2022MCS120009_Assignment02

Santhosh Kumar N

October 11th 2022

Install packages

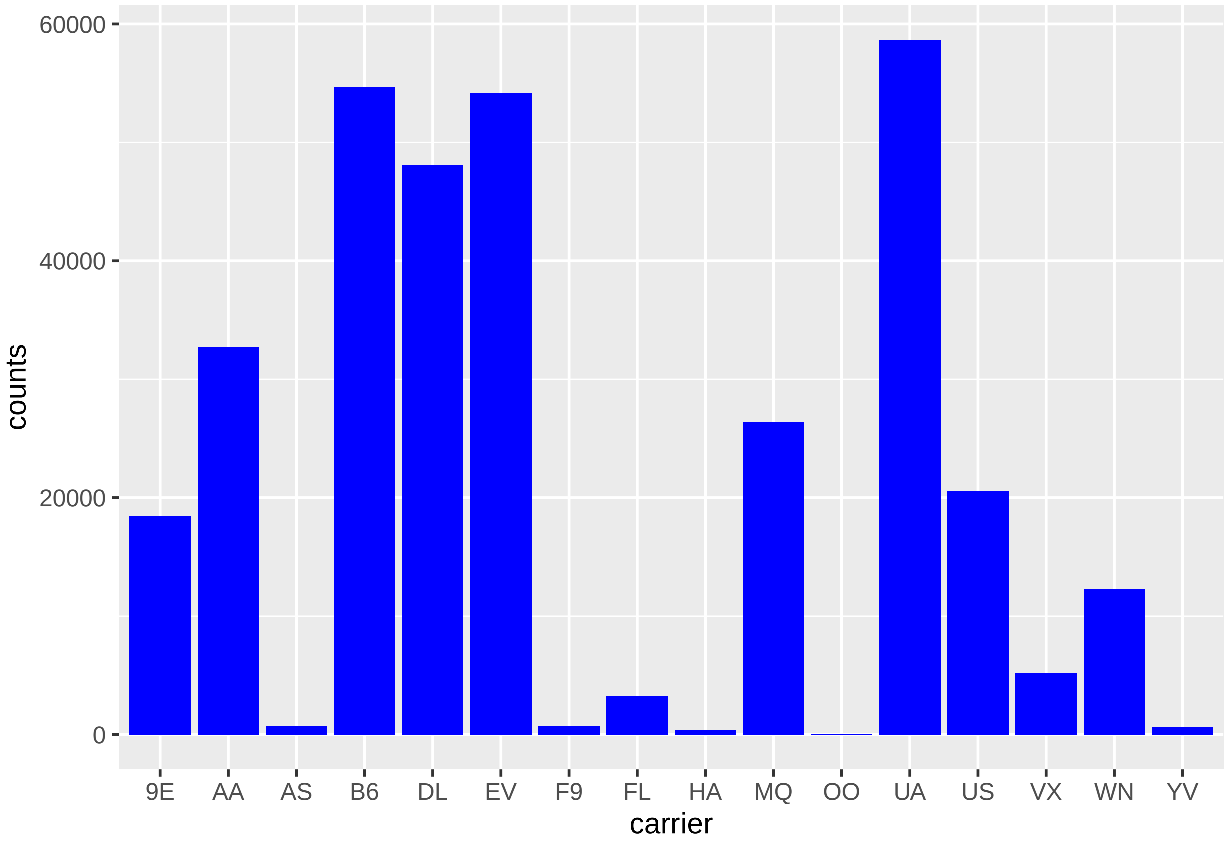
```
library('nycflights13')
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
```

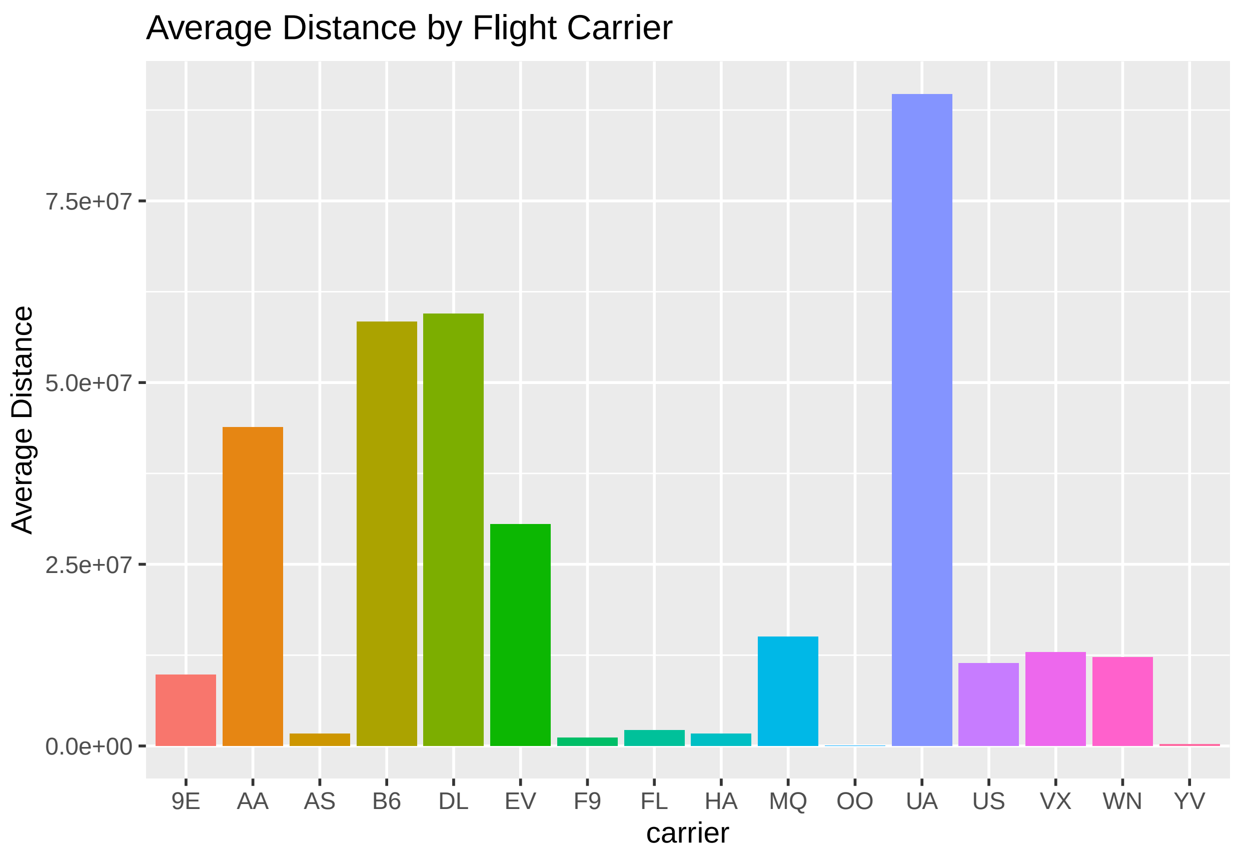
1. Obtain the result as shown below:

```
df = flights

data  = df %>% group_by(carrier) %>% summarize(counts = n())

ggplot(data, aes(x=carrier, y=counts)) + geom_bar(stat = "identity",fill="blue")
```



2. Obtain the plot of average distance travelled by flight carriers as shown below:

```
data = aggregate(df$distance, by=list(carrier=df$carrier), FUN=sum)

ggplot(data=data, aes(x=carrier, y=x,fill=carrier))+ geom_bar(stat="identity") +
  theme(legend.position="none") +
  ylab("Average Distance") +
  ggtitle("Average Distance by Flight Carrier")
```



## 3. Obtain the result shown below:

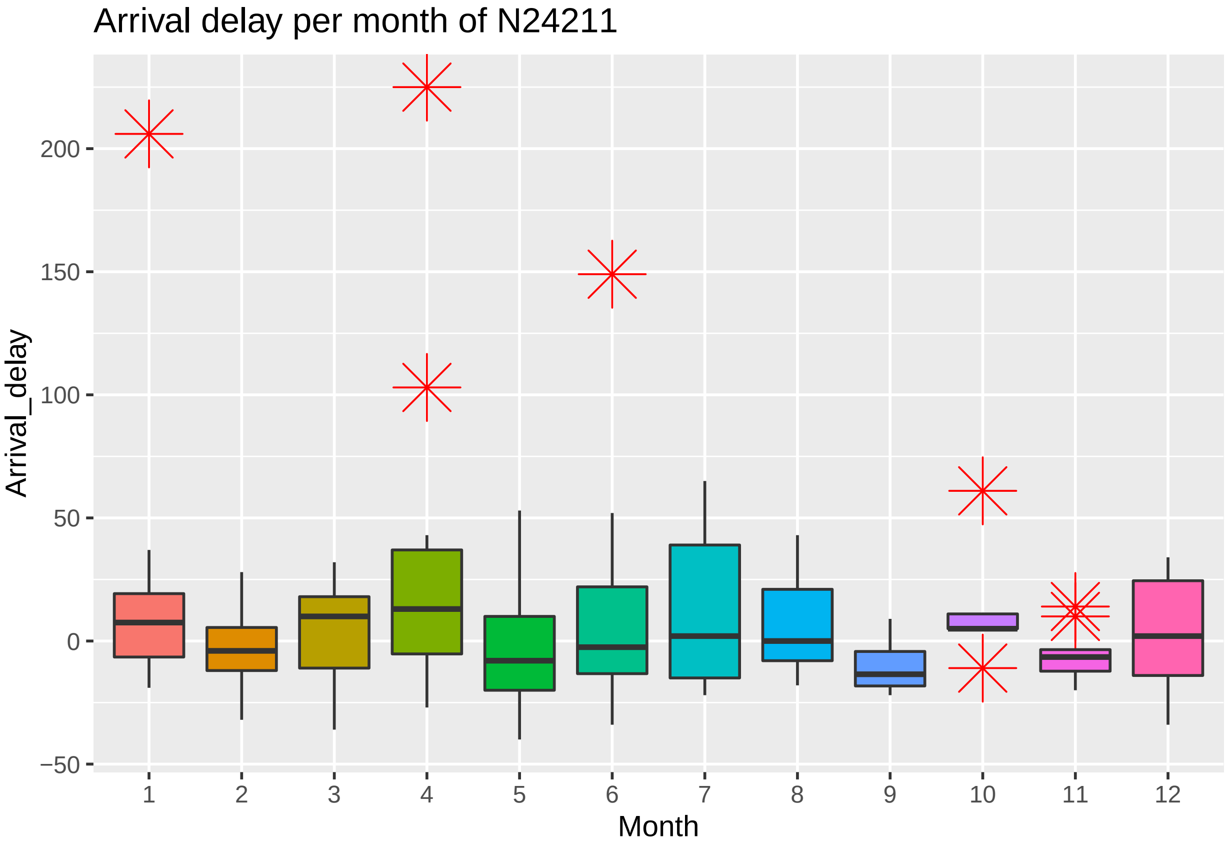
```
filter(flights, month == 8 & origin == "JFK" & dest == "FLL" ) %>% arrange(time_hour)
```

```
## # A tibble: 336 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     8     1      556        600      -4     849     850      -1 B6
##  2  2013     8     1      753        800      -7    1056    1104      -8 DL
##  3  2013     8     1      800        800       0    1129    1053      36 B6
##  4  2013     8     1     1027       1029      -2    1328    1320       8 B6
##  5  2013     8     1     1242       1239       3    1541    1534       7 B6
##  6  2013     8     1     1443       1430      13    1751    1735      16 B6
##  7  2013     8     1     1532       1535      -3    1843    1901     -18 DL
##  8  2013     8     1     1629       1630      -1    2006    1945      21 B6
##  9  2013     8     1     1928       1901      27    2243    2213      30 B6
## 10  2013     8     1     2340       2135     125     232      30     122 B6
## # ... with 326 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

4. Obtain the plot as shown below:

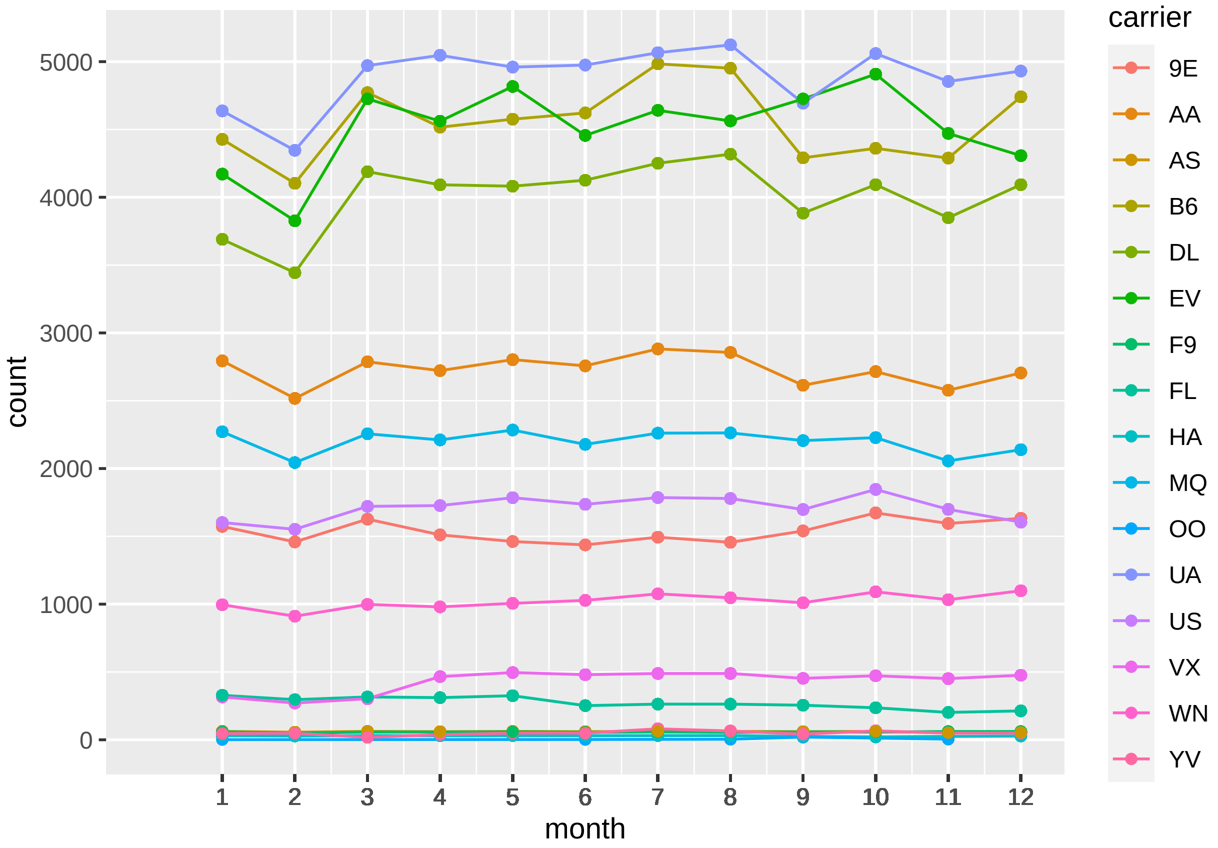
```
data = flights
```

```
df = subset(data,tailnum == 'N24211',select = c('month','arr_delay','tailnum'))

df$month <- as.factor(df$month)


ggplot(df,aes(x=month, y=arr_delay, fill=month)) +
  geom_boxplot(outlier.shape = 8,outlier.size = 8,outlier.colour = 'red') +
  theme(legend.position="none") + xlab("Month") + ylab("Arrival_delay") +
  ggtitle("Arrival delay per month of N24211")
```



5.Obtain the plot as shown below::

```
df= select(flights,c('carrier','month'))

df_with_count<- df %>% group_by(month,carrier) %>% mutate(count=n())

ggplot(data=df_with_count, aes(x=month, y=count, group=carrier,color = carrier)) +
geom_line() + scale_x_continuous(breaks=df_with_count$month,limits=c(0, 12)) +
geom_point()
```



## 6. Write the R code for the following output.

```
head(filter(select(flights,c("carrier","dep_delay","air_time","distance")),carrier== "AA") %>%
       mutate(air_time_hours = air_time /60))
```

```
## # A tibble: 6 x 5
##   carrier dep_delay air_time distance air_time_hours
##   <chr>       <dbl>    <dbl>    <dbl>          <dbl>
## 1 AA              2      160     1089           2.67
## 2 AA             -2      138      733           2.3 
## 3 AA             -1      257     1389           4.28
## 4 AA             -4      152     1085           2.53
## 5 AA             13      153     1096           2.55
## 6 AA             -2      192     1598           3.2 
```

7.Obtain the result as shown below:

```
dff = filter(flights,month == '6',day>=20)

dff %>% arrange(desc(day))
```

```
## # A tibble: 10,424 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>  
##  1  2013     6    30       12       2231     101     352     226      86 B6     
##  2  2013     6    30       21       2300      81     116       8      68 B6     
##  3  2013     6    30       23       2055     208     123    2230     173 WN     
##  4  2013     6    30       25       2359      26      413      350      23 B6
##  5  2013     6    30       43       2250     113      150       14      96 B6
##  6  2013     6    30       56       2245     131      201        3     118 B6
##  7  2013     6    30      116       2359      77      451      344      67 B6
##  8  2013     6    30      153       2245     188      422      135     167 B6
##  9  2013     6    30      217       2359     138      545      340     125 B6
## 10  2013     6    30      525        500      25      703      640      23 US
## # ... with 10,414 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

8. Calculate the number of planes only flew one route but flew that route more than 17 times?

```
# flights with frequency higher than 17
df = flights %>% count(tailnum,origin,dest)

df %>% filter(n>17)
```

```
## # A tibble: 3,642 x 4
##    tailnum origin dest      n
##    <chr>   <chr>  <chr> <int>
##  1 N0EGMQ  EWR    ORD      50
##  2 N0EGMQ  LGA    ATL      61
##  3 N0EGMQ  LGA    BNA      55
##  4 N0EGMQ  LGA    CLT      52
##  5 N0EGMQ  LGA    DTW      24
##  6 N0EGMQ  LGA    MSP      42
##  7 N0EGMQ  LGA    RDU      27
##  8 N102UW  EWR    CLT      23
##  9 N103US  EWR    CLT      24
## 10 N104UW  EWR    CLT      25
## # ... with 3,632 more rows
```

```
# flights origin-destination for each flights
df %>% count(tailnum)
```

```
## # A tibble: 4,044 x 2
##    tailnum     n
##    <chr>   <int>
##  1 D942DN      3
##  2 N0EGMQ     14
##  3 N10156     41
##  4 N102UW      3
##  5 N103US      3
##  6 N104UW      3
##  7 N10575     43
##  8 N105UW      4
##  9 N107US      4
## 10 N108UW      3
## # ... with 4,034 more rows
```

```
# filghts with single route
df %>% filter(n == 1)
```

```
## # A tibble: 12,150 x 4
##    tailnum origin dest      n
##    <chr>   <chr>  <chr> <int>
##  1 D942DN  JFK    MCO       1
##  2 D942DN  LGA    MCO       1
##  3 N0EGMQ  LGA    XNA       1
##  4 N10156  EWR    GSO       1
##  5 N10156  EWR    GSP       1
##  6 N10156  EWR    IAD       1
##  7 N10156  EWR    MHT       1
##  8 N10156  EWR    MSN       1
##  9 N10156  EWR    ORF       1
## 10 N10156  EWR    SDF       1
## # ... with 12,140 more rows
```

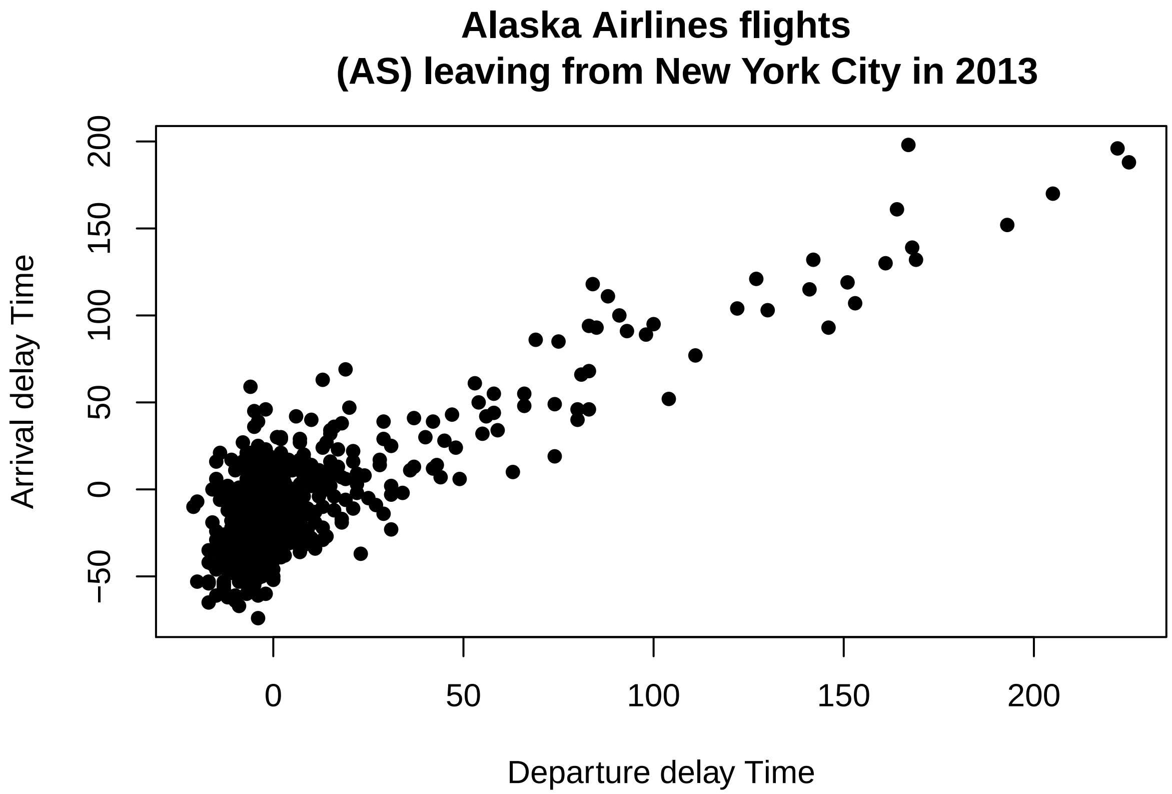
9.a Find out the number of Alaska Airlines flights (AS) leaving from New York City in 2013.

```
nrow(filter(flights, carrier == 'AS' & origin == 'EWR'))
```

```
## [1] 714
```

9 b. Obtain the result as shown below:

```
data = flights

Alaska <- flights[flights$carrier == "AS", ]

plot(Alaska$dep_delay, Alaska$arr_delay,main="Alaska Airlines flights
     (AS) leaving from New York City in 2013",
     xlab = "Departure delay Time" ,ylab = "Arrival delay Time",pch = 16)
```



##

10. Find out the total distance for all flights in the month of December ? What was the average distance per flight?

```
total_distance = sum(filter(flights, month == '12')$distance)

cat("Total Distance:" ,total_distance,"\n")
```

```
## Total Distance: 29954084
```

```
total_flights = nrow(filter(flights, month == '12'))

cat("Total Flights:",total_flights,"\n")
```

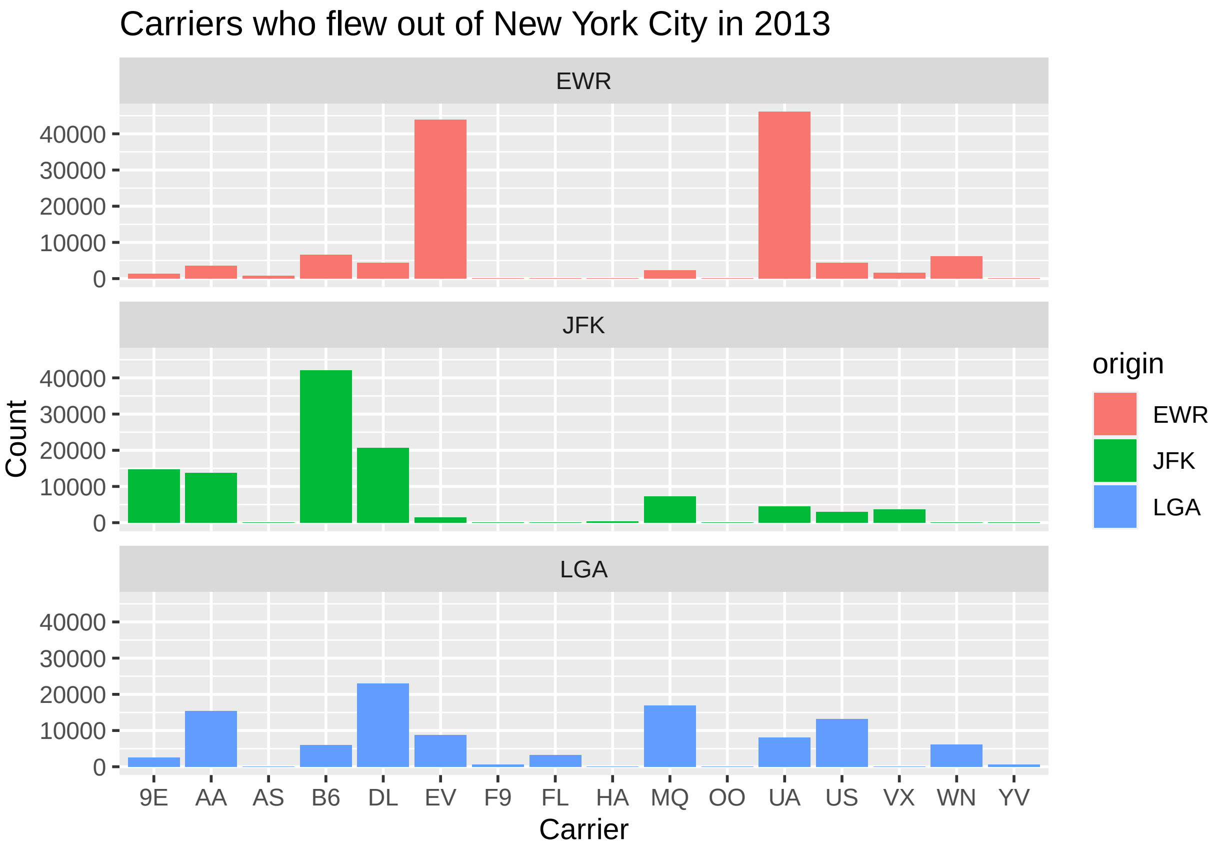
```
## Total Flights: 28135
```

```
avg_dis_flights = total_distance / total_flights

cat("Average distance per flight :", avg_dis_flights,"\n")
```

```
## Average distance per flight : 1064.656
```

11. Obtain the result as shown below:

```
data = flights

carrier.freq <- table(flights$carrier)
carrier.freq <- as.data.frame(carrier.freq)
colnames(carrier.freq) <- c("carrier", "number")

carrier.origin <- table(flights$origin, flights$carrier)
carrier.origin <- as.data.frame(carrier.origin)

colnames(carrier.origin) <- c("origin", "carrier", "number")

ggplot(data = carrier.origin, mapping = aes(x = carrier, y = number, fill = origin)) +
  geom_col() +
  facet_wrap(~ origin, ncol = 1) +
  labs(x = "Carrier", y = "Count",
       title = "Carriers who flew out of New York City in 2013")
```



## 12. Find out the flights that departed in June or July.

```
df = filter(flights, month == 6 | month == 7  ) %>% count(month)

df
```

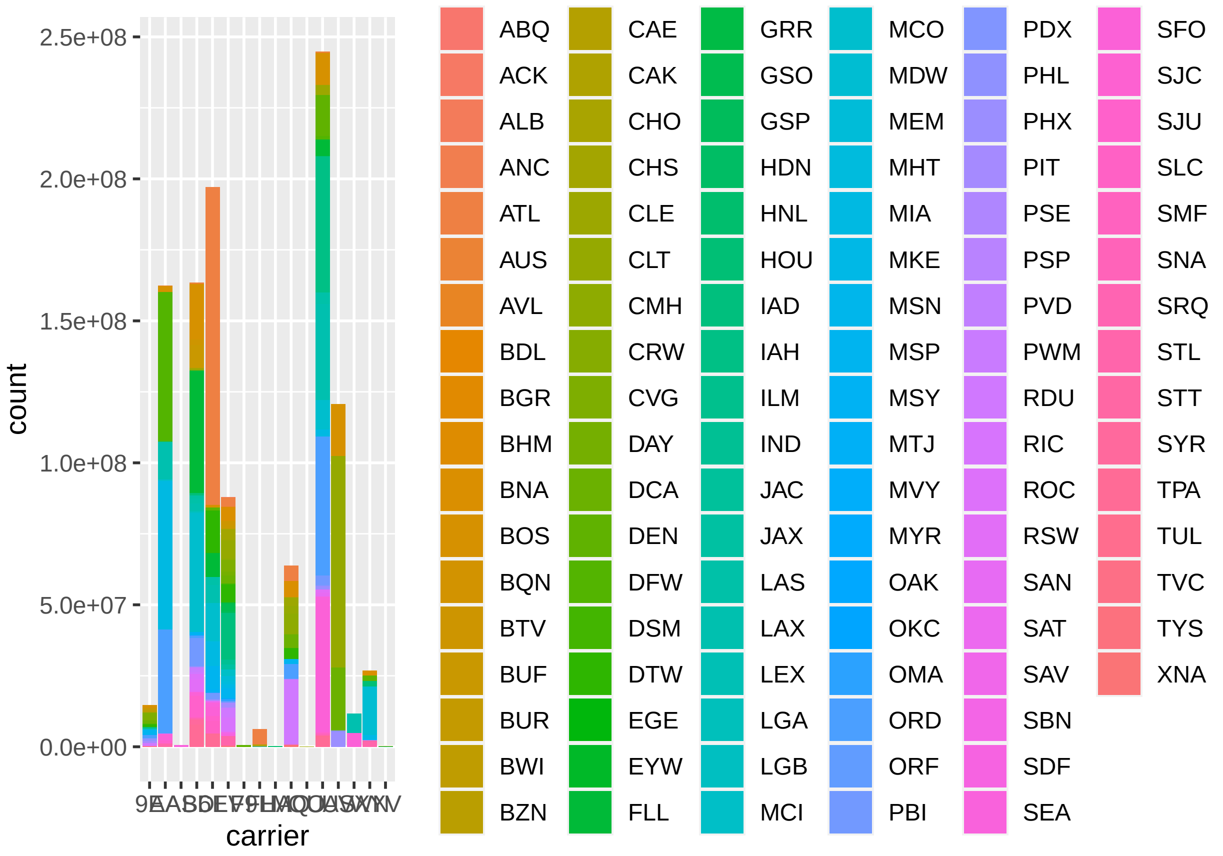
```
## # A tibble: 2 x 2
##   month     n
##   <int> <int>
## 1     6 28243
## 2     7 29425
```

```
cat("Total flights for june and July:",sum(df$n))
```

```
## Total flights for june and July: 57668
```

13. Obtain the following plot about the number of flights for each carrier and their destination.

```
df= select(flights,c('carrier','dest'))

df_with_count<-df %>% group_by(carrier,dest) %>% mutate(count=n())

ggplot(df_with_count, aes(fill = dest, y = count, x = carrier))+
  geom_bar(position = "stack", stat = "identity")+
  theme(plot.title = element_text(hjust = 0.5))
```



14 a Find out the flights that were most delayed on arrival and the flights that left just before the time .

```
filter(flights,dep_time < sched_dep_time)  %>% arrange(desc(arr_delay))
```

```
## # A tibble: 184,782 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     9      641        900    1301    1242    1530    1272 HA
##  2  2013     6    15     1432       1935    1137    1607    2120    1127 MQ
##  3  2013     1    10     1121       1635    1126    1239    1810    1109 MQ
##  4  2013     9    20     1139       1845    1014    1457    2210    1007 AA
##  5  2013     7    22      845       1600    1005    1044    1815     989 MQ
##  6  2013     4    10     1100       1900     960    1342    2211     931 DL
##  7  2013    12     5      756       1700     896    1058    2020     878 AA
##  8  2013     5     3     1133       2055     878    1250    2215     875 MQ
##  9  2013    12    14      830       1845     825    1210    2154     856 DL
## 10  2013     5    19      713       1700     853    1007    1955     852 AA
## # ... with 184,772 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

14 b Find out the flights that weren't delayed on arrival or departure by more than three hours.

```
filter(flights,dep_delay <180 & arr_delay <180) %>% arrange(desc(dep_delay))
```

```
## # A tibble: 322,854 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     2      923        624     179    1051     758     173 EV
##  2  2013     1    22     2009       1710     179    2112    1820     172 EV
##  3  2013     1    31     2354       2055     179     144    2250     174 MQ
##  4  2013    10     7     2329       2030     179      41    2205     156 WN
##  5  2013    11     1     2329       2030     179      34    2205     149 WN
##  6  2013    11    17     2234       1935     179      32    2143     169 EV
##  7  2013    12    11     2104       1805     179    2355    2123     152 UA
##  8  2013    12    12     1308       1009     179    1555    1319     156 UA
##  9  2013    12    17     2358       2059     179     128    2244     164 B6
## 10  2013    12    22     2146       1847     179      14    2121     173 UA
## # ... with 322,844 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

15. Find out the flights which are flying to "IAH" or "HOU",that were operated by carriers UA,AA and DL.

```
filter(filter(flights, dest == "IAH" | dest == "HOU") ,
       carrier == "UA" | carrier == "AA" |carrier == "DL")
```

```
## # A tibble: 7,198 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      623        627      -4     933     932       1 UA
##  4  2013     1     1      728        732      -4    1041    1038       3 UA
##  5  2013     1     1      739        739       0    1104    1038      26 UA
##  6  2013     1     1      908        908       0    1228    1219       9 UA
##  7  2013     1     1     1028       1026       2    1350    1339      11 UA
##  8  2013     1     1     1044       1045      -1    1352    1351       1 UA
##  9  2013     1     1     1114        900     134    1447    1222     145 UA
## 10  2013     1     1     1205       1200       5    1503    1505      -2 UA
## # ... with 7,188 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

16 a Find out the first departure for each day from NYC airport in 2013.

```
filter(flights,origin== 'EWR') %>% group_by(month,day) %>% summarise(First_Dept = min(dep_time))
```

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

```
## # A tibble: 365 x 3
## # Groups:   month [12]
##    month   day First_Dept
##    <int> <int>      <int>
##  1     1     1         NA
##  2     1     2         NA
##  3     1     3         NA
##  4     1     4         NA
##  5     1     5         NA
##  6     1     6         NA
##  7     1     7        454
##  8     1     8         NA
##  9     1     9         NA
## 10     1    10         NA
## # ... with 355 more rows
```

16 b Calculate the total number of flights that flew out daily and monthly from NYC airport in 2013.

```
filter(flights,origin== 'EWR') %>% group_by(month,day) %>% summarise(dailyFlightCount = n())
```

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

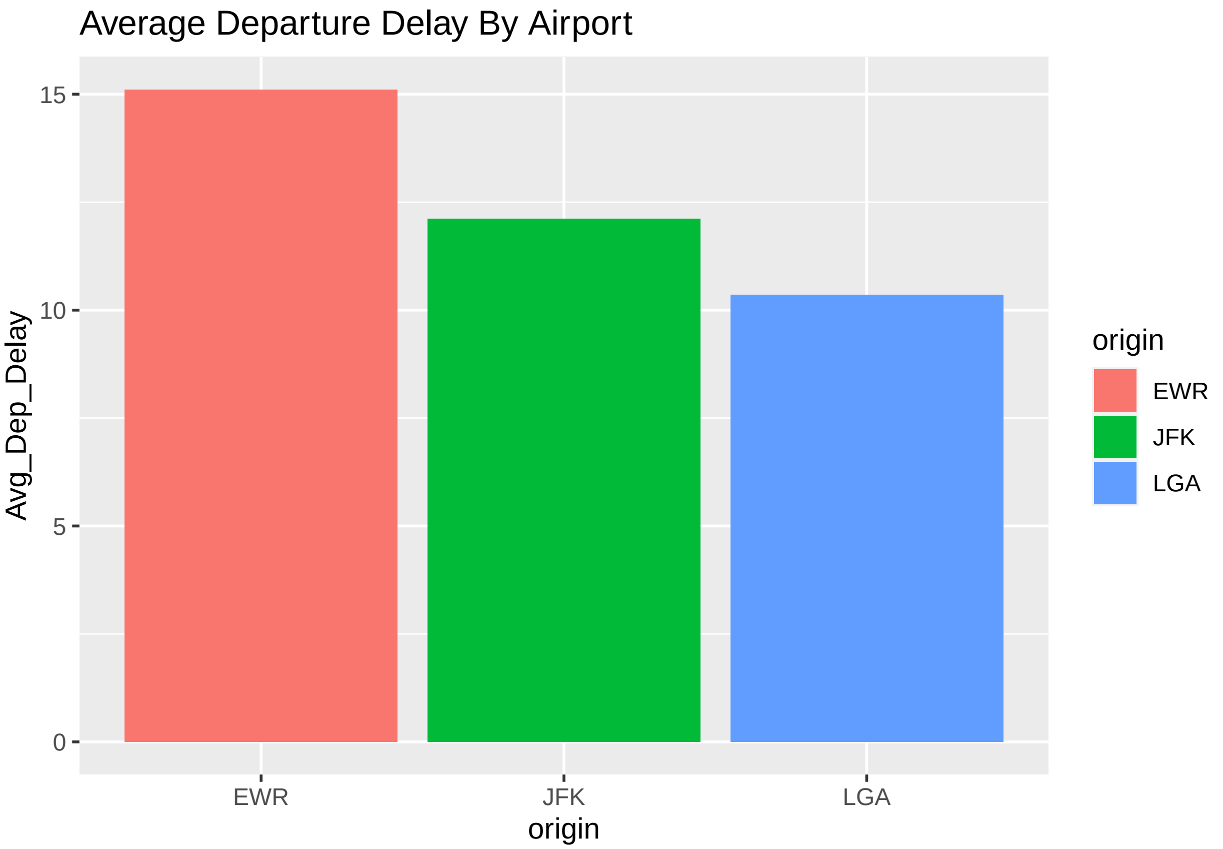
```
## # A tibble: 365 x 3
## # Groups:   month [12]
##    month   day dailyFlightCount
##    <int> <int>            <int>
##  1     1     1              305
##  2     1     2              350
##  3     1     3              336
##  4     1     4              339
##  5     1     5              238
##  6     1     6              301
##  7     1     7              342
##  8     1     8              334
##  9     1     9              336
## 10     1    10              344
## # ... with 355 more rows
```

17. Obtain the plot as shown below:

```
data = flights %>% group_by(origin) %>% summarise_at(vars(dep_delay),funs(mean(.,na.rm=TRUE)))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
ggplot(data=data, aes(x=origin, y=dep_delay,fill=origin))+ geom_bar(stat="identity") +
  ylab("Avg_Dep_Delay") +  ggtitle("Average Departure Delay By Airport")
```

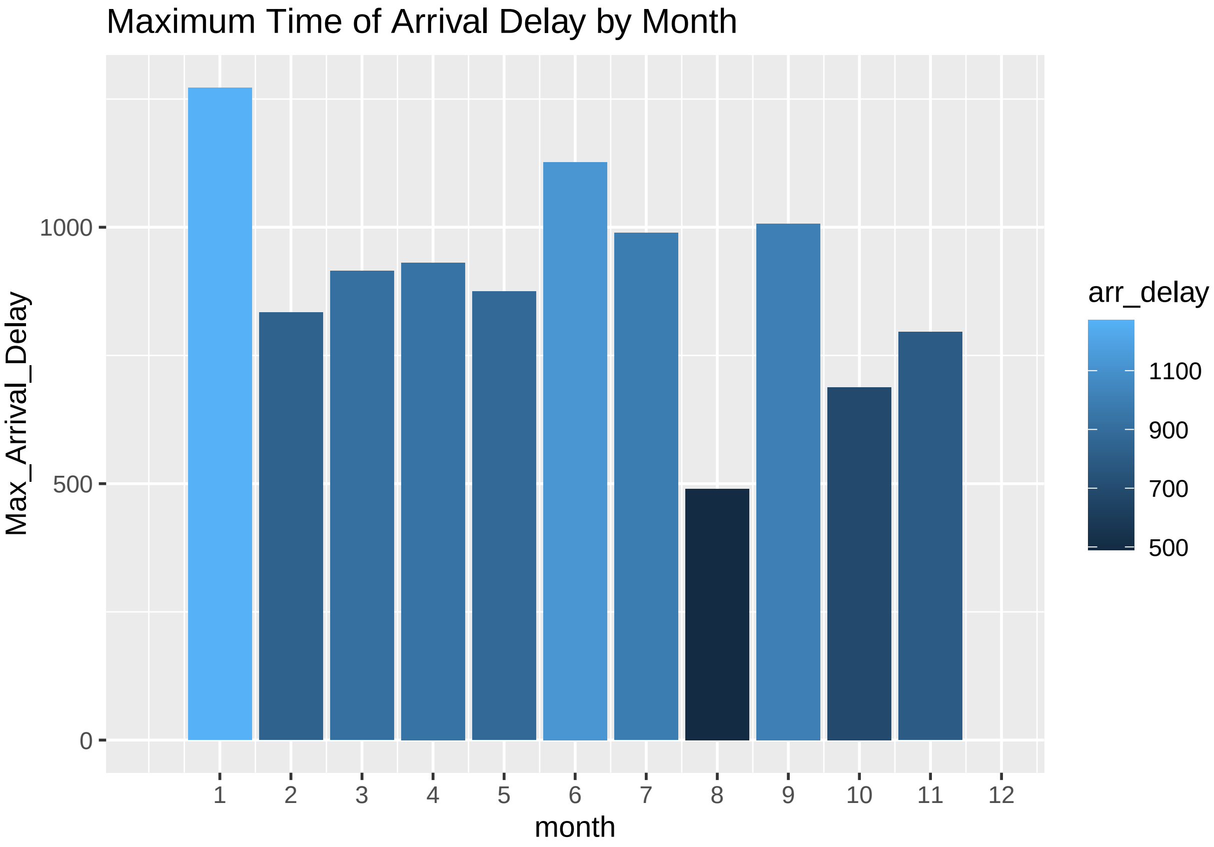


18. Produce the plot of maximum time of arrival delay by month as shown below:

```
df= select(flights,c('month',arr_delay))

df1 = df %>% group_by(month) %>% summarise_at(vars(arr_delay),funs(max(.,na.rm=TRUE)))
```

```r
ggplot(data=df1, aes(x=month, y=arr_delay,fill=arr_delay))+
  geom_bar(stat="identity")+
  scale_x_continuous(breaks=df1$month,limits=c(0, 12)) +
  ylab("Max_Arrival_Delay") +
  ggtitle("Maximum Time of Arrival Delay by Month")
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```


Maximum Time of Arrival Delay by Month
Max_Arrival_Delay
month
arr_delay