

## 5.1

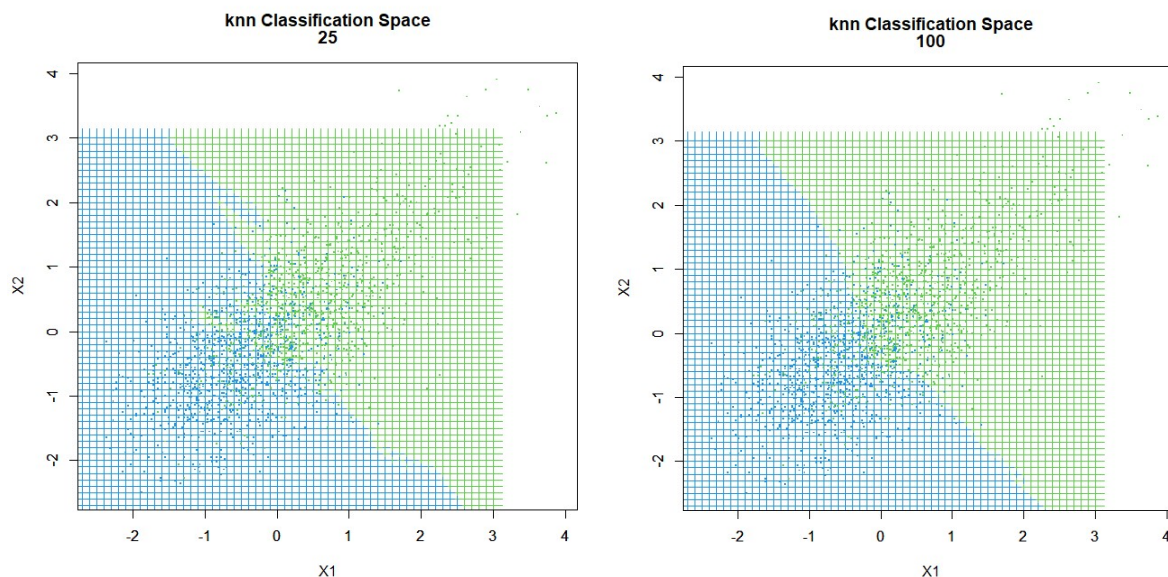
When using the daily range of my stock (DVN) to produce the same outcome in Section 3 LN5, I produced different results. I recreated all the variables from Section 5 of LN4, especially the factor data type in *Y.dvn* (`> Y.dvn = as.factor(Y.dvn)`). I made a natural log transformation of the range values and standardized the lag1 and lag2 variables.

```
> LnX.dvn = log(X.dvn)
> StLnX.dvn = apply(LnX.dvn,2,scale)
> TrainStLnX.dvn = StLnX.dvn[InSample,]
> TestStLnX.dvn = StLnX.dvn[OutSample,]
> NewProbeKnnBig(TrainStLnX.dvn,StProbeX,TrainY.dvn,25)
NULL
```

What was produced was a different color as shown in Section 3. I tried to go back and retrace my steps but I believe the extra lags in my *dvnrange* file created a different result.

```
TestY.dvn
knn.pred  1.390002 HighRisk LowRisk
1.390002      0      0      0
HighRisk      0     656     226
LowRisk       0     282     691
```

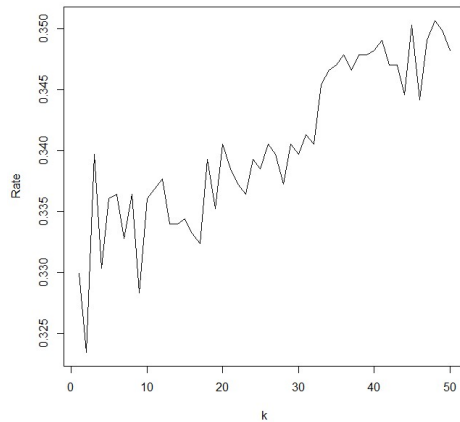
I restarted it and tried to correct the issue but It continued to produce the same result. After resuming the experiment, I noticed that higher values of *k* produced a smoother classification space.



## 5.2

When applying the *ksearch* program for my DVN range data, it produced the following output:

```
junk = kSearch(X.dvn,Y.dvn,1200,50)
plot(junk,type="l",ylab="Rate",xlab="k")
```

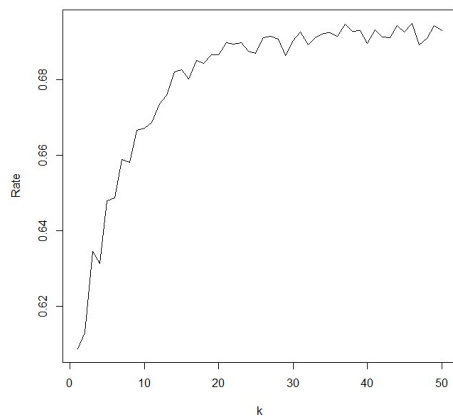


I believe that splitting the selected data causes erratic fluctuations in the plot as it splits k one at a time.

## 5.3

When applying the *kcvsearch* program for my DVN range data, it produced the following output:

```
junk2 = kcvSearch(X.IBM,Y.IBM,1200,50,30)
plot(junk2,type="l",ylab="Rate",xlab="k")
```



With cross validation, k makes random splits and then averages them out to produce smoother results in the plot.

5.4(a)

When searching Google to discover ways to create grids in Excel for R points, the following results are generated:

The screenshot shows a Google search interface with the query "how to create grids of points r using excel". The results are as follows:

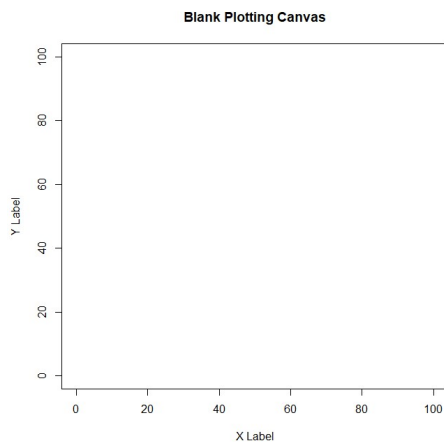
- How to create a grid whose vertex coordinates are shown on ...**  
Apr 15, 2017 - Then, use the GIS to select the grid points that lie within your study area. ... though, with the thesis that creating a grid is not a GIS operation, here is an R based ...  
Convert CSV to Raster Grid in R - Geographic Information ... Jan 10, 2018  
How to plot gridded map from lat-lon and fill values in R .... Aug 29, 2017  
More results from gis.stackexchange.com  
4 answers
- www.dataanalytics.org.uk > gridlines-in-graphs-and-ch...**  
**Gridlines in graphs and charts. Statistics for Ecologists ...**  
Jul 29, 2019 - How to add and format gridlines in Excel charts and R graphs. ... Give the start and end points, this will produce a primitive sequence with an ...  
People also search for:  
grid() in r abline in r  
plot grid
- stackoverflow.com > questions > create-points-from-a-g...**  
**Create points from a grid using R - Stack Overflow**  
Aug 9, 2016 - First follow ?readOGR to create the scot\_BNG object which I'll be using here. Then create gridded object: > scotgrid = spsample(scot\_BNG, n=1000, ...  
**How to create a grid of spatial points - Stack Overflow** 1 answer Jan 22, 2017  
**Methods adding Excel-like features to R plots ...** 4 answers Nov 13, 2011  
**Create a grid from regular sampling points in R ...** 2 answers Nov 3, 2016  
**Excel Function to fill grid with specific data that is ...** 1 answer May 6, 2019  
More results from stackoverflow.com  
1 answer  
Missing: excel | Must include: excel
- nceas.github.io > oss-lessons > 4-tues-spatial-analysis-in-r**  
**Intro to spatial data in R - Open and plot raster and vector data ...**  
We populate that grid with interpolated values generated from the points layer that we created above. In the example below, we use the Inverse Distance Weighted ...  
Missing: excel | Must include: excel
- cran.r-project.org > web > packages > OpenRepGrid** PDF  
**Package 'OpenRepGrid' - CRAN**

I found an interesting article that appeared in the search to allow the user to create test-grids inside R.

<https://bookdown.org/ndphillips/YaRrr/low-level-plotting-functions.html>

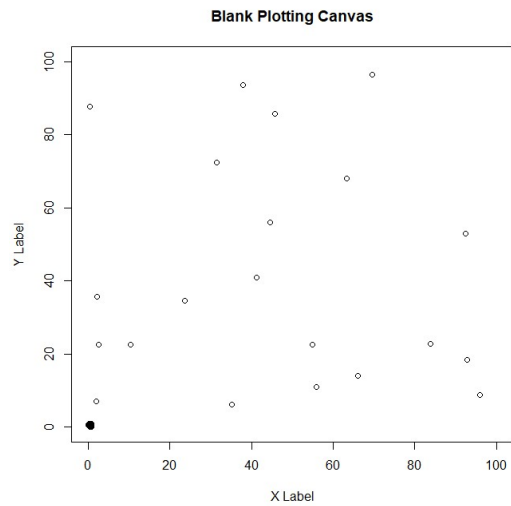
According to the article, you must create a blank plot first using the `plot()` function

```
> plot(x = 1,  
+       xlab = "X Label",  
+       ylab = "Y Label",  
+       xlim = c(0, 100),  
+       ylim = c(0, 100),  
+       main = "Blank Plotting Canvas",  
+       type = "n")
```



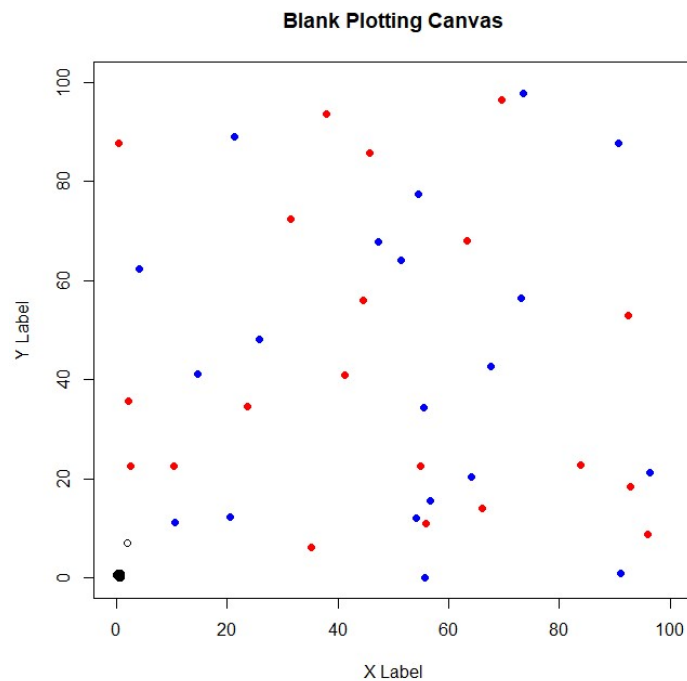
Next, to add points to your plot you must use the `points()` function. I had difficulty with this one where the website had their own dataset. After searching many websites to try to create random points within the plot, I found something that worked. Most R functions I saw didn't work because I didn't have that package. But I found out how to use the `runif()` function to produce random numbers. I created an x and y variable that had `runif()` statements in them, then plotted them with the `points` function

```
> x = runif(20, min = 0, max = 99)  
> y = runif(20, min = 0, max = 99)  
> points(x,y)
```



After that I wanted to create 2 groups of randomly generated points

```
> points(x,y)
> x = runif(20, min = 0, max = 99)
> y = runif(20, min = 0, max = 99)
> points(x,y)
> points(x,y,pch = 16,col="red")
> a = runif(20, min = 0, max = 99)
> b = runif(20, min = 0, max = 99)
> points(a,b,pch = 16,col="blue")
```



This is a very informative website that teaches you how to make variety of plots using R. I will come back to this to create more in the future.

## Appendix

```
ls()
library(class)
dvn = read.csv("dvnrange.csv")
head(dvn,10)
tail(dvn,10)
sample(5,5)
Shuffle = sample(3655,3655)
dvnrange[Shuffle[1:6],]
dvnrange = [Shuffle[1:6],]
dvn[Shuffle[1:6],]
InSample = Shuffle[1:1800]
OutSample = Shuffle[1801:3655]
X.dvn = dvn[,3:5]
Y.dvn = dvn[,2]
median(Y.dvn)
Y.dvn[Y.dvn>1.390002]="HighRisk"
Y.dvn[Y.dvn<1.390002]="LowRisk"
Y.dvn[1:6]
as.factor(Y.dvn[1:6])
Y.dvn = as.factor(Y.dvn)
TrainX.dvn = X.dvn[InSample,]
TrainY.dvn = Y.dvn[InSample,]
TrainY.dvn = Y.dvn[InSample]
TestX.dvn = X.dvn[OutSample,]
TestY.dvn = Y.dvn[OutSample]
knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
table(knn.pred,TestY.dvn)
X.dvn = dvn[,3:4]
table(knn.pred,TestY.dvn)
TrainX.dvn = X.dvn[InSample,]
TestX.dvn = X.dvn[OutSample,]
table(knn.pred,TestY.dvn)
LnX.dvn = log(X.dvn)
StLnX.dvn = apply(LnX.dvn,2,scale)
TrainStLnX.dvn = StLnX.dvn[InSample,]
TestStLnX.dvn = StLnX.dvn[OutSample,]
NewProbeKnnBig(TrainStLnX.dvn,StProbeX,TrainY.dvn,25)
q()
ls()
load("C:\\Users\\whall\\Google Drive\\1 CUNY WORK\\01
NewProbeKnnBig(TrainStLnX.dvn,StProbeX,TrainY.dvn,25)
history(max.show=200)
```

```
junk = kSearch(X.IBM,Y.IBM,1200,50)
plot(junk,type="l",ylab="Rate",xlab="k")
junk = kSearch(X.dvn,Y.dvn,1200,50)
plot(junk,type="l",ylab="Rate",xlab="k")
junk = kSearch(X.dvn,Y.dvn,1800,50)
plot(junk,type="l",ylab="Rate",xlab="k")
junk = kSearch(X.dvn,Y.dvn,1200,50)
plot(junk,type="l",ylab="Rate",xlab="k")
kSearch =
function (X,Y,split=100, d=25)
{
  if(class(Y) != "factor") stop('Y is not factor type')
  if(class(X) != "data.frame") stop('X is not data.frame type')
  rows = nrow(X)
  Hold = rep(NA,d)
  Shuffle=sample(rows,rows)
  for(k in 1:d) {
    InSample=Shuffle[1:split]
    OutSample=Shuffle[(split+1):rows]
    TrainX = X[InSample,]
    TrainY = Y[InSample]
    TestX = X[OutSample,]
    TestY = Y[OutSample]
    knn.pred = knn(TrainX, TestX, TrainY, k)
    table.out = table(knn.pred,TestY)
    Hold[k] = (table.out[1,1]+table.out[2,2])/sum(table.out)
  }
  return(Hold)
}
plot(junk,type="l",ylab="Rate",xlab="k")
junk = kSearch(X.dvn,Y.dvn,1800,50)
plot(junk,type="l",ylab="Rate",xlab="k")
junk2 = kcvSearch(X.IBM,Y.IBM,1200,50,30)
plot(junk,type="l",ylab="Rate",xlab="k")
junk2 = kcvSearch(X.IBM,Y.IBM,1200,50,30)
plot(junk2,type="l",ylab="Rate",xlab="k")
```