STAT480_Homework_2

Bin Feng

```
#include library
library(reshape2)
library(ggplot2)
library(biganalytics)
```

```
## Loading required package: bigmemory
```

```
## Loading required package: foreach
```

```
## Loading required package: biglm
```

```
## Loading required package: DBI
```

```
library(foreach)
library(parallel)
library(doSNOW)
```

```
## Loading required package: iterators
```

```
## Loading required package: snow
```

```
##
## Attaching package: 'snow'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, clusterSplit, makeCluster,
##     parApply, parCapply, parLapply, parRapply, parSapply,
##     splitIndices, stopCluster
```

```
require("knitr")
```

```
## Loading required package: knitr
```

```
#set working directory
opts_knit$set(root.dir = "~/Stat480/RDataScience/AirlineDelays")
```

Question 1

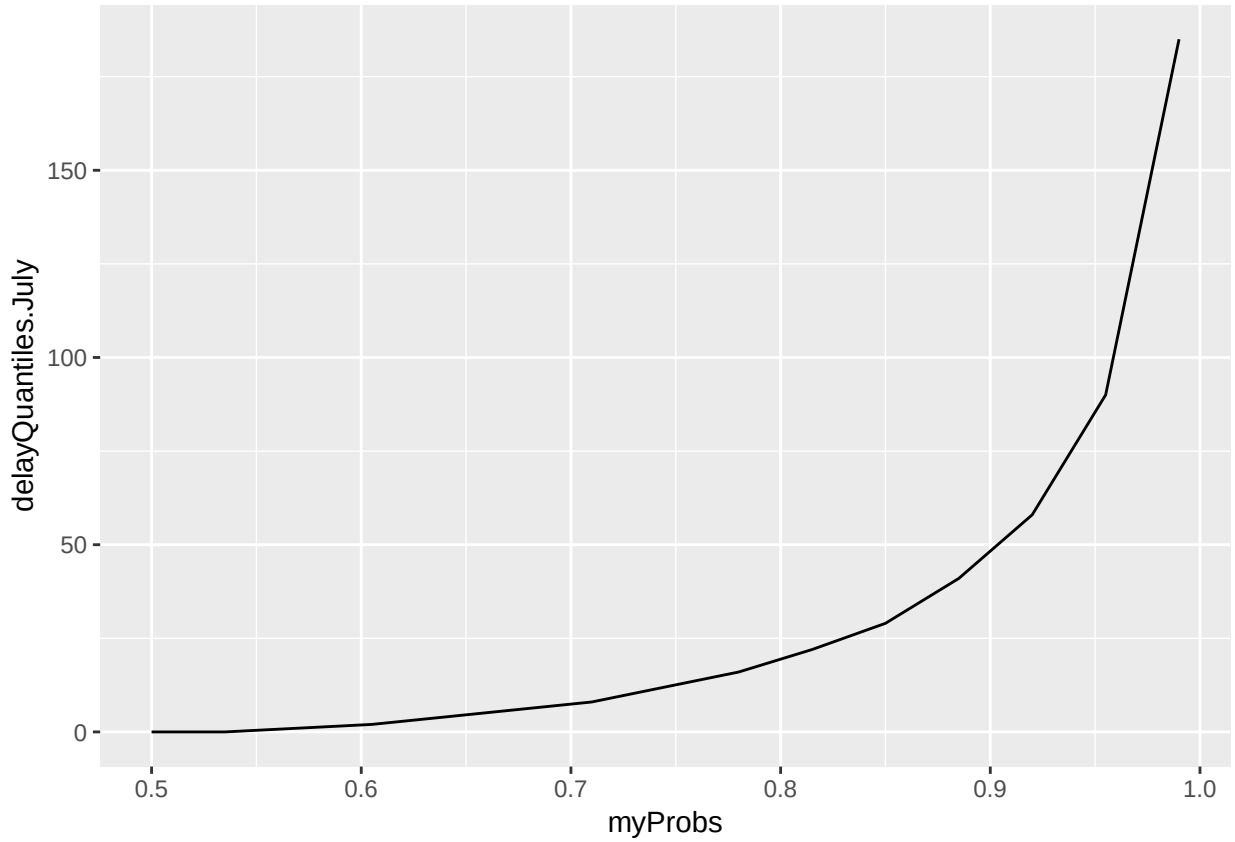
A traveler is planning a trip for July 2009 and wonders about the amount of departure delay they might encounter. They have the data from 2007 and 2008 and want to look at delays that are at least of median length. Obtain the 50th through 99th percentiles for July data in those years and interpret what the results tell us about magnitudes and frequency of delayed departures in July during those two years.

```
#Attach the same big matrix to flight0708 using the descriptor file without creating any new large matr
flight0708 <- attach.big.matrix("air0708.desc")

# machine minus one.
numParallelCores <- max(1, detectCores()-1)
# Create the parallel processes.
cl <- makeCluster(rep("localhost", numParallelCores),
                  type = "SOCK")
```

```
# Register the parallel processes with foreach.
registerDoSNOW(cl)

# Create a variable to hold the quantile probabilities.
myProbs <- seq(0.5, 0.99, 0.035)

month.index <- split(1:nrow(flight0708), flight0708[,"Month"])

delayQuantiles.July <- foreach( month = month.index[7], .combine=cbind) %do% {
  quantile(flight0708[month, "DepDelay"], myProbs,
           na.rm = TRUE)
}
#colnames(delayQuantiles.July) <- c("percentile", "delay")
stopCluster(cl)
delayQuantiles.July
```

```
##   50% 53.5%   57% 60.5%   64% 67.5%   71% 74.5%   78% 81.5%   85% 88.5%
##     0     0     1     2     4     6     8    12    16    22    29    41
##   92% 95.5%   99%
##    58    90   185
```

```
delayQuantiles.July <- as.data.frame(delayQuantiles.July)
# See delay quantiles for the data.

library(ggplot2)

qplot(myProbs, delayQuantiles.July, data = delayQuantiles.July, geom = "line")
```

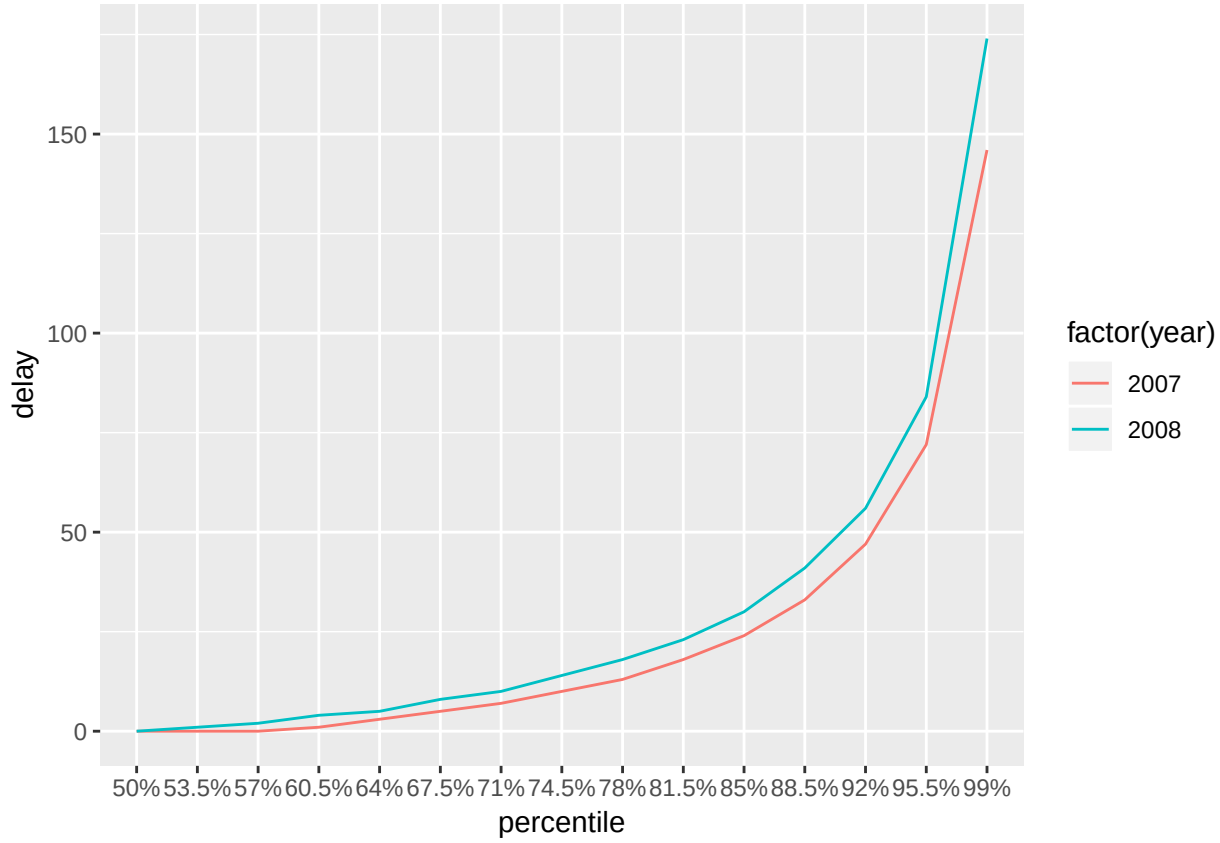


Question 2

The traveler is also curious about differences in departure delay percentiles for July during those two years. Compute and compare the 50th through 99th percentiles for July 2007 and July 2008. Provide an informative visualization along with interpretation of similarities and differences in the delay quantiles.

```
# machine minus one.
numParallelCores <- max(1, detectCores()-1)
# Create the parallel processes.
cl <- makeCluster(rep("localhost", numParallelCores),
                  type = "SOCK")
# Register the parallel processes with foreach.
registerDoSNOW(cl)

# Create a variable to hold the quantile probabilities.
myProbs <- seq(0.5, 0.99, 0.035)

Month07 <- flight0708[which(flight0708[,"Month"] == '7'), ]
year.index <- split(1:nrow(Month07), Month07[,"Year"])

delayQuantiles.July <- foreach( year = year.index, .combine=cbind) %do% {
  quantile(flight0708[year, "DepDelay"], myProbs,
           na.rm = TRUE)
}
# Clean up the column names.
```

```
colnames(delayQuantiles.July) <- names(year.index)

#colnames(delayQuantiles.July) <- c("percentile", "delay")
stopCluster(cl)

dq <- melt(delayQuantiles.July)
names(dq) <- c("percentile", "year", "delay")
ggplot(data=dq,
       aes(x=percentile, y=delay, group=factor(year), color=factor(year))) +
       geom_line()
```



Question 3

Consider month and day of week as continuous linear predictors for departure delay. Obtain a linear regression model for departure delay as a function of month and day of week using the 2007-2008 data. Interpret what the model suggests about the relationship between delay time and the day of week and month. Comment on the usefulness of the model and any issues with using this model.

```
delay.model01 <- lm(flight0708[, "DepDelay"] ~ flight0708[, "Month"] + flight0708[, "DayOfWeek"])
summary(delay.model01)
```

```
##
## Call:
## lm(formula = flight0708[, "DepDelay"] ~ flight0708[, "Month"] +
##     flight0708[, "DayOfWeek"])
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -543.67  -14.85  -11.38   -1.33 2591.15
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               11.203008   0.027608  405.79   <2e-16 ***
## flight0708[, "Month"]     -0.191397   0.002785  -68.73   <2e-16 ***
## flight0708[, "DayOfWeek"]  0.188386   0.004769   39.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.74 on 14165946 degrees of freedom
##   (296994 observations deleted due to missingness)
## Multiple R-squared:  0.0004415,  Adjusted R-squared:  0.0004414
## F-statistic:  3129 on 2 and 14165946 DF,  p-value: < 2.2e-16
```

Question 4

Rather than a straight linear trend, it is suggested that delays might be much worse in winter and not as bad in summer. Likewise, it is suggested that delays might get increasingly worse as the week goes on.

```
delay.model02 <- lm(flight0708[, "DepDelay"] ~ I((flight0708[, "Month"] - 6)^2) + I(flight0708[, "DayOf
summary(delay.model02)
```

```
##
## Call:
## lm(formula = flight0708[, "DepDelay"] ~ I((flight0708[, "Month"] -
##     6)^2) + I(flight0708[, "DayOfWeek"]^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -544.19  -14.97  -11.26   -1.25 2590.41
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     9.8810949  0.0180308  548.01   <2e-16 ***
## I((flight0708[, "Month"] - 6)^2) 0.0312587  0.0008612   36.30   <2e-16 ***
## I(flight0708[, "DayOfWeek"]^2)   0.0234729  0.0005849   40.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.74 on 14165946 degrees of freedom
##   (296994 observations deleted due to missingness)
## Multiple R-squared:  0.0002049,  Adjusted R-squared:  0.0002048
## F-statistic:  1452 on 2 and 14165946 DF,  p-value: < 2.2e-16
```