HW2

Tianze Wang, Kexin Wang

9/24/2019

#0Initialization Run the next line only once

```
#install.packages("nycflights13")
```

Then always run next block

```
library(nycflights13)
nyc <- nycflights13::flights
```

Prob1

```
nyc$air_gain = nyc$dep_delay - nyc$arr_delay
```

1.a

```
mean(nyc$air_gain, na.rm = TRUE)
```

```
## [1] 5.659779
```

```
median(nyc$air_gain, na.rm = TRUE)
```

```
## [1] 7
```

Since both mean and median of the air_gain value is bigger than 0, we could say airlines actually gain time on average.

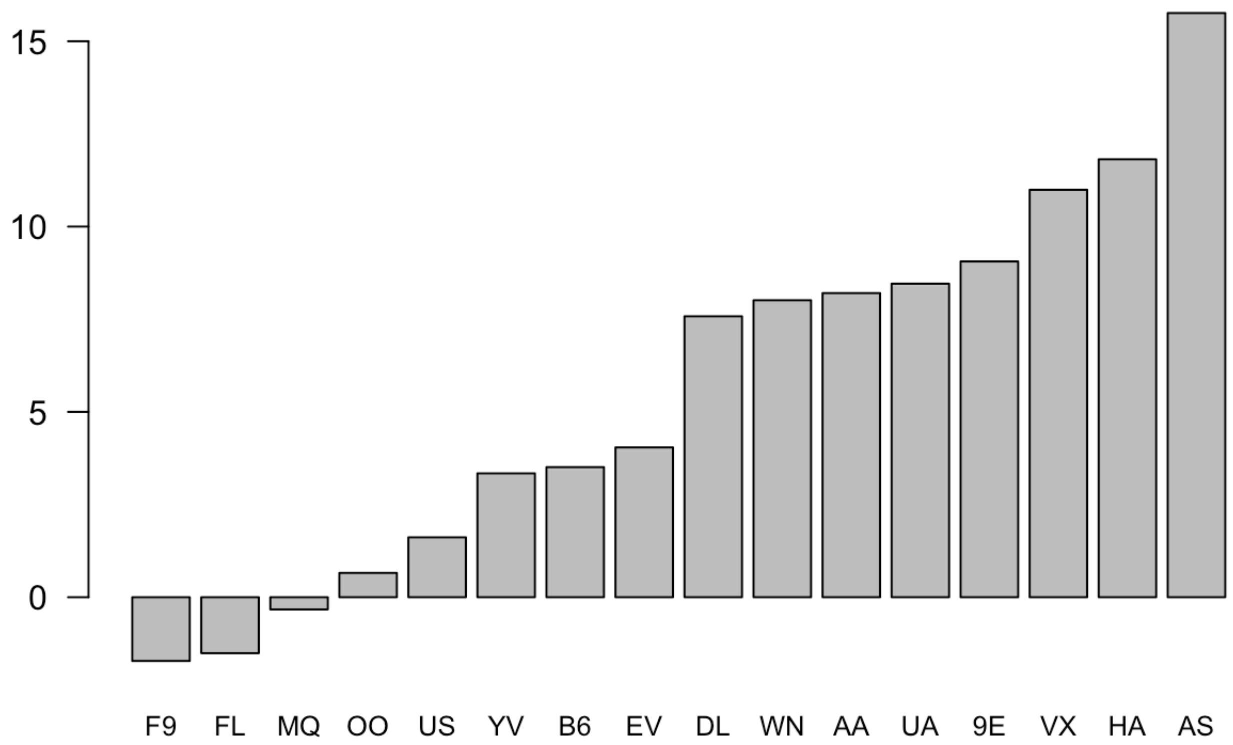
1.b

```
tapply(nyc$air_gain, nyc$carrier , mean, na.rm = TRUE)
```

```
##          9E         AA         AS         B6         DL         EV 
##  9.0599052  8.2048393 15.7616361  3.5095746  7.5796089  4.0424982 
##          F9         FL         HA         MQ         OO         UA 
## -1.7195301 -1.5099213 11.8157895 -0.3293526  0.6551724  8.4588972 
##          US         VX         WN         YV 
##  1.6150976 10.9921814  8.0125374  3.3419118
```

1.c

```
barplot(sort(tapply(nyc$air_gain, nyc$carrier , mean, na.rm = TRUE)), las=1, cex.name
s=.8)
```



2

```
good_airport_monthly <- function(x) {
  nyc_temp = nyc[nyc$month == x,]
  grouped_result = tapply(nyc_temp$arr_delay, nyc_temp$origin, mean, na.rm = TRUE)
  return(names(sort(grouped_result)[1]))
}
good_airport_monthly(6)
```

```
## [1] "LGA"
```

3

Here I choose another method instead of the if else statement.

```
morning <- nyc$dep_time >= 400 & nyc$dep_time < 1200
noon <- nyc$dep_time >= 1200 & nyc$dep_time < 1600
evening <- nyc$dep_time >= 1600 & nyc$dep_time < 2000
night <- nyc$dep_time >= 2000 | nyc$dep_time < 400
nyc$time_region = NA
nyc$time_region[morning] = "morning"
nyc$time_region[noon] = "noon"
nyc$time_region[evening] = "evening"
nyc$time_region[night] = "night"
nyc_atl = nyc[nyc$dest == "ATL",]
names(sort(tapply(nyc_atl$arr_delay, nyc_atl$time_region, mean, na.rm = TRUE))[1])
```

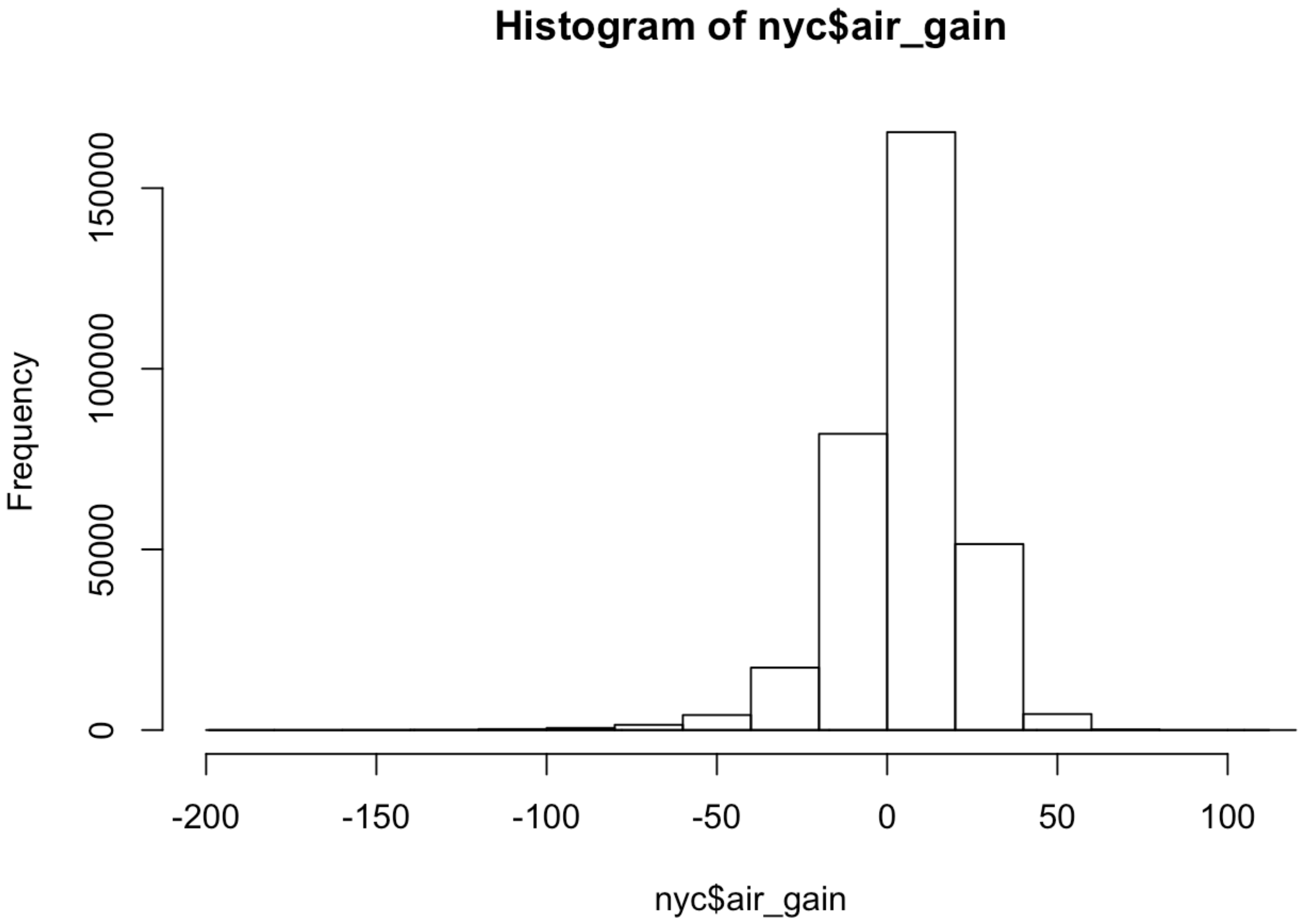
```
## [1] "morning"
```

```
nyc_atl = nyc_atl[complete.cases(nyc_atl),]
nyc$origin = as.factor(nyc$origin)
nyc$time_region = as.factor(nyc$time_region)
```

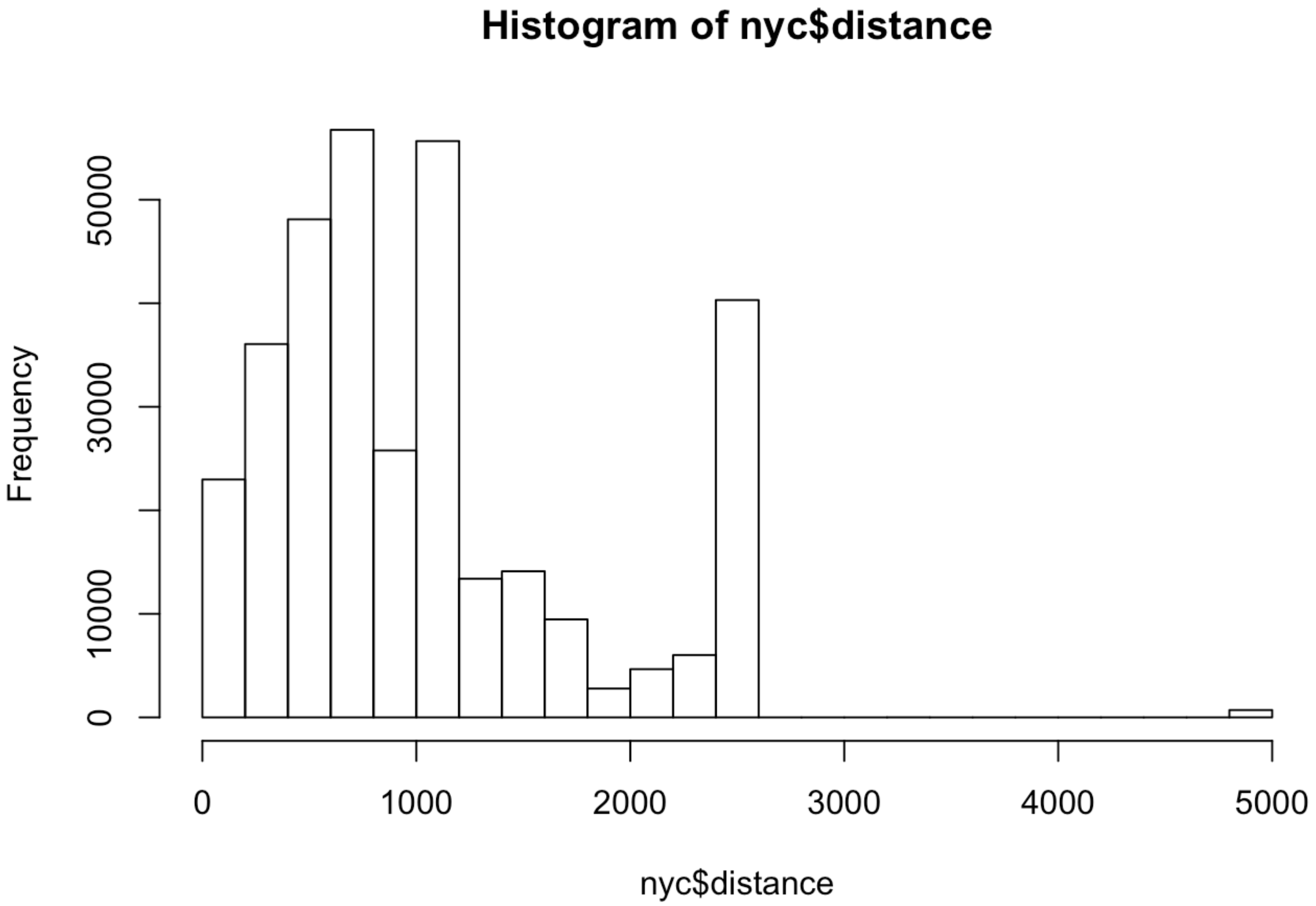
4

First we observe the distribution within these two columns

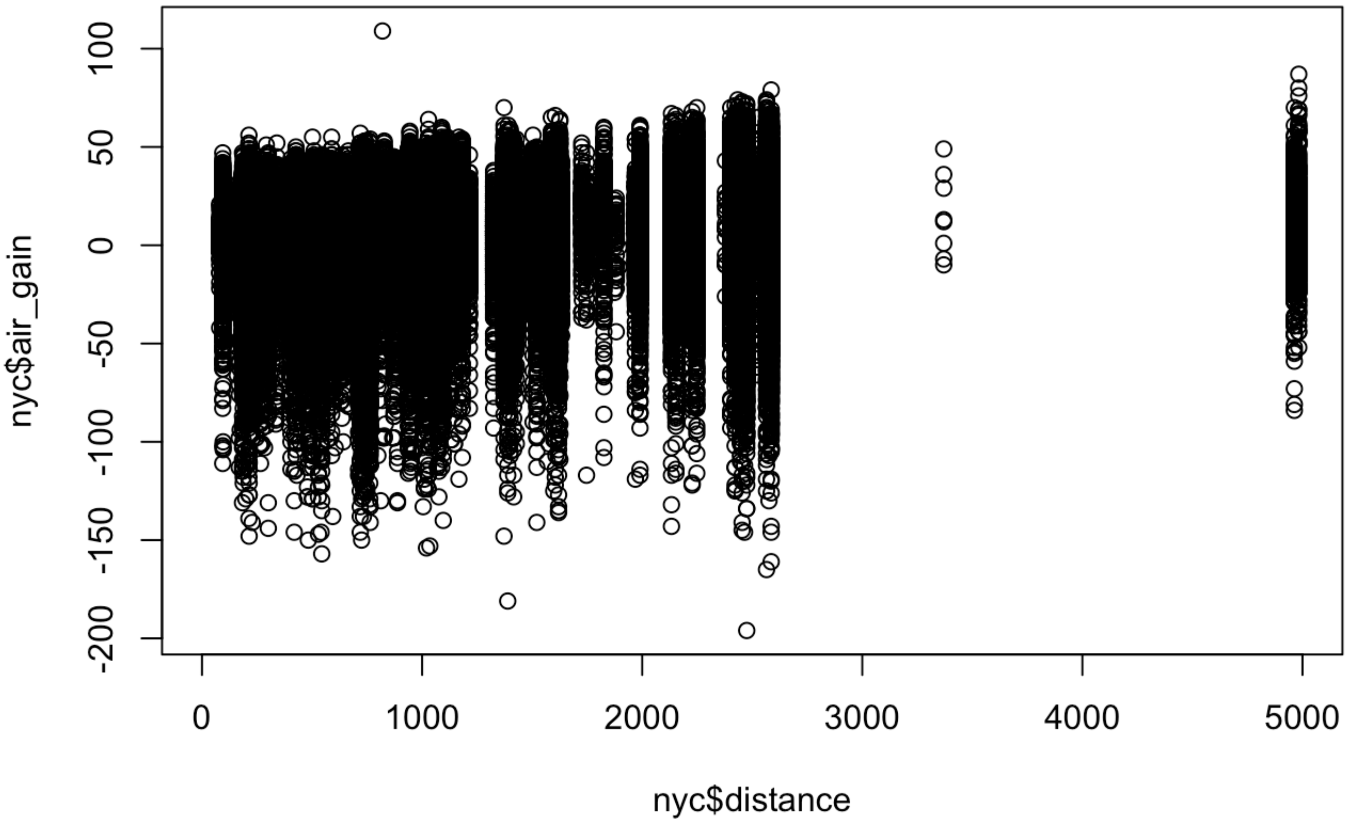
```
hist(nyc$air_gain)
lines(density(nyc$air_gain, na.rm = TRUE))
```



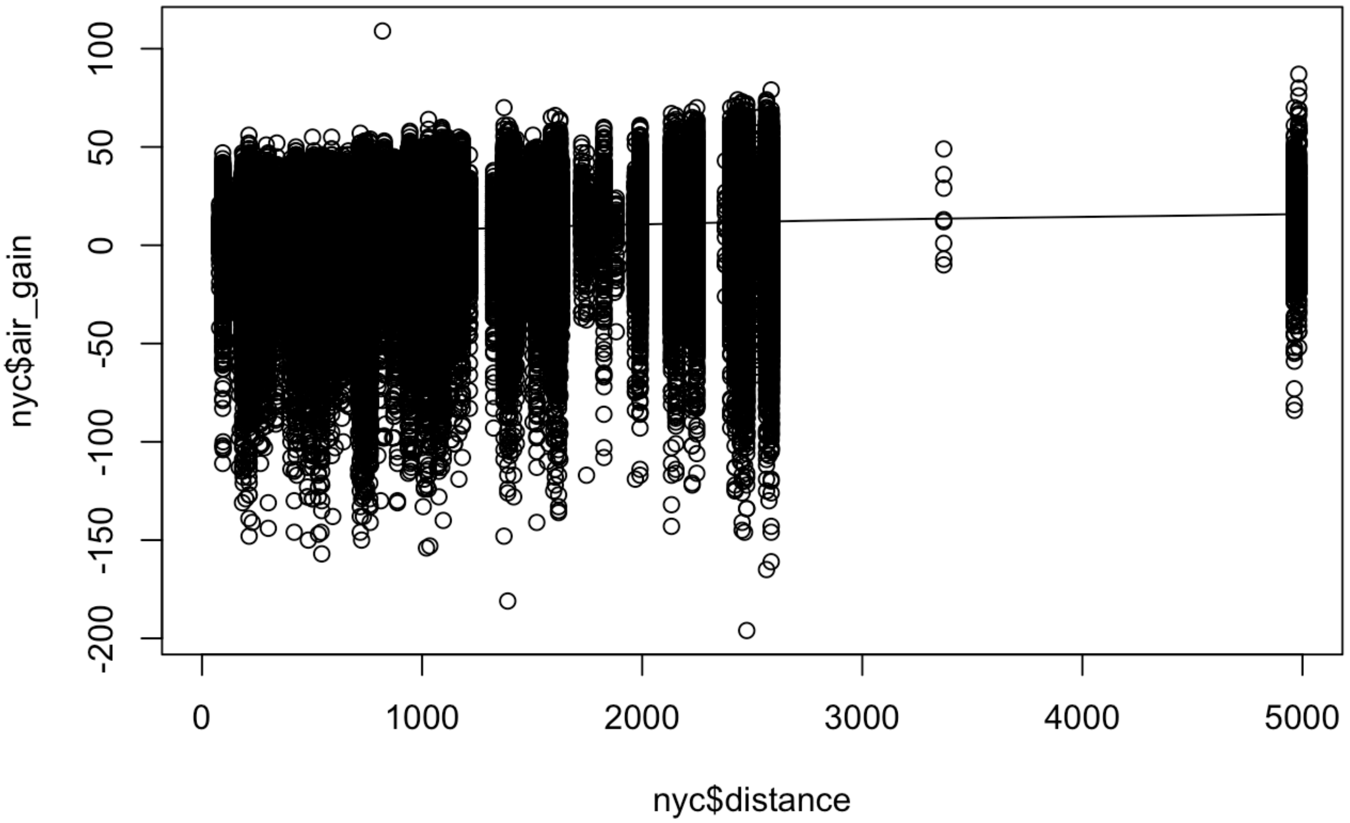
```
hist(nyc$distance)
```



```
plot(nyc$distance, nyc$air_gain)
```



```
scatter.smooth(nyc$distance, nyc$air_gain)
```



```
cor(nyc$distance, nyc$air_gain,use="complete.obs")
```

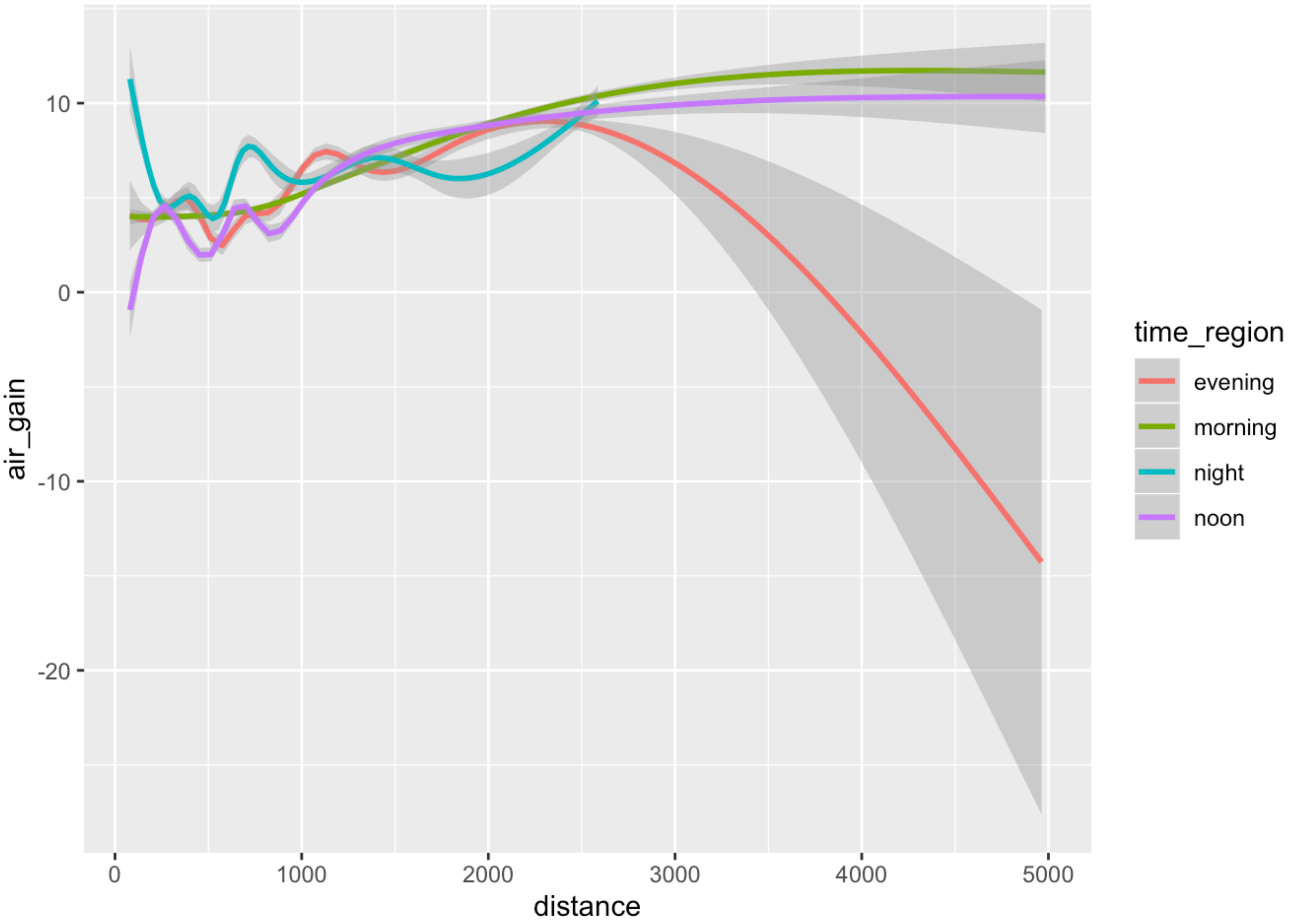
```
## [1] 0.1048957
```

We study the relation with different time zones

```
library(ggplot2)
baseplot <- ggplot(data = nyc, aes(x=distance, y= air_gain, colour = time_region))
baseplot + geom_smooth()
```

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

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

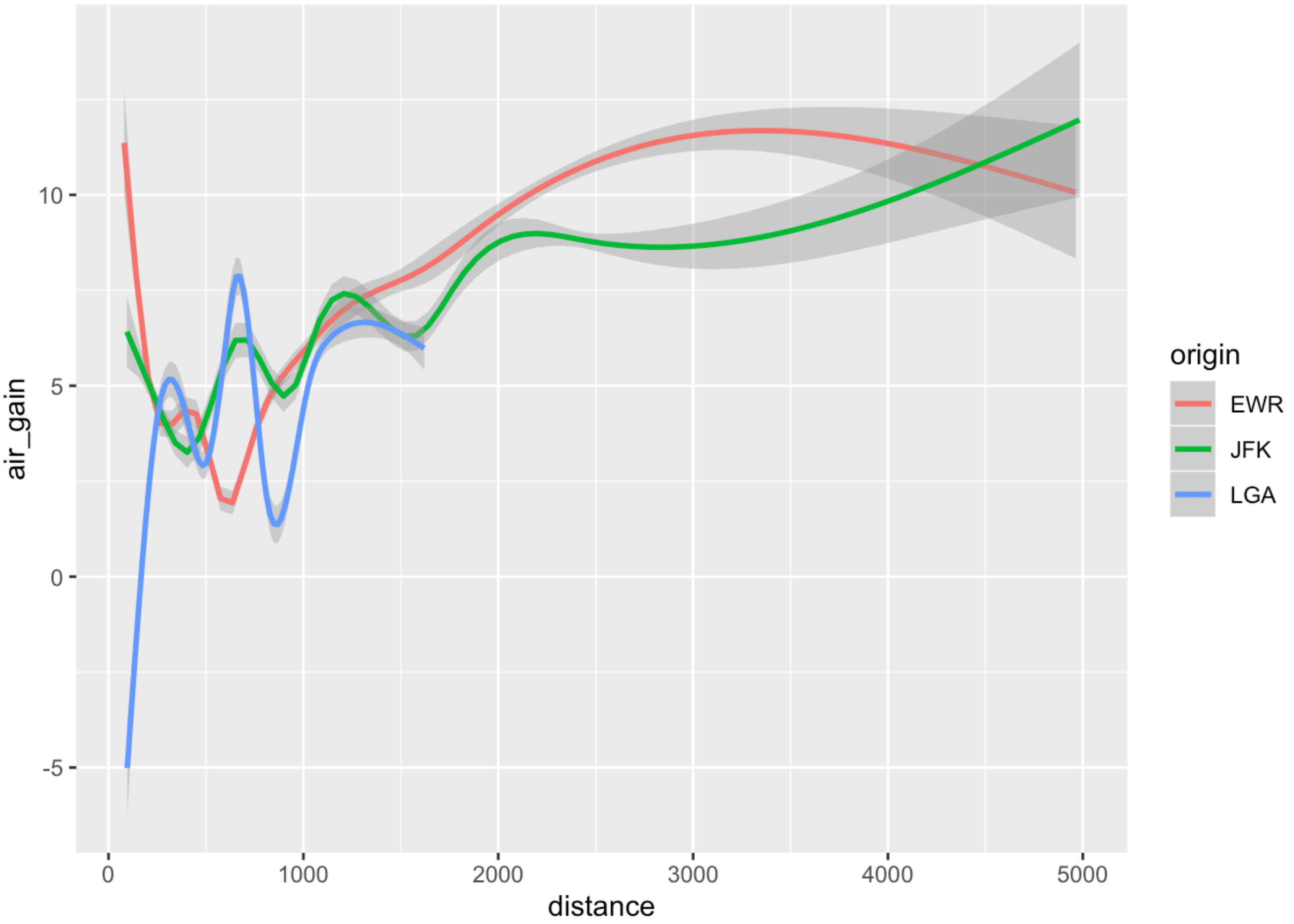


Next we study the relation within different destination

```
library(ggplot2)
baseplot <- ggplot(data = nyc, aes(x=distance, y= air_gain, colour = origin))
baseplot + geom_smooth()
```

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

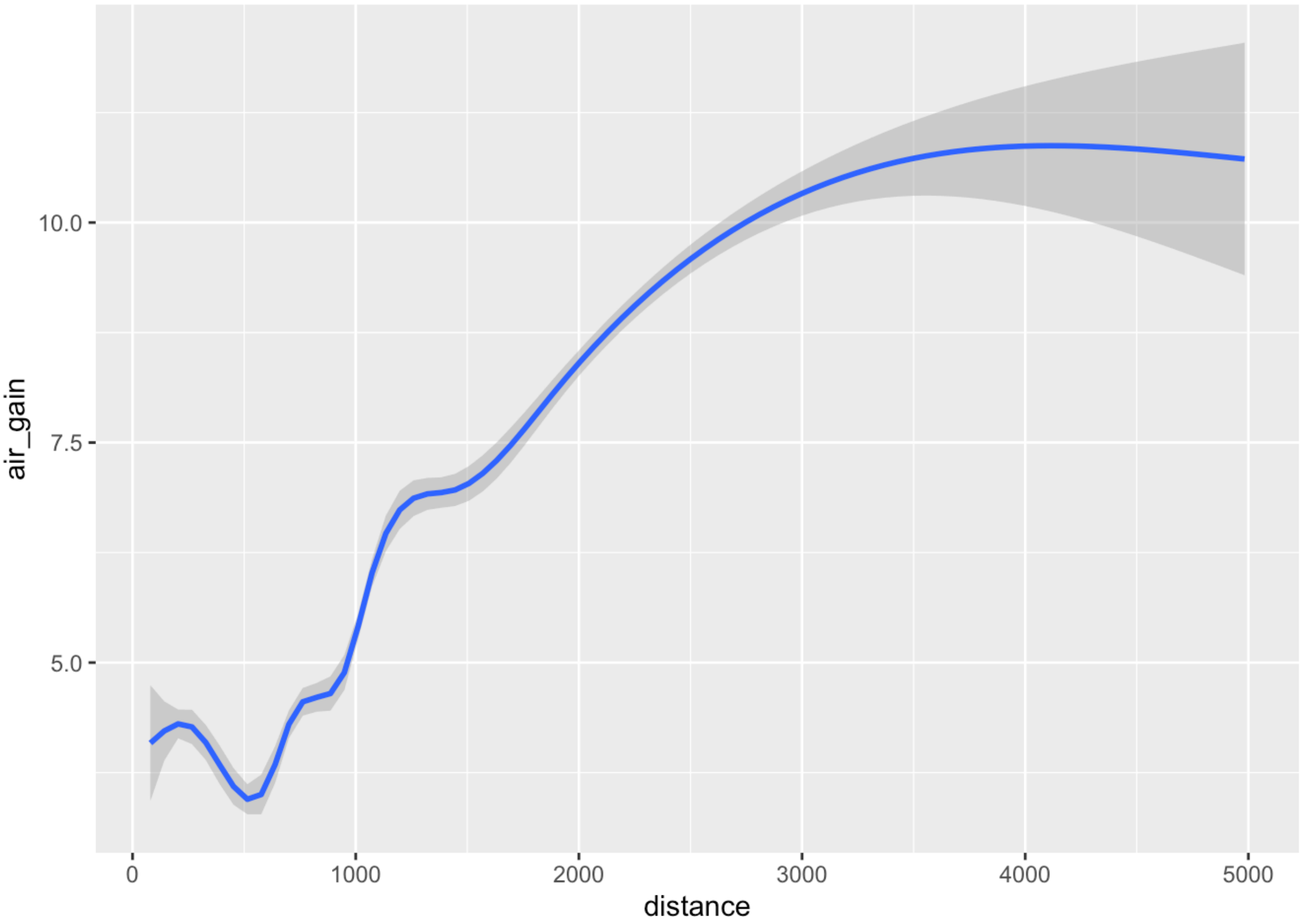
```
## Warning: Removed 9430 rows containing non-finite values (stat_smooth).
```



```
library(ggplot2)
baseplot <- ggplot(data = nyc, aes(x=distance, y= air_gain))
baseplot + geom_smooth()
```

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

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



#

4. Conclusion

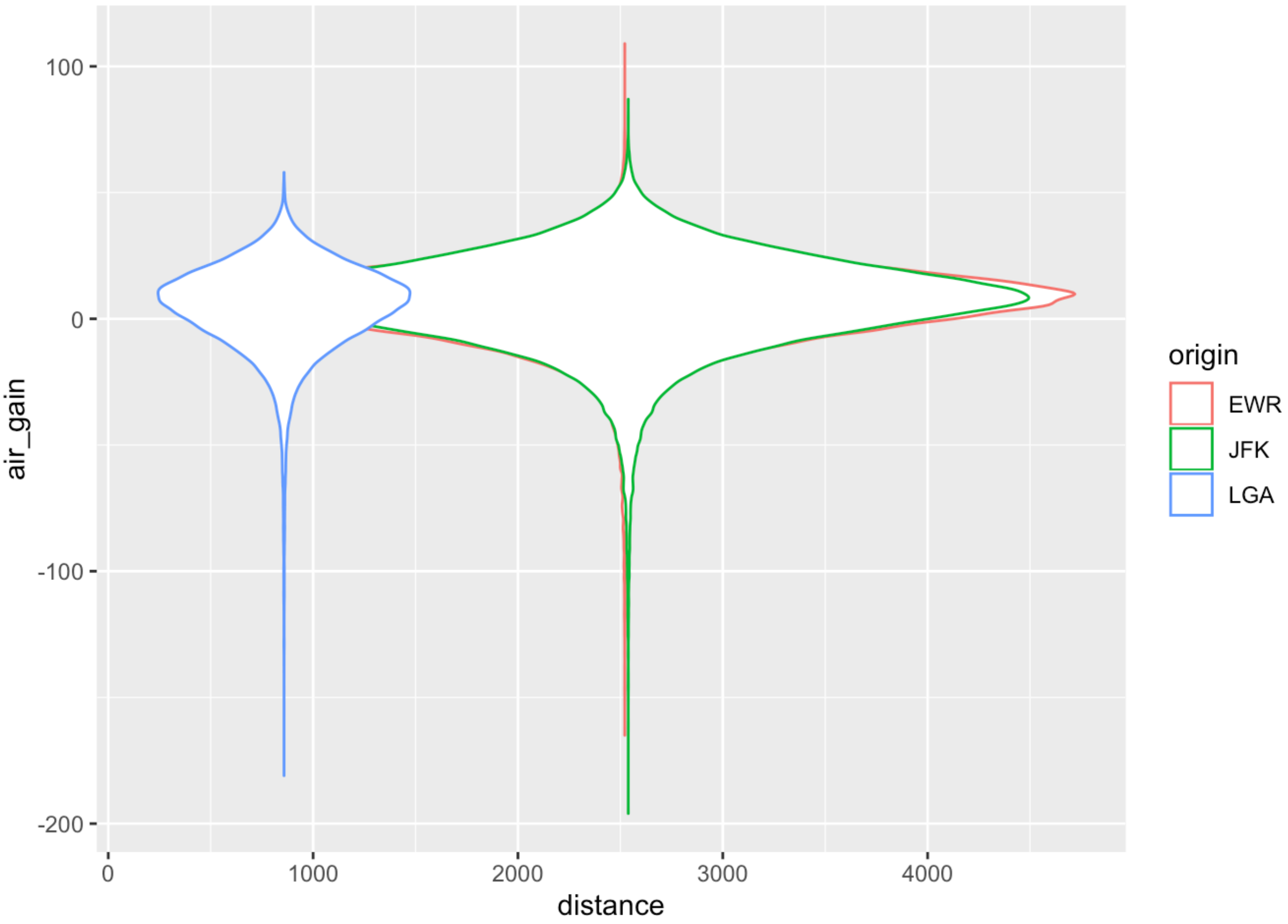
```
From the above figures, we can denote following observations.
1. Generally, the longer the distance, the bigger airgain is. However, airgain usuall
y will not exceed 15 minutes.
2. The airgain for flights travelled with distance less than 2000 have an airgain wit
h bigger sample deviation.
3. Night flights tend to have negative airgain at long distance, which indicates the
flight spend more time on air than estimated. This may due to fatigue of pilots, or c
onstraints on night flights.
```

5

```
library(ggplot2)
baseplot <- ggplot(data = nyc, aes(x=distance, y= air_gain, colour = origin))
baseplot + geom_violin()
```

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

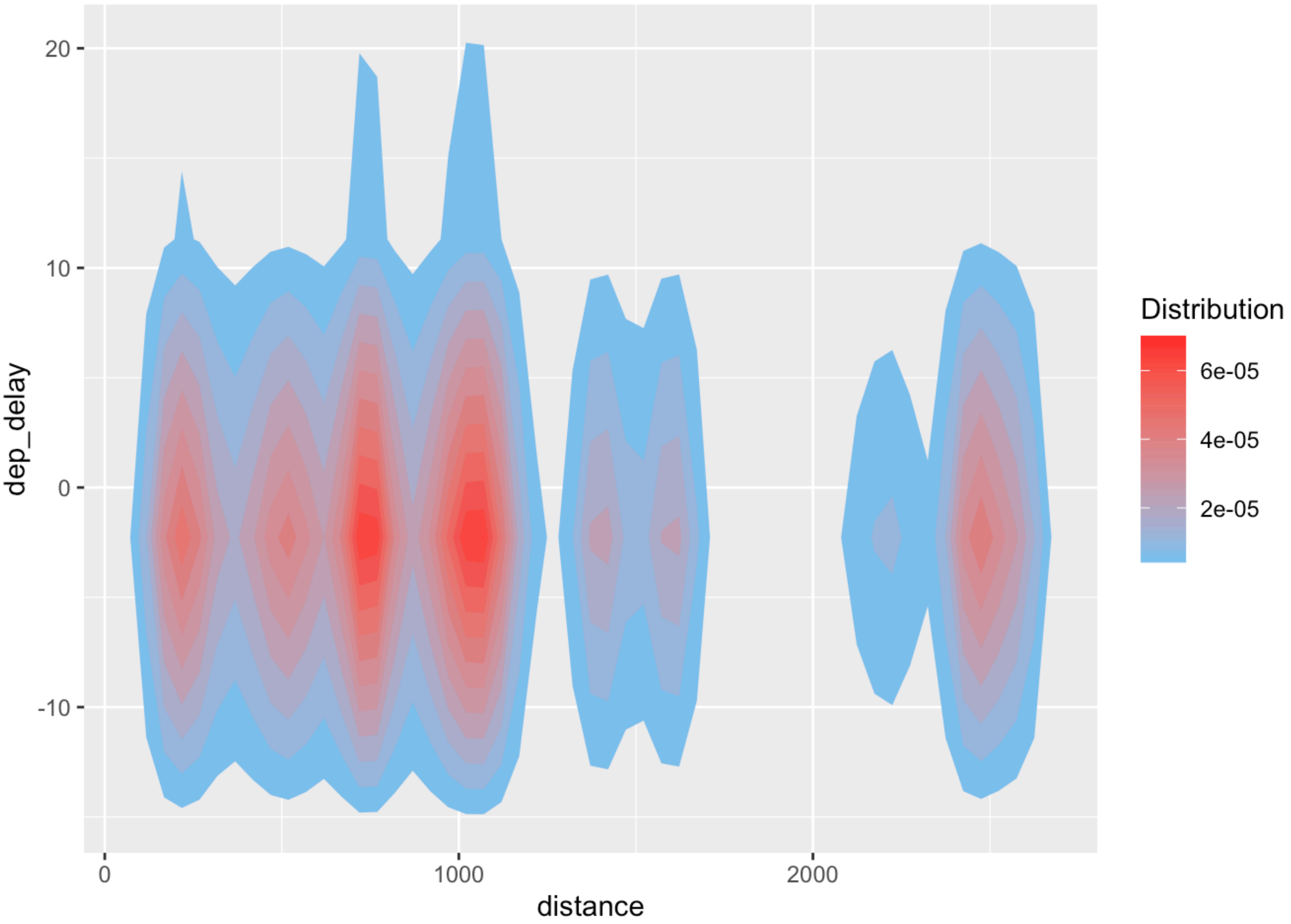
```
## Warning: position_dodge requires non-overlapping x intervals
```



From the violin plot above, we can understand tha EWR and JFK mainly have long distance travels, while LGA mostly has short distance travels.

```
library(ggplot2)
baseplot <- ggplot(data = nyc, aes(x=distance, y= dep_delay, colour = air_gain))
baseplot + stat_density2d(aes(fill=..level..), geom="polygon",bins=12) +   scale_fill
_gradient(low="skyblue2", high="firebrick1", name="Distribution")
```

```
## Warning: Removed 8255 rows containing non-finite values (stat_density2d).
```



This graph is a density graph showing that, considering both distance and dep delay, which combination will laed to most airgain density. Most airgain density is labeled as red, while low density is labeled as blue.