Assignment 5 - CT5102 Visualisation with ggplot2 - Flights data

Chin Zhe Jing 22221970

zhejing

2022-10-13

The aim of this assignment is to use ggplot2 to visualise flight data, which is contained in the package nycflights13. Note that you may have to follow up and search the internet to see the instructions for using certain features, for example using lubridate and finding out how to position a legend in ggplot2. Please ensure that your outputs exactly match the graphs. For example, the command scale_x_continuous(n.breaks = 15) can be used to format the x-axis in a number of the plots.

```
library(nycflights13)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(tibble)
```

The data is contained in the tibble flights, and the idea is to create a new tibble d based on this data. Notice that not all the columns are used, and that there are two new columns, MonthName and Hour.

```
data <- tibble(flights)
data$MonthName <- month(data$time_hour, label = TRUE, abbr = TRUE)
data$Hour <-as.factor(data$hour)

d <- subset(data, select=c('month',
                           'MonthName',
                           'Hour',
                           'origin',
                           'day',
                           'dep_delay',
                           'arr_delay',
                           'air_time',
                           'distance',
                           'carrier'))

d
```

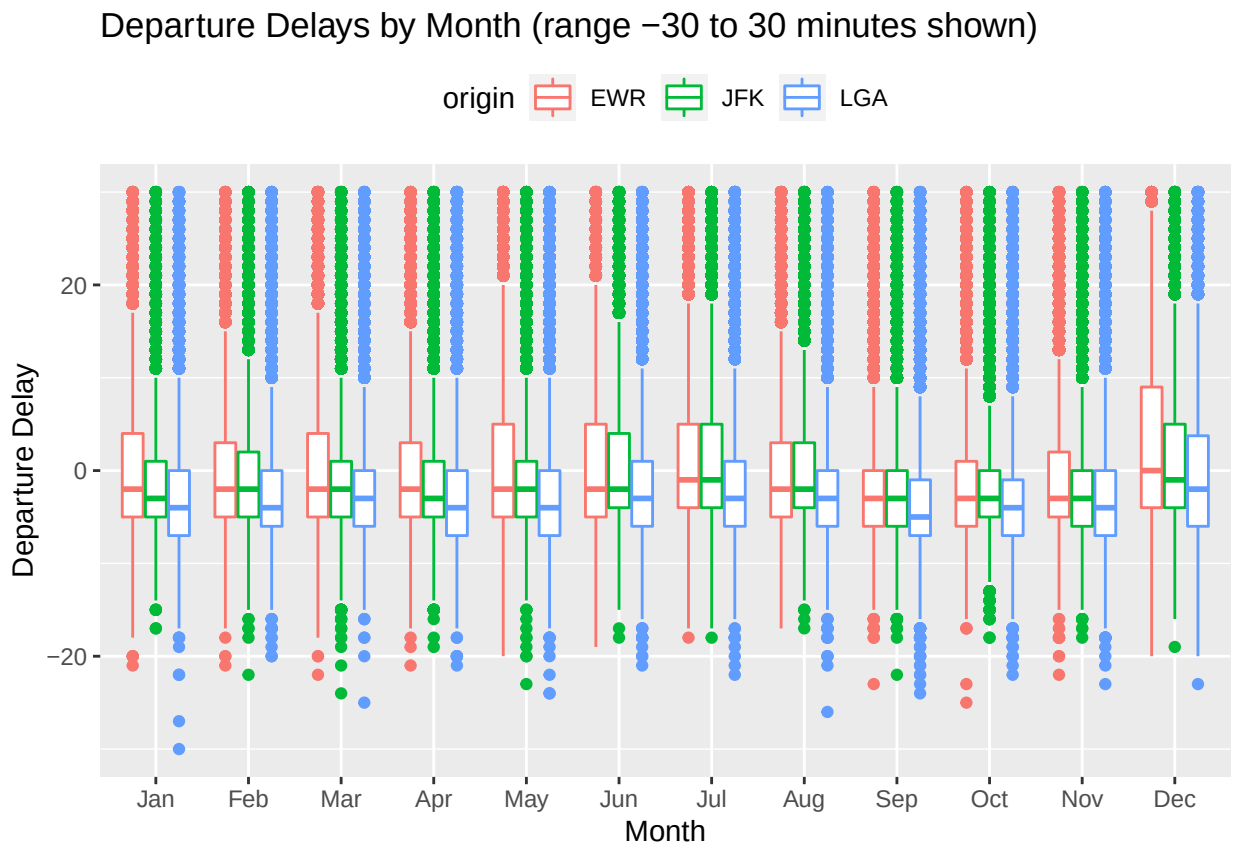
```
## # A tibble: 336,776 x 10
##    month MonthName Hour  origin   day dep_delay arr_de~1 air_t~2 dista~3 carrier
##    <int> <ord>     <fct> <chr>  <int>     <dbl>    <dbl>   <dbl>   <dbl> <chr>
##  1     1 Jan       5     EWR        1         2       11     227    1400 UA
##  2     1 Jan       5     LGA        1         4       20     227    1416 UA
##  3     1 Jan       5     JFK        1         2       33     160    1089 AA
##  4     1 Jan       5     JFK        1        -1      -18     183    1576 B6
##  5     1 Jan       6     LGA        1        -6      -25     116     762 DL
##  6     1 Jan       5     EWR        1        -4       12     150     719 UA
##  7     1 Jan       6     EWR        1        -5       19     158    1065 B6
##  8     1 Jan       6     LGA        1        -3      -14      53     229 EV
##  9     1 Jan       6     JFK        1        -3       -8     140     944 B6
## 10     1 Jan       6     LGA        1        -2        8     138     733 AA
## # ... with 336,766 more rows, and abbreviated variable names 1: arr_delay,
## #   2: air_time, 3: distance
```

```
glimpse(d)
```

```
## Rows: 336,776
## Columns: 10
## $ month     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ MonthName <ord> Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, ~
## $ Hour      <fct> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6, 6, ~
## $ origin    <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA", "JFK~
## $ day       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ dep_delay <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1, 0, ~
## $ arr_delay <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -14, 31~
## $ air_time  <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 158, 3~
## $ distance  <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, 1028,~
## $ carrier   <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "AA", ~
```

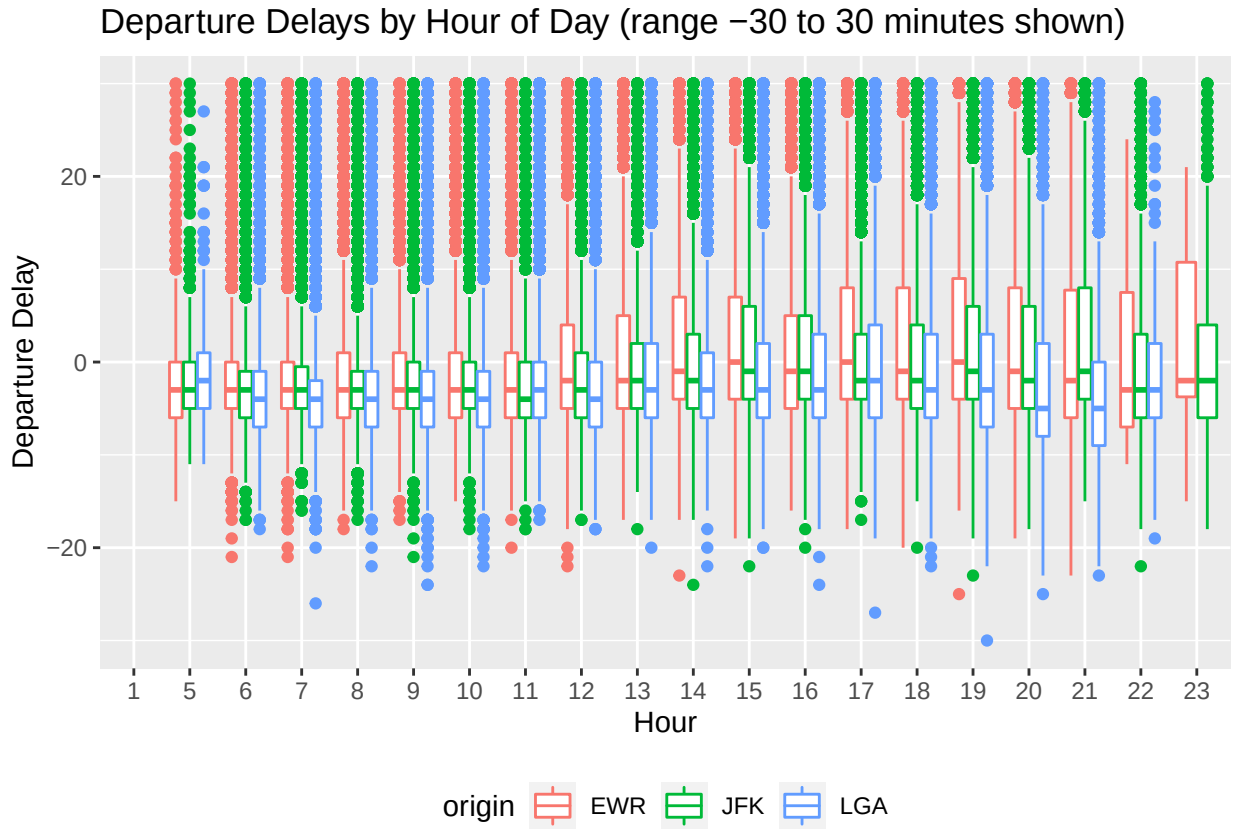
1. Plot a boxplot for the departure delays by month.

```
d_dep_delay_month <- d |>
  subset(origin %in% c('EWR',
                       'JFK',
                       'LGA') &
           dep_delay>=-30 & dep_delay<=30)
ggplot(d_dep_delay_month, aes(x=MonthName, y=dep_delay, colour = origin)) +
  geom_boxplot() +
  xlab("Month") +
  ylab("Departure Delay") +
  ggtitle(("Departure Delays by Month (range -30 to 30 minutes shown)")) +
  theme(legend.position="top")
```



2. Plot a boxplot for the departure delays by hour of the day xlim(levels(data$Hour)) force show all data points

```
ggplot(subset(d, origin %in% c('EWR',
                               'JFK',
                               'LGA') &
                dep_delay>=-30 & dep_delay<=30),
       aes(x=Hour, y=dep_delay, colour = origin)) +
  geom_boxplot() +
  xlab("Hour") +
  xlim(levels(data$Hour)) +
  ylab("Departure Delay") +
  ggtitle(("Departure Delays by Hour of Day (range -30 to 30 minutes shown)")) +
  theme(legend.position="bottom")
```

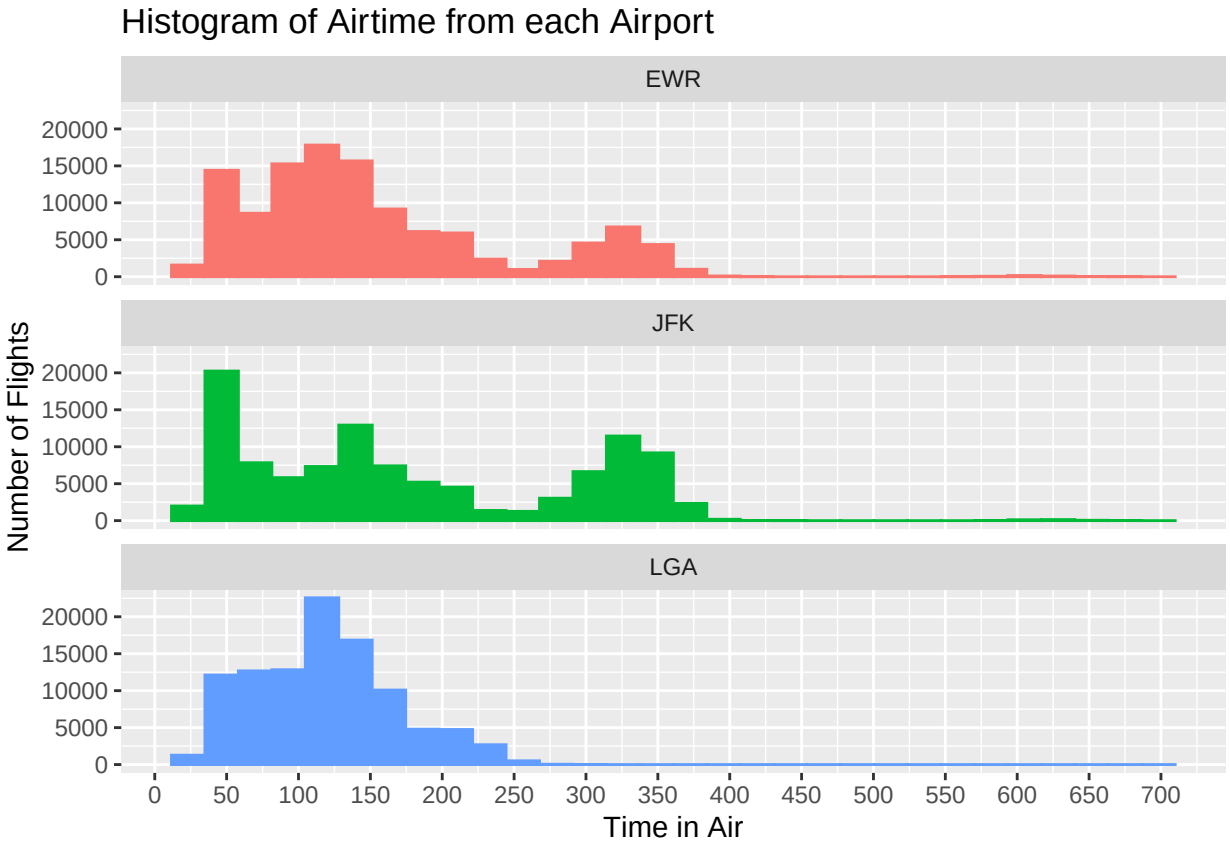


3. Plot a faceted histogram of the air time of flights by origin. Note that three rows should be used, and there is no legend.

```
ggplot(subset(d, origin %in% c('EWR',
                               'JFK',
                               'LGA')),
       aes(x=air_time, colour = origin, fill = origin)) +
  geom_histogram(show.legend = FALSE) +
  xlab("Time in Air") +
  scale_x_continuous(breaks = seq(0, 700, by = 50)) +
  ylab("Number of Flights") +
  ggtitle(("Histogram of Airtime from each Airport")) +
  facet_wrap(~origin, ncol=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

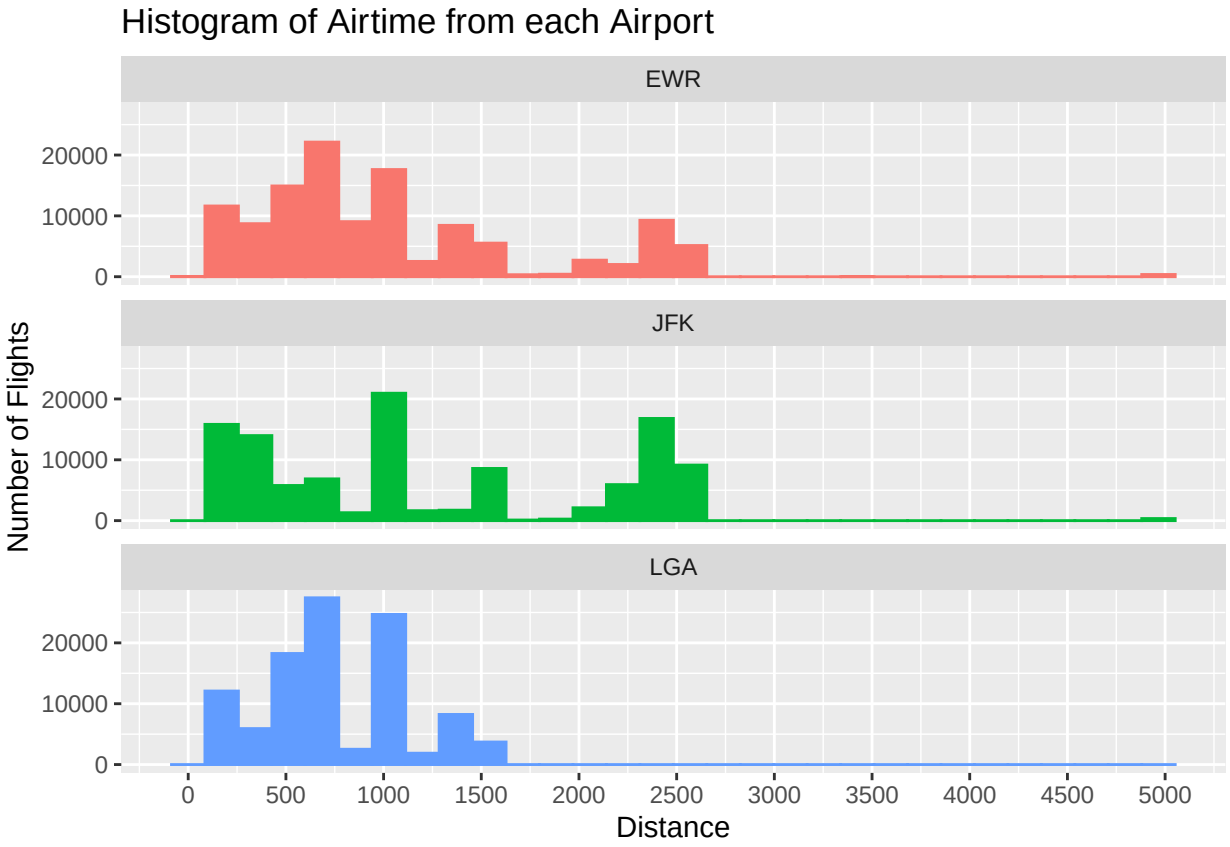
```
## Warning: Removed 9430 rows containing non-finite values (stat_bin).
```



4. Plot a faceted histogram of the flight distance by origin. Note that three rows should be used, and there is no legend.

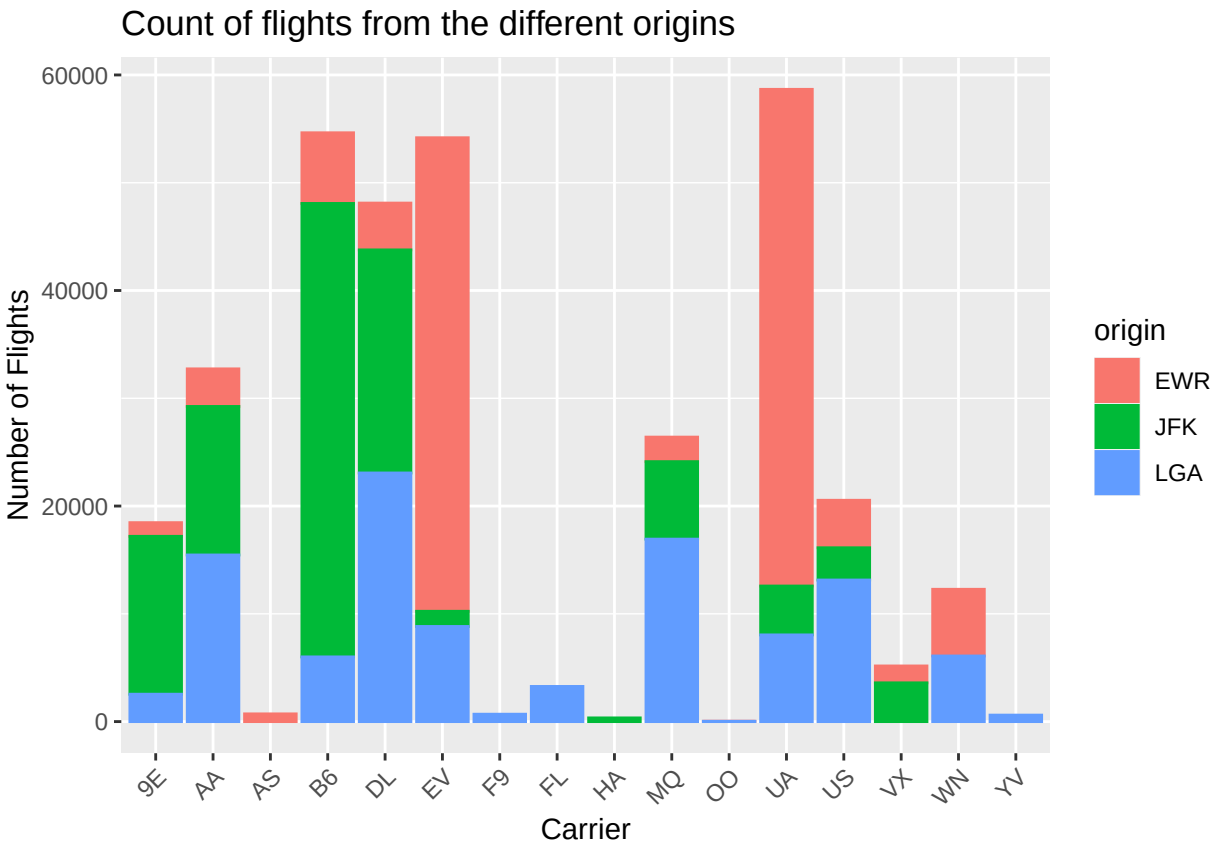
```
ggplot(subset(d, origin %in% c('EWR',
                               'JFK',
                               'LGA')),
       aes(x=distance, colour = origin, fill = origin)) +
  geom_histogram(show.legend = FALSE) +
  xlab("Distance") +
  scale_x_continuous(breaks = seq(0, 5000, by = 500)) +
  ylab("Number of Flights") +
  ggtitle(("Histogram of Airtime from each Airport")) +
  facet_wrap(~origin, ncol=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



5. Plot a bar chart showing the number of flights per carrier

```
ggplot(subset(d, origin %in% c('EWR',
                               'JFK',
                               'LGA')),
       aes(x=carrier, colour = origin, fill = origin, position = "fill")) +
  geom_bar() +
  xlab("Carrier") +
  theme(axis.text.x=element_text(angle=45,hjust=1)) +
  ylab("Number of Flights") +
  ggtitle(("Count of flights from the different origins"))
```



6. With a seed value of 100, select a random sample of 3000 from the tibble, and store this in the variable d1.

```
set.seed(100)
d1_data <- tibble(flights)
d1 <- d1_data[sample(nrow(d1_data), 3000), ]
head(d1)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep~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    11    30      756         759      -3     947    1008     -21 EV
## 2  2013     9    25      757         805      -8    1053    1101      -8 DL
## 3  2013     3    29     1112        1115      -3    1352    1416     -24 DL
## 4  2013    12     8      752         752       0    1040    1046      -6 UA
## 5  2013     5    16     1154        1200      -6    1257    1310     -13 US
## 6  2013    12     6       NA         600      NA      NA     915      NA AA
## # ... with 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
```

```
summary(d1)
```

```
##       year          month             day           dep_time    sched_dep_time
```

```
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   6   Min.   : 500
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 905   1st Qu.: 900
##  Median :2013   Median : 7.000   Median :16.00   Median :1350   Median :1345
##  Mean   :2013   Mean   : 6.629   Mean   :15.79   Mean   :1335   Mean   :1326
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1730   3rd Qu.:1713
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2357   Max.   :2359
##                                                  NA's   :69
##    dep_delay          arr_time    sched_arr_time   arr_delay
##  Min.   :-18.00   Min.   :   1   Min.   :   1   Min.   :-68.000
##  1st Qu.: -5.00   1st Qu.:1109   1st Qu.:1120   1st Qu.:-17.000
##  Median : -2.00   Median :1525   Median :1541   Median : -6.000
##  Mean   : 11.83   Mean   :1497   Mean   :1523   Mean   :  5.853
##  3rd Qu.: 10.00   3rd Qu.:1930   3rd Qu.:1925   3rd Qu.: 13.000
##  Max.   :336.00   Max.   :2400   Max.   :2359   Max.   :360.000
##  NA's   :69       NA's   :74                    NA's   :82
##    carrier              flight         tailnum             origin
##  Length:3000        Min.   :   1.0   Length:3000        Length:3000
##  Class :character   1st Qu.: 634.8   Class :character   Class :character
##  Mode  :character   Median :1552.0   Mode  :character   Mode  :character
##                     Mean   :2019.5
##                     3rd Qu.:3525.0
##                     Max.   :6177.0
##
##      dest              air_time        distance         hour        minute
##  Length:3000        Min.   : 23.0   Min.   :  94   Min.   : 5   Min.   : 0.00
##  Class :character   1st Qu.: 81.0   1st Qu.: 502   1st Qu.: 9   1st Qu.: 7.00
##  Mode  :character   Median :125.0   Median : 816   Median :13   Median :29.00
##                     Mean   :147.7   Mean   :1019   Mean   :13   Mean   :25.99
##                     3rd Qu.:190.0   3rd Qu.:1389   3rd Qu.:17   3rd Qu.:44.00
##                     Max.   :686.0   Max.   :4983   Max.   :23   Max.   :59.00
##                     NA's   :82
##    time_hour
##  Min.   :2013-01-01 07:00:00.0
##  1st Qu.:2013-04-07 16:45:00.0
##  Median :2013-07-07 14:00:00.0
##  Mean   :2013-07-05 17:49:25.2
##  3rd Qu.:2013-10-04 12:00:00.0
##  Max.   :2013-12-31 15:00:00.0
##
```

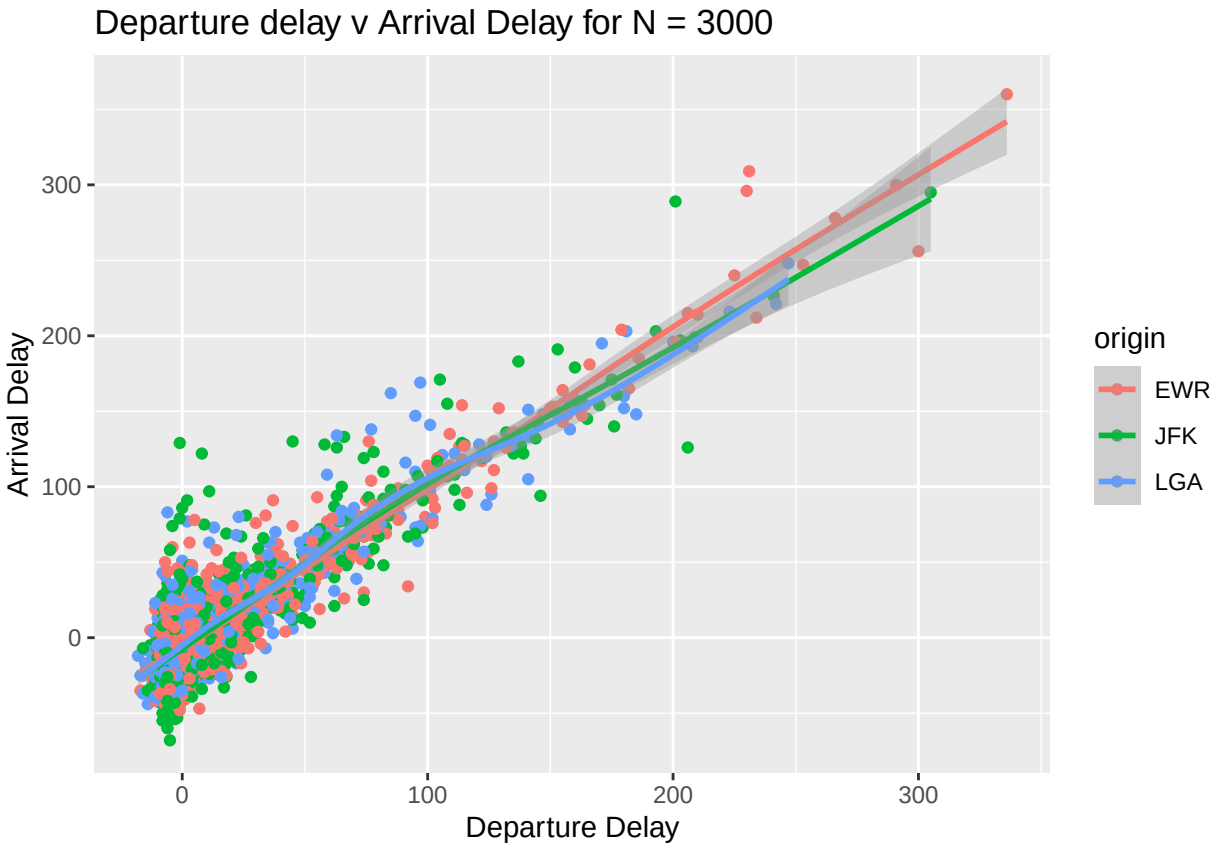
7. Plot the departure delay v arrival delay Use geom_point

```
ggplot(subset(d1, origin %in% c('EWR',
                                'JFK',
                                'LGA')),
       aes(x=dep_delay,
           y=arr_delay,
           colour = origin,
           position = "fill")) +
  geom_point() +
  geom_smooth() +
  xlab("Departure Delay") +
  ylab("Arrival Delay") +
  ggtitle(("Departure delay v Arrival Delay for N = 3000"))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 82 rows containing non-finite values (stat_smooth).

## Warning: Removed 82 rows containing missing values (geom_point).
```



8. Plot the distance v Air Time Use method = lm

```
ggplot(subset(d1, origin %in% c('EWR',
                                'JFK',
                                'LGA')),
       aes(x=distance,
           y=air_time,
           colour = origin,
           position = "fill")) +
  geom_point() +
  geom_smooth(method = lm) +
  xlab("Distance") +
  ylab("Air Time") +
  ggtitle(("Distance v Air Time for N = 3000"))
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 82 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 82 rows containing missing values (geom_point).
```