Lab06_BasicGraphicalParameters

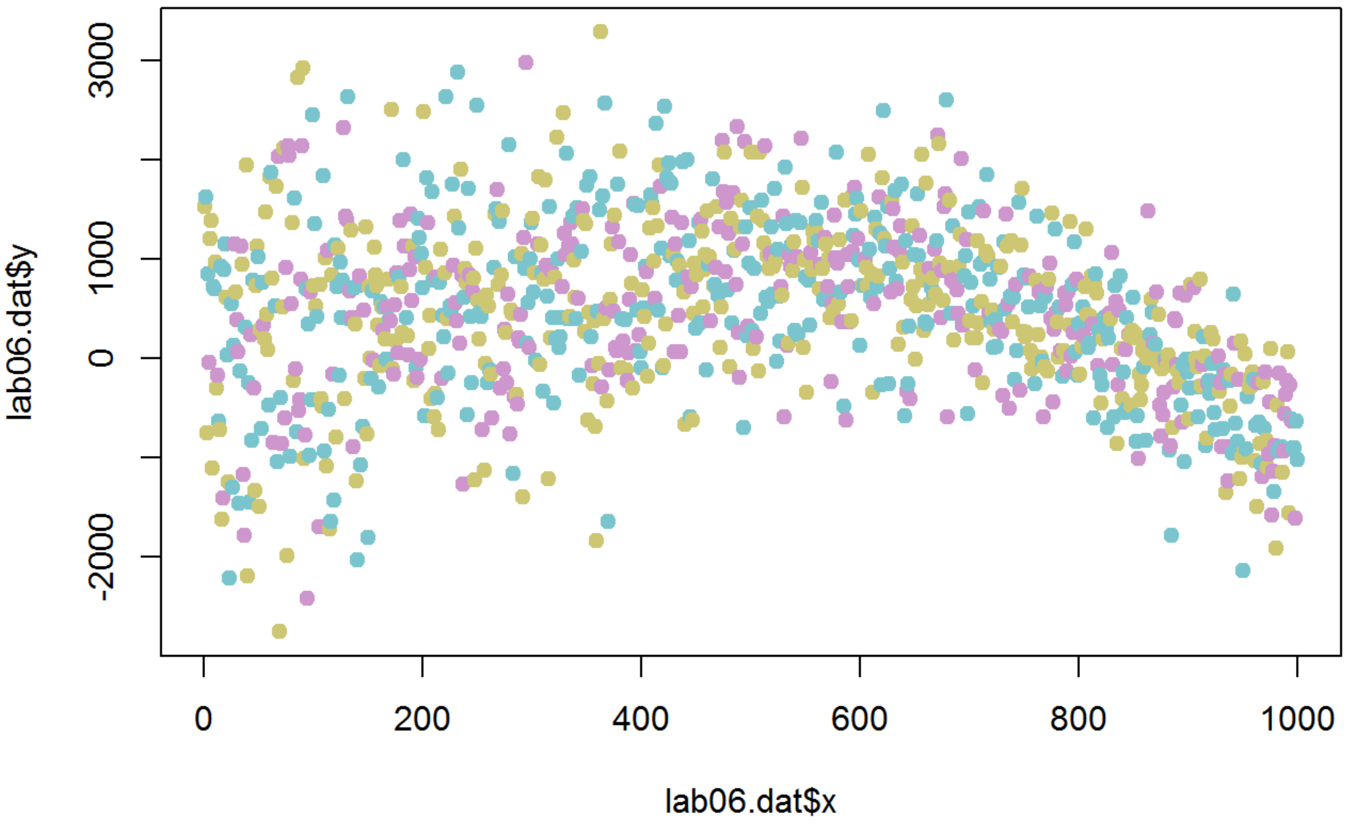
Skyler Halbritter

October 15, 2015

```
lab06.dat <- read.csv( "Lab 06 Data.csv", stringsAsFactors=FALSE )
```

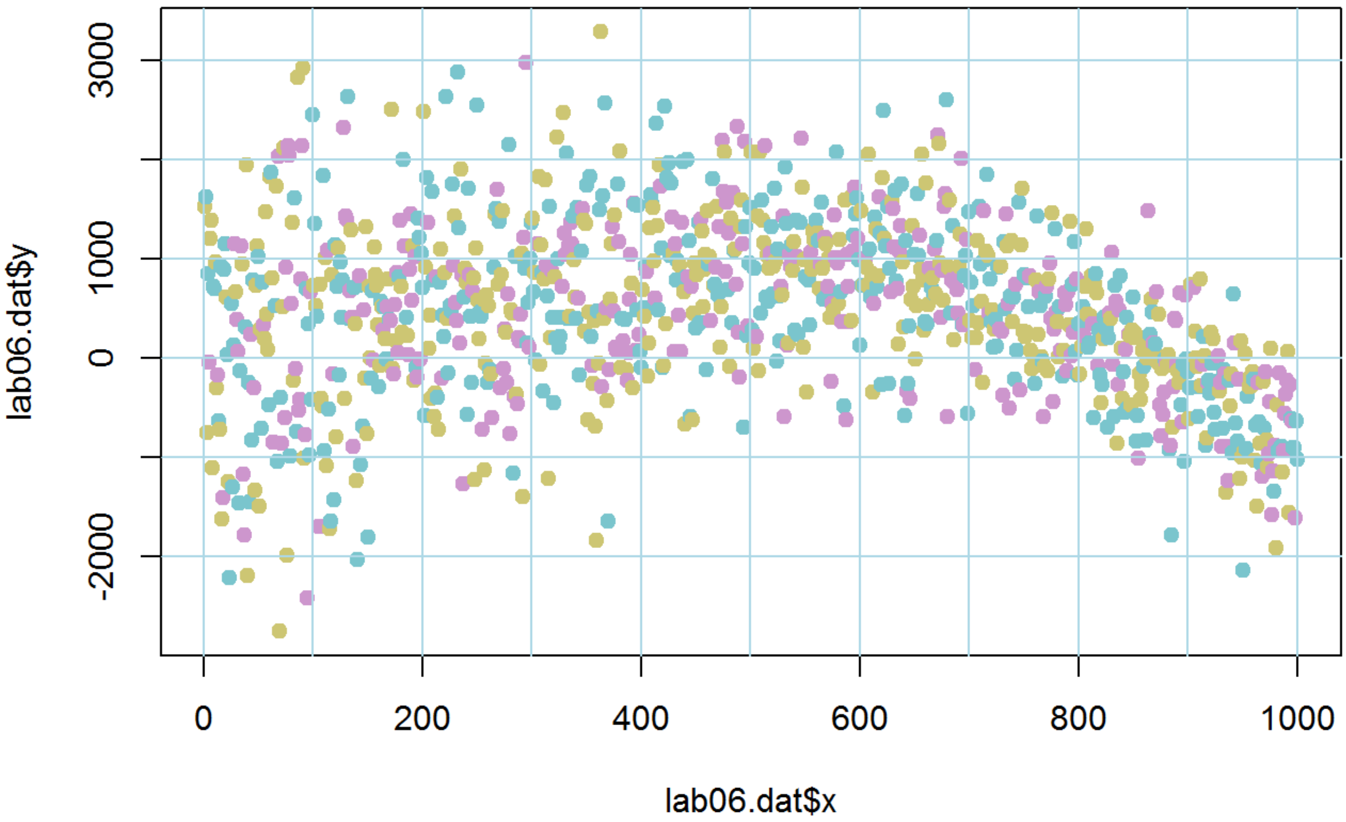
1. Plot your x and y variables as normal using col=groups and pch=19 to represent the group structure.

```
plot(lab06.dat$x, lab06.dat$y, pch=19, col=lab06.dat$groups)
```



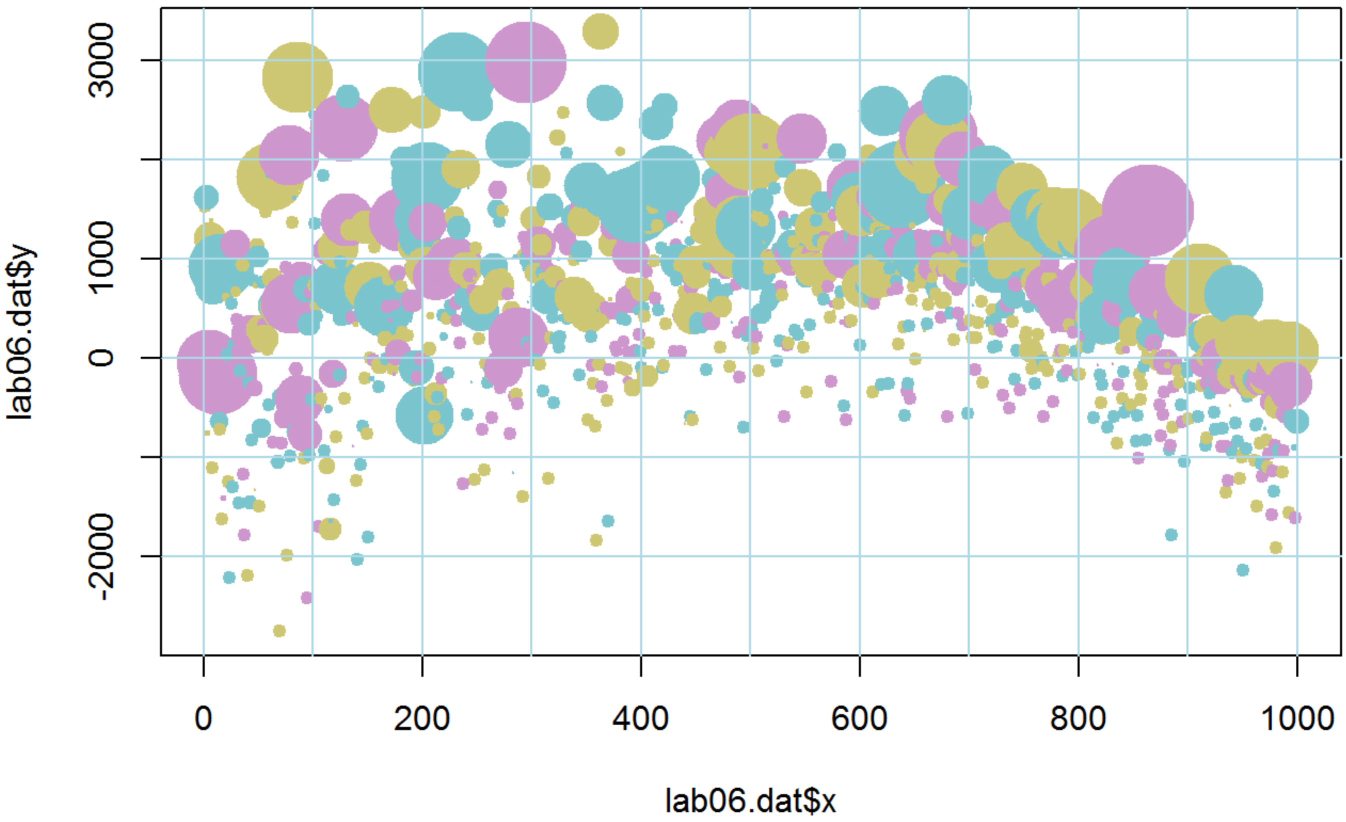
2. Add a coordinate system using the abline() function. Create horizontal lines spaces 1000 apart, and vertical lines spaced 100 apart. I used the color “lightblue” so that it shows as a ground element, not figure.

```
plot(lab06.dat$x, lab06.dat$y, pch=19, col=lab06.dat$groups)
abline(h=seq(from=-2000, to=3000, by=1000), col="lightblue" )
abline(v=seq(from=0, to=1000, by=100), col="lightblue")
```



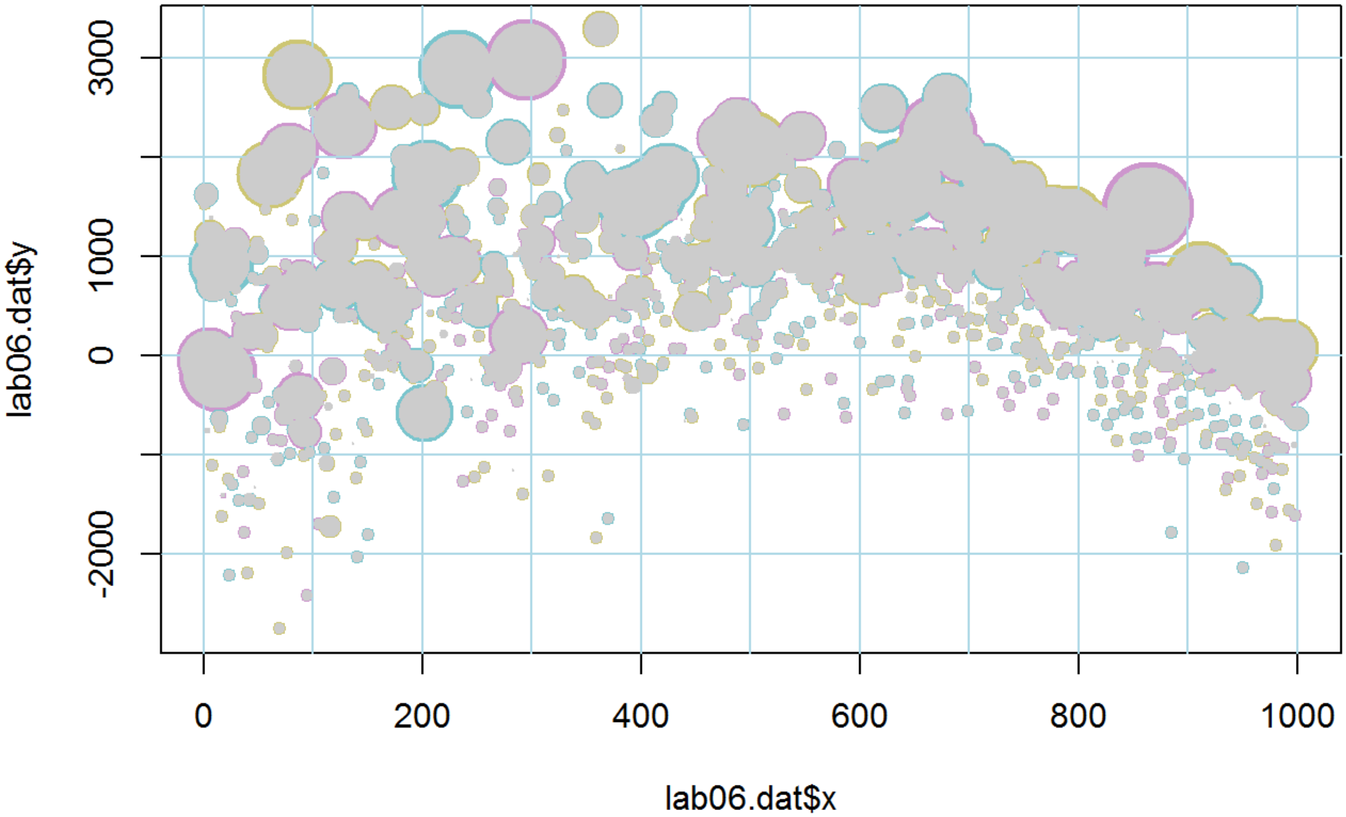
3. Scale the point size relative to the z variable using the cex argument. You can vary point sizes, but still increase and decrease points sizes by scaling z: cex=2*z, for example.

```
plot(lab06.dat$x, lab06.dat$y, pch=19, col=lab06.dat$groups, cex=2*(lab06.dat$z))
abline(h=seq(from=-2000, to=3000, by=1000), col="lightblue")
abline(v=seq(from=0, to=1000, by=100), col="lightblue")
```



4. Tone down the colors by adding a gray center to the plot points using the points() command. Use “gray80” as your color. You will do this by using a slightly smaller cex.

```
plot(lab06.dat$x, lab06.dat$y, pch=19, col=lab06.dat$groups, cex=2*(lab06.dat$z))
abline(h=seq(from=-2000, to=3000, by=1000), col="lightblue")
abline(v=seq(from=0, to=1000, by=100), col="lightblue")
points(lab06.dat$x, lab06.dat$y, pch=19, col="gray80", cex=1.8*(lab06.dat$z))
```

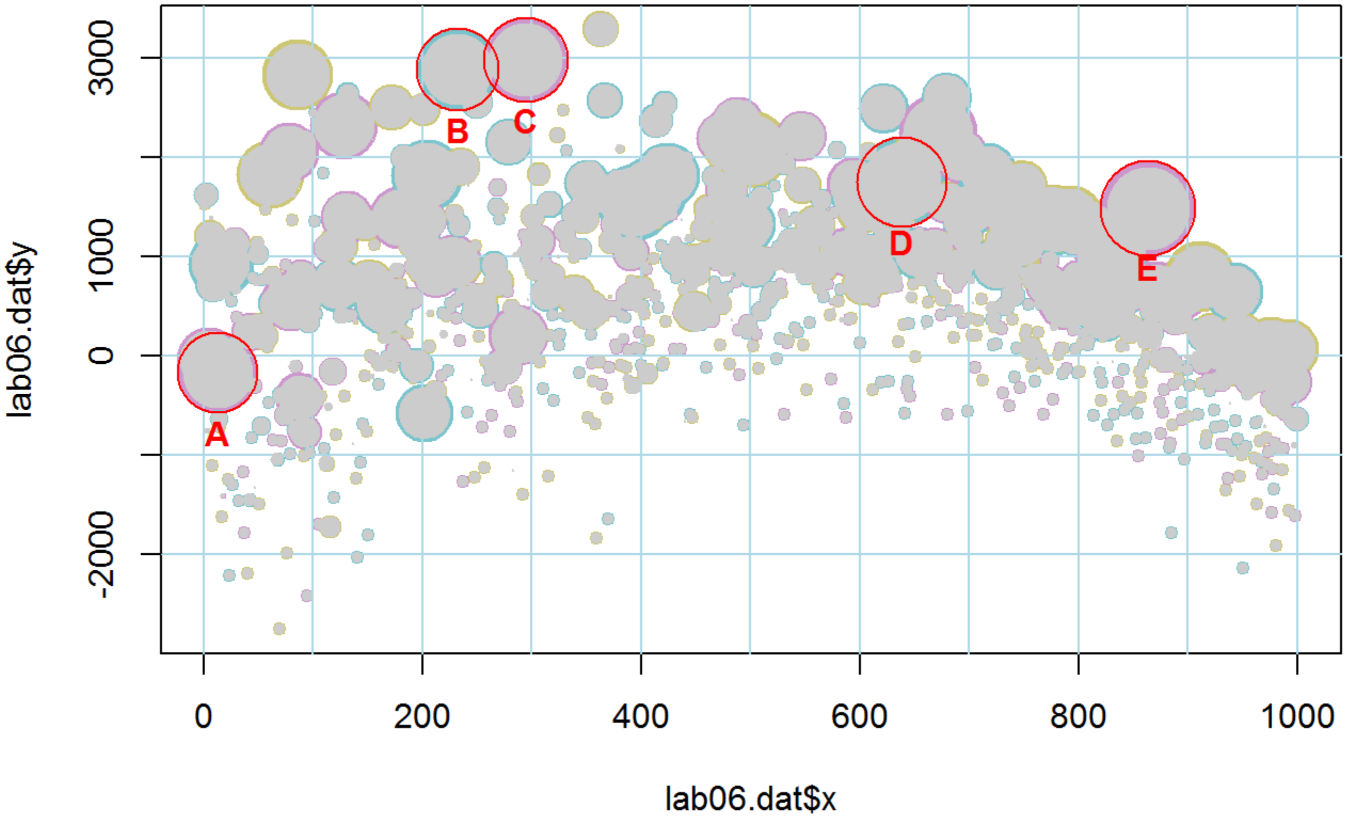


5. Label the five points with the largest z values as A,B,C,D and E. Use a red ring and red text. You can identify these by: these <- which(rank(z) > 995)

```
these.z <- which( rank(lab06.dat$z) > 995)
these.z
```

```
## [1]  13 232 295 638 864
```

```
these.z.vec <- lab06.dat[c(13,232,295,638,864) , ]
plot(lab06.dat$x, lab06.dat$y, pch=19, col=lab06.dat$groups, cex=2*(lab06.dat$z))
abline(h=seq(from=-2000, to=3000, by=1000), col="lightblue")
abline(v=seq(from=0, to=1000, by=100), col="lightblue")
points(lab06.dat$x, lab06.dat$y, pch=19, col="gray80", cex=1.8*(lab06.dat$z))
points( x=these.z.vec$x, y=these.z.vec$y, pch=1, cex=2.1*(these.z.vec$z), col="red" )
text( x=these.z.vec$x, y=these.z.vec$y, labels=c("A", "B", "C", "D", "E"), pos=1, col="red", offset=1.2, font=2 )
```



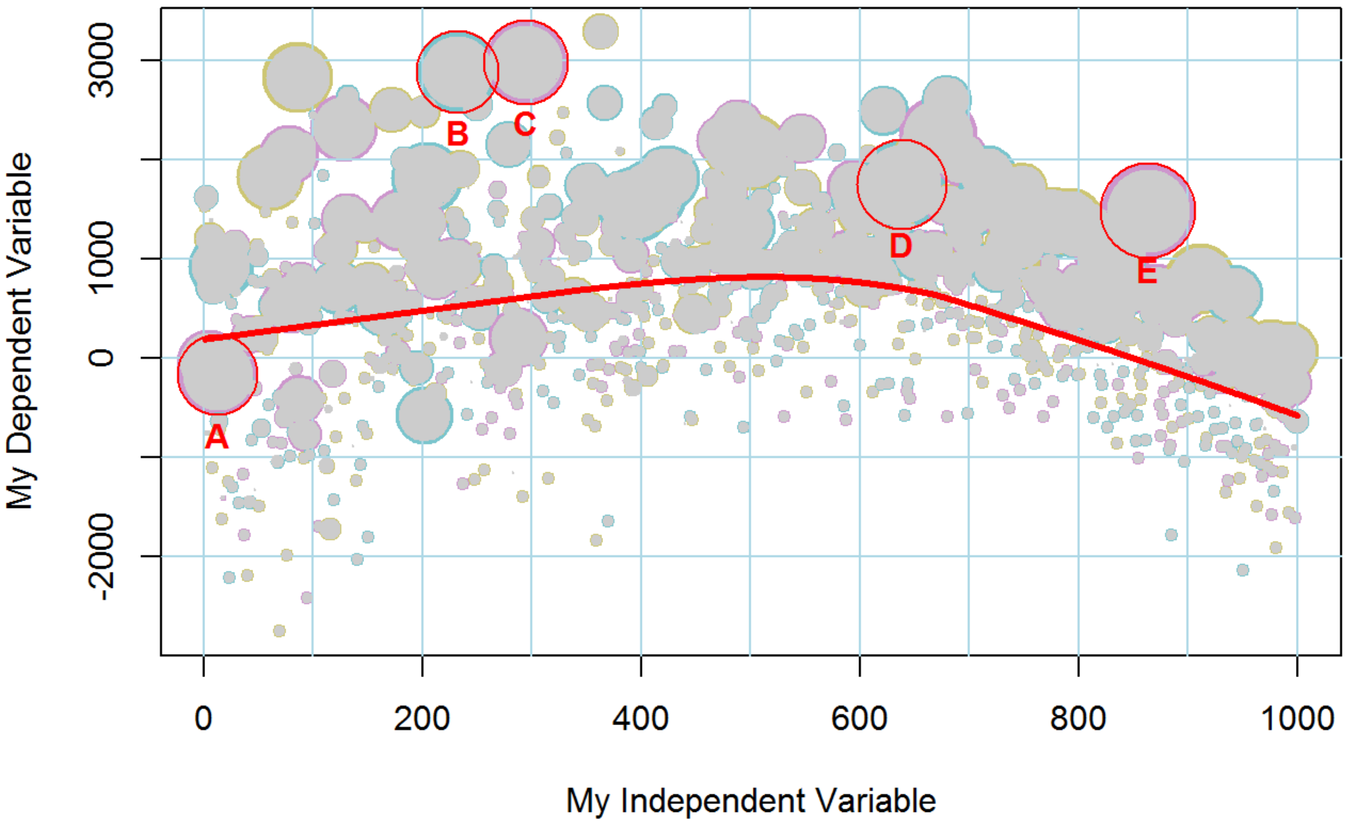
6. Make sure your x and y axes are properly labeled, and use your name as the title of the graph.

```
these.z <- which( rank(lab06.dat$z) > 995)
these.z
```

```
## [1]  13 232 295 638 864
```

```
these.z.vec <- lab06.dat[c(13,232,295,638,864) , ]
plot(
  lab06.dat$x, lab06.dat$y,
  xlab="My Independent Variable",
  ylab="My Dependent Variable",
  main= "Skyler Halbritter: Lab06 Basic Graphical Parameters",
  pch=19,
  col=lab06.dat$groups,
  cex=2*(lab06.dat$z))
abline(h=seq(from=-2000, to=3000, by=1000), col="lightblue")
abline(v=seq(from=0, to=1000, by=100), col="lightblue")
points(lab06.dat$x, lab06.dat$y, pch=19, col="gray80", cex=1.8*(lab06.dat$z))
points( x=these.z.vec$x, y=these.z.vec$y, pch=1, cex=2.1*(these.z.vec$z), col="red" )
text( x=these.z.vec$x, y=these.z.vec$y, labels=c("A", "B", "C", "D", "E"), pos=1, col="red", offset=1.2, font=2 )
lines( lowess( lab06.dat$x, lab06.dat$y ), col="red", lwd=3 )
```

Skyler Halbritter: Lab06 Basic Graphical Parameters


My Dependent Variable
My Independent Variable
A
B
C
D
E