In [47]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch
import seaborn as sns
from sklearn.preprocessing import normalize
```

In [48]:

```
dt = pd.read_csv('EastWestAirlines.csv')
dt
```

Out[48]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	1	28143	0	1	1	1	174	1
1	2	19244	0	1	1	1	215	2
2	3	41354	0	1	1	1	4123	4
3	4	14776	0	1	1	1	500	1
4	5	97752	0	4	1	1	43300	26
...	...	...	...	...	...	...	...	...
3994	4017	18476	0	1	1	1	8525	4
3995	4018	64385	0	1	1	1	981	5
3996	4019	73597	0	3	1	1	25447	8
3997	4020	54899	0	1	1	1	500	1
3998	4021	3016	0	1	1	1	0	0

3999 rows × 12 columns

In [49]:

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ID#                3999 non-null   int64
 1   Balance            3999 non-null   int64
 2   Qual_miles         3999 non-null   int64
 3   cc1_miles          3999 non-null   int64
 4   cc2_miles          3999 non-null   int64
 5   cc3_miles          3999 non-null   int64
 6   Bonus_miles        3999 non-null   int64
 7   Bonus_trans        3999 non-null   int64
 8   Flight_miles_12mo  3999 non-null   int64
 9   Flight_trans_12    3999 non-null   int64
 10  Days_since_enroll  3999 non-null   int64
 11  Award?             3999 non-null   int64
dtypes: int64(12)
memory usage: 375.0 KB
```

In [50]:

```
dt.isna().sum()
```

Out[50]:

```
ID#                  0
Balance              0
Qual_miles           0
cc1_miles            0
cc2_miles            0
cc3_miles            0
Bonus_miles          0
Bonus_trans          0
Flight_miles_12mo    0
Flight_trans_12      0
Days_since_enroll    0
Award?               0
dtype: int64
```

In [51]:

```
dt = dt.drop(['ID#'],axis=1)
```

In [52]:

```
dt
```

Out[52]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight
0	28143	0	1	1	1	174	1	
1	19244	0	1	1	1	215	2	
2	41354	0	1	1	1	4123	4	
3	14776	0	1	1	1	500	1	
4	97752	0	4	1	1	43300	26	
...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	
3995	64385	0	1	1	1	981	5	
3996	73597	0	3	1	1	25447	8	
3997	54899	0	1	1	1	500	1	
3998	3016	0	1	1	1	0	0	

3999 rows × 11 columns

In [53]:

```
# Normalize heterogenous numerical data
```

In [54]:

```
dd_norm = pd.DataFrame(normalize(dt),columns=dt.columns)
dd_norm
```

Out[54]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Fligh
0	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034	
1	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098	
2	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095	
3	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061	
4	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243	
...	...	...	...	...	...	...	...	
3994	0.905810	0.0	0.000049	0.000049	0.000049	0.417949	0.000196	
3995	0.999649	0.0	0.000016	0.000016	0.000016	0.015231	0.000078	
3996	0.944948	0.0	0.000039	0.000013	0.000013	0.326726	0.000103	
3997	0.999592	0.0	0.000018	0.000018	0.000018	0.009104	0.000018	
3998	0.907271	0.0	0.000301	0.000301	0.000301	0.000000	0.000000	

3999 rows × 11 columns

In [55]:

```
dt.describe()
```

Out[55]:

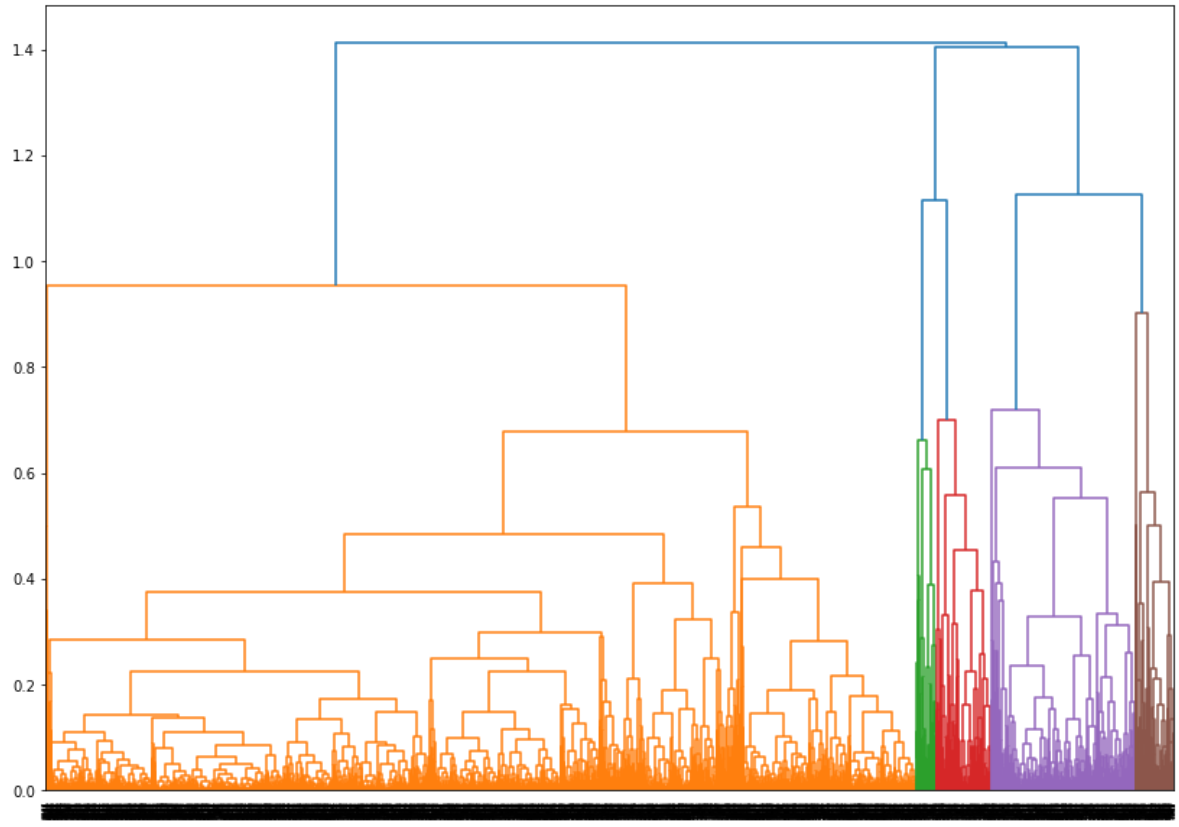
	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bon
count	3.999000e+03	3999.000000	3999.000000	3999.000000	3999.000000	3999.000000	39
mean	7.360133e+04	144.114529	2.059515	1.014504	1.012253	17144.846212	
std	1.007757e+05	773.663804	1.376919	0.147650	0.195241	24150.967826	
min	0.000000e+00	0.000000	1.000000	1.000000	1.000000	0.000000	
25%	1.852750e+04	0.000000	1.000000	1.000000	1.000000	1250.000000	
50%	4.309700e+04	0.000000	1.000000	1.000000	1.000000	7171.000000	
75%	9.240400e+04	0.000000	3.000000	1.000000	1.000000	23800.500000	
max	1.704838e+06	11148.000000	5.000000	3.000000	5.000000	263685.000000	

In [56]:

```
# Dendrograms
```

In [57]:

```
plt.figure(figsize=(14,10))
dendograms=sch.dendrogram(sch.linkage(dd_norm, method='complete'))
plt.show()
```



In [58]:

```
# Clusters
```

In [59]:

```
cl = AgglomerativeClustering(n_clusters=5,affinity='euclidean',linkage='ward')
cl
```

Out[59]:

```
AgglomerativeClustering(n_clusters=5)
```

In [62]:

```
y_cl = cl.fit_predict(dt)
Clusters = pd.DataFrame(y_cl,columns =['Clusters'] )
```

In [63]:

```
Clusters
```

Out[63]:

	Clusters
0	2
1	2
2	2
3	2
4	4
...	...
3994	2
3995	4
3996	4
3997	4
3998	2

3999 rows × 1 columns

In [64]:

```
dt['h_clusterid'] = cl.labels_
dt
```

Out[64]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight
0	28143	0	1	1	1	174	1	
1	19244	0	1	1	1	215	2	
2	41354	0	1	1	1	4123	4	
3	14776	0	1	1	1	500	1	
4	97752	0	4	1	1	43300	26	
...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	
3995	64385	0	1	1	1	981	5	
3996	73597	0	3	1	1	25447	8	
3997	54899	0	1	1	1	500	1	
3998	3016	0	1	1	1	0	0	

3999 rows × 12 columns

In [65]:

```
dt.groupby('h_clusterid').agg(['mean'])
```

Out[65]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_
	mean	mean	mean	mean	mean	mean	
h_clusterid							
0	841016.769231	512.692308	3.346154	1.000000	1.115385	52888.269231	22.3
1	158510.772436	276.342949	3.035256	1.008013	1.060897	35739.006410	17.1
2	22129.604577	95.710755	1.469565	1.018307	1.000000	7136.640732	8.2
3	355242.694030	424.671642	3.059701	1.022388	1.000000	46811.955224	19.5
4	75338.683495	120.885437	2.557282	1.009709	1.007767	22349.167961	14.0

In [ ]:

```

```

In [ ]:

```

```