326.212 Final Project: Part 2

2018-11120 양은주

```
con = dbConnect(SQLite(), "project.sqlite")
```

Part 2. Basic Questions (150 pts)

Q1. Monthly traffic in three airports (60 pts)

1. We first start by exploring the airports table. Using `dplyr::filter()`, find out which airports the codes "SNA", "SJC", and "SMF" belong to.

```
SNA.SJC.SMF = dplyr::tbl(con, "airports") %>%
  filter(IATA %in% c("SNA", "SJC", "SMF")) %>%
  collect() %>%
  print(width = Inf)
```

```
## # A tibble: 3 x 14
##   `Airport ID` Name                                        City
##          <dbl> <chr>                                       <chr>
## 1         3748 Norman Y. Mineta San Jose International Airport San Jose
## 2         3817 Sacramento International Airport            Sacramento
## 3         3867 John Wayne Airport-Orange County Airport    Santa Ana
##   Country       IATA  ICAO  Latitude Longitude Altitude Timezone DST
##   <chr>         <chr> <chr>    <dbl>     <dbl>    <dbl>    <dbl> <chr>
## 1 United States SJC   KSJC      37.4     -122.       62       -8 A
## 2 United States SMF   KSMF      38.7     -122.       27       -8 A
## 3 United States SNA   KSNA      33.7     -118.       56       -8 A
##   `Tz database time zone` Type    Source
##   <chr>                   <chr>   <chr>
## 1 America/Los_Angeles     airport OurAirports
## 2 America/Los_Angeles     airport OurAirports
## 3 America/Los_Angeles     airport OurAirports
```

코드가 "SNA", "SJC", "SMF"인 공항은 각각 John Wayne Airport-Orange County Airport, Norman Y. Mineta San Jose International Airport, Sacramento International Airport였다.

2. Aggregate the counts of flights to all three of these airports at the monthly level (in the flights table) into a new data frame `airportcounts`. You may find `dplyr` functions `group_by()`, `summarise()`, `collect()`, and the pipe operator `%>%` useful.

```
airportcounts = dplyr::tbl(con, "flights") %>%
  filter(Cancelled == 0) %>%
  select(Year, Month, Dest) %>%
  group_by(Year, Month) %>%
  summarise(SNA = sum(Dest == "SNA", na.rm = TRUE),
            SJC = sum(Dest == "SJC", na.rm = TRUE),
            SMF = sum(Dest == "SMF", na.rm = TRUE)) %>%
  ungroup() %>%
  collect() %>%
  print(width = Inf)
```

```
## # A tibble: 216 x 5
##     Year Month   SNA   SJC   SMF
##    <dbl> <dbl> <int> <int> <int>
##  1  2001     1  3561  6144  3261
##  2  2001     2  3183  5532  2902
##  3  2001     3  3560  6221  3322
##  4  2001     4  3467  6090  3274
##  5  2001     5  3660  6381  3411
##  6  2001     6  3673  6293  3443
##  7  2001     7  3860  6540  3655
##  8  2001     8  3927  6585  3680
##  9  2001     9  2883  4732  2846
## 10  2001    10  3465  5533  3342
## # ... with 206 more rows
```

3. Add a new column to airportcounts by constructing a `Date` variable from the variables `Year` and `Month` (using helper functions from the `lubridate` package). Sort the rows in ascending order of dates.

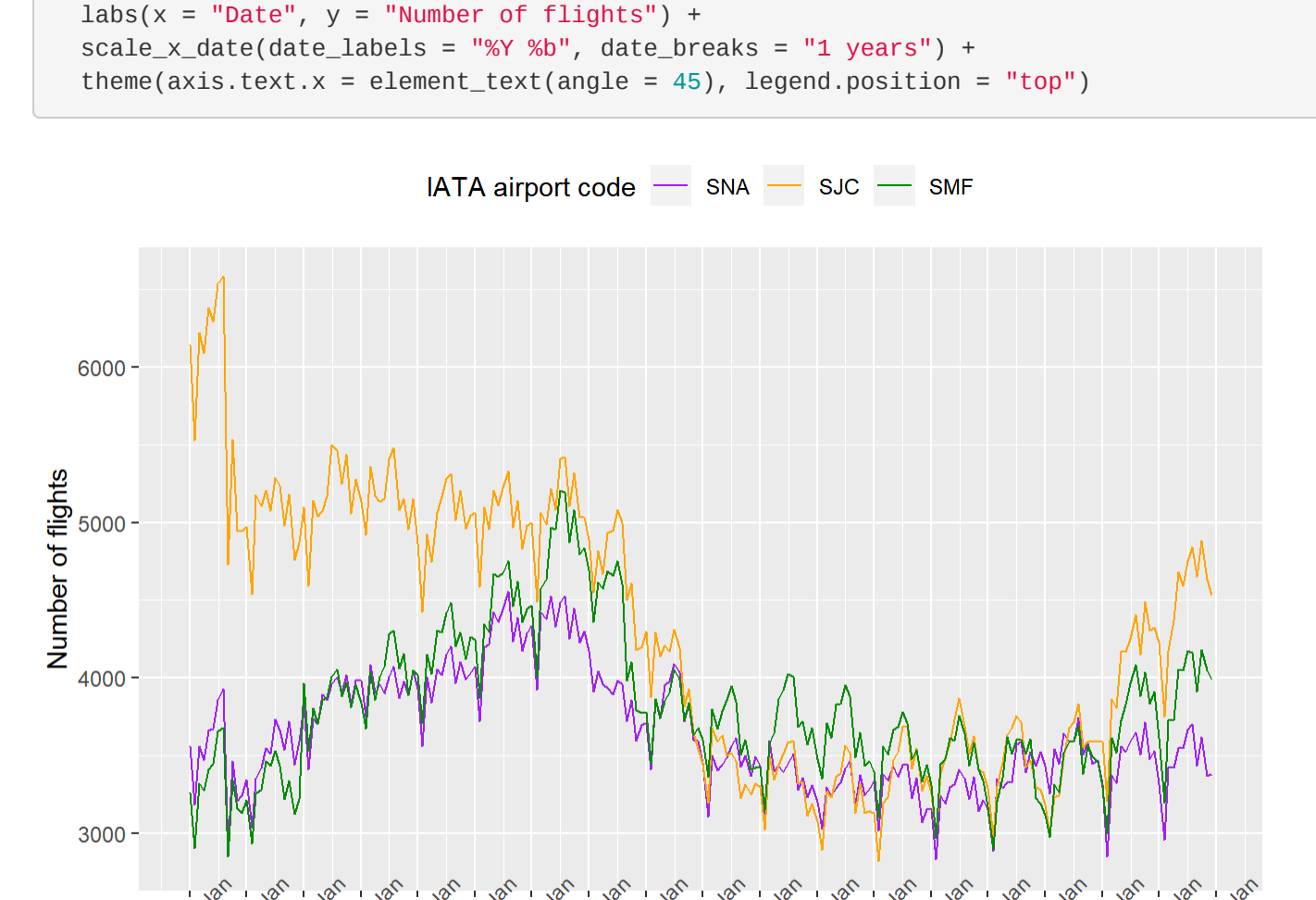
```
airportcounts = airportcounts %>%
  mutate(Date = make_date(Year, Month, 1)) %>%
  arrange(Date) %>%
  print(width = Inf)
```

```
## # A tibble: 216 x 6
##     Year Month   SNA   SJC   SMF Date
##    <dbl> <dbl> <int> <int> <int> <date>
##  1  2001     1  3561  6144  3261 2001-01-01
##  2  2001     2  3183  5532  2902 2001-02-01
##  3  2001     3  3560  6221  3322 2001-03-01
##  4  2001     4  3467  6090  3274 2001-04-01
##  5  2001     5  3660  6381  3411 2001-05-01
##  6  2001     6  3673  6293  3443 2001-06-01
##  7  2001     7  3860  6540  3655 2001-07-01
##  8  2001     8  3927  6585  3680 2001-08-01
##  9  2001     9  2883  4732  2846 2001-09-01
## 10  2001    10  3465  5533  3342 2001-10-01
## # ... with 206 more rows
```

`lubridate::make_date()` 함수를 사용하여 `<date>` type의 object를 생성하기 위해서는 year, month, day를 전부 지정해야 하기 때문에 임의로 1일에 day를 맞추었다.

4. From `airportcounts`, generate a time series plot that plots the number of flights per month in each of the three airports in chronological order.

```
ggplot(data = airportcounts) +
  geom_line(mapping = aes(x = Date, y = SNA, colour = "purple")) +
  geom_line(mapping = aes(x = Date, y = SJC, colour = "orange")) +
  geom_line(mapping = aes(x = Date, y = SMF, colour = "green4")) +
  scale_color_identity(name = "IATA airport code",
                       breaks = c("purple", "orange", "green4"),
                       labels = c("SNA", "SJC", "SMF"),
                       guide = "legend") +
  labs(x = "Date", y = "Number of flights") +
  scale_x_date(date_labels = "%Y %b", date_breaks = "1 years") +
  theme(axis.text.x = element_text(angle = 45), legend.position = "top")
```



2007년 이전까지는 "SJC" 공항이 우위를 점했지만 이후에는 각 공항으로 향하는 항공편의 수가 거의 차이가 나지 않았다. 덧붙이면 겨울에는 항공편의 수가 줄었다가 여름에 다시 늘어나는 연주기성을 공통적으로 관찰할 수 있었다.

5. Find the top ten months (like 2001-09) with the largest number of flights for each of the three airports.

```
SNA = airportcounts %>%
  arrange(desc(SNA)) %>%
  .[1:10, ]
SJC = airportcounts %>%
  arrange(desc(SJC)) %>%
  .[1:10, ]
SMF = airportcounts %>%
  arrange(desc(SMF)) %>%
  .[1:10, ]
```

```
paste(SNA$Year, sprintf("%02d", SNA$Month), sep = "-")
```

```
##  [1] "2006-08" "2007-08" "2007-05" "2007-07" "2007-10" "2006-07" "2006-05"
##  [8] "2007-03" "2006-10" "2007-04"
```

"SNA" 공항으로 향하는 항공편이 가장 많았던 상위 10개의 달은 2006-08, 2007-08, 2007-05, 2007-07, 2007-10, 2006-07, 2006-05, 2007-03, 2006-10, 2007-04이었다.

```
paste(SJC$Year, sprintf("%02d", SJC$Month), sep = "-")
```

```
##  [1] "2001-08" "2001-07" "2001-05" "2001-06" "2001-03" "2001-01" "2001-04"
##  [8] "2001-10" "2001-02" "2003-07"
```

"SJC" 공항으로 향하는 항공편이 가장 많았던 상위 10개의 달은 2001-08, 2001-07, 2001-05, 2001-06, 2001-03, 2001-01, 2001-04, 2001-10, 2001-02, 2003-07이었다.

```
paste(SMF$Year, sprintf("%02d", SMF$Month), sep = "-")
```

```
##  [1] "2007-07" "2007-08" "2007-10" "2007-05" "2007-06" "2007-09" "2007-12"
##  [8] "2007-11" "2008-07" "2006-08"
```

"SMF" 공항으로 향하는 항공편이 가장 많았던 상위 10개의 달은 2007-07, 2007-08, 2007-10, 2007-05, 2007-06, 2007-09, 2007-12, 2007-11, 2008-07, 2006-08이었다.

Q2. Finding reliable airlines (60 pts)

Which airline was most reliable flying from Chicago O'Hare (ORD) to Minneapolis/St. Paul (MSP) in Year 2015?

1. Create a data frame `delays` that contains the average arrival delay for each day in 2015 for four airlines: United (UA), Northwest (NW), American (AA), and Delta (DL). Your data frame must contain only necessary variables, to save the memory space.

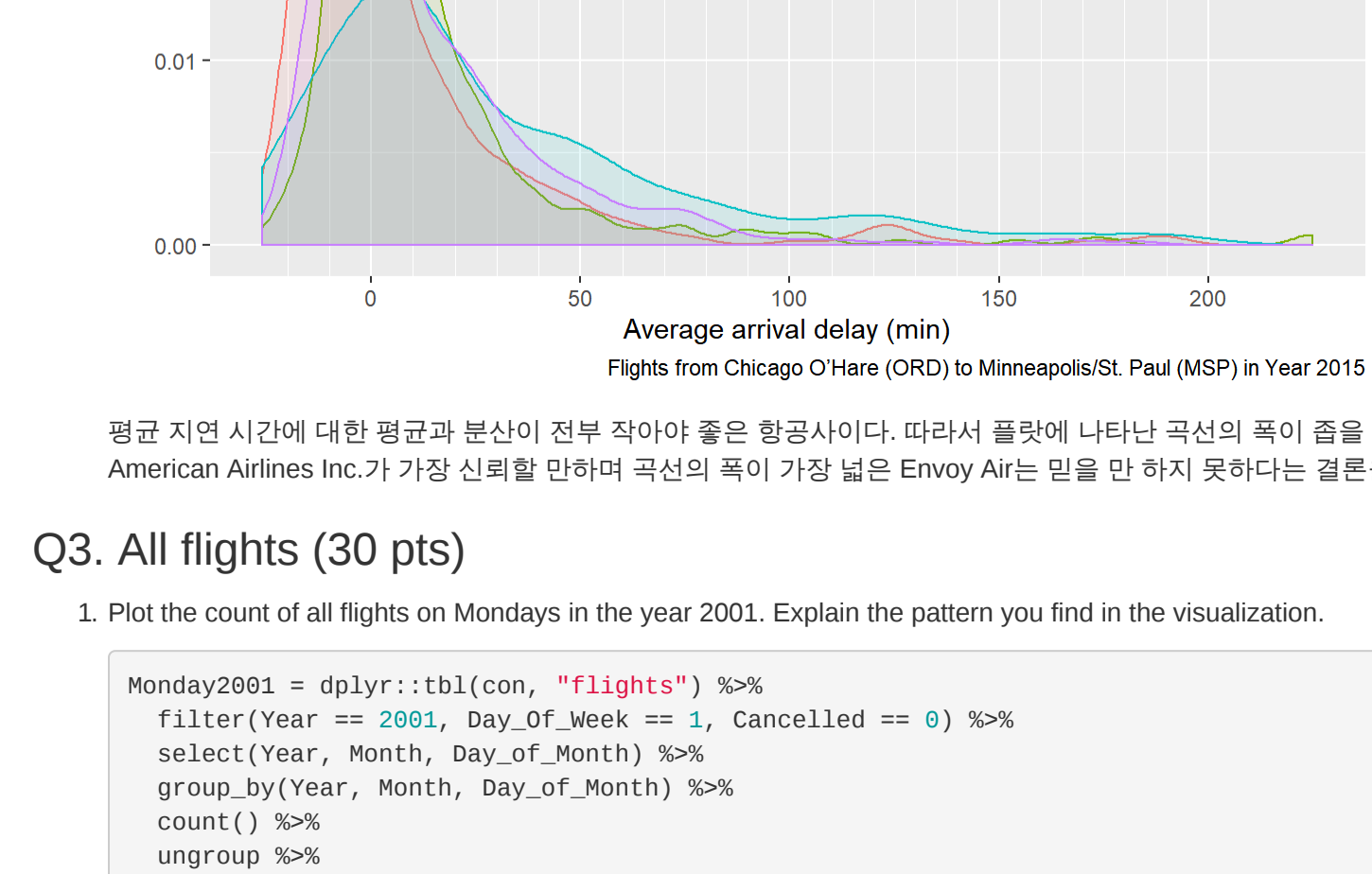
```
delays = dplyr::tbl(con, "flights") %>%
  filter(Origin == "ORD", Dest == "MSP", Year == 2015,
         Op_Unique_Carrier %in% c("UA", "DL", "AA", "MQ"), Cancelled == 0) %>%
  select(Year, Month, Day_of_Month, Op_Unique_Carrier, Arr_Delay) %>%
  group_by(Year, Month, Day_of_Month, Op_Unique_Carrier) %>%
  summarise(Avg_Arr_Delay = mean(Arr_Delay, na.rm = TRUE)) %>%
  ungroup() %>%
  collect() %>%
  print(width = Inf)
```

```
## # A tibble: 1,188 x 5
##     Year Month Day_of_Month Op_Unique_Carrier Avg_Arr_Delay
##    <dbl> <dbl>        <dbl> <chr>                     <dbl>
##  1  2015     1            1 AA                        120.
##  2  2015     1            1 DL                        -12.5
##  3  2015     1            1 MQ                          2.33
##  4  2015     1            2 AA                        -10.5
##  5  2015     1            2 DL                         -1
##  6  2015     1            2 MQ                          3.33
##  7  2015     1            3 AA                        124.
##  8  2015     1            3 DL                         48.5
##  9  2015     1            3 MQ                         20
## 10  2015     1            4 AA                        101.
## # ... with 1,178 more rows
```

2. Compare the average delay of the four airlines by generating density plots comparing them in a single panel. In doing this, use a join function to provide the full names of the airlines in the legend of the plot. Which airline is the most reliable? Which is the least?

```
airlines = dplyr::tbl(con, "airlines") %>%
  collect()
```

```
ggplot(data = left_join(delays, airlines, by = c("Op_Unique_Carrier" = "Code"))) +
  geom_density(mapping = aes(x = Avg_Arr_Delay, fill = Description, colour = Description), alpha = 0.1) +
  labs(x = "Average arrival delay (min)",
       y = "Density",
       fill = "Airlines",
       colour = "Airlines",
       caption = "Flights from Chicago O'Hare (ORD) to Minneapolis/St. Paul (MSP) in Year 2015") +
  scale_x_continuous(minor_breaks = seq(-20, 200, 10)) +
  theme(legend.position = "top")
```



평균 지연 시간에 대한 평균과 분산이 전부 작아야 좋은 항공사이다. 따라서 플랏에 나타난 곡선의 폭이 좁을 뿐만 아니라 왼쪽으로 몰린 American Airlines Inc.가 가장 신뢰할 만하며 곡선의 폭이 가장 넓은 Envoy Air는 믿을 만 하지 못하다는 결론을 내릴 수 있었다.

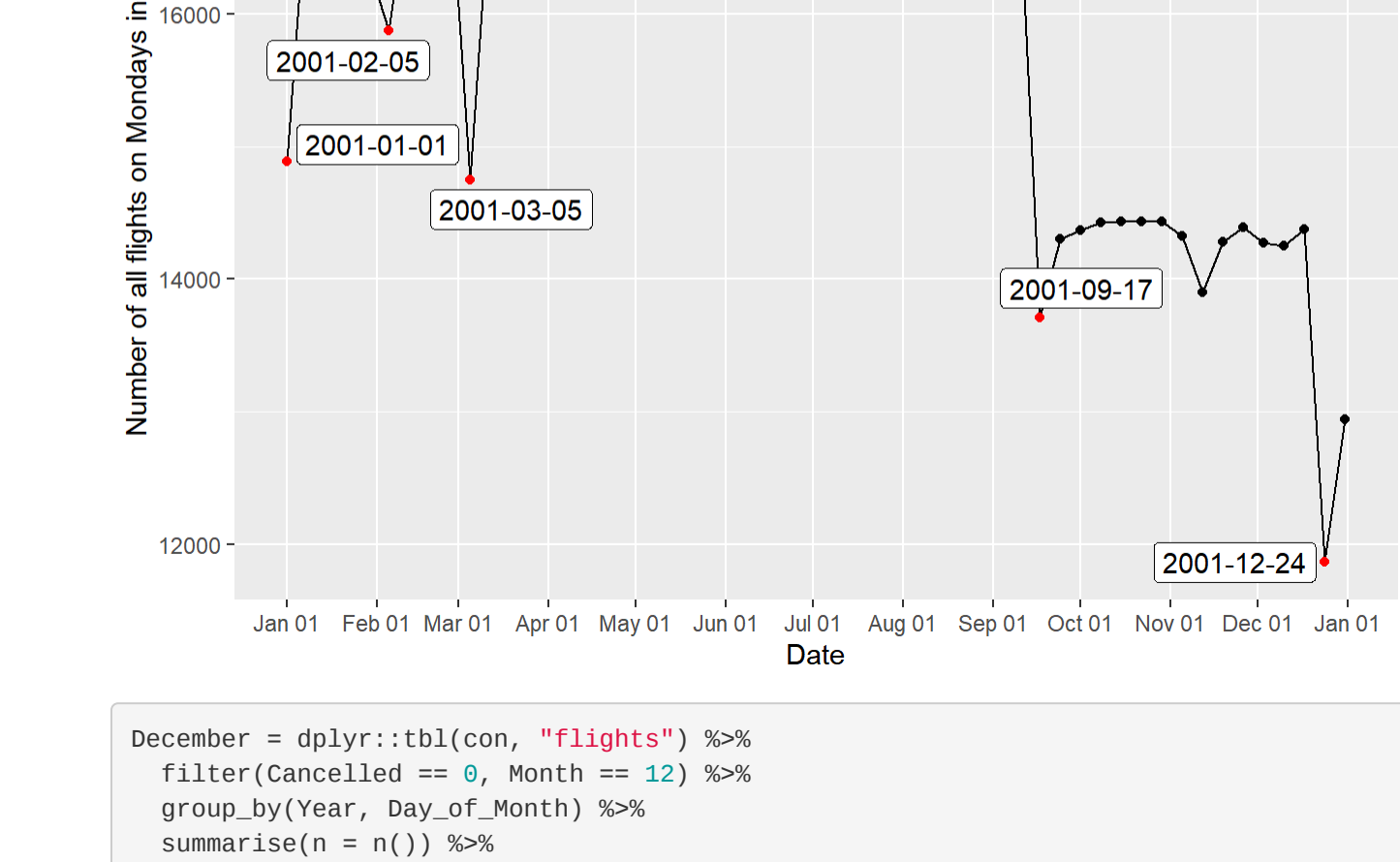
Q3. All flights (30 pts)

1. Plot the count of all flights on Mondays in the year 2001. Explain the pattern you find in the visualization.

```
Monday2001 = dplyr::tbl(con, "flights") %>%
  filter(Year == 2001, Day_Of_Week == 1, Cancelled == 0) %>%
  select(Year, Month, Day_of_Month) %>%
  group_by(Year, Month, Day_of_Month) %>%
  count() %>%
  ungroup %>%
  collect() %>%
  mutate(Date = make_date(Year, Month, Day_of_Month))
```

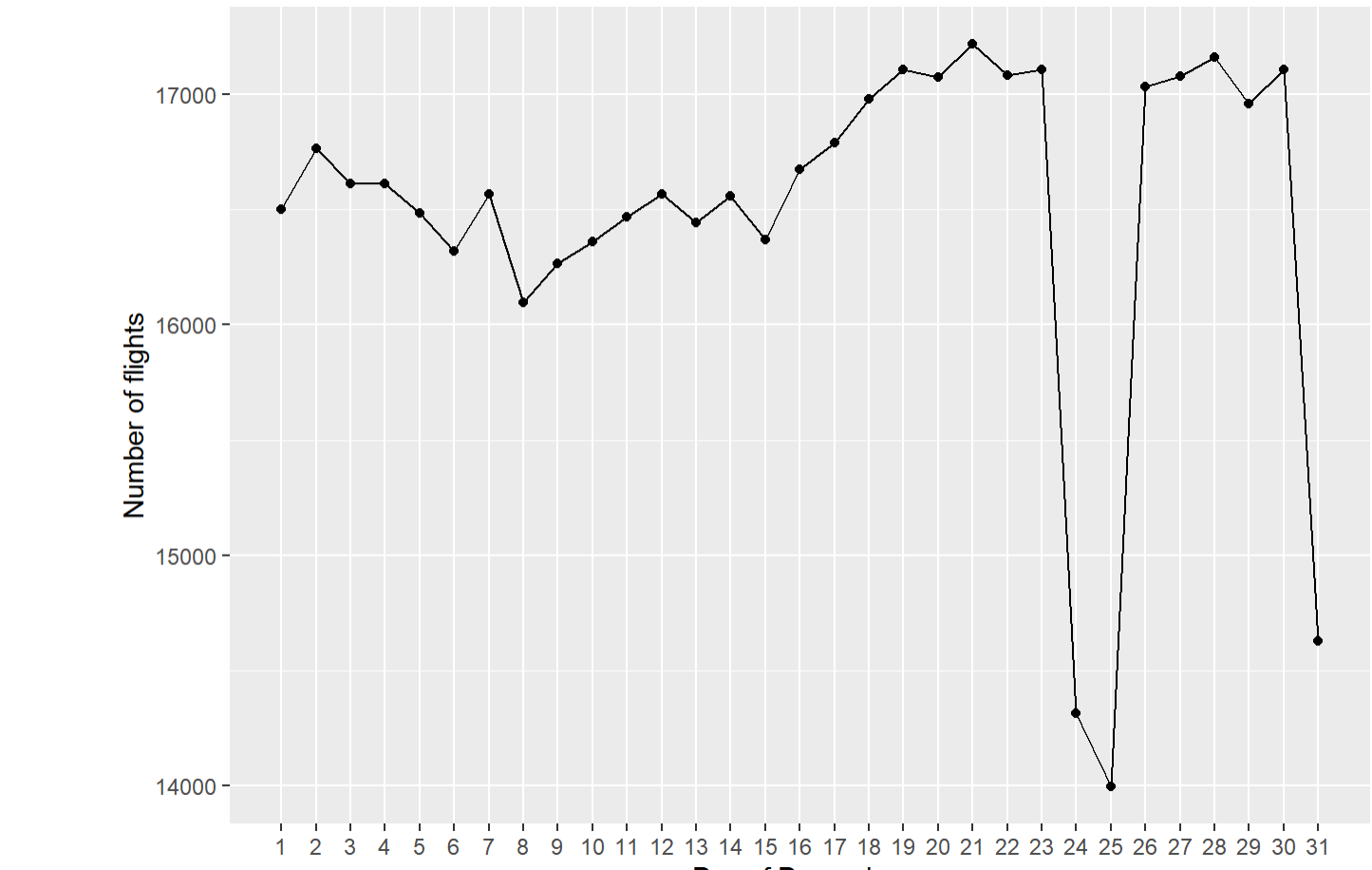
```
Monday2001Label = Monday2001 %>%
  mutate(lag_n = lag(n), lead_n = lead(n), before = lag_n - n, after = lead_n - n) %>%
  filter(row_number(desc(before)) < 5 | row_number(desc(after)) < 5)
```

```
ggplot(data = Monday2001, mapping = aes(x = Date, y = n)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_labels = "%b %d", date_breaks = "1 months", minor_breaks = NULL) +
  labs(y = "Number of all flights on Mondays in 2001") +
  geom_point(colour = "red", data = Monday2001Label) +
  geom_label_repel(data = Monday2001Label, mapping = aes(label = as.character(Date)))
```




```
December = dplyr::tbl(con, "flights") %>%
  filter(Cancelled == 0, Month == 12) %>%
  group_by(Year, Day_of_Month) %>%
  summarise(n = n()) %>%
  group_by(Day_of_Month) %>%
  summarise(n = mean(n)) %>%
  collect()
```

```
ggplot(data = December, mapping = aes(x = Day_of_Month, y = n)) +
  geom_point() +
  geom_line() +
  labs(x = "Day of December", y = "Number of flights") +
  scale_x_continuous(breaks = seq(1, 31, 1), minor_breaks = NULL)
```



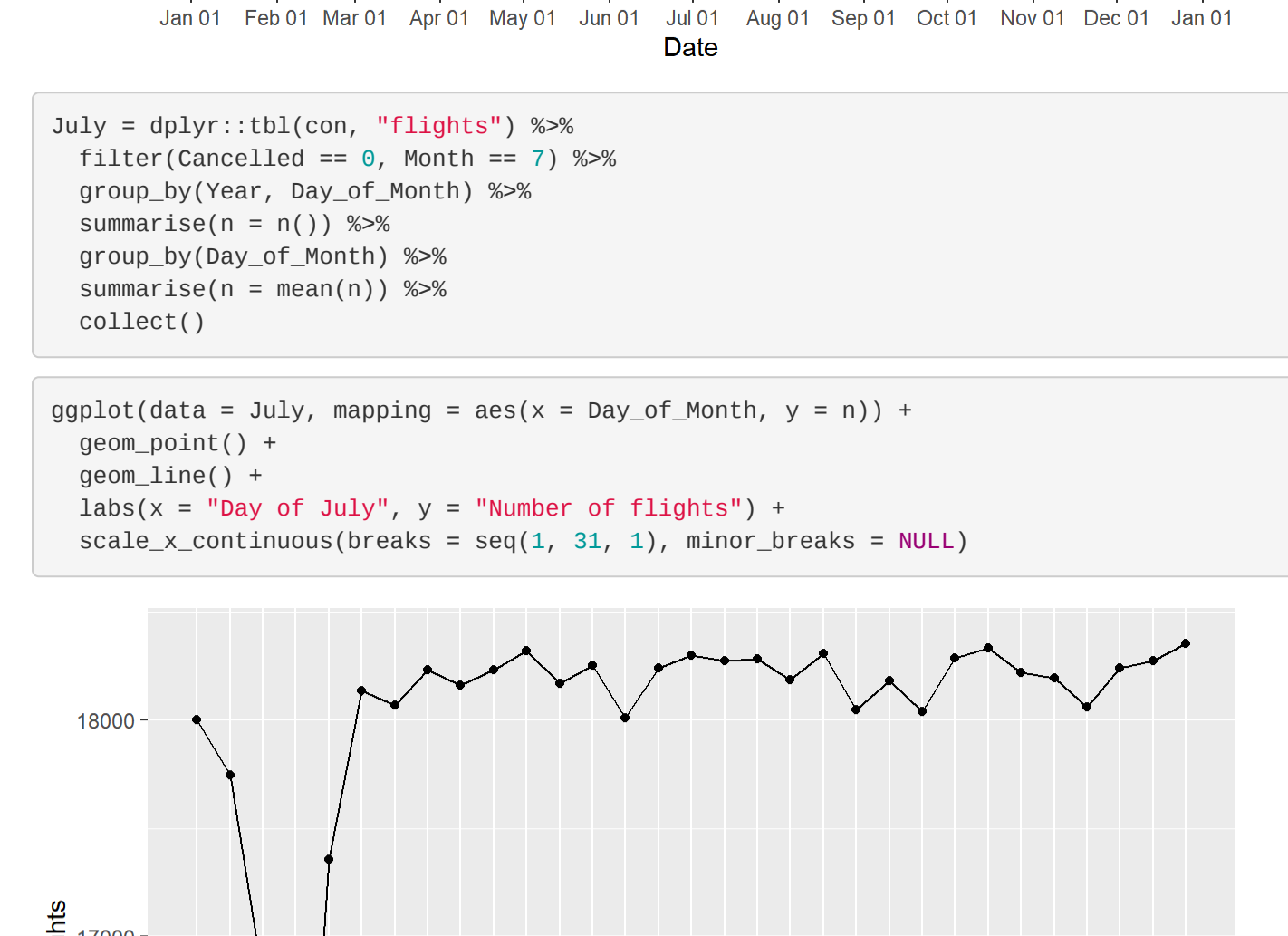
항공편의 수가 큰 폭으로 변동하는 시점을 한눈에 알아볼 수 있게 하기 위해서 `Monday2001Label` 테이블을 생성한 뒤 `geom_label_repel()` 함수를 적용했다. 먼저 9월 17일에 항공편의 수가 급락하여 2001년 말까지 회복되지 않은 것이 눈에 띈다. 이는 2001년 9월 11일에 발생한 최악의 항공기 테러의 영향을 받은 것으로 생각된다[1]. 이를 감안해도 12월 24일에는 유난히 항공편의 수가 적었는데 `December` 테이블에 12월의 각 일마다 연별 평균 항공편의 수를 집계하여 플랏팅한 결과 휴일이거나 휴일 근방인 12월 24일, 25일, 31일에는 항상 항공편의 수가 줄어들었다는 사실을 확인할 수 있었다. 한편, 미국에서 5월의 마지막 Monday는 공휴일인 Memorial Day이며 1월 1일 역시 공휴일인 New Years Day이다. 연휴 기간에는 항공편의 수가 줄어든다는 것을 일반화한다면 1월 1일과 Memorial Day인 5월 28일에 있었던 일시적인 하락 역시 설명할 수 있을 것으로 생각한다. 3월 5일의 하락은 북동부 지역에 있었던 폭설에 영향을 받았던 듯 하지만 `flights` 데이터에서 결항 이유를 나타내는 `Cancellation_Code` 의 값은 2003년부터 제공되기 때문에 정확히 검증할 수는 없었다[2].

2. Repeat 1 for the year 2011.

```
Monday2011 = dplyr::tbl(con, "flights") %>%
  filter(Year == 2011, Day_Of_Week == 1, Cancelled == 0) %>%
  select(Year, Month, Day_of_Month) %>%
  group_by(Year, Month, Day_of_Month) %>%
  count() %>%
  ungroup %>%
  collect() %>%
  mutate(Date = make_date(Year, Month, Day_of_Month))
```

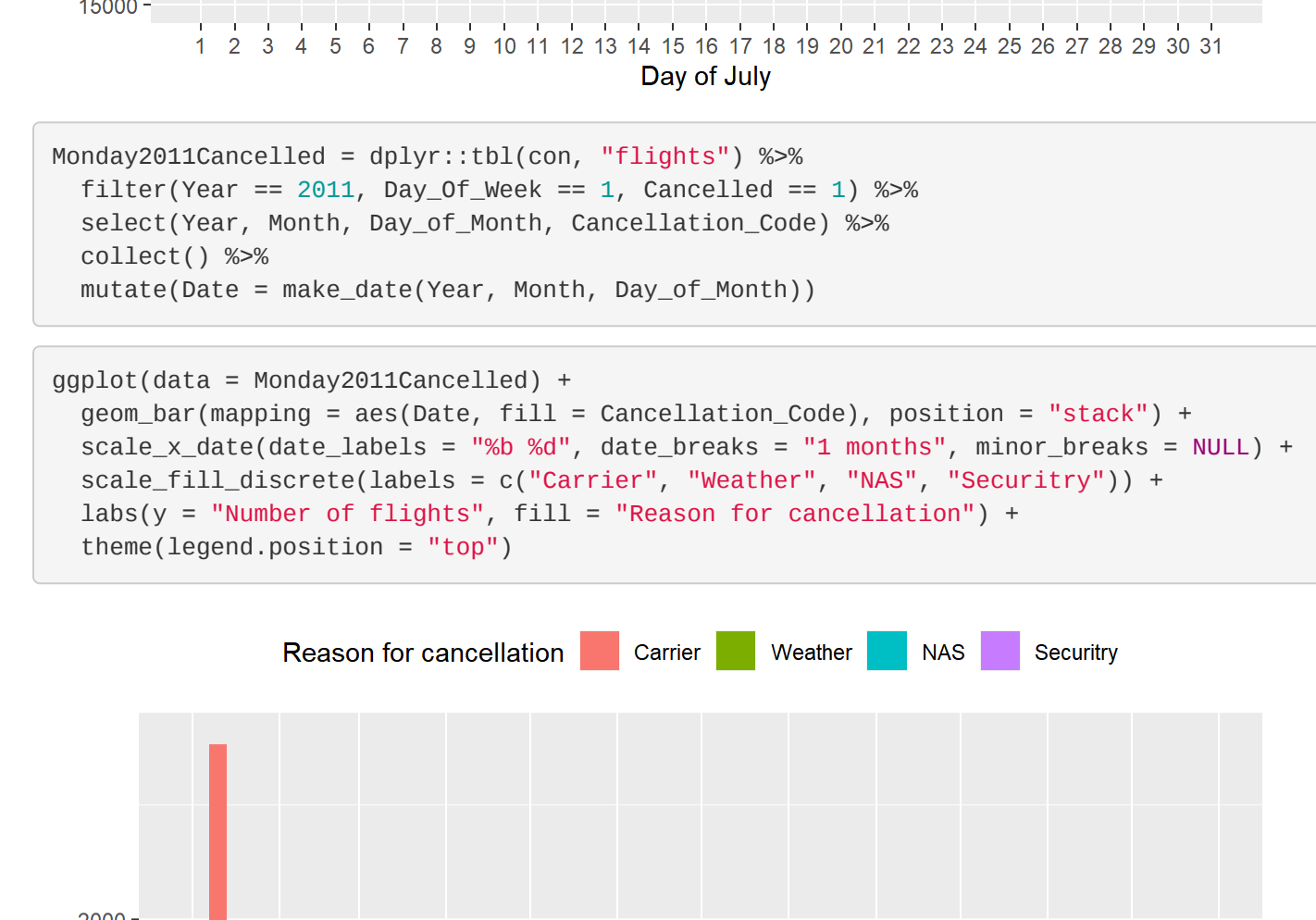
```
Monday2011Label = Monday2011 %>%
  mutate(lag_n = lag(n), lead_n = lead(n), before = lag_n - n, after = lead_n - n) %>%
  filter(row_number(desc(before)) < 5 | row_number(desc(after)) < 5)
```

```
ggplot(data = Monday2011, mapping = aes(x = Date, y = n)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_labels = "%b %d", date_breaks = "1 months", minor_breaks = NULL) +
  labs(y = "Number of all flights on Mondays in 2011") +
  geom_point(colour = "red", data = Monday2011Label) +
  geom_label_repel(data = Monday2011Label, mapping = aes(label = as.character(Date)))
```



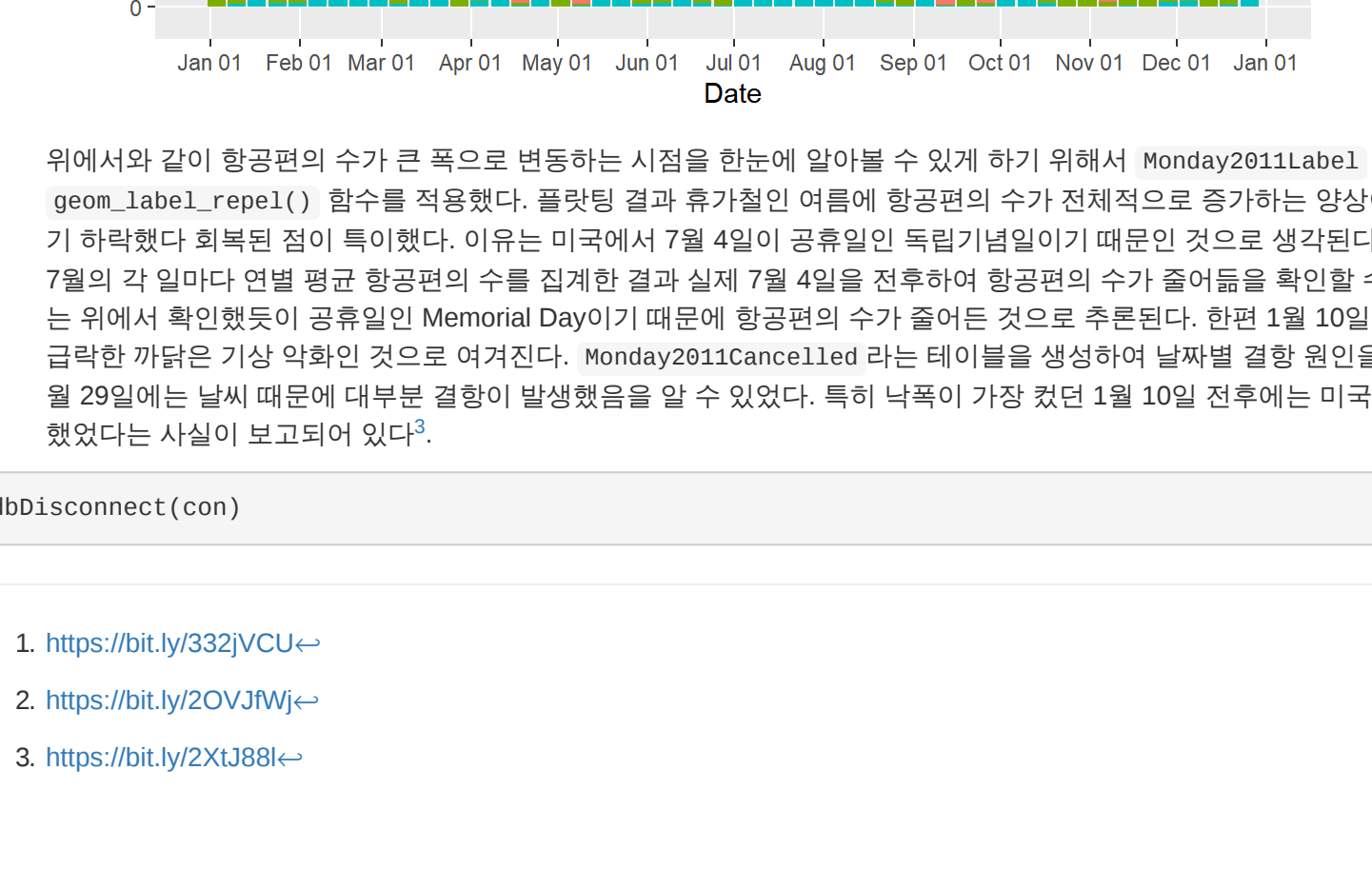

```
July = dplyr::tbl(con, "flights") %>%
  filter(Cancelled == 0, Month == 7) %>%
  group_by(Year, Day_of_Month) %>%
  summarise(n = n()) %>%
  group_by(Day_of_Month) %>%
  summarise(n = mean(n)) %>%
  collect()
```

```
ggplot(data = July, mapping = aes(x = Day_of_Month, y = n)) +
  geom_point() +
  geom_line() +
  labs(x = "Day of July", y = "Number of flights") +
  scale_x_continuous(breaks = seq(1, 31, 1), minor_breaks = NULL)
```




```
Monday2011Cancelled = dplyr::tbl(con, "flights") %>%
  filter(Year == 2011, Day_Of_Week == 1, Cancelled == 1) %>%
  select(Year, Month, Day_of_Month, Cancellation_Code) %>%
  collect() %>%
  mutate(Date = make_date(Year, Month, Day_of_Month))
```

```
ggplot(data = Monday2011Cancelled) +
  geom_bar(mapping = aes(Date, fill = Cancellation_Code), position = "stack") +
  scale_x_date(date_labels = "%b %d", date_breaks = "1 months", minor_breaks = NULL) +
  scale_fill_discrete(labels = c("Carrier", "Weather", "NAS", "Securitry")) +
  labs(y = "Number of flights", fill = "Reason for cancellation") +
  theme(legend.position = "top")
```



위에서와 같이 항공편의 수가 큰 폭으로 변동하는 시점을 한눈에 알아볼 수 있게 하기 위해서 `Monday2011Label` 테이블을 생성한 뒤 `geom_label_repel()` 함수를 적용했다. 플랏팅 결과 휴가철인 여름에 항공편의 수가 전체적으로 증가하는 양상이었지만 7월 4일에만 갑자기 하락했다가 회복된 점이 특이했다. 이유는 미국에서 7월 4일이 공휴일인 독립기념일이기 때문인 것으로 생각된다. `July` 테이블을 생성하여 7월의 각 일마다 연별 평균 항공편의 수를 집계한 결과 실제 7월 4일을 전후하여 항공편의 수가 줄어듦을 확인할 수 있었다. 5월 30일의 경우는 위에서 확인했듯이 공휴일인 Memorial Day이기 때문에 항공편의 수가 줄어든 것으로 추론된다. 한편 1월 10일, 8월 29일에 항공편의 수가 급락한 까닭은 기상 악화인 것으로 여겨진다. `Monday2011Cancelled` 라는 테이블을 생성하여 날짜별 결항 원인을 플랏팅한 결과 1월 10일, 8월 29일에는 날씨 때문에 대부분 결항이 발생했음을 알 수 있었다. 특히 낙폭이 가장 컸던 1월 10일 전후에는 미국에 매우 강한 눈폭풍이 발생했었다는 사실이 보고되어 있다[3].

```
dbDisconnect(con)
```

1. https://bit.ly/332jVCU↩
2. https://bit.ly/2OVJfWj↩
3. https://bit.ly/2XtJ88l↩