**4 Graphics**

**Exercise 1**

1. a Create three different data sets, based on different distributions:

> discrete\_small <- sample(1:10, 100, prob=seq(from=0.1,to=1.0, length.out=10), replace=TRUE)

> exponential <- rexp(1000)

> bi\_modal <- c(rnorm(1000), rnorm(1000, mean=5, sd=2))

discrete\_small <- sample(1:10, 100, prob=seq(from=0.1,to=2.0, length.out=10), replace=TRUE)  
exponential <- rexp(1000)  
bi\_modal <- c(rnorm(1000), rnorm(1000, mean=5, sd=2))  
plot(discrete\_small)  
plot(exponential)  
plot(bi\_modal)  
boxplot(discrete\_small)  
boxplot(exponential)  
boxplot(bi\_modal)  
hist(discrete\_small)  
hist(exponential)  
hist(bi\_modal)  
a <- density(exponential)  
plot(a)

1.b Study the given distributions and describe them briefly.

The exponetial data set is (fairly) continuous; a density plot nicel

y shows the distribution.

A boxplot can be useful to place several datasets next to each other for c

omparison. The histogram

can easily become too coarse for such data. Plotting a trable doesn’t make

any sense here

The bi-modal data is continuous; the density plot again nicely shows t

he distribution, as

does the histogram. The boxplot completely misses the bi-modality

of this data.

2. Make different plots of the three data sets, including boxplot, histogram, density plot. How useful are the different plots for the different data sets. Are some plot types more useful for some data but not for other data?

make all plot types for each of the three datasets in one window

change outer margins to fill space

discrete data

opar <- par(mfrow=c(1,4), mar=c(3,2,2,1)+.1)

boxplot(discrete\_small)

hist(discrete\_small)

plot(density(discrete\_small))

plot(table(discrete\_small))

10

Discrete datasets are best visualized plotting the table of the datai or

making a histogram.

Note that the histogram puts the counts for value ’1’ and value ’2’ in one bin. T

he density plot

suggests continuous data and also suggests that the data can be less then ze

ro, or more than 10.

# discrete data

opar <- par(mfrow=c(1,4), mar=c(3,2,2,1)+.1)

boxplot(exponential)

hist(exponential)

plot(density(exponential))

plot(table(exponential))

**Exercise 2**

Combining histograms

Some functions do not appear to return data, e.g. the function plot(..). In some cases they do but it is hidden from you. You can see the returned data if you assign the (invisibly) returned data from the function to a variable, e.g.

datah <- hist(data)

1. Create three different data sets, based on different distributions:

data1 <- sample(1:10, 100, prob=seq(from=0.4,to=1.0, length.out=10), replace=TRUE)

data2 <- sample(1:10, 100, prob=seq(from=5.0,to=0.1, length.out=10), replace=TRUE)

data3 <- sample(1:10, 100, prob=seq(from=0.4,to=0.8, length.out=10), replace=TRUE)

1. Make histograms for each of the three datasets and assign the result to variables.

data1h <- hist(data1, plot=FALSE)

str(data1h)

List of 7

$ breaks : num [1:10] 1 2 3 4 5 6 7 8 9 10

$ counts : int [1:9] 7 3 5 8 15 17 13 14 18

$ intensities: num [1:9] 0.07 0.03 0.05 0.08 0.15 0.17 0.13 0.

14 0.18

$ density : num [1:9] 0.07 0.03 0.05 0.08 0.15 0.17 0.13 0.14 0.

18

$ mids : num [1:9] 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5

$ xname : chr "data1"

$ equidist : logi TRUE

- attr(\*, "class")= chr "histogram"

> data2h <- hist(data2, plot=FALSE)

> str(data2h)

List of 7

$ breaks : num [1:10] 1 2 3 4 5 6 7 8 9 10

$ counts : int [1:9] 33 15 9 11 10 8 6 7 1

$ intensities: num [1:9] 0.33 0.15 0.09 0.11 0.1 0.08 0.06 0.0

7 0.01

$ density : num [1:9] 0.33 0.15 0.09 0.11 0.1 0.08 0.06 0.07 0.0

1

$ mids : num [1:9] 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5

$ xname : chr "data2"

$ equidist : logi TRUE

- attr(\*, "class")= chr "histogram"

> data3h <- hist(data3, plot=FALSE)

> str(data3h)

List of 7

$ breaks : num [1:10] 1 2 3 4 5 6 7 8 9 10

$ counts : int [1:9] 10 8 13 9 12 8 10 12 18

$ intensities: num [1:9] 0.1 0.08 0.13 0.09 0.12 0.08 0.1 0.12

0.18

$ density : num [1:9] 0.1 0.08 0.13 0.09 0.12 0.08 0.1 0.12 0.18

$ mids : num [1:9] 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5

$ xname : chr "data3"

$ equidist : logi TRUE

- attr(\*, "class")= chr "histogram"

1. Inspect the variables that are created by the hist function. What data in the variables are used for making the histogram?

The output from the hist(..) function is in fact a list (with a ’class label’ histogram) with 7elements. Element

breaks specifies the left and right boundaries of the bins of the histogram.counts the counts per bin,

density and frequency contain estimated density values (from the help page: If ’all(diff(breaks) == 1)’, they are the relative frequencies ’counts/n’). Element mids contains the midpoints of the bins. Note that breaks

contains 1 more value than the number of bins whereas mids contains the same number of values.

4. Plot all three histograms in a single plot. Are the ranges of the three variables compatible? Is this always the case?

> # plot all histograms on top of each other. One histogram may

easily

> # cover others. Using combinations of angle, density, col,

> # border (see "help(plot.histogram)") may help in some cas

es.

> par(mfrow=c(1,2))

> # simple plotting on top

> plot(data1h)

> lines(data2h, col=2)

> lines(data3h, col=3)

> ###########################

> # slightly smarter plotting:

> ###########################

> # make empty plot with proper dimensions

> maxy <- max(c(data1h$counts, data2h$counts, data3h$counts))

> ylim <- c(0, maxy)

> # x-scale is the same in all data

> xlim <- range(data1h$breaks)

> plot(NA, xlim=xlim, ylim=ylim, main='three histograms',xlab='value',ylab='ount’, bty='n')

> lines(data1h, angle=-45, density=10, col=1)

> lines(data2h,border=2, angle=45, density=10, col=2)

> lines(data3h,border=3, angle=90, density=20, col=3}

> par(mfrow=c(1,2))

> # simple line plot using the 'mids' and 'counts' elements

> # make empty plot with proper dimensions

> maxy <- max(c(data1h$counts, data2h$counts, data3h$counts))

> ylim <- c(0, maxy)

> # x-scale is the same in all data

> xlim <- range(data1h$breaks)

> plot(NA, xlim=xlim, ylim=ylim, main='three histograms',xlab='value',ylab='count', bty='n')

> lines(data1h$mids, data1h$count,lwd=2)

> lines(data2h$mids, data2h$counts, col=2,lwd=2)

> lines(data3h$mids, data3h$counts, col=3,lwd=2)

> ###########################

> # Using plottype 's'(step)

> ###########################

> plot(NA, xlim=xlim, ylim=ylim, main='three histograms',xlab='value',ylab='count', bty='n', ty='s')

> lines(data1h$mids, data1h$count,lwd=2, ty='s')

> lines(data2h$mids, data2h$counts, col=2,lwd=2, ty='s')

> lines(data3h$mids, data3h$counts, col=3,lwd=2, ty='s')

**Exercise 3**

Scatter plot; highlight probe types

In this exercise you will make a scatterplot and highlight different subsets of datapoints using different colors.

Download a dataset, which contain the data of the Cy3 and the Cy5 channels of a DamID experiment, from the course website: http://bioinformatics.nki.nl/courses/Rstat\_11\_II/texts/resources/Cy5Cy3.RData, and load it in your R-session:

> load(‘Cy5Cy3.RData’)

> # or:

> load(url(‘http://bioinformatics.nki.nl/courses/Rstat\_11\_II/texts/resources/Cy5Cy3.RData’))

1. Examine the datasets (str(), dim(), etc).

What (kind of information) do you think each column contains (hint; table, unique, str).

> class(Cy3)

[1] "data.frame"

> dim(Cy3)

[1] 71901 11

> str(Cy3)

'

data.frame

'

: 71901 obs. of 11 variables:

$ IMAGE\_ID : chr "108197\_532" "108197\_532" "108197\_532" "1

08197\_532" ...

$ GENE\_EXPR\_OPTION: chr "EXPERIMENTAL" "EXPERIMENTAL" "E

XPERIMENTAL" "EXPERIMENTAL" ...

$ SEQ\_ID : chr "chr13" "chr13" "chr13" "chr13" ...

$ PROBE\_ID : chr "HUMCHR13\_0000041027" "HUMCHR13\_0000064

855" "HUMCHR13\_0000015279" "HUMCHR13\_0000068371"

$ POSITION : int 36927955 48031637 24852094 49639745 267182

13 31611836 37784077 21594546 31938700 20298208

$ X : int 345 148 670 624 247 253 47 240 569 684 ...

$ Y : int 463 864 316 932 297 579 1 472 287 396 ...

$ MATCH\_INDEX : int 65176949 65176951 65176953 65176954 651

76955 65176956 65176962 65176964 65176967 6517697

$ SEQ\_URL : logi NA NA NA NA NA NA ...

$ PM : num 1849 1115 1012 1122 877 ...

$ MM : num 0 0 0 0 0 0 0 0 0 0 ...

> # are the data frames compatible?

> nrow(Cy5) == nrow(Cy3)

[1] TRUE

> all(Cy3$PROBE\_ID == Cy5$PROBE\_ID)

[1] TRUE

> # or:

> table(Cy3$PROBE\_ID == Cy5$PROBE\_ID)

TRUE

71901

> # or:

> all.equal(Cy3$PROBE\_ID, Cy5$PROBE\_ID)

[1] TRUE

2. Create a dataframe from the two datasets with the following columns:

seqname (the name of the sequence on which the probe lies)

start (position of the probe)

log2(Cy5) (the PM value in the Cy5 dataset)

log2(Cy3)

probe

type

> # some columns contain unique values for each probe, some co

lumns

> # contain the same vsalue for all probes. column

'

GENE\_EXPR\_OPTION

'

> # contains a larege number of

'

EXPERIMENTAL

'

probes plus some control

> # probes (RANDOM, and SPIKE-1-5)

> table(Cy3$GENE\_EXPR\_OPTION)

EXPERIMENTAL RANDOM SPIKE\_1 SPIKE\_2

69503 1399 200 199

SPIKE\_3 SPIKE\_4 SPIKE\_5

200 200 200

> data <- data.frame(seqname = Cy3$SEQ\_ID,

start = Cy5$POSITION,

Cy3 = log2(Cy3$PM),

Cy5 = log2(Cy5$PM),

probe\_type = Cy5$GENE\_EXPR\_OPTION,

stringsAsFactors = FALSE)

1. Make density plots of the two channels, Cy3 and Cy5.

> plot(density(data$Cy3), lwd=2, col='green')

> lines(density(data$Cy5), col='red', lwd=2)

> legend(x='topright', inset=0.025, legend=c('Cy5','Cy3'), col=c('red','green'),lwd=2)

1. Make a scatterplot of the two channels Cy3, Cy5. Set proper values for title, axis labels. Select pleasing symbols for the data points.

> # first select proper settings for plot-character, labels, etc

> # see the'example'section of the help file for points (see pchShow)

> # for an overview of different plotting characters

> pch <- 19

> cex <- 0.3

> main <-'Cy5 vs. Cy3'

> xlab <-'Cy3'

> ylab <-'Cy5'

> plot(data$Cy3, data$Cy5, main=main, xlab=xlab, ylab=ylab, pch=pch, cex=cex

1. Make an index vector indicating the data points of the various probe types and highlight them in the plot using different colors and/or symbols of your choice.

> # what probe\_types do we have?

> unique(data$probe\_type)

[1] "EXPERIMENTAL" "SPIKE\_1" "SPIKE\_2"

[4] "SPIKE\_3" "SPIKE\_4" "SPIKE\_5"

[7] "RANDOM"

> # first plot all data points

> pch <- 19

> cex <- 0.3

> main <-'Cy5 vs. Cy3'

> xlab <-'Cy3'

> ylab <-'Cy5'

> plot(data$Cy3, data$Cy5, main=main, xlab=xlab, ylab=ylab, pch=pch, cex=cex)

> # for each probe type (other than 'EXPERIMENTAL') create an index vector

> # to highlight corresponding points in scatter plot. Use a somewhat larger

> # dot for the highlighted points

> cex <- .8

> idx <- data$probe\_type =='SPIKE\_1'

> col <-'red'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> idx <- data$probe\_type =='SPIKE\_2'

> col <-'green'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> idx <- data$probe\_type =='SPIKE\_3'

> col <-'blue'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> idx <- data$probe\_type =='SPIKE\_4'

> col <-'yellow'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> idx <- data$probe\_type =='SPIKE\_5'

> col <-'magenta'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> idx <- data$probe\_type =='RANDOM'

> col <-'pink'

> points(data$Cy3[idx], data$Cy5[idx], pch=pch, cex=cex, col=col)

> legend(x='bottomright', legend=unique(data$probe\_type),

col=c('black','red','

green','blue','

yellow','magenta','pink'), pch=19,cex=.8)

> ############################

> # alternatively you could set up a vector of colors of the same length

> # as the number of rows in 'data' with different colors corresponding to

> # the probe\_type:

> main <-'Cy5 vs. Cy3'

> xlab <-'Cy3'

> ylab <-'Cy5'

> pch <- 19

> # initialize all colors to black

> colors <- rep('black', nrow(data))

> # set different colors for each probe\_type (other than EXPERIMENTAL)

> colors[data$probe\_type =='SPIKE\_1'] <-'red'

> colors[data$probe\_type =='SPIKE\_2'] <-'green'

> colors[data$probe\_type =='SPIKE\_3'] <-'blue'

> colors[data$probe\_type =='SPIKE\_4'] <-'yellow'

> colors[data$probe\_type =='SPIKE\_5'] <-'magenta'

> colors[data$probe\_type =='RANDOM'] <-'pink'

> # similar for the size of the dots:

> cex <- rep(0.3, nrow(data))

> cex[data$probe\_type !='EXPERIMENTAL'] <- 0.8

> # now we can print all different colors in one go:

> plot(data$Cy3, data$Cy5, main=main, xlab=xlab, ylab=ylab, pch=pch,cex=cex, col=colors)

> legend(x='bottomright', legend=unique(data$probe\_type),col=c('black','red','green','blue','yellow','magenta','pink'), pch=19,cex=0.1)