Reproducible Research: Peer Assessment 1

ANCO HAN

2017 06 25

Setting up the Environment

- Working Directory

```
rm(list=ls())
rootDir = '~/DS'
list.files(paste0(rootDir, '/Data'))
```

```
## [1] "activity.csv"
```

- Load Packages

```
library(ggplot2)
library(sqldf)
```

Loading and preprocessing the data

Show any code that is needed to

1. Load the data (i.e. read.csv())

```
myData = read.csv(paste0(rootDir, '/Data/activity.csv'))
summary(myData)
```

```
##      steps                date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##  NA's   :2304     (Other)   :15840
```

```
head(myData); tail(myData)
```

```
##   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
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

2. Process/transform the data (if necessary) into a format suitable for your analysis

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

```
totalCnt = data.frame(value=tapply(myData$steps, myData$date, sum, na.rm=TRUE))
sum(is.na(totalCnt)) + sum(totalCnt<1000, na.rm=FALSE)
```

```
## [1] 10
```

```
sum(is.na(totalCnt)) + sum(totalCnt<1000, na.rm=TRUE)
```

```
## [1] 10
```

```
totalCnt[totalCnt<1000,]
```

```
## 2012-10-01 2012-10-02 2012-10-08 2012-11-01 2012-11-04 2012-11-09
##          0        126          0          0          0          0
## 2012-11-10 2012-11-14 2012-11-15 2012-11-30
##          0          0         41          0
```

```
totalCnt = data.frame(value=tapply(myData$steps, myData$date, sum, na.rm=FALSE))
sum(is.na(totalCnt)) + sum(totalCnt<1000, na.rm=FALSE)
```

```
## [1] NA
```

```
sum(is.na(totalCnt)) + sum(totalCnt<1000, na.rm=TRUE)
```

```
## [1] 10
```

```
totalCnt[totalCnt<1000,]
```

```
##       <NA> 2012-10-02       <NA>       <NA>       <NA>       <NA>
##         NA        126         NA         NA         NA         NA
##       <NA>       <NA> 2012-11-15       <NA>
##         NA         NA         41         NA
```

- The Answer is:

```
totalCnt
```

```
##            value
## 2012-10-01    NA
## 2012-10-02   126
## 2012-10-03 11352
## 2012-10-04 12116
## 2012-10-05 13294
## 2012-10-06 15420
## 2012-10-07 11015
## 2012-10-08    NA
## 2012-10-09 12811
## 2012-10-10  9900
## 2012-10-11 10304
## 2012-10-12 17382
## 2012-10-13 12426
## 2012-10-14 15098
## 2012-10-15 10139
```

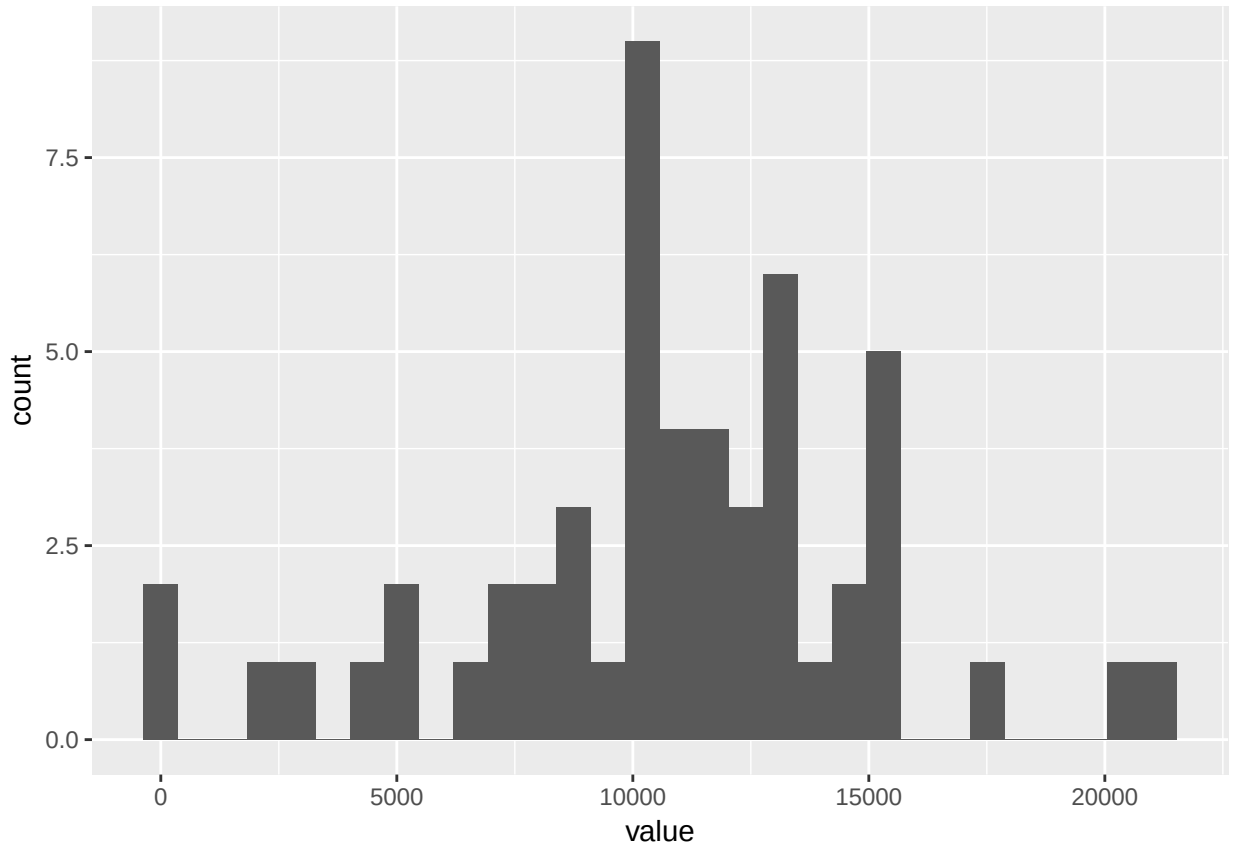
```
## 2012-10-16 15084
## 2012-10-17 13452
## 2012-10-18 10056
## 2012-10-19 11829
## 2012-10-20 10395
## 2012-10-21  8821
## 2012-10-22 13460
## 2012-10-23  8918
## 2012-10-24  8355
## 2012-10-25  2492
## 2012-10-26  6778
## 2012-10-27 10119
## 2012-10-28 11458
## 2012-10-29  5018
## 2012-10-30  9819
## 2012-10-31 15414
## 2012-11-01    NA
## 2012-11-02 10600
## 2012-11-03 10571
## 2012-11-04    NA
## 2012-11-05 10439
## 2012-11-06  8334
## 2012-11-07 12883
## 2012-11-08  3219
## 2012-11-09    NA
## 2012-11-10    NA
## 2012-11-11 12608
## 2012-11-12 10765
## 2012-11-13  7336
## 2012-11-14    NA
## 2012-11-15    41
## 2012-11-16  5441
## 2012-11-17 14339
## 2012-11-18 15110
## 2012-11-19  8841
## 2012-11-20  4472
## 2012-11-21 12787
## 2012-11-22 20427
## 2012-11-23 21194
## 2012-11-24 14478
## 2012-11-25 11834
## 2012-11-26 11162
## 2012-11-27 13646
## 2012-11-28 10183
## 2012-11-29  7047
## 2012-11-30    NA
```

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

```
gg = ggplot(data=totalCnt, aes(x=value))
gg + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```



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

```
meanCnt = tapply(myData$steps, myData$date, mean)
medianCnt = tapply(myData$steps, myData$date, median)
```

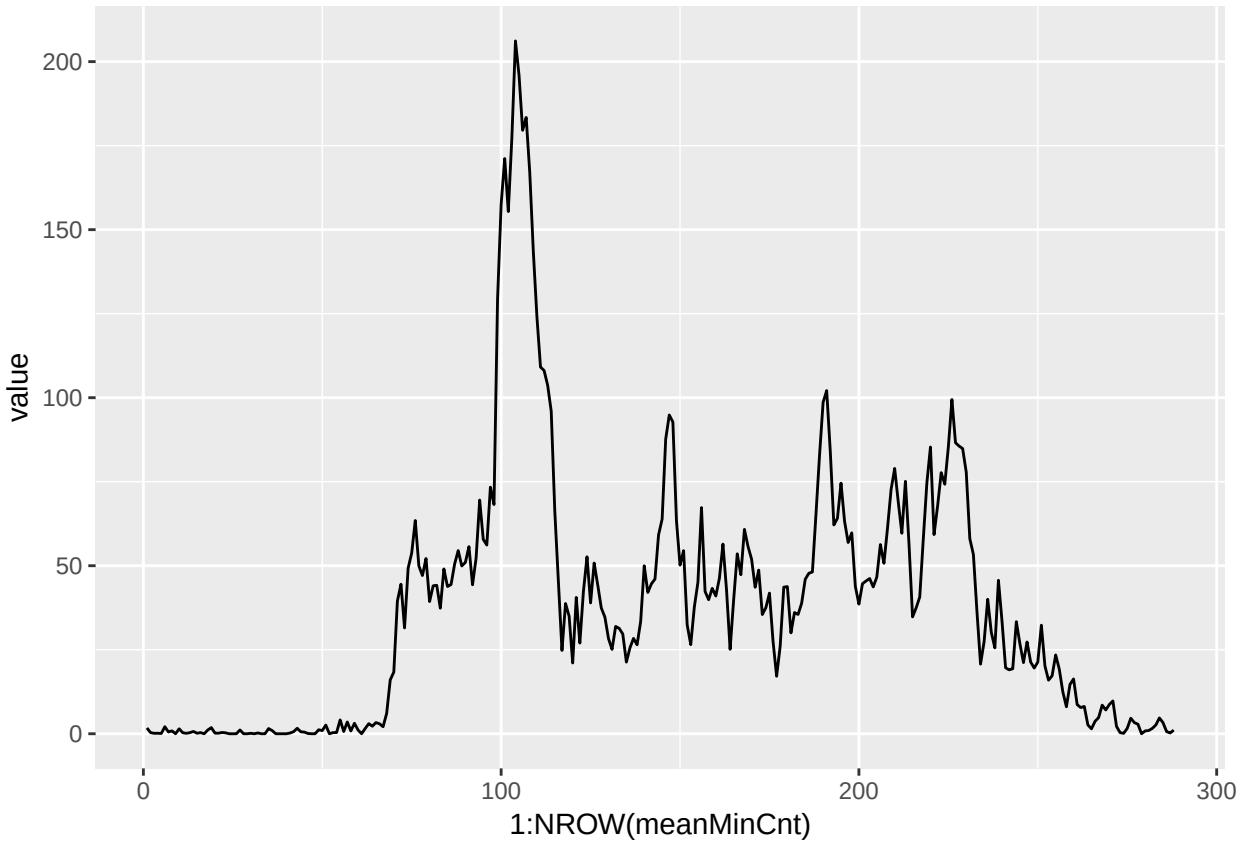
What is the average daily activity pattern?

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)

```
meanMinCnt = data.frame(value=tapply(myData$steps, myData$interval, mean, na.rm=TRUE))
meanMinCnt$idx = rownames(meanMinCnt)
rownames(meanMinCnt) = NULL
head(meanMinCnt)
```

```
##       value idx
## 1 1.7169811   0
## 2 0.3396226   5
## 3 0.1320755  10
## 4 0.1509434  15
## 5 0.0754717  20
## 6 2.0943396  25
```

```
gg2 = ggplot(data=meanMinCnt)
gg2 + geom_line(aes(x=1:NROW(meanMinCnt),y=value))
```



2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
meanMinCnt[meanMinCnt$value == max(meanMinCnt$value),]
```

```
##        value idx
## 104 206.1698 835
```

```
sqldf("
    SELECT *
    FROM ```meanMinCnt```
    WHERE value = (SELECT max(value) FROM ```meanMinCnt```)
    ")
```

```
## Loading required package: tcltk
```

```
##      value idx
## 1 206.1698 835
```

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 NAs)

- The Answer is:

```
(CountNA = sum(is.na(myData$steps)))
```

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

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.

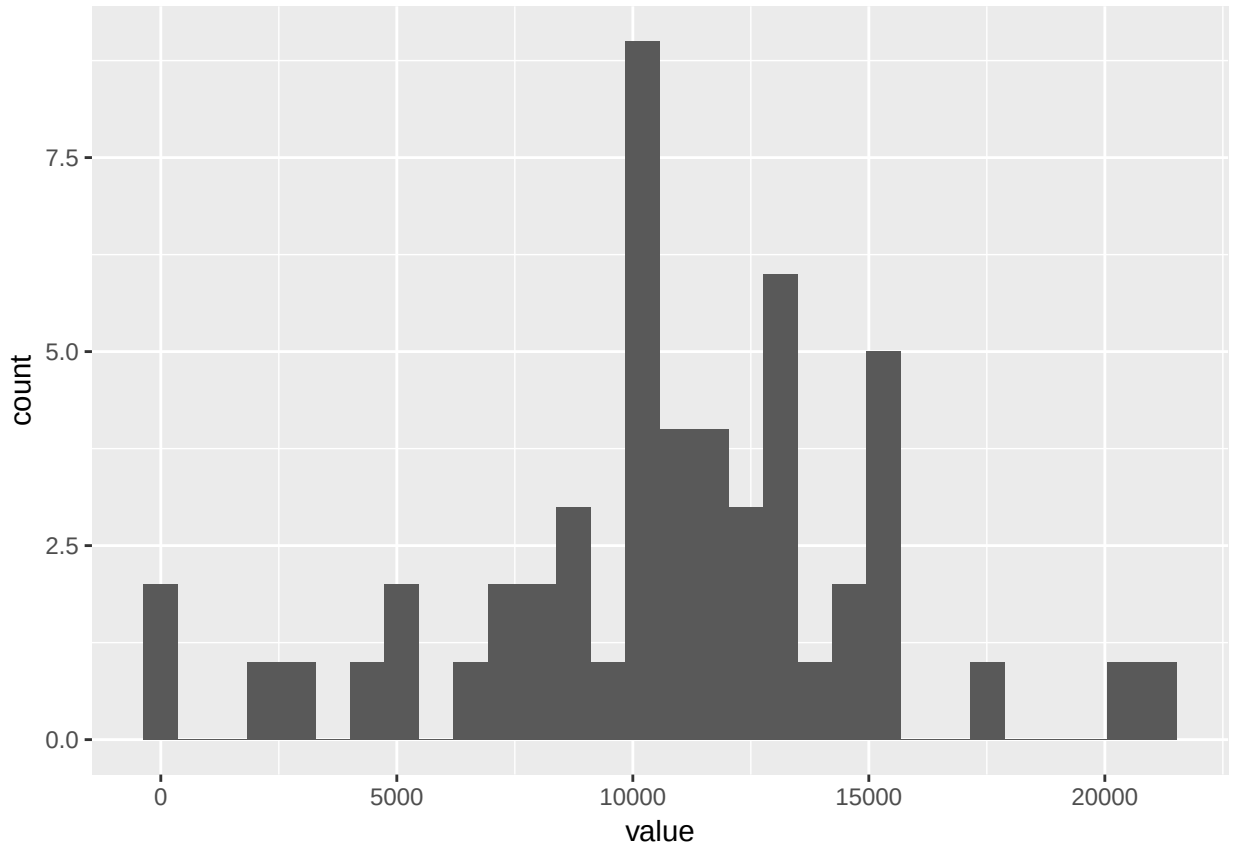
```
# myData[is.na(myData$steps),]
# meanMinCnt

mydf1 =
sqldf("
        SELECT A.date, A.interval, A.steps, round(B.value) AS MeanOfInterval
        FROM ```myData``` A Left Join ```meanMinCnt``` B on A.interval = B.idx;
      ")
mydf1$stepsNew = ifelse(is.na(mydf1$steps), mydf1$MeanOfInterval, mydf1$steps)
head(mydf1)
```

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

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?

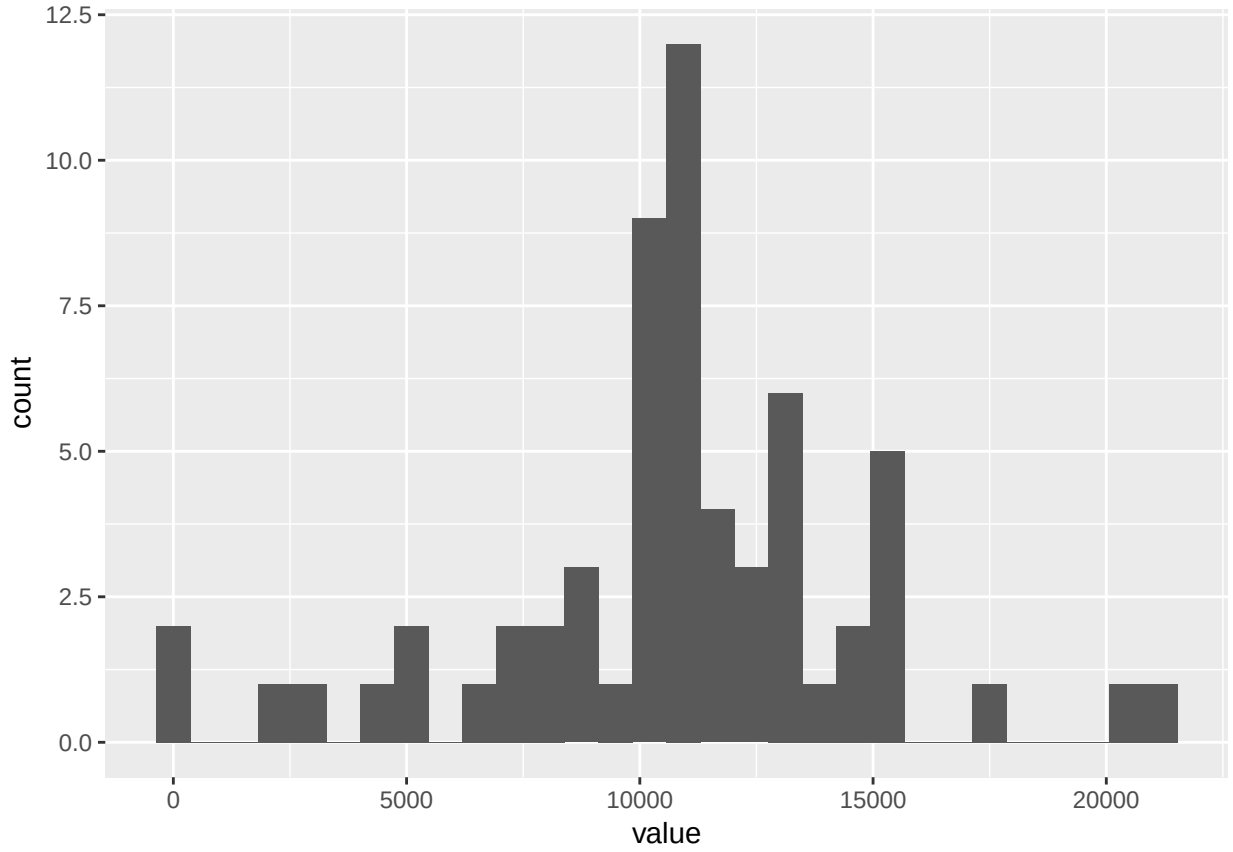
```
gg = ggplot(data=totalCnt, aes(x=value))
gg + geom_histogram()
```




```
head(mydf1)
```

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

```
totalCntNew = data.frame(value=tapply(mydf1$stepsNew, myData$date, sum, na.rm=FALSE))
gg3 = ggplot(data=totalCntNew, aes(x=value))
gg3 + geom_histogram()
```




```
(meanOfDate = data.frame(meanOfDate=tapply(mydf1$stepsNew, mydf1$date, mean)))
```

```
##            meanOfDate
## 2012-10-01 37.3680556
## 2012-10-02  0.4375000
## 2012-10-03 39.4166667
## 2012-10-04 42.0694444
## 2012-10-05 46.1597222
## 2012-10-06 53.5416667
## 2012-10-07 38.2465278
## 2012-10-08 37.3680556
## 2012-10-09 44.4826389
## 2012-10-10 34.3750000
## 2012-10-11 35.7777778
## 2012-10-12 60.3541667
## 2012-10-13 43.1458333
## 2012-10-14 52.4236111
## 2012-10-15 35.2048611
## 2012-10-16 52.3750000
## 2012-10-17 46.7083333
## 2012-10-18 34.9166667
## 2012-10-19 41.0729167
## 2012-10-20 36.0937500
## 2012-10-21 30.6284722
## 2012-10-22 46.7361111
## 2012-10-23 30.9652778
## 2012-10-24 29.0104167
```

```
## 2012-10-25  8.6527778
## 2012-10-26 23.5347222
## 2012-10-27 35.1354167
## 2012-10-28 39.7847222
## 2012-10-29 17.4236111
## 2012-10-30 34.0937500
## 2012-10-31 53.5208333
## 2012-11-01 37.3680556
## 2012-11-02 36.8055556
## 2012-11-03 36.7048611
## 2012-11-04 37.3680556
## 2012-11-05 36.2465278
## 2012-11-06 28.9375000
## 2012-11-07 44.7326389
## 2012-11-08 11.1770833
## 2012-11-09 37.3680556
## 2012-11-10 37.3680556
## 2012-11-11 43.7777778
## 2012-11-12 37.3784722
## 2012-11-13 25.4722222
## 2012-11-14 37.3680556
## 2012-11-15  0.1423611
## 2012-11-16 18.8923611
## 2012-11-17 49.7881944
## 2012-11-18 52.4652778
## 2012-11-19 30.6979167
## 2012-11-20 15.5277778
## 2012-11-21 44.3993056
## 2012-11-22 70.9270833
## 2012-11-23 73.5902778
## 2012-11-24 50.2708333
## 2012-11-25 41.0902778
## 2012-11-26 38.7569444
## 2012-11-27 47.3819444
## 2012-11-28 35.3576389
## 2012-11-29 24.4687500
## 2012-11-30 37.3680556
```

```
(medianOfDate = data.frame(medianOfDate=tapply(mydf1$stepsNew, mydf1$date, median)))
```

```
##            medianOfDate
## 2012-10-01         34.5
## 2012-10-02          0.0
## 2012-10-03          0.0
## 2012-10-04          0.0
## 2012-10-05          0.0
## 2012-10-06          0.0
## 2012-10-07          0.0
## 2012-10-08         34.5
## 2012-10-09          0.0
## 2012-10-10          0.0
## 2012-10-11          0.0
## 2012-10-12          0.0
## 2012-10-13          0.0
## 2012-10-14          0.0
```

```
## 2012-10-15          0.0
## 2012-10-16          0.0
## 2012-10-17          0.0
## 2012-10-18          0.0
## 2012-10-19          0.0
## 2012-10-20          0.0
## 2012-10-21          0.0
## 2012-10-22          0.0
## 2012-10-23          0.0
## 2012-10-24          0.0
## 2012-10-25          0.0
## 2012-10-26          0.0
## 2012-10-27          0.0
## 2012-10-28          0.0
## 2012-10-29          0.0
## 2012-10-30          0.0
## 2012-10-31          0.0
## 2012-11-01         34.5
## 2012-11-02          0.0
## 2012-11-03          0.0
## 2012-11-04         34.5
## 2012-11-05          0.0
## 2012-11-06          0.0
## 2012-11-07          0.0
## 2012-11-08          0.0
## 2012-11-09         34.5
## 2012-11-10         34.5
## 2012-11-11          0.0
## 2012-11-12          0.0
## 2012-11-13          0.0
## 2012-11-14         34.5
## 2012-11-15          0.0
## 2012-11-16          0.0
## 2012-11-17          0.0
## 2012-11-18          0.0
## 2012-11-19          0.0
## 2012-11-20          0.0
## 2012-11-21          0.0
## 2012-11-22          0.0
## 2012-11-23          0.0
## 2012-11-24          0.0
## 2012-11-25          0.0
## 2012-11-26          0.0
## 2012-11-27          0.0
## 2012-11-28          0.0
## 2012-11-29          0.0
## 2012-11-30         34.5
```

```
cbind(
meanCnt
,meanOfDate
,medianCnt
,medianOfDate
)
```

```
##               meanCnt meanOfDate medianCnt medianOfDate
## 2012-10-01         NA 37.3680556        NA         34.5
## 2012-10-02  0.4375000  0.4375000         0          0.0
## 2012-10-03 39.4166667 39.4166667         0          0.0
## 2012-10-04 42.0694444 42.0694444         0          0.0
## 2012-10-05 46.1597222 46.1597222         0          0.0
## 2012-10-06 53.5416667 53.5416667         0          0.0
## 2012-10-07 38.2465278 38.2465278         0          0.0
## 2012-10-08         NA 37.3680556        NA         34.5
## 2012-10-09 44.4826389 44.4826389         0          0.0
## 2012-10-10 34.3750000 34.3750000         0          0.0
## 2012-10-11 35.7777778 35.7777778         0          0.0
## 2012-10-12 60.3541667 60.3541667         0          0.0
## 2012-10-13 43.1458333 43.1458333         0          0.0
## 2012-10-14 52.4236111 52.4236111         0          0.0
## 2012-10-15 35.2048611 35.2048611         0          0.0
## 2012-10-16 52.3750000 52.3750000         0          0.0
## 2012-10-17 46.7083333 46.7083333         0          0.0
## 2012-10-18 34.9166667 34.9166667         0          0.0
## 2012-10-19 41.0729167 41.0729167         0          0.0
## 2012-10-20 36.0937500 36.0937500         0          0.0
## 2012-10-21 30.6284722 30.6284722         0          0.0
## 2012-10-22 46.7361111 46.7361111         0          0.0
## 2012-10-23 30.9652778 30.9652778         0          0.0
## 2012-10-24 29.0104167 29.0104167         0          0.0
## 2012-10-25  8.6527778  8.6527778         0          0.0
## 2012-10-26 23.5347222 23.5347222         0          0.0
## 2012-10-27 35.1354167 35.1354167         0          0.0
## 2012-10-28 39.7847222 39.7847222         0          0.0
## 2012-10-29 17.4236111 17.4236111         0          0.0
## 2012-10-30 34.0937500 34.0937500         0          0.0
## 2012-10-31 53.5208333 53.5208333         0          0.0
## 2012-11-01         NA 37.3680556        NA         34.5
## 2012-11-02 36.8055556 36.8055556         0          0.0
## 2012-11-03 36.7048611 36.7048611         0          0.0
## 2012-11-04         NA 37.3680556        NA         34.5
## 2012-11-05 36.2465278 36.2465278         0          0.0
## 2012-11-06 28.9375000 28.9375000         0          0.0
## 2012-11-07 44.7326389 44.7326389         0          0.0
## 2012-11-08 11.1770833 11.1770833         0          0.0
## 2012-11-09         NA 37.3680556        NA         34.5
## 2012-11-10         NA 37.3680556        NA         34.5
## 2012-11-11 43.7777778 43.7777778         0          0.0
## 2012-11-12 37.3784722 37.3784722         0          0.0
## 2012-11-13 25.4722222 25.4722222         0          0.0
## 2012-11-14         NA 37.3680556        NA         34.5
## 2012-11-15  0.1423611  0.1423611         0          0.0
## 2012-11-16 18.8923611 18.8923611         0          0.0
## 2012-11-17 49.7881944 49.7881944         0          0.0
## 2012-11-18 52.4652778 52.4652778         0          0.0
## 2012-11-19 30.6979167 30.6979167         0          0.0
## 2012-11-20 15.5277778 15.5277778         0          0.0
## 2012-11-21 44.3993056 44.3993056         0          0.0
## 2012-11-22 70.9270833 70.9270833         0          0.0
```

```
## 2012-11-23 73.5902778 73.5902778         0          0.0
## 2012-11-24 50.2708333 50.2708333         0          0.0
## 2012-11-25 41.0902778 41.0902778         0          0.0
## 2012-11-26 38.7569444 38.7569444         0          0.0
## 2012-11-27 47.3819444 47.3819444         0          0.0
## 2012-11-28 35.3576389 35.3576389         0          0.0
## 2012-11-29 24.4687500 24.4687500         0          0.0
## 2012-11-30         NA 37.3680556        NA         34.5
```

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.

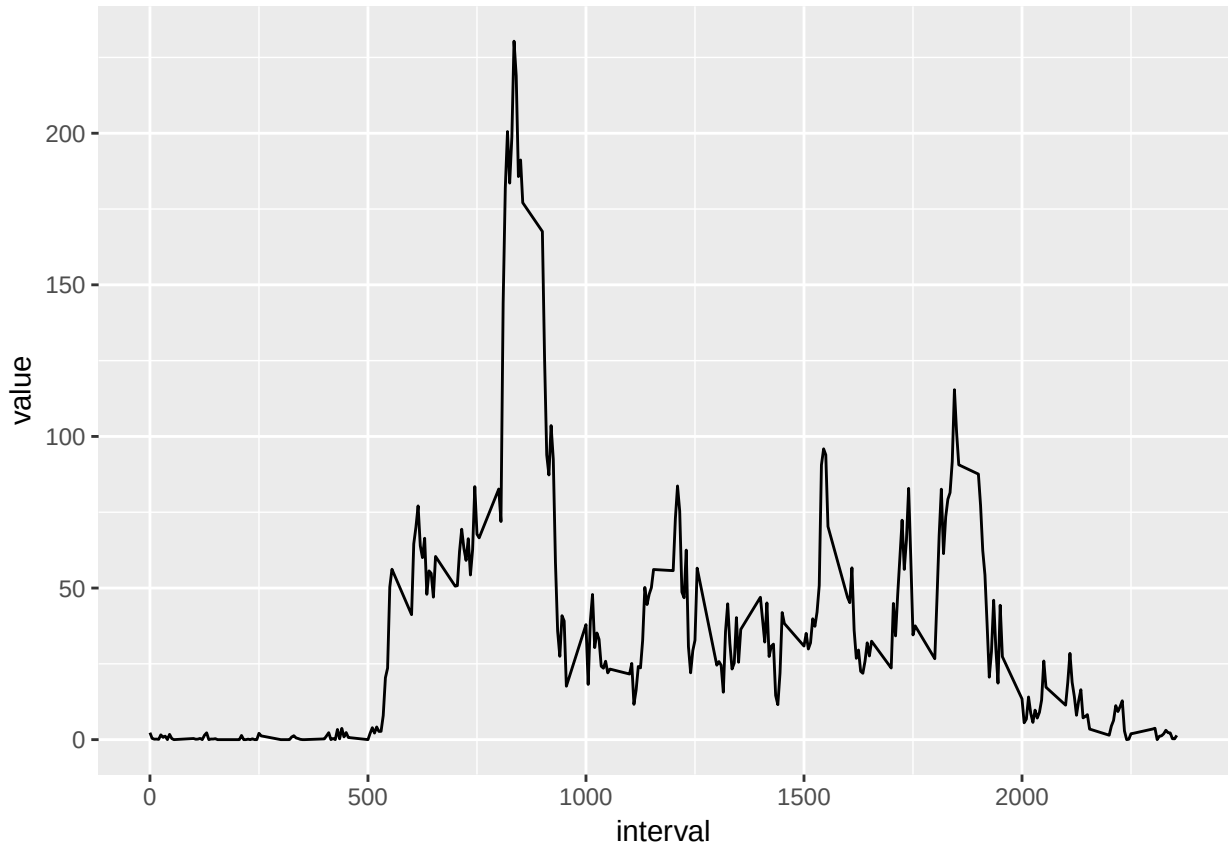
```
mydf1$date = gsub('-', '', mydf1$date)
mydf1$day = weekdays(as.Date(mydf1$date, "%Y%m%d"))
mydf1$weekend_YN = ifelse(mydf1$day %in% c('Sunday', 'Saturday'), 'Y', 'N')
head(mydf1)
```

```
##       date interval steps MeanOfInterval stepsNew    day weekend_YN
## 1 20121001        0    NA              2        2 Monday          N
## 2 20121001        5    NA              0        0 Monday          N
## 3 20121001       10    NA              0        0 Monday          N
## 4 20121001       15    NA              0        0 Monday          N
## 5 20121001       20    NA              0        0 Monday          N
## 6 20121001       25    NA              2        2 Monday          N
```

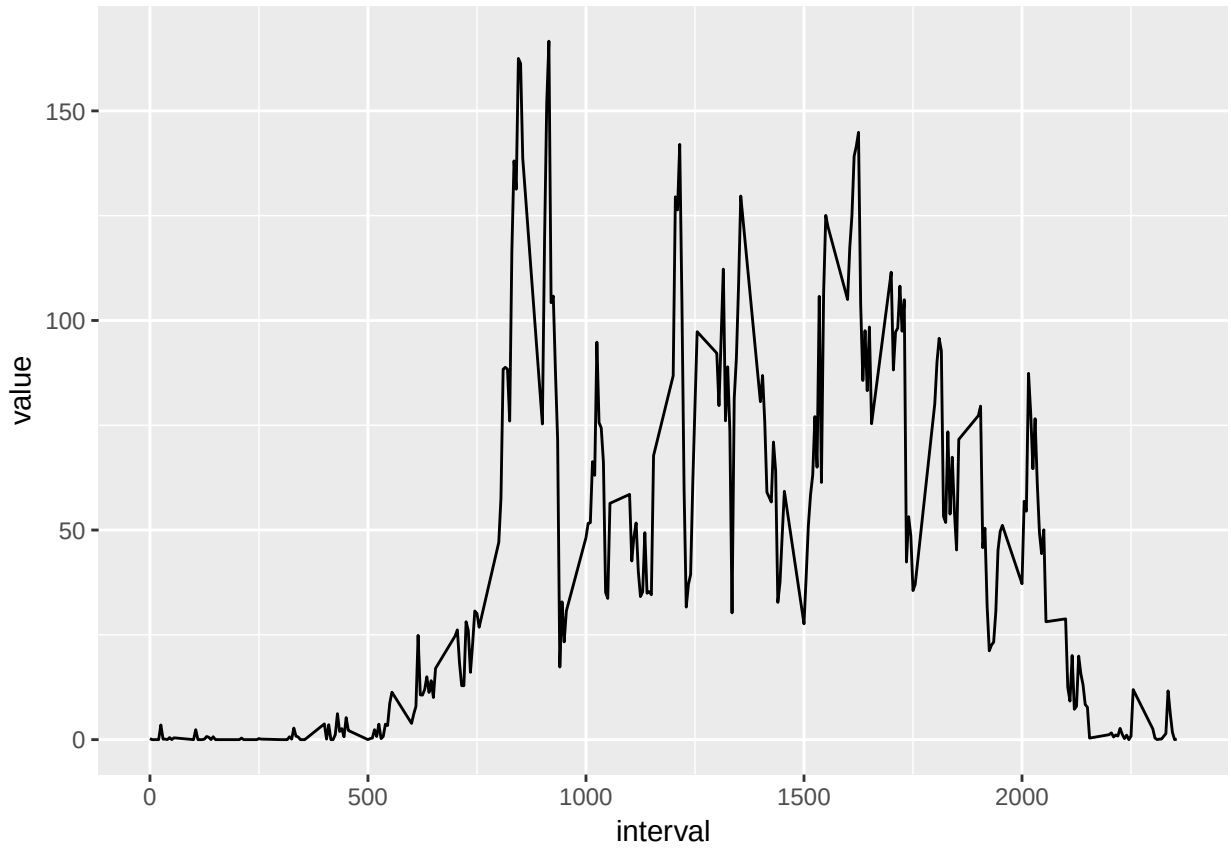
2. Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
weekdayData = mydf1[mydf1$weekend_YN == 'N',]
weekendData = mydf1[mydf1$weekend_YN == 'Y',]

weekdayMean = data.frame(value=tapply(weekdayData$stepsNew, weekdayData$interval, mean))
weekdayMean$interval = rownames(weekdayMean)
gg5 = ggplot(data=weekdayMean)
gg5 + geom_line(aes(x=as.numeric(rownames(weekdayMean)), y=value)) + xlab('interval')
```

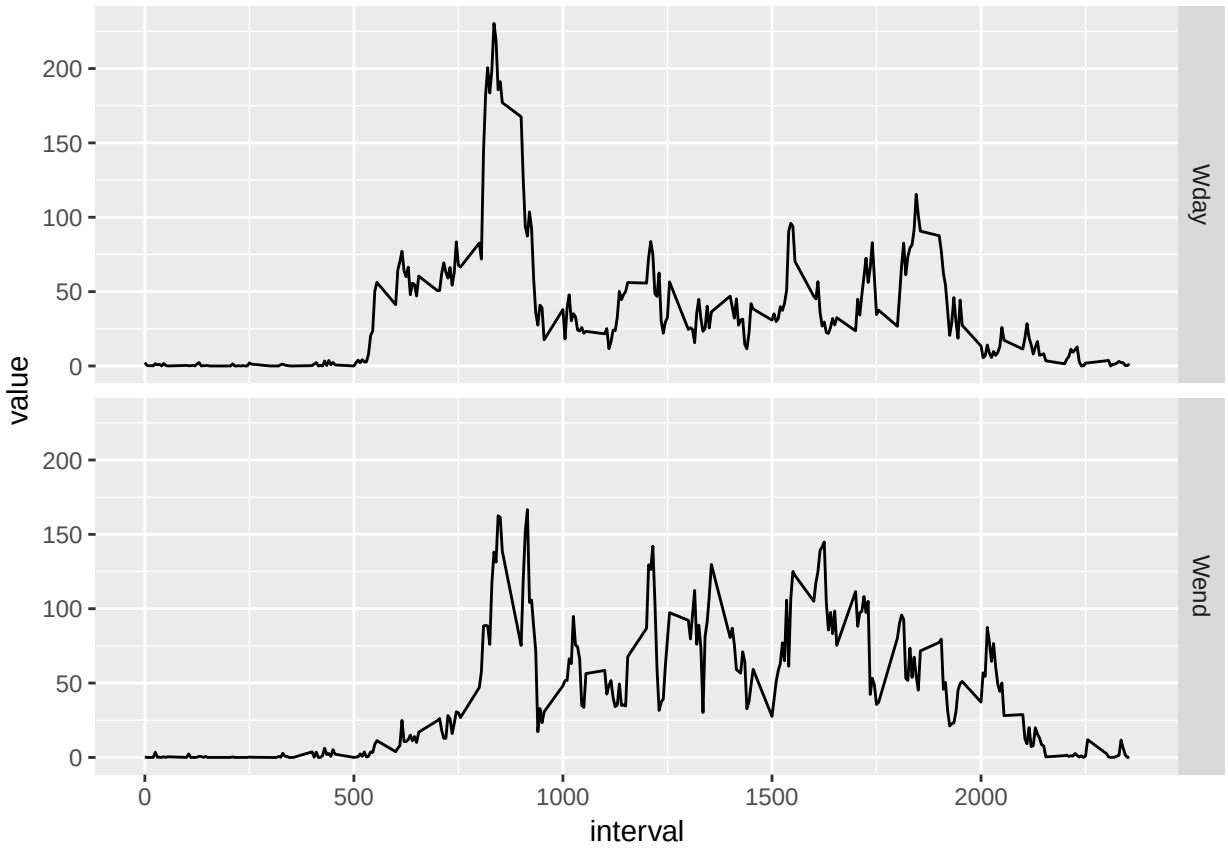



```
weekendMean = data.frame(value=tapply(weekendData$stepsNew, weekendData$interval, mean))
weekendMean$interval = rownames(weekendMean)
gg6 = ggplot(data=weekendMean)
gg6 + geom_line(aes(x=as.numeric(rownames(weekendMean)), y=value)) + xlab('interval')
```




```
weekdayMean$DateType = 'Wday'
weekendMean$DateType = 'Wend'
ddda = rbind(weekdayMean, weekendMean)

ggplot(data=ddda) +
    geom_line(aes(x=as.numeric(interval), y=value)) +
    facet_grid(DateType ~ .) +
    xlab('interval')
```


value
200
150
100
50
0
Wday
Wend
0
500
1000
1500
2000
interval