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

```
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 sum
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:st
##
##    filter, lag
##
## The following objects are masked from 'package:ba
##
##    intersect, setdiff, setequal, union

library(magrittr)
library(ggplot2)
theme_update(text = element_text(family = "",
                                 face = "plain", col
                                 hjust = 0.5, vjust
                                 debug = FALSE))
library(nycflights13)  # for airports
nycflights.airports <- airports
library(fasttime)
library(grattan)

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

library(directlabels)
```

```
pre2008_flights <-
  rbindlist(lapply(list.files(path = "../flights/1987-2008/",
                              pattern = "csv$",
                              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, patt
                              full.names = TRUE),
                   read_and_report))

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

```
Sys.time()
```

```
## [1] "2016-01-12 00:29:32 AEDT"
```

```
flights <- fread("../1987-2015-On-Time-Performance.c
```

```
##
Read 0.0% of 165931626 rows
Read 0.8% of 165931626 rows
Read 1.6% of 165931626 rows
Read 2.4% of 165931626 rows
Read 3.2% of 165931626 rows
Read 4.0% of 165931626 rows
Read 4.8% of 165931626 rows
Read 5.7% of 165931626 rows
Read 6.5% of 165931626 rows
Read 7.3% of 165931626 rows
Read 8.1% of 165931626 rows
Read 9.0% of 165931626 rows
Read 9.8% of 165931626 rows
Read 10.6% of 165931626 rows
Read 11.5% of 165931626 rows
Read 12.3% of 165931626 rows
Read 13.1% of 165931626 rows
Read 14.0% of 165931626 rows
Read 14.8% of 165931626 rows
Read 15.6% of 165931626 rows
Read 16.5% of 165931626 rows
Read 17.3% of 165931626 rows
Read 18.1% of 165931626 rows
Read 19.0% of 165931626 rows
Read 19.8% of 165931626 rows
Read 20.6% of 165931626 rows
Read 21.5% of 165931626 rows
Read 22.3% of 165931626 rows
Read 23.1% of 165931626 rows
Read 23.9% of 165931626 rows
Read 24.8% of 165931626 rows
Read 25.6% of 165931626 rows
Read 26.4% of 165931626 rows
Read 27.2% of 165931626 rows
Read 28.1% of 165931626 rows
Read 28.9% of 165931626 rows
Read 29.7% of 165931626 rows
Read 30.5% of 165931626 rows
Read 31.4% of 165931626 rows
Read 32.2% of 165931626 rows
Read 33.0% of 165931626 rows
Read 33.8% of 165931626 rows
Read 34.7% of 165931626 rows
Read 35.5% of 165931626 rows
Read 36.3% of 165931626 rows
Read 37.1% of 165931626 rows
Read 38.0% of 165931626 rows
Read 38.8% of 165931626 rows
Read 39.6% of 165931626 rows
Read 40.4% of 165931626 rows
Read 41.3% of 165931626 rows
Read 42.1% of 165931626 rows
Read 42.9% of 165931626 rows
```

```
Time", "ArrTime", "Uni
epDelay", "Origin", "D
```

```
Read 43.7% of 165931626 rows
Read 44.6% of 165931626 rows
Read 45.4% of 165931626 rows
Read 46.2% of 165931626 rows
Read 47.0% of 165931626 rows
Read 47.8% of 165931626 rows
Read 48.7% of 165931626 rows
Read 49.5% of 165931626 rows
Read 50.3% of 165931626 rows
Read 51.1% of 165931626 rows
Read 52.0% of 165931626 rows
Read 52.8% of 165931626 rows
Read 53.6% of 165931626 rows
Read 54.4% of 165931626 rows
Read 55.3% of 165931626 rows
Read 56.1% of 165931626 rows
Read 56.9% of 165931626 rows
Read 57.7% of 165931626 rows
Read 58.5% of 165931626 rows
Read 59.3% of 165931626 rows
Read 60.2% of 165931626 rows
Read 61.0% of 165931626 rows
Read 61.8% of 165931626 rows
Read 62.6% of 165931626 rows
Read 63.4% of 165931626 rows
Read 64.3% of 165931626 rows
Read 65.1% of 165931626 rows
Read 65.9% of 165931626 rows
Read 66.7% of 165931626 rows
Read 67.5% of 165931626 rows
```

```
Read 68.4% of 165931626 rows
Read 69.2% of 165931626 rows
Read 70.0% of 165931626 rows
Read 70.8% of 165931626 rows
Read 71.6% of 165931626 rows
Read 72.4% of 165931626 rows
Read 73.2% of 165931626 rows
Read 74.0% of 165931626 rows
Read 74.9% of 165931626 rows
Read 75.7% of 165931626 rows
Read 76.5% of 165931626 rows
Read 77.3% of 165931626 rows
Read 78.1% of 165931626 rows
Read 78.9% of 165931626 rows
Read 79.7% of 165931626 rows
Read 80.6% of 165931626 rows
Read 81.4% of 165931626 rows
Read 82.2% of 165931626 rows
Read 83.0% of 165931626 rows
Read 83.8% of 165931626 rows
Read 84.6% of 165931626 rows
Read 85.4% of 165931626 rows
Read 86.3% of 165931626 rows
Read 87.1% of 165931626 rows
Read 87.9% of 165931626 rows
Read 88.7% of 165931626 rows
Read 89.5% of 165931626 rows
Read 90.3% of 165931626 rows
Read 91.1% of 165931626 rows
Read 91.9% of 165931626 rows
```

```
Read 92.8% of 165931626 rows
Read 93.6% of 165931626 rows
Read 94.4% of 165931626 rows
Read 95.2% of 165931626 rows
Read 96.0% of 165931626 rows
Read 96.8% of 165931626 rows
Read 97.6% of 165931626 rows
Read 98.4% of 165931626 rows
Read 99.2% of 165931626 rows
Read 165931626 rows and 12 (of 29) columns from 15.1

# flights <- readRDS("../1987-2015-On-Time-Performan
```

```
flightsSanFran <- flights[Origin %in% c("SFO", "OAK"
sample.frac = 0.5
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
# We need to concatenate the Year, Month, and DayofM
# to take into account the various time zones of the
integer.cols <- grep("Time$", names(flights))

Sys.time()

## [1] "2016-01-12 00:32:52 AEDT"

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

## [1] "2016-01-12 00:32:52 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 = NULL, DepTime = NU
Sys.time()

## [1] "2016-01-12 00:35:12 AEDT"
```

```
# Now we join it to the airports dataset from nycfli
Sys.time()

## [1] "2016-01-12 00:35:12 AEDT"

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

##             used   (Mb) gc trigger    (Mb)  max u
## Ncells    538470   28.8   11131760   594.6    538
## Vcells 876465018 6686.9 2271447861 17329.8 876465

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)
gc(1,1)

##             used   (Mb) gc trigger    (Mb)   max
## Ncells     538494   28.8    7124326   380.5     5
## Vcells 1033480214 7884.9 2271447861 17329.8 10334

# The joins produce NAs when the airports table isn'
flights <- flights[!is.na(Origin)]
gc(1,1)
```

```
##             used   (Mb) gc trigger    (Mb)   max used   (Mb)
## Ncells     538509   28.8    5699460   304.4     538509   28.8
## Vcells 1033468019 7884.8 2271447861 17329.8 1033468019 7884.8

Sys.time()

## [1] "2016-01-12 00:36:01 AEDT"
```

```
Sys.time()

## [1] "2016-01-12 00:36:01 AEDT"

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

## [1] "2016-01-12 00:45:45 AEDT"
```

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

## [1] "2016-01-12 00:45:45 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-
```

```
unique_dates[,Week := Week - min(Week)]
flights <- flights[unique_dates]
Sys.time()

## [1] "2016-01-12 00:46:10 AEDT"
```

Flights 1987-2015

Hugh P

January 12, 2016

1

There were 164 million flights from 1987-10-01 05:00:00 to 2015-11-01 09:51:00.

2 San Francisco

```
Sys.time()

## [1] "2016-01-12 00:46:11 AEDT"

setkey(flightsSanFran, Year, Month, DayofMonth)
unique_dates <- unique(flightsSanFran)
unique_dates <- unique_dates[,list(Year, Month, Dayo
unique_dates[,Week := (Year - 1987L) * 52 + data.tab
unique_dates[,Week := Week - min(Week)]
flightsSanFran <- flightsSanFran[unique_dates]
Sys.time()

## [1] "2016-01-12 00:46:15 AEDT"
```

```
maxN <- function(x, N=2){
  len <- length(x)
  if(N>len){
    warning('N greater than length(x).  Setting N=le
    N <- length(x)
  }
  sort(x,partial=len-N+1)[len-N+1]
}

setkey(unique_dates, Week)
```

```
flightsSanFran %>%
  #sample_frac(0.05) %>%
  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) %>%
  group_by(SF_airport) %>%                                  Month)) %/% 7]
  mutate(label.text = ifelse(n == maxN(n), paste(" ", SF_airport), NA_cha
  setkey(Week) %>%
  data.table:::merge.data.table(unique(unique_dates)) %>%
  mutate(Date = fastPOSIXct(sprintf("%d-%02d-%02d", Year, Month, DayofMon
         n = n) %>%  # not a sample
  ggplot(aes(x = Date, y = n, color = SF_airport, group = SF_airport)) +
  geom_point() +
  geom_text(aes(label = label.text),
            nudge_y = 0.5,
            nudge_x = 1,
            hjust = 0,
            fontface = "bold",
            size = 5) +
  theme(legend.position = "none") +
  geom_line(size = 0.5) +
  #
```

```
  geom_vline(xintercept = as.numeric(as.POSIXct("200
  scale_x_datetime(date_breaks = "5 years",
                   date_labels = "%Y",
                   minor_breaks = seq(as.POSIXct("198

## Warning:  Removed 2920 rows containing missing va
(geom_text).
```

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

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

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, col
  geom_smooth(span = 0.25, se = FALSE) +
  geom_text(aes(label = ifelse(Date == max(Date),
                               name,
                               NA_character_),
                vjust = ifelse(name == "Southwest" &
```


80

```
                                  -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 = deg
parametric = parametric, :  span too small.  fewer da
values than degrees of freedom.
## Warning in simpleLoess(y, x, w, span, degree = deg
parametric = parametric, :  pseudoinverse used at 200
## Warning in simpleLoess(y, x, w, span, degree = deg
parametric = parametric, :  neighborhood radius 0.171
## Warning in simpleLoess(y, x, w, span, degree = deg
parametric = parametric, :  reciprocal condition numb
0
## Warning in simpleLoess(y, x, w, span, degree = deg
parametric = parametric, :  There are other near sing
as well.  0.029327
## Warning:  Removed 4579 rows containing missing val
(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.

150

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

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

```
majorAirportThreshold = 10

major_airports <-
  flights[ ,.(n = .N), by = Dest][order(-n)] %>% # f
  mutate(TopN = 1:n() <= majorAirportThreshold) %>%
  mutate(AirportOther = ifelse(TopN, Dest, "Other_ai
  select(-n) %>%
  setkey(Dest)

airports_by_volume_by_year <- flights[major_airports

airports_by_volume_by_2014 <-
```


4,

```
  airports_by_volume_by_year %>%
  filter(Year == 2014) %>%
  filter(AirportOther != "AirportOther") %>%
  merge(select(nycflights.airports, faa, name), by.x
  arrange(desc(n))
gc(0,1)

##              used   (Mb) gc trigger    (Mb)   max
## Ncells     695285   37.2    1867598    99.8     6
## Vcells 1290767501 9847.8 3617730857 27601.1 12907

setkey(flights, Dest)
gc(0,1)

##              used   (Mb) gc trigger    (Mb)   max
## Ncells     695181   37.2    1867598    99.8     6
## Vcells 1290762564 9847.8 3617730857 27601.1 12907

airports_by_volume_by_year %>%
  filter(AirportOther != "Other_airport", Year > 198
  merge(select(nycflights.airports, faa, name), by.x
  mutate(name = factor(name, levels = airports_by_vo
  ggplot(aes(x = Year, y = n, group = name, color =
  geom_line()
gc(0,1)

##              used   (Mb) gc trigger    (Mb)   max
## Ncells     702432   37.6    1867598    99.8     7
## Vcells 1290783051 9847.9 3617730857 27601.1 12907
```

```
rel_vol_major_airports <-
  flights[major_airports][ ,.(n = .N * sample.weight.int), by = list(Year
  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

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

rel_vol_major_airports  %>%
  mutate(name = factor(name, levels = rev(last_values$name))) %>%
  ggplot(aes(x = Year, y = rel, group = name, color = name)) +
  geom_line()
```

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

##
Read 43.2% of 486165 rows
Read 78.2% of 486165 rows
Read 486165 rows and 110 (of 110) columns from 0.204 GB file in 00:00:04

city_decoder <-
  otp201510 %>%
  select(contains("Origin")) %>%
  unique
```

2.

```
setkey(city_decoder, OriginCityMarketID)

gc(T,T)

##              used    (Mb) gc trigger    (Mb)   ma
## Ncells     708733    37.9    1867598    99.8
## Vcells 1335752294 10191.0 3617730857 27601.1 1335

city_market_decoder <-
  fread("../metadata//L_CITY_MARKET_ID.csv") %>%
  setnames(old = c("Code", "Description"),
           new = c("OriginCityMarketID", "OriginCity
  setkey(OriginCityMarketID)
city_market_decoder[,OriginCityMarketID := as.integer(OriginCityMarketID)]
city_decoder <- merge(city_decoder, city_market_deco                    = FALSE)
gc(T,T)

##              used    (Mb) gc trigger    (Mb)   ma
## Ncells     714266    38.2    1867598    99.8
## Vcells 1335778567 10191.2 3617730857 27601.1 1335
```

```
market_volume_by_year <-
  flightsSanFran %>%
  filter(Dest %in% c("SFO", "OAK")) %>%
  merge(city_decoder, by = "Origin") %>%
  count(Year, OriginCityMarketDescription) %>%
  mutate(State = gsub("^.*([A-Z]{2}).*$", "\\1", Ori
  filter(n > 3650) %>%
  mutate(Label = ifelse(Year == max(Year), OriginCit
  arrange(Year, desc(n))
```

```
mkt.vol.by.yr <- function(year, colname){
  magrittr::extract2(dplyr::filter(market_volume_by_year, Year == year),
}
market_volume_by_year %>%
  mutate(OriginCityMarketDescription = factor(OriginCityMarketDescriptio
  ggplot(aes(x = Year, y = n, color = OriginCityMarketDescription, group
  #facet_grid(State ~ .) +
  geom_line() +
  #geom_text(aes(label = Label)) +
  #geom_dl(method = list("top.points", dl.trans(y = y+0.25), fontfamily =
  theme(legend.position = "none") -> p
direct.label(p, list("top.points", dl.trans(y = y+0.25), fontface="bold")
```

```
FAA_aircraft <-
  fread("../metadata/planes.csv") %>%
  setnames(old = c("tailnum", "year"), new = c("TailNum", "YearOfReg")) %
  setkey(TailNum)
```

```
714266    38.2
```

```
flights %>%
  group_by(Origin, Dest) %>%
  filter(n() > 50000) %>%
  mutate(Route = paste0(Origin, "-", Dest),
         RevRoute = paste0(Dest, "-", Origin),
         maxRoute = pmax(Route, RevRoute)) %>%
  ggplot(aes(x = ActualElapsedTime)) +
  geom_density(aes(fill = maxRoute), alpha = 0.5) + xlim(0,300)
```

```
## Warning: Removed 1193422 rows containing non-finite
values (stat_density).
```

```
flights %>%
  select(Origin, Dest, ActualElapsedTime) %>%
  group_by(Origin, Dest) %>%
  summarise(average_time = mean(ActualElapsedTime, n
            sd_time = sd(ActualElapsedTime, na.rm =
            n = n()) %>%
  mutate(avg_less_sd = (sd_time - average_time) / average_time) %>%
  arrange(avg_less_sd) %>%
  mutate(Route = paste0(Origin, "-", Dest),
         Label = ifelse(Route %in% c('ROC-JFK', 'SLC
         hasLabel = !is.na(Label)) %>%
  ggplot(aes(x = average_time, y = sd_time)) +
  #geom_point(aes(alpha = n/max(n))) + scale_alpha_i
  geom_point(aes(size = n, fill = hasLabel, alpha =
  scale_fill_manual(values = c(Orange, "red"))  +
  scale_alpha_manual(values = c(0.5, 1)) +
  geom_text(aes(label = Label), color = "red", fontface = "bold", hjust = 1.1, vjust = 0.0, nudge_x = -1, nudge_y = 0.2) +
  coord_cartesian(xlim = c(0,480), ylim = c(0,50)) +
  scale_x_continuous("Average elapsed time", expand
  scale_y_continuous("SD of time", expand = c(0,0))

## Warning:  Removed 886 rows containing missing val
(geom_point).
## Warning:  Removed 8792 rows containing missing va
(geom_text).
```

```
flights %>%
  group_by(Year, Month, DayofMonth) %>%
  summarise(prop_cancelled = mean(Cancelled)) %>%
  ggplot(aes(x = fasttime::fastPOSIXct(paste(Year, Month, DayofMonth, sep
  geom_bar(stat = "identity", width=1)
```

```
 flights %>%
  group_by(Year, Month, DayofMonth) %>%
  summarise(prop_cancelled = mean(Cancelled)) %>%  ,
  ungroup %>%
  mutate(rank = dense_rank(prop_cancelled)) %>%
  ggplot(aes(x = jitter(rank, amount = 0.1), y = prop_cancelled)) + geom_
                                                                + scale_size_area(max
## Warning:  position_stack requires non-overlapping
x intervals
```

```
flights %>%
  filter(Year == 2001, Month == 9, DayofMonth == 11) %>%
  group_by(Origin) %>%
  summarise(latest_departure = max(DepDateTimeZulu)) %>%
  ungroup %>%
  arrange(latest_departure) %>%
  mutate(number_airports_closed = 1:n()) %>%
  ggplot(aes(x = latest_departure, y = number_airports_closed)) +
  geom_line(group = 1) +
  geom_vline(xintercept = as.numeric(as.POSIXct("2001-09-11 09:17:00")))
```

2.1 Effect of 9-11

1.

Figure 2.7: Number of airports closed by UTC (determined by date of last departure)

2.2 Atlanta, Chicago, and Dallas Fort-Worth

```
flights_hubs <- flights[Origin %in% c('ATL', 'ORD', 'DFW')]
```

```
flights_hubs %>%
  filter(Year < 2015 | Month < 9) %>%
  count(Week, Origin) %>%
  setkey(Week) %>%
  data.table:::merge.data.table(unique(unique_dates)) %>%
  mutate(Date = as.Date(paste0(Year, "-", Month, "-", DayofMonth))) %>%
  ggplot(aes(x = Date, y = n * sample.weight.int, color = Origin, group =
  geom_line() +
  geom_point()
```

```
summary.tbl <-
  flights_hubs %>%
  group_by(Origin, Year, Month, DayofMonth) %>%
  summarise(n = n(), average_delay = sum(DepDelay, na.rm = TRUE) / n())

average_delay_by_hub <-
  summary.tbl %>%
  mutate(Date = as.Date(paste0(Year, "-", Month, DayofMonth))) %>%
  ggplot(aes(x = Date, y = average_delay, group = Origin, color = Origin)
  geom_smooth()

## Error in charToDate(x):  character string is not
in a standard unambiguous format
```

```
total_flights_by_hub <-
  summary.tbl %>%
  mutate(Date = as.Date(paste0(Year, "-", Month, Day
         n = n * sample.weight.int) %>%
  ggplot(aes(x = Date, y = n)) +
  geom_smooth(aes(group = Origin, color = Origin))

## Error in charToDate(x):  character string is not
in a standard unambiguous format

gridExtra::grid.arrange(average_delay_by_hub, total_flights_by_hub, ncol = 1)

## Error in arrangeGrob(...):  object 'average_delay_by_hub'
not found
```

```
summary.tbl <-
  flights_hubs %>%
  group_by(Origin, Year, Month, DayofMonth) %>%
  summarise(n = n(), total_depdelay = sum(DepDelay, na.rm = TRUE) / n())

summary.tbl %>% select(-n) %>%
  tidyr::spread(Origin, total_depdelay) %>%
  tidyr::gather(Origin, dep_delay, DFW:ORD) %>%
  ggplot(aes(x = ATL, y = dep_delay, color = Origin)) +
  geom_point(alpha = 0.051) +
  guides(color = guide_legend(override.aes = list(size = 4))) +
  theme(legend.position = c(0.8, 0.8)) +
  geom_smooth() +
  geom_abline(slope = 1) +
  coord_equal() +
  xlim(-10,60) + ylim(-10,60) +
  facet_grid(Origin ~ .)

## Warning:  Removed 226 rows containing non-finite
values (stat_smooth).
## Warning:  Removed 226 rows containing missing values
(geom_point).
```

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

Compiled in 19.5275007327398