Sample Answers Lab 03: Univariate Plots and Maps

Handout date: Wednesday, September 16, 2020

Due date: Wednesday, September 23, 2020 by midnight as Word document into eLearning's SubmitLab03 link

Task 1: Color Brewer (0.6 points)

Link the library **`RColorBrewer`** to your R session and ***explore the online help*** and the function **`display.brewer.all( )`**.

The table below shows three different color palettes with 7 classes, which are tailored toward colorblind readers.

		
Palette 1	*Palette 2*	*Palette 3*

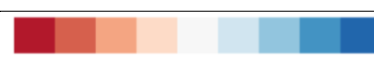
[a] For each palette identify its "name" in the library **`RColorBrewer`** and underlying type, i.e., *sequential*, *diverging* and *qualitative*. Justify your answer. (0.3 points)

Palette 1: Blues is used sequential variables. Sequential palettes are suited to ordered data that progress from low to high. Lightness steps dominate the look of these schemes, with light colors for low data values to dark colors for high data values.

Palette 2: Set2 is used for qualitative variables. Qualitative palettes do not imply magnitude differences between legend classes, and hues are used to create the primary visual differences between classes. Qualitative schemes are best suited to representing nominal or categorical data. Note: For set 2 the maximum numbers of colors is 8, however, for set 3 it increases to 12 different colors.

Palette 3: RdBu is used for variables diverging around a reference value. Diverging palettes put equal emphasis on mid-range critical values and extremes at both ends of the data range. The critical class or break in the middle of the legend is emphasized with light colors and low and high extremes are emphasized with dark colors that have contrasting hues.

[b] Recreate the three palettes with ***9 classes*** (note palette 1 only allows for 8 classes) and put their properly sized images into the table below. (0.3 points)

		
display.brewer.pal(9, "Blues")	*display.brewer.pal(8, "Set2")*	*display.brewer.pal(9, "RdBu")*
Palette 1	*Palette 2*	*Palette 3*

Task 2: Mapping (1.2 points)

You are given three **ESRI** shape files. These files are packed into the zipped file **`Italy.zip`**. All geographies are in the ***lat*/*long*** format and therefore not projected. Note: takes care of projecting these files properly.

The maps you will generate consist of two area layers: [a] the countries neighboring Italy (layer **`Neighbors.shp`**) and [b] added on top of it the 95 provinces of Italy (layer **`Provinces.shp`**). Make sure that the ***frame of the plot window*** is sized properly to embed Italy. The base map below shows these two layers in a properly sized window frame.

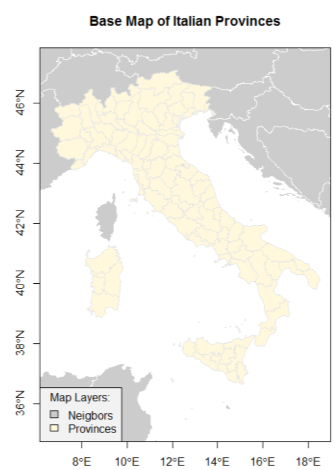
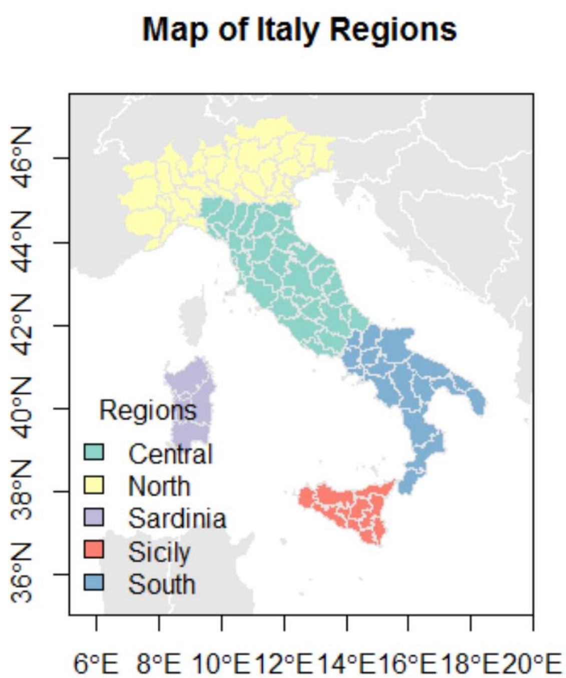


Figure 1: Base Map of the Italian Provinces

You will generate three color maps displaying different map themes. For your lab answers please show these maps in ***color***. Each map should be properly ***framed***, have a proper ***title*** and ***legend***, show the ***neighboring countries*** as spatial reference frame. You can use the mapping functions in the package **`DallasTracts`**. Just show the code for your maps.

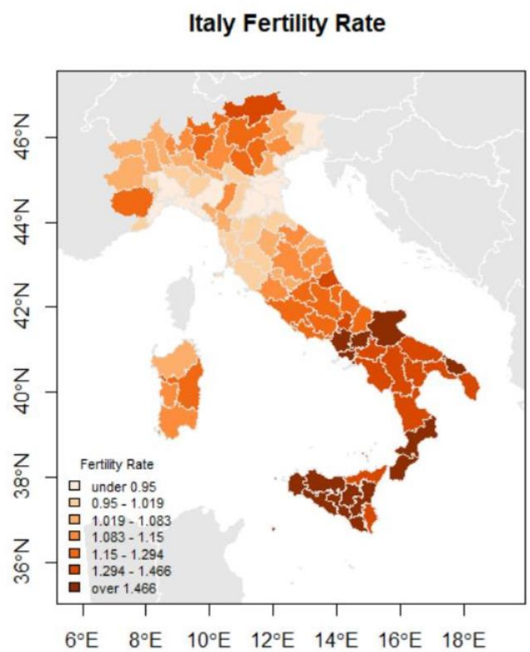
[a] Generate a map showing the Italian regions, which are stored in the variable **`REGION`**. Show the relevant code used to generate the map. (0.3 points)

```
library(TexMix)
library(rgdal)
#Get polygons of neighboring countries
neig.shp <- readOGR(dsn="./Italy",layer = "Neighbors", integer64 ="allow.loss")
#Get polygons of Italy provinces
Italy.shp <- readOGR(dsn="./Italy",layer = "Provinces", integer64 = "allow.loss")
Italy.bbox <- bbox(Italy.shp)
plot(neig.shp,axes = T,col=grey(0.9),border = "white", xlim=Italy.bbox[1,],
ylim=Italy.bbox[2,])
mapColorQual(as.factor(Italy.shp$REGION), Italy.shp,
  map.title="Region Map of Italy",
    legend.title = "Regions",add.to.map=T)
```



[b] Generate a map showing the total fertility rate (number of births per woman) in the Italian provinces using 7 classes, which is stored in the variable **TOTFERTRAT**. Show the relevant code used to generate the map. (0.3 points)

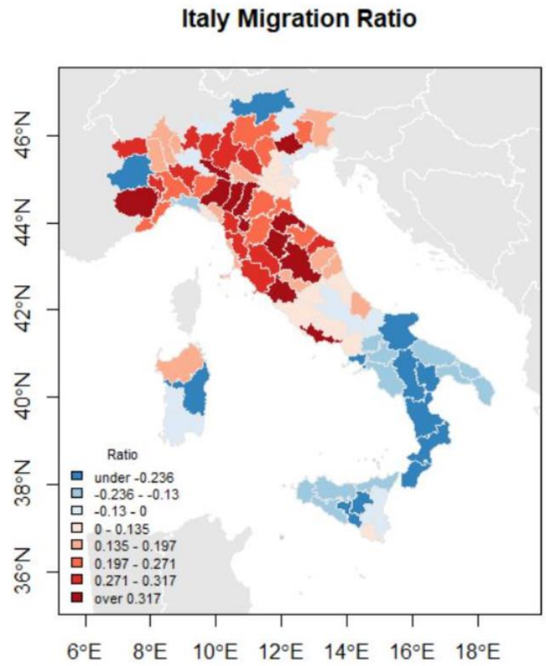
```
plot(neig.shp,axes=T,col=grey(0.9),border="white",
xlim=Italy.bbox[1,], ylim=Italy.bbox[2,])
# addToMap=T over-plots provinces over neighbors
mapColorRamp(Italy.shp$TOTFERTRAT,Italy.shp, breaks=7,
  map.title="Italy Fertility Rate ",
  legend.title="Fertility Rate",add.to.map=T,
  legend.cex=0.7)
```



[c] Generate a map showing the gender ratio in the 95 provinces, which can be calculated with the transformation **`logMigRatio <- log(`*`shp`*`$INFLOW/`*`shp`*`$OUTFLOW)`**, where ***shp*** refers to the name of your imported shape-file. What is the neutral break-point for the variable **`logMigRatio`**. Use in total 8 classes but select the appropriate number of classes for the below and above breakpoint of zero of the underlying distribution of **`logMigRatio`**. Justify your choice. Show the relevant code used to generate the map. (0.6 points)

The natural breaking point is 0 = log 1 when the inflow and outflow are in balance. Values greater than zero indicate population gain due to migration, whereas values below zero mean a province is losing population due to dominant outmigration.

```
Italy.shp$logMigRatio <- log(Italy.shp$INFLOW/Italy.shp$OUTFLOW)
hist(Italy.shp$logMigRatio)
plot(neig.shp,axes=T,col=grey(0.9),border="white",
xlim=Italy.bbox[1,], ylim=Italy.bbox[2,])
sum(Italy.shp$logMigRatio <= 0)
sum(Italy.shp$logMigRatio >= 0)
mapBiPolar(Italy.shp$logMigRatio, Italy.shp,
neg.breaks=3, pos.breaks=5, break.value=0,
map.title="Italy Migration Ratio",
legend.title="Ratio",add.to.map=T,
legend.cex=0.7)
```



Task 3: Statistical Graphs (1.2 points)

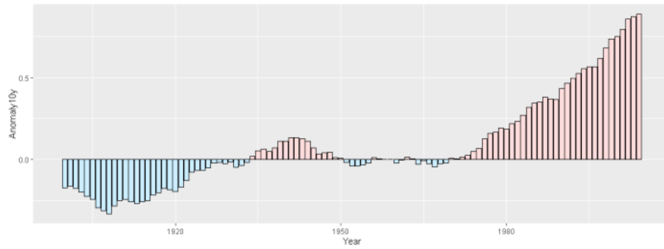
The following graphs are reproductions from the R Graphics Cookbook, 2e. One of the objectives of this task is that you visit the **R Graphics Cookbook** and explore the graphs on display there.

Study the sections associated with the graphs, which are displayed below. Its sections document how these graphs were built.

You job in this is to *reproduce* these graphs and *show the code*, which generated them. The data used in the **R Graphics Cookbook** are available in **`library("gcookbook")`**.

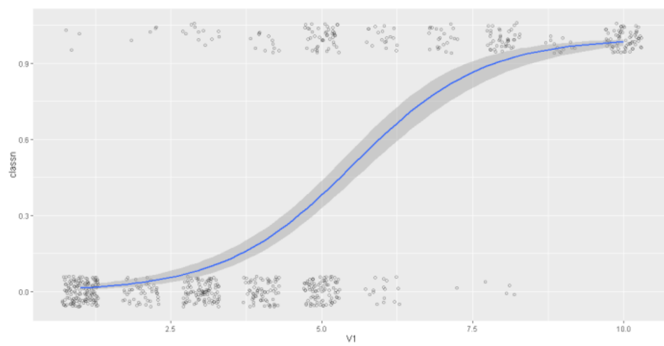
[a] Reproduce the graph below and show the code (0.3 pts)

```
library(gcookbook) # Load gcookbook for the climate data set
library(dplyr)
library(ggplot2)

climate_sub <- climate %>%
  filter(Source == "Berkeley" & Year >= 1900) %>%
  mutate(pos = Anomaly10y >= 0)

ggplot(climate_sub, aes(x = Year, y = Anomaly10y, fill = pos)) +
  geom_col(position = "identity", colour = "black", size = 0.25) +
  scale_fill_manual(values = c("#CCEEFF", "#FFDDDD"), guide = FALSE)
```



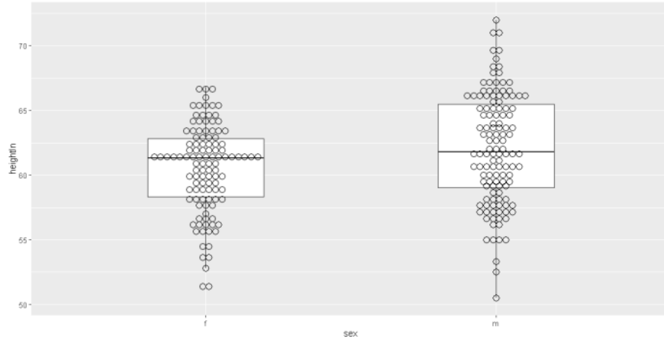
[b] Reproduce the graph below and show the code (0.3 pts)

```
library(MASS)
biopsy_mod <- biopsy %>% mutate(classn = recode(class, benign=0,
malignant=1))

ggplot(biopsy_mod, aes(x=V1, y=classn))+
  geom_point(position=position_jitter(width=0.3, height=0.06), alpha=0.4,
shape=21, size=1.5)+
  stat_smooth(method=glm, method.args=list(family=binomial))
```



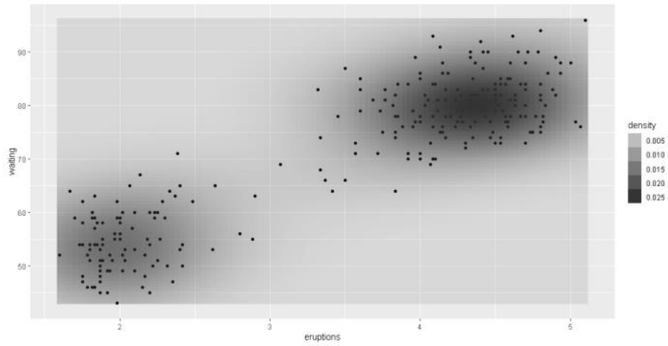
[c] Reproduce the graph below and show the code (0.3 pts)

```
ggplot(heightweight, aes(x=sex, y=heightIn)) +
  geom_boxplot(outlier.colour = NA, width= 0.4)+
  geom_dotplot(binaxis = "y", binwidth = 0.5, stackdir = "center", fill=NA)
```



[d] Reproduce the graph below and show the code (0.3 pts)

```
ggplot(faithful, aes(x=eruptions, y=waiting))+
  geom_point()+
  stat_density2d(aes(alpha=..density..), geom="tile", contour = FALSE)
```



Hint: For some graphs you would also need to look at the data transformation that have been applied before the graph is rendered.

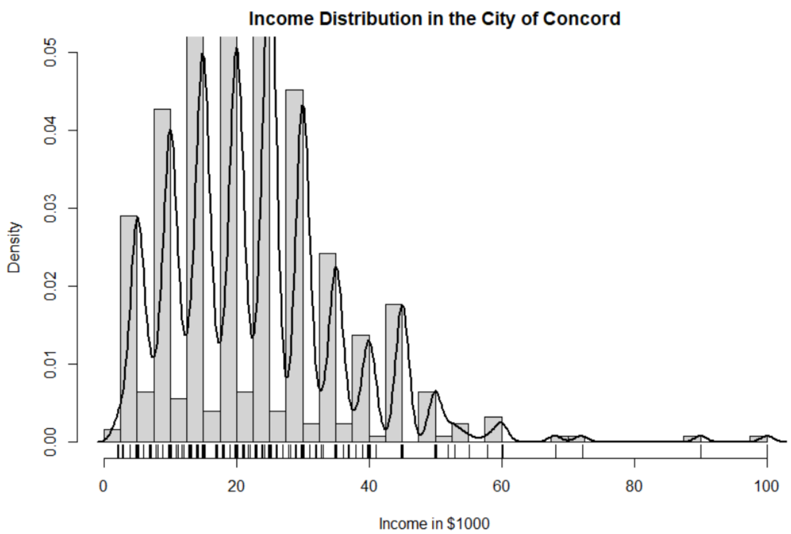
Task 4: Histograms with Kernel density overlays (1 point)

Open the **Concord1.sav** file and generate three histograms with the code

```
hist(concord$income, breaks=seq(0,100,by=???), freq=FALSE,
     ylim=c(0,0.05), xlab="Income in $1000",
     main="Income Distribution in the City of Concord")
rug(jitter(concord$income, factor=1), side=1)
lines(density(concord$income, bw=???), lwd=2)
```

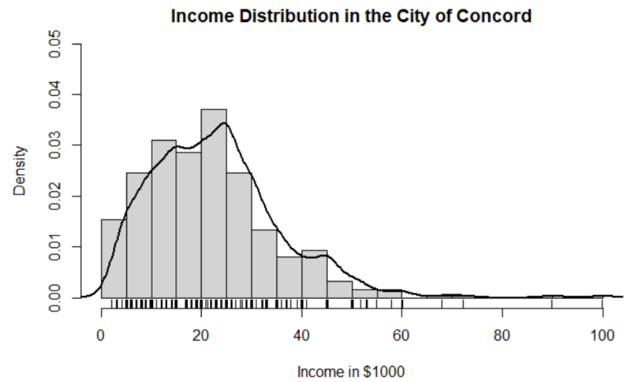
a. Show the histogram with the associated density curve for **by=2.5** and **bw=1**.

```
concord <- foreign::read.spss('Concord1.sav')
hist(concord$income, breaks=seq(0,100,by=2.5), freq=FALSE,
     ylim=c(0,0.05), xlab="Income in $1000",
     main="Income Distribution in the City of Concord")
rug(jitter(concord$income, factor=1), side=1)
lines(density(concord$income, bw=1), lwd=2)
```

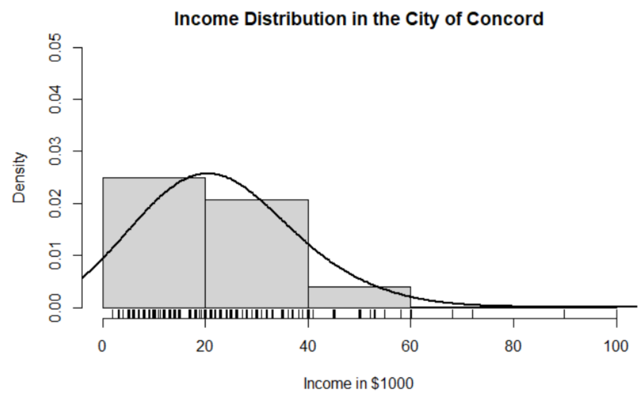


Comment: This histogram provides too much details due to random sampling variations, which makes it difficult to detect the overall characteristic of the distribution.

b. Show the histogram with the associated density curve for **by=5** and **bw=2.5**.



c. Show the histogram with the associated density curve for **`by=20`** and **`bw=10`**.



Comment: This histogram over-smoothed the density and masked relevant variations in the data.

d. Justify, which of the three parameter combinations most realistically reflects the distribution of the income.

The histogram with by = 5 and bw = 2.5 better depicts the distribution of data in the underlying population.

e. Show the ***stem-and-leaves*** plot (function **`stem( )`**)to identify whether the income variable has the tendency to cluster around particular values.

<u>Hint:</u> The income was reported by sampled household heads. Do people have the tendency to round their responses?

```
stem(concord$income)
   0 | 223334
   0 | 55555555555555555555555555555555666677778888899
   1 | 000000000000000000000000000000000000000000000011122223333333444444
   1 | 55555555555555555555555555555555555555555555555555555555777777888888
   2 | 00000000000000000000000000000000000000000000000000000000011111222333
   2 | 55555555555555555555555555555555555555555555555555555555555555555555+3
   3 | 0000000000000000000000000000000000000000000000000000122333
   3 | 55555555555555555555555555567789
   4 | 0000000000000001
   4 | 5555555555555555555555
   5 | 00000000233
   5 | 58
```

```
   6 | 000
   6 | 8
   7 | 2
   7 |
   8 |
   8 |
   9 | 0
   9 |
  10 | 0
```

The data values seem to cluster around 0s and 5s. This indicates that respondents round their incomes.