Reproducible Research_Peer Assignment 1

Hin-Weng Wan

September 16, 2016

Loading and Processing Data

Unzip 'activity.zip in working directory Library:dplyr, ggplot2in RStudio

```
data<-read.csv("activity.csv")
dat1<-na.omit(data)
as.numeric(dat1$month)
```

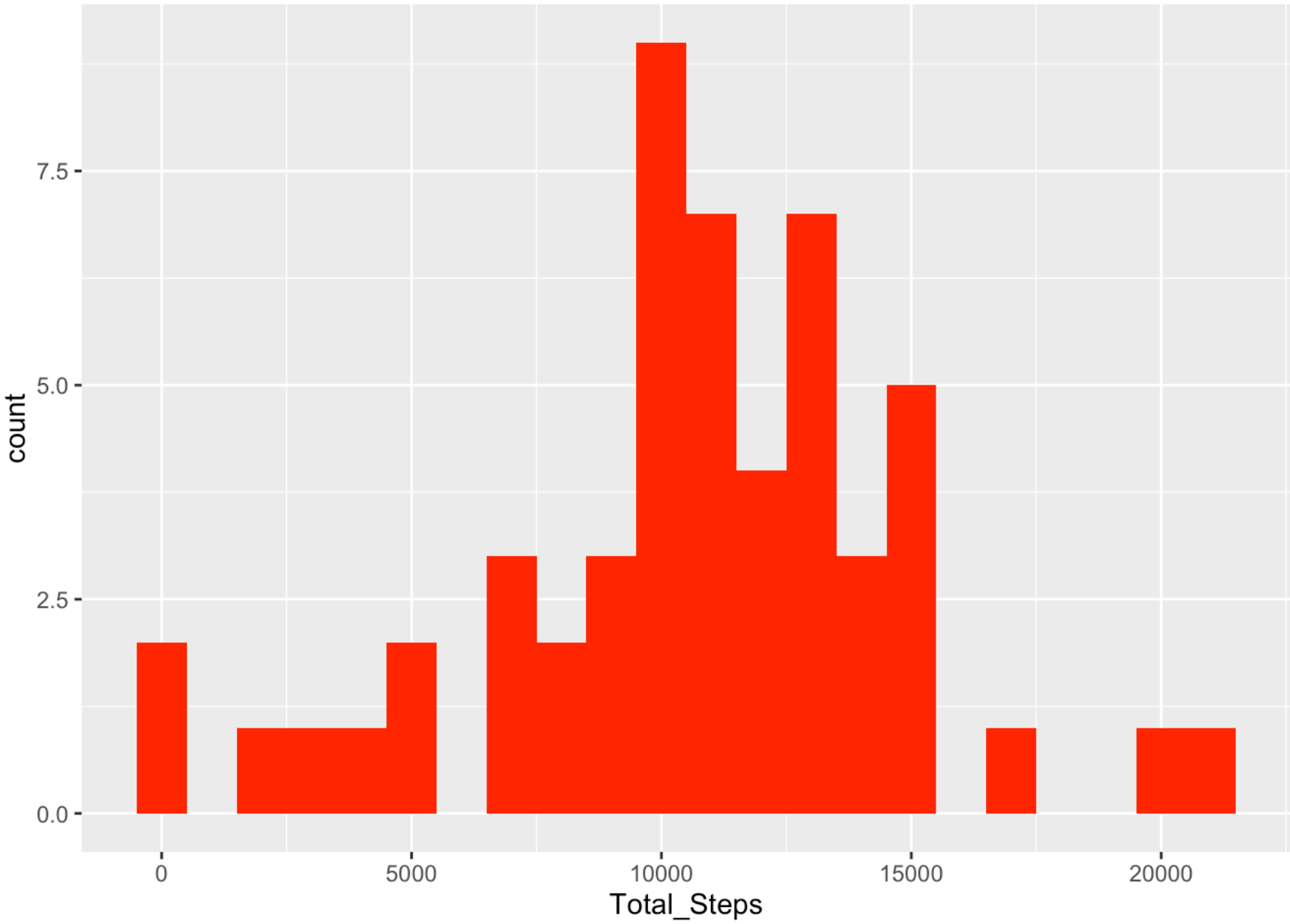
```
## numeric(0)
```

```
View(dat1)
```

Total number of steps each day _histogram

```
totalsteps<-with(dat1,aggregate(steps,list(date=date),sum))
totalsteps<-rename (totalsteps,Total_Steps=x)
View(totalsteps)

ggplot(totalsteps,aes(x=Total_Steps ))+geom_histogram(binwidth=1000,fill="red")
```



Mean and Median of steps each day

```
mean(totalsteps $Total_Steps)
```

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

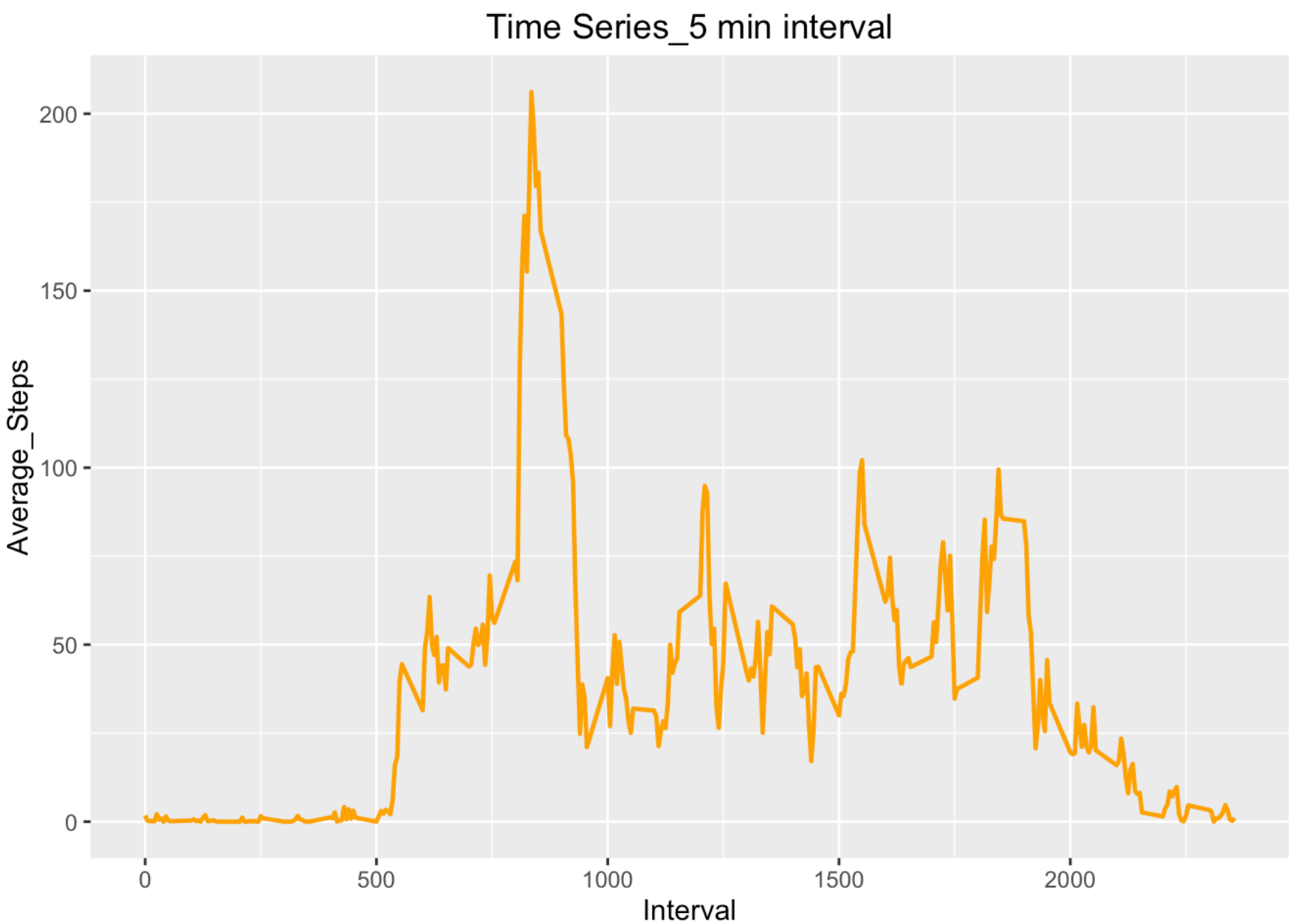
```
median(totalsteps $Total_Steps)
```

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

Time Series Plot _ Average number of steps averaged across all days

```
avsteps <- with(dat1,aggregate(steps,list(interval=interval),mean))
avsteps<-rename(avsteps,Average_Steps=x,Interval=interval )
View(avsteps)

ggplot (avsteps,aes(Interval,Average_Steps))+geom_line(color="orange", size=0.8) +
labs(title ="Time Series_5 min interval")
```



5 minute interval that on average contains the max number of steps is 835 with average steps of 206.2

You can also get the answer from the dataframe generated with the RCode ‘View avsteps’ above.

```
avsteps[avsteps$Average_Steps==max(avsteps$Average_Steps),]
```

```
##     Interval Average_Steps
## 104      835      206.1698
```

Imputing missing values

Total of NAs

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

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

Use the mean of the 5 minute interval for imputation

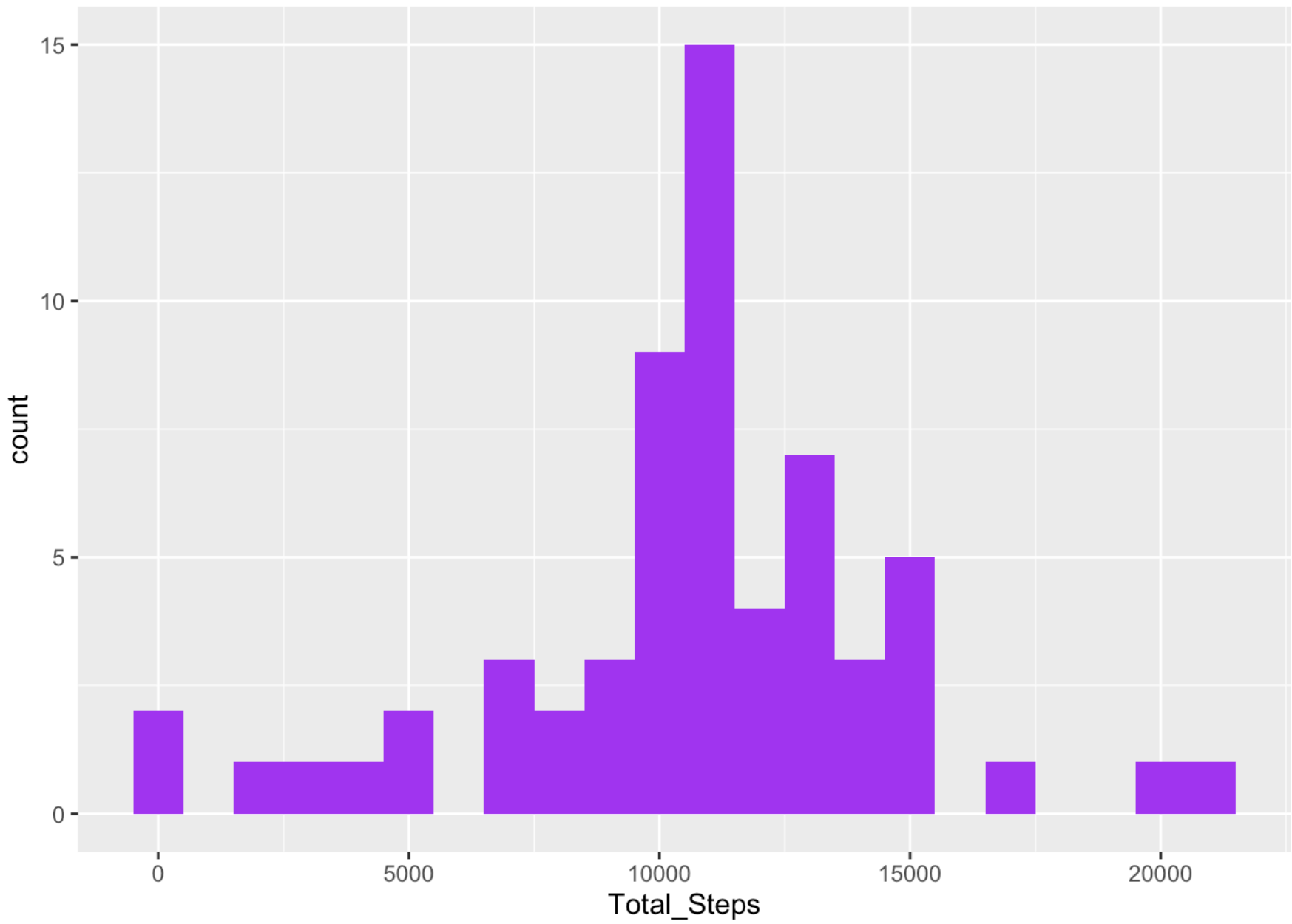
Create a new dataset that is equal to the original data data set but with missing values substituted

```
dat2<-mutate (data,steps=ifelse(is.na(steps),as.integer
(avsteps$Average_Steps)  ,steps ))
View(dat2)
```

Total number of steps taken each day _ histogram

```
totalsteps2<-with(dat2,aggregate(steps,list(date=date),sum))
totalsteps2<-rename (totalsteps2,Total_Steps=x)
View(totalsteps2)

ggplot (totalsteps2,aes(x=Total_Steps))+geom_histogram(binwidth=1000,fill="purple"
)
```



Display the 2 histograms in one plot i.e the top histogram has imputed values and the bottom histogram has no imputed values

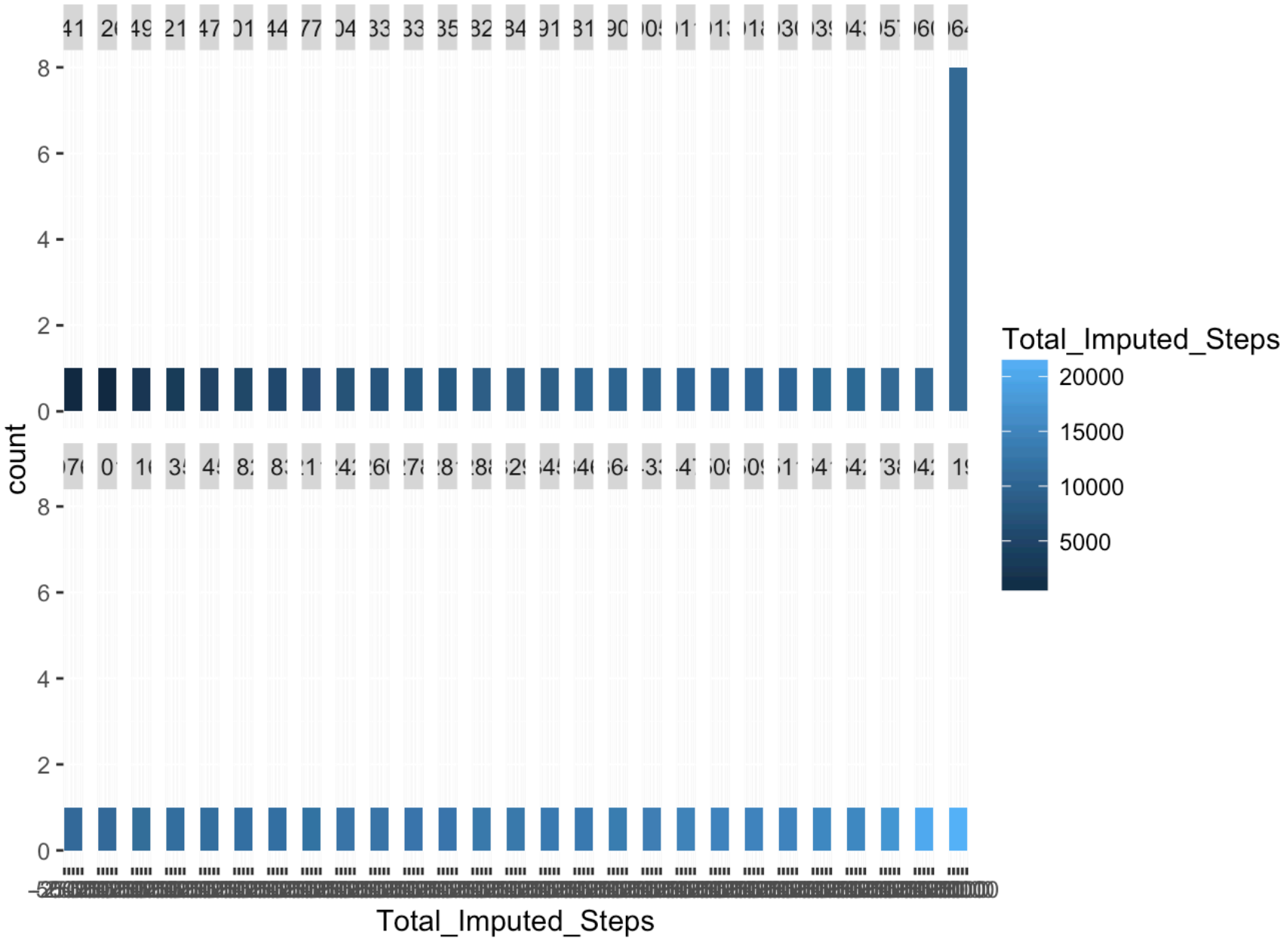
```
Joint<-full_join (totalsteps2,totalsteps,by="date")
Joint<-rename (Joint,Total_Imputed_Steps=Total_Steps.x)
View(Joint)

ggplot (Joint,aes(x=Total_Imputed_Steps,fill=Total_Imputed_Steps))+geom_histogram(
binwidth=100000)+scale_size_identity()+facet_wrap(~Total_Imputed_Steps ,nrow=2,"fr
ee_x")
```

```
## Warning: Coercing `ncol` to be an integer.
```

```
## Warning in sanitise_dim(ncol): NAs introduced by coercion
```

```
## Warning: `ncol` is missing or less than 1 and will be treated as NULL.
```



Mean and Median of steps each day based on imputed values

```
mean(totalsteps2$Total_Steps)
```

```
## [1] 10749.77
```

```
median(totalsteps2$Total_Steps)
```

```
## [1] 10641
```

Mean and Median of steps each day _ Without Imputed Values vs With Imputed Values

Impact from imputing missing values on the estimates of the total daily number of steps is in a mean difference of -16.4and a median difference of -124, where both the imputed mean and median are lower

```
mean1<-mean(totalsteps$Total_Steps)
mean2<-mean(totalsteps2$Total_Steps)
mean2-mean1
```

```
## [1] -16.41819
```

```
median1<-median(totalsteps$Total_Steps)
median2<-median(totalsteps2$Total_Steps)
median2-median1
```

```
## [1] -124
```

Differences in activities between weekdays and weekends

Create a new factor variable with 2 levels -‘weekday’ and ‘weekend’

```
dat2$date<-as.Date(dat2$date)
weekdays1<-c("Monday","Tuesday","Wednesday","Thursday",
             "Friday")
dat2$wday<-c('weekend','weekday')[weekdays(dat2$date)%in%
            weekdays1+1L]

table(dat2$wday)
```

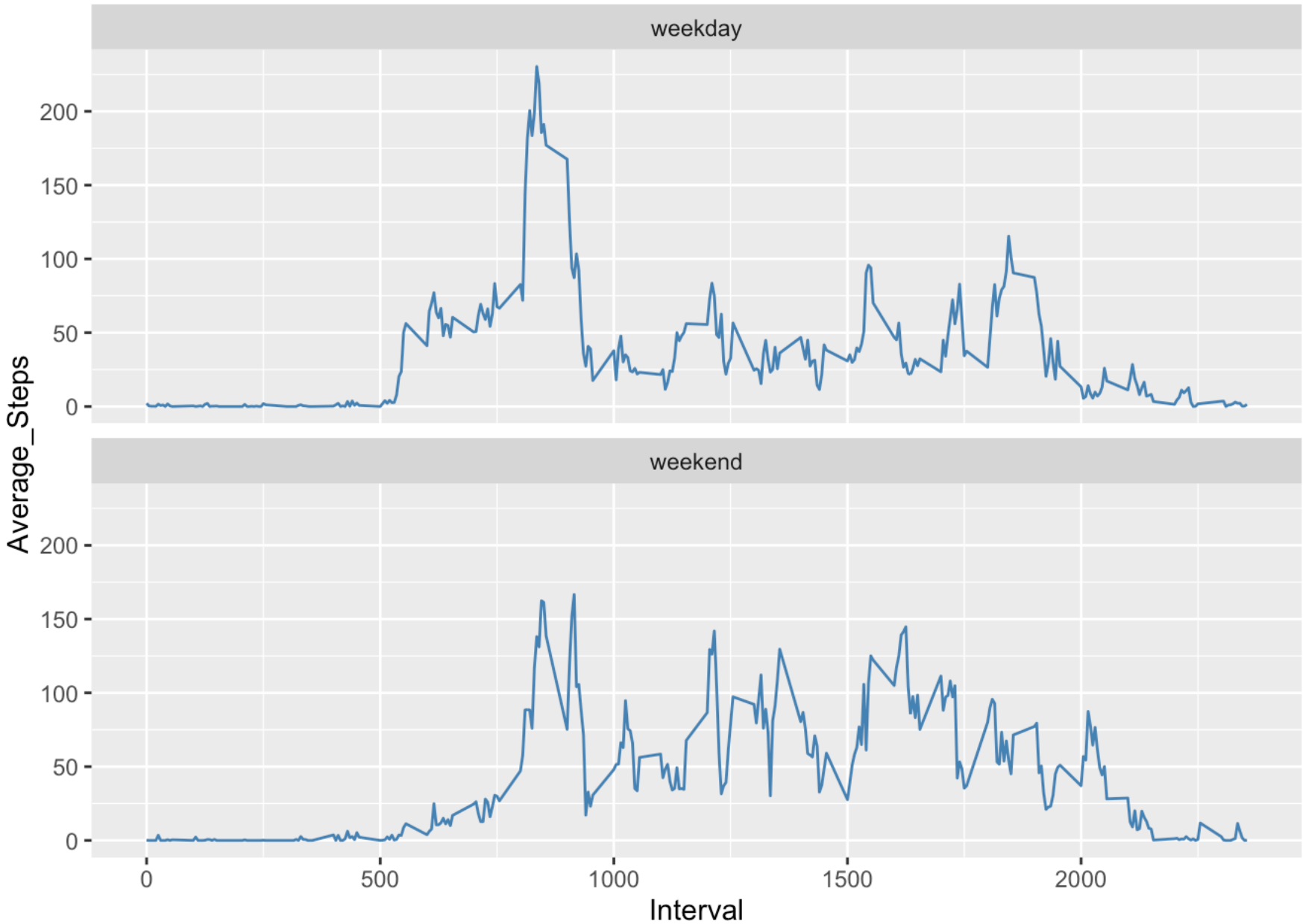
```
##
## weekday weekend
##   12960    4608
```

```
View(dat2)
```

Time Series Panel Plot - interval(x axis) and Average Steps(y axis)

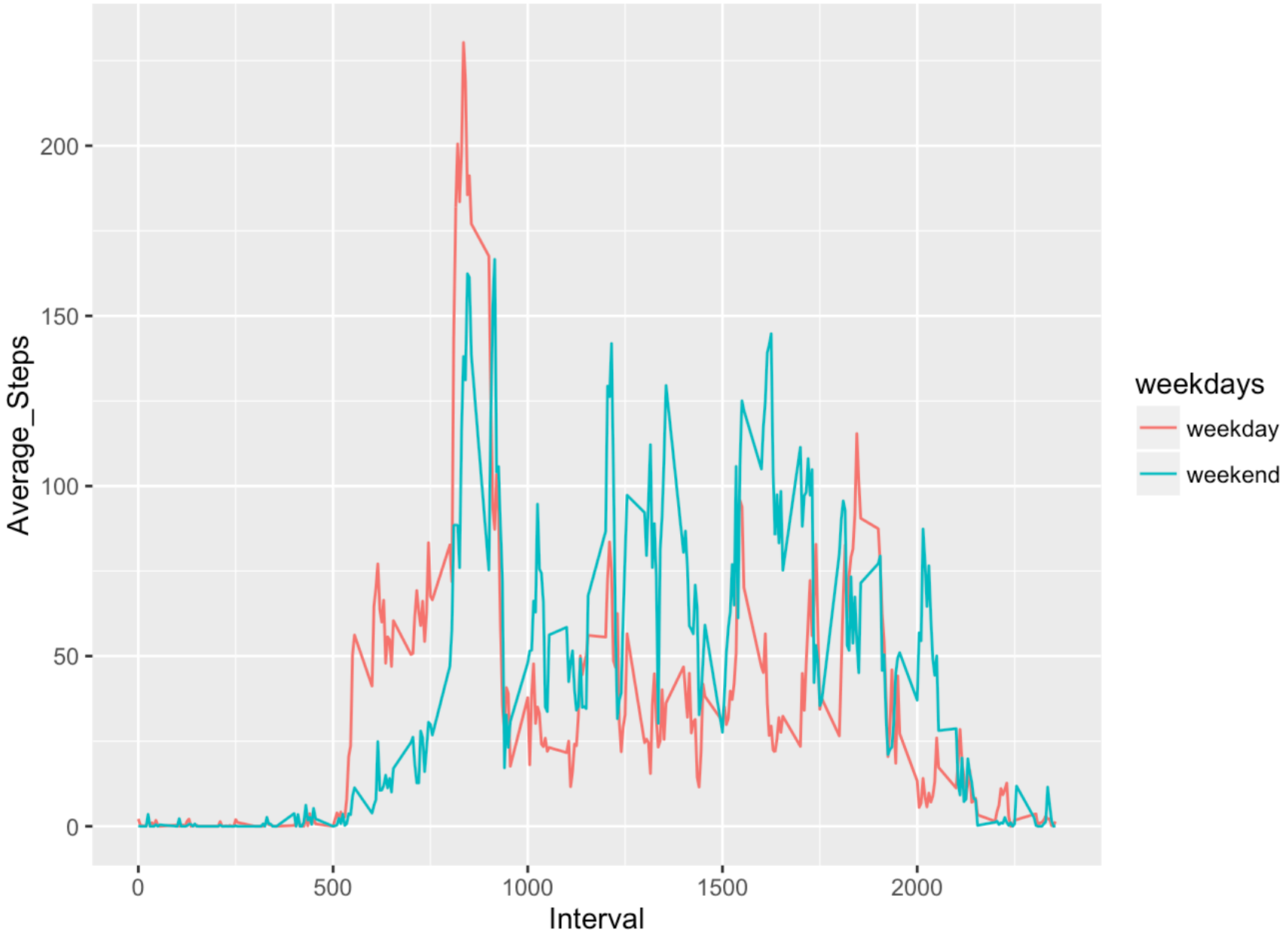
```
avsteps2<- with(dat2,aggregate(steps,list(interval=interval,weekdays=dat2$wday),me
an) )
avsteps2<-rename (avsteps2,Average_Steps=x,Interval=interval)
View(avsteps2)

ggplot (avsteps2,aes(Interval,Average_Steps))+geom_line(color="steelblue")+facet_w
rap(~weekdays,nrow=2)
```



Time Series Single Plot -interval (x axis) and Average Steps (y axis). This overlay of the time series provides a clearer distinction

```
ggplot (avsteps2,aes(Interval,Average_Steps,
color=weekdays))+geom_line ()
```


Average_Steps
200
150
100
50
0
0
500
1000
1500
2000
Interval
weekdays
weekday
weekend