> library("Cairo", lib.loc="~/R/win-library/3.2")

> library("colorspace", lib.loc="~/R/win-library/3.2")

> library("ggplot2", lib.loc="~/R/win-library/3.2")

> library("googleVis", lib.loc="~/R/win-library/3.2")

Welcome to googleVis version 0.5.10

Please read the Google API Terms of Use

before you start using the package:

https://developers.google.com/terms/

Note, the plot method of googleVis will by default use

the standard browser to display its output.

See the googleVis package vignettes for more details,

or visit http://github.com/mages/googleVis.

To suppress this message use:

suppressPackageStartupMessages(library(googleVis))

> library("mapproj", lib.loc="~/R/win-library/3.2")

Loading required package: maps

# maps v3.1: updated 'world': all lakes moved to separate new #

# 'lakes' database. Type '?world' or 'news(package="maps")'. #

> detach("package:maps", unload=TRUE)

Error: package ‘maps’ is required by ‘mapproj’ so will not be detached

> library("maps", lib.loc="~/R/win-library/3.2")

> library("maptools", lib.loc="~/R/win-library/3.2")

Loading required package: sp

Checking rgeos availability: FALSE

Note: when rgeos is not available, polygon geometry computations in maptools depend on gpclib,

which has a restricted licence. It is disabled by default;

to enable gpclib, type gpclibPermit()

> library(googleVis)

> make.state.abbreviation <- function(x){switch(x,

+ "Alaska" = "AK", "Alabama" = "AL", "Arkansas" = "AR",

+ "Arizona" = "AZ", "California" = "CA",

+ "Colorado" = "CO", "Connecticut" = "CT", "District of Columbia" = "DC",

+ "Delaware" = "DE", "Florida" = "FL",

+ "Georgia" = "GA", "Hawaii" = "HI", "Iowa" = "IA",

+ "Idaho" = "ID", "Illinois" = "IL", "Indiana" = "IN",

+ "Kansas" = "KS", "Kentucky" = "KY", "Louisiana" = "LA",

+ "Massachusetts" = "MA", "Maryland" = "MD", "Maine" = "ME",

+ "Michigan" = "MI", "Minnesota" = "MN", "Missouri" = "MO",

+ "Mississippi" = "MS", "Montana" = "MT",

+ "North Carolina" = "NC", "North Dakota" = "ND",

+ "Nebraska" = "NE", "New Hampshire" = "NH", "New Jersey" = "NJ",

+ "New Mexico" = "NM", "Nevada" = "NV", "New York" = "NY",

+ "Ohio" = "OH", "Oklahoma" = "OK", "Oregon" = "OR",

+ "Pennsylvania" = "PA",

+ "Rhode Island" = "RI", "South Carolina" = "SC", "South Dakota" = "SD",

+ "Tennessee" = "TN", "Texas" = "TX",

+ "Utah" = "UT", "Virginia" = "VA", "Vermont" = "VT",

+ "Washington" = "WA", "Wisconsin" = "WI",

+ "West Virginia" = "WV", "Wyoming" = "WY", "")}

> my.data.frame <- read.csv("LSAT\_Scores\_by\_School", header = TRUE)

Error in file(file, "rt") : cannot open the connection

In addition: Warning message:

In file(file, "rt") :

cannot open file 'LSAT\_Scores\_by\_School': No such file or directory

> my.data.frame <- read.csv("LSAT\_Scores\_by\_School.csv", header = TRUE)

Error in file(file, "rt") : cannot open the connection

In addition: Warning message:

In file(file, "rt") :

cannot open file 'LSAT\_Scores\_by\_School.csv': No such file or directory

> setwd("C:/Users/crmo/Desktop/Northwestern Stuff/Data Visualization/Assignment4/LSAT Code")

> my.data.frame <- read.csv("LSAT\_Scores\_by\_School.csv", header = TRUE)

> my.data.frame$State <- as.character(my.data.frame$State)

> my.data.frame$State <- rep("", length = nrow(my.data.frame))

> for(index.for.state in seq(along = my.data.frame$State))

+ my.data.frame$state[index.for.state] <- make.state.abbreviation(my.data.frame$State[index.for.state])

> print(my.data.frame[,c("State", "state")])

State state

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

> y.value.gradient <- c(min(my.data.frame$Median.LSAT),

+ median(my.data.frame$Median.LSAT),

+ max(my.data.frame$Median.LSAT))

> print(my.value.gradient)

Error in print(my.value.gradient) : object 'my.value.gradient' not found

> my.value.gradient <- c(min(my.data.frame$Median.LSAT),

+ median(my.data.frame$Median.LSAT),

+ max(my.data.frame$Median.LSAT))

> print(my.value.gradient)

[1] 160 164 173

> javascript.us.map.object <- gvisGeoChart(my.data.frame, "state", "Median LSAT",

+ options=list(region="US",

+ displayMode="regions",

+ resolution="provinces",

+ colorAxis = "{values: [160, 164, 173], colors: [\'coral', \'lightgray', \'blue']}",

+ width=700, height=500))

> plot(javascript.us.map.object)

> javascript.us.map.object <- gvisGeoChart(my.data.frame, "rank", "Median LSAT",

+ options=list(region="US",

+ displayMode="regions",

+ resolution="provinces",

+ colorAxis = "{values: [160, 164, 173], colors: [\'coral', \'lightgray', \'blue']}",

+ width=700, height=500))

> plot(javascript.us.map.object)

> javascript.us.map.object <- gvisGeoChart(my.data.frame, "state", "Average LSAT",

+ options=list(region="US",

+ displayMode="regions",

+ resolution="provinces",

+ colorAxis = "{values: [160, 164, 173], colors: [\'coral', \'lightgray', \'blue']}",

+ width=700, height=500))

> my.value.gradient <- c(min(my.data.frame$Median.Average.LSAT),

+ median(my.data.frame$Median.Average.LSAT),

+ max(my.data.frame$Median.Average.LSAT))

Warning messages:

1: In min(my.data.frame$Median.Average.LSAT) :

no non-missing arguments to min; returning Inf

2: In is.na(x) : is.na() applied to non-(list or vector) of type 'NULL'

3: In max(my.data.frame$Median.Average.LSAT) :

no non-missing arguments to max; returning -Inf