Problem 1: Predicting Housing Prices in Ames, Iowa, Revisited

a

```
lm.fit = lm(SalePrice ~ ., data=ames.train)
summary(lm.fit)
```

There are 57 statistically significant at the 95% level. R tells us that "Coefficients: (8 not defined because of singularities)", indicating that we have linear dependencies in our data. For example, the total amount of square feet in the basement is the sum of the other area variables for the basement area.

b

```
cpMax <- 0.0005
cpMin <- 0.000
cpStep <- (cpMax - cpMin)/100
refit = TRUE
if (refit){
  set.seed(123) # Reproduce results in case only this chunk is executed
  cv.amestree = train(SalePrice ~.,
                 data = ames.train,
                 method = "rpart",
                 trControl = trainControl(method="cv"),
                 metric = "Rsquared",
                 tuneGrid = data.frame(.cp=seq(cpMin, cpMax, by = cpStep)))
}

cv.amestree$bestTune

##         cp
## 39 0.00019
```

So the best value of cp is 0.00019.

The root node is still the external quality, and for the next levels, the size of the house and garage, as well as the neighborhood are important. The whole tree can be seen in fig. 1

c

```
refitForest = TRUE
if (refitForest){
  set.seed(123)
  rf.cv = train(y = ames.train$SalePrice,
                x = subset(ames.train, select=-c(SalePrice)),
                method="rf",
                nodesize=25,
                ntree=80,
                trControl=trainControl(method="cv", number=10),
                tuneGrid=data.frame(mtry=seq(1,20,1))
  )
}
rf.cv$bestTune

##    mtry
## 19   19
```

So the best value of mtry is 19.

```
important_vars = importance(rf.cv)
tail(important_vars[order(important_vars),], 5)

##   GarageCars    YearBuilt    ExterQual    GrLivArea Neighborhood
## 5.869225e+11 6.157545e+11 8.993100e+11 1.070505e+12 1.818242e+12
```

The 5 most important variables are (in decreasing order): neighborhood, living area of ground floor, the external quality, construction year and the capacity of cars in the garage. This matches fairly well with the results from c).

```
rmse = function(preds, actual) {
  n = length(actual)
  sqe = sum((preds - actual)**2)
  return(sqrt(sqe/n))
```

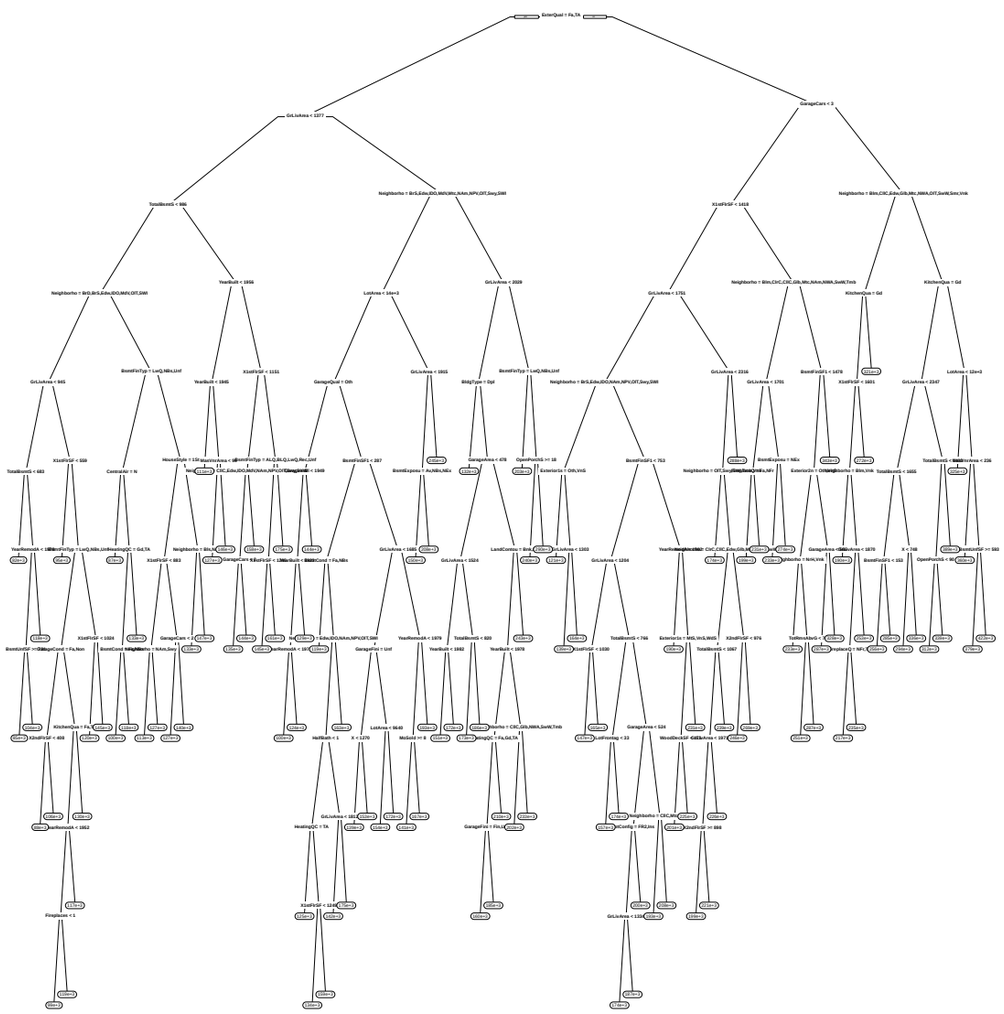

Figure 1: The best tree as chosen by the cross validation

```
}

mae = function(preds, actual) {
  n = length(preds)
  return(sum(abs(preds - actual)/n))
}

rsq = function(preds, actual) {
  avg = mean(actual)

  tss = sum((actual - avg)**2)
  rss = sum((actual - preds)**2)
  return(1 - rss/tss)
}

metricsFn = function(model) {
  preds.test = predict(model, newdata = ames.test)
  preds.train = predict(model, newdata = ames.train)
  cat("RMSE train")
  print(rmse(preds.train, ames.train$SalePrice))

  cat("RMSE test")
  print(rmse(preds.test, ames.test$SalePrice))

  cat("MAE train")
  print(mae(preds.train, ames.train$SalePrice))

  cat("MAE test")
  print(mae(preds.test, ames.test$SalePrice))

  cat("R^2 train")
  print(rsq(preds.train, ames.train$SalePrice))

  cat("R^2 test")
  print(rsq(preds.test, ames.test$SalePrice))
}

metricsFn(lm.fit)

## RMSE train[1] 22476.65
## RMSE test[1] 27292.6
## MAE train[1] 14638.33
## MAE test[1] 16123.29
## R^2 train[1] 0.8980057
## R^2 test[1] 0.8449949

metricsFn(cvtree)

## RMSE train[1] 20406.07
## RMSE test[1] 32378.37
## MAE train[1] 14037.93
## MAE test[1] 22148
## R^2 train[1] 0.9159319
## R^2 test[1] 0.7818445

metricsFn(rf.cv)

## RMSE train[1] 15720.94
## RMSE test[1] 22919.09
## MAE train[1] 10447.38
## MAE test[1] 15306.29
## R^2 train[1] 0.9501036
## R^2 test[1] 0.8906923
```

e

Based of these numbers I would suggest using the random forest model, as it outperforms the other models for every metric. We lose a good amount of interpretability compared to the other models, especially the linear one. The CART model is something in between, but in the end I feel increased prediction quality is the most important thing to consider.

Problem 2: Clustering Stock Returns

```
entries = aggregate(data, by=list(data$Industry), FUN=length)
rownames(entries) = entries$Group.1
entries$N = entries$avg200603
subset(entries, select = (N))

##                               N
## Consumer Discretionary       69
## Consumer Staples             32
## Energy                       38
## Financials                   78
## Health Care                  44
## Industrials                  55
## Information Technology       56
## Materials                    28
## Telecommunications Services   5
## Utilities                    28
```

The amount of companies in each sector can be seen above. There are for example 78 companies in the financials industry.

```
first = which(colnames(returns)=="avg200801")
last = which(colnames(returns)=="avg201012")
aggReturns = aggregate(returns[c(first:last)], by=list(data$Industry), FUN=mean)
rownames(aggReturns) = aggReturns$Group.1
aggReturns = subset(aggReturns, select = -c(Group.1))

df = as.data.frame(t(aggReturns))
df$Month = rownames(df)
df = melt(df, id.vars=c("Month"))
df$Returns = df$value
par(mar = c(0.1, 0.1, 0.1, 0.1))
ggplot(df, aes(x=Month, y=Returns, color=variable, group=variable)) +
  geom_line() +
  theme(axis.text.x = element_text(angle=80, hjust = 1))
```

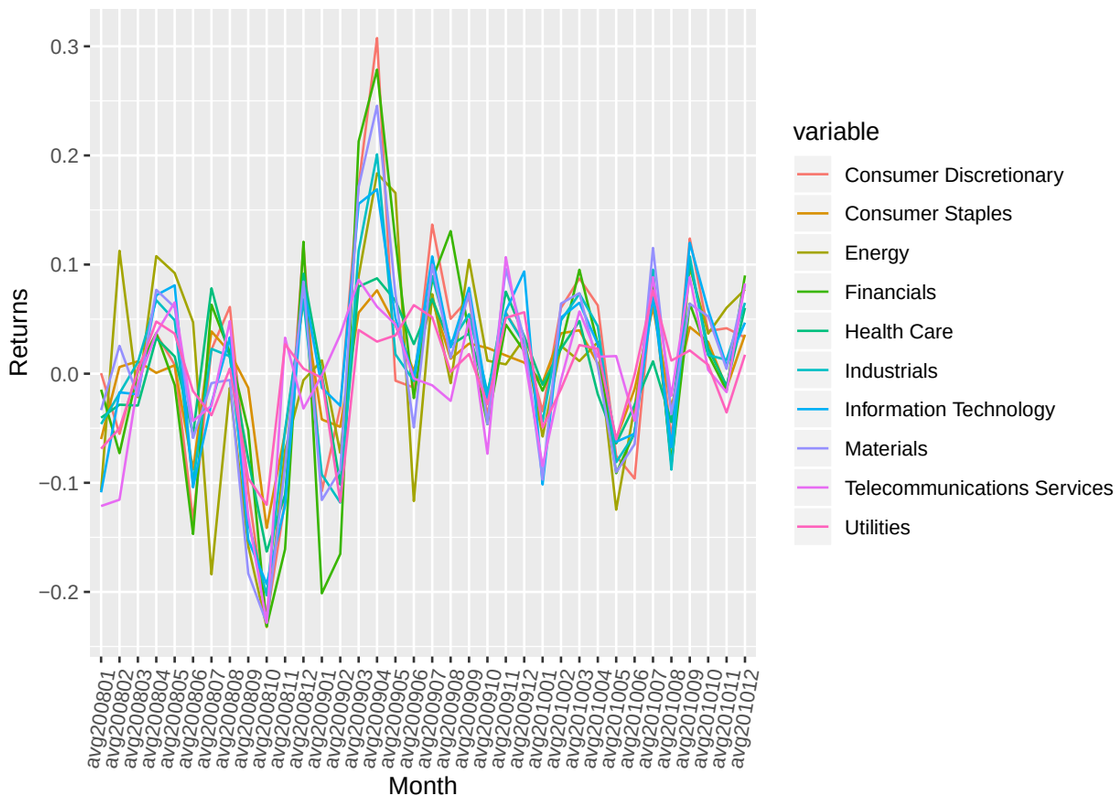

Figure 2: The average monthly stock return for 2008 - 2010

As seen in fig. 2 the sectors develop fairly correlated. The crisis in the fall of 2008 can be seen as a big dip for all sectors, and then the bounce back in 2009.

```r
d <- dist(returns)
hclust.mod <- hclust(d, method="ward.D2")
plot(hclust.mod, labels=F, ylab="Dissimilarity", xlab = "", sub = "")
```

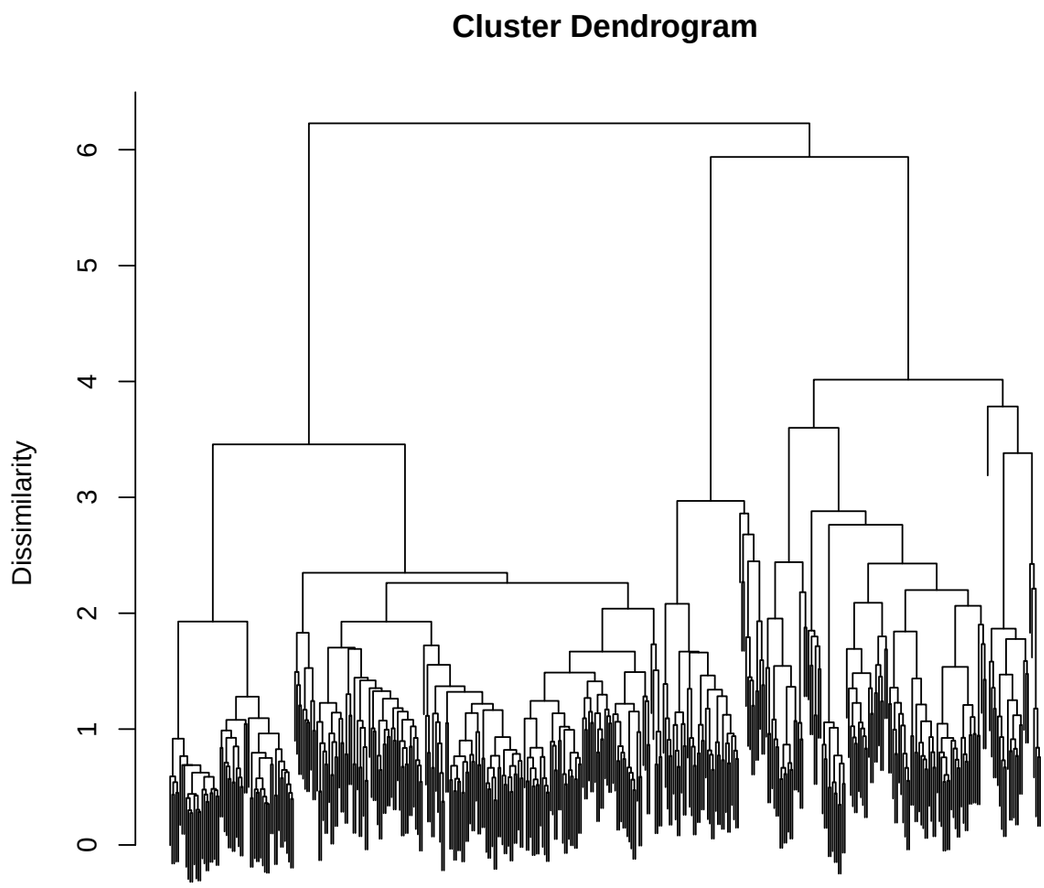

Figure 3: Dendrogram for the stock data

```r
hc.dissim <- data.frame(k = seq_along(hclust.mod$height),
                        dissimilarity = rev(hclust.mod$height))

plot(hc.dissim$k, hc.dissim$dissimilarity, type="l", xlim=c(0,30))
```

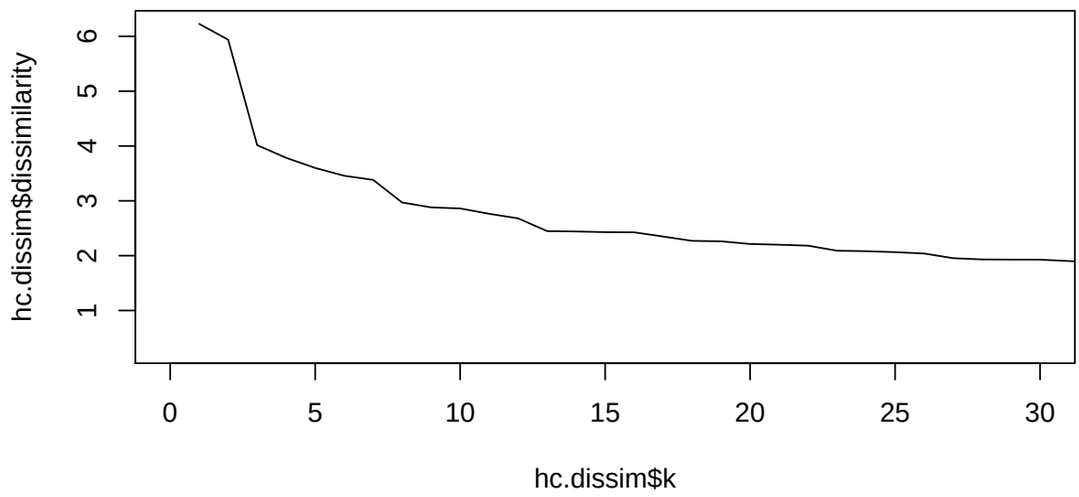

Figure 4: Skree plot for the stock data

b

From the skree plot in fig. 4 a good choice might be $k = 6$, as this value is at the breaking point of the curve.

c

```r
N_CLUSTERS = 6
h.clusters <- cutree(hclust.mod, N_CLUSTERS)
```

```r
cluster.avgR = aggregate(data, by=list(h.clusters), FUN=mean)
cluster.avgR$avg200810

## [1] -0.1510996 -0.2763859 -0.2602381 -0.4932965 -0.1120172 -0.3555063

cluster.avgR$avg200903

## [1] 0.09765364 0.19541286 0.10755846 0.94117177 0.24063092 0.21884056
```

The distribution of industries in each cluster can be seen in fig. 5. Cluster 1 makes up the majority of companies, and this is the cluster for companies who did not experience much volatility (relatively) during the crisis. Most of the companies in the industries serving basic necessities fall in this catgory: utilities, health care, telecom and consumer staples.

The next two clusters (2 and 3) is for the companies in the medium volatility range, with cluster 2 recovering more during March 2009 than cluster 3. Interestingly, we find almost all energy companies in cluster 3, and not cluster 1.

Cluster 6 is the most affected of these 4 clusters, losing 35%.

Cluster 4 is AIG, which crashed spectacularly in 2008, and I would consider it an outlier for this dataset. The fact that they gained 94 percent in March 2009 does not mean much, as their stock was very low at this point compared to before the crash.

Cluster 5 is interesting, as they seem to consist mostly of financial companies that were able to recover to some extent in 2009.

d

```r
set.seed(123) # Reproduce results
km <- kmeans(returns, centers = N_CLUSTERS, iter.max=100)
```

```
df = as.data.frame(table(h.clusters, data$Industry))
ggplot(df, aes(x=h.clusters, y=Freq, fill=Var2)) + geom_bar(stat="identity") +
  xlab("Cluster") + ylab("Count")
```

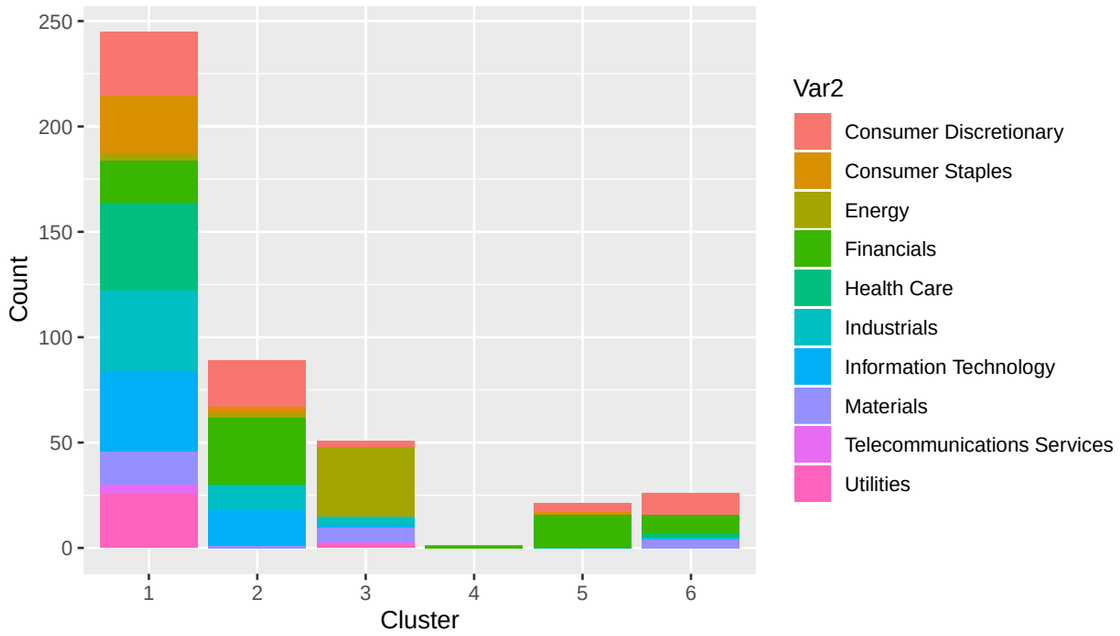

Figure 5: Industries in each hierarchical cluster

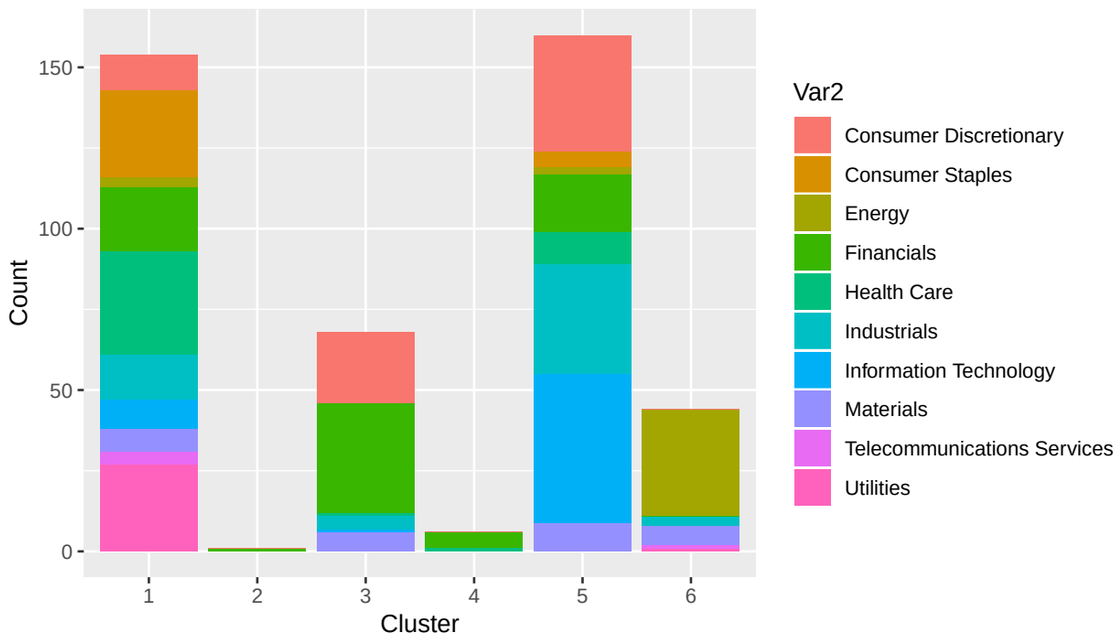

Figure 6: Industries in each kmeans cluster

```r
cluster.avgR = aggregate(data, by=list(km$cluster), FUN=mean)
cluster.avgR$avg200810
```

```
## [1] -0.1116491 -0.4932965 -0.2863172 -0.5038802 -0.2178566 -0.2713985
```

```r
cluster.avgR$avg200903
```

```
## [1] 0.07672832 0.94117177 0.22386421 0.17827922 0.15437714 0.10764831
```

The distribution of industries in the resulting kmeans clusters can be seen in fig. 6. The kmeans cluster 1 and 5 is together pretty much the same as cluster 1 and 2 in the hierarchical clustering, but the kmeans has placed a lot more companies in the second group.

Cluster 2 is again just AIG, while clusters 3 and 4 is similar to clusters 5 and 6 in the hierarchical clustering. Lastly, cluster 6 is very similar to cluster 3 in the hierarchical clustering.

e

Using this model, an investor can identify different classes of companies. The model considers the monthly returns over the period from 2006 to 2016, and groups the companies from these returns. So companies from different clusters have a different patterns in their returns, and this can therefore be used as a tool to diversify investments.