DSSH 6301 - HW 02 Solutions

Here are suggested solutions to the homework assignment. Note that many problems have multiple solutions, so we present here merely one example. If you did not have points taken off for a question, we considered your solution correct. If you have questions about any of these solutions or your own graded homework, please direct questions to Matt Harrigan m.harrigan@neu.edu .

Problem 1

Part a

Note how (1) simple functions and if-then statements can be defined without the explicit use of brackets; (2) the function eliminates NA elements; and (3) it returns "NaN" (Not a Number) if the input is non-numeric. Note also how how all functions return the last quantity without needing an explicit use of "return".

```
mean_fn <- function(x) if (is.numeric(x)) sum(x, na.rm=T)/length(x[!is.na(x)]) else NaN
mean_fn(1:4)
```

```
## [1] 2.5
```

```
mean_fn(c("cat", "dog", "frog"))
```

```
## [1] NaN
```

A simpler version of the function without the error handling, and using brackets and an explicit "return" might be:

```
mean_fn <- function(x){
  mn <- sum(x)/length(x)
  return(mn)
}
```

Part b

```
fn <- function(x) if(sum(x[1:2]) > sum(x[3:4])) x else 0
fn(1:4)
```

```
## [1] 0
```

```
fn(4:1)
```

```
## [1] 4 3 2 1
```

Part c

The two examples given here show the calculation of the Fibonacci sequence by appending a single element to the end of a vector.

```
fib_seq_concat1 <- function(n) {
  if (n < 1)
    stop("n must be a positive integer") # Stops on invalid input

  if (n == 1)
    return(1)

  if (n == 2)
    return(c(1, 1))

  fib <- c(1, 1)

  for (i in 3:n)
    fib <- c(fib, fib[i-2] + fib[i-1])

  return(fib)
}

fib_seq_concat2 <- function(n) {
  if (n < 1)
    stop("n must be a positive integer") # Stops on invalid input

  if (n == 1)
    return(1)

  if (n == 2)
    return(c(1, 1))

  fib <- c(1, 1)

  for (i in 3:n)
    fib[i] <- fib[i-2] + fib[i-1]

  return(fib)
}

fib_seq_concat1(10)
```

```
##  [1]  1  1  2  3  5  8 13 21 34 55
```

```
fib_seq_concat2(10)
```

```
##  [1]  1  1  2  3  5  8 13 21 34 55
```

Part d

```r
m1 <- matrix(1:16, nrow=4, ncol=4)
apply(m1, 1, mean_fn)
```

```
## [1]  7  8  9 10
```

Problem 2

Part a

Only measurements where both Wind and Ozone are not NA are aggregated when using the aggregate function like this. Thus the discrepancy in means for month 5 that you see below.

```r
aggregate(cbind(Wind, Ozone) ~ Month, data=airquality, mean)
```

```
##   Month      Wind    Ozone
## 1     5 11.457692 23.61538
## 2     6 12.177778 29.44444
## 3     7  8.523077 59.11538
## 4     8  8.565385 59.96154
## 5     9 10.075862 31.44828
```

```r
mean(airquality$Wind[airquality$Month == 5])
```

```
## [1] 11.62258
```

```r
mean(airquality$Wind[airquality$Month == 5 & !is.na(airquality$Ozone)])
```

```
## [1] 11.45769
```

Part b

```r
authors <- data.frame(
    surname = c("Tukey", "Venables", "Tierney", "Ripley", "McNeil"),
    nationality = c("US", "Australia", "US", "UK", "Australia"))

books <- data.frame(
    name = c("Tukey", "Venables", "Tierney", "Ripley", "Ripley", "McNeil",
             "R Core"),
    title = c("Exploratory Data Analysis", "Modern Applied Statistics ...",
              "LISP-STAT", "Spatial Statistics", "Stochastic Simulation",
              "Interactive Data Analysis", "An Introduction to R"))

merge(authors, books, by.x="surname", by.y="name", all=T)
```

```
##    surname nationality                         title
## 1   McNeil   Australia     Interactive Data Analysis
## 2   Ripley          UK            Spatial Statistics
## 3   Ripley          UK         Stochastic Simulation
## 4  Tierney          US                     LISP-STAT
## 5    Tukey          US     Exploratory Data Analysis
## 6 Venables   Australia Modern Applied Statistics ...
## 7   R Core        <NA>          An Introduction to R
```

Part c

Note that using `cat` rather than `print` preserves the carriage returns in the output.

```
str <- "To be, or not to be- that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles,
And by opposing end them. To die - to sleep -
No more..."

str <- gsub("to", 2, str, ignore.case = T)
cat(str)
```

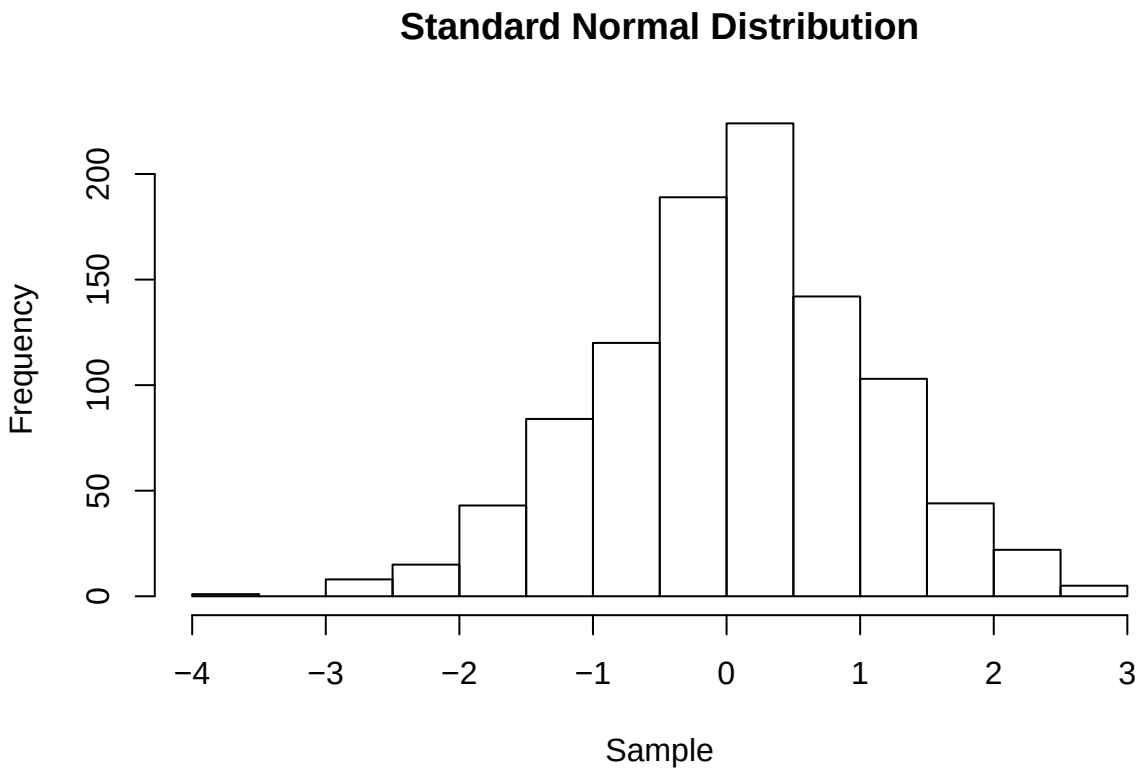
```
## 2 be, or not 2 be- that is the question:
## Whether 'tis nobler in the mind 2 suffer
## The slings and arrows of outrageous fortune
## Or 2 take arms against a sea of troubles,
## And by opposing end them. 2 die - 2 sleep -
## No more...
```

Problem 3

Part a

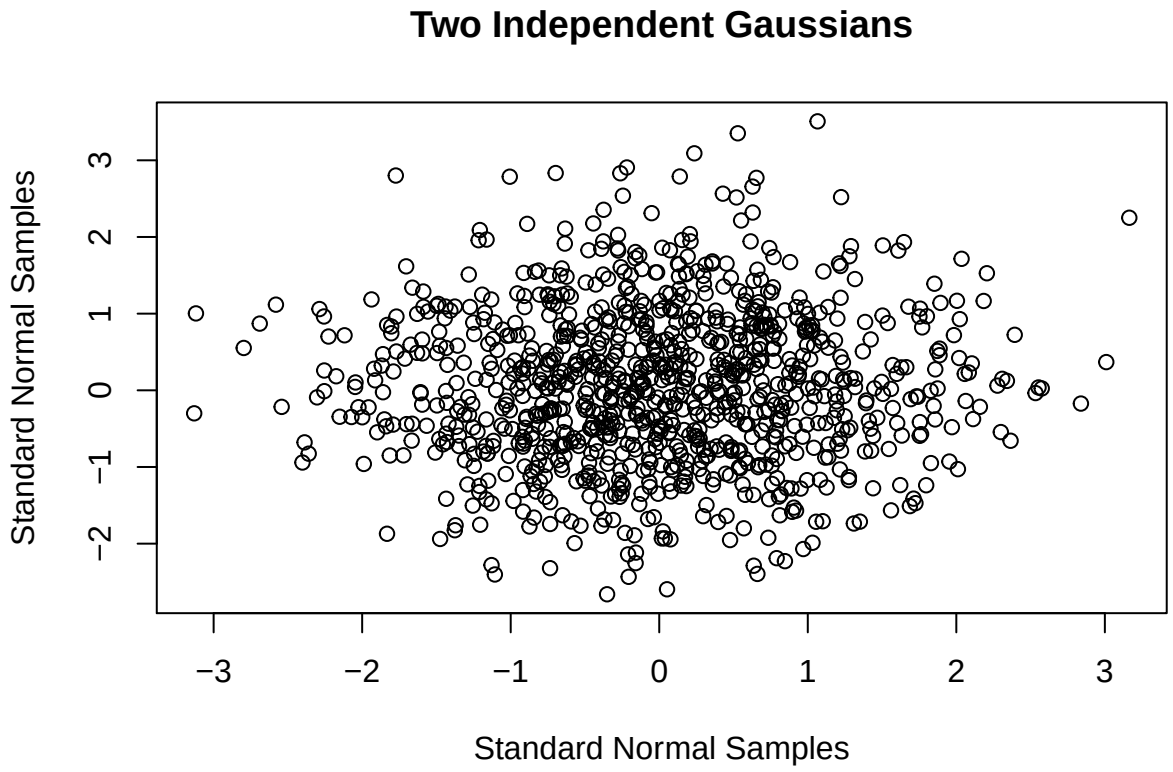
We create 1000 samples from a normal distribution here. More information on this function and its uses will be given later in the course.

```
data <- rnorm(1000)
hist(data, main="Standard Normal Distribution", xlab="Sample")
```

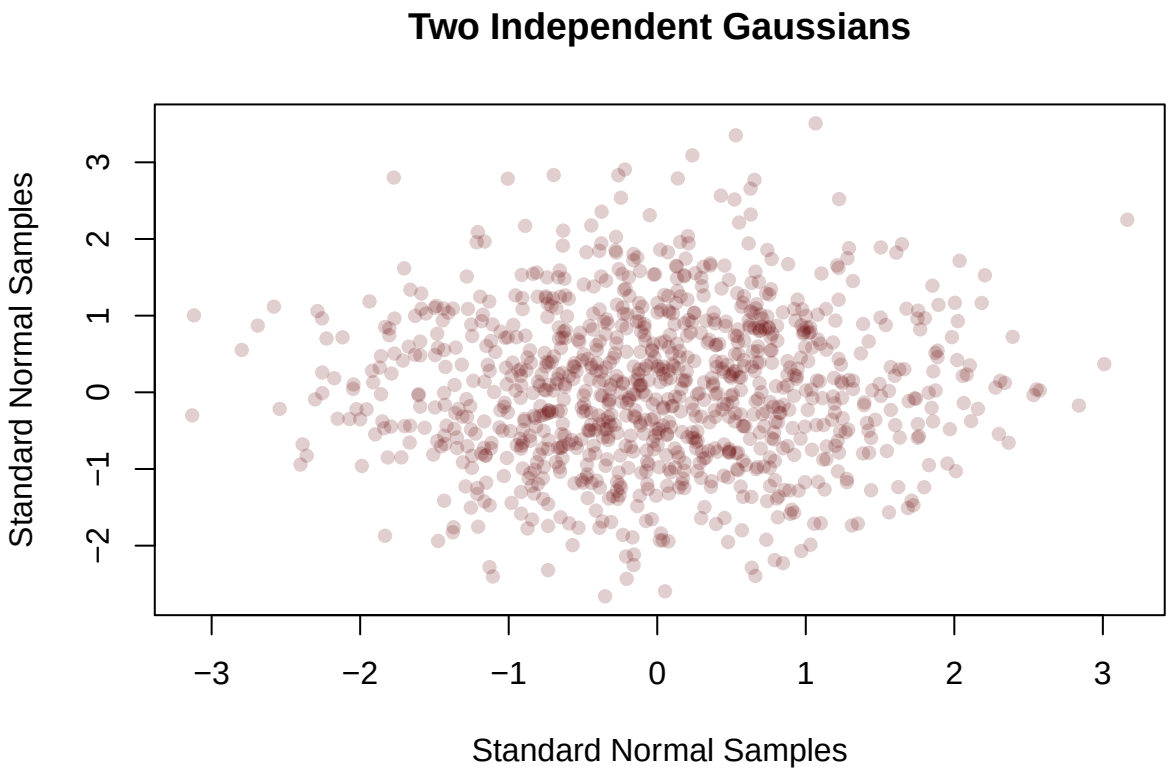


Part b

```
x <- rnorm(1000)
y <- rnorm(1000)

xlab <- "Standard Normal Samples"
ylab <- "Standard Normal Samples"

plot(x, y, main="Two Independent Gaussians", xlab=xlab, ylab=ylab)
```




```
# We add some transparency here for this high density plot.
# The 4th argument adjusts transparency level. See ?rgb for more info.
col <- rgb(100, 0, 0, 50, maxColorValue=255)

plot(x, y, main="Two Independent Gaussians", col=col, pch=16, xlab=xlab,
     ylab=ylab)
```



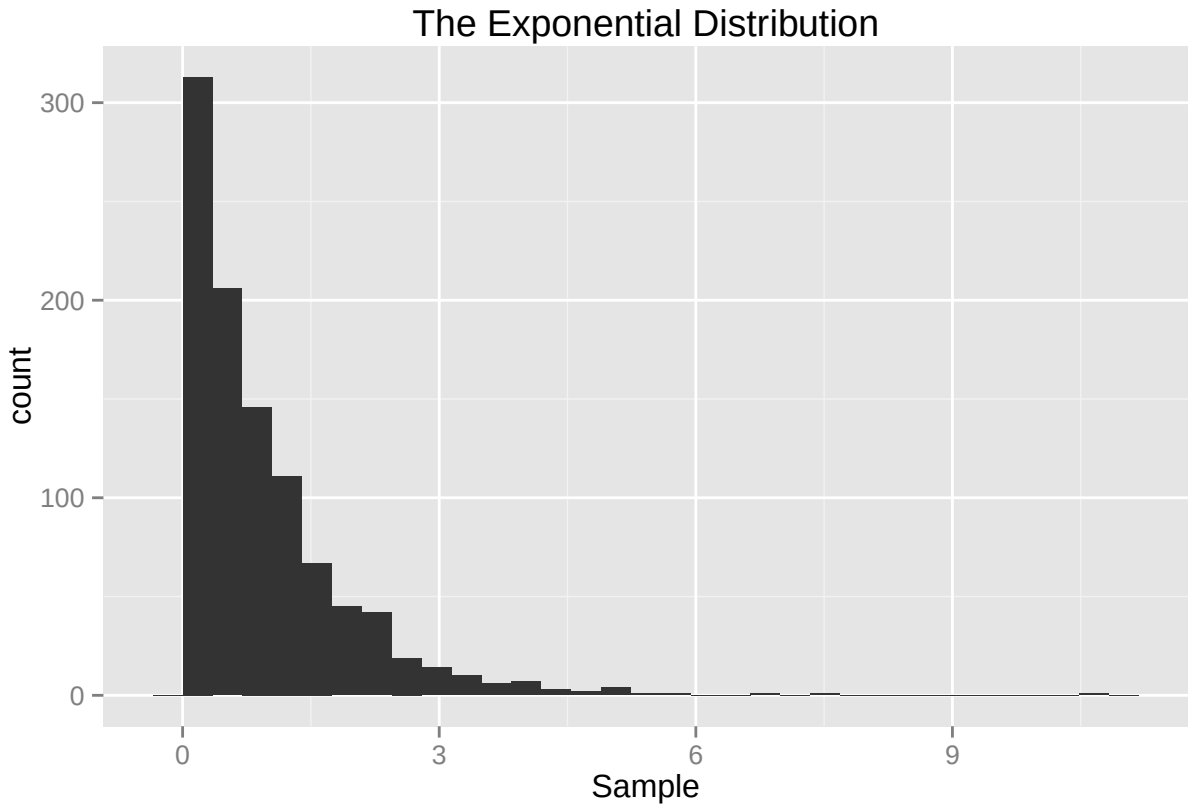
Part c

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
data <- data.frame(Sample=rexp(1000))

main <- "The Exponential Distribution"

ggplot(data, aes(x=Sample)) + geom_histogram() + ggtitle(main)
```

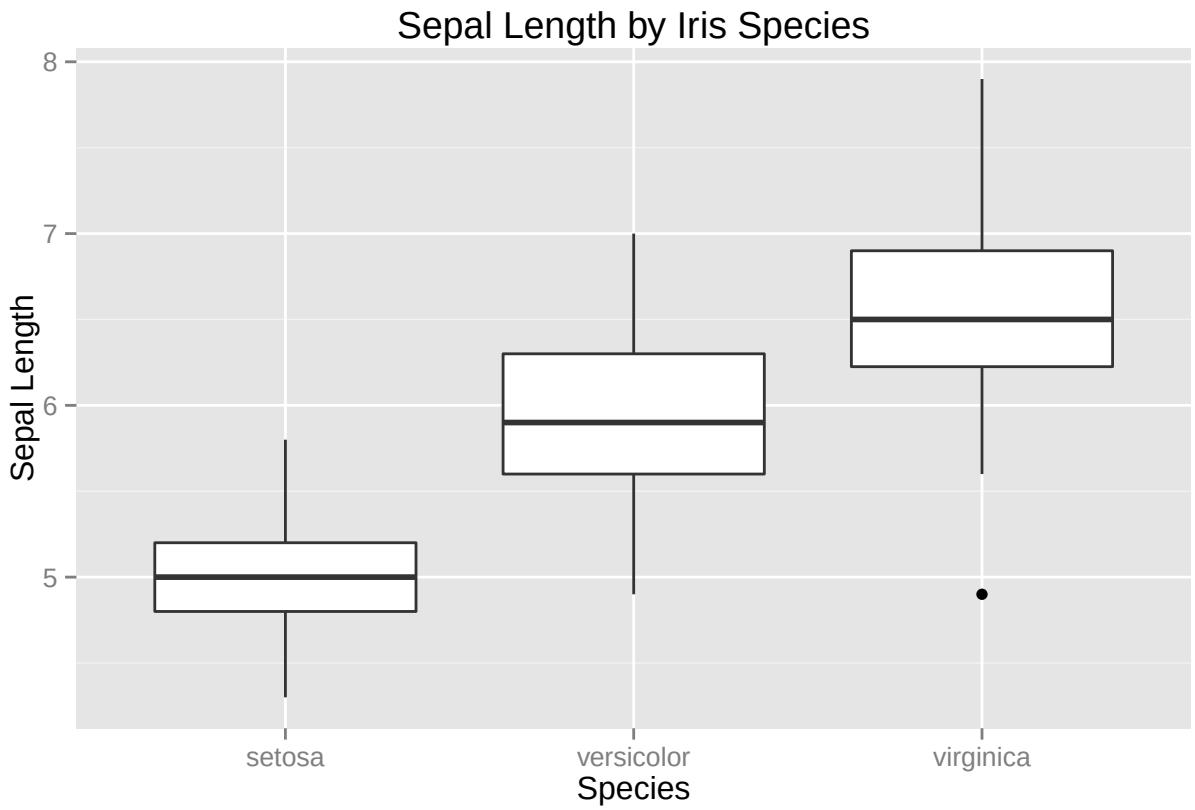
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Part d

Be sure that if you choose to use a dataset that's built into a package (such as `ggplot2`), you must be sure to load the package earlier in your .Rmd file. In this case `iris` is part of the base R and is always available.

```
main <- "Sepal Length by Iris Species"

ggplot(iris, aes(x=Species, y=Sepal.Length)) + geom_boxplot() +
  ggtitle(main) + ylab("Sepal Length")
```

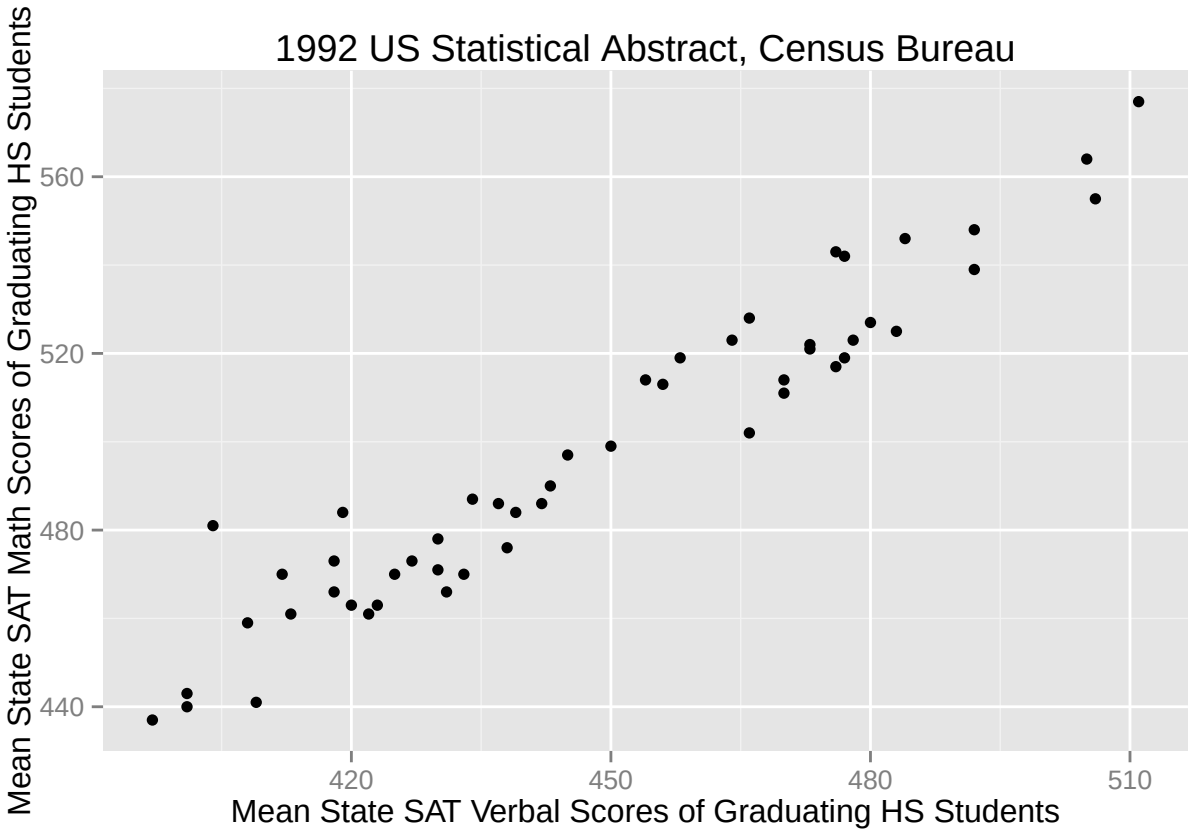


Part e

```
require(car)
```

```
## Loading required package: car
```

```
main <- "1992 US Statistical Abstract, Census Bureau"
xlab <- "Mean State SAT Verbal Scores of Graduating HS Students"
ylab <- "Mean State SAT Math Scores of Graduating HS Students"

ggplot(States, aes(x=SATV, y=SATM)) + geom_point() + ggtitle(main) +
  xlab(xlab) + ylab(ylab)
```


1992 US Statistical Abstract, Census Bureau
Mean State SAT Math Scores of Graduating HS Students
560
520
480
440
420
450
480
510
Mean State SAT Verbal Scores of Graduating HS Students