

```

library(knitr)
.finished <- FALSE
knit_hooks$set(timeit = function(before) {
  if (before) {
    .current.time <- Sys.time()
  } else {
    .duration <- round(difftime(Sys.time(), .current.time), digits = 2)
    if(!.finished)
      write(
        paste0(
          knitr::opts_current$get(name = "label"),
          ": ",
          .duration),
        file = "analysis-post-2008-CHUNKTIMINGS.txt",
        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-post-2008-CHUNKTIMINGS.png",
  timeit = TRUE,
  cache=FALSE,
  out.width = "11in")

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.bitwhich
## operator: ! & | xor != == < <= > >=
## querying: print length any all min max range sum
## bit access: length<- [ [<- [[ [[<-
## for more help type ?bit
##
## Attaching package: 'bit'
##
## The following object is masked from 'package:data.table':
##
##      setattr
##
## The following object is masked from 'package:base':
##
##      xor
##
## Attaching package bit64
## package:bit64 (c) 2011-2012 Jens Oehlschlaegel (C)
## 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)

```

<code>library(nycflights13) # for airports</code>	Read 15.1% of 49153341 rows
<code>nycflights.airports <- airports</code>	Read 15.8% of 49153341 rows
<code>nycflights.planes <- planes</code>	Read 16.5% of 49153341 rows
<code>nycflights.airlines <- as.data.table(airlines)</code>	Read 17.2% of 49153341 rows
<code>for (j in 1:ncol(nycflights.airlines)){</code>	Read 17.9% of 49153341 rows
<code>set(nycflights.airlines, j = j, value = as.character(airlines[,j]))</code>	Read 18.6% of 49153341 rows
<code>}</code>	Read 19.3% of 49153341 rows
<code>nycflights.airlines[,short_name := gsub("\\s.*\$", "", short_name)]</code>	Read 20.0% of 49153341 rows
<code>setnames(nycflights.airlines, "carrier", "UniqueCarrier")</code>	Read 20.6% of 49153341 rows
<code>setkey(nycflights.airlines, UniqueCarrier)</code>	Read 21.3% of 49153341 rows
<code>library(fasttime)</code>	Read 22.0% of 49153341 rows
<code>library(grattan)</code>	Read 22.7% of 49153341 rows
 	Read 23.4% of 49153341 rows
<code>##</code>	Read 24.1% of 49153341 rows
<code>## Attaching package: 'grattan'</code>	Read 24.8% of 49153341 rows
<code>##</code>	Read 25.5% of 49153341 rows
<code>## The following object is masked from 'package:data.table': 'setkey'</code>	Read 26.2% of 49153341 rows
<code>##</code>	Read 26.9% of 49153341 rows
<code>## Orange</code>	Read 27.6% of 49153341 rows
 	Read 28.3% of 49153341 rows
<code>library(directlabels)</code>	Read 29.0% of 49153341 rows
<code>library(ineq) # for Gini()</code>	Read 29.6% of 49153341 rows
 	Read 30.3% of 49153341 rows
<code>convert_week_to_date <- function(DT_with_Week_column)</code>	Read 31.0% of 49153341 rows
<code>stopifnot(is.data.table(DT_with_Week_column), "Week column is not a data table")</code>	Read 31.7% of 49153341 rows
<code>setkey(DT_with_Week_column, Week)</code>	Read 32.4% of 49153341 rows
<code>temp <-</code>	Read 33.1% of 49153341 rows
<code>unique_dates %>%</code>	Read 33.8% of 49153341 rows
<code>group_by(Week) %>%</code>	Read 34.5% of 49153341 rows
<code>summarise(Date = fastPOSIXct(sprintf("%d-%02d-%02d", Week, 1, 1)))</code>	Read 35.2% of 49153341 rows
<code>setkey(Week)</code>	Read 35.8% of 49153341 rows
<code>DT_with_Week_column[temp]</code>	Read 36.5% of 49153341 rows
<code>}</code>	Read 37.2% of 49153341 rows
 	Read 37.9% of 49153341 rows
<code>flights <- fread("../post2008_flights.csv", na.strings = c("NA", "N/A", "None"))</code>	Read 38.6% of 49153341 rows
 	Read 39.3% of 49153341 rows
<code>##</code>	Read 40.0% of 49153341 rows
Read 0.0% of 49153341 rows	Read 40.7% of 49153341 rows
Read 0.7% of 49153341 rows	Read 41.4% of 49153341 rows
Read 1.4% of 49153341 rows	Read 42.1% of 49153341 rows
Read 2.0% of 49153341 rows	Read 42.7% of 49153341 rows
Read 2.7% of 49153341 rows	Read 43.4% of 49153341 rows
Read 3.4% of 49153341 rows	Read 44.1% of 49153341 rows
Read 4.1% of 49153341 rows	Read 44.8% of 49153341 rows
Read 4.8% of 49153341 rows	Read 45.5% of 49153341 rows
Read 5.5% of 49153341 rows	Read 46.2% of 49153341 rows
Read 6.2% of 49153341 rows	Read 46.9% of 49153341 rows
Read 6.9% of 49153341 rows	Read 47.6% of 49153341 rows
Read 7.5% of 49153341 rows	Read 48.3% of 49153341 rows
Read 8.2% of 49153341 rows	Read 49.0% of 49153341 rows
Read 8.9% of 49153341 rows	Read 49.7% of 49153341 rows
Read 9.6% of 49153341 rows	Read 50.3% of 49153341 rows
Read 10.3% of 49153341 rows	Read 51.0% of 49153341 rows
Read 11.0% of 49153341 rows	Read 51.7% of 49153341 rows
Read 11.7% of 49153341 rows	Read 52.4% of 49153341 rows
Read 12.3% of 49153341 rows	Read 53.1% of 49153341 rows
Read 13.0% of 49153341 rows	Read 53.8% of 49153341 rows
Read 13.7% of 49153341 rows	Read 54.5% of 49153341 rows
Read 14.4% of 49153341 rows	Read 55.2% of 49153341 rows
	Read 55.9% of 49153341 rows
	Read 56.6% of 49153341 rows

```
Read 57.2% of 49153341 rows
Read 57.9% of 49153341 rows
Read 58.6% of 49153341 rows
Read 59.3% of 49153341 rows
Read 60.0% of 49153341 rows
Read 60.7% of 49153341 rows
Read 61.4% of 49153341 rows
Read 62.1% of 49153341 rows
Read 62.8% of 49153341 rows
Read 63.5% of 49153341 rows
Read 64.2% of 49153341 rows
Read 64.9% of 49153341 rows
Read 65.5% of 49153341 rows
Read 66.2% of 49153341 rows
Read 66.9% of 49153341 rows
Read 67.6% of 49153341 rows
Read 68.3% of 49153341 rows
Read 69.0% of 49153341 rows
Read 69.7% of 49153341 rows
Read 70.4% of 49153341 rows
Read 71.0% of 49153341 rows
Read 71.7% of 49153341 rows
Read 72.4% of 49153341 rows
Read 73.1% of 49153341 rows
Read 73.8% of 49153341 rows
Read 74.5% of 49153341 rows
Read 75.2% of 49153341 rows
Read 75.9% of 49153341 rows
Read 76.6% of 49153341 rows
Read 77.3% of 49153341 rows
Read 78.0% of 49153341 rows
Read 78.7% of 49153341 rows
Read 79.3% of 49153341 rows
Read 80.0% of 49153341 rows
Read 80.7% of 49153341 rows
Read 81.4% of 49153341 rows
Read 82.1% of 49153341 rows
Read 82.8% of 49153341 rows
Read 83.5% of 49153341 rows
Read 84.2% of 49153341 rows
Read 84.9% of 49153341 rows
Read 85.5% of 49153341 rows
Read 86.2% of 49153341 rows
Read 86.9% of 49153341 rows
Read 87.6% of 49153341 rows
Read 88.3% of 49153341 rows
Read 89.0% of 49153341 rows
Read 89.7% of 49153341 rows
Read 90.4% of 49153341 rows
Read 91.1% of 49153341 rows
Read 91.8% of 49153341 rows
Read 92.4% of 49153341 rows
Read 93.1% of 49153341 rows
Read 93.8% of 49153341 rows
Read 94.5% of 49153341 rows
Read 95.2% of 49153341 rows
Read 95.9% of 49153341 rows
Read 96.6% of 49153341 rows
Read 97.3% of 49153341 rows
Read 98.0% of 49153341 rows
Read 98.7% of 49153341 rows
```

```
Read 99.4% of 49153341 rows
Read 49153341 rows and 65 (of 65) columns from 13.203 GB file in 00:02:52
```

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

```
flights.by.carrier <- flights[, .(n = .N), by = 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 than it may seem
# We need to concatenate the Year, Month, and DayofMonth fields, but we also
# to take into account the various time zones of the airports in the data
integer.cols <- grep("Time$", names(flights))
```

```
Sys.time()
```

```
## [1] "2016-01-30 23:08:42 AEDT"
```

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

```
## [1] "2016-01-30 23:08:42 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", Year,
                                                    sub("([0-9]{2})([0-9]{2})",
                                                    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", Year,
                                                    sub("([0-9]{2})([0-9]{2})",
                                                    perl = TRUE)),
                                                    tz = "GMT")]

  DT[unique_dates]
}
```

```
flights <- create_DepDateTime(flights)
```

```
flights <- create_ArrDateTime(flights)
```

```
#flights[, `:=`(Year = NULL, Month = NULL, DayofMonth = NULL, DepTime = NULL)]
Sys.time()
```

```
# Now we join it to the airports dataset from nycflights1r
Sys.time()
airports <- as.data.table(airports)
airports <- airports[,list(faa, tz)]
setnames(airports, old = c("faa", "tz"), new = c("Origin", "tzOrigin"))
setkey(airports, Origin)
setkey(flights, Origin)
flights <- flights[airports]
setnames(airports, old = c("Origin", "tzOrigin"), new = c("faa", "tz"))
setkey(flights, Dest)
flights <- flights[airports]
rm(airports)
# The joins produce NAs when the airports table isn't loaded
flights <- flights[!is.na(Origin)]
Sys.time()
```

```
Sys.time()
# setting keys doesn't improve timing
flights[,`:=`(DepDateTimeZulu = DepDateTime - lubridate::hms(4), tz = "UTC")]
flights[,`:=`(ArrDateTimeZulu = ArrDateTime - lubridate::hms(4), tz = "UTC")]
Sys.time()
```

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

```
flights_with_timezones <- readRDS("flights-post-2008_with_zuluTimes.rds")
setkey(flights_with_timezones, tempkey)
setkey(flights, tempkey)
flights <- flights[flights_with_timezones]
```

```
# Flights typically follow a weekly cycle, so we should be able to
# Pretty quick!
Sys.time()

## [1] "2016-01-30 23:09:12 AEDT"

setkey(flights, Year, Month, DayofMonth)
unique_dates <-
  unique(flights) %>%
  select(Year, Month, DayofMonth) %>%
  mutate(Week = (Year - 1987L) * 52 + data.table::yday(
    Date = as.Date(paste0(Year, "-", Month, "-", DayofMonth)))
  ) %>%
  group_by(Year, Month) %>%
  mutate(Week = Week - min(Week))
flights <- flights[unique_dates]
Sys.time()

## [1] "2016-01-30 23:09:43 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

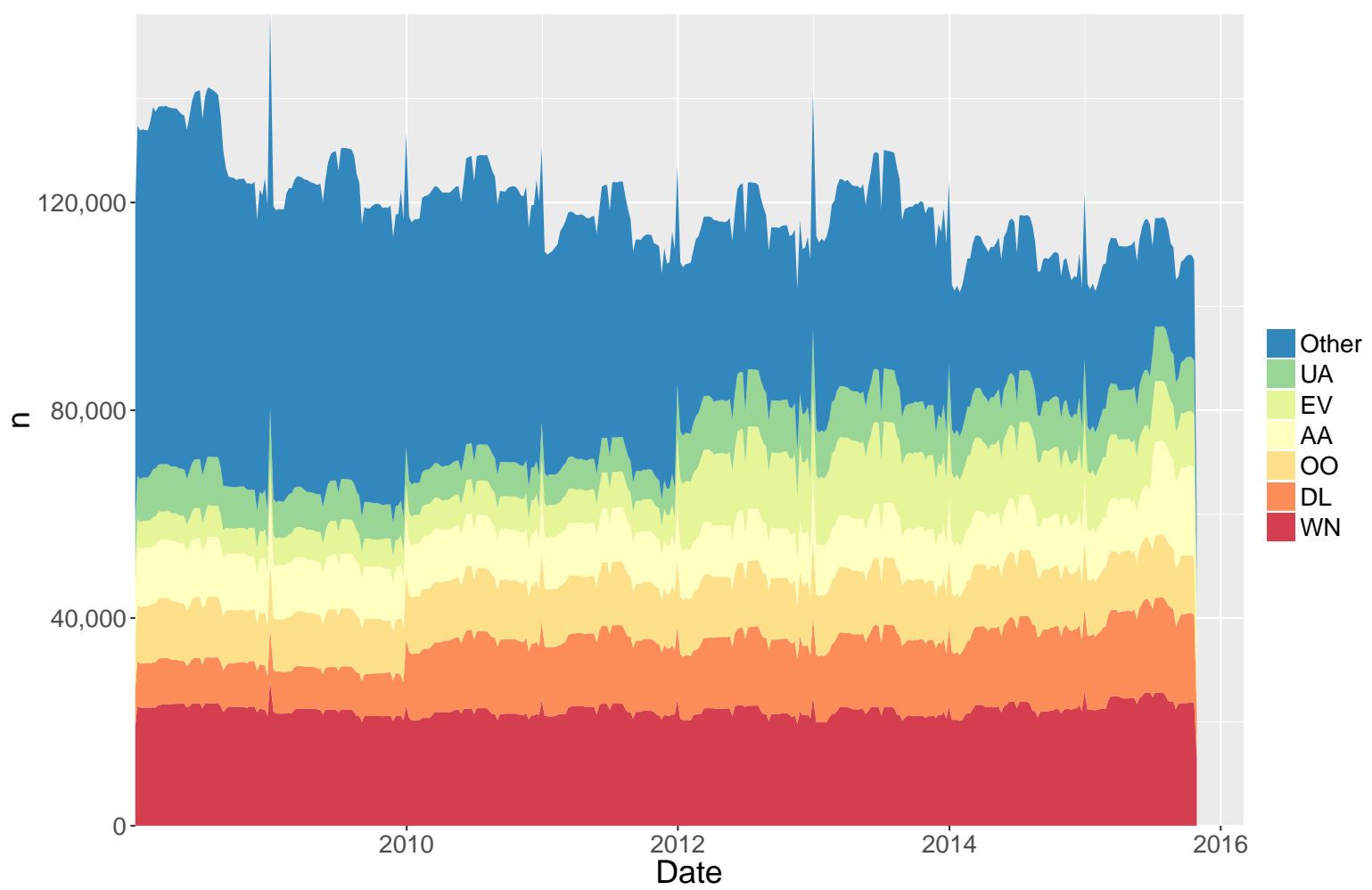
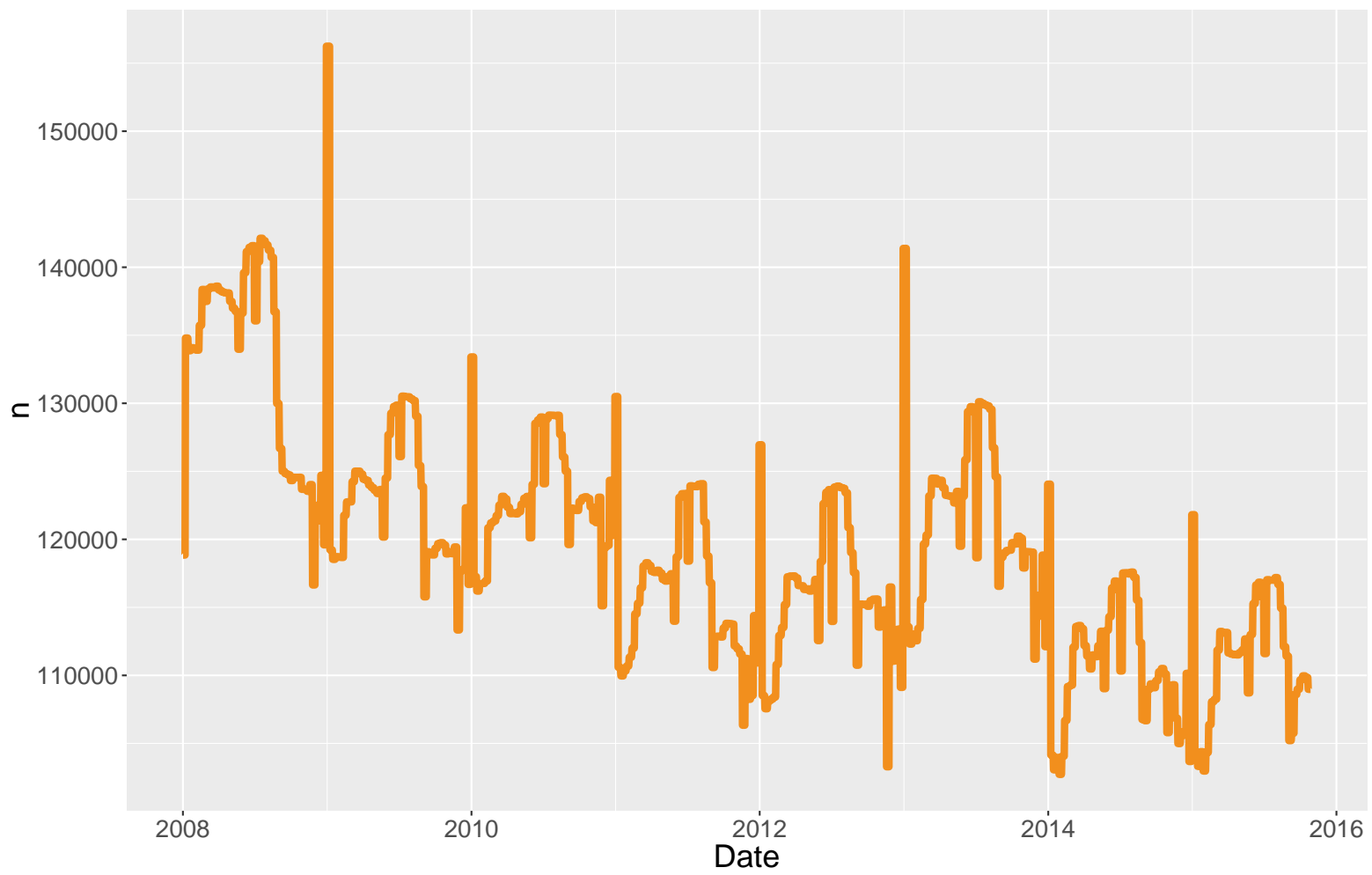
nycflights.airlines[,Carrier_other := ifelse(UniqueCarrier %in% biggest.carriers, UniqueCarrier, "Other")]
```

```
flights.by.week.and.carrier.other <-
  flights.by.week.and.carrier %>%
  group_by(Week,
           Carrier_other = ifelse(UniqueCarrier %in% biggest.carriers, UniqueCarrier, "Other")) %>%
  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.carriers, "Other")))
```

```
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 = 0) %>%
  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_real_)) %>%
  convert_week_to_date %>%
  ggplot(aes(x = Date, y = r, label = label.y)) +
  geom_line() +
  geom_text(aes(label = label.y)) +
  scale_y_continuous(label = comma)
```

```

ggplot(aes(x = Date, y = r, color = Carrier_other, #
geom_line() +
geom_dl(method = "last.qp", aes(label = ifelse(is.na(name)),
# 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()

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 =
  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(UniqueCarrier %in%
# summarise(expected_cancellation = mean(Cancelled))

# })
# system.time({
# flights[,Carrier_other := ifelse(UniqueCarrier %in% names(carriers),
# .[,.(expected_cancellation = mean(Cancelled)), by = list(Year, Month)]

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_cancellation, group = Carrier_other)) +
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) %>%
  summarise(total_ArrDelay = sum(ArrDelay, na.rm = TRUE))

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

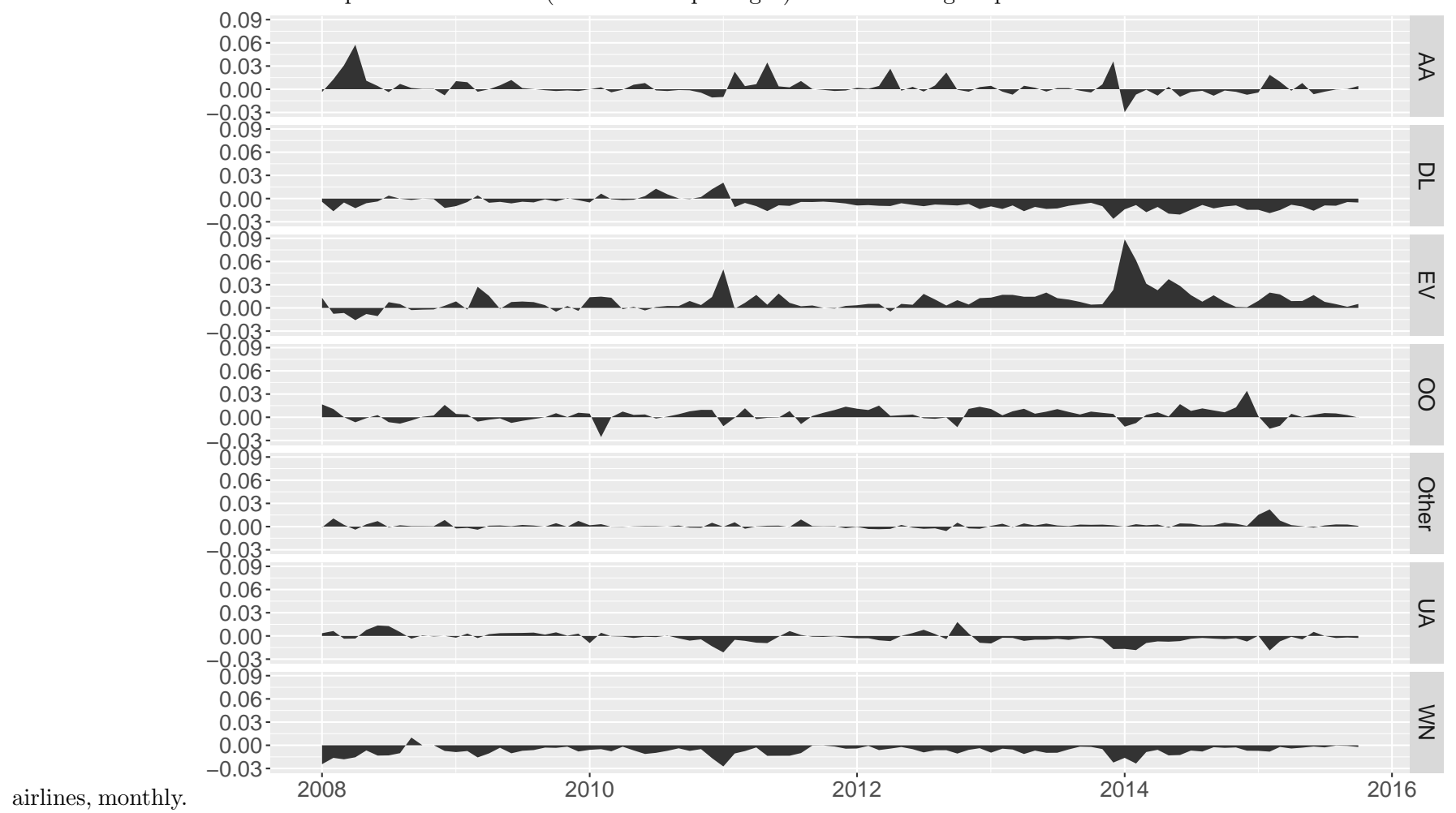
ArrDelays.by.month <-
  flights %>%
  select(Year, Month, ArrDelay) %>%
  group_by(Year, Month) %>%
  summarise(total_ArrDelay = sum(ArrDelay, na.rm = TRUE))

ArrDelays.by.month %>%
  ggplot(aes(Year, Month, total_ArrDelay)) +
  geom_area(group = 1) +
  facet_grid(Carrier_other ~ .) +
  theme(axis.title = element_blank())

```

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



```

    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, UniqueCarrie

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))

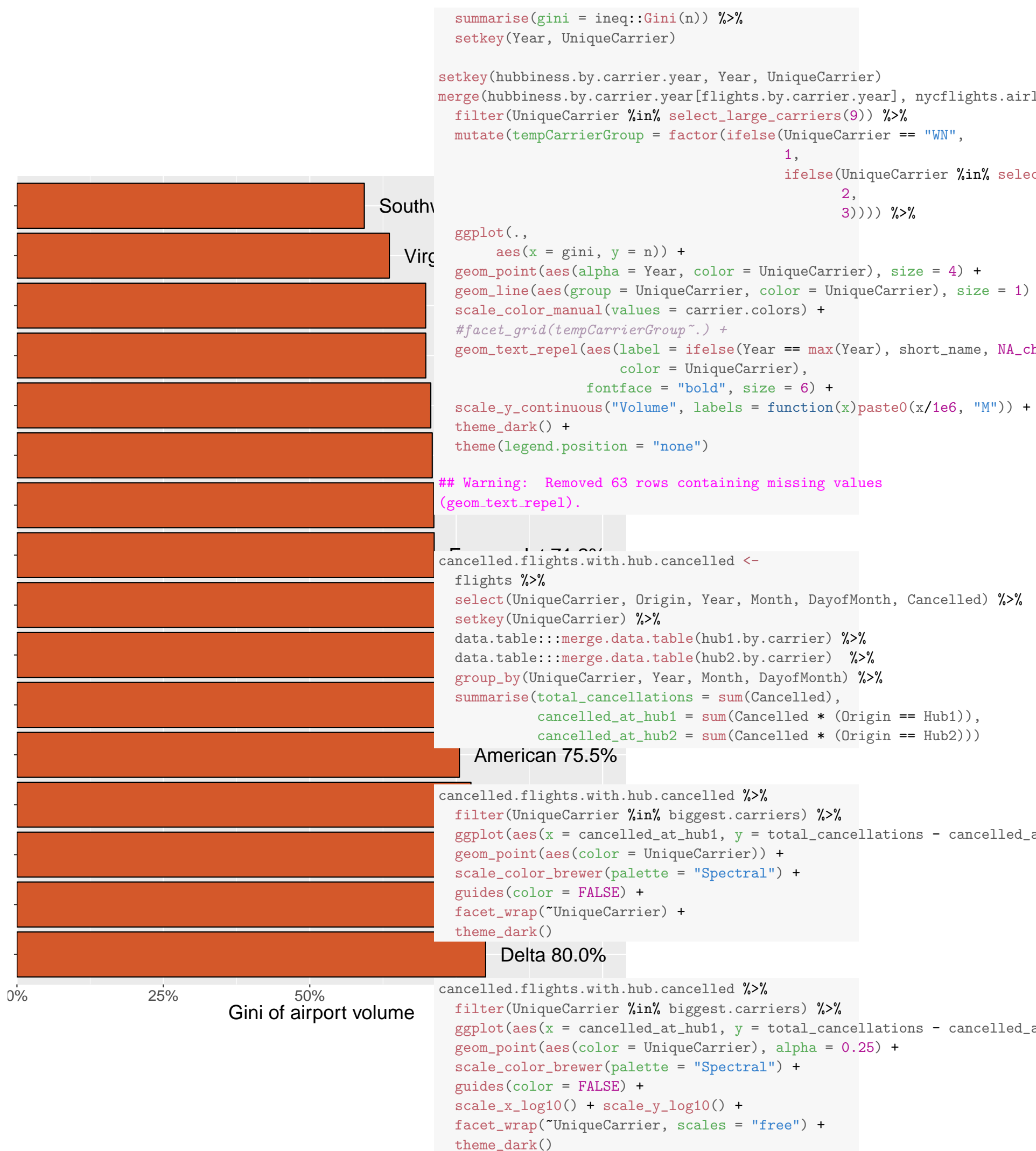
hubbiness.by.carrier %>%
  ungroup %>%
  setkey(UniqueCarrier) %>%
  merge(nycflights.airlines) %>%
  ungroup %>%
  arrange(desc(gini)) %>%
  mutate(short_name = factor(short_name, levels = .$short_name)) %>%
  {
    ggplot(., aes(x = short_name, y = gini, order = gini)) +
      geom_bar(stat = "identity", 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

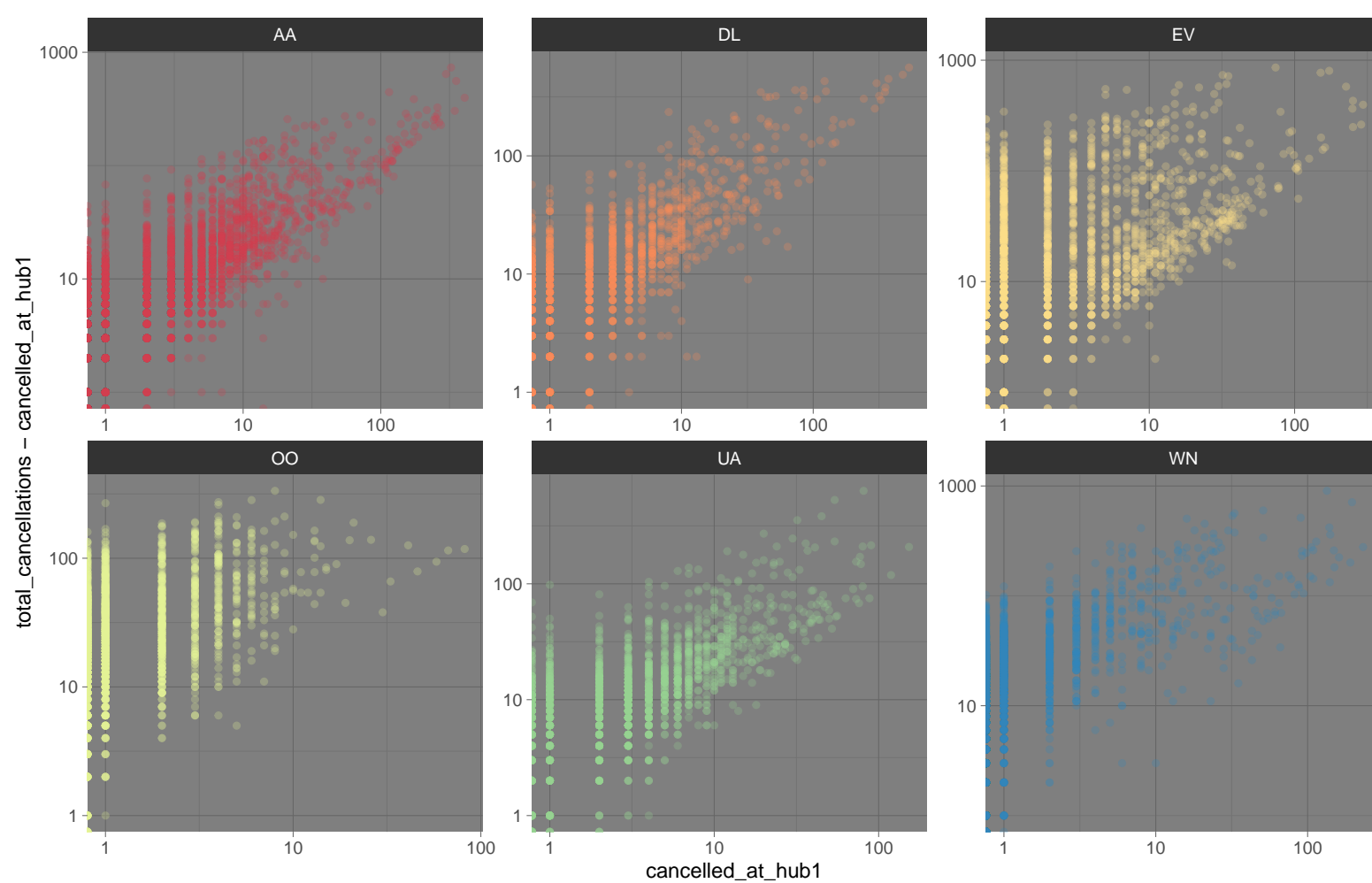
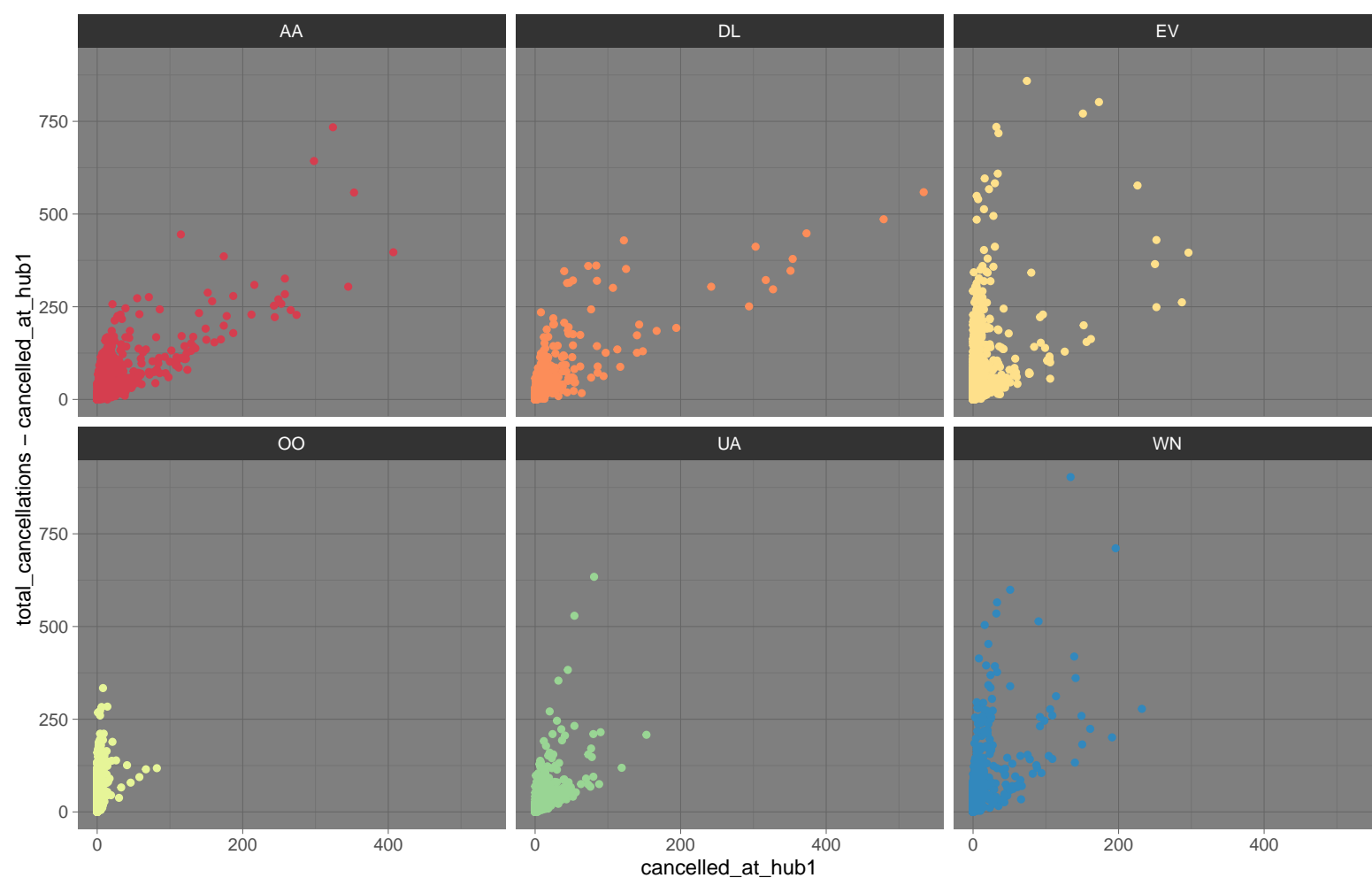
ggplot(hubbiness.by.carrier[flights.by.carrier][nycflights.airlines],
  aes(x = gini, y = n)) +
  geom_point(size = 2) +
  geom_text_repel(aes(label = short_name), fontface = "bold", size = 6) +
  scale_y_continuous("Volume (2008-2015)", labels = function(x)paste0(x/1

## Error in `[.data.table'(hubbiness.by.carrier[flights.by.carrier],
nycflights.airlines): When i is a data.table (or character
vector), x must be keyed (i.e. sorted, and, marked
as sorted) so data.table knows which columns to join
to and take advantage of x being sorted. Call setkey(x,...)
first, see ?setkey.

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) %>%

```



```

cancelled.flights.with.hub.cancelled %>%
  filter(UniqueCarrier %in% biggest.carriers) %>%
  ggplot(aes(x = cancelled_at_hub2, y = total_cancelled_at_hub2, color = UniqueCarrier), alpha = 0.2) +
  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 = cancelled_at_hub1 - cancelled_at_hub2,
         cancelled_rel_other_carriers = total_cancelled_at_hub1 - cancelled_at_hub2)
  ggplot(aes(x = cancelled_at_hub1_rel_other_hubs, y = cancelled_rel_other_carriers, 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 - cancelled_at_hub2,
         cancelled_outside_hub_rel_other_carriers = cancelled_outside_hub - cancelled_at_hub2)
  ggplot(aes(x = cancelled_at_hub1_rel_other_hubs, y = cancelled_outside_hub_rel_other_carriers, color = UniqueCarrier), alpha = 0.2) +
  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.arrydelay.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(avg_arrydelay = sum(ArrDelay, na.rm = TRUE) / n(),
            avg_arrydelay_at_hub = sum(ArrDelay * (Origin == Hub1), na.rm = TRUE) / n(),
            avg_arrydelay_not_at_hub = sum(ArrDelay * (Origin != Hub1), na.rm = TRUE) / n()) %>%
  setkey(Year, Month, DayofMonth) %>%
  data.table::merge.data.table(ArrDelays.avg.by.day)

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

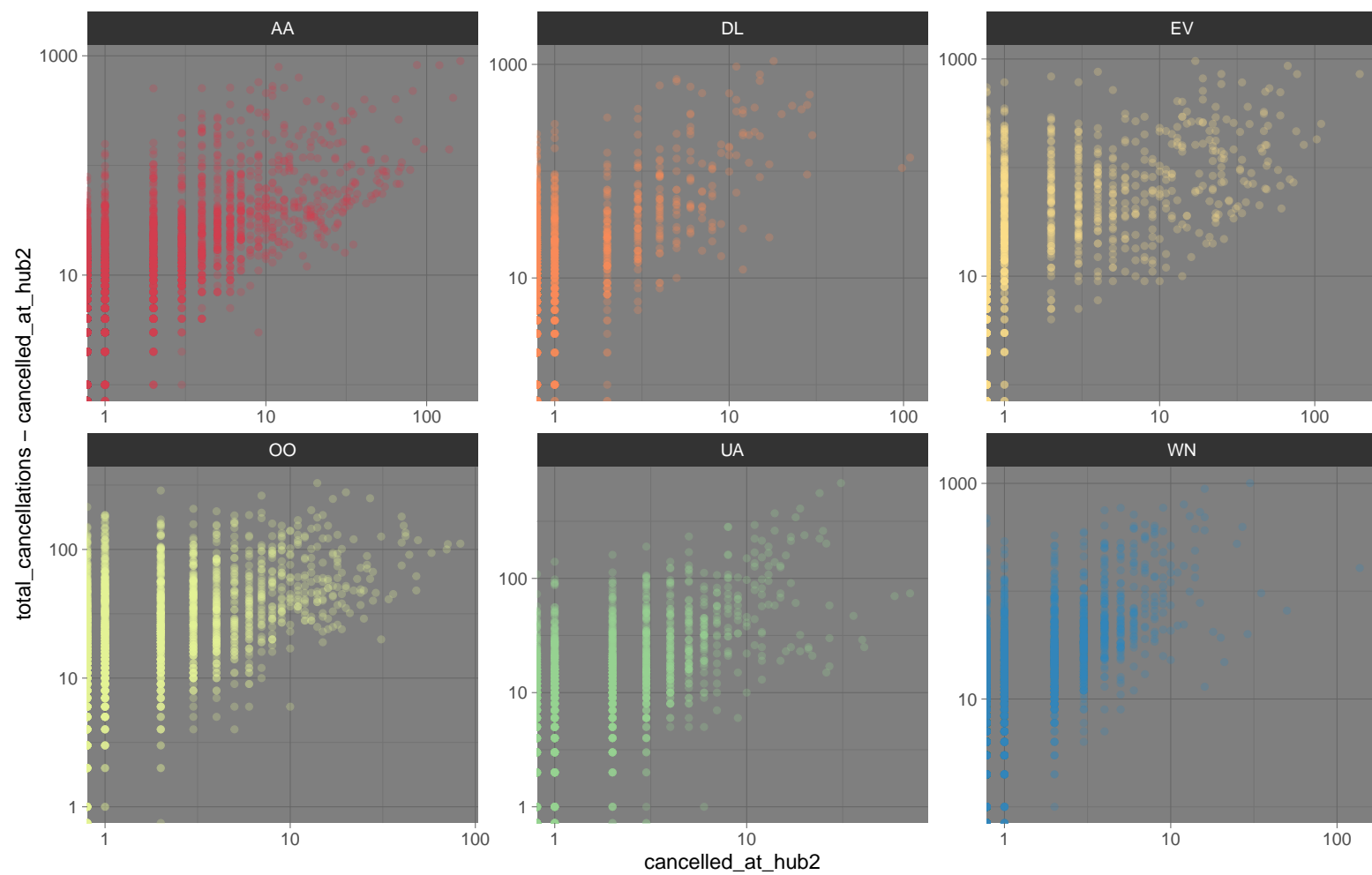
dates.avg.arrydelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)) %>%
  merge(nycflights.airlines, by = "UniqueCarrier") %>%
  ggplot(aes(x = avg_arrydelay_at_hub, y = avg_arrydelay_not_at_hub, color = UniqueCarrier), alpha = 0.33) +
  facet_wrap(~short_name) +
  theme_dark() +
  scale_color_brewer(palette = "Spectral", guide = FALSE)

dates.avg.arrydelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)) %>%
  merge(nycflights.airlines, by = "UniqueCarrier") %>%
  ggplot(aes(x = avg_arrydelay_at_hub, y = avg_arrydelay_not_at_hub, color = UniqueCarrier), alpha = 0.33) +
  facet_wrap(~short_name, scales = "free") +
  theme_dark() +
  scale_color_brewer(palette = "Spectral", guide = FALSE)

dates.avg.arrydelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)) %>%
  merge(nycflights.airlines, by = "UniqueCarrier") %>%
  ggplot(aes(x = avg_arrydelay_at_hub - avg_ArrDelay_allcarriers, y = avg_arrydelay_not_at_hub - avg_ArrDelay_allcarriers, color = UniqueCarrier), alpha = 0.33) +
  facet_wrap(~short_name) +
  theme_dark() +
  geom_abline(slope = 1, color = "white") +
  scale_color_brewer(palette = "Spectral", guide = FALSE)

dates.avg.arrydelay.rel.hub %>%
  filter(UniqueCarrier %in% select_large_carriers(9)) %>%
  merge(nycflights.airlines, by = "UniqueCarrier") %>%
  ggplot(aes(x = avg_arrydelay_at_hub - avg_ArrDelay_allcarriers, y = avg_arrydelay_not_at_hub - avg_ArrDelay_allcarriers, color = UniqueCarrier), 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", nrow = 5749, verbose = TRUE, colClasses = c("integer", : Column 1 ('Code') has been detected as type 'character'. I request from colClasses to read as 'integer' (a lower type) since NAs (or loss of precision) may result.
## Input contains no \n. Taking this to be a filename
## File opened, filesize is 0.000153 GB.
## Memory mapping ... ok
## Detected eol as \r\n (CRLF) in that order, the Windows line ending is \r\n
## Positioned on line 1 after skip or autostart
## This line is the autostart and not blank so searching for the next non-blank line
## Detecting sep ... ','
## Detected 2 columns. Longest stretch was from line 1
## Starting data input on line 1 (either column name or data)
## All the fields on line 1 are character fields. Try to read as character
## Count of eol: 5750 (including 1 at the end)
## Count of sep: 11506
## nrow = MIN( nsep [11506] / ncol [2] - 1, neol [5750] - nblank [1] ) = 5749
## 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", nrow = 5749, verbose = TRUE, colClasses = c("integer", : Column 1 ('Code') has been detected as type 'character'. I 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 integer casting)
## Type codes: 44 (after applying drop or select (if any))
## Allocating 2 column slots (2 - 0 dropped)
## Read 5749 rows. Exactly what was estimated and all data is read
## 0.000s ( 0%) Memory map (rerun may be quicker)
## 0.000s ( 0%) sep and header detection
## 0.000s ( 0%) Count rows (wc -l)
## 0.000s ( 0%) Column type detection (first, middle, last)

## 0.000s ( 0%) Allocation of 5749x2 result (xMB), in RAM
## 0.003s (100%) 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
## 0.003s Total

city.market.decoder[, Code := as.integer(Code)]
city.market.volumes <-
  flights %>%
  select(OriginCityMarketID) %>%
  count(OriginCityMarketID) %>%
  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)] %>%
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code")
  group_by(Year) %>%
  filter(n >= nth(n, 8, order_by = -n)) %>%
  tbl_df %>%
  mutate(Description = factor(Description)) %>%
  mutate(Description = factor(Description,
    levels = city.market.volumes.2014$Description,
    labels = gsub("[A-Z]{2}.*$", "", city.market.volumes.2014$Description))
  filter(Year < max(Year)) %>%
```



```

{
  ggplot(., aes(x = Year, y = n, group = Description)) +
  geom_line() +
  geom_dl(method = "last.points", aes(label = Description)) +
  scale_color_brewer(palette = "Spectral") +
  theme(legend.position = "none") +
  scale_x_continuous(limits = c(min(.$Year), max(.$Year)))
}

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(Week, OriginCityMarketID)] %>%
  merge(city.market.decoder, by.x = "OriginCityMarketID", by.y = "Code") %>%
  group_by(Week) %>%
  filter(OriginCityMarketID %in% city.market.volumes.2014$OriginCityMarketID) %>%
  tbl_df %>%
  mutate(Description = factor(Description)) %>%
  mutate(Description = factor(Description,
                              levels = city.market.volumes.2014$Description,
                              labels = gsub(" ", "[A-Z]", Description))) %>%
  filter(Week < max(Week) & Week > min(Week)) %>%
  mutate(Description.label = ifelse(Week == max(Week), Description, "")) %>%
  {
    ggplot(., aes(x = Week, y = n, group = Description)) +
    geom_line() +
    # geom_text_repel(aes(x = Week, label = Description.label)) +
    # geom_dl(method = "last.points", aes(label = Description.label)) +
    geom_text(aes(label = Description.label, hjust = 0.5)) +
    scale_color_brewer(palette = "Spectral") +
    theme(legend.position = "none") +
    annotate("blank", x = max(.$Week) + 50, y = mean(.$n)) +
    theme_dark()
  }

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

corridor.volumes.by.week <-
  flights[, Corridor := pmin(paste0(OriginCityMarketID, "-", DestCityMarketID),
                             paste0(DestCityMarketID, "-", OriginCityMarketID))] [, .(n = .N),
    by = list(Week, Corridor)] [order(-n)]

corridor.volumes.by.week <-
  flights[, Corridor := sprintf("%s-%s",
                                pmin(OriginCityMarketID, DestCityMarketID),
                                pmax(OriginCityMarketID, DestCityMarketID))] [, .(n = .N),
    by = list(Week, Corridor)] [order(-n)]

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, "-", DestCityMarketID_DS)] %>%
  Corridors %>% select(Corridor, Corridor_DS) %>% setkey(Corridor)
gc(T,T)

##           used      (Mb) gc trigger      (Mb)    max used      (Mb)
## Ncells   66766183  3565.7   90464440  4831.4   66766183  3565.7
## Vcells  3058649203 23335.7  5077629258 38739.3 3058649203 23335.7

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

corridor.volumes.by.week %>%
  filter(Corridor %in% corridor.volumes$Corridor[1:10]) %>%
  setkey(Corridor) %>%
  merge(Corridors) %>%
  mutate(Corridor_DS_x = gsub("^([A-Z].+),.*-([A-Z].+),.*$", "\\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 ~ .)

COMPILATION TIME: 11.5616566181183

COMPILATION.TIME <- round(difftime(Sys.time(), START.TIME, units = "mins"))
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

```