Assignment 3: Data Visualization

Due 5 pm, Friday October 6

1. Stem and leaf plots. Install the package `aplpack`; the function `stem.leaf(...)` from that package will be used to construct the displays. You will work with the following data:

```
set.seed(1234567)
x30 <- rnorm(30)
x100 <- rnorm(100)
x200 <- rnorm(200)
x500 <- rnorm(500)
x1000 <- rnorm(1000)
```

   Stem and leaf plots are then generated as follows:

```
library(aplpack)
stem.leaf(x30)
stem.leaf(x100)
stem.leaf(x200)
stem.leaf(x500)
stem.leaf(x1000)
```

   a. **(3 marks)** What characteristics of the data can you see (or easily determine) from the stem and leaf display? DO NOT hand these displays in!

   b. **(2 marks)** How does the stem and leaf adapt to increasing sample size? How would the stem and leaf display have to be displayed for a sample of $n = 10,000$ and what would be the problems, if any, with that?

2. Quantile functions. Suppose a continuous random variate X has a strictly increasing cumulative distribution function $F_X(x)$, a continuous density $f_X(x)$, and a quantile function $Q_X(p)$.

   a. **(2 marks)** Suppose $U \sim U(0,1)$. Define a random variate $Y = Q_X(U)$. Prove that $Pr(Y \leq a) = F_X(a)$ for any value of a, and hence that Y has the same distribution as does X.

   b. **(4 marks)** Let $Y = aX + b$ for some constants $a > 0$ and b. Prove that the quantile function $Q_Y(p)$ for Y is related to that of X as

$$Q_Y(p) = aQ_X(p) + b.$$

   c. Suppose X is restricted to take only positive values (i.e. $F_X(0) = 0$).

      i. **(2 marks)** Let $Y = \log_a(X) + b$ for some a and b with $a > 1$. Prove that $Q_Y(p) = \log_a(Q_X(p)) + b$

      ii. **(4 marks)** Let $Z = \log_c(X) + d$ for some c and d with $c > 1$. Derive the mathematical relationship between $Q_Z(p)$ and $Q_Y(p)$. What would the parametric curve $(Q_Y(p), Q_Z(p))$ look like for $p \in (0,1)$?

3. Boxplots. The values used to create a boxplot are based on an underlying Gaussian (or Normal) distribution. In this question, you will explore the choices of these values. In `R` the function `qnorm(p)` returns the quantile (i.e. $z = Q(p)$) of a standard normal distribution that corresponds to the cumulative probability `p`. Similarly, `pnorm(z)` returns the value of the cumulative distribution (i.e. $p = F(z)$) for a standard normal distribution at z.

   a. **(1 mark)** Using these functions as appropriate, what is the interquartile range for standard normal?

   b. **(2 marks)** Recall the definition of the upper and lower fences for a box plot,

$$\text{upper fence} = Q3 + c \times IQR$$

$$\text{lower fence} = Q1 - c \times IQR$$

   where $c = 1.5$. Applying these to the $N(0,1)$ distribution, what would be the theoretical values of the lower and upper fences?

c. **(2 marks)** Having just determined the numerical values of the theoretical upper and lower fences, determine the probability that a $N(0,1)$ random variate, say Z, lies outside of one of these fences (i.e. either larger than the upper fence **or** lower than the lower fence)? That is, determine the numerical value of

$$p = Pr((Z < \text{lower fence}) \textbf{ or } (Z > \text{upper fence}))$$

d. **(3 marks)** Suppose that in the previous part of this question, you found the numerical value of p. In a sample of size n from $N(0,1)$, what is the expected number, m say, of values to lie outside the theoretical fences? What is the value of m when $n = 50$?

e. For the standard boxplot c (the constant multiplier of the IQR) is taken to be $c = 1.5$. Suppose we wish to have c change with the size n of the sample. Recall from above that m is the expected number of values in a sample of size n which will lie outside the theoretical fences.

i. **(2 marks)** Write down an expression for the number m as a function of c and n.

ii. **(2 marks)** Using this expression, show how c can be written as a function of m and n.

iii. **(3 marks)** Write a function `getc <- function(m, n) { ... }`, hand it in. Use your function to determine c when $m = 0.35$ for $n = 50, 100, 1000, 10000$.

4. In `R` there are functions that allow calculation of the density (or probability mass) function $f(x)$, the cumulative distribution function $F(x)$, and the quantile function $Q_X(p)$; there are also functions that will generate pseudo-random observations for each distribution. For example for a $N(0,1)$ distribution, the functions are `dnorm(...)`, `pnorm(...)`, `qnorm(...)`, and `rnorm(...)` respectively. To see all of the distributions for which these functions are built-in see `help("distributions")`.

In this question, you will be generating pseudo-random numbers from three different distributions, and four different sample sizes n:

- Gaussian or $N(0,1)$, Student (3) or t_3, and the χ^2_3 distribution.
- $n \in \{50, 100, 1000, 10000\}$

The goal is to compare different visualizations across distributions and to assess the effect of increasing sample size.

Note: So that we will all be looking at the same pictures, we will set a "seed" for the pseudo-random number generation. Be sure to set the seed as shown in each case below.

a. **(3 marks)** Complete (and hand in) the following code to generate the data that we will be considering

```
set.seed(314159)
# The normal data
z50 <- rnorm(...)
z100 <- rnorm(...)
z1000 <- rnorm(...)
z10000 <- rnorm(...)
zlims <- extendrange(c(z50, z100, z1000, z10000))

# The student t (3) data
t50 <- rt(...)
t100 <- rt(...)
t1000 <- rt(...)
t10000 <- rt(...)
tlims <- extendrange(c(t50, t100, t1000, t10000))

# The Chi-squared (3) data
c50 <- rchisq(...)
c100 <- rchisq(...)
c1000 <- rchisq(...)
c10000 <- rchisq(...)
clims <- extendrange(c(c50, c100, c1000, c10000))
```

You will be using these data to answer the remaining parts of this question.

b. For each of the following arrange the corresponding visualizations of the underlying densities in a 2×2 array (e.g. via `savePar <- par(mfrow=c(2,2))`. Each plot in any given array should share the same data limits, the same underlying distribution, and be labelled according to the distribution that generated the sample, and the size of that sample. For each display type (i.e. quantile plot, boxplot, etc.) there should be three arrays (one for each generating distribution) where only the sample size n varies within array.

   Fill all regions with "grey50".

   For each array, comment on how the quality of the display changes as n increases.

   i. **(4 marks)** quantile plots. Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

   ii. **(4 marks)** boxplots. Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

   iii. **(4 marks)** histograms Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

   iv. **(5 marks)** density plots Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

5. Have a look at the video https://youtu.be/HEheh1BH34Q which compares the sizes of various astronomical bodies. The video tries to give some sense of the comparative size of these bodies.

   In the `R` code folder there is a file called `stars.R` that contains measurements of the radius in kilometres of several astronomical bodies in our solar system (in a vector called `solarSystemRadii`) and of the radius of many stars as measured in numbers of solar radii (in a vector called `starRadii`). Load this file into `R` (use `source("directory/stars.R")` with the `directory` changed to wherever you put the file `stars.R`). For example:

```
oldDir <- getwd()
newDir <- "/Users/rwoldford/Documents/Admin/courses/Data\ Visualization/Assignments/Fall17/Assignment\
setwd(newDir)
source("../R/stars.R")
setwd(oldDir)
# Have a look at the contents of each
head(solarSystemRadii)
```

```
##         radius
## Sun     696000
## Jupiter  69911
## Saturn   58232
## Uranus   25362
## Neptune  24622
## Earth     6371
```

```
head(starRadii)
```

```
##                    radius
## binarystarvvcephei   1900
## v354cephei           1520
## mucephei             1420
## kycygni              1420
## v509cassiopeiae       900
## v838monocerotis      1570
```

   a. In the video, the relative size of the planets is encoded in at least three ways.

i. **(3 marks)** Name them.

ii. **(3 marks)** According to Stevens's law, which encoding is most reliably encoded? Which least? (Justify your answers by appeal to this law.)

iii. **(1 mark)** How might position along a common scale have been used instead?

b. For the `solarSystemRadii` data:

Here you are going to draw the various astronomical bodies in our solar system in much the same way as they appear in that video. You will be making use of the `grid` package (as seen in previous assignments).

You will need to recall a few things about `grid`. As before, you will just be drawing circles to represent the various bodies so the function `gridcircle(..)` will be used. Remember that (so far anyway) that you are always drawing within a [0,1] rectangle do that everything you draw will need to be translated to this range (e.g. dividing all radiii by the maximum diameter will allow all circles will fit within the unit square when centred at (0.5, 0.5).

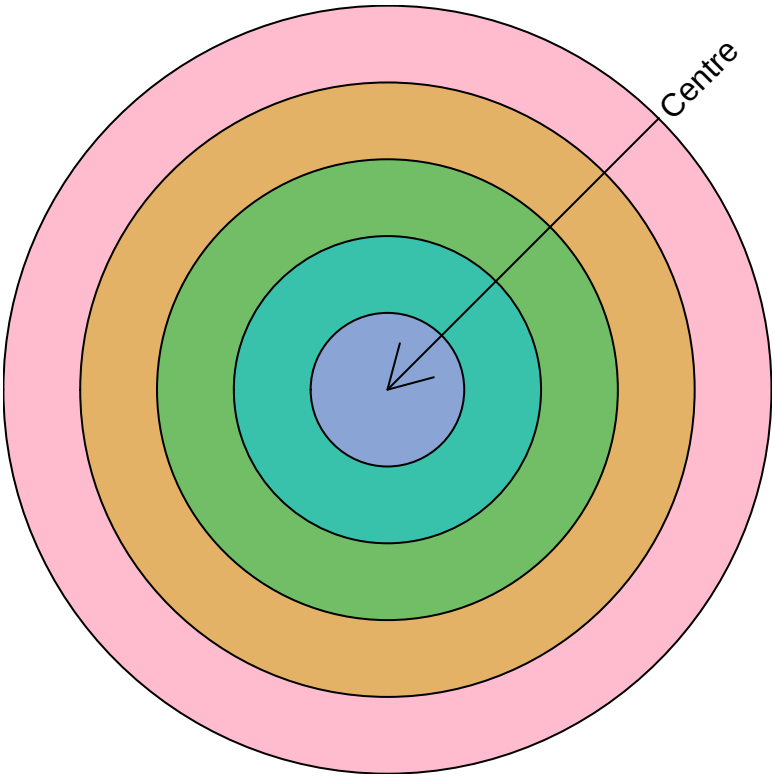
You will also need a palette of colours, one for each body. How to construct a palette of was described during an earlier lecture on numbers and made available on the course web page.

The code below should give you some idea of what is possible. (See help(...) on any function to explore more parameter choices.)

```
library(grid)
library(colorspace)
cols <- rainbow_hcl(n=5, c=100)  # 5 different colours having chroma = 100

grid.newpage()
for (i in 1:5) {
  grid.circle(x=0.5,
      y=0.5,
      r = (6-i)/10,
      gp=gpar(fill=adjustcolor(cols[i], alpha.f = 0.5))
  )
}
# Now add a label and an arrow
xcentre <- 0.5
ycentre <- 0.5
degrees <- 45
radians <- pi * degrees / 180
arrowLength <- 0.5
xarrowFrom <- xcentre + arrowLength * cos(radians)
yarrowFrom <- ycentre + arrowLength * sin(radians)
xarrowTo <- xcentre
yarrowTo <- ycentre
# draw arrow
grid.lines(x=c(xarrowFrom, xarrowTo), y=c(yarrowFrom, yarrowTo),
   arrow=arrow(),
   gp = gpar(col="black", lwd=1, lty = 1))

delta <- 0.01
xText <- xarrowFrom + delta * cos(radians)
yText <- yarrowFrom + delta * sin(radians)
grid.text("Centre", x= xText, y= yText,
  just="left", rot = degrees,
  gp = gpar(col="black"))
```



These functions (and the `grid` package) are to be used to address the following questions.

i. **(5 marks)** Represent each body by a circle whose radius is proportional to the radius in kilometres of that body (use the radius of the sun as the maximum possible radius in the display).

Align the circles so that they are all centred on (0.5, 0.5). Label each of the three largest bodies (excluding the sun) on the plot. Show your code and the picture produced.

IMPORTANT: you need to maintain an aspect ratio of 1 so begin your `r` code chunks in `Rmarkdown` with something like `{r, fig.align="center", fig.width=4, fig.height=4}`

ii. **(2 marks)** Briefly comment on how easily it is to compare the relative sizes of Uranus to Saturn? Of Saturn to Jupiter? Of Jupiter to the Sun? How about comparing Uranus to the Earth or the Moon?

iii. **(4 marks)** According to Stevens's law, what is the range of values we might expect for the ratio of the areas (smaller to larger) of each of the above comparisons? How do these compare to the ratio of actual areas?

iv. **(5 marks)** Consider now all of the bodies in the solar system **excluding the sun**. Using the `grid` package and `grid.circle(...)` etc. lay out all of the remaining bodies from the smallest at the left to the largest at the right with their centres all at $y = 0.5$ but locate them so that they do not overlap. Have the radius of each circles be proportional to the true radius of that body. Mark the Earth, Uranus, Saturn, Jupiter on the plot.

Show your code and your output. Some functions you might find useful are `order(...)` to determine the order of values and possibly `%in%` as in `"foo" %in% c("foo", "bar")` will return `TRUE` if the string `"foo"` can be found in the vector `c("foo", "bar")`; the latter may be helpful in deciding whether to label or not.

c. For the `stars` data,

i. **(3 marks)** Construct a quantile plot of the radii of the `stars`. Describe whatever patterns you see in the data.

ii. **(3 marks)** Construct a quantile plot of the volume of the `stars`. (Recall: the volume of a sphere is $\frac{4}{3}\pi r^3$.) Describe whatever patterns you see in the data.

iii. **(2 marks)** How do these two summaries differ?

iv. **(3 marks)** Construct a quantile plot of the base 10 logarithms of the stellar radii. Describe whatever patterns you see in the data.

v. **(4 marks)** Again construct a quantile plot of the logarithms of the stellar radii **but** this time use base 2 logarithms. In the same plot, overlay the quantiles for the base 10 logarithms. Explain how and why the two sets of quantiles differ.

6. Kernel density estimation.

a. **(4 marks)** Consider the general ASH estimate (non-naive ASH) which looks like

$$\widehat{f}(x,m) \quad = \quad \tfrac{1}{nh}\sum_{|i|<m} w_m(i)v_{k+i}$$

for $x \in B_k$. Here the weights $w_m(i) \geq 0$ and the intervals $B_j = [b_j, b_{j+1})$ indexed by $j = 0, \pm 1, \pm 2, \pm 3, \ldots$ partition the entire real line.

The intervals are each of width $(b_{j+1} - b_j) = \frac{h}{m}$ and v_j is the number of xs in B_j. The total sample size is $n = \sum_{|j|=0}^{\infty} v_j$.

Prove that if $\sum_{|i|<m} w_m(i) = m$ then $\int \widehat{f}(x,m)dx = 1$.

b. **Graduate students (bonus undergraduates)** Recall from the notes the general results on the bias (up to $O(h^4)$) and variance (up to order $O(\frac{h}{n})$) of a "simplified kernel" density estimator.

i. **(4 marks)** Prove that the following kernel is a "simplified kernel" (in the sense of the notes).

$$K(w) = \begin{cases} \frac{15}{32}(1-w^2)(3-7w^2) & w \in [-1,1] \\ \\ 0 & \text{elsewhere.} \end{cases}$$

ii. **(8 marks)** Determine the (approximate) mean squared error of $\tilde{f}_K(x)$ for the above kernel K and for arbitrary $f(x)$.

iii. **(5 marks)** Determine the case for the above K when the true underlying density $f(x)$ is $N(0,1)$.