4.1 This is the history of the steps of testing the cat and dog data set.

```
ls()
history()
library(class)
knn
Dogs[2,4]
Dogs[1:5,3]
Dogs[1:15,2:3]
TrainX
TestX
TrainY
knn.pred = knn(TrainX,TestX,TrainY,3)
knn.pred
cbind(TestY,knn.pred)
table(knn.pred,TestY)
table.out = table(knn.pred,TestY)
table.out
(table.out[1,1]+table.out[2,2])/sum(table.out)
save.image("C:\\Users\\whall\\Google Drive\\1
history()
```

4.2 In my DVN CSV file I tried to create lags. While doing this, there were several empty spaces from the dates that would create a problem in R.

	A	B	C	D	E	F	G	H	I	J	K
1	Date	Open	High	Low	Close	Adj Close	Volume	DVNret	Lag1	Lag2	Lag3
2	1/3/2006	62.76	64.99	62.61	64.69	53.90661	4101200				
3	1/4/2006	64.69	66.26	64	65.99	54.98991	5681000	2.00959			
4	1/5/2006	65.99	66	64.31	65.17	54.30659	5543500	-1.24264	2.00959		
5	1/6/2006	66.25	67.16	65.89	66.59	55.48989	3472000	2.178933	-1.24264	2.00959	
6	1/9/2006	67.1	67.1	65.86	66.45	55.37323	3030600	-0.21024	2.178933	-1.24264	2.00959
7	1/10/2006	66.45	67.6	66.21	66.55	55.45657	3526400	0.150499	-0.21024	2.178933	-1.24264
8	1/11/2006	66.1	66.2	64.54	65.38	54.48159	3972700	-1.7581	0.150499	-0.21024	2.178933
9	1/12/2006	65.5	67.09	65.17	65.49	54.57324	4503600	0.168226	-1.7581	0.150499	-0.21024

In my new csv file, DVN2008.csv I started my data on Jan. 2th, 2008 to July 14th, 2020. By doing this, it created cells without any null data. The columns contain DVNret, Lag1,Lag2,Year, and Direction (TRUE (Up) and False (Down)).

	A	B	C	D	E	F
1	Date	DVNret	Lag1	Lag2	Year	Direction
2	1/2/2008	3.115511	-1.65911	0.321764	2008	
3	1/3/2008	1.374354	3.115511	-1.65911	2008	FALSE
4	1/4/2008	-2.84051	1.374354	3.115511	2008	FALSE
5	1/7/2008	-0.55372	-2.84051	1.374354	2008	TRUE
6	1/8/2008	-2.83967	-0.55372	-2.84051	2008	FALSE
7	1/9/2008	2.338131	-2.83967	-0.55372	2008	TRUE
8	1/10/2008	-0.71678	2.338131	-2.83967	2008	FALSE
9	1/11/2008	-1.66946	-0.71678	2.338131	2008	FALSE

Since there is a null value on 1/2/2008, I had to delete the first row and focus on the data from 1/3/2008 on.

```
> dvn = dvn[-1,]
> head(dvn,10)
     ï..Date     DVNret       Lag1       Lag2 Year Direction
2   1/3/2008  1.3743536  3.1155111 -1.6591059 2008     FALSE
3   1/4/2008 -2.8405051  1.3743536  3.1155111 2008     FALSE
4   1/7/2008 -0.5537244 -2.8405051  1.3743536 2008      TRUE
5   1/8/2008 -2.8396650 -0.5537244 -2.8405051 2008     FALSE
6   1/9/2008  2.3381305 -2.8396650 -0.5537244 2008      TRUE
7  1/10/2008 -0.7167847  2.3381305 -2.8396650 2008     FALSE
8  1/11/2008 -1.6694640 -0.7167847  2.3381305 2008     FALSE
9  1/14/2008  3.4759318 -1.6694640 -0.7167847 2008      TRUE
10 1/15/2008 -3.2150363  3.4759318 -1.6694640 2008     FALSE
```

From this point, I will start Step 1 with *library(class)*. Step 2 has already been completed and will continue to create all variables needed.

```
> Lag1 <- dvn$Lag1
> Lag2 <- dvn$Lag2
> Year <- dvn$Year
> train = (Year<2020)
> dvn2020 = dvn[!train,]
> Direction <- dvn$Direction
> Direction2020 = Direction[!train]
> train.X=cbind(Lag1 ,Lag2)[train ,]
> test.X=cbind (Lag1 ,Lag2)[!train ,]
> train.Direction =Direction [train]
> knn.pred=knn(train.X,test.X,train.Direction,k=1)
```

After, I created a table with these results:

```
          Direction2020
knn.pred FALSE TRUE
   FALSE    45   16
   TRUE     30   43
```

It appears that the percent of correct forcasts is 66% by using the calculation below.

```
> p <- (45+43)/(45+16+30+43)
> p
[1] 0.6567164
```

Now, I will try to increase k to see if that will improve the forcast

```
> knn.pred=knn(train.X,test.X,train.Direction,k=3)
> table(knn.pred,Direction2020)
          Direction2020
knn.pred FALSE TRUE
   FALSE    47   20
   TRUE     28   39
> p <- (47+39)/(47+20+28+39)
> p
[1] 0.641791
```

In this data, by increasing k (k=3), the results were slightly less than when k=1.

4.3

In this section of the exercise, I created a new excel file named “dvnrange.csv” to calculate the historical range of my company, Devon Energy corp (DVN). I calculated the historical ranges and prepared 3 Lags to conduct analysis

	A	B	C	D	E
1	Date	DVNrange	Lag 1	Lag 2	Lag 3
2	1/6/2006	1.270005	1.690002	2.260002	2.379997
3	1/9/2006	1.239997	1.270005	1.690002	2.260002
4	1/10/2006	1.389999	1.239997	1.270005	1.690002
5	1/11/2006	1.659996	1.389999	1.239997	1.270005
6	1/12/2006	1.919998	1.659996	1.389999	1.239997
7	1/13/2006	1.130005	1.919998	1.659996	1.389999
8	1/17/2006	1.160004	1.130005	1.919998	1.659996

.

After setting up *>library(class)* , I imported the csv file and checked its head and tail.

```
> library(class)
> dvnr = read.csv("dvnrange.csv")
> head(dvnr,10)
     i..Date DVNrange    Lag.1    Lag.2    Lag.3
1   1/6/2006 1.270005 1.690002 2.260002 2.379997
2   1/9/2006 1.239997 1.270005 1.690002 2.260002
3  1/10/2006 1.389999 1.239997 1.270005 1.690002
4  1/11/2006 1.659996 1.389999 1.239997 1.270005
5  1/12/2006 1.919998 1.659996 1.389999 1.239997
6  1/13/2006 1.130005 1.919998 1.659996 1.389999
7  1/17/2006 1.160004 1.130005 1.919998 1.659996
8  1/18/2006 2.600006 1.160004 1.130005 1.919998
9  1/19/2006 2.890000 2.600006 1.160004 1.130005
10 1/20/2006 2.500000 2.890000 2.600006 1.160004
> tail(dvnr,10)
      i..Date DVNrange Lag.1 Lag.2 Lag.3
3646  7/1/2020     0.82  0.65  0.44  0.72
3647  7/2/2020     0.52  0.82  0.65  0.44
3648  7/6/2020     0.75  0.52  0.82  0.65
3649  7/7/2020     0.67  0.75  0.52  0.82
```

I shuffled the data then created 2 variables, *InSample* (which has half of the rows) and *OutSample* (that has the remaining rows). *X.dvn* consists the columns 3 and 4, and *Y.dvn* consists of column 2 in dvnrange. After, I calculated the median of *Y.dvn*, which is 1.390002. This is important because the median will determine which ranges were high or low risk. NOTE: looking back at what I did, I realized that I should have set *X.dvn* to *dvnr[,3:5]* to cover all Lags. What I did here was cover only Lags 1 and 2.

```
> sample(5,5)
[1] 4 1 2 3 5
> Shuffle = sample(3655,3655)
> dvnr[Shuffle[1:6],]
      i..Date DVNrange    Lag.1    Lag.2    Lag.3
410  8/23/2007 1.559997 1.879997 1.769997 1.939994
1805 3/11/2013 0.919998 0.530002 1.570000 0.809997
3432 8/26/2019 0.769998 1.320000 0.660000 0.649999
2142 7/11/2014 1.040001 1.409996 1.210007 1.220001
2282 1/30/2015 3.250000 2.299999 3.209999 1.939999
2193 9/23/2014 0.980004 1.520004 1.379997 1.239998
> InSample = Shuffle[1:1800]
> OutSample = Shuffle[1801:3655]
> X.dvn=dvnr[,3:4]
> Y.dvn=dvnr[,2]
> median(Y.dvn)
[1] 1.390002
```

After completing that, I set 2 conditional statements for the variable *Y.dvn* to determine if the range is above or below the median.

```
> Y.dvn[Y.dvn>1.390002]="HighRisk"
> Y.dvn[Y.dvn<=1.390002]="LowRisk"
```

When examining Y.dvn, you can see how the program is catagorizing the range as LowRisk or HighRisk.

```
> Y.dvn[1:6]
[1] "LowRisk"  "LowRisk"  "LowRisk"  "HighRisk" "HighRisk" "LowRisk"
```

Next, it is time to set our Train and Test variables. *TrainX.dvn* are the Lags that were shuffled in rows between 1 and 1800. *TrainY.dvn* is the range column with the same amount of rows. TestX.dvn and TestY.dvn is the OutSample rows (1801:3655)

```
> TrainX.dvn=X.dvn[InSample,]
> TrainY.dvn=Y.dvn[InSample,]
Error in Y.dvn[InSample, ] : incorrect number of dimensions
> TrainY.dvn=Y.dvn[InSample]
> TestX.dvn=X.dvn[OutSample,]
> TestY.dvn=Y.dvn[OutSample]
```

Once the test and train variables are set up, we create our *knn.pred* variable then create our table with a k of 25. This table is showing us the successful predictions and errors within the table. As you can see, "HighRisk" predictions were accurate 675 times with 214 errors. Also, we see that "LowRisk" predictions have 693 accurate predictions with 273 errors.

```
> knn.pred = knn(TrainX.dvn,TestX.dvn,TrainY.dvn,25)
> table(knn.pred,TestY.dvn)
          TestY.dvn
knn.pred   HighRisk LowRisk
  HighRisk      675     214
  LowRisk       273     693
```

Finally, we take our table and assign it to a variable named table.out to make the calculations necessary to find its percentage of accuracy. As shown here, our percent of accuracy of our predictions are 74%.

```
> table.out = table(knn.pred,TestY.dvn)
> (table.out[1,1]+table.out[2,2])/sum(table.out)
[1] 0.7374663
```

APPENDIX

```
ls()
library(class)
SMarketTatum = read.csv("SMarketTatum.csv")
head(SmarketTatum,10)
head(SMarketTatum,10)
Lag1 = SMarketTatum$Lag1
Lag2 = SMarketTatum$Lag2
Year = SMarketTatum$Year
train = (Year<2005)
SMarketTatum.2005 = SMarketTatum[!train]
SMarketTatum.2005 = SMarketTatum[!train,]
Direction = SMarketTatum$Direction
Direction.2005 = Direction[!train,]
Direction.2005 = Direction[!train]
train.X = cbind(Lag,Lag2)[train,]
train.X = cbind(Lag1,Lag2)[train,]
test.X = cbind(Lag1,Lag2)[!train,]
train.Direction = Direction[train]
knn.pred = knn(train.X,test.X,train.Direction,k=1)
table(knn.pred,Direction.2005)
history()
p = (84+43)/(84+43+68+57)
p
knn.pred = knn(train.X,test.X,train.Direction,k=3)
table(knn.pred,Direction.2005)
history()

dvn = read.csv("DVN.csv")
head(dvn,10)
dvn = dvn[-1, ]
head(dvn,10)
q()
dvn = read.csv("DVN2008.csv")
head(dvn,10)
dvn = dvn[-1,]
head(dvn,10)
library(class)
knn
Lag1 <- dvn$Lag1
Lag2 <- dvn$Lag2
Year <- dvn$Year
train = (Year<2020)
dvn2020 = dvn[!train,]
Direction <- dvn$Direction
Direction2020 = Direction[!train]
train.X=cbind(Lag1 ,Lag2)[train ,]
test.X=cbind (Lag1 ,Lag2)[!train ,]
train.Direction =Direction [train]
knn.pred=knn(train.X,test.X,train.Direction,k=1)
table(knn.pred,Direction2020)
p <- (45+43)/(45+16+30+43)
p
knn.pred=knn(train.X,test.X,train.Direction,k=3)
table(knn.pred,Direction2020)
p <- (47+39)/(47+20+28+39)
p
dvnrange = read.csv("dvnrange.csv")
head(dvnrange,10)
tail(dvnrange,10)
dvnrange = dvnrange[-1:-2,]
head(dvnrange,10)
history()
```

```
history(max.show=inf)
history(max.show)
history(inf)
history(max.show=100)

dvnr = read.csv("dvnrange.csv")
head(dvnr,10)
tail(dvnr,10)
history(max.show=150)
sample(5,5)
Shuffle = sample(3655,3655)
dvnr[Shuffle[1:6],]
InSample = Shuffle[1:1800]
OutSample = Shuffle[1801:3655]
X.dvn=dvnr[,3:4]
Y.dvn=dvnr[,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])
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)
table.out = table(knn.pred,TestY.dvn)
(table.out[1,1]+table.out[2,2])/sum(table.out)
history()
```