Code ▾

R Notebook for data incubator_minjie XU

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

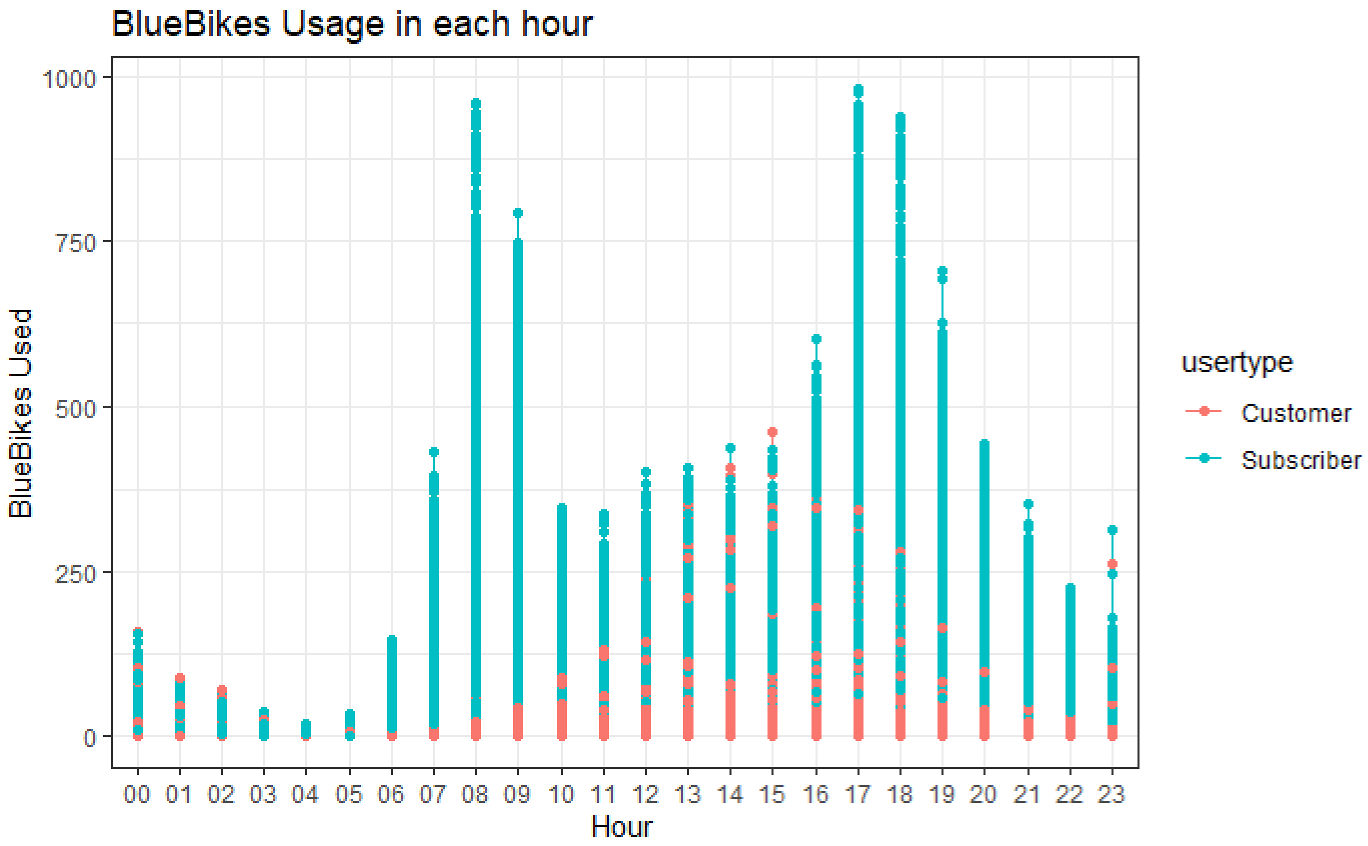
Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
library(data.table) # for faster loading on large dataset
library(tidyverse) # for ggplot, dplyr
library(tis) # for holiday function
busystation<-NULL
timeall<-NULL
hourlist<-seq(0,23,by=1)
for (y in c(2017,2018,2019)){
        for (m in seq(1,12,by=1)){
                if (y == 2019 & m == 4) { return() }
                ## load blue bike rental data from https://s3.amazonaws.com/hubway-data/index.html
                path<-paste0("C:\\Users\\Minjie\\Documents\\2018 Fall Master\\Data Mining\\project\\hubway\\",y,sprintf("%02i",m),"-bluebikes-tripdata.csv")
                bikedata <- as.data.frame(fread(path))

                # dataset for busy station definition and visulization
                bikedata$hour<-as.factor(substr(bikedata$starttime,12,13))
                NS<-bikedata$`end station latitude`-bikedata$`start station latitude`
                bikedata$direction_NS<-"S"
                bikedata$direction_NS[NS>0]<-"N"
                bikedata$direction_NS[NS==0]<-"0"
                EW<-bikedata$`end station longitude`-bikedata$`start station longitude`
                bikedata$direction_EW<-"W"
                bikedata$direction_EW[EW>0]<-"E"
                bikedata$direction_EW[EW==0]<-"0"
                bikedata$duration_min <- round(bikedata$tripduration/60, 0)
                bikedata1<-bikedata[c("starttime","stoptime","start station id","start station latitude","start station longitude",
                                      "end station id","end station latitude","end station longitude",
                                      "direction_NS","direction_EW","duration_min","usertype")]
                busystation<-rbind(busystation,bikedata1)

                # # dataset for time series and ML analysis
                # bikedata2<-bikedata[c("starttime","usertype")]
                # bikedataall<-rbind(bikedataall,bikedata2)


                for(d in seq(1,31,by=1)){
                        # Check if leap year
                        if (y%%400 == 0){
                                leap = TRUE
                        } else if (y%%100 == 0){
                                leap = FALSE
                        } else if (y%%4 == 0){
                                leap = TRUE
                        } else leap = FALSE

                        # check date of every month
                        if (m == 2 & leap & d > 29){
                                next
                        } else if (m == 2 & d > 28){
                                next
                        } else if (m %in% c(4, 6, 9, 11) & d > 30){
                                next
                        }

                        #make a complete time list to storage dates and hours
                        timelist<-cbind(rep(paste(y,m,d,sep="-"),24),hourlist)
                        timeall<-rbind(timeall,timelist)

                }

        }
}
# remove daylight savings hours
timeall<-as.data.frame(timeall)
colnames(timeall) <- c("date","hour")
timeall<-timeall[!(timeall$date %in% c("2017-3-12","2018-3-11","2019-3-10") & timeall$hour==2),]
winterdaylight<-data.frame(date=c("2017-11-5","2018-11-4"),hour=c(2,2))
timeall<-rbind(timeall,winterdaylight)
# dataset for time series and ML analysis
bikedataall<-busystation[,c("starttime","usertype")]
bikedataall$newtime<-format(round(as.POSIXct(bikedataall$starttime, format="%Y-%m-%d %H:%M:%S"),
units="hours"))
```

Hide

```
#combine bike usage in function of hour
bikedata3<-bikedataall %>%
        group_by(newtime,usertype) %>%
        summarize(count = n())
bikedata3$hour<-substr(bikedata3$newtime,12,13)
bikedata3 %>% ggplot(aes(x = hour, y = count,color=usertype)) +
        geom_line() +
        geom_point() +
        labs(title = "BlueBikes Usage in each hour",
             x = "Hour",  y = "BlueBikes Used") +
        theme_bw()
```



Hide

```
bikedataall<-bikedata3[,-4]
```

Hide

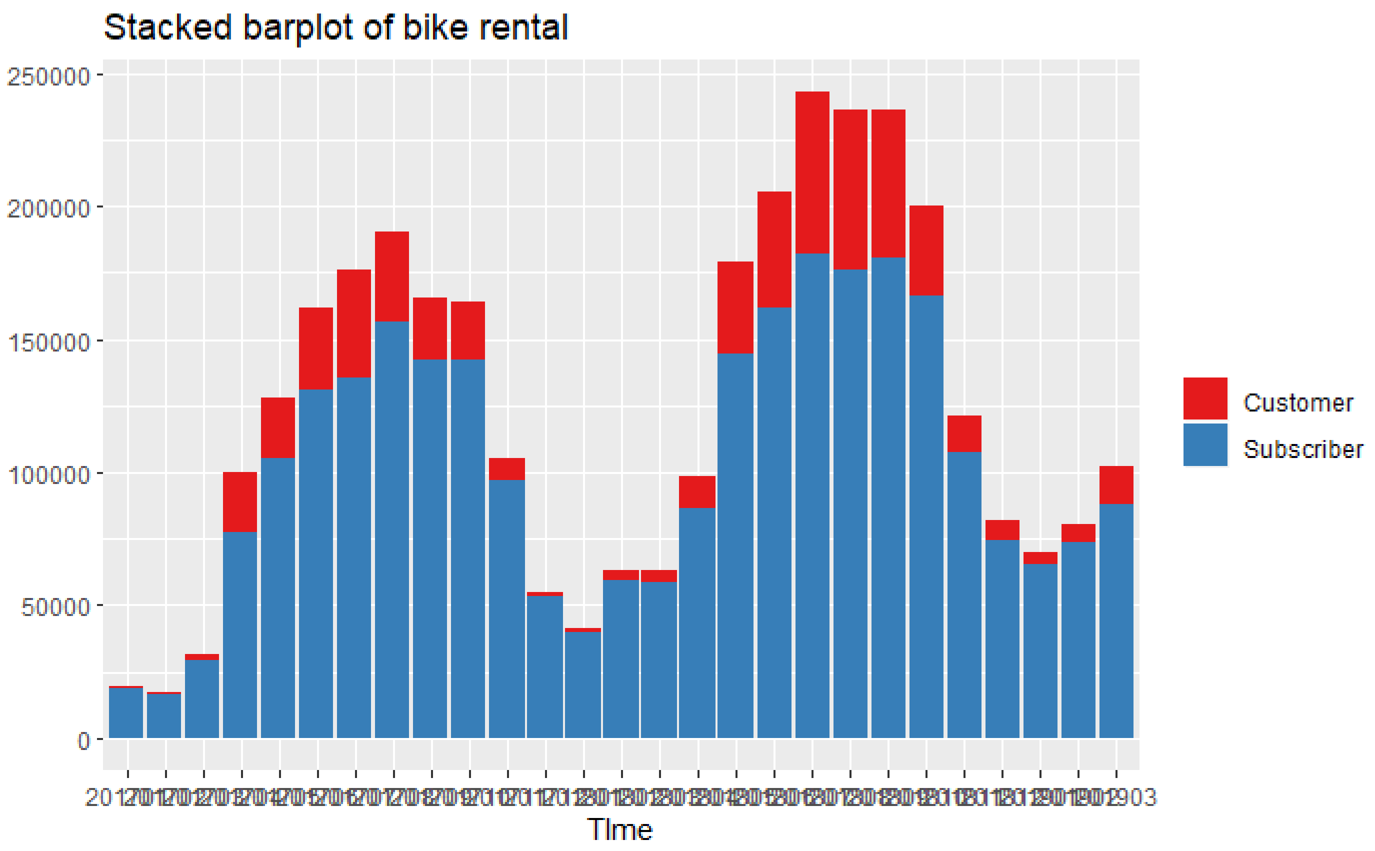
```
## busiest start and return bike station visulization and data analysis
#handling busystation start
busystation$newtime<-format(round(as.POSIXct(busystation$starttime, format="%Y-%m-%d %H:%M:%S"),
units="hours"))
busystation$date=substr(busystation$newtime,1,10)
busystation$hour<-substr(busystation$newtime,12,13)
busystation<-busystation[,-1]
busystation$year<-substr(busystation$newtime,1,4)
busystation$month<-substr(busystation$newtime,6,7)
busystation<-busystation[!busystation$newtime=="2019-04-01 00:00:00",]
```

Hide

```
yearmonth1<-busystation%>%
        filter(usertype=="Subscriber") %>%
        group_by(year,month) %>%
        summarize(total=n())
yearmonth1$type<-"Subscriber"
yearmonth2<-busystation%>%
        filter(usertype=="Customer") %>%
        group_by(year,month) %>%
        summarize(total=n())
yearmonth2$type<-"Customer"
yearmonth1<-rbind(yearmonth1,yearmonth2)
```

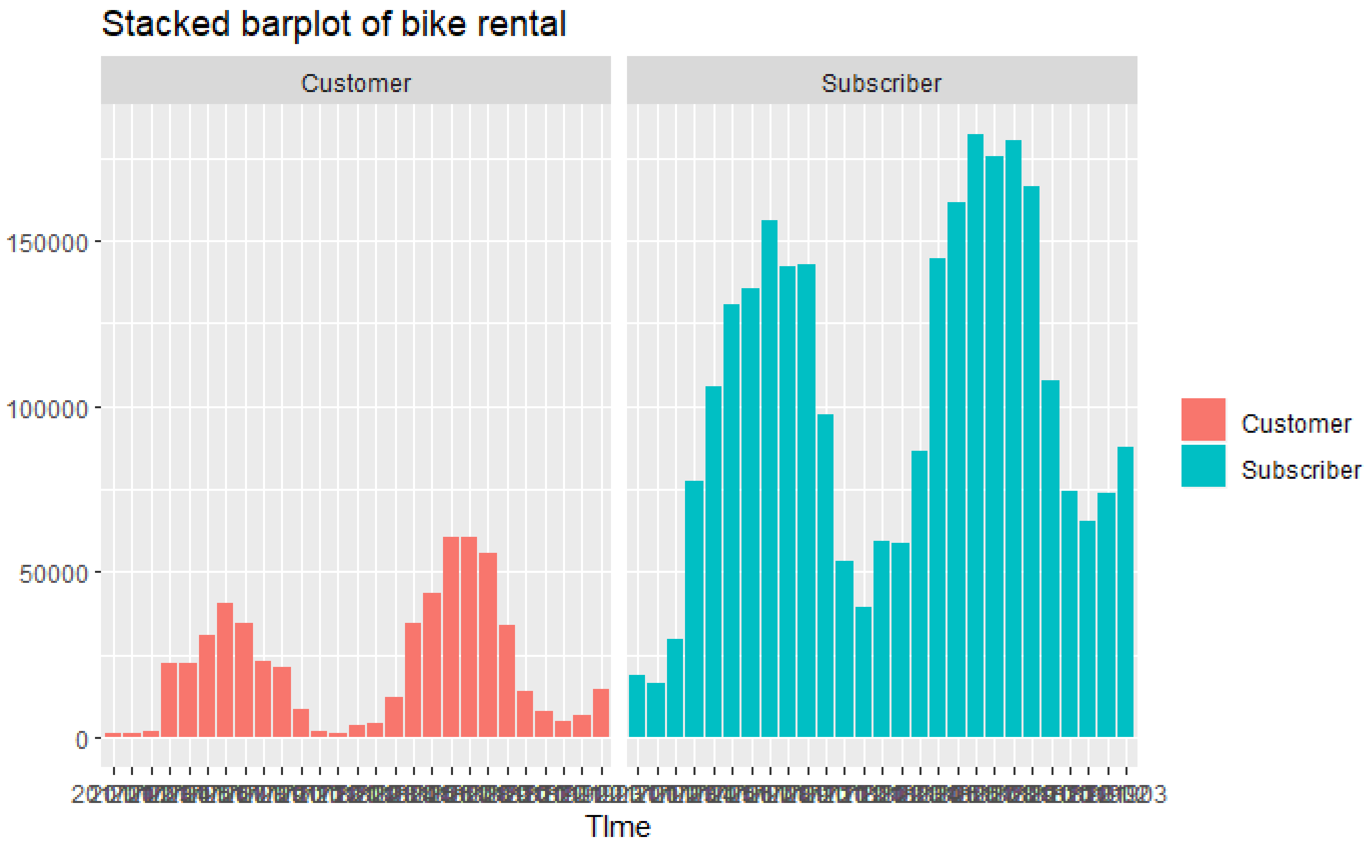
Hide

```
#stacked barchart
ggplot(yearmonth1, aes(fill=type, y=total, x=paste(year,month,sep=''))) +
        geom_bar( stat="identity") +
        scale_fill_brewer(palette = "Set1") +
        labs(x = "TIme", y = NULL, fill = NULL ,title = "Stacked barplot of bike rental")
```



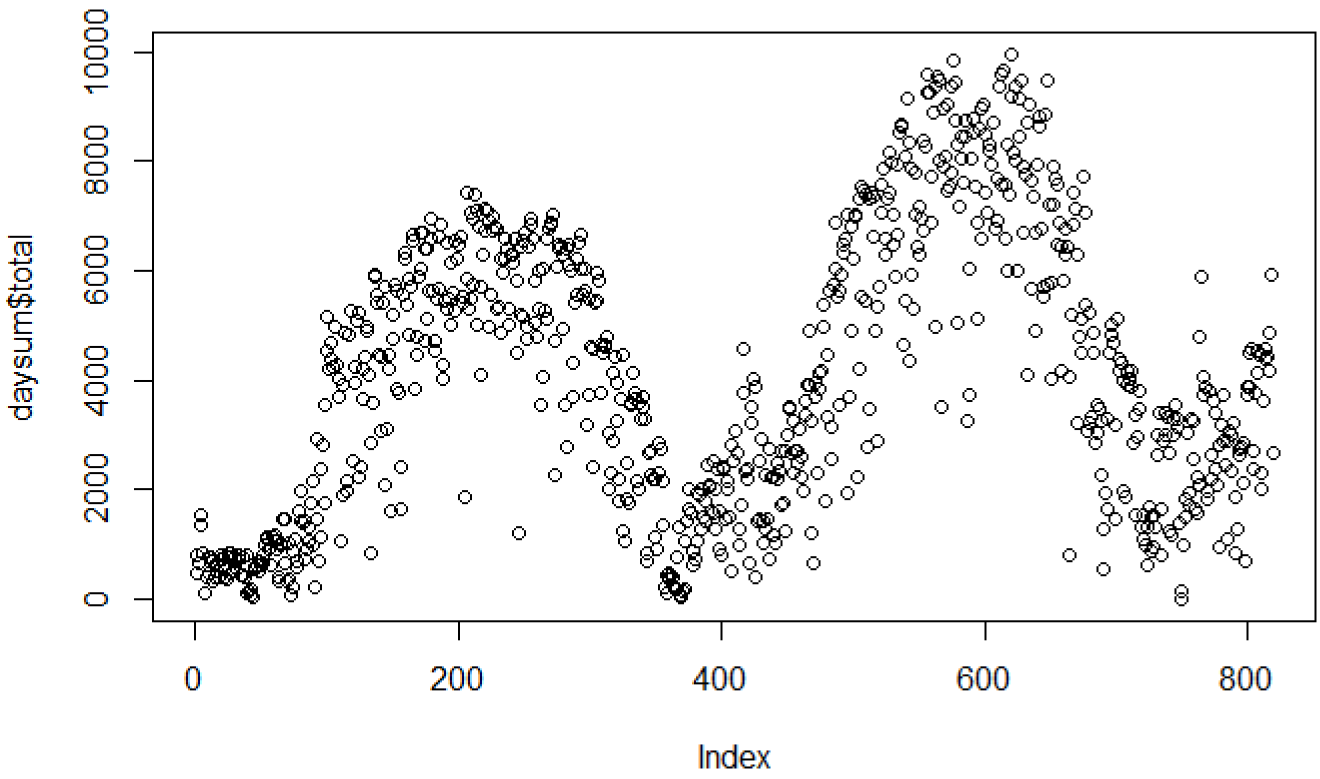
Hide

```
# Faceting
ggplot(yearmonth1, aes(fill=type, y=total, x=paste(year,month,sep=''))) +
        geom_bar( stat="identity") +
        facet_wrap(~type)+
        labs(x = "TIme", y = NULL, fill = NULL ,title = "Stacked barplot of bike rental")
```



Hide

```
daysum<-busystation %>%
        group_by(date) %>%
        summarise(total=n())
plot(daysum$total)
```

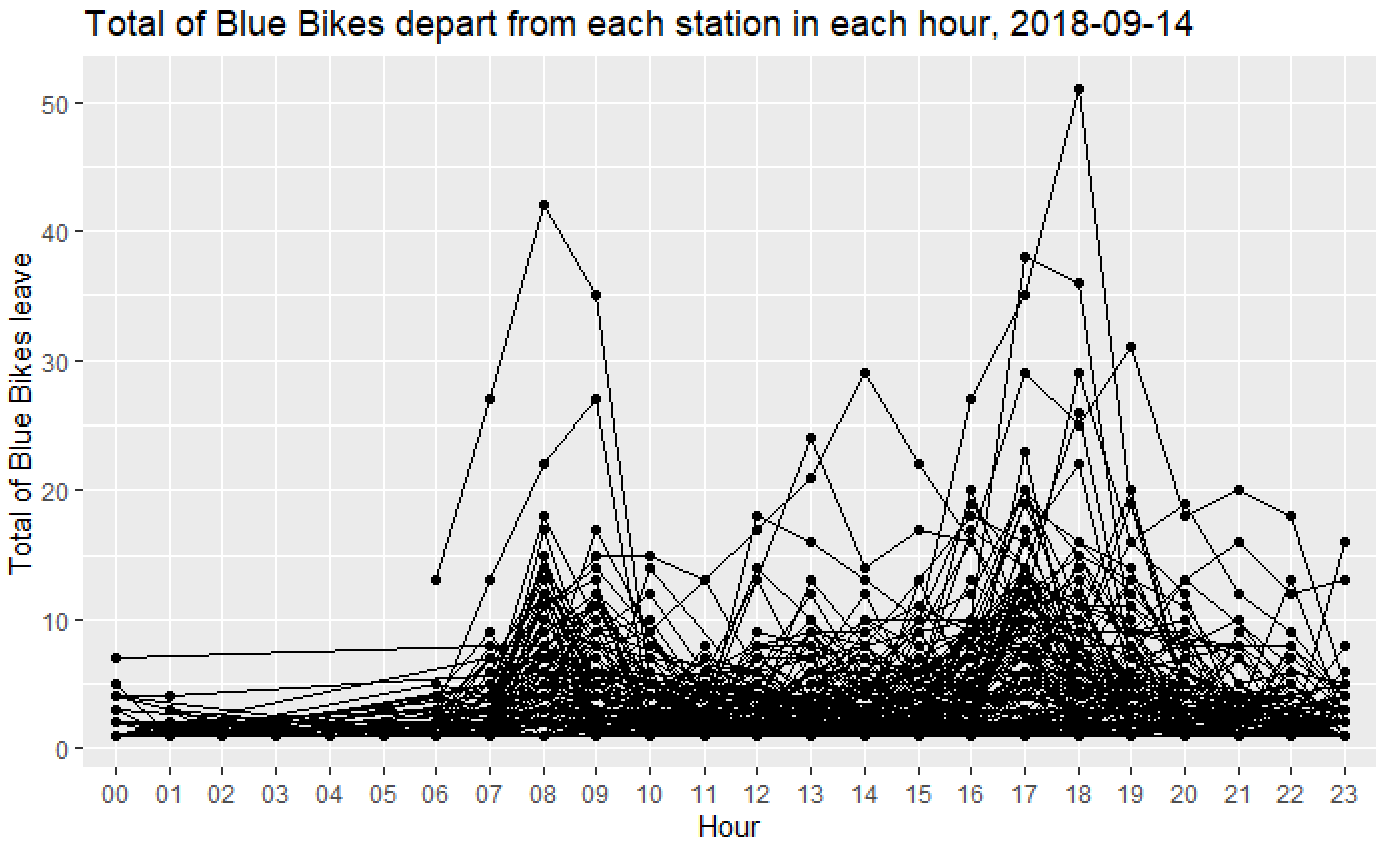


Hide

```
daysum<-daysum[order(-daysum$total,daysum$date), ]
```

Hide

```
busiestday<-busystation %>%
        filter(date==unlist(daysum[1,1])) %>%
        group_by(hour,`start station id`,`start station latitude`,`start station longitude`) %>%
        summarize(total=n())
busiestday %>% ggplot(aes(x = hour, y = total,group=`start station id`)) +
        geom_line() +
        geom_point() +
        labs(title = "Total of Blue Bikes depart from each station in each hour, 2018-09-14",
             x = "Hour",  y = "Total of Blue Bikes leave")
```



Hide

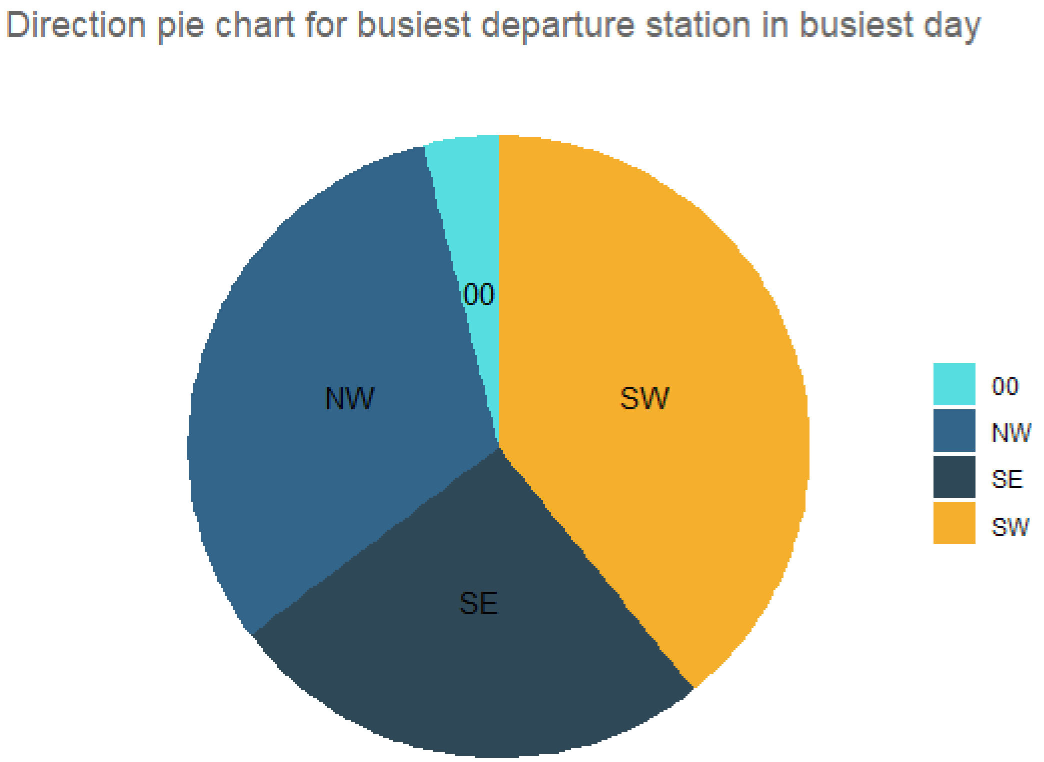
```
sort1.start<-busiestday[order(-busiestday$total,busiestday$hour),]
```

Hide

```
#count each direction number
direction1<-busystation %>%
        filter(date==unlist(daysum[1,1])&hour==unlist(sort1.start$hour[1])
               &`start station id`==unlist(sort1.start$`start station id`[1])) %>%
        group_by(direction_NS,direction_EW) %>%
        summarize(total=n())
#count each direction rental time
usetime1<-busystation %>%
        filter(date==unlist(daysum[1,1])&hour==unlist(sort1.start$hour[1])
               &`start station id`==unlist(sort1.start$`start station id`[1])) %>%
        group_by(direction_NS,direction_EW) %>%
        summarize(mean_time = mean(duration_min, na.rm = TRUE))
direction1$meantime<-usetime1$mean_time
direction1$to_dire<-paste0(direction1$direction_NS,direction1$direction_EW)
direction1<-direction1[,c(-1,-2)]
```

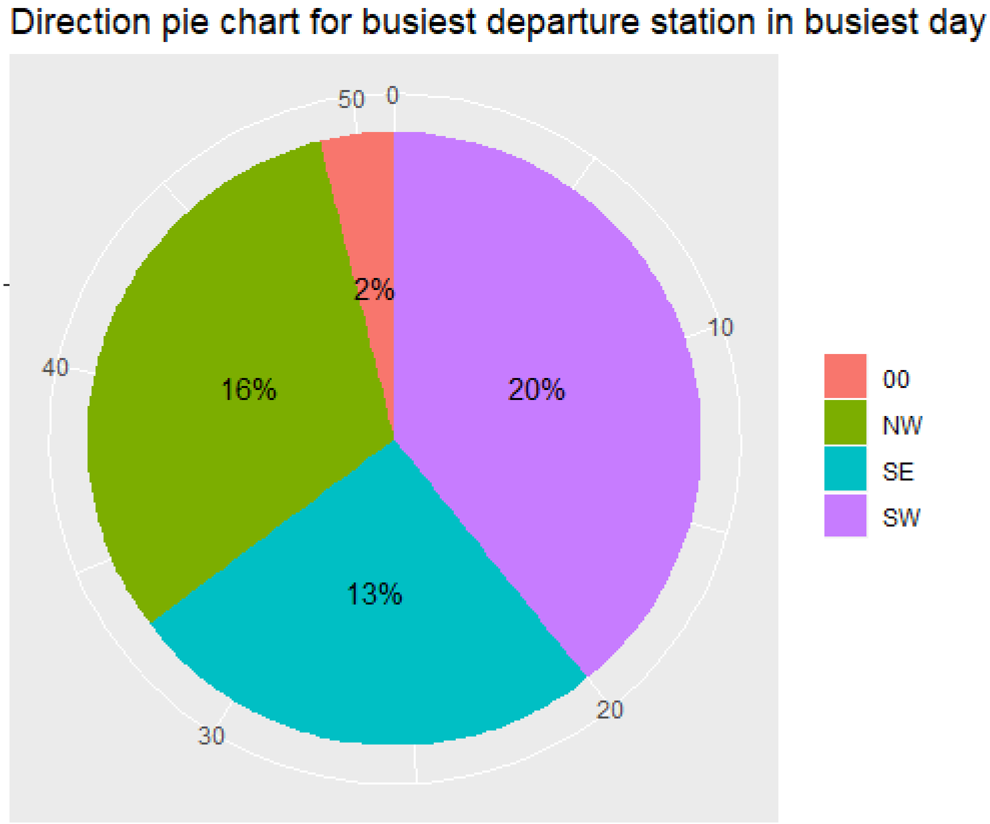
Hide

```
# Create a basic bar
ggplot(direction1, aes(x="", y=total, fill=to_dire)) +
        geom_bar(stat="identity", width=1)+
        coord_polar("y", start=0) +
        geom_text(aes(label = to_dire), position = position_stack(vjust = 0.5))+
        scale_fill_manual(values=c("#55DDE0", "#33658A", "#2F4858", "#F6AE2D", "#F26419", "#9999
99"))+
        labs(x = NULL, y = NULL, fill = NULL, title = "Direction pie chart for busiest departure
station in busiest day ")+
        theme_classic() + theme(axis.line = element_blank(),
                                axis.text = element_blank(),
                                axis.ticks = element_blank(),
                                plot.title = element_text(hjust = 0.5, color = "#666666"))
```



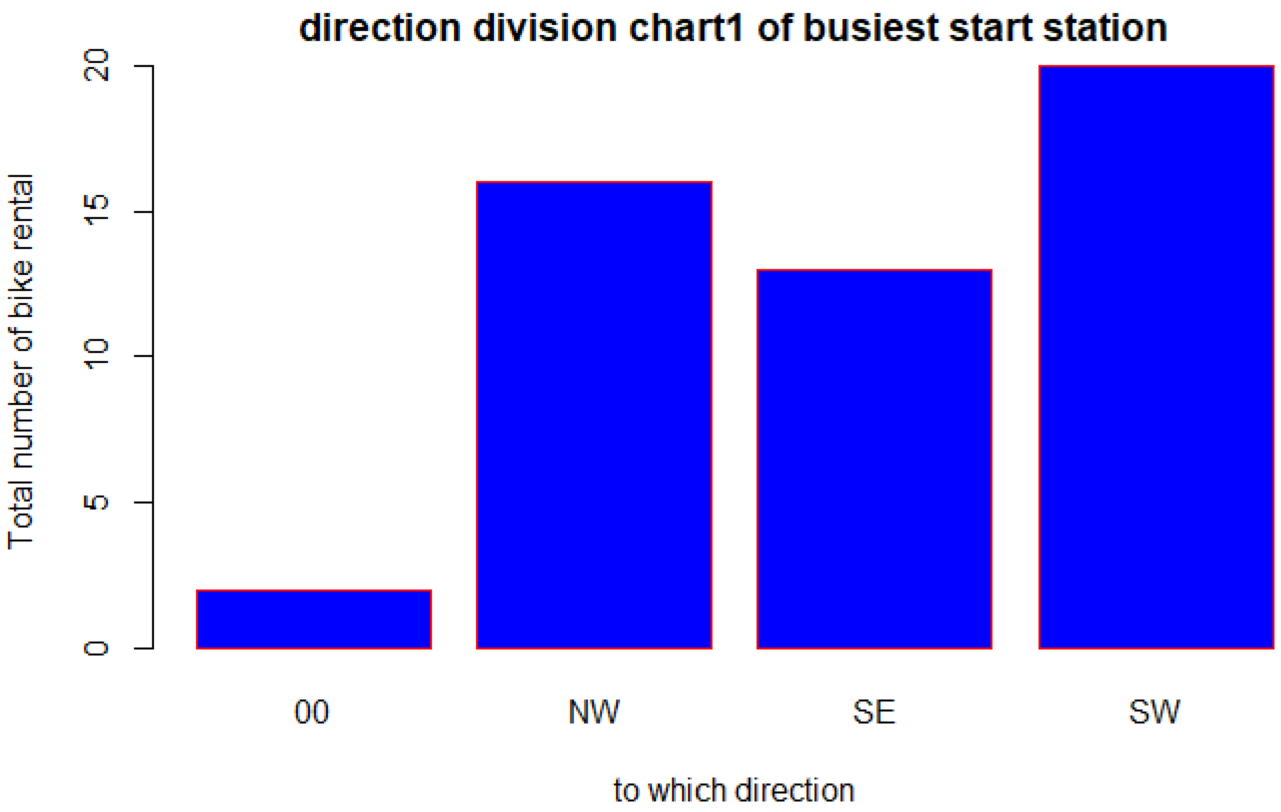
Hide

```
ggplot(direction1, aes(x="", y=total, fill=to_dire)) +
        geom_bar(stat="identity", width=1) +
        coord_polar("y", start=0) +
        geom_text(aes(label = paste0(total, "%")), position = position_stack(vjust = 0.5)) +
        labs(x = NULL, y = NULL, fill = NULL ,title = "Direction pie chart for busiest departure
station in busiest day")
```



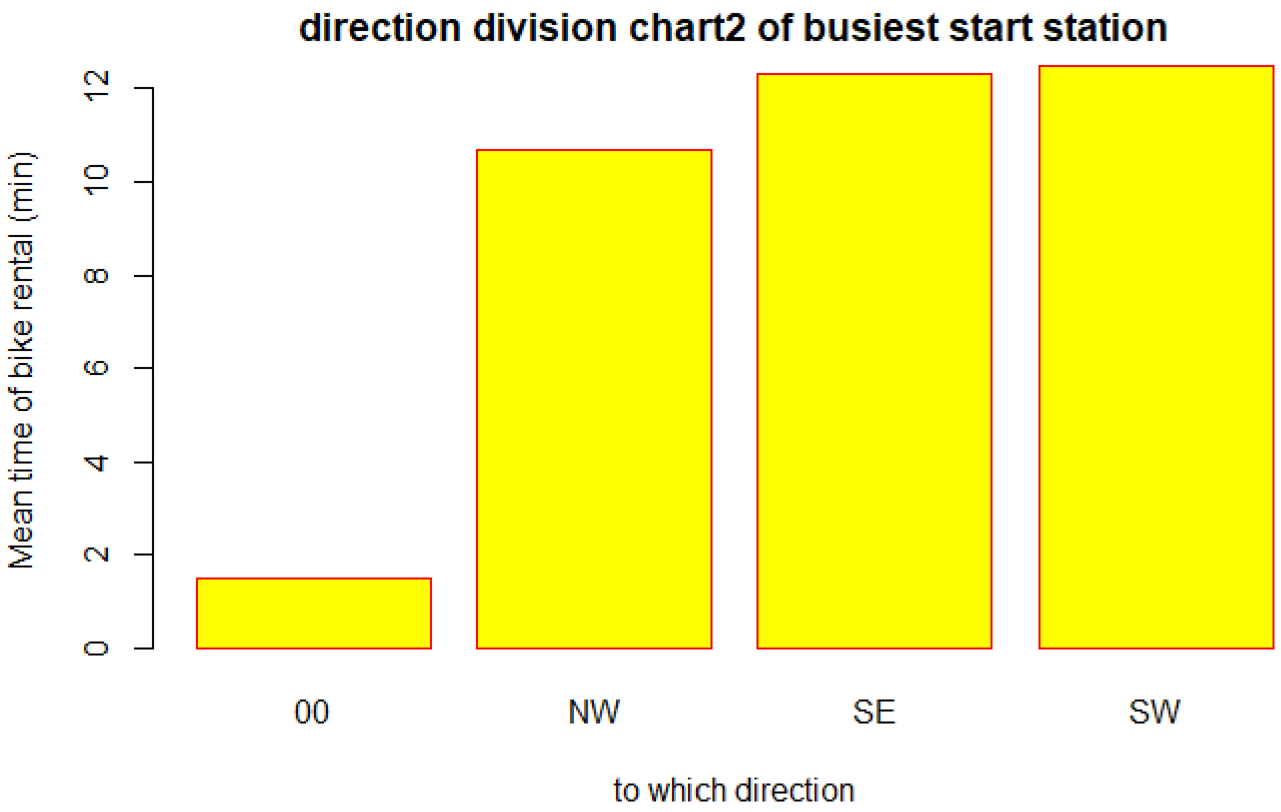
Hide

```
barplot(direction1$total,names.arg=direction1$to_dire,xlab="to which direction",
        ylab="Total number of bike rental",col="blue",
        main="direction division chart1 of busiest start station",border="red")
```


direction division chart1 of busiest start station

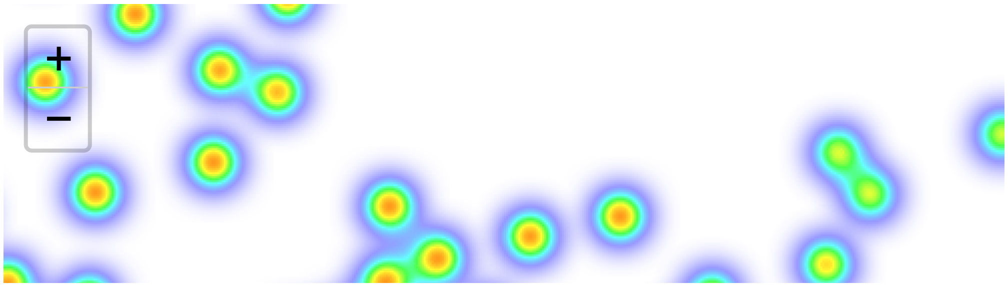
Hide

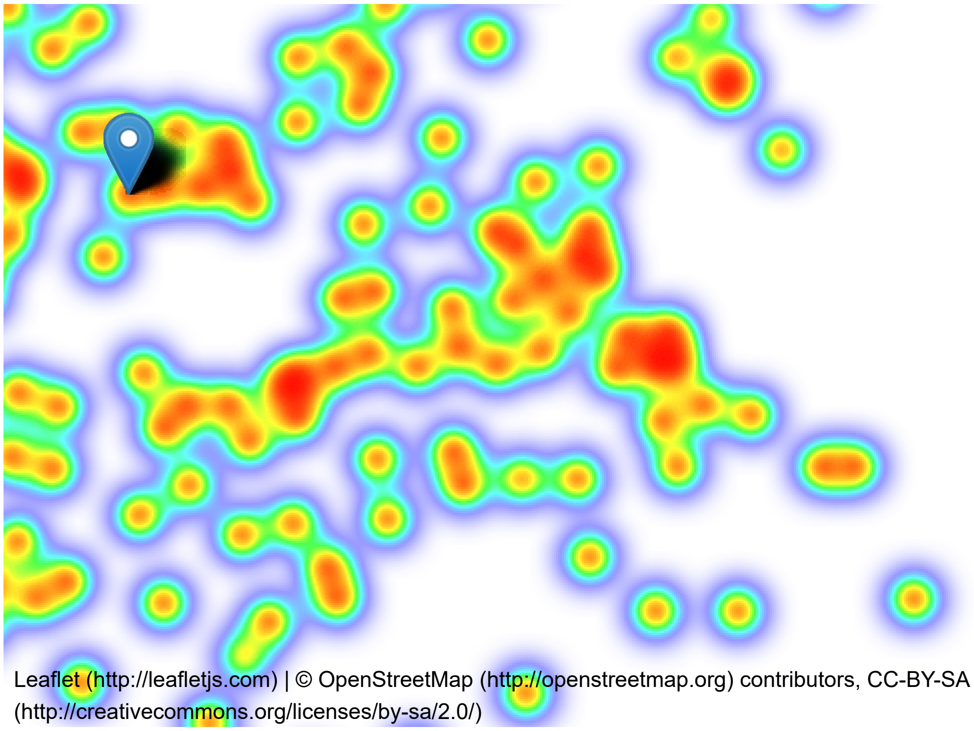
```
barplot(direction1$meantime,names.arg=direction1$to_dire,xlab="to which direction",
        ylab="Mean time of bike rental (min)",col="yellow",
        main="direction division chart2 of busiest start station",border="red")
```


direction division chart2 of busiest start station

Hide

```
library(leaflet) # interactive mapping
library(leaflet.extras) #extra mapping for leaflet
# 2018-09-14 18:00, how busy the stations are to start bike rental
busiestday[busiestday$hour==unlist(sort1.start[1,1]),] %>%
        leaflet() %>%
        setView(lng = -71.0589, lat = 42.3601, zoom = 13) %>%
        addTiles( ) %>%
        addHeatmap(lng = busiestday$`start station longitude`, lat = busiestday$`start station l
atitude`,
                   max = 2, radius = 15) %>%
        addMarkers(lng = busiestday$`start station longitude`[busiestday$`start station id`
                                                              == unlist(sort1.start$`start stati
on id`[1])],
                   lat = busiestday$`start station latitude`[busiestday$`start station id`
                                                             == unlist(sort1.start$`start statio
n id`[1])],
                   popup =busiestday$`start station id`[ busiestday$`start station id`
                                                         ==unlist(sort1.start$`start station id`
[1])])
```





Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)

```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.
```