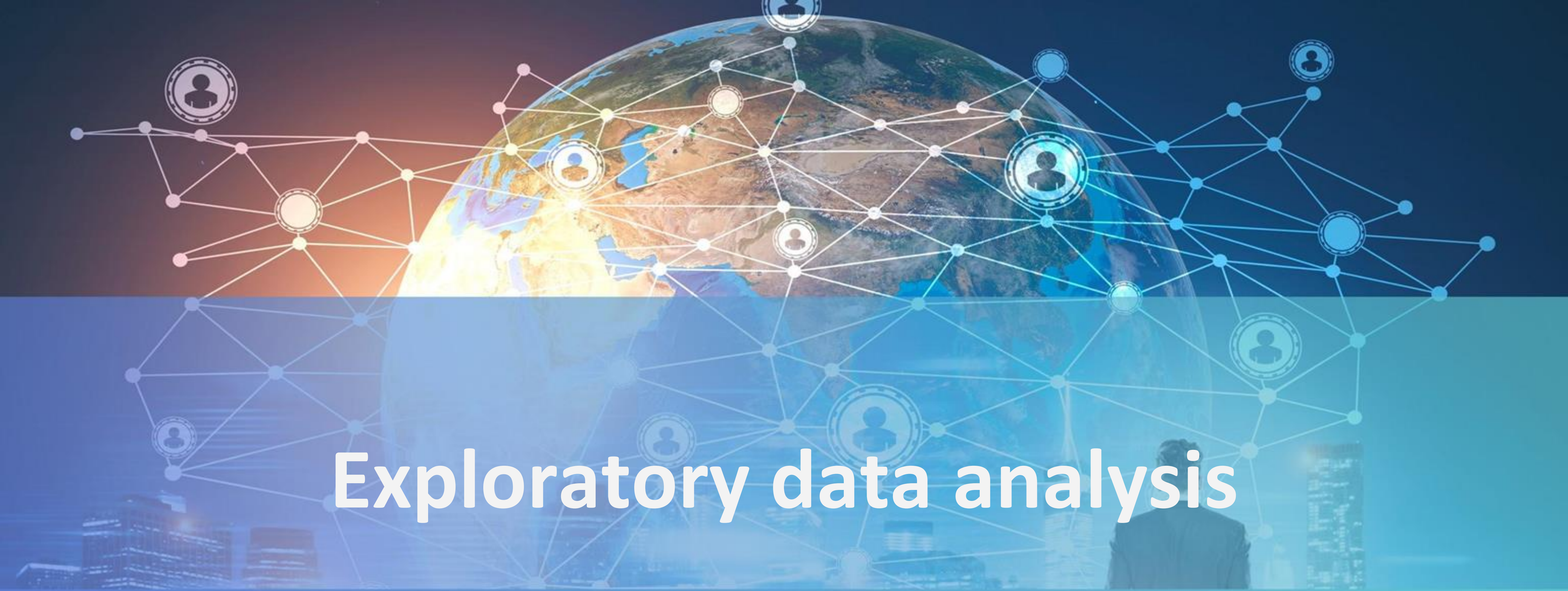


Exploratory data analysis

GITAA
Transforming careers

In this lecture

- Frequency tables
- Two-way tables
- Two-way table - joint probability
- Two-way table - marginal probability
- Two-way table - conditional probability
- Correlation

Importing data into Spyder

- Importing necessary libraries

```
import os
```

‘os’ library to change the working directory

```
import pandas as pd
```

‘pandas’ library to work with dataframes

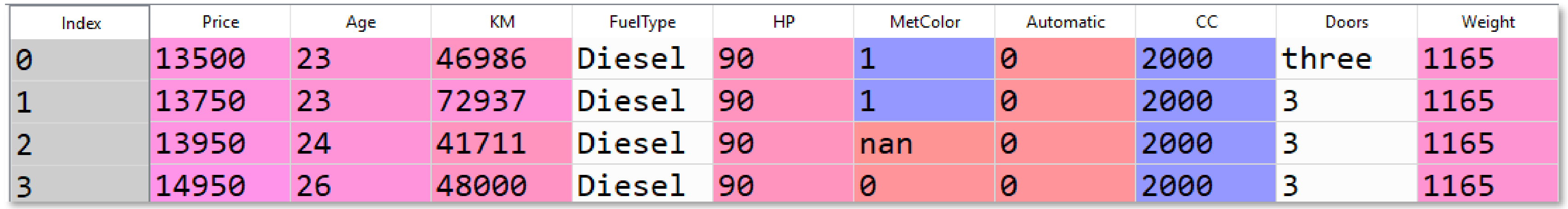
- Changing the working directory

```
os. chdir("D:\Pandas")
```

Importing data into Spyder

- Importing data

```
cars_data = pd.read_csv('Toyota.csv',index_col=0,
                          na_values=["??","????"])
```



Index	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	three	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	nan	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165

- Creating copy of original data

```
cars_data2 = cars_data.copy()
```

Frequency tables

pandas.crosstab()

- To compute a simple cross-tabulation of one, two (or more) factors
- By default computes a frequency table of the factors

```
pd.crosstab(index=cars_data2['FuelType'],columns='count',
            dropna=True)
```

Size of data

1436– Original data

1336 – after dropping nan values

```
Out[3]:
col_0     count
FuelType
CNG          15
Diesel      144
Petrol     1177
```

Most of the cars have petrol as fuel type

Two-way tables

pandas.crosstab()

- To look at the frequency distribution of gearbox types with respect to different fuel types of the cars

```
pd.crosstab(index   = cars_data2['Automatic'],
            columns = cars_data2['FuelType'],
            dropna  = True)
```

```
Out[5]:
FuelType   CNG  Diesel  Petrol
Automatic
0           15     144    1104
1            0       0      73
```

Automatic

0- Manual gear box

1- Automatic gearbox

Two-way table - joint probability

pandas.crosstab()

- Joint probability is the likelihood of two independent events happening at the same time

```
pd.crosstab(index       = cars_data2['Automatic'],
            columns     = cars_data2['FuelType'],
            normalize = True,
            dropna      = True)
```

```
Out[16]:
FuelType          CNG     Diesel     Petrol
Automatic
0            0.010801   0.108011   0.828083
1            0.000000   0.000000   0.053105
```

Two-way table - marginal probability

pandas.crosstab()

- Marginal probability is the probability of the occurrence of the single event

```
pd.crosstab(index     =cars_data2['Automatic'],
            columns   = cars_data2['FuelType'],
            margins   = True,
            dropna    = True,
            normalize = True)
```

probability of cars having manual gear box when the fuel type are CNG or Diesel or Petrol is 0.95

```
Out[17]:
FuelType        CNG    Diesel    Petrol       All
Automatic
0          0.010801  0.108011  0.828083  0.946895
1          0.000000  0.000000  0.053105  0.053105
All        0.010801  0.108011  0.881188  1.000000
```

Two-way table - conditional probability

pandas.crosstab()

- Conditional probability is the probability of an event (A), given that another event (B) has already occurred
- Given the type of gear box, probability of different fuel type

```
pd.crosstab(index      = cars_data2['Automatic'],
            columns    = cars_data2['FuelType'],
            margins    = True,
            dropna     = True,
            normalize = 'index')
```

```
Out[19]:
FuelType          CNG    Diesel    Petrol
Automatic
0            0.011407  0.114068  0.874525
1            0.000000  0.000000  1.000000
All          0.010801  0.108011  0.881188
```

$\rightarrow$ Row sum = 1

Two-way table - conditional probability

pandas.crosstab()

- Conditional probability is the probability of an event (A), given that another event (B) has already occurred

```
pd.crosstab(index      = cars_data2['Automatic'],
            columns    = cars_data2['FuelType'],
            margins    = True, dropna = True,
            normalize = 'columns')
```

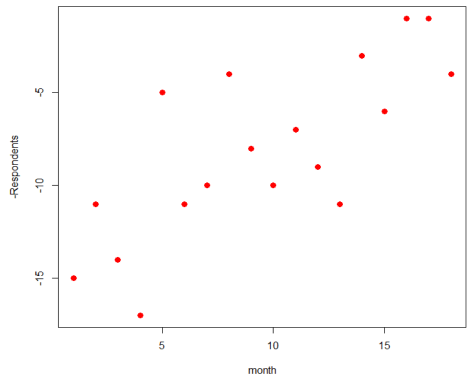
```
Out[20]:
FuelType   CNG  Diesel    Petrol       All
Automatic
0          1.0     1.0  0.939734  0.946895
1          0.0     0.0  0.060266  0.053105
```

Column sum = 1

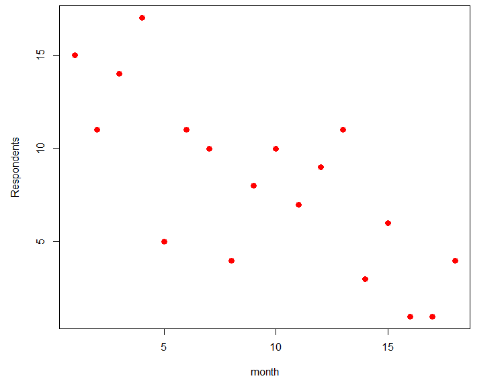
Correlation

- Correlation: the strength of association between two variables
- Visual representation of correlation: Scatter plots

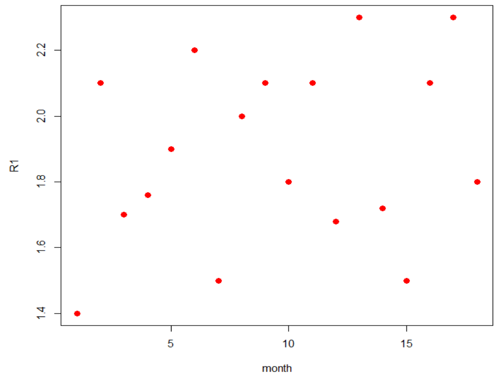
Positive trend



Negative trend



Little or no correlation



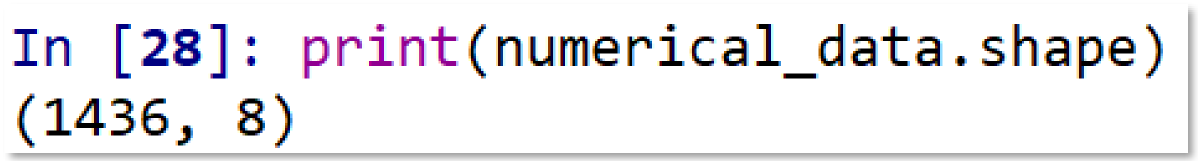
Correlation

```
DataFrame.corr(self, method='pearson’)
```

- To compute pairwise correlation of columns excluding NA/null values
- Excluding the categorical variables to find the Pearson’s correlation

```
numerical_data = cars_data2.select_dtypes(exclude=[object])
```

- Let’ s check the no. of variables available under ***numerical_data***



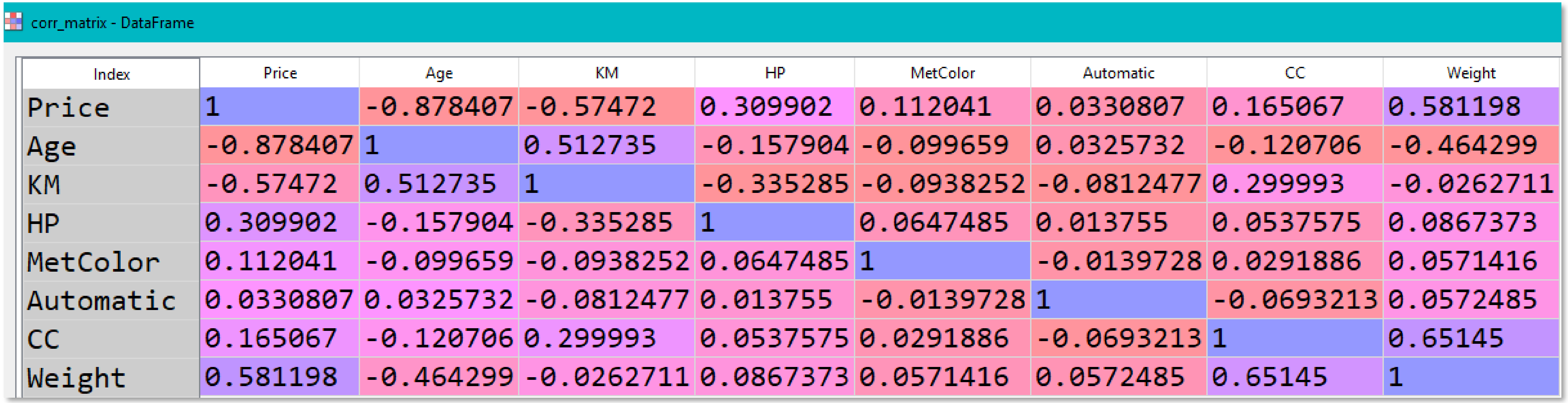
```
In [28]: print(numerical_data.shape)
(1436, 8)
```

Correlation

```
DataFrame.corr(self, method='pearson’)
```

- Correlation between numerical variables

```
corr_matrix = numerical_data.corr()
```


corr_matrix - DataFrame

Index	Price	Age	KM	HP	MetColor	Automatic	CC	Weight
Price	1	-0.878407	-0.57472	0.309902	0.112041	0.0330807	0.165067	0.581198
Age	-0.878407	1	0.512735	-0.157904	-0.099659	0.0325732	-0.120706	-0.464299
KM	-0.57472	0.512735	1	-0.335285	-0.0938252	-0.0812477	0.299993	-0.0262711
HP	0.309902	-0.157904	-0.335285	1	0.0647485	0.013755	0.0537575	0.0867373
MetColor	0.112041	-0.099659	-0.0938252	0.0647485	1	-0.0139728	0.0291886	0.0571416
Automatic	0.0330807	0.0325732	-0.0812477	0.013755	-0.0139728	1	-0.0693213	0.0572485
CC	0.165067	-0.120706	0.299993	0.0537575	0.0291886	-0.0693213	1	0.65145
Weight	0.581198	-0.464299	-0.0262711	0.0867373	0.0571416	0.0572485	0.65145	1

Summary

- Frequency tables
- Two-way tables
- Two-way table - joint probability
- Two-way table - marginal probability
- Two-way table - conditional probability
- Correlation

```
ror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
ier_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
 = bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactly
```

THANK YOU