sol genomics network

Basic Graphs With R

sol genomics network

Boyce Thompson Institute for Plant Research
Tower Road
Ithaca, New York 14853-1801
U.S.A.

by
Aureliano Bombarely Gomez

Basic Graphs with R:

1. What kind of graphs can produce R ?
2. Software and documentation.
3. Steps and functions.
4. Case I: Tomato gene expression
5. Case II: Sequence size distribution.

Basic Graphs with R:

1. What kind of graphs can produce R ?
2. Software and documentation.
3. Steps and functions.
4. Case I: Tomato gene expression
5. Case II: Sequence size distribution.

1. What kind of graphs can produce R ?

Depends of the package:

+ Basic package:

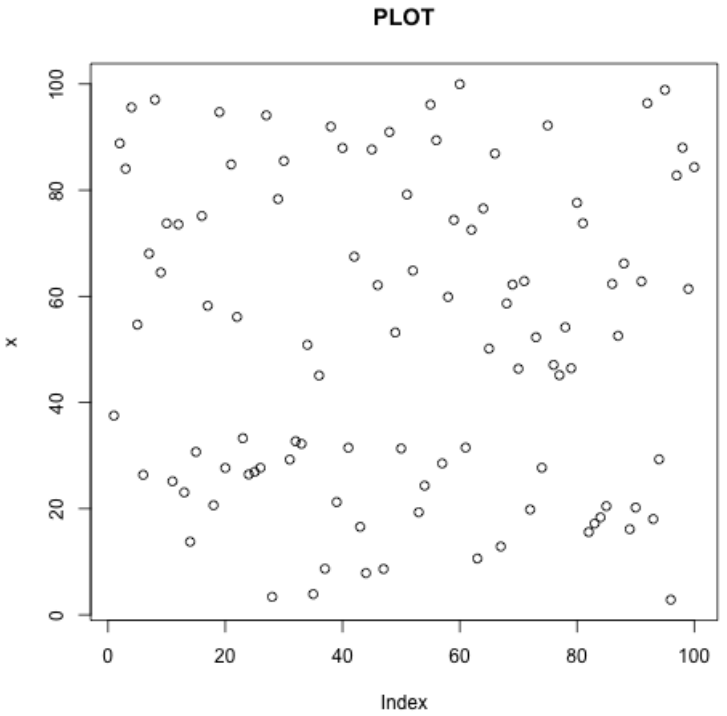
Plot,	simple plots
Hist,	density plots
Dotchart,	simple dotplots
Barplot,	simple bar graphs
Pie,	simple pie graphs
Boxplot,	boxplot graphs

1. What kind of graphs can produce R ?

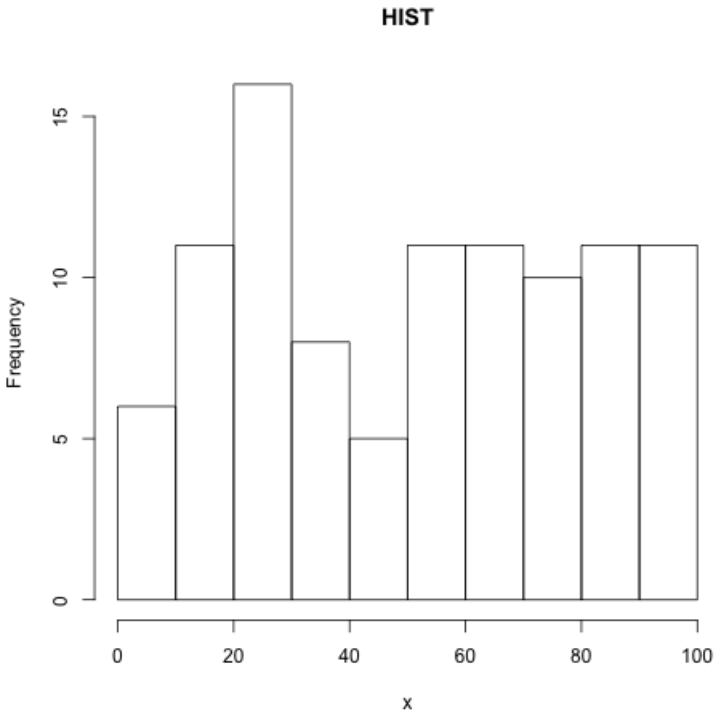
sol genomics network

Example:

```
> x ← runif(100, 0, 100)
```




```
> plot(x, main="PLOT")
```



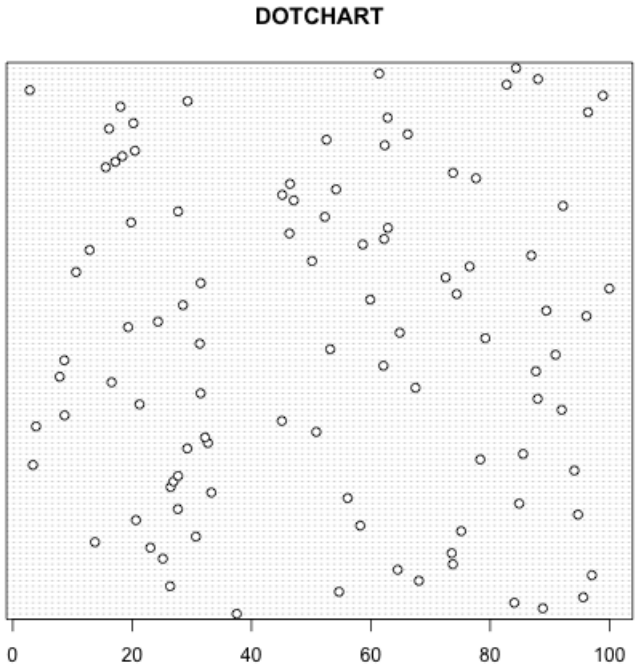

```
> hist(x, main="HIST")
```

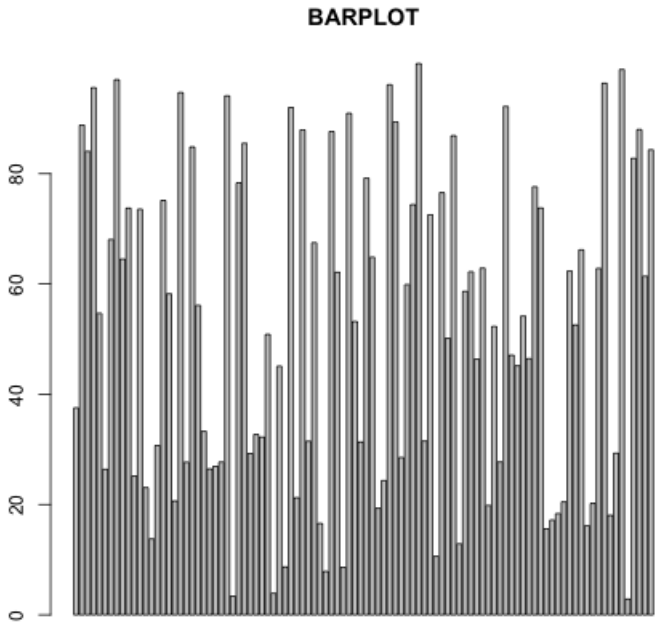
1. What kind of graphs can produce R ?

sol genomics network

Example:

```
> x ← runif(100, 0, 100)
```






```
> dotchart(x, main="DOTCHART")
```

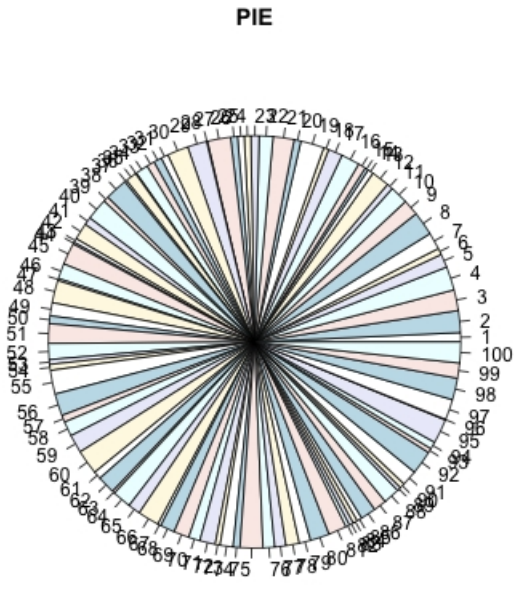
```
> barplot(x, main="BARPLOT")
```

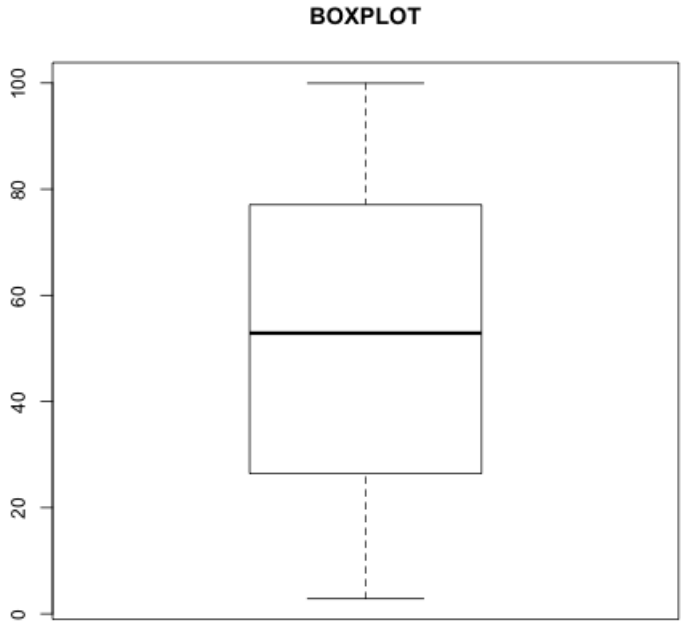
1. What kind of graphs can produce R ?

sol genomics network

Example:

```
> x ← runif(100, 0, 100)
```






```
> pie(x, main=”PIE”)
```

```
> boxplot(x, main=”BOXPLOT”)
```

1. What kind of graphs can produce R ?

sol genomics network

Depends of the package:

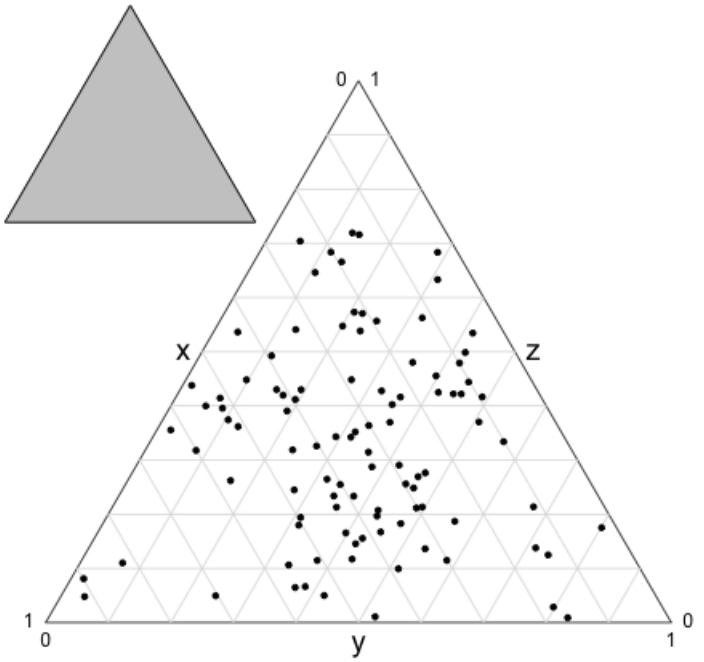
+ Ade4 package:

TrianglePlot, trianglegraphs

```
> x ← runif(100, 0, 100)
> y ← c(1:100)
> z ← sample(1:100)
> xyz ← as.data.frame(cbind(x, y, z))

>library("ade4")

>triangle.plot(xyz)
```



1. What kind of graphs can produce R ?

Depends of the package:

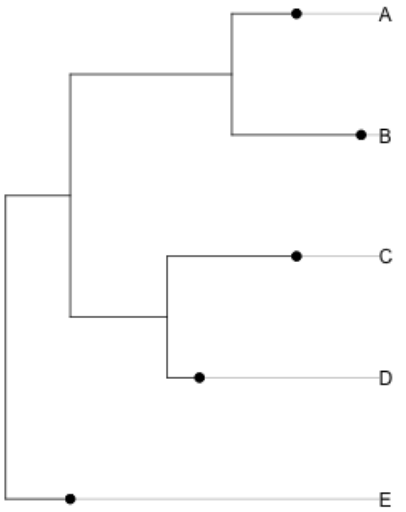
+ Ade4 package:

Plotphylog, phylogenetic trees

```
> tree ←
  "(((A:2,B:4):5,(C:4,D:1):3):2,E:2);"

>library("ade4")

> plot.phylog(newick2phylog(tree))
```



Basic Graphs with R:

1. What kind of graphs can produce R ?

2. Software and documentation.

3. Steps and functions.

4. Case I: Tomato gene expression

5. Case II: Sequence size distribution.

2. Software and documentation.

sol genomics network

WEB:

OFICIAL WEB: http://www.r-project.org/index.html

QUICK-R: http://www.statmethods.net/index.html

R GRAPH GALLERY:

http://addictedtor.free.fr/graphiques/

BOOKS:

Introductory Statistics with R (Statistics and Computing), P. Dalgaard

[available as manual at R project web]

The R Book, MJ. Crawley

Basic Graphs with R:

1. What kind of graphs can produce R ?

2. Software and documentation.

3. Steps and functions.

4. Case I: Tomato gene expression

5. Case II: Sequence size distribution.

3. Steps and functions..

sol genomics network

Graphical procedures step by step:

1) Enable Graphical Device

(X11/QuartzWindows/ by default)

2) Global graphical parameters for the graphical device

3) High level plot function

4) Low level plot function

3. Steps and functions..

Graphical procedures step by step:

1) Enable Graphical Device

bmp(), tiff(), jpeg(), png(), postscript(), pdf().

2) Global graphical parameters for the graphical device

par()

3) High level plot function

plot(), hist(), dotchart(), barplot(), pie()...

4) Low level plot function

legend(), points(), axis(), text(), mtext()...

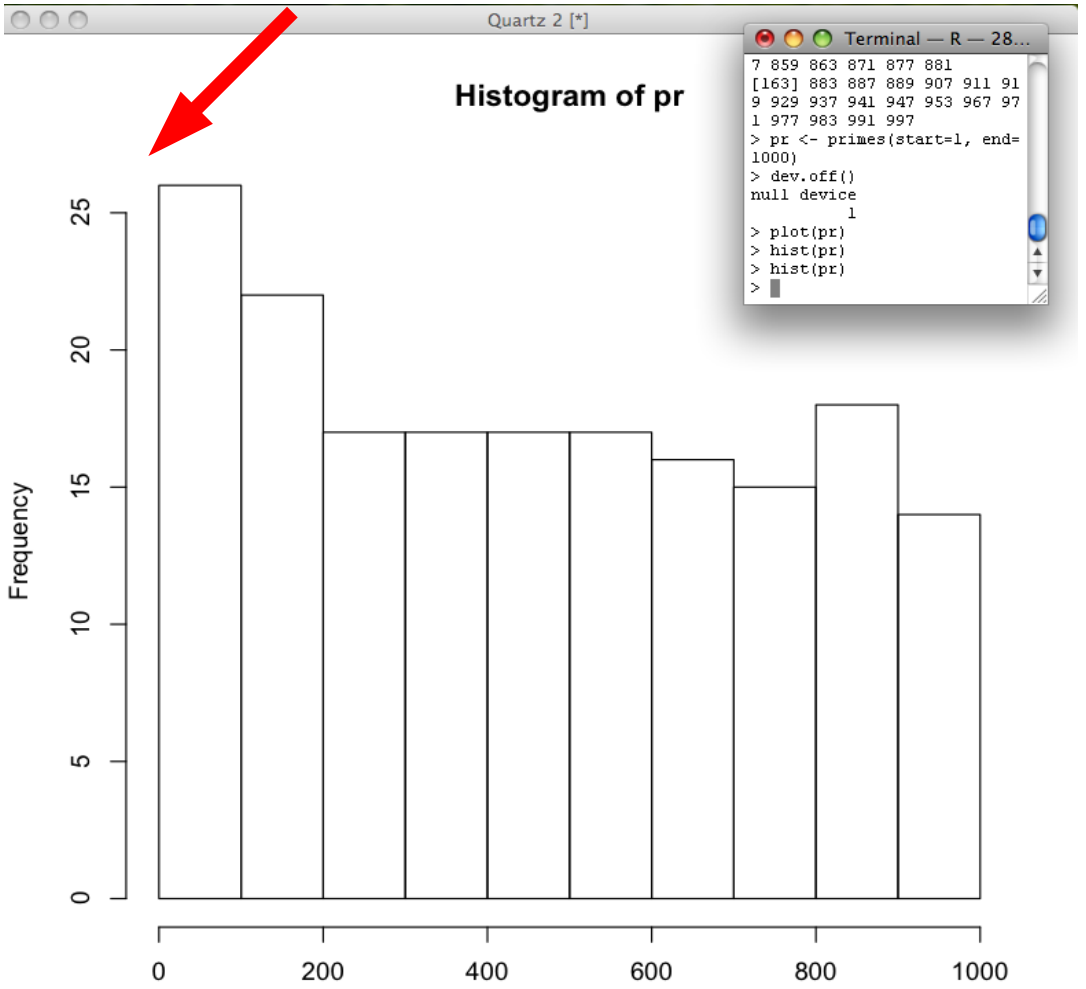
3. Steps and functions..

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>hist(pr)
```

Shorter Y axis

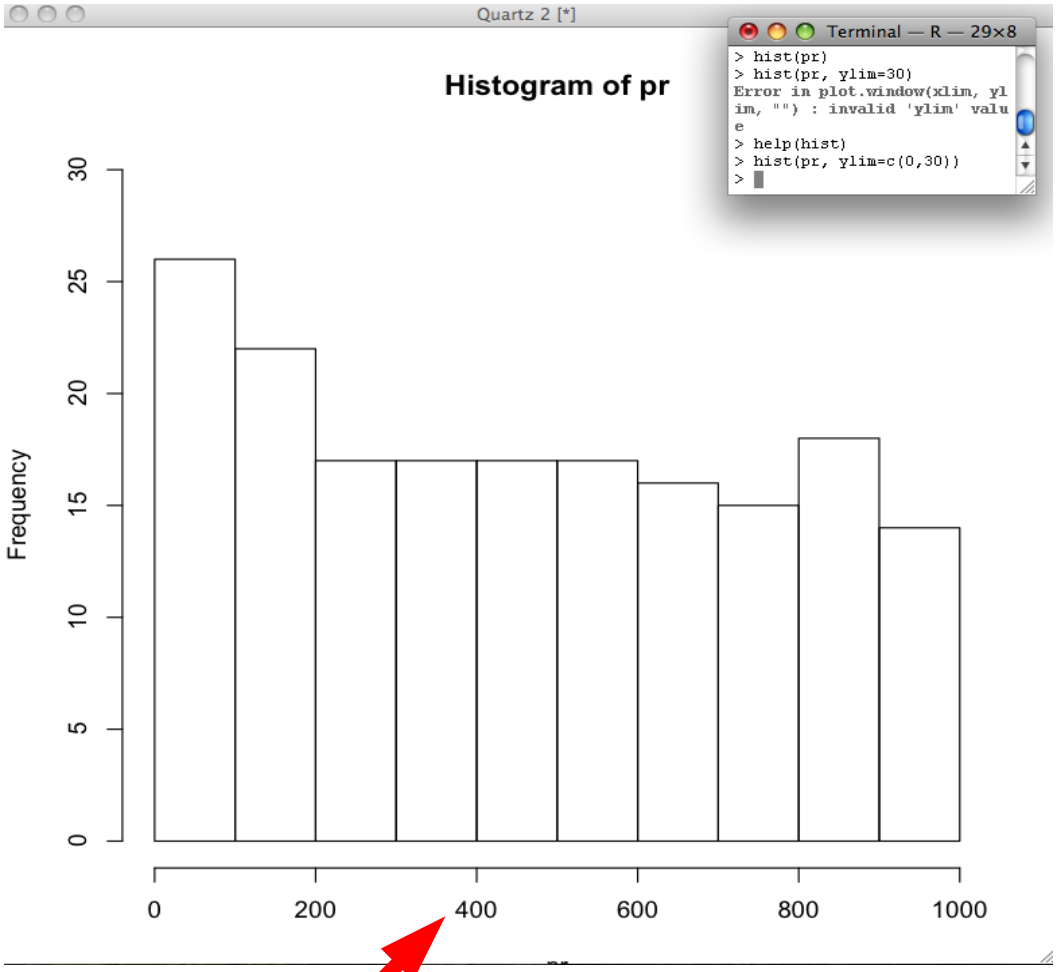


3. Steps and functions..

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>hist(pr, ylim=c(0,30))
```



No X axis title

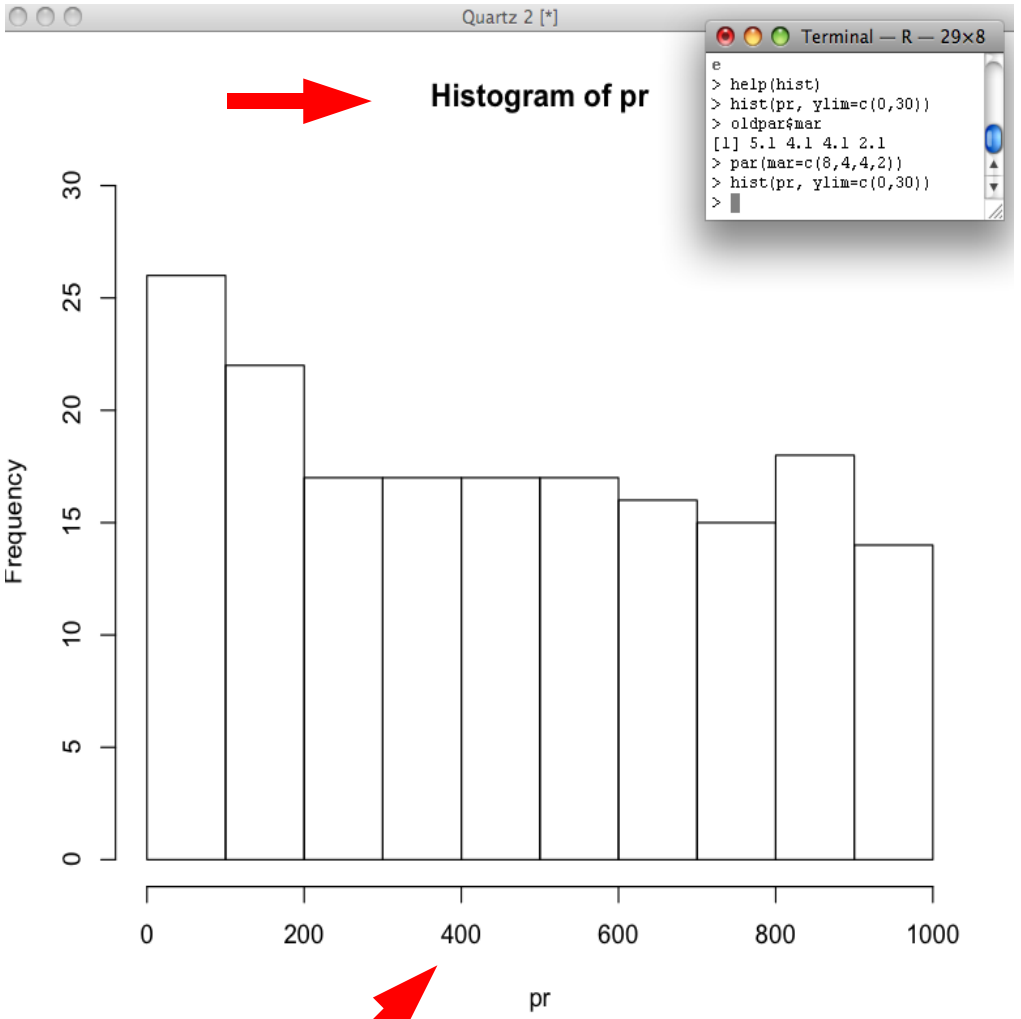
3. Steps and functions..

sol genomics network

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>par( mar=c(8,4,4,2))
>hist(pr, ylim=c(0,30))
```



Wrong titles

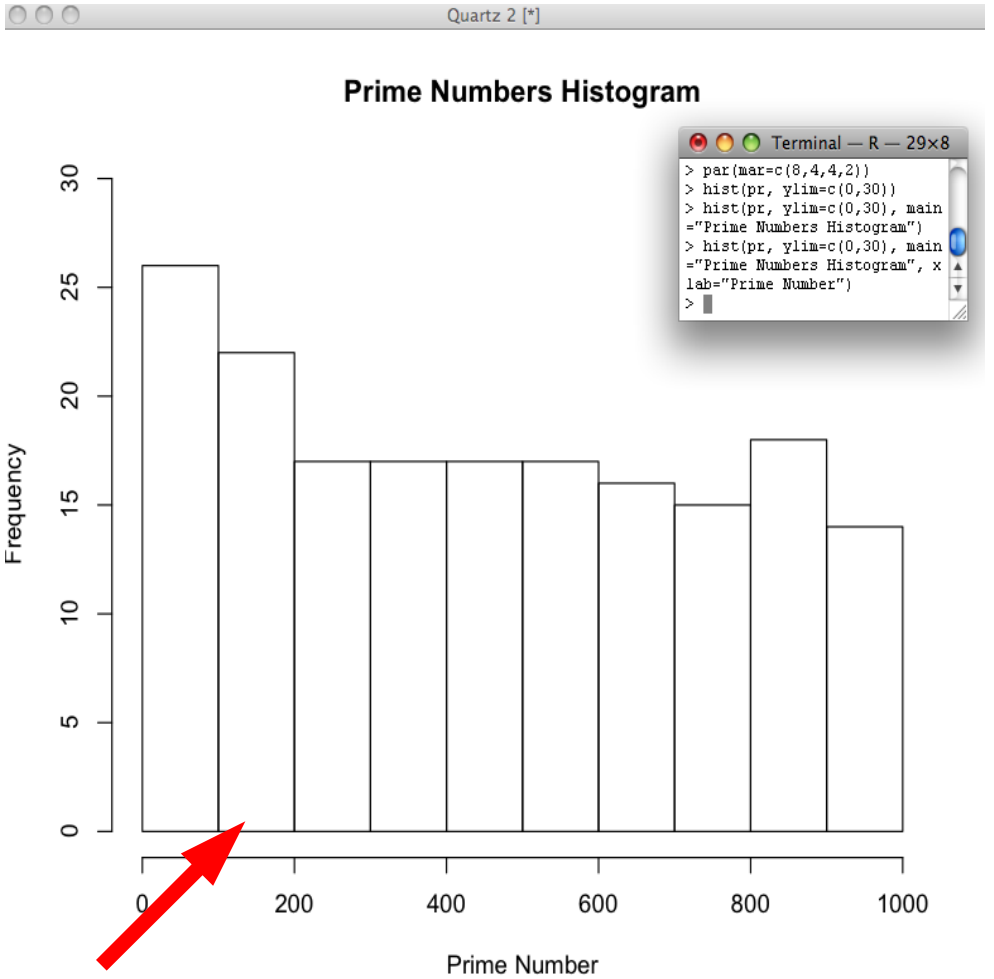
3. Steps and functions..

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>par( mar=c(8,4,4,2))

>hist(pr, ylim=c(0,30),
main="Prime Number Histogram",
xlab="Prime Number")
```



No colors

3. Steps and functions..

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>par( mar=c(8,4,4,2))

>hist(pr,
ylim=c(0,30),
main="Prime Number Histogram",
xlab="Prime Number",
col="green")
```

3. Steps and functions..

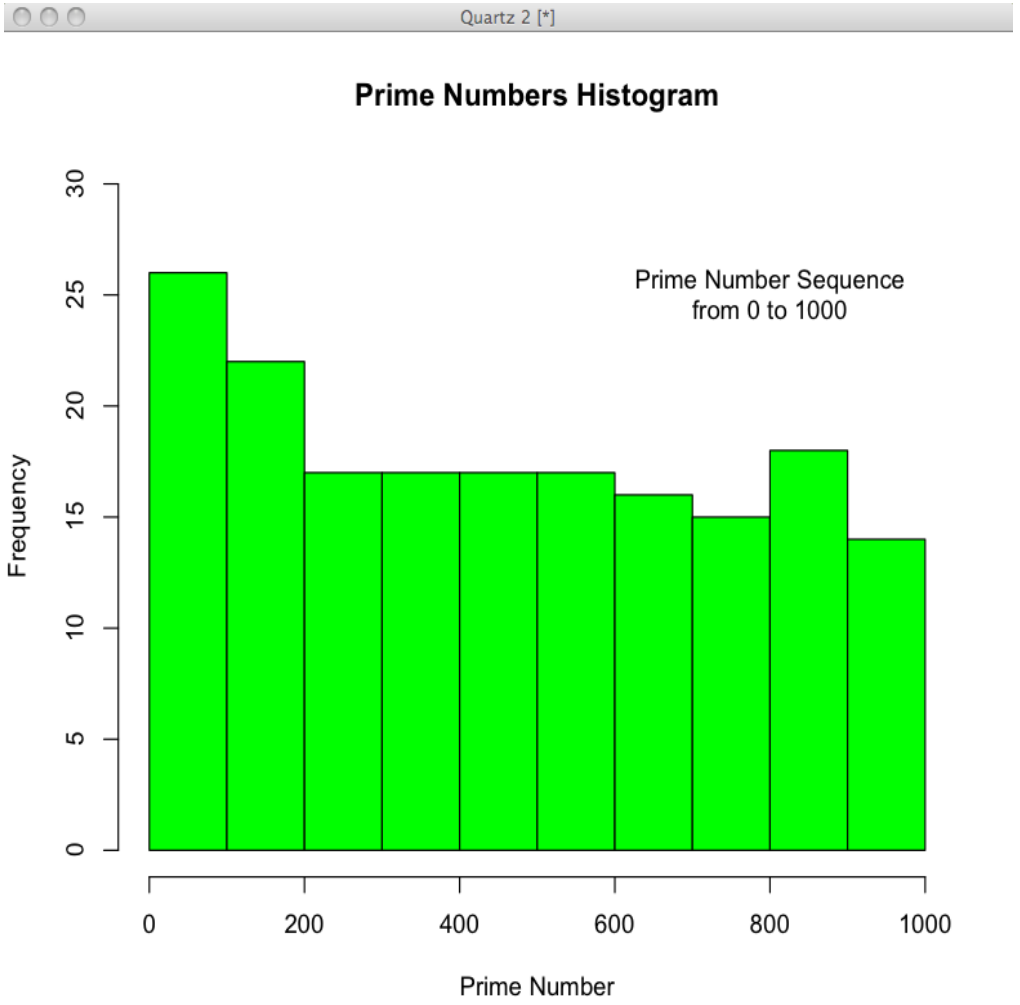
Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>par( mar=c(8,4,4,2))

>hist(pr,
ylim=c(0,30),
main="Prime Number Histogram",
xlab="Prime Number",
col="green")

>text(800, 25, "Prime Number
Sequence\nfrom 0 to 1000")
```



3. Steps and functions..

Example II:

```
>library("schoolmath")
> pr ← primes(start=1, end=1000)

>bmp(filename="MyFile.bmp")

>par( mar=c(8,4,4,2))

>hist(pr,
ylim=c(0,30),
main="Prime Number Histogram",
xlab="Prime Number",
col="green")

>text(800, 25, "Prime Number
Sequence\nfrom 0 to 1000")
```

>library("schoolmath")
> pr ← primes(start=1, end=1000)

1) ENABLE GRAPHICAL DEVICE

2) GLOBAL GRAPHICAL PARAMS.

3) HIGH-LEVEL PLOTTING COMMAND

4) LOW-LEVEL PLOTTING COMMAND

Basic Graphs with R:

1. What kind of graphs can produce R ?

2. Software and documentation.

3. Steps and functions.

4. Case 1: Tomato gene expression

5. Case II: Sequence size distribution.

4. Case 1: Tomato gene expression

Case I: Two Gene Expressions

1- Load the files: “betatubulin.tab”

“cystathioninegsynthase.tab”

2- Calculate the mean each sample for each each gene and store in a new dataframe.

Case I: Two Gene Expressions

1- Solution:

```
csyn <-read.delim(file="cystathioninegsynthase.tab")
btub <-read.delim(file="betatubulin.tab")
```

2- Solution:

```
btub_mean <- apply(btub, 1, mean, na.rm=TRUE)
csyn_mean <- apply(csym, 1, mean, na.rm=TRUE)
```

Case I: Two Gene Expressions

3- Calculate the sd each tissue and each gene and store in a new dataframe.

4- Create two dataframe with means (1) and sd (2)

Case I: Two Gene Expressions

3- Solution:

```
btub_sd <- apply(btub, 1, sd, na.rm=TRUE)
csyn_sd <- apply(csym, 1, sd, na.rm=TRUE)
```

4- Solution:

```
gene_mean <- cbind(btub_mean, csyn_mean)
gene_sd <- cbind(btub_sd, csyn_sd)
```

4. Case 1: Tomato gene expression

Case I: Two Gene Expressions

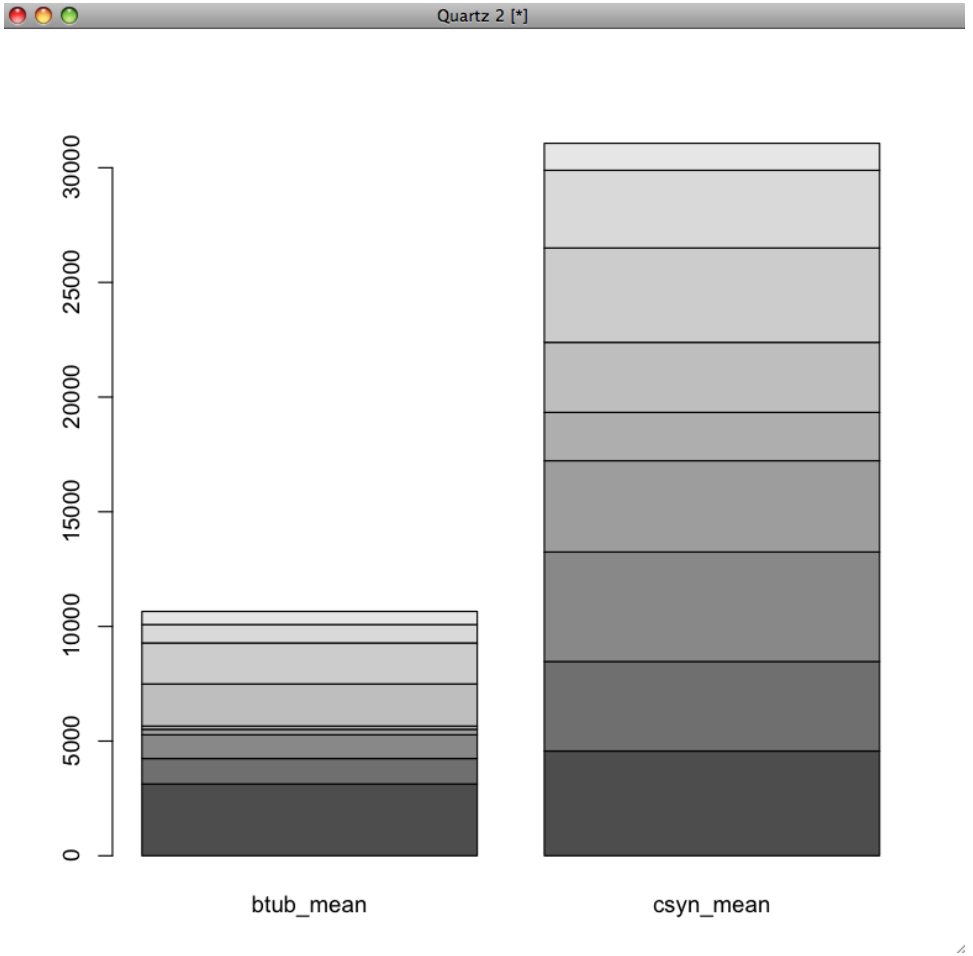
5- Create a simple bargraph with gene expression means.

6- Separate the samples in different columns

Case I: Two Gene Expressions

5- Solution:

barplot(gene_mean)

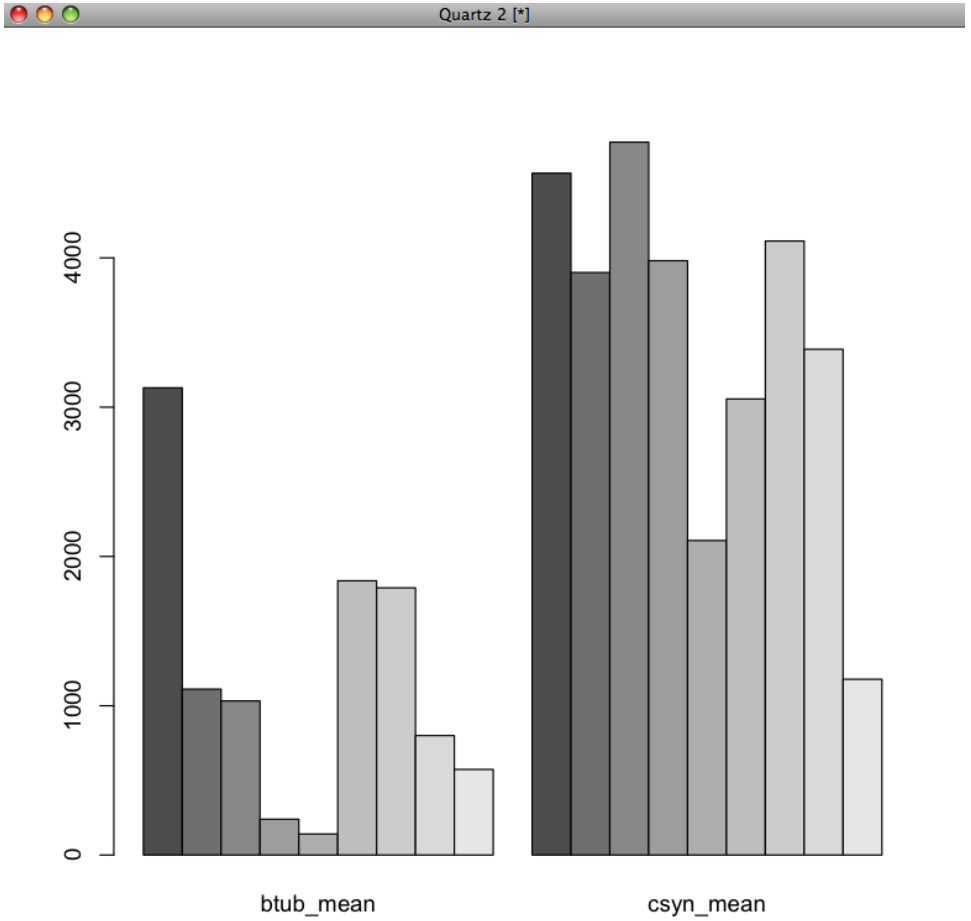


4. Case 1: Tomato gene expression

Case I: Two Gene Expressions

6- Solution:

```
barplot(gene_mean,
    beside=TRUE)
```



Case I: Two Gene Expressions

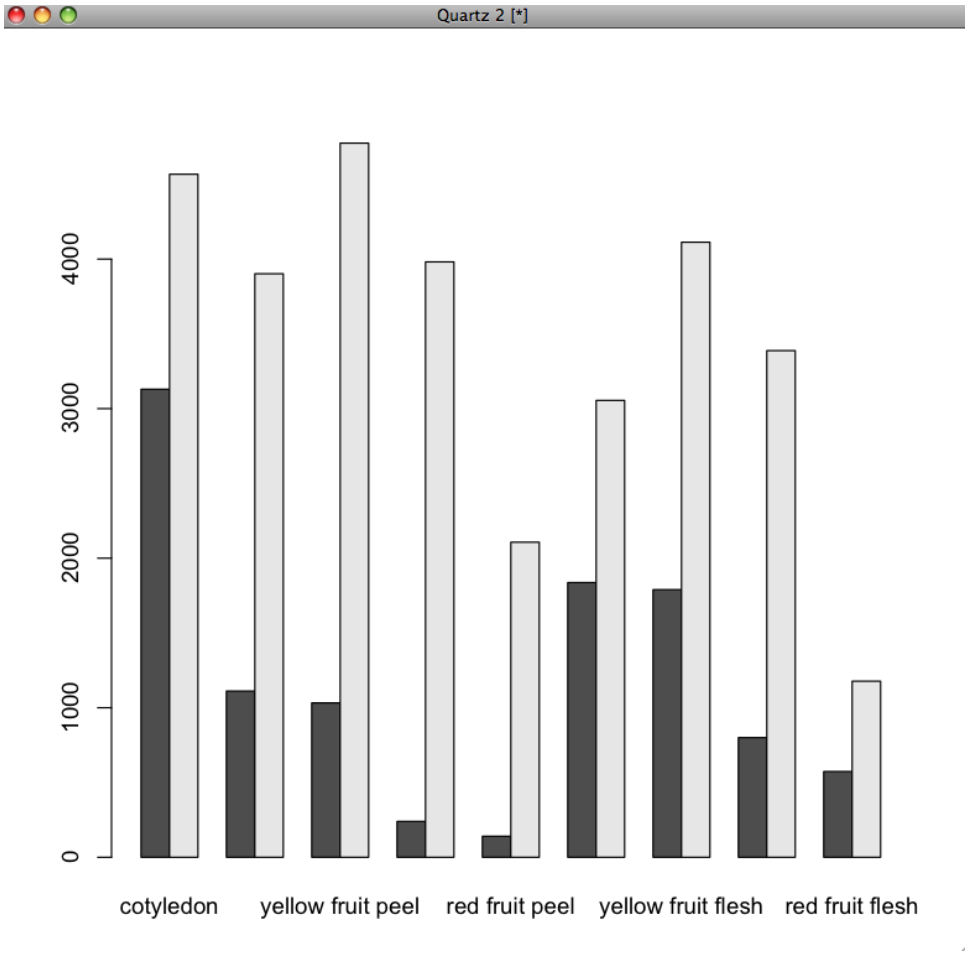
7- Cluster the bars by sample, not by gene

8- Flip the sample labels 90 degrees (vertical)

Case I: Two Gene Expressions

7- Solution:

```
barplot(t(gene_mean),
    beside=TRUE)
```

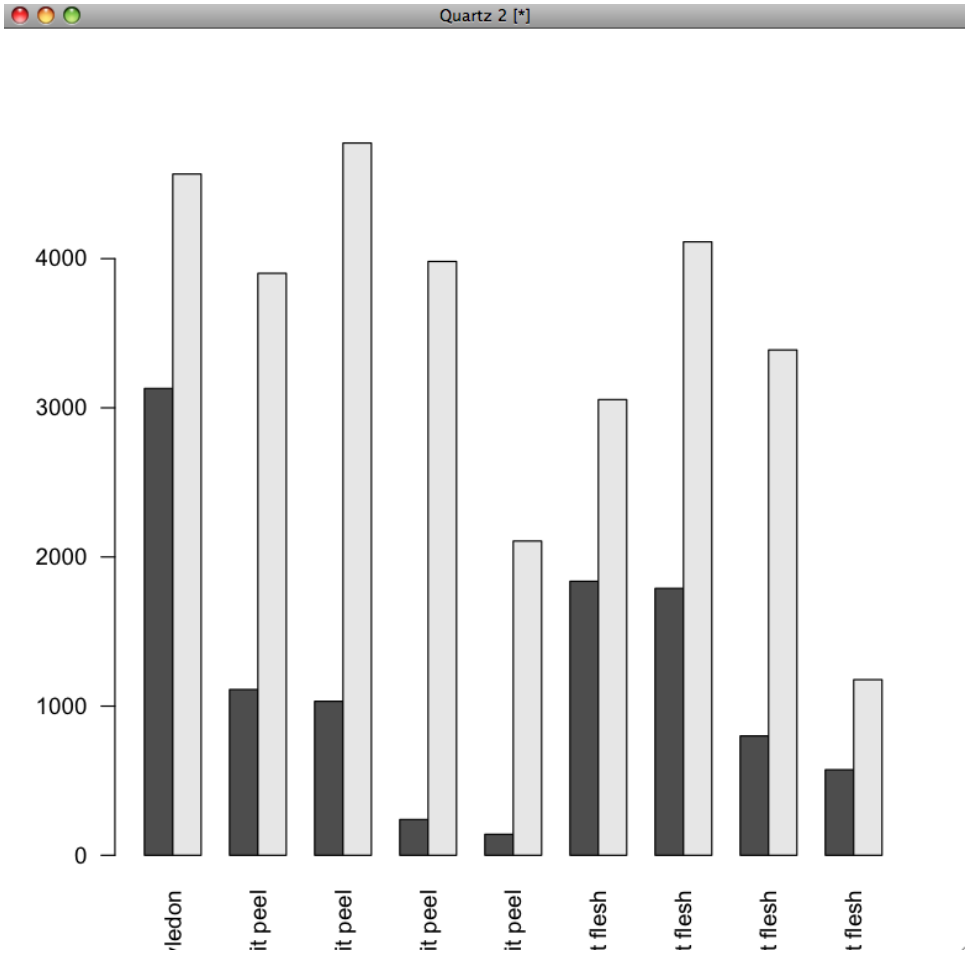


4. Case 1: Tomato gene expression

Case I: Two Gene Expressions

8- Solution:

```
barplot(t(gene_mean),
    beside=TRUE,
    las=2)
```



Case I: Two Gene Expressions

9- Increase the margin size enough to fit the xlabels

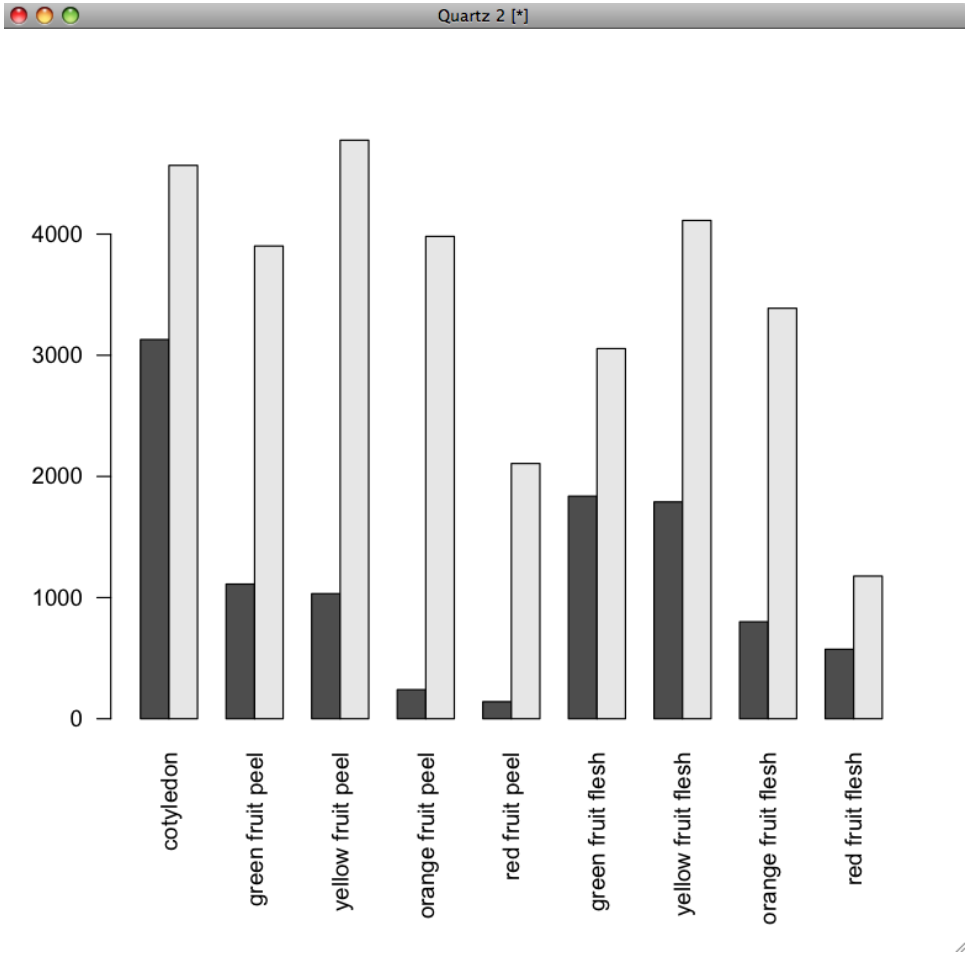
10- Increase the y axis limit to 5000

Case I: Two Gene Expressions

9- Solution:

```
par(mar=c(10,4,4,2))

barplot(t(gene_mean),
    beside=TRUE,
    las=2)
```

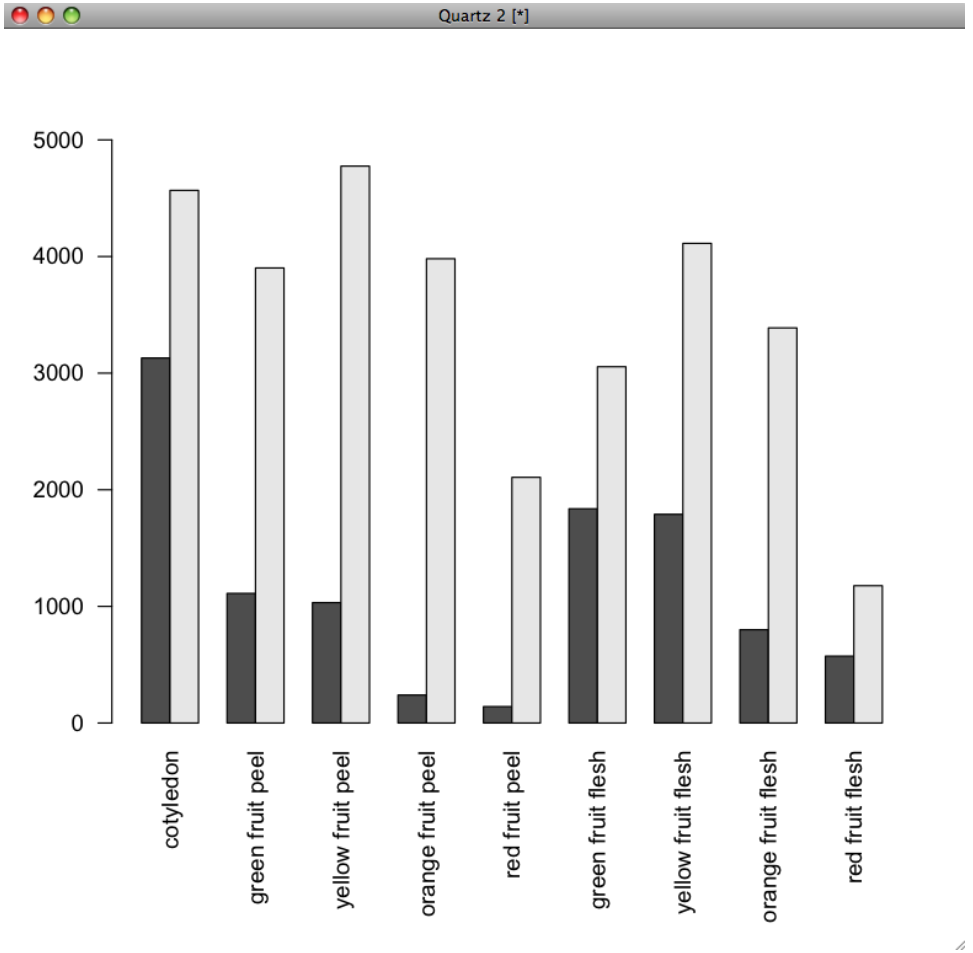


Case I: Two Gene Expressions

10- Solution:

```
par(mar=c(10,4,4,2))

barplot(t(gene_mean),
    beside=TRUE,
    las=2,
    ylim=c(0,5000))
```



4. Case 1: Tomato gene expression

Case I: Two Gene Expressions

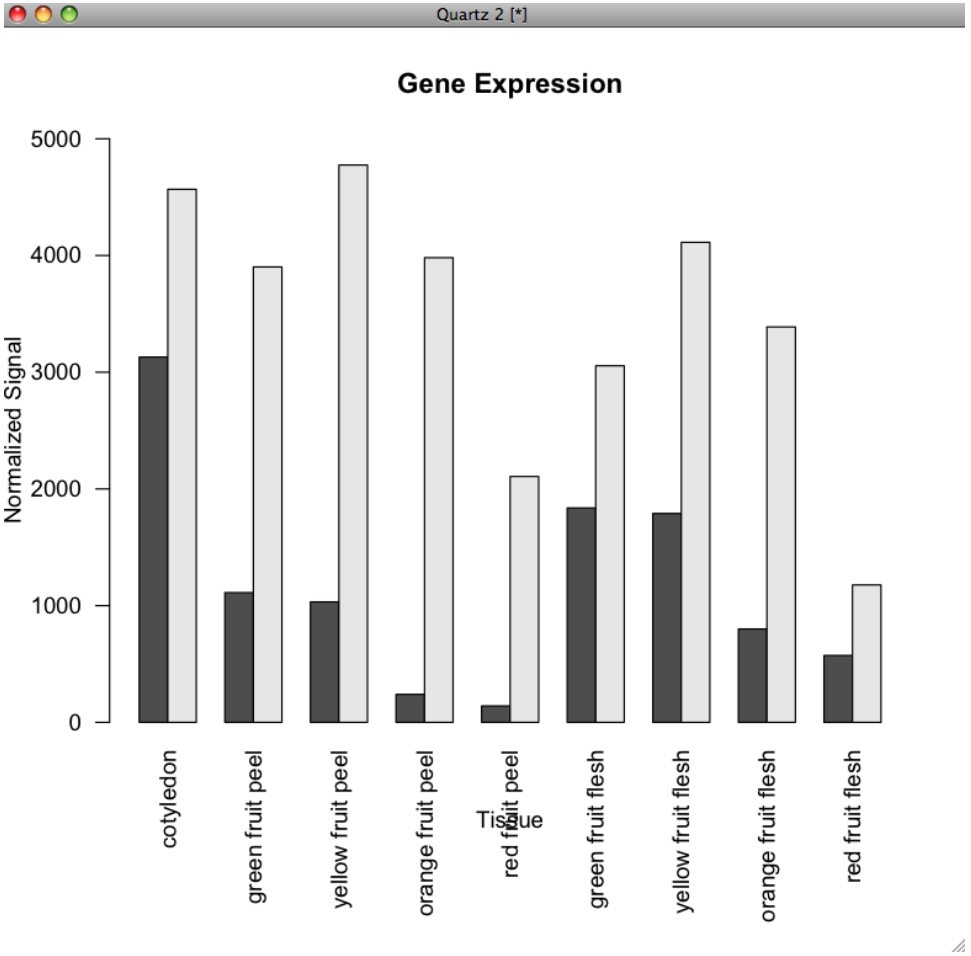
11- Add a title (“Gene Expression”), a x axis (“Tissue”) label and y axis label (“Normalized Signal”)

12- Fix x axis label under x labels and reassign new margins

Case I: Two Gene Expressions

11- Solution:

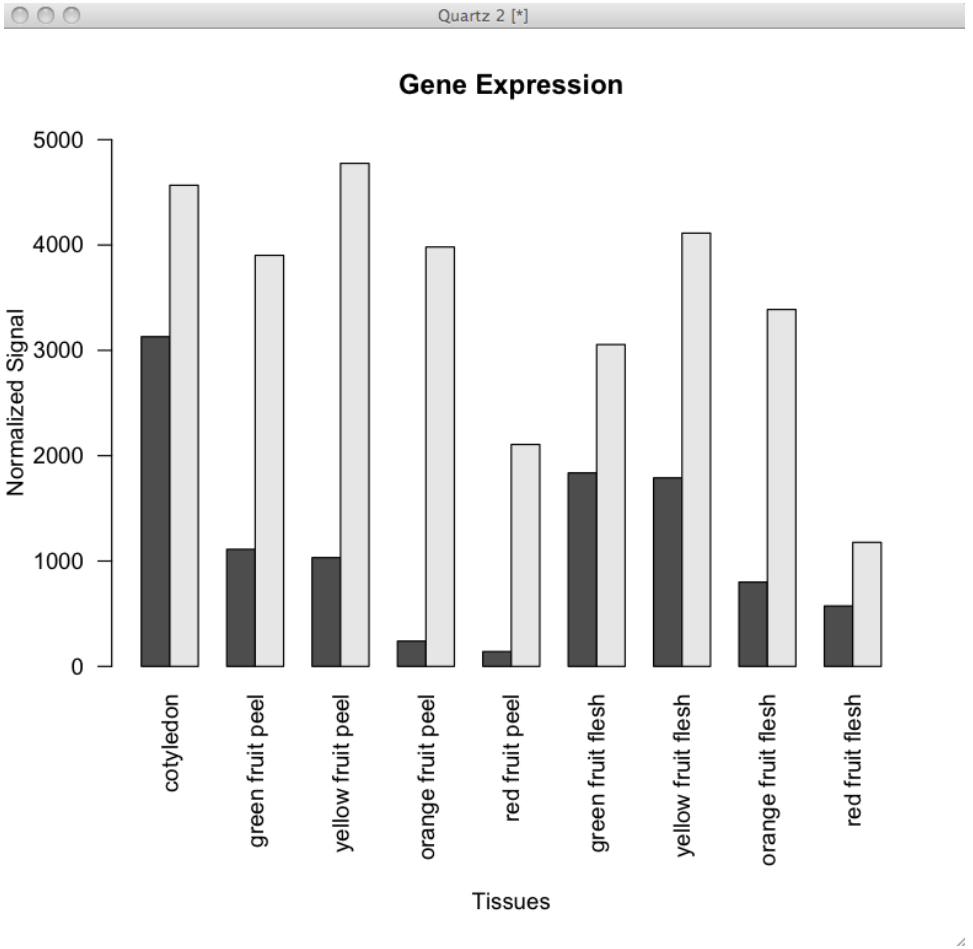
```
par(mar=c(10,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        ylim=c(0,5000),
        main="Gene Expression",
        xlab="Tissue",
        ylab="Normalized Signal")
```



Case I: Two Gene Expressions

12- Solution:

```
par(mar=c(12,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        ylim=c(0,5000),
        main=”Gene Expression”,
        ylab=”Normalized Signal”)
mtext("Tissues", side=1, line=8)
```



Case I: Two Gene Expressions

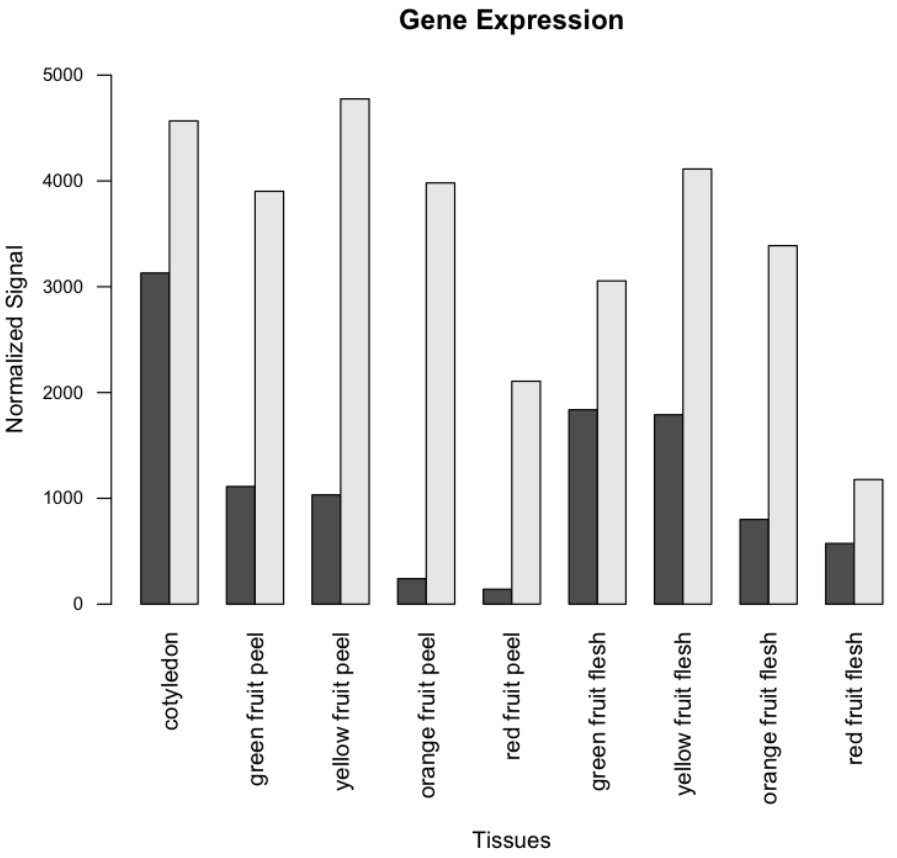
13- Reduce the font size for y. labels 80%

14- Change bar colors to “blue” and “light blue”

Case I: Two Gene Expressions

13- Solution:

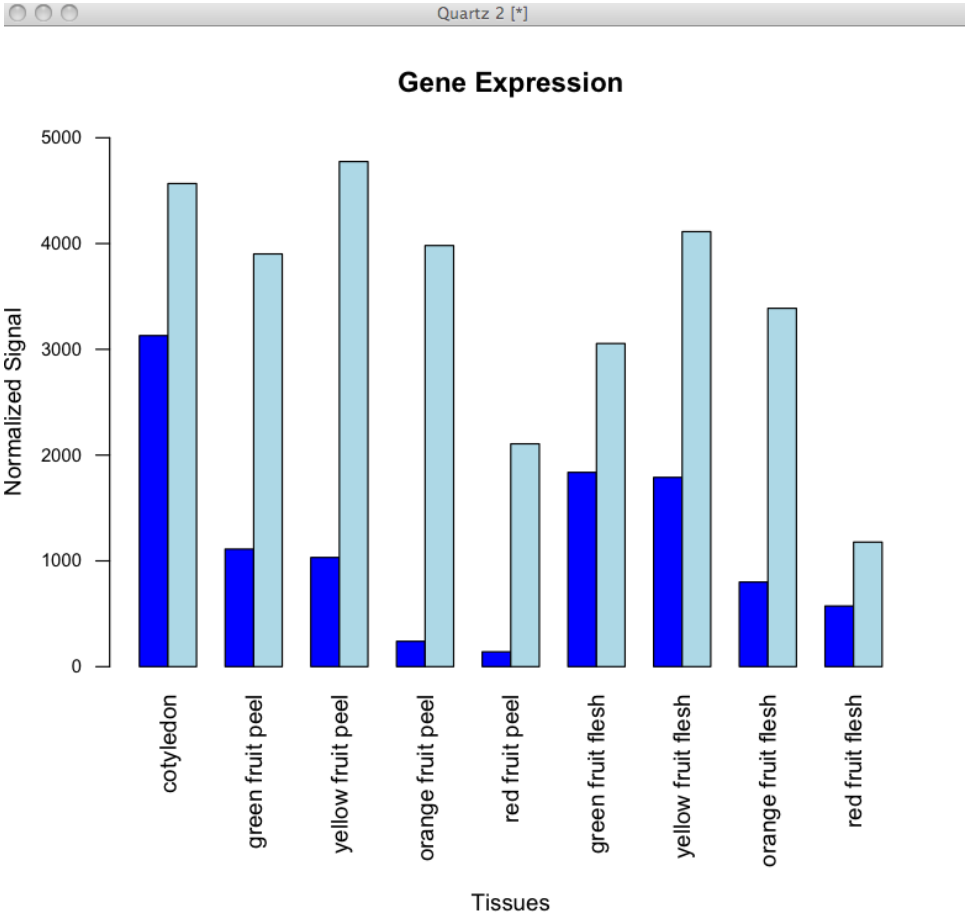
```
par(mar=c(12,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        cex.axis=0.8
        ylim=c(0,5000),
        main=”Gene Expression”,
        ylab=”Normalized Signal”)
mtext("Tissues", side=1, line=8)
```



Case I: Two Gene Expressions

14- Solution:

```
par(mar=c(12,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        cex.axis=0.8,
        col=c("blue", "light blue"),
        ylim=c(0,5000),
        main="Gene Expression",
        ylab="Normalized Signal")
mtext("Tissues", side=1, line=8)
```



Case I: Two Gene Expressions

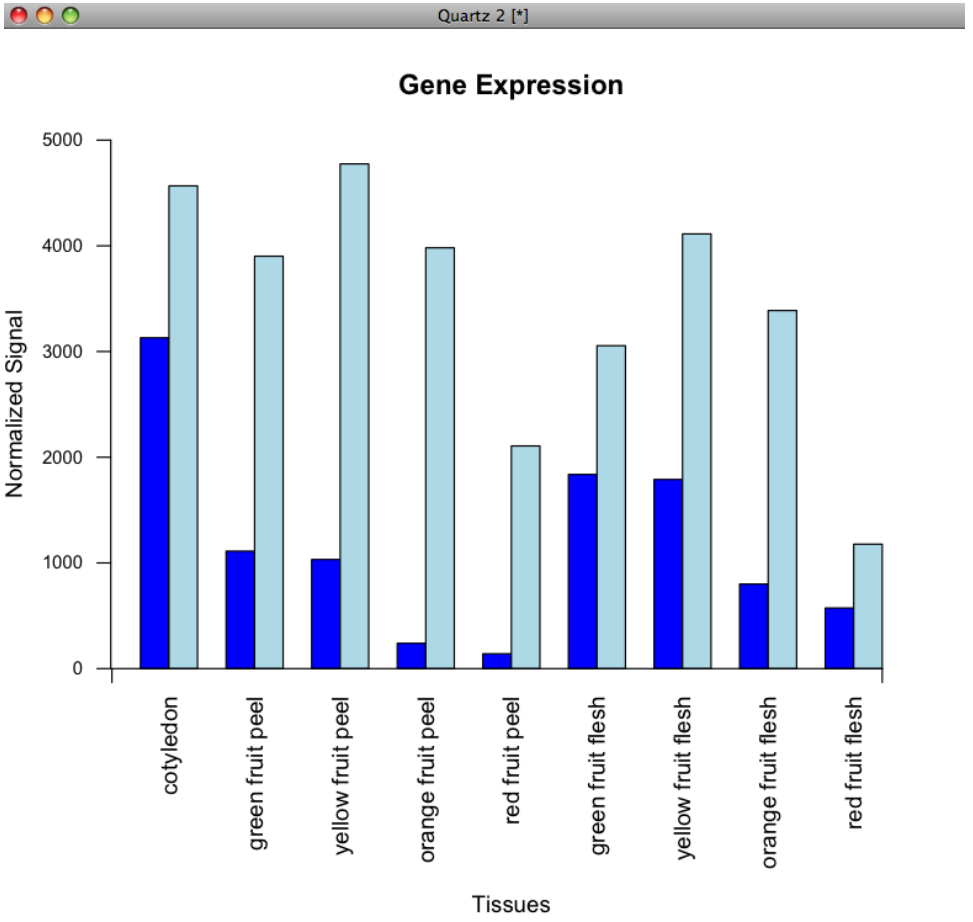
15- Draw a x.axis line

16- Draw a legend

Case I: Two Gene Expressions

15- Solution:

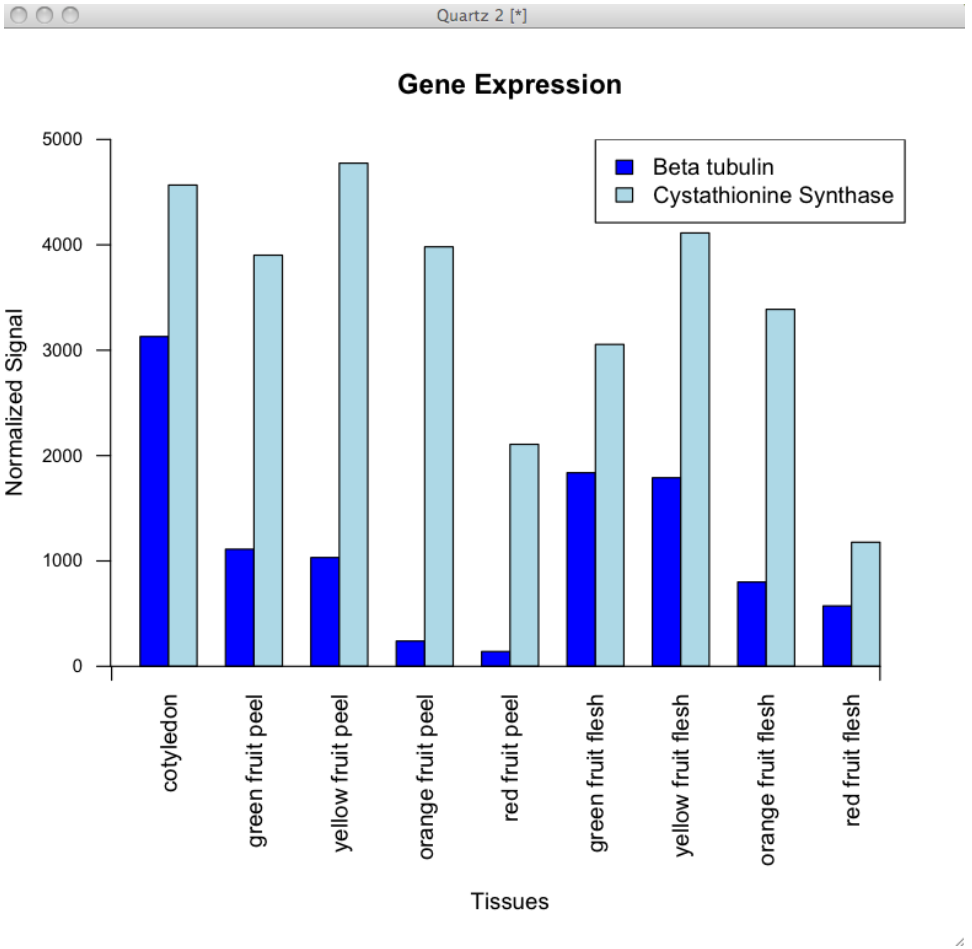
```
par(mar=c(12,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        cex.axis=0.8,
        col=c("blue", "light blue"),
        ylim=c(0,5000),
        main="Gene Expression",
        ylab="Normalized Signal")
mtext("Tissues", side=1, line=8)
axis(1, at=c(0,27), label=FALSE)
```



Case I: Two Gene Expressions

16- Solution:

```
par(mar=c(12,4,4,2))
barplot(t(gene_mean),
        beside=TRUE,
        las=2,
        cex.axis=0.8,
        col=c("blue", "light blue"),
        ylim=c(0,5000),
        main="Gene Expression",
        ylab="Normalized Signal")
mtext("Tissues", side=1, line=8)
axis(1, at=c(0,27), label=FALSE)
legend(17, 5000, c("Beta tubulin","Cystathionine Synthase"), fill=c("blue","light blue"))
```



Case I: Two Gene Expressions

17- Draw the error bars using SD values.

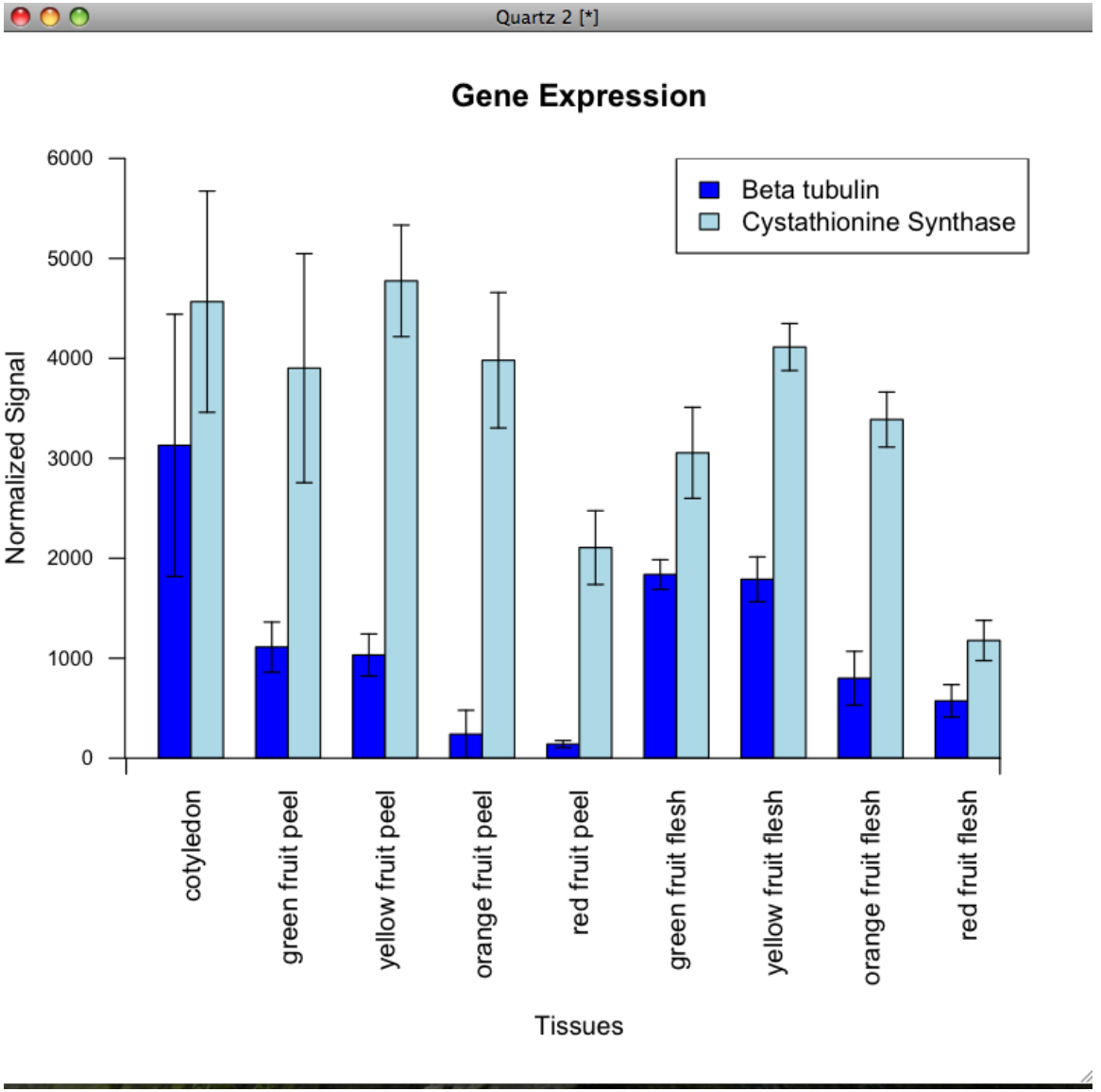
Case I: Two Gene Expressions

17- Solution:

```
par(mar=c(12,4,4,2))
barplot(t(gene_mean), beside=TRUE, las=2, cex.axis=0.8, col=c(“blue”, “light blue”),
        ylim=c(0,5000),  main=”Gene Expression”, ylab=”Normalized Signal”)
mtext("Tissues", side=1, line=8)
axis(1, at=c(0,27), label=FALSE)
legend(17, 5000, c("Beta tubulin","Cystathionine Synthase"), fill=c("blue","light blue"))
btub_xpos ← seq(0,24,3) + 1.5
csyn_xpos ← seq(0,24,3) + 2.5
btub_sdplus ← btub_mean + btub_sd;          btub_sdless ← btub_mean – btub_sd
csyn_sdplus ← csyn_mean + csyn_sd;          csyn_sdless ← csyn_mean – csyn_sd
arrows(btub_xpos, btub_sdpluss, btub_xpos, btub_sdless, code=3, angle=90, length=0.1)
arrows(csyn_xpos, csyn_sdpluss, csyn_xpos, csyn_sdless, code=3, angle=90, length=0.1)
```

Case I: Two Gene Expressions

17- Solution:



Basic Graphs with R:

1. What kind of graphs can produce R ?
2. Software and documentation.
3. Steps and functions.
4. Case I: Tomato gene expression
5. Case II: Sequence size distribution.

5. Case II: Sequence size distribution.

Case II: Sequence size distribution.

454 sequence files in .sff format.

0) Extract the list of accessions/lengths

```
sffinfo -acc -seq 454.sff | grep '>' |
    cut -d ' ' -f1,2 |
        sed -r 's/>//' |
            sed -r 's/ length=/\t/' >
                sffinfo.length
```

Case II: Sequence size distribution.

1) Load the data into R environment.

2) Count number of accessions.

3) Plot an histogram.

Case II: Sequence size distribution.

1) Load the data into R environment.

Solution: seqsL <- read.delim("sffinfo.length", header=FALSE)

2) Count number of accessions.

Solution: summary(seqsL)

3) Plot an histogram.

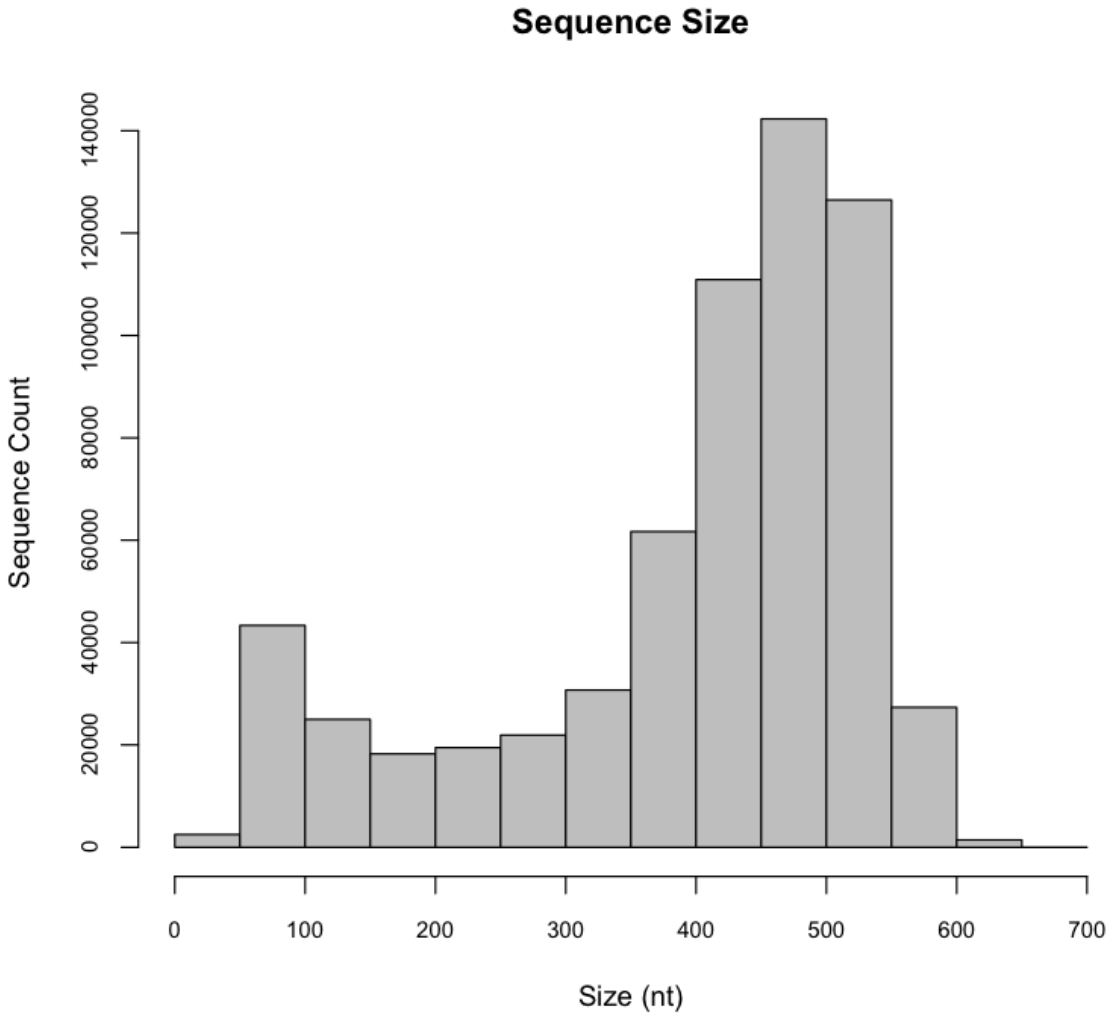
Solution: par(mar = c(8,4,4,2))

```
hist(seqsL$V2, main="Sequence Size", cex.axis=0.8,
ylab="Sequence Count", col=c("gray"), xlab="Size (nt)")
```

Case II: Sequence size distribution.

3) Plot an histogram.

Solution:



Case II: Sequence size distribution.

4) Plot density of the distribution

Solution:

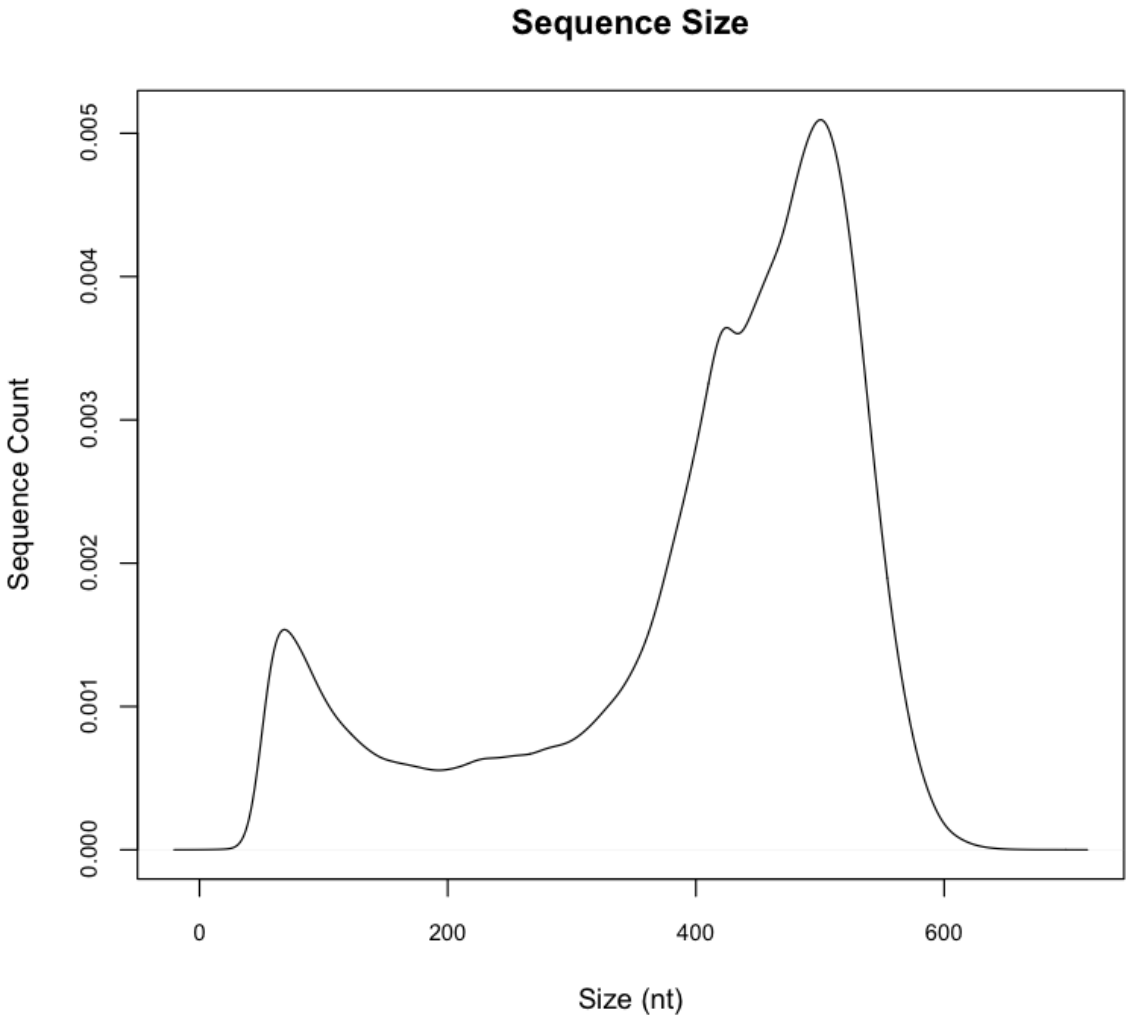
```
par(mar=c(8,4,4,2))

plot(density(seqsL$V2), main="Sequence Size",
cex.axis=0.8, ylab="Sequence Count", xlab="Size (nt)")
```

Case II: Sequence size distribution.

4) Plot density of the distribution

Solution:



Case II: Sequence size distribution.

5) Plot a point for max. y value and a text with the x value.

Case II: Sequence size distribution.

5) Plot a point for max. y value and a text with the x value.

Solution:

```
D ← density(seqsL$V2)
Dmtx ← cbind(D$x, D$y)
XfYmax ← Dmtx[order(Dmtx[,2], decreasing=TRUE),][1]
YfYmax ← Dmtx[order(Dmtx[,2], decreasing=TRUE),][1]
points(XfYmax, YfYmax, cex=1.5, pch=16)
text(XfYmax + 50, YfYmax, round(XfYmax, digit=3))
```

Case II: Sequence size distribution.

5) Plot a point for max. y value and a text with the x value.

Solution: