Analysis in R

Priyanshi Garg

2025-04-22

1. Intro to R

This project is part of my learning journey to develop and practice my skills in data analysis using R.

I have chosen the palmerpenguins dataset because it is widely used for practicing R and provides a clean, real-world dataset that's great for exploring data science concepts.

In this project, I will cover the full analysis workflow — including installing and loading packages, exploring and cleaning data, performing transformations, visualizing patterns, and summarizing key insights.

2. Getting started

To begin working with the data, I need to download and load the necessary packages — tidyverse for data manipulation and visualization, and palmerpenguins for the dataset itself. In R, this is done using the install.packages() function to install the package, and the library() function to load it into the session.

I will start by installing data

```
install.packages("palmerpenguins")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(palmerpenguins)
```

now lets install tidyverse

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
```

```
## -- 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 errors
```

Viewing data

```
#View(penguins_raw)
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

```
head(penguins,4)
```

```
## # A tibble: 4 x 8
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
## 1 Adelie  Torgersen           39.1          18.7               181        3750
## 2 Adelie  Torgersen           39.5          17.4               186        3800
## 3 Adelie  Torgersen           40.3          18                 195        3250
## 4 Adelie  Torgersen           NA            NA                  NA          NA
## # i 2 more variables: sex <fct>, year <int>
```

```
colnames(penguins)
```

```
## [1] "species"           "island"            "bill_length_mm"
## [4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
## [7] "sex"               "year"
```

3. Data Cleaning

In this stage, I prepared the dataset for analysis and visualization by performing essential data cleaning steps. This included checking and correcting data types, identifying and handling missing values, and removing any duplicate rows. I also renamed the dataset to penguins_cleaned to reflect the cleaned version for further use in the project.

Firstly I will understand my dataset

```
colSums(is.na(penguins))
```

```
summary(penguins)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
```

```
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

now we see some na values which we need to clean and renaming the columns.

```
penguins_cleaned<-penguins %>%
  drop_na() %>%
  rename(bill_length=bill_length_mm,
         bill_depth=bill_depth_mm)
summary(penguins_cleaned)
```

```
##       species          island     bill_length      bill_depth
##  Adelie   :146   Biscoe   :163   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :123   1st Qu.:39.50   1st Qu.:15.60
##  Gentoo   :119   Torgersen: 47   Median :44.50   Median :17.30
##                                  Mean   :43.99   Mean   :17.16
##                                  3rd Qu.:48.60   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172       Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190       1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197       Median :4050                Median :2008
##  Mean   :201       Mean   :4207                Mean   :2008
##  3rd Qu.:213       3rd Qu.:4775                3rd Qu.:2009
##  Max.   :231       Max.   :6300                Max.   :2009
```

Check and convert data types By checking the data type we will get to know about the different data types of each column. If any data type is wrong then we can change it.

```
str(penguins_cleaned)
```

```
## tibble [333 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length      : num [1:333] 39.1 39.5 40.3 36.7 39.3 38.9 39.2 41.1 38.6 34.6 ...
##  $ bill_depth       : num [1:333] 18.7 17.4 18 19.3 20.6 17.8 19.6 17.6 21.2 21.1 ...
##  $ flipper_length_mm: int [1:333] 181 186 195 193 190 181 195 182 191 198 ...
##  $ body_mass_g      : int [1:333] 3750 3800 3250 3450 3650 3625 4675 3200 3800 4400 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 2 1 2 2 ...
##  $ year             : int [1:333] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

year is "int" i.e. integer data type. We will convert it into "date" data type using as.date function

```
penguins_cleaned$year <- as.Date(penguins_cleaned$year)
head(penguins_cleaned,4)
```

```
## # A tibble: 4 x 8
##   species island    bill_length bill_depth flipper_length_mm body_mass_g sex
##   <fct>   <fct>           <dbl>      <dbl>             <int>       <int> <fct>
## 1 Adelie  Torgersen        39.1       18.7               181        3750 male
## 2 Adelie  Torgersen        39.5       17.4               186        3800 female
## 3 Adelie  Torgersen        40.3       18                 195        3250 female
## 4 Adelie  Torgersen        36.7       19.3               193        3450 female
## # i 1 more variable: year <date>
```

Check for duplicates. True means that there is duplication in data and false means no duplication.

```
duplicated(penguins_cleaned)
```

there is no duplicate rows. But if there was any duplication we can use distinct function or unique to remove it.

```
distinct(penguins_cleaned,.keep_all= FALSE)
```

```
## # A tibble: 333 x 8
##    species island    bill_length bill_depth flipper_length_mm body_mass_g sex
##    <fct>   <fct>           <dbl>      <dbl>             <int>       <int> <fct>
##  1 Adelie  Torgersen        39.1       18.7               181        3750 male
##  2 Adelie  Torgersen        39.5       17.4               186        3800 female
##  3 Adelie  Torgersen        40.3       18                 195        3250 female
##  4 Adelie  Torgersen        36.7       19.3               193        3450 female
##  5 Adelie  Torgersen        39.3       20.6               190        3650 male
##  6 Adelie  Torgersen        38.9       17.8               181        3625 female
##  7 Adelie  Torgersen        39.2       19.6               195        4675 male
##  8 Adelie  Torgersen        41.1       17.6               182        3200 female
##  9 Adelie  Torgersen        38.6       21.2               191        3800 male
## 10 Adelie  Torgersen        34.6       21.1               198        4400 male
## # i 323 more rows
## # i 1 more variable: year <date>
```

```
# or use unique
# unique(penguins_cleaned, incomparables = FALSE)
```

no duplicate value was present in data so no row is removed.

Add/remove columns (mutate(), select()).

```
new<-mutate(penguins_cleaned, body_mass_kg=body_mass_g / 1000)
```

I gave the table a new name . To save the new column that I made. Now I will delete the old column

```
penguins_cleaned<- new %>%
  select(-body_mass_g)
head(penguins_cleaned,4)
```

```
## # A tibble: 4 x 8
##   species island    bill_length bill_depth flipper_length_mm sex    year
##   <fct>   <fct>           <dbl>      <dbl>             <int> <fct>  <date>
## 1 Adelie  Torgersen        39.1       18.7               181 male   1975-07-01
## 2 Adelie  Torgersen        39.5       17.4               186 female 1975-07-01
## 3 Adelie  Torgersen        40.3       18                 195 female 1975-07-01
## 4 Adelie  Torgersen        36.7       19.3               193 female 1975-07-01
## # i 1 more variable: body_mass_kg <dbl>
```

```
#View(penguins_cleaned)
```

Use pipes (%>%) for chaining operations. And using nested functions like filter(arrange (penguin))

```
#nested functions

filtered <- arrange ( filter(penguins_cleaned, island=="Torgersen" ),year )
summary(filtered)
```

```
##       species          island    bill_length      bill_depth
##  Adelie   :47   Biscoe    : 0   Min.   :33.50   Min.   :15.90
```

```
##  Chinstrap: 0   Dream    : 0   1st Qu.:36.65   1st Qu.:17.45
##  Gentoo   : 0   Torgersen:47   Median :39.00   Median :18.40
##                                Mean   :39.04   Mean   :18.45
##                                3rd Qu.:41.10   3rd Qu.:19.25
##                                Max.   :46.00   Max.   :21.50
##  flipper_length_mm     sex          year             body_mass_kg
##  Min.   :176.0     female:24   Min.   :1975-07-01   Min.   :2.900
##  1st Qu.:187.5     male  :23   1st Qu.:1975-07-01   1st Qu.:3.337
##  Median :191.0                 Median :1975-07-02   Median :3.700
##  Mean   :191.5                 Mean   :1975-07-02   Mean   :3.709
##  3rd Qu.:195.5                 3rd Qu.:1975-07-03   3rd Qu.:4.000
##  Max.   :210.0                 Max.   :1975-07-03   Max.   :4.700
```

```
head(filtered)
```

```
## # A tibble: 6 x 8
##   species island    bill_length bill_depth flipper_length_mm sex    year
##   <fct>   <fct>           <dbl>      <dbl>             <int> <fct>  <date>
## 1 Adelie  Torgersen        39.1       18.7               181 male   1975-07-01
## 2 Adelie  Torgersen        39.5       17.4               186 female 1975-07-01
## 3 Adelie  Torgersen        40.3       18                 195 female 1975-07-01
## 4 Adelie  Torgersen        36.7       19.3               193 female 1975-07-01
## 5 Adelie  Torgersen        39.3       20.6               190 male   1975-07-01
## 6 Adelie  Torgersen        38.9       17.8               181 female 1975-07-01
## # i 1 more variable: body_mass_kg <dbl>
```

```
#we  can  do  the same  with pipes
filtered_data<- penguins_cleaned %>%
  filter(island=="Biscoe") %>%
  arrange(year)
head(filtered_data)
```

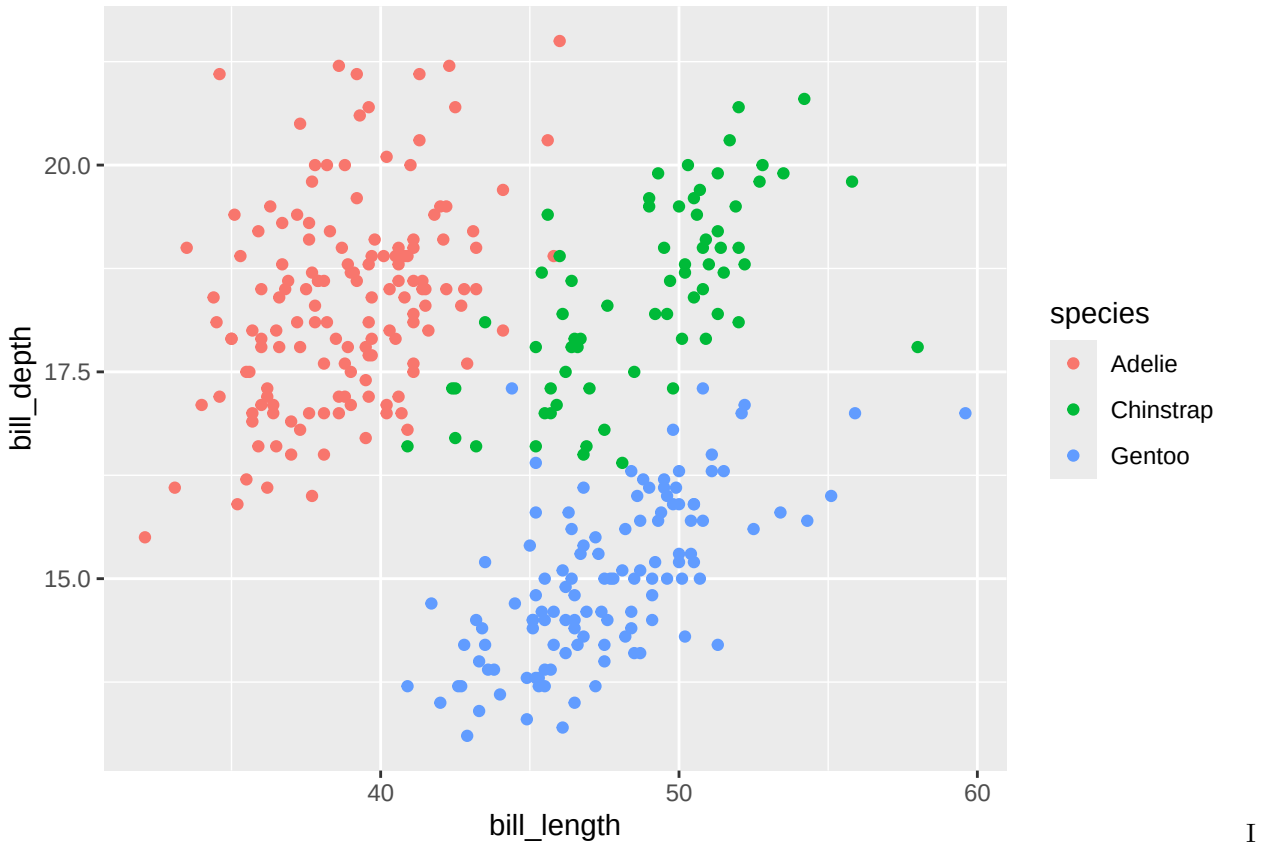
```
## # A tibble: 6 x 8
##   species island bill_length bill_depth flipper_length_mm sex    year
##   <fct>   <fct>        <dbl>      <dbl>             <int> <fct>  <date>
## 1 Adelie  Biscoe        37.8       18.3               174 female 1975-07-01
## 2 Adelie  Biscoe        37.7       18.7               180 male   1975-07-01
## 3 Adelie  Biscoe        35.9       19.2               189 female 1975-07-01
## 4 Adelie  Biscoe        38.2       18.1               185 male   1975-07-01
## 5 Adelie  Biscoe        38.8       17.2               180 male   1975-07-01
## 6 Adelie  Biscoe        35.3       18.9               187 female 1975-07-01
## # i 1 more variable: body_mass_kg <dbl>
```

4.Data Visualization

In this stage, I worked with the penguins_cleaned dataset to practice creating visualizations using the ggplot2 package. I created a few charts to explore different relationships in the data and experimented with various features such as aesthetics, geometries (geom_*), faceting, labels, and annotations to enhance the clarity and presentation of the plots.

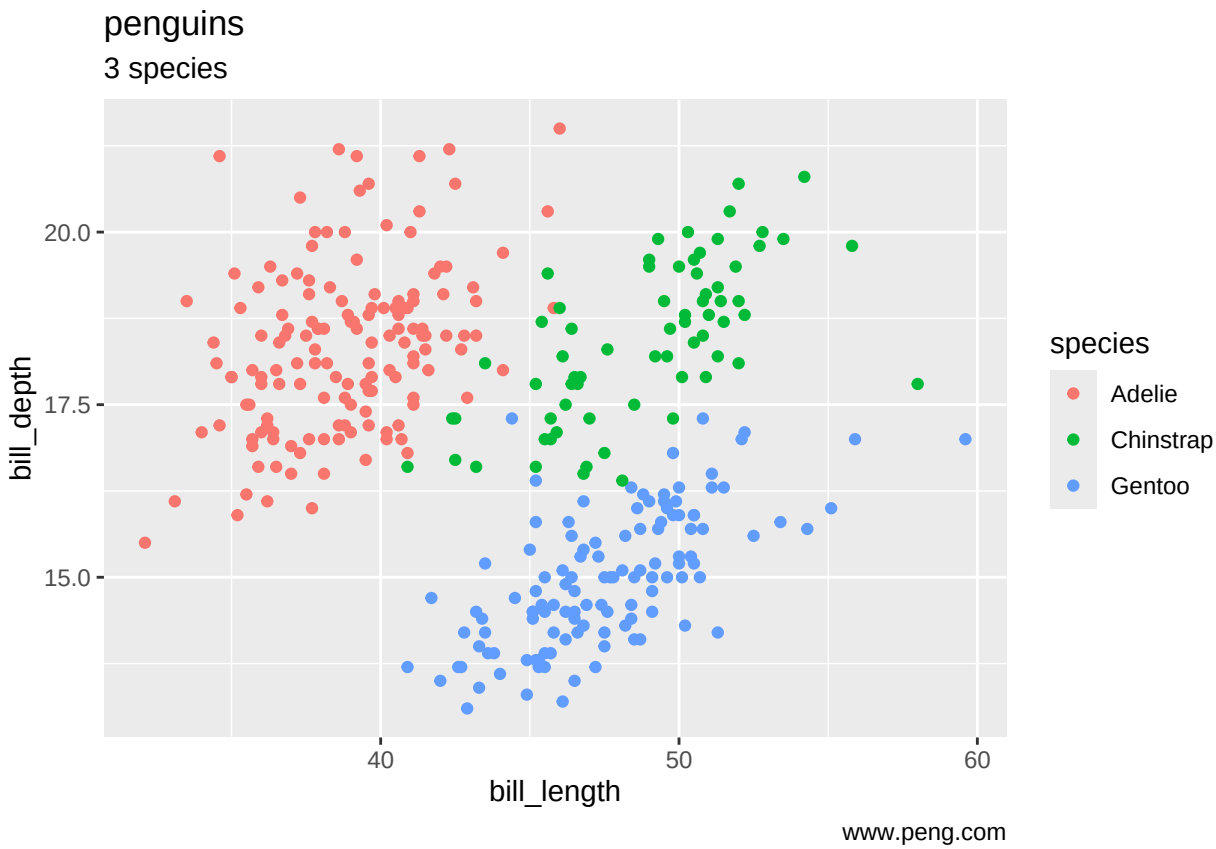
```
library(ggplot2)
```

```
chart1<- ggplot(penguins_cleaned,mapping = aes(x=bill_length,y=bill_depth,color=species) )+geom_point()
chart1
```



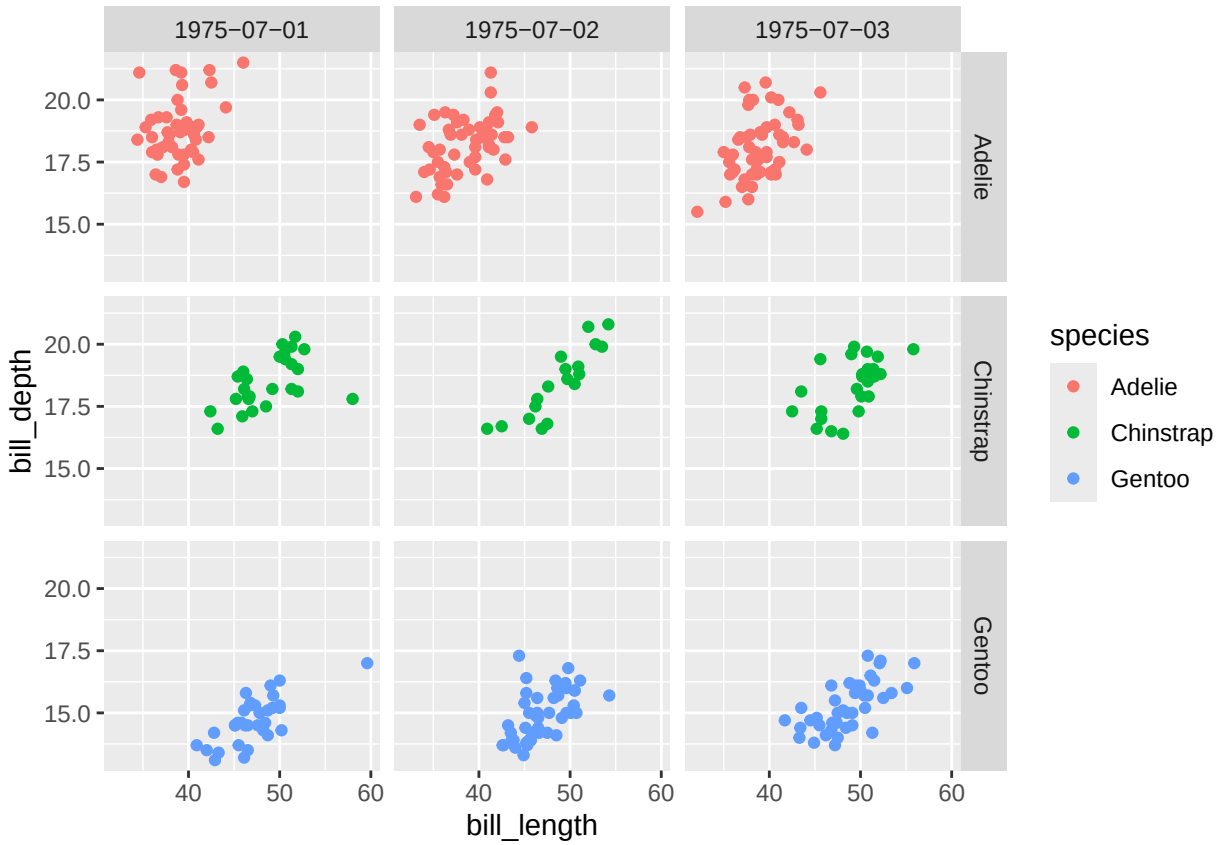
I have created a simple scattered plot chart.I have named this chart as “chart1” to refer in coming up steps. I can add so many more elements in this. ###Add axis labels, titles, and caption in this chart.

```
chart1+labs(title = "penguins",caption = "www.peng.com",subtitle = "3 species" )
```



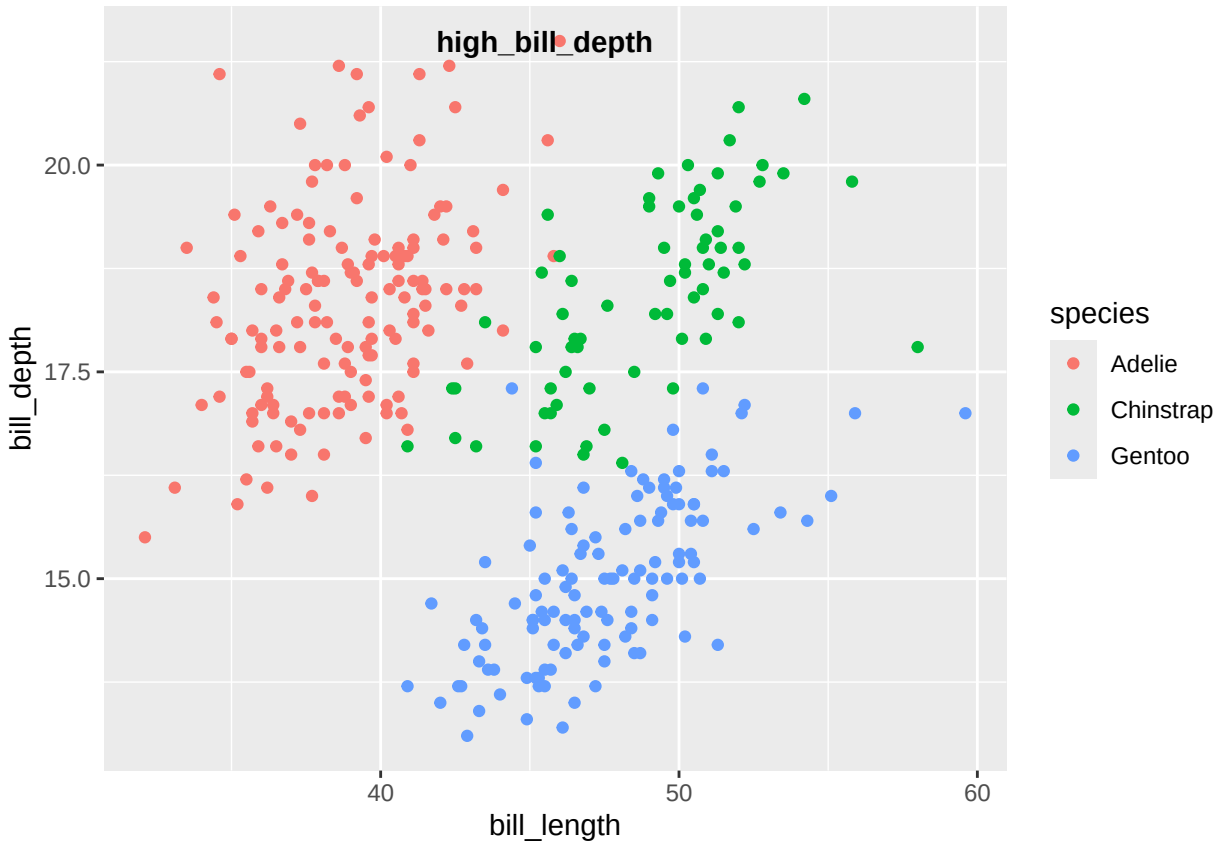
Use facet_wrap() to compare categories.

```
chart1+facet_grid(species~year)
```



now lets add a annotation

```
chart1+annotate('text',x=45.5,y=21.5,label="high_bill_depth",fontface = 'bold')
```



5. Summary Statistics

In this section, I used summarizing tools such as group_by() and summarise() to generate key insights from the dataset. These functions allowed me to calculate group-wise statistics, such as minimum and maximum values of bill length across different islands, helping me better understand the structure and patterns within the data.

```
penguins_cleaned %>%
group_by(island) %>%
 drop_na() %>%
summarise(max_bill_length= max(bill_length), min_bill_length= min(bill_length))
```

```
## # A tibble: 3 x 3
##   island    max_bill_length min_bill_length
##   <fct>               <dbl>           <dbl>
## 1 Biscoe               59.6            34.5
## 2 Dream                58              32.1
## 3 Torgersen            46              33.5
```

8. Summary & Learnings

This project took me through the complete data analysis workflow — from installing and loading a data set to cleaning, transforming, and visualizing the data using R. It was a great experience that helped reinforce both my technical and documentation skills.

I'm especially proud of the following:

- Learning how to document my work clearly using RMarkdown
- Practicing clean and readable code through tidyverse tools like dplyr and ggplot2
- Presenting my analysis in a structured, professional format

As I continue learning, I plan to add more insights and observations from the data set, along with additional visualizations or statistical summaries.