HW4_DATA412

Shengling Hu

2/21/2022

Load relative package and the dataset

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(nycflights13)
data("flights")
help(flights)
head(flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## 4  2013     1     1      544            545        -1     1004           1022
## 5  2013     1     1      554            600        -6      812            837
## 6  2013     1     1      554            558        -4      740            728
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
nrow(flights)
```

```
## [1] 336776
```

```r
flights%>%slice(1:3)
```

```
## # A tibble: 3 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Worst Plane to Fly

```r
flights %>%
  group_by(tailnum) %>%
  summarize(mean_dep = mean(dep_delay, na.rm = TRUE), n = n())%>%
  arrange(desc(mean_dep)) %>%# the three worst planes with their mean average depature dely record
  slice(1:3)
```

```
## # A tibble: 3 x 3
##   tailnum mean_dep     n
##   <chr>      <dbl> <int>
## 1 N844MH       297     1
## 2 N922EV       274     1
## 3 N587NW       272     1
```

The worst three tailnums are N844MH, N922EV, N1587NW, they all just made one trip

The worst three tailnums and which flew more than 15 trips

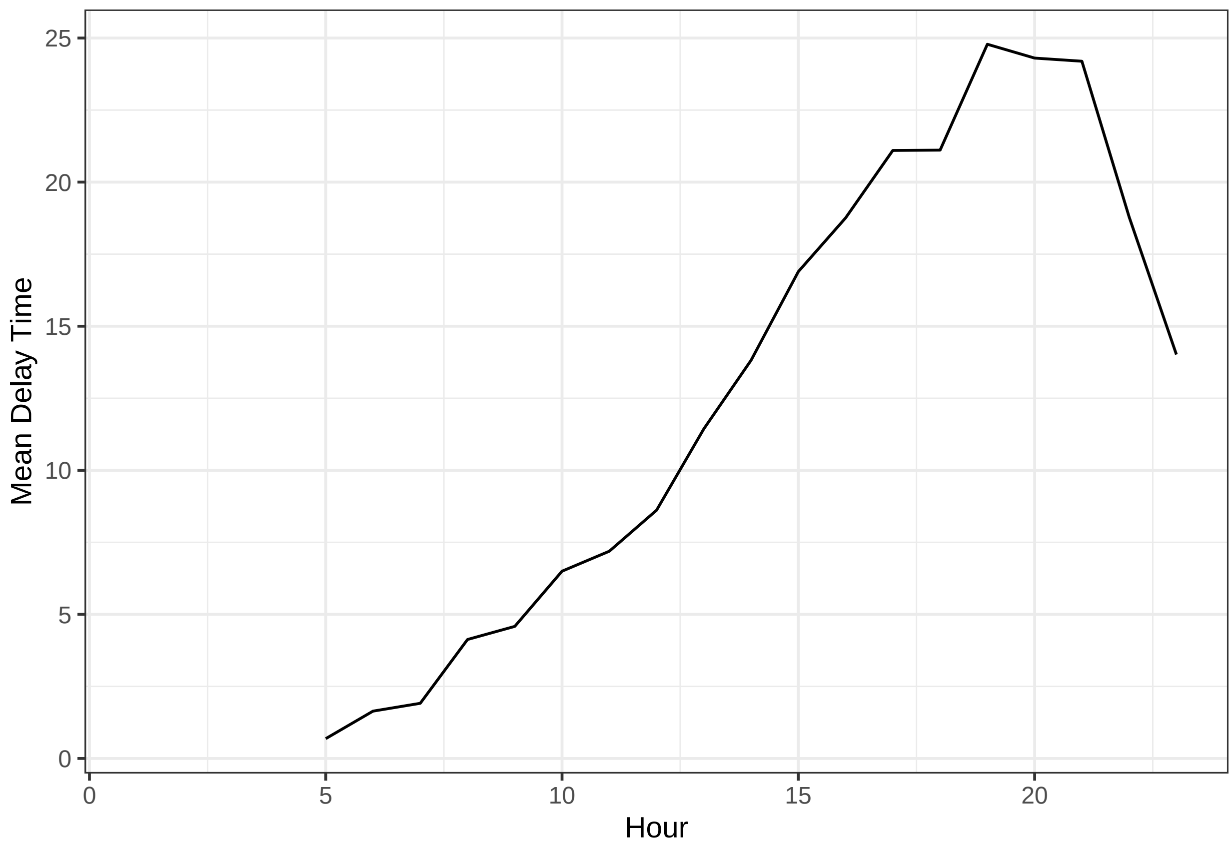
```r
flights %>%
  group_by(tailnum) %>%
  summarize(mean_dep = mean(dep_delay, na.rm = TRUE), n = n())%>%
  filter(n>15)%>%
  arrange(desc(mean_dep)) %>%# the three worst planes with their mean average depature dely record
  slice(1:3)
```

```
## # A tibble: 3 x 3
##   tailnum mean_dep     n
##   <chr>      <dbl> <int>
## 1 N184DN      54.7    16
## 2 N203FR      53.5    41
## 3 N645MQ      52.5    25
```

Best Time of Day to Fly

```
flights %>%
  group_by(hour) %>%
  summarize(mean_dep = mean(dep_delay, na.rm = TRUE), n = n())%>%
  ggplot(aes(x = hour, y = mean_dep)) +
  geom_line() +
  theme_bw() +
  xlab("Hour") +
  ylab("Mean Delay Time")
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



#The time I need to choose is 5:00 am.

#Worst Trips for each Destination

```
flights%>%
  group_by(dest)%>%
  mutate(total_de = sum(dep_delay, na.rm = TRUE))%>%
  mutate(pro_de=dep_delay/total_de, na.rm = TRUE)%>%
  arrange(dest, desc(pro_de))%>%
  select(year, month, day, dest, flight, pro_de, total_de)%>%
  filter(pro_de == max(pro_de, na.rm=TRUE))
```

```
## Warning in max(pro_de, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```

```
## # A tibble: 106 x 7
## # Groups:   dest [104]
##     year month   day dest  flight  pro_de total_de
##    <int> <int> <int> <chr>  <int>   <dbl>    <dbl>
##  1  2013    12    14 ABQ       65 0.0407      3490
##  2  2013     7    23 ACK     1491 0.128       1711
##  3  2013     1    25 ALB     4309 0.0326      9897
##  4  2013     8    17 ANC      887 0.728        103
##  5  2013     7    22 ATL     2047 0.00425   211391
##  6  2013     7    10 AUS      503 0.0111     31496
##  7  2013     6    14 AVL     4519 0.103       2154
##  8  2013     2    21 BDL     4103 0.0345      7301
##  9  2013    12     1 BGR     5309 0.0354      7011
## 10  2013     4    10 BHM     5038 0.0402      8077
## # ... with 96 more rows
```

The total delay >= 0.

```
flights%>%
  group_by(dest)%>%
  mutate(total_de = sum(dep_delay, na.rm = TRUE))%>%
  filter(total_de > 0)%>%
  mutate(pro_de=dep_delay/total_de, na.rm = TRUE)%>%
  arrange(dest, desc(pro_de))%>%
  select(year, month, day, dest, flight, pro_de, total_de)%>%
  filter(pro_de == max(pro_de, na.rm=TRUE))
```

```
## # A tibble: 102 x 7
## # Groups:   dest [102]
##     year month   day dest  flight  pro_de total_de
##    <int> <int> <int> <chr>  <int>   <dbl>    <dbl>
##  1  2013    12    14 ABQ       65 0.0407      3490
##  2  2013     7    23 ACK     1491 0.128       1711
##  3  2013     1    25 ALB     4309 0.0326      9897
##  4  2013     8    17 ANC      887 0.728        103
##  5  2013     7    22 ATL     2047 0.00425   211391
##  6  2013     7    10 AUS      503 0.0111     31496
##  7  2013     6    14 AVL     4519 0.103       2154
##  8  2013     2    21 BDL     4103 0.0345      7301
##  9  2013    12     1 BGR     5309 0.0354      7011
## 10  2013     4    10 BHM     5038 0.0402      8077
## # ... with 92 more rows
```

find the worst flight number

```
flights%>%
  group_by(flight)%>%
  mutate(total_de = sum(dep_delay, na.rm = TRUE))%>%
  mutate(pro_de=dep_delay/total_de, na.rm = TRUE)%>%
  arrange(dest, desc(pro_de))%>%
  select(year, month, day, dest, flight, pro_de, total_de)%>%
  filter(pro_de == max(pro_de))
```

```
## # A tibble: 1,994 x 7
## # Groups:   flight [1,902]
##     year month   day dest  flight pro_de total_de
##    <int> <int> <int> <chr>  <int>  <dbl>    <dbl>
##  1  2013     5    30 ACK     1191  0.170      595
##  2  2013    12    31 ALB     4551  1.33        15
##  3  2013     7     2 ALB     6041  1           -3
##  4  2013    11    17 ALB     4550  0.911      157
##  5  2013     3     9 ALB     4263  0.646       48
##  6  2013     6     8 ALB     5963  0.532      201
##  7  2013    12    19 ALB     4470  0.393      178
##  8  2013     1    12 ATL     1713  7.33         6
##  9  2013    12    18 ATL     1716  5.33         3
## 10  2013    11     2 ATL     1358  2.86        21
## # ... with 1,984 more rows
```

```
flights %>%
  group_by(dest)%>%
  summarize(total_car = n_distinct(carrier), na.rm = TRUE)%>%
  filter(total_car>3)%>%
  arrange(total_car)
```

```
## # A tibble: 38 x 3
##    dest  total_car na.rm
##    <chr>     <int> <lgl>
##  1 BUF           4 TRUE
##  2 BWI           4 TRUE
##  3 CHS           4 TRUE
##  4 CVG           4 TRUE
##  5 DFW           4 TRUE
##  6 FLL           4 TRUE
##  7 JAX           4 TRUE
##  8 MCO           4 TRUE
##  9 MKE           4 TRUE
## 10 RSW           4 TRUE
## # ... with 28 more rows
```

The airpots are BOS, CLT, ORD, TPA

Calculate the daily correlation for each airpot

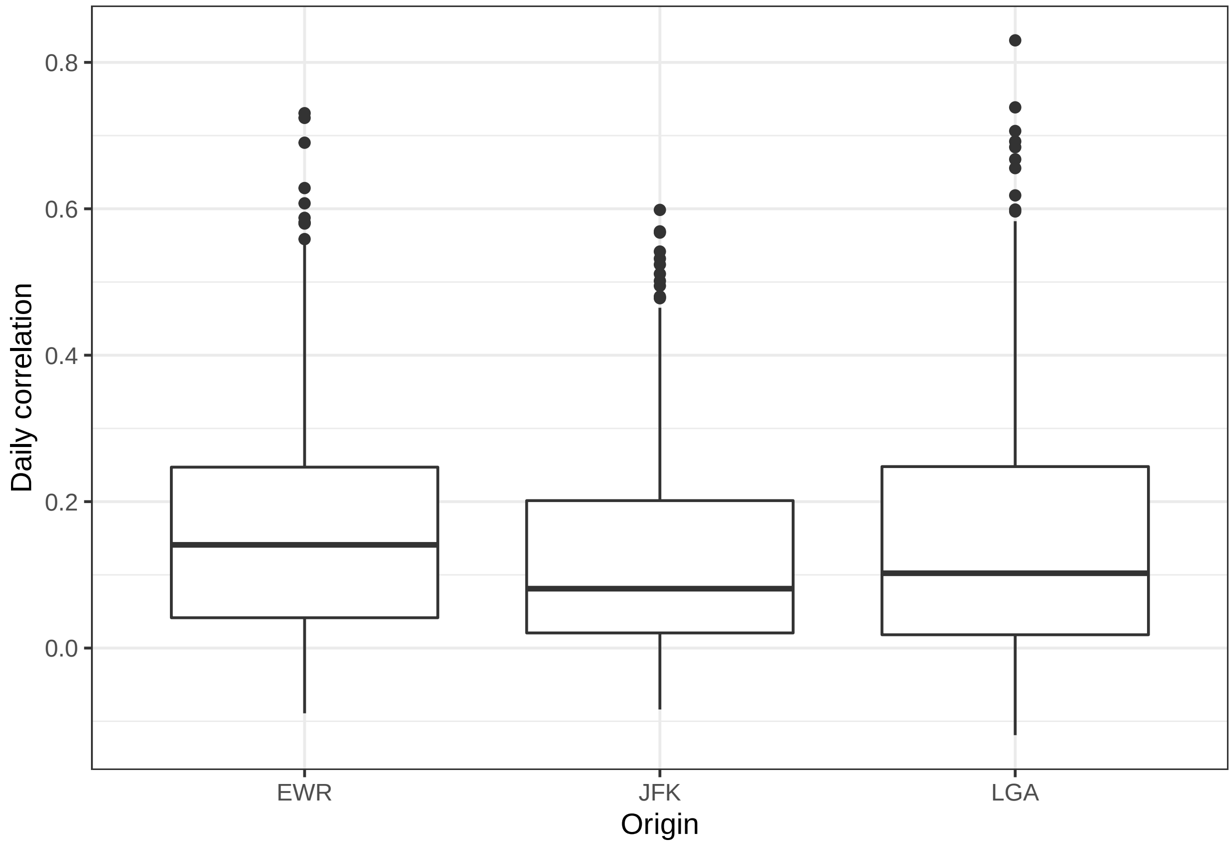
```
flights %>%
  group_by(origin,year, month,day) %>%
  arrange(dep_time) %>%
  mutate(lagdep_delay = lag(dep_delay))->new_flights

new_flights %>%
  group_by(origin,year, month,day)%>%
  arrange(dep_time) %>%
  summarize(cor_delay = cor(lagdep_delay, dep_delay, use = "pairwise.complete.obs"))
```

```
## `summarise()` has grouped output by 'origin', 'year', 'month'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 1,095 x 5
## # Groups:   origin, year, month [36]
##    origin  year month   day cor_delay
##    <chr>  <int> <int> <int>     <dbl>
##  1 EWR     2013     1     1    0.247
##  2 EWR     2013     1     2    0.105
##  3 EWR     2013     1     3    0.0761
##  4 EWR     2013     1     4    0.124
##  5 EWR     2013     1     5   -0.0630
##  6 EWR     2013     1     6    0.0335
##  7 EWR     2013     1     7    0.0606
##  8 EWR     2013     1     8    0.0288
##  9 EWR     2013     1     9    0.0246
## 10 EWR     2013     1    10    0.0431
## # ... with 1,085 more rows
```

```
new_flights %>%
  group_by(origin,year, month,day)%>%
  arrange(dep_time) %>%
  summarize(cor_delay = cor(lagdep_delay, dep_delay, use = "pairwise.complete.obs"))%>%
  ggplot( mapping = aes(x = origin, y = cor_delay))+
  geom_boxplot()+
  theme_bw() +
  xlab("Origin") +
  ylab("Daily correlation")
```

```
## `summarise()` has grouped output by 'origin', 'year', 'month'. You can override
## using the `.groups` argument.
```




```
new_flights %>%
  group_by(origin,year, month,day)%>%
  arrange(dep_time) %>%
  summarize(cor_delay = cor(lagdep_delay, dep_delay, use = "pairwise.complete.obs"))%>%
  group_by(origin)%>%
  summarize(mean_daily_cor = mean(cor_delay, na.rm=TRUE),
            median_daily_cor=median(cor_delay, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'origin', 'year', 'month'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 3 x 3
##   origin mean_daily_cor median_daily_cor
##   <chr>           <dbl>            <dbl>
## 1 EWR             0.165           0.141
## 2 JFK             0.126           0.0811
## 3 LGA             0.154           0.102
```

Based on the boxplot, we can find that all the distribution of the daily correlation for the three airpots is skewed to the right.

Combine the boxplot and the numerical summary, the airpot EWR has the higest average daily correlation between subsequent flight delays.

Part two

load the dataset

```
data("starwars")
```

determine the individuals have missing value and arranged in ascending order of height

```
starwars%>%
  filter(is.na(gender))%>%
  select(name, height)%>%
  arrange(height)
```

```
## # A tibble: 4 x 2
##   name           height
##   <chr>           <int>
## 1 Sly Moore         178
## 2 Ric Olié          183
## 3 Quarsh Panaka     183
## 4 Captain Phasma     NA
```

change the NA value into nonbinary

```
starwars%>%
  mutate(gender=replace(gender, is.na(gender), 'nonbinary'))->starwars
```

#calculate BMI

```
starwars%>%
  mutate(gender=replace(gender, is.na(gender), 'nonbinary'))%>%
  mutate(height_m=height/100,  na.rm = TRUE)%>%
  mutate(BMI=mass/(height_m^2),  na.rm = TRUE)->starwars
```

calculate mean and median for each gender

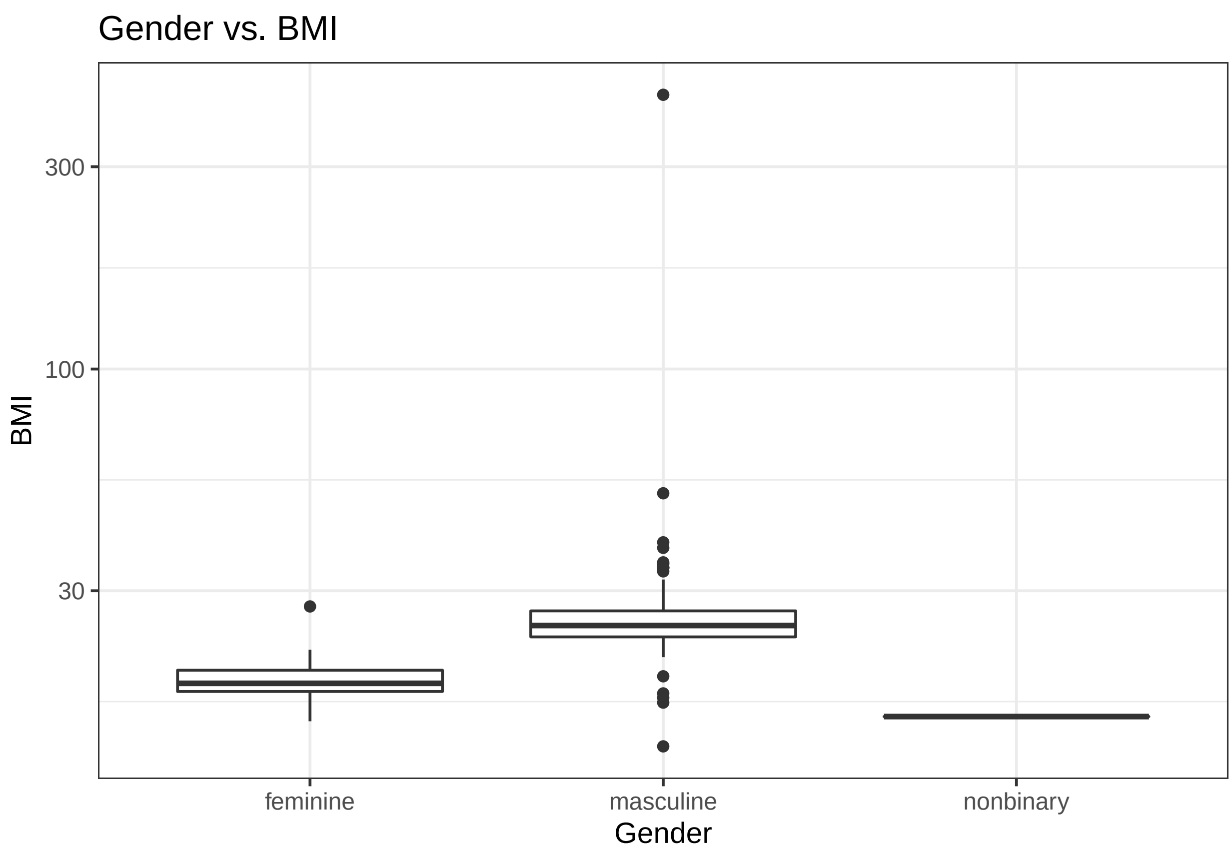
```
starwars%>%
  group_by(gender)%>%
  summarize(mean_height = mean(height, na.rm = TRUE), median_height = median(height, na.rm= TRUE), n=n()
```

```
## # A tibble: 3 x 4
##   gender    mean_height median_height     n
##   <chr>           <dbl>         <dbl> <int>
## 1 feminine         165.          166.    17
## 2 masculine        177.          183     66
## 3 nonbinary        181.          183      4
```

plot for BMI vs. gender

```
ggplot(data = starwars, mapping = aes(gender, BMI)) +
  geom_boxplot()+
  theme_bw()+
  scale_y_log10()+
  xlab("Gender") +
  ylab("BMI") +
  ggtitle("Gender vs. BMI")
```

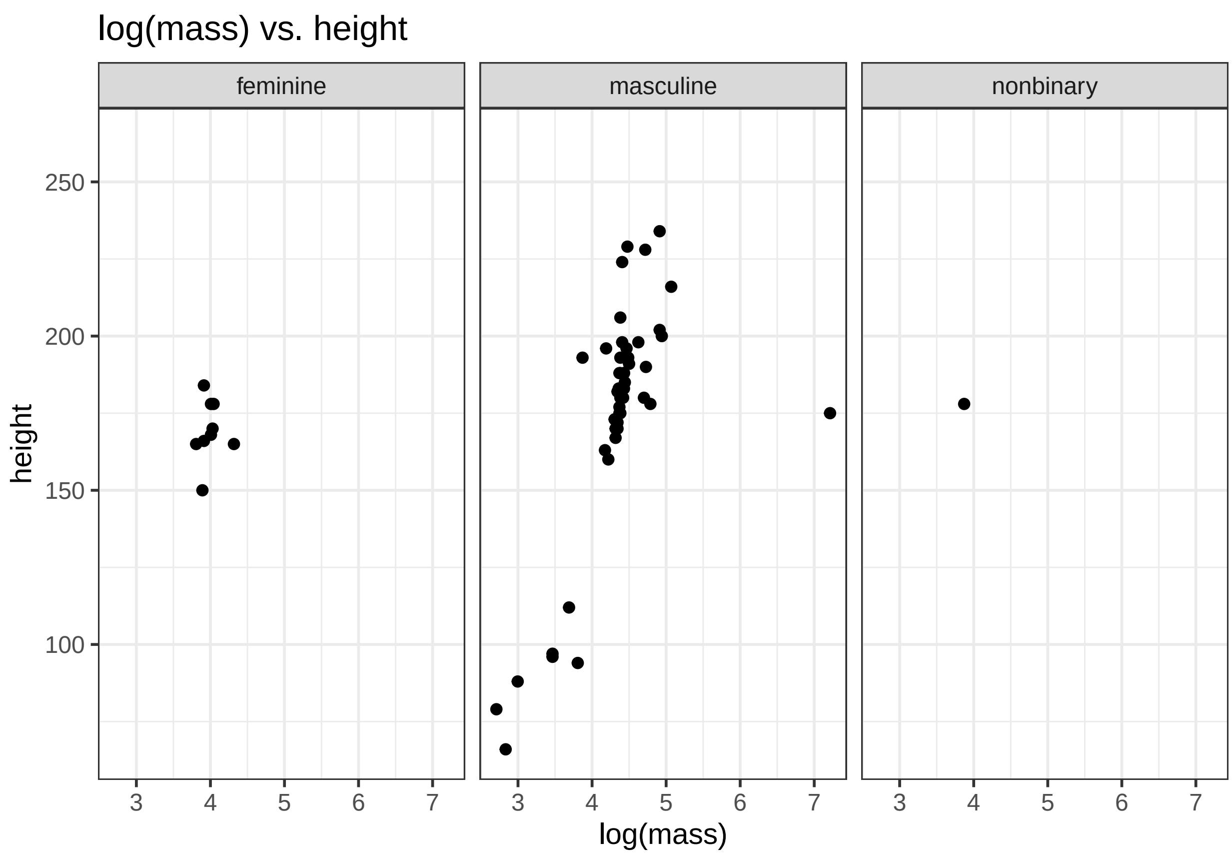
```
## Warning: Removed 28 rows containing non-finite values (stat_boxplot).
```



scatter plot for log(mass) vs. height, faceting by gender

```
ggplot(data = starwars, mapping = aes(x = log(mass) , y = height)) +
  geom_point() +
  theme_bw()+
  xlab("log(mass)") +
  ylab("height") +
  ggtitle("log(mass) vs. height")+
  facet_wrap( ~ gender)
```

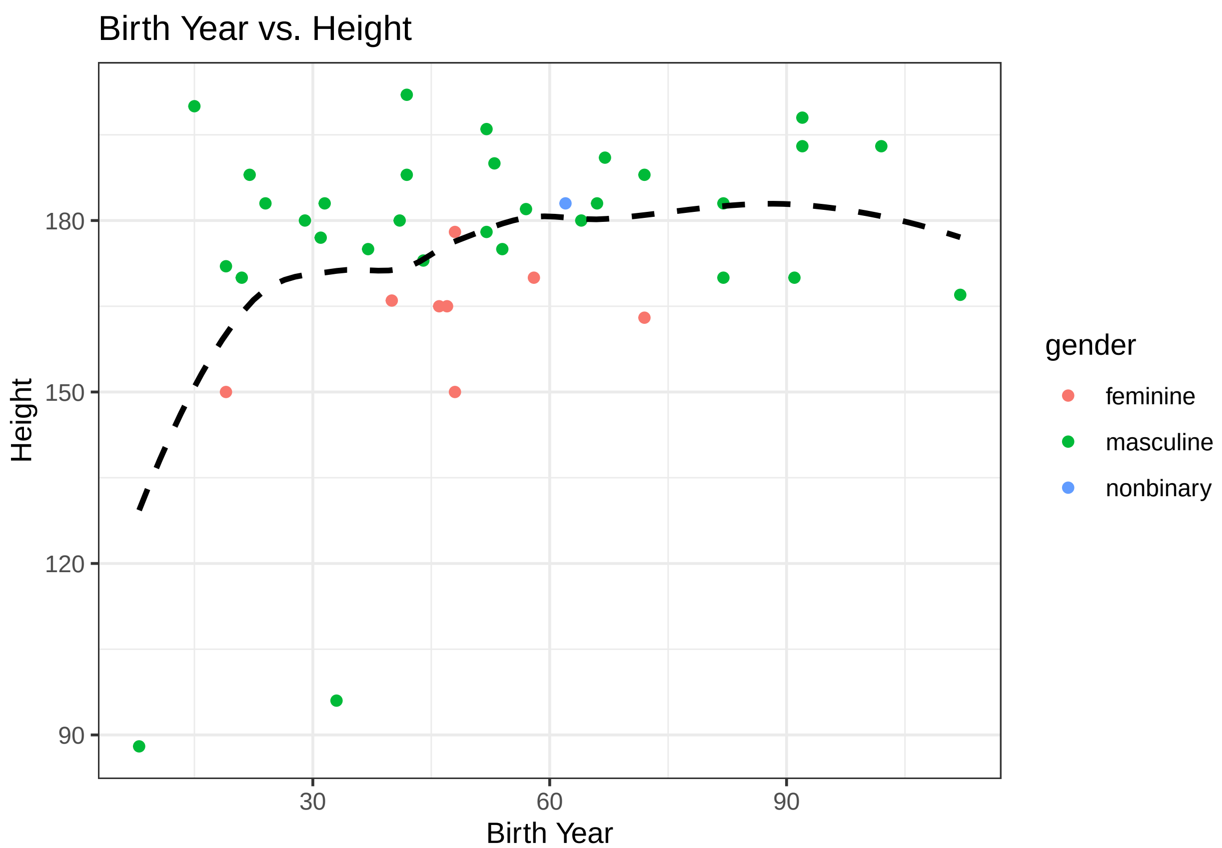
```
## Warning: Removed 28 rows containing missing values (geom_point).
```



reproduce the plot

```
starwars%>%
  filter(birth_year <= 150)%>%
  ggplot(mapping = aes(birth_year, height, color = gender)) +
  geom_point()+
  theme_bw()+
  xlab("Birth Year") +
  ylab("Height")+
  ggtitle("Birth Year vs. Height")+
  geom_smooth(aes(birth_year, height),color="black", linetype = "dashed", se = FALSE)
```

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



#load the dataset

```
library(remotes)
library(palmerpenguins)
data(package = 'palmerpenguins')
data('penguins')
```

#calculate the fb_ratio

```
penguins%>%
  mutate(max_billlen=max(bill_length_mm, na.rm = TRUE), fb_ratio=flipper_length_mm/max_billlen, na.rm=TI
```

#eliminate the NA value of fb.ratio, and show the highest four penguins of each sex

```
penguins%>%
  filter(!is.na(fb_ratio))%>%
  group_by(sex)%>%
  arrange(desc(fb_ratio))%>%
  slice(1:4)
```

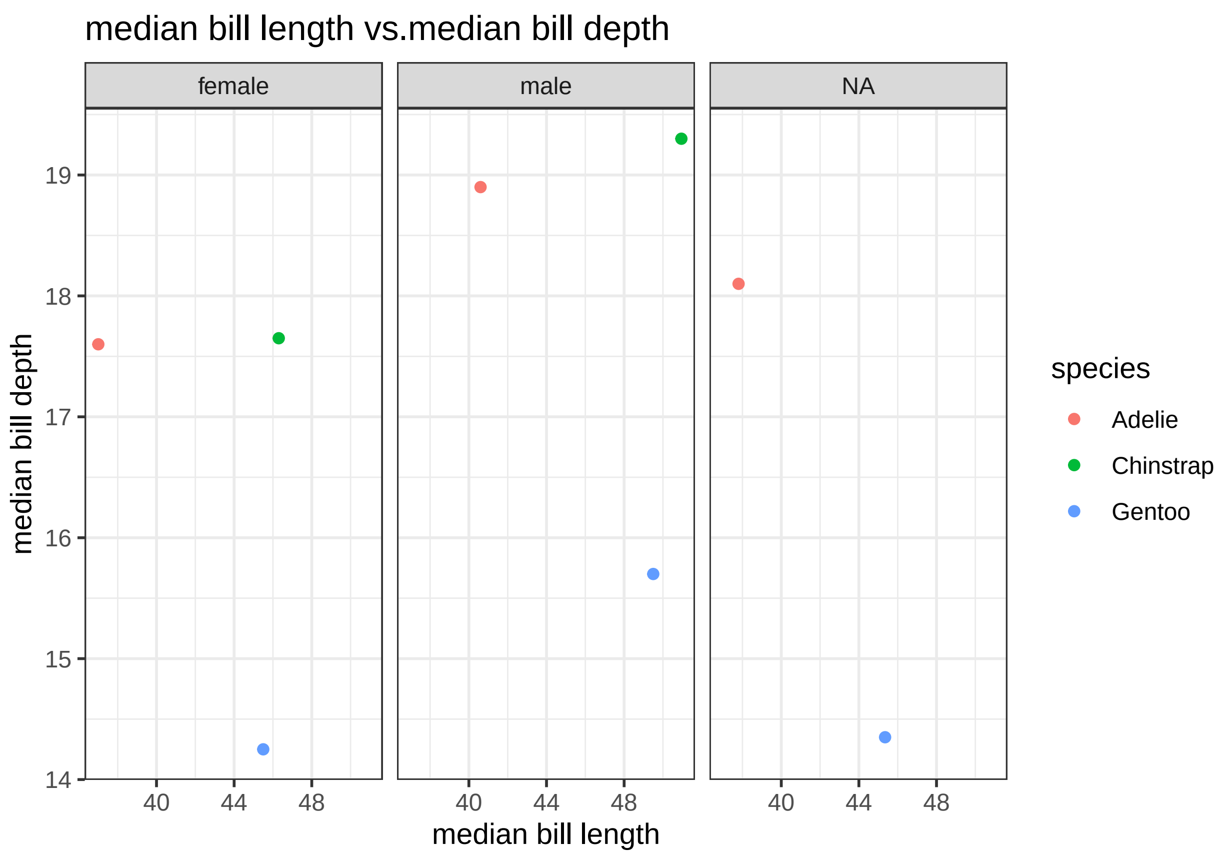
```
## # A tibble: 12 x 11
## # Groups:   sex [3]
##    species island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>           <dbl>         <dbl>             <int>       <int>
##  1 Gentoo  Biscoe           46.9          14.6               222        4875
```

```
##   2 Gentoo  Biscoe            49.1          14.8                 220         5150
##   3 Gentoo  Biscoe            43.5          14.2                 220         4700
##   4 Gentoo  Biscoe            45.8          14.2                 219         4700
##   5 Gentoo  Biscoe            54.3          15.7                 231         5650
##   6 Gentoo  Biscoe            50            16.3                 230         5700
##   7 Gentoo  Biscoe            59.6          17                   230         6050
##   8 Gentoo  Biscoe            49.8          16.8                 230         5700
##   9 Gentoo  Biscoe            44.5          15.7                 217         4875
##  10 Gentoo  Biscoe            44.5          14.3                 216         4100
##  11 Gentoo  Biscoe            47.3          13.8                 216         4725
##  12 Gentoo  Biscoe            46.2          14.4                 214         4650
## # ... with 5 more variables: sex <fct>, year <int>, max_billlen <dbl>,
## #   fb_ratio <dbl>, na.rm <lgl>
```

#For each species and sex, calculate the median of the numeric variables

```
penguins%>%
  group_by(species, sex)%>%
  summarize(median_bilen = median(bill_length_mm, na.rm = TRUE),median_bidep=median(bill_depth_mm,na.rm
  ggplot(mapping = aes(x = median_bilen , y = median_bidep, color=species)) +
  geom_point() +
  theme_bw()+
  xlab("median bill length") +
  ylab("median bill depth") +
  ggtitle("median bill length vs.median bill depth")+
  facet_wrap( ~ sex)
```

```
## `summarise()` has grouped output by 'species'. You can override using the
## `.groups` argument.
```



#The median bill depth values of male penguins for these three species are higher than female penguins for these three species

The total number of rows with no missing values

```
sum(!is.na(penguins))
```

```
## [1] 3763
```

#unique values for each of the columns that end in "_mm" for each sex

```
penguins%>%
  group_by(sex)%>%
  summarize(unique_bilen = n_distinct(bill_length_mm,na.rm = TRUE),
            unique_bidep = n_distinct(bill_depth_mm,na.rm = TRUE),
            unique_flip = n_distinct(flipper_length_mm,na.rm = TRUE))
```

```
## # A tibble: 3 x 4
##   sex    unique_bilen unique_bidep unique_flip
##   <fct>         <int>        <int>       <int>
## 1 female           97           56          41
## 2 male            110           58          49
## 3 <NA>              7            9           8
```