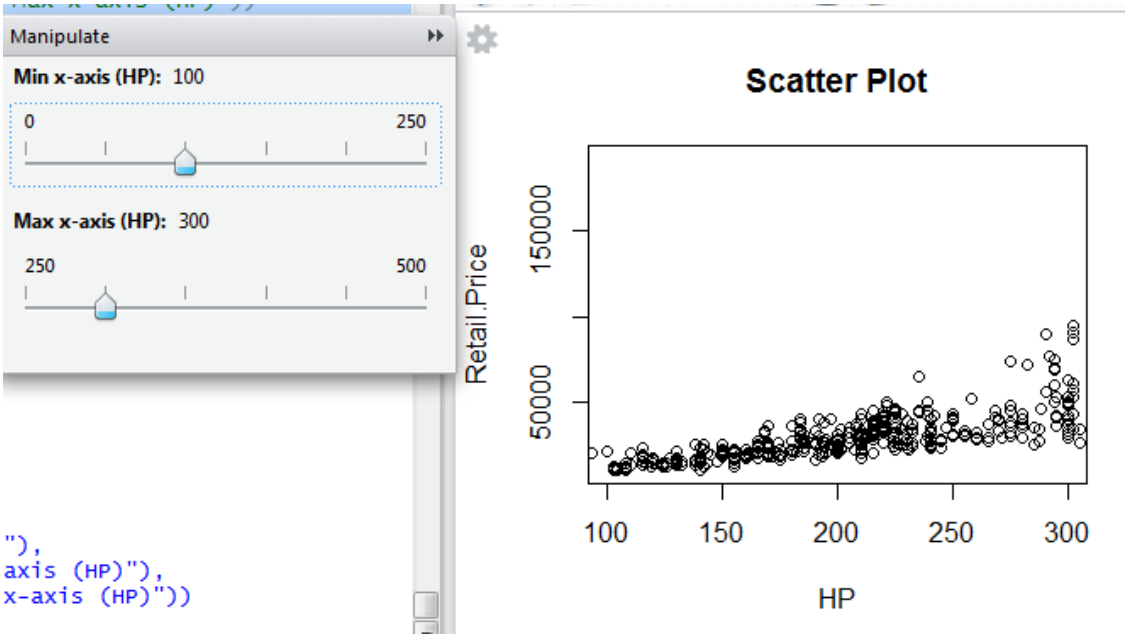
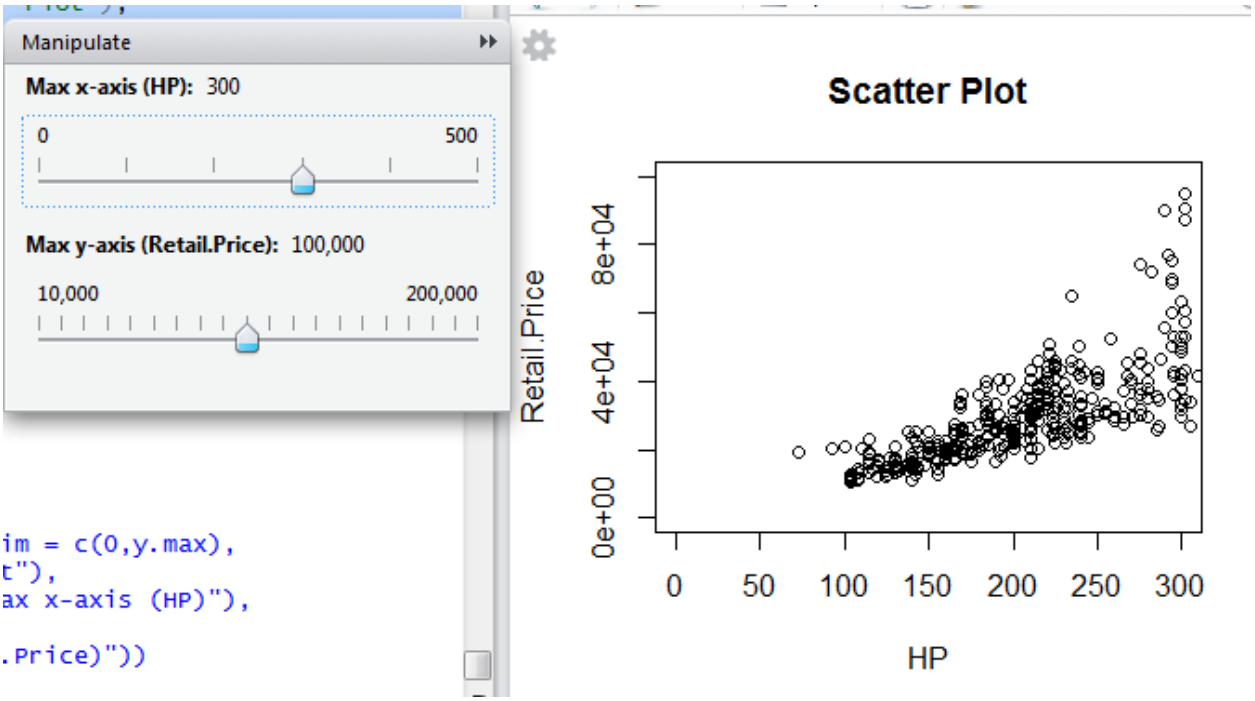
Lab 3
Steven Lin

Exercise 1

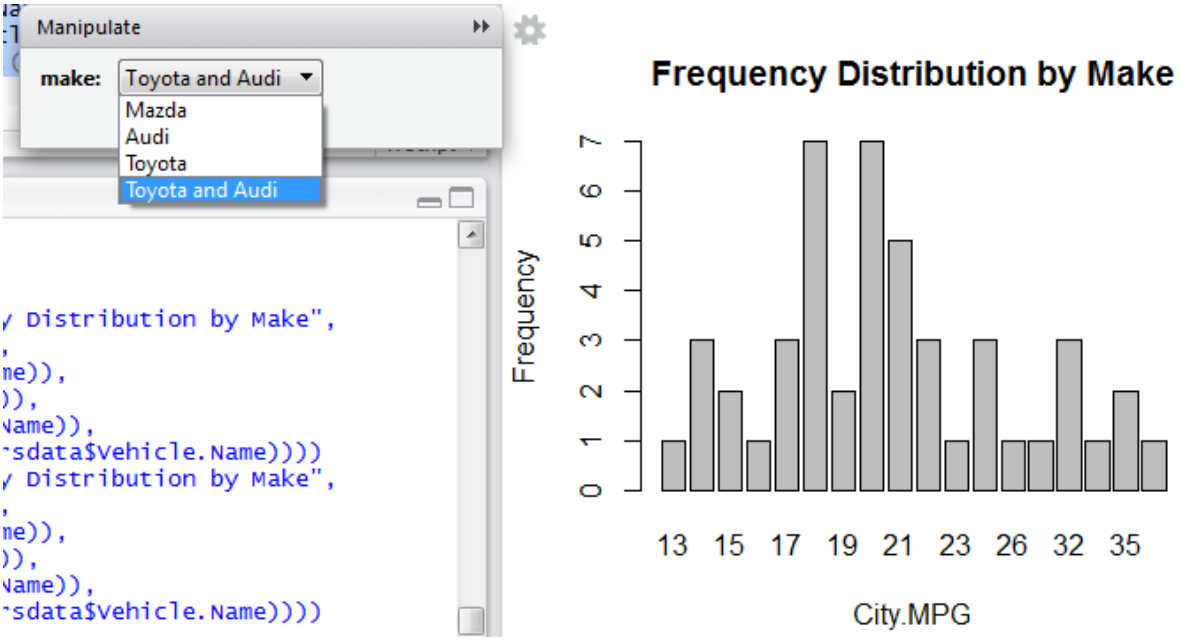
a)



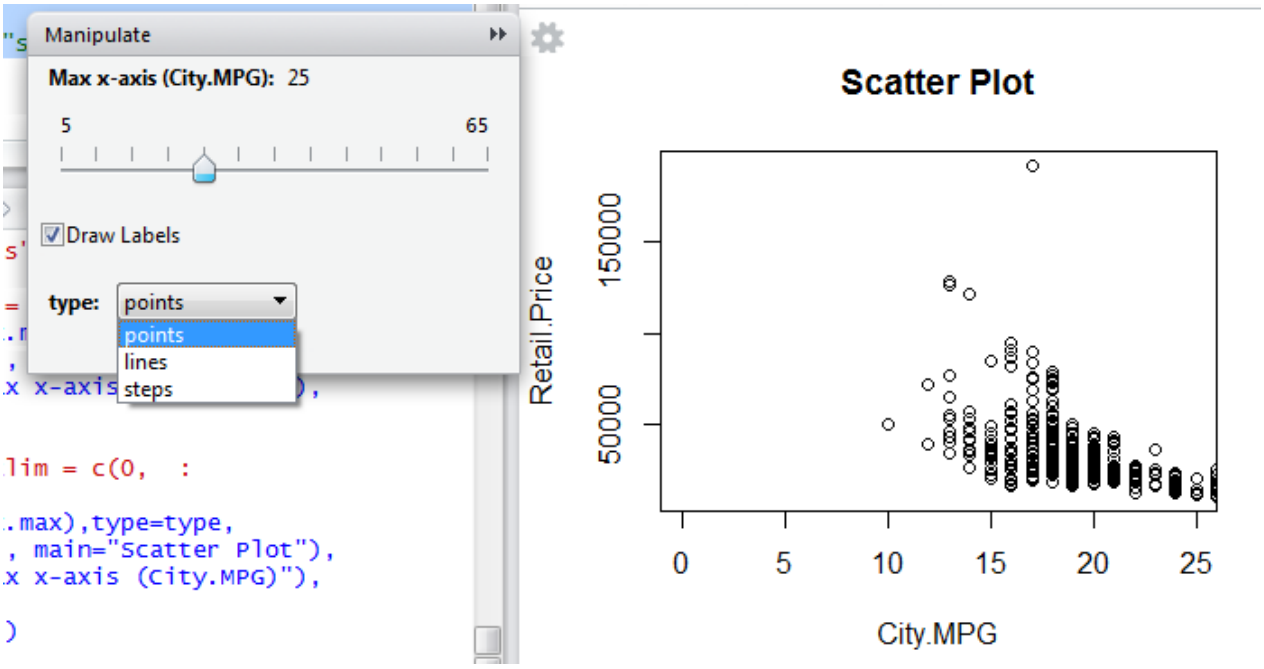
b)



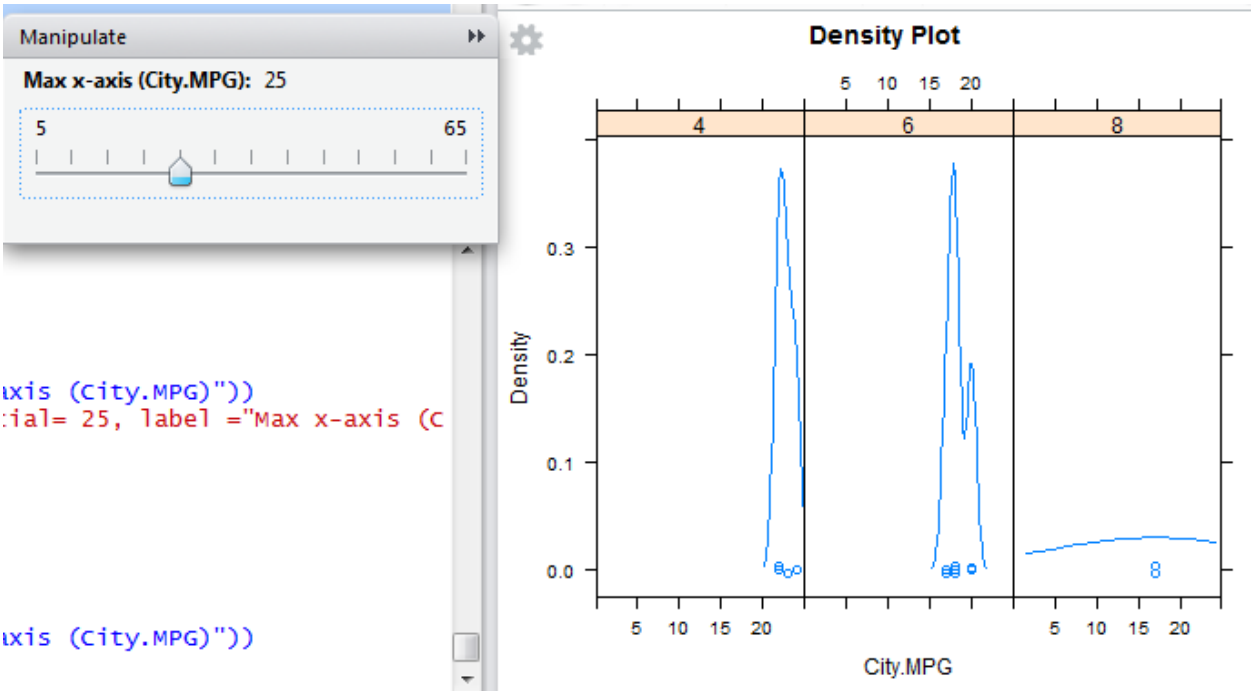
c)


Manipulate
make:
Toyota and Audi
Mazda
Audi
Toyota
Toyota and Audi
Frequency Distribution by Make
Frequency
0 1 2 3 4 5 6 7
13 15 17 19 21 23 26 32 35
City.MPG

d)

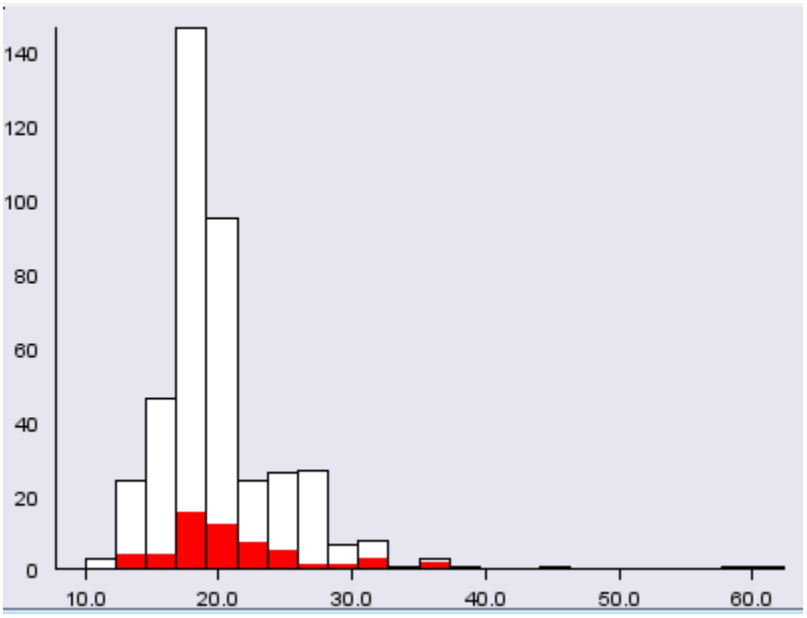

Manipulate
Max x-axis (City.MPG): 25
5
65
Draw Labels
type:
points
points
lines
steps
Scatter Plot
Retail.Price
50000
150000
0 5 10 15 20 25
City.MPG

e)



Exercise 2

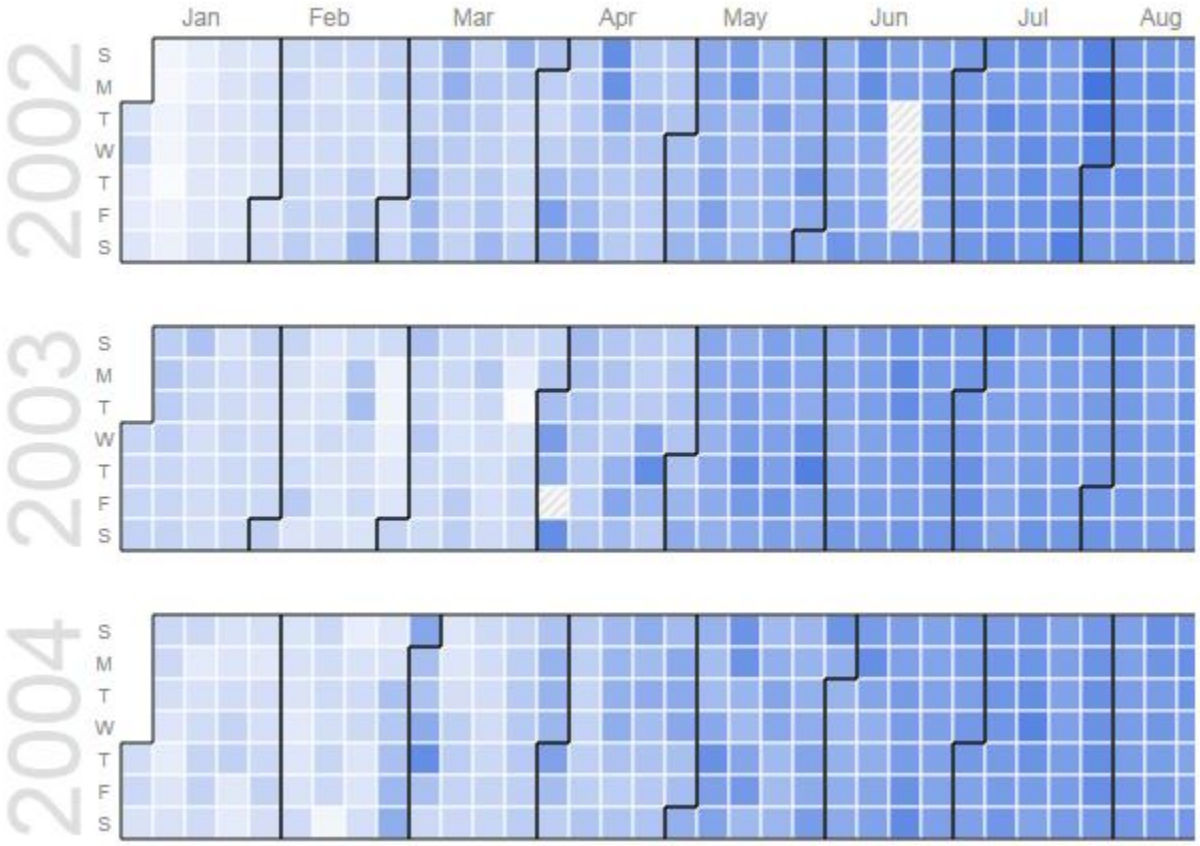
a) proportion of the dataset selected = 13.5514 %



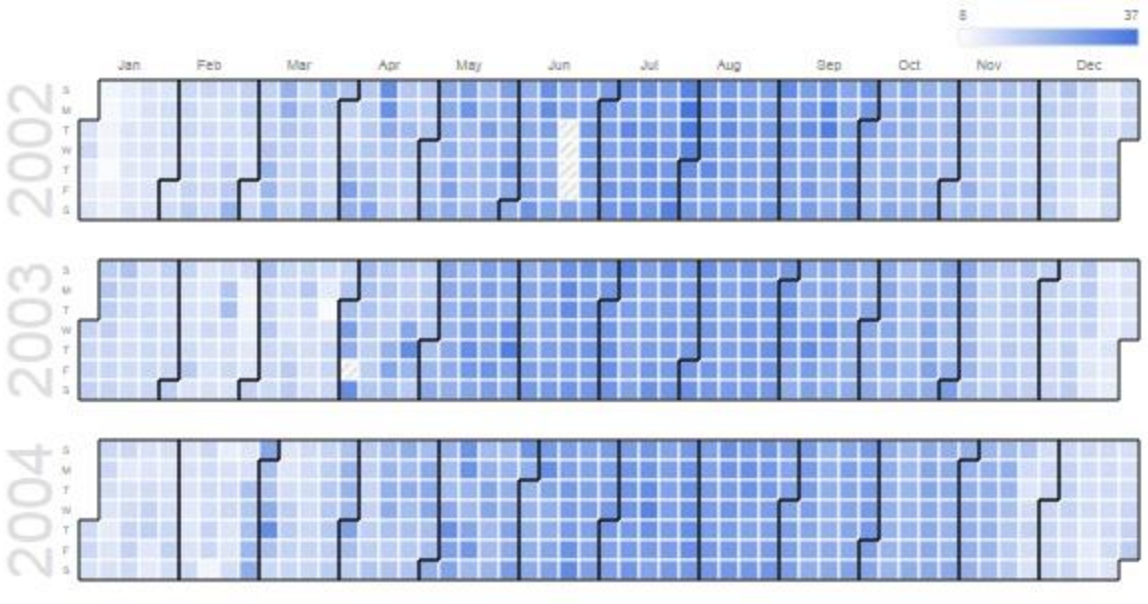
Exercise 3

a)

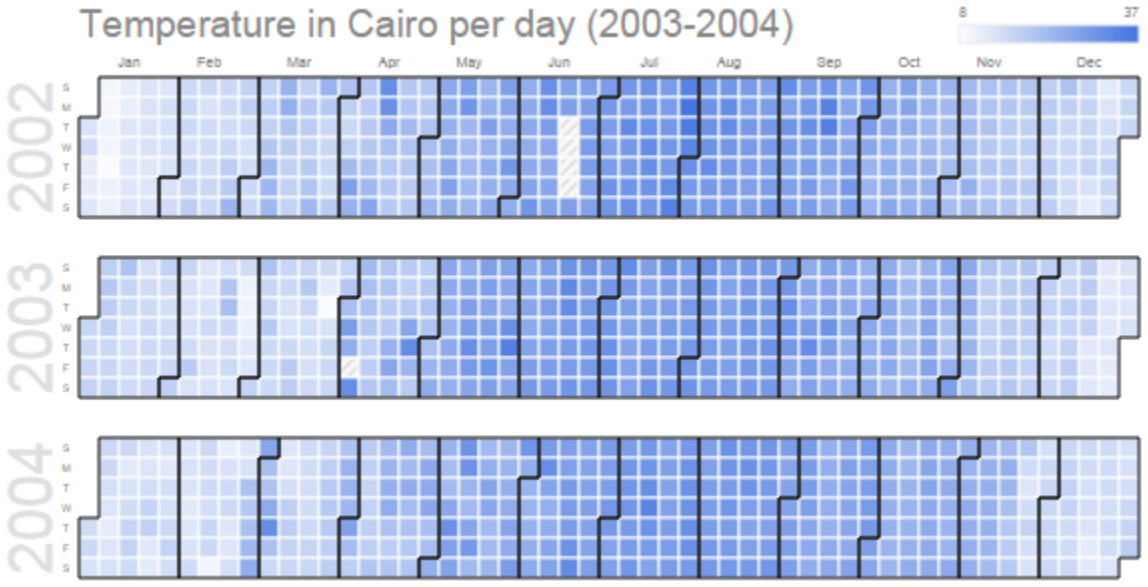
i)



ii)

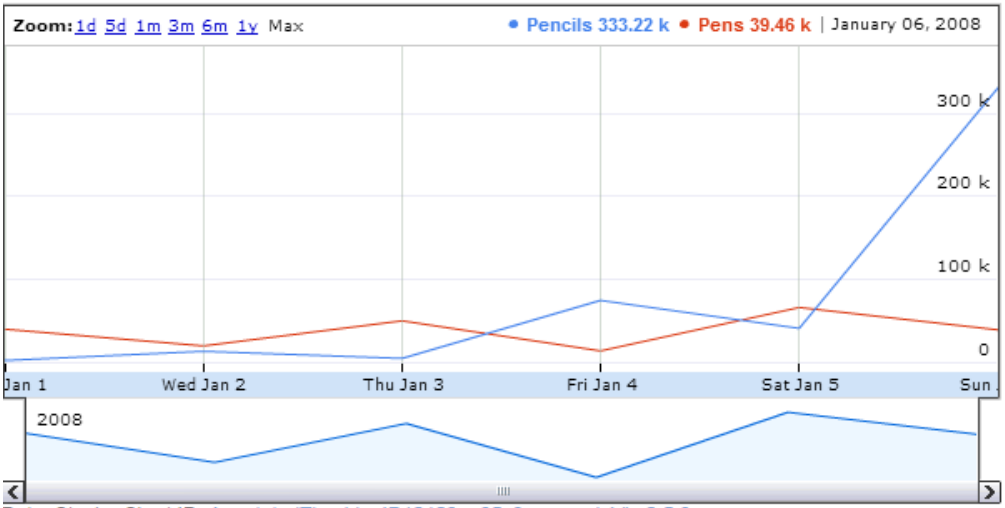

8
37
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2002
2003
2004

iii)


Temperature in Cairo per day (2003-2004)
8
37
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2002
2003
2004

b)

i)



Data: Stock • Chart ID: AnnotatedTimeLineID12103ee85c9c • googleVis-0.5.8
R version 3.1.3 (2015-03-09) • Google Terms of Use • Documentation and Data Policy

ii)

Data: Stock • Chart ID: AnnotatedTimeLineID12103a4b6b3b • googleVis-0.5.8
R version 3.1.3 (2015-03-09) • Google Terms of Use • Documentation and Data Policy

iii)

Data: Stock • Chart ID: AnnotatedTimeLineID12101551a50 • googleVis-0.5.8
R version 3.1.3 (2015-03-09) • Google Terms of Use • Documentation and Data Policy

Code

```
a# Lab session 3 exercise

##### Load data ######################################################################

# My PC
main = "C:/Users/Steven/Documents/Academics/3_Graduate School/2014-2015 ~ NU/"

# Aginity
#main = "\\\\nas1/labuser169"

course = "MSIA_411_Data_Visualization"
datafolder = "/Lab/Data"
setwd(file.path(main,course, datafolder))

carsdata =read.csv("04cars data.csv",header=TRUE,
                   na.strings=c("","*","NA"))


##### Ex1  ###########################################################
# Part a ####
# Create a scatterplot for horsepower ("HP") against retail price
# ("Retail.Price") - horsepower on the x-axis and retail price on the
# y-axis. Add two sliders for the axis: "x.min" between 0 and 250,
# and "x.max" between 250 and 500

library(manipulate)

#slider & different options
manipulate(plot(carsdata$HP,carsdata$Retail.Price,xlim=c(x.min,x.max),
                xlab = "HP", ylab = "Retail.Price", main="Scatter Plot"),
           x.min = slider(0,250,step=50,initial = 100, label = "Min x-axis (HP)"),
           x.max = slider(250,500,step=50, initial =300, label = "Max x-axis (HP)"))

# Part b ####
# For the same scatterplot as in part (a) create one slider for the
# x-axis between 0 and 500, and one slider for the y-axis between
# 10,000 and 200,000.

manipulate(plot(carsdata$HP,carsdata$Retail.Price,xlim=c(0,x.max), ylim = c(0,y.max),
                xlab = "HP", ylab = "Retail.Price", main="Scatter Plot"),
           x.max = slider(0,500, step= 100, initial = 300, label = "Max x-
axis (HP)"),
           y.max = slider(10000,200000, step = 10000,
                          initial=100000, label = "Max y-axis (Retail.Price)"))

# Part c ####
# Create a barplot for "City.MPG" with a picker for the following car
# makes/combinations of car makes: Mazda, Audi, Toyota, Toyota and Audi.
carsdata$make = "Other"

makeList = c("Mazda", "Audi", "Toyota")
for (mfg in makeList){
  index = grep(mfg,(carsdata$Vehicle.Name))
  carsdata[index,"make"] = mfg

}

```

```
#picker with groups
manipulate(barplot(table(carsdata[make,"City.MPG"]),main="Barplot by Make",
                   xlab="City.MPG", ylab="Frequency", beside=TRUE),
           make = picker("Mazda"= grep("Mazda",(carsdata$Vehicle.Name)),
                         "Audi"=  grep("Audi",(carsdata$Vehicle.Name)),
                         "Toyota"=grep("Toyota",(carsdata$Vehicle.Name)),
                         "Toyota and Audi"= grep("Toyota|Audi",(carsdata$Vehicle.Name))
))

# Part d ####
# Create a scatterplot of "City.MPG" versus "Retail.Price", which will
# contain the following: sliderfor the x-axis between 5 and 65, by increments
# of 5 and initial value 25; picker with points, line
# and step (as shown in class); checkbox with "draw labels"

#ann: a logical value indicating whether the default annotation (title and x and y axis
 labels) should appear on the plot.

manipulate(plot(carsdata$City.MPG,carsdata$Retail.Price,xlim=c(0,x.max),type=type,
                xlab = "City.MPG", ylab = "Retail.Price",ann=label, main="Scatter Plot"
),
           x.max = slider(5,65, step= 5, initial = 25, label = "Max x-
axis (City.MPG)"),
           label = checkbox(T, "Draw Labels"),
           type = picker("points"="p", "lines" = "l", "steps"="s"))

# Part e ####

# For this part you will combine the manipulate package with the lattice package.
# Select the first 20 rows of the "04cars data" set and, using the lattice package,
# create density plots for "City.MPG" for cars with 4, 6 and 8 cylinders.
# You must include a slider for the x-axis.

sample = carsdata[1:20,]
library(lattice)

manipulate(densityplot(~City.MPG|factor(Cyl),
                       main="Density Plot",
                       xlab="City.MPG",
                       xlim=c(0,x.max),
                       data = sample,
                       layout = c(3,1)),
           x.max = slider(5,65,step =5, initial= 25, label ="Max x-axis (City.MPG)"))

##### Ex2  ################################################################

      # Part a ####
      # Create a histogram for "City.MPG". Create a selection with all Toyota, Mazda a
nd Audi
      # vehicles in the histogram (part of the histogram should turn red after you cre
ate the
      # selection). What proportion of the dataset was selected?

      library(iplots)
      library(JGR)

      carsdata=read.csv("04cars data.csv",
                        header=TRUE,na.strings=c("","*","NA"))

      attach(carsdata)

```

```
# histograms
ihist(City.MPG)

#interactive selection:
#get indices
iset.selected()

#select cases from here:
index = grep(("Toyota|Mazda|Audi"),Vehicle.Name)
iset.select(index)

# proportion of the dataset selected
length(index)/dim(carsdata)[1]


detach(carsdata)

##### Ex3  ##################################################################

# Part a ####
library(googleVis)
demo(WorldBank)

data(Cairo)

# i)
# Inspect the Cairo data set and fill in the the gaps ("") in the
# command above appropriately (leave the options list empty for the
# time being). Create a google calendar plot

calendar1 = gvisCalendar(data=Cairo, datevar = "Date",
                         numvar = "Temp", options = list())
plot(calendar1)

# ii)
# You will notice that not all of the months are shown in the plot
# of part (i) (can see up to half of August).
# Find a way to make all the months show up (including the legend).
# Hint: you need to adjust the size of the figure.

calendar2 = gvisCalendar(data=Cairo, datevar = "Date",
                         numvar = "Temp",
                         options = list(calendar="{ cellSize: 10 }"))
plot(calendar2)

# iii)
# Add a title to your calendar chart. Also, change the font size
# of the years on the side of the calendar to 30.


calendar3 = gvisCalendar(data=Cairo, datevar = "Date",
                         numvar = "Temp",
                         options = list(title = "Temperature in Cairo per day (2003-2004)",
                                                  calendar="{cellSize: 10,
                                                            yearLabel: {fontSize: 30}}"))

plot(calendar3)


# Par b  ####
```

```
172.
173.        data(Stock)
174.        # i
175.        # Inspect the the Stock data set, fill in the gaps ("") in the command
176.        # above appropriately, and plot the time line. Leave the options list empty.
177.
178.
179.        TL1 = gvisAnnotatedTimeLine(data=Stock, datevar="Date",
180.                                    numvar="Value", idvar="Device",date.format = "%Y/%m/
    %d",
181.                                    titlevar="Title", annotationvar="Annotation",
182.                                    options=list())
183.        plot(TL1)
184.
185.        # ii
186.        # Find a way to make the annotations visible on the time line.
187.
188.        TL2 = gvisAnnotatedTimeLine(data=Stock, datevar="Date",
189.                                    numvar="Value", idvar="Device",date.format = "%Y/%m/
    %d",
190.                                    titlevar="Title", annotationvar="Annotation",
191.                                    options=list(displayAnnotations=TRUE))
192.        plot(TL2)
193.
194.
195.        # iii
196.        # Notice that there are two types of devices in the data set, namely pencils
197.        # and pens. Since their values differ quite a bit, create two y-axes on the
198.        # timeline - one for pencils, and one for pens.
199.
200.        TL3 = gvisAnnotatedTimeLine(data = Stock, datevar="Date",
201.                                    numvar="Value", idvar="Device",
202.                                    titlevar="Title", annotationvar="Annotation",
203.                                    options=list(displayAnnotations=TRUE,
204.                                                 scaleColumns='[0,1]',
205.                                                scaleType='allmaximized'))
206.        plot(TL3)
```