Clustering

Load the data

Dataset: Kansas City House Data via Kaggle.

```
df <- read.csv("kc_final.csv", header=TRUE)
set.seed(1234)
df <- df[sample(1:nrow(df), 10000, replace=FALSE),]
df <- subset(df, select=-c(X, id, date))
```

Explore statistically and graphically

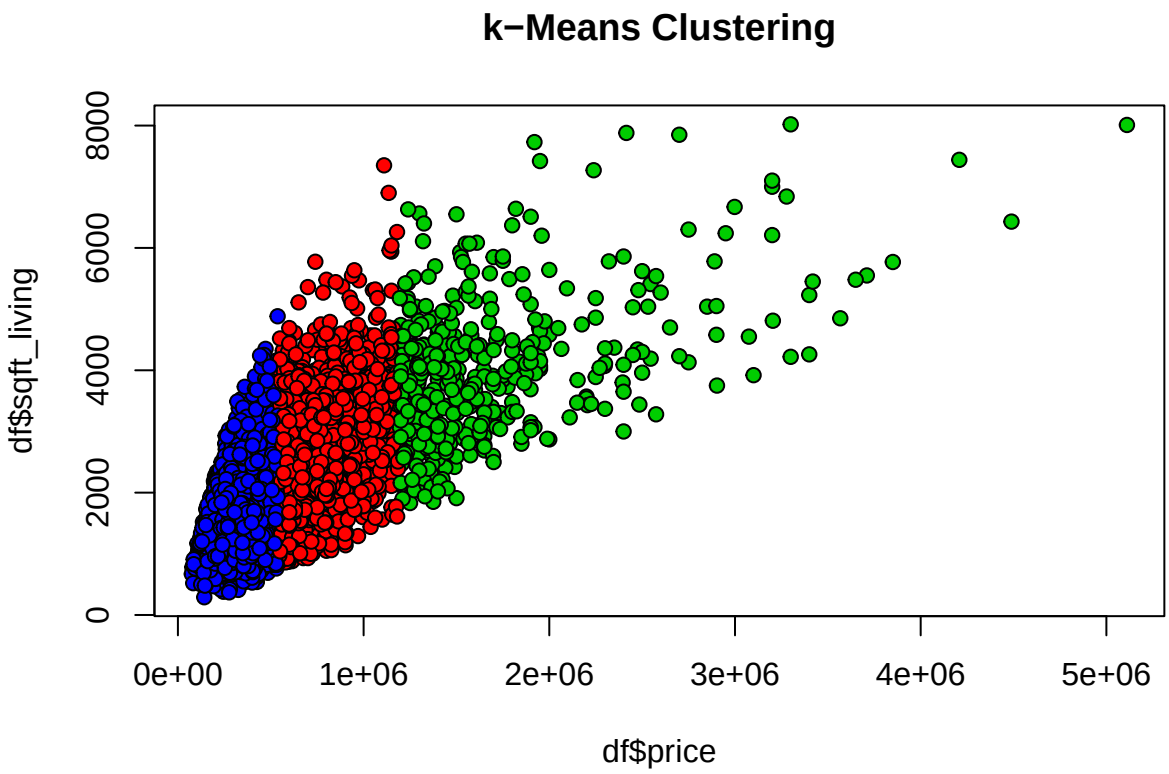
A good graphical illustration of the data may be seen below in the k-Means example (input data is basically the same as what is shown geometrically in that graph, and the colors are generated as a result of the clustering). We can print out part of the dataframe to learn about the format of data it contains.

```
str(df)
```

```
## 'data.frame':    10000 obs. of  19 variables:
##  $ price        : num  600000 606000 660000 537000 975000 ...
##  $ bedrooms     : int  3 3 3 4 3 3 4 4 3 3 ...
##  $ bathrooms    : num  1 2 3.5 2.5 2.5 1.5 2.5 1.5 2.25 1.5 ...
##  $ sqft_living  : int  940 1980 2740 1990 2530 1210 2320 1840 1560 2290 ...
##  $ sqft_lot     : int  19000 7680 3785 2660 7000 10588 9264 7076 35026 9600 ...
##  $ floors       : num  1 1.5 2 2 2.5 1 2 1.5 1 1 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 4 0 0 0 0 0 ...
##  $ condition    : int  3 4 3 3 3 4 3 3 3 4 ...
##  $ grade        : int  6 6 9 8 9 7 8 7 7 7 ...
##  $ sqft_above   : int  940 1070 2190 1990 2530 1210 2320 1840 1290 2290 ...
##  $ sqft_basement: int  0 910 550 0 0 0 0 0 270 0 ...
##  $ yr_built     : int  1945 1911 2001 2012 1915 1958 1994 1957 1985 1967 ...
##  $ yr_renovated : int  0 0 0 0 1999 0 0 0 0 0 ...
##  $ zipcode      : int  98004 98033 98034 98034 98136 98002 98188 98106 98092 98042 ...
##  $ lat          : num  47.6 47.7 47.7 47.7 47.5 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  2280 1330 2060 1990 2380 1408 2320 1510 1660 1310 ...
##  $ sqft_lot15   : int  19000 8704 3457 2665 7000 10588 9129 7320 35160 9600 ...
```

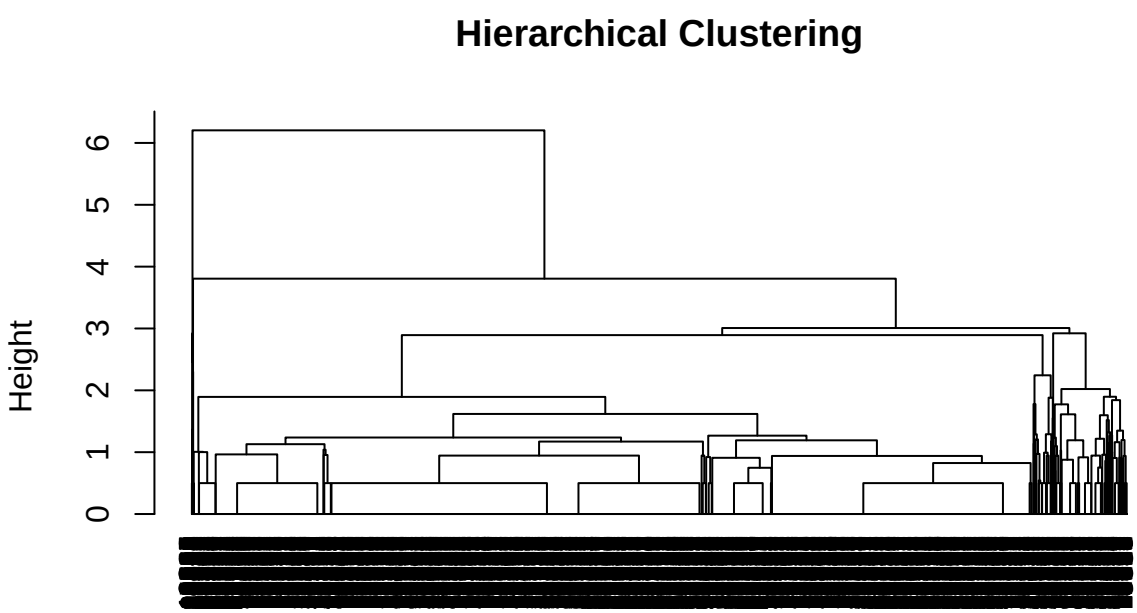
k-Means

```
kmeans1 <- kmeans(df[c("price", "sqft_living")], 3, nstart=20)
plot(df$price, df$sqft_living, pch=21, bg=c("red", "green3", "blue")[unclass(kmeans1$cluster)], main="k
```



Hierarchical

```
dist1 <- dist(df[c("bedrooms", "floors", "view")])
hclust1 <- hclust(dist1, method="average")
plot(hclust1, hang=-1, cex=0.8, main="Hierarchical Clustering")
```



Model-based

```
library(mclust)
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
mclust1 <- Mclust(df[c("bedrooms", "floors", "view")])
summary(mclust1)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEV (ellipsoidal, equal volume and shape) model with 8 components:
##
##  log-likelihood     n df       BIC       ICL
##       -15956.03 10000 58 -32446.26 -42909.91
##
## Clustering table:
##    1    2    3    4    5    6    7    8
##    0    0  509 8715  335  161  280    0
```

Comparison and analysis

The k-Means clustering yields the most intuitive results, as it simply groups together points on a graph similarly to how humans would identify clumps of points near each other. It clearly identifies three classes, low-, medium-, and high-end houses. The hierarchical clustering algorithm is quite impractical on this dataset; it generates a hugely complex clustering system which is difficult to reason about even enough to determine how to improve it. The complexity grows as more input attributes are considered. Model-based clustering is a little strange and difficult to interpret/visualize, but its convenience lies in the fact that it attempts to automatically select the optimum clustering model based on various statistical metrics.