Sample Answer Lab 03: Univariate Plots and Maps

Handout date: Wednesday, September 18, 2019
Due date: Wednesday, September 25, 2019 at the beginning of class as hardcopy
This lab counts 4 % toward your total grade

Task 1: Color Brewer (0.6 points)

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

 *Palette 1*	 *Palette 2*	 *Palette 3*

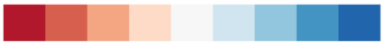
[a] For each palette identify its "name" in the library RColorBrewer and underlying type, i.e., sequential, diverging and qualitative. Justify your answer. (0.3 points)

Palette 1: 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 2: 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 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*** and put their properly sized images into the table below. (0.3 points)

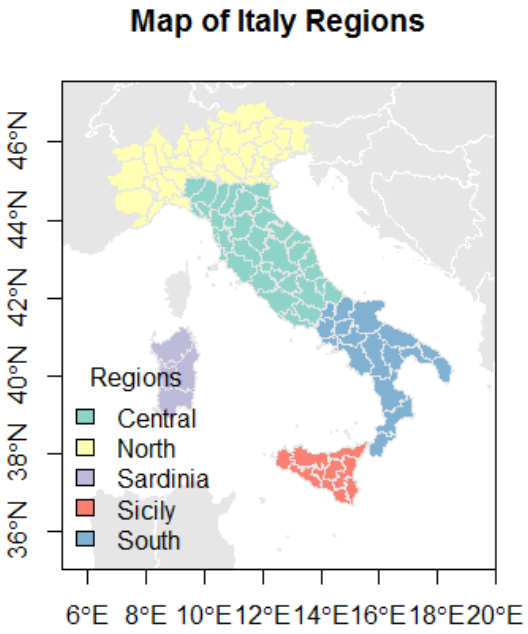
 *display.brewer.pal(8, "Set2")*	 *display.brewer.pal(9, "Blues")*	 *display.brewer.pal(9, "RdBu")*

Task 2: Mapping (1.2 points)

[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)

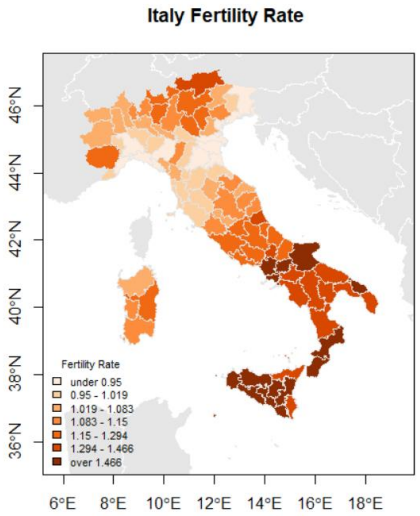
```
library(DallasTracts)
library(rgdal)
```

```
#Get polygons of neighboring countries
neig.shp <- readOGR(dsn="./Data",layer = "Neighbors", integer64 =
"allow.loss")
#Get polygons of Italy provinces
Italy.shp <- readOGR(dsn="./Data",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(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)
```

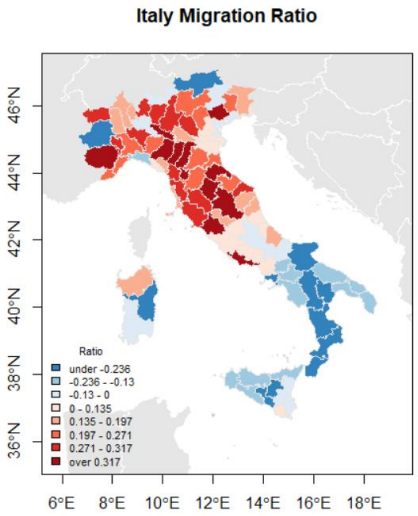


[c] Generate a map showing the gender ratio in the 95 provinces, which can be calculated with the transformation **`logMigRatio <- log(`*`shp`*`$INFLOW/`*`shp`*`$OUTFLOW)`**, where ***`shp`*** refers to the name of your imported shape-file. What is the neutral break-point for the variable **`logMigRatio`**.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



Study Kabacoff's Chapter 6 "Basic Graphs" pp 117-136

In order to solve the remaining tasks, you need to study Kabacoff's Chapter 6 "Basic Graphs" pp 117-136 and adjust the code examples given there to generate the requested graphs for the different datasets.

Task 3: Bar-plots and their Derivatives (1.3 points)

[a] Discuss how the organization of the **`Titanic`** data in line 4 changes for the **`Titanic.df`** data-frame in line 7 to the **`Titanic.pas.df`** data-frame in line 11. (0.3 points)

Comment: The original **`Titanic`** data format is in cross-tables, which display class and sex information by age and survival status. **`Titanic.df`** data-frame in line 7 lists the frequency of records based on the different ***factor level combinations***. If we replicate those factor level records by the frequency count in **`Titanic.df`** data-frame, we will get the **`Titanic.pas.df`** data-frame in line 11. This table consists of one record for each passenger and crew onboard of the Titanic with their status in the categories age, sex, class and survival.

[b] What are the **`sapply( )`** function calls in lines 8, 12 and 13 doing? (0.2 points)

Comment: For each column in the dataframe, **`sapply()`** returns the result of applying function to the corresponding variables.

[c] How do the two tables in lines 16 and 17 differ? (0.1 points)

Comment: The entries in the table in line 16 are sorted according to the order of the factor levels in the variable class whereas the entries in the table in line 17 are sorted descending according to the frequency of observations in each factor level.

[d] What is the difference between the two cross-tabulations in lines 20 and 21? (0.1 points)

Comment: The rows and columns are exchanged.

[e] Recreate the bar-plot shown below and show the code that you have used to generate the plot. Which of the tables in lines 16 and 17 should you use? (0.2 points)

```
class.cnt <- sort(table(Titanic.pas.df$Class), decreasing = T)
barplot(class.cnt, xlab = "Class", ylab = "Frequency",
        main="Composition of People on Board of the Titanic ")
```

Comment: The table in line 17 is used because that table is sorted by descending order. **`Class`** is a nominal variable therefore it is good to first display the most frequent category, then the next frequent category and so forth.

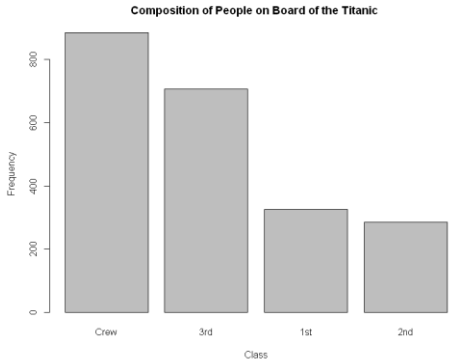


Figure 1: Histogram sorted by frequency

[f] Recreate both bar-plots shown below and show the specific code that you have used to generate these plots. Briefly discuss what the difference between both plots is. Hint: look at the scales for the y-axes. (0.4 points)

```
surByClass <- table(Titanic.pas.df$Survived, Titanic.pas.df$Class)
barplot(surByClass, xlab="Class", ylab="Frequency",
        main="Survival by Class on Board of the Titanic",
        legend=rownames(surByClass))
vcd::spine(classBySur,
           main="Percentage of Survival by Class on Board of the Titanic")
```

Comment: Left figure displays survival status of passengers and crew by absolute frequency counts. However, right figure displays the proportion of survived people by relative frequency, which makes it easier to compare information across different groups.

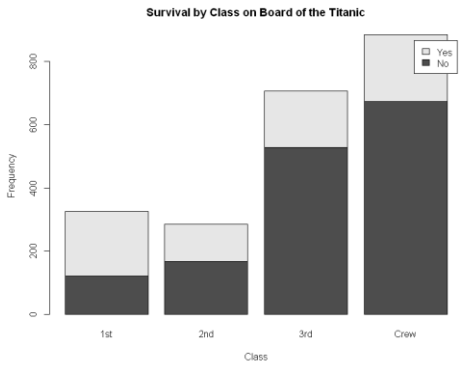


Figure 2: Stacked bar-plot

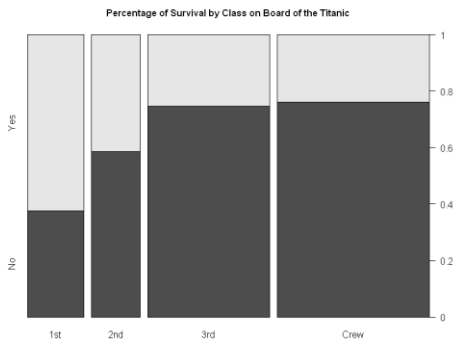


Figure 3: Spinogram

Task 4: Histograms with Kernel density overlays (0.9 points)

Open the old faithful geyser dataset with **`data("faithful", package="datasets")`**. For information on this dataset consult the online help.

Generate for the variable **`eruptions`** time a ***density histogram*** with observation ticks at the bottom of the x-axis and an overlaid ***density curve***. The code generating a plot is: **`lines(density( ))`**). The code generating a plot is:

```
hist(faithful$eruptions,
     breaks = seq(1.5,5.3,by=selBy), probability=TRUE,
     main="Eruption Times of Old Faithful")
rug(jitter(faithful$eruptions))
lines(density(faithful$eruptions, bw=selBw), col="red")
```

[a] Explain what the **`breaks`**-option of the **`hist( )`** function is defining. (0.1 points)

Comment: **`breaks`**-option of the **`hist( )`** function defines the class intervals of the histogram bins. For histograms these need to be of equal width. The **`seq( )`** function generates equal width intervals with a given lower and upper bound.

[b] Explain why the option **`probability=TRUE`** needs to be set to true in order for the kernel density curve to display properly on top of the histogram. (0.1 points)

Comment: If option **`probability=TRUE`** is set, the histogram is displayed at the relative frequency scale [0, 1], which matches the scale of kernel density curve. If **`probability=FALSE`** is set, the histogram will be displayed the bins in absolute frequencies. Then the overlaid kernel density curve would need to be multiplied by the total number of counts (rows in the data-frame) to match the histogram.

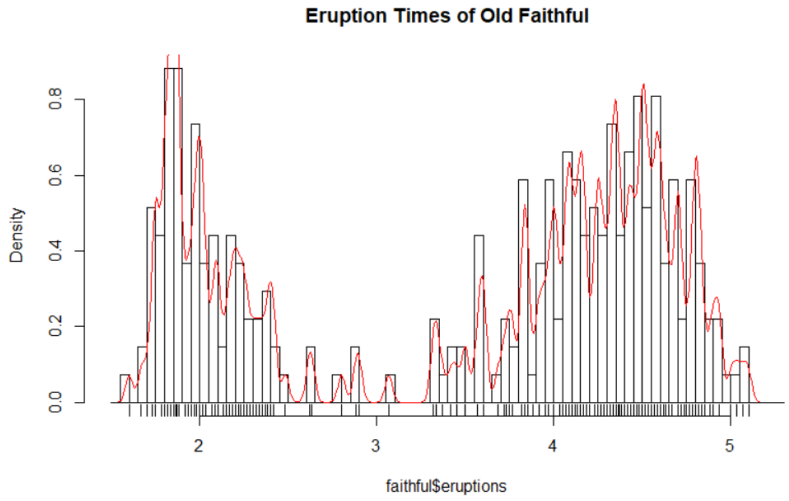
[c] Explain what the option **`bw`** in the **`density( )`** function is doing. See for instance, https://en.wikipedia.org/wiki/Kernel_density_estimation or BBR pp 410-415. (0.1 points)

Comment: Bandwidth is the smoothing parameter of kernels. A large **bw** value will over-smooth the density and mask the local variation in the data. However, a small **bw** value will yield a density driven by individual observations and it becomes granular. It will be difficult to interpret the overall pattern in the distribution.

One can view both the histogram and the kernel density curve as a model for an underlying population that is derived from the available sample data. The aim is to draw both in a way that they best reflect the distribution of the underlying but unknown population.

[d] Generate and show the histogram-density plot with the parameters **selBy=0.05** and **selBw=0.02**. What is wrong with the selected set of parameters? (0.2 points)

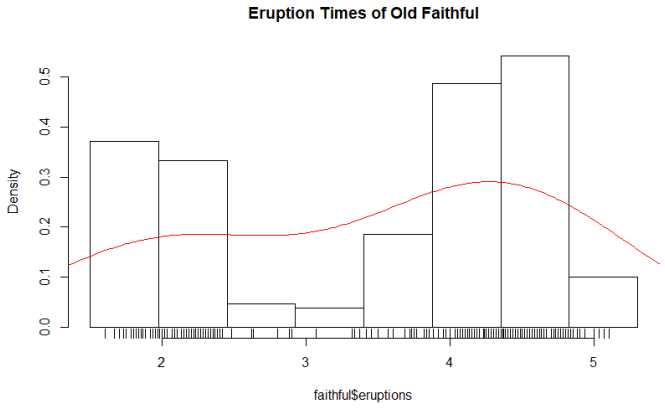
```
selBy=0.05
selBw=0.02
hist(faithful$eruptions,
     breaks = seq(1.5,5.3,by=selBy), probability=T,
     main="Eruption Times of Old Faithful")
rug(jitter(faithful$eruptions))
lines(density(faithful$eruptions, bw=selBw), col="red")
```



Comment: This histograms (selBy=0.05 and selBw=0.02) provides too much details due to random sampling variations, which makes it difficult to detect the overall characteristic of the distribution.

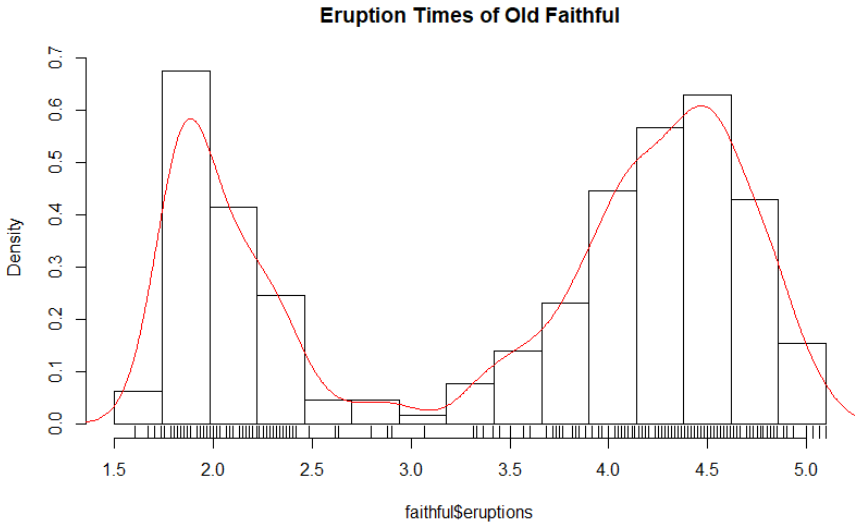
[e] Generate and show the histogram-density plot with the parameters **selBy=0.475** and **selBw=0.8**. What is wrong with the selected set of parameters? (0.2 points)

Comment: This histogram (selBy=0.475 and selBw=0.8) over-smooths the density and mask relevant variations in the data.



[f] Find a set of parameters **`selBy`** and **`selBw`** by experimentation that informatively displays the underlying distribution of the eruption times. Show your final histogram-density plot and ***justify*** your selection of parameters. (0.2 points)

```
selBy=0.25
selBw=0.12
hist(faithful$eruptions,
     breaks = seq(1.6,5.1,by=selBy), probability=TRUE,
     main="Waiting Times inbetween Old Faithful's Eruptions")
rug(jitter(faithful$eruptions))
lines(density(faithful$eruptions, bw=selBw), col="red")
```



Comment: Tasks [d] and [e] suggest that the appropriate values for selBy and selBw should be between 0.05 and 0.475, 0.02 and 0.8 respectively. The histogram above (selBy =0.25 and selBw =0.12) better depicts the distribution of data in the underlying population.