```r
START.TIME <- Sys.time()
knitr::opts_chunk$set(fig.show = 'hide',
                      fig.width = 8.4,
                      fig.height = 5,
                      out.width = "8.4in")
```

```r
library(data.table)
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
```

```r
library(magrittr)
library(ggplot2)
library(nycflights13)  # for airports
nycflights.airports <- airports
library(fasttime)
library(grattan)
```

```
## Loading required package:  devEMF
##
## Attaching package:  'grattan'
##
## The following object is masked from 'package:datasets':
##
##     Orange
```

```r
pre2008_flights <-
  rbindlist(lapply(list.files(path = "../flights/1987-2008/",
                    pattern = "csv$",
```

```r
                     full.names = TRUE), fread))

pre2008.names <-
  names(pre2008_flights)

read_and_report <-
  function(filename){
    year <- gsub("^.*(2[0-9]{3}).{3,4}csv$", "\\1", filename)
    if(grepl("1.csv", filename, fixed = TRUE))
      cat(year)
    fread(filename, select = pre2008.names, showProgress = FALSE)
  }

gc(1,1)
post2008_flights <-
  rbindlist(lapply(list.files(path = "../flights", recursive = TRUE, pattern = "2[0-9]{3
                              full.names = TRUE),
                   read_and_report))

flights <- rbindlist(list(pre2008_flights, post2008_flights), use.names = TRUE)
readr::write_csv(flights, path = "../1987-2015-On-Time-Performance.csv")
```

```r
Sys.time()

## [1] "2016-01-05 22:31:02 AEDT"

flights <- fread("../1987-2015-On-Time-Performance.csv")

##
Read 0.0% of 165931626 rows
Read 0.5% of 165931626 rows
Read 1.0% of 165931626 rows
Read 1.5% of 165931626 rows
Read 2.0% of 165931626 rows
Read 2.6% of 165931626 rows
Read 3.1% of 165931626 rows
Read 3.6% of 165931626 rows
Read 4.1% of 165931626 rows
Read 4.6% of 165931626 rows
Read 5.1% of 165931626 rows
Read 5.7% of 165931626 rows
Read 6.2% of 165931626 rows
Read 6.7% of 165931626 rows
```

```
Read 7.2% of 165931626 rows
Read 7.7% of 165931626 rows
Read 8.3% of 165931626 rows
Read 8.8% of 165931626 rows
Read 9.3% of 165931626 rows
Read 9.8% of 165931626 rows
Read 10.3% of 165931626 rows
Read 10.8% of 165931626 rows
Read 11.4% of 165931626 rows
Read 11.9% of 165931626 rows
Read 12.4% of 165931626 rows
Read 12.9% of 165931626 rows
Read 13.4% of 165931626 rows
Read 14.0% of 165931626 rows
Read 14.5% of 165931626 rows
Read 15.0% of 165931626 rows
Read 15.5% of 165931626 rows
Read 16.0% of 165931626 rows
Read 16.5% of 165931626 rows
Read 17.1% of 165931626 rows
Read 17.6% of 165931626 rows
Read 18.1% of 165931626 rows
Read 18.6% of 165931626 rows
Read 19.1% of 165931626 rows
Read 19.7% of 165931626 rows
Read 20.2% of 165931626 rows
Read 20.7% of 165931626 rows
Read 21.2% of 165931626 rows
Read 21.7% of 165931626 rows
Read 22.2% of 165931626 rows
Read 22.8% of 165931626 rows
Read 23.3% of 165931626 rows
Read 23.8% of 165931626 rows
Read 24.3% of 165931626 rows
Read 24.8% of 165931626 rows
Read 25.3% of 165931626 rows
Read 25.9% of 165931626 rows
Read 26.4% of 165931626 rows
Read 26.9% of 165931626 rows
Read 27.4% of 165931626 rows
Read 27.9% of 165931626 rows
Read 28.5% of 165931626 rows
Read 29.0% of 165931626 rows
```

```
Read 29.5% of 165931626 rows
Read 30.0% of 165931626 rows
Read 30.5% of 165931626 rows
Read 31.0% of 165931626 rows
Read 31.6% of 165931626 rows
Read 32.1% of 165931626 rows
Read 32.6% of 165931626 rows
Read 33.1% of 165931626 rows
Read 33.6% of 165931626 rows
Read 34.2% of 165931626 rows
Read 34.7% of 165931626 rows
Read 35.2% of 165931626 rows
Read 35.7% of 165931626 rows
Read 36.2% of 165931626 rows
Read 36.7% of 165931626 rows
Read 37.3% of 165931626 rows
Read 37.8% of 165931626 rows
Read 38.3% of 165931626 rows
Read 38.8% of 165931626 rows
Read 39.3% of 165931626 rows
Read 39.8% of 165931626 rows
Read 40.4% of 165931626 rows
Read 40.9% of 165931626 rows
Read 41.4% of 165931626 rows
Read 41.9% of 165931626 rows
Read 42.4% of 165931626 rows
Read 43.0% of 165931626 rows
Read 43.5% of 165931626 rows
Read 44.0% of 165931626 rows
Read 44.5% of 165931626 rows
Read 45.0% of 165931626 rows
Read 45.5% of 165931626 rows
Read 46.1% of 165931626 rows
Read 46.6% of 165931626 rows
Read 47.1% of 165931626 rows
Read 47.6% of 165931626 rows
Read 48.1% of 165931626 rows
Read 48.7% of 165931626 rows
Read 49.2% of 165931626 rows
Read 49.7% of 165931626 rows
Read 50.2% of 165931626 rows
Read 50.7% of 165931626 rows
Read 51.3% of 165931626 rows
```

```
Read 51.8% of 165931626 rows
Read 52.3% of 165931626 rows
Read 52.8% of 165931626 rows
Read 53.4% of 165931626 rows
Read 53.9% of 165931626 rows
Read 54.4% of 165931626 rows
Read 54.9% of 165931626 rows
Read 55.5% of 165931626 rows
Read 56.0% of 165931626 rows
Read 56.5% of 165931626 rows
Read 57.0% of 165931626 rows
Read 57.5% of 165931626 rows
Read 58.1% of 165931626 rows
Read 58.6% of 165931626 rows
Read 59.1% of 165931626 rows
Read 59.6% of 165931626 rows
Read 60.2% of 165931626 rows
Read 60.7% of 165931626 rows
Read 61.2% of 165931626 rows
Read 61.7% of 165931626 rows
Read 62.3% of 165931626 rows
Read 62.8% of 165931626 rows
Read 63.3% of 165931626 rows
Read 63.8% of 165931626 rows
Read 64.4% of 165931626 rows
Read 64.9% of 165931626 rows
Read 65.4% of 165931626 rows
Read 65.9% of 165931626 rows
Read 66.5% of 165931626 rows
Read 67.0% of 165931626 rows
Read 67.5% of 165931626 rows
Read 68.0% of 165931626 rows
Read 68.6% of 165931626 rows
Read 69.1% of 165931626 rows
Read 69.6% of 165931626 rows
Read 70.1% of 165931626 rows
Read 70.6% of 165931626 rows
Read 71.2% of 165931626 rows
Read 71.7% of 165931626 rows
Read 72.2% of 165931626 rows
Read 72.7% of 165931626 rows
Read 73.2% of 165931626 rows
Read 73.7% of 165931626 rows
```

```
Read 74.2% of 165931626 rows
Read 74.7% of 165931626 rows
Read 75.3% of 165931626 rows
Read 75.8% of 165931626 rows
Read 76.3% of 165931626 rows
Read 76.8% of 165931626 rows
Read 77.3% of 165931626 rows
Read 77.8% of 165931626 rows
Read 78.3% of 165931626 rows
Read 78.8% of 165931626 rows
Read 79.4% of 165931626 rows
Read 79.9% of 165931626 rows
Read 80.4% of 165931626 rows
Read 80.9% of 165931626 rows
Read 81.4% of 165931626 rows
Read 81.9% of 165931626 rows
Read 82.4% of 165931626 rows
Read 82.9% of 165931626 rows
Read 83.5% of 165931626 rows
Read 84.0% of 165931626 rows
Read 84.5% of 165931626 rows
Read 85.0% of 165931626 rows
Read 85.5% of 165931626 rows
Read 86.0% of 165931626 rows
Read 86.5% of 165931626 rows
Read 87.0% of 165931626 rows
Read 87.5% of 165931626 rows
Read 88.1% of 165931626 rows
Read 88.6% of 165931626 rows
Read 89.1% of 165931626 rows
Read 89.6% of 165931626 rows
Read 90.1% of 165931626 rows
Read 90.6% of 165931626 rows
Read 91.1% of 165931626 rows
Read 91.6% of 165931626 rows
Read 92.2% of 165931626 rows
Read 92.7% of 165931626 rows
Read 93.2% of 165931626 rows
Read 93.7% of 165931626 rows
Read 94.2% of 165931626 rows
Read 94.7% of 165931626 rows
Read 95.2% of 165931626 rows
Read 95.7% of 165931626 rows
```

```
Read 96.3% of 165931626 rows
Read 96.8% of 165931626 rows
Read 97.3% of 165931626 rows
Read 97.8% of 165931626 rows
Read 98.3% of 165931626 rows
Read 98.8% of 165931626 rows
Read 99.3% of 165931626 rows
Read 99.8% of 165931626 rows
Read 165931626 rows and 29 (of 29) columns from 15.111 GB file in 00:03:50

# flights <- readRDS("../1987-2015-On-Time-Performance.rds")
```

```
flightsSanFran <- flights[Origin %in% c("SFO", "OAK") | Dest %in% c("SFO", "OAK")]
sample.frac = 0.2
sample.weight.int = as.integer(round(1/sample.frac))
flights <- flights[sample(.N, .N * sample.frac)]
```

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

Sys.time()

## [1] "2016-01-05 22:35:40 AEDT"

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

## [1] "2016-01-05 22:35:40 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, Month, DayofM
                                                    sub("([0-9]{2})([0-9]{2})", "\\1:\\2:0
                                                        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, Month, DayofM
                                                   sub("([0-9]{2})([0-9]{2})", "\\1:\\2:0
                                                       perl = TRUE)),
                                           tz = "GMT")]
  DT[unique_dates]
}
flights <- create_DepDateTime(flights)
flights <- create_ArrDateTime(flights)
#flights[,`:=`(Year = NULL, Month = NULL, DayofMonth = NULL, DepTime = NULL, ArrTime = N
Sys.time()

## [1] "2016-01-05 22:37:32 AEDT"
```

```
# Now we join it to the airports dataset from nycflights13 to obtain time zone informati
Sys.time()

## [1] "2016-01-05 22:37:32 AEDT"

airports <- as.data.table(airports)
airports <- airports[,list(faa, tz)]
gc(1,1)

##             used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells    533584   28.5   11554252   617.1    533584   28.5
## Vcells 819117392 6249.4 2325188006 17739.8 819117392 6249.4

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("Dest", "tzDest"))
setkey(flights, Dest)
flights <- flights[airports]
rm(airports)
gc(1,1)

##             used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells    533639   28.5    9243401   493.7    533639   28.5
## Vcells 878975212 6706.1 2325188006 17739.8 878975212 6706.1
```

```r
# The joins produce NAs when the airports table isn't present in the flights table.
flights <- flights[!is.na(Origin)]
gc(1,1)

##              used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells     533613   28.5    7394720   395.0    533613   28.5
## Vcells 878952337 6705.9 2325188006 17739.8 878952337 6705.9

Sys.time()

## [1] "2016-01-05 22:38:08 AEDT"
```

```r
Sys.time()

## [1] "2016-01-05 22:38:08 AEDT"

setkey(flights, DepDateTime)
flights[,`:=`(DepDateTimeZulu = DepDateTime - lubridate::hours(tzOrigin),
              ArrDateTimeZulu = ArrDateTime - lubridate::hours(tzDest) )]
Sys.time()

## [1] "2016-01-05 22:42:11 AEDT"
```

```r
# Flights typically follow a weekly cycle, so we should obtain the week in the dataset.
# Pretty quick!
Sys.time()

## [1] "2016-01-05 22:42:11 AEDT"

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

## [1] "2016-01-05 22:42:20 AEDT"
```

Flights 1987-2015

Hugh P

January 5, 2016

1

There were 164 million flights from 1987-10-01 05:00:00 to 2015-11-01 09:10:00.

2 San Francisco

```
Sys.time()

## [1] "2016-01-05 22:42:21 AEDT"

setkey(flightsSanFran, Year, Month, DayofMonth)
unique_dates <- unique(flightsSanFran)
unique_dates <- unique_dates[,list(Year, Month, DayofMonth)]
unique_dates[,Week := (Year - 1987L) * 52 + data.table::yday(sprintf("%d-%02d-%02d", Year
unique_dates[,Week := Week - min(Week)]
flightsSanFran <- flightsSanFran[unique_dates]
Sys.time()

## [1] "2016-01-05 22:42:23 AEDT"
```

```
setkey(unique_dates, Week)
flightsSanFran %>%
  filter(!(Origin %in% c("SFO", "OAK") & Dest %in% c("SFO", "OAK"))) %>%
  mutate(SF_airport = ifelse(Origin %in% c("SFO", "OAK"),
                             Origin,
                             Dest)) %>%
  count(Week, SF_airport) %>%
  setkey(Week) %>%
  data.table:::merge.data.table(unique(unique_dates)) %>%
  mutate(Date = fastPOSIXct(sprintf("%d-%02d-%02d", Year, Month, DayofMonth), tz = "GMT")
         n = n) %>%  # not a sample
  ggplot(aes(x = Date, y = n, color = SF_airport, group = SF_airport)) +
  geom_point() +
  geom_line(size = 0.5) +
  #
  geom_vline(xintercept = as.numeric(as.POSIXct("2001-09-11")))
```

```
carriers <- as.data.table(airlines)
if("carrier" %in% names(carriers))
  setnames(carriers, old = "carrier", new = "UniqueCarrier")
```

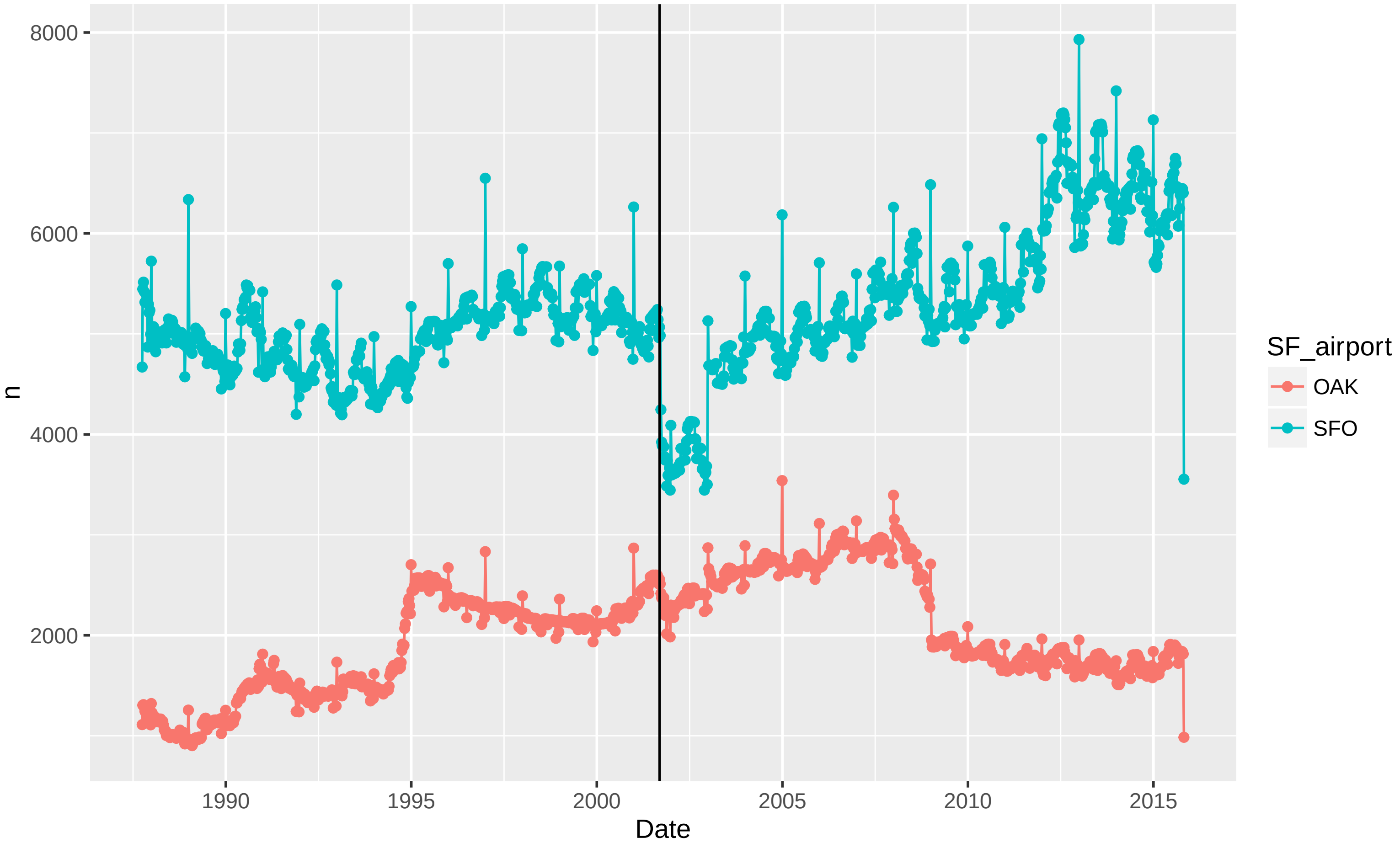


Figure 2.1: Number of depatures over time from Oakland and San Francisco Intl.

```r
setkey(carriers, UniqueCarrier)
set(carriers, j = 1L, value = as.character(carriers[[1L]]))
set(carriers, j = 2L, value = gsub("^([A-Za-z]+)\\s.*$", "\\1", carriers[[2L]]))

flightsSanFran %>%
  filter(Origin %in% c("SFO", "OAK")) %>%
  count(Year, Month, Origin, UniqueCarrier) %>%
  group_by(UniqueCarrier) %>%
  filter(sum(n) > (2015 - 1987) * 12 * 30)  %>%
  mutate(Date = Year + (Month - 1)/12) %>%
  setkey(UniqueCarrier) %>%
  merge(carriers) %>%
  ggplot(aes(x = Date, y = n * sample.weight.int, color = name, group = interaction(name
  geom_smooth(span = 0.25, se = FALSE) +
  geom_text(aes(label = ifelse(Date == max(Date),
                               name,
                               NA_character_),
                vjust = ifelse(name == "Southwest" & Origin == "SFO",
                                -0.5,
                                0.5)),
            nudge_x = 0.75,
            size = 5) + theme(legend.position = "none") +
  annotate("blank", x = 2019, y = 0) +
  facet_grid(Origin ~ .) +
  theme(text = element_text(size = 16))

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
:  span too small.  fewer data values than degrees of freedom.
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
:  pseudoinverse used at 2002.1
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
:  neighborhood radius 0.17125
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
:  reciprocal condition number  0
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
:  There are other near singularities as well.  0.029327
## Warning:  Removed 4579 rows containing missing values (geom_text).
```

After September 11, flights from SFO fell, whereas OAK's volume did notFlights fell more in SFO than they did in OAK because most of OAK's flights are from Southwest, which did not change its flight patterns. Furthermore, United was affected more than most airlines from the aftermath of the attacks.

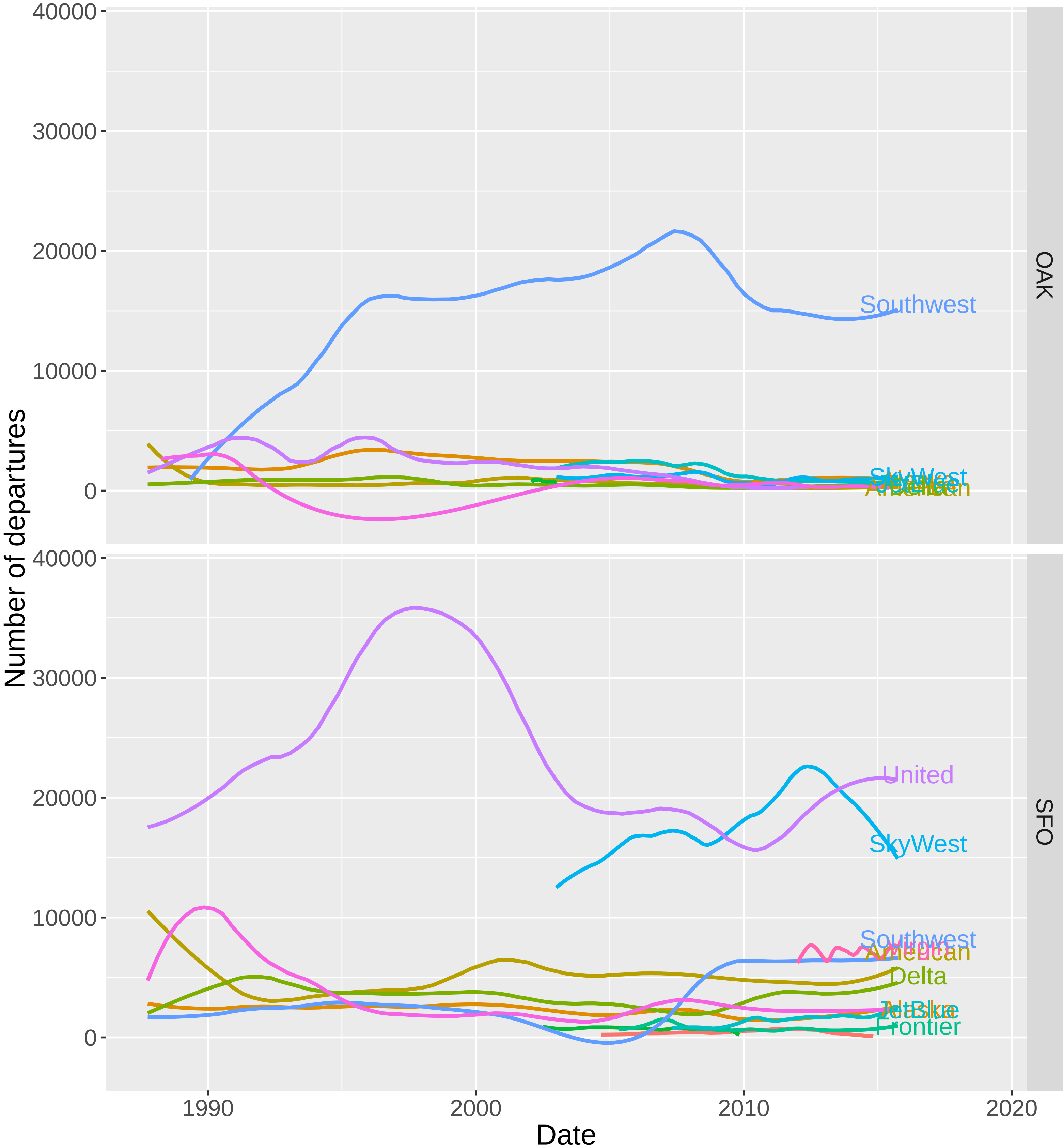
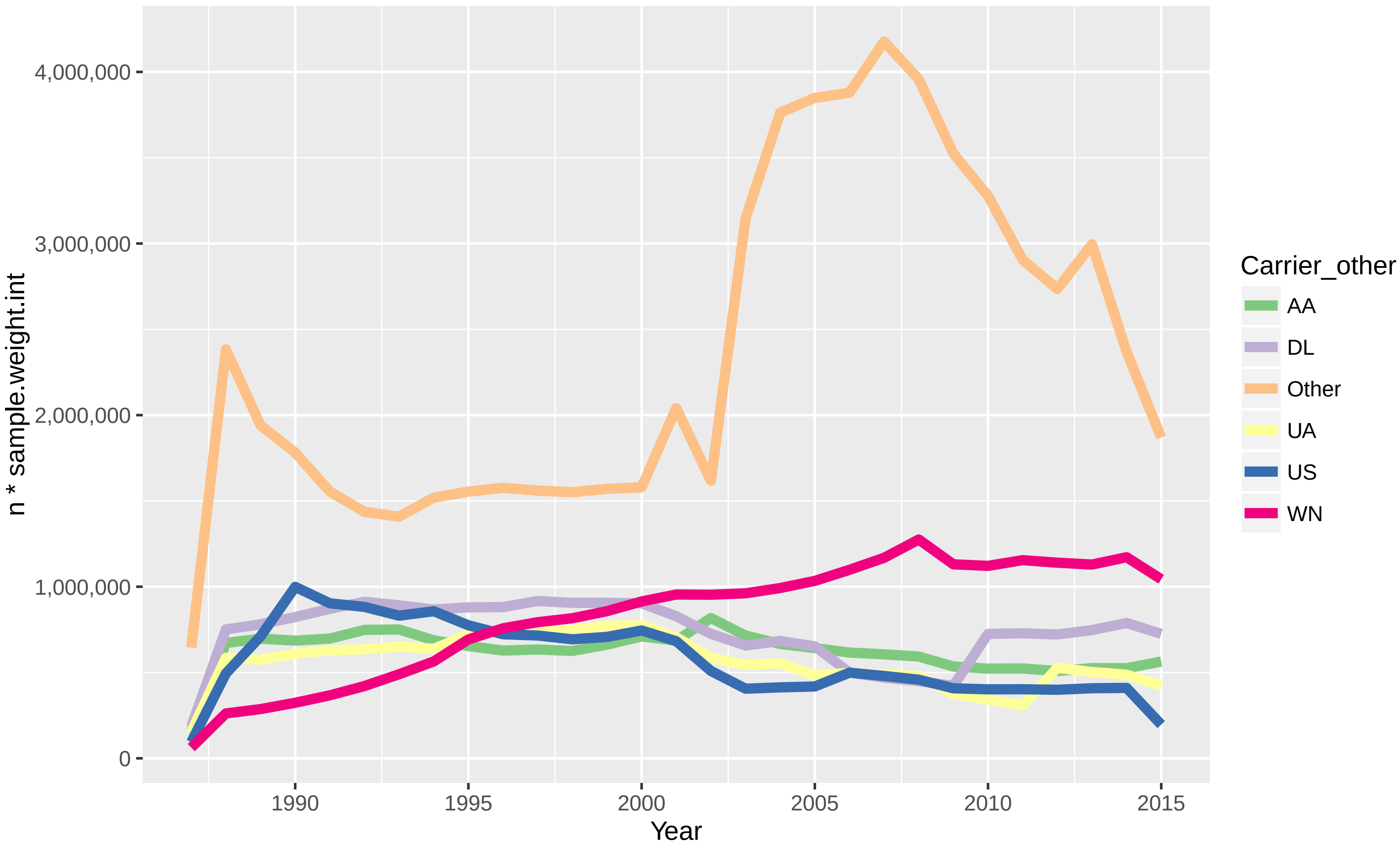


Figure 2.2: Number of depatures over time from Oakland and San Francisco Intl.

```r
top_5_carriers <-
  flights %>%
  count(UniqueCarrier) %>%
  arrange(desc(n)) %>%
  mutate(TopN = 1:n() <= 5) %>%
  mutate(Carrier_other = ifelse(TopN, UniqueCarrier, "Other")) %>%
  select(-n) %>%
  setkey(UniqueCarrier)

flights %>%
  setkey(UniqueCarrier) %>%
  merge(top_5_carriers) %>%
  count(Carrier_other, Year) %>%
  ggplot(aes(x = Year, y = n * sample.weight.int, color = Carrier_other, group = Carrier_
  geom_line() +
  scale_colour_brewer(palette = "Accent") +
  scale_y_continuous(label = scales::comma)
```

```r
majorAirportThreshold = 10

airports_by_volume_by_year <- flights[major_airports][ ,.(n = .N * sample.weight.int), by

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

airports_by_volume_by_2014 <-
  airports_by_volume_by_year %>%
  filter(Year == 2014) %>%
  filter(AirportOther != "AirportOther") %>%
  merge(select(nycflights.airports, faa, name), by.x = "AirportOther", by.y = "faa") %>%
  arrange(desc(n))

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

gc(0,1)

##              used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells     691026   37.0    2423100   129.5    691026   37.0
## Vcells 988301793 7540.2 2790305607 21288.4 988301793 7540.2

major_airports <-
  flights[ ,.(n = .N), by = Dest][order(-n)] %>% # flights %>% count(Dest) %>% arrange(d
  mutate(TopN = 1:n() <= majorAirportThreshold) %>%
  mutate(AirportOther = ifelse(TopN, Dest, "Other_airport")) %>%
```


4,000,000
3,000,000
2,000,000
1,000,000
0
n * sample.weight.int
1990
1995
2000
2005
2010
2015
Year
Carrier_other
AA
DL
Other
UA
US
WN

```
  select(-n) %>%
  setkey(Dest)
setkey(flights, Dest)
gc(0,1)

##              used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells    691069   37.0    2423100   129.5    691069   37.0
## Vcells 988303478 7540.2 2790305607 21288.4 988303478 7540.2

airports_by_volume_by_year %>%
  filter(AirportOther != "Other_airport", Year > 1987L, Year < 2015L)  %>%
  merge(select(nycflights.airports, faa, name), by.x = "AirportOther", by.y = "faa") %>%
  mutate(name = factor(name, levels = airports_by_volume_by_2014$name)) %>%
  ggplot(aes(x = Year, y = n, group = name, color = name)) +
  geom_line()

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

gc(0,1)

##              used   (Mb) gc trigger    (Mb)  max used   (Mb)
## Ncells    691045   37.0    2423100   129.5    691045   37.0
## Vcells 988301906 7540.2 2790305607 21288.4 988301906 7540.2
```

```
rel_vol_major_airports <-
  flights[major_airports][ ,.(n = .N * sample.weight.int), by = list(Year, AirportOther)]
  filter(AirportOther != "Other_airport", Year > 1987L, Year < 2015L) %>%
  arrange(Year) %>%
  group_by(AirportOther) %>%
  mutate(rel = n/first(n)) %>%
  merge(select(nycflights.airports, faa, name), by.x = "AirportOther", by.y = "faa")

last_values <-
  rel_vol_major_airports %>%
  filter(Year == max(Year)) %>%
  arrange(rel)

otp201510 <-
  fread("../dep_delay/On_Time_On_Time_Performance_2015_10.csv")

##
Read 57.6% of 486165 rows
Read 92.6% of 486165 rows
```

```
Read 486165 rows and 110 (of 110) columns from 0.204 GB file in 00:00:04
```

```r
otp201510 %>%
  select(contains("Origin"))
```

```
##         OriginAirportID OriginAirportSeqID OriginCityMarketID Origin
##      1:           12478            1247803              31703    JFK
##      2:           12478            1247803              31703    JFK
##      3:           12478            1247803              31703    JFK
##      4:           12478            1247803              31703    JFK
##      5:           12478            1247803              31703    JFK
##     ---
## 486161:           13830            1383002              33830    OGG
## 486162:           13830            1383002              33830    OGG
## 486163:           13830            1383002              33830    OGG
## 486164:           13830            1383002              33830    OGG
## 486165:           12173            1217302              32134    HNL
##         OriginCityName OriginState OriginStateFips OriginStateName
##      1:   New York, NY          NY              36        New York
##      2:   New York, NY          NY              36        New York
##      3:   New York, NY          NY              36        New York
##      4:   New York, NY          NY              36        New York
##      5:   New York, NY          NY              36        New York
##     ---
## 486161:    Kahului, HI          HI              15          Hawaii
## 486162:    Kahului, HI          HI              15          Hawaii
## 486163:    Kahului, HI          HI              15          Hawaii
## 486164:    Kahului, HI          HI              15          Hawaii
## 486165:   Honolulu, HI          HI              15          Hawaii
##         OriginWac
##      1:        22
##      2:        22
##      3:        22
##      4:        22
##      5:        22
##     ---
## 486161:         2
## 486162:         2
## 486163:         2
## 486164:         2
## 486165:         2
```

```r
rel_vol_major_airports  %>%
  mutate(name = factor(name, levels = rev(last_values$name))) %>%
```

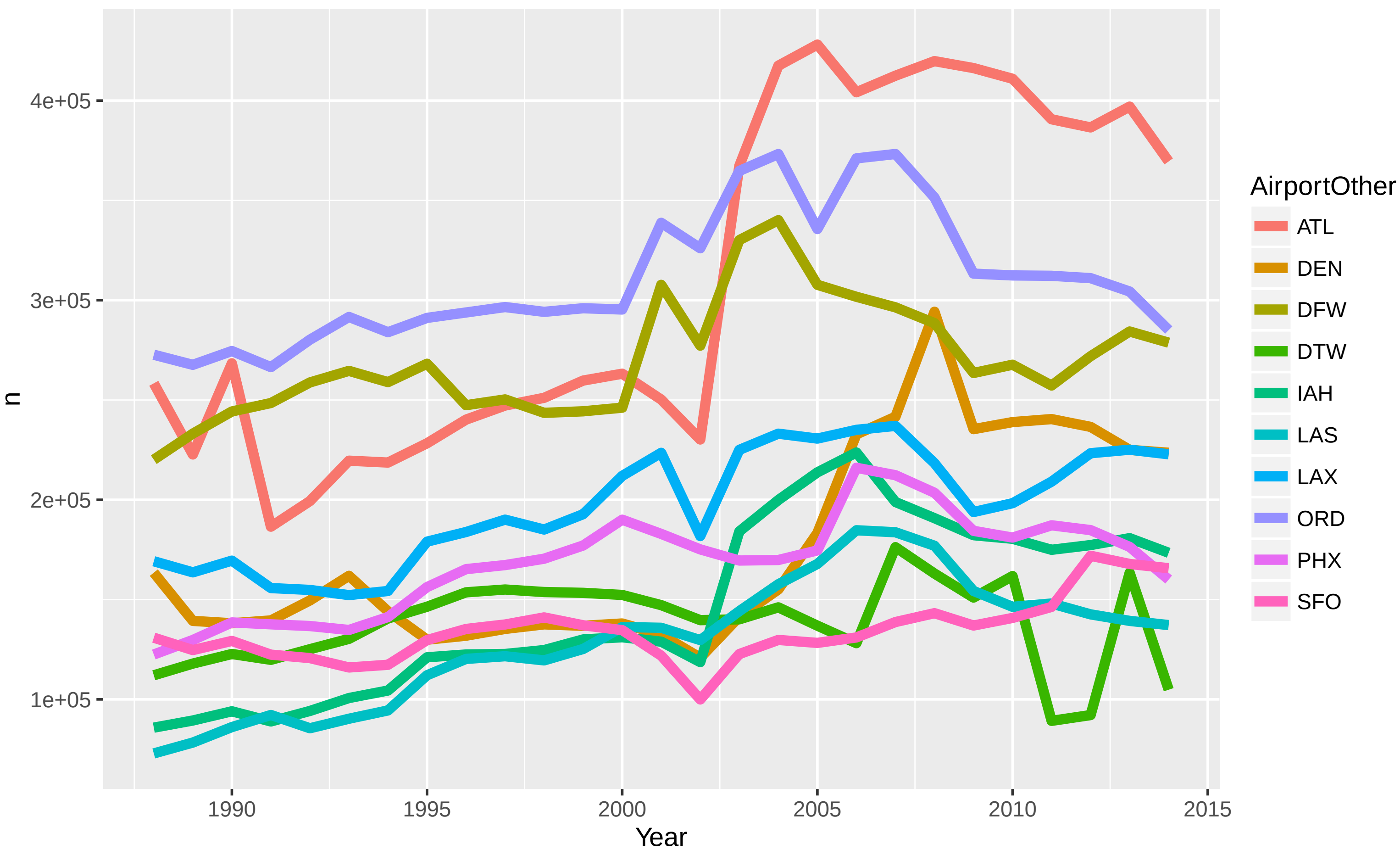


Figure 2.3: Annual flights by the top 10 airports by total volume.

```
ggplot(aes(x = Year, y = rel, group = name, color = name)) +
geom_line()
```

```
FINISH.TIME <- Sys.time()
```

Compiled in 12.2163051843643

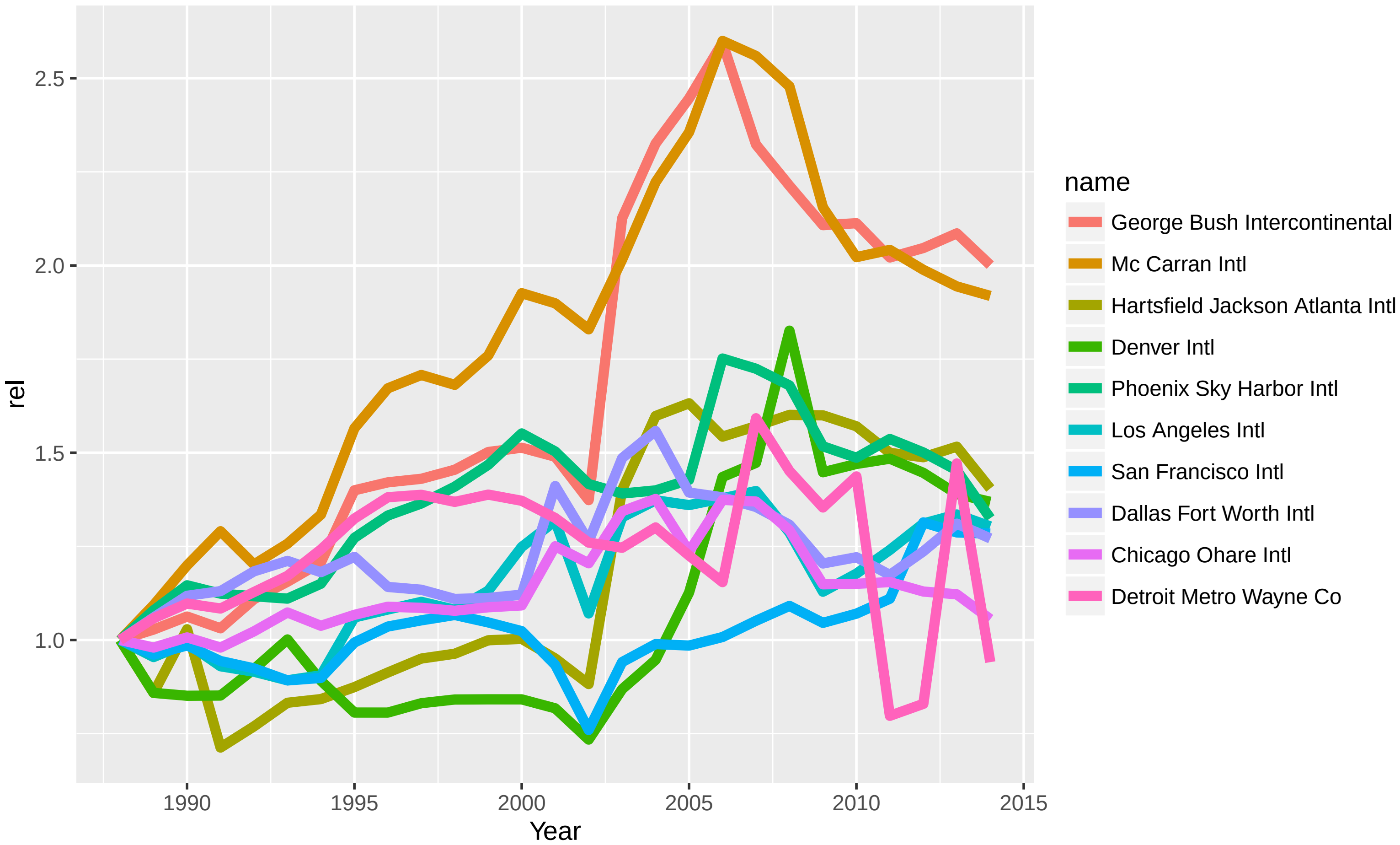


Figure 2.4: Annual flights by airport, 1988 = 1.