You will need the following file for this lab:

- Assignment 5.dat

The file is a text file and you can open it using a text editor such as Notepad to view its data. You want to read the data from the text file and create a data frame from the data on the text file. The problems are two-folds:

- First, the data are surrounded by lines of text
- Second, they are in different formats.

A good way to handle it is to open a file connection, and read in the data sequentially. For each of the three parts of data, we must specify what deviates from the standards for `read.table()`, which you can check with `?read.table`.

Question 1

(1/1 point)

First, let's set the working directory to the folder where you stored the data file.

Which function should you use to set your working directory?

- getwd()
- cd()
- dir()
- setwd() ✔

EXPLANATION

Suppose that you have saved the data file in your C:\Lab folder, you can set the working directory to that folder by using the following command:

```
setwd("C:/Lab")
```

Question 2

(1/1 point)

Proceed by opening a connection to the data file.

Which command should you use to perform the task?

- [x] f1<-file("Assignment 5.dat",open="r")
- [ ] f1<-scan("Assignment 5.dat")
- [ ] f1<-readLines("Assignment 5.dat")
- [ ] f1<-dget("Assignment 5.dat")

EXPLANATION

The following command will open a connection to the file you want to read from:

```
f1<-file("Assignment 5.dat",open="r")
```

Question 3

(1/1 point)

For the first part of the data, we must skip 4 lines, specify the comment character to "%", and read in 7 lines (the one with comments doesn't count). Read the first part of the data to a data frame.

Which command should you use to perform the task?

- my.data<-read.table(f1,skip=4,sep="%",nrows=7)
- my.data<-read.table(f1,skip=4,na.strings="%",nrows=7)
- my.data<-read.table(f1,skip=4,comment.char="%",nrows=7) ✔
- my.data<-read.table(f1,skip=4,what="%",nrows=7)

EXPLANATION

You can use the following command to perform the task:

```
my.data<-read.table(f1,skip=4,comment.char="%",nrows=7)
```

Question 4

(1/1 point)

For the second part of the data, we must skip 3 lines, specify the separator to ";", the decimal point to ",", and read in 2 lines. Read the second part of the data to another data frame.

Which command should you use to perform the task?

- (•) my.data2<-read.table(f1,skip=3,sep=";",dec=",",nrows=2) ✔
- ( ) my.data2<-read.table(f1,skip=3,na.strings=";",dec=",",nrows=2)
- ( ) my.data2<-read.table(f1,skip=3,comment.char=";",dec=",",nrows=2)
- ( ) my.data2<-read.table(f1,skip=3,what=";",dec=",",nrows=2)

EXPLANATION

You can use the following command to perform the task:

```
my.data2<-read.table(f1,skip=3,sep=";",dec=",",nrows=2)
```

Question 5

(1/1 point)

For the third part of the data, we must skip 5 lines, and specify the separator to ",". -9999 is a common way to specify missing values. It could also be a real observation, but we will treat it as a missing value and therefore specify na.strings to be "-9999". Read the third part of the data to another different data frame.

Which command should you use to perform the task?

- my.data3<-read.table(f1,skip=5,dec="-9999",sep=",",nrows=2)
- my.data3<-read.table(f1,skip=5,na.strings="-9999",sep=",",nrows=2) ✔
- my.data3<-read.table(f1,skip=5,comment.char="-9999",sep=",",nrows=2)

- my.data3<-read.table(f1,skip=5,what="-9999",sep=",",nrows=2)

EXPLANATION

You can use the following command to perform the task:

```
my.data3<-read.table(f1,skip=5,na.strings="-9999",sep=",",nrows=2)
```

Question 6

(1/1 point)

Now, combine the three data frames you created to a single data frame. You can bind rows of different data frames together with the `rbind()` function.

Which command should you use to perform the task?

- my.all.data<-rbind(my.data..my.data3)

- my.all.data<-rbind(my.data[1:3])
- my.all.data<-rbind(my.data+my.data2+my.data3)
- my.all.data<-rbind(my.data,my.data2,my.data3) ✔

EXPLANATION

You can use the following command to perform the task:

```
my.all.data<-rbind(my.data,my.data2,my.data3)
```