Diabetes data: comparing distributions

13 marks (undergrads) plus potential *8 marks* bonus

21 marks (grads)

Comparing distributions

Download the `diabetes` data from the course website. In that file, there is a dataset on various measurements of 145 patients. Once you load this file into your R session (or equivalently, execute its contents there) there will be a data set called `diabetes`.

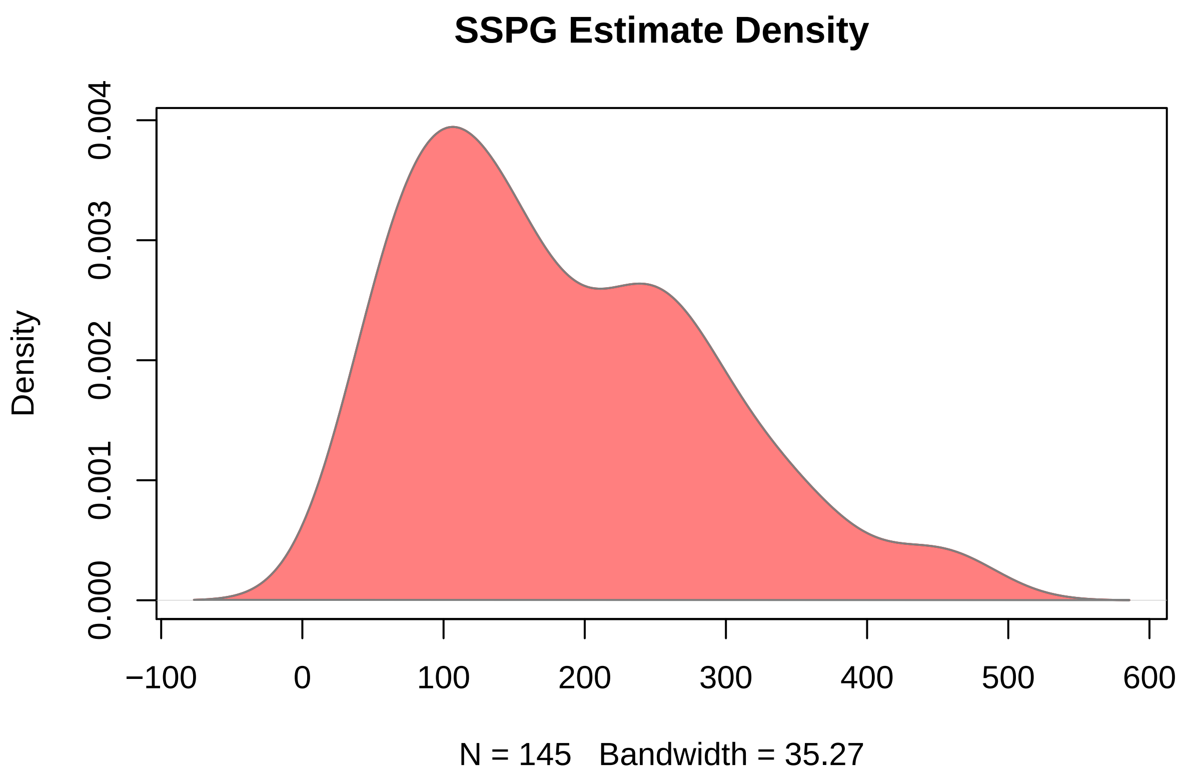
```r
# For example, you could use the source command.
# Here the file is stored in the current directory
load("diabetes.Rda")
# Once loaded the data is available as the data frame `diabetes'
head(diabetes)
```

```
##   PatientNumber RelativeWeight FastingPlasmaGlucose GlucoseArea
## 1             1           0.81                   80         356
## 2             2           0.95                   97         289
## 3             3           0.94                  105         319
## 4             4           1.04                   90         356
## 5             5           1.00                   90         323
## 6             6           0.76                   86         381
##   InsulinArea SSPG ClinClass
## 1         124   55         3
## 2         117   76         3
## 3         143  105         3
## 4         199  108         3
## 5         240  143         3
## 6         157  165         3
```

The variate `SSPG` stands for steady state plasma glucose which measures the patient's insulin resistance, a pathological condition where the body's cells fail to respond to the hormone insulin.

a. **(3 marks)** Produce a plot of a density estimate of `SSPG` and comment on what you see.

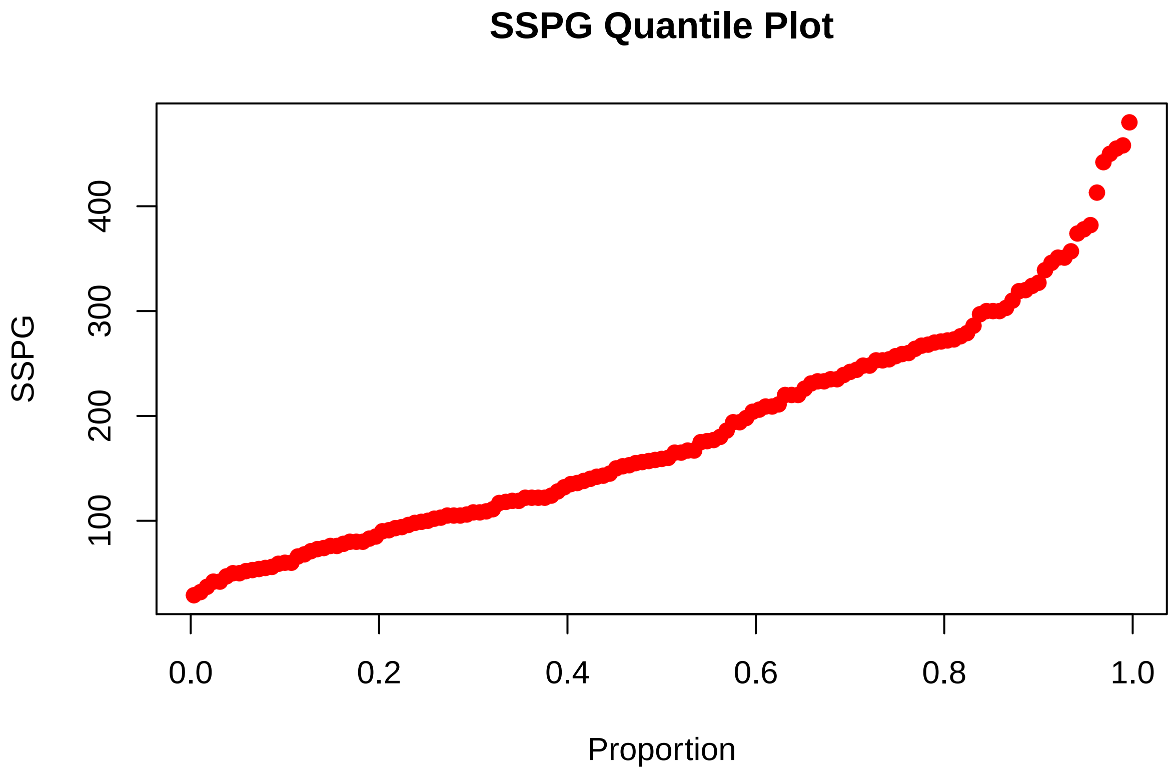
```r
plot(density(diabetes$SSPG),col = "red", main = "SSPG Estimate Density")
polygon(density(diabetes$SSPG), border = "grey50", col = adjustcolor("red", 0.5))
```



The graph is an approxmiately bimodal graph with a peak T about 100 and another peak at approxmately 250. There is a heavier right tail in the graph, and the graph seems to be more right skewed. As different peaks have differnt height and the tails are different, the graph itself is not symmetric.

b. **(3 marks)** Construct a quantile plot of `SSPG` and comment on the shape of its distribution.

```
plot(ppoints(diabetes$SSPG), sort(diabetes$SSPG), type = "b",col = "red", pch = 19, xlab = "Proportion", yl
```



The graph approxmately increses at a linear shape. But there is a convex shape in the end. There may be a heavier right tail at the end.

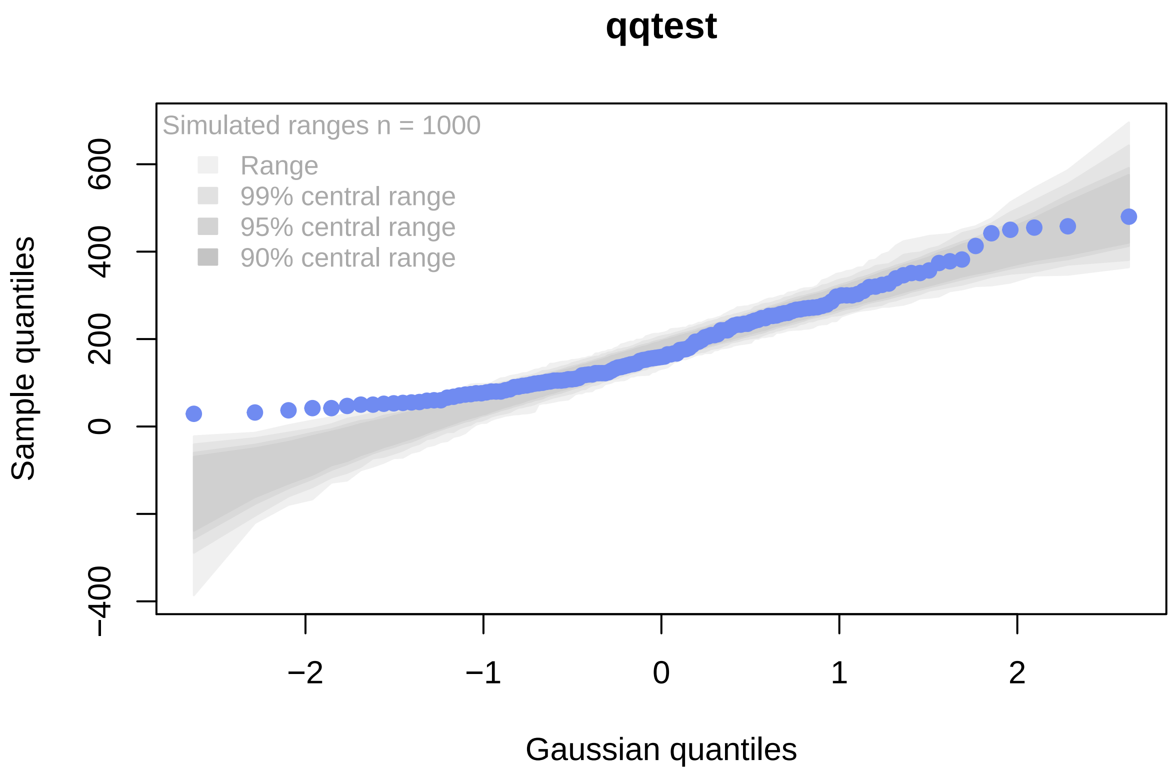
c. **(3 marks)** Use `qqtest` to construct a qqplot that compares `SSPG` to a standard normal distribution. Include envelopes in the plot. Comment on the distribution of `SSPG` and whether it might reasonably be regarded as a sample from some normal distribution. Explain your reasoning

   Important: Before every `qqtest` execute `set.seed(3124159)` so that we are all seeing the same plots.

```
set.seed(3124159)
library("qqtest")
```

```
## Warning: package 'qqtest' was built under R version 3.5.3
```

```
qqtest(diabetes$SSPG)
```



As shown in the graph, most of points lies inside the envolope, but some points, especially ones on the left side with extrme values do not lie in. Those are the evidence against the hypothesis that the points are not from the normal distribution.

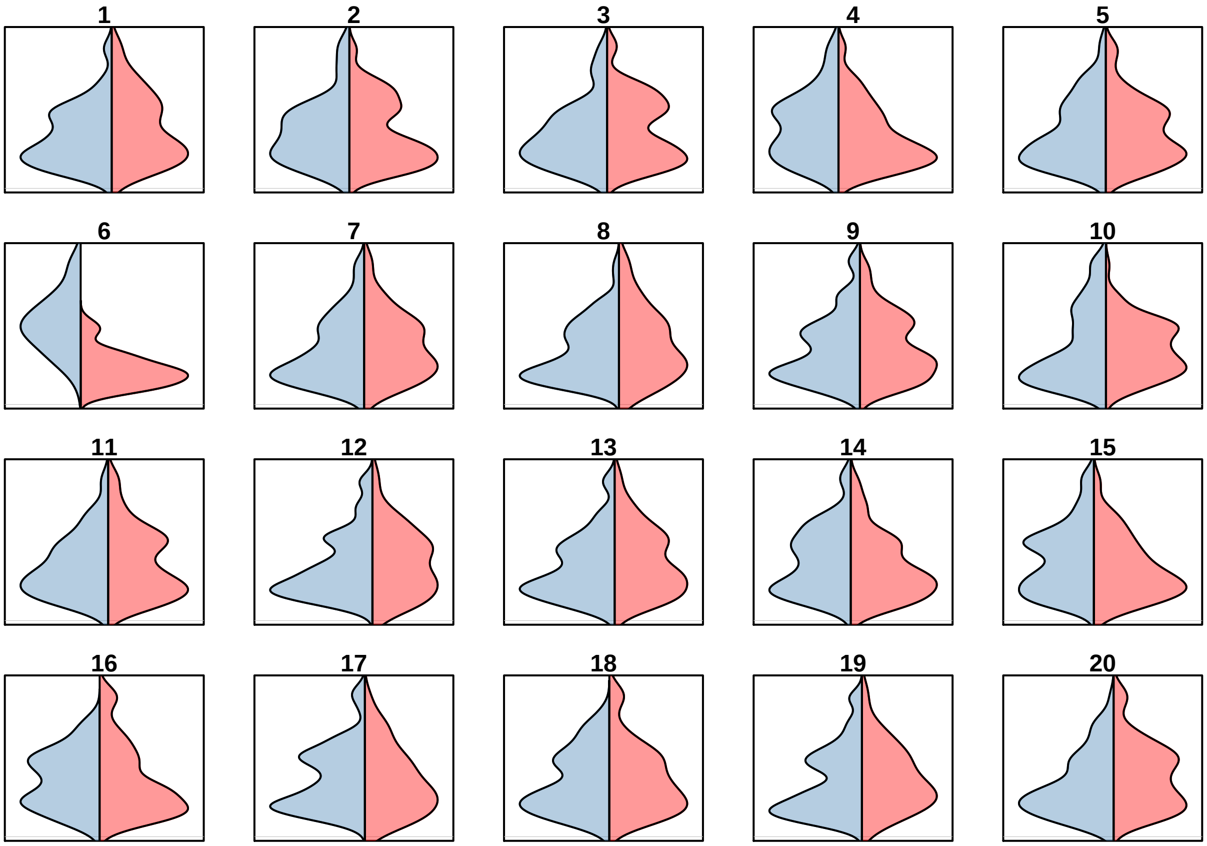
d. The last variate, `ClinClass`, represents the classification of each patient according to the 1979 medical criteria into one of three groups: 1 = "Overt Diabetic", 2 = "Chemical Diabetic", and 3 = "Normal".

   i. **(4 marks)** Construct a back to back density line-up plot to assess whether the normal and diabetic (chemical and overt combined) `SSPG` values come from the same distribution. Use `set.seed(3124159)` and show your code. What conclusions do you draw?

```
set.seed(3124159)
mixRandomly <- function(data) {
  # Note that data need not be a data frame
  # It is expected to be a list with an x and a y component
  # (possibly of different lengths)
  x <- data$x
  y <- data$y
  n_x <- length(x)
  n_y <- length(y)
  mix <- c(x, y)
  select4x <- sample(1:(n_x + n_y),
                     n_x,
                     replace = FALSE)
  new_x <- mix[select4x]
  # The mixing occurs
  y.new <- mix[-select4x]
  list(x = new_x, y = y.new)
}

hideLocation <- function(trueLoc, nSubjects) {
```

```r
  possibleBaseVals <- 3:min(2 * nSubjects, 50)
  # remove easy base values
  possibleBaseVals <- possibleBaseVals[possibleBaseVals != 10 & possibleBaseVals != 5]
  base <- sample(possibleBaseVals, 1)
  offset <- sample(5:min(5 * nSubjects, 125), 1)
  # return location information (trueLoc hidden)
  list(trueLoc=paste0("log(", base^(trueLoc + offset), ", base=", base, ") - ", offset))
}

revealLocation <- function(hideLocation) { eval(parse(text = hideLocation$trueLoc)) }

back2back <- function(data, subjectNo) {
  ylim <- extendrange(c(data$x, data$y))
  Xdensity <- density(data$x, bw = "SJ")
  Ydensity <- density(data$y, bw = "SJ")
  Ydensity$y <- -Ydensity$y
  xlim <- extendrange(c(Xdensity$y, Ydensity$y))
  xyswitch <- function(xy_thing) {
    yx_thing <- xy_thing
    yx_thing$x <- xy_thing$y
    yx_thing$y <- xy_thing$x
    yx_thing }
  plot( xyswitch(Xdensity),
        col = "red",
        main = paste(subjectNo), # display subject number
        cex.main = 1.1, # increase subject number size
        ylab = "", xlab = "", xaxt = "n", yaxt = "n", xlim = xlim, ylim = ylim)
  polygon(xyswitch(Xdensity), col = adjustcolor("red", 0.4))
  lines(xyswitch(Ydensity), col = "steelblue")
  polygon(xyswitch(Ydensity), col = adjustcolor("steelblue", 0.4))
}

lineup <- function(data,
                   showSubject = NULL,
                   generateSubject = NULL, trueLoc = NULL,
                   layout = c(5, 4)) {
  # Get the number of subjects in total
  nSubjects <- layout[1] * layout[2]
  if (is.null(trueLoc)) {
    trueLoc <- sample(1:nSubjects, 1) }
  if (is.null(showSubject)) {
    stop("need a plot function for the subject") }
  if (is.null(generateSubject)) {
    stop("need a function to generate subject") }
  # Need to decide which subject to present
  presentSubject <- function(subjectNo) {
    if (subjectNo != trueLoc) {
      data <- generateSubject(data) }
    showSubject(data, subjectNo) }
  # This does the plotting
  savePar <- par(mfrow = layout, mar = c(1, 1, 1, 1), oma = rep(0, 4))
  sapply(1:nSubjects, FUN = presentSubject)
  par(savePar)
  # hide the true location but return information to reconstruct it.
  print(hideLocation(trueLoc, nSubjects))
  print(revealLocation(hideLocation(trueLoc, nSubjects)))
```

```
}

index <- with(diabetes, which(ClinClass == 3))
sspg <- diabetes$SSPG
normal <- sspg[index]
diabetic <- sspg[-index]
data <- list(x = normal, y = diabetic)
lineup(data, generateSubject = mixRandomly, showSubject = back2back, layout = c(4, 5))
```



```
## $trueLoc
## [1] "log(7.51141330201283e+30, base=22) - 17"
##
## [1] 6
```

From the graph and the result of the code we can see that the graph 6 is most differenct compare to other 19 graphs. As it has huge diffence in not only the location and the concetration between the red and the blue region, it is an evidence against the null hypothesis.

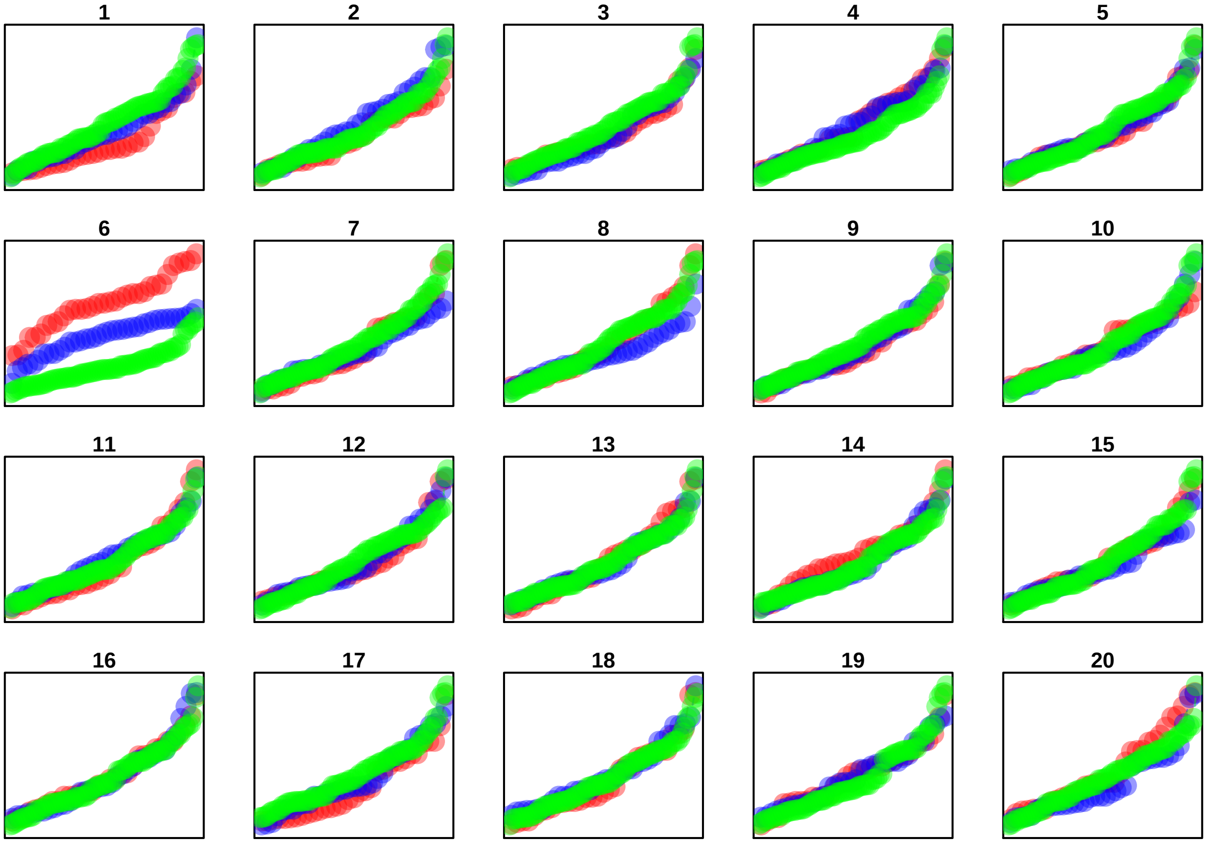
```
ii. **Grad students, bonus undergraduates**  **(8 marks)** Consider the following code:
```

```
set.seed(3124159)
mixRandomly <- function(data) {
  # Note that data need not be a data frame
  # It is expected to be a list with an x, y and z component
  # (possibly of different lengths)
  x <- data$x
  y <- data$y
  z <- data$z
  n_x <- length(x)
  n_y <- length(y)
  n_z <- length(z)
  mix <- c(x, y, z)
  select4x <- sample(1:(n_x + n_y + n_z), n_x, replace = FALSE)
  new_x <- mix[select4x]
  # The mixing occurs
  yz.new <- mix[-select4x]
  select4y <- sample(1:(n_y + n_z), n_y, replace = FALSE)
  y.new <- yz.new[select4y]
  z.new <- yz.new[-select4y]
  list(x = new_x, y = y.new, z = z.new) }


hideLocation <- function(trueLoc, nSubjects) {
  possibleBaseVals <- 3:min(2 * nSubjects, 50)
  # remove easy base values
  possibleBaseVals <possibleBaseVals[possibleBaseVals != 10 & possibleBaseVals != 5]
  base <- sample(possibleBaseVals, 1)
  offset <- sample(5:min(5 * nSubjects, 125), 1)
  # return location information (trueLoc hidden)
  list(trueLoc=paste0("log(", base^(trueLoc + offset), ", base=", base, ") - ", offset)) }


revealLocation <- function(hideLocation) { eval(parse(text = hideLocation$trueLoc)) }


myQuantilePlot <- function(data, subjectNo) {
  ylim <- extendrange(c(data$x, data$y, data$z))
  n_x <- length(data$x)
  n_y <- length(data$y)
  n_z <- length(data$z)
  p_x <- ppoints(n_x)
  p_y <- ppoints(n_y)
  p_z <- ppoints(n_z)
  plot( p_x, sort(data$x), type = "b", col = adjustcolor("red", 0.4), pch = 19, cex = 2, ylim = ylim, main
)
  points( p_y, sort(data$y), type = "b", col = adjustcolor("blue", 0.4), pch = 19, cex = 2 )
  points( p_z, sort(data$z), type = "b", col = adjustcolor("green", 0.4), pch = 19, cex = 2 )

}

lineup <- function(data, showSubject = NULL, generateSubject = NULL, trueLoc = NULL, layout = c(5, 4)) {
  # Get the number of subjects in total
  nSubjects <- layout[1] * layout[2]
  if (is.null(trueLoc)) {
    trueLoc <- sample(1:nSubjects, 1) }
  if (is.null(showSubject)) {
    stop("need a plot function for the subject") }
  if (is.null(generateSubject)) {
    stop("need a function to generate subject") }
  # Need to decide which subject to present
  presentSubject <- function(subjectNo) {
    if (subjectNo != trueLoc) {
      data <- generateSubject(data) }
    showSubject(data, subjectNo) }
  # This does the plotting
  savePar <- par(mfrow = layout, mar = c(1, 1, 1, 1), oma = rep(0, 4))
  sapply(1:nSubjects, FUN = presentSubject)
  par(savePar)
  # hide the true location but return information to reconstruct it.
  print(hideLocation(trueLoc, nSubjects))
  print(revealLocation(hideLocation(trueLoc, nSubjects)))
}



overt_i <- with(diabetes, which(ClinClass == 1))
chemical_i <- with(diabetes, which(ClinClass == 2))
normal_i <- with(diabetes, which(ClinClass == 3))
sspg <- diabetes$SSPG
overt <- sspg[overt_i]
chemical <- sspg[chemical_i]
normal <- sspg[normal_i]
data <- list(x = overt, y = chemical, z = normal)
lineup(data, generateSubject = mixRandomly, showSubject = myQuantilePlot, layout = c(4, 5))
```

```
## Warning in possibleBaseVals < possibleBaseVals[possibleBaseVals != 10 & :
## longer object length is not a multiple of shorter object length
```

```
## $trueLoc
## [1] "log(4.12220233558324e+117, base=37) - 69"
```

```
## Warning in possibleBaseVals < possibleBaseVals[possibleBaseVals != 10 & :
## longer object length is not a multiple of shorter object length
```



```
## [1] 6
```

```
    The function `mixRandomly` will need to be rewritten to handle `data` being a list of three samples.  W
```

Graph 6 is clearly different to other graphs. The code result confirms that. The p value is 1/20 with the null hypothesis that the 3 clinical classes have the same distribution. Graph 6 is an evidence against the hypothesis.