Homework 6

Samriddh gupta

Model Building Homework

Libraries

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------ tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts ---------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(purrr)
library(modelr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

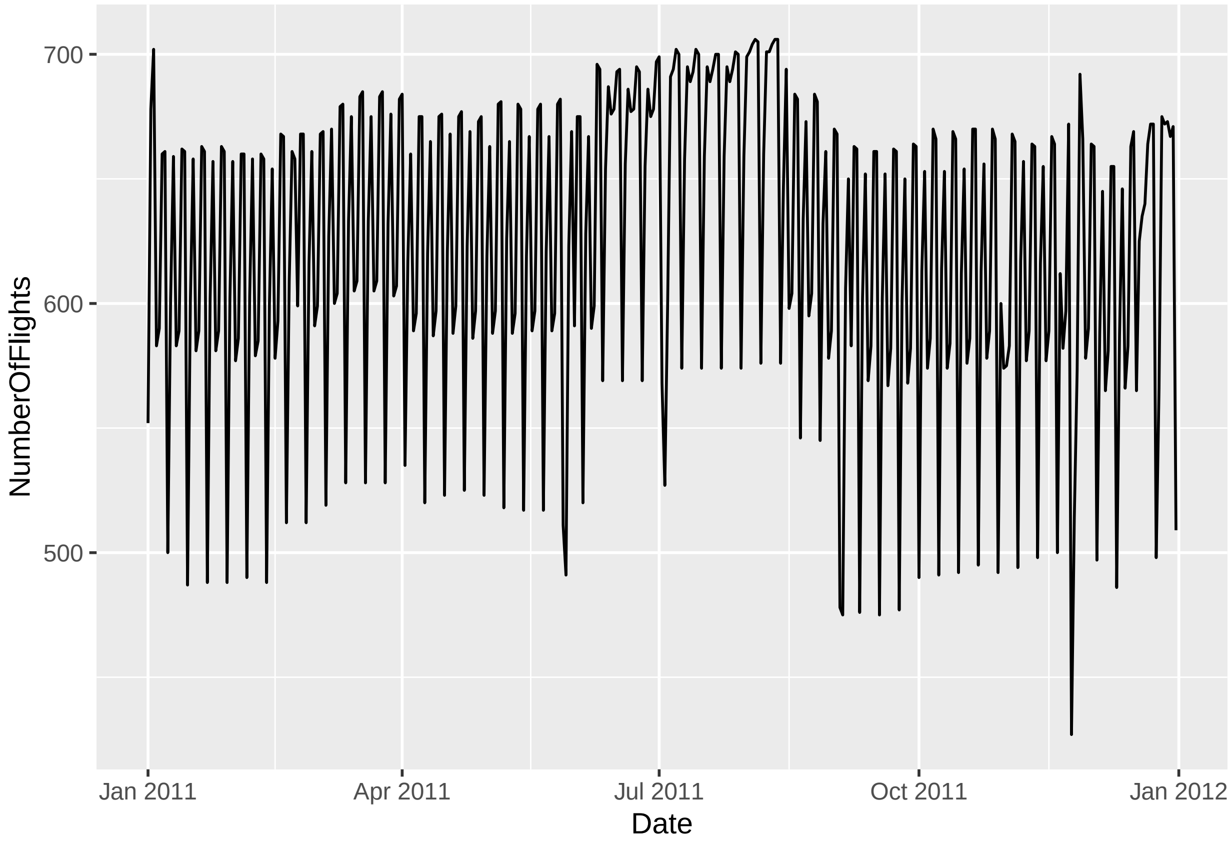
```
## The following object is masked from 'package:base':
##
##     date
```

1. (2 pts) Create a dataset from hflights that has the Date, Day of the Week, and the number of flights for that date. Visualize the data both as a line graph (# of flights vs Date) and as a boxplot for each day of the week.

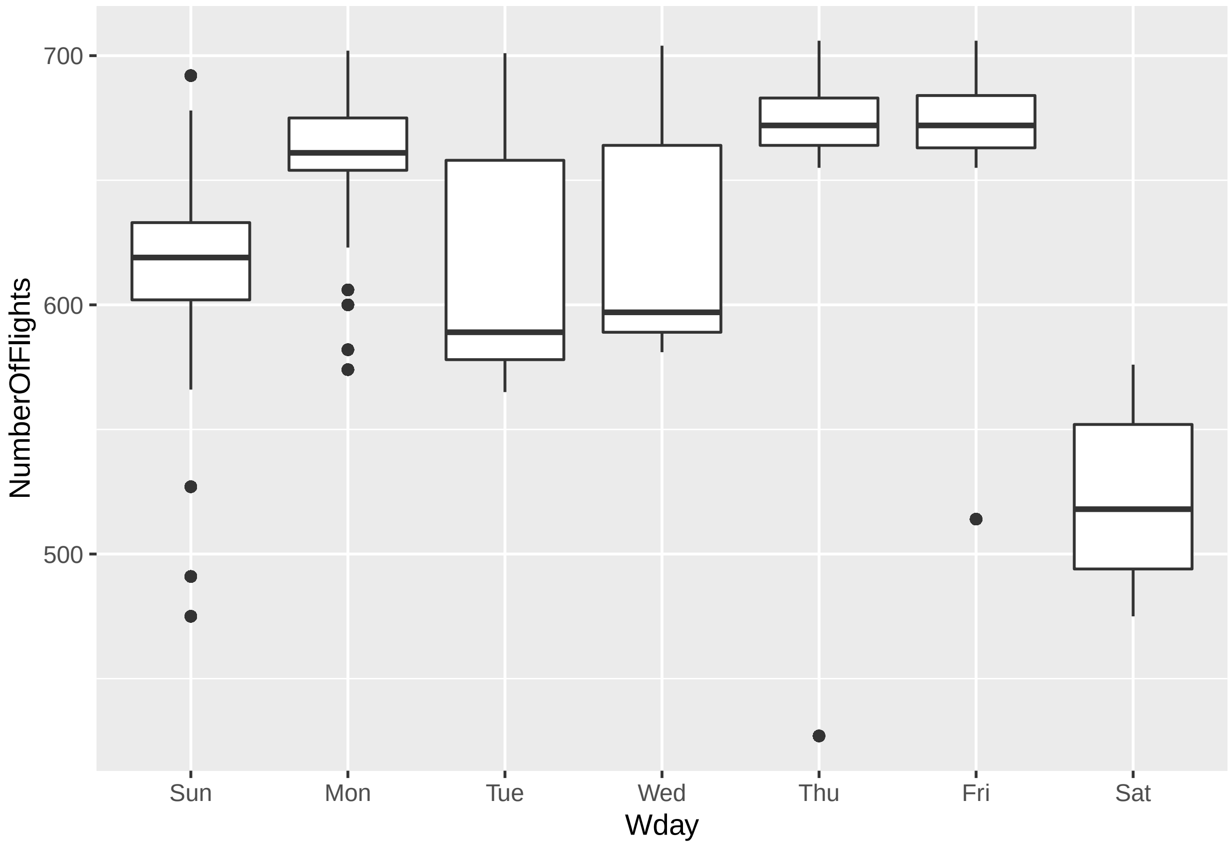
```
data1<-hflights::hflights

data1 %>%
  mutate(Date=make_date(Year,Month,DayofMonth)) %>%
  select(Date,DayOfWeek) %>%
  group_by(Date) %>%
  mutate(NumberOfFlights=n())->
  data2

data2<-data2 %>%
  mutate(Wday=wday(Date,label = TRUE))

head(data2)
```

```
## # A tibble: 6 x 4
## # Groups:   Date [6]
##   Date       DayOfWeek NumberOfFlights Wday
##   <date>         <int>           <int> <ord>
## 1 2011-01-01         6             552 Sat
## 2 2011-01-02         7             678 Sun
## 3 2011-01-03         1             702 Mon
## 4 2011-01-04         2             583 Tue
## 5 2011-01-05         3             590 Wed
## 6 2011-01-06         4             660 Thu
```

```
ggplot(data = data2,mapping = aes(Date,NumberOfFlights))+
  geom_line()
```



```r
g<-ggplot(data = data2,mapping = aes(Wday,NumberOfFlights))+
  geom_boxplot()

g
```

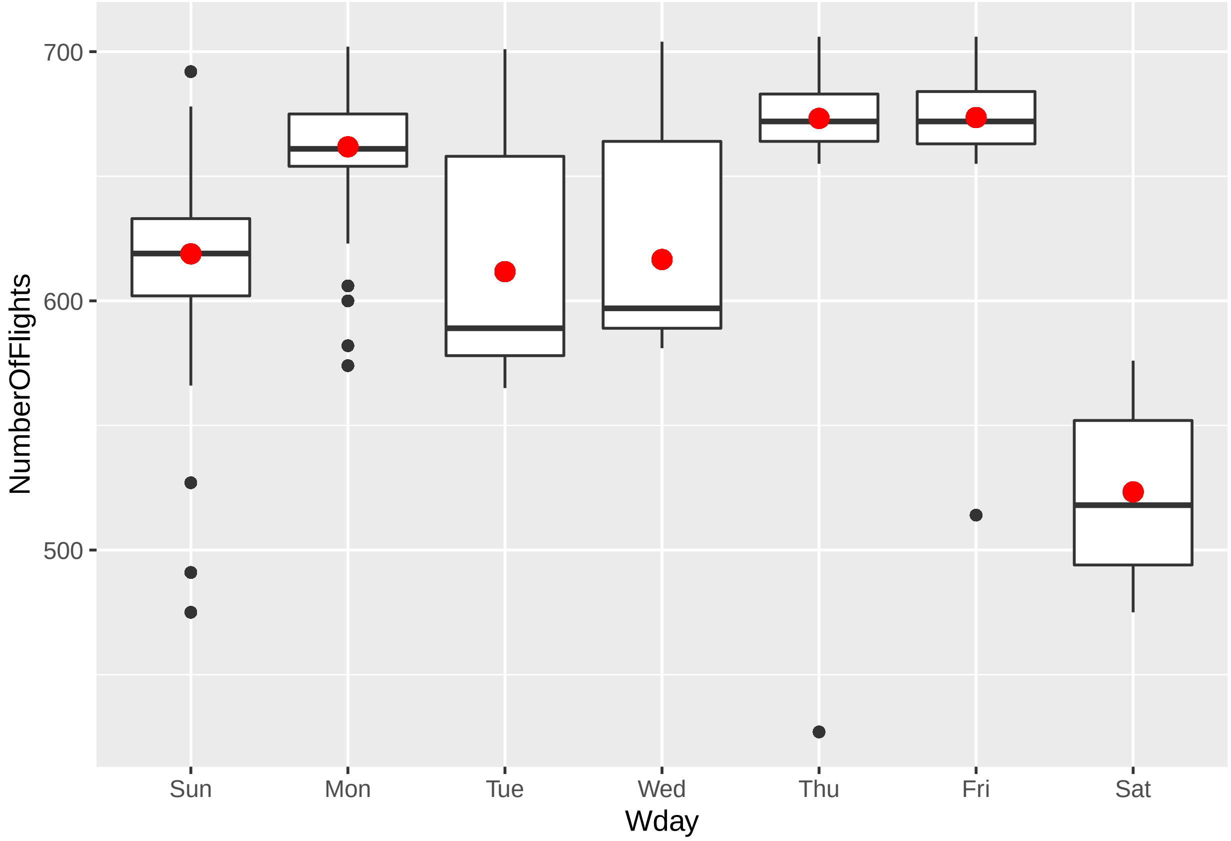


2. (1 pt) Construct a model using day of the week as the predictor. What does this model tell us? Visualize the residuals.

```r
mod <- lm(NumberOfFlights ~ Wday, data = data2)

grid <- data2 %>%
  data_grid(Wday) %>%
  add_predictions(mod, "NumberOfFlights")
grid
```

```
## # A tibble: 2,555 x 3
## # Groups:   Date [365]
##    Date       Wday  NumberOfFlights
##    <date>     <ord>           <dbl>
##  1 2011-01-01 Sun              619.
##  2 2011-01-01 Mon              662.
##  3 2011-01-01 Tue              612.
##  4 2011-01-01 Wed              617.
##  5 2011-01-01 Thu              673.
##   6 2011-01-01 Fri           674.
##   7 2011-01-01 Sat           523.
##   8 2011-01-02 Sun           619.
##   9 2011-01-02 Mon           662.
##  10 2011-01-02 Tue           612.
## # ... with 2,545 more rows
```

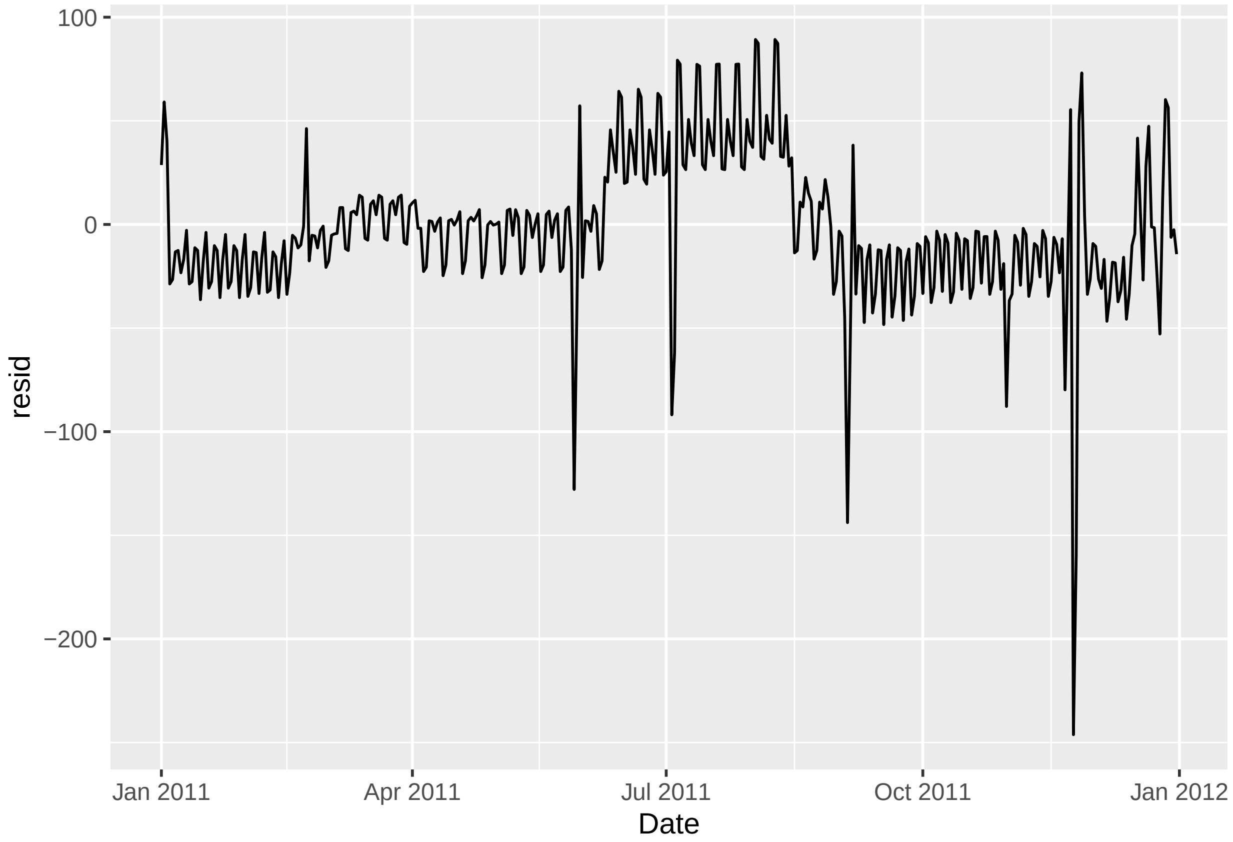
```
## The model predicts the Number of the flights on any given week day.
g+geom_point(data = grid,color = "red", size = 3)
```



```
data2 <- data2 %>%
  add_residuals(mod)
data2
```

```
## # A tibble: 227,496 x 5
## # Groups:   Date [365]
##    Date       DayOfWeek NumberOfFlights Wday   resid
##    <date>         <int>           <int> <ord>  <dbl>
##  1 2011-01-01         6             552 Sat     28.7
##  2 2011-01-02         7             678 Sun     59.1
##  3 2011-01-03         1             702 Mon     40.2
##  4 2011-01-04         2             583 Tue    -28.7
##  5 2011-01-05         3             590 Wed    -26.6
##  6 2011-01-06         4             660 Thu    -13.2
##  7 2011-01-07         5             661 Fri    -12.6
##   8 2011-01-08         6             500 Sat   -23.3
##   9 2011-01-09         7             602 Sun   -16.9
## 10 2011-01-10         1             659 Mon    -2.83
## # ... with 227,486 more rows
```

```
ggplot(data = data2,mapping = aes(Date,resid))+
  geom_line()
```



(2 pts) Add a variable to account for seasonal variation. You can adjust the breaks something like this (feel free to change the dates)

season <- function(date) { cut(date, breaks = ymd(20110101, 20110301, 20110605, 201130905, 20120101), labels = c("winter","spring", "summer", "fall")) }

mod1 <- lm(n ~ wday * season, data = daily) daily_res <- daily %>% add_residuals(mod1, "resid")

```
data2<-data2 %>%
  mutate(season=cut(Date,
                    breaks = ymd(20110101, 20110301, 20110605, 20110825, 20120101),
                    labels = c("winter","spring", "summer", "fall")))

mod1 <- lm(NumberOfFlights ~ Wday * season, data = data2)
daily_res <- data2 %>%
  add_residuals(mod1, "resid")

head(daily_res)
```

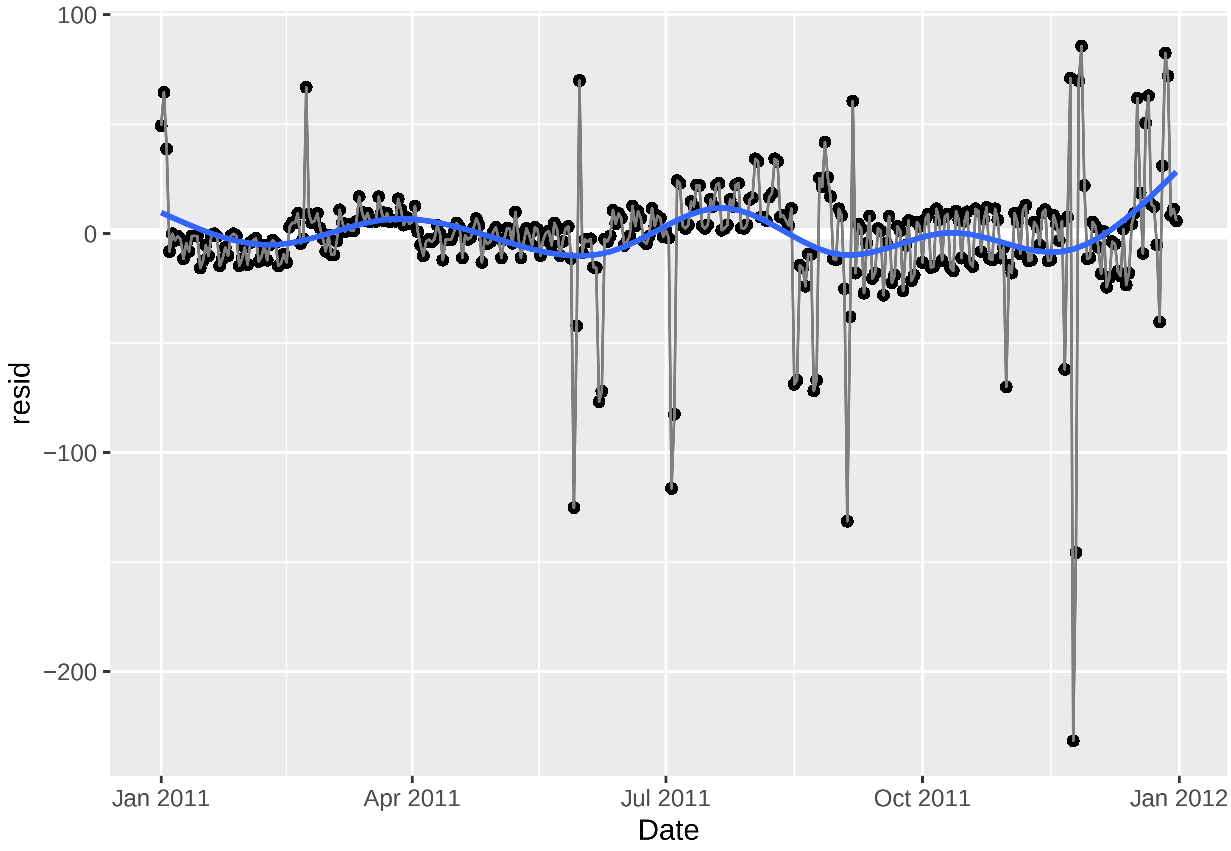
```
## # A tibble: 6 x 6
## # Groups:   Date [6]
##   Date       DayOfWeek NumberOfFlights Wday    resid season
##   <date>         <int>           <int> <ord>   <dbl> <fct>
## 1 2011-01-01         6             552 Sat   49.3    winter
## 2 2011-01-02         7             678 Sun   64.5    winter
## 3 2011-01-03         1             702 Mon   38.7    winter
## 4 2011-01-04         2             583 Tue   -8.13   winter
## 5 2011-01-05         3             590 Wed   -0.0284 winter
## 6 2011-01-06         4             660 Thu   -3.01   winter
```

4. (3 1/2 pts)

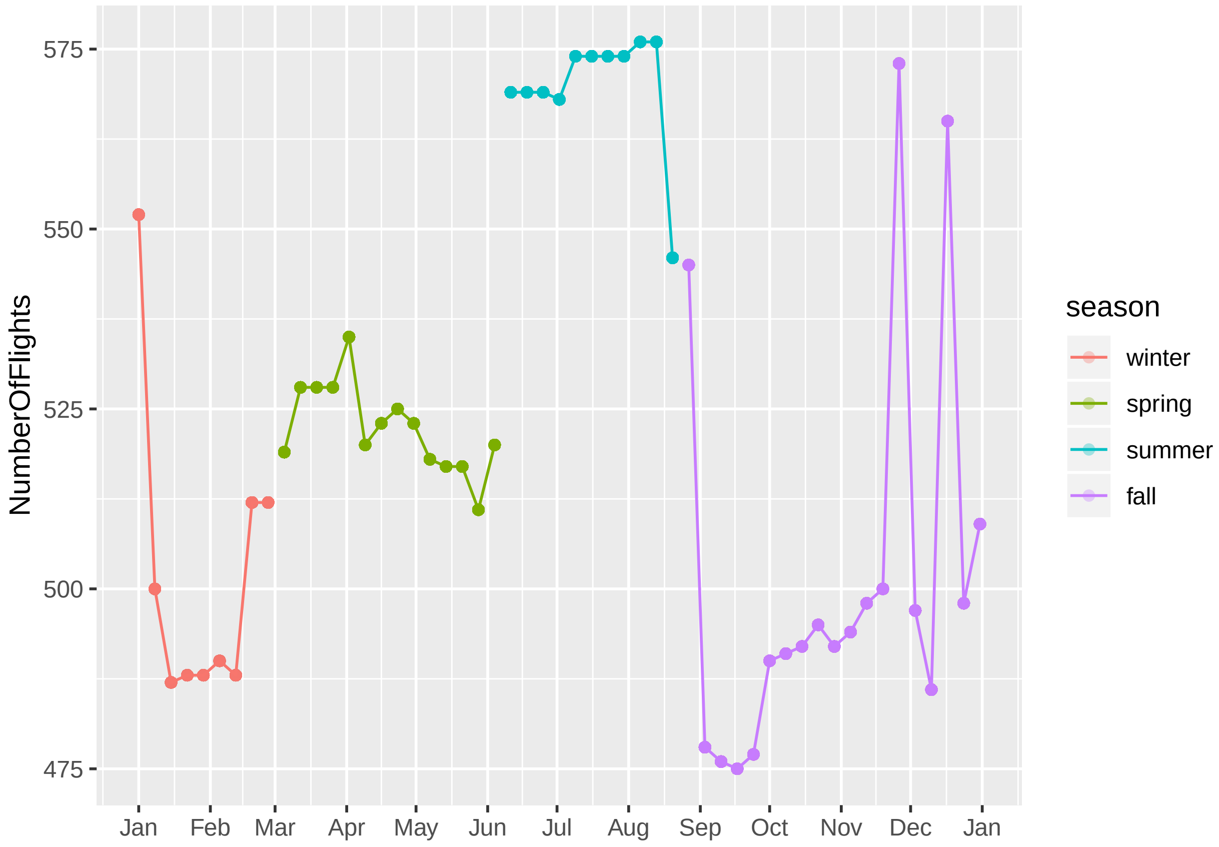
a) (1/2 pt) Identify the dates with the largest residual values. What do you think is the cause for the days with the highest and lowest residual values?

```
daily_res %>%
  filter(abs(resid) > 100 ) %>%
  ggplot(data= daily_res,mapping=aes(Date, resid)) +
  geom_ref_line(h = 0) +
  geom_point(alpha = 1/3) +
  geom_line(color = "grey50") +
  geom_smooth(se = FALSE, span = 0.20)
```

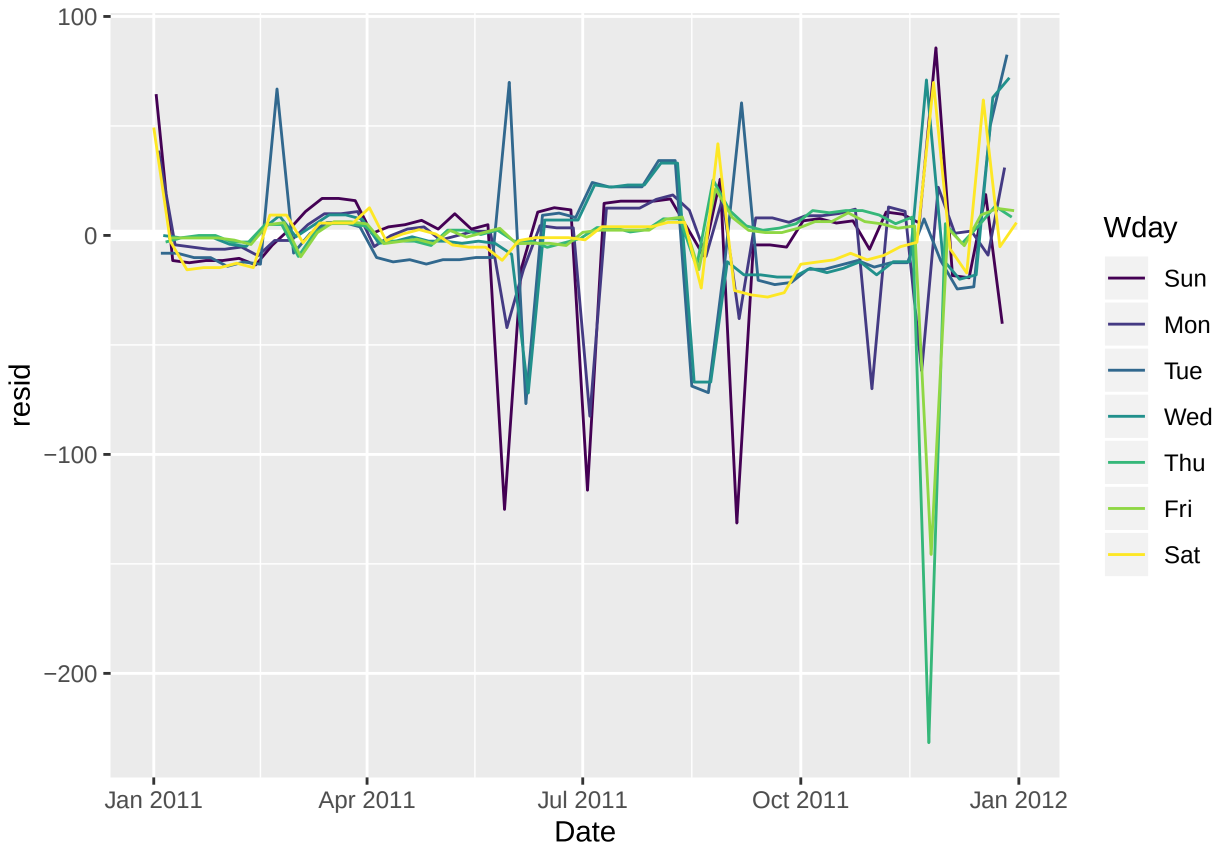
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
daily_res %>%
  filter(Wday == "Sat") %>%
  ggplot(aes(Date, NumberOfFlights, colour = season)) +
  geom_point(alpha = 1/3) +
  geom_line() +
  scale_x_date(NULL, date_breaks = "1 month", date_labels = "%b")
```



```
daily_res %>%
  ggplot(aes(Date, resid, color = Wday)) +
  geom_ref_line(h = 0) +
  geom_line()
```



```
## Most of the dates on the models appears to be holidays(If we look it at American Holiday calenders)
```

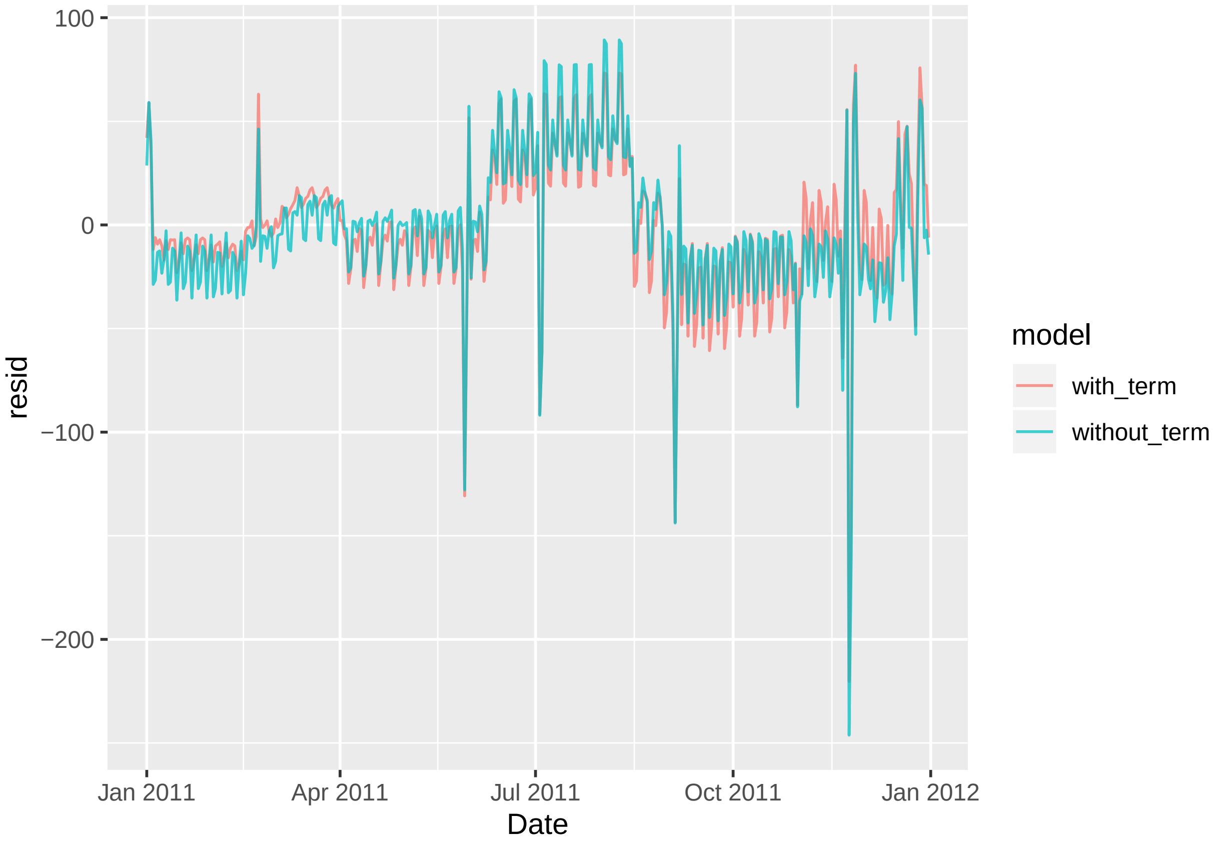
b) (1 pt) Add a variable to identify dates fitting this criterion.

```
data3<-data2 %>%
  mutate(Quarter=cut(Date,
                     breaks = ymd(20110101, 20110401, 20110701, 20111101, 20120101),
                     labels = c("Q1","Q2", "Q3", "Q4")))
tail(data3)
```

```
## # A tibble: 6 x 7
## # Groups:   Date [1]
##   Date       DayOfWeek NumberOfFlights Wday  resid season Quarter
##   <date>         <int>           <int> <ord> <dbl> <fct>  <fct>
## 1 2011-12-06         2             565 Tue   -46.7 fall   Q4
## 2 2011-12-06         2             565 Tue   -46.7 fall   Q4
## 3 2011-12-06         2             565 Tue   -46.7 fall   Q4
## 4 2011-12-06         2             565 Tue   -46.7 fall   Q4
## 5 2011-12-06         2             565 Tue   -46.7 fall   Q4
## 6 2011-12-06         2             565 Tue   -46.7 fall   Q4
```

c) (2 pts) Build a model to explain your data using the variables you now have. Visualize the residuals.

```
mod1 <- lm(NumberOfFlights ~ Wday, data = data3)
mod2 <- lm(NumberOfFlights ~ Wday * Quarter, data = data3)

data3 %>%
  gather_residuals(without_term = mod1, with_term = mod2) %>%
  ggplot(aes(Date, resid, colour = model)) +
  geom_line(alpha = 0.75)
```



5. (1 1/2 pts) Use what you have learned above to predict the number of flights for 2020 per day. Print a graph that overlays the number of flights in 2011 with your number of predicted flights in 2020. How many flights do you predict for each day June 20 - July 10 of 2020?

```
data4<-data3%>%
  data_grid(Wday,Quarter,Date =seq(ymd(20200101),ymd(20201231),by=1) ) %>%
  add_predictions(mod2)
```