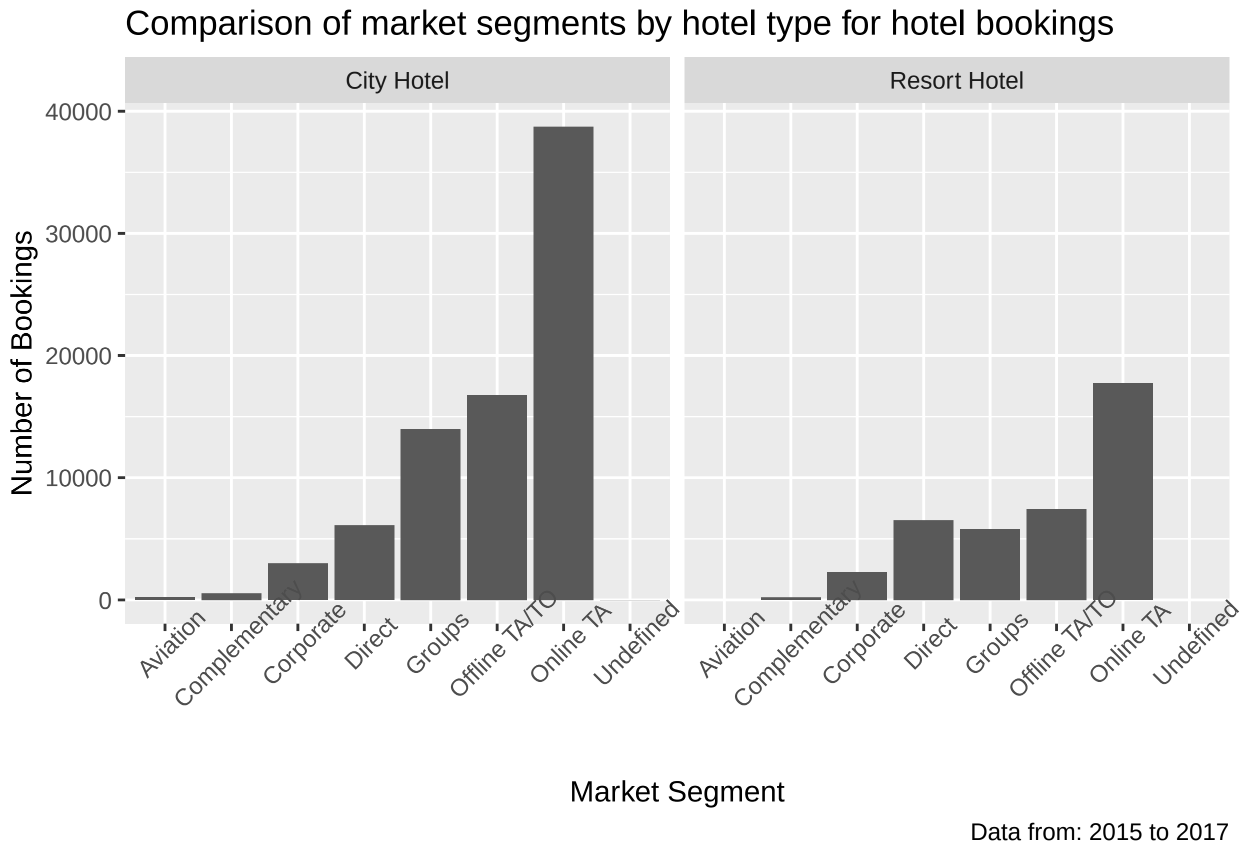
```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       caption=paste0("Data from: ", mindate, " to ", maxdate))
```



This code chunk makes a slight change to the visualization you created in the last chunk; now the "data from: 2015 to 2017" subtitle is in the bottom right corner.

Now you want to clean up the x and y axis labels to make sure they are really clear. To do that, you can add to the `labs()` function and use `x=` and `y=`. Feel free to change the text of the label and play around with it:

```r
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")
```



Now you have the data visualization from earlier, but now the x and y axis labels have been changed from 'market_segment' and 'count' to 'Market Segment' and 'Number of Bookings' so that the chart is clearer.

Step 5: Saving your chart

Now, it's time to save what you just created so you can easily share with stakeholders.

You can use the `ggsave()` function to do just that! It will save your image as a 7x7 at the file path you input by default, which makes it simple to export your plots from R.

The `ggsave()` function in the code chunk below will save the last plot that was generated, so if you ran something after running the code chunk above, run that code chunk again.

Then run the following code chunk to save that plot as a .png file named `hotel_booking_chart`, which makes it clear to your stakeholders what the .png file contains. Now you should be able to find this file in your 'Files' tab in the bottom right of your screen. Check it out!

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

Practice quiz

What dimensions did you save this picture as using `ggsave()`? Note your answer and respond in the Coursera platform.

A: 5x5 B: 10x10 C: 7x7 D: 25x25

If you wanted to make your chart bigger and more rectangular to fit the slide show presentation, you could specify the height and width of your .png in the `ggsave()` command. Edit the code chunk below to create a 16x8 .png image:

```
ggsave('hotel_booking_chart.png',
       width=16,
       height=8)
```

Activity Wrap Up

Now you have finished creating and exporting a data visualization with annotations in `ggplot2`, you can share what you created with key stakeholders to give them insights into your data findings. These skills will allow you to create, annotate, and share your data visualizations directly from your `R` console space. You can practice these skills by modifying the code chunks in the rmd file, or use this code as a starting point in your own project console. You will continue learning more about `ggplot2` functions in this course, but with the skills you have already been practicing, you will be able to generate plots, utilize aesthetic functions, apply filters, and create annotations to explain your data.

When you complete this activity, save it by clicking the File tab in the upper left of the RStudio console and then selecting Save. You will use this document (or any other R activity you have completed and saved in this course) for an upcoming activity.