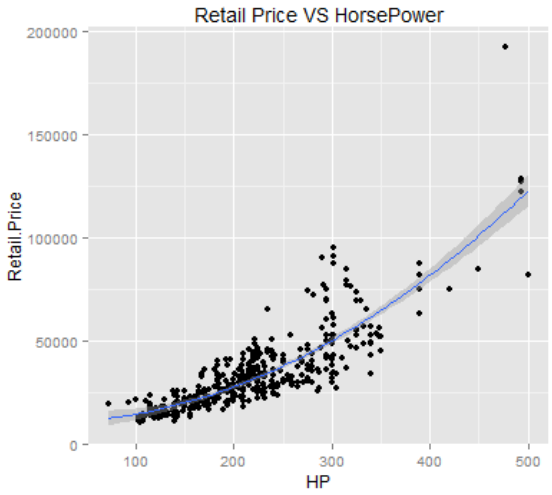
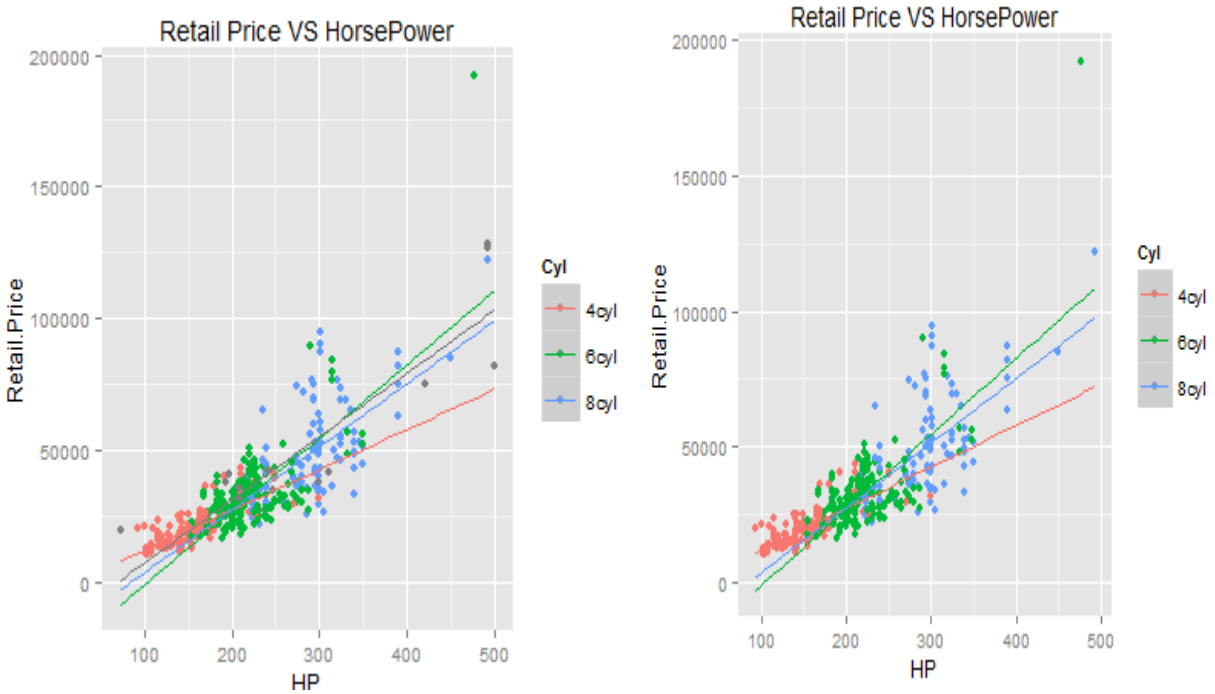
Lab 4
Steven Lin

Exercise 1

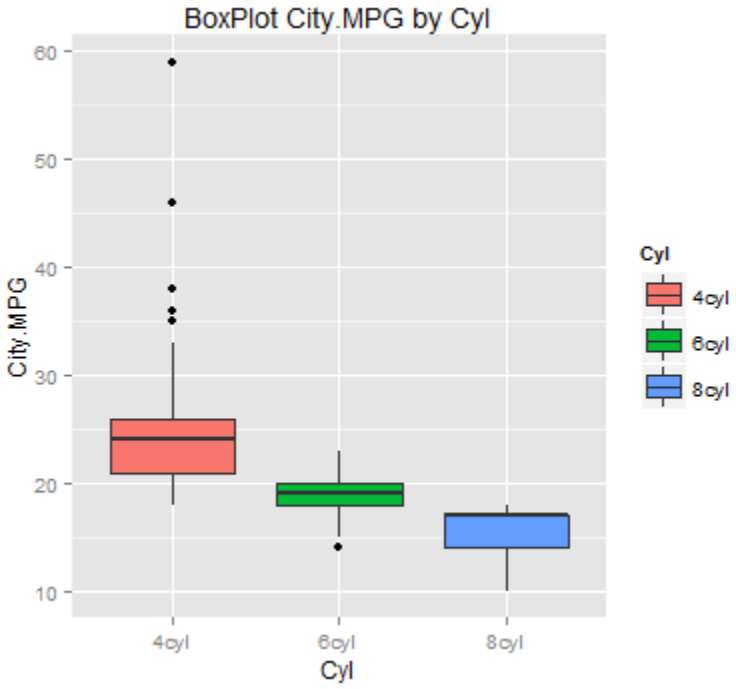
a)



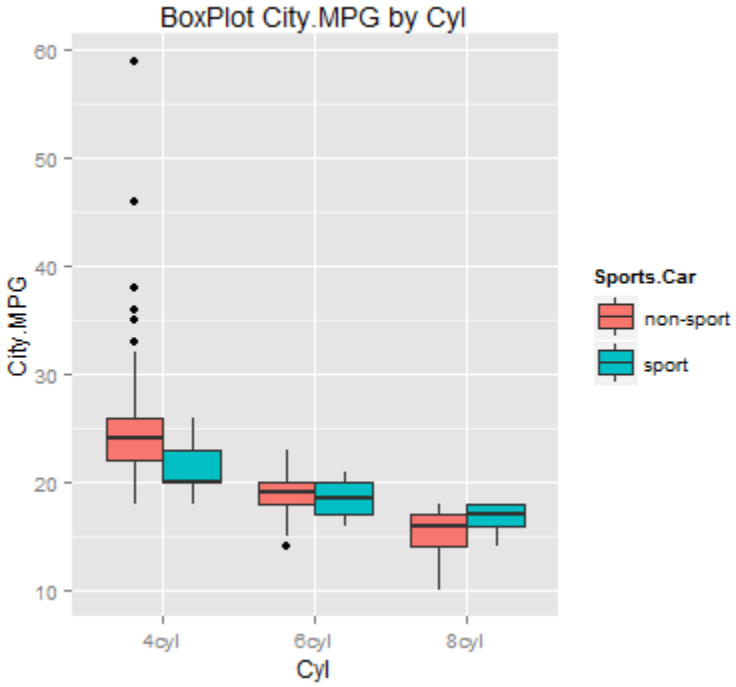
b) *Note: plot on the left has all the points where point with cylinders other than 4, 6 and 8 are plotted in gray because using factor on cylinders on the entire data converts these points to NA. Plot in the right only plots the subset of data with cylinders 4, 6 and 8. For the next questions I use the subset data.*



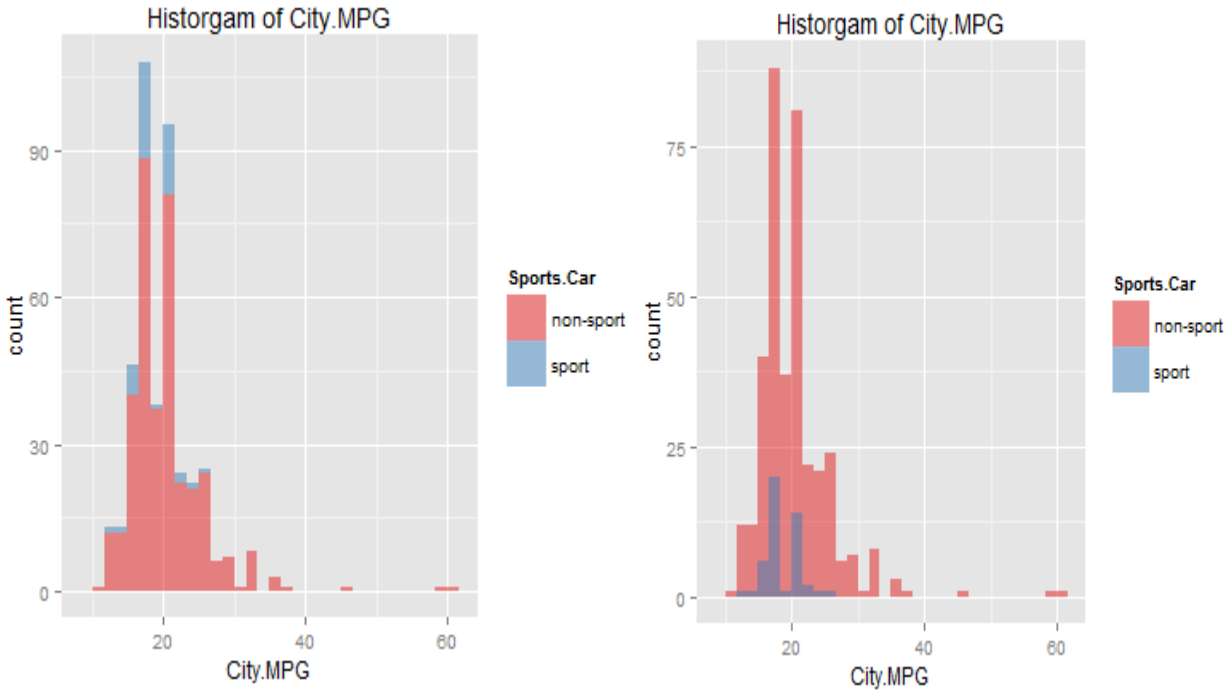
c)


BoxPlot City.MPG by Cyl
City.MPG
60
50
40
30
20
10
4cyl
6cyl
8cyl
Cyl
Cyl
4cyl
6cyl
8cyl

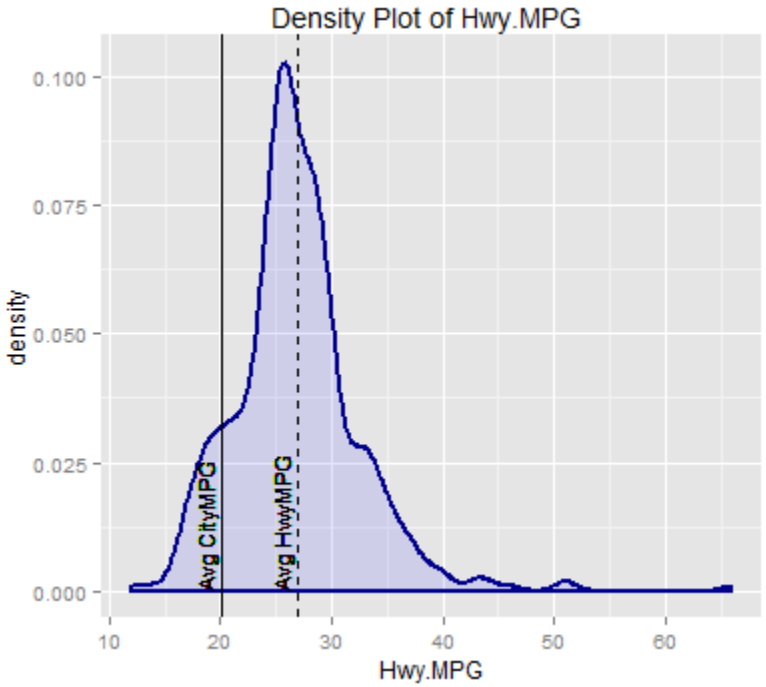
d)


BoxPlot City.MPG by Cyl
City.MPG
60
50
40
30
20
10
4cyl
6cyl
8cyl
Cyl
Sports.Car
non-sport
sport

e) *Note: not sure what "on top of each other" refers to. So I included "stacked" (left) and "overlaid" (right) histograms. The overlaid histogram makes much more sense to use with transparency since you can clearly compare the distributions of sports vs. non-sports.*

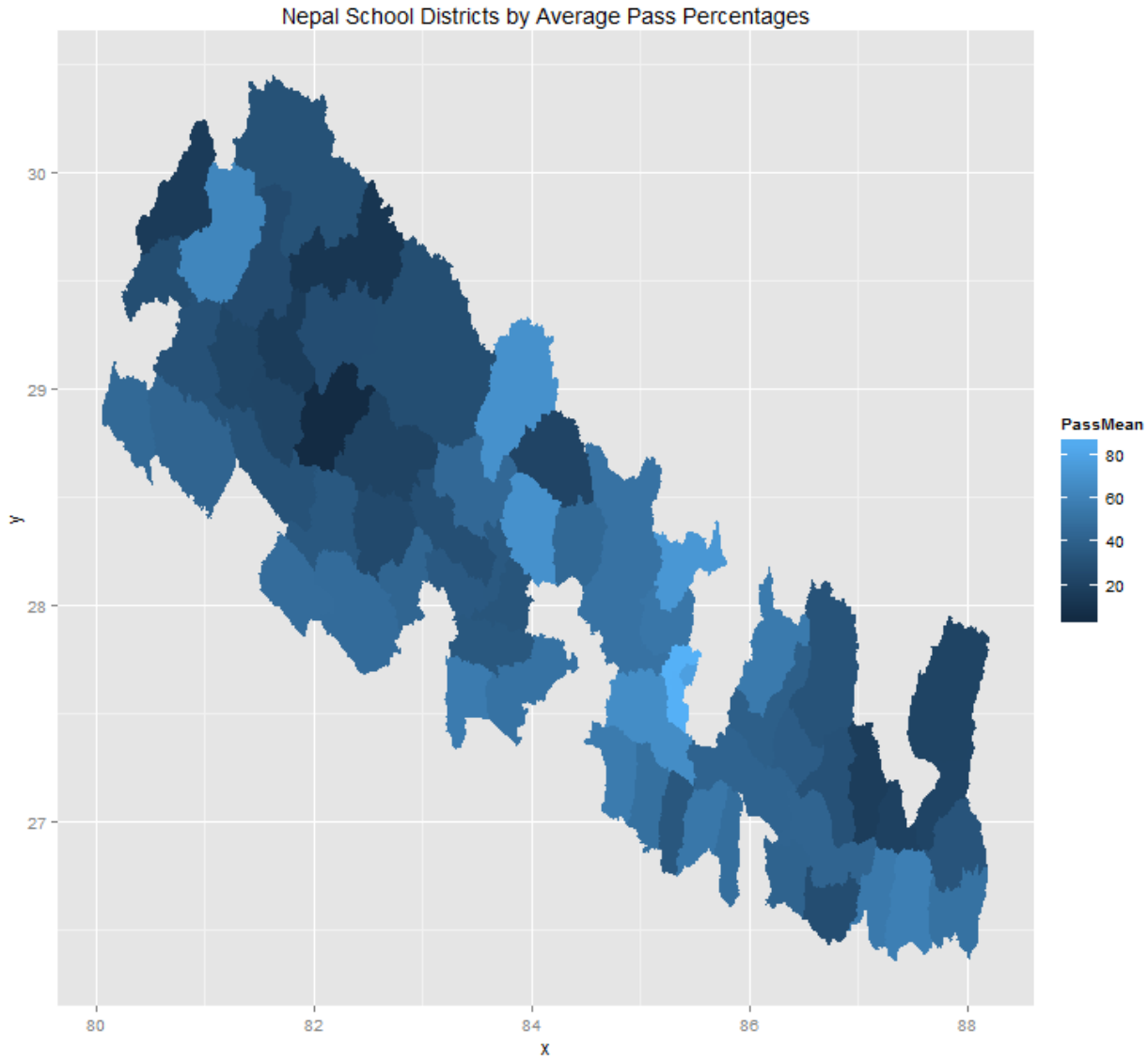


f)

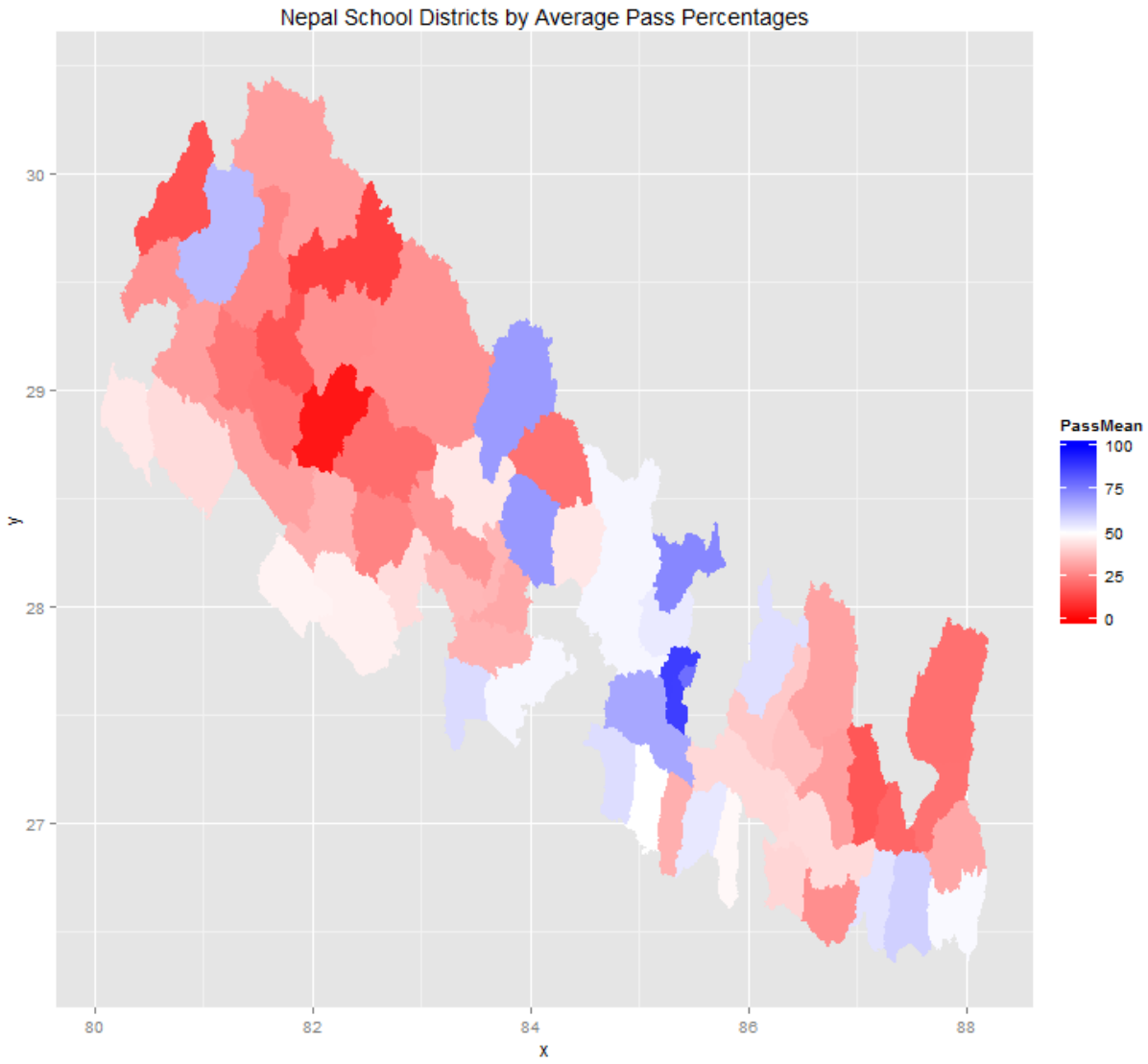


Exercise 2

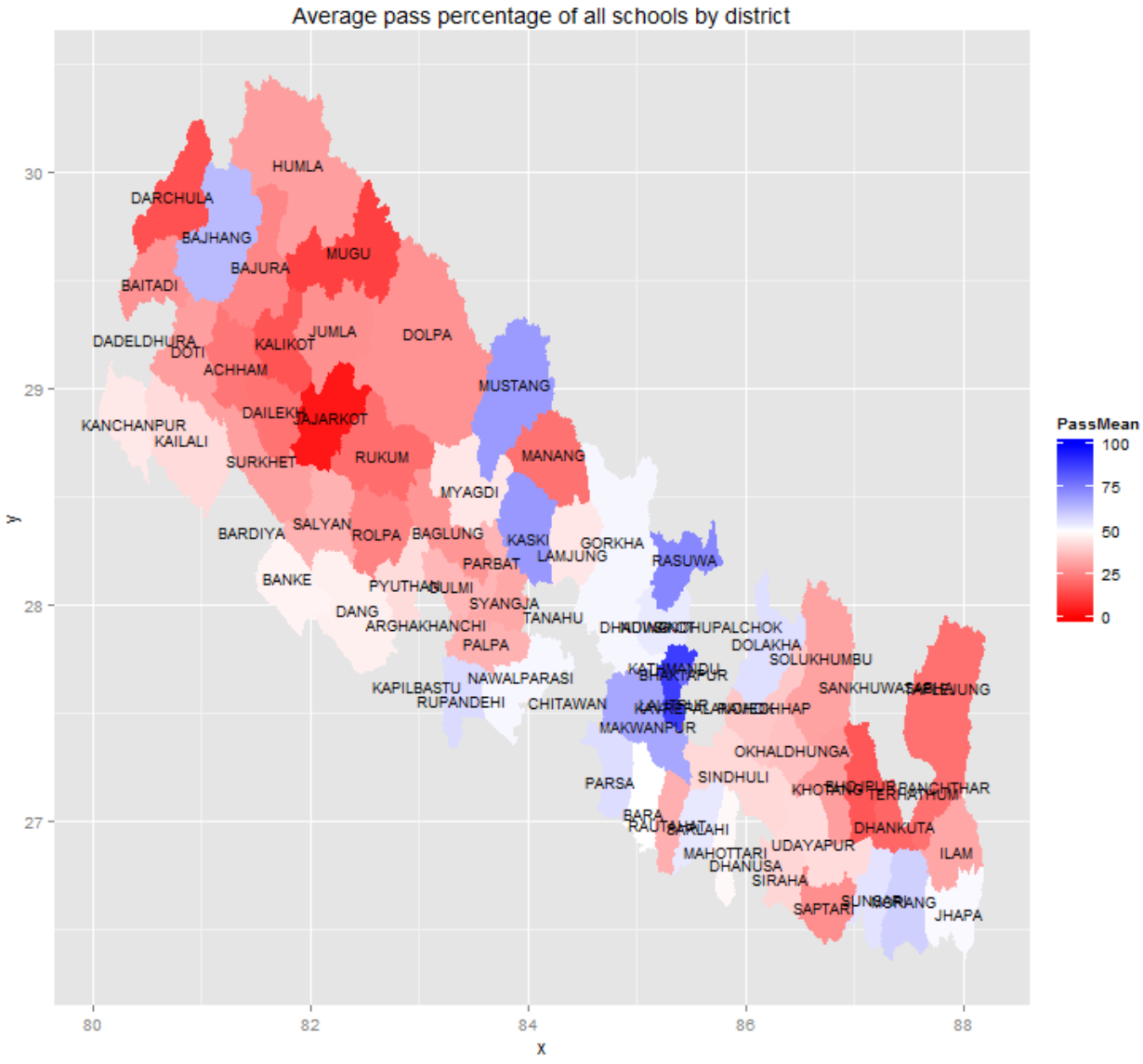
a)


Nepal School Districts by Average Pass Percentages
30
29
28
27
y
80
82
84
86
88
x
PassMean
80
60
40
20

b)


Nepal School Districts by Average Pass Percentages
30
29
28
27
y
80
82
84
86
88
x
PassMean
100
75
50
25
0

c)


Average pass percentage of all schools by district
HUMLA
DARCHULA
BAJHANG
BAJURA
MUGU
BAITADI
DADELDHURA
DOTI
KALIKOT
JUMLA
DOLPA
ACHHAM
MUSTANG
DAILEKH
JAJARKOT
KANCHANPUR
KAILALI
SURKHET
RUKUM
MANANG
MYAGDI
BARDIYA
SALYAN
ROLPA
BAGLUNG
KASKI
GORKHA
LAMJUNG
PARBAT
RASUWA
BANKE
PYUTHAN
GULMI
SYANGJA
DANG
ARGHAKHANCHI
TANAHU
PALPA
DOLAKHA
SOLUKHUMBU
NAWALPARASI
KAPILBASTU
RUPANDEHI
CHITAWAN
MAKWANPUR
OKHALDHUNGA
SINDHULI
PARSA
TERHATHUM
DHANKUTA
UDAYAPUR
MAHOTTARI
DHANUSA
SIRAHA
SAPTARI
JHAPA
ILAM
PassMean
100
75
50
25
0
30
29
28
27
y
80
82
84
86
88
x

Code

```
# Lab session 4 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  ##########################################################################

library(ggplot2)

## Part a ####

# Using qplot() create a scatter plot of "HP" against "Retail.Price."
# Fit a second degree polynomial through the data, i.e. Retail.Price
# = HP2. Make sure that the datapoints and the fitted line are
# shown on the same figure.

qplot(HP,Retail.Price, data = carsdata, geom =c("point","smooth"),
      formula=y~poly(x,2), method="lm",main="Retail Price VS HorsePower")

## Part b ####

# Select all the cars with 4, 6 and 8 cylinders. For each cylinder
# category regress (using "lm") "HP" on Retail.Price." In total you
# need to have three regression lines. Remove the 95% confidence
# interval around the lines.

table(carsdata$Cyl)

dim(carsdata) # 428 rows
mydata = carsdata

mydata$Cyl = factor(mydata$Cyl,levels=c(4,6,8),labels=c("4cyl","6cyl","8cyl"))

# data points with no cyl = 4,6,8
index = which(!(carsdata$Cyl %in% c(4,6,8)))

carsdata[index,]
mydata[index,]

dim(mydata) # 428 rows
table(mydata$Cyl)

# when using factors, cyl other than 4,6 and 8 get NA, so when plotting they
# are ignored

```

```r
sub = carsdata[(carsdata$Cyl %in% c(4,6,8)),]
sub$Cyl = factor(sub$Cyl,levels=c(4,6,8),labels=c("4cyl","6cyl","8cyl"))

# points that are not 4,6 and 8 are plotted in gray color
qplot(HP,Retail.Price, data = mydata,
      geom = c("point","smooth"),formula=y~x, level =0, fullrange=T,
      color=Cyl,method="lm",main="Retail Price VS HorsePower")

# subset data 4,6 and 8
qplot(HP,Retail.Price, data = sub,
      geom = c("point","smooth"),formula=y~x, level =0, fullrange=T,
      color=Cyl,method="lm",main="Retail Price VS HorsePower")

## Part c ####

# For the same subset of the data as in part (b) (cars with 4,
# 6 and 8 cylinders), use ggplot() to a create boxplot for
# "City.MPG" for all three categories.
# (The final figure should have 3 boxplots and "City.MPG" on
# the y-axis.) Give a different fill color to each boxplot.

ggplot(sub, aes(x=Cyl, y=City.MPG)) +
  geom_boxplot(aes(fill = Cyl))+
  ggtitle("BoxPlot City.MPG by Cyl")

## Part d ####

# On the final figure from part (c) add another layer of boxplots
# in the following way: for every category of cylinders create
# two boxplots, one corresponding to sports cars ("Sports.Car" = 1),
# and the other corresponding to non-sports cars ("Sports.Car" = 0).
# Your resulting figure should have 6 boxplots.
# Use two fill colors; one color for sports cars boxplots
# and another for non-sports cars boxplots. You must use ggplot().

table(sub$Sports.Car)
sub$Sports.Car = factor(sub$Sports.Car,levels=c(0,1), labels=c("non-sport","sport"))

ggplot(sub, aes(x=Cyl, y=City.MPG)) +
  geom_boxplot(aes(fill = Sports.Car))+
  ggtitle("BoxPlot City.MPG by Cyl")


      ## Part e ####

      # Using ggplot() create a histogram for "City.MPG" conditional
      # on whether a car is a sports car or not (so you have to end
      # up with 2 histograms). Both histograms must be shown on the same
      # figure on top of each other. Use two different colors and
      # set the transparency control alpha = 0.5

      mydata = carsdata
      mydata$Sports.Car = factor(mydata$Sports.Car,levels=c(0,1),
                                 labels=c("non-sport","sport"))

      # default  = stacked
      ggplot(mydata, aes(x = City.MPG)) +
        geom_histogram(aes(fill=Sports.Car ),alpha = 0.5)+
        ggtitle("Historgam of City.MPG")+
        scale_fill_brewer(palette="Set1")

```

```
      # overlayed
      ggplot(mydata, aes(x=City.MPG, fill=Sports.Car)) +
        geom_histogram(position="identity", alpha=0.5)+
        ggtitle("Historgam of City.MPG") +
        scale_fill_brewer(palette="Set1")

      ## Part f ####

      # Using ggplot() create a density plot for "Hwy.MPG" and draw
      # a dashed line showing the mean of "Hwy.MPG" and a straight
      # line showing the mean of "City.MPG".

      meanHwyMPG = mean(mydata$Hwy.MPG,na.rm=T)
      meanCityMPG = mean(mydata$City.MPG,na.rm=T)


      ggplot(mydata, aes(x=Hwy.MPG))+
        geom_density(colour="darkblue", size=1, fill="blue", alpha=0.1) +
        geom_vline(aes(xintercept=meanHwyMPG ), linetype="dashed")+
        geom_vline(aes(xintercept=meanCityMPG))+
        ggtitle("Density Plot of Hwy.MPG ") +
        geom_text(data=mydata, mapping=aes(x=c(meanHwyMPG,meanCityMPG),
                                           y=0, label=c("Avg HwyMPG", "Avg CityMPG")),

                  size=4, angle=90, vjust=-0.4, hjust=0)


      ##### Ex1  ################################################################

      # Consider the "nepal.csv" data set.

      # Part a ####
      # Go to GADM.org and download the shape file for Nepal's map (
      # you will download a zip file that contains "NPL_adm3.shp").
      # Re-run the code from lab 4 to set up Nepal's map such that each
      # region's color corresponds to the average pass percentage of all
      # schools in that district.

      library(plyr)
      library(rgeos)
      library(maptools)
      library(sp)
      library(rgdal)
      library(ggplot2)
      library(Rcpp)
      #library(gpclib)

      # http://gadm.org/download
      nepal = read.csv("nepal.csv",header=TRUE,na.strings=c("","*","NA"))

      str(nepal)

      #donwload shapefile
      #go to GADM.org, select the country you are interested in, and donwload
      #the shp file (you will donwload a zip file
      #that contains the shp file you're interested in)

      np_dist = readShapeSpatial("NPL_adm/NPL_adm3.shp")
      plot(np_dist)
```

```
#This function turns a map into a data frame than can more easily be plotted with ggplot2.
np_dist = fortify(np_dist,region="NAME_3")

#When I was using fortify function, I didn't specify the region and later I set it as "NAME_3"#. Here is the reason why I can only use "NAME_3".

#In the unzipped folder, you can see such files: NPL_adm1, NPL_adm2, NPL_adm3, NPL_adm4. #"adm1" means administrative level 1 such as "West", "East", "Central". "adm2" can refer to #"State". As the data I read into R is NPL_adm3, I should specify the region to the column #which represents this level. So, "NAME_3" is for "NPL_adm3".

#In other words, if you load dataset "NPL_adm2", you have to specify the region="NAME_2".

# since each row contains data about 1 school, we want to take the average
# of schools in the same district.
# use ddply to do averaging

# Take the mean of PASS.PERCENT by District
# Note the use of the '.' function to allow District to be used without quoting

distrpassave = ddply(nepal, .(District), summarize, PassMean = mean(PASS.PERCENT))
distrpassave = ddply(nepal, ~District, summarize, PassMean = mean(PASS.PERCENT))


np_dist$id = toupper(np_dist$id)  #change ids to uppercase

ggplot() +
  geom_map(data = distrpassave, aes(map_id = District, fill = PassMean), map = np_dist)+
  expand_limits(x = np_dist$long, y = np_dist$lat) +
  ggtitle("Nepal School Districts by Average Pass Percentages")

# Part b ####
# Change the colors on the map. Make the low values red, the mid
# values white, and the high values blue. Set the midpoint to 50.

ggplot() +
  geom_map(data = distrpassave, aes(map_id = District, fill = PassMean), map = np_dist) +
  expand_limits(x = np_dist$long, y = np_dist$lat) +
  scale_fill_gradient2(low="red", mid = "white", midpoint = 50, high = "blue",limits=c(0,100)) +
  ggtitle("Nepal School Districts by Average Pass Percentages")

# Part c ####
# Add the district names to the map.

distlabels = ddply(np_dist, .(id), summarize, long = mean(long), lat = mean(lat))

ggplot() +
  geom_map(data = distrpassave, aes(map_id = District, fill = PassMean), map = np_dist) +
  expand_limits(x = np_dist$long, y = np_dist$lat) +
  scale_fill_gradient2(low="red", mid = "white", midpoint = 50, high = "blue",limits=c(0,100)) +
  ggtitle("Average pass percentage of all schools by district")+
  geom_text(data = distlabels,aes(long,lat,label=id),size=3)
```