Class Exercise 1

We will work with two separate datasets, `LakeHuron` and `Loblolly`. The dataset `LakeHuron` measures the water level of Lake Huron, one of the 5 Great Lakes, from 1875 to 1972. `Loblolly` measures the growth and age of different loblolly seed varieties. “Loblollies” are loblolly pines, a fast-growing species important to the commercial timber industry.

Let’s look at `LakeHuron` first:

```
data(LakeHuron)
LakeHuron
```

Does `LakeHuron` look like any of the data types we have already seen? Why or why not? `LakeHuron` is actually a *time series* data set with data type `ts`; try the following query:

```
is.ts(LakeHuron)
```

We can confirm this by looking at its class attribute:

```
attributes(LakeHuron)
```

Note that the attribute `tsp` tells us when the series starts, when it ends, and its increment (E.g., monthly data to be plotted on a yearly scale would have an increment of 12); the `class` attribute identifies the data type. Both of these attributes are used by the `plot` command to create a time series plot with the proper time scale. Note that the command itself is quite simple, but uses built-in rules for plotting a `ts` object:

```
plot(LakeHuron)
```

Notice anything unusual in the time series? Lake Huron’s outlet, the St. Clair River, has been extensively dredged over the years, creating a long-term decrease in lake level that apparently leveled off decades ago. Concerns over a continuing decrease in the levels of Lakes Michigan, Superior and Huron continue to this day.

Let’s try plotting every 5 years’ data. Notice how easily multiple commands can be nested in **R**.

```
plot(LakeHuron[seq(1,100,by=5)])
```

This plot appears quite different from our earlier time series plot–what has been changed? The following command should save the 5-year subset as a time series object that can be more properly plotted. Are any issues still unaddressed? How would you resolve them?

```
plot(ts(LakeHuron[seq(1,100,by=5)],start=1875,frequency=0.2))
```

Next we will work with the `Loblolly` data set.

```
data(Loblolly)
Loblolly
```

What kind of data set is this? Is it a matrix or a data frame?

```
is.matrix(Loblolly)
is.data.frame(Loblolly)
```

Since it is not a matrix, you might anticipate that commands commonly used with matrices would not work. Try these:

```
dim(Loblolly)
Loblolly[1:5,]
```

Did they work? Clearly, some matrix commands can be applied to data frames.

Next we confirm that `Loblolly$Seed` is a factor; here we first type the variable name by itself; does the way in which **R** prints the variable provide clues to the data type?

```
Loblolly$Seed
is.factor(Loblolly$Seed)
```

Type

```
names(Loblolly)
```

These names are not particularly descriptive; we can change them (not in the datasets library, but in our local workspace) if we'd like, then construct a scatterplot for two of the variables. Are the resulting names more satisfactory? What might be a disadvantage?

```
names(Loblolly)=c("Height (Ft)","Age (Yrs)","Seed Variety")
names(Loblolly)
Loblolly
```