Week2Project

```
library(ggplot2);library(stats);library(knitr);library(base);
library(dplyr);library(reshape2);library(gridExtra)
```

Loading and preprocessing the data

```
dat=read.csv('/Users/apple/Desktop/Cousera/Data Science/7. Reproducibale Research/activity.csv')
# Let us check the data structure
dat$date=as.Date(dat$date,format = "%Y-%m-%d")
str(dat)
```

```
## '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 ...
```

1. Is there missing value? Yes, we can delete the observations with missing value first, if need, we will add them back to do further analysis.

```
# sum(is.na(date))=0; sum(is.na(interval)) =0    date and interval columns have no missing value
# delete the observations with missing "steps" variable.
dat2=dat[which(is.na(dat$steps)==F),]
str(dat2)
```

```
## 'data.frame':    15264 obs. of  3 variables:
##  $ steps   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ date    : Date, format: "2012-10-02" "2012-10-02" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

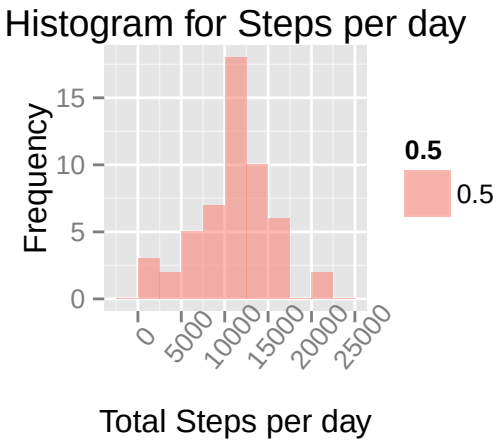
What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
Tsteps=aggregate(steps~date,dat2,sum)
```

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

```
gH=ggplot(Tsteps,aes(steps,alpha=0.5))+geom_histogram(fill="salmon",binwidth=2500)
gH=gH+ggtitle("Histogram for Steps per day")+ylab('Frequency')+xlab('Total Steps per day');
gH=gH+theme(axis.text.x=element_text(angle=50))
gH
```



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

```
avg.Step=round(mean(Tsteps$steps,na.rm = T),2)    ;  med.Step=round(median(Tsteps$steps),2)
print(paste('The mean for Total Steps per day is:',avg.Step,'and the meadian of it is:',med.Step))
```

```
## [1] "The mean for Total Steps per day is: 10766.19 and the meadian of it is: 10765"
```

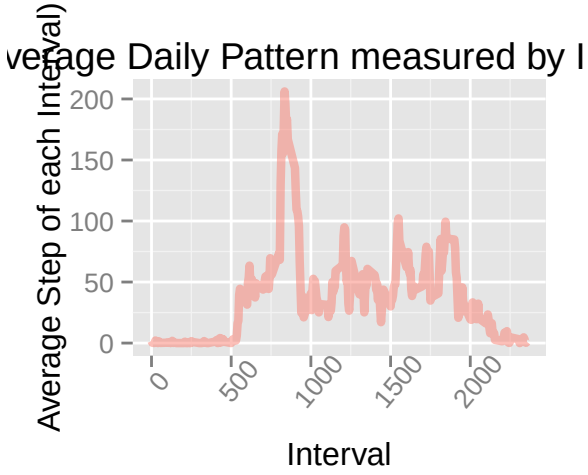
What is the average daily activity pattern?

1. Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
dat3=arrange(dat2,interval)
# In order to to draw a time seiries, the Interval must be integer rather than factor
print(paste("The interval is in range : [",min(dat3$interval),max(dat3$interval),"]."))
```

```
## [1] "The interval is in range : [ 0 2355 ]."
```

```
datStep=aggregate(steps~interval,dat3,mean)
g=ggplot(datStep,aes(interval,steps))+geom_line(color='salmon',lwd=1.5,alpha=0.5)
g=g+ggtitle("Average Daily Pattern measured by Interval")+ylab("Average Step of each Interval)")
g=g+xlab('Interval')+ theme(axis.text.x=element_text(angle=50)); g
```



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

```
maxStep=max(datStep$steps)
print(paste('The 5 minute Interval with Identifier',datStep[which(datStep$steps==maxStep),][1],
            'has the maximum steps:',as.integer(maxStep) ,"."))
```

```
## [1] "The 5 minute Interval with Identifier 835 has the maximum steps: 206 ."
```

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(the total number of rows with NA)

```
#sum(is.na(dat$date));sum(is.na(dat$interval))   # There is no missing value in date and interval
miss=sum(is.na(dat$steps))
percent=round((sum(is.na(dat$steps))/dim(dat)[1]),4 )
print(paste('The',miss,'missing values account for', percent*100,'% as in population.'))
```

```
## [1] "The 2304 missing values account for 13.11 % as in population."
```

2. Ways to deal with the missing value?

- *Using the average steps per day.*
  If we fill in missing value by this way, we should figure out, how many days with missing values?

```
day=dat[which(is.na(dat$steps)==1),]  ;  tab=table((as.factor(day$date))); tab
```

```
##
## 2012-10-01 2012-10-08 2012-11-01 2012-11-04 2012-11-09 2012-11-10
##        288        288        288        288        288        288
## 2012-11-14 2012-11-30
##        288        288
```

```
print(paste('There are',dim(table(as.factor(dat$date))),'days in total.'))
```

```
## [1] "There are 61 days in total."
```

```
print(paste('There are',dim(tab),'days with missing steps value.'))
```

```
## [1] "There are 8 days with missing steps value."
```

- *Using the average steps per interval.*
  If we fill in missing value by this way, we should figure out, how many kinds of interval with missing values?

```
tab2=table((as.factor(day$interval)));
print(paste('There are',dim(table(as.factor(dat$interval))),'kinds of interval in total.'))
```

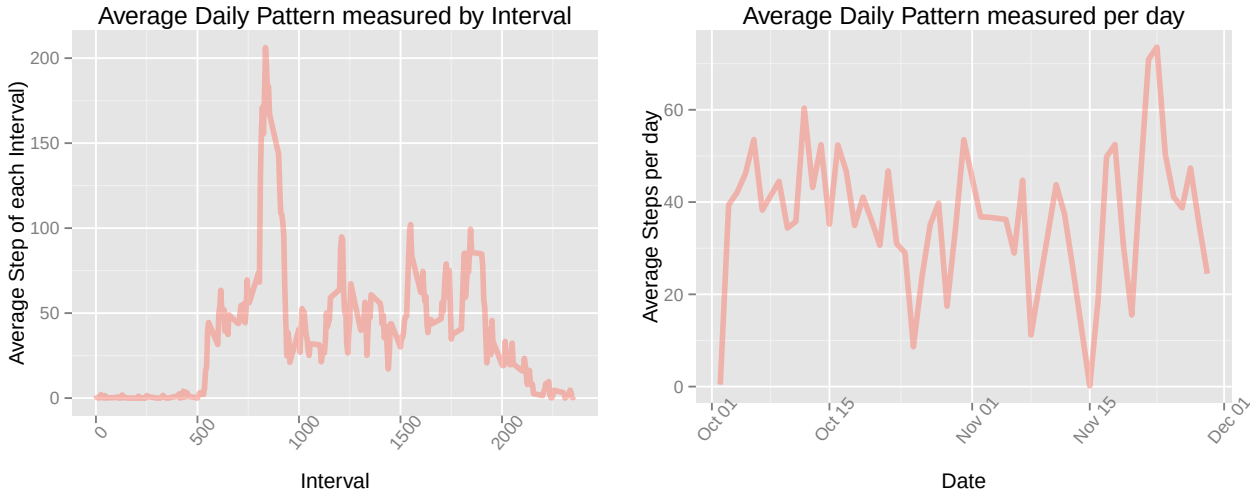
```
## [1] "There are 288 kinds of interval in total."
```

```
print(paste('There are',dim(tab2),' kinds of interval with missing steps value.'))
```

```
## [1] "There are 288  kinds of interval with missing steps value."
```

- *Compare the average steps both mearsured by per interval and per day.*

```
# This is average steps per day
avg=aggregate(steps~date,dat2,mean)
g1=ggplot(avg,aes(date,steps,)) + geom_line(lwd=1.5,col='salmon',alpha=0.5)
g1=g1+ggtitle("Average Daily Pattern measured per day")+xlab('Date')+ylab('Average Steps per day')
g1=g1+theme(axis.text.x=element_text(angle=50))
grid.arrange(g,g1,ncol=2)
```



Firstly, from the comparison figures, there is a large variation either thoughout the day or throught out the interval. Although, the outlier of average steps measured by interval is more significant.
Second, the days with missing value account for 13% as total days. However, each kind of interval has the missing value.
Most Importantly, the 8 days, with missing data, has no records in our "avg" dataset, which is the dataset stores the average steps per day.

```
# dataset "tab" stores the 8 days with missing steps data.
# And "avg" stores the average steps per day. And unfortunately, they have no intersection !
f=as.Date(names(tab))
sum(avg$date %in% f)
```

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

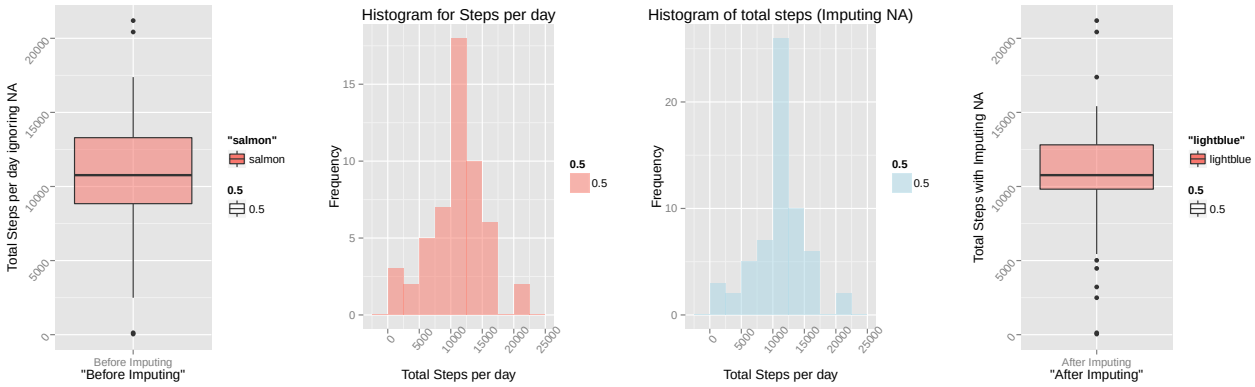
To sum up, we have to fill 'NA' with the average step per interval in this way.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
- Create the table stores the value we need to replace the missing value in 'dat'.

```
# 1)"datStep" dataset stores the average steps per interval. 2) "dat" is the original dataset with "NA"
# Sort "dat" by interval, and using the value in "datStep" to fill in the 'NA' in "dat"
new=arrange(dat,interval)
for (i in 1:nrow(new)){
    if (is.na(new$steps[i])==T)            # filter :  the step is missing.
      {   intervalIndex=new$interval[i]    # save the specified interval index with missing value.
         steps=datStep$steps[which(datStep$interval==intervalIndex)]
         # corresponding average step for each specified interval index.
         new$steps[i]=steps
      }
}
str(new)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  1.72 0 0 47 0 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-02" ...
##  $ interval: int  0 0 0 0 0 0 0 0 0 0 ...
```

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?

```
# Recall the histogram of total number of steps each day is gH( ignoring the missing value).
# There are 8 days has no records.
new=arrange(new,date)          ;    T=aggregate(steps~date,new,sum)
gH2=ggplot(T,aes(steps,alpha=0.5))+geom_histogram(fill='lightblue',binwidth=2500)
gH2=gH2+ggtitle("Histogram of total steps (Imputing NA)")
gH2=gH2+xlab('Total Steps per day') +ylab('Frequency')
gH2=gH2+theme(axis.text.x=element_text(angle=50))
b1=qplot(y=Tsteps$steps,x='Before Imputing', geom='boxplot',fill="salmon", alpha=0.5)
b1=b1+ylab("Total Steps per day ignoring NA")+theme(axis.text.y=element_text(angle=50))
b2=qplot(y=T$steps,     x='After Imputing' ,geom='boxplot', fill="lightblue",alpha=0.5)
b2=b2+ylab("Total Steps with Imputing NA")+theme(axis.text.y=element_text(angle=50))
grid.arrange(b1,gH,gH2,b2,ncol=4)
```



```
# Compare the dataset before and after imputing "NA"
avg=round(mean(T$steps),2)    ;    med=round(median(T$steps),2)
t=data.frame( MinTotStep=c(min(Tsteps$steps), min(T$steps)),
              Q1_TotStep=c( quantile(Tsteps$steps)[2],quantile(T$steps)[2] ),
              AvgTotStep=c(avg.Step,avg),
              MedTotStep=c(med.Step,med),
              Q3_TotStep=c( quantile(Tsteps$steps)[4],quantile(T$steps)[4] ),
              MaxTotStep=c(max(Tsteps$steps), max(T$steps)),
              StdTotStep=c(sd(Tsteps$steps),sd(T$steps)))
rownames(t)=c('Ignore NA','Imputing NA')  ; kable(t)
```

	MinTotStep	Q1_TotStep	AvgTotStep	MedTotStep	Q3_TotStep	MaxTotStep	StdTotStep
Ignore NA	41	8841	10766.19	10765.00	13294	21194	4269.180
Imputing NA	41	9819	10766.19	10766.19	12811	21194	3974.391

To sum up - Before and after the imputing "NA", the skewness of distribution of steps is around the same. That is, the mean, median, max and min stay the same, but the 1st and 3st quantile of steps with imputing slides closer to the mean. Moreover, imputing 'NA' leads to a smaller standard deviation. - Imputing has decreased the variation of steps, making the value concentrate around the mean. As a result,the number outliers increase a little.(these new outliers were not outliers before imputing)

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 identifier with level – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
new$indicator='Weekday'
new$indicator[which( weekdays(new$date) %in% c('Saturday','Sunday') ) ]='Weekend'
new$indicator=as.factor(new$indicator)
# now,we divide the new to 2 tables
wkend=new[which(new$indicator=='Weekend'),]
wkday=new[which(new$indicator=='Weekday'),]
```

2. Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

- Let us check the distribution table first:

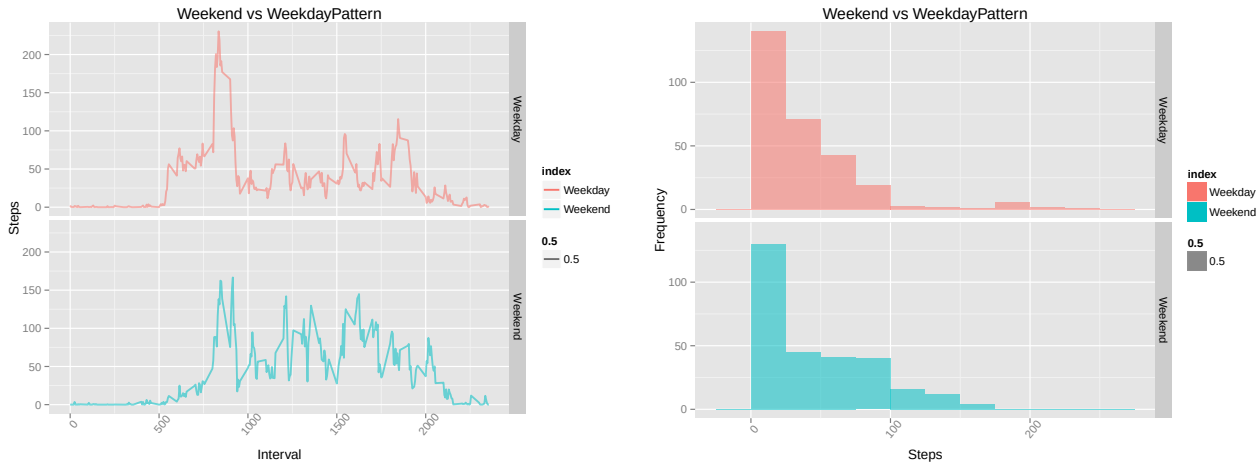
```
w1=aggregate(steps~interval,wkend,mean)
w2=aggregate(steps~interval,wkday,mean)
w=data.frame( TotalObs=c(dim(w1)[1] , dim(w2)[1]) ,
              MinAvgStep=c(min(w1$steps), min(w2$steps)),
              Q1_AvgStep=c(quantile(w1$steps)[2],quantile(w2$steps)[2] ),
              AvgAvgStep=c(quantile(w1$steps)[3],quantile(w2$steps)[3] ),
              MedAvgStep=c(median(w1$steps), median(w2$steps)),
              Q3_AvgStep=c(quantile(w1$steps)[4],quantile(w2$steps)[4] ),
              MaxAvgStep=c(max(w1$steps), max(w2$steps)),
```

```
              StdAvgStep=c(sd(w1$steps),sd(w2$steps))           )
w=round(w,2) ; rownames(w)=c('Weekend Pattern','Weekday Pattern')  ; kable(w)
```

	TotalObs	MinAvgStep	Q1_AvgStep	AvgAvgStep	MedAvgStep	Q3_AvgStep	MaxAvgStep
Weekend Pattern	288	0	1.24	32.34	32.34	74.65	166.64
Weekday Pattern	288	0	2.25	25.80	25.80	50.85	230.38

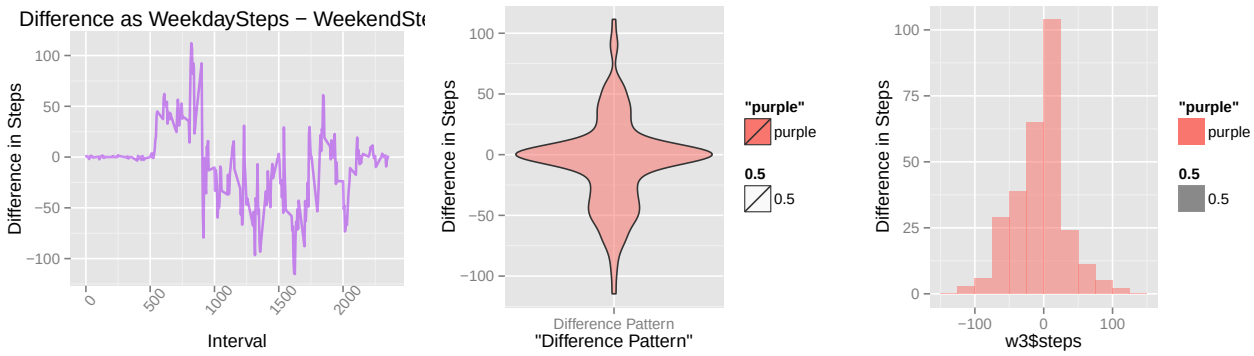
- The panel plot of Weekend Pattern vs Weekday Pattern.

```
w1$index='Weekend' ; w2$index='Weekday'  ; w=rbind(w1,w2)  ;w$index=as.factor(w$index)
g1=ggplot(w,aes(interval,steps,col=index,alpha=0.5))+geom_line(lwd=0.8)
g1=g1+facet_grid(index~.)        # setting the vertical panel
g1=g1+ggtitle("Weekend vs WeekdayPattern")+xlab("Interval")+ylab('Steps')
g1=g1+theme(axis.text.x=element_text(angle=50));
g2=ggplot(w,aes(steps,fill=index,alpha=0.5))+geom_histogram(binwidth=25)
g2=g2+facet_grid(index~.)
g2=g2+ggtitle("Weekend vs WeekdayPattern")+xlab("Steps")+ylab('Frequency')
g2=g2+theme(axis.text.x=element_text(angle=50))
grid.arrange(g1,g2,ncol=2)
```



- The panel of the difference between Weekday and Weekend.

```
w3=w2; w3$steps=w2$steps-w1$steps
g3=ggplot(w3,aes(interval,steps))+geom_line(color='purple',alpha=0.5,lwd=0.8)
g3=g3+ggtitle("Difference as WeekdaySteps - WeekendSteps")
g3=g3+xlab("Interval")+ylab('Difference in Steps')+theme(axis.text.x=element_text(angle=50))
g4=qplot(y=w3$steps,x='Difference Pattern', geom='violin', fill= "purple",alpha=0.5)
g4=g4+ylab('Difference in Steps')
g5=qplot(w3$steps, geom='histogram', binwidth=25, fill= "purple",alpha=0.5)
g5=g5+ylab('Difference in Steps')
grid.arrange(g3,g4,g5,ncol=3)
```



To sum up:
1. When interval is less than 900, the weekday has higher average daily steps than that in weekend. However, the situation is reversed during the interval larger than 900.
2. The distribution of the difference between weekday and weekend is around normal with the mean=0.

```
summary(w3$steps)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -115.300  -28.690    0.000   -6.756    6.071  112.200
```