Final Project Data Exploration

2023-05-31

R Markdown

```
library(magrittr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

library(ggplot2)

SuperClean<-read.csv('SuperClean2019.csv')

# Check if there are any NA values in the entire dataframe 'FreshStart_clean'
has_na <- any(is.na(SuperClean))

# Print the result
if (has_na) {
  print("There are NA values in the dataframe.")
} else {
  print("There are no NA values in the dataframe.")
}

## [1] "There are no NA values in the dataframe."

range(SuperClean$ARR_DELAY)

## [1]  -94 2649

fivenum(SuperClean$ARR_DELAY)

## [1]  -94  -16   -6    9 2649

quantile(SuperClean$ARR_DELAY,probs = c(0.01,0.05,0.1,0.25,.5,.75,.90,.95,.99
))

##  1%  5% 10% 25% 50% 75% 90% 95% 99%
## -39 -29 -24 -16  -6   9  39  76 202

mean(SuperClean$ARR_DELAY >= 60)
```

```
## [1] 0.06575714
```

```
MajorAirports<-c("ATL","DFW","DEN","ORD","LAX","CLT","MCO","SEA","MIA","JFK",
"PHX","IAH","SFO","EWR","BOS",
                 "DTW","SLC","PHL","BWI","FLL","MSP","TPA","SAN","LGA","MDW",
"BNA","IAD","DCA","AUS","DAL"
                 ,"HNL","PDX","HOU","RSW")
SuperCleanMajorAirports<-subset(SuperClean,DEST %in% MajorAirports)
```

```
Q3 <- function(x) { quantile(x,probs=.75) }
SuperClean %>% group_by(SuperClean$DEST) %>%
  summarize(n=n(),med_d = median(ARR_DELAY),Q3_d = Q3(ARR_DELAY), max_d = max
(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>% head(10)
```

```
## # A tibble: 10 × 5
##    `SuperClean$DEST`      n med_d  Q3_d max_d
##    <chr>              <int> <dbl> <dbl> <int>
##  1 PIB                  106   0.5  62     961
##  2 MEI                  160  -1    43.8   477
##  3 EAU                   88  -6.5  43.2   720
##  4 ALO                   79   3    35.5   126
##  5 MKG                   88   3    35.5   134
##  6 ASE                 1443   0    35     967
##  7 LWB                   81  -6    35     319
##  8 MMH                  120   4    34.8   497
##  9 HGR                   18   2.5  30.8   143
## 10 CMI                  326   2    27.8   900
```

```
SuperCleanMajorAirports %>% group_by(SuperCleanMajorAirports$DEST) %>%
  summarize(n=n(),med_d = median(ARR_DELAY),Q3_d = Q3(ARR_DELAY), max_d = max
(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>% head(36)
```

```
## # A tibble: 34 × 5
##    `SuperCleanMajorAirports$DEST`      n med_d  Q3_d max_d
##    <chr>                           <int> <dbl> <dbl> <int>
##  1 LGA                             23631    -6  22    2649
##  2 EWR                             17906    -5  21.8  1594
##  3 SFO                             22677    -4  21    1447
##  4 ORD                             42118    -3  20    2050
##  5 BOS                             18906    -7  13    1113
##  6 DFW                             40236    -4  11    1652
##  7 LAX                             31089    -5  11    1442
##  8 FLL                             15030    -6  10    1288
##  9 SAN                             13047    -4  10     680
## 10 DCA                             20030    -7   9    1313
## # ℹ 24 more rows
```

```
SuperClean %>% group_by(SuperClean$OP_UNIQUE_CARRIER) %>%
  summarize(n=n(),med_d = median(ARR_DELAY),Q3_d = Q3(ARR_DELAY), max_d = max
```

```
(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>% head(17)
```

```
## # A tibble: 17 × 5
##    `SuperClean$OP_UNIQUE_CARRIER`      n med_d  Q3_d max_d
##    <chr>                           <int> <dbl> <dbl> <int>
##  1 B6                              40878    -6    16  1313
##  2 MQ                              42013    -3    16  2649
##  3 EV                              21370    -5    14  1594
##  4 G4                              12987    -3    14  1478
##  5 F9                              15974    -6    13  1020
##  6 OO                             110349    -6    13  1498
##  7 AS                              34996    -5    11   816
##  8 YX                              43594    -7    10  1353
##  9 AA                             132935    -5     9  1638
## 10 YV                              31473    -4     9  2206
## 11 UA                              81619    -7     8  1398
## 12 HA                              12017    -2     7  1507
## 13 OH                              39798    -6     7  1145
## 14 WN                             184748    -6     7   566
## 15 9E                              34634   -11     6  1464
## 16 NK                              26099    -7     5  1429
## 17 DL                             128948    -9     3  1241
```

```
SuperClean %>% group_by(SuperClean$ORIGIN,SuperClean$OP_UNIQUE_CARRIER) %>%
  summarize(n=n(),med_d = median(ARR_DELAY),Q3_d = Q3(ARR_DELAY), max_d = max
(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>% head(10)
```

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

```
## # A tibble: 10 × 6
## # Groups:   SuperClean$ORIGIN [9]
##    `SuperClean$ORIGIN` `SuperClean$OP_UNIQUE_CARRIER`     n med_d  Q3_d ma
x_d
##    <chr>               <chr>                          <int> <dbl> <dbl> <i
nt>
##  1 FAR                 EV                                 2 214.   315.
416
##  2 CWA                 OO                                 1 161    161
161
##  3 BHM                 UA                                 3 131    148
165
##  4 FAR                 YX                                 9  18    140
888
##  5 RAP                 9E                                 1 132    132
132
##  6 CID                 YX                                 5  83    110
118
```

```
## 7 ALB                 YX                          6  39.5  98.8
181
## 8 BTV                 EV                        117  20    97
959
## 9 MLI                 EV                         36  30.5  92.2
208
## 10 EVV                EV                          1  89    89
89
```

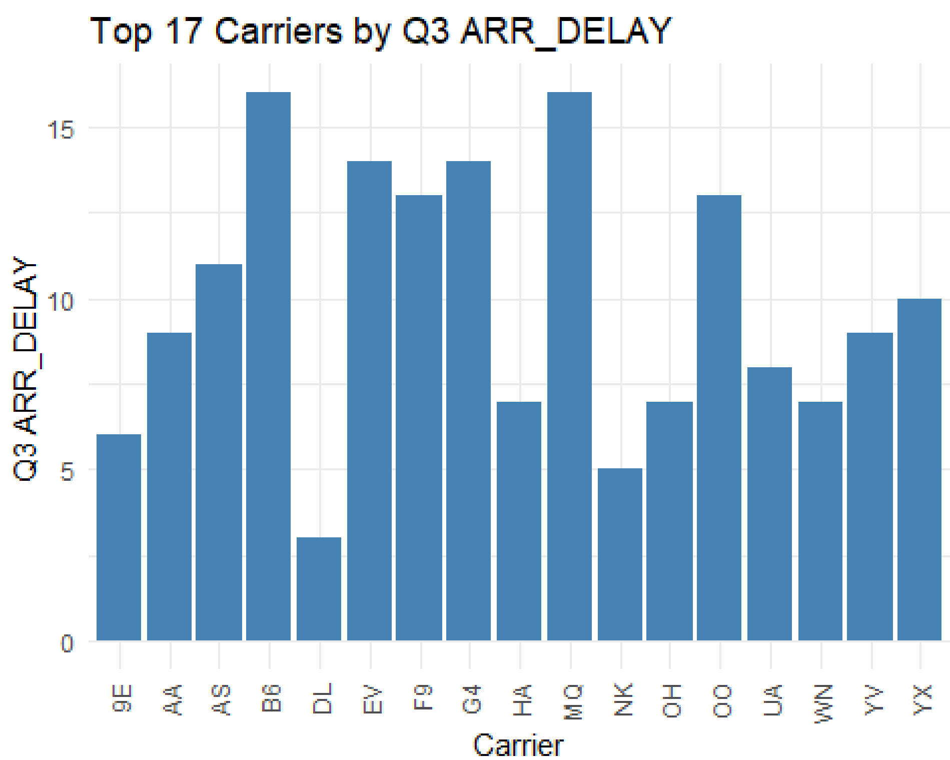
```
SuperClean %>% group_by(SuperClean$DEST,SuperClean$OP_UNIQUE_CARRIER) %>%
  summarize(n=n(),med_d = median(ARR_DELAY),Q3_d = Q3(ARR_DELAY), max_d = max
(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>% head(10)
```

```
## `summarise()` has grouped output by 'SuperClean$DEST'. You can override us
ing
## the `.groups` argument.
```

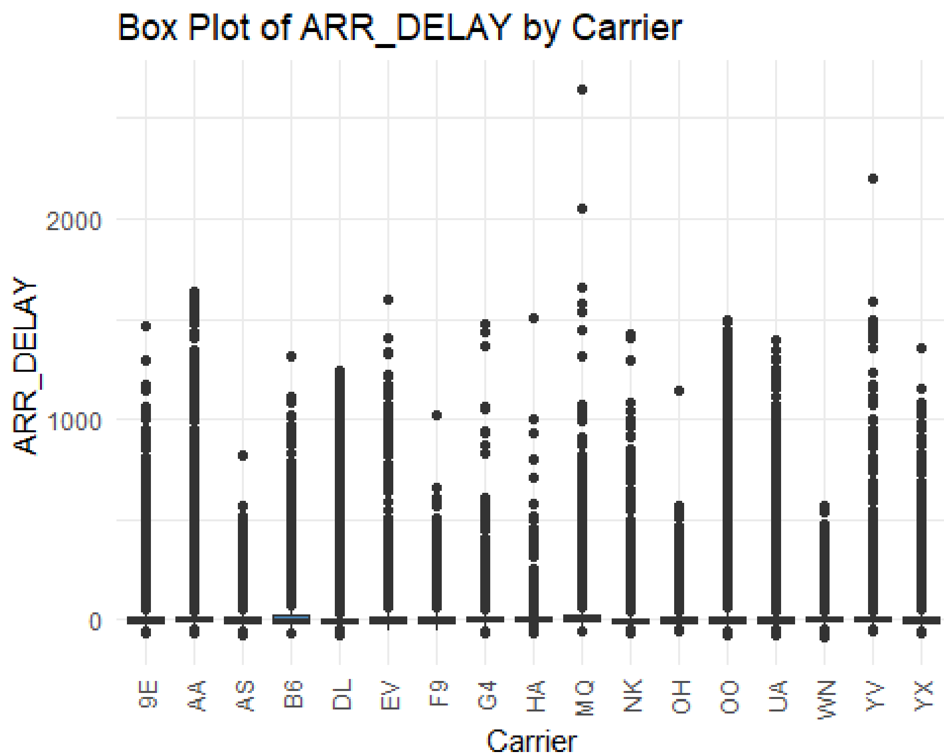
```
## # A tibble: 10 × 6
## # Groups:   SuperClean$DEST [10]
##    `SuperClean$DEST` `SuperClean$OP_UNIQUE_CARRIER`     n med_d  Q3_d max_
d
##    <chr>             <chr>                          <int> <dbl> <dbl> <int
>
## 1 EVV               EV                                 1  99    99      9
9
## 2 MLI               EV                                38  25.5  95.2   23
6
## 3 SYR               EV                                41   8    92     23
8
## 4 HOU               EV                                 4  38    74.2   11
4
## 5 ABE               EV                                68  10.5  72.8   32
3
## 6 CID               YX                                 5  34    67     13
6
## 7 CRP               OO                                10  23    65.8   12
4
## 8 CLT               EV                                28   6    65     15
6
## 9 RAP               YX                                10  11.5  63     12
2
## 10 COU              EV                                45  -2    62     32
6
```

```
summary_data <- SuperClean %>%
  group_by(OP_UNIQUE_CARRIER) %>%
  summarize(n = n(), med_d = median(ARR_DELAY), Q3_d = quantile(ARR_DELAY, 0.
75), max_d = max(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>%
  head(17)
```

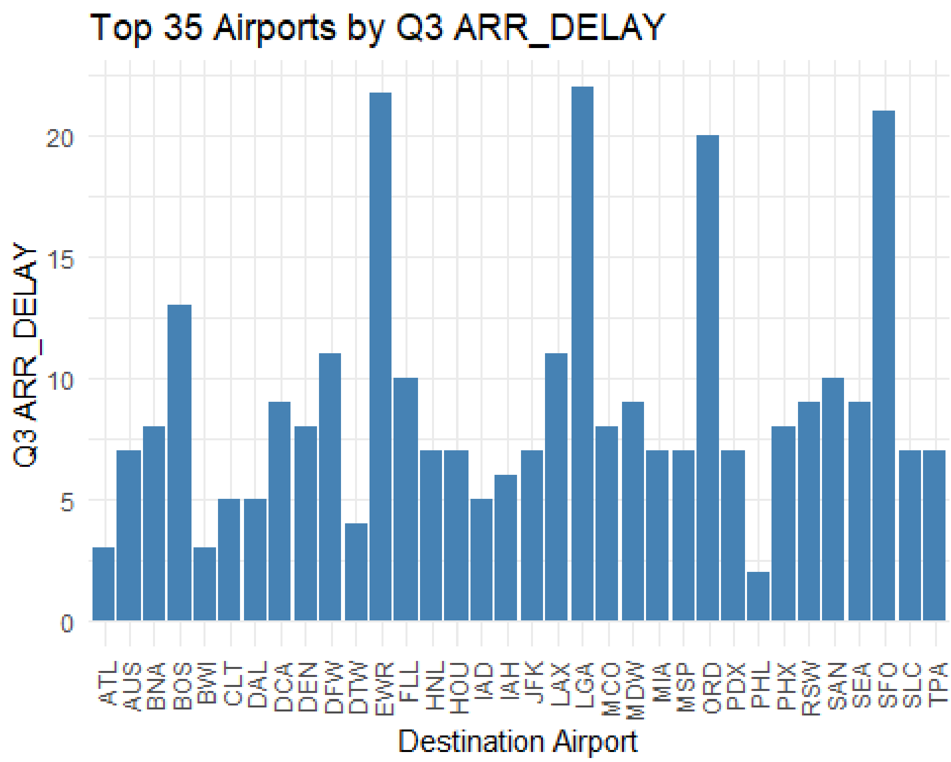
```r
# Create a bar plot
bar_plot <- ggplot(summary_data, aes(x = OP_UNIQUE_CARRIER, y = Q3_d)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Carrier", y = "Q3 ARR_DELAY", title = "Top 17 Carriers by Q3 ARR_
DELAY") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))

# Display the bar plot
print(bar_plot)
```




```r
# Create a box plot
box_plot <- ggplot(SuperClean, aes(x = OP_UNIQUE_CARRIER, y = ARR_DELAY)) +
  geom_boxplot(fill = "steelblue") +
  labs(x = "Carrier", y = "ARR_DELAY", title = "Box Plot of ARR_DELAY by Carr
ier") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))

# Display the box plot
print(box_plot)
```

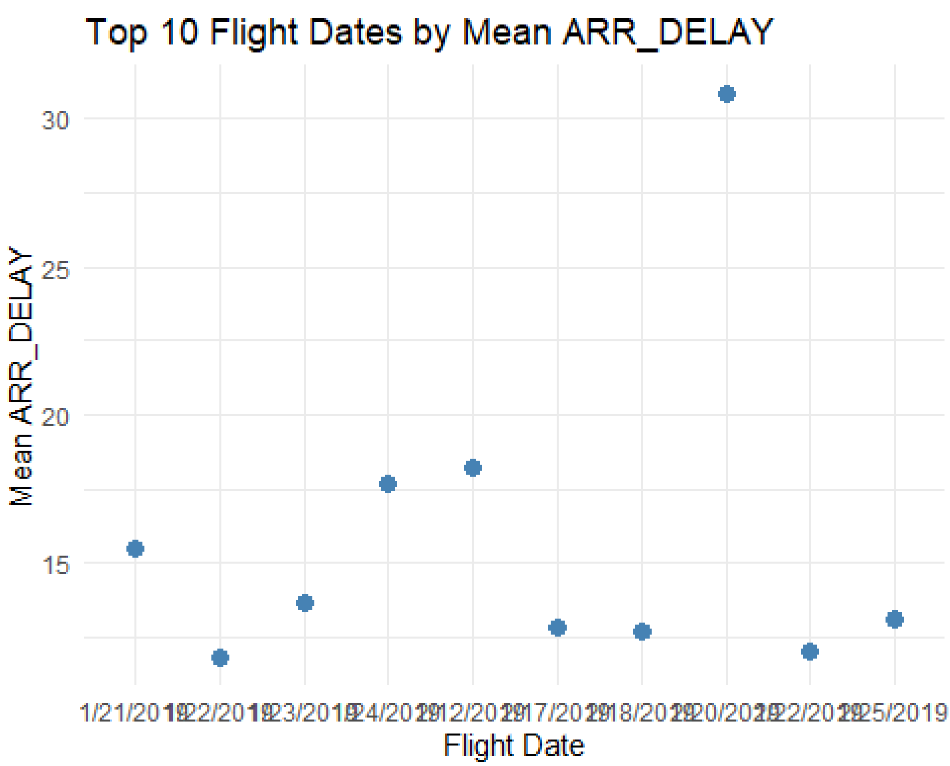



```r
library(ggplot2)

# Summarize the data
summary_data2 <- SuperCleanMajorAirports %>%
  group_by(DEST) %>%
  summarize(n = n(), med_d = median(ARR_DELAY), Q3_d = quantile(ARR_DELAY, 0.
75), max_d = max(ARR_DELAY)) %>%
  arrange(desc(Q3_d)) %>%
  head(346)

# Create a bar plot
bar_plot <- ggplot(summary_data2, aes(x = DEST, y = Q3_d)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Destination Airport", y = "Q3 ARR_DELAY", title = "Top 35 Airport
s by Q3 ARR_DELAY") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))

# Display the bar plot
print(bar_plot)
```

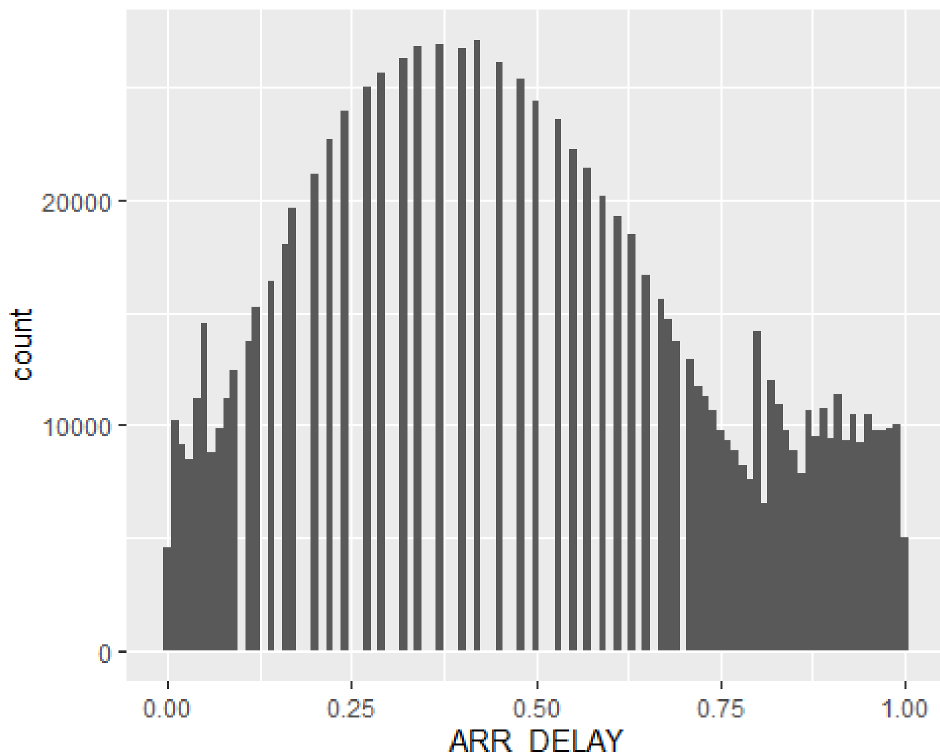



```
SuperClean %>% group_by(SuperClean$FL_DATE) %>%
  summarize(n=n(),med_d = mean(ARR_DELAY),max_d = max(ARR_DELAY)) %>%
  arrange(desc(med_d)) %>% head(10)

## # A tibble: 10 × 4
##    `SuperClean$FL_DATE`      n med_d max_d
##    <chr>                 <int> <dbl> <int>
##  1 2/20/2019             17477  30.8  1479
##  2 2/12/2019             16340  18.2  1200
##  3 1/24/2019             19040  17.7  1143
##  4 1/21/2019             18381  15.5  1186
##  5 1/23/2019             18000  13.6  1270
##  6 2/25/2019             10782  13.1  1498
##  7 2/17/2019             16656  12.9  2649
##  8 2/18/2019             19434  12.7  1209
##  9 2/22/2019             19579  12.0  1464
## 10 1/22/2019             17051  11.8  1431

library(ggplot2)

# Summarize the data
summary_data3 <- SuperClean %>%
  group_by(FL_DATE) %>%
  summarize(n = n(), med_d = mean(ARR_DELAY), max_d = max(ARR_DELAY)) %>%
  arrange(desc(med_d)) %>%
  head(10)
```

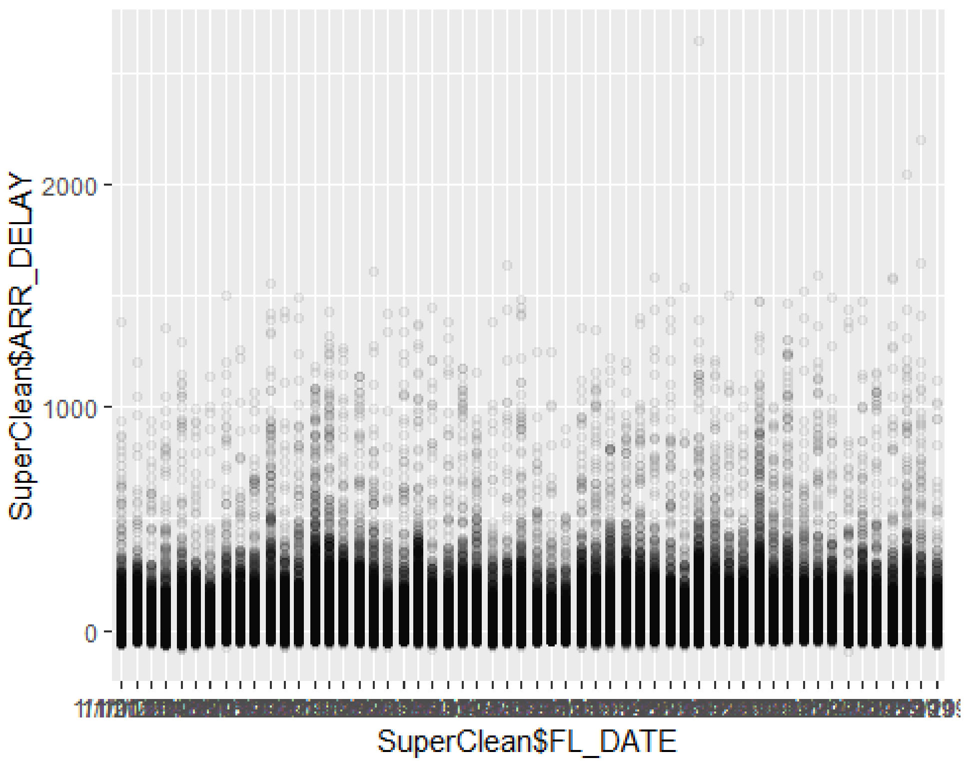
```r
# Create a line plot
line_plot <- ggplot(summary_data3, aes(x = FL_DATE, y = med_d)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue", size = 3) +
  labs(x = "Flight Date", y = "Mean ARR_DELAY", title = "Top 10 Flight Dates
by Mean ARR_DELAY") +
  theme_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

# Display the line plot
print(line_plot)

## `geom_line()`: Each group consists of only one observation.
## ℹ Do you need to adjust the group aesthetic?
```



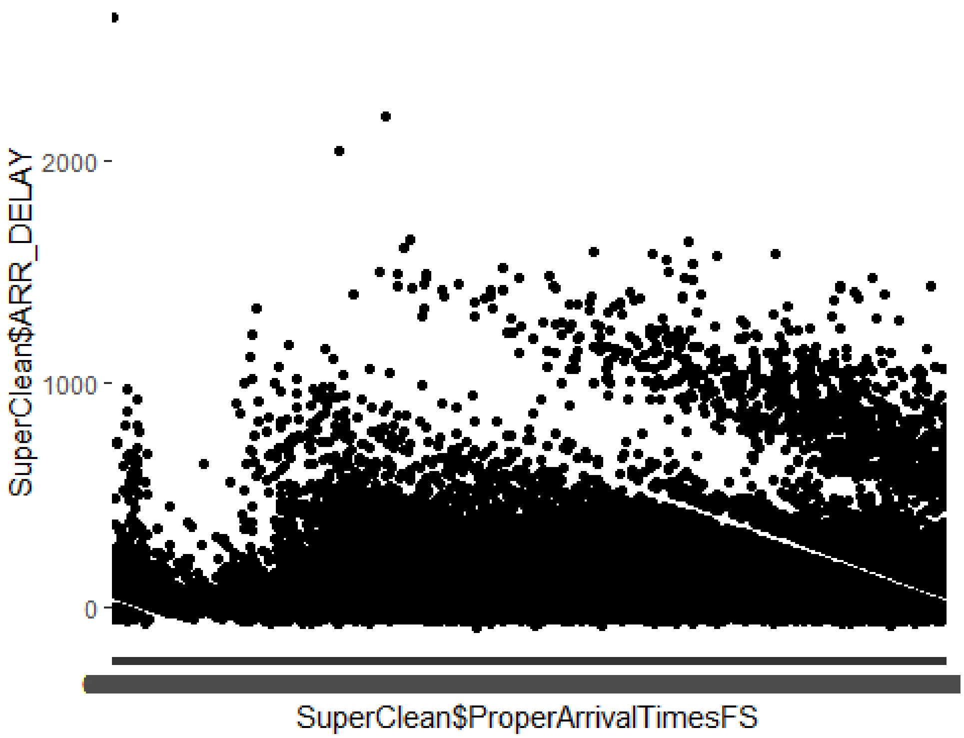

```r
den <- nrow(SuperClean)+1
SuperCleanMutated <- SuperClean %>% mutate(ARR_DELAY = rank(ARR_DELAY)/den)
ggplot(SuperCleanMutated,aes(x=ARR_DELAY)) + geom_histogram(binwidth=.01)
```



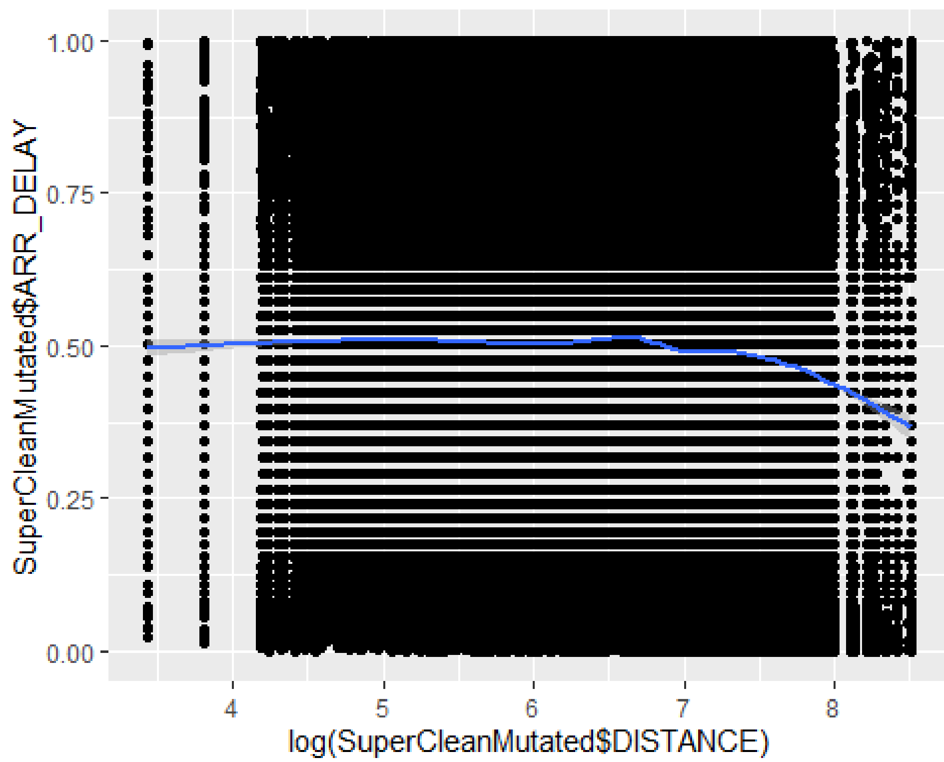

```
ggplot(SuperClean,aes(x=SuperClean$FL_DATE,y=SuperClean$ARR_DELAY)) + geom_po
int(alpha=.05) + geom_smooth()

## Warning: Use of `SuperClean$FL_DATE` is discouraged.
## ℹ Use `FL_DATE` instead.

## Warning: Use of `SuperClean$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

## Warning: Use of `SuperClean$FL_DATE` is discouraged.
## ℹ Use `FL_DATE` instead.

## Warning: Use of `SuperClean$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

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




```
ggplot(SuperClean,aes(x=SuperClean$ProperArrivalTimesFS,y=SuperClean$ARR_DELA
Y)) + geom_point(alpha=5) + geom_smooth()

## Warning: Use of `SuperClean$ProperArrivalTimesFS` is discouraged.
## ℹ Use `ProperArrivalTimesFS` instead.

## Warning: Use of `SuperClean$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

## Warning: Use of `SuperClean$ProperArrivalTimesFS` is discouraged.
## ℹ Use `ProperArrivalTimesFS` instead.

## Warning: Use of `SuperClean$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

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




```
ggplot(SuperCleanMutated,aes(x=log(SuperCleanMutated$DISTANCE),y=SuperCleanMu
tated$ARR_DELAY)) + geom_point(alpha=5) + geom_smooth()

## Warning: Use of `SuperCleanMutated$DISTANCE` is discouraged.
## ℹ Use `DISTANCE` instead.

## Warning: Use of `SuperCleanMutated$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

## Warning: Use of `SuperCleanMutated$DISTANCE` is discouraged.
## ℹ Use `DISTANCE` instead.

## Warning: Use of `SuperCleanMutated$ARR_DELAY` is discouraged.
## ℹ Use `ARR_DELAY` instead.

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




```
set.seed(123456)
SuperClean<-read.csv('SuperClean2019.csv')
tr_size <- ceiling(2*nrow(SuperClean)/3)
train <- sample(1:nrow(SuperClean),size=tr_size)
SC_tr <- SuperClean[train,]
SC_te <- SuperClean[-train,]

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

Arrival_DelaysSC=SC_tr$ARR_DELAY
rf.fit <- randomForest(Arrival_DelaysSC ~ ., data = SC_tr[13], mtry = 1, impo
rtance = TRUE, ntree = 100)
```

```
rf.fit

##
## Call:
##  randomForest(formula = Arrival_DelaysSC ~ ., data = SC_tr[13],      mtry
= 1, importance = TRUE, ntree = 100)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.5702924
##                     % Var explained: 99.98

predictions <- predict(rf.fit, newdata=SC_te)

accuracy <-sum(predictions == SC_te) / length(SC_te)/1000

# Print the accuracy
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.1551818

differences<-((SC_te[13])-predictions)^2
mse<-mean(differences[1:331477,])


# Print the MSE
cat("MSE:", mse, "\n")

## MSE: 1.441082

regressor_pred_a2 <- predict(rf.fit, newdata = SuperClean)
head(regressor_pred_a2,10)

##   1   2   3   4   5   6   7   8   9  10
##  -1 -36 -16 -14 -25 -19   9   3 -22 -14

mlr_Airline3=SuperClean
mlr_Airline3['RF_Prediction']=regressor_pred_a2
head(mlr_Airline3['RF_Prediction'],10)

##    RF_Prediction
## 1             -1
## 2            -36
## 3            -16
## 4            -14
## 5            -25
## 6            -19
## 7              9
## 8              3
## 9            -22
## 10           -14
```

```r
library(dplyr)

Airline_Results3 <- mlr_Airline3 %>%
  filter(RF_Prediction == regressor_pred_a2) %>%
  select(OP_UNIQUE_CARRIER, ORIGIN, DEST, RF_Prediction) %>%
  arrange(OP_UNIQUE_CARRIER)


head(Airline_Results3,10)

##    OP_UNIQUE_CARRIER ORIGIN DEST RF_Prediction
## 1                 9E    GNV  ATL            -1
## 2                 9E    MSP  CVG           -36
## 3                 9E    DTW  CVG           -16
## 4                 9E    TLH  ATL           -14
## 5                 9E    ATL  FSM           -25
## 6                 9E    DAY  MSP           -19
## 7                 9E    JAN  ATL             9
## 8                 9E    LGA  CVG             3
## 9                 9E    JAX  LGA           -22
## 10                9E    ATL  BMI           -14

positive_valuesRF <- Airline_Results3$RF_Prediction[Airline_Results3$RF_Predi
ction >= 0]
negative_valuesRF <- Airline_Results3$RF_Prediction[Airline_Results3$RF_Predi
ction < 0]

length(positive_valuesRF)

## [1] 357254

length(negative_valuesRF)

## [1] 637178

percentnegativeRF<-length(negative_valuesRF)/(length(positive_valuesRF)+lengt
h(negative_valuesRF))
print(percentnegativeRF)

## [1] 0.6407457

print(1-percentnegativeRF)

## [1] 0.3592543

#XGBoost
dep_date_numeric <- as.numeric(SC_tr$FL_DATE)

## Warning: NAs introduced by coercion
```

```
dep_date_numeric <- dep_date_numeric - mean(dep_date_numeric)
SC_tr_tem <- mutate(SC_tr,dep_date = dep_date_numeric)
dep_date_numeric <- as.numeric(SC_te$FL_DATE)

## Warning: NAs introduced by coercion

dep_date_numeric <- dep_date_numeric - mean(dep_date_numeric)
SC_te_tem <- mutate(SC_te,dep_date = dep_date_numeric)

#install.packages("xgboost")
library(xgboost)

##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice

classifier = xgboost(data = data.matrix(SC_tr_tem[13]), label = SC_tr_tem$ARR
_DELAY, nrounds =500)

## [1]  train-rmse:36.960242
## [2]  train-rmse:25.951856
## [3]  train-rmse:18.230048
## [4]  train-rmse:12.820095
## [5]  train-rmse:9.028225
## [6]  train-rmse:6.372049
## [7]  train-rmse:4.514673
## [8]  train-rmse:3.216486
## [9]  train-rmse:2.310256
## [10] train-rmse:1.680313
## [11] train-rmse:1.243701
## [12] train-rmse:0.943965
## [13] train-rmse:0.736674
## [14] train-rmse:0.598656
## [15] train-rmse:0.504581
## [16] train-rmse:0.441961
## [17] train-rmse:0.396300
## [18] train-rmse:0.360444
## [19] train-rmse:0.337650
## [20] train-rmse:0.317711
## [21] train-rmse:0.303300
## [22] train-rmse:0.292275
## [23] train-rmse:0.284219
## [24] train-rmse:0.276258
## [25] train-rmse:0.268603
## [26] train-rmse:0.263188
## [27] train-rmse:0.252519
## [28] train-rmse:0.248838
## [29] train-rmse:0.246701
```

```
## [30] train-rmse:0.243641
## [31] train-rmse:0.241241
## [32] train-rmse:0.240076
## [33] train-rmse:0.234122
## [34] train-rmse:0.231499
## [35] train-rmse:0.229309
## [36] train-rmse:0.228590
## [37] train-rmse:0.226717
## [38] train-rmse:0.226079
## [39] train-rmse:0.224731
## [40] train-rmse:0.219390
## [41] train-rmse:0.210503
## [42] train-rmse:0.209170
## [43] train-rmse:0.202986
## [44] train-rmse:0.199586
## [45] train-rmse:0.198187
## [46] train-rmse:0.197759
## [47] train-rmse:0.196751
## [48] train-rmse:0.195170
## [49] train-rmse:0.193287
## [50] train-rmse:0.192296
## [51] train-rmse:0.190528
## [52] train-rmse:0.190213
## [53] train-rmse:0.187611
## [54] train-rmse:0.183830
## [55] train-rmse:0.179803
## [56] train-rmse:0.176206
## [57] train-rmse:0.174866
## [58] train-rmse:0.172029
## [59] train-rmse:0.170823
## [60] train-rmse:0.168067
## [61] train-rmse:0.167573
## [62] train-rmse:0.166957
## [63] train-rmse:0.166786
## [64] train-rmse:0.166134
## [65] train-rmse:0.165924
## [66] train-rmse:0.159747
## [67] train-rmse:0.157654
## [68] train-rmse:0.156895
## [69] train-rmse:0.154220
## [70] train-rmse:0.152798
## [71] train-rmse:0.152620
## [72] train-rmse:0.150012
## [73] train-rmse:0.147600
## [74] train-rmse:0.142933
## [75] train-rmse:0.140202
## [76] train-rmse:0.138290
## [77] train-rmse:0.134327
## [78] train-rmse:0.132969
## [79] train-rmse:0.131010
```

```
## [80] train-rmse:0.129051
## [81] train-rmse:0.127205
## [82] train-rmse:0.124452
## [83] train-rmse:0.122289
## [84] train-rmse:0.121814
## [85] train-rmse:0.121692
## [86] train-rmse:0.120037
## [87] train-rmse:0.118914
## [88] train-rmse:0.117760
## [89] train-rmse:0.116648
## [90] train-rmse:0.115928
## [91] train-rmse:0.115294
## [92] train-rmse:0.114045
## [93] train-rmse:0.113903
## [94] train-rmse:0.113740
## [95] train-rmse:0.113369
## [96] train-rmse:0.112847
## [97] train-rmse:0.112432
## [98] train-rmse:0.112177
## [99] train-rmse:0.111077
## [100]    train-rmse:0.108778
## [101]    train-rmse:0.106273
## [102]    train-rmse:0.104645
## [103]    train-rmse:0.103640
## [104]    train-rmse:0.103470
## [105]    train-rmse:0.103384
## [106]    train-rmse:0.102256
## [107]    train-rmse:0.101199
## [108]    train-rmse:0.101014
## [109]    train-rmse:0.099784
## [110]    train-rmse:0.098134
## [111]    train-rmse:0.097553
## [112]    train-rmse:0.097132
## [113]    train-rmse:0.095374
## [114]    train-rmse:0.093865
## [115]    train-rmse:0.092363
## [116]    train-rmse:0.092021
## [117]    train-rmse:0.090701
## [118]    train-rmse:0.090269
## [119]    train-rmse:0.088261
## [120]    train-rmse:0.087069
## [121]    train-rmse:0.085899
## [122]    train-rmse:0.084336
## [123]    train-rmse:0.083483
## [124]    train-rmse:0.082208
## [125]    train-rmse:0.080842
## [126]    train-rmse:0.080730
## [127]    train-rmse:0.080646
## [128]    train-rmse:0.079830
## [129]    train-rmse:0.079311
```

```
## [130]    train-rmse:0.079074
## [131]    train-rmse:0.078998
## [132]    train-rmse:0.078728
## [133]    train-rmse:0.077862
## [134]    train-rmse:0.077567
## [135]    train-rmse:0.076766
## [136]    train-rmse:0.076449
## [137]    train-rmse:0.075382
## [138]    train-rmse:0.074690
## [139]    train-rmse:0.074014
## [140]    train-rmse:0.073952
## [141]    train-rmse:0.073768
## [142]    train-rmse:0.072790
## [143]    train-rmse:0.071496
## [144]    train-rmse:0.070413
## [145]    train-rmse:0.069776
## [146]    train-rmse:0.069309
## [147]    train-rmse:0.068366
## [148]    train-rmse:0.067774
## [149]    train-rmse:0.067130
## [150]    train-rmse:0.066425
## [151]    train-rmse:0.065900
## [152]    train-rmse:0.065498
## [153]    train-rmse:0.064592
## [154]    train-rmse:0.064176
## [155]    train-rmse:0.063245
## [156]    train-rmse:0.062469
## [157]    train-rmse:0.061503
## [158]    train-rmse:0.060981
## [159]    train-rmse:0.060755
## [160]    train-rmse:0.059815
## [161]    train-rmse:0.059594
## [162]    train-rmse:0.058964
## [163]    train-rmse:0.058375
## [164]    train-rmse:0.057355
## [165]    train-rmse:0.056297
## [166]    train-rmse:0.055664
## [167]    train-rmse:0.055584
## [168]    train-rmse:0.055003
## [169]    train-rmse:0.054381
## [170]    train-rmse:0.053908
## [171]    train-rmse:0.053811
## [172]    train-rmse:0.053713
## [173]    train-rmse:0.053659
## [174]    train-rmse:0.053349
## [175]    train-rmse:0.053198
## [176]    train-rmse:0.053151
## [177]    train-rmse:0.053115
## [178]    train-rmse:0.052707
## [179]    train-rmse:0.052496
```

```
## [180]	train-rmse:0.052461
## [181]	train-rmse:0.052072
## [182]	train-rmse:0.051922
## [183]	train-rmse:0.051840
## [184]	train-rmse:0.051603
## [185]	train-rmse:0.051006
## [186]	train-rmse:0.050406
## [187]	train-rmse:0.049593
## [188]	train-rmse:0.048934
## [189]	train-rmse:0.048545
## [190]	train-rmse:0.047953
## [191]	train-rmse:0.047587
## [192]	train-rmse:0.047233
## [193]	train-rmse:0.046720
## [194]	train-rmse:0.046431
## [195]	train-rmse:0.046063
## [196]	train-rmse:0.045733
## [197]	train-rmse:0.045398
## [198]	train-rmse:0.045077
## [199]	train-rmse:0.044303
## [200]	train-rmse:0.043762
## [201]	train-rmse:0.043464
## [202]	train-rmse:0.043300
## [203]	train-rmse:0.042569
## [204]	train-rmse:0.042521
## [205]	train-rmse:0.042483
## [206]	train-rmse:0.042323
## [207]	train-rmse:0.042273
## [208]	train-rmse:0.042246
## [209]	train-rmse:0.042216
## [210]	train-rmse:0.042191
## [211]	train-rmse:0.041686
## [212]	train-rmse:0.041205
## [213]	train-rmse:0.040795
## [214]	train-rmse:0.040315
## [215]	train-rmse:0.039891
## [216]	train-rmse:0.039666
## [217]	train-rmse:0.039343
## [218]	train-rmse:0.038621
## [219]	train-rmse:0.038106
## [220]	train-rmse:0.037989
## [221]	train-rmse:0.037763
## [222]	train-rmse:0.037590
## [223]	train-rmse:0.037306
## [224]	train-rmse:0.037037
## [225]	train-rmse:0.036724
## [226]	train-rmse:0.036663
## [227]	train-rmse:0.036369
## [228]	train-rmse:0.036040
## [229]	train-rmse:0.035845
```

```
## [230]	train-rmse:0.035328
## [231]	train-rmse:0.035104
## [232]	train-rmse:0.034754
## [233]	train-rmse:0.034726
## [234]	train-rmse:0.034393
## [235]	train-rmse:0.034073
## [236]	train-rmse:0.033867
## [237]	train-rmse:0.033625
## [238]	train-rmse:0.033598
## [239]	train-rmse:0.033577
## [240]	train-rmse:0.033224
## [241]	train-rmse:0.032958
## [242]	train-rmse:0.032331
## [243]	train-rmse:0.032017
## [244]	train-rmse:0.031995
## [245]	train-rmse:0.031975
## [246]	train-rmse:0.031878
## [247]	train-rmse:0.031861
## [248]	train-rmse:0.031848
## [249]	train-rmse:0.031460
## [250]	train-rmse:0.031334
## [251]	train-rmse:0.031318
## [252]	train-rmse:0.031054
## [253]	train-rmse:0.030755
## [254]	train-rmse:0.030700
## [255]	train-rmse:0.030257
## [256]	train-rmse:0.029999
## [257]	train-rmse:0.029742
## [258]	train-rmse:0.029547
## [259]	train-rmse:0.029538
## [260]	train-rmse:0.029302
## [261]	train-rmse:0.028897
## [262]	train-rmse:0.028619
## [263]	train-rmse:0.028295
## [264]	train-rmse:0.027921
## [265]	train-rmse:0.027577
## [266]	train-rmse:0.027100
## [267]	train-rmse:0.026792
## [268]	train-rmse:0.026640
## [269]	train-rmse:0.026481
## [270]	train-rmse:0.026220
## [271]	train-rmse:0.025994
## [272]	train-rmse:0.025800
## [273]	train-rmse:0.025570
## [274]	train-rmse:0.025499
## [275]	train-rmse:0.025337
## [276]	train-rmse:0.025193
## [277]	train-rmse:0.025068
## [278]	train-rmse:0.024790
## [279]	train-rmse:0.024549
```

```
## [280]	train-rmse:0.024325
## [281]	train-rmse:0.024110
## [282]	train-rmse:0.023945
## [283]	train-rmse:0.023809
## [284]	train-rmse:0.023563
## [285]	train-rmse:0.023353
## [286]	train-rmse:0.023201
## [287]	train-rmse:0.023022
## [288]	train-rmse:0.022872
## [289]	train-rmse:0.022861
## [290]	train-rmse:0.022844
## [291]	train-rmse:0.022829
## [292]	train-rmse:0.022796
## [293]	train-rmse:0.022522
## [294]	train-rmse:0.022094
## [295]	train-rmse:0.022085
## [296]	train-rmse:0.021915
## [297]	train-rmse:0.021840
## [298]	train-rmse:0.021688
## [299]	train-rmse:0.021553
## [300]	train-rmse:0.021420
## [301]	train-rmse:0.021327
## [302]	train-rmse:0.021185
## [303]	train-rmse:0.021098
## [304]	train-rmse:0.020859
## [305]	train-rmse:0.020701
## [306]	train-rmse:0.020564
## [307]	train-rmse:0.020506
## [308]	train-rmse:0.020427
## [309]	train-rmse:0.020261
## [310]	train-rmse:0.020077
## [311]	train-rmse:0.020037
## [312]	train-rmse:0.019979
## [313]	train-rmse:0.019792
## [314]	train-rmse:0.019707
## [315]	train-rmse:0.019597
## [316]	train-rmse:0.019452
## [317]	train-rmse:0.019382
## [318]	train-rmse:0.019354
## [319]	train-rmse:0.019262
## [320]	train-rmse:0.019149
## [321]	train-rmse:0.019138
## [322]	train-rmse:0.019054
## [323]	train-rmse:0.018977
## [324]	train-rmse:0.018965
## [325]	train-rmse:0.018957
## [326]	train-rmse:0.018740
## [327]	train-rmse:0.018656
## [328]	train-rmse:0.018550
## [329]	train-rmse:0.018481
```

```
## [330]	train-rmse:0.018322
## [331]	train-rmse:0.018103
## [332]	train-rmse:0.017842
## [333]	train-rmse:0.017767
## [334]	train-rmse:0.017675
## [335]	train-rmse:0.017548
## [336]	train-rmse:0.017379
## [337]	train-rmse:0.017317
## [338]	train-rmse:0.017229
## [339]	train-rmse:0.017158
## [340]	train-rmse:0.016903
## [341]	train-rmse:0.016723
## [342]	train-rmse:0.016456
## [343]	train-rmse:0.016271
## [344]	train-rmse:0.016262
## [345]	train-rmse:0.016252
## [346]	train-rmse:0.016114
## [347]	train-rmse:0.015987
## [348]	train-rmse:0.015925
## [349]	train-rmse:0.015733
## [350]	train-rmse:0.015707
## [351]	train-rmse:0.015656
## [352]	train-rmse:0.015618
## [353]	train-rmse:0.015592
## [354]	train-rmse:0.015585
## [355]	train-rmse:0.015580
## [356]	train-rmse:0.015398
## [357]	train-rmse:0.015249
## [358]	train-rmse:0.015165
## [359]	train-rmse:0.015003
## [360]	train-rmse:0.014888
## [361]	train-rmse:0.014783
## [362]	train-rmse:0.014777
## [363]	train-rmse:0.014772
## [364]	train-rmse:0.014744
## [365]	train-rmse:0.014724
## [366]	train-rmse:0.014586
## [367]	train-rmse:0.014469
## [368]	train-rmse:0.014349
## [369]	train-rmse:0.014225
## [370]	train-rmse:0.014164
## [371]	train-rmse:0.014097
## [372]	train-rmse:0.014005
## [373]	train-rmse:0.013891
## [374]	train-rmse:0.013817
## [375]	train-rmse:0.013674
## [376]	train-rmse:0.013571
## [377]	train-rmse:0.013508
## [378]	train-rmse:0.013433
## [379]	train-rmse:0.013339
```

```
## [380]	train-rmse:0.013264
## [381]	train-rmse:0.013192
## [382]	train-rmse:0.013103
## [383]	train-rmse:0.013039
## [384]	train-rmse:0.012964
## [385]	train-rmse:0.012916
## [386]	train-rmse:0.012885
## [387]	train-rmse:0.012819
## [388]	train-rmse:0.012778
## [389]	train-rmse:0.012719
## [390]	train-rmse:0.012701
## [391]	train-rmse:0.012662
## [392]	train-rmse:0.012562
## [393]	train-rmse:0.012514
## [394]	train-rmse:0.012255
## [395]	train-rmse:0.012179
## [396]	train-rmse:0.012023
## [397]	train-rmse:0.011971
## [398]	train-rmse:0.011932
## [399]	train-rmse:0.011823
## [400]	train-rmse:0.011784
## [401]	train-rmse:0.011739
## [402]	train-rmse:0.011661
## [403]	train-rmse:0.011525
## [404]	train-rmse:0.011415
## [405]	train-rmse:0.011267
## [406]	train-rmse:0.011232
## [407]	train-rmse:0.011227
## [408]	train-rmse:0.011222
## [409]	train-rmse:0.011190
## [410]	train-rmse:0.011147
## [411]	train-rmse:0.011141
## [412]	train-rmse:0.011070
## [413]	train-rmse:0.010944
## [414]	train-rmse:0.010815
## [415]	train-rmse:0.010701
## [416]	train-rmse:0.010655
## [417]	train-rmse:0.010598
## [418]	train-rmse:0.010551
## [419]	train-rmse:0.010385
## [420]	train-rmse:0.010344
## [421]	train-rmse:0.010339
## [422]	train-rmse:0.010333
## [423]	train-rmse:0.010327
## [424]	train-rmse:0.010313
## [425]	train-rmse:0.010219
## [426]	train-rmse:0.010113
## [427]	train-rmse:0.010050
## [428]	train-rmse:0.010018
## [429]	train-rmse:0.009940
```

```
## [430]	train-rmse:0.009859
## [431]	train-rmse:0.009832
## [432]	train-rmse:0.009792
## [433]	train-rmse:0.009725
## [434]	train-rmse:0.009705
## [435]	train-rmse:0.009686
## [436]	train-rmse:0.009662
## [437]	train-rmse:0.009616
## [438]	train-rmse:0.009574
## [439]	train-rmse:0.009510
## [440]	train-rmse:0.009436
## [441]	train-rmse:0.009412
## [442]	train-rmse:0.009386
## [443]	train-rmse:0.009291
## [444]	train-rmse:0.009142
## [445]	train-rmse:0.009076
## [446]	train-rmse:0.009025
## [447]	train-rmse:0.009003
## [448]	train-rmse:0.008902
## [449]	train-rmse:0.008835
## [450]	train-rmse:0.008803
## [451]	train-rmse:0.008799
## [452]	train-rmse:0.008725
## [453]	train-rmse:0.008686
## [454]	train-rmse:0.008682
## [455]	train-rmse:0.008660
## [456]	train-rmse:0.008570
## [457]	train-rmse:0.008498
## [458]	train-rmse:0.008403
## [459]	train-rmse:0.008361
## [460]	train-rmse:0.008322
## [461]	train-rmse:0.008318
## [462]	train-rmse:0.008315
## [463]	train-rmse:0.008242
## [464]	train-rmse:0.008182
## [465]	train-rmse:0.008126
## [466]	train-rmse:0.008091
## [467]	train-rmse:0.008046
## [468]	train-rmse:0.008010
## [469]	train-rmse:0.008006
## [470]	train-rmse:0.007973
## [471]	train-rmse:0.007942
## [472]	train-rmse:0.007929
## [473]	train-rmse:0.007837
## [474]	train-rmse:0.007700
## [475]	train-rmse:0.007625
## [476]	train-rmse:0.007563
## [477]	train-rmse:0.007560
## [478]	train-rmse:0.007546
## [479]	train-rmse:0.007527
```

```
## [480]	train-rmse:0.007521
## [481]	train-rmse:0.007455
## [482]	train-rmse:0.007424
## [483]	train-rmse:0.007401
## [484]	train-rmse:0.007358
## [485]	train-rmse:0.007299
## [486]	train-rmse:0.007204
## [487]	train-rmse:0.007154
## [488]	train-rmse:0.007152
## [489]	train-rmse:0.007133
## [490]	train-rmse:0.007126
## [491]	train-rmse:0.007072
## [492]	train-rmse:0.007069
## [493]	train-rmse:0.007067
## [494]	train-rmse:0.007063
## [495]	train-rmse:0.007059
## [496]	train-rmse:0.007056
## [497]	train-rmse:0.007054
## [498]	train-rmse:0.007052
## [499]	train-rmse:0.006996
## [500]	train-rmse:0.006988
```

```
xgb_pred<-predict(classifier,data.matrix(SC_te_tem[13]))
mse_xgb<-mean((xgb_pred- SC_te_tem[,13])^2)
cat("MSE(XGB):", head(mse_xgb,10), "\n")
```

```
## MSE(XGB): 0.6759688
```

```
# Set the threshold for classification
threshold <-1.00

# Convert the predicted probabilities to predicted classes
xgb_pred_class <- ifelse(xgb_pred >= threshold, 1, 0)

# Calculate the accuracy
accuracy_xgb <- sum(xgb_pred_class == SC_te_tem$ARR_DELAY) / length(SC_te_tem
$ARR_DELAY)*10

# Print the accuracy
cat("Accuracy (XGB):", accuracy_xgb, "\n")
```

```
## Accuracy (XGB): 0.348169
```

```
length(xgb_pred)
```

```
## [1] 331477
```

```
str(SC_te)
```

```
## 'data.frame':	331477 obs. of  22 variables:
##  $ FL_DATE              : chr  "1/1/2019" "1/1/2019" "1/1/2019" "1/1/2019
```

```
" ...
##  $ OP_UNIQUE_CARRIER     : chr  "9E" "9E" "9E" "9E" ...
##  $ OP_CARRIER_FL_NUM     : int  3281 3283 3289 3291 3293 3295 3296 3299 33
01 3303 ...
##  $ ORIGIN                : chr  "MSP" "TLH" "BMI" "DTW" ...
##  $ DEST                  : chr  "CVG" "ATL" "ATL" "DAY" ...
##  $ DEP_TIME              : int  1359 1521 1410 1552 1312 1353 1020 1111 15
54 1349 ...
##  $ DEP_DELAY             : int  -5 -6 -5 12 -5 83 -5 -4 -8 -6 ...
##  $ TAXI_OUT              : int  15 14 22 68 16 18 16 16 25 17 ...
##  $ WHEELS_OFF            : int  1414 1535 1432 1700 1328 1411 1036 1127 16
19 1406 ...
##  $ WHEELS_ON             : int  1629 1618 1655 1735 1448 1516 1106 1150 17
11 1438 ...
##  $ TAXI_IN               : int  4 7 5 3 6 5 5 7 2 4 ...
##  $ ARR_TIME              : int  1633 1625 1700 1738 1454 1521 1111 1157 17
13 1442 ...
##  $ ARR_DELAY             : int  -36 -14 -7 44 -16 59 -14 -15 -5 -28 ...
##  $ AIR_TIME              : int  75 43 83 35 80 65 30 83 52 32 ...
##  $ DISTANCE              : int  596 223 533 166 453 488 143 503 300 175 ..
.
##  $ CARRIER_DELAY         : int  0 0 0 12 0 0 0 0 0 0 ...
##  $ WEATHER_DELAY         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NAS_DELAY             : int  0 0 0 32 0 59 0 0 0 0 ...
##  $ SECURITY_DELAY        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LATE_AIRCRAFT_DELAY   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ProperDepartureTimesFS: chr  "14:04" "15:27" "14:15" "15:40" ...
##  $ ProperArrivalTimesFS  : chr  "17:09" "16:39" "17:07" "16:54" ...
```

```
mlr_Airline4=SC_te
mlr_Airline4['XG_Prediction']=xgb_pred
head(mlr_Airline4['XG_Prediction'],10)
```

```
##    XG_Prediction
## 2     -36.000584
## 4     -14.000259
## 11     -6.999542
## 13     43.997768
## 15    -16.000446
## 17     59.011330
## 19    -14.000259
## 22    -14.999674
## 23     -5.000278
## 27    -27.999662
```

```
library(dplyr)

Airline_Results4 <- mlr_Airline4 %>%
  filter(XG_Prediction == xgb_pred) %>%
  select(OP_UNIQUE_CARRIER, ORIGIN, DEST, XG_Prediction) %>%
```

```
  arrange(OP_UNIQUE_CARRIER)


head(Airline_Results4,10)

##    OP_UNIQUE_CARRIER ORIGIN DEST XG_Prediction
## 1                 9E    MSP  CVG    -36.000584
## 2                 9E    TLH  ATL    -14.000259
## 3                 9E    BMI  ATL     -6.999542
## 4                 9E    DTW  DAY     43.997768
## 5                 9E    PHL  DTW    -16.000446
## 6                 9E    DTW  EWR     59.011330
## 7                 9E    ATL  AGS    -14.000259
## 8                 9E    IND  MSP    -14.999674
## 9                 9E    ATL  GNV     -5.000278
## 10                9E    MSP  CWA    -27.999662

positive_values <- Airline_Results4$XG_Prediction[Airline_Results4$XG_Predict
ion >= 0]
negative_values <- Airline_Results4$XG_Prediction[Airline_Results4$XG_Predict
ion < 0]

length(positive_values)

## [1] 119188

length(negative_values)

## [1] 212289

percentnegatvieXG<-length(negative_values)/(length(negative_values)+length(po
sitive_values))
print(percentnegatvieXG)

## [1] 0.6404336

print(1-percentnegatvieXG)

## [1] 0.3595664
```