

```
filepath = '/content/newegg.csv'
```

```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv(filepath)
```

## ✓ Datasets of the Computer Components

data

	Unnamed: 0	brand_name	items_Desciption	ratings	prices	Category
0	0	AMD	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	14,723.99	cpu
1	1	AMD	Kingston 16GB (2 x 8GB) 240- Pin DDR3 SDRAM DDR...	(9)	378.99	cpu
2	2	AMD	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(4)	2,834.99	cpu
3	3	AMD	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(42)	661.99	cpu
4	4	AMD	AMD Ryzen Threadripper 3990X 64-Core 2.9 GHz S...	(691)	1,588.99	cpu
...	...	...	...	...	...	...
2700	195	NaN	EVGA SuperNOVA 1000 T2 220- T2-1000-X1 80+ TITA...	NaN	NaN	power
2701	196	NaN	Thermaltake TR2 TR-600 600W ATX12V v2.3 SLI Re...	NaN	NaN	power

### Changing the values in the Category Column due to dataset typographical error

```
data.loc[data["Category"] == "cpu", "Category"] = "CPU"
data.loc[data["Category"] == "gpu", "Category"] = "GPU"
data.loc[data["Category"] == "motherboard", "Category"] = "Motherboard"
data.loc[data["Category"] == "moniter", "Category"] = "Monitor"
data.loc[data["Category"] == "storege", "Category"] = "Storage"
data.loc[data["Category"] == "ram", "Category"] = "RAM"
data.loc[data["Category"] == "power", "Category"] = "Power"
```

```
data
```

	Unnamed: 0	brand_name	items_Desciption	ratings	prices	Category
<b>0</b>	0	AMD	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	14,723.99	CPU
<b>1</b>	1	AMD	Kingston 16GB (2 x 8GB) 240- Pin DDR3 SDRAM DDR...	(9)	378.99	CPU
<b>2</b>	2	AMD	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(4)	2,834.99	CPU
<b>3</b>	3	AMD	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(42)	661.99	CPU
<b>4</b>	4	AMD	AMD Ryzen Threadripper 3990X 64-Core 2.9 GHz S...	(691)	1,588.99	CPU
...	...	...	...	...	...	...
<b>2700</b>	195	NaN	EVGA SuperNOVA 1000 T2 220- T2-1000-X1 80+ TITA...	NaN	NaN	Power
<b>2701</b>	196	NaN	Thermaltake TR2 TR-600 600W ATX12V v2.3 SLI Re...	NaN	NaN	Power

### Changing the name of the columns to easily identify

```
data.columns = ['Item_No', 'Brand_Name', 'Item_Name', 'Ratings', 'Price', 'Category']
data
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
0	0	AMD	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	14,723.99	CPU
1	1	AMD	Kingston 16GB (2 x 8GB) 240-Pin DDR3 SDRAM DDR...	(9)	378.99	CPU
2	2	AMD	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(4)	2,834.99	CPU
3	3	AMD	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(42)	661.99	CPU
4	4	AMD	AMD Ryzen Threadripper 3990X 64-Core 2.9 GHz S...	(691)	1,588.99	CPU
...	...	...	...	...	...	...
2700	195	NaN	EVGA SuperNOVA 1000 T2 220- T2-1000-X1 80+ TITA...	NaN	NaN	Power
2701	196	NaN	Thermaltake TR2 TR-600 600W ATX12V v2.3 SLI Re...	NaN	NaN	Power

### Checking the amount of CPU, Motherboard, GPU, Monitor, Storage, Ram, Power Components and getting the total number of components

```

cpu_amt = len(data[data['Category'] == "CPU"])
gpu_amt = len(data[data['Category'] == "GPU"])
mobo_amt = len(data[data['Category'] == "Motherboard"])
mon_amt = len(data[data['Category'] == "Monitor"])
stg_amt = len(data[data['Category'] == "Storage"])
ram_amt = len(data[data['Category'] == "RAM"])
psu_amt = len(data[data['Category'] == "Power"])

print("Number of CPUs:",cpu_amt)
print("Number of GPUs:",gpu_amt)
print("Number of Motherboards:",mobo_amt)
print("Number of Monitors:",mon_amt)
print("Number of Storage:",stg_amt)
print("Number of RAMs:",ram_amt)
print("Number of PSUs:",psu_amt)

total = (cpu_amt + gpu_amt + mobo_amt + mon_amt + stg_amt + ram_amt + psu_amt)
print("=====")
print("Total Number of Components:",total)

```

Number of CPUs: 135  
 Number of GPUs: 410  
 Number of Motherboards: 440

```
Number of Monitors: 920
Number of Storage: 440
Number of RAMs: 160
Number of PSUs: 200
=====
Total Number of Components: 2705
```

## Identifying the column names of the dataset

```
col_num = 1
for x in data.columns:
    print(f"{col_num}.",x)
    col_num+=1
```

1. Item\_No
2. Brand\_Name
3. Item\_Name
4. Ratings
5. Price
6. Category

## ✓ Creating new dataframe for each category

### Dataframe for CPUs

```
cpu_df = pd.DataFrame(data)
cpu_dataframe = cpu_df[cpu_df['Category'] == 'CPU'].copy()
cpu_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
<b>0</b>	0	AMD	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	14,723.99	CPU
<b>1</b>	1	AMD	Kingston 16GB (2 x 8GB) 240-Pin DDR3 SDRAM DDR...	(9)	378.99	CPU
<b>2</b>	2	AMD	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(4)	2,834.99	CPU
<b>3</b>	3	AMD	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(42)	661.99	CPU
<b>4</b>	4	AMD	AMD Ryzen Threadripper 3990X 64- Core 2.9 GHz S...	(691)	1,588.99	CPU
...	...	...	...	...	...	...
<b>130</b>	130	NaN	Refurbished: AMD Athlon 64 X2 4200+ Brisbane D...	NaN	NaN	CPU
<b>131</b>	131	NaN	Refurbished: Intel Core 2 Duo E7200 Wolfdale-3...	NaN	NaN	CPU

### Dataframe for GPUs

```
gpu_df = pd.DataFrame(data)
gpu_dataframe = gpu_df[gpu_df['Category'] == 'GPU'].copy()
gpu_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
135	0	Sapphire Tech	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	1,058.99	GPU
136	1	MSI	Kingston 16GB (2 x 8GB) 240-Pin DDR3 SDRAM DDR...	(2)	1,542.99	GPU
137	2	GIGABYTE	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(5)	1,096.99	GPU
138	3	Sapphire Tech	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(11)	586.99	GPU
139	4	EVGA	SAPPHIRE PULSE Radeon RX 5600 XT DirectX 12 10...	(1)	1,208.99	GPU
...	...	...	...	...	...	...
540	405	NaN	Sapphire - 11295-01-20G - Video Card 11295-01-...	NaN	NaN	GPU
541	406	NaN	GIGABYTE AMD Radeon RX Vega 56 DirectX 12 8GB ...	NaN	NaN	GPU

### Dataframe for Motherboards

```
mobo_df = pd.DataFrame(data)
mobo_dataframe = mobo_df[mobo_df['Category'] == 'Motherboard'].copy()
mobo_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
545	0	MSI	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	1,397.99	Motherboard
546	1	MSI	Kingston 16GB (2 x 8GB) 240-Pin DDR3 SDRAM DDR...	(148)	1,133.99	Motherboard
547	2	MSI	CORSAIR Vengeance LPX 32GB (4 x 8GB) 288-Pin D...	(12)	434.99	Motherboard
548	3	ASUS	CORSAIR Vengeance LPX 128GB (4 x 32GB) 288-Pin...	(128)	359.99	Motherboard
549	4	MSI	MSI MEG X570 ACE Gaming Motherboard AMD AM4 SA...	(9)	283.99	Motherboard
...	...	...	...	...	...	...
980	435	NaN	Refurbished: GIGABYTE Z370XP SLI (rev. 1.0) LG...	NaN	NaN	Motherboard
981	436	NaN	Refurbished: MSI Z170A GAMING M3 LGA 1151 Inte...	NaN	NaN	Motherboard

## Dataframe for Monitors

```
mon_df = pd.DataFrame(data)
mon_dataframe = mon_df[mon_df['Category'] == 'Monitor'].copy()
mon_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
985	0	MSI	Acer XZ271U Abmiiiphzx 27" Quad HD 2560 x 1440 ...	(157)	1,020.99	Monitor
986	1	MSI	MSI Optix AG32C 32" Red LED Non- Glare Super Na...	(28)	944.99	Monitor
987	2	MSI	MSI Optix MAG272QR 27" WQHD 2560 x 1440 (2K) 1...	(417)	1,322.99	Monitor
988	3	MSI	MSI Optix MAG27C 27" Full HD 1920 x 1080 1ms (...)	(157)	680.99	Monitor
989	4	ASUS	MSI Optix AG32C 32" Red LED Non- Glare Super Na...	(417)	676.99	Monitor
...	...	...	...	...	...	...
1900	915	NaN	Lenovo ThinkVision T24i-19 23.8" Full HD VGA D...	NaN	NaN	Monitor
1901	916	NaN	Dell E2720HS 27" 1920x1080 Full HD LED IPS 5ms...	NaN	NaN	Monitor

## Dataframe for Storages

```
stg_df = pd.DataFrame(data)
stg_dataframe = stg_df[stg_df['Category'] == 'Storage'].copy()
stg_dataframe
```



	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
1905	0	Seagate	StarTech.com USB3C2ESAT3 3 ft 1m USB C to eSAT...	(1)	1,131.99	Storage
1906	1	Seagate	CRU 31350-1279-0000 Usb 3.0 Writeblocker; Bloc...	(187)	574.99	Storage
1907	2	Seagate	SanDisk 256GB Ultra SDXC UHS- I/Class 10 Memory...	(9)	536.99	Storage
1908	3	Seagate	Corsair 110Q CC-9011184-WWW Black Steel / Plast...	(644)	1,919.99	Storage
1909	4	Seagate	Seagate Technology ST12000NM001G Hard Drive 12...	(644)	1,065.99	Storage
...	...	...	...	...	...	...
2340	435	NaN	Western Digital Blue WD10EALX 1TB 7200 RPM 32M...	NaN	NaN	Storage
2341	436	NaN	Lenovo 1TB PCI-Express 3.0 x4 NVME TLC Interna...	NaN	NaN	Storage
...	...	...	...	...	...	...

### Dataframe for RAMs

```
ram_df = pd.DataFrame(data)
ram_dataframe = ram_df[ram_df['Category'] == 'RAM'].copy()
ram_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
2345	0	Corsair	Thermaltake Level 20 RS Motherboard Sync ARGB ...	(43)	415.99	RAM
2346	1	Kingston Technology Corp.	Rosewill SRM-01B-450 Micro ATX Mini Tower Desk...	(103)	432.16	RAM
2347	2	Corsair	Antec Performance Series P82 Flow ATX Mid-Towe...	(86)	653.99	RAM
2348	3	Corsair	Phanteks Eclipse P300A High Airflow Full-Metal...	(10)	2,380.99	RAM
2349	4	Corsair	CORSAIR Vengeance RGB Pro 16GB (2 x 8GB) 288-P...	(1)	733.99	RAM
...	...	...	...	...	...	...
2500	155	NaN	CORSAIR Vengeance 8GB (2 x 4GB) 240-Pin DDR3 S...	(2)	NaN	RAM
2501	156	NaN	G.SKILL TridentZ RGB Series 32GB (2 x 16GB) 28...	NaN	NaN	RAM

### Dataframe for Power Supplies

```
psu_df = pd.DataFrame(data)
psu_dataframe = psu_df[psu_df['Category'] == 'Power'].copy()
psu_dataframe
```

	Item_No	Brand_Name	Item_Name	Ratings	Price	Category
2505	0	EVGA	EVGA SuperNOVA 850 GA, 80 Plus Gold 850W, Full...	(1)	491.99	Power
2506	1	EVGA	EVGA 850 B5, 80 Plus BRONZE 850W, Fully Modula...	(1)	NaN	Power
2507	2	EVGA	EVGA SuperNOVA 750 G5, 80 Plus Gold 750W, Full...	(47)	567.99	Power
2508	3	EVGA	EVGA SuperNOVA 550 GA, 80 Plus Gold 550W, Full...	(1)	720.12	Power
2509	4	EVGA	EVGA SuperNOVA 850 GA, 80 Plus Gold 850W, Full...	(1)	490.99	Power
...	...	...	...	...	...	...
2700	195	NaN	EVGA SuperNOVA 1000 T2 220-T2-1000-X1 80+ TITA...	NaN	NaN	Power
2701	196	NaN	Thermaltake TR2 TR-600 600W ATX12V v2.3 SLI Re...	NaN	NaN	Power

## ✓ Deleting the Items with no price and getting its mean price

### CPU (Central Processing Unit)

```
cpu_dataframe.loc[:, 'Price'] = pd.to_numeric(cpu_dataframe.Price, errors = 'coerce')
cpu_dataframe = cpu_dataframe.dropna()
cpu_dataframe

res = 0
total_row = len(cpu_dataframe)
for x in cpu_dataframe['Price']:
    res += x

mean_price_cpu = round((res/total_row), 2)

print("Average Price of CPU Components:", mean_price_cpu)
```

Average Price of CPU Components: 538.23

### GPU (Graphics Processing Unit)

```
gpu_dataframe.loc[:, 'Price'] = pd.to_numeric(gpu_dataframe.Price, errors = 'coerce')
gpu_dataframe = gpu_dataframe.dropna()
gpu_dataframe

res2 = 0
total_row2 = len(gpu_dataframe)
for x in gpu_dataframe['Price']:
    res2 += x

mean_price_gpu = round((res2/total_row2),2)
print("Average Price of GPU Components:", mean_price_gpu)
```

Average Price of GPU Components: 622.16

## Motherboard

```
mobo_dataframe.loc[:, 'Price'] = pd.to_numeric(mobo_dataframe.Price, errors = 'coerce')
mobo_dataframe = mobo_dataframe.dropna()
mobo_dataframe

res3 = 0
total_row3 = len(mobo_dataframe)
for x in mobo_dataframe['Price']:
    res3 += x

mean_price_mobo = round((res3/total_row3),2)
print("Average Price of Motherboard Components:", mean_price_mobo)
```

Average Price of Motherboard Components: 538.76

## Monitor

```
mon_dataframe.loc[:, 'Price'] = pd.to_numeric(mon_dataframe.Price, errors = 'coerce')
mon_dataframe = mon_dataframe.dropna()
mon_dataframe

res4 = 0
total_row4 = len(mon_dataframe)
for x in mon_dataframe['Price']:
    res4 += x

mean_price_mon = round((res4/total_row4),2)
print("Average Price of Monitor Components:", mean_price_mon)
```

Average Price of Monitor Components: 620.92

## Storage

```
stg_dataframe.loc[:, 'Price'] = pd.to_numeric(stg_dataframe.Price, errors = 'coerce')
stg_dataframe = stg_dataframe.dropna()
stg_dataframe

res5 = 0
total_row5 = len(stg_dataframe)
for x in stg_dataframe['Price']:
    res5 += x

mean_price_stg = round((res5/total_row5),2)
print("Average Price of Storage Components:", mean_price_stg)
```