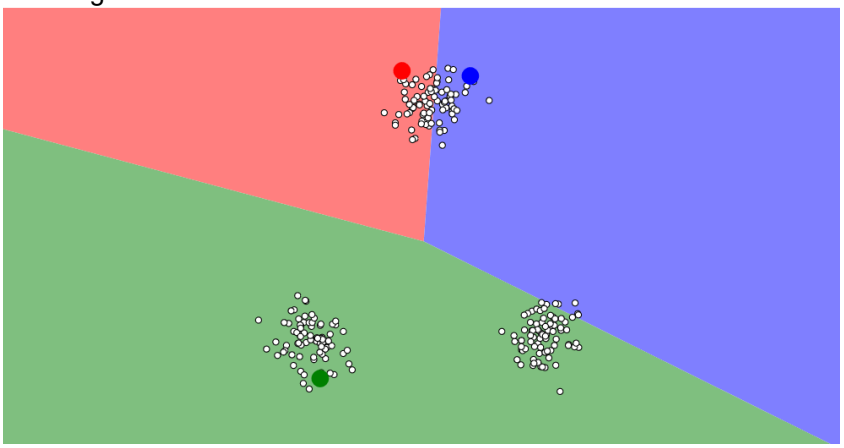
# *k*-means++ Clustering

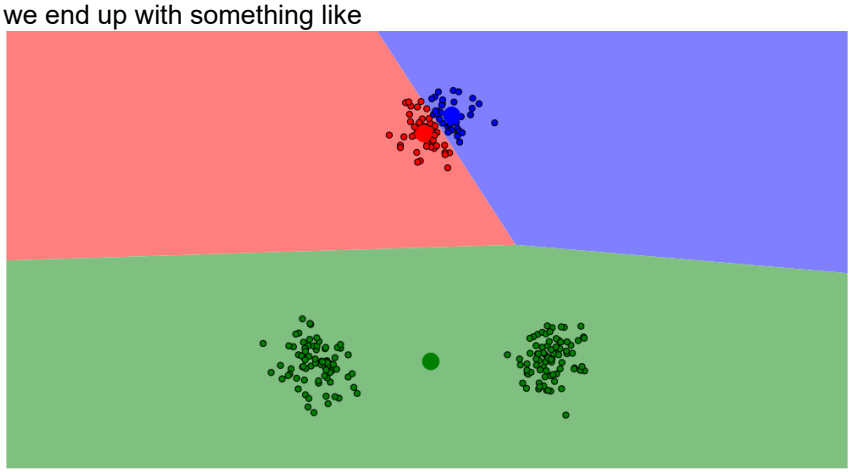
## Theme

## Part 1: *k*-means Animation

- If we e.g. start with



we end up with something like

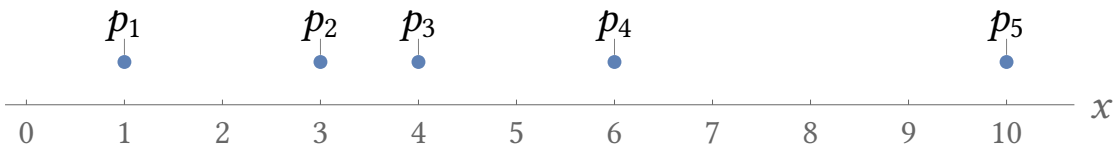


which is not really a good clustering result since the two clusters in the bottom end up in one cluster.

- The problem is that the distance of the bottom point clouds is smaller to the green cluster centre than to one of the top ones. So, we are already in a local minima and further iterations won't improve anything. This means that we are stuck with this bad result solely because of the bad initialization.

# Part 2: *k*-means++ Algorithm

In[13]:=
```
data = {1, 3, 4, 6, 10};
dataLabels = Subscript[it["p"], #] & /@ Range[Length[data]];
ListPlot[MapThread[Callout[{#1, 0.1}, #2, Top] &, {data, dataLabels}],
 PlotTheme → "myTheme",
 AspectRatio → 1 / 10,
 PlotRange → {Automatic, {0, 0.25}},
 Axes → {True, False},
 AxesLabel → {x},
 Ticks → {Range[0, 10], Automatic}
]
```




In[16]:=
```
c₁ = data[[2]]
```

Out[16]= 3

Distances $d_i$

In[17]:=
```
distances = DistanceMatrix[{c₁}, data, DistanceFunction → SquaredEuclideanDistance] // Flatten
```

Out[17]= {4, 0, 1, 9, 49}

Probabilities $P_i$

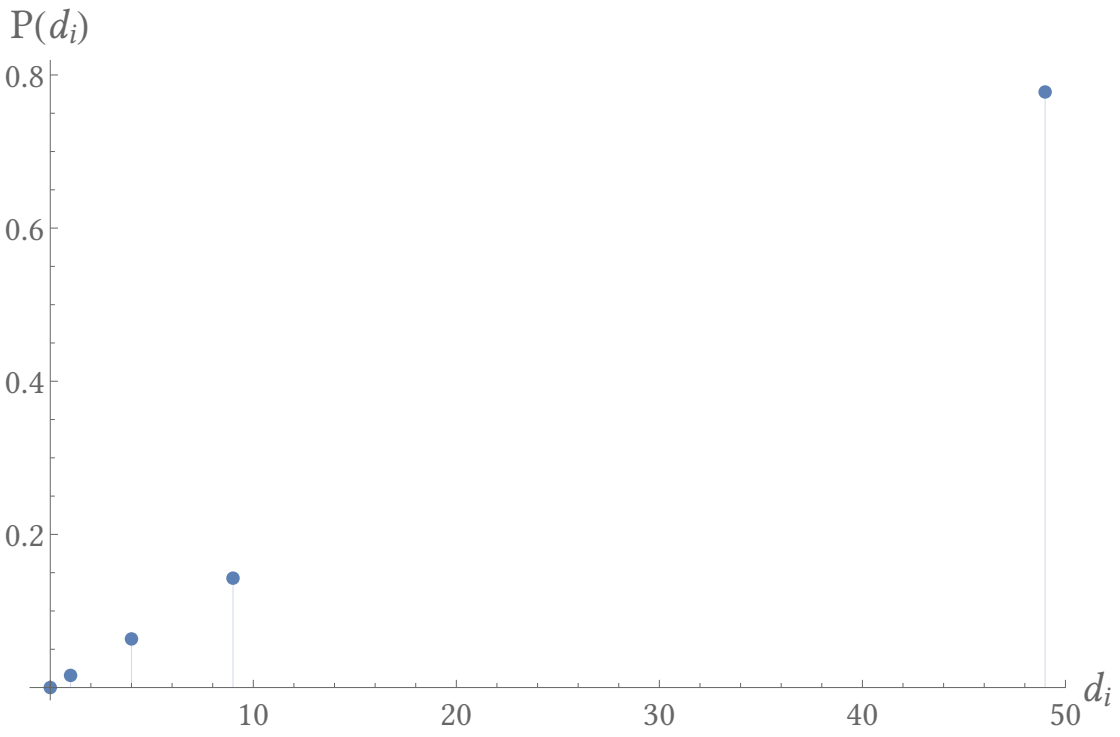
In[18]:=
$$\frac{\text{distances}}{\text{Total[distances]}} \text{ // N}$$

Out[18]= {0.0634921, 0., 0.015873, 0.142857, 0.777778}

And the plot of the discrete probability distribution.

In[19]:=
```
ListPlot[{distances, distances/Total[distances]}^T,
 PlotTheme → "myTheme",
 PlotRange → All,
 AxesLabel → {"d_i", "P(d_i)"},
 Filling → Axis
]
```

Out[19]=




In[20]:=
```
SeedRandom[1337];
c2 = RandomChoice[distances/Total[distances] → data]
```

Out[21]= 10

New distance values.

In[22]:=
```
distances2 =
 MapThread[SquaredEuclideanDistance[#1, #2] &, {Flatten[Nearest[{c1, c2}, data], 1], data}]
```

Out[22]= {4, 0, 1, 9, 0}

And new probability values.

In[23]:=
```
distances2/Total[distances2]
```

Out[23]= $\left\{\frac{2}{7}, 0, \frac{1}{14}, \frac{9}{14}, 0\right\}$

The first data point $p_1$ is selected 4 times, $p_2$ not at all, $p_3$ once, $p_4$ 9 times and $p_5$ again not at all.

---

# Part 3: Video

- https://www.youtube.com/watch?v=BIQDlmZDuf8
- There is no guarantee that always the point with the largest distance is chosen and this is also not intended. Points with higher distances just have a higher probability of being chosen.
- The argument for this might be that we want the algorithm to be robuster against outliers. If we always chose the point with the largest distance, we might end up selecting one outlier after another (outliers tend to be far away).