```
library(knitr)
.finished <- FALSE
knit_hooks$set(timeit = function(before) {
    if (before) {
      .current.time <<- Sys.time()
    } else {
      .duration <- round(difftime(Sys.time(), .curre
      if(!.finished)
        write(
          paste0(
            knitr::opts_current$get(name = "label"),
            ": ",
            .duration),
          file = "analysis-post-2008-CHUNKTIMINGS.tx
          ncolumns = 1,
          append = TRUE)
    }
})
file.remove("analysis-post-2008-CHUNKTIMINGS.txt")
```

```
## [1] TRUE
```

```
START.TIME <- Sys.time()
knitr::opts_chunk$set(fig.show = 'hide',
                      fig.width = 11,
                      fig.height = 7,
                      fig.path = atlas <- "atlas-pos
                      timeit = TRUE,
                      cache=FALSE,
                      out.width = "11in")
```

```
# use RDS: allow previously generated files to be re
# saves time but might be dangerous. Must rely on `t
useRDS = TRUE
```

```
library(tidyr)
library(data.table)
library(bit64)
```

```
## Loading required package:  bit
## Attaching package bit
## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL
## creators:  bit bitwhich
## coercion:  as.logical as.integer as.bit as.bitwhi
which
## operator:  !  & | xor != ==
## querying:  print length any all min max range sun
summary
## bit access:  length<- [ [<- [[ [[<-
## for more help type ?bit
##
## Attaching package:  'bit'
##
## The following object is masked from 'package:data
##
##    setattr
##
## The following object is masked from 'package:base
##
##    xor
```

```
##
## Attaching package bit64
## package:bit64 (c) 2011-2012 Jens Oehlschlaegel (GPL-2
with commercial restrictions)
## creators:  integer64 seq :
## coercion:  as.integer64 as.vector as.logical as.integer
as.double as.character as.bin
## logical operator:  !  & | xor != == < <= >= >
## arithmetic operator:  + - * / %/% %% ^
## math:  sign abs sqrt log log2 log10
## math:  floor ceiling trunc round
## querying:  is.integer64 is.vector [is.atomic} [length]
is.na format print
## aggregation:  any all min max range sum prod
## cumulation:  diff cummin cummax cumsum cumprod
## access:  length<- [ [<- [[ [[<-
## combine:  c rep cbind rbind as.data.frame
## for more help type ?bit64
##
## Attaching package:  'bit64'
##
## The following object is masked from 'package:bit':
##
##    still.identical
##
## The following objects are masked from 'package:base':
##
##    %in%, :, is.double, match, order, rank
```

```
library(dplyr)
```

```
##
## Attaching package:  'dplyr'
##
## The following objects are masked from 'package:data.table':
##
##    between, last
##
## The following objects are masked from 'package:stats':
##
##    filter, lag
##
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union
```

```
library(magrittr)
```

```
##
## Attaching package:  'magrittr'
##
## The following object is masked from 'package:tidyr':
##
##    extract
```

```
library(ggplot2)
theme.text.size = 18
text.size = (5/14) * theme.text.size
theme_update(text = element_text(family = "",
                                 face = "plain", colour = "black", size =
                                 lineheight = 0.9,
                                 hjust = 0.5, vjust = 0.5,
                                 angle = 0, margin = margin(),
                                 debug = FALSE))
update_geom_defaults("text", list(size = text.size))
update_geom_defaults("line", list(size = 2))
library(ggrepel)
library(scales)
library(nycflights13)  # for airports
nycflights.airports <- airports
nycflights.planes   <- planes
nycflights.airlines <- as.data.table(airlines)
for (j in 1:ncol(nycflights.airlines)){
  set(nycflights.airlines, j = j, value = as.charact
}
nycflights.airlines[,short_name := gsub("\\s.*$", ""
setnames(nycflights.airlines, "carrier", "UniqueCarr
setkey(nycflights.airlines, UniqueCarrier)
library(fasttime)
library(grattan)
```

```
##
## Attaching package:  'grattan'
##
## The following object is masked from 'package:data
##
##    Orange
```

```
library(directlabels)
library(ineq)  # for Gini()
```

```
convert_week_to_date <- function(DT_with_Week_column
  stopifnot(is.data.table(DT_with_Week_column), "Wee
  setkey(DT_with_Week_column, Week)
  temp <-
    unique_dates %>%
    group_by(Week) %>%
    summarise(Date = fastPOSIXct(sprintf("%d-%02d-%0 ) %>%
    setkey(Week)

  DT_with_Week_column[temp]
}
```

```
flights <- fread("../post2008_flights.csv", na.strin
```

```
##
Read 0.0% of 49153341 rows
Read 0.7% of 49153341 rows
Read 1.4% of 49153341 rows
Read 2.1% of 49153341 rows
Read 2.8% of 49153341 rows
Read 3.5% of 49153341 rows
Read 4.2% of 49153341 rows
Read 4.8% of 49153341 rows
Read 5.5% of 49153341 rows
Read 6.2% of 49153341 rows
Read 6.9% of 49153341 rows
Read 7.6% of 49153341 rows
Read 8.3% of 49153341 rows
Read 9.0% of 49153341 rows
Read 9.7% of 49153341 rows
Read 10.4% of 49153341 rows
Read 11.1% of 49153341 rows
Read 11.8% of 49153341 rows
Read 12.5% of 49153341 rows
Read 13.1% of 49153341 rows
Read 13.8% of 49153341 rows
Read 14.5% of 49153341 rows
Read 15.2% of 49153341 rows
Read 15.9% of 49153341 rows
Read 16.6% of 49153341 rows
Read 17.3% of 49153341 rows
Read 18.0% of 49153341 rows
Read 18.7% of 49153341 rows
Read 19.4% of 49153341 rows
Read 20.1% of 49153341 rows
Read 20.8% of 49153341 rows
Read 21.5% of 49153341 rows
Read 22.2% of 49153341 rows
Read 22.9% of 49153341 rows
Read 23.6% of 49153341 rows
Read 24.3% of 49153341 rows
Read 25.0% of 49153341 rows
Read 25.7% of 49153341 rows
Read 26.3% of 49153341 rows
Read 27.1% of 49153341 rows
Read 27.7% of 49153341 rows
Read 28.4% of 49153341 rows
Read 29.1% of 49153341 rows
Read 29.8% of 49153341 rows
Read 30.5% of 49153341 rows
Read 31.2% of 49153341 rows
Read 31.9% of 49153341 rows
Read 32.6% of 49153341 rows
Read 33.3% of 49153341 rows
Read 34.0% of 49153341 rows
Read 34.7% of 49153341 rows
Read 35.4% of 49153341 rows
Read 36.1% of 49153341 rows
Read 36.8% of 49153341 rows
Read 37.5% of 49153341 rows
Read 38.1% of 49153341 rows
Read 38.8% of 49153341 rows
Read 39.5% of 49153341 rows
Read 40.2% of 49153341 rows
Read 40.9% of 49153341 rows
Read 41.6% of 49153341 rows
Read 42.3% of 49153341 rows
Read 43.0% of 49153341 rows
Read 43.7% of 49153341 rows
Read 44.4% of 49153341 rows
Read 45.1% of 49153341 rows
Read 45.8% of 49153341 rows
Read 46.4% of 49153341 rows
Read 47.1% of 49153341 rows
Read 47.8% of 49153341 rows
Read 48.5% of 49153341 rows
Read 49.2% of 49153341 rows
Read 49.9% of 49153341 rows
Read 50.6% of 49153341 rows
Read 51.3% of 49153341 rows
Read 52.0% of 49153341 rows
Read 52.7% of 49153341 rows
Read 53.4% of 49153341 rows
Read 54.1% of 49153341 rows
Read 54.7% of 49153341 rows
Read 55.4% of 49153341 rows
Read 56.1% of 49153341 rows
Read 56.8% of 49153341 rows
Read 57.5% of 49153341 rows
Read 58.2% of 49153341 rows
Read 58.9% of 49153341 rows
Read 59.6% of 49153341 rows
Read 60.3% of 49153341 rows
Read 61.0% of 49153341 rows
Read 61.7% of 49153341 rows
Read 62.4% of 49153341 rows
Read 63.0% of 49153341 rows
Read 63.7% of 49153341 rows
Read 64.4% of 49153341 rows
Read 65.1% of 49153341 rows
Read 65.8% of 49153341 rows
Read 66.5% of 49153341 rows
Read 67.2% of 49153341 rows
Read 67.9% of 49153341 rows
Read 68.6% of 49153341 rows
Read 69.3% of 49153341 rows
Read 70.0% of 49153341 rows
Read 70.7% of 49153341 rows
Read 71.3% of 49153341 rows
Read 72.0% of 49153341 rows
Read 72.7% of 49153341 rows
Read 73.4% of 49153341 rows
Read 74.1% of 49153341 rows
Read 74.8% of 49153341 rows
Read 75.5% of 49153341 rows
Read 76.2% of 49153341 rows
Read 76.9% of 49153341 rows
Read 77.6% of 49153341 rows
Read 78.3% of 49153341 rows
Read 79.0% of 49153341 rows
Read 79.6% of 49153341 rows
Read 80.3% of 49153341 rows
Read 81.0% of 49153341 rows
Read 81.7% of 49153341 rows
Read 82.4% of 49153341 rows
Read 83.1% of 49153341 rows
Read 83.8% of 49153341 rows
Read 84.5% of 49153341 rows
Read 85.2% of 49153341 rows
Read 85.9% of 49153341 rows
Read 86.6% of 49153341 rows
Read 87.3% of 49153341 rows
Read 87.9% of 49153341 rows
Read 88.6% of 49153341 rows
Read 89.3% of 49153341 rows
Read 90.0% of 49153341 rows
Read 90.7% of 49153341 rows
Read 91.4% of 49153341 rows
Read 92.1% of 49153341 rows
Read 92.8% of 49153341 rows
Read 93.5% of 49153341 rows
Read 94.2% of 49153341 rows
Read 94.9% of 49153341 rows
Read 95.6% of 49153341 rows
```

```
Read 96.2% of 49153341 rows
Read 96.9% of 49153341 rows
Read 97.6% of 49153341 rows
Read 98.3% of 49153341 rows
Read 99.0% of 49153341 rows
Read 99.7% of 49153341 rows
Read 49153341 rows and 65 (of 65) columns from 13.203 GB file in 00:03:19

flights[,tempkey := 1:.N]
```

```
flights.by.carrier <- flights[, .(n = .N), keyby = UniqueCarrier]

select_large_carriers <- function(ranking){
  flights.by.carrier %>%
    arrange(desc(n)) %>%
    head(ranking) %$%
    UniqueCarrier
}
```

```
carrier.colors <- RColorBrewer::brewer.pal(11, "Spectral")
names(carrier.colors) <- select_large_carriers(11)
```

```
# First we want a time for each flight. This is more difficult that it m
# We need to concatenate the Year, Month, and DayofMonth fields, but we c
# to take into account the various time zones of the airports in the dat
integer.cols <- grep("Time$", names(flights))

Sys.time()

## [1] "2016-02-05 21:34:52 AEDT"

for (j in integer.cols){
  set(flights, j = j, value = as.integer(flights[[j]]))
}
Sys.time()

## [1] "2016-02-05 21:34:53 AEDT"
```

```
# See stackoverflow: links and comments under my question
create_DepDateTime <- function(DT){
  setkey(DT, Year, Month, DayofMonth, DepTime)
  unique_dates <- unique(DT[,list(Year, Month, DayofMonth, DepTime)])
  unique_dates[,DepDateTime := fastPOSIXct(sprintf("%d-%02d-%02d %s", Yea
                                                    sub("([0-9]{2})([0-9]{
                                                        perl = TRUE)),
                                           tz = "GMT")]
  DT[unique_dates]
}

create_ArrDateTime <- function(DT){
  setkey(DT, Year, Month, DayofMonth, ArrTime)
  unique_dates <- unique(DT[,list(Year, Month, DayofMonth, ArrTime)])
  unique_dates[,ArrDateTime := fastPOSIXct(sprintf("%d-%02d-%02d %s", Yea
                                                    sub("([0-9]{2})([0-9]{
                                                        perl = TRUE)),
                                           tz = "GMT")]
  DT[unique_dates]
}
```

```
flights <- create_DepDateTime(flights)
flights <- create_ArrDateTime(flights)
#flights[,`:=`(Year = NULL, Month = NULL, DayofMonth
Sys.time()
```

```
# Now we join it to the airports dataset from nycfli
Sys.time()
airports <- as.data.table(airports)
airports <- airports[,list(faa, tz)]
setnames(airports, old = c("faa", "tz"), new = c("Or
setkey(airports, Origin)
setkey(flights, Origin)
flights <- flights[airports]
setnames(airports, old = c("Origin", "tzOrigin"), ne
setkey(flights, Dest)
flights <- flights[airports]
rm(airports)
# The joins produce NAs when the airports table isn'
flights <- flights[!is.na(Origin)]
Sys.time()
```

```
Sys.time()
# setting keys doesn't improve timing
flights[,`:=`(DepDateTimeZulu = DepDateTime - lubrid
flights[,`:=`(ArrDateTimeZulu = ArrDateTime - lubrid
Sys.time()
```

```
flights %>%
  select(tempkey, DepDateTime, ArrDateTime, tzOrigin
  saveRDS(file = "flights-post-2008_with_zuluTimes.r
```

```
flights_with_timezones <- readRDS("flights-post-2008_with_zuluTimes.rds")

## Warning in gzfile(file, "rb"):  cannot open compr
file 'flights-post-2008_with_zuluTimes.rds', probable
reason 'No such file or directory'
## Error in gzfile(file, "rb"):  cannot open the con

setkey(flights_with_timezones, tempkey)

## Error in setkey(flights_with_timezones, tempkey):
object 'flights_with_timezones' not found

setkey(flights, tempkey)
flights <- flights[flights_with_timezones]

## Error in eval(expr, envir, enclos):  object 'flig
not found
```

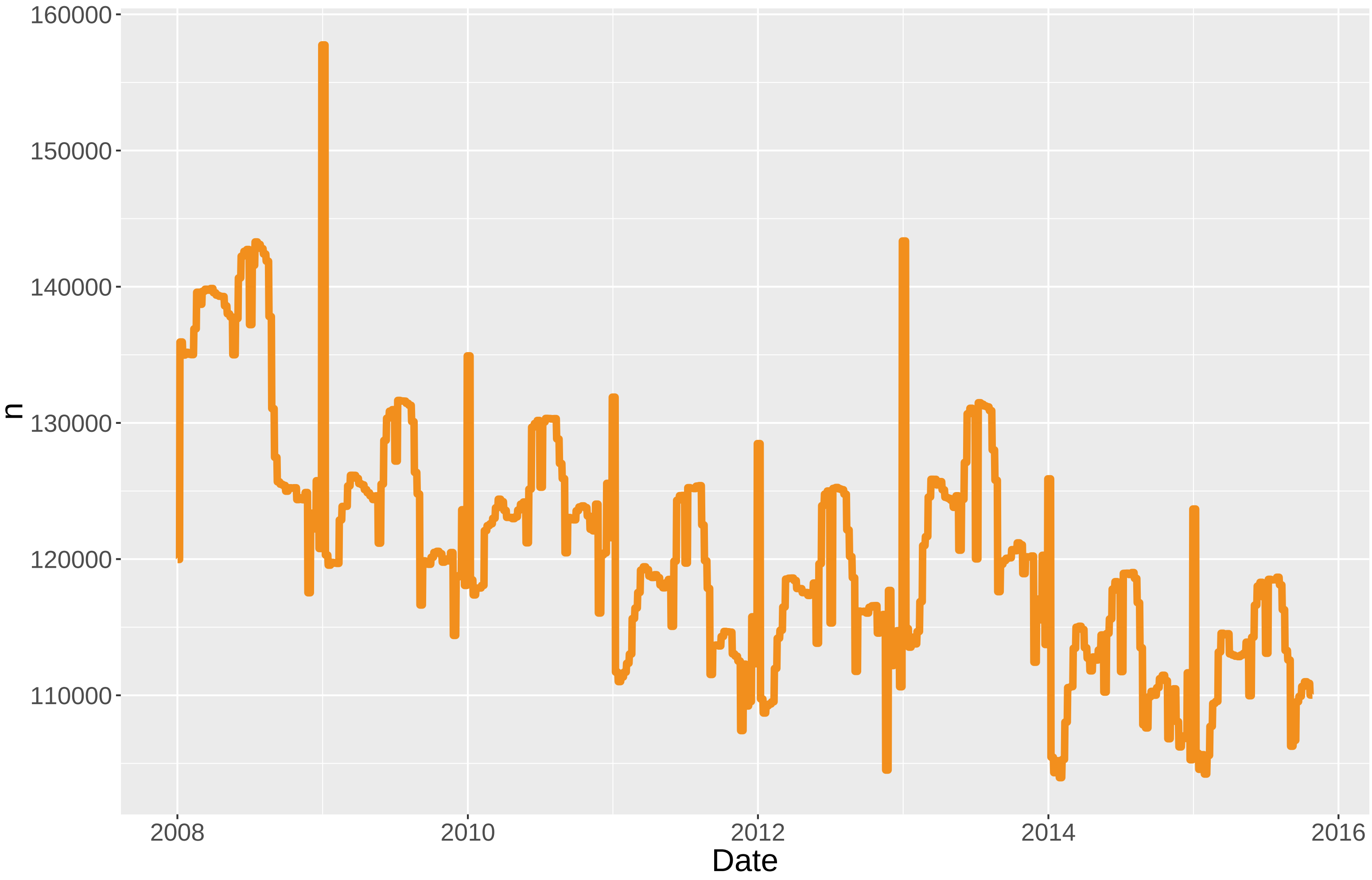
```
# Flights typically follow a weekly cycle, so we sho
# Pretty quick!
Sys.time()

## [1] "2016-02-05 21:34:54 AEDT"

setkey(flights, Year, Month, DayofMonth)
unique_dates <-
  unique(flights) %>%
```

```
  select(Year, Month, DayofMonth) %>%
  mutate(Week = (Year - 1987L) * 52 + data.table::yday(sprintf("%d-%02d-%
         Week = Week - min(Week))
flights <- flights[unique_dates]
Sys.time()

## [1] "2016-02-05 21:35:23 AEDT"
```

```
setkey(unique_dates, Week)
flights[,.(n = .N), keyby = Week][unique_dates]  %>%
  filter(Week < max(Week)) %>%
  mutate(Date = fastPOSIXct(paste0(Year, "-", Month, "-", DayofMonth))) %
  ggplot(aes(x = Date, y = n)) +
  geom_line(group = 1) +
  scale_y_continuous()
```

```
setkey(unique_dates, Week)
flights[,.(n = .N), keyby = Week][unique_dates]  %>%
  distinct(Week) %>%
  filter(Week < max(Week)) %>%
  mutate(difference = n - lag(n, 1, default = mean(.$n)),
         Date = fastPOSIXct(paste0(Year, "-", Month, "-", DayofMonth)),
         diff.lab = ifelse(ntile(difference, 100) == 100,
                           paste0(Year, "-", Month, "-", DayofMonth),
                           NA)) %>%
  ggplot(aes(x = Date, y = n)) +
  geom_line(group = 1, size = 2) +
  geom_point() +
  geom_text(aes(label = diff.lab)) +
  scale_y_continuous(label = comma)

## Warning:  Removed 403 rows containing missing values
(geom_text).
```

```
flights.by.week.and.carrier <-
  flights[,.(n = .N), by = list(Week, UniqueCarrier)]

biggest.carriers <-
  flights[,.(n = .N), by = UniqueCarrier][order(-n)] %>%
  filter(row_number(-n) <= 6) %$%
  UniqueCarrier

nycflights.airlines[,Carrier_other := ifelse(UniqueCarrier %in% biggest.c

flights.by.week.and.carrier.other <-
  flights.by.week.and.carrier %>%
  group_by(Week,
           Carrier_other = ifelse(UniqueCarrier %in% biggest.carriers, Un
  summarise(n = sum(n)) %>%
  merge(airlines, by.x = "Carrier_other", by.y = "carrier", all.x = TRUE)
  mutate(Carrier_other = factor(Carrier_other, levels = c(biggest.carrier

flights.by.week.and.carrier.other %>%
  convert_week_to_date %>%
  arrange(Date, Carrier_other) %>%
  ggplot(aes(x = Date, y = n, fill = Carrier_other)) +
  geom_area() +
  scale_y_continuous(label = scales::comma) +
```



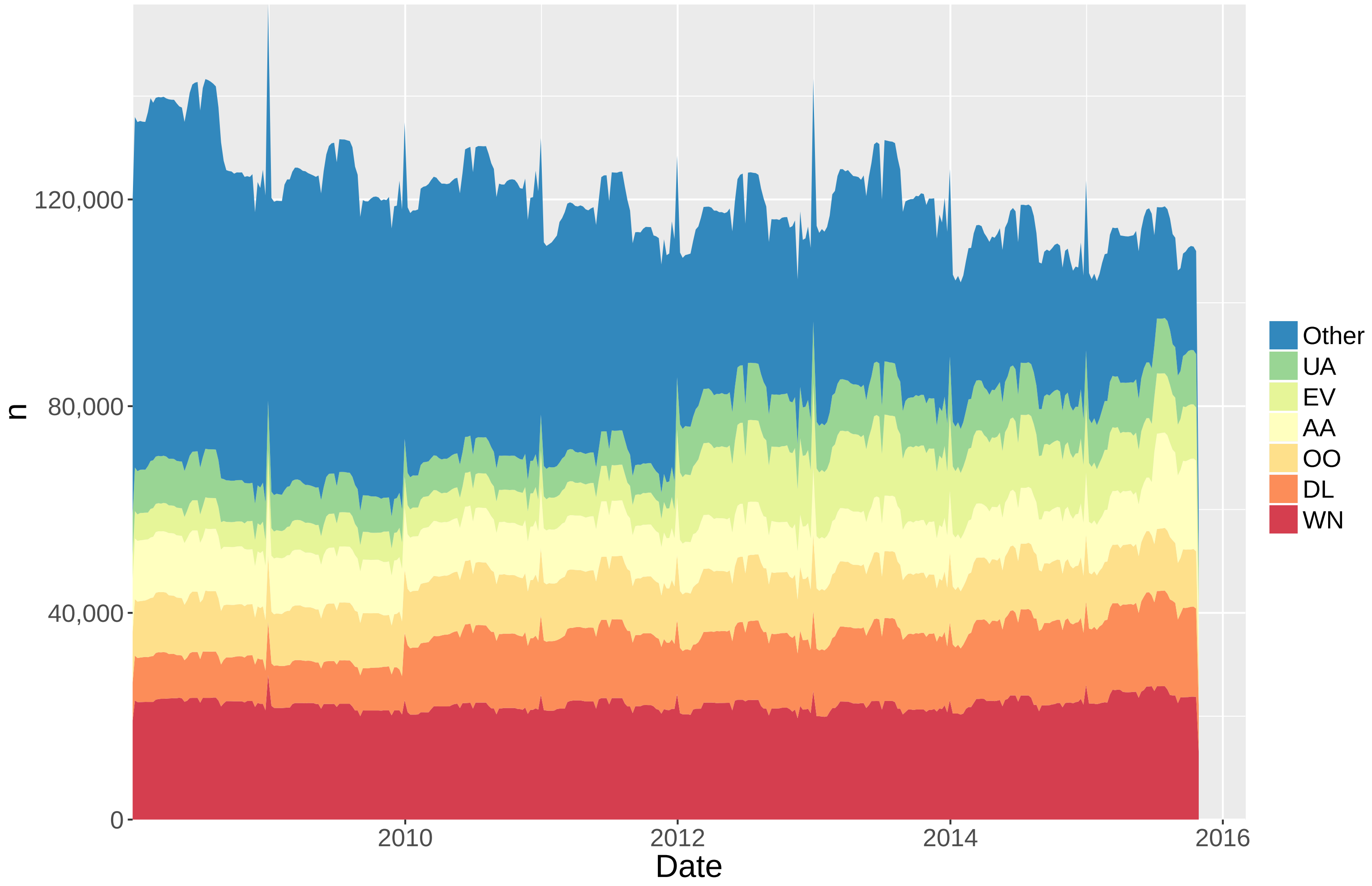
```
  scale_fill_brewer("", palette = "Spectral") +
  guides(fill = guide_legend(reverse = TRUE)) +
  annotate("blank", x = fastPOSIXct('2016-03-01'), y
  scale_x_datetime(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0), label = comma)
  theme(legend.position = "right")

## Scale for 'y' is already present.  Adding another
scale for 'y', which
## will replace the existing scale.
```

```
flights.by.week.and.carrier.other %>%
  group_by(Carrier_other) %>%
  mutate(r = n/first(n)) %>%
  filter(Week < max(Week)) %>%
  mutate(label.y = ifelse(Week == max(Week), r, NA_r
  convert_week_to_date %>%
  ggplot(aes(x = Date, y = r, color = Carrier_other,
  geom_line() +
  geom_dl(method = "last.qp", aes(label = ifelse(is.
#   geom_text(aes(y = label.y, label = name
#                  ), hjust = 0, nudge_x = 1) +
  #scale_color_brewer(palette = "Spectral") +
  guides(color = guide_legend(reverse = TRUE)) +
  annotate("blank", x = fastPOSIXct('2016-09-01'), y
  scale_x_datetime(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0), label = comma)
  theme(legend.position = "none")
```

```
cancellations.by.week <-
  flights %>%
  select(Week, Cancelled) %>%
  group_by(Week) %>%
  summarise(total_cancellations = sum(Cancelled))

cancellations.by.week %>%
  convert_week_to_date %>%
  ggplot(aes(x = Date, y = total_cancellations)) +
  geom_line(group = 1)
```

```
cancellations.by.month <-
  flights %>%
  select(Year, Month, Cancelled) %>%
  group_by(Year, Month) %>%
  summarise(total_cancellations = sum(Cancelled))

cancellations.by.month %>%
  ggplot(aes(x = Year + Month/12, y = total_cancellations)) +
  geom_line()                                       name)))) +
```

```
cancellations.by.year.carrier.other <-
  flights %>%
  select(Year, UniqueCarrier, Cancelled) %>%
  group_by(Year, UniqueCarrier) %>%
  summarise(total_cancellations = sum(Cancelled)) %>%
  setkey(UniqueCarrier) %>%
  .[nycflights.airlines] %>%
  group_by(Year, Carrier_other) %>%
  summarise(total_cancellations = sum(total_cancellations))
```



```
cancellations.by.year.carrier.other %>%
  mutate(Carrier_other_f = factor(Carrier_other, levels = c(biggest.carriers, "Other"))) %>%
  arrange(Year, Carrier_other_f) %>%
  ggplot(aes(x = Year, y = total_cancellations, fill
  geom_area() +
  guides(fill = guide_legend(reverse = TRUE)) +
  scale_fill_brewer(palette = "Spectral")
```

```
expected.cancellations.by.month <-
#   system.time({
#   flights %>%
#   select(Year, Month, UniqueCarrier, Cancelled) %>
#   group_by(Year, Month, Carrier_other = ifelse(Uni                   ther")) %>%
#   summarise(expected_cancellation = mean(Cancelled
#   })
# system.time({
#   flights[,Carrier_other := ifelse(UniqueCarrier %in% biggest.carriers, UniqueCarrier, "Other")] %>%
#   .[,.(expected_cancellation = mean(Cancelled)), by = list(Year, Month, Carrier_other)]})

  flights %>%
  select(Year, Month, UniqueCarrier, Cancelled) %>%
  # Get Carrier_other variable
    setkey(UniqueCarrier) %>%
    .[nycflights.airlines] %>%
  group_by(Year, Month, Carrier_other) %>%
  summarise(expected_cancellation = mean(Cancelled))
```

```
expected.cancellations.by.month %>%
  ggplot(aes(x = Year + Month/12, y = expected_cance                   er)) +
  geom_line()
```

```
expected.cancellations.by.week <-
  flights %>%
  select(Week, UniqueCarrier, Cancelled) %>%
  # Get Carrier_other variable
    setkey(UniqueCarrier) %>%
    .[nycflights.airlines] %>%
  group_by(Week, Carrier_other) %>%
  summarise(expected_cancellation = mean(Cancelled))

expected.cancellations.by.week %>%
  group_by(Week) %>%
  mutate(difference = expected_cancellation - mean(expected_cancellation)
  ggplot(aes(x = Week, y = difference)) +
  geom_area(group = 1) +
  facet_grid(Carrier_other ~ .)
```

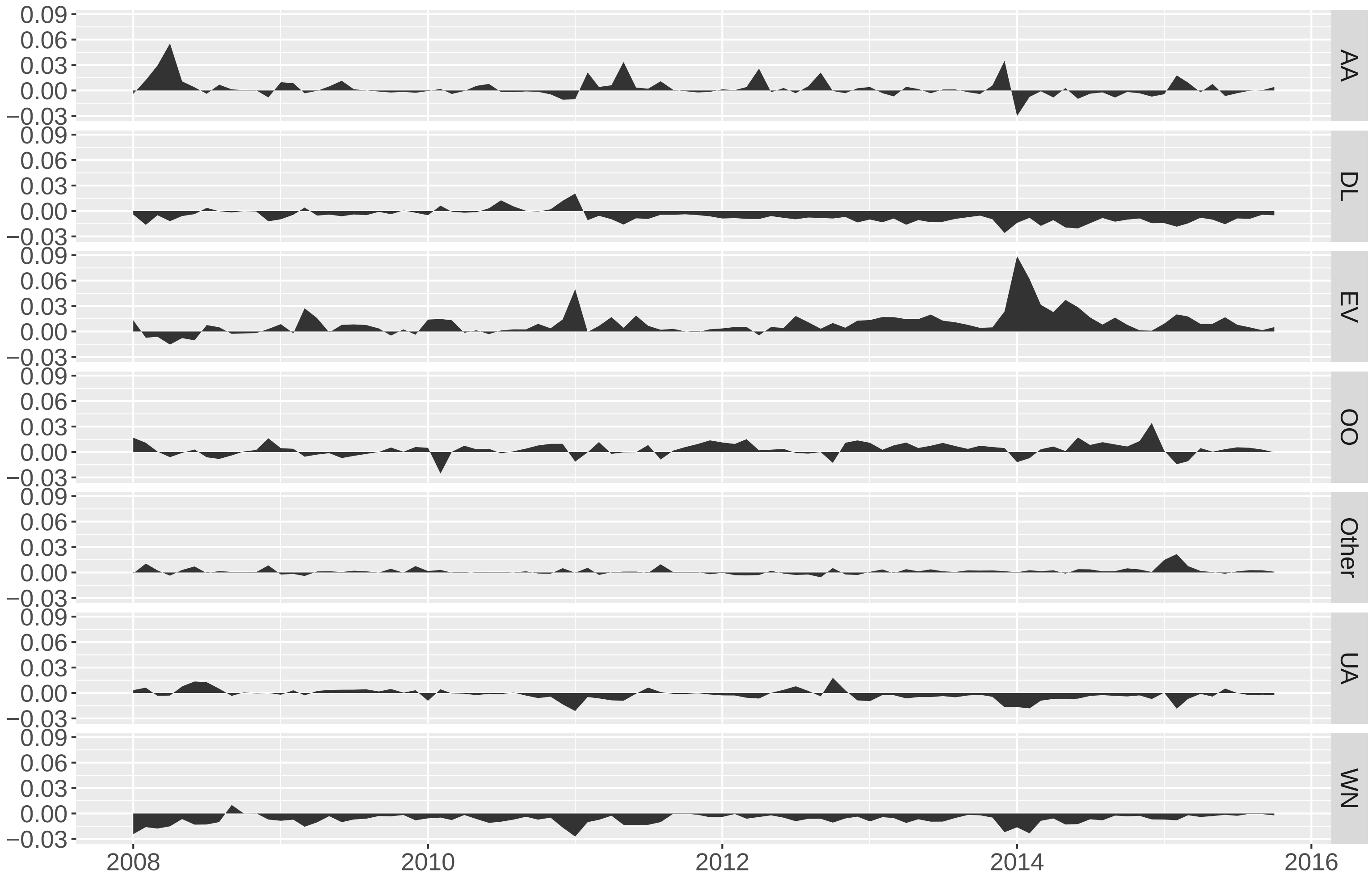
```
expected.cancellations.by.month %>%
  group_by(Year, Month) %>%
  mutate(difference = expected_cancellation - mean(expected_cancellation)
  ggplot(aes(x = as.Date(paste0(Year, "-", Month, "-01")), y = difference
  geom_area(group = 1) +
  facet_grid(Carrier_other ~ .) +
  theme(axis.title = element_blank())
```

```
ArrDelays.by.week <-
  flights %>%
  select(Week, ArrDelay) %>%
  group_by(Week) %>%
```

Figure 0.1: Southwest airlines (and Delta Air Lines from the start of 2011) have had consistently lower cancellation rates. ExpressJet has had substantially higher.

Figure 0.2: *

The difference of each airline's expected cancellation (cancellations per flight) from the average expected cancellation across all airlines, monthly.



```
  summarise(total_ArrDelay = sum(ArrDelay, na.rm = T

ArrDelays.by.week %>%
  ggplot(aes(Week, total_ArrDelay)) +
  geom_area(group = 1) +
  geom_hline(yintercept = 0, color = "black")
```

```
ArrDelays.by.month <-
  flights %>%
  select(Year, Month, ArrDelay) %>%
  group_by(Year, Month) %>%
  summarise(total_ArrDelay = sum(ArrDelay, na.rm = T

ArrDelays.by.month %>%
  ggplot(aes(as.Date(sprintf("%d-%02d-01", Year, Mon
  geom_area(group = 1) +
  geom_hline(yintercept = 0, color = "black")
```

```
ArrDelays.by.month %<>%
  ungroup %>%
  mutate(rel_delay = total_ArrDelay/mean(total_ArrDelay))

cancellations.by.month %<>%
  ungroup %>%
  mutate(rel_cancellations = total_cancellations / m

setkey(ArrDelays.by.month, Year, Month)
setkey(cancellations.by.month, Year, Month)
ArrDelays.by.month[cancellations.by.month] %>%
  select(Year, Month, starts_with("rel")) %>%
  melt.data.table(measure.vars = c("rel_delay", "rel
  ggplot(aes(as.Date(sprintf("%d-%02d-01", Year, Mon
  geom_bar(stat = "identity", position = "stack", wi
  theme(legend.position = "top")

## Warning:  Stacking not well defined when ymin !=
0
## Warning:  position_stack requires non-overlapping
x intervals
```

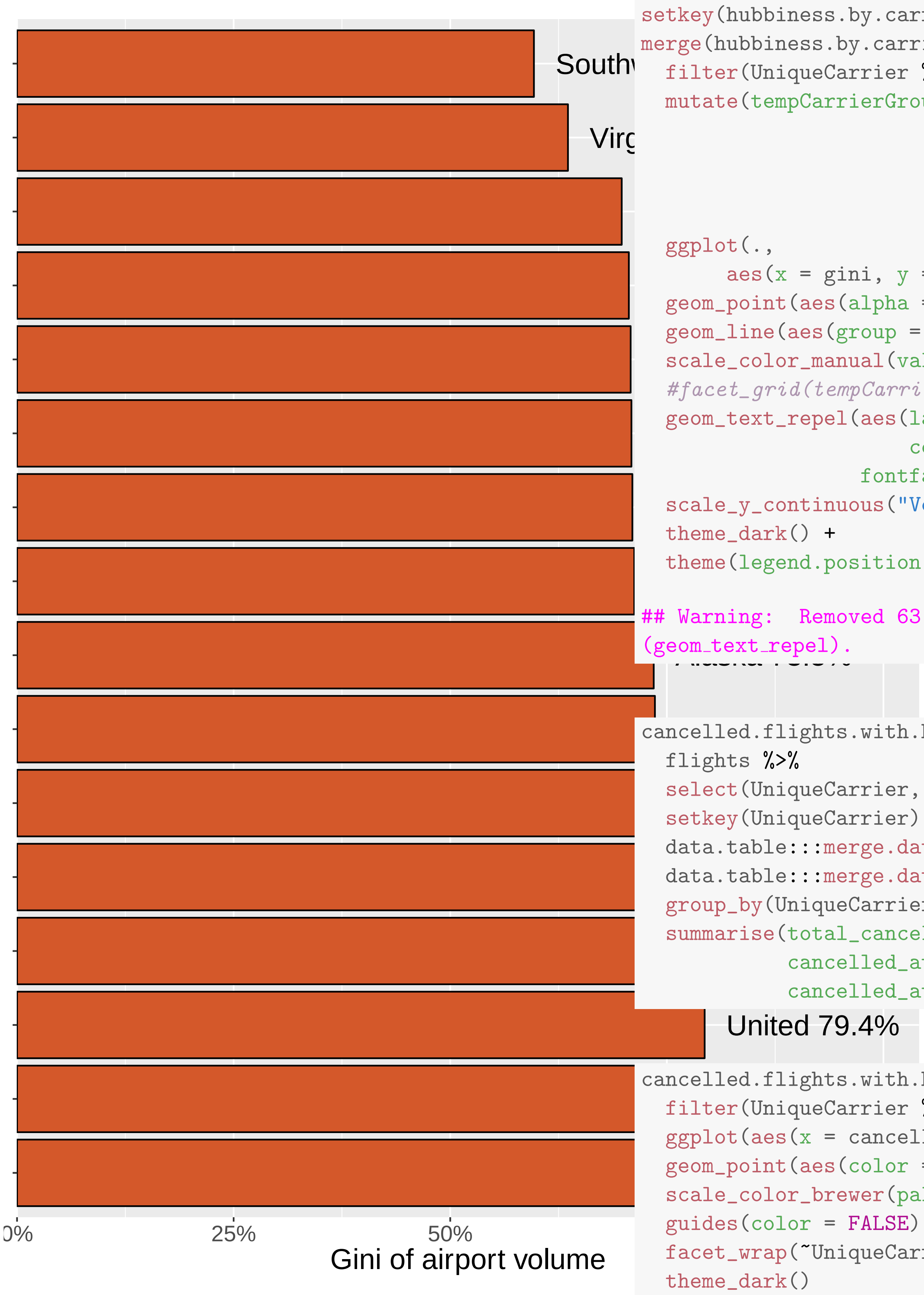
0.1 Which airport causes the most delays

```
# system.time({
# flights.by.origin <-
#   count(flights, Origin) %>%
#   arrange(desc(n))
# })
# 8 s.

flights.by.origin <-
    flights[,.(n = .N), by = Origin][order(-n)]
# 0.27s

flights.by.airport.carrier <-
#   flights %>%
#   count(Origin, UniqueCarrier) %>%
#   arrange(desc(n))
  flights[,.(n = .N), by = list(Origin, UniqueCarrier)][order(-n)]

hubs <-
  flights.by.airport.carrier %>%
  group_by(UniqueCarrier) %>%
  filter(n >= nth(n, order_by = -1*n, 2))

hub1.by.carrier <-
  hubs %>%
  group_by(UniqueCarrier) %>%
  filter(n == max(n)) %>%
  select(-n) %>%
  setnames("Origin", "Hub1") %>%
  setkey(UniqueCarrier)

hub2.by.carrier <-
  hubs %>%
  group_by(UniqueCarrier) %>%
  filter(n != max(n)) %>%
  select(-n) %>%
  setnames("Origin", "Hub2") %>%
  setkey(UniqueCarrier)
```

```
# Define hubbiness to be the Gini coefficient of each carrier.
hubbiness.by.carrier <-
  flights %>%
  select(UniqueCarrier, Origin) %>%
  group_by(UniqueCarrier, Origin) %>%
  tally() %>%
  ungroup %>%
  group_by(UniqueCarrier) %>%
  summarise(gini = ineq::Gini(n))                     ) %>%

hubbiness.by.carrier %>%
  ungroup %>%
  setkey(UniqueCarrier) %>%
  merge(nycflights.airlines) %>%
  ungroup %>%
  arrange(desc(gini)) %>%
  mutate(short_name = factor(short_name, levels = .$short_name)) %>%
  {
    ggplot(., aes(x = short_name, y = gini, order = gini)) +
      geom_bar(stat = "identity", width = 0.9) +
      coord_flip() +
      geom_text(aes(label = paste(short_name, percent(gini))), hjust = 0,
      theme(axis.title.y = element_blank(), axis.text.y = element_blank()
      scale_y_continuous("Gini of airport volume", expand = c(0,0), limit
  }
```

```
ggplot(hubbiness.by.carrier[flights.by.carrier][nycflights.airlines],
       aes(x = gini, y = n)) +
  geom_point(size = 2) +
  geom_text_repel(aes(label = short_name), fontface = "bold", size = 6) +
  scale_y_continuous("Volume (2008-2015)", labels = function(x)paste0(x/1
```

```
flights.by.carrier.year <-
  flights[,.(n = .N), by = list(Year, UniqueCarrier)]
setkey(flights.by.carrier.year, Year, UniqueCarrier)
```

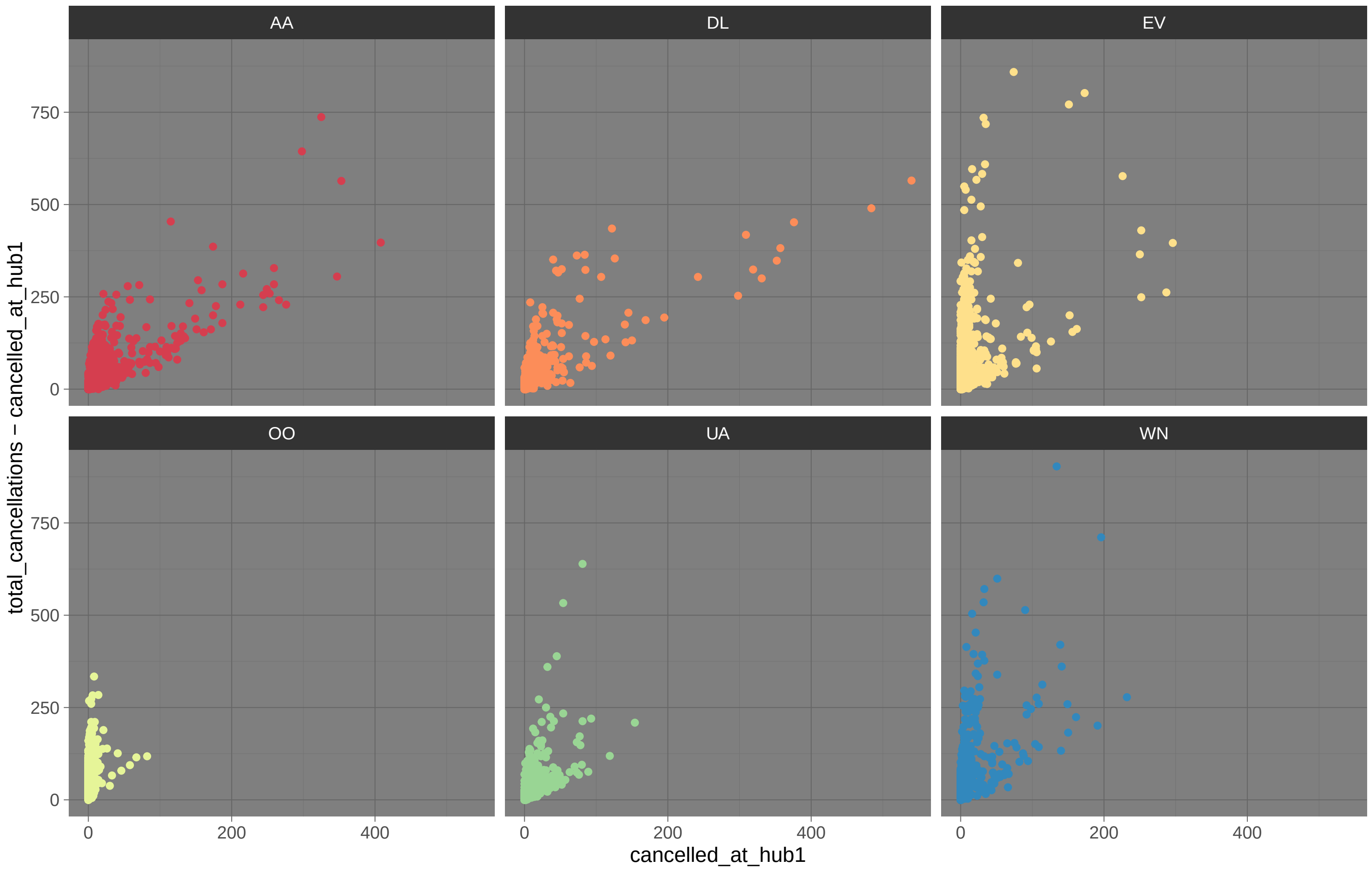


```
hubbiness.by.carrier.year <-
  flights %>%
  select(Year, UniqueCarrier, Origin) %>%
  count(Year, UniqueCarrier, Origin) %>%
  group_by(Year, UniqueCarrier) %>%
  summarise(gini = ineq::Gini(n)) %>%
  setkey(Year, UniqueCarrier)

setkey(hubbiness.by.carrier.year, Year, UniqueCarrier)
merge(hubbiness.by.carrier.year[flights.by.carrier.year], nycflights.airl
  filter(UniqueCarrier %in% select_large_carriers(9)) %>%
  mutate(tempCarrierGroup = factor(ifelse(UniqueCarrier == "WN",
                                          1,
                                          ifelse(UniqueCarrier %in% selec
                                                 2,
                                                 3)))) %>%
  ggplot(.,
       aes(x = gini, y = n)) +
  geom_point(aes(alpha = Year, color = UniqueCarrier), size = 4) +
  geom_line(aes(group = UniqueCarrier, color = UniqueCarrier), size = 1)
  scale_color_manual(values = carrier.colors) +
  #facet_grid(tempCarrierGroup~.) +
  geom_text_repel(aes(label = ifelse(Year == max(Year), short_name, NA_ch
                      color = UniqueCarrier),
                  fontface = "bold", size = 6) +
  scale_y_continuous("Volume", labels = function(x)paste0(x/1e6, "M")) +
  theme_dark() +
  theme(legend.position = "none")

## Warning:  Removed 63 rows containing missing values
(geom_text_repel).
```

```
cancelled.flights.with.hub.cancelled <-
  flights %>%
  select(UniqueCarrier, Origin, Year, Month, DayofMonth, Cancelled) %>%
  setkey(UniqueCarrier) %>%
  data.table:::merge.data.table(hub1.by.carrier) %>%
  data.table:::merge.data.table(hub2.by.carrier)  %>%
  group_by(UniqueCarrier, Year, Month, DayofMonth) %>%
  summarise(total_cancellations = sum(Cancelled),
            cancelled_at_hub1 = sum(Cancelled * (Origin == Hub1)),
            cancelled_at_hub2 = sum(Cancelled * (Origin == Hub2)))
```

```
cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  ggplot(aes(x = cancelled_at_hub1, y = total_cancellations - cancelled_a
  geom_point(aes(color = UniqueCarrier)) +
  scale_color_brewer(palette = "Spectral") +
  guides(color = FALSE) +
  facet_wrap(~UniqueCarrier) +
  theme_dark()
```

```
cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  ggplot(aes(x = cancelled_at_hub1, y = total_cancellations - cancelled_a
  geom_point(aes(color = UniqueCarrier), alpha = 0.25) +
  scale_color_brewer(palette = "Spectral") +
  guides(color = FALSE) +
```



```
  scale_x_log10() + scale_y_log10() +
  facet_wrap(~UniqueCarrier, scales = "free") +
  theme_dark()
```

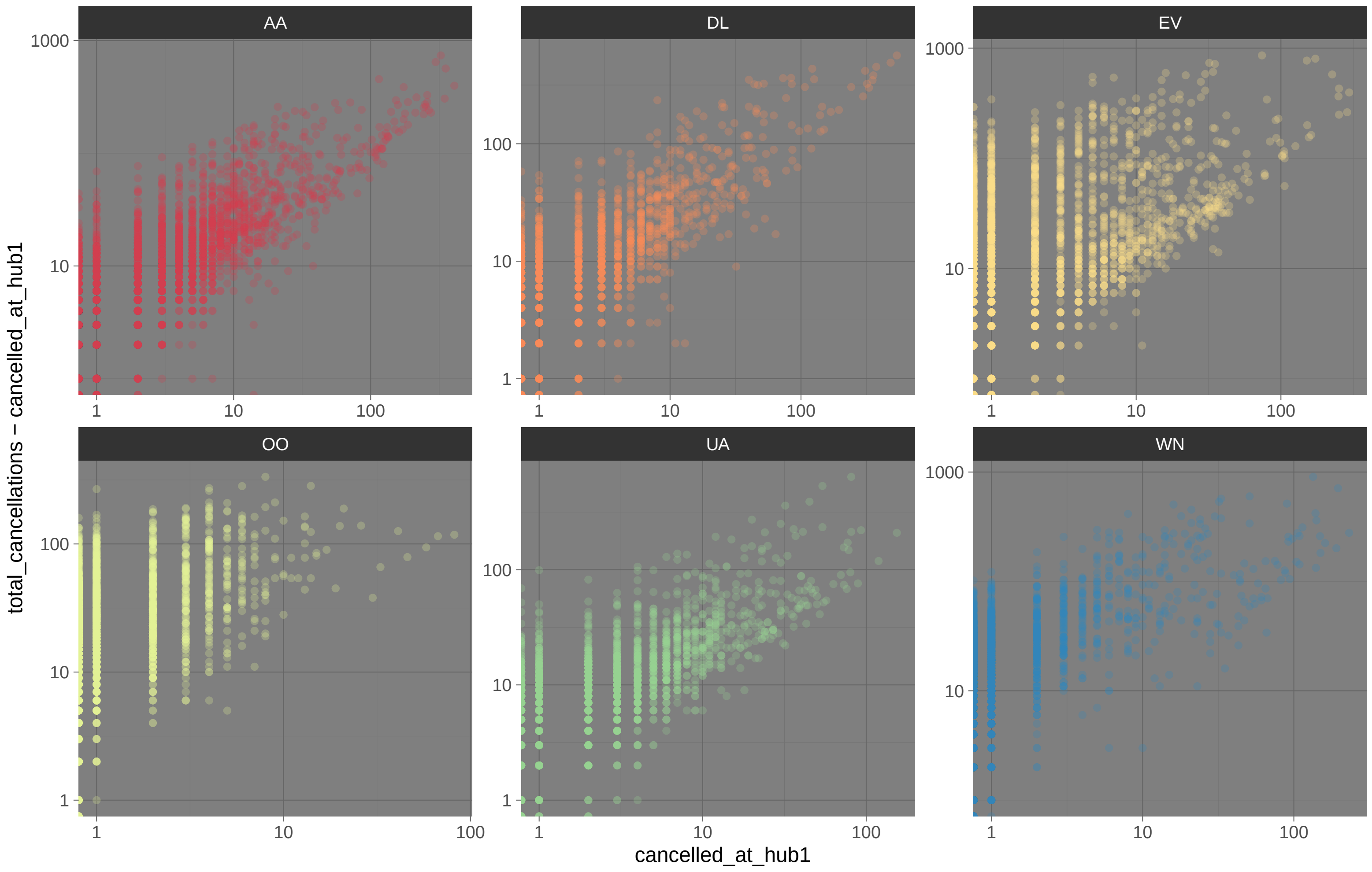
```
cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  ggplot(aes(x = cancelled_at_hub2, y = total_cancel
  geom_point(aes(color = UniqueCarrier), alpha = 0.25) +
  scale_color_brewer(palette = "Spectral") +
  guides(color = FALSE) +
  scale_x_log10() + scale_y_log10() +
  facet_wrap(~UniqueCarrier, scales = "free") +
  theme_dark()
```

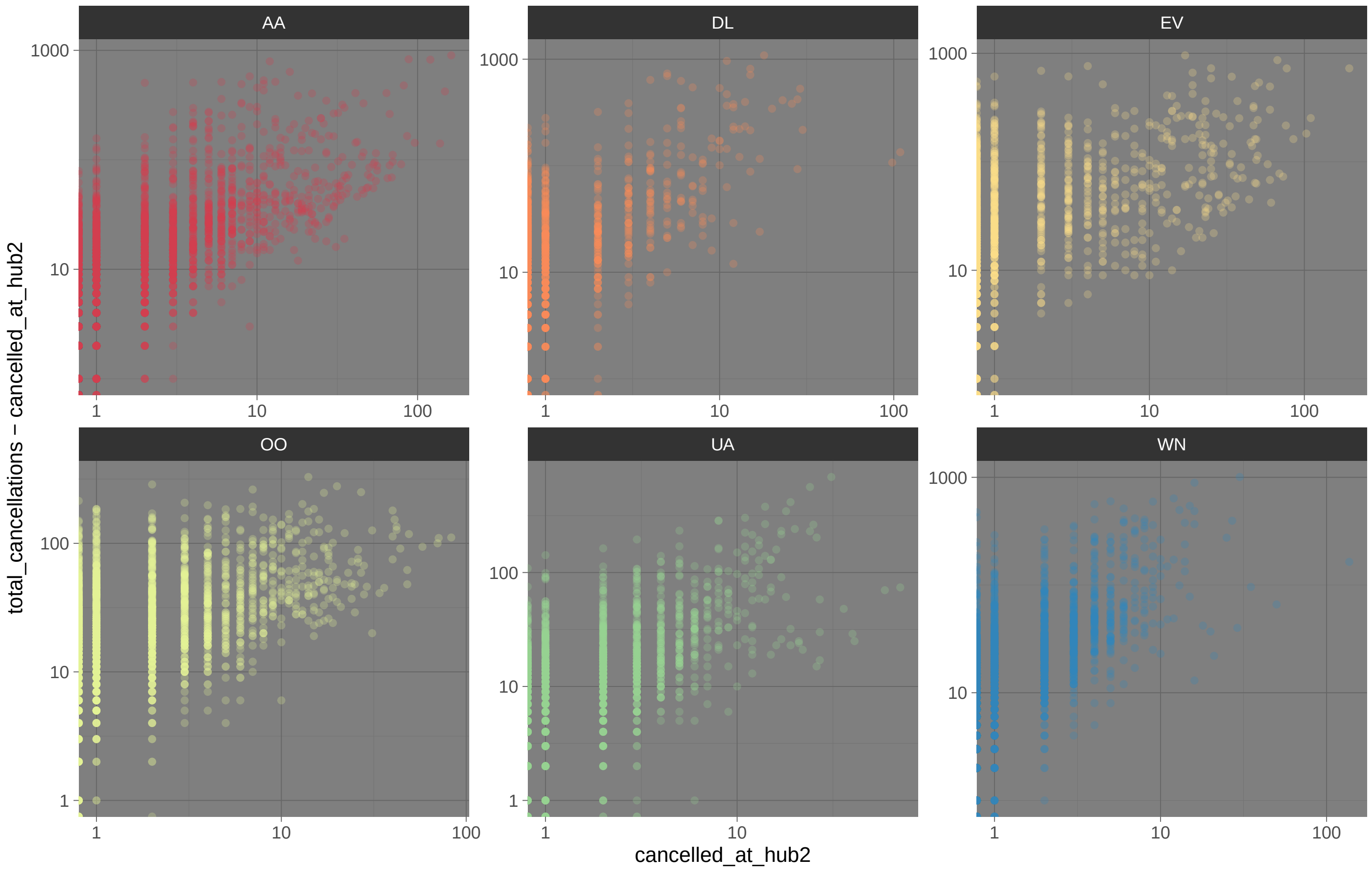
```
cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  group_by(Year, Month, DayofMonth) %>%
  mutate(cancelled_at_hub1_rel_other_hubs = cancelle
         cancelled_rel_other_carriers = total_cancel
  ggplot(aes(x = cancelled_at_hub1_rel_other_hubs, y
  geom_point(aes(color = UniqueCarrier), alpha = 0.2
  scale_color_brewer(palette = "Spectral") +
  guides(color = FALSE) +
  facet_wrap(~UniqueCarrier, scales = "free") +
  theme_dark()
```

```
cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  group_by(Year, Month, DayofMonth) %>%
```

```
  mutate(cancelled_at_hub1_rel_other_hubs = cancelled_at_hub1 - mean(canc
         cancelled_outside_hub_rel_other_carriers = (total_cancellations
  ggplot(aes(x = cancelled_at_hub1_rel_other_hubs, y = cancelled_outside_
  geom_point(aes(color = UniqueCarrier), alpha = 0.25) +
  scale_color_brewer(palette = "Spectral") +
  guides(color = FALSE) +
  facet_wrap(~UniqueCarrier, scales = "free") +
  theme_dark()
```

```
ArrDelays.by.day <-
  flights %>%
  select(Year, Month, DayofMonth, ArrDelay) %>%
  group_by(Year, Month, DayofMonth) %>%
  summarise(total_ArrDelay_allcarriers = sum(ArrDelay, na.rm = TRUE)) %>%
  setkey(Year, Month, DayofMonth)

ArrDelays.avg.by.day <-
  flights %>%
  select(Year, Month, DayofMonth, ArrDelay) %>%
  group_by(Year, Month, DayofMonth) %>%
  summarise(avg_ArrDelay_allcarriers = sum(ArrDelay, na.rm = TRUE)/n()) %
  setkey(Year, Month, DayofMonth)

dates.arrdelay.rel.hub <-
  flights %>%
  select(Year, Month, DayofMonth, UniqueCarrier, Origin, ArrDelay) %>%
  setkey(UniqueCarrier) %>%
  data.table:::merge.data.table(hub1.by.carrier) %>%
  group_by(Year, Month, DayofMonth, UniqueCarrier) %>%
  summarise(total_arrdelay = sum(ArrDelay, na.rm = TRUE),
            arrdelay_at_hub = sum(ArrDelay * (Origin == Hub1), na.rm = TR
```


AA
DL
EV
OO
UA
WN
total_cancellations − cancelled_at_hub1
cancelled_at_hub1
1000
100
10
1
1
10
100


AA
DL
EV
OO
UA
WN
total_cancellations − cancelled_at_hub2
cancelled_at_hub2
1000
100
10
1
1
10
100

```
            arrdelay_not_at_hub = sum(ArrDelay * (Or
  setkey(Year, Month, DayofMonth) %>%
  data.table:::merge.data.table(ArrDelays.by.day)

dates.avg.arrdelay.rel.hub <-
  flights %>%
  select(Year, Month, DayofMonth, UniqueCarrier, Ori
  setkey(UniqueCarrier) %>%
  data.table:::merge.data.table(hub1.by.carrier) %>%
  group_by(Year, Month, DayofMonth, UniqueCarrier) %
  summarise(avg_arrdelay = sum(ArrDelay, na.rm = TRU
            avg_arrdelay_at_hub = sum(ArrDelay * (Or
            avg_arrdelay_not_at_hub = sum(ArrDelay *
  setkey(Year, Month, DayofMonth) %>%
  data.table:::merge.data.table(ArrDelays.avg.by.day


dates.arrdelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)
  ggplot(aes(x = total_arrdelay, y = arrdelay_at_hub
  geom_point(alpha = 0.33) +
  facet_wrap(~UniqueCarrier) +
  theme_dark() +
  scale_color_brewer(palette = "Spectral")
```

```
dates.avg.arrdelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)
  merge(nycflights.airlines, by = "UniqueCarrier") %
  ggplot(aes(x = avg_arrdelay_at_hub, y = avg_arrdel
  geom_point(alpha = 0.33) +
  facet_wrap(~short_name) +
  theme_dark() +
  scale_color_brewer(palette = "Spectral", guide = F
```

```
dates.avg.arrdelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)
  merge(nycflights.airlines, by = "UniqueCarrier") %
  ggplot(aes(x = avg_arrdelay_at_hub, y = avg_arrdel
  geom_point(alpha = 0.33) +
  facet_wrap(~short_name, scales = "free") +
  theme_dark() +
  scale_color_brewer(palette = "Spectral", guide = F
```

```
dates.avg.arrdelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)
  merge(nycflights.airlines, by = "UniqueCarrier") %
  ggplot(aes(x = avg_arrdelay_at_hub - avg_ArrDelay_
  geom_point(alpha = 0.33) +
  facet_wrap(~short_name) +
  theme_dark() +
  geom_abline(slope = 1, color = "white") +
  scale_color_brewer(palette = "Spectral", guide = F
```

```
dates.avg.arrdelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)
  merge(nycflights.airlines, by = "UniqueCarrier") %
  ggplot(aes(x = avg_arrdelay_at_hub - avg_ArrDelay_
```

```
  geom_point(alpha = 0.33) +
  facet_wrap(~short_name, scales = "free") +
  theme_dark() +
  scale_color_manual(values = carrier.colors)
```

```
city.market.decoder <- fread("../metadata/L_CITY_MARKET_ID.csv", verbose

## Input contains no \n. Taking this to be a filename to open
## File opened, filesize is 0.000153 GB.
## Memory mapping ... ok
## Detected eol as \r\n (CRLF) in that order, the Windows standard.
## Positioned on line 1 after skip or autostart    ›1)) %>%
## This line is the autostart and not blank so searching up for the last
## Detecting sep ... ','
## Detected 2 columns. Longest stretch was from line 1 to line 30
## Starting data input on line 1 (either column names or first row of dat
## All the fields on line 1 are character fields. Treating as the column
## Count of eol: 5750 (including 1 at the end)
## Count of sep: 11506
## nrow = MIN( nsep [11506] / ncol [2] -1, neol [5750] - nblank [1] ) = 5
## Type codes (   first 5 rows): 44
## Type codes (+ middle 5 rows): 44
## Type codes (+   last 5 rows): 44

## Warning in fread("../metadata/L_CITY_MARKET_ID.csv",
verbose = TRUE, colClasses = c("integer", :  Column
1 ('Code') has been detected as type 'character'.  Ignoring
request from colClasses to read as 'integer' (a lower
type) since NAs (or loss of precision) may result.

## Type codes: 44 (after applying colClasses and integer64)
## Type codes: 44 (after applying drop or select (if supplied)
## Allocating 2 column slots (2 - 0 dropped)
## Read 5749 rows. Exactly what was estimated and allocated up front
##    0.010s ( 62%) Memory map (rerun may be quicker)
##    0.000s (  0%) sep and header detection
##    0.001s (  6%) Count rows (wc -l)
##    0.000s (  0%) Column type detection (first, middle and last 5 rows)
##    0.001s (  6%) Allocation of 5749x2 result (xMB) in RAM
##    0.004s ( 25%) Reading data
##    0.000s (  0%) Allocation for type bumps (if any), including gc time
##    0.000s (  0%) Coercing data already read in type bumps (if any)
##    0.000s (  0%) Changing na.strings to NA
##    0.016s        Total

city.market.decoder[,Code := as.integer(Code)]
city.market.volumes <-
  flights %>%
  select(OriginCityMarketID) %>%
  count(OriginCityMarketID) %>%                      ›lay_allcarriers, colo
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code")
  arrange(n)
```

```
city.market.volumes.2014 <-
  flights[Year == 2014, .(n = .N), by = OriginCityMarketID] %>%
  filter(n >= nth(n, 8, order_by = -n)) %>%
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code")
  arrange(desc(n))

flights[,.(n = .N), by = list(Year, OriginCityMarketID)] %>%arriers, colo
```

```
  merge(city.market.decoder, by.x = "OriginCityMarke
  group_by(Year) %>%
  filter(n >= nth(n, 8, order_by = -n)) %>%
  tbl_df %>%
  mutate(Description = factor(Description)) %>%
  mutate(Description = factor(Description,
                              levels = city.market.v
                              labels = gsub(", [A-Z]                              ) %>%
  filter(Year < max(Year)) %>%
  {
  ggplot(., aes(x = Year, y = n, group = Description
  geom_line() +
  geom_dl(method = "last.points", aes(label = Descri
  scale_color_brewer(palette = "Spectral") +
  theme(legend.position = "none") +
  scale_x_continuous(limits = c(min(.$Year), max(.$Y
  }
```

```
city.market.volumes.2014 <-
  flights[Year == 2014, .(n = .N), by = OriginCityMa
  filter(n >= nth(n, 8, order_by = -n)) %>%
  merge(city.market.decoder, by.x = "OriginCityMarke
  arrange(desc(n))

flights[,.(n = .N), by = list(Week, OriginCityMarketID)] %>%
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code") %>%
  group_by(Week) %>%
  filter(OriginCityMarketID %in% city.market.volumes
  tbl_df %>%
  mutate(Description = factor(Description)) %>%
  mutate(Description = factor(Description,
                              levels = city.market.v
                              labels = gsub(", [A-Z]

  filter(Week < max(Week) & Week > min(Week)) %>%
  mutate(Description.label = ifelse(Week == max(Week
  {
    ggplot(., aes(x = Week, y = n, group = Descripti
      geom_line() +
      # geom_text_repel(aes(x = Week, label = Descri
      # geom_dl(method = "last.points", aes(label =
      geom_text(aes(label = Description.label), hjus
      scale_color_brewer(palette = "Spectral") +
      theme(legend.position = "none") +
      annotate("blank", x = max(.$Week) + 50, y = me
      theme_dark()

  }

## Warning:  Removed 3240 rows containing missing va
(geom_text).
```

```
flights %>%
  select(tempkey, OriginCityMarketID, DestCityMarket
  .[, Corridor := pmin(paste0(OriginCityMarketID, "-
                       paste0(DestCityMarketID, "-",
  select(tempkey, Corridor) %>%
  setkey(tempkey) %>%
  saveRDS(file = "flights-with-corridor.rds", compre
```

```
corridor.volumes.by.week <-
  flights[, Corridor := pmin(paste0(OriginCityMarketID, "-", DestCityMark
                             paste0(DestCityMarketID, "-", OriginCityMark
```

```
flights.with.corridor <-
  readRDS("flights-with-corridor.rds")

## Warning in gzfile(file, "rb"):  cannot open compressed
file 'flights-with-corridor.rds', probable reason 'No
such file or directory'
## Error in gzfile(file, "rb"):  cannot open the connection

flights <-
  flights %>%
  setkey(tempkey) %>%
  .[flights.with.corridor]

## Error in eval(expr, envir, enclos):  object 'flights.with.corridor'
not found

corridor.volumes.by.week <- flights[,(.n = .N), by = list(Week, Corridor)

## Error in eval(expr, envir, enclos):  object 'Corridor'
not found
```

```
if (!useRDS){
  Corridors <-
  data.table::CJ(OriginCityMarketID = city.market.decoder$Code,
                 DestCityMarketID = city.market.decoder$Code) %>%
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code")
  setnames("Description", "OriginCityMarketID_DS") %>%
  merge(city.market.decoder, by.x = "DestCityMarketID", by.y = "Code") %>
  setnames("Description", "DestCityMarketID_DS")

Corridors[,Corridor := paste0(OriginCityMarketID, "-", DestCityMarketID)]
Corridors[,Corridor_DS := paste0(OriginCityMarketID_DS, "-", DestCityMark
Corridors %<>% select(Corridor, Corridor_DS) %>% setkey(Corridor)
gc(T,T)
} else {
  Corridors <- readRDS("Corridors.rds")
}

## Warning in gzfile(file, "rb"):  cannot open compressed
file 'Corridors.rds', probable reason 'No such file
or directory'
## Error in gzfile(file, "rb"):  cannot open the connection
```

```
corridor.volumes <-
  corridor.volumes.by.week %>%
  group_by(Corridor) %>%
  summarise(total_volume = sum(n)) %>%
  arrange(desc(total_volume))

## Error in eval(expr, envir, enclos):  object 'corridor.volumes.by.week'
not found

corridor.volumes.by.week %>%
  filter(Corridor %in% corridor.volumes$Corridor[1:10]) %>%
  setkey(Corridor) %>%
  merge(Corridors) %>%
  mutate(Corridor_DS_x = gsub("ˆ([A-Z].+),.*-([A-Z].+),.*$", "\\1-\\2", Corridor_DS)) %>%
  filter(Week < max(Week)) %>%
  group_by(Corridor) %>%
  mutate(maxn = max(n)) %>%
  ungroup %>%
  mutate(Facet = rank(maxn) %% 5) %>%
  ggplot(aes(x = Week, y = n, group = Corridor_DS_x, color = Corridor_DS_x)) +
  geom_line() +
  scale_x_continuous(limits = c(0, 450)) +
  geom_dl(method = "last.points", aes(label = Corridor_DS_x)) +
  facet_grid(Facet ˜ .)
```

```
## Error in eval(expr, envir, enclos):  object ’corridor.volumes.by.week’
not found
```

COMPILATION TIME: 5.64845073223114

```
COMPILATION.TIME <- round(difftime(Sys.time(), START.TIME, units = "mins"), 1)
write("=======",
file = "analysis-post-2008-CHUNKTIMINGS.txt",
append = TRUE)
write(paste0("Compilation time: ", COMPILATION.TIME),
      file = "analysis-post-2008-CHUNKTIMINGS.txt",
      append = TRUE)
finished <- TRUE
```