Stat 302
Statistical Software and Its Applications
Classification and Clustering

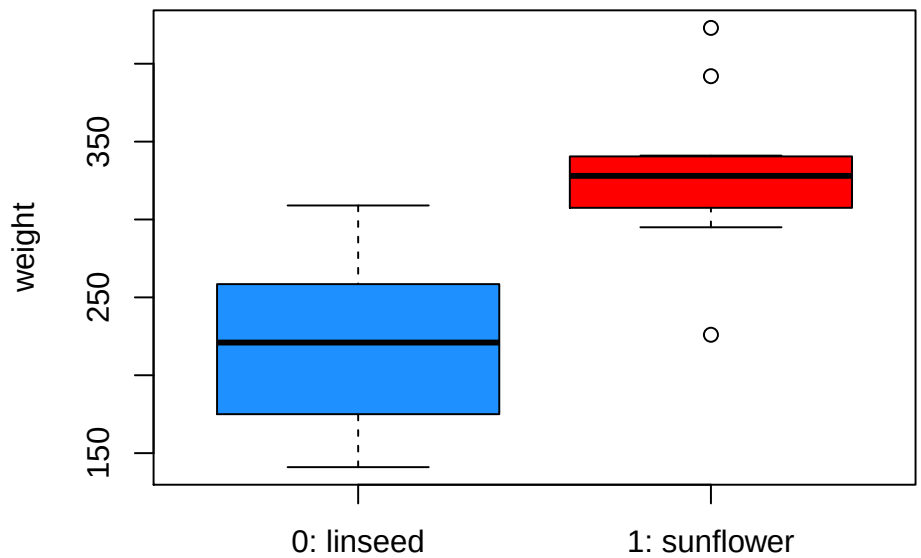
Yen-Chi Chen

Department of Statistics, University of Washington

Autumn 2016

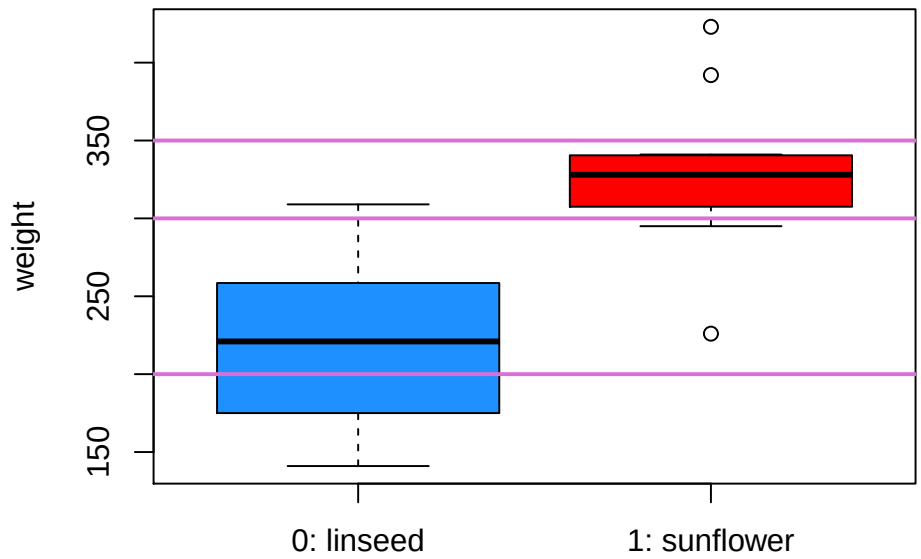
Motivating Example: chickwts Dataset – 1

For simplicity, we consider `linseed` and `sunflower` feed. Here is the boxplot for `weight` of chicken from these two groups.



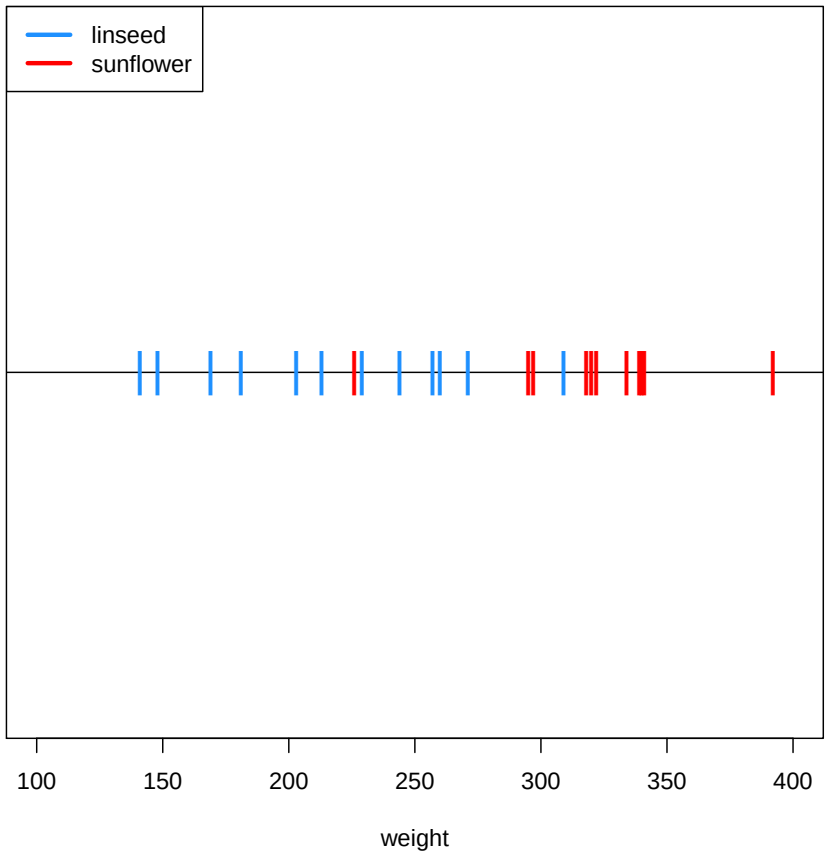
$\longrightarrow$ If now I give you a chicken with some value of `weight`, which feed will you predict?

Motivating Example: chickwts Dataset – 2



$\longrightarrow$ What feed will you predict for `weight = 200, 300` and `350`?

Motivating Example: chickwts Dataset – 3



$\longrightarrow$ What feed will you predict for `weight = 200, 300` and `350`?

kNN Method – 1

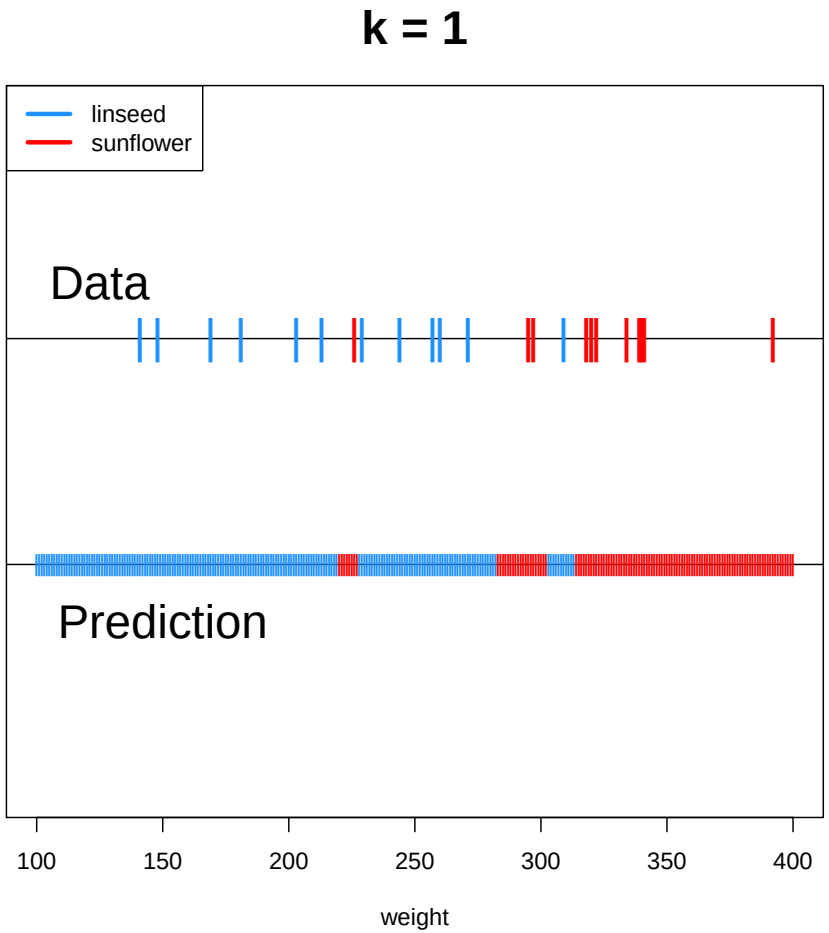
- A simple method is called *k-nearest neighbor* (kNN) method.
- For a given point `Sepal.length=x`, we find the k observations whose `Sepal.length` is closest to `x`.
- Different k yields different result.
- In R, we will use the function `knn()` in the library `class`.

```
> x_grid <- seq(from = 100, to=400, by=1)
> label_grid <- knn(train=as.matrix(data_x),
+                   test=as.matrix(x_grid),
+                   cl = label_y,k=11)
```

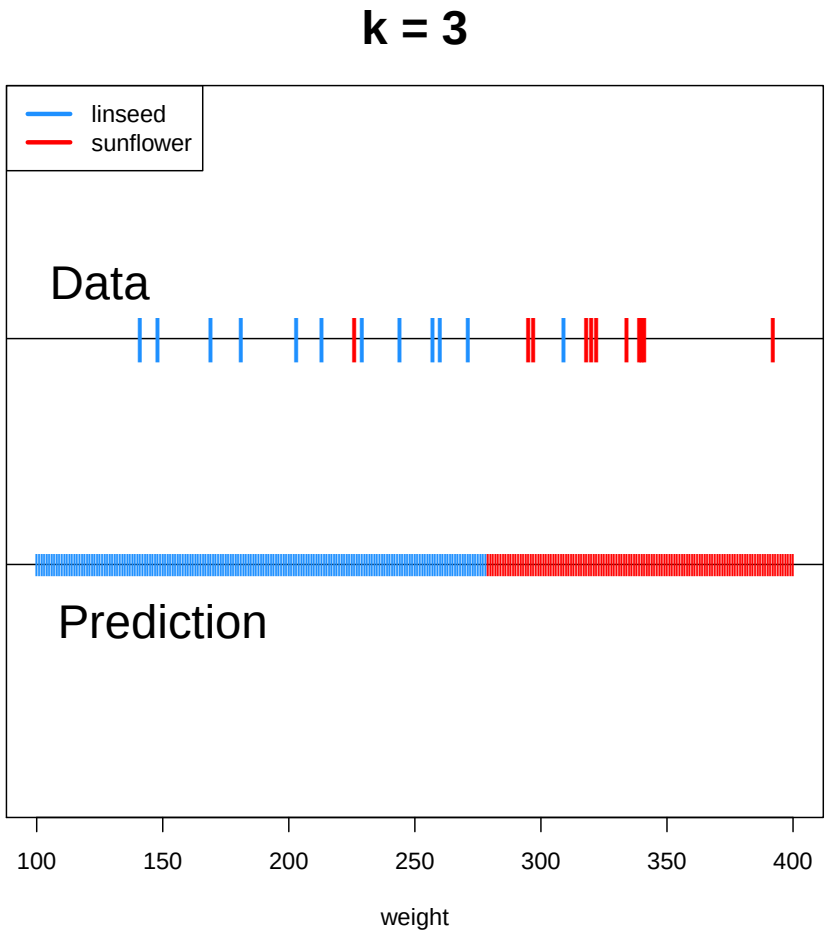
kNN Method – 2

```
> plot(NULL, xlim=c(100,400), ylim=c(-2,1), xlab="weight",
+      yaxt="n", ylab="", main="k = 1", cex.main=2)
> abline(h=0)
> points(x=data_x[label_y==0], y=rep(0, sum(label_y==0)),
+        col="dodgerblue", cex=2, pch="|")
> points(x=data_x[label_y==1], y=rep(0, sum(label_y==1)),
+        col="red", cex=2, pch="|")
> abline(h=-1)
> points(x=x_grid[label_grid==0], y=rep(-1, sum(label_grid==0)),
+        col="dodgerblue", cex=1, pch="|")
> points(x=x_grid[label_grid==1], y=rep(-1, sum(label_grid==1)),
+        col="red", cex=1, pch="|")
> text("Data", x=125, y=0.25, cex=2)
> text("Prediction", x=150, y=-1.25, cex=2)
> legend("topleft",c("linseed","sunflower"),
+        col=c("dodgerblue","red"), lwd=3)
```

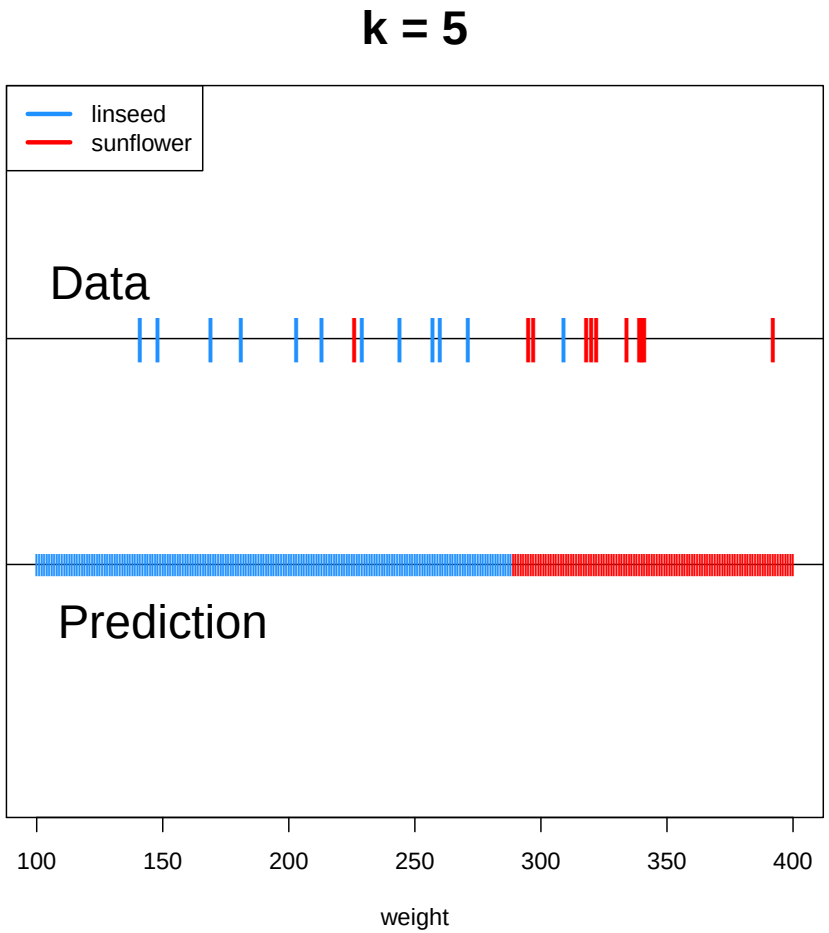
kNN Method – 3


k = 1
linseed
sunflower
Data
Prediction
100
150
200
250
300
350
400
weight

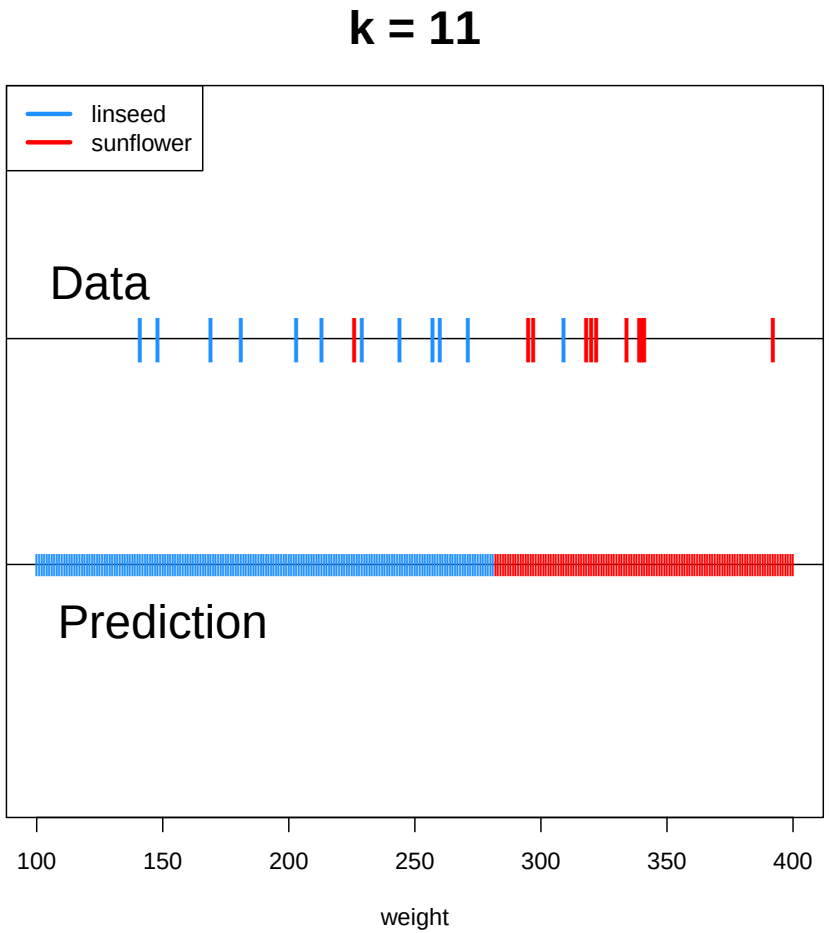
kNN Method – 4


k = 3
linseed
sunflower
Data
Prediction
100
150
200
250
300
350
400
weight

kNN Method – 5


k = 5
linseed
sunflower
Data
Prediction
100
150
200
250
300
350
400
weight

kNN Method – 6


k = 11
linseed
sunflower
Data
Prediction
100
150
200
250
300
350
400
weight

kNN Method – 7

- The quantity k matters a lot!
- How are we going to choose it?
- Generally, we choose k to minimize the *prediction error*.
- Basic idea: we know the actual labels of the data so we can evaluate our prediction error.
- However, we are reusing the data for many times and choose the optimal error.
- We would generally end up being too optimistic about the optimal prediction error–this is called *overfitting*.
- There are some ways to deal with this problem such as *cross-validation*; if you are interested in the classification problem, I highly recommend you to learn this concept.

Classification based on Probability – 1

- We can do classification based on a statistical (probability) model.
- The task of classification is: having observed `weight`, which `feed` should be predict.
- A statistical model for this task is $P(\texttt{feed}|\texttt{weight})$.
- In our case, `feed= linseed` or `sunflower`; so we are interested in the quantities

$$P(\texttt{feed} = \texttt{linseed}|\texttt{weight}),$$
$$P(\texttt{feed} = \texttt{sunflower}|\texttt{weight}).$$

- A common strategy is to choose the `feed` that has a higher probability.

Classification based on Probability – 2

- Using the Bayes rule, comparing the above two probabilities is the same as comparing the following two quantities:

$$p(\texttt{weight}|\texttt{feed} = \texttt{linseed}) \cdot P(\texttt{feed} = \texttt{linseed}),$$
$$p(\texttt{weight}|\texttt{feed} = \texttt{sunflower}) \cdot P(\texttt{feed} = \texttt{sunflower}).$$

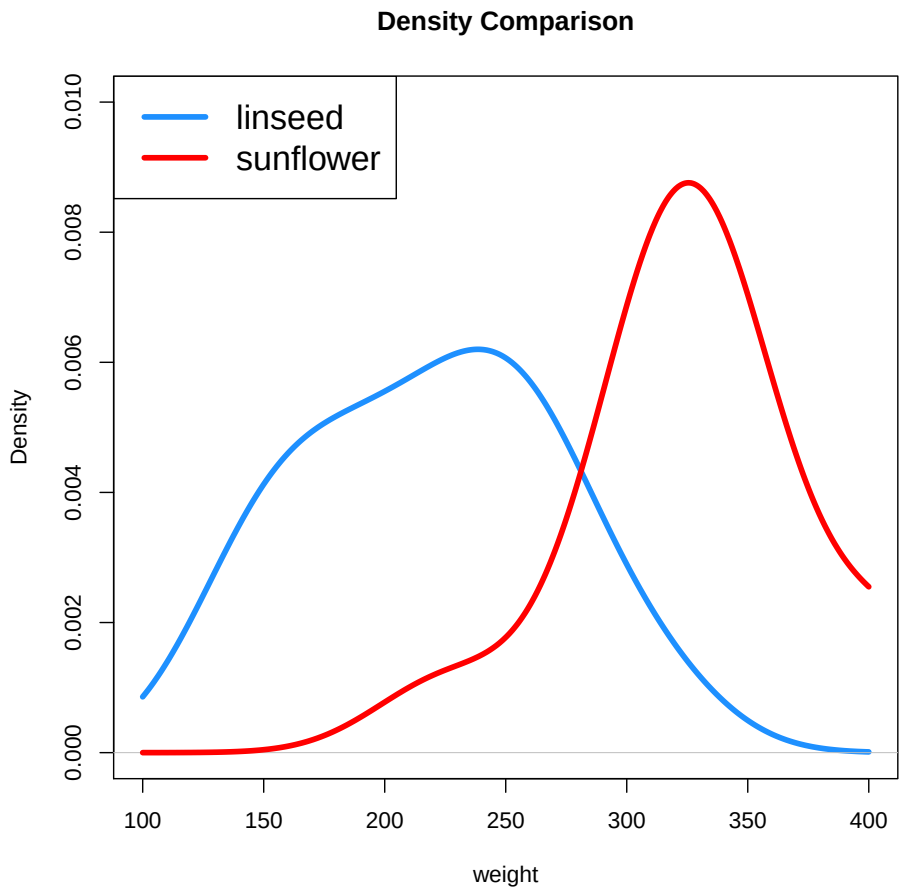
- Based on the data, the above quantities can be replaced by

$$\hat{p}(\texttt{weight}|\texttt{feed} = \texttt{linseed}) \cdot N(\texttt{feed} = \texttt{linseed}),$$
$$\hat{p}(\texttt{weight}|\texttt{feed} = \texttt{sunflower}) \cdot N(\texttt{feed} = \texttt{sunflower}),$$

  where $\hat{p}(\texttt{weight}|\texttt{feed} = \texttt{X})$ is the estimated density using those observations whose `feed = X` and $N(\texttt{feed} = \texttt{linseed})$ is the number of observations whose `feed = X`.
- If the two groups (`feed`) have equal size, we can directly compare their density.

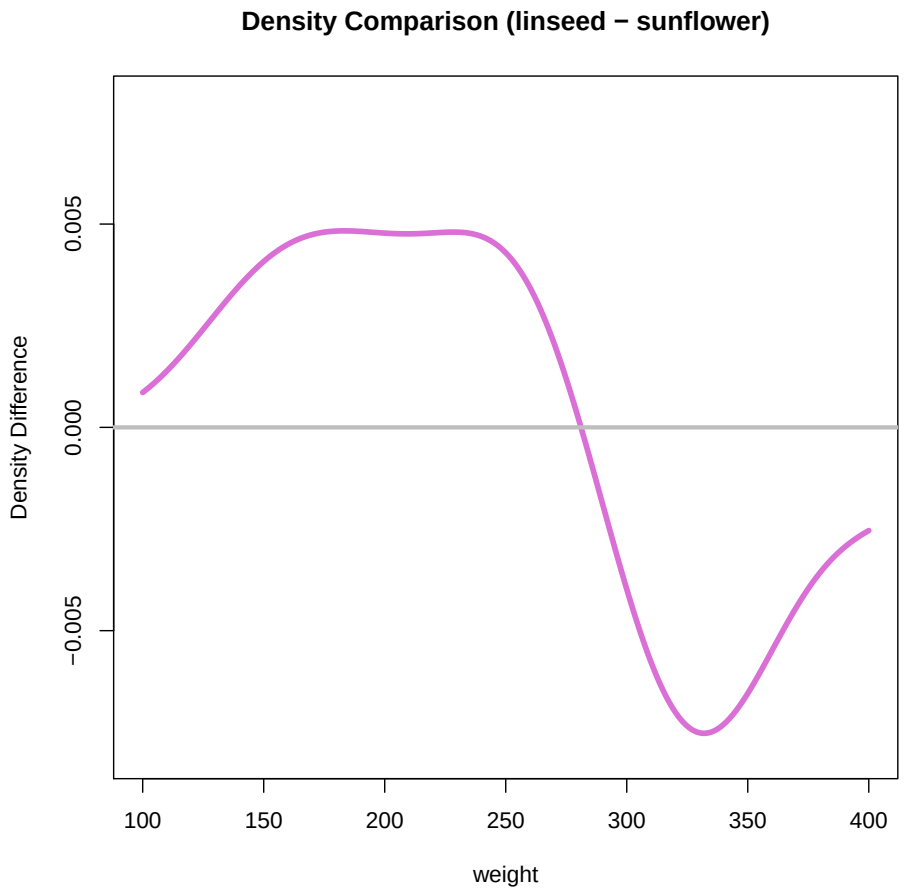
Classification based on Probability – 3

```
> plot(density(data_x[label_y==0], bw=30, from=100, to=400),
+      lwd=4, col="dodgerblue", ylim=c(0,0.01),
+      main="Density Comparison", xlab="weight")
> lines(density(data_x[label_y==1], bw=30, from=100, to=400),
+       lwd=4, col="red")
> legend("topleft",c("linseed","sunflower"),
+        col=c("dodgerblue","red"), lwd=4, cex=1.5)
```



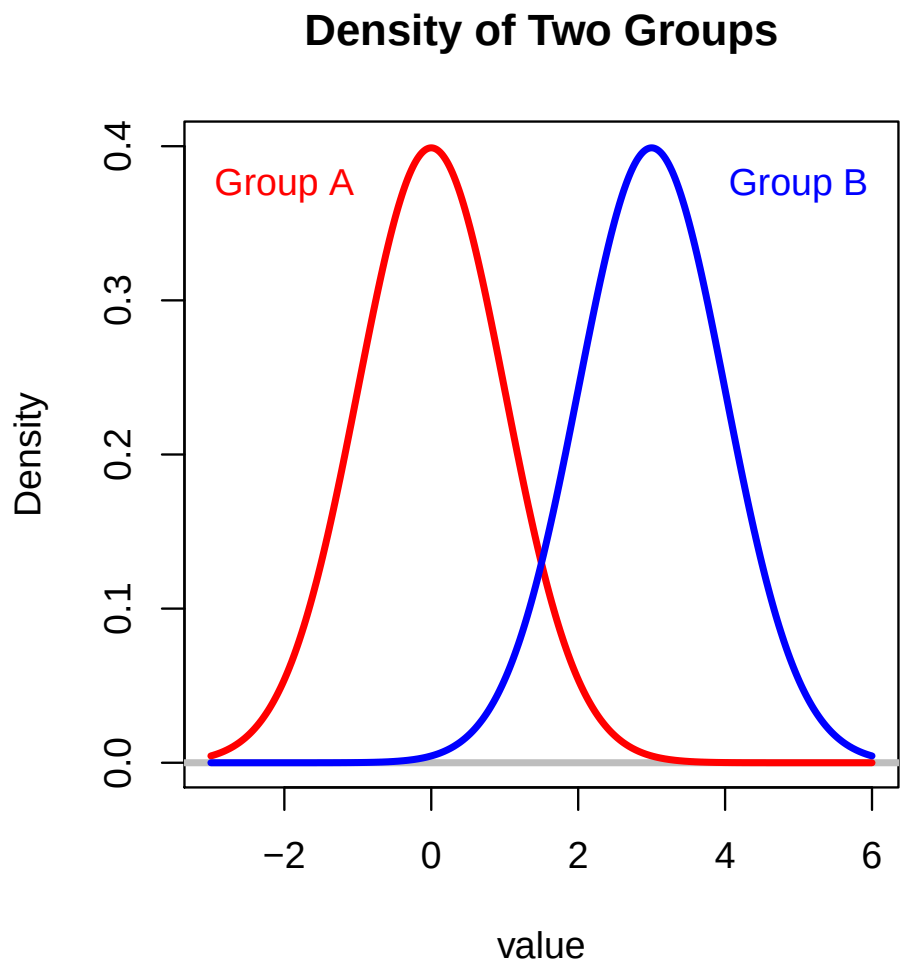
Classification based on Probability – 4

```
group0 <-density(data_x[label_y==0], bw=30, from=100, to=400)
group1 <-density(data_x[label_y==1], bw=30, from=100, to=400)
plot(group0$x, group0$y-group1$y,
     lwd=4, col="orchid", ylim=c(-0.008,0.008),
     main="Density Comparison (linseed - sunflower)",
     xlab="weight", type="l", ylab="Density Difference")
abline(h=0, lwd=3, col="gray")
```



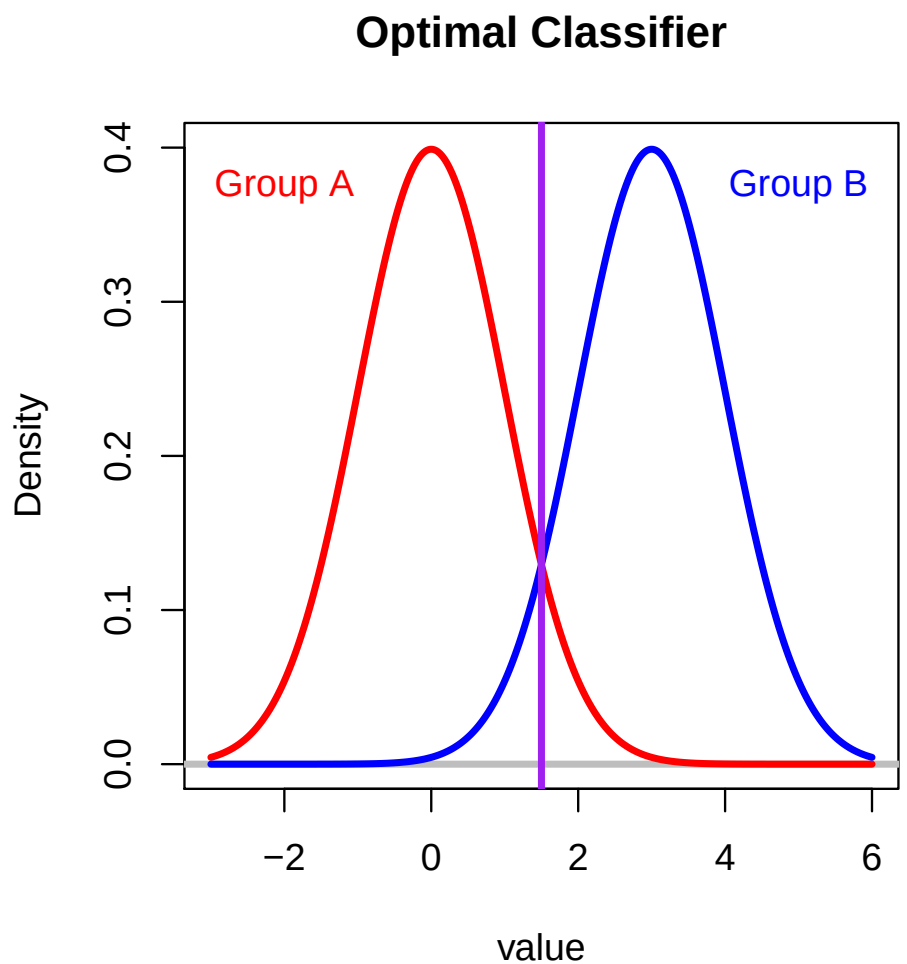
Classification based on Probability – 5

Consider the following example: the group A is from red distribution and group B is from the blue distribution. And we have equal probability to obtain a new observation from each group. Question: what is the best classifier?



Classification based on Probability – 6

The best classifier will be the purple line: we classify a point to be in the red group if its value is less than 1.5; otherwise we classify it to be in the blue group.



Classification based on Probability – 7

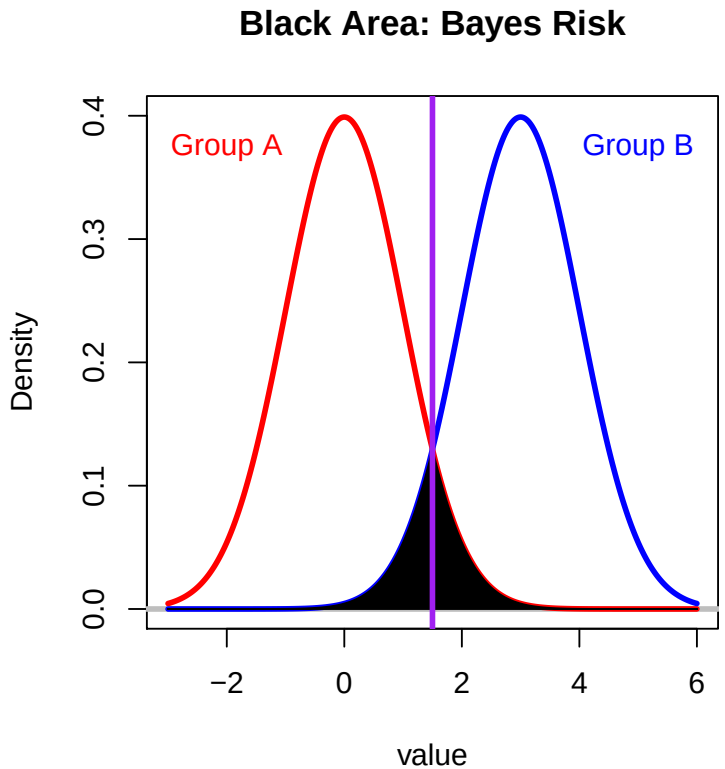
- How do we understand this optimal classifier?
- Given a value x, the probability of being in red/blue is

$$P(\text{Red Group}|x), \quad P(\text{Blue Group}|x).$$

- If we know these two probabilities, the optimal classifier is to classify x into the group with higher probability.
- Thus, the optimal classifier outputs a label

$$\begin{cases} \textit{red} & \text{if } P(\text{Red Group}|x) > P(\text{Blue Group}|x) \\ \textit{blue} & \text{if } P(\text{Red Group}|x) < P(\text{Blue Group}|x) \end{cases}.$$

However, even the optimal classifier has some prediction error, which is characterized by the black region.



There are some chances that points from the red group may be above 1.5. This unavoidable probability is called the *Bayes risk*.

Bayes Risk – 1

- The Bayes risk is *the error probability of the best classifier*.
- It is the error purely due to the randomness.
- The existence of the Bayes risk implies that even if we have done our best, there is still some misclassification errors.
- Just like many decision-making problems, we still make mistakes even we have made our best choice.

Bayes Risk – 2

```
> data_x <- c(rnorm(10000), rnorm(10000, mean=3))
> label_y <- c(rep(0,10000), rep(1,10000))
>
> classifier_y <- data_x>= 1.5
>   # the best classifier
> err <- sum(classifier_y!=label_y)/20000
> err
[1] 0.06835
```

$\longrightarrow$ this is the Bayes risk.

Classifiers: Comments

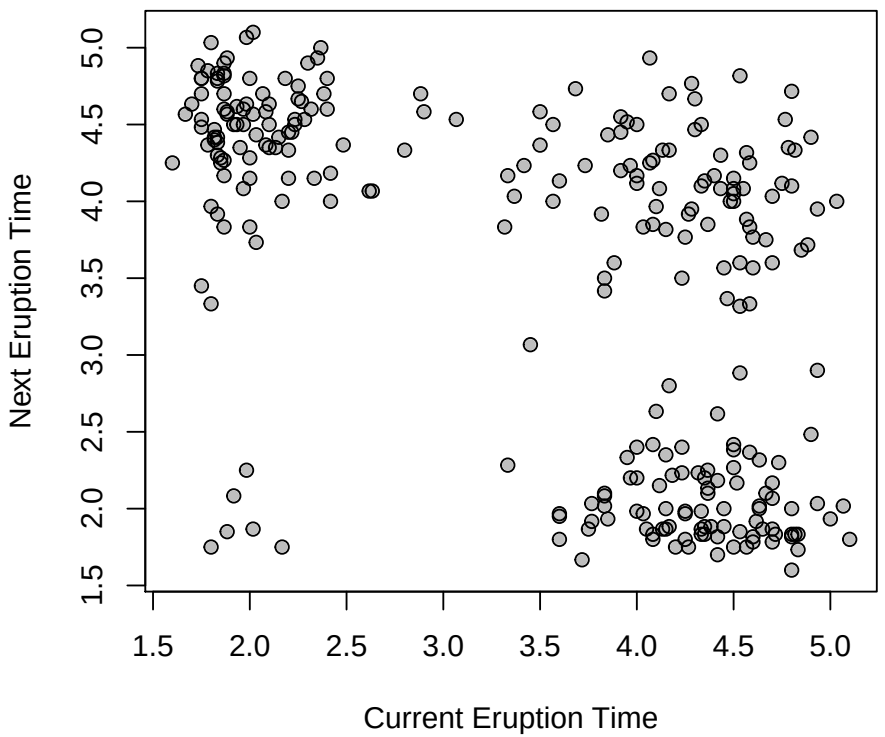
- In machine learning, classification and prediction is one of the main goals.
- An example: given an email, how Google can classify it as important/unimportant/spam?
- Here are a list of some common approaches:
  - Logistic Regression.
  - Support Vector Machine.
  - Random Forest.
  - Naive Bayes.
  - Boosting.
  - Deep Learning/Neural Network.
- You would learn more in STAT 425 (Introduction to Statistical Machine Learning).

Clustering: Introduction

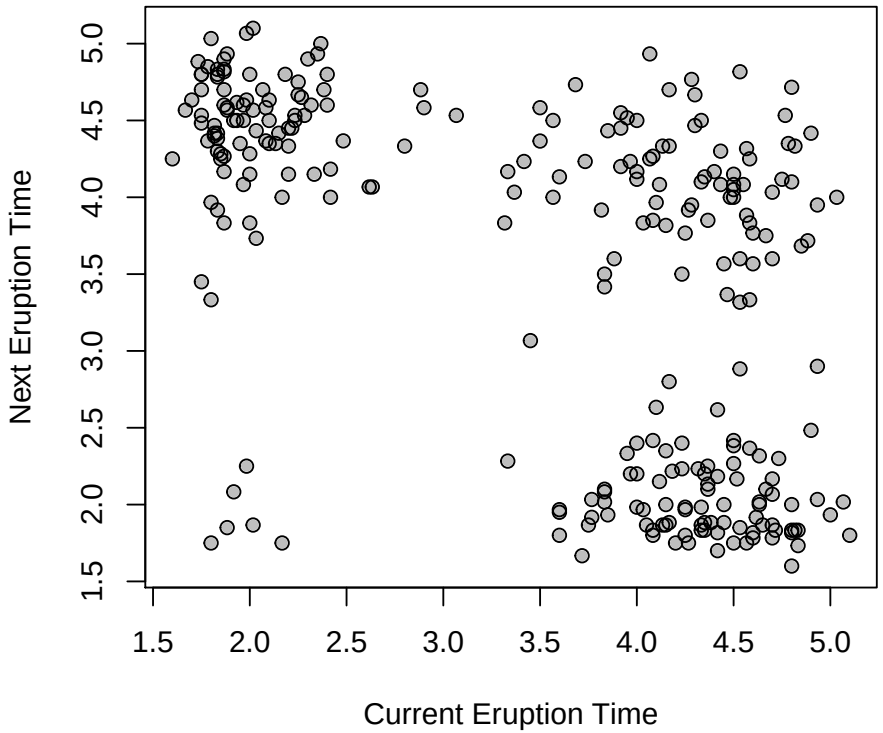
- Clustering is to group data into clusters.
- Ideally, we want points within the same clusters are similar to each other; points in different clusters are different from each other.
- Namely, we want to increase *within group similarity* and decrease *between group similarity*.
- Why do we want to do clustering?$\rightarrow$ in some scientific analysis, a cluster may correspond to observations generated by the same/similar procedure.
- A main difference between classification and clustering is that in classification, we have labels for our observations, but in clustering, we do not have labels.

Clustering: Old Faithful Dataset – 1

```
> data1 <- cbind(faithful$eruptions[1:271],
+                faithful$eruptions[2:272])
> plot(data1, xlab="Current Eruption Time",
+      ylab="Next Eruption Time", col="gray",
+      pch=20)
> points(data1)
```



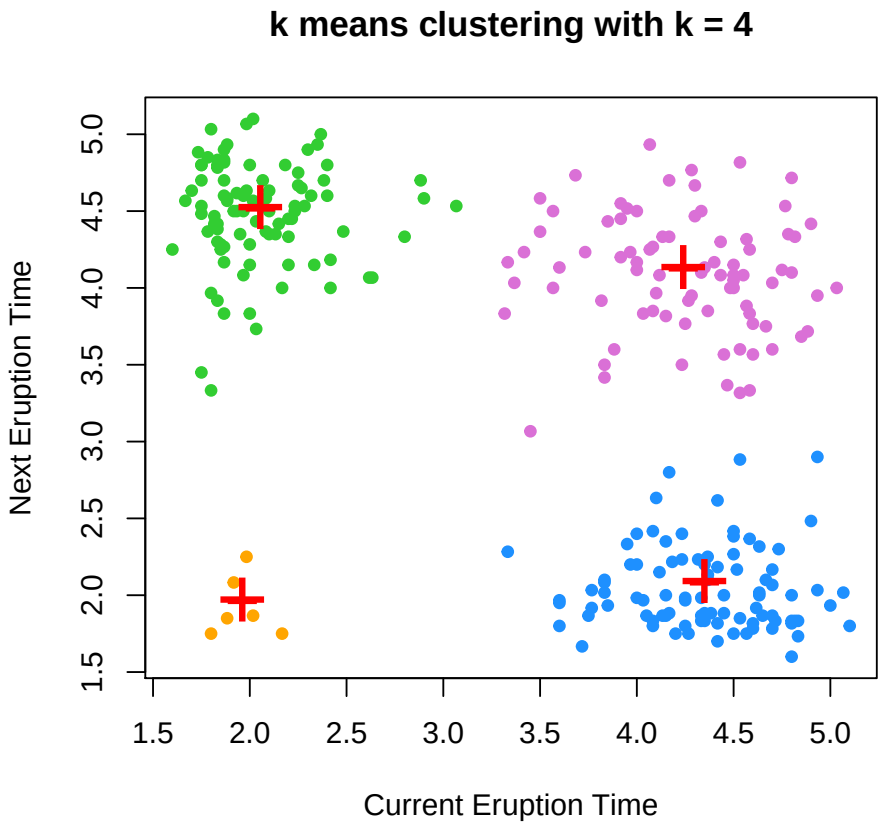
Clustering: Old Faithful Dataset – 2



- The dataset seems to have 4 structures (clusters) mixed together.
- Question: is there any way we can partition data points into the 4 clusters?

k-means Clustering – 1

Here we will introduce a common method: k-means clustering.

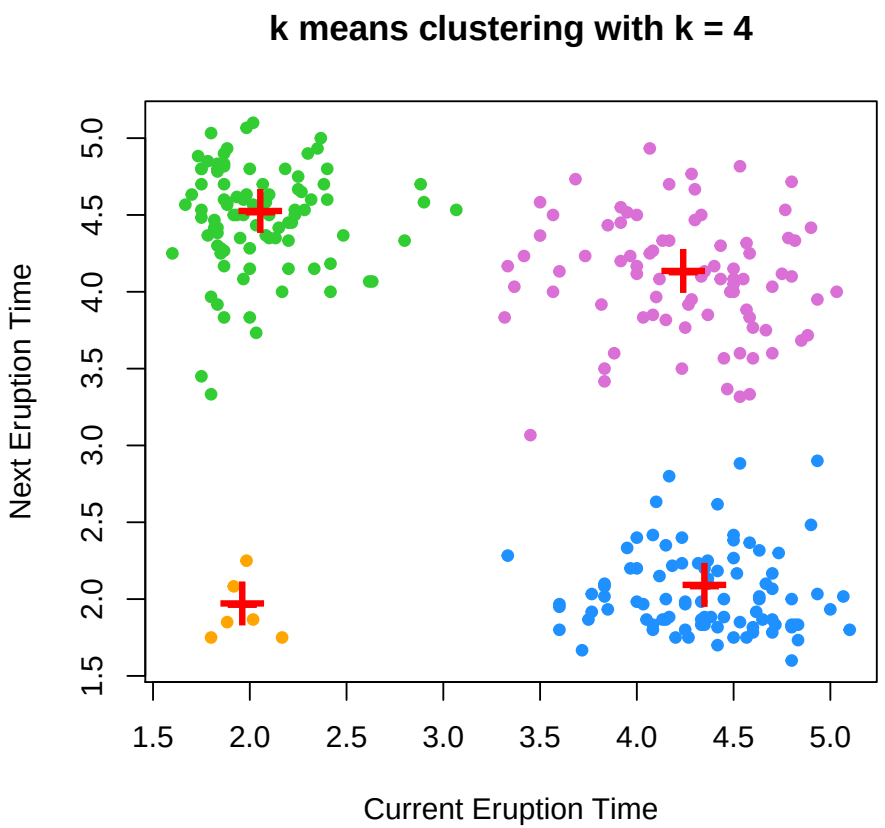


k-means Clustering – 2

- k-means is to partition the data points into k groups.
- The idea is that we find the best k points (these points are called 'centers') such that
  - every data point is assigned to the closest center, and
  - the sum of square of within cluster distance is minimized.
- The sum of square of of within cluster distance is called the k-means objective.
- There is an algorithm for computing the k-means clustering.
- However, this algorithm will stop at a local minimum of the k-means objective.
- So in practice, we need to run the algorithm multiple times and check the within cluster distance to make sure the result is a global minimum.

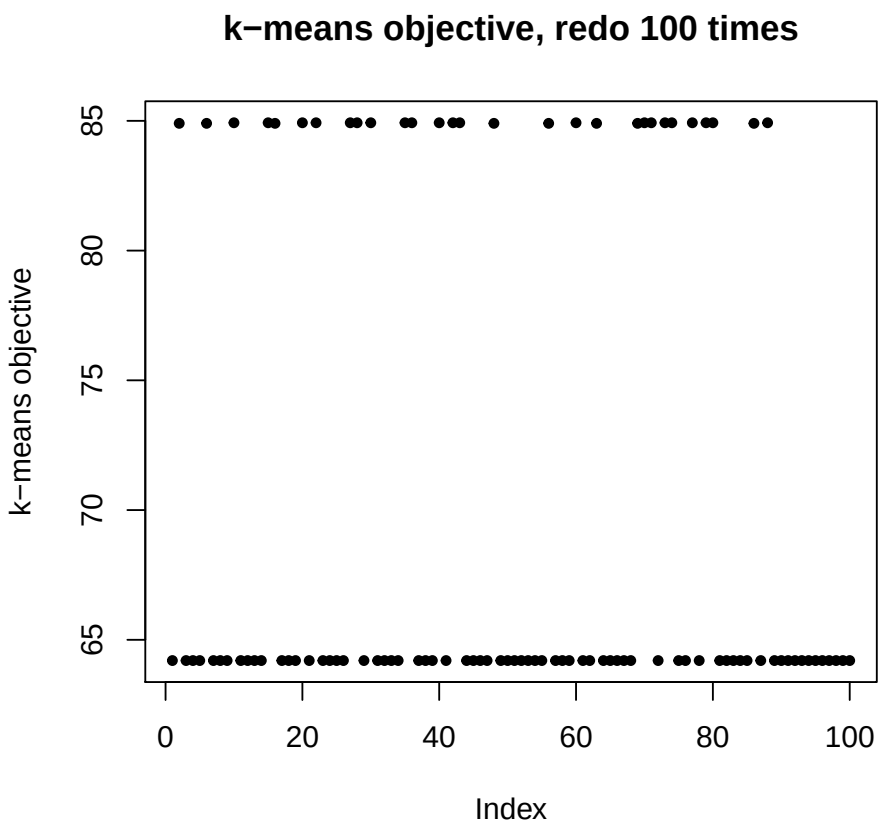
k-means Clustering – 3

```
> data1_km <- kmeans(data1, centers=4)
> col4 <- c("dodgerblue","orchid","limegreen","orange")
> plot(data1, xlab="Current Eruption Time",
+      ylab="Next Eruption Time",
+      col=col4[data1_km$cluster], pch=20, cex=1.2,
+      main="k means clustering with k = 4")
> points(data1_km$centers, col="red",pch="+", cex=3)
```



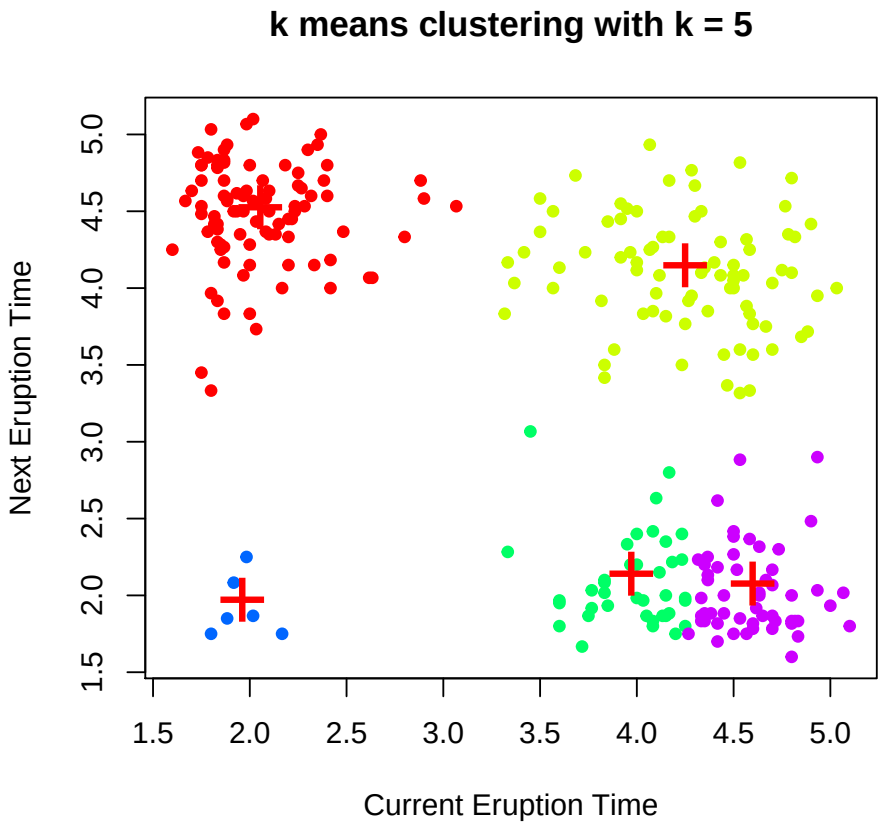
k-means Clustering – 4

```
> k_obj = rep(NA, 100)
> for(w in 1:100){
+   data1_km <- kmeans(data1, centers=4)
+   k_obj[w] <- data1_km$tot.withinss
+ }
> plot(k_obj, pch=20, ylab="k-means objective",
+      main="k-means objective, redo 100 times")
```

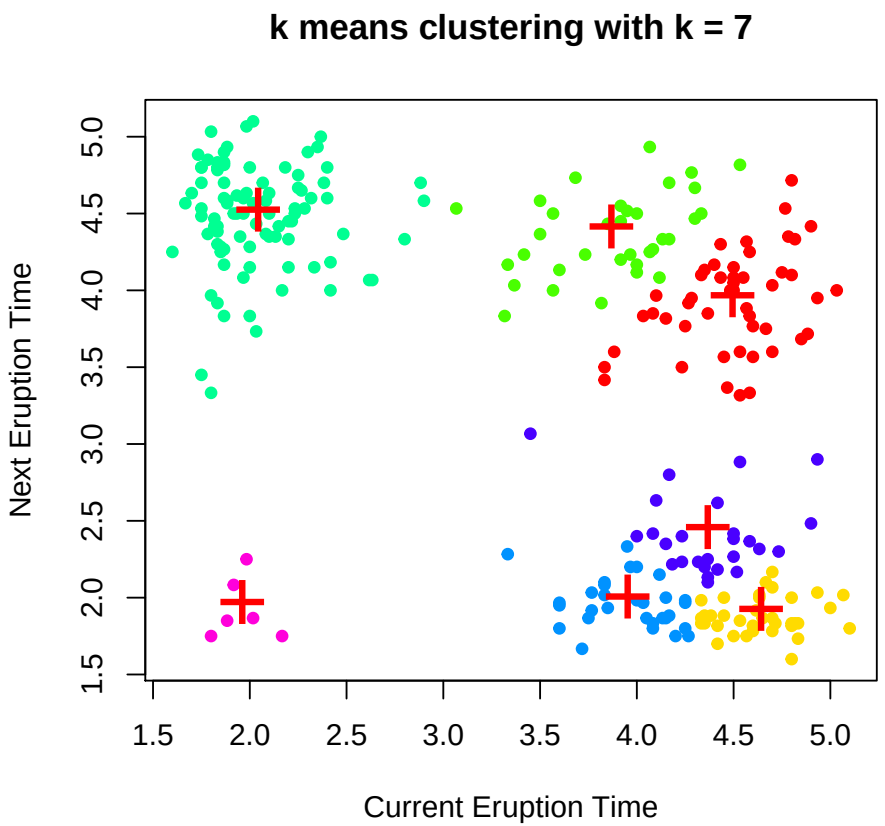


k-means Clustering – 5

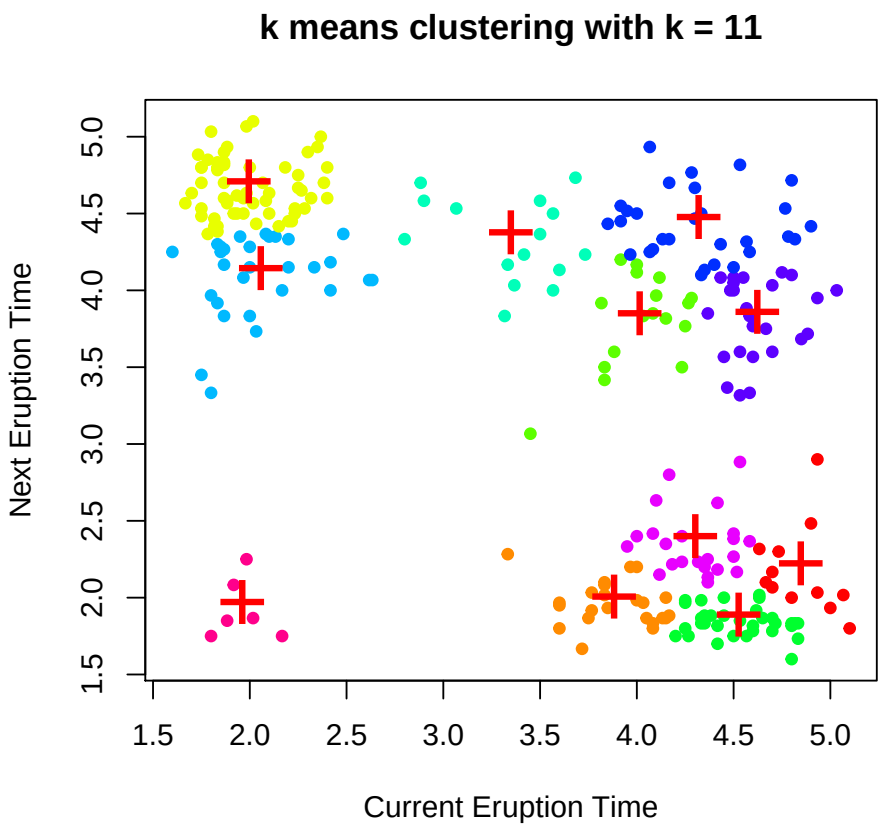
Different k gives you different results.



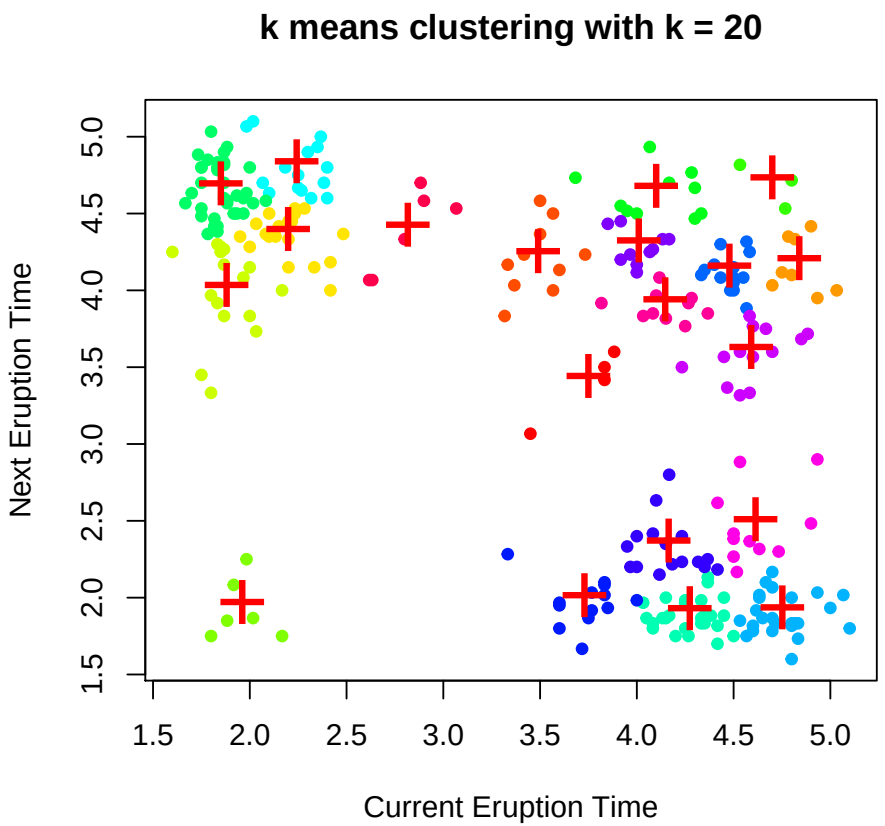
k-means Clustering – 6



k-means Clustering – 7



k-means Clustering – 8



k-means Clustering – 9

- How to choose the number of cluster k is also a hard question.
- Generally we need to look at the data first and the decide it.
- k-means has many applications in compression–we can compress the entire dataset using the k centers to reduce the size of the dataset.
- It is also known as vector quantization.

Clustering: Comments

- Due to time constraint, we only cover k-means clustering.
- There are many many other clustering techniques.
- Hierarchical clustering, spectral clustering, mean shift clustering, ...
- Clustering is still a very popular research topic in statistics and machine learning.
- In scientific or engineering fields, clustering is also a common task.
- In industry, people use clustering to explore the structure of a complex dataset.
- I highly recommend you to learn more about clustering.