

Week 2 Tasks Solutions

```
library(tidyverse)
library(ggplot2)
library(nycflights13)
library(readr)
library(knitr)
library(fivethirtyeight)
```

Further Task 1

In this task we will work with the dataset analysed and reported in the 2016 article from FiveThirtyEight.com entitled “Some People Are Too Superstitious To Have A Baby On Friday The 13th” [here](#).

The data set is called `US_births_2000_2014` and is in the `fivethirtyeight` package.

1. Create an object called `US_births_2013` which focuses only on data corresponding to 2013 births.

```
#Further Task 1.1 Solution
US_births_2013 <- US_births_2000_2014 |> filter(year == 2013)
```

2. By only choosing births data for the years 2010, 2011, 2012, and 2014 create a new dataframe called `US_births_small` and check that this resulting data frame has 1461 rows. Note that there are many different ways to do this, but try and come up with three different ways using:

- the “or” operator `|`
- the `%in%` operator
- the “not” operator `!` or combinations of them.

```
#Further Task 1.2 Solution
US_births_small <- US_births_2000_2014 |>
  filter(year %in% c(2010, 2011, 2012, 2014))
```

```
US_births_small <- US_births_2000_2014 |>
  filter(!(year %in% c(2000:2009, 2013, 2015)))
```

```
US_births_small <- US_births_2000_2014 |>
  filter(year == 2010 | year == 2011 | year == 2012 | year == 2014)
```

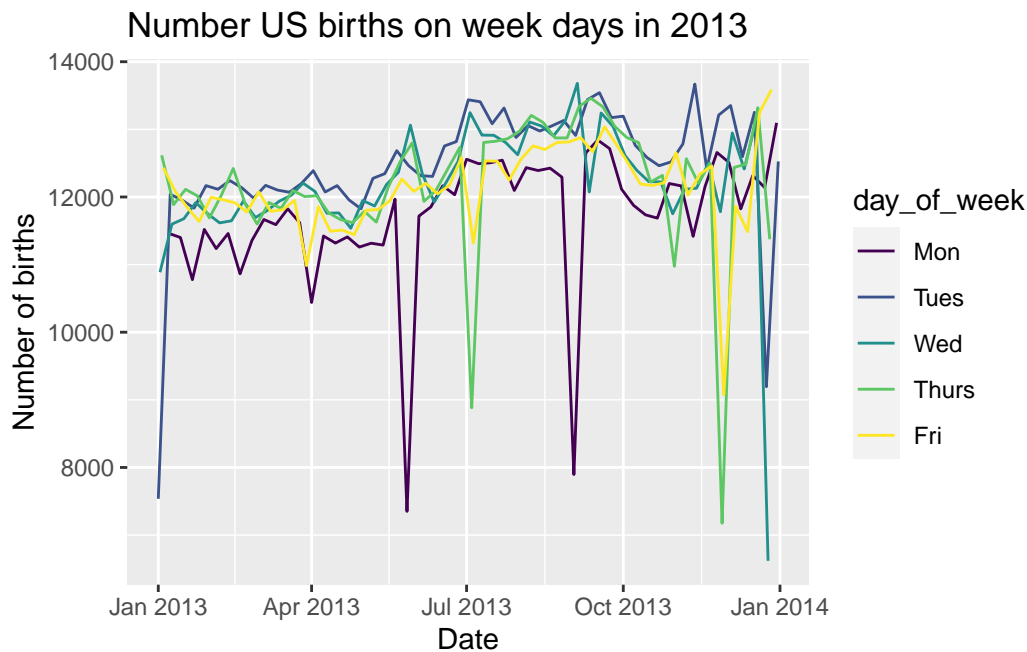
3. Suppose we are interested in choosing rows for only weekdays (not Saturdays or Sundays) for `day_of_week` in year 2013. Write the code to do so and give the name `US_births_weekdays_2013` to the resulting data frame. (Note that you may want to run `US_births_2000_2014 |> distinct(day_of_week)` to identify the specific values of `day_of_week`.)

```
#Further Task 1.3 Solution
```

```
US_births_weekdays_2013 <- US_births_2000_2014 |>
  filter(!(day_of_week %in% c('Sat','Sun')), year==2013)
# You could also use & instead of the last comma, i.e.
US_births_weekdays_2013 <- US_births_2000_2014 |>
  filter(!(day_of_week %in% c('Sat','Sun')) & year==2013)
```

. Using what you covered in **Week 1: Visualization**, produce an appropriate plot looking at the pattern of births on all weekdays in 2013 coloured by the particular day of the week. (Remember to load the package `ggplot2`).

```
#Further Task 1.4 Solution
ggplot(US_births_weekdays_2013, aes(x=date,y=births, color=day_of_week))+
  geom_line()+
  labs(x='Date', y='Number of births',
       title='Number US births on week days in 2013')
```



5. The plot in the previous task has shown there are some outliers in the data for US births on weekdays in 2013. We can use the `summarize` function to get an idea for how these outliers may affect the shape of the `births` variable in `US_births_weekdays_2013`. Write some code to calculate the mean and median values for all weekday birth totals in 2013. Store this aggregated data in the data frame `birth_summ`. What do these values suggest about the effects of the outliers?

```
#Further Task 1.5 Solution
birth_summ <- US_births_weekdays_2013 |>
  summarize(mean_births = mean(births),
```

```

        median_births = median(births))
birth_summ

```

```

# A tibble: 1 x 2
  mean_births median_births
    <dbl>         <int>
1    12145.         12192

```

#Comparing the mean and the median values we see they are not very different,
#showing that the potential outliers are not distorting the shape of

6. Instead of looking at the overall mean and median across all of 2013 weekdays, calculate the mean and median for each of the five different weekdays throughout 2013. Using the same names for the columns as in the `birth_summ` data frame in the previous exercise, create a new data frame called `birth_day_summ`.

```

#Further Task 1.6 Solution
birth_day_summ <- US_births_weekdays_2013 |>
  summarize( mean_births = mean(births),
             median_births = median(births), .by = c(day_of_week))
birth_day_summ

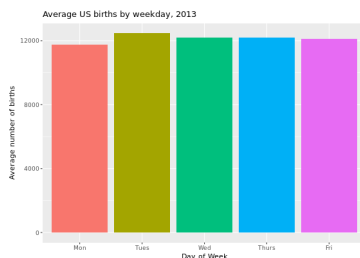
```

```

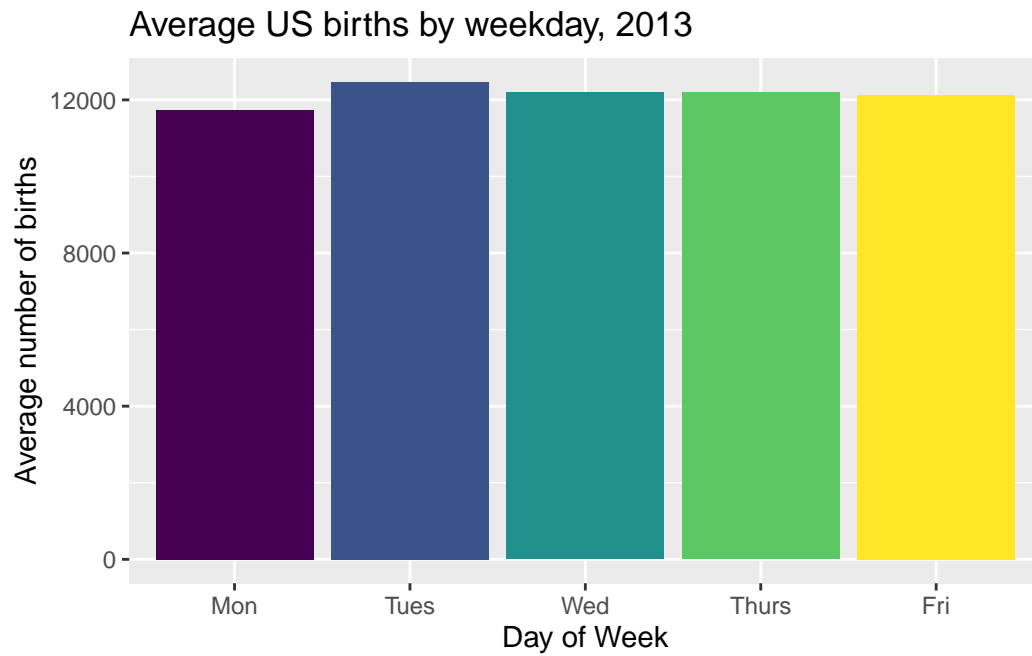
# A tibble: 5 x 3
  day_of_week mean_births median_births
    <ord>         <dbl>         <dbl>
1 Tues          12469.          12519
2 Wed           12190.          12194
3 Thurs         12194.          12270
4 Fri           12133.          12126.
5 Mon           11732.          11838.

```

7. Using the aggregated data in the `birth_day_summ` data frame, produce this barplot.



```
#Further Task 1.7 Solution
ggplot(birth_day_summ, aes(x=day_of_week,y=mean_births, fill=day_of_week))+
  geom_col(show.legend = FALSE)+
  labs(x='Day of Week', y='Average number of births',
       title='Average US births by weekday, 2013')
```



Further Task 2

In this task we will work with the dataset analysed and reported in the 2014 article from FiveThirtyEight.com entitled “41 Percent Of Fliers Think You’re Rude If You Recline Your Seat” [here](#).

The data set is called `flying` and is in the `fivethirtyeight` package.

1. Write code to determine the proportion of respondents in the survey that responded with “Very” when asked if a passenger reclining their seat was rude. You should determine this proportion across the different levels of `age` and `gender` resulting in a data frame of size 8 x 3. Assign the name `prop_very` to this calculated proportion in this aggregated data frame.

Tip

We can obtain proportions using the `mean()` function applied to logical values. For example suppose we want to count the proportion of “heads” in five tosses of a fair coin. If the results of the five tosses are stored in
`tosses <- c("heads", "tails", "tails", "heads", "heads")`
then we can use `mean(tosses == "heads")` to get the resulting answer of 0.6.

Tip

Including the function `na.omit(TRUE)` in the ‘pipe’ (`|>`) removes all entries that are not complete whereas including the argument `na.rm=TRUE` in the `mean()` function removes just those entries where the relevant variable value is missing.

```
#Further Task 2.1 Solution
two_group_prop_1 <- flying |>
  summarize(prop_very = mean(recline_rude=='Very', na.rm=TRUE),
            .by=c(gender,age))
#OR
two_group_prop_2 <- flying |>
  na.omit(TRUE) |>
  summarize(prop_very = mean(recline_rude=='Very'),
            .by = c(gender,age))
```

```
#Compare the two summaries...
two_group_prop_1
```

```
# A tibble: 9 x 3
  gender age    prop_very
  <chr> <ord>    <dbl>
1 <NA> <NA>    0.0909
2 Male 30-44    0.0811
3 Male 18-29    0.152
4 Male 45-60    0.0804
5 Male > 60    0.0606
6 Female > 60    0.129
7 Female 30-44    0.0541
8 Female 45-60    0.0410
9 Female 18-29    0.0860
```

```
two_group_prop_2
```

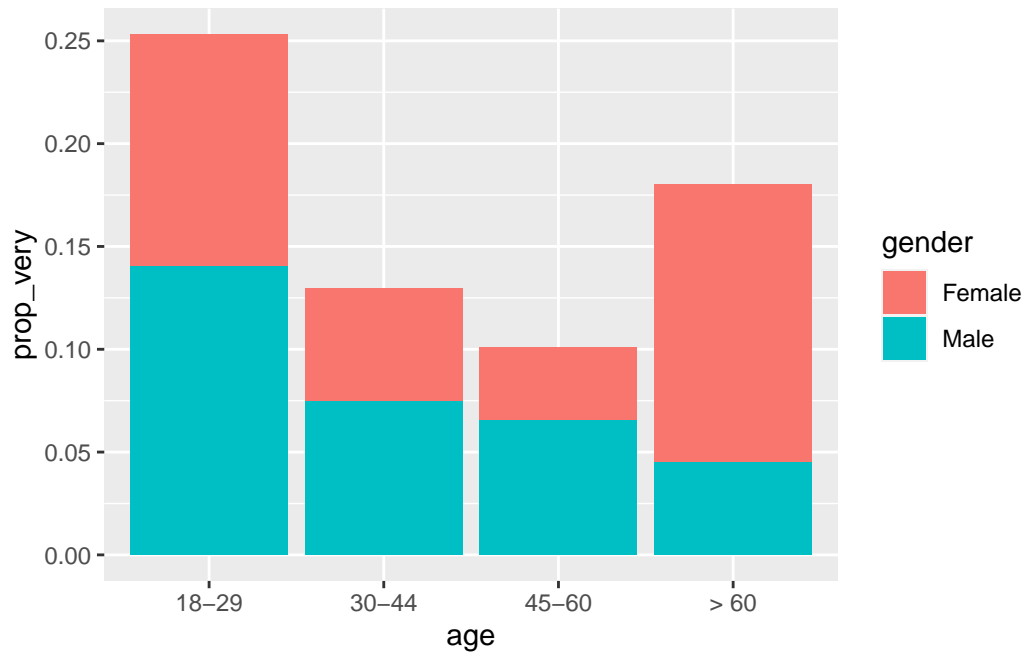
```
# A tibble: 8 x 3
  gender age    prop_very
  <chr> <ord>    <dbl>
1 Male 30-44    0.075
2 Male 18-29    0.140
3 Male 45-60    0.0658
4 Male > 60    0.0455
5 Female > 60    0.135
6 Female 30-44    0.0548
7 Female 45-60    0.0353
8 Female 18-29    0.113
```

2. Using the aggregated data you've created, produce two bar plots (one stacked, the other side-by-side) to show the differences between the sexes of the proportion of people who believe reclining your seat is 'very' rude, within each age group.

What gender and age-range pairings have the highest and lowest proportions of thinking reclining airline seats are very rude in this survey?

What stands out to you as you review these proportions?

```
#Further Task 2.2 Solution
ggplot(two_group_prop_2, aes(x=age,y=prop_very,fill=gender))+geom_col()
```



```
ggplot(two_group_prop_2, aes(x=age,y=prop_very,fill=gender))+geom_col(position = "dodge")
```

