```
tgpp<-read.csv("tgpp.csv")
```

1. What are the names of the columns in this dataset?
2. How many rows and columns does this data file have?

```
str(tgpp)
```

```
## 'data.frame':    4080 obs. of  11 variables:
##  $ plot     : int  205 205 205 205 205 205 205 205 205 205 ...
##  $ year     : int  1998 1998 1998 1998 1998 1998 1998 1998 1998 1998 ...
##  $ record_id: int  187 188 189 190 191 192 193 194 195 196 ...
##  $ corner   : int  NA 1 2 3 4 1 2 3 4 1 ...
##  $ scale    : num  100 10 10 10 10 1 1 1 1 0.1 ...
##  $ richness : int  60 36 34 37 33 21 23 19 25 10 ...
##  $ easting  : int  727000 727000 727000 727000 727000 727000 727000 727000 727000 727000 ...
##  $ northing : int  4080000 4080000 4080000 4080000 4080000 4080000 4080000 4080000 4080000 40
80000 ...
##  $ slope    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ ph       : num  6.9 6.9 6.9 6.9 6.9 6.9 6.9 6.9 6.9 6.9 ...
##  $ yrsslb   : num  0.39 0.39 0.39 0.39 0.39 0.39 0.39 0.39 0.39 0.39 ...
```

```
#1 plot, year, record_id, corner, scale, richness, easting, northing, slope, ph, yrsslb
#2 11 columns, 4080 rows
```

3. What kind of object is each data column? Hint: checkout the function sapply().

```
sapply(tgpp, class)
```

```
##      plot      year record_id    corner     scale  richness   easting
## "integer" "integer" "integer" "integer" "numeric" "integer" "integer"
##  northing     slope        ph    yrsslb
## "integer" "integer" "numeric" "numeric"
```

4. What are the values of the the datafile for rows 1, 5, and 8 at columns 3, 7, and 10

```
tgpp[c(1,5,8), c(3,7,10)]
```

```
##   record_id easting  ph
## 1       187  727000 6.9
## 5       191  727000 6.9
## 8       194  727000 6.9
```

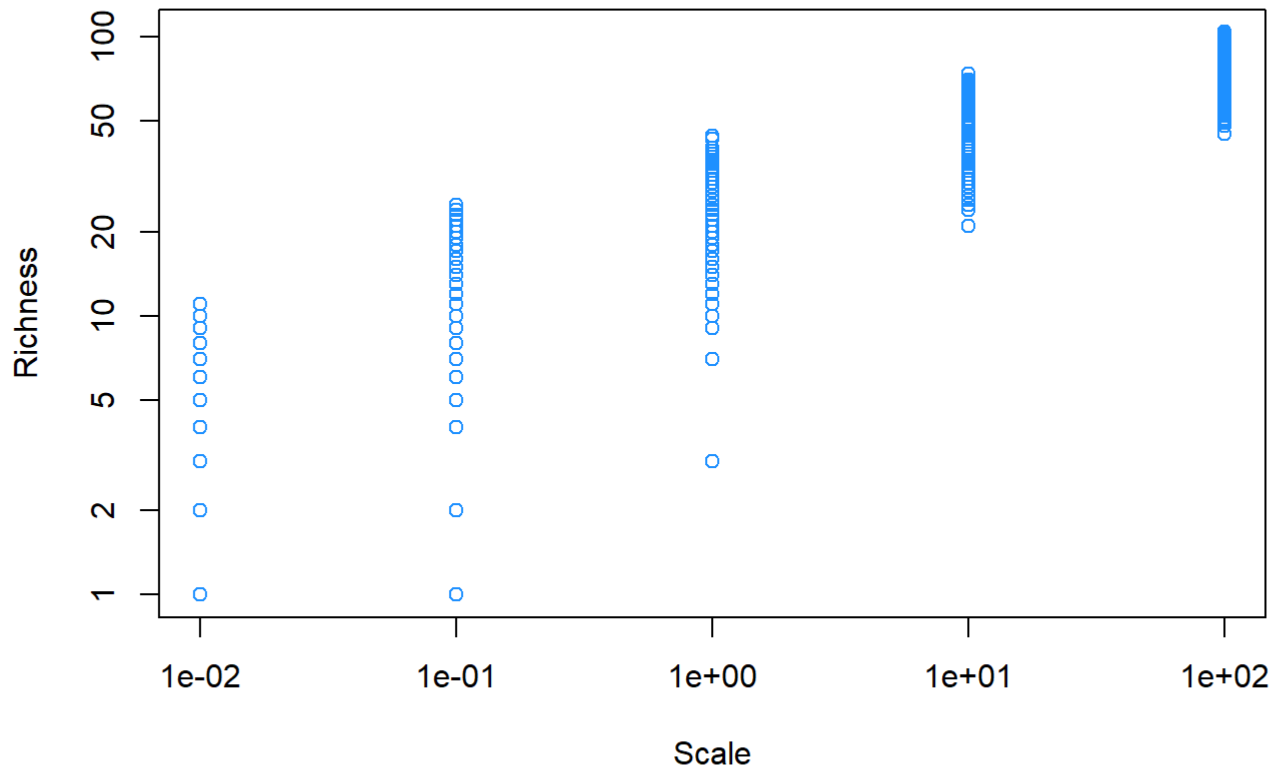
5. Create a pdf of the relationship between the variables “scale” and “richness”. Scale is the area in square meters of the quadrat in which richness was recorded. Be sure to label your axes clearly, and choose a color you find pleasing for the points. To get a list of available stock colors use the function colors().

```
pdf("./scale_richness.pdf")
plot(tgpp$scale, tgpp$richness, xlab = "Scale", ylab="Richness", col="dodgerblue")
```

6. What happens to your plot when you set the plot argument log equal to ‘xy’. plot(…, log=‘xy’)

```
plot(tgpp$scale, tgpp$richness, log = "xy", xlab = "Scale", ylab="Richness", col="dodgerblue")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 4 y values <= 0 omitted
## from logarithmic plot
```



Graph is more spread out and looks nicer. It was log transformed