NICOLA PULIGHEDDU

In [ ]:

ESE 1

In [ ]:

```
import pandas as pd

df = pd.read_excel('coalpublic2013.xls',header = 3)
df.dropna().head()
```

In [16]:

```
somma = df["Production (short tons)"].sum()
somma
```

Out[16]:

```
984841779
```

In [19]:

```
media = df["Production (short tons)"].mean()
media
```

Out[19]:

```
679201.2268965517
```

In [20]:

```
minimum = df["Production (short tons)"].min()
minimum
```

Out[20]:

```
0
```

In [21]:

```
maximum = df["Production (short tons)"].max()
maximum
```

Out[21]:

```
111005549
```

In [ ]:

ESE 2

In [27]:

```
employ_df = pd.read_excel('employee.xlsx')
employ_df.head()
```

Out[27]:

	emp_id	first_name	last_name	hire_date
0	100	Steven	King	2003-06-17
1	101	Neena	Kochhar	2005-09-21
2	102	Lex	De Haan	2001-01-13
3	103	Alexander	Hunold	2006-01-03
4	104	Bruce	Ernst	2007-05-21

In [32]:

```
employ_df[employ_df["hire_date"].between("2005-01-01","2005-12-31")].sort_values(by=["hire_date"])
```

Out[32]:

	emp_id	first_name	last_name	hire_date
5	105	David	Austin	2005-06-25
17	117	Sigal	Tobias	2005-07-24
1	101	Neena	Kochhar	2005-09-21
10	110	John	Chen	2005-09-28
11	111	Ismael	Sciarra	2005-09-30
16	116	Shelli	Baida	2005-12-24

In [ ]:

ESE 3

In [33]:

```
df_diamonds = pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-data/master/diamonds.csv")
```

In [34]:

```
df_diamonds
```

Out[34]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

53940 rows × 10 columns

In [61]:

```
mask1 = df_diamonds["x"] > 5
mask2 = df_diamonds["y"] > 5
mask3 = df_diamonds["z"] > 5

filt_diamonds = df_diamonds[mask1 & mask2 & mask3]
filt_diamonds
```

Out[61]:

	carat	cut	color	clarity	depth	table	price	x	y	z
11778	1.83	Fair	J	I1	70.0	58.0	5083	7.34	7.28	5.12
13002	2.14	Fair	J	I1	69.4	57.0	5405	7.74	7.70	5.36
13118	2.15	Fair	J	I1	65.5	57.0	5430	8.01	7.95	5.23
13562	1.96	Fair	F	I1	66.6	60.0	5554	7.59	7.56	5.04
13757	2.22	Fair	J	I1	66.7	56.0	5607	8.04	8.02	5.36
...	...	...	...	...	...	...	...	...	...	...
27748	2.00	Very Good	G	SI1	63.5	56.0	18818	7.90	7.97	5.04
27749	2.29	Premium	I	VS2	60.8	60.0	18823	8.50	8.47	5.16
48410	0.51	Very Good	E	VS1	61.8	54.7	1970	5.12	5.15	31.80
49189	0.51	Ideal	E	VS1	61.8	55.0	2075	5.15	31.80	5.12
49905	0.50	Very Good	G	VVS1	63.7	58.0	2180	5.01	5.04	5.06

1457 rows × 10 columns

In [ ]:

ESE 4

In [76]:

```
# mask1 = df_diamonds["cut"] == "Premium"
# mask2 = df_diamonds["cut"] == "Ideal"
# diam_prem_ideal = df_diamonds[mask1 | mask2]
# diam_prem_ideal

# o anche
# diam_prem_ideal = df_diamonds[(df_diamonds.cut.isin(["Premium","Ideal"]))]




diam_prem_ideal = df_diamonds[(df_diamonds.cut == "Premium") | (df_diamonds.cut == "Ideal") ]
diam_prem_ideal
```

Out[76]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
11	0.23	Ideal	J	VS1	62.8	56.0	340	3.93	3.90	2.46
12	0.22	Premium	F	SI1	60.4	61.0	342	3.88	3.84	2.33
...	...	...	...	...	...	...	...	...	...	...
53931	0.71	Premium	F	SI1	59.8	62.0	2756	5.74	5.73	3.43
53934	0.72	Premium	D	SI1	62.7	59.0	2757	5.69	5.73	3.58
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50

	carat	cut	color	clarity	depth	table	price	x	y	z
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

35342 rows × 10 columns

```
In [ ]:
```

```
In [ ]:
```