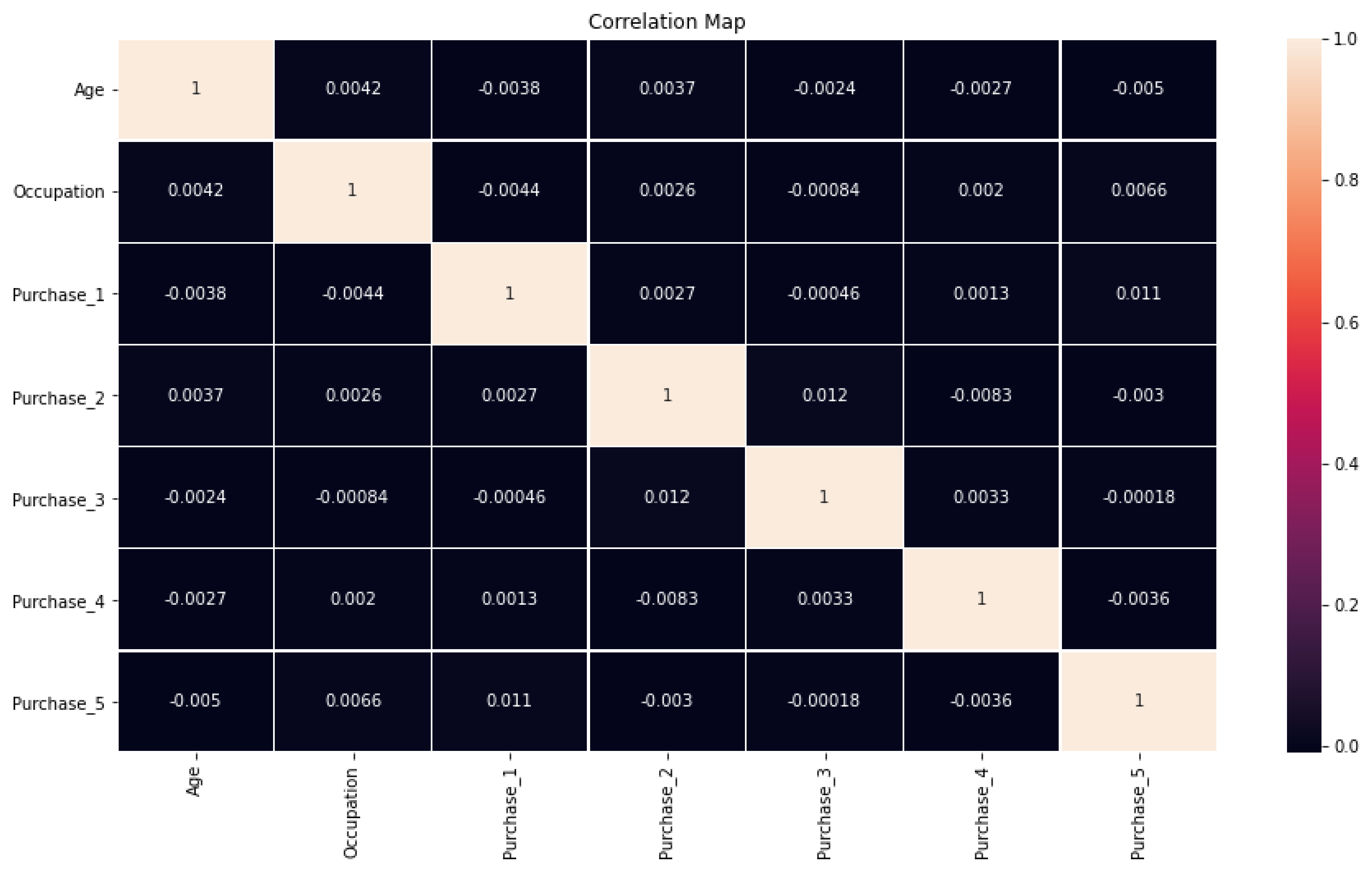
In [2]:
```
# Motivation: Try helping the company to achieve their bussines goals using their data.
# First we want to check if the are any correlated variables in our original data set.
# For this mission we need to import few liabraries that will help us to come up with a conclusion
import seaborn as sns
from matplotlib import pyplot as plt
from pyspark.sql import SparkSession
import pandas as pd
```

In [3]:
```
#loading the original data set with pandas liabrary
df=pd.read_csv("data_project.txt")
spark = SparkSession.builder.appName("new2").getOrCreate()
df1=spark.read.csv("data_project.txt",header=True,inferSchema=True)
```

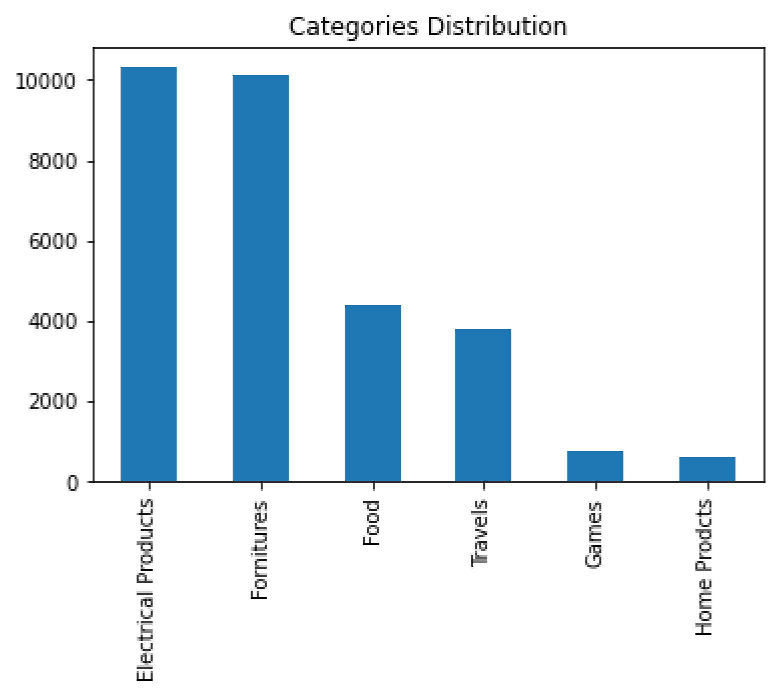
In [4]:
```
#Create a heat map that can help us to find correlation between variables.
plt.figure(figsize=(15,8))
sns.heatmap(df.corr(),annot=True,linewidth=0.5)
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.title("Correlation Map")
plt.show()
```



In [7]:
```
# Conclusion: the fact that we have created random data affects the correlation between the variables.
# The distribution of the data is uniform and every piece of data has the same probability to appear.
# This is the reason why i have added new varables to the original data set.
# Now, lets load the modified data set to produce some analysis.
df1=spark.read.csv("question_8_scenario.txt",header=True,inferSchema=True)
df1.show()
```

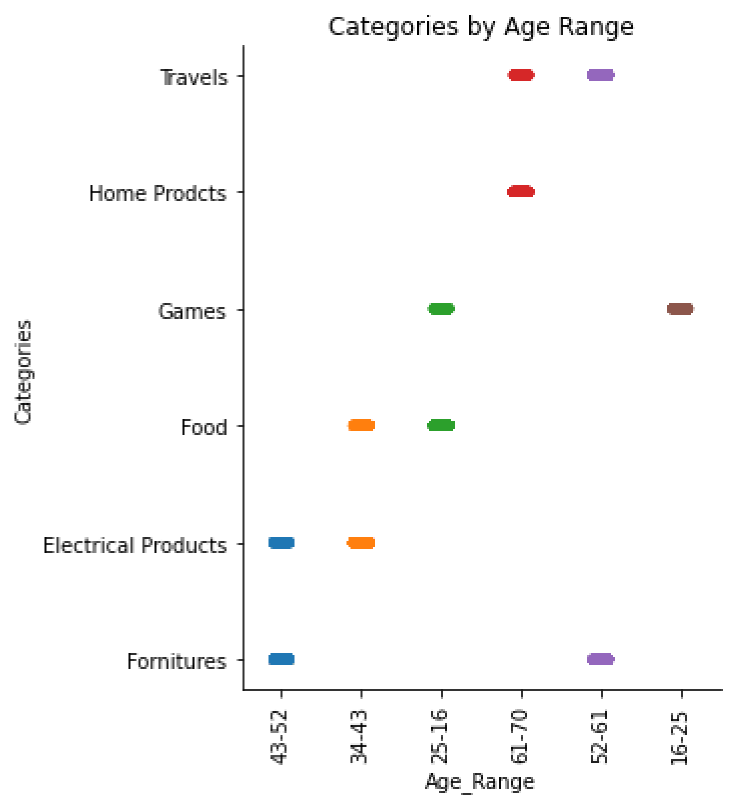
```
+---+----------+----------+----------+----------+----------+----------+------------------+-------------------+-
--------+
|Age|Occupation|Purchase_1|Purchase_2|Purchase_3|Purchase_4|Purchase_5|Range_of_purchases|         Categories|A
ge_Range|
+---+----------+----------+----------+----------+----------+----------+------------------+-------------------+-
--------+
| 50|        14|       795|      2902|       762|     12695|       458|       15000-20000|         Fornitures|
   43-52|
| 39|         7|      2307|      8776|      1218|      6389|     14237|       10000-15000|Electrical Products|
   34-43|
| 32|         5|      1314|      2056|       488|     10218|     14929|        2500-10000|               Food|
   25-16|
| 46|        18|      2310|      9043|      2086|       908|      5199|       15000-20000|         Fornitures|
   43-52|
| 46|        18|      4560|      3267|      1222|     16235|      4881|       15000-20000|         Fornitures|
   43-52|
| 48|         4|      3775|      2835|      1039|      3392|      4469|       15000-20000|         Fornitures|
   43-52|
| 64|         1|       385|      6416|      1172|      2403|      2381|           0-10000|       Home Prodcts|
   61-70|
| 31|        12|      2042|      4154|       795|     23827|     13004|        2500-10000|               Food|
   25-16|
| 49|         8|      2364|       413|      2409|      2505|      8347|       15000-20000|         Fornitures|
   43-52|
| 32|         6|      1551|      2308|        64|     18931|      4218|        2500-10000|               Food|
   25-16|
| 56|        12|      1498|      6427|       770|     23558|     12276|       20000-25000|            Travels|
   52-61|
| 50|         9|      2031|      3822|      1821|      9445|      3349|       15000-20000|         Fornitures|
   43-52|
| 27|         7|      3334|      2010|       928|      3251|     11242|        2500-10000|               Food|
   25-16|
| 40|         9|      1238|      7434|      1909|     18500|      1985|       10000-15000|Electrical Products|
   34-43|
| 56|        18|      3671|      8001|      2279|     13194|     14352|       20000-25000|            Travels|
   52-61|
| 48|        14|      2339|       981|       974|     24425|      1739|       15000-20000|         Fornitures|
   43-52|
| 51|        11|      4395|      1333|      1064|     10394|      9179|       15000-20000|         Fornitures|
   43-52|
| 54|         3|      2735|      5942|      1866|     22108|      9554|       20000-25000|            Travels|
   52-61|
| 38|        12|      3462|      5182|      1117|      4873|      9649|       10000-15000|Electrical Products|
   34-43|
| 64|         3|      4449|      3848|      1464|     14832|     10705|           0-10000|       Home Prodcts|
   61-70|
+---+----------+----------+----------+----------+----------+----------+------------------+-------------------+-
--------+
only showing top 20 rows
```

In [8]:
```
# I have added few columns: Range_of_purchases, Categories and Age_Range.
# these categories based on the Age column that has normal distribution from the original data set.
# lets load the data using pandas and create bar plot to check the distribution of the Categories column.
df2 = pd.read_csv("question_8_scenario.txt")
df2["Categories"].value_counts().plot(kind='bar')
plt.xticks(rotation=90)
plt.title("Categories Distribution")
plt.show()
```

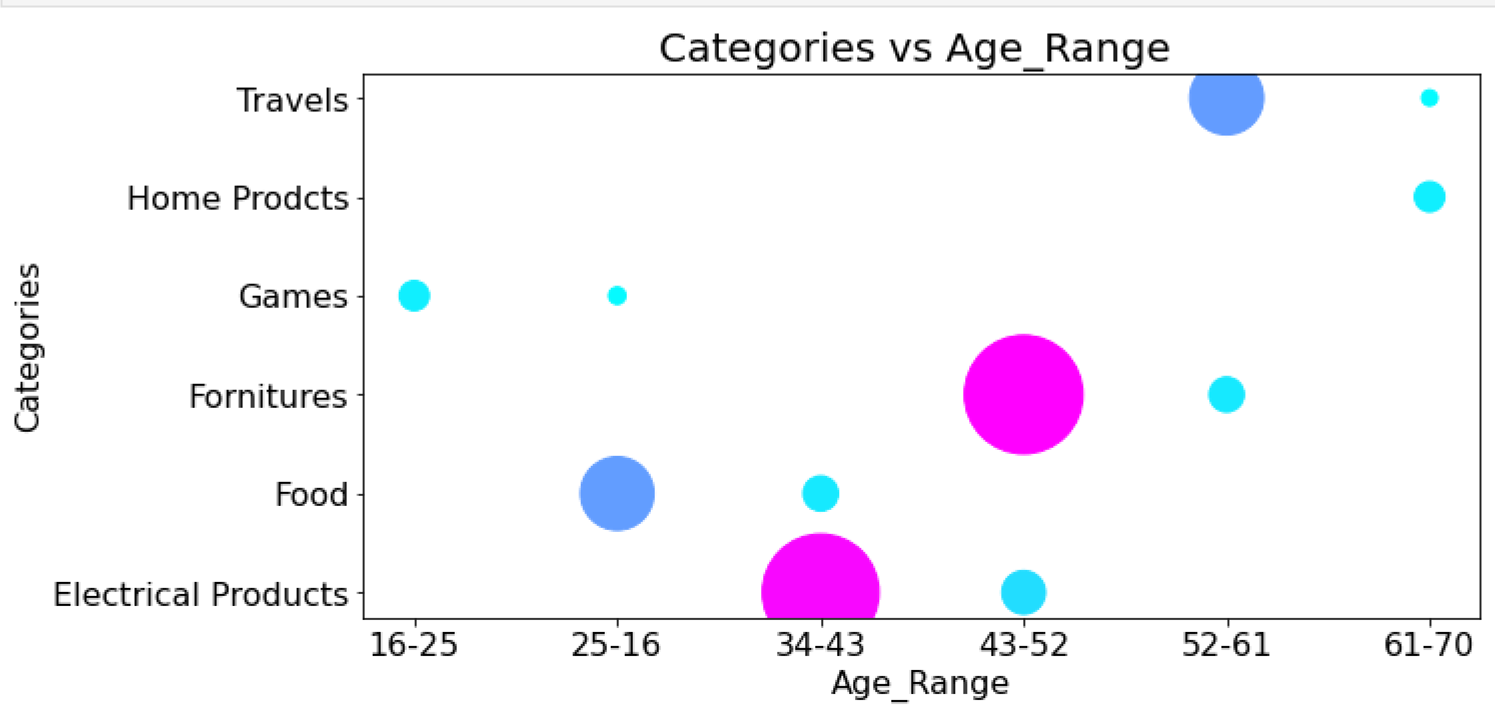


In [10]:
```
# Conclusion: the best sellers categories in the company are: Elctrical Products and Fornitures.
# Lets write a query to give the exact number of sales from each category:
df1.groupby("Categories").count().orderBy("count",ascending=False).show()
# lets say that we want to check which ages buy the Elctrical Products and Fornitures by using cat plot:
sns.catplot(x='Age_Range',y='Categories',data=df2)
plt.xticks(rotation=90)
plt.title("Categories by Age Range")
plt.show()
```

```
+-------------------+-----+
|         Categories|count|
+-------------------+-----+
|Electrical Products|10299|
|         Fornitures|10134|
|               Food| 4390|
|            Travels| 3788|
|              Games|  786|
|       Home Prodcts|  603|
+-------------------+-----+
```



In [11]:
```
# Conclusion: we can see that the ages between 43-52, 34-43 and 52-61 buy the categories: Electrical Products a
# Now, we want to check between these ages who buy the most from the bestsellers categories.
# To achieve this mission we will import bubble_plot liabrary:
from bubble_plot.bubble_plot import bubble_plot
bubble_plot(df1,'Age_Range','Categories',normalization_by_all=True)
plt.show()
```



In [16]:
```
# The query that shows the number of buyers in the ages between 34-43 that buy Electrical Products:
from pyspark.sql.functions import *
df1.where((col("Categories")=="Electrical Products")&(col("Age_Range")=="34-43")).count()
```

Out[16]:
```
9045
```

In [17]:
```
# The query that shows the number of buyers in the ages between 43-52 that buy Fornitures:
df1.where((col("Categories")=="Fornitures")&(col("Age_Range")=="43-52")).count()
```

Out[17]:
```
9327
```

In [ ]:
```
# Final conclusion: we can say to the company that the ages between 34 and 43 buy Electrical Products the most.
# The ages between 43 and 52 buy Fornitures the most.
# My recommendation: Try to find out the reasons why the company has a success in these categories.
# Afterwards try to implement the keys of this success on other categories as well.
```