# AssignmentRlab01

Ehsan Eslmai Shafigh

2023-04-13

## Assignment 1

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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
```

```
library(nycflights13)
```

### Exercise 1: American Airlines Employees

**1)** We start by importing the data:

```
aa = read.delim(file = "american_airline_empl.txt" , sep = "")
da = read.delim(file = "delta_airline_empl.txt" , sep = "")
fe = read.delim(file = "federal_express_empl.txt" , sep = "")
ua = read.delim(file = "united_airline_empl.txt" , sep = "")
```

There are a couple of columns which contain strings, which we would like to convert to numbers:

```
aa$Full.time = as.numeric(gsub("," , "" , aa$Full.time ))
aa$Part.time = as.numeric(gsub("," , "" , aa$Part.time ))
aa$Grand = as.numeric(gsub("," , "" , aa$Grand ))

da$Full.time = as.numeric(gsub("," , "" , da$Full.time ))
da$Part.time = as.numeric(gsub("," , "" , da$Part.time ))
da$Grand = as.numeric(gsub("," , "" , da$Grand ))

fe$Full.time = as.numeric(gsub("," , "" , fe$Full.time ))
fe$Part.time = as.numeric(gsub("," , "" , fe$Part.time ))
```

```
fe$Grand = as.numeric(gsub("," , "" , fe$Grand ))

ua$Full.time = as.numeric(gsub("," , "" , ua$Full.time ))
ua$Part.time = as.numeric(gsub("," , "" , ua$Part.time ))
ua$Grand = as.numeric(gsub("," , "" , ua$Grand ))
```

We convert the imported data into a tibble:

```
aa = tibble(aa)
da = tibble(da)
fe = tibble(fe)
ua = tibble(ua)
```

**2)** To successfully merge the tibbles, we need to change the names of some columns:

```
aa = rename(aa , 'American Airlines FT' = 'Full.time' , 'American Airlines PT' = 'Part.time' , 'America
da = rename(da , 'Delta Airlines FT' = 'Full.time' , 'Delta Airlines PT' = 'Part.time' , 'Delta Airline
fe = rename(fe , 'Federal Express FT' = 'Full.time' , 'Federal Express PT' = 'Part.time' , 'Federal Exp
ua = rename(ua , 'United Airlines FT' = 'Full.time' , 'United Airlines PT' = 'Part.time' , 'United Airl
```

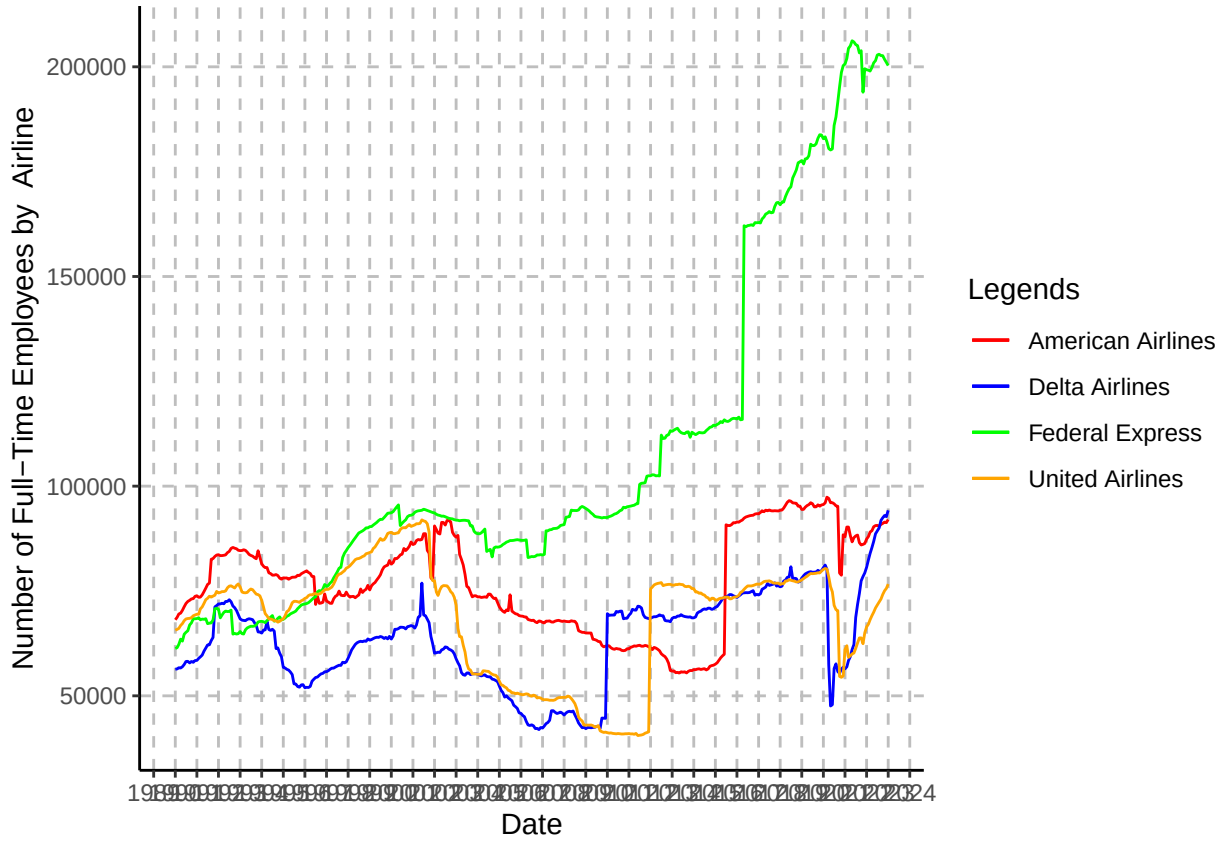
We need to combine the two columns corresponding to the year and month, to create a new column of the date type. For a practical reason, we need also to add a column of days:

```
aa$day = 1
aa = unite(aa, date , Year , Month , day , sep = "-")
aa$date = ymd(aa$date)

da$day = 1
da = unite(da, date , Year , Month , day , sep = "-")
da$date = ymd(da$date)

fe$day = 1
fe = unite(fe, date , Year , Month , day , sep = "-")
fe$date = ymd(fe$date)

ua$day = 1
ua = unite(ua, date , Year , Month , day , sep = "-")
ua$date = ymd(ua$date)

df = merge(aa , da , by = 'date')
df = merge(df , fe , by = 'date')
df = merge(df , ua , by = 'date')
```

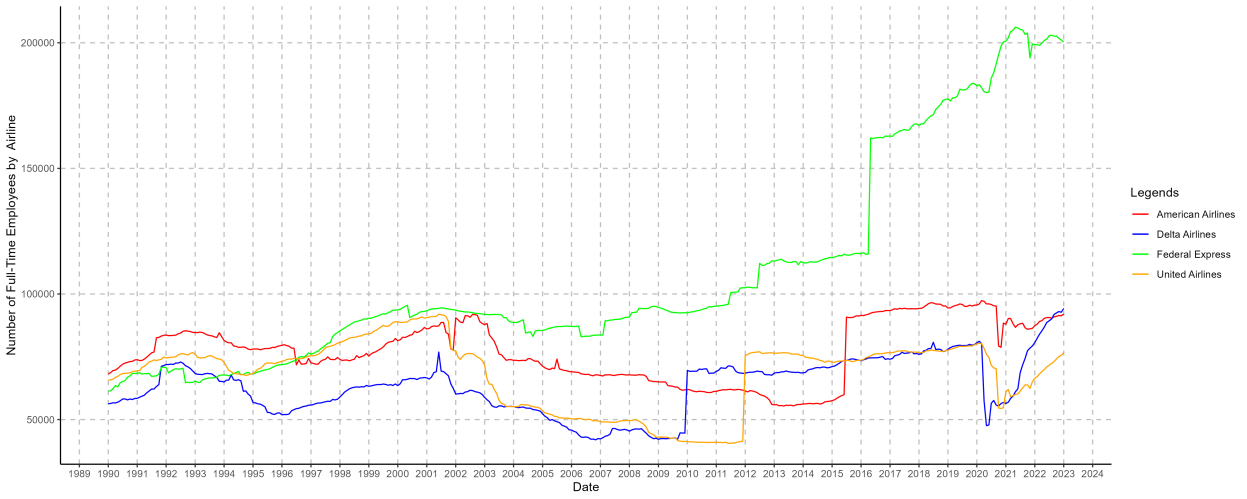
```
## Warning in merge.data.frame(df, ua, by = "date"): column names 'Total.x',
## 'Total.y' are duplicated in the result
```

**3)** one plot for the full-time employees:

```r
df = select(df , -Total.x , -Total.y)

ggplot(df, aes(x = date)) +
  geom_line(aes(y = `American Airlines FT`, color = "American Airlines")) +
  geom_line(aes(y = `Delta Airlines FT`, color = "Delta Airlines")) +
  geom_line(aes(y = `Federal Express FT`, color = "Federal Express")) +
  geom_line(aes(y = `United Airlines FT`, color = "United Airlines")) +
  scale_color_manual(name = "Legends", values = c("American Airlines" = "red", "Delta Airlines" = "blue
  labs(x = "Date", y = "Number of Full-Time Employees by  Airline") +
  theme_classic()+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ theme(
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_blank(),
    panel.background = element_blank()
  )
```

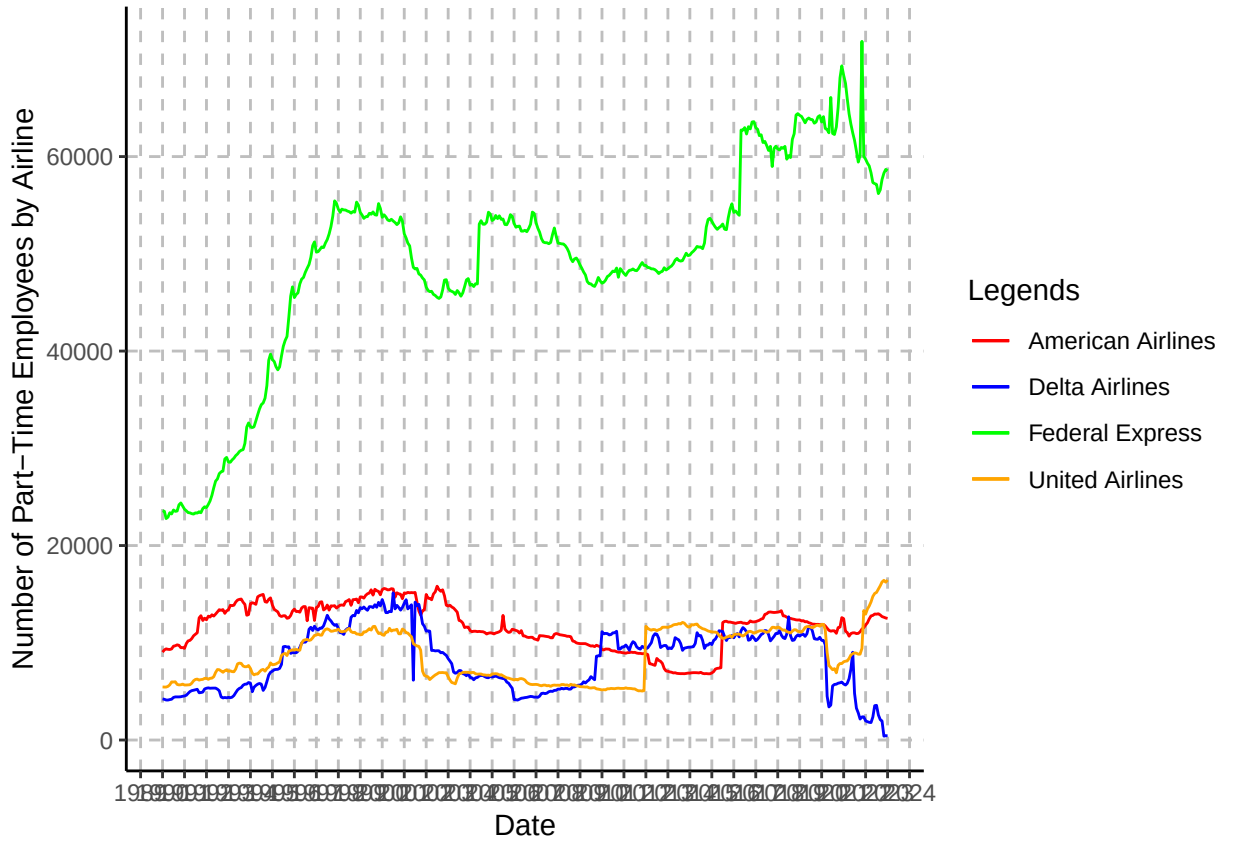



```r
ggsave("full time employees.png", width = 15, height = 6)
```

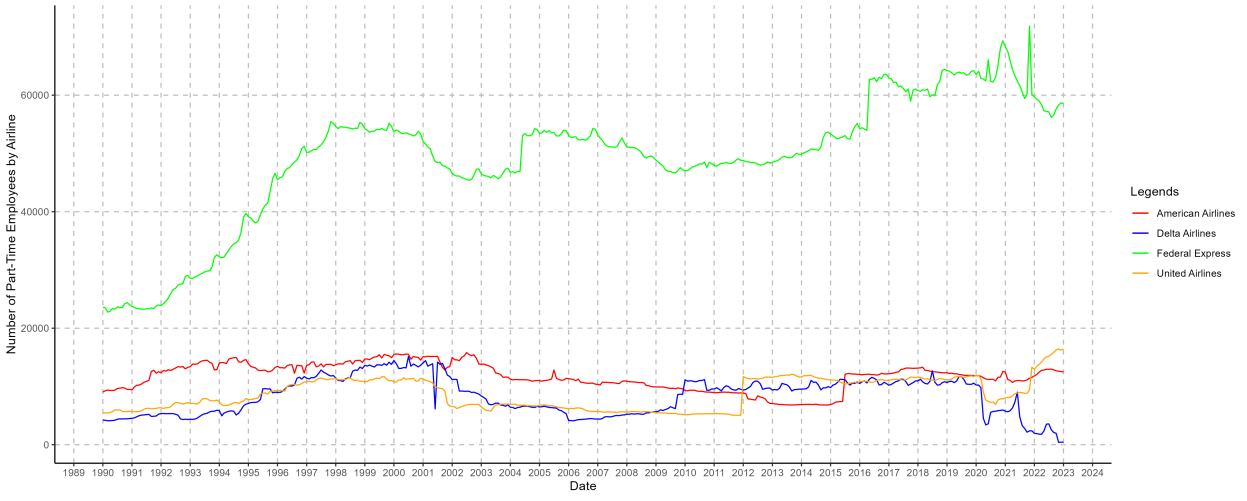



Another plot for the part-time workers:

```
ggplot(df, aes(x = date)) +
  geom_line(aes(y = `American Airlines PT`, color = "American Airlines")) +
  geom_line(aes(y = `Delta Airlines PT`, color = "Delta Airlines")) +
  geom_line(aes(y = `Federal Express PT`, color = "Federal Express")) +
  geom_line(aes(y = `United Airlines PT`, color = "United Airlines")) +
  scale_color_manual(name = "Legends", values = c("American Airlines" = "red", "Delta Airlines" = "blue
  labs(x = "Date", y = "Number of Part-Time Employees by Airline") +
  theme_classic()+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ theme(
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_blank(),
    panel.background = element_blank()
  )
```




```
ggsave("part time employees.png", width = 15, height = 6)
```




```
print(df[which.max(df$`American Airlines Grand`),1])
```

**4)**

```
## [1] "2018-06-01"
```

```
print(df[which.max(df$`Delta Airlines Grand`),1])
```

```
## [1] "2023-01-01"
```

```
print(df[which.max(df$`Federal Express Grand`),1])
```

```
## [1] "2021-03-01"
```

```
print(df[which.max(df$`United Airlines Grand`),1])
```

```
## [1] "2001-03-01"
```

```
print(df[which.min(df$`American Airlines Grand`),1])
```

```
## [1] "2013-09-01"
```

```
print(df[which.min(df$`Delta Airlines Grand`),1])
```

```
## [1] "2006-11-01"
```

```
print(df[which.min(df$`Federal Express Grand`),1])
```

```
## [1] "1990-01-01"
```

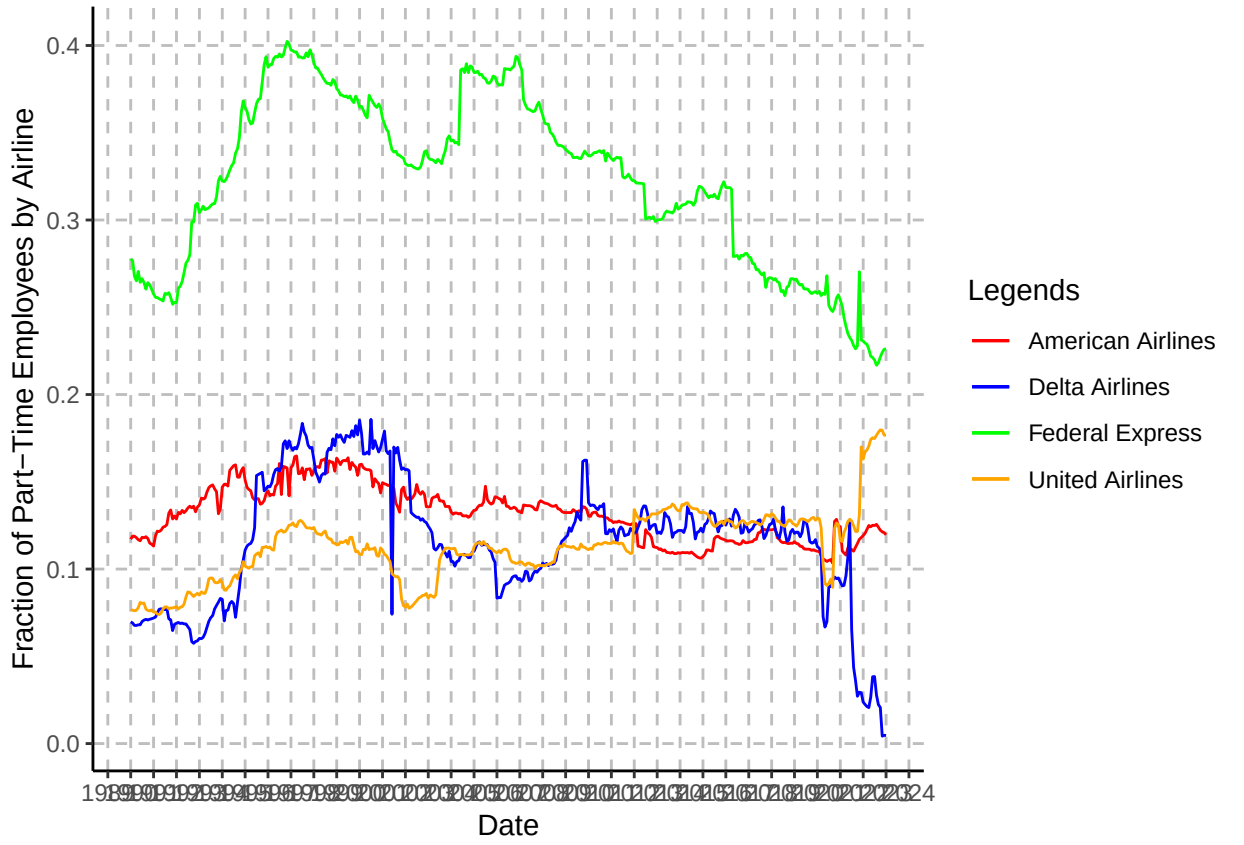
```
print(df[which.min(df$`United Airlines Grand`),1])
```

```
## [1] "2011-06-01"
```

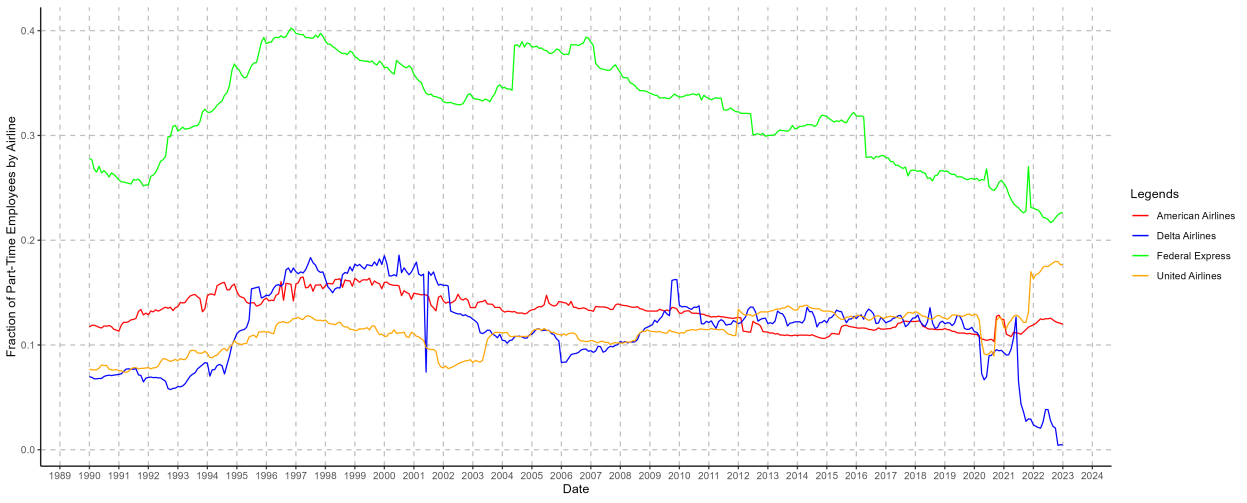
**5)** We create a new column containing the fraction of part-time workers to the total workers:

```
df$ptf_aa = df$`American Airlines PT`/(df$`American Airlines FT` + df$`American Airlines PT`)
df$ptf_da = df$`Delta Airlines PT`/(df$`Delta Airlines FT` + df$`Delta Airlines PT`)
df$ptf_fe = df$`Federal Express PT`/(df$`Federal Express FT` + df$`Federal Express PT`)
df$ptf_ua = df$`United Airlines PT` / (df$`United Airlines FT` + df$`United Airlines PT`)
```

```
ggplot(df, aes(x = date)) +
  geom_line(aes(y = `ptf_aa`, color = "American Airlines")) +
  geom_line(aes(y = `ptf_da`, color = "Delta Airlines")) +
  geom_line(aes(y = `ptf_fe`, color = "Federal Express")) +
  geom_line(aes(y = `ptf_ua`, color = "United Airlines")) +
  scale_color_manual(name = "Legends", values = c("American Airlines" = "red", "Delta Airlines" = "blue
  labs(x = "Date", y = "Fraction of Part-Time Employees by Airline") +
  theme_classic()+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ theme(
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_blank(),
    panel.background = element_blank()
  )
```



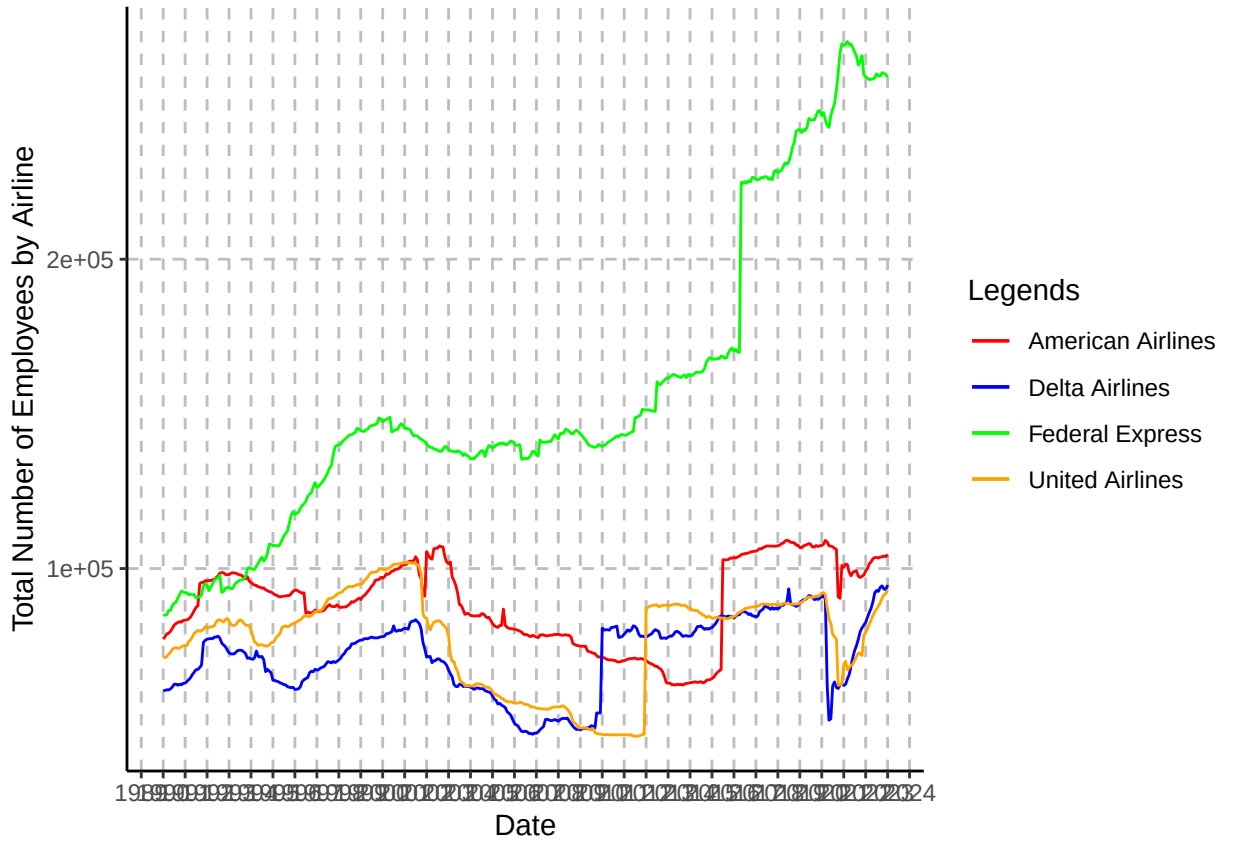

```
ggsave("fraction of part time employees.png", width = 15, height = 6)
```



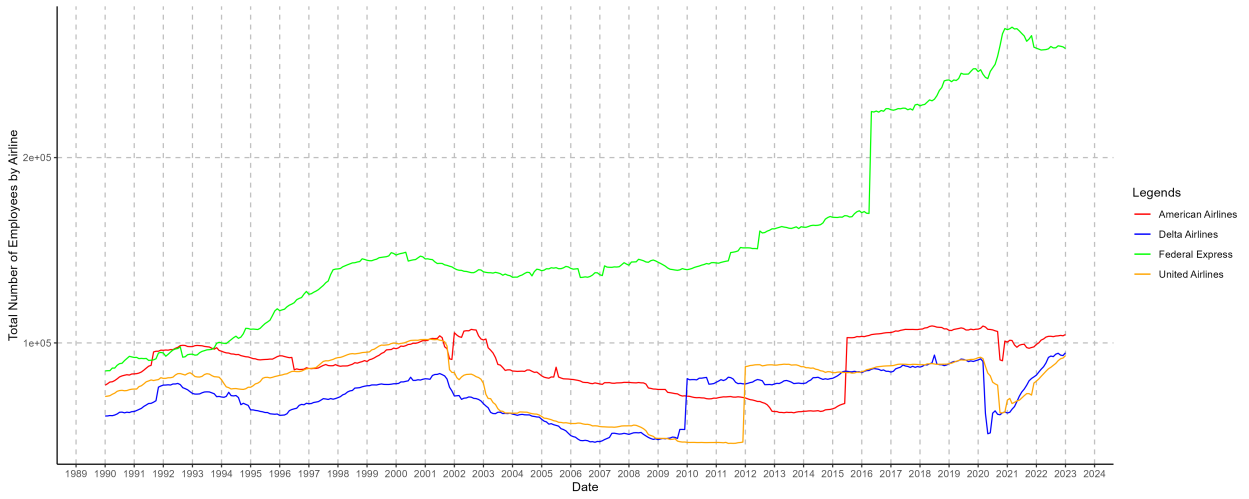

**6)** We have to take a look at the total employees of the four companies:

```
ggplot(df, aes(x = date)) +
  geom_line(aes(y = `American Airlines Grand`, color = "American Airlines")) +
  geom_line(aes(y = `Delta Airlines Grand`, color = "Delta Airlines")) +
  geom_line(aes(y = `Federal Express Grand`, color = "Federal Express")) +
```

```
  geom_line(aes(y = `United Airlines Grand`, color = "United Airlines")) +
  scale_color_manual(name = "Legends", values = c("American Airlines" = "red", "Delta Airlines" = "blue
  labs(x = "Date", y = "Total Number of Employees by Airline") +
  theme_classic()+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ theme(
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_blank(),
    panel.background = element_blank()
  )
```




```
ggsave("total employees.png", width = 15, height = 6)
```




We see that unlike the other companies, the number of employees in the Federal Express has increased during the period, while for the others it has declined initially and then returned almost to the pre-covid values.

### Exercise 2: Data Frames & Tibble

we begin with importing the data:

```
nyc_flights = flights
```

In order to plot the data we form a column corresponding to the date of each flight:

```
nyc_flights = unite(nyc_flights , date , year , month , day , sep = "-" )
nyc_flights$date = ymd(nyc_flights$date)
```

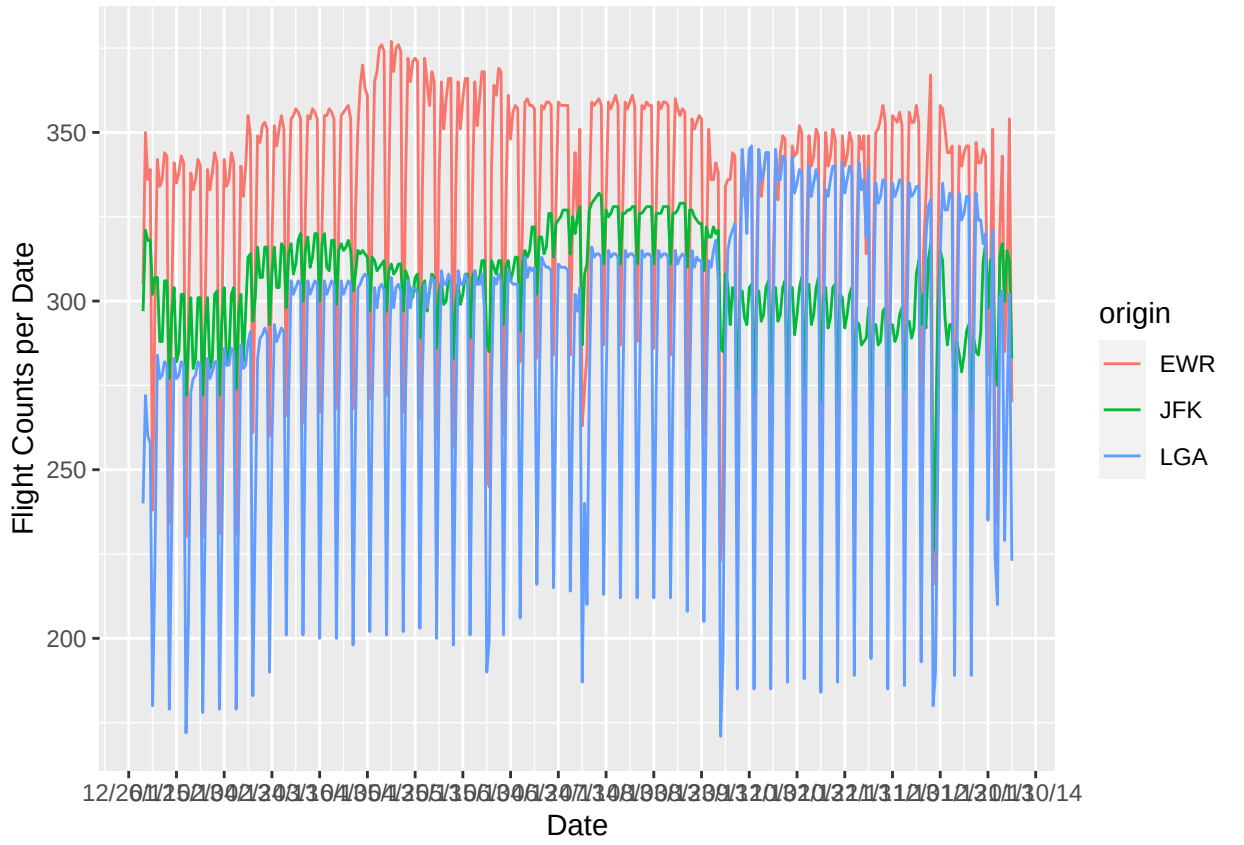
**1.1)** We group by the dates and the origin, and summarize the results as the counts of flights:

```
flight_count_summary = nyc_flights %>%
  group_by(date , origin) %>%
  summarize(flight_counts = n())
```

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

we plot the results:

```
ggplot(data = flight_count_summary, aes(x = date, y = flight_counts, color = origin)) +
  labs(x = "Date", y = "Flight Counts per Date") +
  geom_line()+
  scale_x_date(date_breaks = "20 day", date_labels = "%D")
```




```
ggsave("flights per day.png", width = 15, height = 6)
```


**1.2)** to plot the flights per week, first we create a column containing the week number

```
flights_week = flight_count_summary
flights_week = flights_week %>%
  mutate(week = week(date))
```

then we add a column containing the name of the day of the given date

```
flights_week = flights_week %>%
  mutate(day = wday(date, label = TRUE))
```

Next we filter the weekend

```
flights_workdays = flights_week %>%
  filter(!day %in% c("Sat", "Sun"))
```

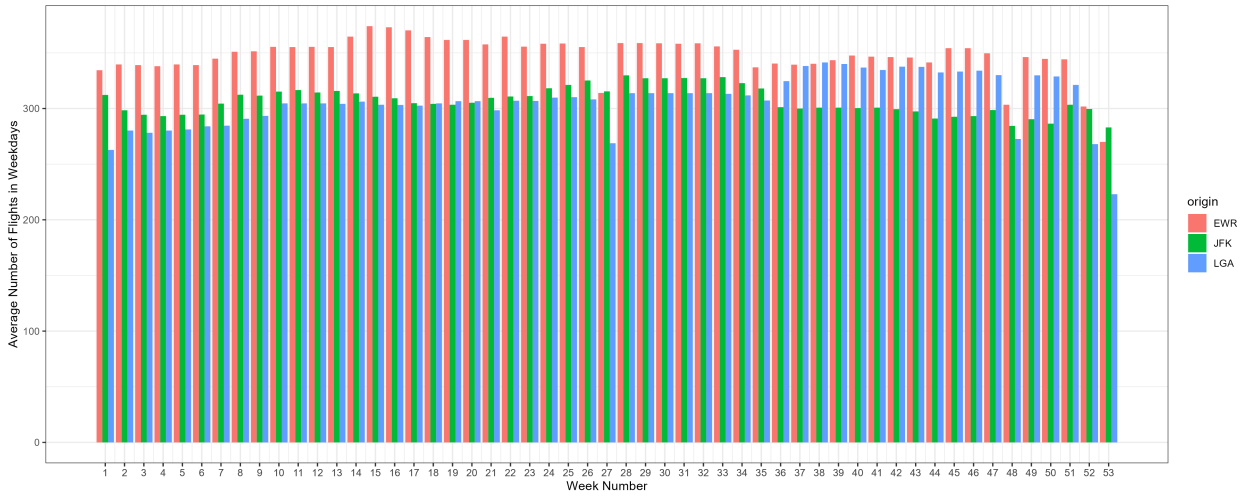
Finally we average over the weekdays and create a column

```
flights_weekdays_avg = flights_workdays %>%
  group_by(week, origin) %>%
  summarize(avg_flights = mean(flight_counts))
```

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

```
ggplot(flights_weekdays_avg, aes(x = week, y = avg_flights, fill = origin)) +
  geom_col(position = "dodge") +
  labs(x = "Week Number", y = "Average Number of Flights in Weekdays") +
  theme_bw() +
  scale_x_continuous(breaks = flights_weekdays_avg$week)
```

```
ggsave("average flight in weekdays.png", width = 15, height = 6)
```

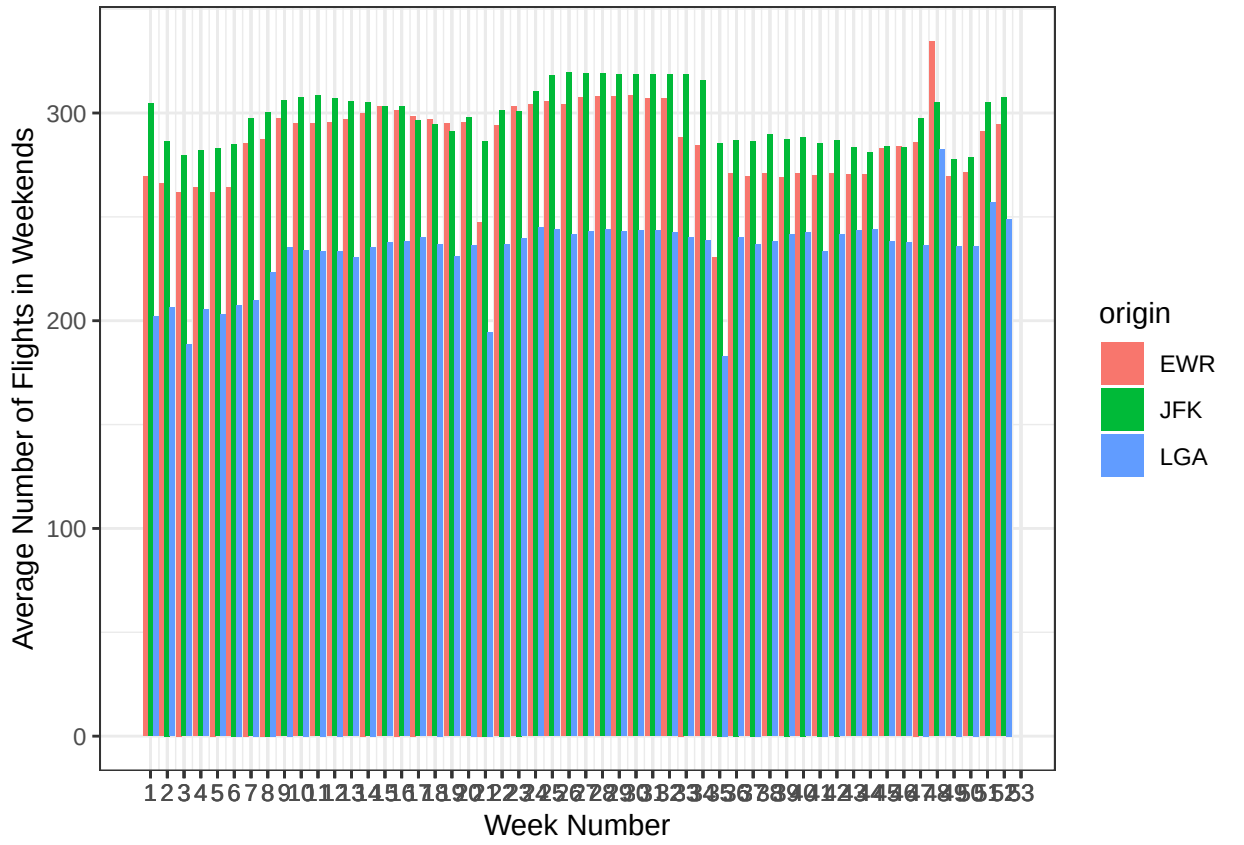



To extract the same information for the the weekends this time we exclude the weekdays from the original data:

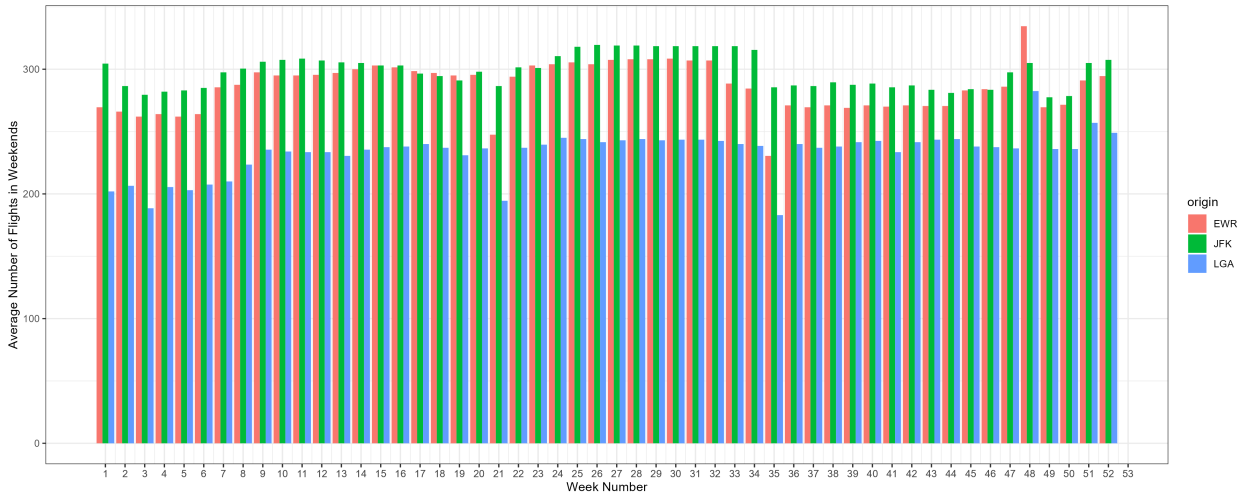
```
flights_weekends = flights_week %>%
  filter(!day %in% c("Mon", "Tue" , "Wed" , "Thu" , "Fri"))

flights_weekends_avg = flights_weekends %>%
  group_by(week, origin) %>%
  summarize(avg_flights = mean(flight_counts))
```

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

```
ggplot(flights_weekends_avg, aes(x = week, y = avg_flights, fill = origin)) +
  geom_col(position = "dodge") +
  labs(x = "Week Number", y = "Average Number of Flights in Weekends") +
  theme_bw() +
  scale_x_continuous(breaks = flights_weekdays_avg$week)
```




```
ggsave("average flight in weekends.png", width = 15, height = 6)
```

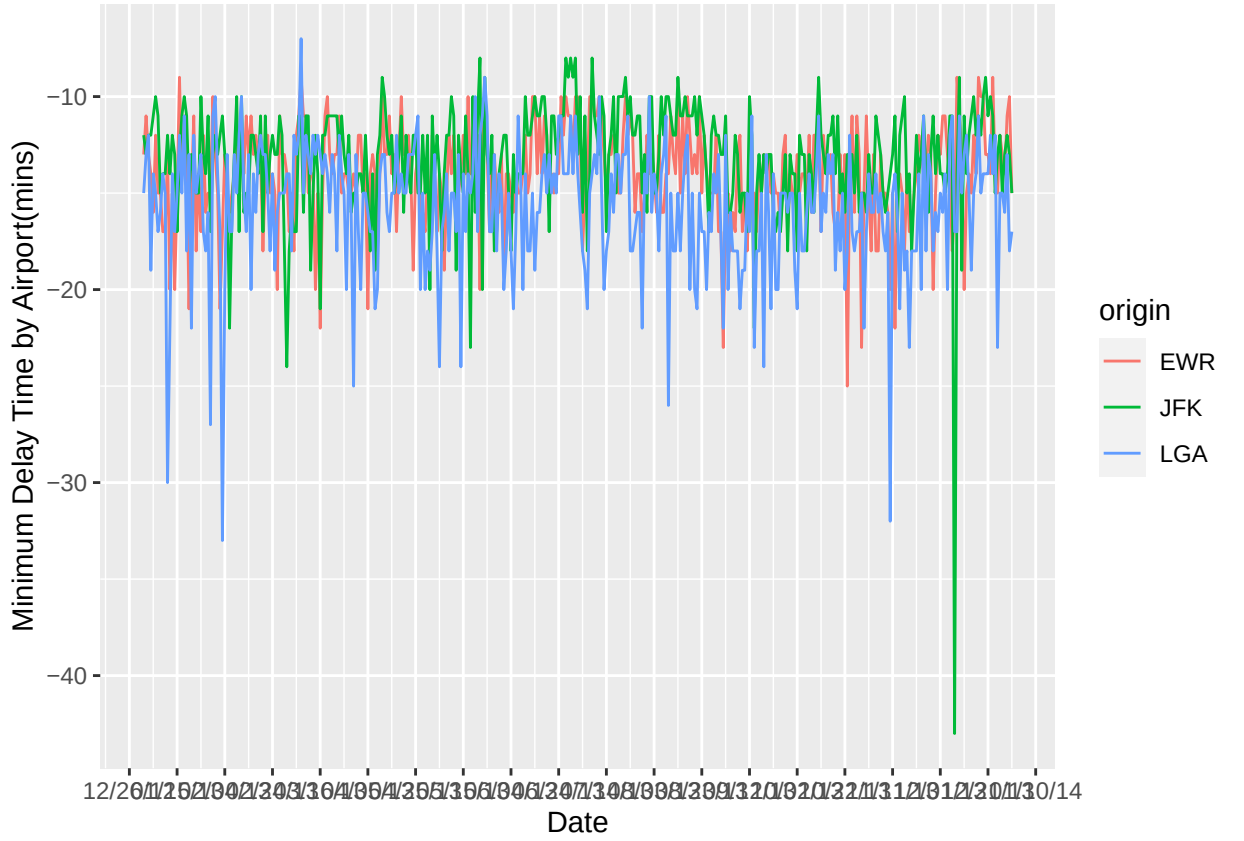



**2)** Now we use the group by method again to extract the minimum and maximum and average departure delay of the flights in each day

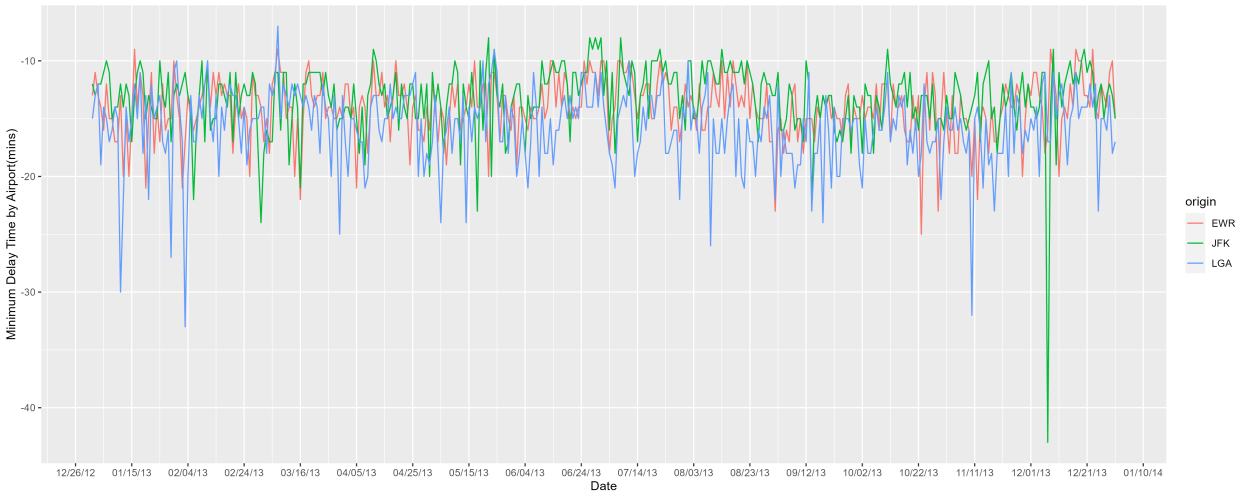
```
flight_delay_summary = nyc_flights %>%
  group_by(date , origin) %>%
  summarize(minimum_delay = min(dep_delay,na.rm = TRUE) , maximum_delay = max(dep_delay,na.rm = TRUE), a
```

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

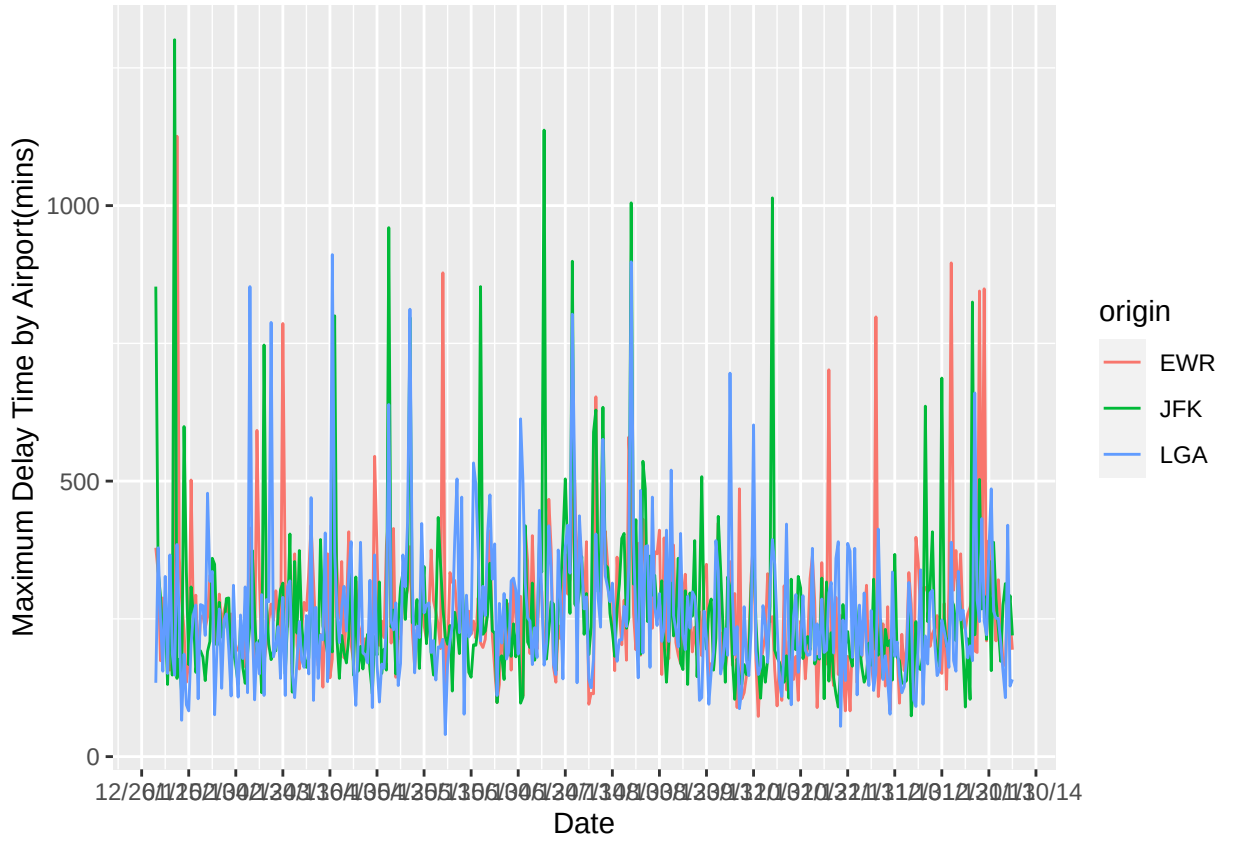
```
ggplot(data = flight_delay_summary, aes(x = date, y = minimum_delay, color = origin)) +
  labs(x = "Date", y = "Minimum Delay Time by Airport(mins)") +
  geom_line()+
  scale_x_date(date_breaks = "20 day", date_labels = "%D")
```




```
ggsave("minimum delay by airport.png", width = 15, height = 6)
```

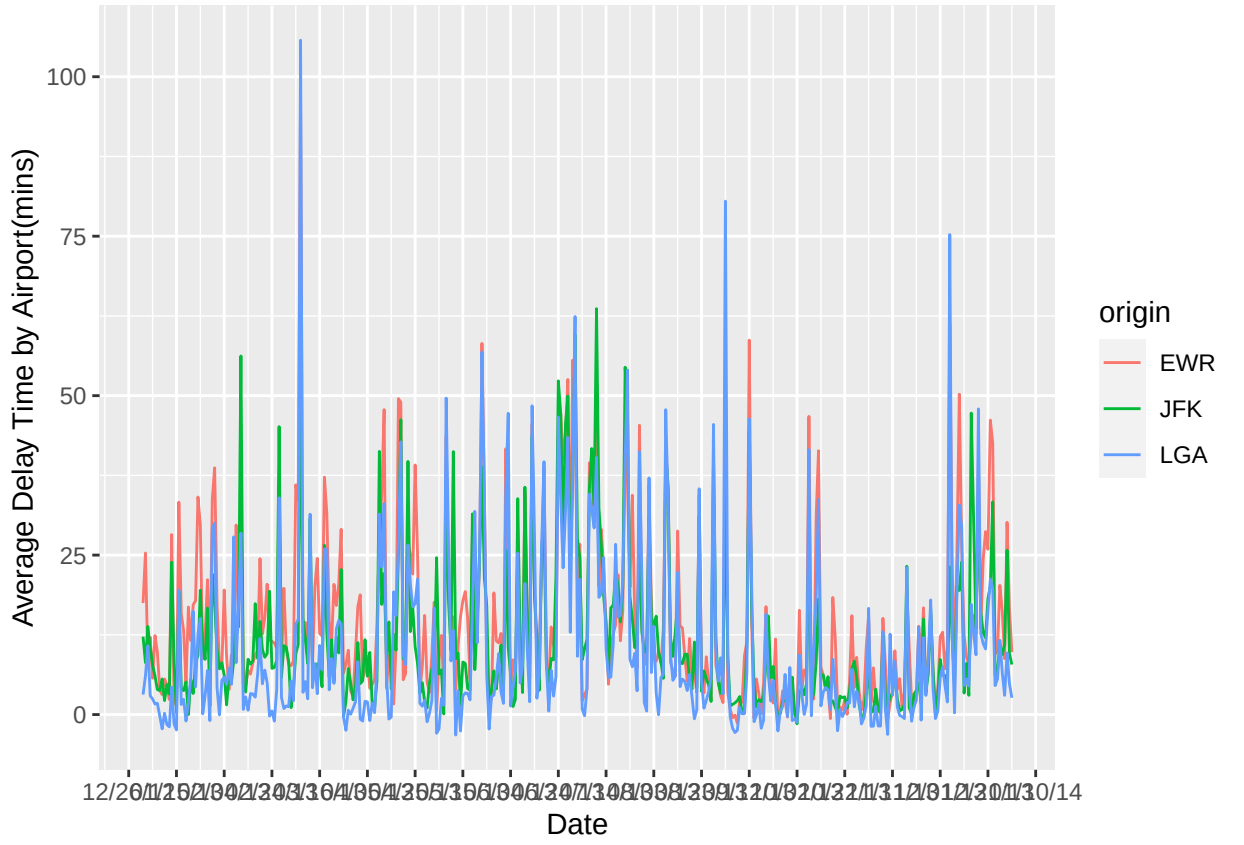


```
ggplot(data = flight_delay_summary, aes(x = date, y = maximum_delay, color = origin)) +
  labs(x = "Date", y = "Maximum Delay Time by Airport(mins)") +
  geom_line()+
  scale_x_date(date_breaks = "20 day", date_labels = "%D")
```

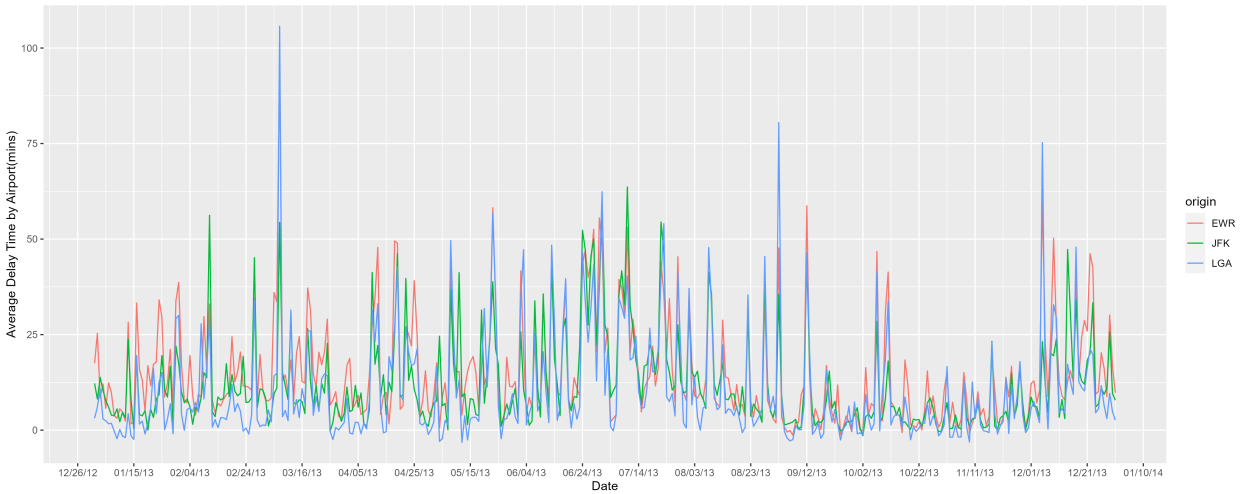



```
ggsave("maximum delay by airport.png", width = 15, height = 6)
```

```
ggplot(data = flight_delay_summary, aes(x = date, y = average_delay, color = origin)) +
  labs(x = "Date", y = "Average Delay Time by Airport(mins)") +
  geom_line()+
  scale_x_date(date_breaks = "20 day", date_labels = "%D")
```

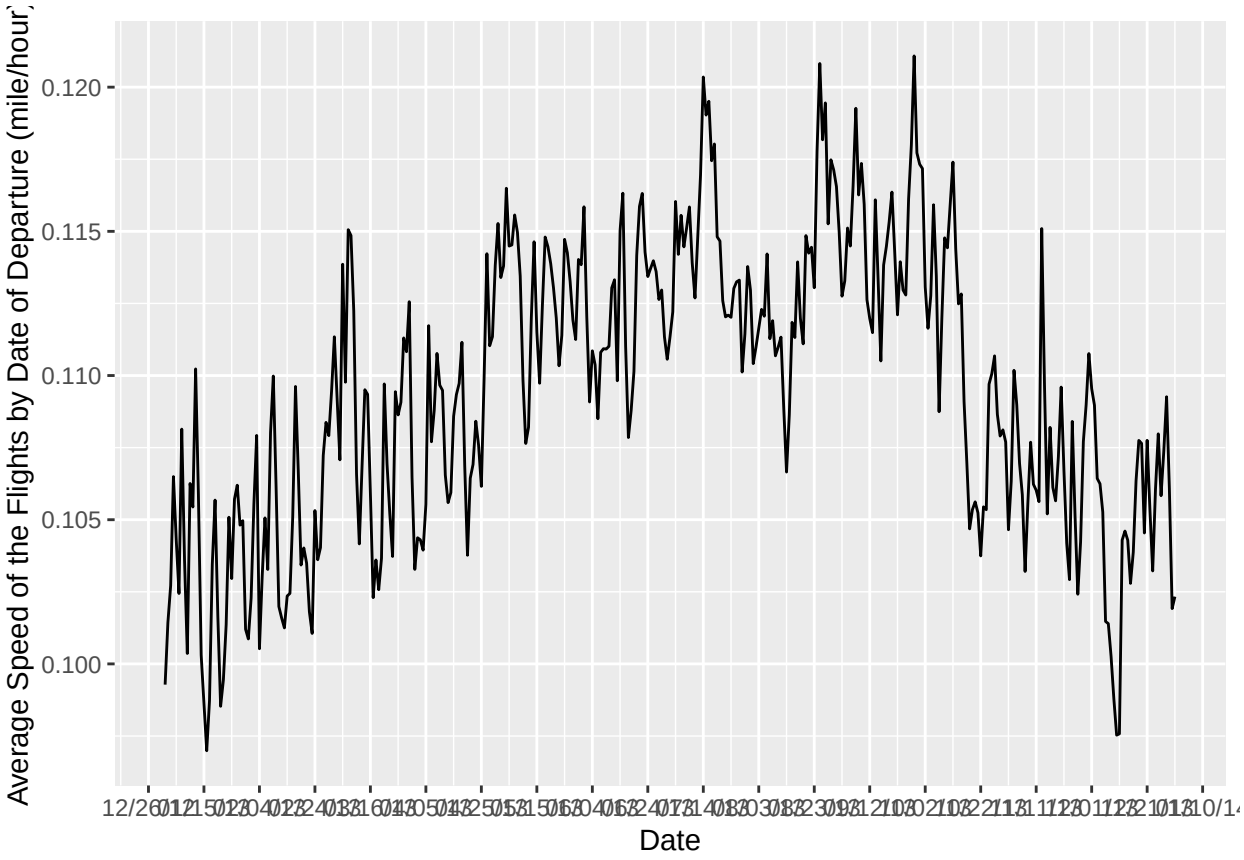



```
ggsave("average delay by airport.png", width = 15, height = 6)
```

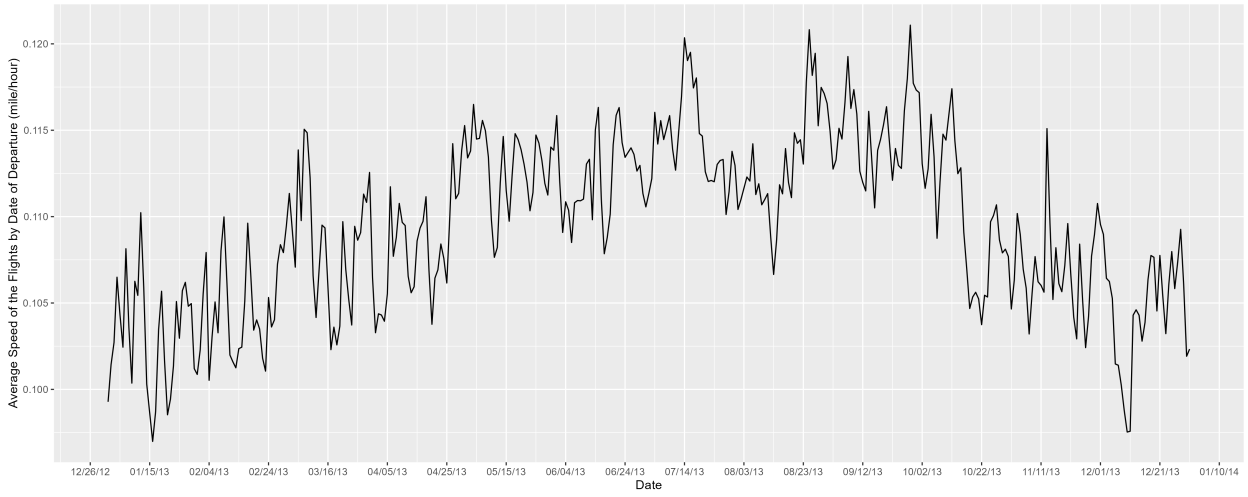



**3)** Next we calculate the average speed of each plane, and then average over for each day

```r
flights_speed = nyc_flights %>%
  mutate(speed = distance / air_time / 60)

flights_speed_summary = flights_speed %>%
  group_by(date) %>%
  summarize(average_speeds = mean(speed,na.rm = TRUE))

ggplot(data = flights_speed_summary, aes(x = date, y = average_speeds)) +
  labs(x = "Date", y = "Average Speed of the Flights by Date of Departure (mile/hour)") +
  geom_line()+
  scale_x_date(date_breaks = "20 day", date_labels = "%D")
```




```r
ggsave("average speed of the flights.png", width = 15, height = 6)
```



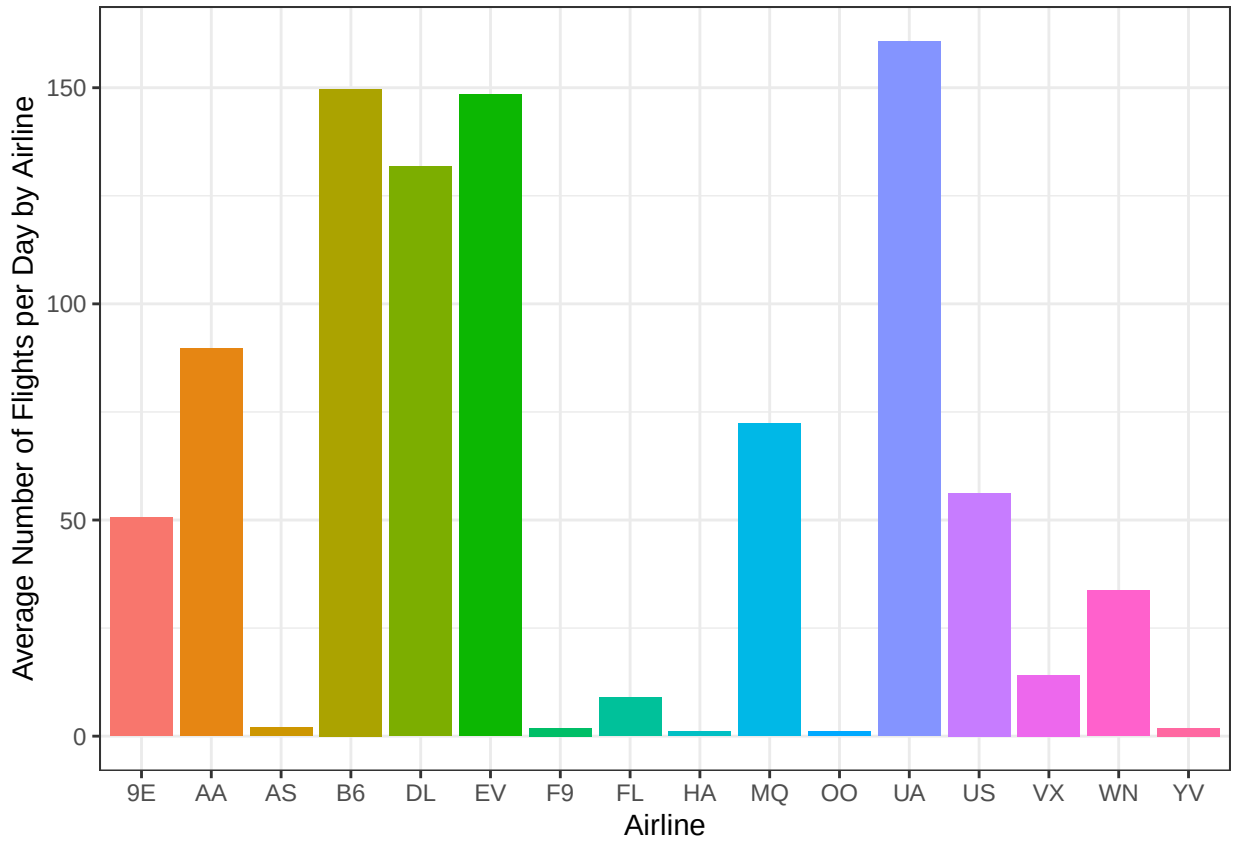

**4)** To get the information about the airlines, we group by the carrier and dates first

```
flights_airlines_daysummary = nyc_flights %>%
  group_by(carrier , date) %>%
  summarize(flight_count = n())
```

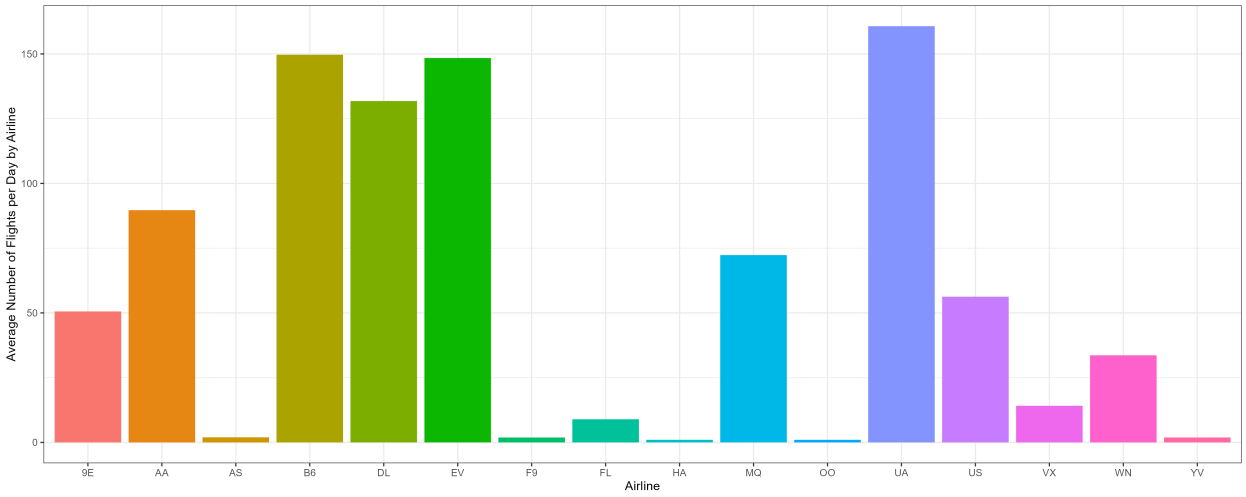
```
## `summarise()` has grouped output by 'carrier'. You can override using the
## `.groups` argument.
```

now to get the average number of flights per day of an airline, we group by airline again and this time we average over the flights count to summarize

```
airlines_average_day = flights_airlines_daysummary %>%
  group_by(carrier) %>%
  summarize(average_flight_perday = mean(flight_count , na.rm = TRUE))

ggplot(airlines_average_day , aes(x = carrier, y = average_flight_perday , fill = carrier)) +
  geom_col(position = "dodge") +
  labs(x = "Airline", y = "Average Number of Flights per Day by Airline") +
  theme_bw()+
  theme(legend.position = "none")
```




```
ggsave("average flights per day by airline.png", width = 15, height = 6)
```




As we see "UA" airline has the largest number of flights on average per day

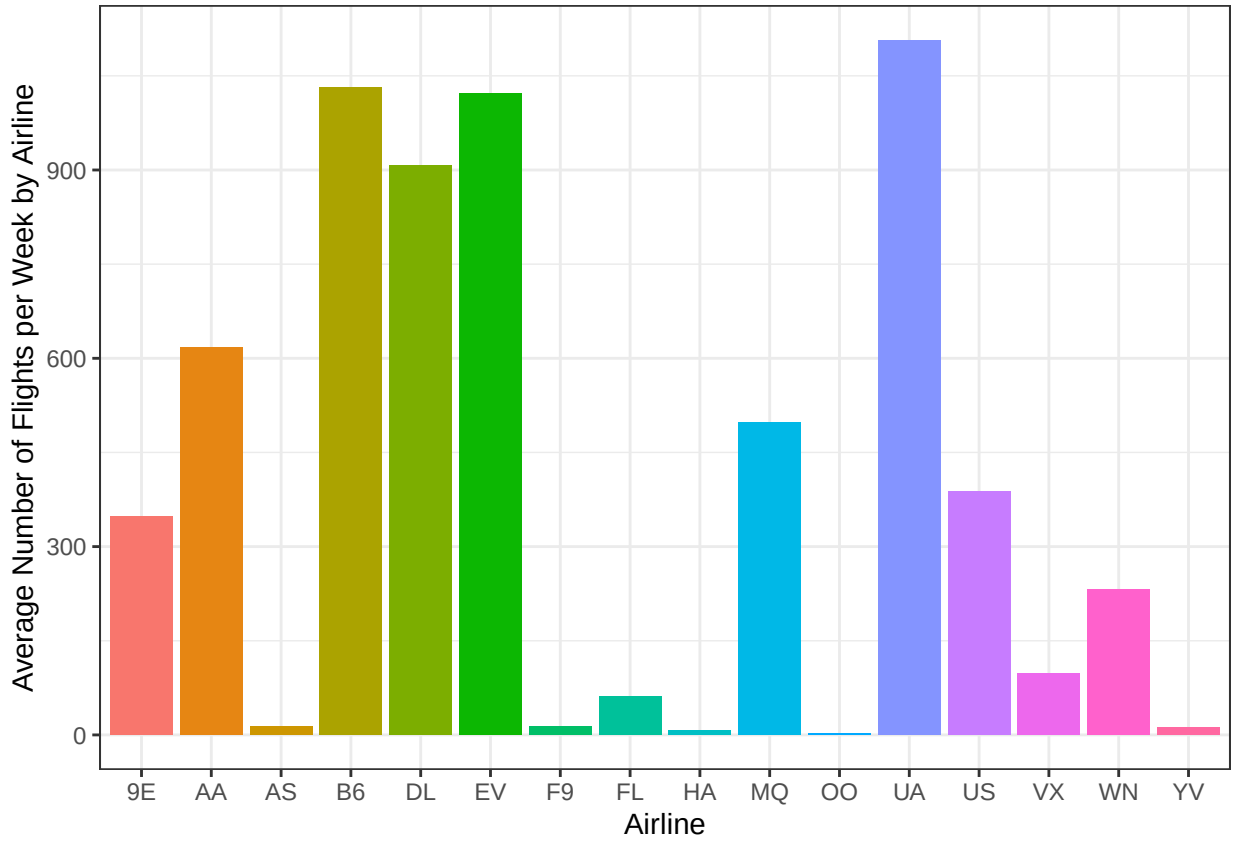
To get the same data for weeks, we add a week column to the initial tibble

```
flights_airlines_weeksummary = nyc_flights %>%
  mutate(week = week(date))

flights_airlines_weeksummary = flights_airlines_weeksummary %>%
```

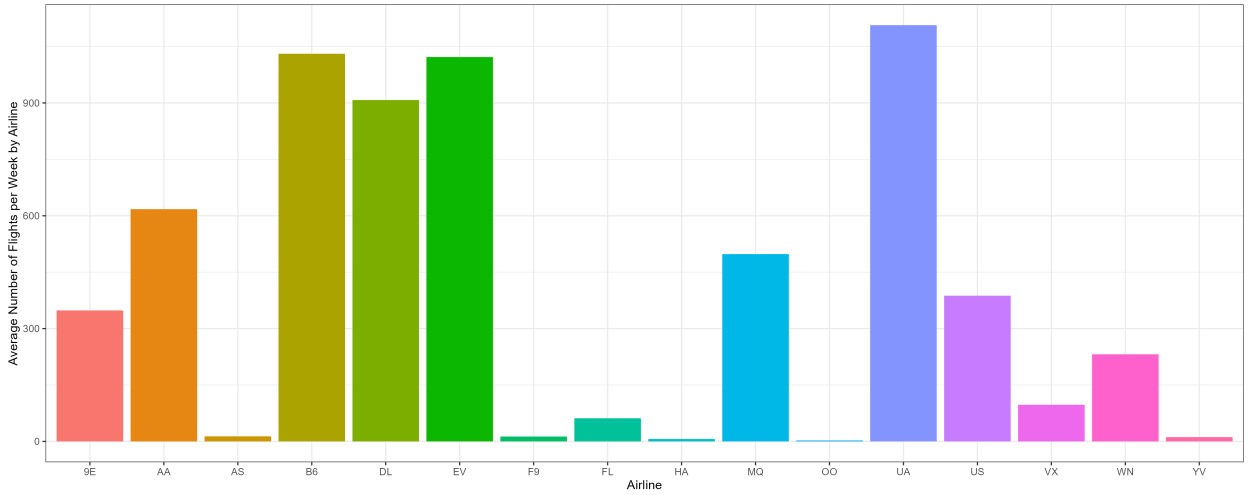
```
  group_by(carrier , week) %>%
  summarize(flights_count = n())
```

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

```
airlines_average_week = flights_airlines_weeksummary %>%
  group_by(carrier) %>%
  summarize(average_flight_perweek = mean(flights_count , na.rm = TRUE))

ggplot(airlines_average_week , aes(x = carrier, y = average_flight_perweek , fill = carrier)) +
  geom_col(position = "dodge") +
  labs(x = "Airline", y = "Average Number of Flights per Week by Airline") +
  theme_bw()+
  theme(legend.position = "none")
```




```
ggsave("average flights per week by airline.png", width = 15, height = 6)
```



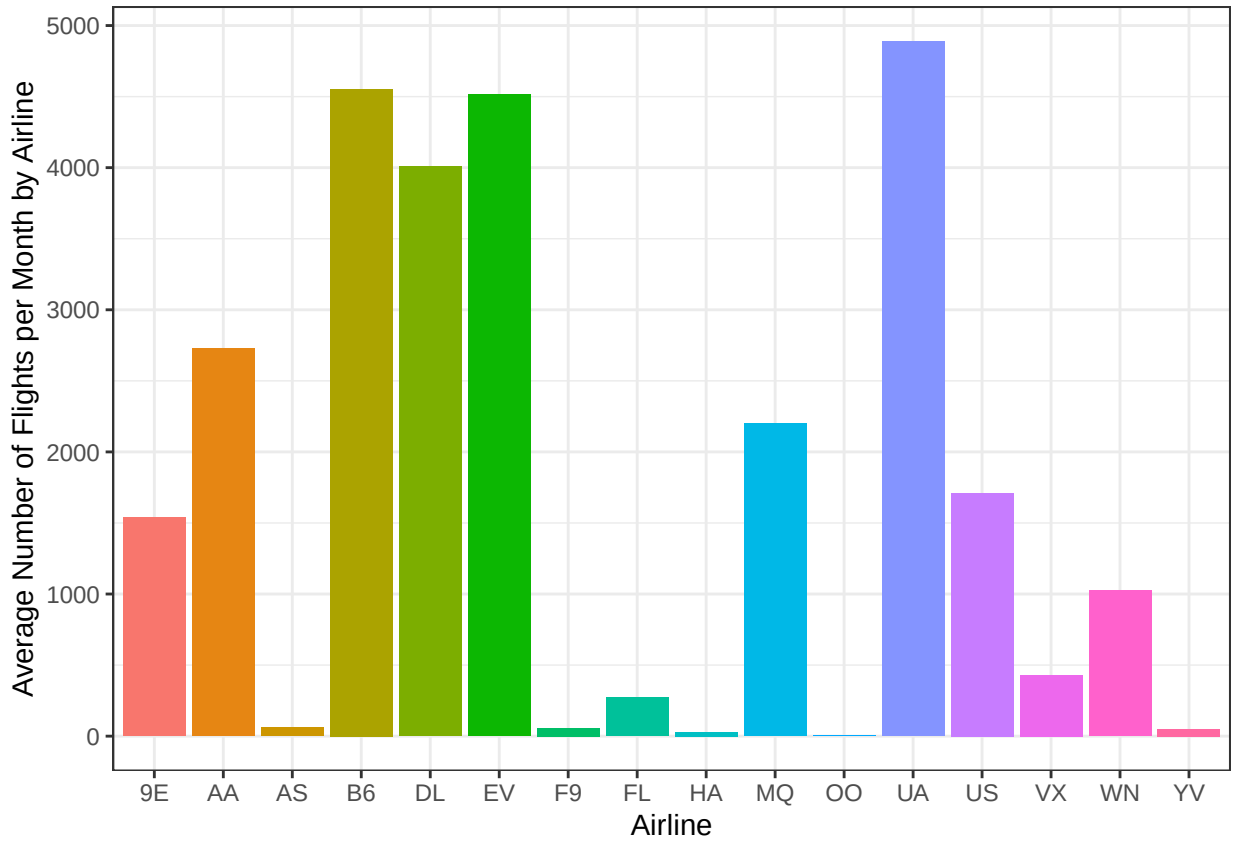

"UA" airlines has also the largest number of flights per week.

The same can be done for months:

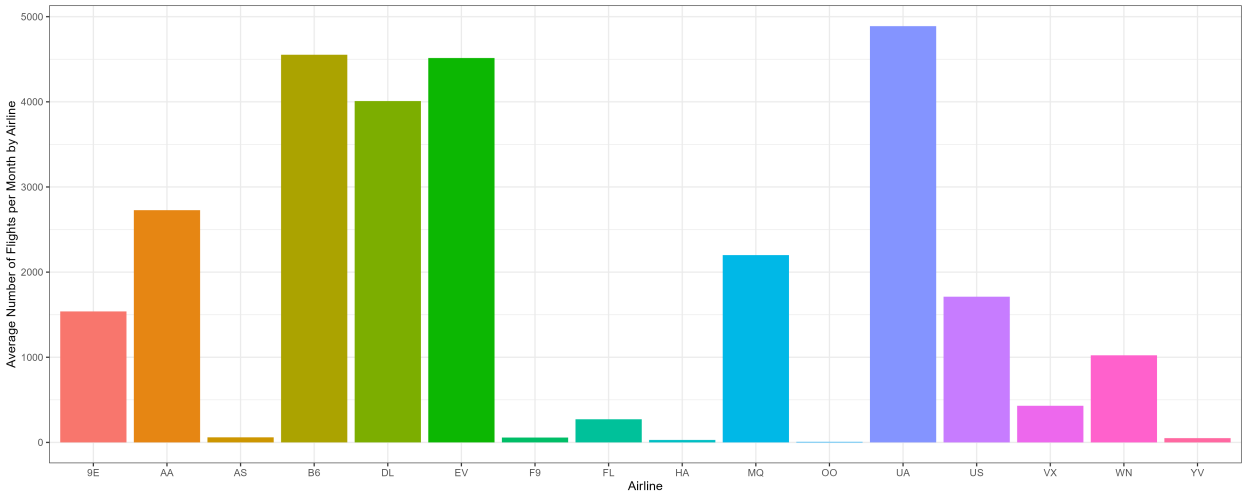
```
flights_airlines_monthsummary = nyc_flights %>%
  mutate(month = month(date))

flights_airlines_monthsummary = flights_airlines_monthsummary %>%
  group_by(carrier , month) %>%
  summarize(flights_count = n())
```

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

```
airlines_average_month = flights_airlines_monthsummary %>%
  group_by(carrier) %>%
  summarize(average_flight_permonth = mean(flights_count , na.rm = TRUE))

ggplot(airlines_average_month , aes(x = carrier, y = average_flight_permonth , fill = carrier)) +
  geom_col(position = "dodge") +
  labs(x = "Airline", y = "Average Number of Flights per Month by Airline") +
  theme_bw()+
  theme(legend.position = "none")
```




```
ggsave("average flights per month by airline.png", width = 15, height = 6)
```




As we see the lowest number of flights per month belongs to "OO" airlines.

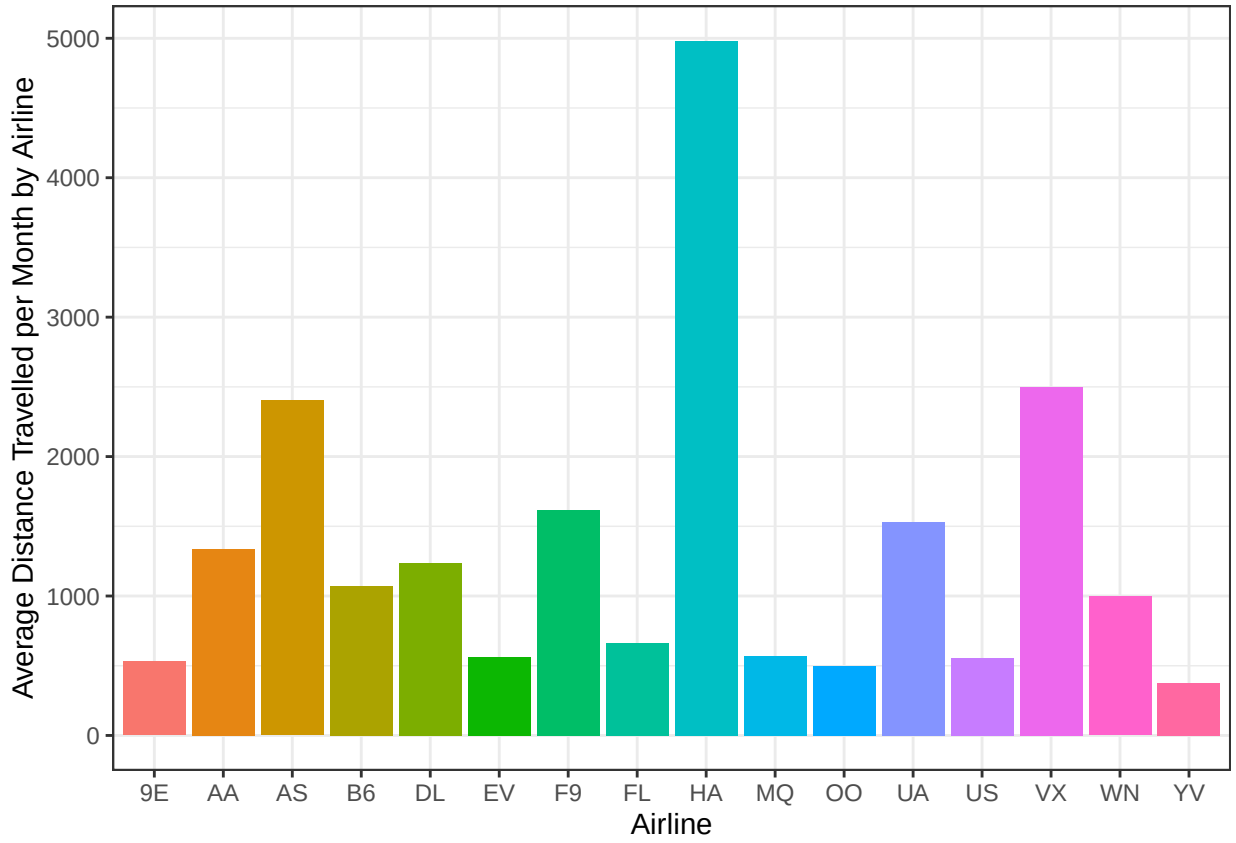
to get information about distance we return to the original tibble

```
airlines_distance_per_month = nyc_flights %>%
  mutate(month = month(date))

airlines_distance_summary = airlines_distance_per_month %>%
```

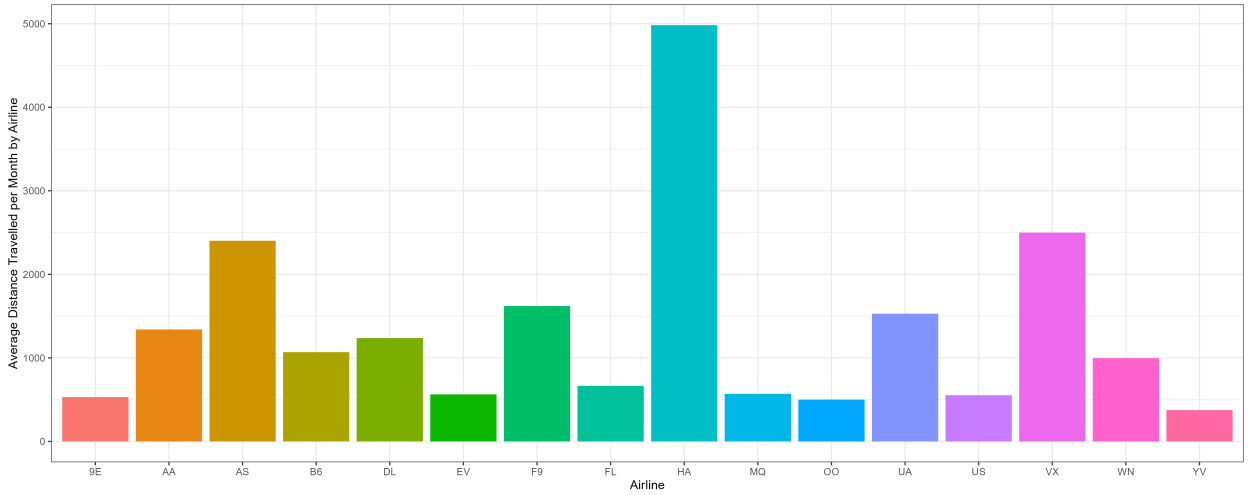
```
  group_by(carrier , month) %>%
  summarize(distance_travelled = sum(distance))
```

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

```
airlines_distance_per_month = airlines_distance_per_month %>%
  group_by(carrier) %>%
  summarize(average_distance_travelled = mean(distance))

ggplot(airlines_distance_per_month, aes(x = carrier, y = average_distance_travelled , fill = carrier)) +
  geom_col(position = "dodge") +
  labs(x = "Airline", y = "Average Distance Travelled per Month by Airline") +
  theme_bw()+
  theme(legend.position = "none")
```




```
ggsave("distance travelled per month by airline.png", width = 15, height = 6)
```




As we see the "HA" airline has the largest value.