In [235…

**import** pandas **as** pd **import** numpy **as** np **import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**from** scipy **import** stats

**from** sklearn.cluster **import** KMeans, AgglomerativeClustering, DBSCAN, AffinityPropagation

**from** sklearn.metrics **import** silhouette\_score

**import** plotly **as** py

**import** plotly.graph\_objs **as** go

**import** scipy.cluster.hierarchy **as** sch

**from** itertools **import** product

**from** scipy.stats **import** pearsonr

**from** mpl\_toolkits.mplot3d **import** Axes3D **from** mpl\_toolkits.mplot3d **import** Axes3D **import** plotly **as** py

**import** plotly.graph\_objs **as** go

In [6]:

da\_ta **=** pd**.**read\_csv('Mall\_Customers.csv')*#importing the da\_ta set*

da\_ta**.**head()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[6]: |  | **CustomerID** | **Gender** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
|  | **0** | 1 | Male | 19 | 15 | 39 |
|  | **1** | 2 | Male | 21 | 15 | 81 |
|  | **2** | 3 | Female | 20 | 16 | 6 |
|  | **3** | 4 | Female | 23 | 16 | 77 |
|  | **4** | 5 | Female | 31 | 17 | 40 |

In [7]:

print(da\_ta**.**shape)

(200, 5)

In [4]:

da\_ta**.**describe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[4]: | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
|  | **count** 200.000000 | 200.000000 | 200.000000 | 200.000000 |
|  | **mean** 100.500000 | 38.850000 | 60.560000 | 50.200000 |
|  | **std** 57.879185 | 13.969007 | 26.264721 | 25.823522 |
|  | **min** 1.000000 | 18.000000 | 15.000000 | 1.000000 |
|  | **25%** 50.750000 | 28.750000 | 41.500000 | 34.750000 |
|  | **50%** 100.500000 | 36.000000 | 61.500000 | 50.000000 |
|  | **75%** 150.250000 | 49.000000 | 78.000000 | 73.000000 |
|  | **max** 200.000000 | 70.000000 | 137.000000 | 99.000000 |

In [5]:

Out[5]:

CustomerID 0

da\_ta**.**isnull()**.**sum()

Gender 0

Age 0

Annual Income (k$) 0

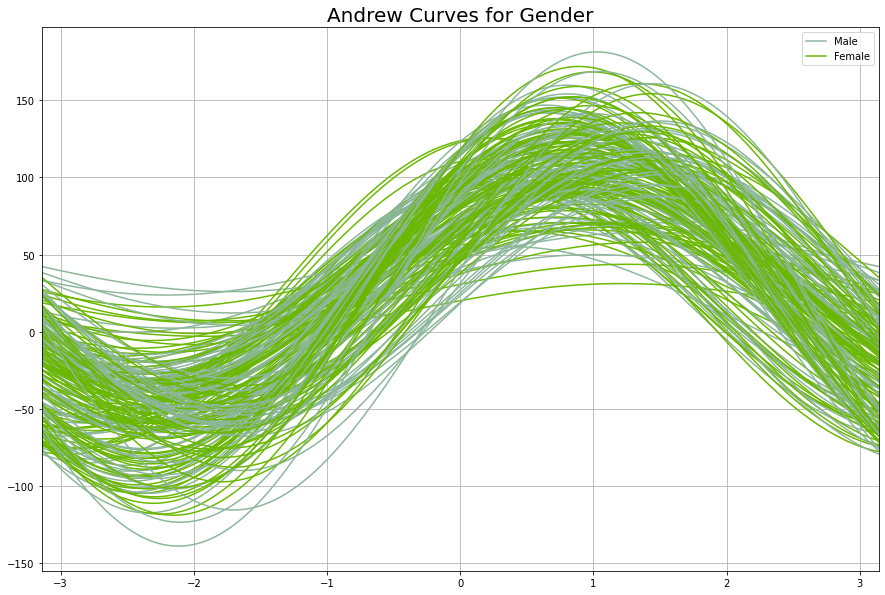
Spending Score (1-100) 0

dtype: int64

In [218…

plt**.**rcParams['figure.figsize'] **=** (15, 10) pd**.**plotting**.**andrews\_curves(da\_ta**.**drop("CustomerID", axis**=**1), "Gender") plt**.**title('Andrew Curves for Gender', fontsize **=** 20)

plt**.**show()



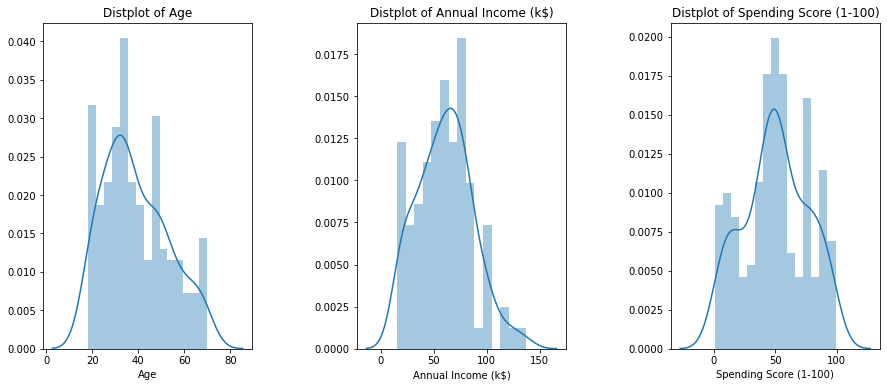
In [9]:

plt**.**figure(1 , figsize **=** (15 , 6)) number **=** 0

**for** x **in** number ['Age' , 'Anumbernumberual Inumbercome (k$)' , 'Spenumberdinumberg Score number **+=** 1

plt**.**subplot(1 , 3 , number) plt**.**subplots\_adjust(hspace **=** 0.5 , wspace **=** 0.5) snumbers**.**distplot(da\_ta[x] , binumbers **=** 15) plt**.**title('Distplot of {}'**.**format(x))

plt**.**show()



In [ ]:

In [10]:

Out[10]:

In [13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x23f7f4d8c88>

sns**.**countplot(y **=** 'Gender' , da\_ta **=** da\_ta)



*#Subsetting the dataframes for different gender groups* ma\_da\_ta\_le **=** data[data['Gender'] **==** 'Male'] fe\_da\_ta\_ma\_le **=** data[data['Gender'] **==** 'Female']

*#Explore the Age , Income and Spending score for Males*

plt**.**figure(1 , figsize **=** (15 , 6)) n **=** 0

**for** x **in** ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']: n **+=** 1

plt**.**subplot(1 , 3 , n) plt**.**subplots\_adjust(hspace **=** 0.5 , wspace **=** 0.5) sns**.**distplot(ma\_da\_ta\_le[x] , bins **=** 20) plt**.**title('Distplot of male {}'**.**format(x))

plt**.**show()

*#Explore the Age, Income and Spending score of Females*

plt**.**figure(1 , figsize **=** (15 , 6)) n **=** 0

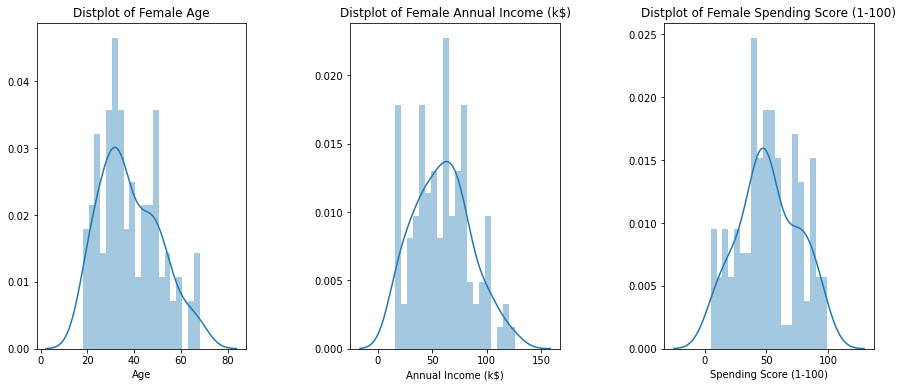
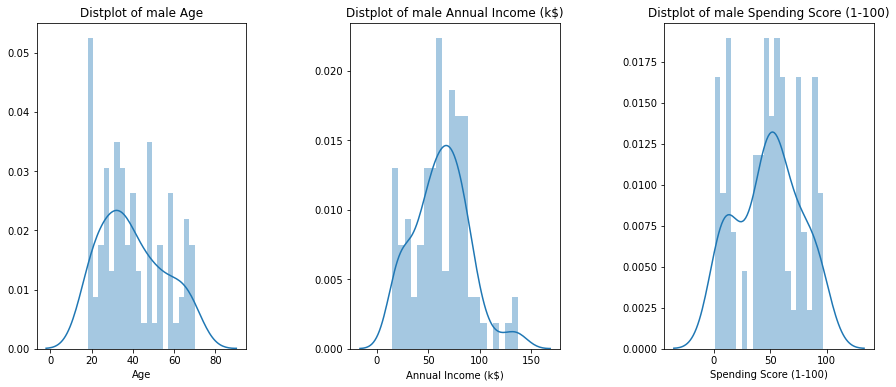
**for** x **in** ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:

n **+=** 1

plt**.**subplot(1 , 3 , n)

plt**.**subplots\_adjust(hspace **=** 0.5 , wspace **=** 0.5) sns**.**distplot(fe\_da\_ta\_ma\_le[x] , bins **=** 20) plt**.**title('Distplot of Female {}'**.**format(x))

plt**.**show()



In [18]:

*#Investigate different age groups for males and females*

age\_da\_male\_ta **=** da\_ta[da\_ta['Gender']**==**'Male']['Age'] *# subset with males age*

fe\_age\_da\_male\_ta **=** da\_ta[da\_ta['Gender']**==**'Female']['Age'] *# subset with females age*

In [24]:

age\_bins **=** range(15,75,5)

*# males histogram*

fig2, (ax1, ax2) **=** plt**.**subplots(1, 2, figsize**=**(12,5), sharey**=True**) sns**.**distplot(males\_age, bins**=**age\_bins, kde**=False**, color**=**'#0066ff', ax**=**ax1, hist\_kws**=**dict( ax1**.**set\_xticks(age\_bins)

ax1**.**set\_ylim(top**=**25) ax1**.**set\_title('Males') ax1**.**set\_ylabel('Count')

ax1**.**text(45,23, "TOTAL count: {}"**.**format(age\_da\_male\_ta**.**count())) ax1**.**text(45,22, "Mean age: {:.1f}"**.**format(age\_da\_male\_ta**.**mean()))

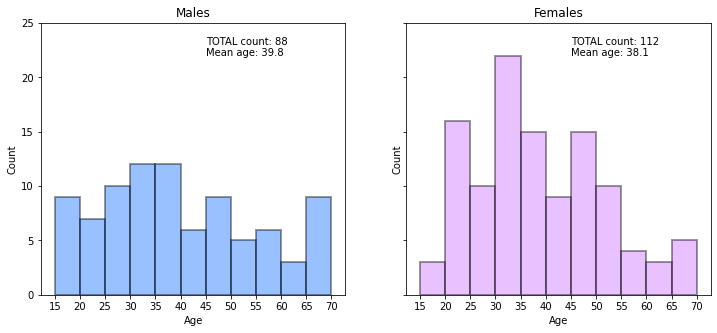
*# females histogram*

sns**.**distplot(females\_age, bins**=**age\_bins, kde**=False**, color**=**'#cc66ff', ax**=**ax2, hist\_kws**=**dic ax2**.**set\_xticks(age\_bins)

ax2**.**set\_ylim(top**=**25) ax2**.**set\_title('Females') ax2**.**set\_ylabel('Count')

ax2**.**text(45,23, "TOTAL count: {}"**.**format(fe\_age\_da\_male\_ta**.**count())) ax2**.**text(45,22, "Mean age: {:.1f}"**.**format(fe\_age\_da\_male\_ta**.**mean()))

plt**.**show()



In [29]:

Out[29]:

*#finding the maximum Expendicture with respect to GENDER and AGE*

spend\_by\_male **=** da\_ta[da\_ta['Gender']**==**'Male']['Spending Score (1-100)'] *# subset with ma*

fe\_spend\_by\_male **=** da\_ta[da\_ta['Gender']**==**'Female']['Spending Score (1-100)'] *# subset wi*

spending\_bin **=** range(0,105,5)

*# males histogram*

fig2, (ax1, ax2) **=** plt**.**subplots(1, 2, figsize**=**(18,5), sharey**=True**) sns**.**distplot(spend\_by\_male, bins**=**spending\_bin, kde**=False**, color**=**'#0066ff', ax**=**ax1, hist\_k ax1**.**set\_xticks(spending\_bin)

ax1**.**set\_ylim(top**=**25) ax1**.**set\_title('Males') ax1**.**set\_ylabel('Count')

ax1**.**text(50,15, "Mean spending score: {:.1f}"**.**format(spend\_by\_male**.**mean())) ax1**.**text(50,14, "Median spending score: {:.1f}"**.**format(spend\_by\_male**.**median())) ax1**.**text(50,13, "Std. deviation score: {:.1f}"**.**format(spend\_by\_male**.**std()))

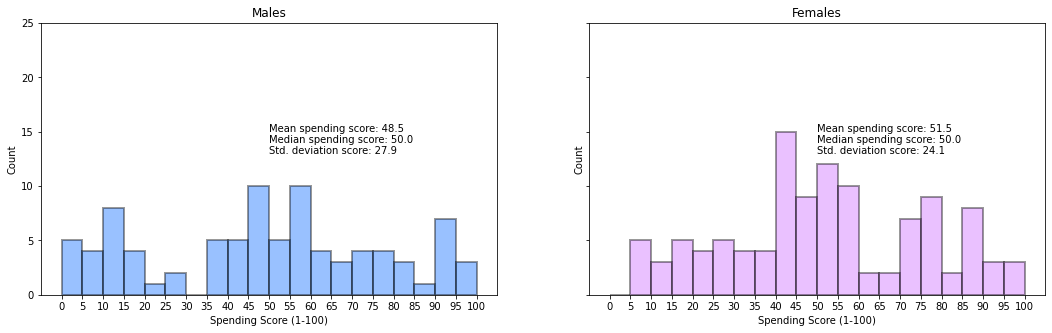
*# females histogram*

sns**.**distplot(fe\_spend\_by\_male, bins**=**spending\_bin, kde**=False**, color**=**'#cc66ff', ax**=**ax2, his ax2**.**set\_xticks(spending\_bin)

ax2**.**set\_ylim(top**=**25) ax2**.**set\_title('Females') ax2**.**set\_ylabel('Count')

ax2**.**text(50,15, "Mean spending score: {:.1f}"**.**format(fe\_spend\_by\_male**.**mean())) ax2**.**text(50,14, "Median spending score: {:.1f}"**.**format(fe\_spend\_by\_male**.**median())) ax2**.**text(50,13, "Std. deviation score: {:.1f}"**.**format(fe\_spend\_by\_male**.**std()))

Text(50, 13, 'Std. deviation score: 24.1')



In [40]:

Age\_median **=** da\_ta**.**groupby(["Gender",pd**.**cut(da\_ta['Age'], age\_bins)])**.**median() Age\_median**.**index **=** Age\_median**.**index**.**set\_names(['Gender', 'Age\_group']) Age\_median**.**reset\_index(inplace**=True**)

In [41]:

Age\_median**.**head(10)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[41]: | **Gender** | **Age\_group** | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
|  | **0** Female | (15, 20] | 112.0 | 19.0 | 63.0 | 50.0 |
|  | **1** Female | (20, 25] | 46.0 | 23.0 | 39.0 | 65.0 |
|  | **2** Female | (25, 30] | 139.5 | 29.0 | 74.5 | 73.0 |
|  | **3** Female | (30, 35] | 126.0 | 32.0 | 70.0 | 69.0 |
|  | **4** Female | (35, 40] | 138.5 | 38.0 | 73.5 | 41.0 |
|  | **5** Female | (40, 45] | 137.0 | 44.0 | 73.0 | 28.0 |
|  | **6** Female | (45, 50] | 82.0 | 49.0 | 54.0 | 44.5 |
|  | **7** Female | (50, 55] | 87.0 | 54.0 | 57.0 | 43.0 |
|  | **8** Female | (55, 60] | 74.0 | 58.0 | 50.0 | 35.0 |
|  | **9** Female | (60, 65] | 79.0 | 64.0 | 51.5 | 39.0 |

In [42]:

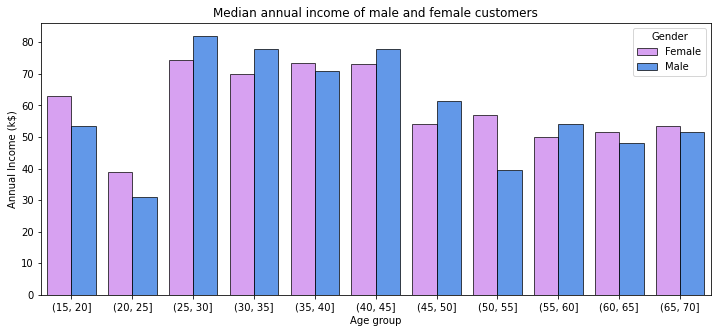
fig, ax **=** plt**.**subplots(figsize**=**(12,5))

sns**.**barplot(x**=**'Age\_group', y**=**'Annual Income (k$)', hue**=**'Gender', data**=**Age\_median, palette**=**['#cc66ff','#0066ff'],

alpha**=**0.7,edgecolor**=**'k', ax**=**ax)

ax**.**set\_title('Median annual income of male and female customers') ax**.**set\_xlabel('Age group')

plt**.**show()



In [43]:

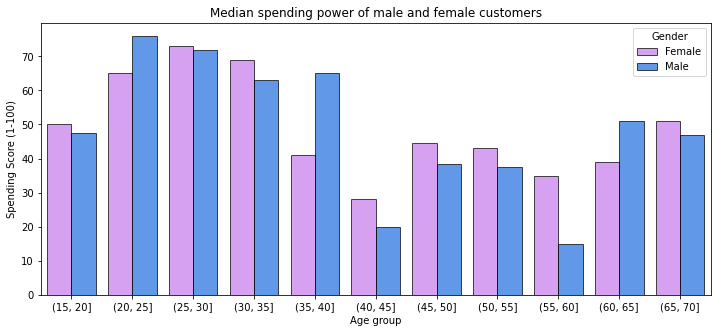
fig, ax **=** plt**.**subplots(figsize**=**(12,5))

sns**.**barplot(x**=**'Age\_group', y**=**'Spending Score (1-100)', hue**=**'Gender', data**=**Age\_median, palette**=**['#cc66ff','#0066ff'],

alpha**=**0.7,edgecolor**=**'k', ax**=**ax)

ax**.**set\_title('Median spending power of male and female customers') ax**.**set\_xlabel('Age group')

plt**.**show()



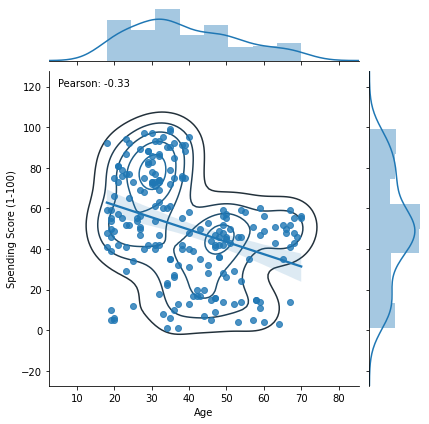
In [58]:

*# calculating Pearson's correlation*

corr, \_ **=** pearsonr(da\_ta['Age'], da\_ta['Spending Score (1-100)'])

joint\_ploting **=** (sns**.**jointplot('Age', 'Spending Score (1-100)', data**=**da\_ta, kind**=**'reg'))**.**plot\_joint(sns**.**kdeplot, zorder**=**0, n\_levels**=**6)

plt**.**text(5,120, 'Pearson: {:.2f}'**.**format(corr)) plt**.**show()



In [60]:

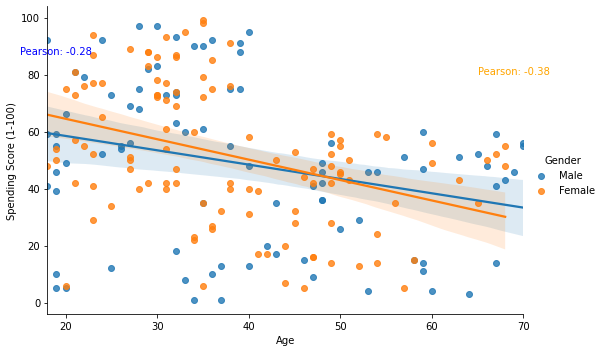
*# calculating Pearson's correlation betweem ,age groups and spending power*

corr1, \_ **=** pearsonr(age\_da\_male\_ta**.**values, spend\_by\_male**.**values) corr2, \_ **=** pearsonr(fe\_age\_da\_male\_ta**.**values, fe\_spend\_by\_male**.**values)

sns**.**lmplot('Age', 'Spending Score (1-100)' , data**=**da\_ta, hue**=**'Gender', aspect**=**1.5)

plt**.**text(15,87, 'Pearson: {:.2f}'**.**format(corr1), color**=**'blue') plt**.**text(65,80, 'Pearson: {:.2f}'**.**format(corr2), color**=**'orange')

plt**.**show()



# K-MEANS

In [209…

*#Trying K-Means Clustering for AGE AND SPENDING SCORE TO SEE THE VARIATION..*

X1 **=** data[['Age' , 'Spending Score (1-100)']]**.**iloc[: , :]**.**values inter **=** []

point\_s **=** []

**for** n **in** range(2 , 11):

algorithm **=** (KMeans(n\_clusters **=** n ,init**=**'k-means++', n\_init **=** 10 ,max\_iter**=**300, tol**=**0.0001, random\_state**=** 111 , algorithm**=**'elkan') )

algorithm**.**fit(X1) inter**.**append(algorithm**.**inter\_)

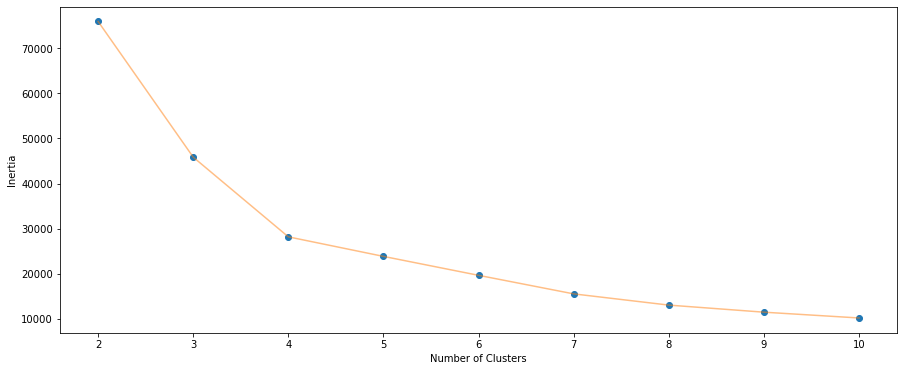
silhouette\_avg **=** silhouette\_score(X1, algorithm**.**labels\_) point\_s**.**append(silhouette\_avg) *# data for the silhouette score method*

In [211…

*#The Elbow Chart*

plt**.**figure(1 , figsize **=** (15 ,6)) plt**.**plot(np**.**arange(2 , 11) , inter , 'o')

plt**.**plot(np**.**arange(2 , 11) , inter , '-' , alpha **=** 0.5) plt**.**xlabel('Number of Clusters') , plt**.**ylabel('inter') plt**.**show()



# Similarly we can go for the combinations of other variables

In [79]:

X **=** da\_ta[['Age' ,'Annual Income (k$)' ,'Spending Score (1-100)']] inter\_2 **=** []

point\_s\_2 **=** []

**for** n **in** range(2 , 11):

algorithm **=** KMeans(n\_clusters **=** n ,init**=**'k-means++', n\_init **=** 10 ,max\_iter**=**300, tol**=**0.0001, random\_state**=** 111 , algorithm**=**'elkan')**.**fit(X)

inter\_2**.**append(algorithm**.**inter\_2\_)

silhouette\_avg **=** silhouette\_score(X, algorithm**.**labels\_) point\_s\_2**.**append(silhouette\_avg) *# data for the silhouette score method*

In [81]:

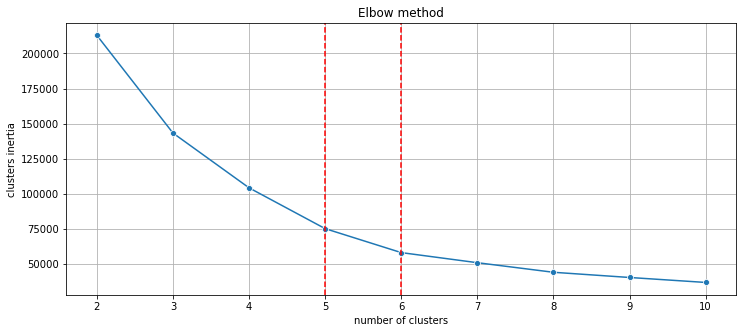
fig, ax **=** plt**.**subplots(figsize**=**(12,5))

ax **=** sns**.**lineplot(np**.**arange(2 , 11), inertia, marker**=**'o', ax**=**ax) ax**.**set\_title("Elbow method")

ax**.**set\_xlabel("number of clusters") ax**.**set\_ylabel("clusters inertia") ax**.**axvline(5, ls**=**"--", c**=**"red")

ax**.**axvline(6, ls**=**"--", c**=**"red") plt**.**grid()

plt**.**show()



# A choice of 5 or 6 clusters seems to be fair. Let's see the silhouette score.

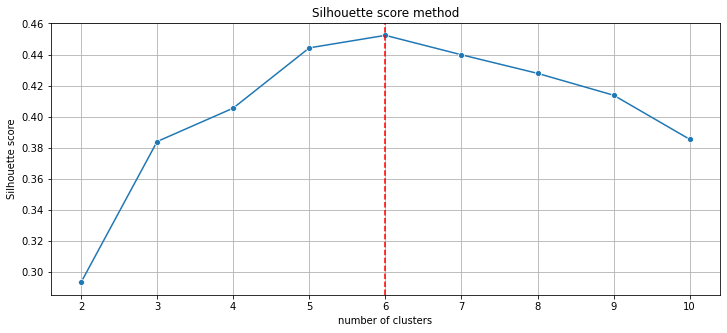
In [82]:

fig, ax **=** plt**.**subplots(figsize**=**(12,5))

ax **=** sns**.**lineplot(np**.**arange(2 , 11), s\_scores, marker**=**'o', ax**=**ax) ax**.**set\_title("Silhouette score method")

ax**.**set\_xlabel("number of clusters") ax**.**set\_ylabel("Silhouette score") ax**.**axvline(6, ls**=**"--", c**=**"red") plt**.**grid()

plt**.**show()



# Silhouette score method indicates the best options would be respectively 6 or 5 clusters. Let's compare both.

In [91]:

*#For clusters of K=6*

K\_means\_6 **=** (KMeans(n\_clusters **=** 6 ,init**=**'k-means++', n\_init **=** 10 ,max\_iter**=**300,

tol**=**0.0001, random\_state**=** 111 , algorithm**=**'elkan') )

K\_Means\_6**.**fit(X)

labels\_6 **=** K\_Means\_6**.**labels\_

center\_6 **=** K\_Means\_6**.**cluster\_centers\_ KM6\_df **=** da\_ta**.**copy() KM6\_df['labels'] **=** labels6

In [94]:

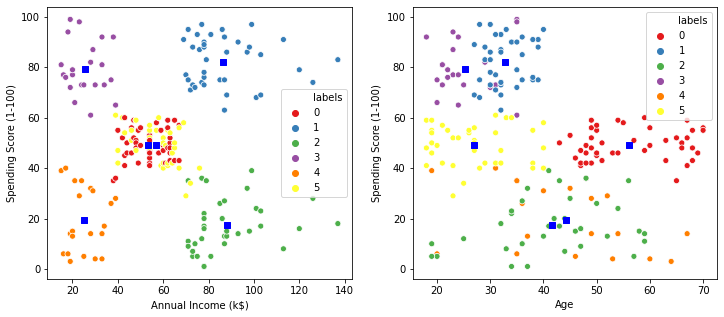
fig1, (axes) **=** plt**.**subplots(1,2,figsize**=**(12,5))

scat\_1 **=** sns**.**scatterplot('Annual Income (k$)', 'Spending Score (1-100)', data**=**KM6\_df, hue**=**'labels', ax**=**axes[0], palette**=**'Set1', legend**=**'full')

sns**.**scatterplot('Age', 'Spending Score (1-100)', data**=**K\_Means\_6\_df, hue**=**'labels', palette**=**'Set1', ax**=**axes[1], legend**=**'full')

axes[0]**.**scatter(center\_6[:,1],center\_6[:,2], marker**=**'s', s**=**40, c**=**"blue")

axes[1]**.**scatter(center\_6[:,0],center\_6[:,2], marker**=**'s', s**=**40, c**=**"blue") plt**.**show()



# K-Means algorithm generated the following 6 clusters:

younger clients with **medium** annual and **medium** spending score clients with **high** annual income and **low** spending score

younger clients with **medium** annual and **medium** spending score clients with **high** annual income and **high** spending score

# clients with **low** annual income and **low** spending score clients with **low** annual income and **high** spending score

In [100…

K\_Means\_clustering\_sizes **=** K\_Means\_6\_df**.**groupby('labels')**.**size()**.**to\_frame() K\_Means\_clustering\_sizes**.**columns **=** ["KM\_size"]

K\_Means\_clustering\_sizes

|  |  |  |
| --- | --- | --- |
| Out[100… | **labels** | **KM\_size** |
|  | **0** | 45 |
|  | **1** | 39 |
|  | **2** | 35 |
|  | **3** | 22 |
|  | **4** | 21 |
|  | **5** | 38 |

In [104…

In [110…

*#For clusters of K=5*

K\_Means\_5 **=** (KMeans(n\_clusters **=** 5 ,init**=**'k-means++', n\_init **=** 10 ,max\_iter**=**300,

tol**=**0.0001, random\_state**=** 111 , algorithm**=**'elkan') )

K\_Means\_5**.**fit(X)

labels\_5 **=** K\_Means\_5**.**labels\_

center\_5 **=** K\_Means\_5**.**cluster\_centers\_ K\_Means\_5\_df **=** da\_ta**.**copy() K\_Means\_5\_df['labels'] **=** labels\_5

fig **=** plt**.**figure(figsize**=**(7, 7))

ax **=** Axes3D(fig, rect**=**[0, 0, .99, 1], elev**=**20, azim**=**210) ax**.**scatter(K\_Means\_6\_df['Age'],

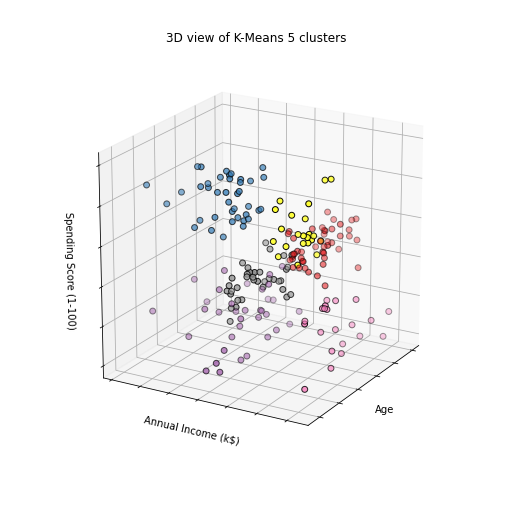
K\_Means\_6\_df['Annual Income (k$)'], K\_Means\_6\_df['Spending Score (1-100)'], c**=**K\_Means\_6\_df['labels'],

s**=**35, edgecolor**=**'k', cmap**=**plt**.**cm**.**Set1)

ax**.**w\_xaxis**.**set\_ticklabels([]) ax**.**w\_yaxis**.**set\_ticklabels([]) ax**.**w\_zaxis**.**set\_ticklabels([]) ax**.**set\_xlabel('Age') ax**.**set\_ylabel('Annual Income (k$)') ax**.**set\_zlabel('Spending Score (1-100)')

ax**.**set\_title('3D view of K-Means 5 clusters') ax**.**dist **=** 12

plt**.**show()



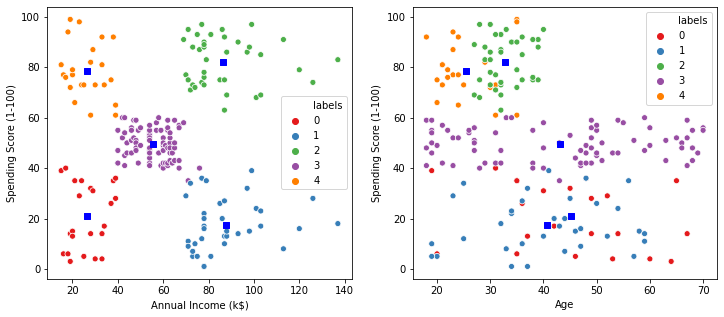
In [111…

fig1, (axes) **=** plt**.**subplots(1,2,figsize**=**(12,5))

scat\_1 **=** sns**.**scatterplot('Annual Income (k$)', 'Spending Score (1-100)', data**=**KM5\_df, hue**=**'labels', ax**=**axes[0], palette**=**'Set1', legend**=**'full')

sns**.**scatterplot('Age', 'Spending Score (1-100)', data**=**K\_Means\_5\_df, hue**=**'labels', palette**=**'Set1', ax**=**axes[1], legend**=**'full')

axes[0]**.**scatter(center\_5[:,1],center\_5[:,2], marker**=**'s', s**=**40, c**=**"blue")

axes[1]**.**scatter(center\_5[:,0],center\_5[:,2], marker**=**'s', s**=**40, c**=**"blue") plt**.**show()

# K-Means algorithm generated the following 5 clusters: clients with **low** annual income and **high** spending score

clients with **medium** annual income and **medium** spending score clients with **high** annual income and **low** spending score

# clients with **high** annual income and **high** spending score clients with **low** annual income and **low** spending score

There are no distinct groups is terms of customers age.

In [112…

K\_Means\_clust\_sizes5 **=** KM5\_df**.**groupby('labels')**.**size()**.**to\_frame() K\_Means\_clust\_sizes5**.**columns **=** ["KM\_size"]

K\_Means\_clust\_sizes5

|  |  |  |
| --- | --- | --- |
| Out[112… | **labels** | **KM\_size** |
|  | **0** | 23 |
|  | **1** | 36 |
|  | **2** | 39 |
|  | **3** | 79 |
|  | **4** | 23 |

In [113…

fig **=** plt**.**figure(figsize**=**(7, 7))

ax **=** Axes3D(fig, rect**=**[0, 0, .99, 1], elev**=**20, azim**=**210) ax**.**scatter(KM5\_df['Age'],

In [221…

*#Using Dendrograms to find out optimal clusters*

dendrogram **=** sch**.**dendrogram(sch**.**linkage(X, method **=** 'ward')) plt**.**title('Dendrogam', fontsize **=** 20) plt**.**xlabel('Customers')

plt**.**ylabel('Ecuclidean Distance') plt**.**show()

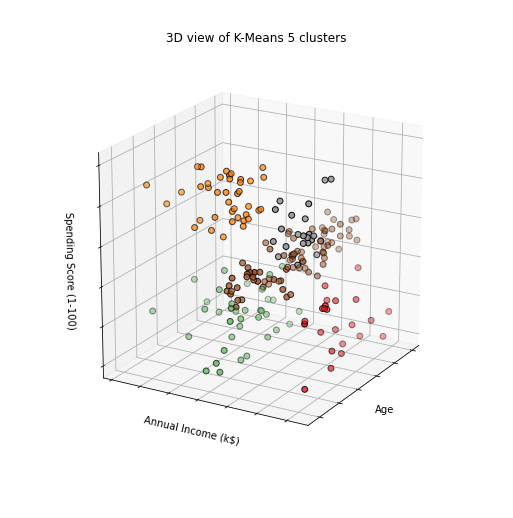
KM5\_df['Annual Income (k$)'], KM5\_df['Spending Score (1-100)'], c**=**KM5\_df['labels'],

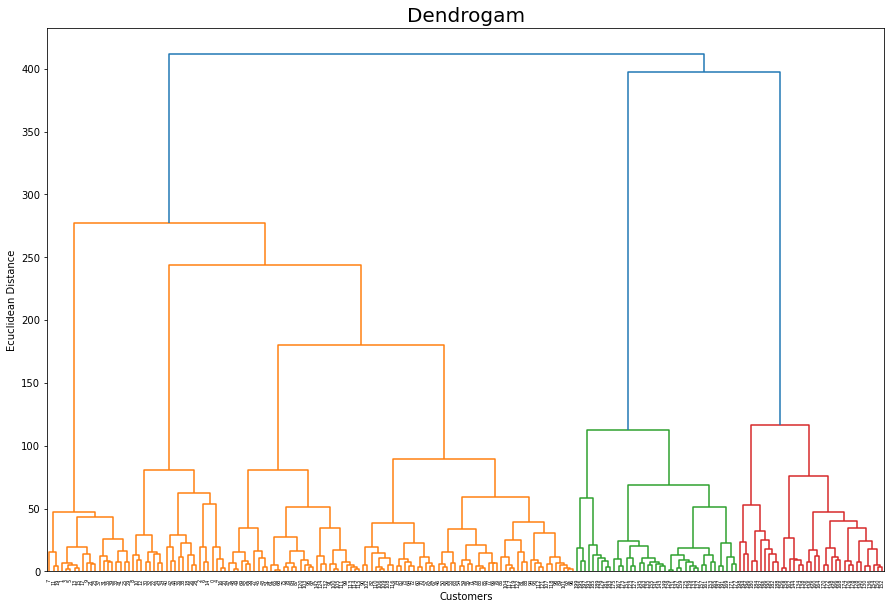
s**=**35, edgecolor**=**'k', cmap**=**plt**.**cm**.**Set1)

ax**.**w\_xaxis**.**set\_ticklabels([]) ax**.**w\_yaxis**.**set\_ticklabels([]) ax**.**w\_zaxis**.**set\_ticklabels([]) ax**.**set\_xlabel('Age') ax**.**set\_ylabel('Annual Income (k$)') ax**.**set\_zlabel('Spending Score (1-100)')

ax**.**set\_title('3D view of K-Means 5 clusters') ax**.**dist **=** 12

plt**.**show()





In [226…

clustering\_agglomerative **=** AgglomerativeClustering(n\_clusters **=** 5, affinity **=** 'euclidean' clustering\_agglomerative**.**fit(X)

labels\_clustering\_agglomerative **=** clustering\_agglomerative**.**labels\_ clustering\_agglomerative\_df **=** data**.**copy() clustering\_agglomerative\_df['labels'] **=** labels\_clustering\_agglomerative

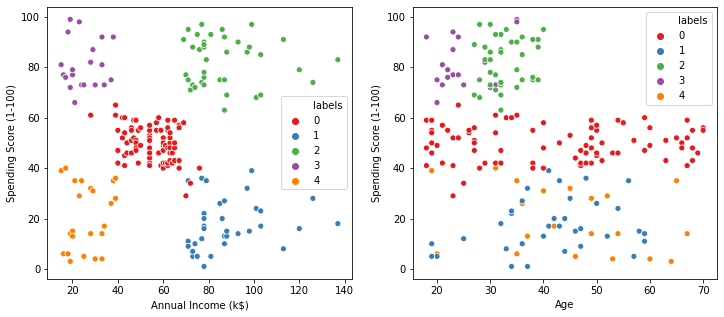
In [234…

fig1, (axes) **=** plt**.**subplots(1,2,figsize**=**(12,5))

sns**.**scatterplot('Annual Income (k$)', 'Spending Score (1-100)', data**=**clustering\_agglomera hue**=**'labels', ax**=**axes[0], palette**=**'Set1', legend**=**'full')

sns**.**scatterplot('Age', 'Spending Score (1-100)', data**=**clustering\_agglomerative\_df, hue**=**'labels', palette**=**'Set1', ax**=**axes[1], legend**=**'full')

plt**.**show()



In [121…

eps\_values **=** np**.**arange(8,12.75,0.25) *# eps values to be investigated*

min\_samples **=** np**.**arange(3,10) *# min\_samples values to be investigated*

DBSCAN\_params **=** list(product(eps\_values, min\_samples))*# creates a mxn combinations of eps*

DBSCAN

In [123…

no\_of\_clusters **=** [] sil\_score **=** []

**for** p **in** DBSCAN\_params:

DBSCN\_clustering **=** DBSCAN(eps**=**p[0], min\_samples**=**p[1])**.**fit(X) no\_of\_clusters**.**append(len(np**.**unique(DBSCN\_clustering**.**labels\_))) sil\_score**.**append(silhouette\_score(X, DBSCN\_clustering**.**labels\_))

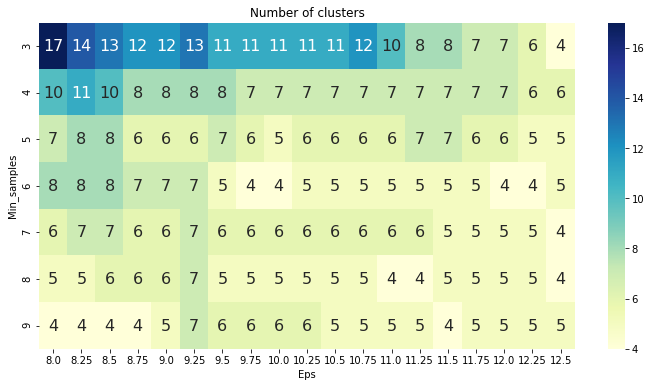
In [124…

tmp **=** pd**.**DataFrame**.**from\_records(DBSCAN\_params, columns **=**['Eps', 'Min\_samples']) tmp['No\_of\_clusters'] **=** no\_of\_clusters

pivot\_1 **=** pd**.**pivot\_table(tmp, values**=**'No\_of\_clusters', index**=**'Min\_samples', columns**=**'Eps' fig, ax **=** plt**.**subplots(figsize**=**(12,6))

sns**.**heatmap(pivot\_1, annot**=True**,annot\_kws**=**{"size": 16}, cmap**=**"YlGnBu", ax**=**ax)

ax**.**set\_title('Number of clusters') plt**.**show()



# Range of clusters is between 17 to 4. Now we see which cluster has the maximum value from the heat map and choose the corresponding min and eps values

In [127…

tmp **=** pd**.**DataFrame**.**from\_records(DBSCAN\_params, columns **=**['Eps', 'Min\_samples']) tmp['Sil\_score'] **=** sil\_score

pivot\_1 **=** pd**.**pivot\_table(tmp, values**=**'Sil\_score', index**=**'Min\_samples', columns**=**'Eps') fig, ax **=** plt**.**subplots(figsize**=**(18,6))

sns**.**heatmap(pivot\_1, annot**=True**, annot\_kws**=**{"size": 10}, cmap**=**"YlGnBu", ax**=**ax)

plt**.**show()



# The highest value of 0.26 is for 12.5 and 4

In [129…

DBSCN\_clustering **=** DBSCAN(eps**=**12.5, min\_samples**=**4)**.**fit(X) DBSCAN\_cluster **=** X**.**copy()

DBSCAN\_cluster**.**loc[:,'Cluster'] **=** DBSCN\_clustering**.**labels\_ *# append labels to points*

In [130…

DBSCAN\_clustered\_sizes **=** DBSCAN\_cluster**.**groupby('Cluster')**.**size()**.**to\_frame()

DBSCAN\_clustered\_sizes**.**columns **=** ["DBSCAN\_size"] DBSCAN\_clustered\_sizes

|  |  |  |
| --- | --- | --- |
| Out[130… | **Cluster** | **DBSCAN\_size** |
|  | **-1** | 18 |
|  | **0** | 112 |
|  | **1** | 8 |
|  | **2** | 34 |
|  | **3** | 24 |
|  | **4** | 4 |

# We have

Outliers = 18 denoted by cluster -1

# Core clusters = 0-4 where 0 has around 112 data points

In [131…

outliers **=** DBSCAN\_cluster[DBSCAN\_cluster['Cluster']**==-**1] fig2, (axes) **=** plt**.**subplots(1,2,figsize**=**(12,5))

sns**.**scatterplot('Annual Income (k$)', 'Spending Score (1-100)', data**=**DBSCAN\_clustered[DBSCAN\_clustered['Cluster']**!=-**1], hue**=**'Cluster', ax**=**axes[0], palette**=**'Set1', legend**=**'full', s**=**45)

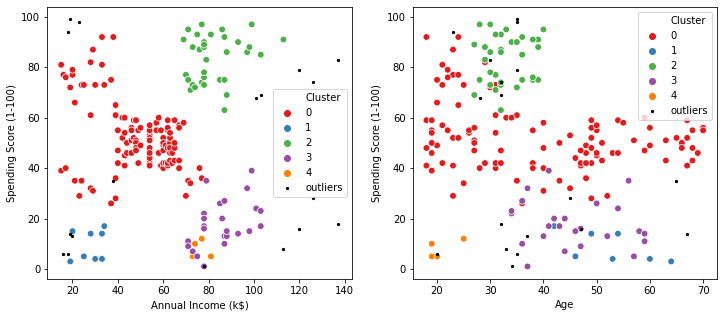
sns**.**scatterplot('Age', 'Spending Score (1-100)',

data**=**DBSCAN\_cluster[DBSCAN\_cluster['Cluster']**!=-**1], hue**=**'Cluster', palette**=**'Set1', ax**=**axes[1], legend**=**'full', s**=**45)

axes[0]**.**scatter(outliers['Annual Income (k$)'], outliers['Spending Score (1-100)'], s**=**5, axes[1]**.**scatter(outliers['Age'], outliers['Spending Score (1-100)'], s**=**5, label**=**'outliers axes[0]**.**legend()

axes[1]**.**legend() plt**.**setp(axes[0]**.**get\_legend()**.**get\_texts(), fontsize**=**'10') plt**.**setp(axes[1]**.**get\_legend()**.**get\_texts(), fontsize**=**'10')

plt**.**show()



In [245…

no\_of\_clusters **=** []

preferences **=** range(**-**20000,**-**500,200) af\_sil\_score **=** [] *# silouette scores*

**for** p **in** preferences:

AF **=** AffinityPropagation(preference**=**p, max\_iter**=**200)**.**fit(X) no\_of\_clusters**.**append((len(np**.**unique(AF**.**labels\_)))) af\_sil\_score**.**append(silhouette\_score(X, AF**.**labels\_))

af\_results **=** pd**.**DataFrame([preferences, no\_of\_clusters, af\_sil\_score], index**=**['preference af\_results**.**sort\_values(by**=**'sil\_score', ascending**=False**)**.**head()

|  |  |  |  |
| --- | --- | --- | --- |
| Out[245… | **preference** | **clusters** | **sil\_score** |
|  | **41** -11800.0 | 6.0 | 0.451649 |
|  | **14** -17200.0 | 6.0 | 0.451440 |
|  | **55** -9000.0 | 6.0 | 0.451440 |
|  | **26** -14800.0 | 6.0 | 0.451440 |
|  | **32** -13600.0 | 6.0 | 0.451440 |

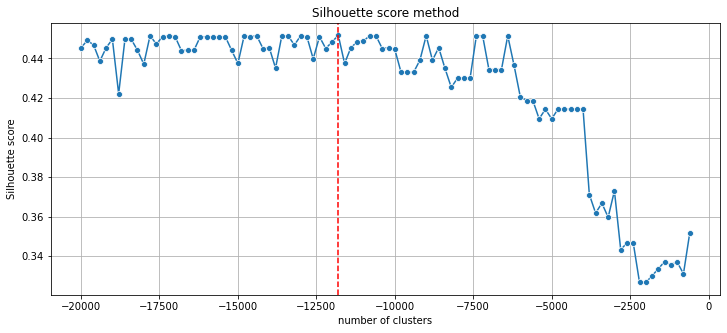
In [246…

fig, ax **=** plt**.**subplots(figsize**=**(12,5))

ax **=** sns**.**lineplot(preferences, af\_sil\_score, marker**=**'o', ax**=**ax) ax**.**set\_title("Silhouette score method")

ax**.**set\_xlabel("number of clusters") ax**.**set\_ylabel("Silhouette score") ax**.**axvline(**-**11800, ls**=**"--", c**=**"red") plt**.**grid()

plt**.**show()



In [248…

AF **=** AffinityPropagation(preference**=-**11800)**.**fit(X)

In [250…

AF\_clustered **=** X**.**copy()

AF\_clustered**.**loc[:,'Cluster'] **=** AF**.**labels\_ *# append labels to points*

In [251…

AF\_clust\_sizes **=** AF\_clustered**.**groupby('Cluster')**.**size()**.**to\_frame() AF\_clust\_sizes**.**columns **=** ["AF\_size"]

AF\_clust\_sizes

|  |  |  |
| --- | --- | --- |
| Out[251… | **Cluster** | **AF\_size** |
|  | **0** | 22 |
|  | **1** | 22 |
|  | **2** | 44 |
|  | **3** | 39 |
|  | **4** | 34 |
|  | **5** | 39 |

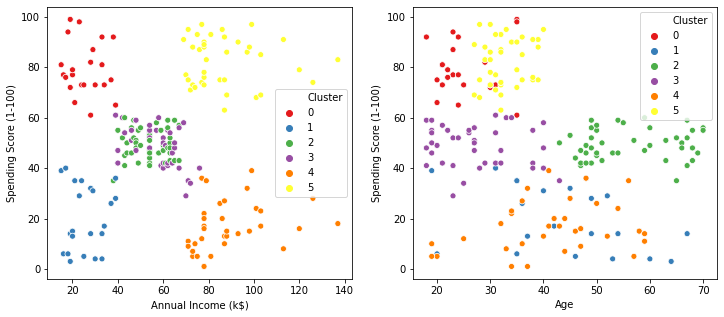
In [252…

fig3, (ax\_af) **=** plt**.**subplots(1,2,figsize**=**(12,5))

scat\_1 **=** sns**.**scatterplot('Annual Income (k$)', 'Spending Score (1-100)', data**=**AF\_clustere hue**=**'Cluster', ax**=**ax\_af[0], palette**=**'Set1', legend**=**'full')

sns**.**scatterplot('Age', 'Spending Score (1-100)', data**=**AF\_clustered, hue**=**'Cluster', palette**=**'Set1', ax**=**ax\_af[1], legend**=**'full')

plt**.**setp(ax\_af[0]**.**get\_legend()**.**get\_texts(), fontsize**=**'10') plt**.**setp(ax\_af[1]**.**get\_legend()**.**get\_texts(), fontsize**=**'10') plt**.**show()



In [254…

clusters **=** pd**.**concat([KM\_clust\_sizes,KM\_clust\_sizes5, DBSCAN\_clust\_sizes, AF\_clust\_sizes] clusters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[254… |  | **KM\_size** | **KM\_size** | **DBSCAN\_size** | **AF\_size** |
|  | **-1** | NaN | NaN | 18.0 | NaN |
|  | **0** | 45.0 | 23.0 | 112.0 | 22.0 |
|  | **1** | 39.0 | 36.0 | 8.0 | 22.0 |
|  | **2** | 35.0 | 39.0 | 34.0 | 44.0 |
|  | **3** | 22.0 | 79.0 | 24.0 | 39.0 |
|  | **4** | 21.0 | 23.0 | 4.0 | 34.0 |
|  | **5** | 38.0 | NaN | NaN | 39.0 |

# From the above comparisons, it is clear that DBSCAN failed to generate reasonable clusters. It is due to its problems in recognising clusters of various densities (which are present in this case).

In turn, K-Means and Affinity Propagation algorithms created reasonable 6 clusters.