In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```python
data = pd.read_csv('driver-data.csv')
data.head()
```

Out[2]:

	id	mean_dist_day	mean_over_speed_perc
0	3423311935	71.24	28
1	3423313212	52.53	25
2	3423313724	64.54	27
3	3423311373	55.69	22
4	3423310999	54.58	25

In [3]:
```python
data.describe()
```

Out[3]:

	id	mean_dist_day	mean_over_speed_perc
count	4.000000e+03	4000.000000	4000.000000
mean	3.423312e+09	76.041523	10.721000
std	1.154845e+03	53.469563	13.708543
min	3.423310e+09	15.520000	0.000000
25%	3.423311e+09	45.247500	4.000000
50%	3.423312e+09	53.330000	6.000000
75%	3.423313e+09	65.632500	9.000000
max	3.423314e+09	244.790000	100.000000

In [4]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    4000 non-null   int64  
 1   mean_dist_day         4000 non-null   float64
 2   mean_over_speed_perc  4000 non-null   int64  
dtypes: float64(1), int64(2)
memory usage: 93.9 KB
```

In [5]:
```python
data.columns
```

Out[5]:
```
Index(['id', 'mean_dist_day', 'mean_over_speed_perc'], dtype='object')
```

In [6]:
```python
features = data.drop(['id'], axis = 1)
features.shape
```

Out[6]:
```
(4000, 2)
```

In [7]:
```python
from sklearn.cluster import KMeans
```

In [8]:
```python
my_cluster_model = KMeans(n_clusters=2)
```

In [9]:
```python
my_cluster_model.fit(features)
```

Out[9]:
```
KMeans(n_clusters=2)
```

In [10]:
```python
data['cluster'] = my_cluster_model.labels_
```

In [11]:
```python
data
```

Out[11]:

	id	mean_dist_day	mean_over_speed_perc	cluster
0	3423311935	71.24	28	0
1	3423313212	52.53	25	0
2	3423313724	64.54	27	0
3	3423311373	55.69	22	0
4	3423310999	54.58	25	0
...	...	...	...	...
3995	3423310685	160.04	10	1
3996	3423312600	176.17	5	1
3997	3423312921	170.91	12	1
3998	3423313630	176.14	5	1
3999	3423311533	168.03	9	1

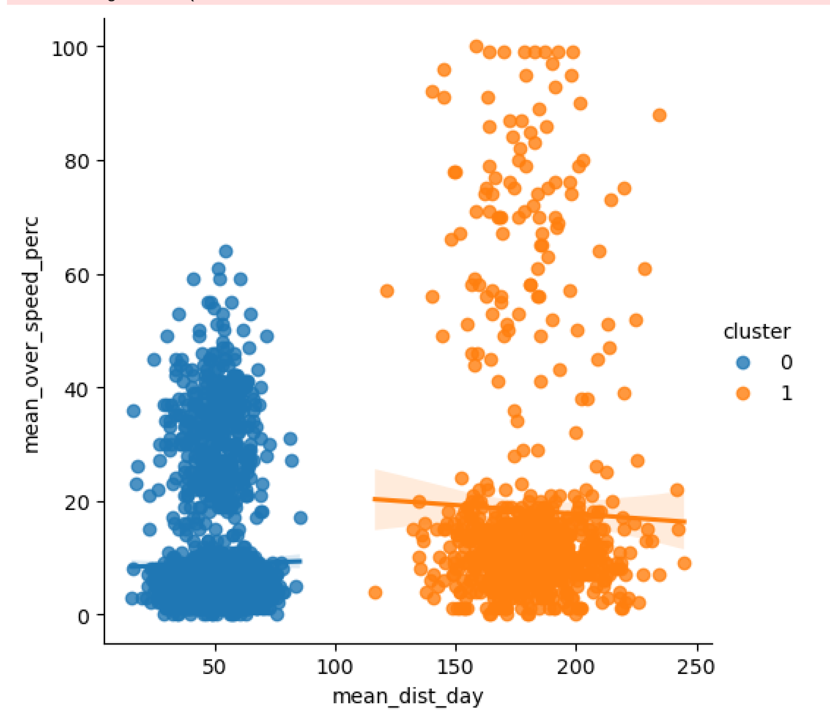
4000 rows × 4 columns

In [12]:
```python
my_cluster_model.cluster_centers_
```

Out[12]:
```
array([[ 50.04763438,   8.82875   ],
       [180.017075  ,  18.29      ]])
```

In [13]:
```python
sns.lmplot('mean_dist_day','mean_over_speed_perc',data=data,hue='cluster');  #seaborn.lmplot() method is used to draw a scatter plot.
plt.show()
```

```
C:\Users\91760\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [14]:
```python
my_cluster_model = KMeans(n_clusters=3)
my_cluster_model.fit(features)
data['cluster'] = my_cluster_model.labels_
sns.lmplot('mean_dist_day','mean_over_speed_perc',data=data,hue='cluster');
```

```
C:\Users\91760\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

In [15]:
```python
my_cluster_model.cluster_centers_
```

Out[15]:
```
array([[ 50.04763438,   8.82875   ],
       [180.34311782,  10.52011494],
       [177.83509615,  70.28846154]])
```

In [16]:
```python
my_cluster_model = KMeans(n_clusters=4)
my_cluster_model.fit(features)
data['cluster'] = my_cluster_model.labels_
sns.lmplot('mean_dist_day','mean_over_speed_perc',data=data,hue='cluster');
```

```
C:\Users\91760\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

In [17]:
```python
my_cluster_model.cluster_centers_
```

Out[17]:
```
array([[ 50.46127059,  32.42823529],
       [180.34311782,  10.52011494],
       [ 49.98428468,   5.21441441],
       [177.83509615,  70.28846154]])
```

In [18]:
```python
my_cluster_model = KMeans(n_clusters=5)
my_cluster_model.fit(features)
data['cluster'] = my_cluster_model.labels_
sns.lmplot('mean_dist_day','mean_over_speed_perc',data=data,hue='cluster');
```

```
C:\Users\91760\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

In [19]:
```python
my_cluster_model.labels_
```

Out[19]:
```
array([4, 4, 4, ..., 1, 1, 1])
```

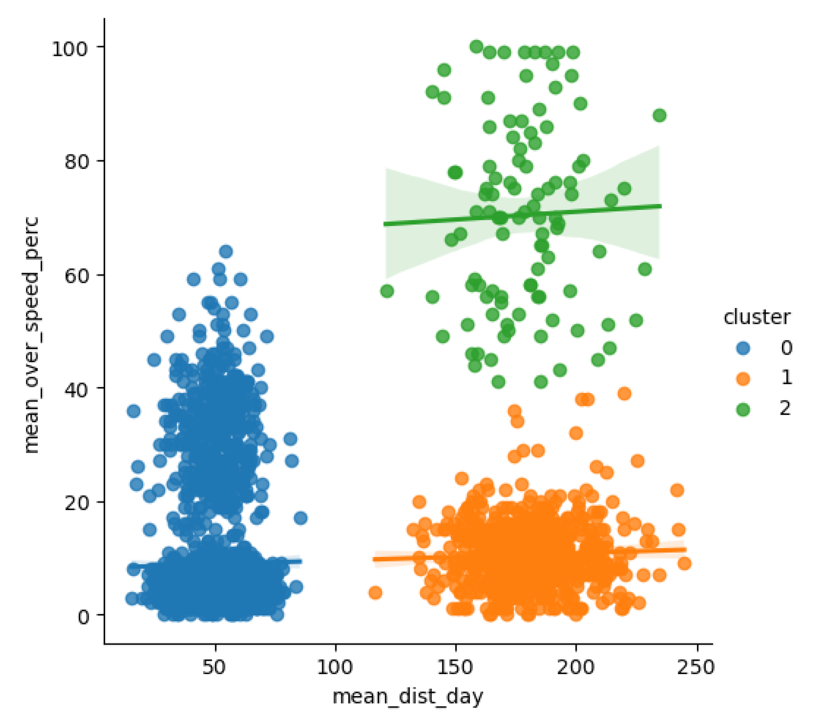
Finding K value

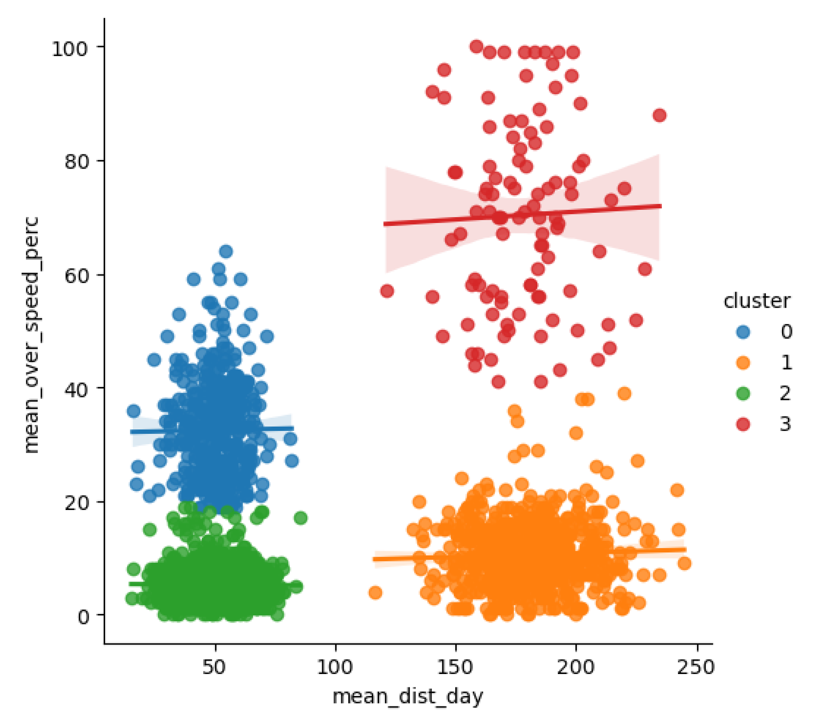
In [20]:
```python
wcss=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,random_state=1)
    kmeans.fit(features)
    wcss.append(kmeans.inertia_)  #the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster.
```

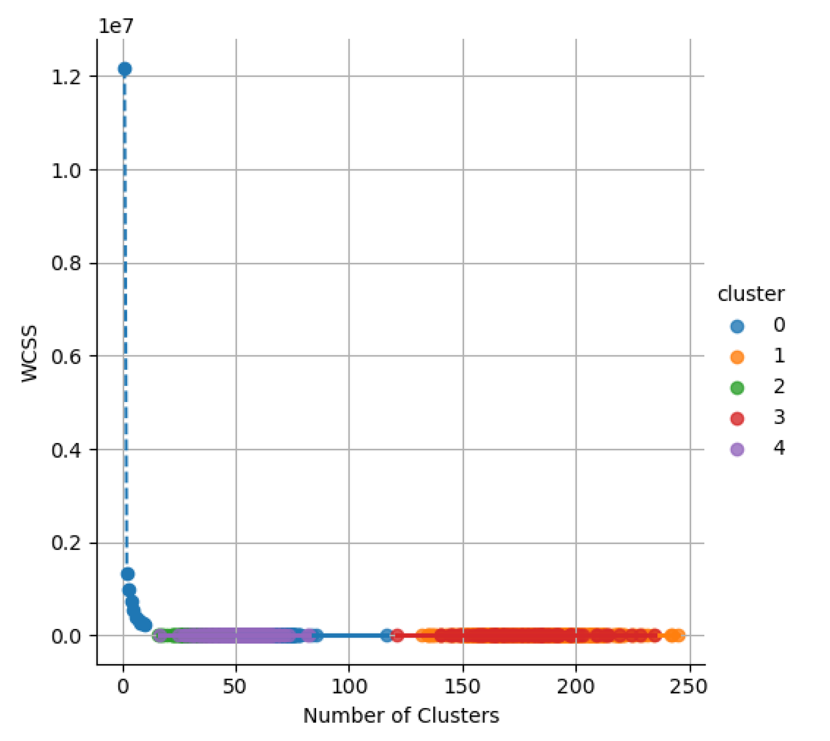
In [21]:
```python
wcss
```

Out[21]:
```
[12184626.129627984,
 1316420.8509477184,
 992634.0606702471,
 719601.1096991899,
 534657.9839435453,
 372837.86302033614,
 319748.1023106628,
 276936.24498565786,
 252830.16872695583,
 230421.39651097317]
```

In [22]:
```python
plt.plot(range(1,11),wcss,"o--")
plt.grid()
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```

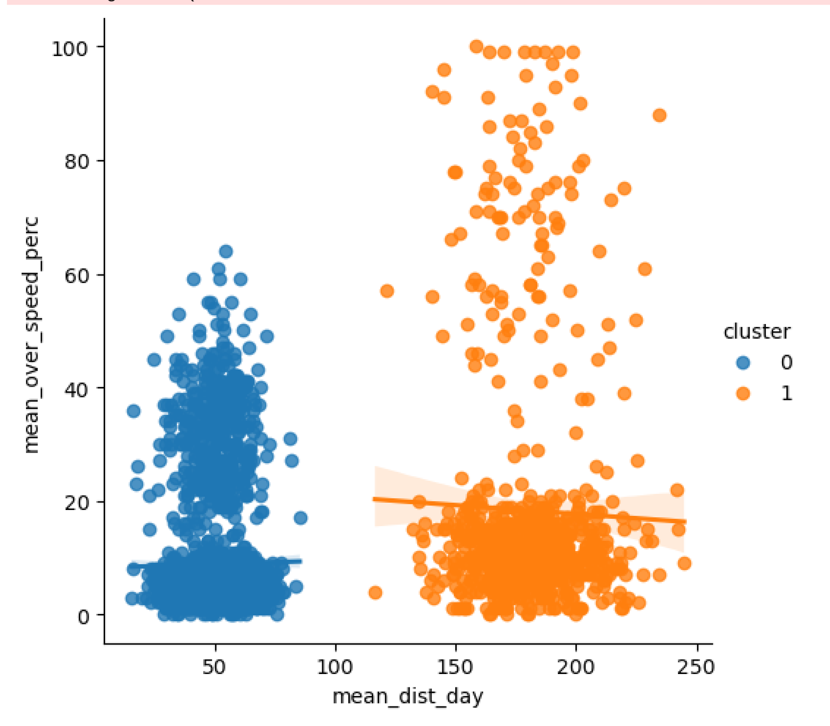






In [23]:
```python
my_cluster_model = KMeans(n_clusters=2)
m = my_cluster_model.fit(features)
data['cluster'] = m.labels_
sns.lmplot('mean_dist_day','mean_over_speed_perc',data=data,hue='cluster');
plt.show()
```

```
C:\Users\91760\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positiona
l argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [24]:
```python
from sklearn.metrics import silhouette_score
# s = (b - a) / max(a, b)
```

In [25]:
```python
range_n_clusters = [2, 3, 4, 5, 6,7,8,9,10]
for n_clusters in range_n_clusters:
    my_cluster_model = KMeans(n_clusters=n_clusters)
    m = my_cluster_model.fit_predict(features)
    silhouette_avg = silhouette_score(features, m)
    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
```

```
For n_clusters = 2 The average silhouette_score is : 0.8490223286225532
For n_clusters = 3 The average silhouette_score is : 0.8231396834167266
For n_clusters = 4 The average silhouette_score is : 0.5907475009381601
For n_clusters = 5 The average silhouette_score is : 0.5126643886251191
For n_clusters = 6 The average silhouette_score is : 0.48574775768218625
For n_clusters = 7 The average silhouette_score is : 0.458532906359744
For n_clusters = 8 The average silhouette_score is : 0.4490407137981559
For n_clusters = 9 The average silhouette_score is : 0.4461317576094622
For n_clusters = 10 The average silhouette_score is : 0.43462210471932766
```

In [ ]:

In [ ]: