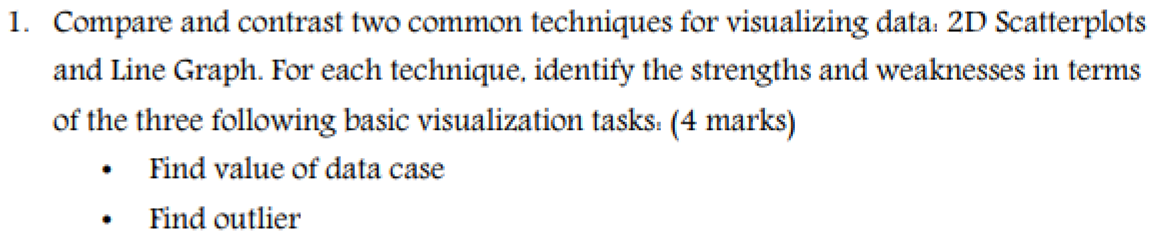
CSE 3020
Data Visualization
Digital Assignment 1

NITIN RANJAN, 18BCE0272

SLOT F2



1. Compare and contrast two common techniques for visualizing data: 2D Scatterplots and Line Graph. For each technique, identify the strengths and weaknesses in terms of the three following basic visualization tasks: (4 marks)
   - Find value of data case
   - Find outlier

Solution:

2D scatterplots and Line plots are probably the easiest way of visualizing data that compares two varying quantities. However, probably the most important difference between both is what an observer can infer from the visualizations. While Scatterplots are more qualitative and demonstrate the pattern / trend or spread of data, line graphs are very quantitative and each point along a variable can be traced to the other variable through a roughly approximated equation.

Now, based on the expected tasks:

1. To find value of data case:

2D Scatter-plots –

The visualization will depend on the density of the plot. If the data points stick very closely with each other and there are a minimal number of stray data points, then it becomes easy to approximate a regression function which can then be used for finding the value of a Data case. Thus, a good goodness of fit in the model can minimize most of the data approximation issues.

However, a data that is more spread can create a very high percentage of error in this approach. So, in this case, the best way is to ascertain the intercepts on both axes if perpendiculars are dropped from that point.

Thus, in 2D scatter plots, while it is quite easy in most cases to ascertain the type of regression the variables follows, it can be a really tedious task to approximate the value of a data case.

Line Graphs –

The approximation of value of a data case is very easy in the case of line graphs. Line graphs represent a set of data points expressed as a mathematical function. The line yields a slope expressing the relationship between the points mathematically and hence it is quite easy as compared to scatterplot where visualization is more important than rigorous mathematics.

However, in a plane where data is more spread, line graph fails too as it is a rigorous mathematical equation and as the points move away from the line, the error increases.

2. Find outlier

2D Scatter-plots –

Finding outlier is easy in scatter plots because this process is fairly observation based. Data points outside a ‘cluster’ or a group of points can easily be detected. However, in a closely knit cluster, i.e. in a case where the data points are quite close to each other, it is natural to have the regression model set along a specific line or curve, and as points move away from the curve, the error increases.

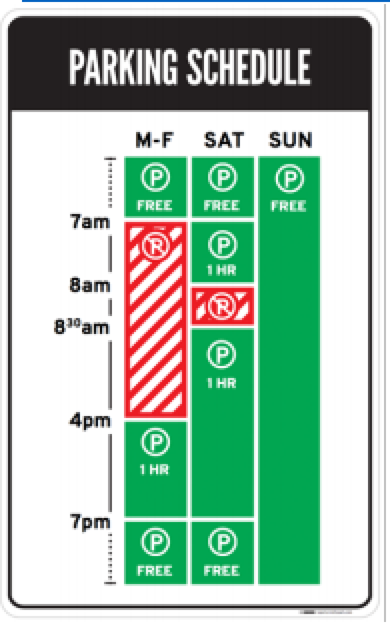
However, in scatter plots, outlier is quite clearly a very distant point and the regression curve is assumed to represent all the close by points quite accurately.

Line Graphs –

In line graphs, an outlier has to be a really isolated point on the plane. This is because line graphs quite naturally assume that most points lie on the line itself and are usually not used to visualize a plane where the data points are scattered. So, finding an outlier in visualizations that actually fit with a line graph is very easy, easier than 2D scatterplots.

However, if on a common scattered dataset, it is easier to point out to an outlier in a scatterplot.

2. Deconstruct, Reconstruct (6 marks)
    - Reference: https://rpubs.com/poojasuresh/531223



- Identify the major issues on the above design (2 marks)
- Create a better visualisation design for the above diagram (4 marks)

suc

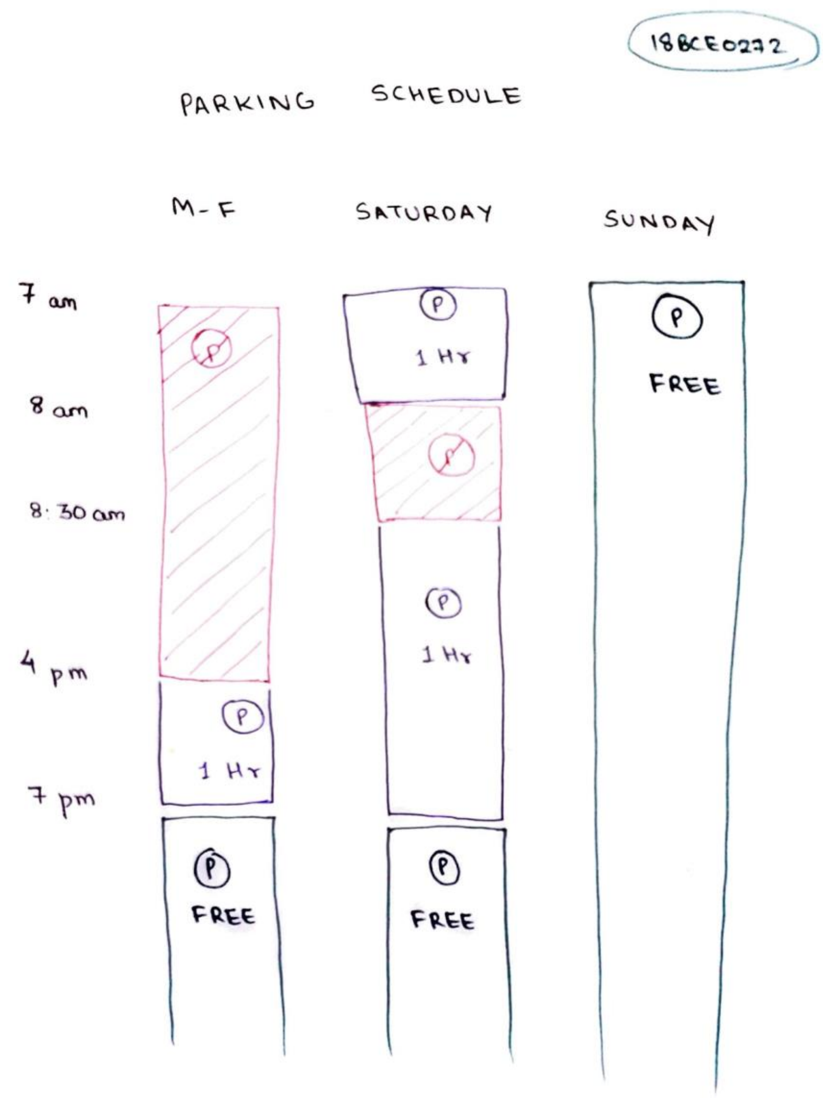
The major issues of the given design are:

1. Not using sufficient types of colours:
   The no Parking are all demonstrated in red stripes. However, the part of the schedule where parking is allowed is of again two types – free hours and hour specific slots. The free hours and hour specific slots have been assigned the same colours in the design. This creates a design that is less informative to the user.
2. Improper segmentation of information being conveyed :
   The free slots extend from 7pm to 7am on all weekdays. There was no necessity to create two segments to illustrate this information. A single slot 7pm – 7 am is sufficient and the day can start with 7am onwards. This would prevent an unnecessary assumption that the observer must understand the cyclic time frame and presents information as a linear chin which is easier to understand.

18BCE0272
NITIN PRAMOD RANJAN

RECONSTRUCTION

A better visualization design will look like:



18BCE0272
NITIN PRAMOD RANJAN

I have implemented the design using a R heatmap. A heatmap should give a very clear estimate of the design. And each colour on the map represents a particular value of the design – no parking, 1 hr parking, or free parking.

The new graph eliminates the design issues by creating a mapping where each category of data is represented as a unique colour with as much linearity as possible.

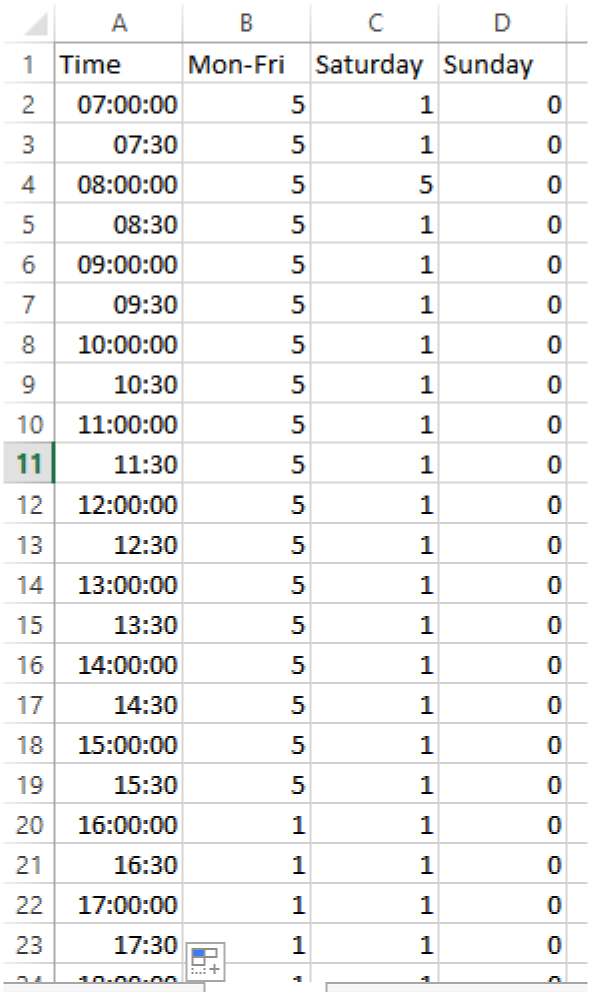
Reconstruction using R

Step 1: create a csv file

#0 is free parking

#1 is 1 hour parking

#5 is no parking to depict the idea of infinity



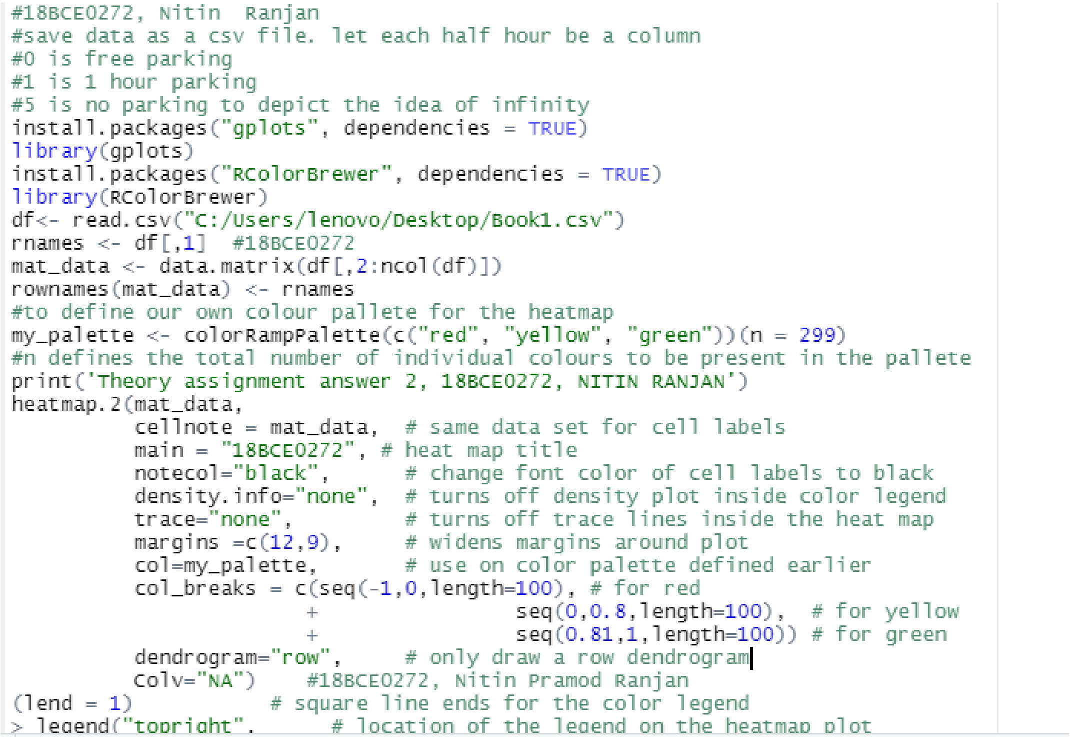
	A	B	C	D
1	Time	Mon-Fri	Saturday	Sunday
2	07:00:00	5	1	0
3	07:30	5	1	0
4	08:00:00	5	5	0
5	08:30	5	1	0
6	09:00:00	5	1	0
7	09:30	5	1	0
8	10:00:00	5	1	0
9	10:30	5	1	0
10	11:00:00	5	1	0
11	11:30	5	1	0
12	12:00:00	5	1	0
13	12:30	5	1	0
14	13:00:00	5	1	0
15	13:30	5	1	0
16	14:00:00	5	1	0
17	14:30	5	1	0
18	15:00:00	5	1	0
19	15:30	5	1	0
20	16:00:00	1	1	0
21	16:30	1	1	0
22	17:00:00	1	1	0
23	17:30	1	1	0

Step 2 : Code in r -

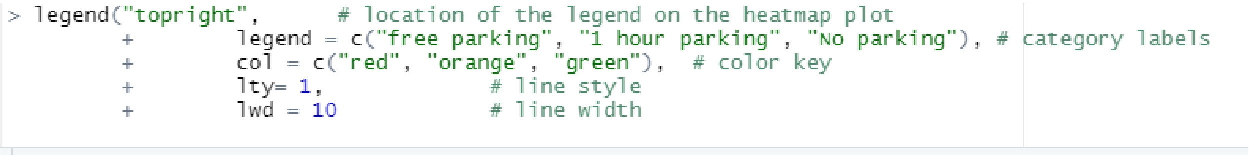
```
#18BCE0272, Nitin  Ranjan
#0 is free parking
#1 is 1 hour parking
#5 is no parking to depict the idea of infinity
#save data as a csv file. let each half hour be a column
install.packages("gplots", dependencies = TRUE)
library(gplots)
install.packages("RColorBrewer", dependencies = TRUE)
library(RColorBrewer)
df<- read.csv("C:/Users/lenovo/Desktop/Book1.csv")
rnames <- df[,1]  #18BCE0272
mat_data <- data.matrix(df[,2:ncol(df)])
rownames(mat_data) <- rnames
#to define our own colour pallete for the heatmap
my_palette <- colorRampPalette(c("red", "yellow", "green"))(n = 299)
#n defines the total number of individual colours to be present in the pallete
print('Theory assignment answer 2, 18BCE0272, NITIN RANJAN')
heatmap.2(mat_data,
        cellnote = mat_data,  # same data set for cell labels
        main = "18BCE0272", # heat map title
        notecol="black",      # change font color of cell labels to black
        density.info="none",  # turns off density plot inside color legend
        trace="none",         # turns off trace lines inside the heat map
        margins =c(12,9),     # widens margins around plot
        col=my_palette,       # use on color palette defined earlier
        col_breaks = c(seq(-1,0,length=100), # for red
                +           seq(0,0.8,length=100),  # for yellow

                +           seq(0.81,1,length=100)) # for green

      dendrogram="row",     # only draw a row dendrogram

      Colv="NA")    #18BCE0272, Nitin Pramod Ranjan

(lend = 1)          # square line ends for the color legend

> legend("topright",      # location of the legend on the heatmap plot

      +       legend = c("free parking", "1 hour parking", "No parking"), # category labels

      +       col = c("red", "orange", "green"),  # color key

      +       lty= 1,            # line style

      +       lwd = 10           # line width

#18BCE0272
```

18BCE0272
NITIN PRAMOD RANJAN

Snapshot:

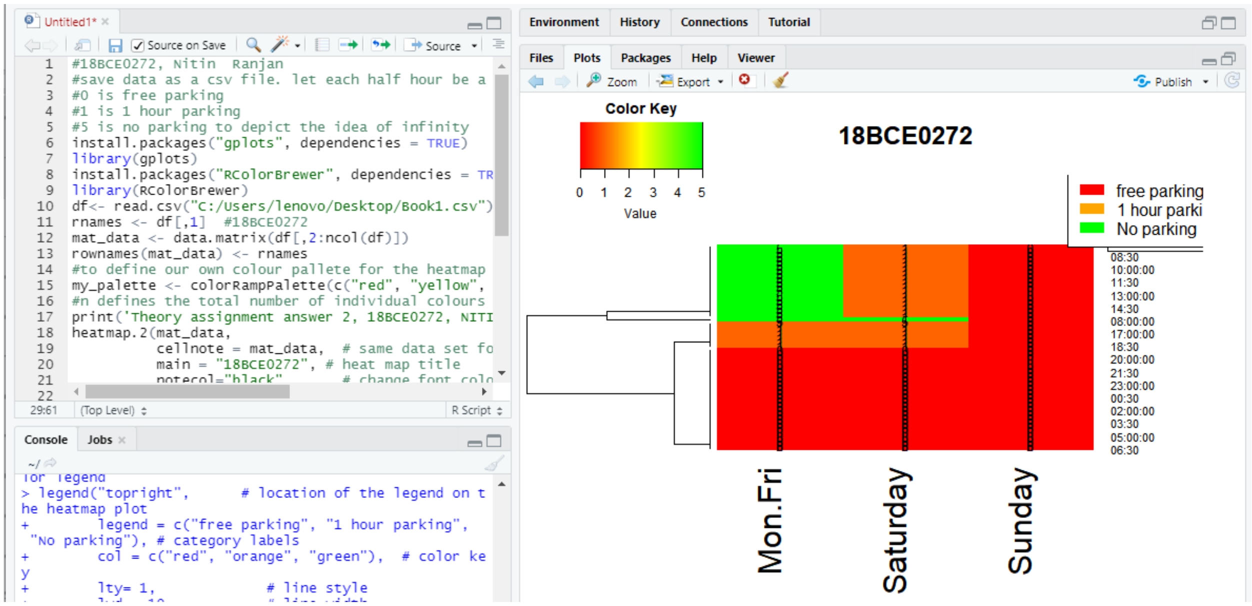


```
#18BCE0272, Nitin  Ranjan
#save data as a csv file. let each half hour be a column
#0 is free parking
#1 is 1 hour parking
#5 is no parking to depict the idea of infinity
install.packages("gplots", dependencies = TRUE)
library(gplots)
install.packages("RColorBrewer", dependencies = TRUE)
library(RColorBrewer)
df<- read.csv("C:/Users/lenovo/Desktop/Book1.csv")
rnames <- df[,1]  #18BCE0272
mat_data <- data.matrix(df[,2:ncol(df)])
rownames(mat_data) <- rnames
#to define our own colour pallete for the heatmap
my_palette <- colorRampPalette(c("red", "yellow", "green"))(n = 299)
#n defines the total number of individual colours to be present in the pallete
print('Theory assignment answer 2, 18BCE0272, NITIN RANJAN')
heatmap.2(mat_data,
          cellnote = mat_data,  # same data set for cell labels
          main = "18BCE0272", # heat map title
          notecol="black",      # change font color of cell labels to black
          density.info="none",  # turns off density plot inside color legend
          trace="none",         # turns off trace lines inside the heat map
          margins =c(12,9),     # widens margins around plot
          col=my_palette,       # use on color palette defined earlier
          col_breaks = c(seq(-1,0,length=100), # for red
                  +                 seq(0,0.8,length=100),  # for yellow
                  +                 seq(0.81,1,length=100)) # for green
          dendrogram="row",     # only draw a row dendrogram
          Colv="NA")    #18BCE0272, Nitin Pramod Ranjan
(lend = 1)             # square line ends for the color legend
> legend("topright",      # location of the legend on the heatmap plot
```



```
> legend("topright",      # location of the legend on the heatmap plot
         +        legend = c("free parking", "1 hour parking", "No parking"), # category labels
         +        col = c("red", "orange", "green"),  # color key
         +        lty= 1,             # line style
         +        lwd = 10            # line width
```

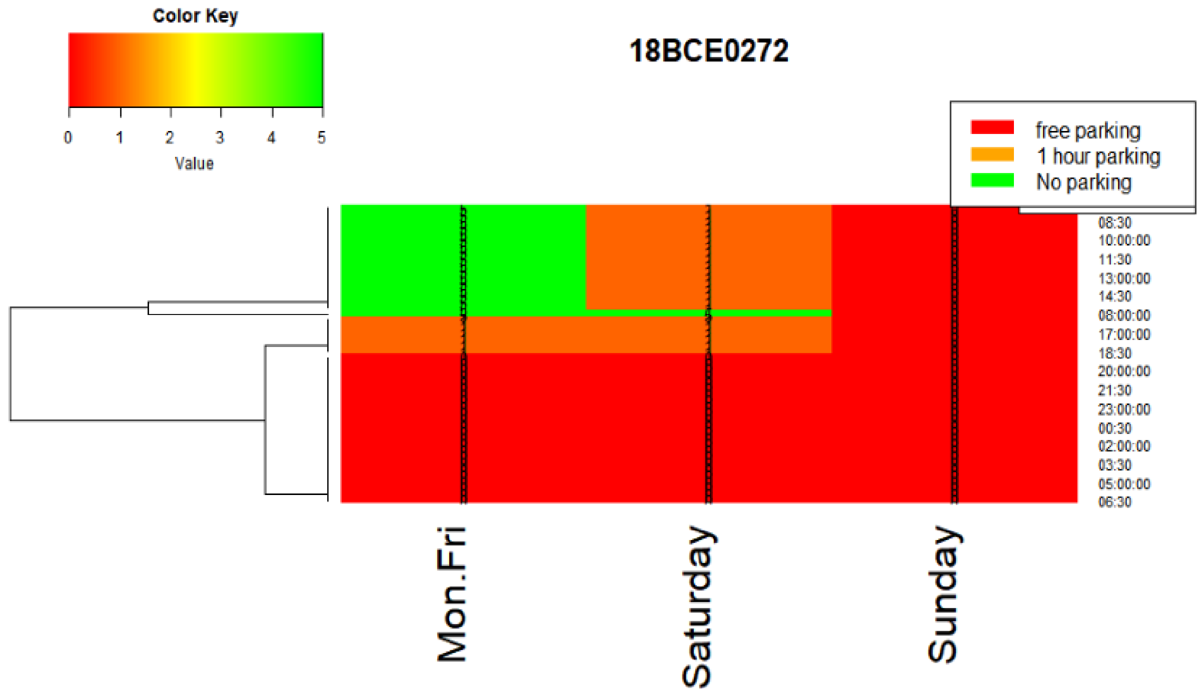
18BCE0272
NITIN PRAMOD RANJAN



```
#18BCE0272, Nitin  Ranjan
#save data as a csv file. let each half hour be a
#0 is free parking
#1 is 1 hour parking
#5 is no parking to depict the idea of infinity
install.packages("gplots", dependencies = TRUE)
library(gplots)
install.packages("RColorBrewer", dependencies = TR
library(RColorBrewer)
df<- read.csv("C:/Users/lenovo/Desktop/Book1.csv")
rnames <- df[,1]  #18BCE0272
mat_data <- data.matrix(df[,2:ncol(df)])
rownames(mat_data) <- rnames
#to define our own colour pallete for the heatmap
my_palette <- colorRampPalette(c("red", "yellow",
#n defines the total number of individual colours
print('Theory assignment answer 2, 18BCE0272, NITI
heatmap.2(mat_data,
          cellnote = mat_data,  # same data set fo
          main = "18BCE0272", # heat map title
          notecol="black"       # change font colo
```

```
for legend
> legend("topright",       # location of the legend on t
he heatmap plot
+         legend = c("free parking", "1 hour parking",
 "No parking"), # category labels
+         col = c("red", "orange", "green"),  # color ke
y
+         lty= 1,             # line style
```

OUTPUT:



Color Key
Value
18BCE0272
free parking
1 hour parking
No parking
08:30
10:00:00
11:30
13:00:00
14:30
08:00:00
17:00:00
18:30
20:00:00
21:30
23:00:00
00:30
02:00:00
03:30
05:00:00
06:30
Mon.Fri
Saturday
Sunday