In [36]:
```
import pandas as pd
car_sales = pd.read_csv("data/car-sales.csv")
```

In [37]:
```
# Convert String column to lower case
car_sales["Make"] = car_sales["Make"].str.lower()
car_sales
```

Out[37]:

	Make	Colour	Odometer (KM)	Doors	Price
0	toyota	White	150043	4	$4,000.00
1	honda	Red	87899	4	$5,000.00
2	toyota	Blue	32549	3	$7,000.00
3	bmw	Black	11179	5	$22,000.00
4	nissan	White	213095	4	$3,500.00
5	toyota	Green	99213	4	$4,500.00
6	honda	Blue	45698	4	$7,500.00
7	honda	Blue	54738	4	$7,000.00
8	toyota	White	60000	4	$6,250.00
9	nissan	White	31600	4	$9,700.00

In [38]:
```
# Open car-sales-missing-data.csv file (It contains some missing data)
car_sales_missing = pd.read_csv("data/car-sales-missing-data.csv")
car_sales_missing.head()
```

Out[38]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.0	4.0	$4,000
1	Honda	Red	87899.0	4.0	$5,000
2	Toyota	Blue	NaN	3.0	$7,000
3	BMW	Black	11179.0	5.0	$22,000
4	Nissan	White	213095.0	4.0	$3,500

In [39]:
```
# Replace the missing values in Odometer column by the mean value of that column
car_sales_missing["Odometer"] = car_sales_missing["Odometer"].fillna(car_sales_missing[
car_sales_missing.head()


# Without assign into same column we can use inplace parameter to replace Nan values au
# car_sales_missing["Odometer"].fillna(car_sales_missing["Odometer"].mean(), inplace=Tr
```

Out[39]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000

	Make	Colour	Odometer	Doors	Price
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500

In [40]:
```
# Drop rows which having Nan in Odometer column
# We use inplace=True to overwrite the table
car_sales_missing["Odometer"].dropna(inplace=True)
car_sales_missing
```

Out[40]:

	Make	Colour	Odometer	Doors	Price
0	Toyota	White	150043.000000	4.0	$4,000
1	Honda	Red	87899.000000	4.0	$5,000
2	Toyota	Blue	92302.666667	3.0	$7,000
3	BMW	Black	11179.000000	5.0	$22,000
4	Nissan	White	213095.000000	4.0	$3,500
5	Toyota	Green	92302.666667	4.0	$4,500
6	Honda	NaN	92302.666667	4.0	$7,500
7	Honda	Blue	92302.666667	4.0	NaN
8	Toyota	White	60000.000000	NaN	NaN
9	NaN	White	31600.000000	4.0	$9,700

In [41]:
```
car_sales = pd.read_csv("data/car-sales.csv")

# Create new Series
seates_column = pd.Series([3, 4, 5, 6, 7])
# Add that Series to car_sales DataFrame as a column
car_sales["Seats"] = seates_column
car_sales # After row 5 all the values will be NaN
```

Out[41]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	Toyota	White	150043	4	$4,000.00	3.0
1	Honda	Red	87899	4	$5,000.00	4.0
2	Toyota	Blue	32549	3	$7,000.00	5.0
3	BMW	Black	11179	5	$22,000.00	6.0
4	Nissan	White	213095	4	$3,500.00	7.0
5	Toyota	Green	99213	4	$4,500.00	NaN
6	Honda	Blue	45698	4	$7,500.00	NaN

	Make	Colour	Odometer (KM)	Doors	Price	Seats
7	Honda	Blue	54738	4	$7,000.00	NaN
8	Toyota	White	60000	4	$6,250.00	NaN
9	Nissan	White	31600	4	$9,700.00	NaN

In [42]:
```
# Replace the NaN values by 5
car_sales.fillna(5, inplace=True)
car_sales
```

Out[42]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats
0	Toyota	White	150043	4	$4,000.00	3.0
1	Honda	Red	87899	4	$5,000.00	4.0
2	Toyota	Blue	32549	3	$7,000.00	5.0
3	BMW	Black	11179	5	$22,000.00	6.0
4	Nissan	White	213095	4	$3,500.00	7.0
5	Toyota	Green	99213	4	$4,500.00	5.0
6	Honda	Blue	45698	4	$7,500.00	5.0
7	Honda	Blue	54738	4	$7,000.00	5.0
8	Toyota	White	60000	4	$6,250.00	5.0
9	Nissan	White	31600	4	$9,700.00	5.0

In [43]:
```
# To Add a Column to a DataFrame using a list (Not a Series)
# length of the list should equal to number of rows
fuel_economy = [1.5, 2.6, 3.4, 2.5, 1.2, 2.3, 1.4, 4.5, 6.3, 2.5]
car_sales["Fuel Economy"] = fuel_economy
car_sales
```

Out[43]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel Economy
0	Toyota	White	150043	4	$4,000.00	3.0	1.5
1	Honda	Red	87899	4	$5,000.00	4.0	2.6
2	Toyota	Blue	32549	3	$7,000.00	5.0	3.4
3	BMW	Black	11179	5	$22,000.00	6.0	2.5
4	Nissan	White	213095	4	$3,500.00	7.0	1.2
5	Toyota	Green	99213	4	$4,500.00	5.0	2.3
6	Honda	Blue	45698	4	$7,500.00	5.0	1.4
7	Honda	Blue	54738	4	$7,000.00	5.0	4.5
8	Toyota	White	60000	4	$6,250.00	5.0	6.3
9	Nissan	White	31600	4	$9,700.00	5.0	2.5

In [44]:
```
# Creating a new column and assign 1 value to all the rows
car_sales["Wheels"] = 4
car_sales
```

Out[44]:

	Make	Colour	Odometer (KM)	Doors	Price	Seats	Fuel Economy	Wheels
0	Toyota	White	150043	4	$4,000.00	3.0	1.5	4
1	Honda	Red	87899	4	$5,000.00	4.0	2.6	4
2	Toyota	Blue	32549	3	$7,000.00	5.0	3.4	4
3	BMW	Black	11179	5	$22,000.00	6.0	2.5	4
4	Nissan	White	213095	4	$3,500.00	7.0	1.2	4
5	Toyota	Green	99213	4	$4,500.00	5.0	2.3	4
6	Honda	Blue	45698	4	$7,500.00	5.0	1.4	4
7	Honda	Blue	54738	4	$7,000.00	5.0	4.5	4
8	Toyota	White	60000	4	$6,250.00	5.0	6.3	4
9	Nissan	White	31600	4	$9,700.00	5.0	2.5	4

In [45]:
```
# Remove Seats column
# axis=1 is used to indicate that we are refering to a column (axis=0 => row)
# inplace=True => Overwrite the DataFrame
car_sales.drop("Seats", axis=1, inplace=True)
car_sales
```

Out[45]:

	Make	Colour	Odometer (KM)	Doors	Price	Fuel Economy	Wheels
0	Toyota	White	150043	4	$4,000.00	1.5	4
1	Honda	Red	87899	4	$5,000.00	2.6	4
2	Toyota	Blue	32549	3	$7,000.00	3.4	4
3	BMW	Black	11179	5	$22,000.00	2.5	4
4	Nissan	White	213095	4	$3,500.00	1.2	4
5	Toyota	Green	99213	4	$4,500.00	2.3	4
6	Honda	Blue	45698	4	$7,500.00	1.4	4
7	Honda	Blue	54738	4	$7,000.00	4.5	4
8	Toyota	White	60000	4	$6,250.00	6.3	4
9	Nissan	White	31600	4	$9,700.00	2.5	4

In [46]:
```
# Shuffle the DataFrame by rows
# frac=1 => get 100% of rows
car_sales_shuffled = car_sales.sample(frac=1)
car_sales_shuffled
```

Out[46]:

	Make	Colour	Odometer (KM)	Doors	Price	Fuel Economy	Wheels
3	BMW	Black	11179	5	$22,000.00	2.5	4
2	Toyota	Blue	32549	3	$7,000.00	3.4	4
4	Nissan	White	213095	4	$3,500.00	1.2	4
6	Honda	Blue	45698	4	$7,500.00	1.4	4
0	Toyota	White	150043	4	$4,000.00	1.5	4
8	Toyota	White	60000	4	$6,250.00	6.3	4
5	Toyota	Green	99213	4	$4,500.00	2.3	4
1	Honda	Red	87899	4	$5,000.00	2.6	4
9	Nissan	White	31600	4	$9,700.00	2.5	4
7	Honda	Blue	54738	4	$7,000.00	4.5	4

In [47]:

```
# Get only 25% of data by shuffling (get randomly)
car_sales.sample(frac=0.25)
```

Out[47]:

	Make	Colour	Odometer (KM)	Doors	Price	Fuel Economy	Wheels
9	Nissan	White	31600	4	$9,700.00	2.5	4
1	Honda	Red	87899	4	$5,000.00	2.6	4

In [48]:

```
# To reset the shuffled DataFrame
# drop=True => Avoid getting 2 index columns
car_sales_shuffled.reset_index(drop=True)
```

Out[48]:

	Make	Colour	Odometer (KM)	Doors	Price	Fuel Economy	Wheels
0	BMW	Black	11179	5	$22,000.00	2.5	4
1	Toyota	Blue	32549	3	$7,000.00	3.4	4
2	Nissan	White	213095	4	$3,500.00	1.2	4
3	Honda	Blue	45698	4	$7,500.00	1.4	4
4	Toyota	White	150043	4	$4,000.00	1.5	4
5	Toyota	White	60000	4	$6,250.00	6.3	4
6	Toyota	Green	99213	4	$4,500.00	2.3	4
7	Honda	Red	87899	4	$5,000.00	2.6	4
8	Nissan	White	31600	4	$9,700.00	2.5	4
9	Honda	Blue	54738	4	$7,000.00	4.5	4

In [49]:

```
# Convert Odometer KM Column to Miles by values by 1.6
# Use lambda function
```

```
car_sales["Odometer (KM)"] = car_sales["Odometer (KM)"].apply(lambda x:x/1.6)
car_sales
```

Out[49]:

	Make	Colour	Odometer (KM)	Doors	Price	Fuel Economy	Wheels
0	Toyota	White	93776.875	4	$4,000.00	1.5	4
1	Honda	Red	54936.875	4	$5,000.00	2.6	4
2	Toyota	Blue	20343.125	3	$7,000.00	3.4	4
3	BMW	Black	6986.875	5	$22,000.00	2.5	4
4	Nissan	White	133184.375	4	$3,500.00	1.2	4
5	Toyota	Green	62008.125	4	$4,500.00	2.3	4
6	Honda	Blue	28561.250	4	$7,500.00	1.4	4
7	Honda	Blue	34211.250	4	$7,000.00	4.5	4
8	Toyota	White	37500.000	4	$6,250.00	6.3	4
9	Nissan	White	19750.000	4	$9,700.00	2.5	4