CourseProject1

Loading necessary packages

```
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(lubridate)
```

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

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

```
library(ggplot2)
```

Loading and preprocessing the data

Show any code that is needed to

1. Load the data (i.e. `read.csv()`)
2. Process/transform the data (if necessary) into a format suitable for your analysis

Loading the data

read the data using read.table()

```
data=read.table("activity.csv", sep=',', na.strings = NA, header=T)
```

Converting data format

change the date into dateformat

```
data$date = ymd(data$date)
```

check the data

```
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```
str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
summary(data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day
2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
3. Calculate and report the mean and median of the total number of steps taken per day

Calculate the total number of Steps taken per day

manipulate data using dplyr and group by date

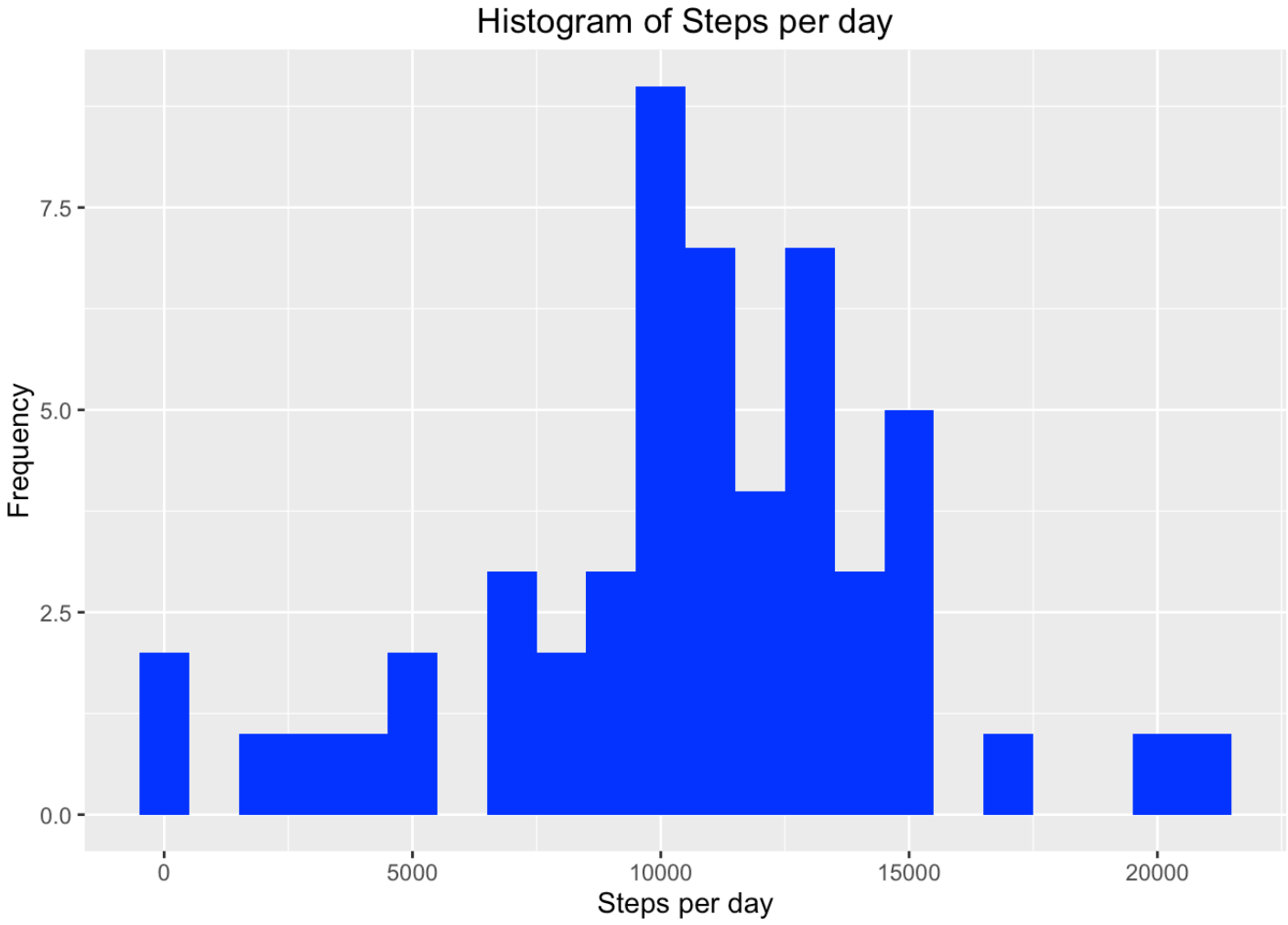
```
steps = data %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps=sum(steps)) %>%
  print
```

```
## Source: local data frame [53 x 2]
##
##          date steps
##        <date> <int>
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## ..        ...   ...
```

Make a histogram of the total number of steps taken each day

make the histogram using ggplot

```
ggplot(steps, aes(x=steps))+
  geom_histogram(fill="blue", binwidth=1000)+
  labs(title="Histogram of Steps per day", x = "Steps per day", y = "Frequency")
```



Calculate the mean and median of the total number of steps taken per day

```
mean_stp = mean(steps$steps, na.rm=T)
median_stp = median(steps$steps, na.rm=T)
mean_stp
```

```
## [1] 10766.19
```

```
median_stp
```

```
## [1] 10765
```

What is the average daily activity pattern?

1.Make a time series plot (i.e. `type` = “`l`”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis) 2.Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

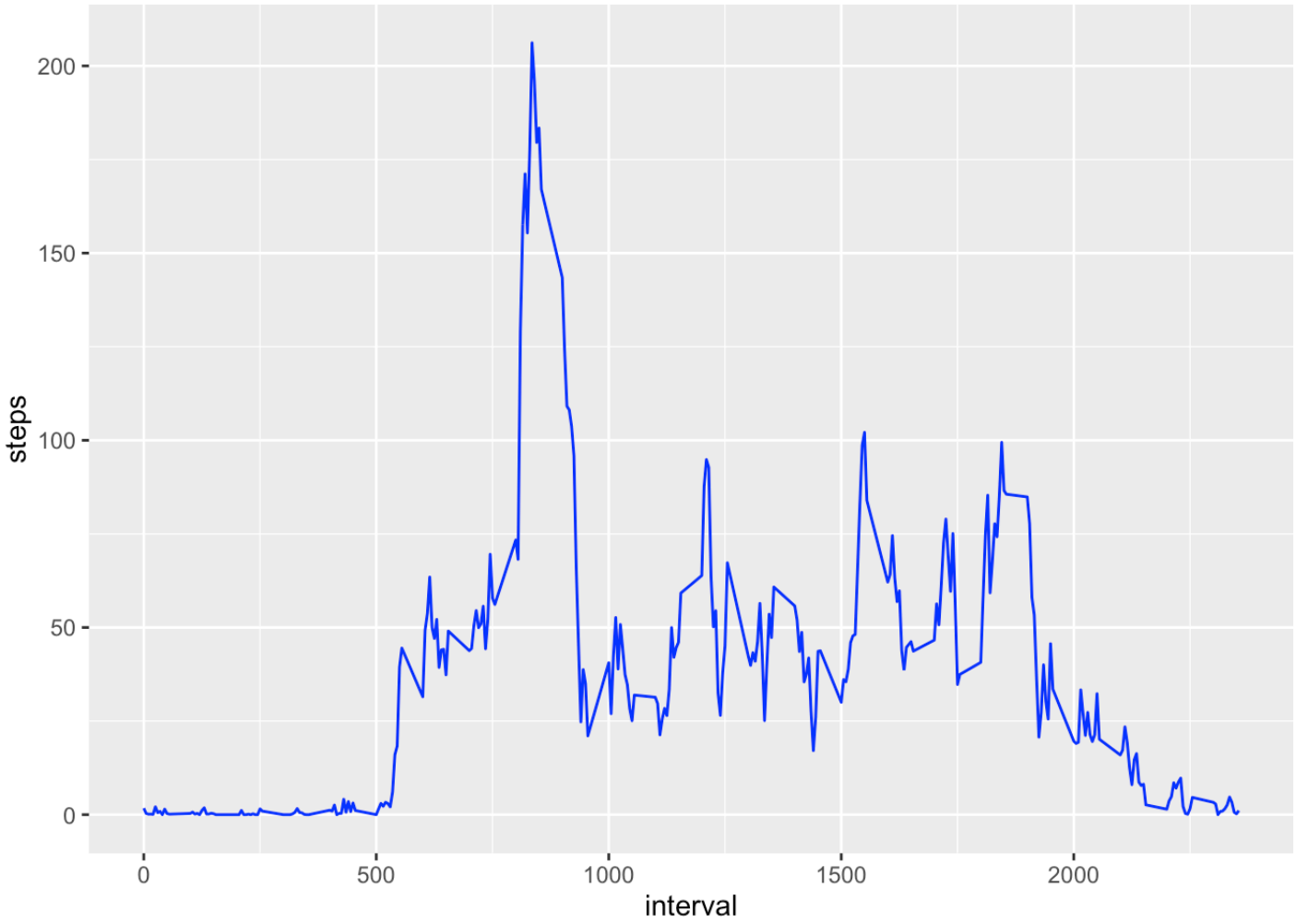
Make a time series plot

calculate the average number of steps taken in each 5-minut interval per day using dplyr and group by interval

```
interval = data %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(steps = mean(steps))
head(interval)
```

```
## Source: local data frame [6 x 2]
##
##   interval     steps
##      <int>     <dbl>
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
ggplot(interval, aes(x=interval,y=steps))+
  geom_line(color = "blue")
```



Find out the maximum steps

use which.max() to find out the maximum steps, on average, across all the days

```
interval[which.max(interval$steps),]
```

```
## Source: local data frame [1 x 2]
##
##   interval    steps
##      <int>    <dbl>
## 1      835 206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as `NA`). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with `NA`s)
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

1. calculate and report the total number of missing values

```
sum(is.na(data$steps))
```

```
## [1] 2304
```

2. fill in all of the missing values in the dataset with the mean in the same 5-minute inerval

```
data_full <- data
nas <- is.na(data_full$steps)
avg_interval <- tapply(data_full$steps, data_full$interval, mean, na.rm=TRUE, simplif
y=TRUE)
```

3. create new dataset as the original with filling in the missing values

```
data_full$steps[nas] <- avg_interval[as.character(data_full$interval[nas])]
sum(is.na(data_full$steps))
```

```
## [1] 0
```

4. calculate the mean and median total number of steps taken per day and make a histogram of the total number of steps taken each day

```
steps_full <- data_full %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
  print
```

```
## Source: local data frame [61 x 2]
##
##          date    steps
##        <date>    <dbl>
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## ..        ...      ...
```

```
ggplot(steps_full, aes(x = steps)) +
  geom_histogram(fill = "blue", binwidth = 1000) +
  labs(title = "Histogram of Steps per day, including missing values", x = "Steps per
day", y = "Frequency")
```

Are there differences in activity patterns between weekdays and weekends?

For this part the `weekdays()` function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.
2. Make a panel plot containing a time series plot (i.e. `type` = "`l`") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

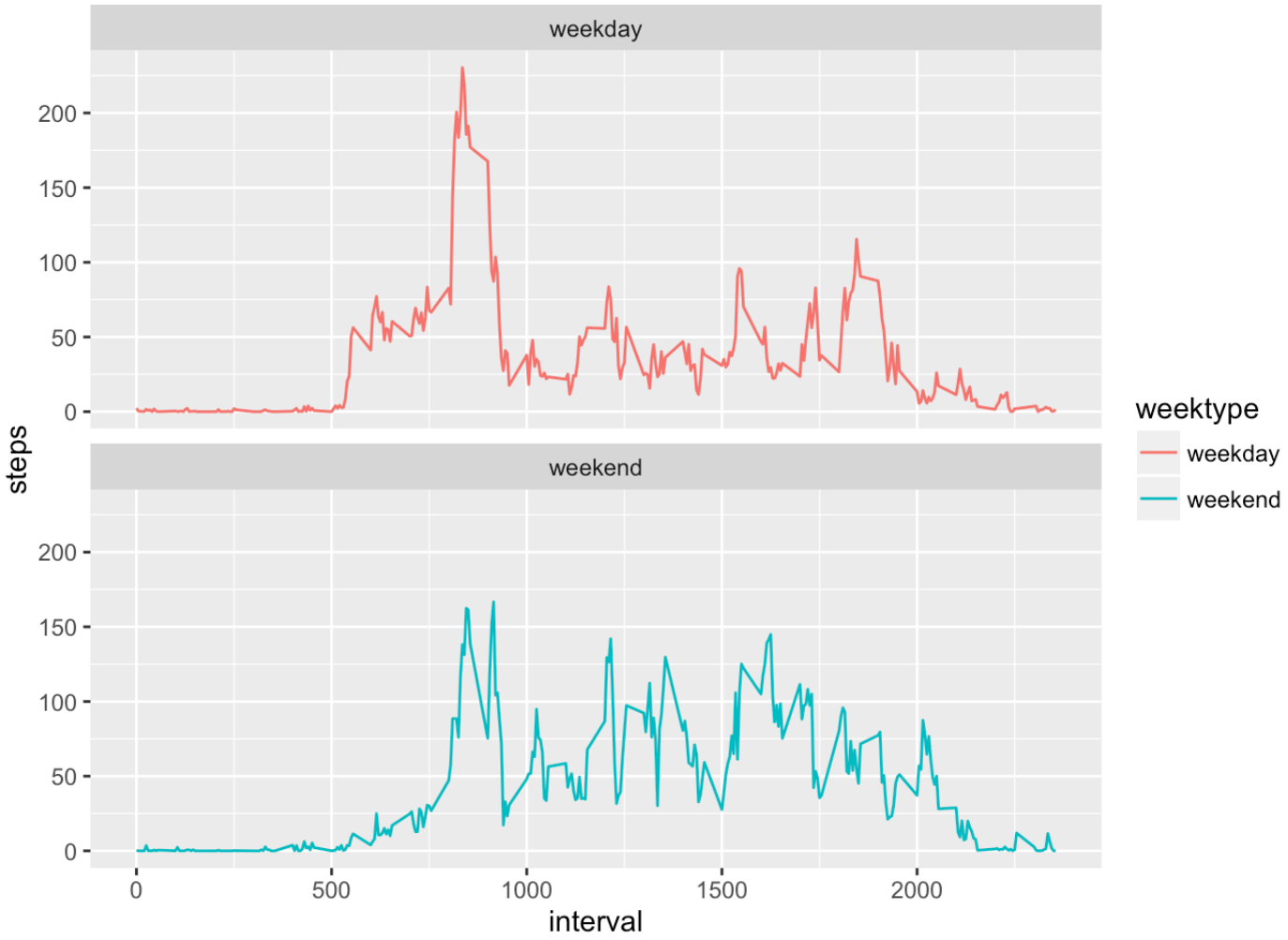
1. Create a new factor variable

```
data_full$date = ymd(data_full$date)
data_full <- mutate(data_full, weektype = ifelse(weekdays(data_full$date) == "Saturda
y" | weekdays(data_full$date) == "Sunday", "weekend", "weekday"))
data_full$weektype <- as.factor(data_full$weektype)
head(data_full)
```

```
##       steps       date interval weektype
## 1 1.7169811 2012-10-01        0  weekday
## 2 0.3396226 2012-10-01        5  weekday
## 3 0.1320755 2012-10-01       10  weekday
## 4 0.1509434 2012-10-01       15  weekday
## 5 0.0754717 2012-10-01       20  weekday
## 6 2.0943396 2012-10-01       25  weekday
```

Average number of steps per 5-minute interval is computed again

```
interval_full = data_full %>%
  group_by(interval,weektype) %>%
  summarise(steps= mean(steps))
s = ggplot(interval_full, aes(x=interval, y=steps, color=weektype))+
  geom_line()+
  facet_wrap(~weektype, ncol = 1, nrow =2)
print(s)
```


weekday
weekend
steps
interval
0
50
100
150
200
0
500
1000
1500
2000
weektype
weekday
weekend