STT461 S25 IA23 - Data Manipulation

Kaiwen Jiang

Use dplyr package

With a random seed of 300, randomly draw 10 sample points from $N(0, 1)$, and calculate their mean.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(300)
mean(rnorm(10))
```

```
## [1] 0.7674236
```

Load in the data PenguinsClean.rda.

```
load("PenguinsClean.rda")
```

Filter: select observations

Filter the observations that have bill length over 39mm, bill depth over 19mm, and live on Dream Island, save as data1. How many observations are there in data1?

```
data1 <- filter(penguins, bill_length_mm > 39, bill_depth_mm > 19, island == 'Dream')
nrow(data1)
```

```
## [1] 26
```

Mutate: add new variables

Add a new column called body_mass_kg to data1, turn body mass from grams to kilograms. name the new dataset data2.

```
data2 <- mutate(data1, body_mass_kg = body_mass_g / 1000)
```

Group by and summarize

Group the penguins in data2 by their species. Name the new groupby data gb1.

```
gb1 <- group_by(data2, species)
```

Calculate the count, the mean bill length, bill depth, flipper length and body mass (kg) for each species. Save the data as sum1. Print it out.

```
sum1 <- summarise(gb1, n = n(), meanbill_len_mm = mean(bill_length_mm), meanbill_depth_mm = mean(bill_d
sum1
```

```
## # A tibble: 2 x 6
##   species      n meanbill_len_mm meanbill_depth_mm meanflip_len meanbody_mass_kg
##   <fct>    <int>           <dbl>             <dbl>        <dbl>            <dbl>
## 1 Adelie       6            41.2              20.2         194.             4.15
## 2 Chinstr~    20            51.2              19.8         201.             3.97
```

Calculate the proportion of each species, add it to the data sum1 as a new variable called freq. Save the new data as sum2. Print it out.

```
sum2 <- mutate(sum1, freq = n/sum(n))
sum2
```

```
## # A tibble: 2 x 7
##   species      n meanbill_len_mm meanbill_depth_mm meanflip_len meanbody_mass_kg
##   <fct>    <int>           <dbl>             <dbl>        <dbl>            <dbl>
## 1 Adelie       6            41.2              20.2         194.             4.15
## 2 Chinstr~    20            51.2              19.8         201.             3.97
## # i 1 more variable: freq <dbl>
```

Join

Join the sum2 data right to data2. Call it jt1. How many rows and columns are there in jt1?

```
jt1 <- inner_join(data2,sum2, by = 'species')
jt1
```

```
## # A tibble: 26 x 15
##    species   island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>     <fct>           <dbl>         <dbl>             <int>       <int>
##  1 Adelie    Dream            39.2          21.1               196        4150
##  2 Adelie    Dream            39.8          19.1               184        4650
##  3 Adelie    Dream            44.1          19.7               196        4400
##  4 Adelie    Dream            42.3          21.2               191        4150
##  5 Adelie    Dream            41.3          20.3               194        3550
##  6 Adelie    Dream            40.2          20.1               200        3975
##  7 Chinstrap Dream            50            19.5               196        3900
```

```
##  8 Chinstrap Dream            51.3          19.2               193        3650
##  9 Chinstrap Dream            52.7          19.8               197        3725
## 10 Chinstrap Dream            51.3          19.9               198        3700
## # i 16 more rows
## # i 9 more variables: sex <fct>, year <int>, body_mass_kg <dbl>, n <int>,
## #   meanbill_len_mm <dbl>, meanbill_depth_mm <dbl>, meanflip_len <dbl>,
## #   meanbody_mass_kg <dbl>, freq <dbl>
```

```
dim(jt1)
```

```
## [1] 26 15
```

Pipe

Use pipe to replicate the process above. Save the data as jt2. How many rows and columns are there in jt2.

```
jt2 <- penguins %>% filter(bill_length_mm >39, bill_depth_mm >19, island == 'Dream') %>%
  mutate(body_mass_kg = body_mass_g / 1000) %>%  #data 2
  inner_join(penguins %>% filter(bill_length_mm >39, bill_depth_mm >19, island == 'Dream') %>% #piped i
  mutate(body_mass_kg = body_mass_g / 1000) %>%
  group_by(species) %>%
  summarise(n = n(), meanbill_len_mm = mean(bill_length_mm), meanbill_depth_mm = mean(bill_depth_mm), m
  mutate(freq = n/sum(n)), by = 'species')
jt2
```

```
## # A tibble: 26 x 15
##    species   island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>     <fct>           <dbl>         <dbl>             <int>       <int>
##  1 Adelie    Dream            39.2          21.1               196        4150
##  2 Adelie    Dream            39.8          19.1               184        4650
##  3 Adelie    Dream            44.1          19.7               196        4400
##  4 Adelie    Dream            42.3          21.2               191        4150
##  5 Adelie    Dream            41.3          20.3               194        3550
##  6 Adelie    Dream            40.2          20.1               200        3975
##  7 Chinstrap Dream            50            19.5               196        3900
##  8 Chinstrap Dream            51.3          19.2               193        3650
##  9 Chinstrap Dream            52.7          19.8               197        3725
## 10 Chinstrap Dream            51.3          19.9               198        3700
## # i 16 more rows
## # i 9 more variables: sex <fct>, year <int>, body_mass_kg <dbl>, n <int>,
## #   meanbill_len_mm <dbl>, meanbill_depth_mm <dbl>, meanflip_len <dbl>,
## #   meanbody_mass_kg <dbl>, freq <dbl>
```

Use ggplot2 package

Load in (or install first) ggplot2 package.

```
library(ggplot2)
```

Load in the data PenguinsClean.rda.

```
load("PenguinsClean.rda")
```

Set up the canvas

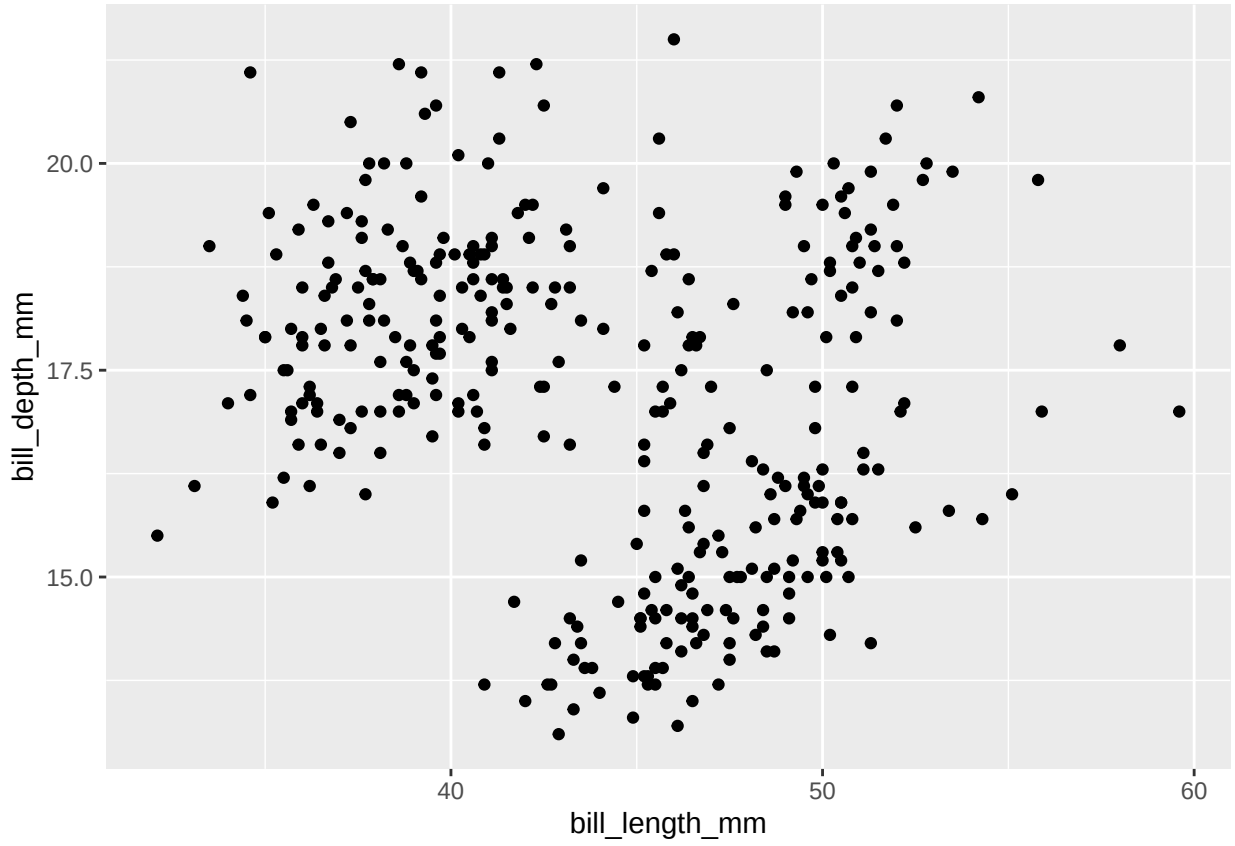
Try the ggplot() function. What do you see?

```
ggplot(data = penguins)
```


Add geom features.

Add a scatterplot with bill_length as x-axis and bill_depth as y-axis.

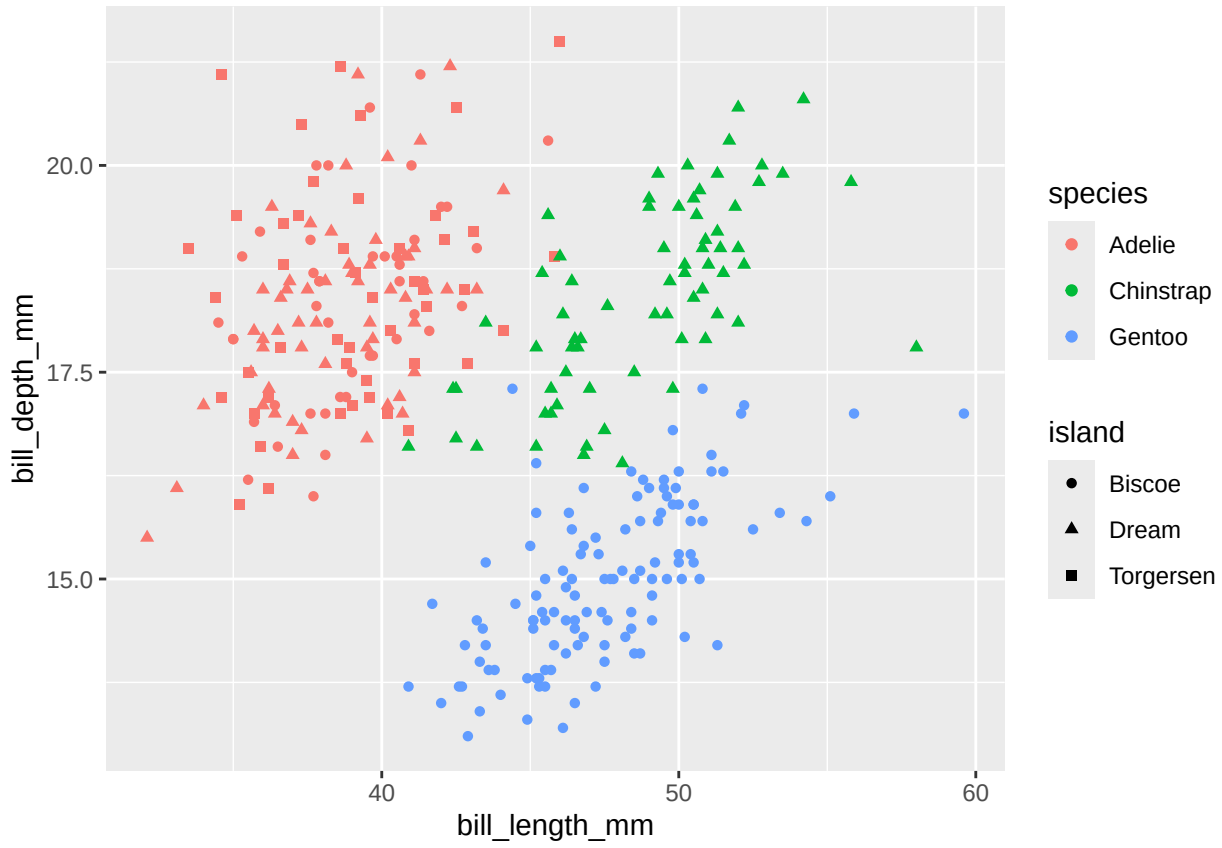
```
ggplot(data = penguins)+
  geom_point(mapping = aes(x = bill_length_mm, y= bill_depth_mm))
```



Edit aesthetics

Use color to represent the species, and shape of the dots to represent the inhabited islands.

```
ggplot(data = penguins)+
  geom_point(mapping = aes(x = bill_length_mm, y= bill_depth_mm, colour = species, shape = island))
```

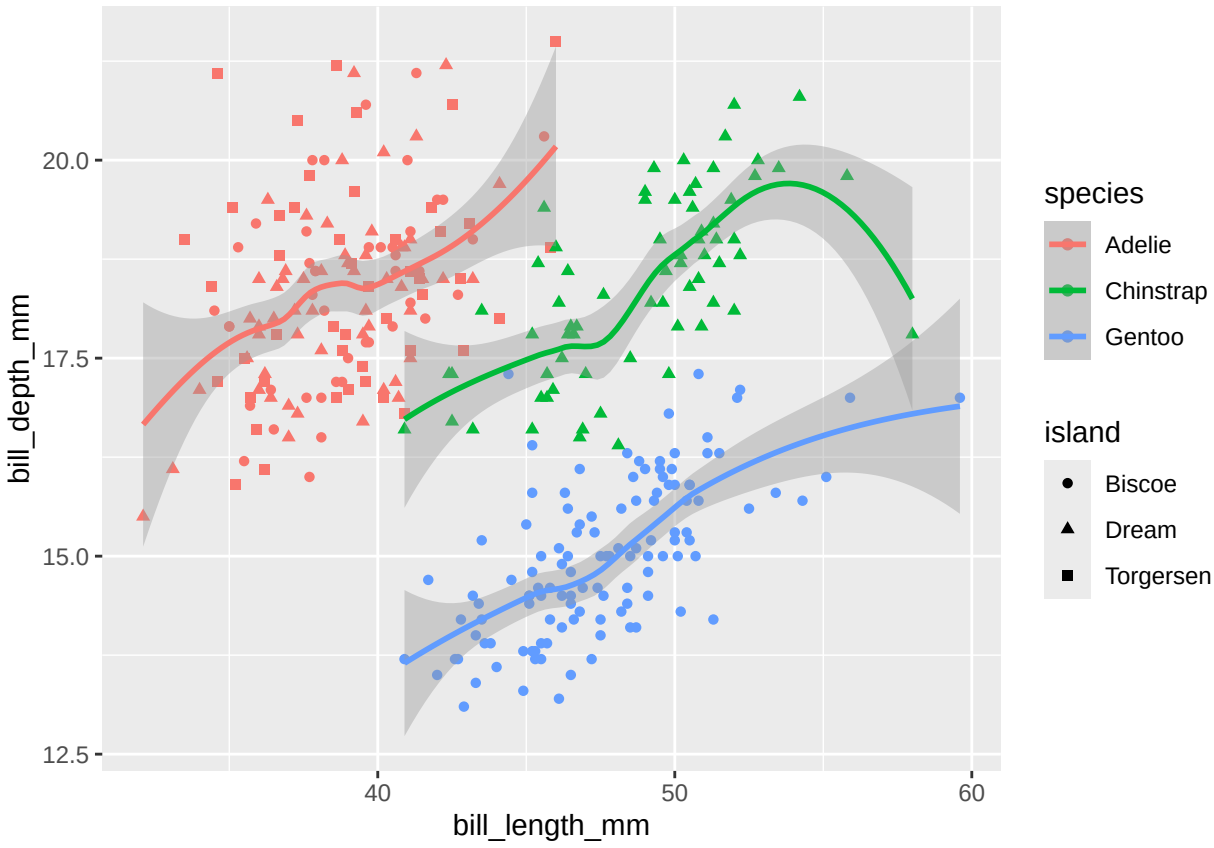


Layers in ggplot2 2

Add fitted trend lines to the graph with geom_smooth().

```
ggplot(data = penguins)+
  geom_point(mapping = aes(x = bill_length_mm, y= bill_depth_mm, colour = species, shape = island))+
  geom_smooth(aes(x = bill_length_mm, y= bill_depth_mm, colour = species))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


bill_depth_mm
20.0
17.5
15.0
12.5
40
50
60
bill_length_mm
species
Adelie
Chinstrap
Gentoo
island
Biscoe
Dream
Torgersen