

Lesson 4: Annotations Solutions

Annotating and saving solutions

This document contains the solutions for the annotating and saving activity. You can use these solutions to check your work and ensure that your code is correct or troubleshoot your code if it is returning errors. If you haven't completed the activity yet, we suggest you go back and finish it before reading the solutions.

If you experience errors, remember that you can search the internet and the RStudio community for help: <https://community.rstudio.com/#>

Step 1: Import your data

The data in this example is originally from the article Hotel Booking Demand Datasets (<https://www.sciencedirect.com/science/article/pii/S2352340918315191>), written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.

The data was downloaded and cleaned by Thomas Mock and Antoine Bichat for #TidyTuesday during the week of February 11th, 2020 (<https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-02-11/readme.md>).

You can learn more about the dataset here: <https://www.kaggle.com/jessemostipak/hotel-booking-demand>

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’ package (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!

```
install.packages('ggplot2')
```

```

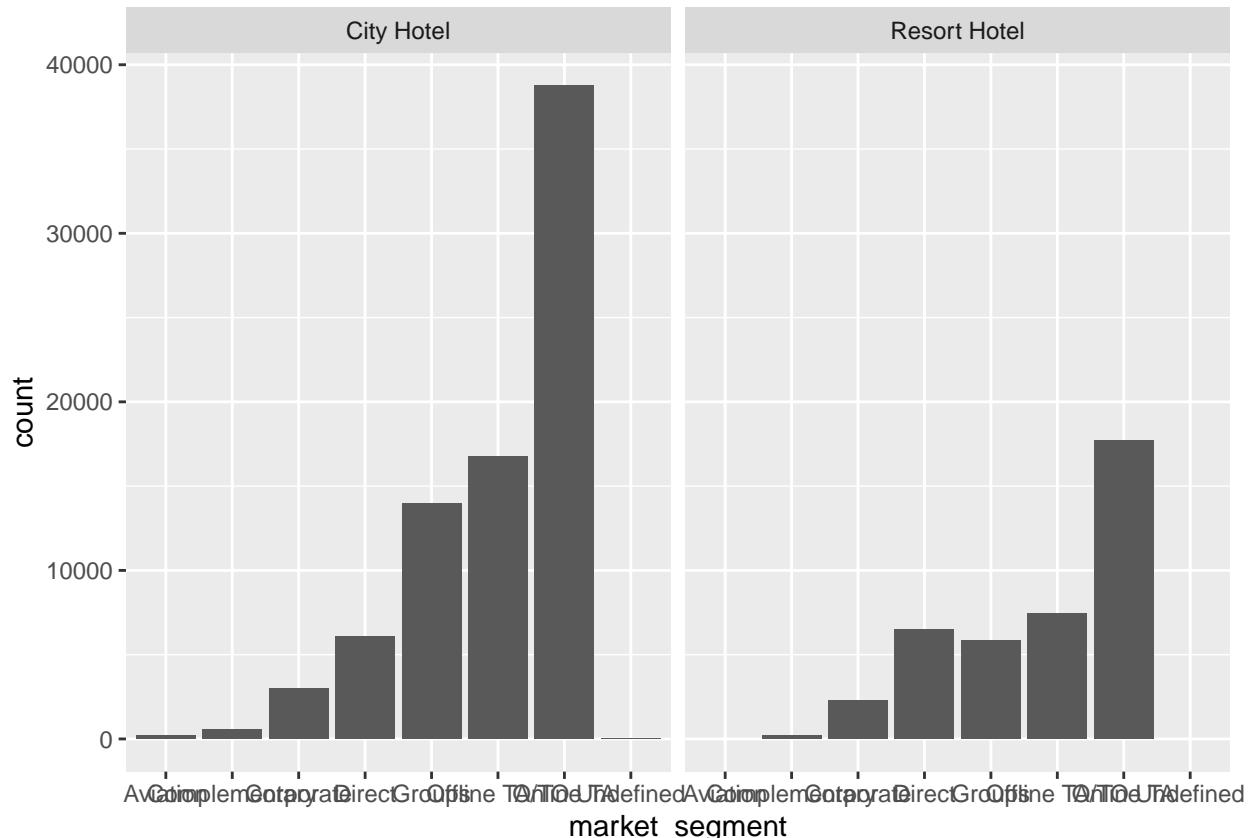
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
library(ggplot2)

```

Step 4: Annotating your chart

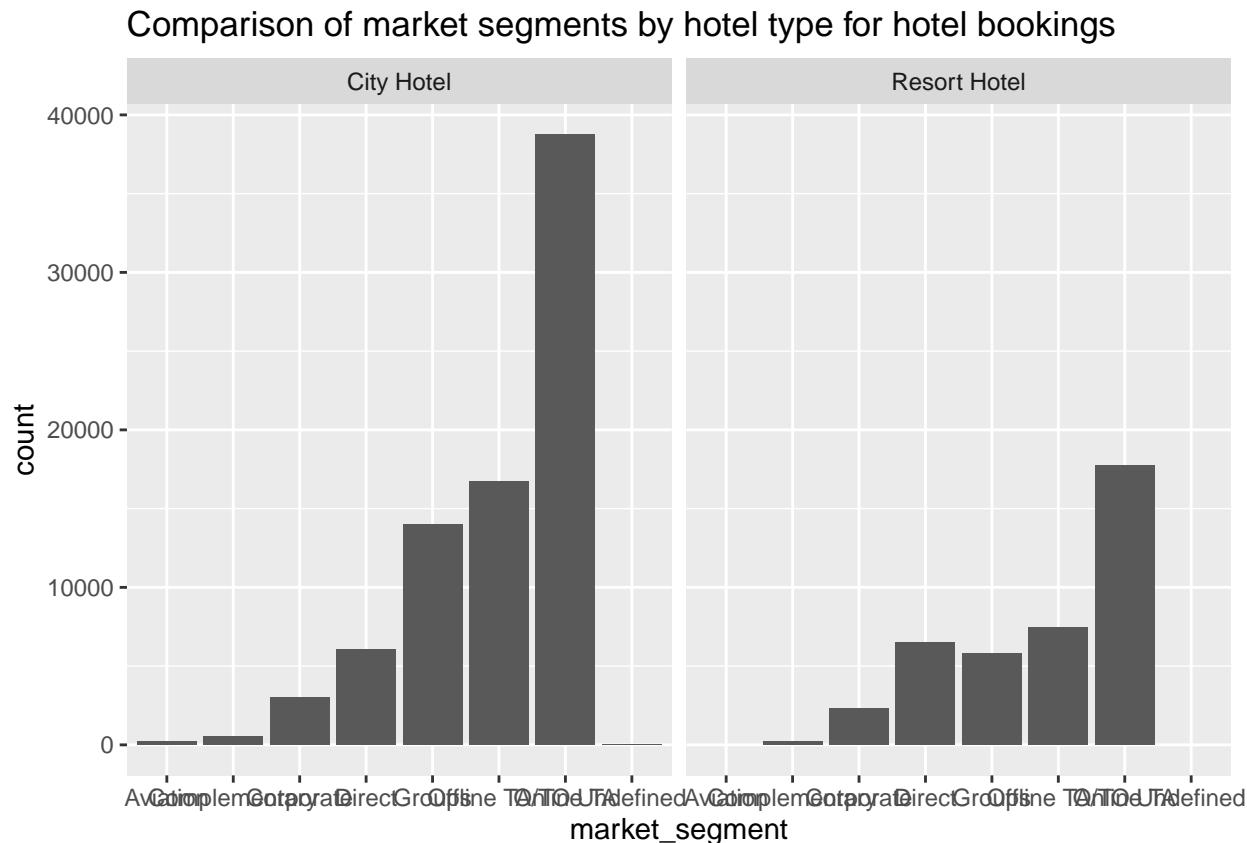
As a refresher, here is the chart you created before:

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



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:

```
ggplot(data = hotel_bookings) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Comparison of market segments by hotel type for hotel bookings")
```



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:

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

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

And the `max()` function:

```
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:

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

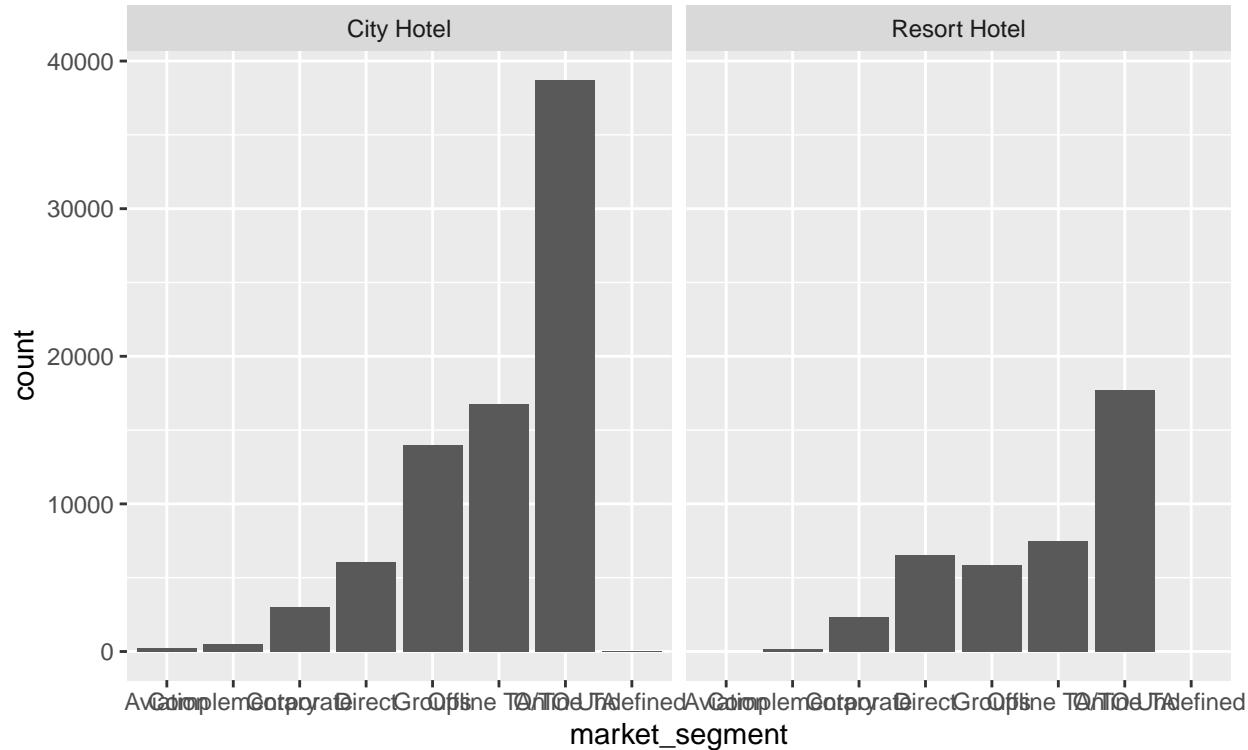
Now, you will add in a subtitle using `subtitle=` in the `labs()` function. Then, you can use the `paste0()`

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:

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

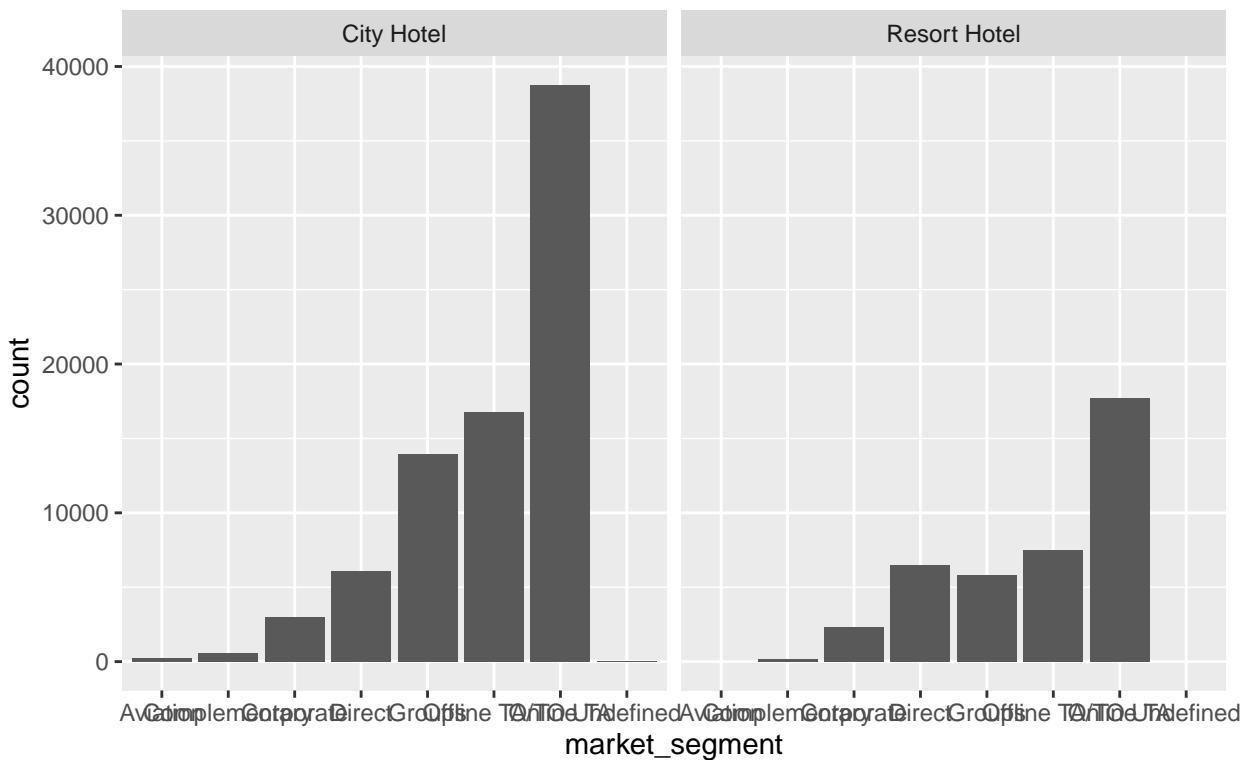
Data from: 2015 to 2017



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) +
  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

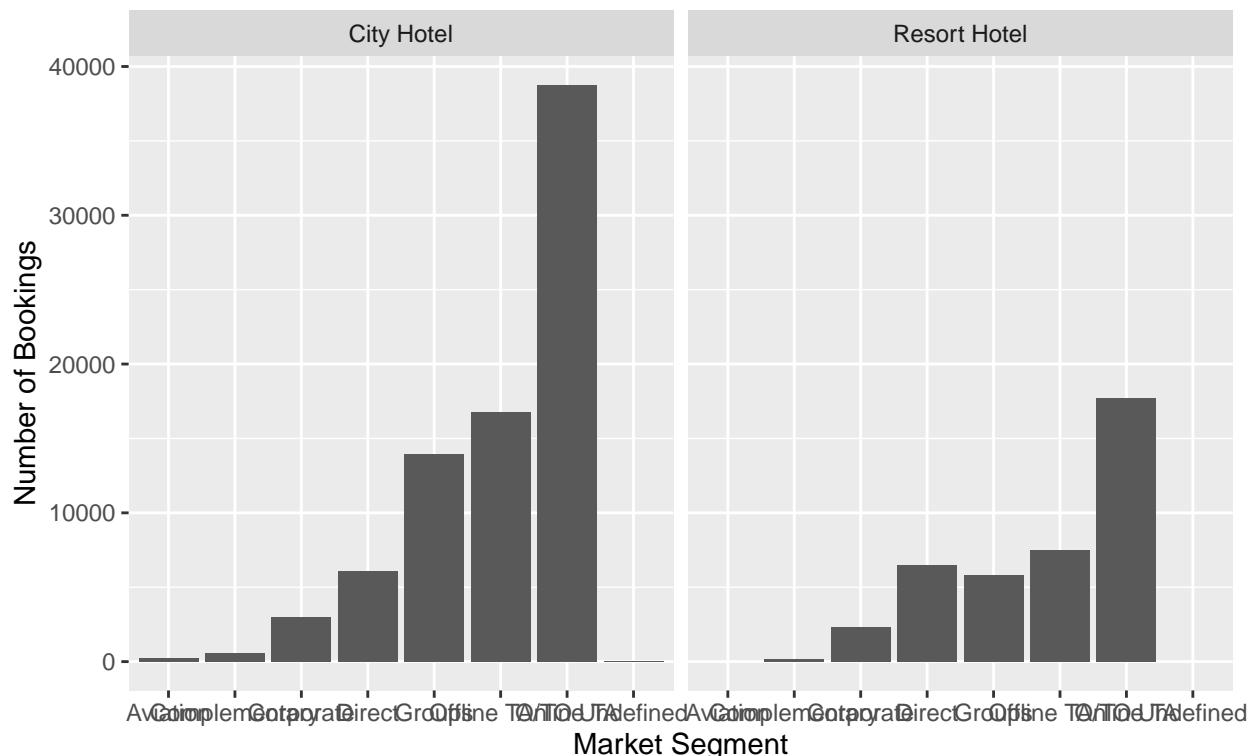


Data from: 2015 to 2017

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:

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

Step 5: Saving your chart

The `ggsave()` function was used to save the last plot that was generated, so if you have run something after running the code chunk above, then run that code chunk again.

Then run the following code chunk to save that plot as a .png file named `city_payment_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')
```

```
## Saving 6.5 x 4.5 in image
```

Practice quiz

What are the default dimensions that `ggsave()` saved your image as?

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

Answer: C. The default dimensions of this `ggsave()` image are 7x7. You can see these dimensions listed after you run the code chunk.

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)
```