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 InsulinArea
## 1             1           0.81                   80         356         124
## 2             2           0.95                   97         289         117
## 3             3           0.94                  105         319         143
## 4             4           1.04                   90         356         199
## 5             5           1.00                   90         323         240
## 6             6           0.76                   86         381         157
##   SSPG ClinClass
## 1   55         3
## 2   76         3
## 3  105         3
## 4  108         3
## 5  143         3
## 6  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
library(ggplot2)
```

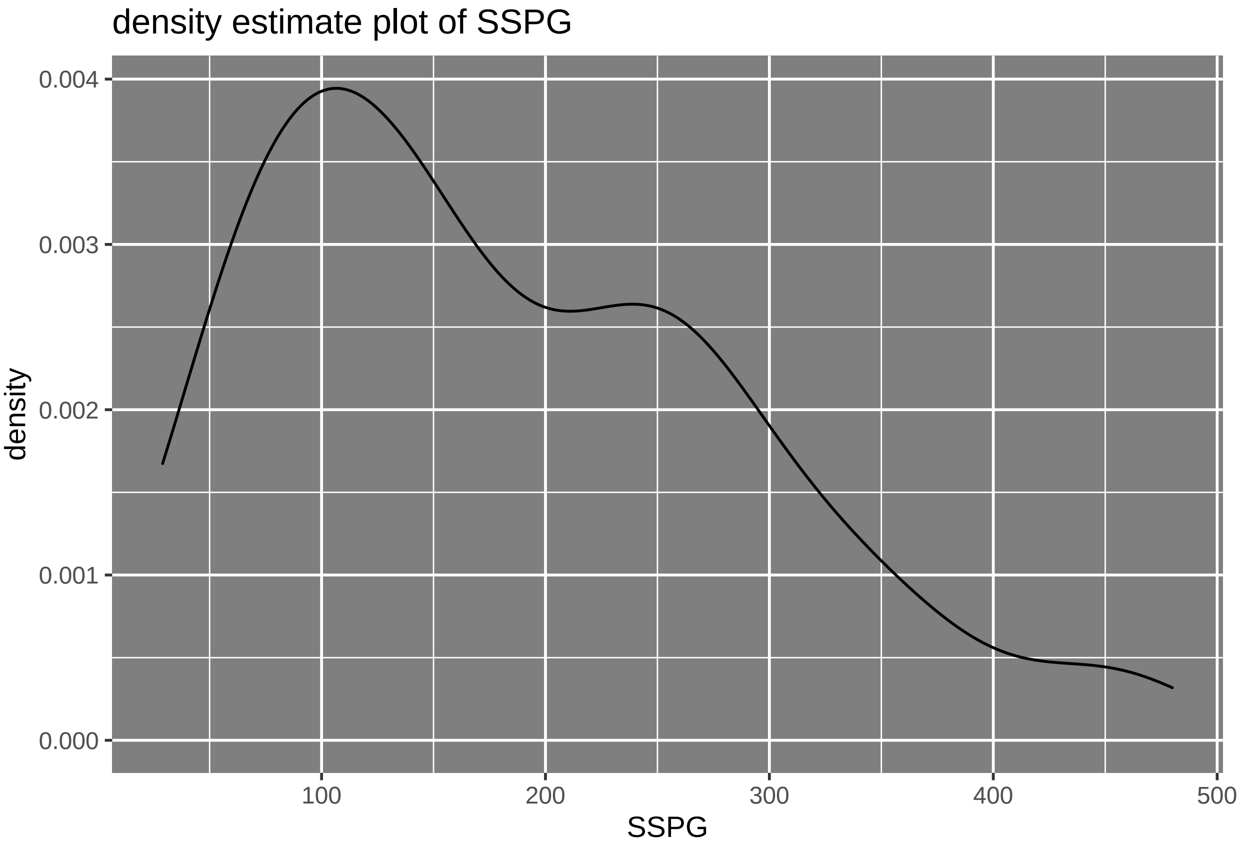
```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```r
library(egg)
```

```
## Warning: package 'egg' was built under R version 4.0.3
```

```
## Loading required package: gridExtra
```

```r
ggplot(data=data.frame(diabetes), mapping = aes(x=SSPG)) +
    geom_density(kernel="gaussian")+
    theme(panel.background = element_rect(fill = "grey50"))+
    labs(title="density estimate plot of SSPG")
```

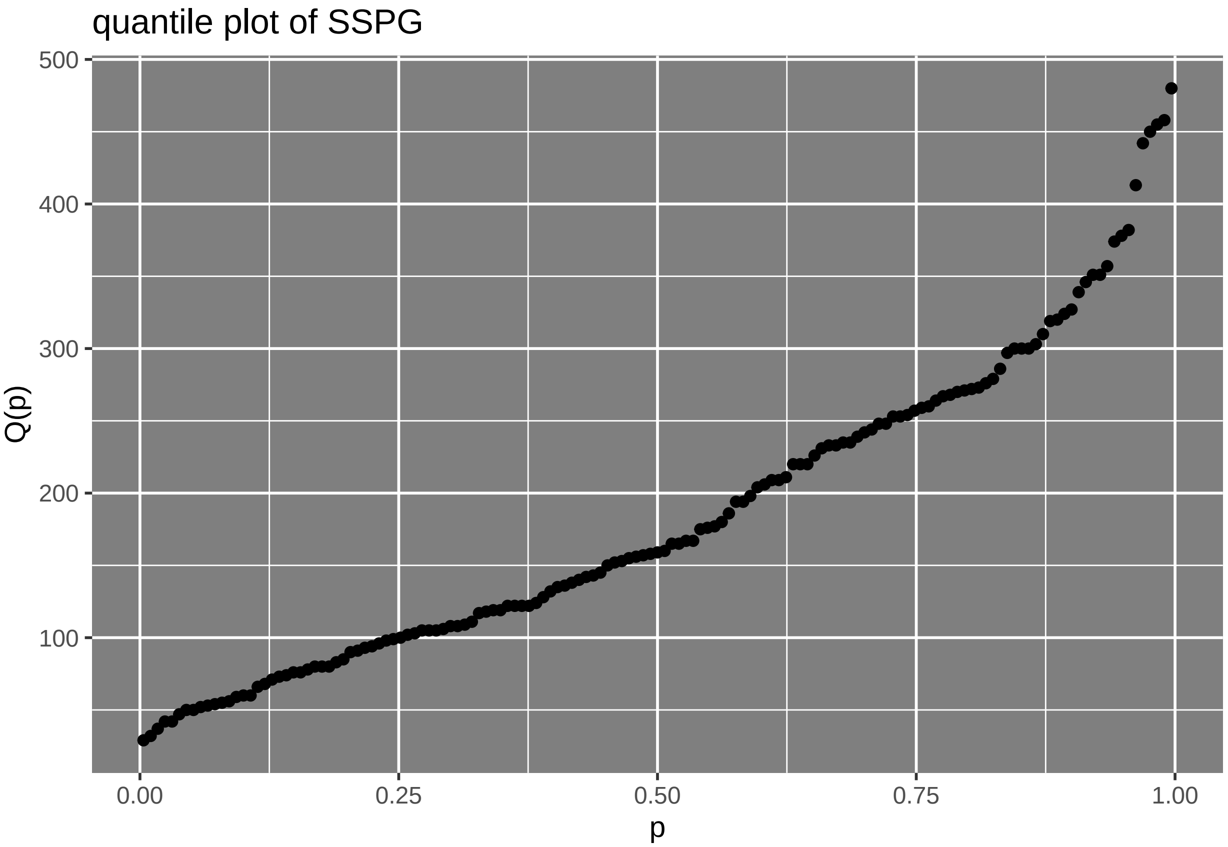


we see the data is multimodal and is skewed to the right

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

```
ggplot(data=diabetes, mapping = aes(x=ppoints(length(SSPG)), y=sort(SSPG))) +
  geom_point(kernel="rectangular")+
  labs(x="p",y="Q(p)")+
  theme(panel.background = element_rect(fill = "grey50"))+
  labs(title="quantile plot of SSPG")
```

```
## Warning: Ignoring unknown parameters: kernel
```



SSPG values are more concentrated at small values below 200, after that the Q(p) increases quickly as p increases.

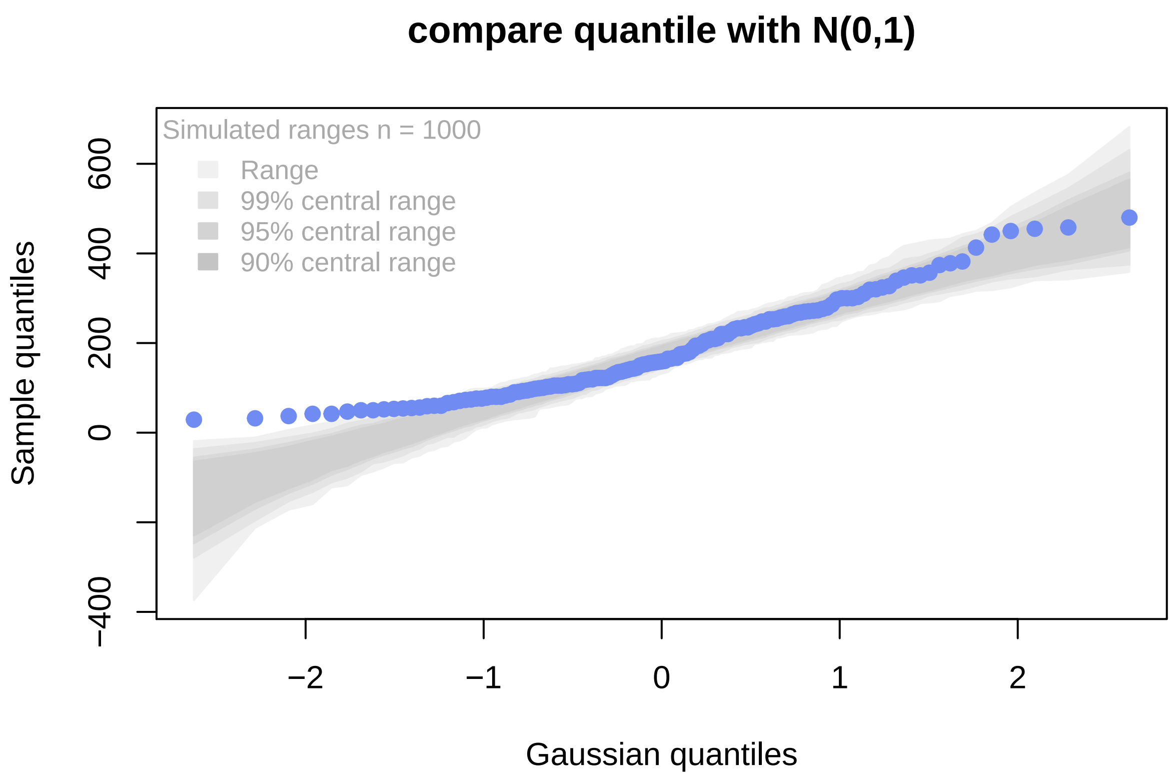
c. **(3 marks)** Use `qqtest` to construct a qqplot that compares `SSPG` to a standard normal distribution. Include envelopes in the plot. Comment on the distribution of `SSPG` and whether it might reasonably be regarded as a sample from some normal distribution. Explain your reasoning

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

```r
library(qqtest)
```

```
## Warning: package 'qqtest' was built under R version 4.0.3
```

```r
set.seed(3124159)
qqtest(diabetes$SSPG,main="compare quantile with N(0,1)")
```



this does not look like nomral distribution as some data points in the lower quantile are clearly outside the confidence interval envelope

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)
ind=diabetes$ClinClass==3
SSPG.normal = diabetes$SSPG[ind]
SSPG.other = diabetes$SSPG[!ind]
data=list(x=SSPG.normal, y=SSPG.other)


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="firebrick",main=paste( subjectNo),# display subject number
       cex.main = 2,
       # increase subject number size
```

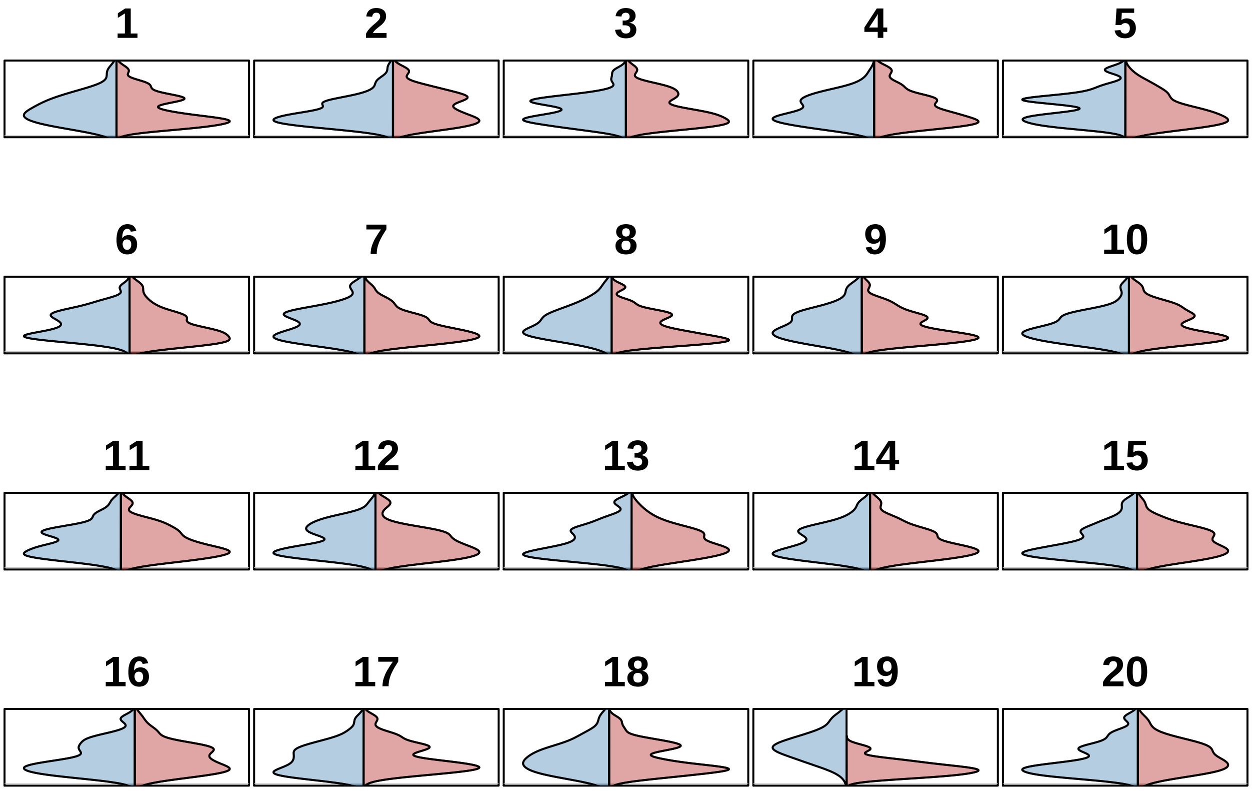
```r
       ylab="", xlab="", xaxt="n", yaxt="n",xlim=xlim, ylim=ylim)
  polygon(xyswitch(Xdensity), col=adjustcolor("firebrick", 0.4))
  lines(xyswitch(Ydensity), col="steelblue")
  polygon(xyswitch(Ydensity), col=adjustcolor("steelblue", 0.4))
}

mixRandomly = function(data){
  x <- data$x
  y <- data$y
  m <-length(x)
  n <-length(y)
  mix <-c(x,y)
  select4x <-sample(1:(m+n),m,replace = FALSE)
  new_x <- mix[select4x]
  # The mixing occurs
  new_y <- mix[-select4x]
  list(x=new_x, y=new_y)
}


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(2.5, 0.1, 3, 0.1), oma=rep(0,4))
  sapply(1:nSubjects, FUN = presentSubject)
  par(savePar)
  # hide the true location but return information to reconstruct it.
  hideLocation(trueLoc, nSubjects)
}
hideLocation <-function(trueLoc, nSubjects){
  possibleBaseVals <- 3:min(2*nSubjects, 50)
  # remove easy base value
  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))}


lineup(data, generateSubject = mixRandomly, showSubject = back2back, layout = c(4,5))
```



```
## $trueLoc
## [1] "log(7.93531459841716e+47, base=13) - 24"
```

```r
log(7.93531459841716e+47, base=13) - 24
```

```
## [1] 19
```

it's quite obvious that 19 does not look like other graph, and turns out it is the real data, so given two
the same distribution, there is only 5% chance of we selectng it right.
we have some evidence against the null hypothese of they are from the same distribution.

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

~~~~
```r
data <- list(x=x, y=y, z=z)
lineup(data,
   generateSuspect = mixRandomly,
   showSuspect = myQuantilePlot,
   layout=c(5,4))
```
~~~~

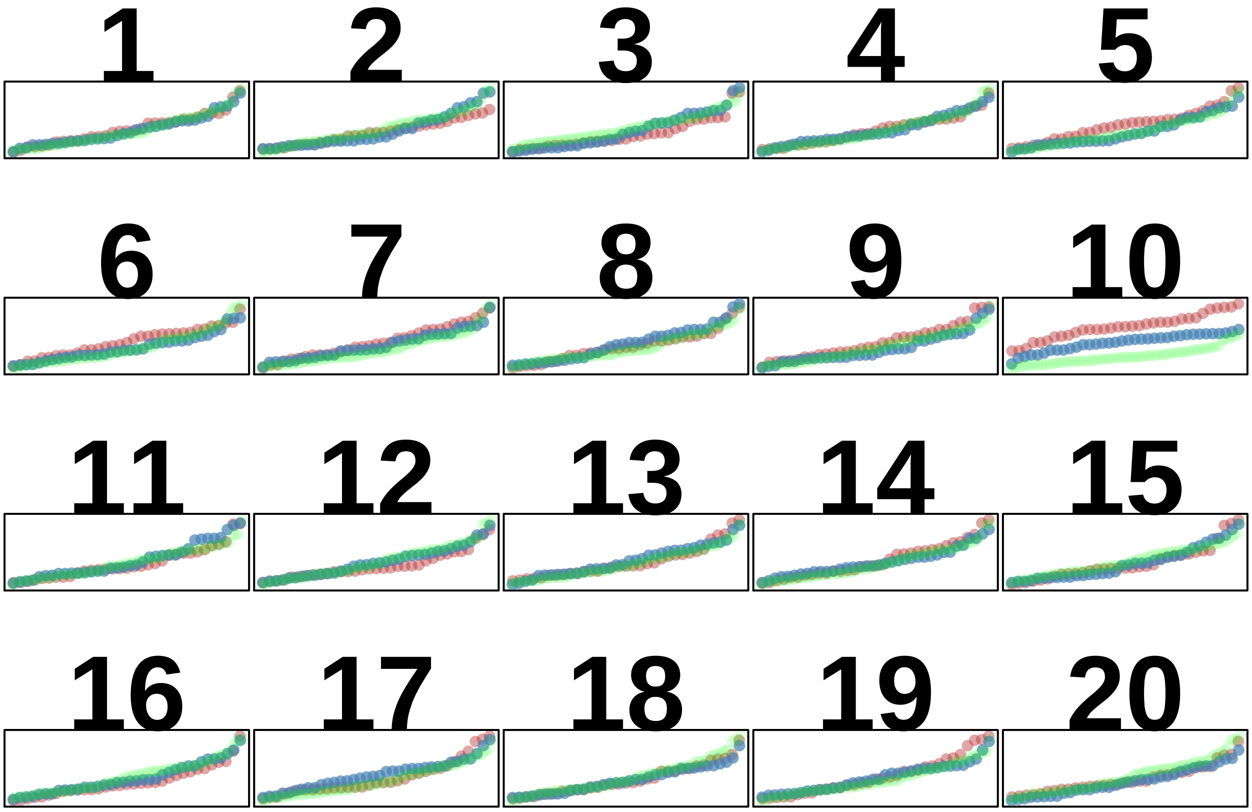
    The function `mixRandomly` will need to be rewritten to handle `data` being a list of three samples. W

```r
set.seed(314159)
myquantiles <-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$x), type="b",
  col=adjustcolor("firebrick", 0.4),  pch=19, cex=1,ylim = ylim,main=paste( subjectNo),cex.main = 5, ylab="
  points(p_y,sort(data$y), type="b",col=adjustcolor("steelblue", 0.7),  pch=19, cex=1)
  points(p_z,sort(data$z), type="b",col=adjustcolor("green", 0.1),  pch=19, cex=1)
}
data=list(x=diabetes$SSPG[(diabetes$ClinClass)==1],
          y=diabetes$SSPG[(diabetes$ClinClass)==2],
          z=diabetes$SSPG[(diabetes$ClinClass)==3])

mymixRandomly = function(data){
  x <- data$x
  y <- data$y
  z <- data$z
  m <-length(x)
  n <-length(y)
  o <-length(z)
  mix <-c(x,y,z)
  select4xyz <-sample(1:(m+n+o),m+n+o,replace = FALSE)
  select4x=select4xyz[1:m]
  select4y=select4xyz[(m+1):(m+n)]
  select4z=select4xyz[(m+n+1):(m+n+o)]
  new_x <- mix[select4x]
  new_y <- mix[select4y]
  new_z <- mix[select4z]
  list(x=new_x, y=new_y, z=new_z)
}

lineup(data, generateSubject = mymixRandomly, showSubject = myquantiles, layout = c(4,5))
```



```
## $trueLoc
## [1] "log(1.21416805764108e+83, base=8) - 82"
```

```
log(1.47573952589676e+87, base=20) - 57
```

```
## [1] 10
```

We observe that it's quite obvious figure 10 is different from other plots. Indeed, figure 10 represents the true data. This suggest that we have some strong evidence against overt diabetic, chemical diabetic and normal patient have same SSPG, there is only 5% possibility of we choosing 10 given H_0 is true