Assignment 3

STAT 442: Data Visualization

Azoacha Forcheh, 20558994

1. **Stem and leaf plots.**

   (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!

   - Plot 1: The density of the data seems to take on the shape of a somewhat flat bell curve centred at around $-0.4 - 0$, with very little spread. It appears to be trimodal, with all the modes appearing between -1 and 1.
   - Plot 2: The density of the data seems to take on the shape of a steep bell curve centred at around $-0.4 - 0$, with an average amount of spread. It appears to be unimodal at around $-0.4 - 0$.
   - Plot 3: The density of the data seems to take on the shape of the bell curve of a standard normal distribution, centred at around $-0.4 - 0$, with somewhat even spread on either side of the peak. It appears to be unimodal at around $-0.4 - 0$. In addition, the value of -3.4 seems to be an outlier.
   - Plot 4: The density of the data seems to take on the shape of a very steep bell curve, centred at around $0 - 0.4$, with very little spread and very short tails. It appears to be unimodal at around $-0.4 - 0$. In addition, the value of -3.4 seems to be an outlier.
   - Plot 5: The data appears to be bimodal, with the modes at around $-0.3 - -0.2$ and $0.2 - 0.3$. In addition, the values of $-3.1, -3.0$ seems to be an outlier.

   (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?

   The scale of the plot decreases as n increases. With these plots, the number of parts of each stem was reduced from 5 (i.e. the possible digits of the leaves were split into 2 groups of 5) for $n \in \{30, 100, 200\}$ to 2 for $n \in \{500, 1000\}$ (i.e. the possible digits of the leaves were split into 5 groups of 2).

   In addition, for the stems not containing 8 and 9, the label for the stems change from the value of the digit for (e.g. -1.) to the first letter of the digits of the group ('t' for 2 and 3, 'f' for 4 and 5, 's' for 6 and 7).

   According to the documentation for **stem.leaf**, the possible number of parts that the stem can be separated into is one of 1, 2, or 5. So for $n = 10,000$, each time will be separated into 5 groups of 2 digits. As a result, the stems of the plot will be extremely long as n is extremely high. This will make it easier to determine the general pattern of the data as it will be spread more horizontally instead of vertically, making the shape of the density clearer. However, the tradeoff is that the detailed pattern becomes much harder to interpret due to the high number of data points and the comparatively small number of parts in each stem.

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.

   Note that $F_X(x)$ is continuous and strictly increasing. Hence, $Q_X(x) = F_X^{-1}(x) \implies F_X(x) = Q_X^{-1}(x)$. So:

$$\begin{aligned} Pr(Y \leq a) &= Pr(Q_X(U) \leq a) \\ &= Pr(U \leq Q_X^{-1}(a)) \\ &= Q_X^{-1}(a) \qquad \text{since } U \sim U(0,1) \\ &= F_X(a) \end{aligned}$$

(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

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

First, note that since $F_X(x)$ is a strictly increasing c.d.f and $Y = aX + b$ for some constants $a > 0$ (so Y increases as X increases) and b, $F_Y(y)$ is also a continuous, strictly continuous c.d.f. We use this to prove the given statement as follows:

$$\begin{aligned}
Pr(X \leq Q_X(p)) &= F_X(Q_X(p)) \\
&= F_X(F_X^{-1}(p)) && \text{since } F_X \text{ is continous and strictly increasing} \\
&= p \\
p &= Pr(X \leq Q_X(p)) \\
&= Pr(aX + b \leq aQ_X(p) + b) \\
&= Pr(Y \leq aQ_X(p) + b) \\
&= F_Y(aQ_X(p) + b) \\
Q_Y(p) &= Q_Y(F_Y(aQ_X(p) + b)) \\
&= F_Y^{-1}(F_Y(aQ_X(p) + b)) && \text{since } F_Y \text{ is continous and strictly increasing} \\
&= aQ_X(p) + b
\end{aligned}$$

(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 $Q_Y(p) = \log_a(Q_X(p)) + b$.

Note that $\log_a$ is a monotonic, strictly increasing function for any constant $a > 1$. Hence, since $F_X(x)$ is a strictly increasing c.d.f and $Y = \log_a(X) + b$ for some constants $a > 1$ and b, $F_Y(y)$ is also a strictly increasing c.d.f. We use this to prove the given statement as follows:

$$\begin{aligned}
p &= Pr(X \leq Q_X(p)) && \text{proven in (b)} \\
&= Pr(\log_a(X) + b \leq \log_a(Q_X(p)) + b) \\
&= Pr(Y \leq \log_a(Q_X(p)) + b) \\
&= F_Y(\log_a(Q_X(p)) + b) \\
Q_Y(p) &= Q_Y(F_Y(\log_a(Q_X(p)) + b)) \\
&= F_Y^{-1}(F_Y(\log_a(Q_X(p)) + b)) && \text{since } F_Y \text{ is continous and strictly increasing} \\
&= \log_a(Q_X(p)) + b
\end{aligned}$$

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 $p \in (0, 1)$?

Note that

$$Z = \log_c(X) + d \implies Q_Z(p) = \log_c(Q_X(p)) + d \implies Q_X(p) = c^{Q_Z(p)-d}$$

and

$$Q_Y(p) = \log_a(Q_X(p)) + b \implies Q_X(p) = a^{Q_Y(p)-b}$$

Hence, we can derive the mathematical relationship between $Q_Z(p)$ and $Q_Y(p)$ as:

$$\begin{aligned}
a^{Q_Y(p)-b} &= c^{Q_Z(p)-d} \\
Q_Z(p) - d &= \log_c a^{Q_Y(p)-b} \\
&= (Q_Y(p) - b)\log_c a \\
Q_Z(p) &= (log_c a)Q_Y(p) - b\log_c a + d
\end{aligned}$$

For the parametric curve, note that X is restricted to take only positive values (i.e. $F_X(0) = 0$).

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?

```
q3 = qnorm(.75)
q1 = qnorm(.25)
iqr_norm = q3 - q1
```

   By definition, the IQR equals the difference between the 3rd quartile and the 1st. So the IQR for the standard normal is 1.3489795, as calculated above.

   (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 = 1.5

# using the calculated values from (a)
upper_fence = q3 + c*iqr_norm
lower_fence = q1 - c*iqr_norm
```

   From the calculations above, the theoretical values of the lower and upper fences -2.697959 and 2.697959 are respectively.

   (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

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

$$\begin{aligned} p &= Pr((Z < \text{lower fence}) \text{ or } (Z > \text{upper fence})) \\ &= Pr(Z < -2.697959) + Pr(Z > 2.697959) \\ &= 1 - Pr(Z \leq 2.697959) + 1 - Pr(Z \leq 2.697959) \\ &= 2 - 2Pr(2.697959) \\ &= 0.006976603 \end{aligned}$$

   , where $2 - 2Pr(2.697959)$ was calculated using the following R code:

```
pr = pnorm(upper_fence)
p = 2*(1- pr)
```

   (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 $n = 50$?

   Note that m/n is the expected proportion of values that would like outside the theoretical fences, which is equivalent to p. Hence, m would equal the sample size multiplied by the probability of a value falling outside of the theoretical values, i.e. $m = pn$.

```
# m when n = 50
m50 = p*50
```

   The value of m when $n = 50$ is 0.3488302.

(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.

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(50)
z100 <- rnorm(100)
z1000 <- rnorm(1000)
z10000 <- rnorm(10000)
zlims <- extendrange(c(z50, z100, z1000, z10000))

# The student t (3) data
t50 <- rt(50, 3)
t100 <- rt(100, 3)
t1000 <- rt(1000, 3)
t10000 <- rt(10000, 3)
tlims <- extendrange(c(t50, t100, t1000, t10000))

# The Chi-squared (3) data
c50 <- rchisq(50, 3)
c100 <- rchisq(100, 3)
c1000 <- rchisq(1000, 3)
c10000 <- rchisq(10000, 3)
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.

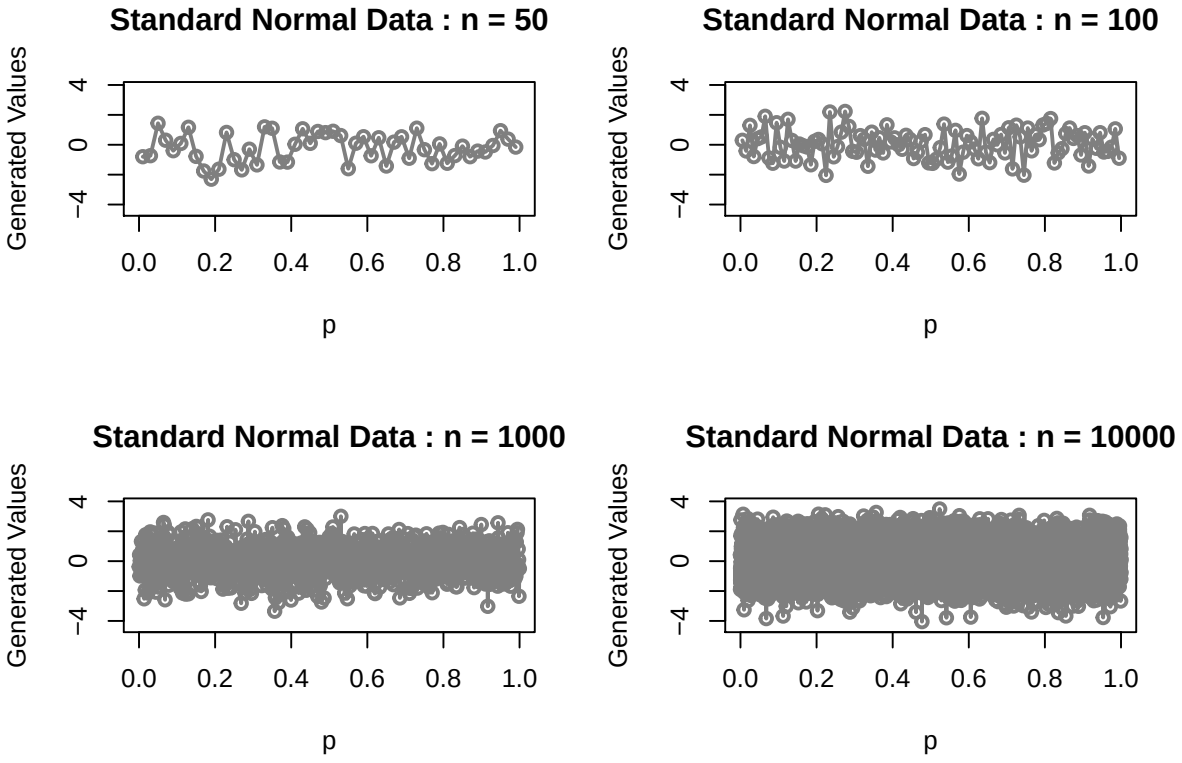
Fill all regions with "grey50".

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.

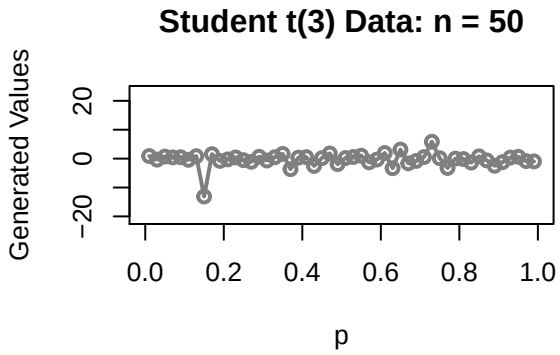
```
# For general use
sample_sizes = c(50, 100, 1000, 10000)
data_norm = list(z50, z100, z1000, z10000)
data_t3 = list(t50, t100, t1000, t10000)
data_chi = list(c50, c100, c1000, c10000)
```

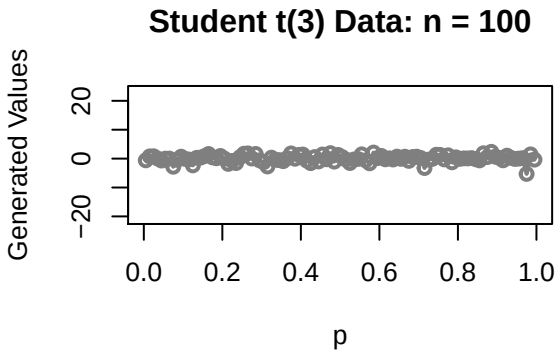
```
# normal distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  p = ppoints(n)
  title = "Standard Normal Data : n = "

  plot(x = p, y=data_norm[[i]],
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Generated Values",
      main=paste0(title, n),
      xlim =c(0,1), ylim =zlims)
}
```

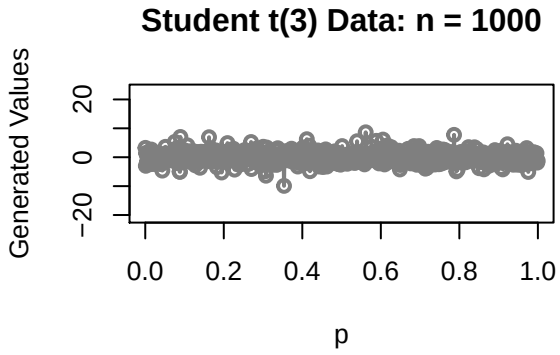


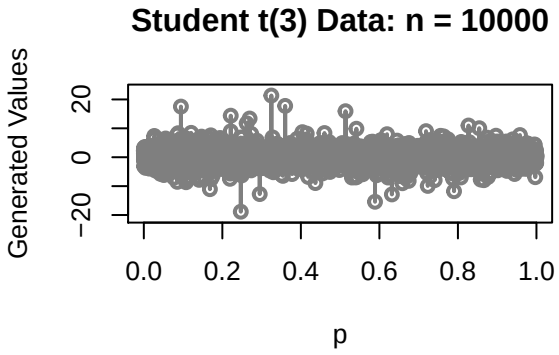

```
# student t(3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  p = ppoints(n)
  title = "Student t(3) Data: n = "

  plot(x = p, y=data_t3[[i]],
```

```
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Generated Values",
      main=paste0(title, n),
      xlim =c(0,1), ylim =tlims)
}
```

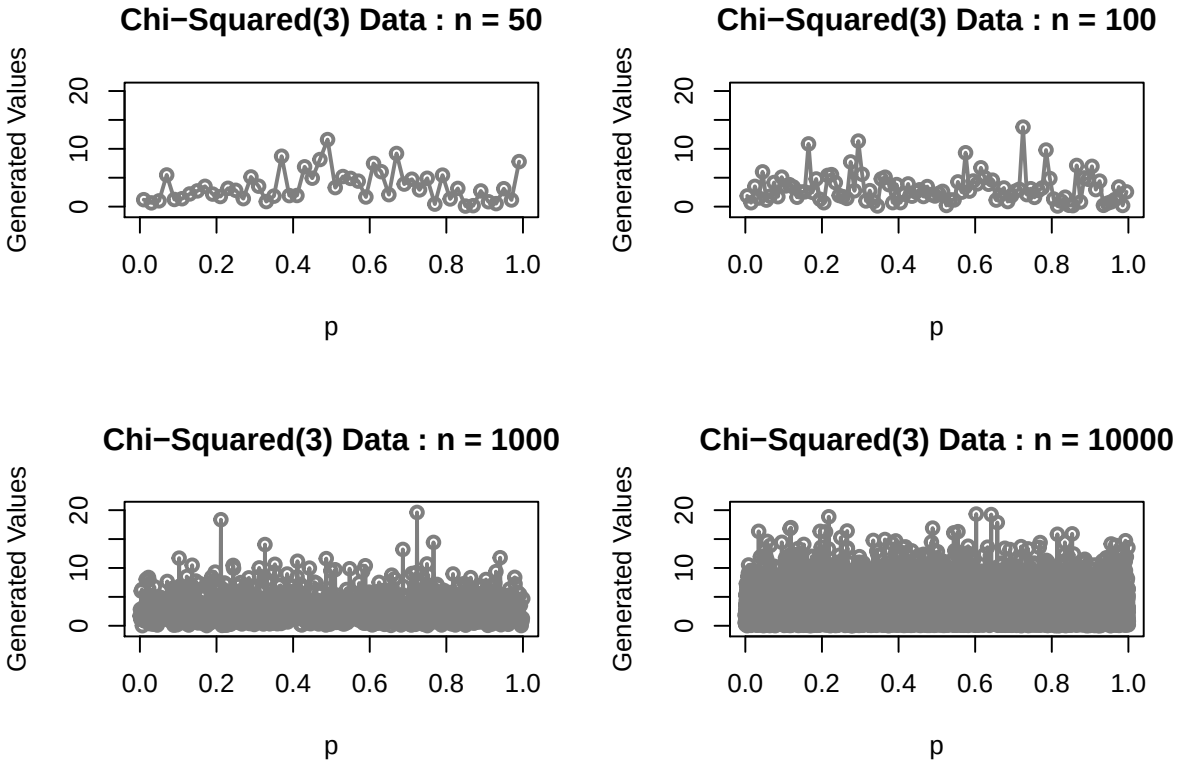






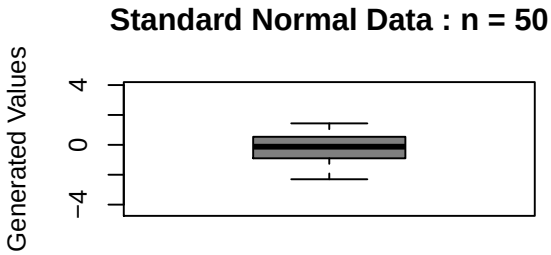


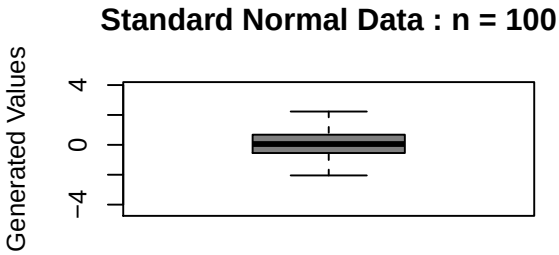

```
# Chi-Squared (3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  p = ppoints(n)
  title = "Chi-Squared(3) Data : n = "

  plot(x = p, y=data_chi[[i]],
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Generated Values",
      main=paste0(title, n),
      xlim =c(0,1), ylim =clims)
}
```

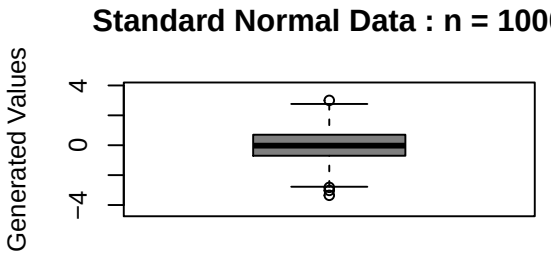


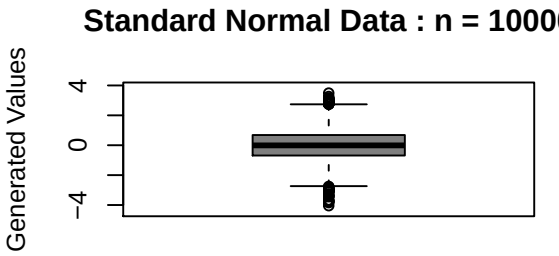
ii. **(4 marks)** boxplots. Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

```
# normal distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Standard Normal Data : n = "

  boxplot(x=data_norm[[i]], col="grey50",
      ylab="Generated Values",
      main=paste0(title, n),
      ylim=zlims)
}
```

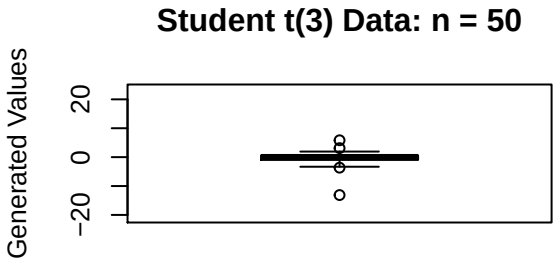


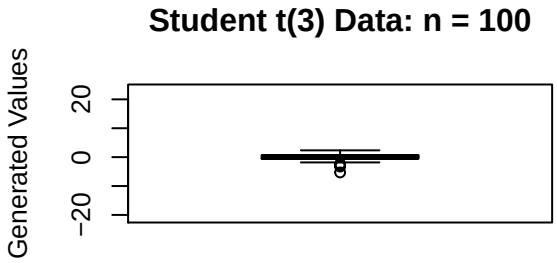


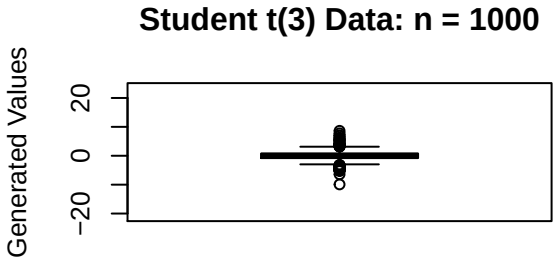


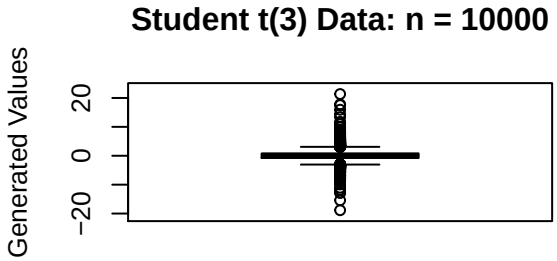



```r
# student t(3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Student t(3) Data: n = "

  boxplot(x=data_t3[[i]], col="grey50",
      ylab="Generated Values",
      main=paste0(title, n),
      ylim=tlims)
}
```

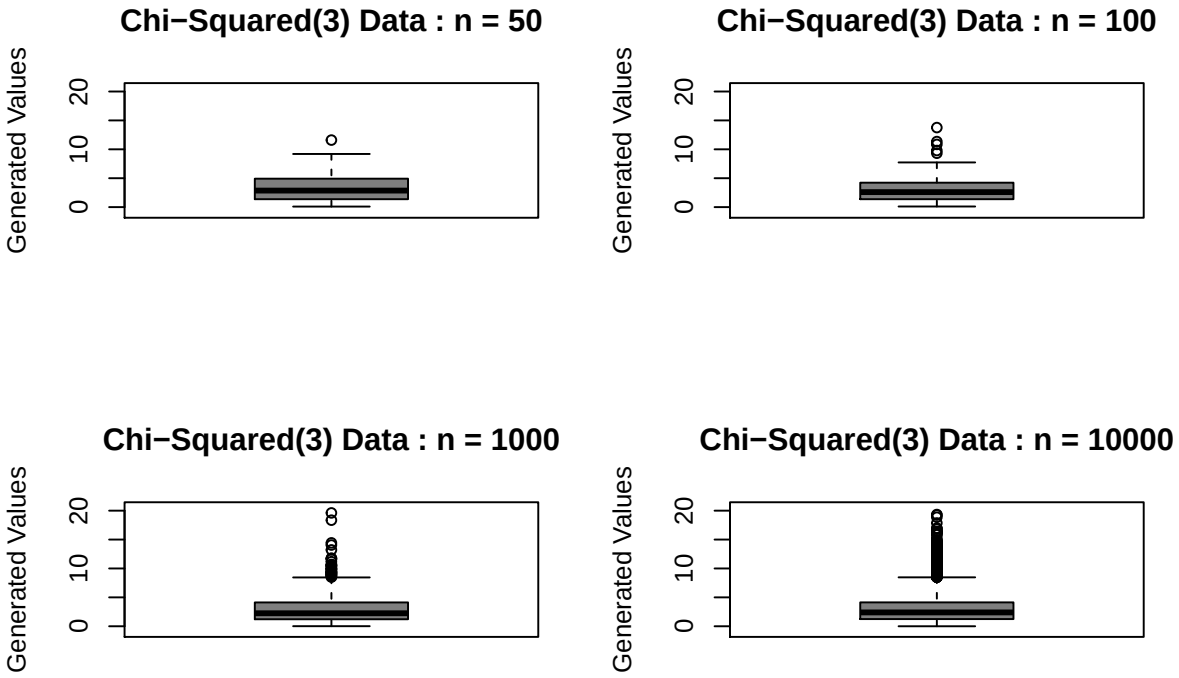








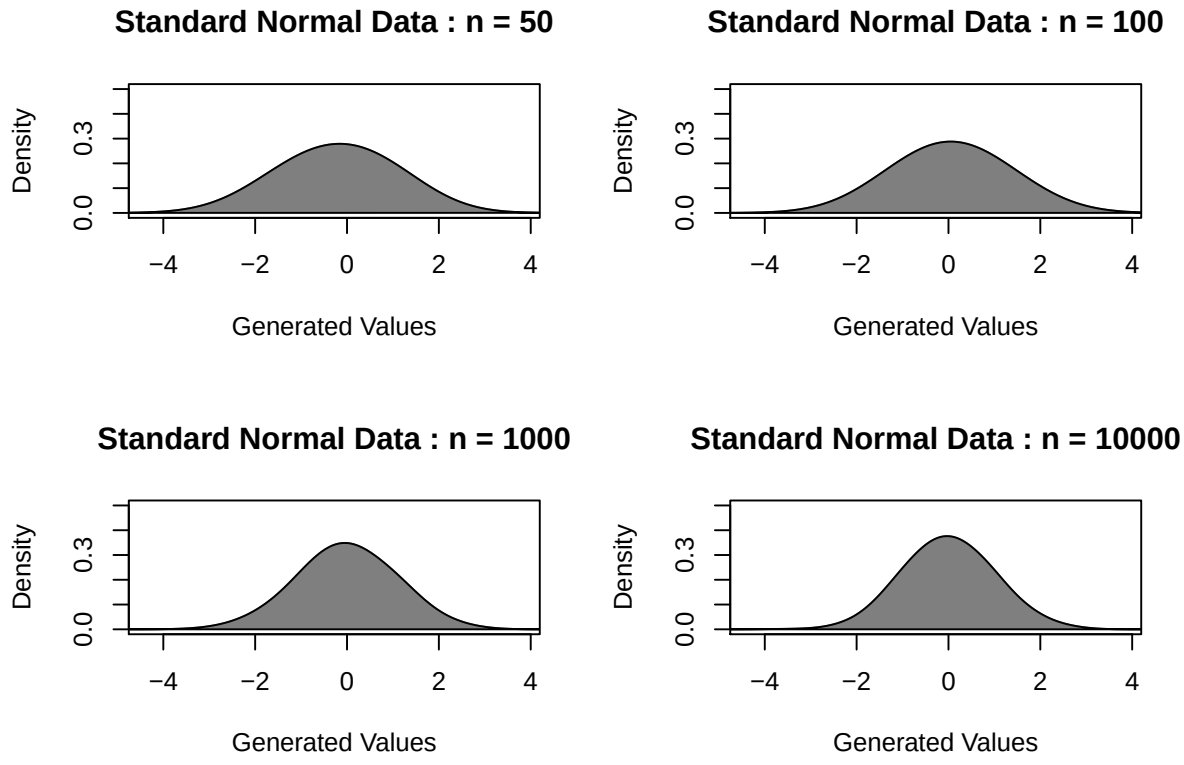
```r
# Chi-Squared (3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Chi-Squared(3) Data : n = "

  boxplot(x=data_chi[[i]], col="grey50",
      ylab="Generated Values",
      main=paste0(title, n),
      ylim=clims)
}
```



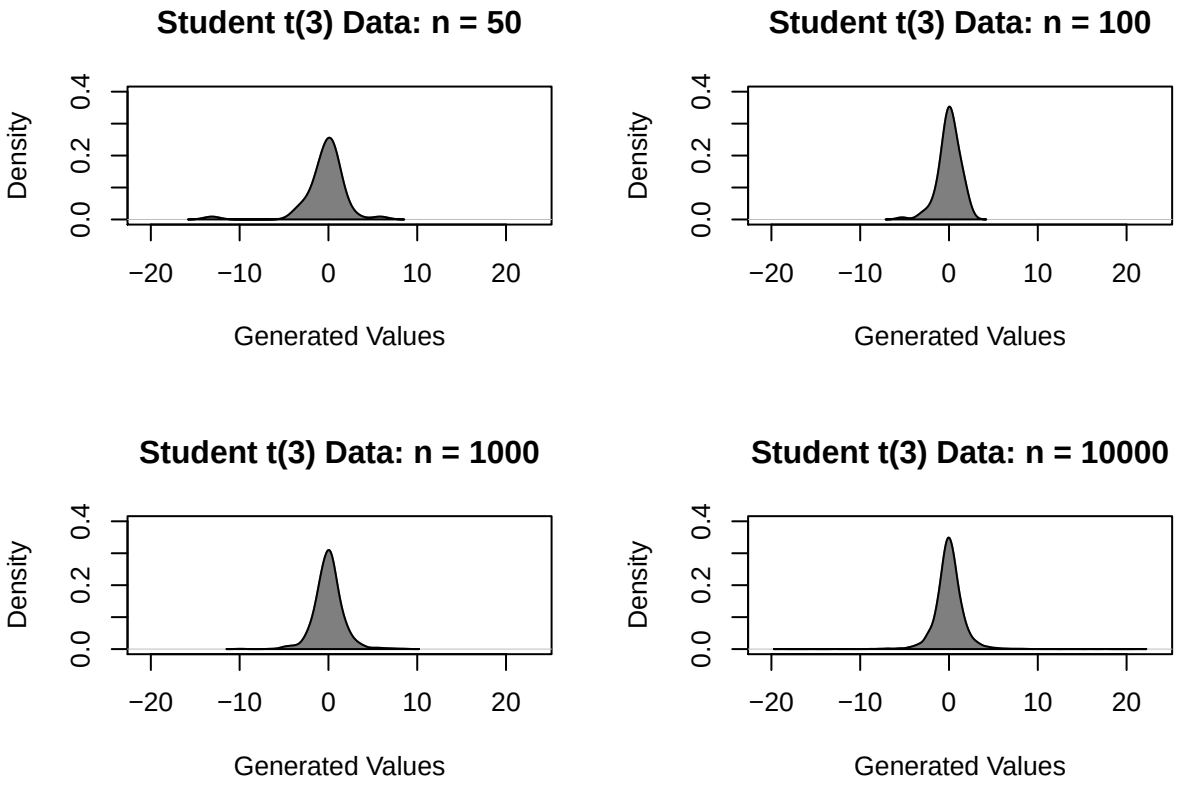
iii. **(4 marks)** histograms Produce the three arrays of changing n, one for each distribution ($N(0,1)$, t_3, and χ^2_3). Submit each displayed array and comment on how the quality of the display changes as n increases.

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.

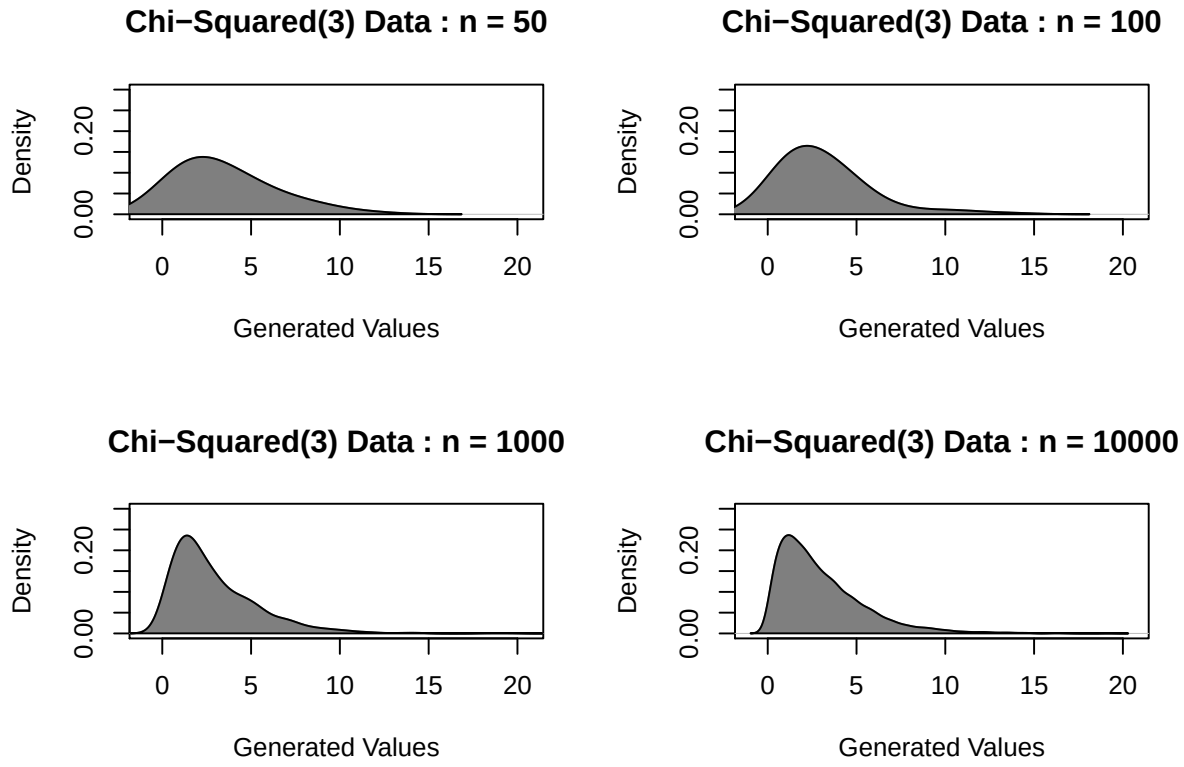
```r
# normal distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Standard Normal Data : n = "
  dens = density(data_norm[[i]], bw="SJ", adjust = 2)

  plot(dens, col="grey50",
       xlab="Generated Values",
       main=paste0(title, n),
       xlim=zlims, ylim=c(0,0.5))
  polygon(dens, col="grey50")
}
```




```
# student t(3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Student t(3) Data: n = "
  dens = density(data_t3[[i]], bw="SJ", adjust = 2)

  plot(dens, col="grey50",
       xlab="Generated Values",
       main=paste0(title, n),
       xlim=tlims, ylim=c(0,0.4))
  polygon(dens, col="grey50")
}
```




```r
# Chi-Squared (3) distribution array
par(mfrow=c(2,2))

for(i in 1:4) {
  n = sample_sizes[i]
  title = "Chi-Squared(3) Data : n = "
  dens = density(data_chi[[i]], bw="SJ", adjust = 2)

  plot(dens, col="grey50",
       xlab="Generated Values",
       main=paste0(title, n),
       xlim=clims, ylim=c(0,0.3))
  polygon(dens, col="grey50")
}
```



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:

```
source("./stars.R")
# 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.

      Volume, position along a common scale, length

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

According to Steven's law, length is most reliable encoding as it creates the smallest bias when encoding the relative size of the planets, which is due to its higher empirically observed range of values $0.9 \leq \beta \leq 1.1$, making $p(x) \approx cx$. The least reliable encoding is volume, which has the largest bias of the 3 when encoding the relative size of the planets, due to its lower empirically observed range of values $0.5 \leq \beta \leq 0.8$.

iii. **(1 marks)** How might position along a common scale have been used instead? As the bodies being considered were all spheres, the position along a common scale could have been used to display and compare the radii (a measurement of length) of the bodies instead of the volumes, which is only dependent on the radii and is a less accurate encoding according to Steven's law.

(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).

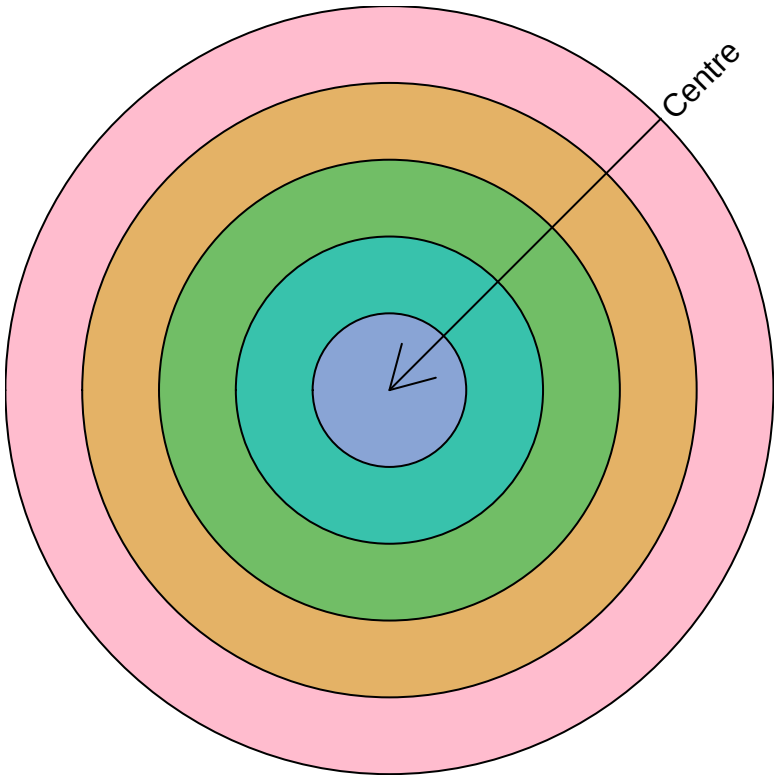
You will need to recall a few things about `grid`. As before, you will just be drawing circles to represent the various bodies so the function `gridcircle(..)` will be used. Remember that (so far anyway) that you are always drawing within a [0,1] rectangle do that everything you draw will need to be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

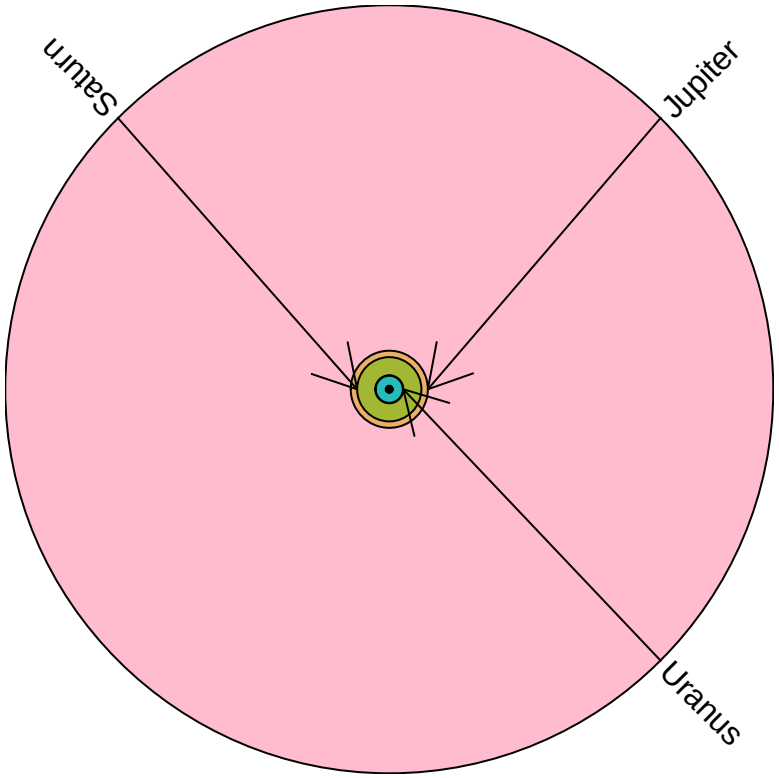
i. **(5 marks)** Represent each body by a circle whose radius is proportional to the radius in kilometres of that body (use the radius of the sun as the maximum possible radius in the display).

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}`

```
library(grid)
library(colorspace)

xcentre = 0.5
ycentre = 0.5
cols = rainbow_hcl(6, c = 100)
n = dim(solarSystemRadii)[1]
max_diam = solarSystemRadii['Sun',]

grid.newpage()
for (i in 1:n) {
  grid.circle(x=xcentre,
              y=ycentre,
              r = (solarSystemRadii[i,]/max_diam)/2,
              gp=gpar(fill=adjustcolor(cols[i], alpha.f = 0.5))
  )
}

degs = c(45, 135, 315)
```

```
for (i in 2:4) {
  degrees = degs[i-1]
  radians = pi * degrees / 180
  arrowLength = 0.5
  xarrowFrom = xcentre + arrowLength * cos(radians)
  yarrowFrom = ycentre + arrowLength * sin(radians)

  if (degrees == 135) {
    xarrowTo = xcentre - (solarSystemRadii[i,]/max_diam)/2
  } else {
    xarrowTo = xcentre + (solarSystemRadii[i,]/max_diam)/2
  }

  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)

  body_name = rownames(solarSystemRadii)[i]
  grid.text(body_name, x= xText, y= yText,
            just="left", rot = degrees,
            gp = gpar(col="black"))
}
```



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?

Because the two circles representing the planets are both relatively large and are overlaid immediately (i.e. with no other circle between the two) over each other, and because the colors of the circles are diverging, it is easy to see that Uranus is much smaller than Saturn as we can still see much of the area of the circle representing Saturn. However, it is difficult to determine by what proportion this difference is as we comparing the two via their areas.

For Saturn and Jupiter, it is once again easy to compare the two planets for the exact reasons mentioned above, with it being easy to determine that Saturn is slightly larger than Jupiter. The same applies for comparing Jupiter to the Sun, though it is harder to make this comparison and most of Jupiter is not visible under the overlaying circles.

Due to the Moon and Earth being significantly smaller than the Sun, they are very difficult to see and compare with any of the other bodies. One can tell that the Moon and Earth are smaller than Uranus, but it is extremely difficult to determine the ratio of difference.

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?

$$\begin{aligned}
\text{Uranus vs. Saturn} &: [(\frac{25362^2\pi}{58232^2\pi})^{0.9}, (\frac{25362^2\pi}{58232^2\pi})^{0.6}] = [0.2239955, 0.3688299] \\
\text{Saturn vs. Jupiter} &: [(\frac{58232^2\pi}{69911^2\pi})^{0.9}, (\frac{58232^2\pi}{69911^2\pi})^{0.6}] = [0.7196298, 0.8030442] \\
\text{Jupiter vs. the Sun} &: [(\frac{69911^2\pi}{696000^2\pi})^{0.9}, (\frac{69911^2\pi}{696000^2\pi})^{0.6}] = [0.01597663, 0.06343421] \\
\text{Earth vs. Uranus} &: [(\frac{6371^2\pi}{25362^2\pi})^{0.9}, (\frac{6371^2\pi}{25362^2\pi})^{0.6}] = [0.08318469, 0.1905588] \\
\text{The Moon vs. Uranus} &: [(\frac{1737.1^2\pi}{25362^2\pi})^{0.9}, (\frac{1737.1^2\pi}{25362^2\pi})^{0.6}] = [0.00796984, 0.0398994]
\end{aligned}$$

In comparison, the actual ratios are:

$$\begin{aligned}
\text{Uranus vs. Saturn} &: 0.1896896 \\
\text{Saturn vs. Jupiter} &: 0.6937969 \\
\text{Jupiter vs. the Sun} &: 0.01008957 \\
\text{Earth vs. Uranus} &: 0.004691186 \\
\text{The Moon vs. Uranus} &: 0.06310274
\end{aligned}$$

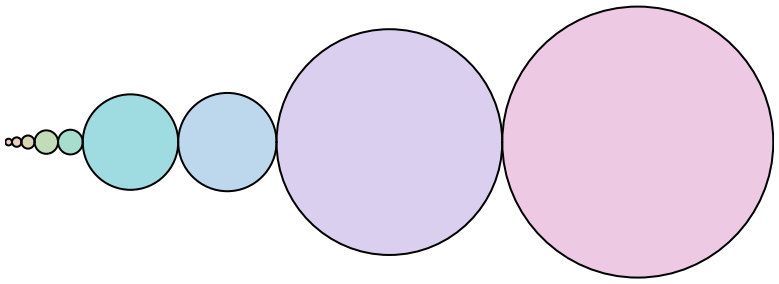
Every one of the ratio of actual areas falls outside and to the left of the calculated ranges for the ratios. The demonstrates the bias and reduced accuracy of using area as a visual encoding of magnitude.

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.

```
cols = rainbow_hcl(n-1)

asc_order = order(solarSystemRadii)
planets = solarSystemRadii[asc_order[! asc_order %in% c(1)],]
planet_names = rownames(solarSystemRadii)[asc_order[! asc_order %in% c(1)]]
total_length = 2*sum(planets)
planetsr = planets/total_length

xcenters = replicate(n-1, 0)
```

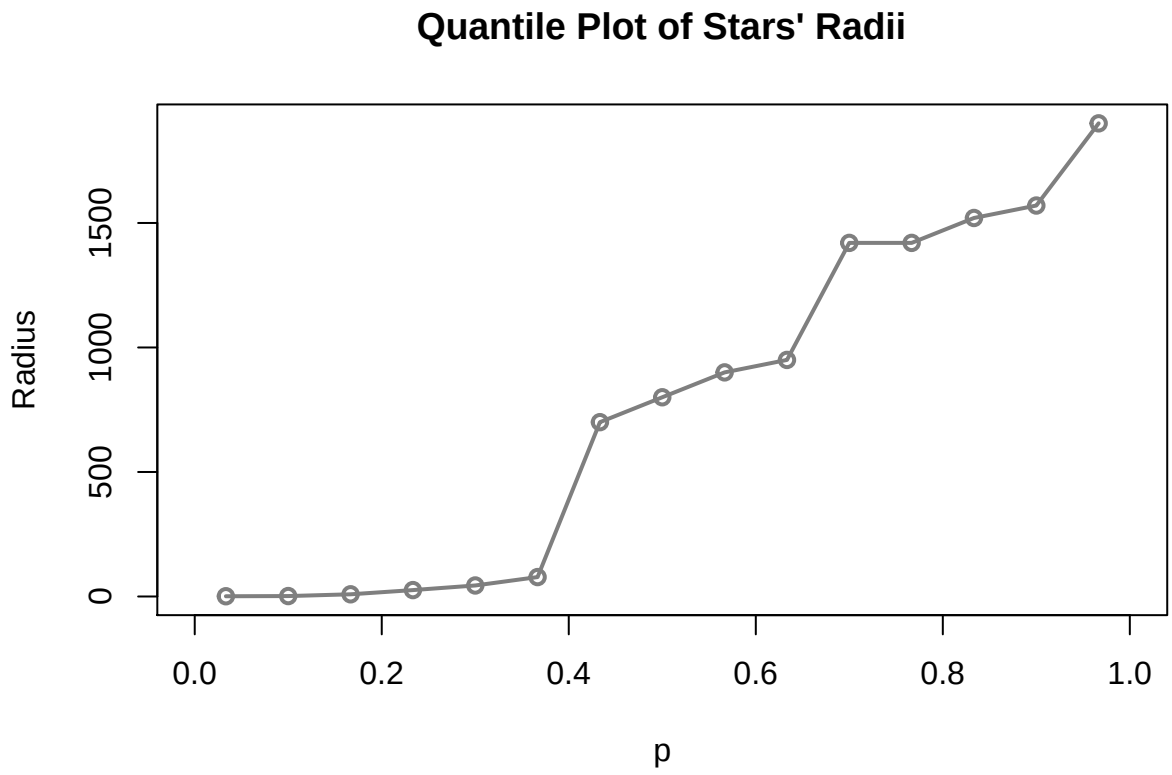
```
for (i in 1:(n-1)) {
  xcenters[i] = (2*sum(planetsr[1:i-1])) + planetsr[i]
}

grid.newpage()
xcenter = 0
ycenter = 0.5

for (i in 1:(n-1)) {
  grid.circle(x=xcenters[i],
              y=ycenter,
              r = planetsr[i],
              gp=gpar(fill=adjustcolor(cols[i], alpha.f = 0.5)))

  body_name = planet_names[i]
  if (body_name %in% c('Earth', 'Uranus', 'Saturn', 'Jupiter')) {
    grid.text(body_name, x=xcenters[i], y=0.09,
              just="left",
              gp = gpar(col="black"))
  }
}
```



(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.

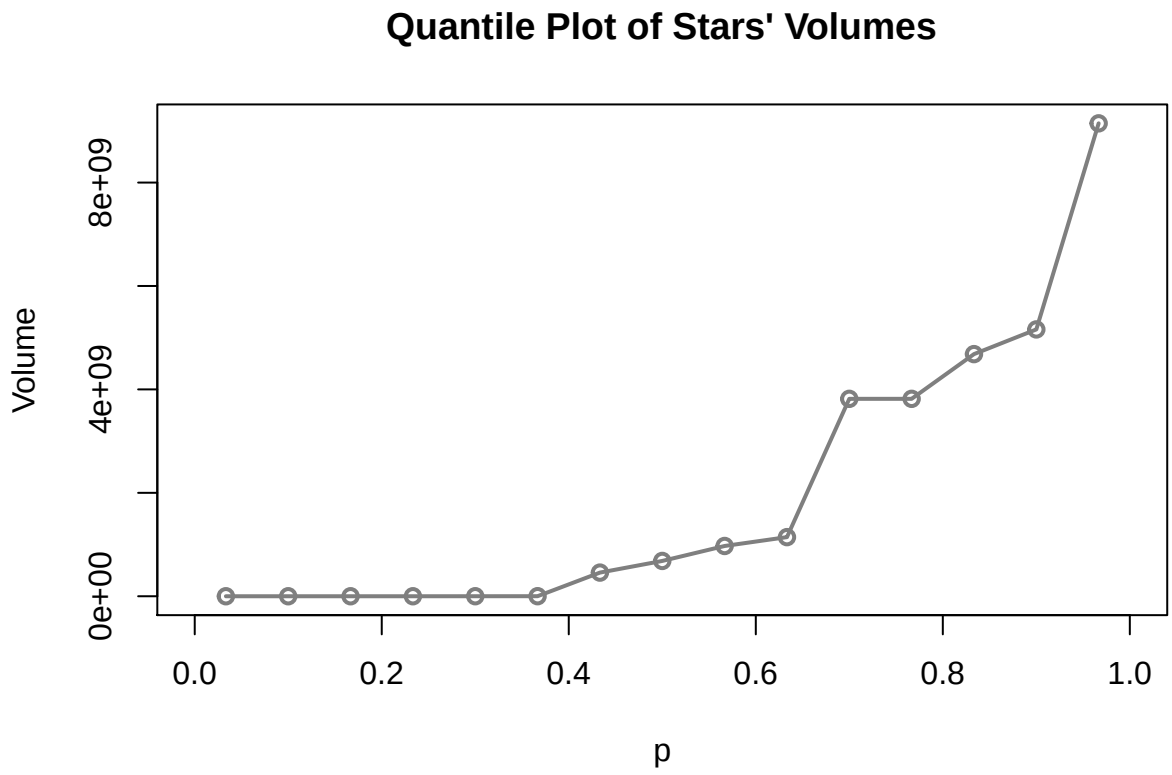
```
n = dim(starRadii)[1]
p = ppoints(n)
starRadii = starRadii[order(starRadii),]
plot(x = p, y=starRadii,
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Radius",
      main="Quantile Plot of Stars' Radii",
      xlim =c(0,1))
```



The data appears to be somewhat discrete with the main values being at 0, $\sim 700 - 1000$, 1500.

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.

```
starVolumes = 4*(starRadii^3)/3
plot(x = p, y=starVolumes,
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Volume",
      main="Quantile Plot of Stars' Volumes",
      xlim =c(0,1))
```

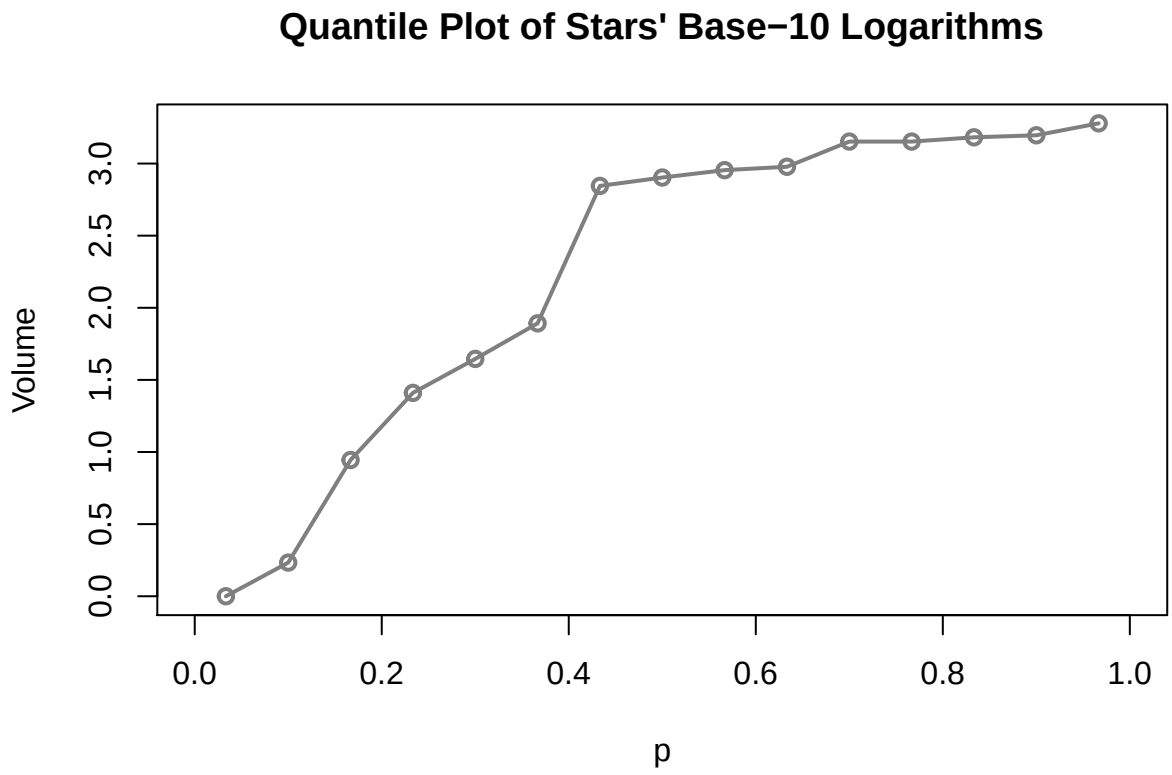


iii. **(2 marks)** How do these two summaries differ?

The plots have similar shapes, but

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

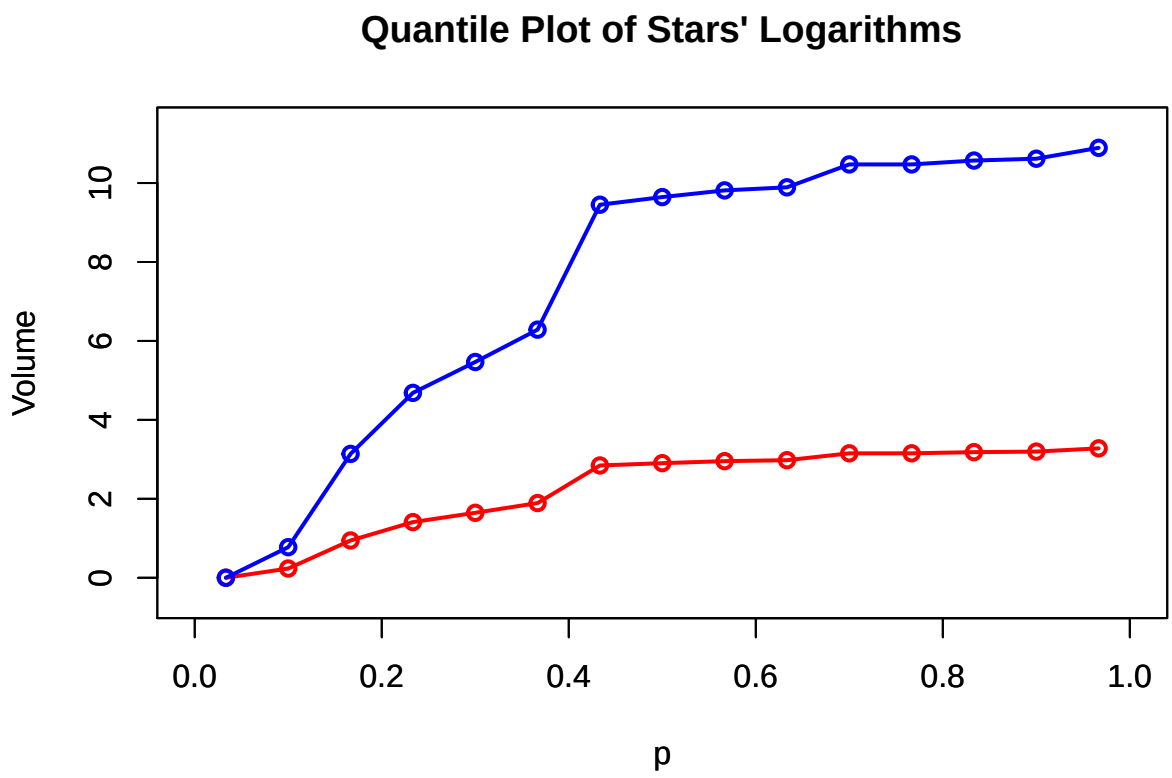
```
starLogs = log(starRadii, base=10)
plot(x = p, y=starLogs,
      type="o", lwd=2, col="grey50",
      xlab="p", ylab="Volume",
      main="Quantile Plot of Stars' Base-10 Logarithms",
      xlim =c(0,1))
```



The data seems to have a decreasing cumulative distribution function as $Q(p)$ and p have an inverse relationship. There is a sharp drop in $Q(p)$ at $p \approx 0.55$.

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.

```
starLog2s = log(starRadii, base=2)
ylims = extendrange(c(starLogs, starLog2s))

plot(x = p, y=starLogs,
      type="o", lwd=2, col="red",
      xlab="p", ylab="Volume",
      main="Quantile Plot of Stars' Logarithms",
      xlim =c(0,1), ylim=ylims)

par(new = TRUE)

plot(x = p, y=starLog2s,
      type="o", lwd=2, col="blue",
      ylab="Volume",
      xlim =c(0,1), ylim=ylims)
```


Quantile Plot of Stars' Logarithms
Volume
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
p