[3]:
```
import pandas as pd
import numpy as np
df = pd.read_csv("lv2-2305-2.csv")
df.head()
```

[3]:

	REG_NO	GRE	TOEFL	UNIV_RATING	SOP	LOR	CGPA	RESEARCH	ADMIT
0	1	337	118	B	4.5	4.5	9.65	1	Admitted
1	2	324	107	B	4.0	4.5	8.87	1	Admitted
2	3	316	104	C	3.0	3.5	8.00	1	Denied
3	4	322	110	C	3.5	2.5	8.67	1	Admitted
4	5	314	103	D	2.0	3.0	8.21	0	Denied

[9]:
```
df1 = df.copy()
```

*[10]:
```
### Step 1-1
df1_a = df1[df1["ADMIT"] == "Admitted"]
df1_d = df1[df1["ADMIT"] == "Denied"]
df1_a.head()
```

[10]:

	REG_NO	GRE	TOEFL	UNIV_RATING	SOP	LOR	CGPA	RESEARCH	ADMIT
0	1	337	118	B	4.5	4.5	9.65	1	Admitted
1	2	324	107	B	4.0	4.5	8.87	1	Admitted
3	4	322	110	C	3.5	2.5	8.67	1	Admitted
5	6	330	115	A	4.5	3.0	9.34	1	Admitted
6	7	321	109	C	3.0	4.0	8.20	1	Admitted

[11]:
```
df1_d.head()
```

[11]:

	REG_NO	GRE	TOEFL	UNIV_RATING	SOP	LOR	CGPA	RESEARCH	ADMIT
2	3	316	104	C	3.0	3.5	8.00	1	Denied
4	5	314	103	D	2.0	3.0	8.21	0	Denied
7	8	308	101	D	3.0	4.0	7.90	0	Denied
8	9	302	102	E	2.0	1.5	8.00	0	Denied
9	10	323	108	C	3.5	3.0	8.60	0	Denied

[44]:
```
### Step 1-2
df1_a["GRE_R"] = df1_a["GRE"].rank(method='min', ascending=False)
df1_a["TOEFL_R"] = df1_a["TOEFL"].rank(method='min', ascending=False)
df1_a["CGPA_R"] = df1_a["CGPA"].rank(method='min', ascending=False)
print(df1_a.head())
df1_d["GRE_R"] = df1_d["GRE"].rank(method='min', ascending=False)
df1_d["TOEFL_R"] = df1_d["TOEFL"].rank(method='min', ascending=False)
df1_d["CGPA_R"] = df1_d["CGPA"].rank(method='min', ascending=False)
print(df1_d.head())
```

```
   REG_NO  GRE  TOEFL UNIV_RATING  SOP  LOR  CGPA  RESEARCH     ADMIT  GRE_R  \
0       1  337    118           B  4.5  4.5  9.65         1  Admitted   16.0
1       2  324    107           B  4.0  4.5  8.87         1  Admitted   97.0
3       4  322    110           C  3.5  2.5  8.67         1  Admitted  120.0
5       6  330    115           A  4.5  3.0  9.34         1  Admitted   51.0
6       7  321    109           C  3.0  4.0  8.20         1  Admitted  131.0

   TOEFL_R  CGPA_R
0     18.0    18.0
1    153.0   126.0
3    109.0   154.0
5     45.0    47.0
6    135.0   179.0
   REG_NO  GRE  TOEFL UNIV_RATING  SOP  LOR  CGPA  RESEARCH   ADMIT  GRE_R  \
2       3  316    104           C  3.0  3.5  8.00         1  Denied   50.0
4       5  314    103           D  2.0  3.0  8.21         0  Denied   71.0
7       8  308    101           D  3.0  4.0  7.90         0  Denied  125.0
8       9  302    102           E  2.0  1.5  8.00         0  Denied  166.0
9      10  323    108           C  3.5  3.0  8.60         0  Denied   15.0

   TOEFL_R  CGPA_R
2    102.0   155.0
4    123.0   114.0
7    150.0   168.0
8    136.0   155.0
9     31.0    39.0
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_a["GRE_R"] = df1_a["GRE"].rank(method='min', ascending=False)
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_a["TOEFL_R"] = df1_a["TOEFL"].rank(method='min', ascending=False)
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_a["CGPA_R"] = df1_a["CGPA"].rank(method='min', ascending=False)
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_d["GRE_R"] = df1_d["GRE"].rank(method='min', ascending=False)
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_d["TOEFL_R"] = df1_d["TOEFL"].rank(method='min', ascending=False)
C:\Users\PC\AppData\Local\Temp\ipykernel_9716\1869587703.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1_d["CGPA_R"] = df1_d["CGPA"].rank(method='min', ascending=False)
```

[46]:
```
### Step 1-3
from scipy.stats import spearmanr
correlation1, p_value1 = spearmanr(df1_a["GRE_R"], df1_a["CGPA_R"])
correlation2, p_value2 = spearmanr(df1_a["TOEFL_R"], df1_a["CGPA_R"])

print(f"스피어만 상관계수1: {correlation1.round(2)}")
print(f"p-value1: {p_value1}")
print(f"스피어만 상관계수2: {correlation2.round(2)}")
print(f"p-value2: {p_value2}")
```

```
스피어만 상관계수1: 0.78
p-value1: 1.2346594088259062e-37
스피어만 상관계수2: 0.74
p-value2: 2.65633884835193e-32
```

[41]:
```
A = correlation1
A_VAR = df1_a["GRE_R"]
A_VAR
```

[41]:
```
0      165.0
1       76.5
3       56.0
5      127.5
6       45.0
       ...
394    120.5
395     76.5
396     88.5
397    127.5
399    146.5
Name: GRE_R, Length: 180, dtype: float64
```

[45]:
```
### Step 1-4
from scipy.stats import spearmanr
correlation3, p_value3 = spearmanr(df1_d["GRE_R"], df1_d["CGPA_R"])
correlation4, p_value4 = spearmanr(df1_d["TOEFL_R"], df1_d["CGPA_R"])

print(f"스피어만 상관계수3: {correlation3.round(2)}")
print(f"p-value3: {p_value3}")
print(f"스피어만 상관계수4: {correlation4.round(2)}")
print(f"p-value4: {p_value4}")
```

```
스피어만 상관계수3: 0.6
p-value3: 1.7069327747848276e-22
스피어만 상관계수4: 0.62
p-value4: 1.918727502244615e-24
```

[50]:
```
B = correlation4
B_VAR = df1_d["CGPA_R"]
B_VAR
```

[50]:
```
2      155.0
4      114.0
7      168.0
8      155.0
9       39.0
       ...
387    136.0
388    185.0
390    107.0
391     30.0
398     13.0
Name: CGPA_R, Length: 220, dtype: float64
```

[52]:
```
abs(A-B).round(2)
```

[52]:
```
np.float64(0.16)
```

[ ]:
```
```