|  |
| --- |
| > install.packages("palmerpenguins")  Error in install.packages : Updating loaded packages  > install.packages("palmerpenguins")  Installing package into ‘/cloud/lib/x86\_64-pc-linux-gnu-library/4.3’  (as ‘lib’ is unspecified)  trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/palmerpenguins\_0.1.1.tar.gz'  Content type 'application/x-gzip' length 3001165 bytes (2.9 MB)  ==================================================  downloaded 2.9 MB  \* installing \*binary\* package ‘palmerpenguins’ ...  \* DONE (palmerpenguins)  The downloaded source packages are in  ‘/tmp/RtmpPZbiTj/downloaded\_packages’  > library(penguins)  Error in library(penguins) : there is no package called ‘penguins’  > library(palmerpenguins)  > library(tidyverse)  ── **Attaching core tidyverse packages** ──────────  ✔ forcats 1.0.0 ✔ readr 2.1.4  ✔ ggplot2 3.4.2 ✔ 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()  ℹ Use the conflicted package to force all conflicts to become errors  > penguins%>%arrange(bill\_length\_mm)  # A tibble: 344 × 8  species island bill\_length\_mm bill\_depth\_mm  *<fct>* *<fct>* *<dbl>* *<dbl>*  1 Adelie Dream 32.1 15.5  2 Adelie Dream 33.1 16.1  3 Adelie Torger… 33.5 19  4 Adelie Dream 34 17.1  5 Adelie Torger… 34.1 18.1  6 Adelie Torger… 34.4 18.4  7 Adelie Biscoe 34.5 18.1  8 Adelie Torger… 34.6 21.1  9 Adelie Torger… 34.6 17.2  10 Adelie Biscoe 35 17.9  # ℹ 334 more rows  # ℹ 4 more variables: flipper\_length\_mm <int>,  # body\_mass\_g <int>, sex <fct>, year <int>  # ℹ Use `print(n = ...)` to see more rows  > penguins%>%arrange(-bill\_length\_mm)  # A tibble: 344 × 8  species island bill\_length\_mm bill\_depth\_mm  *<fct>* *<fct>* *<dbl>* *<dbl>*  1 Gentoo Biscoe 59.6 17  2 Chinstr… Dream 58 17.8  3 Gentoo Biscoe 55.9 17  4 Chinstr… Dream 55.8 19.8  5 Gentoo Biscoe 55.1 16  6 Gentoo Biscoe 54.3 15.7  7 Chinstr… Dream 54.2 20.8  8 Chinstr… Dream 53.5 19.9  9 Gentoo Biscoe 53.4 15.8  10 Chinstr… Dream 52.8 20  # ℹ 334 more rows  # ℹ 4 more variables: flipper\_length\_mm <int>,  # body\_mass\_g <int>, sex <fct>, year <int>  # ℹ Use `print(n = ...)` to see more rows  > penguins2 <- penguins %>% arrange(-bill\_length\_mm)  > View(penguins2)  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize(mean\_bill\_length\_mm)  **Error in `summarize()`:**  ℹ In argument: `mean\_bill\_length\_mm`.  ℹ In group 1: `island = Biscoe`.  **Caused by error:**  ! object 'mean\_bill\_length\_mm' not found  Run `rlang::last\_trace()` to see where the error occurred.  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize(mean\_bill\_length\_mm=mean(bill\_length\_mm))  # A tibble: 3 × 2  island mean\_bill\_length\_mm  *<fct>* *<dbl>*  1 Biscoe 45.2  2 Dream 44.2  3 Torgersen 39.0  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize(max\_bill\_length\_mm=max(bill\_length\_mm))  # A tibble: 3 × 2  island max\_bill\_length\_mm  *<fct>* *<dbl>*  1 Biscoe 59.6  2 Dream 58  3 Torgersen 46  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize(min\_bill\_length\_mm=min(bill\_length\_mm))  # A tibble: 3 × 2  island min\_bill\_length\_mm  *<fct>* *<dbl>*  1 Biscoe 34.5  2 Dream 32.1  3 Torgersen 33.5  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize((max\_bill\_length\_mm=max(bill\_length\_mm)),(mean\_bill\_length\_mm=mean(bill\_length\_mm)))  # A tibble: 3 × 3  island (max\_bill\_length\_mm = max(bill\_le…¹  *<fct>* *<dbl>*  1 Biscoe 59.6  2 Dream 58  3 Torgersen 46  # ℹ abbreviated name:  # ¹​`(max\_bill\_length\_mm = max(bill\_length\_mm))`  # ℹ 1 more variable:  # `(mean\_bill\_length\_mm = mean(bill\_length\_mm))` <dbl>  > penguins%>%group\_by(island)%>%drop\_na()%>% summarize(max\_bill=max(bill\_length\_mm),mean\_bill=mean(bill\_length\_mm))  # A tibble: 3 × 3  island max\_bill mean\_bill  *<fct>* *<dbl>* *<dbl>*  1 Biscoe 59.6 45.2  2 Dream 58 44.2  3 Torgersen 46 39.0  > penguins%>%group\_by(species,island)%>%drop\_na()%>% summarize(max\_bill=max(bill\_length\_mm),mean\_bill=mean(bill\_length\_mm))  `summarise()` has grouped output by 'species'.  You can override using the `.groups` argument.  # A tibble: 5 × 4  # Groups: species [3]  species island max\_bill mean\_bill  *<fct>* *<fct>* *<dbl>* *<dbl>*  1 Adelie Biscoe 45.6 39.0  2 Adelie Dream 44.1 38.5  3 Adelie Torgersen 46 39.0  4 Chinstrap Dream 58 48.8  5 Gentoo Biscoe 59.6 47.6  > penguins%>%filter(species=="Adelie")  # A tibble: 152 × 8  species island bill\_length\_mm bill\_depth\_mm  *<fct>* *<fct>* *<dbl>* *<dbl>*  1 Adelie Torger… 39.1 18.7  2 Adelie Torger… 39.5 17.4  3 Adelie Torger… 40.3 18  4 Adelie Torger… NA NA  5 Adelie Torger… 36.7 19.3  6 Adelie Torger… 39.3 20.6  7 Adelie Torger… 38.9 17.8  8 Adelie Torger… 39.2 19.6  9 Adelie Torger… 34.1 18.1  10 Adelie Torger… 42 20.2  # ℹ 142 more rows  # ℹ 4 more variables: flipper\_length\_mm <int>,  # body\_mass\_g <int>, sex <fct>, year <int>  # ℹ Use `print(n = ...)` to see more rows |
|  |
| |  | | --- | | > penguins %>%  select(-species)  penguins%>%  rename(island\_new=island)  ##to change casing  rename\_with(penguins, toupper) | |

#load data set , use nesting , sorting on basis of length

data("ToothGrowth")

View(ToothGrowth)

filtered\_tg<- filter(ToothGrowth, dose==0.5)

View(filtered\_tg)

arrange(filtered\_tg,len)

View(filtered\_tg)

arrange(filter(ToothGrowth,dose==0.5),len)

View(filtered\_tg)

#piping(%>%)

filtered\_TW<- ToothGrowth%>%

filter(dose==0.5)%>%

group\_by(supp)%>%

summarise(mean\_len = mean(len, na.rm = T), .group="drop")

itle: "Lesson 2: Create your own data frame"

output: html\_document

---

## Background for this activity

This activity is focused on creating and using data frames in `R`. A data frame is a collection of columns containing data, similar to a spreadsheet or SQL table. Data frames are one of the basic tools you will use to work with data in `R`. And you can create data frames from different data sources.

There are three common sources for data:

- A`package` with data that can be accessed by loading that `package`

- An external file like a spreadsheet or CSV that can be imported into `R`

- Data that has been generated from scratch using `R` code

Wherever data comes from, you will almost always want to store it in a data frame object to work with it. Now, you can start creating and exploring data frames with the code chunks in the RMD space. To interact with the code chunk, click the green arrow in the top-right corner of the chunk. The executed code will appear in the RMD space and your console.

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 Lesson2\_Dataframe\_Solutions .rmd file in the Solutions folder under Week 3 for the complete, correct code.

## Step 1: Load packages

Start by installing the required package; in this case, you will want to install `tidyverse`. If you have already installed and loaded `tidyverse` in this session, feel free to skip the code chunks in this step.

```{r}

install.packages("tidyverse")

```

Once a package is installed, you can load it by running the `library()` function with the package name inside the parentheses:

```{r}

library(tidyverse)

```

## Step 2: Create data frame

Sometimes you will need to generate a data frame directly in `R`. There are a number of ways to do this; one of the most common is to create individual vectors of data and then combine them into a data frame using the `data.frame()` function.

Here's how this works. First, create a vector of names by inserting four names into this code block between the quotation marks and then run it:

```{r}

names <- c("", "", "", "")

```

Then create a vector of ages by adding four ages separated by commas to the code chunk below. Make sure you are inputting numeric values for the ages or you might get an error.

```{r}

age <- c(, , , )

```

With these two vectors, you can create a new data frame called `people`:

```{r}

people <- data.frame(names, age)

```

## Step 3: inspect the data frame

Now that you have this data frame, you can use some different functions to inspect it.

One common function you can use to preview the data is the `head()` function, which returns the columns and the first several rows of data. You can check out how the `head()` function works by running the chunk below:

```{r}

head(people)

```

In addition to `head()`, there are a number of other useful functions to summarize or preview your data. For example, the `str()` and `glimpse()` functions will both provide summaries of each column in your data arranged horizontally. You can check out these two functions in action by running the code chunks below:

```{r}

str(people)

```

```{r}

glimpse(people)

```

You can also use `colnames()` to get a list the column names in your data set. Run the code chunk below to check out this function:

```{r}

colnames(people)

```

Now that you have a data frame, you can work with it using all of the tools in `R`. For example, you could use `mutate()` if you wanted to create a new variable that would capture each person's age in twenty years. The code chunk below creates that new variable:

```{r}

mutate(people, age\_in\_20 = age + 20)

```

## Step 4: Try it yourself

To get more familiar with creating and using data frames, use the code chunks below to create your own custom data frame.

First, create a vector of any five different fruits. You can type directly into the code chunk below; just place your cursor in the box and click to type. Once you have input the fruits you want in your data frame, run the code chunk.

```{r}

fruits<- c("banana", "peach", "raspberrys", "nuts", "baingan")

```

Now, create a new vector with a number representing your own personal rank for each fruit. Give a 1 to the fruit you like the most, and a 5 to the fruit you like the least. Remember, the scores need to be in the same order as the fruit above. So if your favorite fruit is last in the list above, the score `1` needs to be in the last position in the list below. Once you have input your rankings, run the code chunk.

```{r}

personal\_fav<-c(3, 5, 5, 4, 1)

```

Finally, combine the two vectors into a data frame. You can call it `fruit\_ranks`. Edit the code chunk below and run it to create your data frame.

```{r}

fruit\_ranks<- data.frame(fruits, personal\_fav)

head(fruit\_ranks)

arrange(fruit\_ranks, personal\_fav)

```

After you run this code chunk, it will create a data frame with your fruits and rankings.

## Activity Wrap Up

In this activity, you learned how to create data frames, view them with summary functions like `head()` and `glimpse()`, and then made changes with the `mutate()` function. You can continue practicing these skills by modifying the code chunks in the rmd file, or use this code as a starting point in your own project console. As you explore data frames, consider how they are similar and different to the tables you have worked with in other data analysis tools like spreadsheets and SQL. Data frames are one of the most basic building blocks you will need to work with data in `R`. So understanding how to create and work with data frames is an important first step to analyzing data.

Make sure to mark this activity as complete in Coursera.

|  |
| --- |
| > head(hotel\_bookings)  # A tibble: 6 × 32  hotel is\_canceled lead\_time arrival\_date\_year  *<chr>* *<dbl>* *<dbl>* *<dbl>*  1 Reso… 0 342 2015  2 Reso… 0 737 2015  3 Reso… 0 7 2015  4 Reso… 0 13 2015  5 Reso… 0 14 2015  6 Reso… 0 14 2015  # ℹ 28 more variables:  # arrival\_date\_month <chr>,  # arrival\_date\_week\_number <dbl>,  # arrival\_date\_day\_of\_month <dbl>,  # stays\_in\_weekend\_nights <dbl>,  # stays\_in\_week\_nights <dbl>, adults <dbl>,  # children <dbl>, babies <dbl>, …  > arrange(hotel\_bookings, lead\_time)  # A tibble: 119,390 × 32  hotel is\_canceled lead\_time  *<chr>* *<dbl>* *<dbl>*  1 Resort Hotel 0 0  2 Resort Hotel 0 0  3 Resort Hotel 0 0  4 Resort Hotel 0 0  5 Resort Hotel 0 0  6 Resort Hotel 0 0  7 Resort Hotel 0 0  8 Resort Hotel 0 0  9 Resort Hotel 0 0  10 Resort Hotel 0 0  # ℹ 119,380 more rows  # ℹ 29 more variables:  # arrival\_date\_year <dbl>,  # arrival\_date\_month <chr>,  # arrival\_date\_week\_number <dbl>,  # arrival\_date\_day\_of\_month <dbl>,  # stays\_in\_weekend\_nights <dbl>, …  # ℹ Use `print(n = ...)` to see more rows  > arrange(hotel\_bookings, desc(lead\_time))  # A tibble: 119,390 × 32  hotel is\_canceled lead\_time  *<chr>* *<dbl>* *<dbl>*  1 Resort Hotel 0 737  2 Resort Hotel 0 709  3 City Hotel 1 629  4 City Hotel 1 629  5 City Hotel 1 629  6 City Hotel 1 629  7 City Hotel 1 629  8 City Hotel 1 629  9 City Hotel 1 629  10 City Hotel 1 629  # ℹ 119,380 more rows  # ℹ 29 more variables:  # arrival\_date\_year <dbl>,  # arrival\_date\_month <chr>,  # arrival\_date\_week\_number <dbl>,  # arrival\_date\_day\_of\_month <dbl>,  # stays\_in\_weekend\_nights <dbl>, …  # ℹ Use `print(n = ...)` to see more rows  > hotel\_bookingsv2<-arrange(hotel\_bookings, desc(lead\_time))  > head(hotel\_bookingsv2)  # A tibble: 6 × 32  hotel is\_canceled lead\_time arrival\_date\_year  *<chr>* *<dbl>* *<dbl>* *<dbl>*  1 Reso… 0 737 2015  2 Reso… 0 709 2016  3 City… 1 629 2017  4 City… 1 629 2017  5 City… 1 629 2017  6 City… 1 629 2017  # ℹ 28 more variables:  # arrival\_date\_month <chr>,  # arrival\_date\_week\_number <dbl>,  # arrival\_date\_day\_of\_month <dbl>,  # stays\_in\_weekend\_nights <dbl>,  # stays\_in\_week\_nights <dbl>, adults <dbl>,  # children <dbl>, babies <dbl>, …  > max(hotel\_bookings$lead\_time)  [1] 737  > min(hotel\_bookings$lead\_time)  [1] 0  > mean(hotel\_bookings$lead\_time)  [1] 104.0114  > mean(hotel\_bookings\_v2$lead\_time)  Error: object 'hotel\_bookings\_v2' not found  > View(hotel\_bookingsv2)  > mean(hotel\_bookingsv2$lead\_time)  [1] 104.0114  > hotel\_bookings\_city<-hotel\_bookings%>%  + filter(hotel=="City Hotel")  > head(hotel\_bookings\_city)  # A tibble: 6 × 32  hotel is\_canceled lead\_time arrival\_date\_year  *<chr>* *<dbl>* *<dbl>* *<dbl>*  1 City… 0 6 2015  2 City… 1 88 2015  3 City… 1 65 2015  4 City… 1 92 2015  5 City… 1 100 2015  6 City… 1 79 2015  # ℹ 28 more variables:  # arrival\_date\_month <chr>,  # arrival\_date\_week\_number <dbl>,  # arrival\_date\_day\_of\_month <dbl>,  # stays\_in\_weekend\_nights <dbl>,  # stays\_in\_week\_nights <dbl>, adults <dbl>,  # children <dbl>, babies <dbl>, …  > mean(hotel\_bookings\_city$lead\_time)  [1] 109.7357  > hotel\_summary <-  + hotel\_bookings %>%  + group\_by(hotel) %>%  + summarise(average\_lead\_time=mean(lead\_time),  + min\_lead\_time=min(lead\_time),  + max\_lead\_time=max(lead\_time))  > head(hotel\_summary)  # A tibble: 2 × 4  hotel average\_lead\_time min\_lead\_time  *<chr>* *<dbl>* *<dbl>*  1 City Hotel 110. 0  2 Resort Hotel 92.7 0  # ℹ 1 more variable: max\_lead\_time <dbl>  > sum\_cancelled<-sum(hotel\_bookings$is\_canceled)  > sum\_cancelled  [1] 44224 |
|  |
| |  | | --- | | > | |

install.packages("SimDesign")

library(SimDesign)

See ?SimFunctions to get started with SimDesign

> actual\_Temp<- c(68.3, 45.5, 67, 59)

> predicted\_Temp<- c(70, 58.9, 63, 67)

> bias(actual\_Temp, predicted\_Temp)

> ggplot(data=hotel\_bookings)+

+ geom\_point(mapping = aes(x=lead\_time,y=children))

Warning message:

Removed 4 rows containing missing values

(`geom\_point()`).

> geom\_point(mapping = aes(x=stays\_in\_weekend\_nights,y=children))

mapping: x = ~stays\_in\_weekend\_nights, y = ~children

geom\_point: na.rm = FALSE

stat\_identity: na.rm = FALSE

position\_identity

> ggplot(data=hotel\_bookings)+

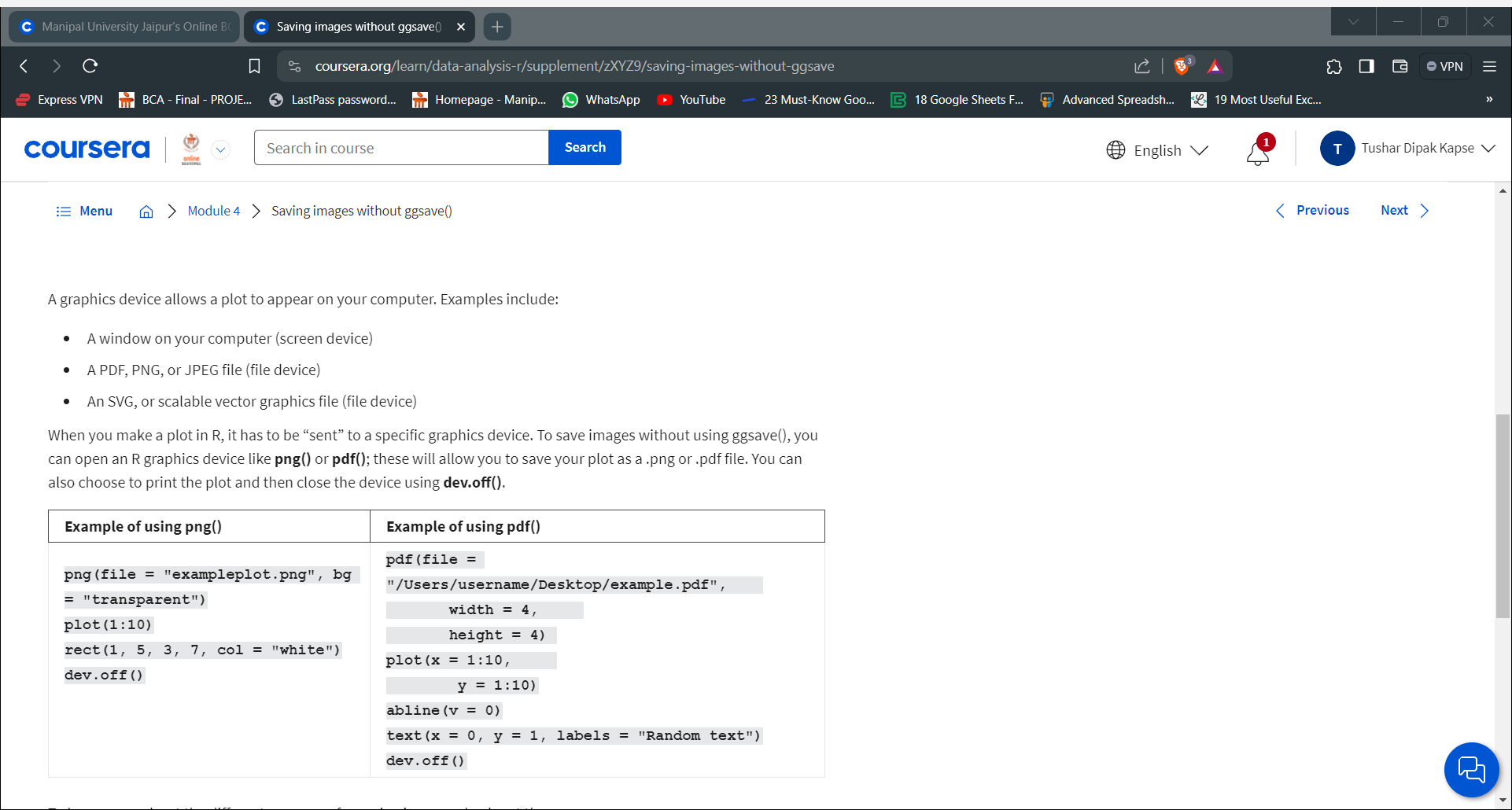
+ geom\_point(mapping = aes(x=stays\_in\_weekend\_nights,y=children))

Warning message:

Removed 4 rows containing missing values

(`geom\_point()`).

|  |
| --- |
| ggplot(data=penguins)+  + geom\_smooth(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g))+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g))  `geom\_smooth()` using method = 'loess' and formula =  'y ~ x'  Warning messages:  1: Removed 2 rows containing non-finite values  (`stat\_smooth()`).  2: Removed 2 rows containing missing values  (`geom\_point()`).  > ggplot(data=penguins)+  + geom\_smooth(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g,linetype=species))  `geom\_smooth()` using method = 'loess' and formula =  'y ~ x'  Warning message:  Removed 2 rows containing non-finite values  (`stat\_smooth()`).  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut))  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut,color=cut))  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut,fill=cut))  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut,fill=clarity)) |
|  |
| |  | | --- | | > | |
| > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g))+  + facet\_wrap(~species)  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`).  > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g,color=species))+  + facet\_wrap(~species)  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`).  > > ggplot(data=diamonds)+  Error: unexpected '>' in ">"  > ggplot(data=diamonds)+  + + geom\_bar(mapping = aes(x=cut,color=cut))+  + facet\_wrap(~species)  **Error in `+.gg`:**  ! Cannot use `+` with a single argument  ℹ Did you accidentally put `+` on a new line?  Run `rlang::last\_trace()` to see where the error occurred.  > ggplot(data=diamonds)+  + + + geom\_bar(mapping = aes(x=cut,color=cut))+  + + facet\_wrap(~cut)  **Error in `+.gg`:**  ! Cannot use `+` with a single argument  ℹ Did you accidentally put `+` on a new line?  Run `rlang::last\_trace()` to see where the error occurred.  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut,color=cut))+  + facet\_wrap(~cut)  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=cut,fill=clarity))+  + facet\_wrap(~cut)  > ggplot(data=diamonds)+  + geom\_bar(mapping = aes(x=color,fill=cut))+  + facet\_wrap(~cut)  > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g,color=species))+  + facet\_grid(sex~species)  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`). |
|  |
| |  | | --- | | > | |
| > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g, color=species))+  + labs(title = "Palmers PENGUINS: Body mass vs Flipper length")  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`).  > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g, color=species))+  + labs(title = "Palmers PENGUINS: Body mass vs Flipper length",subtitle = "sample of three penguins species",caption = "data created by palmer")  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`).  > ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g, color=species))+  + labs(title = "Palmers PENGUINS: Body mass vs Flipper length",subtitle = "sample of three penguins species",caption = "data created by palmer")+  + annotate("text",x=220,y=3500,label="the gentos are largest",color="purple",frontface="bold",size=3,angle=45)  Warning messages:  1: In annotate("text", x = 220, y = 3500, label = "the gentos are largest", :  Ignoring unknown parameters: `frontface`  2: Removed 2 rows containing missing values  (`geom\_point()`).  > p<-ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g, color=species))+  + labs(title = "Palmers PENGUINS: Body mass vs Flipper length",subtitle = "sample of three penguins species",caption = "data created by palmer")+  + annotate("text",x=220,y=4500,label="the gentos are largest",color="purple",frontface="bold",size=3,angle=45)  Warning message:  In annotate("text", x = 220, y = 4500, label = "the gentos are largest", :  Ignoring unknown parameters: `frontface`  > p  Warning message:  Removed 2 rows containing missing values  (`geom\_point()`).  > p<-ggplot(data=penguins)+  + geom\_point(mapping = aes(x=flipper\_length\_mm,y=body\_mass\_g, color=species))+  + labs(title = "Palmers PENGUINS: Body mass vs Flipper length",subtitle = "sample of three penguins species",caption = "data created by palmer")  > p+ annotate("text",x=400,y=4500,label="the gentos are largest",color="purple",frontface="bold",size=3,angle=45)  Warning messages:  1: In annotate("text", x = 400, y = 4500, label = "the gentos are largest", :  Ignoring unknown parameters: `frontface`  2: Removed 2 rows containing missing values  (`geom\_point()`).  > p+ annotate("text",x=260,y=4500,label="the gentos are largest",color="purple",frontface="bold",size=3,angle=60)  Warning messages:  1: In annotate("text", x = 260, y = 4500, label = "the gentos are largest", :  Ignoring unknown parameters: `frontface`  2: Removed 2 rows containing missing values (`geom\_point()`).  > p+ annotate("text",x=200,y=3000,label="the gentos are largest",color="purple",frontface="bold",size=3,angle=30)  Warning messages:  1: In annotate("text", x = 200, y = 3000, label = "the gentos are largest", :  Ignoring unknown parameters: `frontface`  2: Removed 2 rows containing missing values (`geom\_point()`).  Tosave plot  ggplot(“Nameoffile.filetype”)  or |
|  |
| |  | | --- | | > | |



+ <ggplot(data = hotel\_bookings) +

+ geom\_bar(mapping = aes(x = market\_segment)) +

+ facet\_wrap(~hotel) +

+ labs(title="hotel bookings")

Error: unexpected symbol in:

" facet\_wrap(~hotel) +

labs(title="hotel"

> ggplot(data = hotel\_bookings) +

+ geom\_bar(mapping = aes(x = market\_segment)) +

+ facet\_wrap(~hotel) +

+ labs(title="The hotel Bookings")

> mindate <- min(hotel\_bookings$arrival\_date\_year)

> maxdate <- max(hotel\_bookings$arrival\_date\_year)

> 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))

> 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")

> ggsave('hotel\_booking\_chart.png', width=7,

+ height=7)

> t<-ggplot(data = hotel\_bookings) +

+ geom\_bar(mapping = aes(x = market\_segment,fill=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")

> t

