Flight Delay Analysis (Unsupervised Learning)

Joseph O'Malley

15 June, 2019

Contents

Data Overview

These data are from the U.S. Department of Transportation (retrieved from Kaggle.com) and tracks the on-time performance of domestic flights operated by large carriers. It includes 5.8 million flight records throughout the calendar year of 2015 (the MCI subset contains 77,320 flights). The set of 31 feature columns includes: origin, destination, date information, scheduled depart time (of day), wheels up/down time, taxi time, cancellation/diverted (y/n), cancellation reason, type of delay (weather, late aircraft, etc). There are two additional columns I plan to add that include latitude and longitude of departure city, bringing the total columns to 33.

```
# The dataset of employee satisfaction
flights_df <- read.csv(flight_data, sep = ",", header = T)
lat_lon_df <- read.csv(airport_detail, sep = ",", header = T)

# Check basic info
dim(flights_df)
```

```
## [1] 5819079      31
```

Problem description:

In building this model, I look to explore associations that lead to longer delays. I expect time of year and time of day to have effects on delays, especially when combined with weather related factors like origin/destination airports and latitude/longitude. Additionally, I expect airports to have strong effects on taxi times and departure delays. My models will aim to look at the relationships between some of these interconnected feature columns and reduce these features into factors that capture their combined meaning.

The models I plan to use will give decision makers at Southwest insight into what conditions (or combination of conditions) are more prone to delays. This information can help them to schedule flights more

strategically to avoid delays that kill customer loyalty and top line revenue. It could potentially inform certain departure/arrival cities to target/avoid, flight time of day, flight time of year, regions to avoid, and an combination of these (and other) feature columns. Lastly, it could show competitor airlines that are performing better/worse in certain conditions and allow management to analyze their processes to see what makes them more efficient - potentially mimicking parts of their processes in those conditions.

Analysis Methods:

There are a number of methods that I have had exposure to available to use for this analysis. The initial starting point for my analysis is using correspondence analysis, which looks at combinations of crosstabs to analyze similarity (of airlines, in my case). The only disadvantage is that it requires a specific aggregated data format, but allow me to look at specific areas of interest. The next step to explore between delays is Association Rules. This method also allows for tight control over what relationships to explore and puts it in a quantified odds ratio for benchmarking, but it does have the drawback of not being able to handle continuous datatypes (requiring "binning" – as does correspondence analysis).

Cluster analysis comes in several forms and groups similar data across the entire set of columns. These clusters may be tuned in several ways and assigned to a given observation. I could then look at common characteristics of these observations and compare them to the bench marks. Certain flights could receive review based on being put into a more "high risk" cluster. However, without specific knowledge of what made a particular flight high risk, I prefer a more quantifiable feature (or combination).

Using Factor Analysis, I will be able to assign a set number of "factors" that reduce the common variance by giving different weightings to the full set. This reduces dimensionality and will allow me to interpret the underlying meaning. Additionally, when combined with a simple linear regression it can show how each of these "factors" interacts with the dependent variable ("Departure delay").

Of the methods discussed, I plan to (using R) do:

EDA -> Correspondence Analysis -> Association Rules -> Factor Analysis -> Regression

Exploratory Data Analysis

```
## reduce dataset to only flights in and out of Kansas City
flights_df2 <- flights_df[flights_df$ORIGIN_AIRPORT == "MCI" |
    flights_df$DESTINATION_AIRPORT == "MCI", ]

## Join latitude/longitude info for both
flights_df2 <- merge(x = flights_df2, y = lat_lon_df, by.x = "ORIGIN_AIRPORT",
    by.y = "IATA_CODE", all.x = TRUE)
flights_df2 <- merge(x = flights_df2, y = lat_lon_df, by.x = "DESTINATION_AIRPORT",
    by.y = "IATA_CODE", all.x = TRUE)
```

```
# Check basic info of combined dataset
head(flights_df2)
dim(flights_df2)
summary(flights_df2)
str(flights_df2)
```

```
names(flights_df2)
```

```
##  [1] "DESTINATION_AIRPORT" "ORIGIN_AIRPORT"      "YEAR"
##  [4] "MONTH"               "DAY"                 "DAY_OF_WEEK"
##  [7] "AIRLINE"             "FLIGHT_NUMBER"       "TAIL_NUMBER"
## [10] "SCHEDULED_DEPARTURE" "DEPARTURE_TIME"      "DEPARTURE_DELAY"
## [13] "TAXI_OUT"            "WHEELS_OFF"          "SCHEDULED_TIME"
## [16] "ELAPSED_TIME"        "AIR_TIME"            "DISTANCE"
## [19] "WHEELS_ON"           "TAXI_IN"             "SCHEDULED_ARRIVAL"
## [22] "ARRIVAL_TIME"        "ARRIVAL_DELAY"       "DIVERTED"
## [25] "CANCELLED"           "CANCELLATION_REASON" "AIR_SYSTEM_DELAY"
## [28] "SECURITY_DELAY"      "AIRLINE_DELAY"       "LATE_AIRCRAFT_DELAY"
## [31] "WEATHER_DELAY"       "AIRPORT.x"           "CITY.x"
## [34] "STATE.x"             "COUNTRY.x"           "LATITUDE.x"
## [37] "LONGITUDE.x"         "AIRPORT.y"           "CITY.y"
## [40] "STATE.y"             "COUNTRY.y"           "LATITUDE.y"
## [43] "LONGITUDE.y"
```

```
## drop unnecessary columns
flights_df2 <- flights_df2[, -which(names(flights_df2) %in%
    c("AIRPORT.x", "AIRPORT.y", "COUNTRY.x", "COUNTRY.y",
        "YEAR"))]

dim(flights_df2)
```

```
## [1] 77320    38
```

subset to departure MCI flights

```
## origin = MCI
flights_MCI_origin_df <- flights_df2[flights_df2$ORIGIN_AIRPORT ==
    "MCI", ]
flights_MCI_origin_df <- flights_MCI_origin_df[, -which(names(flights_MCI_origin_df) %in%
    c("CITY.x", "STATE.x", "LATITUDE.x", "LONGITUDE.x"))]

## desitination = MCI
flights_MCI_departure_df <- flights_df2[flights_df2$DESTINATION_AIRPORT ==
    "MCI", ]
flights_MCI_departure_df <- flights_MCI_departure_df[,
    -which(names(flights_MCI_departure_df) %in% c("CITY.y",
        "STATE.y", "LATITUDE.y", "LONGITUDE.y"))]

dim(flights_MCI_origin_df)
```

```
## [1] 38665    34
```

```
dim(flights_MCI_departure_df)
```

```
## [1] 38655    34
```

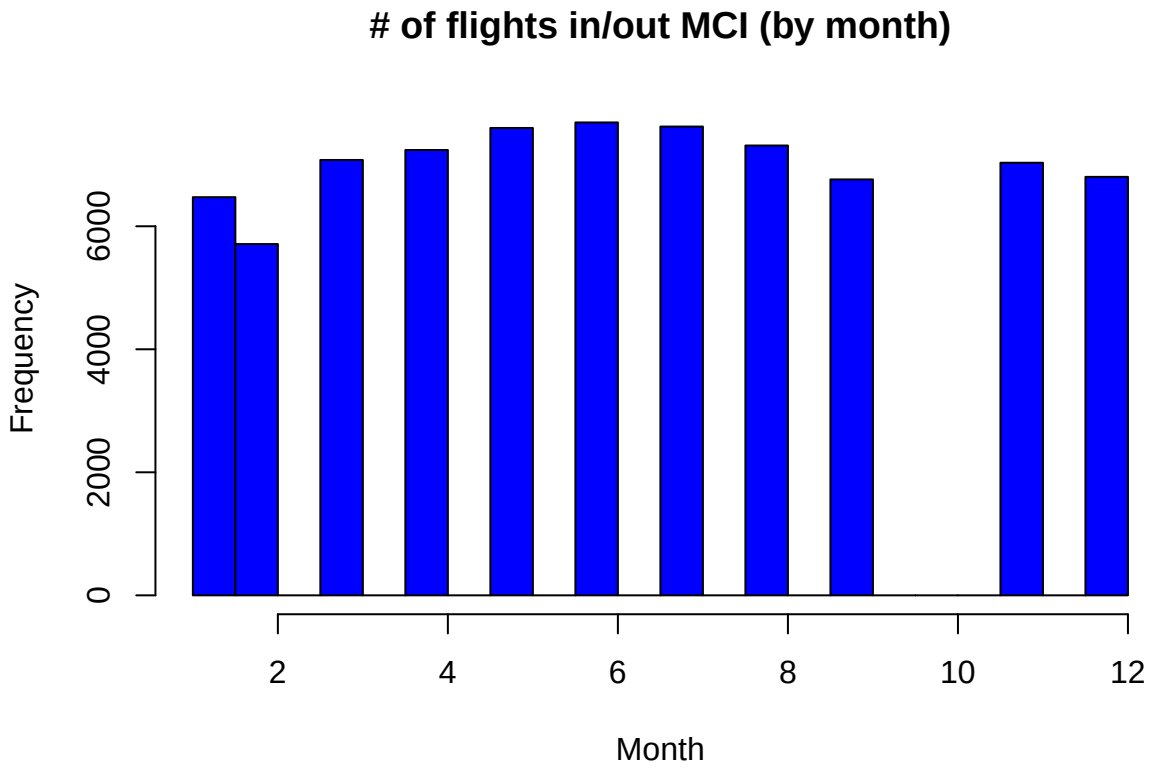
```
summary(flights_MCI_origin_df)
summary(flights_MCI_departure_df)
```

subset to southwest flights

```
southwest_flights <- flights_df2[which(flights_df2$AIRLINE ==
    "WN"), ]
```

SW flights by month

```
prop.table(table(flights_df2$MONTH))
hist(flights_df2$MONTH, col = "blue", main = "# of flights in/out MCI (by month)",
    xlab = "Month")
```



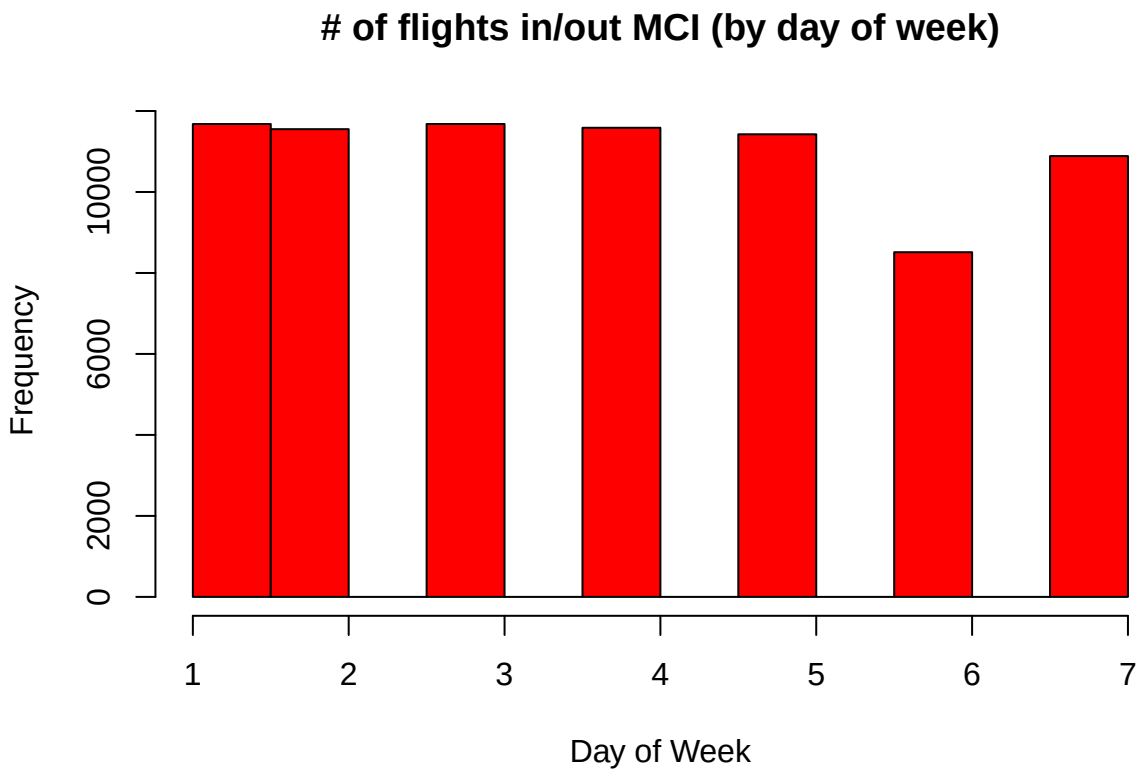

```
## no 10th month, with the highest travel month of June -
## 9.94%, lowest in February 7.39%
prop.table(table(southwest_flights$MONTH))
## Southwest had the most flights in July 10.3%, lowest in
## February 7.4%
```

SW flights by Day of week

```
prop.table(table(flights_df2$DAY_OF_WEEK))
```

```
##
##         1         2         3         4         5         6         7
## 0.1510347 0.1494051 0.1510476 0.1498448 0.1477625 0.1101009 0.1408044
```

```r
hist(flights_df2$DAY_OF_WEEK, col = "red", main = "# of flights in/out MCI (by day of week)",
    xlab = "Day of Week")
```




```r
prop.table(table(flights_df2$AIRLINE))
```

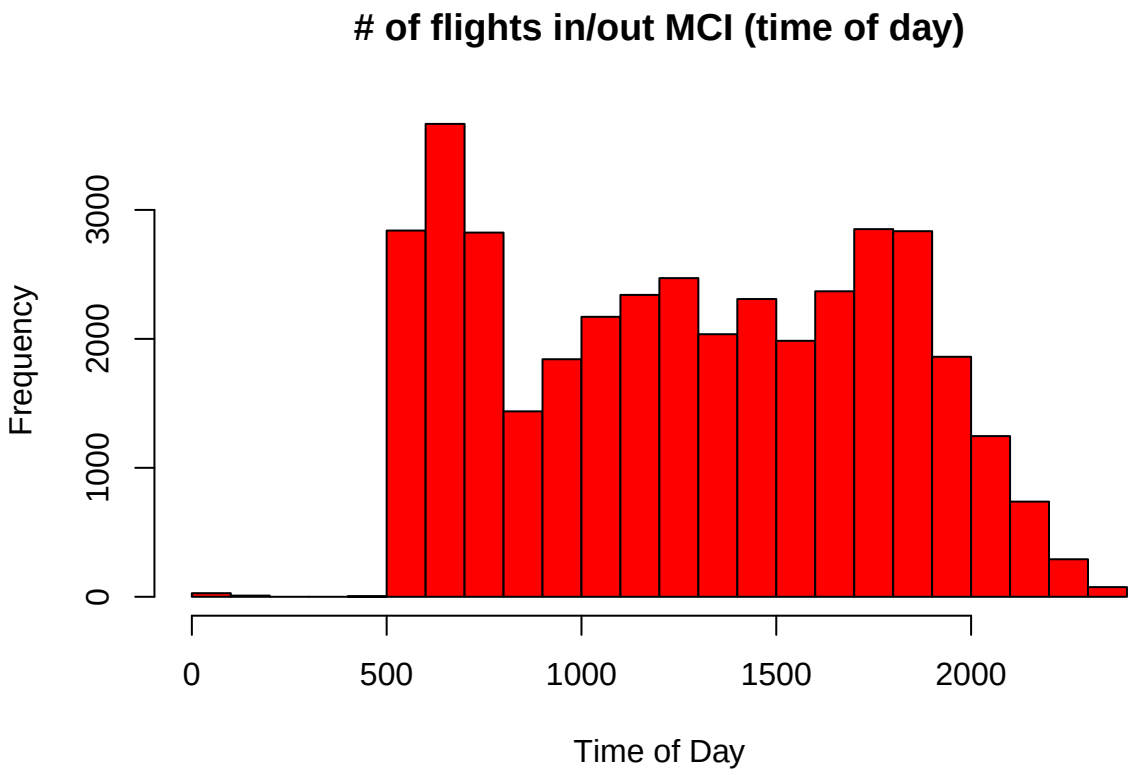
```
##
##          AA          AS          B6          DL          EV          F9
## 0.098448008 0.010243145 0.000000000 0.131414899 0.086717538 0.008988619
##          HA          MQ          NK          OO          UA          US
## 0.000000000 0.003052250 0.036704604 0.041308846 0.015494051 0.031143301
##          VX          WN
## 0.000000000 0.536484739
```

```r
summary(flights_MCI_origin_df$DISTANCE)
## distance ranged from 152-1499 miles, with a median of 643
southwest_origin_mci <- flights_MCI_origin_df[which(flights_MCI_origin_df$AIRLINE ==
    "WN"), ]
summary(southwest_origin_mci$DISTANCE)
## distance ranged from 237-1489, median of 666
```

bin continuous variables

```
flights_df2$DEPARTURE_TIME_binned <- cut(flights_df2$DEPARTURE_TIME,
    breaks = c(0, 9, 1300, 1800, 2400), labels = c("12p-9a",
        "9a-1p", "1-6p", "6-12p"), right = FALSE, ordered_result = TRUE)

## repeat process for all continuous variables based on
## distribution & logical breakoints
```

```
hist(flights_MCI_origin_df$DEPARTURE_TIME, col = "red", main = "# of flights in/out MCI (time of day)",
    xlab = "Time of Day")
```



check new binned columns (binned by quantile dist)

```
names(flights_df2)
```

```
##  [1] "DESTINATION_AIRPORT"        "ORIGIN_AIRPORT"
##  [3] "MONTH"                      "DAY"
##  [5] "DAY_OF_WEEK"                "AIRLINE"
##  [7] "FLIGHT_NUMBER"              "TAIL_NUMBER"
##  [9] "SCHEDULED_DEPARTURE"        "DEPARTURE_TIME"
## [11] "DEPARTURE_DELAY"            "TAXI_OUT"
## [13] "WHEELS_OFF"                 "SCHEDULED_TIME"
## [15] "ELAPSED_TIME"               "AIR_TIME"
```

```
## [17] "DISTANCE"                    "WHEELS_ON"
## [19] "TAXI_IN"                     "SCHEDULED_ARRIVAL"
## [21] "ARRIVAL_TIME"                "ARRIVAL_DELAY"
## [23] "DIVERTED"                    "CANCELLED"
## [25] "CANCELLATION_REASON"         "AIR_SYSTEM_DELAY"
## [27] "SECURITY_DELAY"              "AIRLINE_DELAY"
## [29] "LATE_AIRCRAFT_DELAY"         "WEATHER_DELAY"
## [31] "CITY.x"                      "STATE.x"
## [33] "LATITUDE.x"                  "LONGITUDE.x"
## [35] "CITY.y"                      "STATE.y"
## [37] "LATITUDE.y"                  "LONGITUDE.y"
## [39] "DEPARTURE_TIME_binned"       "SCHEDULED_DEPARTURE_binned"
## [41] "DEPARTURE_DELAY_binned"      "TAXI_OUT_binned"
## [43] "WHEELS_OFF_bin"              "ELAPSED_TIME_bin"
## [45] "SCHEDULED_TIME_bin"          "AIR_TIME_bin"
## [47] "WHEELS_ON_bin"               "TAXI_IN_binned"
## [49] "SCHEDULED_ARRIVAL_binned"    "ARRIVAL_TIME_binned"
## [51] "ARRIVAL_DELAY_binned"        "AIR_SYSTEM_DELAY_binned"
## [53] "SECURITY_DELAY_binned"       "AIRLINE_DELAY_binned"
## [55] "LATE_AIRCRAFT_DELAY_binned" "WEATHER_DELAY_binned"
## [57] "LATITUDE.x_binned"           "LATITUDE.y_binned"
## [59] "LONGITUDE.x_binned"          "LONGITUDE.y_binned"
```

EDA findings:

I started by doing some exploratory data analysis. I started by looking by looking to see how time of year affected flight delays (Figure 1.2) and found there was no 10th month, with the highest travel month of June - 9.94%, lowest in February 7.39%. Then looked at Southwest specifically seeing it had the most flights in July 10.3%, lowest in February 7.4%. I then moved to days of the week (Figure 1.2), with days ranging their days ranging form 14-15.1%, lowest on Saturday 11.0%. Distance ranged from 152-1499 miles, with a median of 643, while Southwest distance ranged from 237-1489. median of 666. I concluded my EDA by seeing what the competitive landscape was in Kansas City and found 53.6% of flights were on southwest, delta second with 13.1% - 3 major carriers have none (Jetblue, Hawaiian, Virgin).

Correspondence Analysis

subset to Southwest airlines flights

```
## install.packages('descr', 'FactoMineR', 'factoextra')

library("FactoMineR")
library("descr")
library("factoextra")

## analyze Southwest flights
southwest_flights <- flights_df2[which(flights_df2$AIRLINE ==
    "WN"), ]
summary(southwest_flights)

southwest_flights_ca <- southwest_flights[, which(names(southwest_flights) %in%
    c("AIRLINE", "MONTH", "DAY_OF_WEEK", "SCHEDULED_ARRIVAL_binned",
        "DEPARTURE_TIME_binned", "SCHEDULED_DEPARTURE_binned",
```

```
        "DEPARTURE_DELAY_binned", "TAXI_OUT_binned", "WHEELS_OFF_bin",
        "ELAPSED_TIME_bin", "SCHEDULED_TIME_bin", "AIR_TIME_bin",
        "WHEELS_ON_bin", "TAXI_IN_binned", "ARRIVAL_TIME_binned",
        "ARRIVAL_DELAY_binned", "AIR_SYSTEM_DELAY_binned", "SECURITY_DELAY_binned",
        "AIRLINE_DELAY_binned", "LATE_AIRCRAFT_DELAY_binned",
        "WEATHER_DELAY_binned", "LATITUDE.x_binned", "LATITUDE.y_binned",
        "LONGITUDE.x_binned", "LONGITUDE.y_binned"))]

summary(southwest_flights_ca)
```

transform data for conjoint analysis

```
southwest_flights_ca1 <- transform(southwest_flights_ca, freq.neib = ave(seq(nrow(southwest_flights_ca)),
    DEPARTURE_DELAY_binned, FUN = length))
mean(southwest_flights_ca1$freq.neib)
```

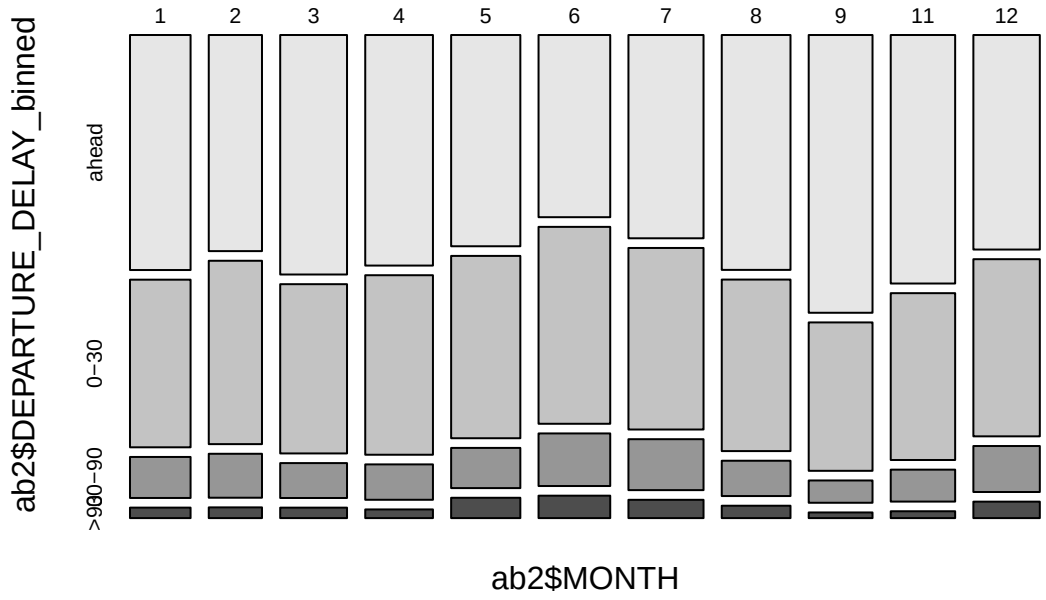
```
## [1] 16682.88
```

```
## eliminate low frequency items
ab2 <- southwest_flights_ca1[southwest_flights_ca1$freq.neib >
    100, ]
names(ab2)
```

```
##  [1] "MONTH"                      "DAY_OF_WEEK"
##  [3] "AIRLINE"                    "DEPARTURE_TIME_binned"
##  [5] "SCHEDULED_DEPARTURE_binned" "DEPARTURE_DELAY_binned"
##  [7] "TAXI_OUT_binned"            "WHEELS_OFF_bin"
##  [9] "ELAPSED_TIME_bin"           "SCHEDULED_TIME_bin"
## [11] "AIR_TIME_bin"               "WHEELS_ON_bin"
## [13] "TAXI_IN_binned"             "SCHEDULED_ARRIVAL_binned"
## [15] "ARRIVAL_TIME_binned"        "ARRIVAL_DELAY_binned"
## [17] "AIR_SYSTEM_DELAY_binned"    "SECURITY_DELAY_binned"
## [19] "AIRLINE_DELAY_binned"       "LATE_AIRCRAFT_DELAY_binned"
## [21] "WEATHER_DELAY_binned"       "LATITUDE.x_binned"
## [23] "LATITUDE.y_binned"          "LONGITUDE.x_binned"
## [25] "LONGITUDE.y_binned"         "freq.neib"
```

flight delays by month

```
## crosstab data
southwest_flights_ca1$delay_by_month <- paste(ab2$DEPARTURE_DELAY_binned,
    ab2$MONTH, sep = "")

# head(ab)
ab3 <- crosstab(ab2$DEPARTURE_DELAY_binned, ab2$MONTH)
```



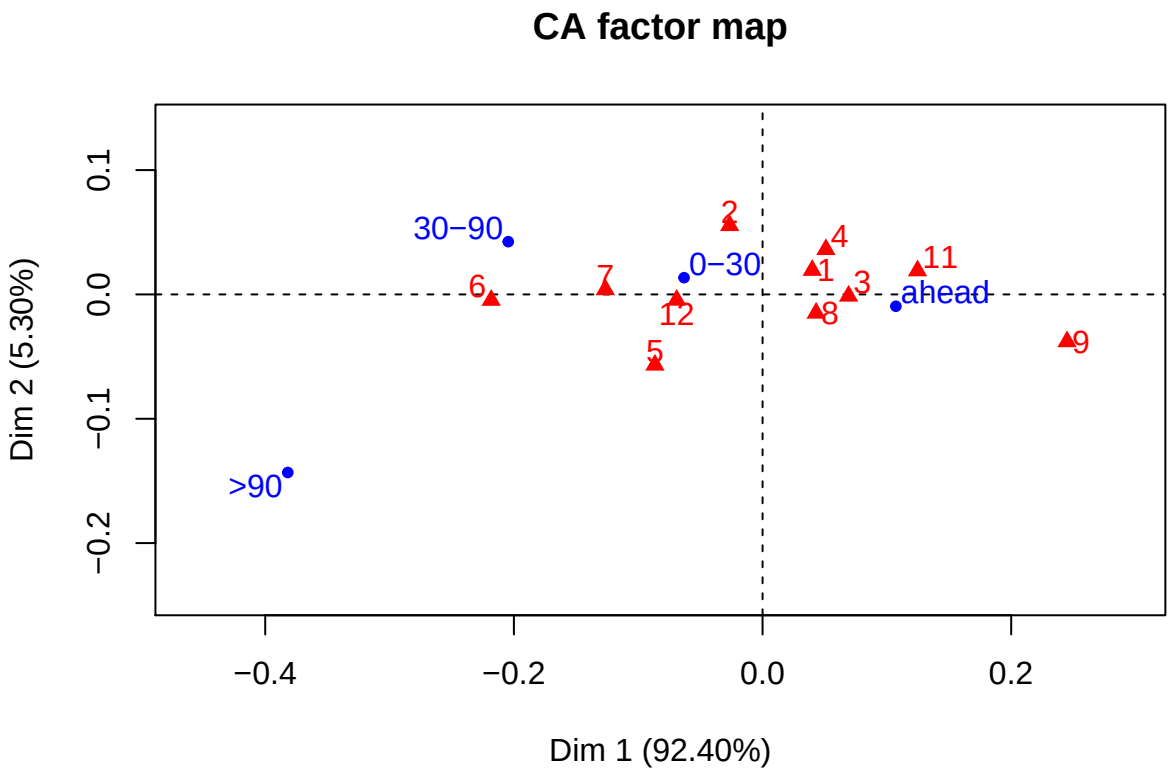

```
## transform data to crosstab table
ab3$tab
```

```
##                            ab2$MONTH
## ab2$DEPARTURE_DELAY_binned    1    2    3    4    5    6    7    8    9
##                      ahead 1771 1431 1991 1942 1818 1625 1899 2009 2205
##                      0-30  1263 1214 1407 1512 1569 1756 1696 1467 1178
##                      30-90  309  291  290  298  346  469  476  302  177
##                      >90     79   71   87   73  175  200  171  106   45
##                            ab2$MONTH
## ab2$DEPARTURE_DELAY_binned   11   12
##                      ahead 1992 1773
##                      0-30  1337 1464
##                      30-90  256  380
##                      >90     55  136
```

```
## test for statistical significance
chisq.test(ab3$tab)
```

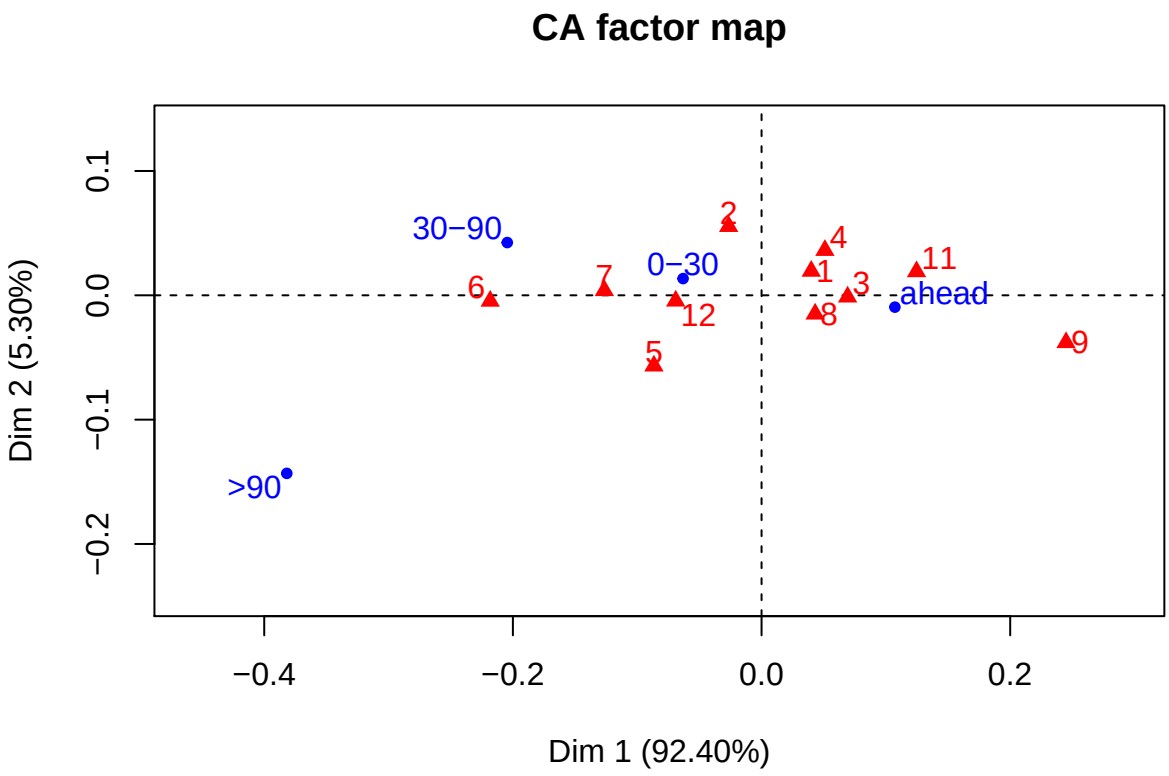
```
##
##  Pearson's Chi-squared test
##
## data:  ab3$tab
## X-squared = 674.8, df = 30, p-value < 2.2e-16
```

```
####### correspondence analysis The general format of CA
CA(ab3$tab, ncp = 5, graph = TRUE)
```




```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 11 categories
## The chi square of independence between the two variables is equal to 674.7953 (p-value =  8.74975e-1
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

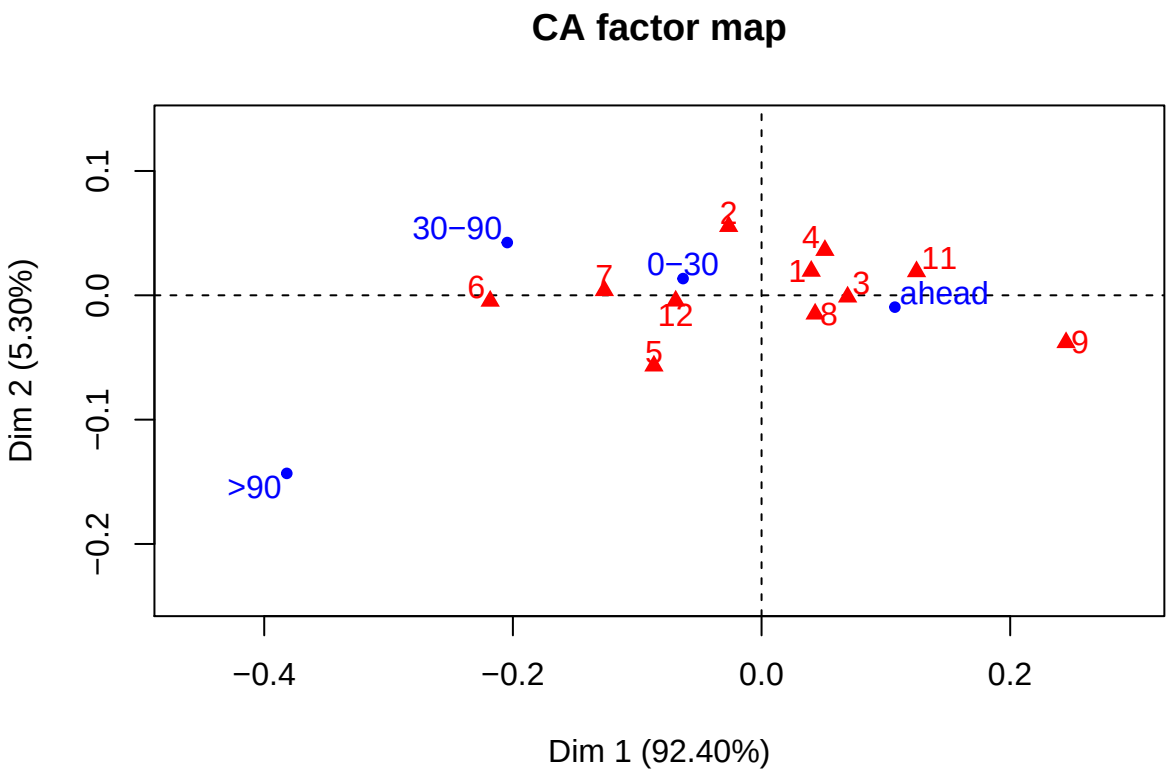
```
hou.ca <- CA(ab3$tab, graph = TRUE)
```



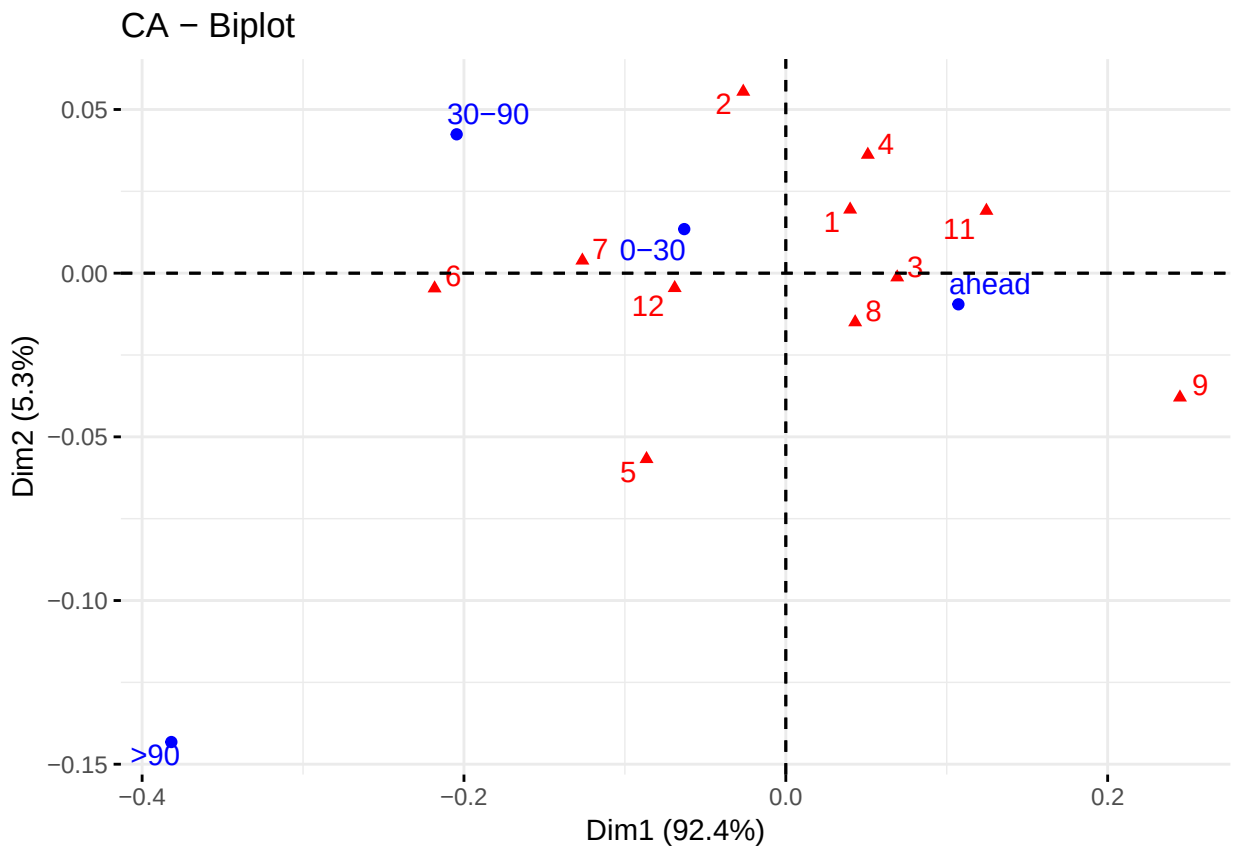

```
print(hou.ca)
```

```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 11 categories
## The chi square of independence between the two variables is equal to 674.7953 (p-value =  8.74975e-1
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

```
ab.ca = CA(ab3$tab, ncp = 5, graph = TRUE)
```




```
fviz_ca_biplot(ab.ca, repel = TRUE)
```

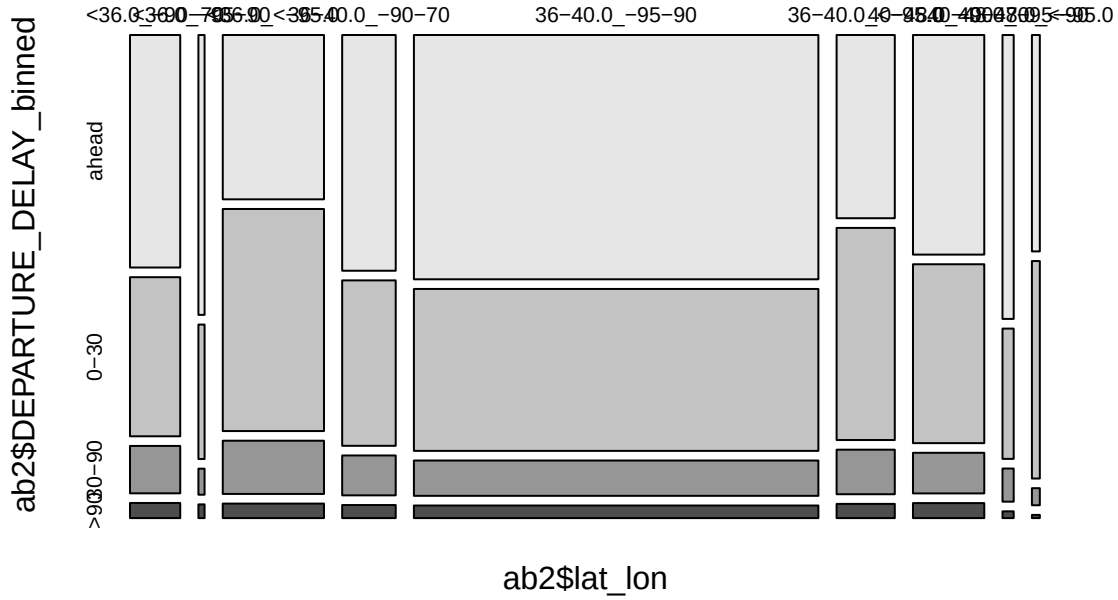


flights by region

```
southwest_flights_ca2 <- transform(southwest_flights_ca, freq.neib = ave(seq(nrow(southwest_flights_ca)),
    DEPARTURE_DELAY_binned, FUN = length))
mean(southwest_flights_ca2$freq.neib)
```

```
## [1] 16682.88
```

```
## eliminate low frequency items
ab2 <- southwest_flights_ca2[southwest_flights_ca2$freq.neib >
    100, ]
# names(ab2)

## crosstab data
ab2$lat_lon <- paste(ab2$LATITUDE.x_binned, ab2$LONGITUDE.x_binned,
    sep = "_")

## test for statistical significance names(ab2)
ab3 <- crosstab(ab2$DEPARTURE_DELAY_binned, ab2$lat_lon)
```




```
## transform data to crosstab table
ab3$tab
```

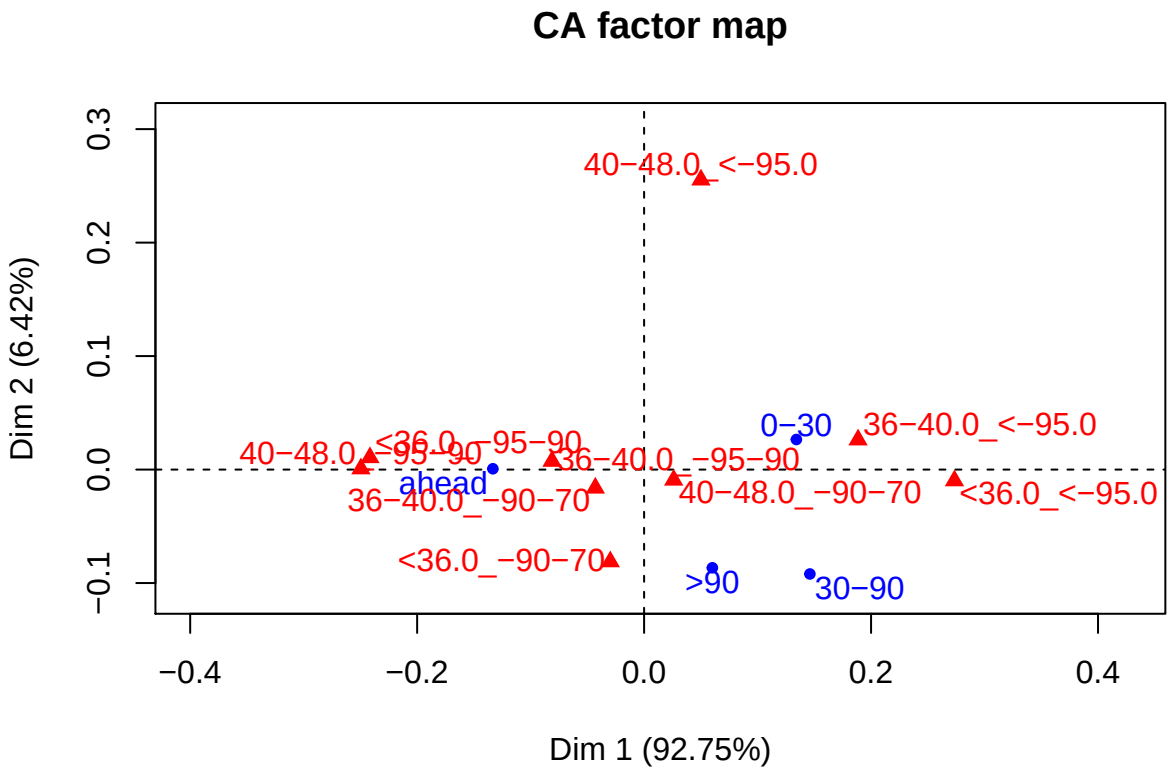
```
##                            ab2$lat_lon
## ab2$DEPARTURE_DELAY_binned <36.0_-90-70 <36.0_-95-90 <36.0_<-95.0
##                       ahead         1384          204         1970
##                       0-30           947           98         2662
##                       30-90          282           19          638
##                       >90             90           10          174
##                            ab2$lat_lon
## ab2$DEPARTURE_DELAY_binned 36-40.0_-90-70 36-40.0_-95-90 36-40.0_<-95.0
##                       ahead           1495          11705           1263
##                       0-30            1049           7757           1462
##                       30-90            253           1692            307
##                       >90               83            604             98
##                            ab2$lat_lon
## ab2$DEPARTURE_DELAY_binned 40-48.0_-90-70 40-48.0_-95-90 40-48.0_<-95.0
##                       ahead           1855            377            203
##                       0-30            1511            173            204
##                       30-90            343             44             16
##                       >90              127              9              3
```

```
chisq.test(ab3$tab)
```

```
##
```

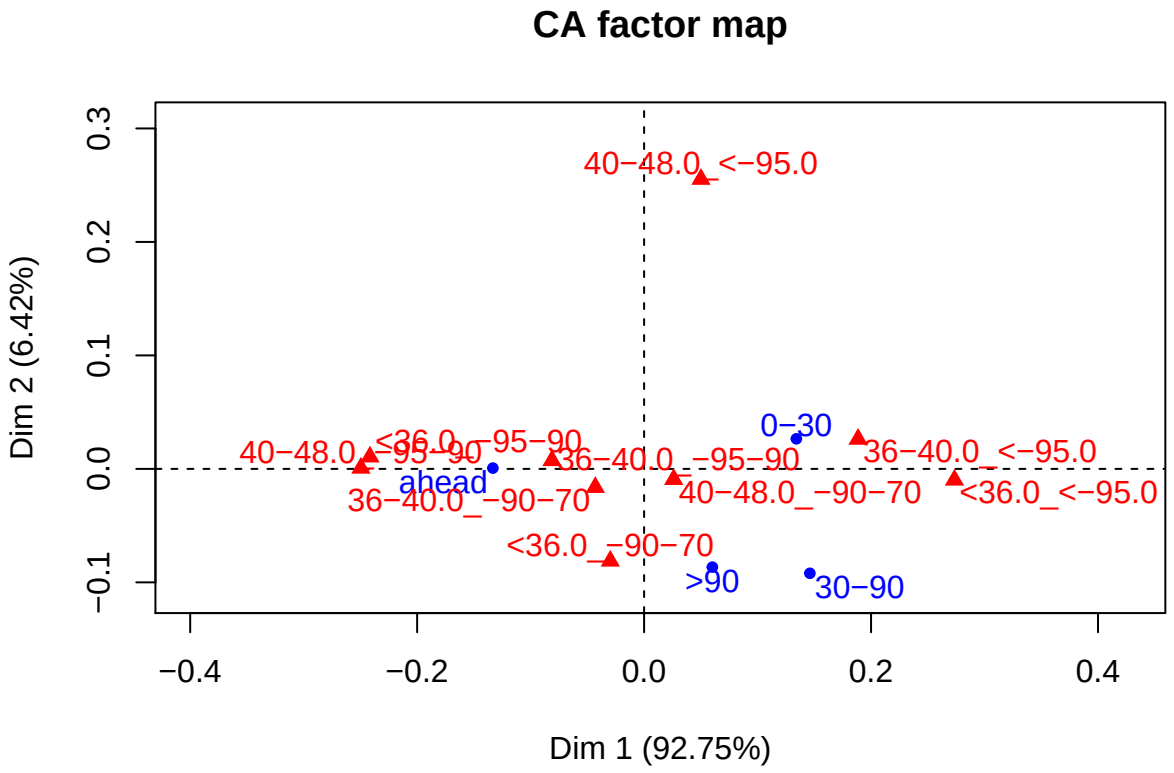
```
##  Pearson's Chi-squared test
##
## data:  ab3$tab
## X-squared = 787.06, df = 24, p-value < 2.2e-16
```

```
####### correspondence analysis The general format of CA
CA(ab3$tab, ncp = 5, graph = TRUE)
```




```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 9 categories
## The chi square of independence between the two variables is equal to 787.0579 (p-value =  1.117925e-
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

```r
# X : a data frame (contingency table) ncp : number of
# dimensions kept in the final results. graph : a logical
# value. If TRUE a graph is displayed.

hou.ca <- CA(ab3$tab, graph = TRUE)
```

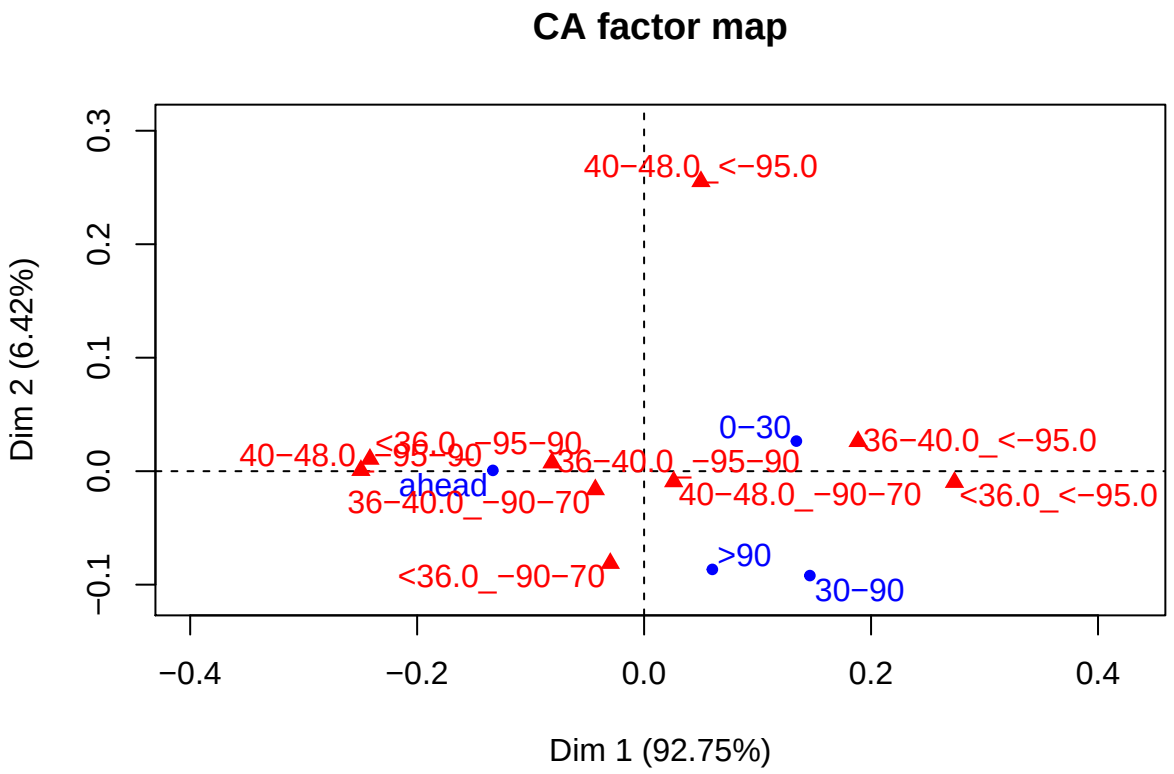



```r
print(hou.ca)
```

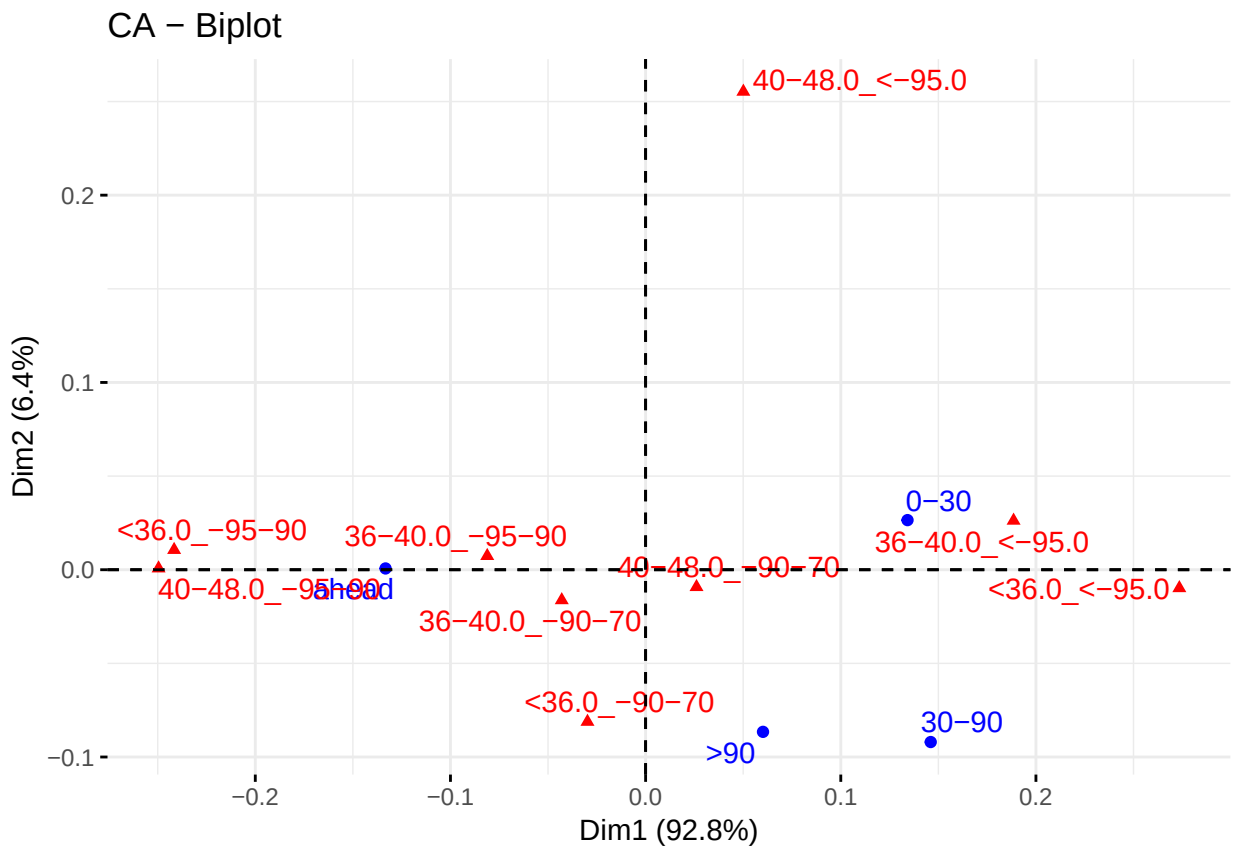
```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 9 categories
## The chi square of independence between the two variables is equal to 787.0579 (p-value =  1.117925e-
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
```

```
## 12 "$call$marge.row" "weights of the rows"
```

```
ab.ca = CA(ab3$tab, ncp = 5, graph = TRUE)
```




```
fviz_ca_biplot(ab.ca, repel = TRUE)
```



delays by airline

```
## susbset to top 4 airlines out of MCI
top_4_air_df <- flights_df2[flights_df2$AIRLINE == "DL" | flights_df2$AIRLINE ==
    "WN" | flights_df2$AIRLINE == "AA" | flights_df2$AIRLINE ==
    "EV", ]

## analyze Southwest flights
all_flights_ca <- top_4_air_df[, which(names(top_4_air_df) %in%
    c("AIRLINE", "MONTH", "DAY_OF_WEEK", "SCHEDULED_ARRIVAL_binned",
        "DEPARTURE_TIME_binned", "SCHEDULED_DEPARTURE_binned",
        "DEPARTURE_DELAY_binned", "TAXI_OUT_binned", "WHEELS_OFF_bin",
        "ELAPSED_TIME_bin", "SCHEDULED_TIME_bin", "AIR_TIME_bin",
        "WHEELS_ON_bin", "TAXI_IN_binned", "ARRIVAL_TIME_binned",
        "ARRIVAL_DELAY_binned", "AIR_SYSTEM_DELAY_binned", "SECURITY_DELAY_binned",
        "AIRLINE_DELAY_binned", "LATE_AIRCRAFT_DELAY_binned",
        "WEATHER_DELAY_binned", "LATITUDE.x_binned", "LATITUDE.y_binned",
        "LONGITUDE.x_binned", "LONGITUDE.y_binned"))]

all_flights_ca2 <- transform(all_flights_ca, freq.neib = ave(seq(nrow(all_flights_ca)),
    DEPARTURE_DELAY_binned, FUN = length))
mean(all_flights_ca2$freq.neib)
```

```
## [1] 27861.11
```

```
## eliminate low frequency items
ab2 <- all_flights_ca2[all_flights_ca2$freq.neib > 500, ]
# names(ab2)

## crosstab data
ab2$lat_lon <- paste(ab2$LATITUDE.x_binned, ab2$LONGITUDE.x_binned,
    sep = "_")

# names(ab2)
ab3 <- crosstab(ab2$DEPARTURE_DELAY_binned, ab2$AIRLINE)
```


```
## transform data to crosstab table
ab3$tab
```

```
##                            ab2$AIRLINE
## ab2$DEPARTURE_DELAY_binned    AA    DL    EV    WN
##                      ahead  4694  6500  4236 20456
##                      0-30   2024  2914  1475 15863
##                      30-90   549   517   514  3594
##                      >90     230   210   264  1198
```

```
## test for statistical significance
chisq.test(ab3$tab)
```

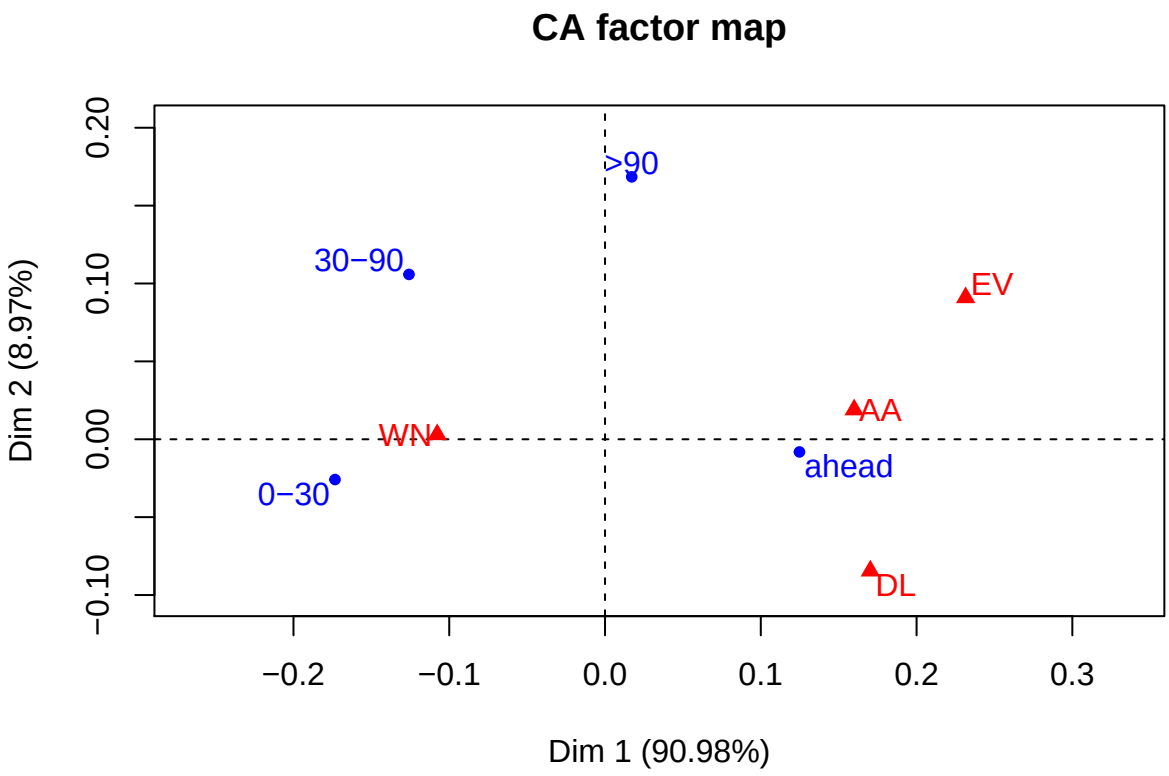
```
##
```

```
##  Pearson's Chi-squared test
##
## data:  ab3$tab
## X-squared = 1440.9, df = 9, p-value < 2.2e-16
```

```
####### correspondence analysis The general format of CA
CA(ab3$tab, ncp = 5, graph = TRUE)
```

```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 4 categories
## The chi square of independence between the two variables is equal to 1440.866 (p-value =  1.142858e-3
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

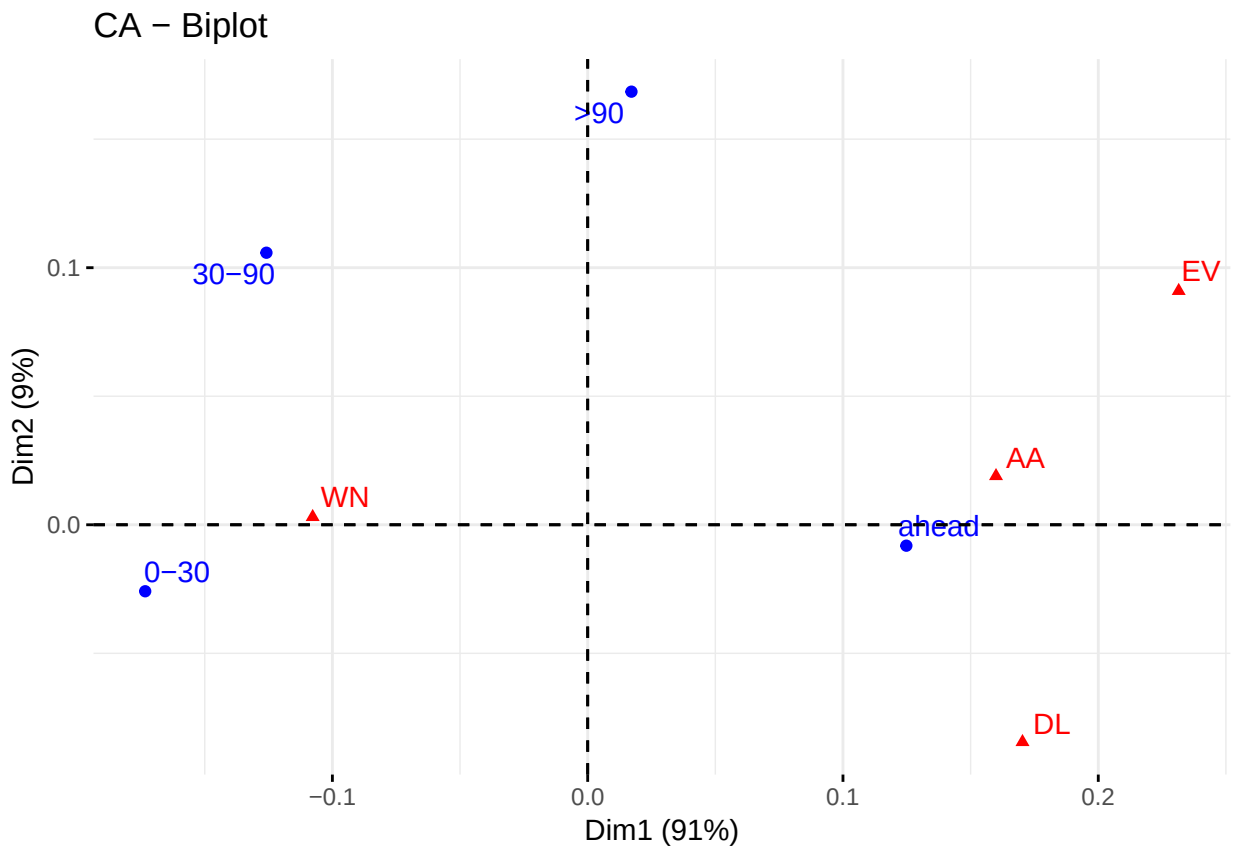
```
# X : a data frame (contingency table) ncp : number of
# dimensions kept in the final results. graph : a logical
# value. If TRUE a graph is displayed.

hou.ca <- CA(ab3$tab, graph = TRUE)
```




```
print(hou.ca)
```

```
## **Results of the Correspondence Analysis (CA)**
## The row variable has  4  categories; the column variable has 4 categories
## The chi square of independence between the two variables is equal to 1440.866 (p-value =  1.142858e-3
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$col"            "results for the columns"
## 3  "$col$coord"      "coord. for the columns"
## 4  "$col$cos2"       "cos2 for the columns"
## 5  "$col$contrib"    "contributions of the columns"
## 6  "$row"            "results for the rows"
## 7  "$row$coord"      "coord. for the rows"
## 8  "$row$cos2"       "cos2 for the rows"
## 9  "$row$contrib"    "contributions of the rows"
## 10 "$call"           "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"
```

```
ab.ca = CA(ab3$tab, ncp = 5, graph = TRUE)
fviz_ca_biplot(ab.ca, repel = TRUE)
```



correspondence analysis findings:

I then moved on to Correspondence Analysis, which allows us to analyze how different features interact and required binning of variables then subsetting (created bins based on quantile distribution and smoothed to round numbers when applicable). I first looked at departure_delay by month (Figure 2.1), by airline (Figure 2.2), region (Latitude/Longitude) (Figure 2.3 & 2.4). Top 4 airlines account for 84.8% of all flights (Figure 2.5). EV is skywest airlines, which serves as a regional connector for AA, Delta, United, and Alaskan airline. 2.9 percent of sw flights had "severe" delays. . . 90minuted> (Delta with just 2.1 percent), but when you consider skywest operates some of their connecting flights (with 4.1 percent extended delays), it may not be so bad. Additionally, looking at the chart show that lattitude/longitudes in west, specifically in the Southwest United States (ironically) –chart right – have more short-mid range delays (0-90 minutes) than average. I finished by running a significance testing for relationships between these binned variables, and all easily passed (Figure 2.6)

Association Rules

```
# install.packages('arules')
library(arules)
library(arulesViz)
library(data.table)

## analyze Southwest flights
southwest_flights <- flights_df2[which(flights_df2$AIRLINE ==
```

```
    "WN"), ]
## create uniqueID columns
southwest_flights$FLIGHT_ID <- seq.int(nrow(southwest_flights))

## subset flights for A-rules/market-basket analysis (mba)
southwest_flights_mba <- southwest_flights[, which(names(southwest_flights) %in%
    c("FLIGHT_ID", "MONTH", "DAY_OF_WEEK", "SCHEDULED_ARRIVAL_binned",
        "DEPARTURE_TIME_binned", "SCHEDULED_DEPARTURE_binned",
        "DEPARTURE_DELAY_binned", "TAXI_OUT_binned", "WHEELS_OFF_bin",
        "ELAPSED_TIME_bin", "SCHEDULED_TIME_bin", "AIR_TIME_bin",
        "WHEELS_ON_bin", "TAXI_IN_binned", "ARRIVAL_TIME_binned",
        "ARRIVAL_DELAY_binned", "AIR_SYSTEM_DELAY_binned", "SECURITY_DELAY_binned",
        "AIRLINE_DELAY_binned", "LATE_AIRCRAFT_DELAY_binned",
        "WEATHER_DELAY_binned", "LATITUDE.x_binned", "LATITUDE.y_binned",
        "LONGITUDE.x_binned", "LONGITUDE.y_binned", "DESTINATION_AIRPORT",
        "ORIGIN_AIRPORT"))]
```

```
# Check basic info
summary(southwest_flights_mba)
str(southwest_flights_mba)


### identify each unique flight
tid <- as.character(southwest_flights_mba[["FLIGHT_ID"]])
southwest_flights_mba <- southwest_flights_mba[, -which(names(southwest_flights_mba) %in%
    c("FLIGHT_ID"))]

### convert all column datatypes to factors
for (i in 1:ncol(southwest_flights_mba)) southwest_flights_mba[[i]] <- as.factor(southwest_flights_mba[

### convert dataframe to transaction set (for A-Rules)
trans <- as(southwest_flights_mba, "transactions")

### set transactionIDs
transactionInfo(trans)[["FLIGHT_ID"]] <- tid
```

inspect transactional data format

```
inspect(trans[1:2])
```

```
##     items                              transactionID FLIGHT_ID
## [1] {DESTINATION_AIRPORT=ABQ,
##      ORIGIN_AIRPORT=MCI,
##      MONTH=6,
##      DAY_OF_WEEK=2,
##      DEPARTURE_TIME_binned=1-6p,
##      SCHEDULED_DEPARTURE_binned=1-6p,
##      DEPARTURE_DELAY_binned=ahead,
##      TAXI_OUT_binned=10-15,
##      WHEELS_OFF_bin=1-6p,
##      ELAPSED_TIME_bin=90-120,
##      SCHEDULED_TIME_bin=120-180,
```

```
##      AIR_TIME_bin=90-120,
##      WHEELS_ON_bin=3-7p,
##      TAXI_IN_binned=<10,
##      SCHEDULED_ARRIVAL_binned=3-7p,
##      ARRIVAL_TIME_binned=3-7p,
##      ARRIVAL_DELAY_binned=ahead,
##      AIRLINE_DELAY_binned=no_delay,
##      LATITUDE.x_binned=36-40.0,
##      LATITUDE.y_binned=<36.0,
##      LONGITUDE.x_binned=-95-90,
##      LONGITUDE.y_binned=<-95.0}                    1         1
## [2] {DESTINATION_AIRPORT=ABQ,
##      ORIGIN_AIRPORT=MCI,
##      MONTH=4,
##      DAY_OF_WEEK=4,
##      DEPARTURE_TIME_binned=1-6p,
##      SCHEDULED_DEPARTURE_binned=1-6p,
##      DEPARTURE_DELAY_binned=ahead,
##      TAXI_OUT_binned=10-15,
##      WHEELS_OFF_bin=1-6p,
##      ELAPSED_TIME_bin=120-180,
##      SCHEDULED_TIME_bin=120-180,
##      AIR_TIME_bin=90-120,
##      WHEELS_ON_bin=3-7p,
##      TAXI_IN_binned=<10,
##      SCHEDULED_ARRIVAL_binned=3-7p,
##      ARRIVAL_TIME_binned=3-7p,
##      ARRIVAL_DELAY_binned=0-30,
##      AIRLINE_DELAY_binned=0-10,
##      LATITUDE.x_binned=36-40.0,
##      LATITUDE.y_binned=<36.0,
##      LONGITUDE.x_binned=-95-90,
##      LONGITUDE.y_binned=<-95.0}                    2         2
```

```
## crosstab data
seg.trans <- trans
summary(seg.trans)
```

```
## transactions as itemMatrix in sparse format with
##  41481 rows (elements/itemsets/transactions) and
##  1358 columns (items) and a density of 0.01665613
##
## most frequent items:
##            TAXI_IN_binned=<10     LATITUDE.y_binned=36-40.0
##                         37141                         28037
##     LATITUDE.x_binned=36-40.0    ARRIVAL_DELAY_binned=ahead
##                         28000                         24724
## AIRLINE_DELAY_binned=no_delay                       (Other)
##                         24724                        795634
##
## element (itemset/transaction) length distribution:
## sizes
##    11    13    15    18    21    22    24    25    26
##   370     7    20    76    13 33420     1    30  7544
```

```
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   22.00   22.00   22.62   22.00   26.00
##
## includes extended item information - examples:
##                      labels           variables levels
## 1 DESTINATION_AIRPORT=10135 DESTINATION_AIRPORT  10135
## 2 DESTINATION_AIRPORT=10136 DESTINATION_AIRPORT  10136
## 3 DESTINATION_AIRPORT=10140 DESTINATION_AIRPORT  10140
##
## includes extended transaction information - examples:
##   transactionID FLIGHT_ID
## 1             1         1
## 2             2         2
## 3             3         3
```

```
seg.rules <- apriori(seg.trans, parameter = list(support = 0.02,
    conf = 0.4, target = "rules", maxlen = 4))
```

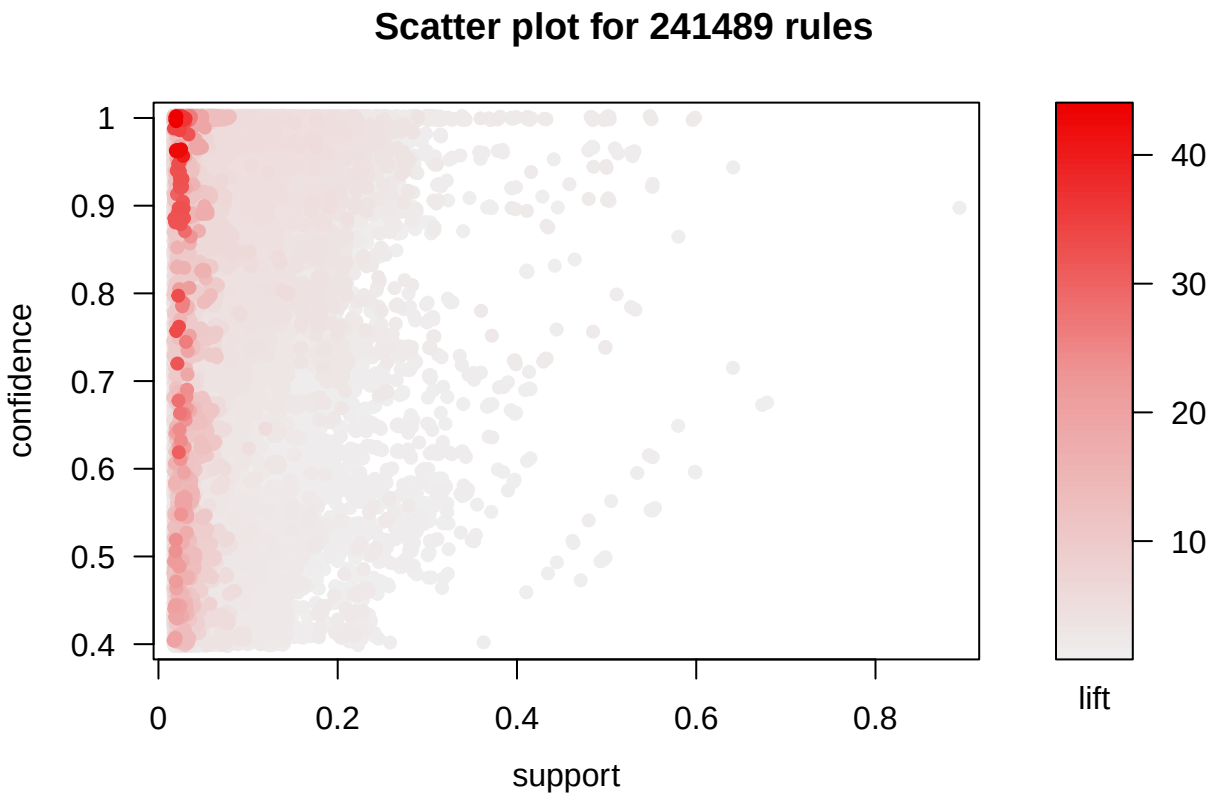
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5    0.02      1
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 829
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[157 item(s), 41481 transaction(s)] done [0.19s].
## sorting and recoding items ... [111 item(s)] done [0.01s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 done [0.87s].
## writing ... [241489 rule(s)] done [0.02s].
## creating S4 object  ... done [0.06s].
```

```
summary(seg.rules)
```

```
## set of 241489 rules
##
## rule length distribution (lhs + rhs):sizes
##      1      2      3      4
##     13   1513  32769 207194
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   4.000   3.852   4.000   4.000
##
## summary of quality measures:
```

```
##     support           confidence          lift             count
##  Min.   :0.02001   Min.   :0.4000   Min.   : 0.5919   Min.   :  830
##  1st Qu.:0.02522   1st Qu.:0.5713   1st Qu.: 1.0633   1st Qu.: 1046
##  Median :0.03469   Median :0.7723   Median : 1.5420   Median : 1439
##  Mean   :0.04710   Mean   :0.7577   Mean   : 2.1410   Mean   : 1954
##  3rd Qu.:0.05456   3rd Qu.:0.9609   3rd Qu.: 2.9490   3rd Qu.: 2263
##  Max.   :0.89537   Max.   :1.0000   Max.   :43.8488   Max.   :37141
##
## mining info:
##       data ntransactions support confidence
##  seg.trans         41481    0.02        0.4
```

```
plot(seg.rules)
```



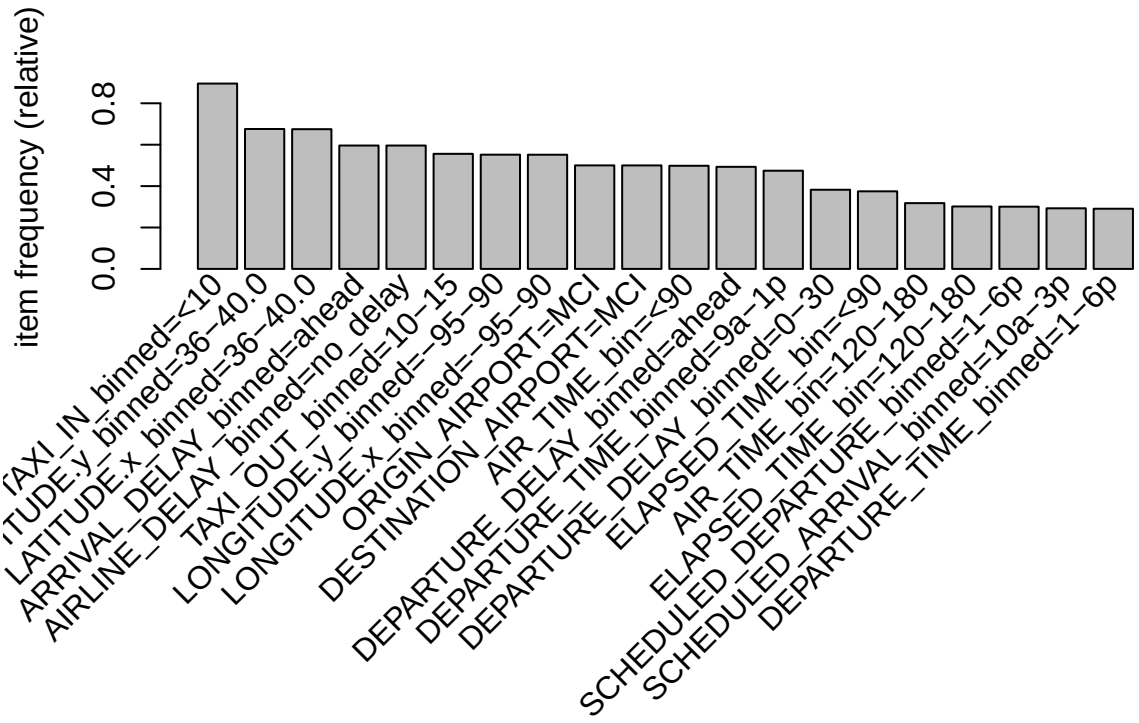
inspect top rules

```
## looking at the orginal set of rules yields 241489 rules
## (with support of 2%, confidence of 40%)
seg.hi <- head(sort(seg.rules, by = "lift"), 20)

inspect(seg.hi)
```

inspecting the top rules shows that certain lat/lon and cities are covariate, as are delays

adjust parameters and check top rules

```
sw_flights_trans <- as(southwest_flights_mba, "transactions")
itemFrequencyPlot(sw_flights_trans, topN = 20)
```




```
basic_rules <- apriori(sw_flights_trans, parameter = list(support = 0.05,
    conf = 0.5, target = "rules", maxlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2074
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[157 item(s), 41481 transaction(s)] done [0.17s].
## sorting and recoding items ... [90 item(s)] done [0.01s].
```

```
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 done [0.56s].
## writing ... [63062 rule(s)] done [0.01s].
## creating S4 object  ... done [0.03s].
```

```
## sort rules by 'lift' -> odds ratio
basic_rules <- sort(basic_rules, by = c("lift"))
inspect(basic_rules[1:5])
```

```
##     lhs                             rhs                           support confidence     lift count
## [1] {AIR_TIME_bin=<90,
##      LATITUDE.y_binned=<36.0,
##      LONGITUDE.y_binned=<-95.0} => {DESTINATION_AIRPORT=DAL} 0.05098720  0.9869342 18.92696  2115
## [2] {AIR_TIME_bin=<90,
##      LATITUDE.x_binned=<36.0,
##      LONGITUDE.x_binned=<-95.0} => {ORIGIN_AIRPORT=DAL}      0.05144524  0.9015632 17.25784  2134
## [3] {AIR_TIME_bin=<90,
##      TAXI_IN_binned=<10,
##      LATITUDE.x_binned=<36.0}   => {ORIGIN_AIRPORT=DAL}      0.05055327  0.8934810 17.10313  2097
## [4] {AIR_TIME_bin=<90,
##      LATITUDE.x_binned=<36.0}   => {ORIGIN_AIRPORT=DAL}      0.05144524  0.8921405 17.07747  2134
## [5] {DESTINATION_AIRPORT=MCI,
##      AIR_TIME_bin=<90,
##      LATITUDE.x_binned=<36.0}   => {ORIGIN_AIRPORT=DAL}      0.05144524  0.8921405 17.07747  2134
```

picked up inferred relations (i.e. {AIR_TIME_bin=<90, LATITUDE.y_binned=<36.0,LONGITUDE.y_binned=<-95.0} => {DESTINATION_AIRPORT=DAL})

analyze rules with Day of Week

```
day_of_week_rules <- apriori(seg.trans, parameter = list(support = 0.001,
    confidence = 0.05), appearance = list(lhs = c("ORIGIN_AIRPORT=MCI",
    "DAY_OF_WEEK=1", "DAY_OF_WEEK=2", "DAY_OF_WEEK=3", "DAY_OF_WEEK=4",
    "DAY_OF_WEEK=5", "DAY_OF_WEEK=6", "DAY_OF_WEEK=7")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 41
##
## set item appearances ...[8 item(s)] done [0.00s].
## set transactions ...[157 item(s), 41481 transaction(s)] done [0.24s].
## sorting and recoding items ... [152 item(s)] done [0.01s].
```

```
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 done [0.08s].
## writing ... [1282 rule(s)] done [0.00s].
## creating S4 object  ... done [0.02s].
```

```
month_rules_partial <- subset(day_of_week_rules, items %pin%
    "DEPARTURE_DELAY_binned")
inspect(sort(month_rules_partial[1:10], by = c("lift", "count",
    "confidence")))
```

```
##      lhs                rhs                              support
## [1]  {DAY_OF_WEEK=4} => {DEPARTURE_DELAY_binned=30-90} 0.014681420
## [2]  {DAY_OF_WEEK=5} => {DEPARTURE_DELAY_binned=30-90} 0.013090331
## [3]  {DAY_OF_WEEK=3} => {DEPARTURE_DELAY_binned=30-90} 0.013596586
## [4]  {}              => {DEPARTURE_DELAY_binned=0-30}  0.382416046
## [5]  {}              => {DEPARTURE_DELAY_binned=ahead} 0.493141438
## [6]  {}              => {DEPARTURE_DELAY_binned=30-90} 0.086642077
## [7]  {DAY_OF_WEEK=2} => {DEPARTURE_DELAY_binned=30-90} 0.012993901
## [8]  {DAY_OF_WEEK=1} => {DEPARTURE_DELAY_binned=30-90} 0.012632289
## [9]  {DAY_OF_WEEK=7} => {DEPARTURE_DELAY_binned=30-90} 0.011234059
## [10] {DAY_OF_WEEK=6} => {DEPARTURE_DELAY_binned=30-90} 0.008413491
##      confidence lift      count
## [1]  0.09875142 1.1397628   609
## [2]  0.08942688 1.0321414   543
## [3]  0.08898706 1.0270652   564
## [4]  0.38241605 1.0000000 15863
## [5]  0.49314144 1.0000000 20456
## [6]  0.08664208 1.0000000  3594
## [7]  0.08531181 0.9846464   539
## [8]  0.08353260 0.9641112   524
## [9]  0.08135475 0.9389750   466
## [10] 0.07611778 0.8785313   349
```

```
dow_rules_partial <- subset(day_of_week_rules, items %pin% "AIR_TIME_bin")
inspect(sort(dow_rules_partial[1:5], by = c("lift", "count",
    "confidence")))
```

```
##     lhs                rhs                       support     confidence
## [1] {DAY_OF_WEEK=6} => {AIR_TIME_bin=180+}    0.006629541 0.05997819
## [2] {}              => {AIR_TIME_bin=<90}     0.498228104 0.49822810
## [3] {}              => {AIR_TIME_bin=120-180} 0.317832261 0.31783226
## [4] {}              => {AIR_TIME_bin=90-120}  0.115040621 0.11504062
## [5] {}              => {AIR_TIME_bin=180+}    0.057472096 0.05747210
##     lift     count
## [1] 1.043605   275
## [2] 1.000000 20667
## [3] 1.000000 13184
## [4] 1.000000  4772
## [5] 1.000000  2384
```

– 1.13x more likely to be delayed 30-90 minutes on Thursday – looked at day of week by destination and found nothing significant – day of the week by time in the air, longer flights (1.16 lift) 2-3 hours on Saturdays

extended weather delays by destination

```
weather_rules <- apriori(sw_flights_trans, parameter = list(support = 2e-04,
    confidence = 0.2, maxlen = 4), appearance = list(lhs = c("WEATHER_DELAY_binned=10+")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5   2e-04      1
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 8
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[157 item(s), 41481 transaction(s)] done [0.25s].
## sorting and recoding items ... [156 item(s)] done [0.01s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [100 rule(s)] done [0.00s].
## creating S4 object  ... done [0.02s].
```

```
weather_rules_partial <- subset(weather_rules, items %pin% "LON" |
    items %pin% "LAT")
inspect(sort(weather_rules_partial[1:5], by = c("lift", "count",
    "confidence")))
```

```
##     lhs    rhs                          support   confidence lift count
## [1] {}  => {LONGITUDE.y_binned=-90-70} 0.2310938 0.2310938  1    9586
## [2] {}  => {LONGITUDE.x_binned=<-95.0} 0.2179793 0.2179793  1    9042
## [3] {}  => {LONGITUDE.y_binned=<-95.0} 0.2171114 0.2171114  1    9006
## [4] {}  => {LATITUDE.x_binned=<36.0}   0.2053952 0.2053952  1    8520
## [5] {}  => {LATITUDE.y_binned=<36.0}   0.2048167 0.2048167  1    8496
```

Association Rules findings:

The analysis using A-Rules first required transforming the dataset into a transactional style dataset (long) grouped by unique flightID. I tested several values for lift, support, and confidence. There was a tradeoff for finding rules that had a large enough sample size to be generalizable and rules that gave us high value (lift)... eventually we settled on the tunings to: seg.rules <- apriori(seg.trans, parameter=list(support=0.02, conf=0.4, target="rules",maxlen=4)) This initial criteria led to a set Initial Criteria led to 241,489 "rules" (Figure 3.1). So, I inspected the results in "lift" order, which is effectively an odds ratio. The initial set was difficult to interpret and picked up some inferred relationships (i.e. {AIR_TIME_bin=<90, LATITUDE.y_binned=<36.0,LONGITUDE.y_binned=<-95.0} => {DESTINATION_AIRPORT=DAL}). I could then tune filter these list to include certain values of interest using "lhs" and "rhs" (i.e. effects of latitude and longitude on different delay types) (See figure 3.2). I started my process by looking at natural relationships to day of the week (Figure 3.3) and found 1.13x more likely

to be delayed 30-90 minutes on Thursday. I further filtered day of week by destination and found nothing significant. Then looking at day of the week by time in the air, found that flights tended to be longer (1.16 lift) 2-3 hours on Saturdays. I then analyzed how latitude/longitude related to weather delays and found 5.7x more likely to be on a delayed flight between 12p-9a and South and West US more likely to get delayed for weather ({WEATHER_DELAY_binned=10+} => {LONGITUDE.x_binned=-90-70} 0.001639305 0.2753036 1.191307 68)

Factor Analysis

subset to Southwest Airlines flights

```
## install.packages('mltools', 'nFactors', 'GPArotation',
## 'psych')
require(mltools)
require(nFactors)
library(psych)
library(GPArotation)

southwest_flights <- flights_df2[which(flights_df2$AIRLINE ==
    "WN"), ]
## create uniqueID columns
southwest_flights$FLIGHT_ID <- seq.int(nrow(southwest_flights))

## subset
southwest_flights_fa <- southwest_flights[, which(names(southwest_flights) %in%
    c("MONTH", "DAY_OF_WEEK", "SCHEDULED_ARRIVAL", "DEPARTURE_TIME",
        "SCHEDULED_DEPARTURE", "TAXI_OUT", "WHEELS_OFF", "ELAPSED_TIME",
        "SCHEDULED_TIME", "AIR_TIME", "WHEELS_ON", "TAXI_IN",
        "ARRIVAL_TIME", "ARRIVAL_DELAY", "AIR_SYSTEM_DELAY",
        "SECURITY_DELAY", "AIRLINE_DELAY", "WEATHER_DELAY", "LATITUDE.x",
        "LATITUDE.y", "LONGITUDE.x", "LONGITUDE.y"))]

southwest_flights_fa_dependent <- southwest_flights[, which(names(southwest_flights) %in%
    c("DEPARTURE_DELAY"))]

### convert columns to be one hot encoded to factors
southwest_flights_fa$MONTH <- as.factor(southwest_flights_fa$MONTH)
southwest_flights_fa$DAY_OF_WEEK <- as.factor(southwest_flights_fa$DAY_OF_WEEK)

### one-hot encode factor columns
southwest_flights_fa <- as.data.table(southwest_flights_fa)
southwest_flights_fa <- one_hot(southwest_flights_fa, cols = c("MONTH",
    "DAY_OF_WEEK"), sparsifyNAs = FALSE, dropCols = TRUE, dropUnusedLevels = FALSE)

## drop column to avoid perfect multicolinearity
southwest_flights_fa <- as.data.frame(southwest_flights_fa)
southwest_flights_fa <- southwest_flights_fa[, -which(names(southwest_flights_fa) %in%
    c("DAY_OF_WEEK_2"))]

# We replace missing observation with variable means (this
# step depends on the real research needs and your decisions
# on the missing observations)
southwest_flights_fa[] <- lapply(southwest_flights_fa, function(x) {
```

```
    x[is.na(x)] <- mean(x, na.rm = TRUE)
    x
})
```

scale dataframe

```
southwest_flights_fa.scaled <- scale(southwest_flights_fa)
southwest_flights_fa <- as.data.frame(southwest_flights_fa.scaled)

## check dimensions & summary of scaled df
summary(southwest_flights_fa)
dim(southwest_flights_fa)
```

find optimal number of Factors

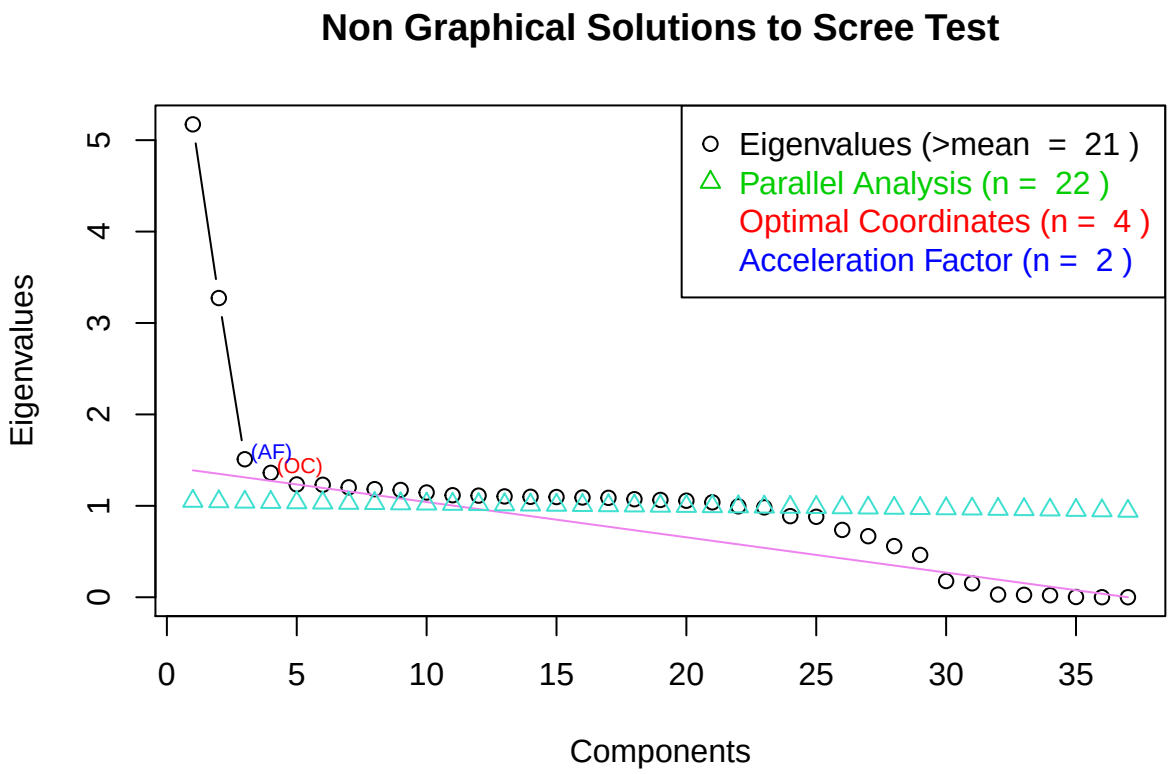
```
# nScree give some indicators about the suggested number of
# factors, in this case 1 or 14
nScree(southwest_flights_fa)
```

```
##   noc naf nparallel nkaiser
## 1   4   2        21      21
```

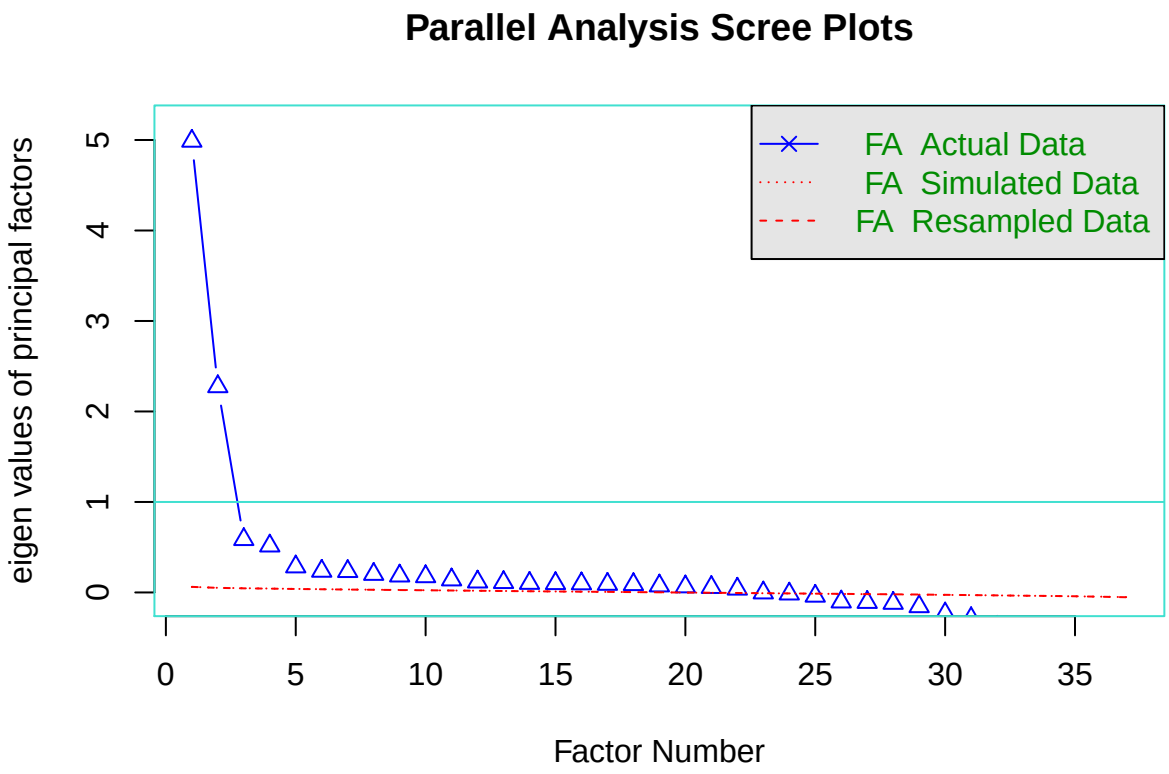
```
# Get eigenvalues. They should be >1 in order to be
# considered, but not necessarily included
eig <- eigen(cor(southwest_flights_fa))

# print top 20 eigenvalues
print(eig$values[1:25])
```

```
##  [1] 5.1729880 3.2727674 1.5094746 1.3607787 1.2342600 1.2302093 1.2022417
##  [8] 1.1825236 1.1735566 1.1458150 1.1158462 1.1115260 1.1029603 1.0987040
## [15] 1.0960090 1.0911638 1.0874391 1.0722472 1.0643861 1.0557564 1.0381174
## [22] 0.9944317 0.9829307 0.8869631 0.8799957
```

```
# This way gives the graph showing suggested number of
# factors.
ap <- parallel(subject = nrow(southwest_flights_fa), var = ncol(southwest_flights_fa))
nS <- nScree(x = eig$values, aparallel = ap$eigen$qevpea)
plotnScree(nS)
```




```
parallel <- fa.parallel(southwest_flights_fa, fm = "minres",
    fa = "fa")
```




```
## Parallel analysis suggests that the number of factors =  23  and the number of components =  NA
```

```
parallel
```

```
## Call: fa.parallel(x = southwest_flights_fa, fm = "minres", fa = "fa")
## Parallel analysis suggests that the number of factors =  23  and the number of components =  NA
##
##  Eigen Values of
##
##  eigen values of factors
##  [1]  4.99  2.27  0.59  0.51  0.28  0.24  0.23  0.20  0.18  0.17  0.14
## [12]  0.12  0.11  0.10  0.10  0.10  0.09  0.09  0.07  0.06  0.06  0.04
## [23]  0.00 -0.02 -0.04 -0.10 -0.11 -0.12 -0.16 -0.24 -0.29 -0.38 -0.50
## [34] -0.82 -0.98 -1.00 -1.00
##
##  eigen values of simulated factors
##  [1]  0.06  0.05  0.05  0.04  0.04  0.04  0.03  0.03  0.03  0.02  0.02
## [12]  0.02  0.02  0.01  0.01  0.01  0.01  0.00  0.00  0.00  0.00 -0.01
## [23] -0.01 -0.01 -0.01 -0.02 -0.02 -0.02 -0.02 -0.03 -0.03 -0.03 -0.04
## [34] -0.04 -0.04 -0.05 -0.05
##
##  eigen values of components
##  [1] 5.17 3.27 1.51 1.36 1.23 1.23 1.20 1.18 1.17 1.15 1.12 1.11 1.10 1.10
## [15] 1.10 1.09 1.09 1.07 1.06 1.06 1.04 0.99 0.98 0.89 0.88 0.74 0.67 0.56
## [29] 0.46 0.18 0.15 0.03 0.03 0.02 0.00 0.00 0.00
```

```
##
##  eigen values of simulated components
## [1] NA
```

```
# We run FA using 'Varimax' rotation with 'minres' (minimum
# residual) The 'scores=' gives the factor scores that we
# will use in future steps. This is another important step
southwest_flights_fa4 <- fa(southwest_flights_fa, nfactors = 4,
    rotate = "Varimax", fm = "minres", scores = "regression")
# southwest_flights_fa4

# The root mean square of residuals (RMSR) is 0.05. This is
# acceptable as this value should be closer to 0.  RMSEA
# ranges from 0 to 1, with smaller values indicating better
# model fit.


# print top factor loadings, which can then interpret what
# variables are loaded on what factor and we can term the
# factor with a new name. Both positive and negative loading
# are useful for indicating the meaning of a factor
print(southwest_flights_fa4$loadings, cutoff = 0.1, sort = TRUE)
```

```
##
## Loadings:
##                      MR1    MR2    MR3    MR4
## SCHEDULED_DEPARTURE  0.942                -0.170
## DEPARTURE_TIME       0.964                -0.226
## WHEELS_OFF           0.963                -0.221
## WHEELS_ON            0.871                 0.431
## SCHEDULED_ARRIVAL    0.884                 0.190
## ARRIVAL_TIME         0.851                 0.451
## SCHEDULED_TIME               0.985
## ELAPSED_TIME                 0.996  0.125
## AIR_TIME                     0.998
## TAXI_OUT                     0.137  0.520
## AIR_SYSTEM_DELAY                    0.591
## MONTH_1
## MONTH_2
## MONTH_3
## MONTH_4
## MONTH_5
## MONTH_6
## MONTH_7
## MONTH_8
## MONTH_9
## MONTH_11
## MONTH_12
## DAY_OF_WEEK_1
## DAY_OF_WEEK_3
## DAY_OF_WEEK_4
## DAY_OF_WEEK_5
## DAY_OF_WEEK_6
## DAY_OF_WEEK_7
```

```
## TAXI_IN                           0.207
## ARRIVAL_DELAY        0.225        0.238 -0.334
## SECURITY_DELAY
## AIRLINE_DELAY                           -0.186
## WEATHER_DELAY                     0.106
## LATITUDE.x                -0.159
## LONGITUDE.x
## LATITUDE.y                -0.181
## LONGITUDE.y               -0.339
##
##                   MR1   MR2   MR3   MR4
## SS loadings     5.078 3.175 0.793 0.739
## Proportion Var 0.137 0.086 0.021 0.020
## Cumulative Var 0.137 0.223 0.244 0.264
```

```
# (optional) function visually shows the loading
# fa.diagram(retail.fa)
```

name factors (based on loading weights)

```
# rename factors based on weightings analyzed.  This
# dimesnionally reduces the number and simplifies
# interpretation
southwest_flights_fa4.scores <- as.data.frame(southwest_flights_fa4$scores)

colnames(southwest_flights_fa4.scores) <- c("scheduled_v_actual_timing",
    "time_in_air_and_latlon", "delays_and_taxi_time", "MR4")
southwest_flights_fa4.scores
```

join factors to dependent variables for regression

```
southwest_flights_fa.final <- cbind(southwest_flights_fa_dependent,
    southwest_flights_fa4.scores)

names(southwest_flights_fa.final)
```

```
## [1] "southwest_flights_fa_dependent" "scheduled_v_actual_timing"
## [3] "time_in_air_and_latlon"         "delays_and_taxi_time"
## [5] "MR4"
```

```
# head(southwest_flights_fa.final)

## write to csv for part 2
## write.csv(southwest_flights_fa.final,
## file='southwest_flights_fa.csv')
```

Factor Analysis findings:

I first had to one hot encode the factor columns (Day of week & Month) then scaled (zscores) the data and removed the dependent variable (departure_delay), and columns to avoid perfect multicollineaity. Initial

metrics suggested (1, 4, 2, 21, 21) (Figure 4.1). After looking at the eigenvalues – found that 21 were above the needed 1.0 threshold (top 3 – 5.17, 3.27, 1.53)... showing significant drop off in common variance explained. I decided to move forward with 4 (based on the chart) though the top 2 had a majority of the explanatory power but included 4 based on the chart to reduce "noise" in the top 2 features and see if there was any intepretible pattern in the additional factors. The top group weighted columns by grouping (see figure 4.2):

1- "timing_vs_scheduled" – scheduled departure, dep_time, wheels_off, wheels_on, Scheduled_arrived, actual_arrived 2- "time_in_air_lat/lon" – scheduled time, elapsed time, Scheduled_arrived, actual_arrived, latitude.x/y (negative weight), longitude.y (negative weight) 3- "delays_and_taxi_time" – taxi in/out, arrival/weather/air_system delays 4- "MR4" – mix of negative weights

Factor Analysis: OLS Regression

OLS Regression - using newly formed factors

```
summary(southwest_flights_fa.final)
```

```
##  southwest_flights_fa_dependent scheduled_v_actual_timing
##  Min.   :-15.00                 Min.   :-3.105245
##  1st Qu.: -3.00                 1st Qu.:-0.844493
##  Median :  0.00                 Median :-0.006127
##  Mean   : 10.15                 Mean   : 0.000000
##  3rd Qu.: 10.00                 3rd Qu.: 0.834848
##  Max.   :495.00                 Max.   : 3.235188
##  NA's   :370
##  time_in_air_and_latlon delays_and_taxi_time      MR4
##  Min.   :-30.7258       Min.   :-853.0867    Min.   :-9.26488
##  1st Qu.: -0.8378       1st Qu.:  -0.5542    1st Qu.:-0.05814
##  Median : -0.3235       Median :  -0.1294    Median : 0.09730
##  Mean   :  0.0000       Mean   :   0.0000    Mean   : 0.00000
##  3rd Qu.:  0.8551       3rd Qu.:   0.4122    3rd Qu.: 0.29951
##  Max.   :  3.6200       Max.   :  91.1612    Max.   :32.33457
##
```

```
dim(southwest_flights_fa.final)
```

```
## [1] 41481     5
```

```
## drop NA records of dependent variable
southwest_flights_fa.final <- southwest_flights_fa.final[!is.na(southwest_flights_fa.final$southwest_fli
    ]
## droped 370/41481 records
dim(southwest_flights_fa.final)
```

```
## [1] 41111     5
```

```
## look at prior distribution of target - Grocery
## food
summary(southwest_flights_fa.final$southwest_flights_fa_dependent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -15.00   -3.00    0.00   10.15   10.00  495.00
```

```
## create linear regression(s) to check feature
## importance
Sample.model1 <- lm(data = southwest_flights_fa.final,
    southwest_flights_fa_dependent ~ scheduled_v_actual_timing +
        time_in_air_and_latlon + delays_and_taxi_time +
        MR4)

summary(Sample.model1)
```

```
##
## Call:
## lm(formula = southwest_flights_fa_dependent ~ scheduled_v_actual_timing +
##     time_in_air_and_latlon + delays_and_taxi_time + MR4, data = southwest_flights_fa.final)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79.91 -11.68  -5.08   1.93 514.95
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 10.16218    0.13871  73.263  < 2e-16 ***
## scheduled_v_actual_timing    7.07116    0.13860  51.020  < 2e-16 ***
## time_in_air_and_latlon       0.75872    0.13801   5.498 3.87e-08 ***
## delays_and_taxi_time        -0.42638    0.02155 -19.783  < 2e-16 ***
## MR4                        -11.11660    0.14575 -76.272  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.12 on 41106 degrees of freedom
## Multiple R-squared:  0.1716, Adjusted R-squared:  0.1715
## F-statistic:  2129 on 4 and 41106 DF,  p-value: < 2.2e-16
```

```
## get odds ratio
exp(Sample.model1$coefficients)
```

```
##               (Intercept) scheduled_v_actual_timing
##              2.590470e+04              1.177514e+03
##    time_in_air_and_latlon      delays_and_taxi_time
##              2.135535e+00              6.528693e-01
##                       MR4
##              1.486349e-05
```

FA Regression findings:

I then joined the resulting factor columns as features to the hold out dependent variable (continuous - departure_delay) and looked at the prior distribution of the dependent variable I then hen performed linear regression and found that all factors reached statistical significance (Figure 5.2), though the r2 was still just 0.171 - 'scheduled_vs_actual_timing' had the highest positive weight (aka increased the delay the most, with increase).

Analysis Finidngs

results

I was pleased with the output of my analysis as it led to several solid findings. Delta has less "severe" delays out of MCI. Based on analysis of Latitude/Longitude, SW united states has more delays and weather related delays relative to other regions. Passengers are 13 percent more likely to be delayed 30-90 minutes on Thursday. Flights are 5.7x more likely to be on a delayed flight between 12a-9a. In further analysis I would look into reason for this disparity (primarily night delayed flights that went out early the next morning?)

reccomendations

In the Kansas City market, Delta ostensibly has better performance with "severe" delays out of MCI. Management should analyze Delta staffing/policies – to try to understand why they have less delays. Without pricing data it's tough to tell, but possibly move focus to areas outside of the Southwest. Based on analysis of Latitude/Longitude, the southwest United States has more delays and weather delays relative to other regions. Lastly look at increasing staff on Thursdays, (pilots, flex shifts, etic.).

limitations

Granularity in understanding the exact issue that caused a flight to depart late (i.e. reason code). It's hard to look at these delays in isolation of pricing, if anything this could be an interesting feature. Looking into additional cities and their interconnectedness of these flight networks would be huge in understanding causes of delays (i.e. weather in another city with a plane scheduled from MCI flight is). More recent Data would help the validity of model (this data comes from 2015). An ideal state could be doing trend analysis and near real time "at-risk" flight analysis.

"'