

```
#####
# Plotting exercises
#####

# Example from Quick-R
attach(mtcars)
plot(wt,mpg,main="Scatterplot Example")
abline(lm(mpg~wt), col="red") # regression line (y~x)
lines(lowess(wt,mpg), col=terrain.colors(30)[20]) # lowess line (x,y)

#####
# Below is an example that builds slowly, from plotting expression values for
# a single gene up to automating all aspects of the plot so it can be done
# for many genes in a large data.frame.

# Read in data
infile = 'dd_dev_transcriptome.txt'
data = read.table(infile, header=TRUE)
summary(data)

# Make a plot of expression over time (cols 3-9) for the first gene in the list
plot(c(0,4,8,12,16,20,24),data[1,3:9]) # Note, in this case, you must specify yourself the x-values: c(0,4,8,12,16,20,24)

# Make it a line plot -- see ?plot for different options
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o')

# Make it a nice color
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col='dodgerblue3')

# Change the x-axis and y-axis labels
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col='dodgerblue3', xlab='Timepoint (hrs)', ylab='Normalized Number of Reads' )

# Add a title
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col='dodgerblue3', xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main="Expression of Gene
DDB_G0267178 During Development" )

# Make Title a Variable
curr.title = paste("Expression of Gene ", data[1,1], " During Development")
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col='dodgerblue3', xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)

# Plot with a randomly chosen color
cols=colors()
curr.title = paste("Expression of Gene ", data[1,1], " During Development")
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=sample(colors(),1), xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)

# Plot and save to file
outfile = paste("Plot_of_", data[1,1], ".pdf", sep="")
pdf(file=outfile)
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col='dodgerblue3', xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)
dev.off()

# Plot and save to file in a Loop for the first 10 genes
for (i in 1:10) {
  outfile = paste("Plot_of_", data[i,1], ".pdf", sep="")
  pdf(file=outfile)
  plot(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=sample(colors(),1), xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)
  dev.off()
}

# Plot first 10 genes, save to a SINGLE file
outfile = paste("Plot_of_", data[1,1], ".pdf", sep="")
pdf(file=outfile)
for (i in 1:10) {
  plot(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=sample(colors(),1), xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)
}
dev.off()

# Plot the first 10 genes, all on the same GRAPH
# Note: This works by plotting the first plot with plot() and subsequent plots using points(), which doesn't overwrite the previous plot each time
outfile = paste("Plot_of_First10_Genes.pdf", sep="")
pdf(file=outfile)

plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=sample(colors(),1), xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)

for (i in 2:10) {
  points(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=sample(colors(),1), xlab='Timepoint (hrs)', ylab='Normalized Number of Reads',main=curr.title)
}

dev.off()

# Fix the max y-axis and give a better title
max_value = max(data[1:10,3:9]) # Determine the max value of the data being plotted and set to variable "max_value"
```

```

curr.title = "Plot of First 10 Genes" # Create a better plot title
outfile = "Plot_of_First10_Genes.pdf"
pdf(file=outfile)

plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=terrain.colors(10)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title, ylim=c(0,max_value)) # see par() for details on these additional arguments

for (i in 2:10) {
curr.col=terrain.colors(10)[i]
points(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=terrain.colors(10)[i], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title)
}
dev.off()

# Add a legend
max_value = max(data[1:10,3:9]) # Determine the max value of the data being plotted and set to variable "max_value"
curr.title = "Plot of First 10 Genes" # Create a better plot title

outfile = paste("Plot_of_First10_Genes.pdf", sep="")
pdf(file=outfile)
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=terrain.colors(10)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title, ylim=c(0,max_value))

for (i in 2:10) {
curr.col=terrain.colors(10)[i]
points(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=terrain.colors(10)[i], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title)
}
legend("topright", legend=data[1:10,1],col=terrain.colors(10),lty=1,pch=19)
dev.off()

# Make number of genes (rows) to plot a variable
num_genes = 20
max_value = max(data[1:num_genes,3:9]) # Determine the max value of the data being plotted and set to variable "max_value"
curr.title = paste("Plot of First ", num_genes, " Genes", sep="")
outfile = paste("Plot_of_First", num_genes, "_Genes.pdf", sep="")

pdf(file=outfile)
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title, ylim=c(0,max_value))

for (i in 2:num_genes) {
curr.col=terrain.colors(num_genes)[i]
points(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=terrain.colors(num_genes)[i], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title)
}
legend("topright", legend=data[1:num_genes,1],col=terrain.colors(num_genes),lty=1,pch=19,cex=0.4)
dev.off()

##### PLOTTING MULTIPLE PANELS

# Set parameter for number of number of rows and columns
par(mfrow=c(2,2)) # two rows, two columns
plot(c(0,4,8,12,16,20,24),data[1,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=paste("Gene ",data[1,1]), ylim=c(0,max_value))
plot(c(0,4,8,12,16,20,24),data[2,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=paste("Gene ",data[2,1]), ylim=c(0,max_value))
plot(c(0,4,8,12,16,20,24),data[3,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=paste("Gene ",data[3,1]), ylim=c(0,max_value))
plot(c(0,4,8,12,16,20,24),data[4,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=paste("Gene ",data[4,1]), ylim=c(0,max_value))

##### SAME, but USE A LOOP
num_genes=20 # Must be divisible by 5!
num_rows=num_genes/5
num_cols=num_genes/num_rows
par(mfrow=c(num_rows,num_cols))

for (i in 1:num_genes){
curr.gene=data[i,1]
curr.title=paste("Gene ", curr.gene, sep="")
plot(c(0,4,8,12,16,20,24),data[i,3:9],type='o',col=terrain.colors(num_genes)[1], pch=19, xlab='Timepoint (hrs)', ylab='Normalized Number of
Reads',main=curr.title, ylim=c(0,max_value))
}

#### BARPLOT
par(mfrow=c(1,1)) #reset par() back to a single plot
d=data[data[,6]<=3000,]
boxplot(d[,6]~d$type)

```

```
##### HEATMAPS -- note: takes a while to plot
heatmap(as.matrix(data[1:2000,3:9]),labRow=NULL)
```

```
install.packages('RColorBrewer')
require('RColorBrewer')
hmc1=brewer.pal(5,"YlGnBu")
heatmap(as.matrix(data[1:2000,3:9]),labRow=NULL,col=hmc1)
```

```
install.packages('gplots')
require('gplots')
a=data[order(data$tp.hr00),3:9]
heatmap.2(as.matrix(a),Rowv=FALSE,scale='row',trace='none',dendrogram='both')
```