p8105_hw3_jj3205

Jia Ji (jj3205)

2022-10-15

Problem 1

```
library(p8105.datasets)
data("instacart")
```

Problem 2

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.0     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)

acce_df =
  read_csv("data/accel_data.csv") %>%
  pivot_longer(
    cols = activity.1:activity.1440,
    names_to = "activity_number",
    values_to = "activity_counts",
    names_prefix = "activity.",
  ) %>%
  mutate(
    is_weekend = (day == "Saturday" | day == "Sunday"),
```

```
    day = factor(day, levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Satu
  )
```

```
## Rows: 35 Columns: 1443
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (1): day
## dbl (1442): week, day_id, activity.1, activity.2, activity.3, activity.4, ac...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
acce_df
```

```
## # A tibble: 50,400 x 6
##     week day_id day    activity_number activity_counts is_weekend
##    <dbl>  <dbl> <fct>  <chr>                     <dbl> <lgl>
##  1     1      1 Friday 1                          88.4 FALSE
##  2     1      1 Friday 2                          82.2 FALSE
##  3     1      1 Friday 3                          64.4 FALSE
##  4     1      1 Friday 4                          70.0 FALSE
##  5     1      1 Friday 5                          75.0 FALSE
##  6     1      1 Friday 6                          66.3 FALSE
##  7     1      1 Friday 7                          53.8 FALSE
##  8     1      1 Friday 8                          47.8 FALSE
##  9     1      1 Friday 9                          55.5 FALSE
## 10     1      1 Friday 10                         43.0 FALSE
## # ... with 50,390 more rows
```

The accel_data_clean dataset contains 50400 observations and 6 variables.
The variables records these information for each observation: week, day_id, which day in a week, is it weekend or not, activity number, and activity counts.

Traditional analyses of accelerometer data focus on the total activity over the day. Using the tidied dataset, we will aggregate across minutes to create a total activity variable for each day, and create a table showing these totals.

```
total_activity =
  acce_df %>%
  group_by(week, day) %>%
  summarise(total_activity = sum(activity_counts)) %>%
  pivot_wider(
    names_from = "day",
    values_from = "total_activity"
  )
knitr::kable(total_activity)
```

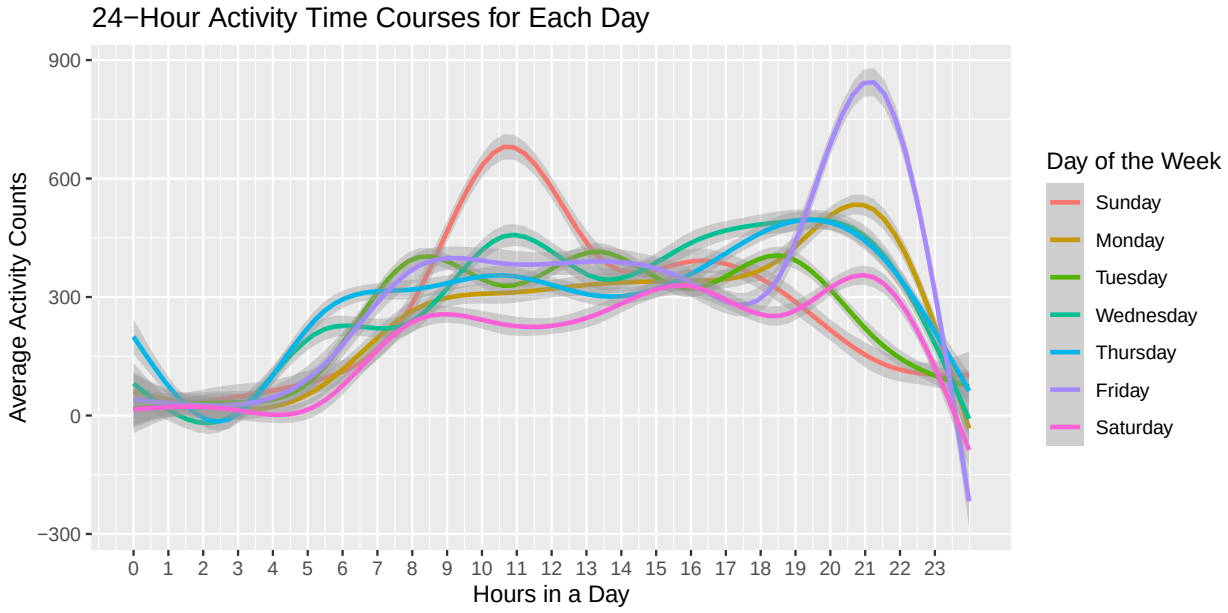
week	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	631105	78828.07	307094.2	340115	355923.6	480542.6	376254
2	422018	295431.00	423245.0	440962	474048.0	568839.0	607175
3	467052	685910.00	381507.0	468869	371230.0	467420.0	382928
4	260617	409450.00	319568.0	434460	340291.0	154049.0	1440
5	138421	389080.00	367824.0	445366	549658.0	620860.0	1440

From the table, we can see that as time passes day by day, the total activity counts were oscillating up and

down. And activity counts on weekends are relatively lower than the counts on weekdays.

Accelerometer data allows the inspection activity over the course of the day. Now we will make a single-panel plot that shows the 24-hour activity time courses for each day and use color to indicate day of the week.

```
acce_df %>%
  mutate(activity_number = as.numeric(activity_number)) %>%
  group_by(day, activity_number) %>%
  summarize(avg_value = mean(activity_counts)) %>%
  ggplot(aes(x = activity_number, y = avg_value, color = day)) +
  geom_smooth() +
  scale_x_continuous(
    breaks = (0:23)*60 + 1,
    labels = c(0:23),
    name = "Hours in a Day"
  ) +
  labs(
    title = "24-Hour Activity Time Courses for Each Day",
    x = "Activity Number (hrs)",
    y = "Average Activity Counts",
    color = "Day of the Week"
  ) +
  theme(legend.position = "right")
```



The average activity counts for all days in a week are the lowest during the time period of around 23:50 p.m. to 6:00 a.m, because at this period the test subject is sleeping and cannot move around frequently.
There are also significant higher peaks of average activity counts at around 10:30 a.m of Sunday and around 21:00 p.m of Friday. This is probably because the test subject was doing some special activity (like doing exercises).

Problem 3

```
library(p8105.datasets)
data("ny_noaa")
```

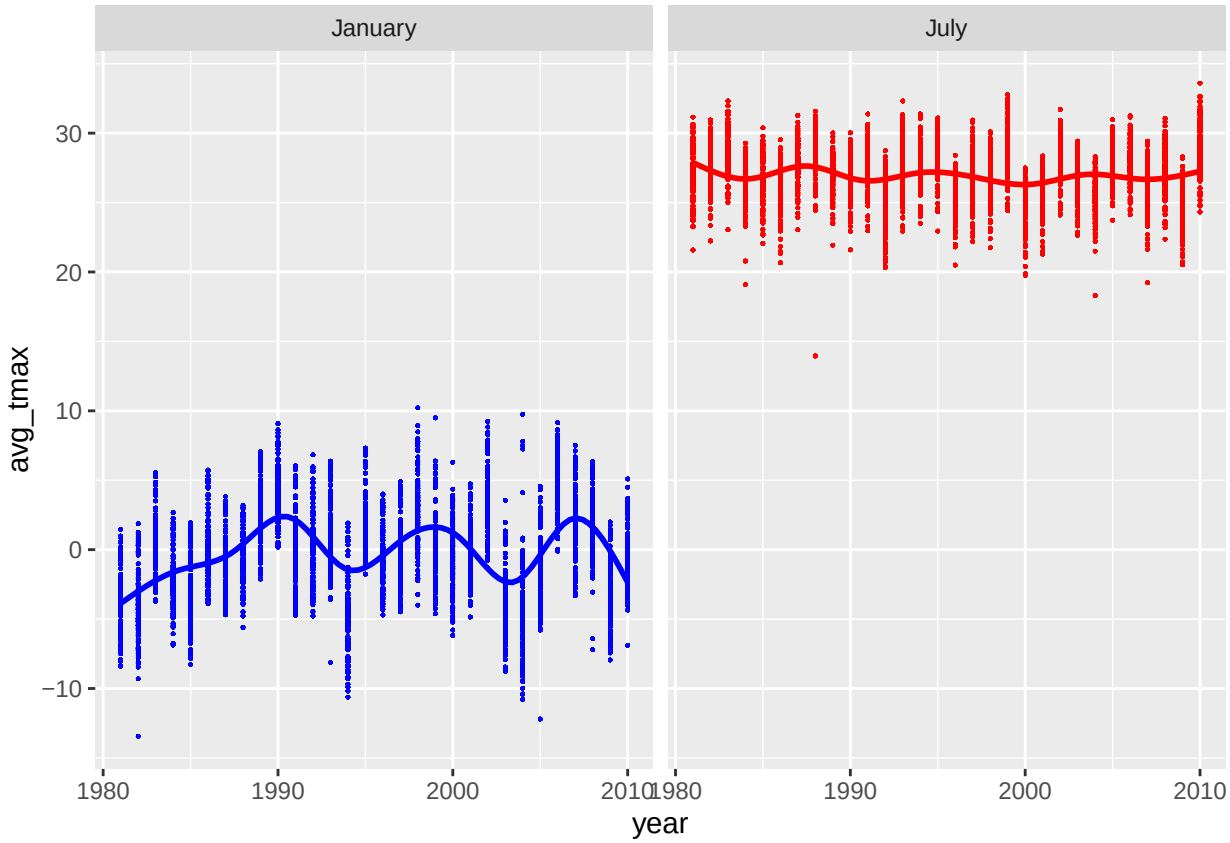
```
snow_df = ny_noaa %>%
  separate(date, c("year", "month", "day"), sep = "-") %>%
  mutate(
    year = as.numeric(year),
    month = as.numeric(month),
    day = as.numeric(day),
    prcp = as.numeric(prcp)*0.1,
    tmax = as.numeric(tmax)*0.1,
    tmin = as.numeric(tmin)*0.1)
```

“

The tidied dataset contains 9 columns and 2595176 rows. Variables are: weather station ID, year, month, day, precipitation (in mm), snowfall (in mm), snow depth (in mm), maximum temperature (in degree C), minimum temperature (in degree C). There are a lot of missing values in this data set: The precipitation variable has 145838 missing values; The snowfall has 381221 missing values; The snow depth variable has 591786 missing values; The minimum temperature has 1134420 missing values; The maximum temperature has 1134358 missing values.

For snowfall, most commonly observed values is 0, there are 2008508 rows with a 0 snowfall record. Plot showing the average max temperature in January and in July in each station across years:

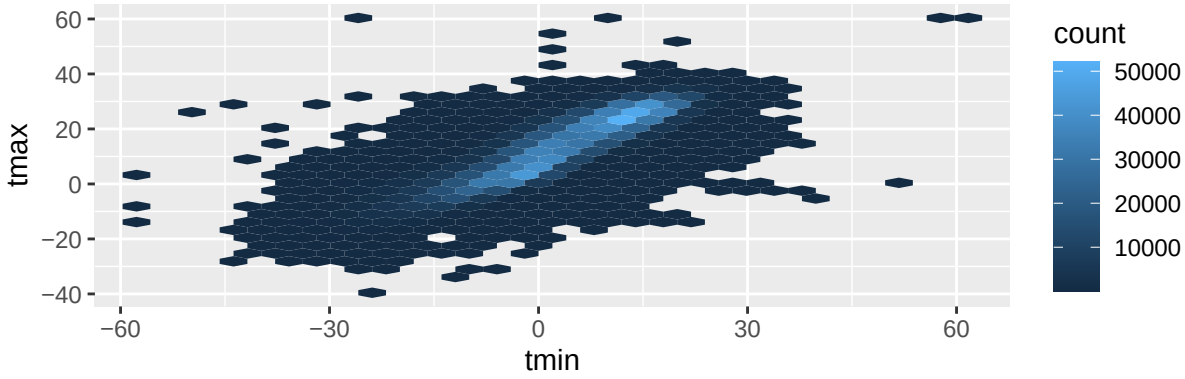
```
snow_df %>%
  filter(month == 1 | month == 7) %>%
  group_by(id, year, month) %>%
  mutate(
    avg_tmax = mean(tmax, na.rm = TRUE),
    month = month.name[month]) %>%
  ggplot(aes(x = year, y = avg_tmax, color = month)) +
  geom_point(size = 0.1)+
  geom_smooth(alpha = 0.5, se = FALSE) +
  facet_grid(~month)+
  theme(legend.position="none")+
  scale_color_manual(values=c("blue", "red"))
```

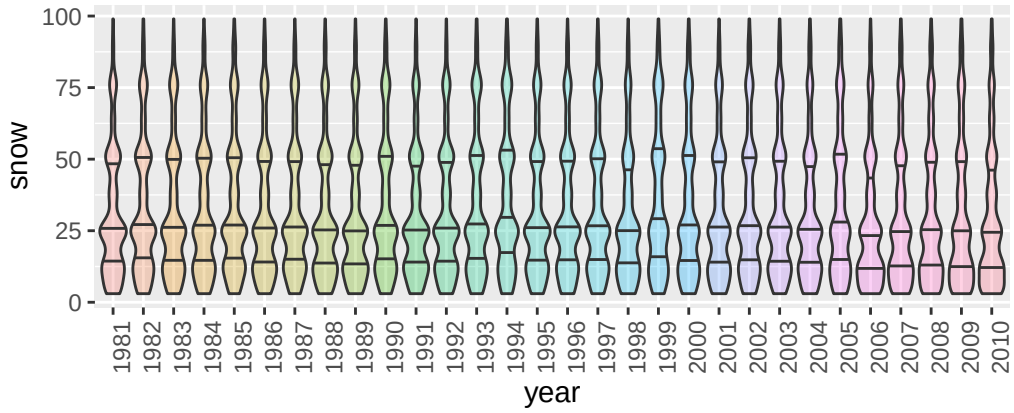


Observable and interpretable structure: It is warmer in July overall. The average maximum temperature among different stations has a smaller fluctuation range among July over the years, as compared to January. It seems that there is not apparent global warming trend from this plot. There seem to be extremely cold winters around 1993~1994 and around 2002-2004.

Make a two-panel plot that consist of maximum temperature vs minimum temperature for the full dataset distribution of snowfall values (between 0 and 100) over the years.

```
library("patchwork")
snow_a <-
  snow_df %>%
  ggplot(aes(x = tmin, y = tmax)) +
  geom_hex()
snow_b <-
snow_df %>%
  filter(snow < 100 & snow > 0) %>%
  mutate(year = factor(year)) %>%
  ggplot(aes(x = year, y = snow)) +
  geom_violin(aes(fill = year), alpha = 0.3, draw_quantiles = c(0.25, 0.5, 0.75))+
  theme(axis.text.x = element_text(angle = 90),
        legend.position="none")
snow_a/snow_b
```


tmax
60
40
20
0
−20
−40
−60
−30
0
30
60
tmin
count
50000
40000
30000
20000
10000


snow
100
75
50
25
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
year