PS5

Q1.1

Q1.1.1

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(nycflights13)
library(dplyr)


head(flights,5)
```

```
## # A tibble: 5 × 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
## # ℹ 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>
```

Q1.1.2

a. There is no code more than three character
b. Also there is no code that contains digits

```
 #(a)
flights %>%
  filter(nchar(dest) !=3)
```

```
## # A tibble: 0 × 19
## # ℹ 19 variables: year <int>, month <int>, day <int>, dep_time <int>,
## #   sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   sched_arr_time <int>, 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>
```

```
#(b)
flights %>%
  filter(grepl(c(0.9), dest)) #I googled this function
```

```
## # A tibble: 0 × 19
## # ℹ 19 variables: year <int>, month <int>, day <int>, dep_time <int>,
## #   sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   sched_arr_time <int>, 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>
```

Q1.1.3

The total values are 336,776 and missing arr_delay values are 9430
Compare to the total values, looks fine

```
flights %>%
  filter(is.na(arr_delay)) %>%
  summarise(n())
```

```
## # A tibble: 1 × 1
##   `n()`
##   <int>
## 1  9430
```

Q1.1.4

The max delay is 1272min and the min delay is -86(assuming arriving early).
I heard about the delay that over a day therefore these delay data is plausible.

```
max(flights$arr_delay, na.rm = TRUE)
```

```
## [1] 1272
```

```
min(flights$arr_delay, na.rm = TRUE)
```

```
## [1] -86
```

Q1.1.5

```
flights %>%
  group_by(dest) %>%
  summarise(mean = mean(arr_delay, na.rm = TRUE)) %>%
  arrange(desc(mean)) %>%
  head(3)
```

```
## # A tibble: 3 × 2
##   dest   mean
##   <chr> <dbl>
## 1 CAE    41.8
## 2 TUL    33.7
## 3 OKC    30.6
```

Q1.1.6

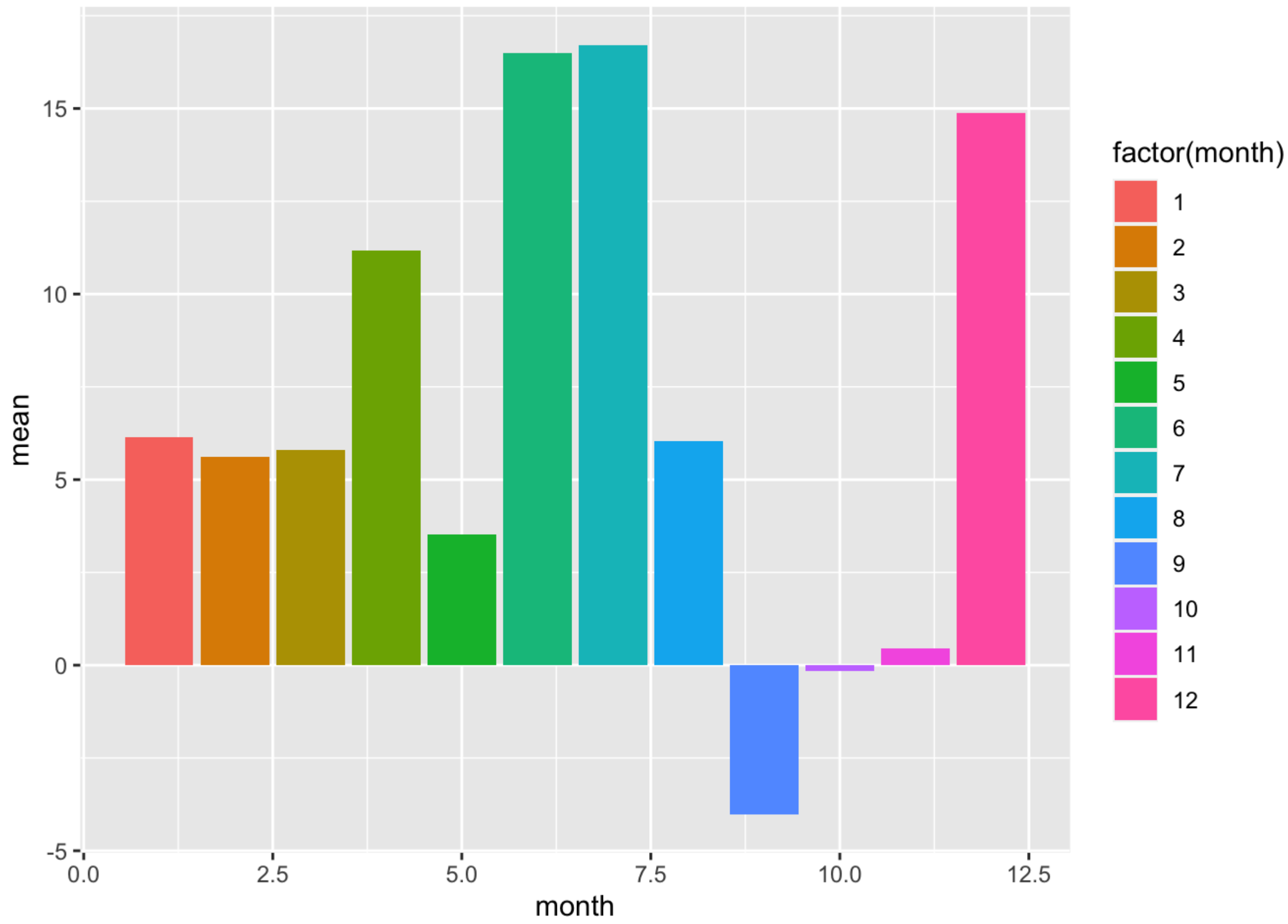
```
flights %>%
  group_by(month) %>%
  summarise(mean = mean(arr_delay, na.rm = TRUE))
```

```
## # A tibble: 12 × 2
##    month   mean
##    <int>  <dbl>
##  1     1  6.13
##  2     2  5.61
##  3     3  5.81
##  4     4 11.2
##  5     5  3.52
##  6     6 16.5
##  7     7 16.7
##  8     8  6.04
##  9     9 -4.02
## 10    10 -0.167
## 11    11  0.461
## 12    12 14.9
```

Q1.1.7

From July and Aug, the average of delay is peaked
September’s average of delay is negative and increased after till December.

```
flights %>%
  group_by(month) %>%
  summarise(mean = mean(arr_delay, na.rm = TRUE)) %>%
  ggplot(aes(month, mean, fill = factor(month)))+
  geom_col()
```



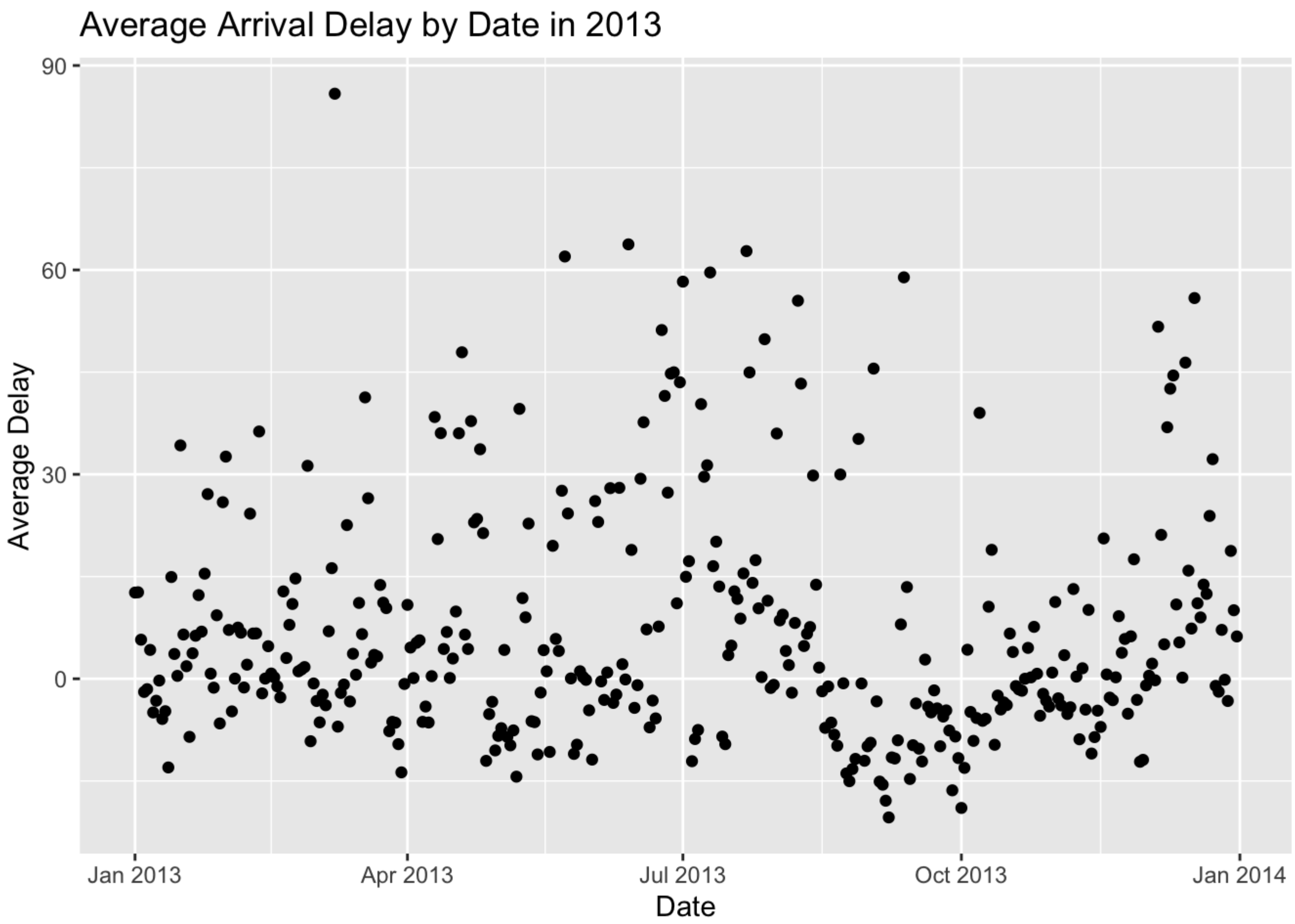
Q1.1.8

Overall, it is align with the bar graph comparing month and mean of the delay.

Delay is more common during summer
Fall looks like having less delay

```r
library(lubridate)

flights %>%
  filter(year == 2013) %>%
  mutate(year_month_date= make_date(year = year, month = month, day = day)) %>%
  group_by(year_month_date) %>%
  summarise(mean_delay = mean(arr_delay, na.rm = TRUE)) %>%
  ggplot(aes(year_month_date, mean_delay))+
  geom_point() +
  labs(title = "Average Arrival Delay by Date in 2013", x = "Date", y = "Average Delay")
```



Q1.2

Q1.2.1

```
flights %>%
  filter(dest == "ORD") %>%
  nrow()
```

```
## [1] 17283
```

Q1.2.2

```
flights %>%
  filter(dest == "ORD") %>%
  arrange(air_time) %>%
  summarize(year, month, day, dep_time,air_time,carrier,origin) %>%
  head(1)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## ℹ Please use `reframe()` instead.
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## # A tibble: 1 × 7
##    year month   day dep_time air_time carrier origin
##   <int> <int> <int>    <int>    <dbl> <chr>   <chr>
## 1  2013     8    21     1604       87 UA      EWR
```

Q1.2.3

I checked the Google flights and the flights are usually in between 2h30m to 2h50m The shortest flight time is 87m, which is 1h 27min
Can be short however not impossible.

Q1.2.4

The longest time is 198min, which is similar to the one I searched

```
flights %>%
  filter(dest == "ORD") %>%
  arrange(desc(air_time)) %>%
  select(year, month, day, dep_time,air_time,carrier,origin) %>%
  head(1)
```

```
## # A tibble: 1 × 7
##    year month   day dep_time air_time carrier origin
##   <int> <int> <int>    <int>    <dbl> <chr>   <chr>
## 1  2013     4    10      638      198 UA      EWR
```

Q1.2.5

```
flights %>%
  filter(dest == "ORD", !is.na(air_time), !is.na(distance)) %>%
  mutate(mph = distance/(air_time/60)) %>%
  arrange(mph) %>%
  slice(c(1:3, (n()-2):n())) %>% ##I googled this trick
  select(year, month, day, air_time,carrier,dep_delay, mph)
```

```
## # A tibble: 6 × 7
##    year month   day air_time carrier dep_delay   mph
##   <int> <int> <int>    <dbl> <chr>       <dbl> <dbl>
## 1  2013     4    10      198 UA             -2  218.
## 2  2013     9    19      192 UA             -4  229.
## 3  2013     9    19      188 UA              1  229.
## 4  2013     5     8       92 MQ              1  469.
## 5  2013     7    12       92 MQ             38  469.
## 6  2013     8    21       87 UA              9  496.
```

Q1.2.6

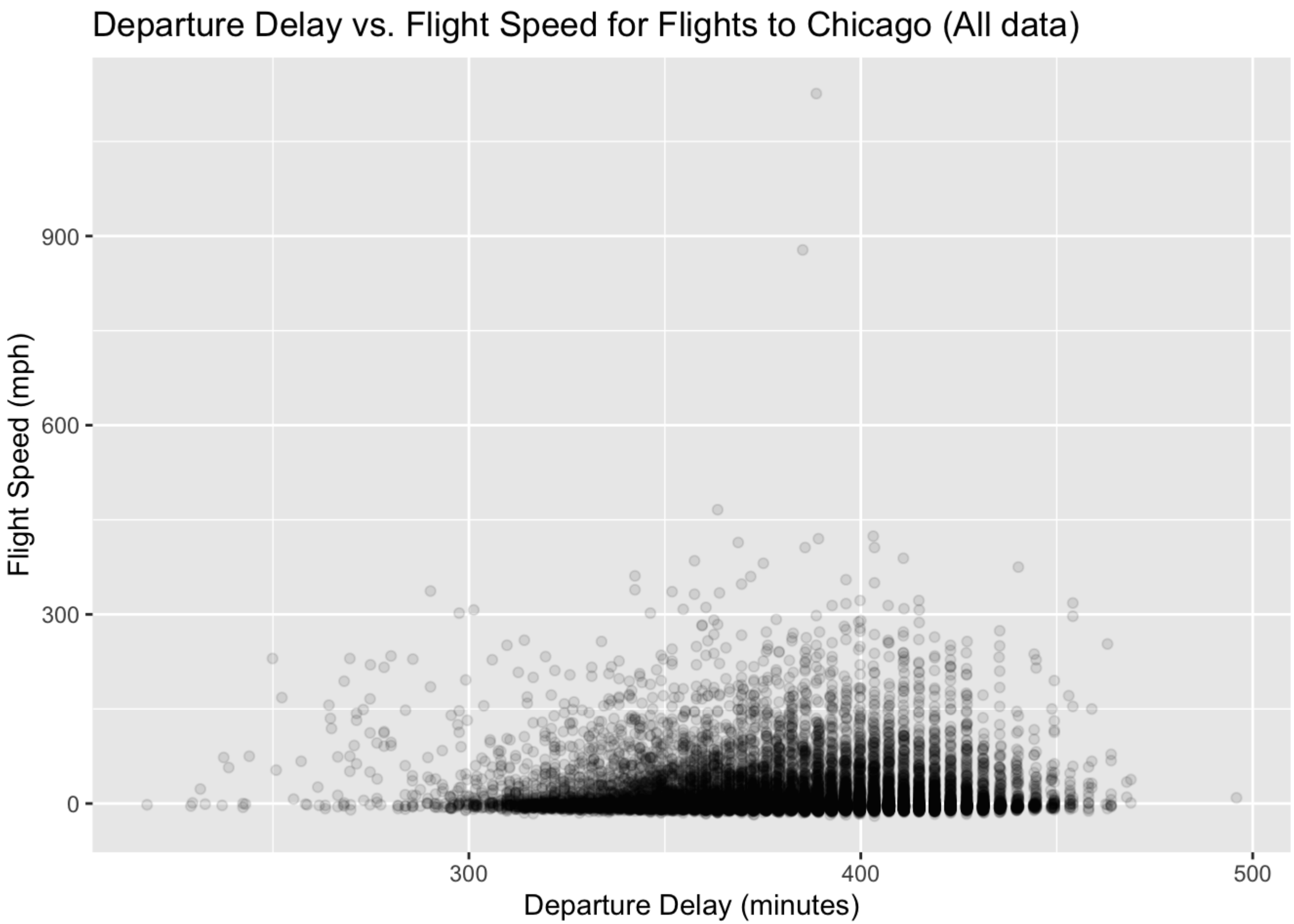
Initially, I experimented with line and box plots. However, a line plot was unsuitable as the data points do not have a sequential or connected nature. As for the box plot, it resulted in a single aggregated box which was ineffective for discerning any distinct patterns within the data.

Consequently, I opted for a scatter plot and also focused on mph(y-axis) from -20 to 300. This visualization effectively displays the overall data excluding outliers, allowing for the detection of any patterns or trends.

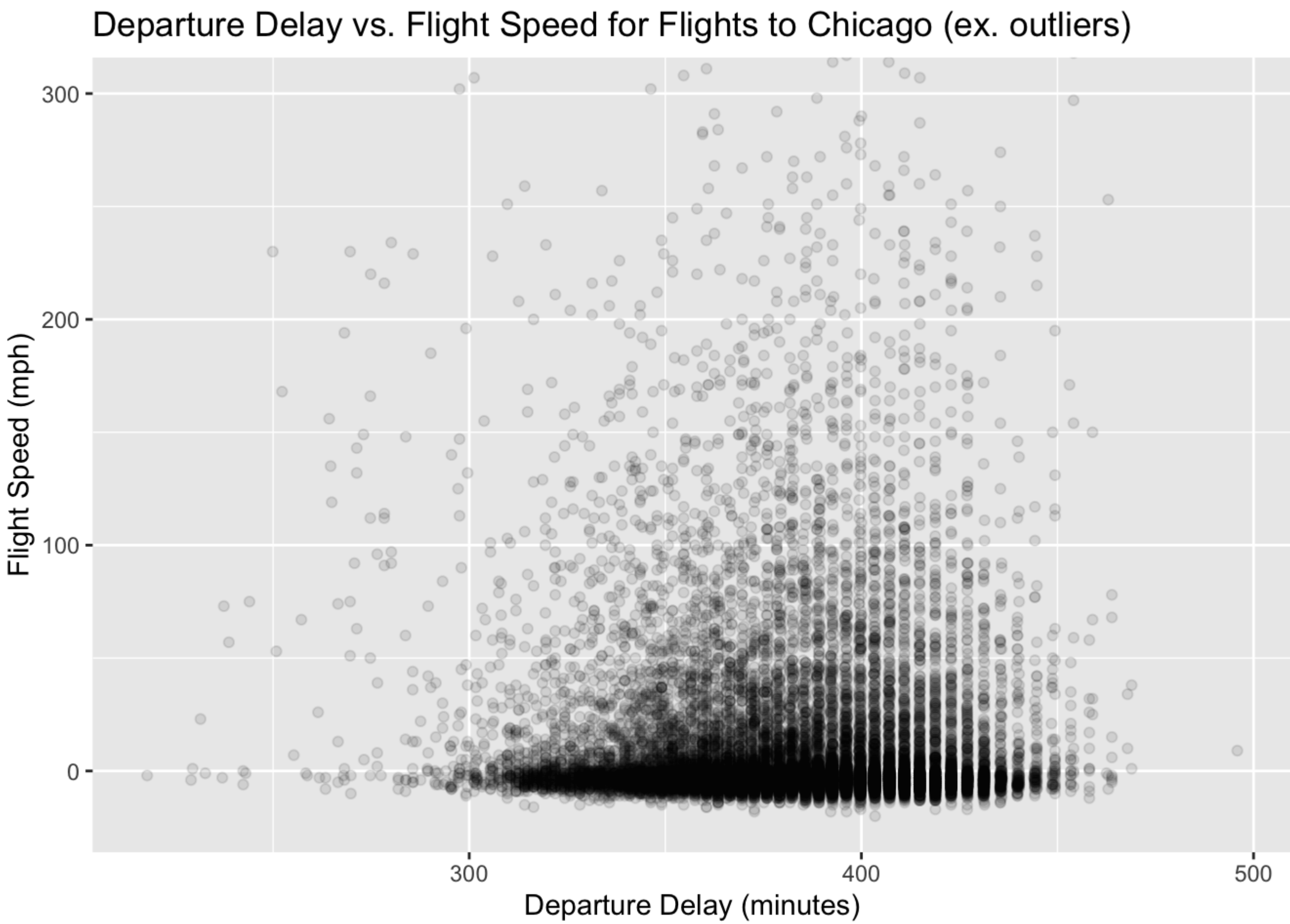
I am seeing that deperture delay is highly accumulated between 300 to 450 min
Also flight speed increase along with departure delay till 400 min and started to decrease after

```
flights_mutate <- flights

flights_mutate %>%
  mutate(mph = distance/(air_time/60)) %>%
  filter(dest == "ORD", !is.na(mph), !is.na(dep_delay)) %>%
  ggplot(aes(mph,dep_delay,)) +
  geom_point(alpha = 0.1) +
  labs(title = "Departure Delay vs. Flight Speed for Flights to Chicago (All data)",
       x = "Departure Delay (minutes)",
       y = "Flight Speed (mph)")
```




```
flights_mutate %>%
  mutate(mph = distance/(air_time/60)) %>%
  filter(dest == "ORD", !is.na(mph), !is.na(dep_delay)) %>%
  ggplot(aes(mph,dep_delay,)) +
  geom_point(alpha = 0.1) +
  coord_cartesian(ylim = c(-20, 300)) +
  labs(title = "Departure Delay vs. Flight Speed for Flights to Chicago (ex. outliers)",
       x = "Departure Delay (minutes)",
       y = "Flight Speed (mph)")
```



Q2.1

Q2.1.1

```
gapmider <- read_delim("../gapminder.csv.bz2")
```

```
## Rows: 13055 Columns: 25
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: "\t"
## chr  (6): iso3, name, iso2, region, sub-region, intermediate-region
## dbl (19): time, totalPopulation, fertilityRate, lifeExpectancy, childMortali...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(gapmider)
```

```
## [1] 13055    25
```

Q2.1.2

There are some NA values for certain columns however for the ones we are going to use are ok

```
head(gapmider,5)
```

```
## # A tibble: 5 × 25
##   iso3  name  iso2  region   `sub-region`             `intermediate-region`  time
##   <chr> <chr> <chr> <chr>    <chr>                    <chr>                 <dbl>
## 1 ABW   Aruba AW    Americas Latin America and the … Caribbean              1960
## 2 ABW   Aruba AW    Americas Latin America and the … Caribbean              1961
## 3 ABW   Aruba AW    Americas Latin America and the … Caribbean              1962
## 4 ABW   Aruba AW    Americas Latin America and the … Caribbean              1963
## 5 ABW   Aruba AW    Americas Latin America and the … Caribbean              1964
## # ℹ 18 more variables: totalPopulation <dbl>, fertilityRate <dbl>,
## #   lifeExpectancy <dbl>, childMortality <dbl>, youthFemaleLiteracy <dbl>,
## #   youthMaleLiteracy <dbl>, adultLiteracy <dbl>, GDP_PC <dbl>,
## #   accessElectricity <dbl>, agriculturalLand <dbl>, agricultureTractors <dbl>,
## #   cerealProduction <dbl>, fertilizerHa <dbl>, co2 <dbl>,
## #   greenhouseGases <dbl>, co2_PC <dbl>, pm2.5_35 <dbl>, battleDeaths <dbl>
```

```
tail(gapmider,5)
```

```
## # A tibble: 5 × 25
##   iso3  name     iso2  region `sub-region`       `intermediate-region`  time
##   <chr> <chr>    <chr> <chr>  <chr>              <chr>                 <dbl>
## 1 ZWE   Zimbabwe ZW    Africa Sub-Saharan Africa Eastern Africa         2015
## 2 ZWE   Zimbabwe ZW    Africa Sub-Saharan Africa Eastern Africa         2016
## 3 ZWE   Zimbabwe ZW    Africa Sub-Saharan Africa Eastern Africa         2017
## 4 ZWE   Zimbabwe ZW    Africa Sub-Saharan Africa Eastern Africa         2018
## 5 ZWE   Zimbabwe ZW    Africa Sub-Saharan Africa Eastern Africa         2019
## # ℹ 18 more variables: totalPopulation <dbl>, fertilityRate <dbl>,
## #   lifeExpectancy <dbl>, childMortality <dbl>, youthFemaleLiteracy <dbl>,
## #   youthMaleLiteracy <dbl>, adultLiteracy <dbl>, GDP_PC <dbl>,
## #   accessElectricity <dbl>, agriculturalLand <dbl>, agricultureTractors <dbl>,
## #   cerealProduction <dbl>, fertilizerHa <dbl>, co2 <dbl>,
## #   greenhouseGases <dbl>, co2_PC <dbl>, pm2.5_35 <dbl>, battleDeaths <dbl>
```

Q2.1.3

ISO3 : 253
ISO2 : 249
name : 250

```
iso3_n <- gapmider %>%
  distinct(iso3) %>%
  nrow()

iso2_n <- gapmider %>%
  distinct(iso2) %>%
  nrow()

name_n <- gapmider %>%
  distinct(name) %>%
  nrow()
```

Q2.1.4

```
#(a)

gapmider %>%
  group_by(iso2) %>%
  summarise(names = toString(unique(name)), count = n_distinct(name)) %>%
  filter(count>1) %>%
  select(names)
```

```
## # A tibble: 1 × 1
##   names
##   <chr>
## 1 NA, Namibia
```

```
#(b)

gapmider %>%
  group_by(name) %>%
  summarise(iso3s = toString(unique(iso3)), count = n_distinct(iso3)) %>%
  filter(count>1) %>%
  select(iso3s)
```

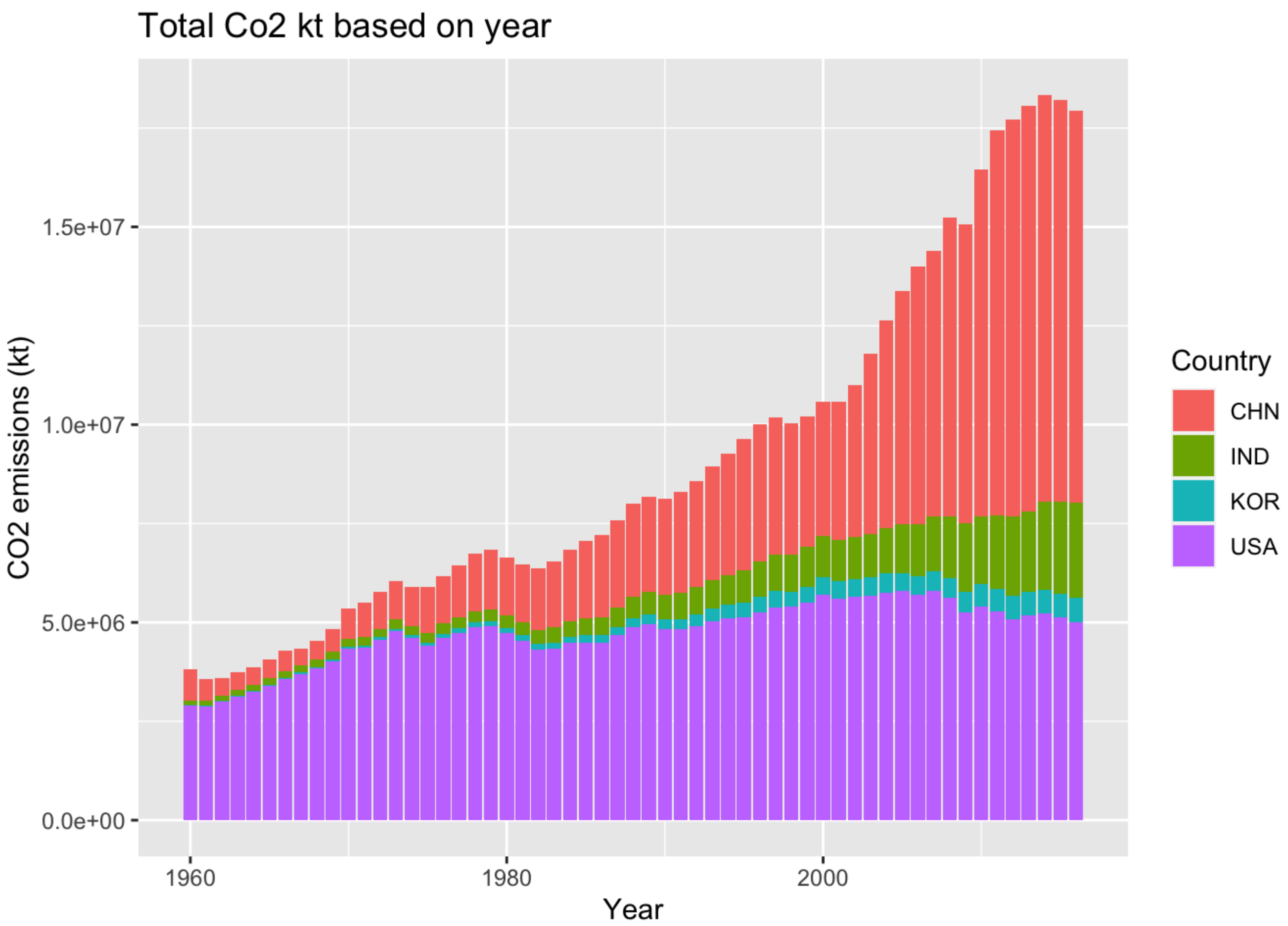
```
## # A tibble: 1 × 1
##   iso3s
##   <chr>
## 1 CHANISL, GBM, KOS, NLD_CURACAO
```

Q2.2

Q2.2.1

It has been increasing over time

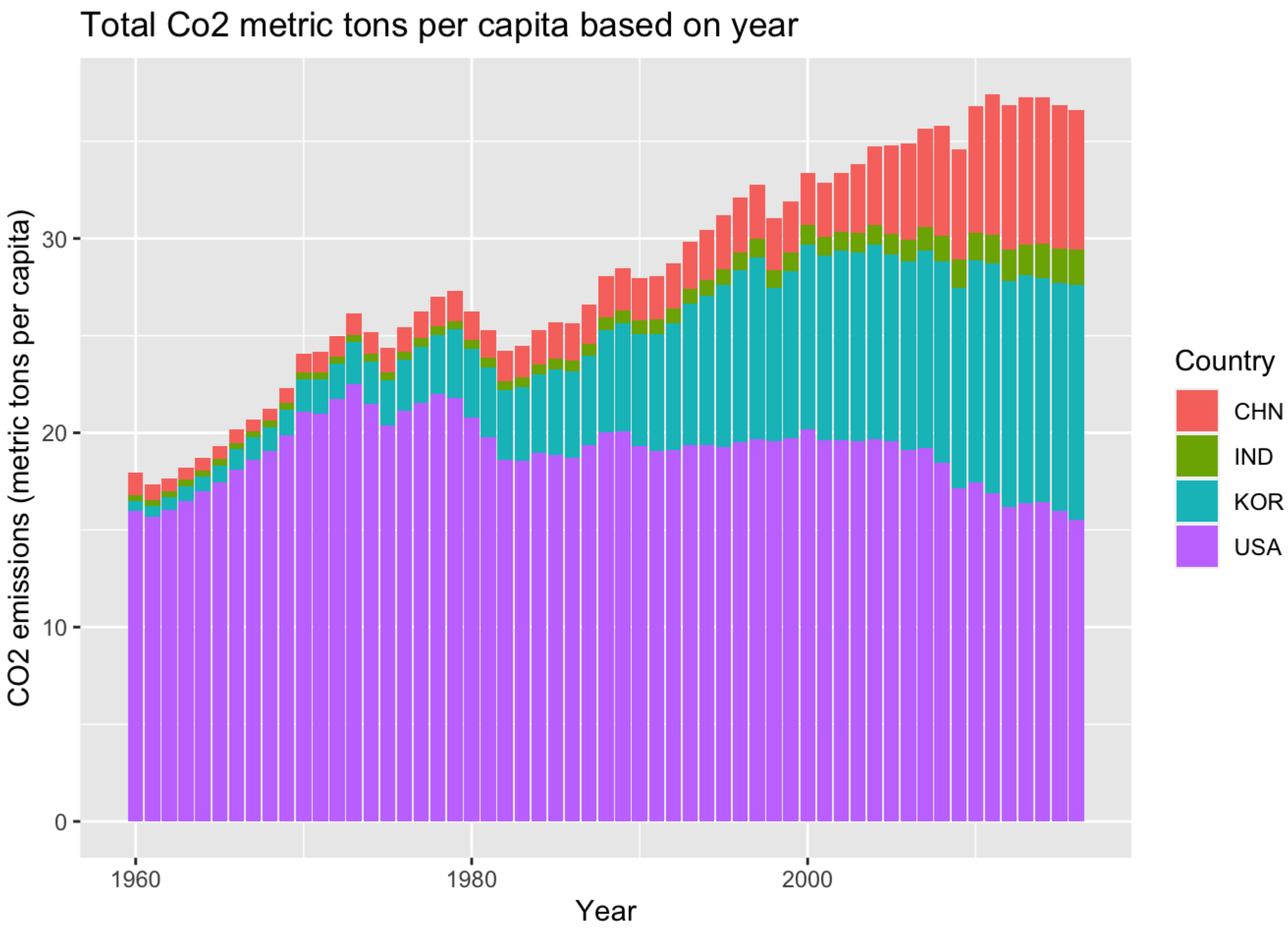
```r
gapmider %>%
  filter(iso3 %in% c('CHN', 'USA', 'IND', 'KOR'), !is.na(co2)) %>%
  ggplot(aes(time, co2, fill=iso3)) +
  geom_col() +
    labs(
       x = "Year",
       y = "CO2 emissions (kt)",
       fill = "Country",
       title = "Total Co2 kt based on year")
```



Q2.2.2

CO2 per capita was keep increasing till 1970-1080, decreased slightly after, and then increasing till recent.

```
gapmider %>%
  filter(iso3 %in% c('CHN', 'USA', 'IND', 'KOR'), !is.na(co2_PC) ) %>%
  ggplot(aes(time, co2_PC, fill = iso3)) +
  geom_col() +
  labs(
       x = "Year",
       y = "CO2 emissions (metric tons per capita)",
       fill = "Country",
       title = "Total Co2 metric tons per capita based on year")
```



Q2.2.3

Only showing the 1960 and 2016 since the result dataset is long, only showing the relevant data for next question Q2.2.4

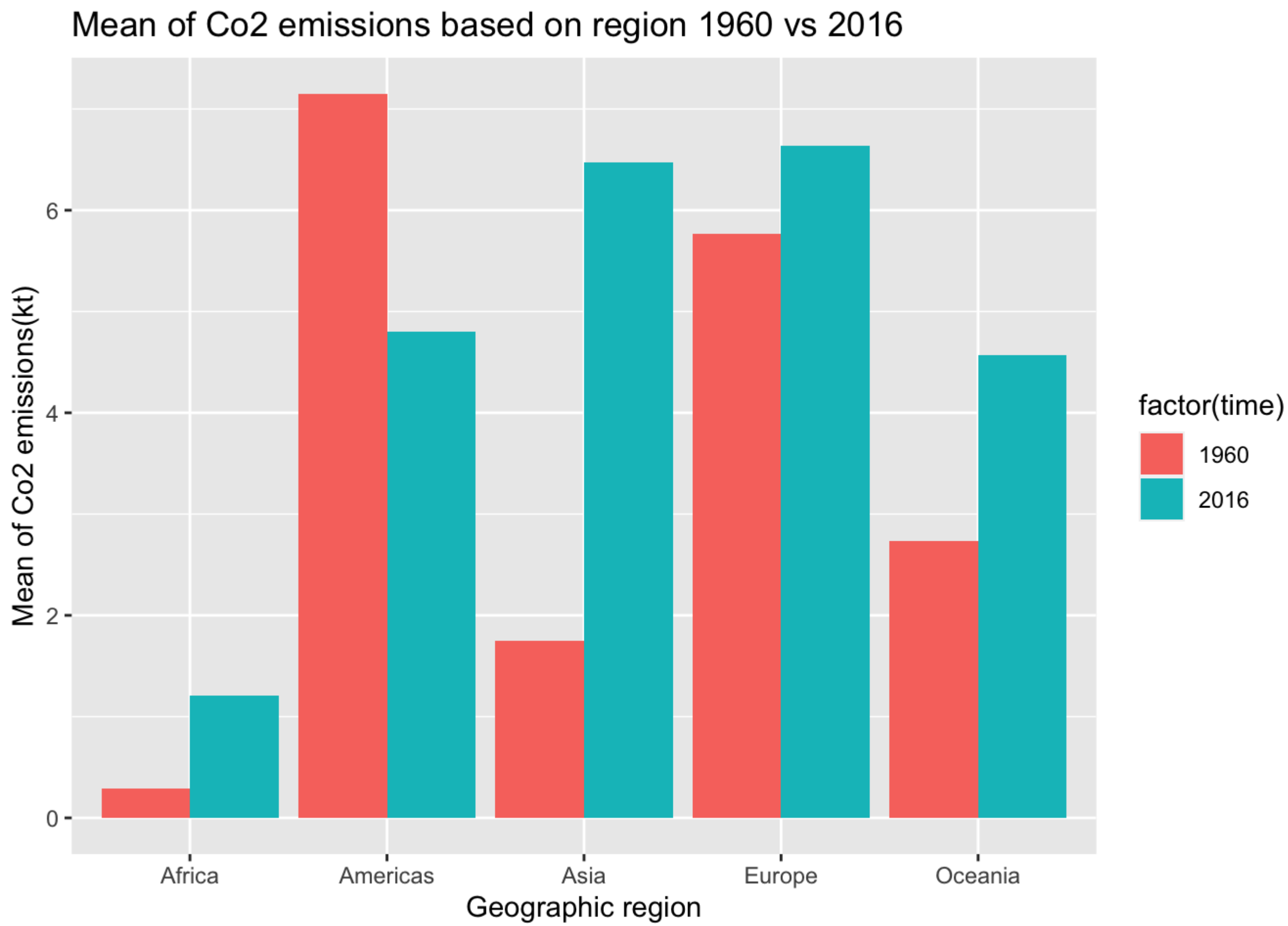
```
gapmider %>%
  group_by(region, time) %>%
  filter(time %in% c(1960, 2016), !is.na(co2_PC), !is.na(region)) %>%
  summarise(mean = mean(co2_PC))
```

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

```
## # A tibble: 10 × 3
## # Groups:   region [5]
##    region    time  mean
##    <chr>    <dbl> <dbl>
##  1 Africa    1960 0.291
##  2 Africa    2016 1.20
##  3 Americas  1960 7.15
##  4 Americas  2016 4.80
##  5 Asia      1960 1.74
##  6 Asia      2016 6.47
##  7 Europe    1960 5.77
##  8 Europe    2016 6.64
##  9 Oceania   1960 2.73
## 10 Oceania   2016 4.57
```

Q2.2.4

```
gapmider %>%
  filter(time %in% c(1960, 2016), !is.na(co2_PC), !is.na(region)) %>%
  group_by(region, time) %>%
  summarise(mean = mean(co2_PC), .groups = 'drop') %>%
  ggplot(aes(region, mean, fill = factor(time))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
       x = "Geographic region",
       y = "Mean of Co2 emissions(kt)",
       color = "Year",
       title = "Mean of Co2 emissions based on region 1960 vs 2016")
```

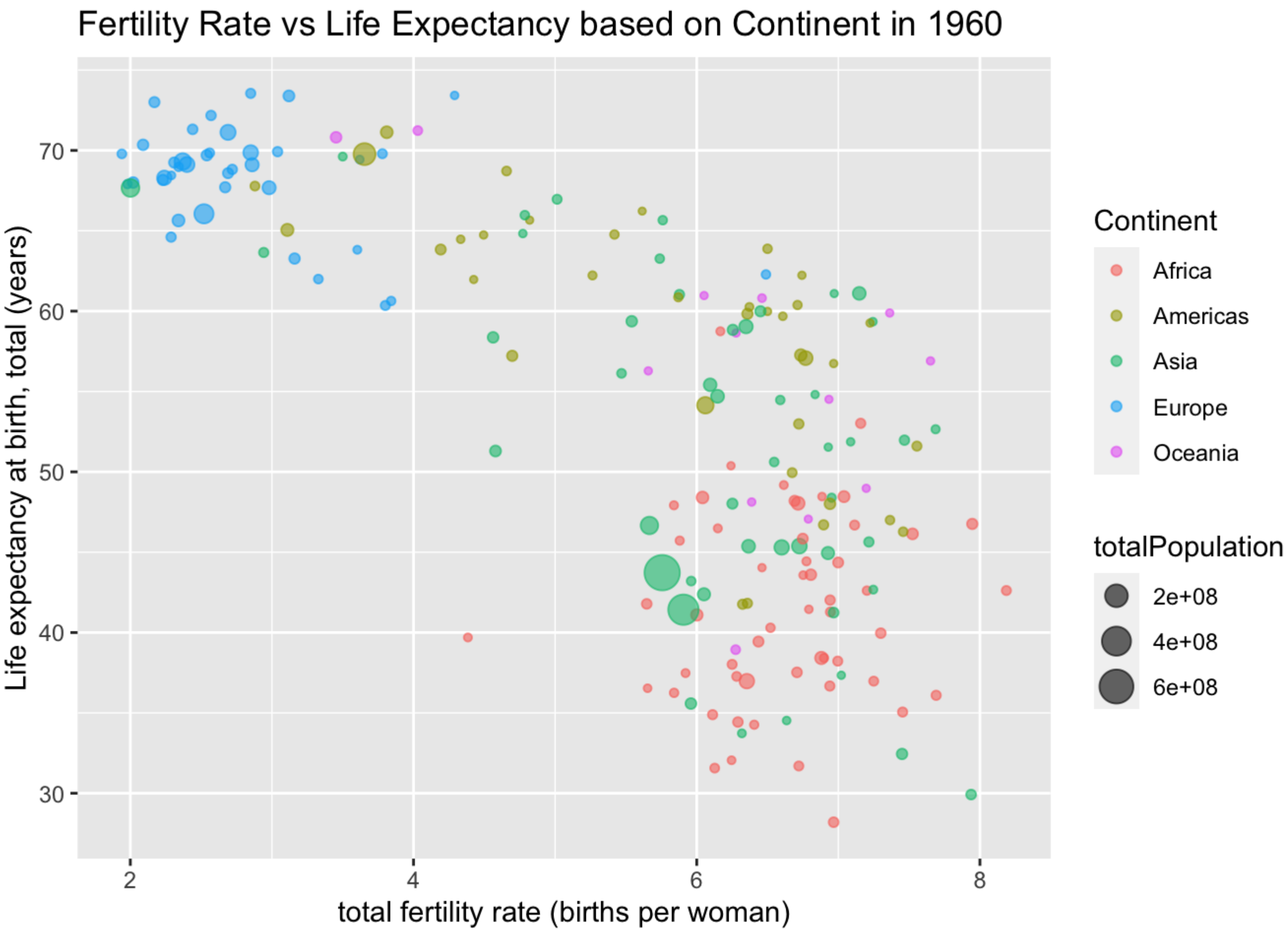


Q2.3

Q2.3.1

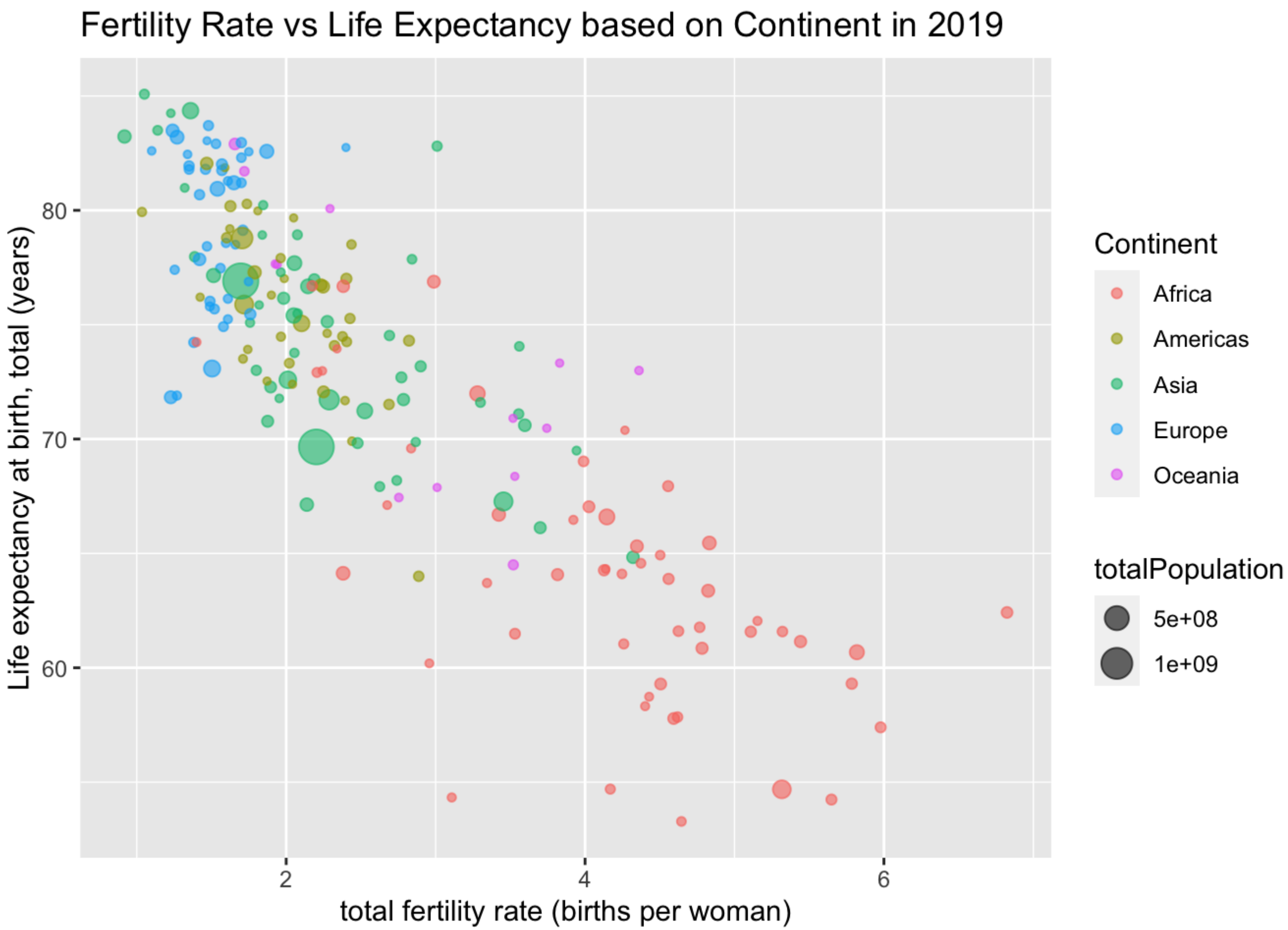
Generally, Europe exhibits very low fertility and high life expectancy, while Africa is at the opposite end. America and Asia are somewhat scattered in the mid-range; however, some populous Asian countries show high fertility and low life expectancy.

```
gapmider %>%
  filter(time == 1960,!is.na(fertilityRate), !is.na(lifeExpectancy), !is.na(region), !is.na(totalPopulation)) %>%
  ggplot(aes(fertilityRate,lifeExpectancy, size = totalPopulation, color = factor(region)))+
  geom_point(alpha = 0.6) +
  labs(
       x = "total fertility rate (births per woman)",
       y = "Life expectancy at birth, total (years)",
       color = "Continent",
       title = "Fertility Rate vs Life Expectancy based on Continent in 1960")
```



Q2.3.2

```
gapmider %>%
  filter(time == 2019,!is.na(fertilityRate), !is.na(lifeExpectancy), !is.na(region), !is.na(totalPopulation)) %>%
  ggplot(aes(fertilityRate,lifeExpectancy, size = totalPopulation, color = factor(region)))+
  geom_point(alpha = 0.6) +
  labs(
       x = "total fertility rate (births per woman)",
       y = "Life expectancy at birth, total (years)",
       color = "Continent",
       title = "Fertility Rate vs Life Expectancy based on Continent in 2019")
```



Q2.3.3

Globally, fertility rates have decreased while life expectancy has increased. Particularly, the Americas and Asia are characterized by low fertility and high life expectancy. However, Europe and Africa remain on opposite ends of this spectrum.

Q2.3.4

The previous scatterplot, encompassing all countries, makes direct comparisons challenging. However, the spread of data points aligns well with the overall mean values.

```
gapmider %>%
  filter(time %in% c(1960, 2019), !is.na(lifeExpectancy), !is.na(region)) %>%
  group_by(region, time) %>%
  summarise(mean=mean(lifeExpectancy))
```

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

```
## # A tibble: 10 × 3
## # Groups:   region [5]
##    region    time  mean
##    <chr>    <dbl> <dbl>
##  1 Africa    1960  41.5
##  2 Africa    2019  64.1
##  3 Americas  1960  58.6
##  4 Americas  2019  75.8
##  5 Asia      1960  51.6
##  6 Asia      2019  74.6
##  7 Europe    1960  68.3
##  8 Europe    2019  79.4
##  9 Oceania   1960  56.4
## 10 Oceania   2019  73.5
```

Q2.3.5

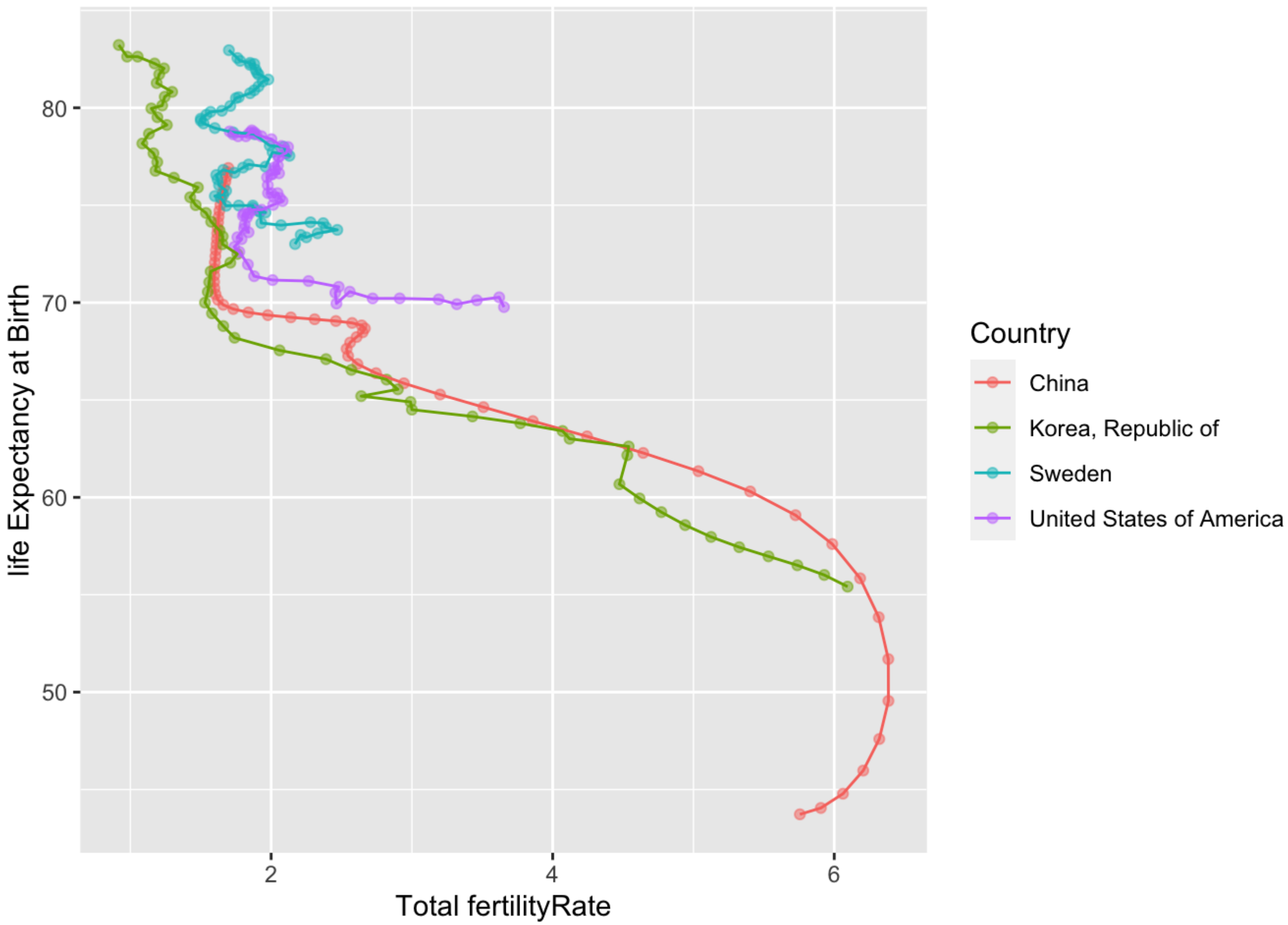
```
gapmider %>%
  filter(time %in% c(1960, 2019), !is.na(lifeExpectancy), !is.na(region)) %>%
  group_by(region, time) %>%
  summarise(mean=mean(lifeExpectancy)) %>%
  ungroup() %>%
  mutate(growth = mean-lag(mean)) %>%
  filter(time == 2019)
```

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

```
## # A tibble: 5 × 4
##   region    time  mean growth
##   <chr>    <dbl> <dbl>  <dbl>
## 1 Africa    2019  64.1   22.6
## 2 Americas  2019  75.8   17.2
## 3 Asia      2019  74.6   23.0
## 4 Europe    2019  79.4   11.1
## 5 Oceania   2019  73.5   17.1
```

Q2.3.6

```
gapmider %>%
  filter(iso3 %in% c("USA", "CHN", "SWE", "KOR"), !is.na(fertilityRate)) %>%
  group_by(name) %>%
  ggplot(aes(fertilityRate,lifeExpectancy, color = factor(name))) +
  geom_point(alpha= 0.5) +
  geom_path() +
  labs(
       x = "Total fertilityRate",
       y = "life Expectancy at Birth",
       color = "Country")
```



Q2.3.7

```
gapmider %>%
  filter(time %in% c(1960, 2019), !is.na(lifeExpectancy), !is.na(name)) %>%
  group_by(time) %>%
  mutate(rank = rank(desc(lifeExpectancy))) %>%
  filter(name == "United States of America") %>%
  select(time, name, lifeExpectancy, rank)
```

```
## # A tibble: 2 × 4
## # Groups:   time [2]
##    time name                     lifeExpectancy  rank
##   <dbl> <chr>                             <dbl> <dbl>
## 1  1960 United States of America           69.8    17
## 2  2019 United States of America           78.8    46
```