Analysing columns

Problem 1: Import MPG dataset abd store as the pandas dataframe with name *mpg*

```
import pandas as pd

mpg =
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/MPG.cs
v")

mpg
```

```
      mpg  cylinders  displacement  horsepower  weight
acceleration  \
0    18.0          8         307.0       130.0    3504          12.0

1    15.0          8         350.0       165.0    3693          11.5

2    18.0          8         318.0       150.0    3436          11.0

3    16.0          8         304.0       150.0    3433          12.0

4    17.0          8         302.0       140.0    3449          10.5

..    ...        ...           ...         ...     ...           ...

393  27.0          4         140.0        86.0    2790          15.6

394  44.0          4          97.0        52.0    2130          24.6

395  32.0          4         135.0        84.0    2295          11.6

396  28.0          4         120.0        79.0    2625          18.6

397  31.0          4         119.0        82.0    2720          19.4


     model_year  origin                       name
0            70     usa  chevrolet chevelle malibu
1            70     usa          buick skylark 320
2            70     usa         plymouth satellite
3            70     usa              amc rebel sst
4            70     usa                ford torino
..          ...     ...                        ...
393          82     usa            ford mustang gl
394          82  europe                  vw pickup
395          82     usa              dodge rampage
396          82     usa                ford ranger
397          82     usa                 chevy s-10
```

```
[398 rows x 9 columns]
```

Problem 2: Copy MPG dataframe as car

```
car = mpg.copy()

car

      mpg  cylinders  displacement  horsepower  weight  
acceleration  \
0    18.0          8         307.0       130.0    3504          12.0   

1    15.0          8         350.0       165.0    3693          11.5   

2    18.0          8         318.0       150.0    3436          11.0   

3    16.0          8         304.0       150.0    3433          12.0   

4    17.0          8         302.0       140.0    3449          10.5   

..    ...        ...           ...         ...     ...           ...   

393  27.0          4         140.0        86.0    2790          15.6   

394  44.0          4          97.0        52.0    2130          24.6   

395  32.0          4         135.0        84.0    2295          11.6   

396  28.0          4         120.0        79.0    2625          18.6   

397  31.0          4         119.0        82.0    2720          19.4   


     model_year  origin                       name  
0            70     usa  chevrolet chevelle malibu  
1            70     usa          buick skylark 320  
2            70     usa         plymouth satellite  
3            70     usa              amc rebel sst  
4            70     usa                ford torino  
..          ...     ...                        ...  
393          82     usa            ford mustang gl  
394          82  europe                  vw pickup  
395          82     usa              dodge rampage  
396          82     usa                ford ranger  
397          82     usa                 chevy s-10  

[398 rows x 9 columns]
```

Problem 3: Drop column name cylinders from original dataframe (mpg) and inspect what happened to the copy(car).

```
mpg = mpg.drop("cylinders", axis = 1)
```

```
mpg.columns
```

```
Index(['mpg', 'displacement', 'horsepower', 'weight', 'acceleration',
       'model_year', 'origin', 'name'],
      dtype='object')
```

```
car.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model_year', 'origin', 'name'],
      dtype='object')
```

Note :No changes in cars dataframe

Problem 4: Analysing car dataframe

```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64  
 7   origin        398 non-null    object 
 8   name          398 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

```
car.describe()
```

```
              mpg   cylinders  displacement  horsepower  
weight  \
count  398.000000  398.000000    398.000000  392.000000   398.000000

mean    23.514573    5.454774    193.425879  104.469388  2970.424623

std      7.815984    1.701004    104.269838   38.491160   846.841774

min      9.000000    3.000000     68.000000   46.000000  1613.000000
```

```
25%     17.500000    4.000000    104.250000   75.000000  2223.750000

50%     23.000000    4.000000    148.500000   93.500000  2803.500000

75%     29.000000    8.000000    262.000000  126.000000  3608.000000

max     46.600000    8.000000    455.000000  230.000000  5140.000000
```

```
       acceleration  model_year
count    398.000000  398.000000
mean      15.568090   76.010050
std        2.757689    3.697627
min        8.000000   70.000000
25%       13.825000   73.000000
50%       15.500000   76.000000
75%       17.175000   79.000000
max       24.800000   82.000000
```

Problem 5: Provide unique values in each columns cylinders and origin

```
car[["cylinders","origin"]].value_counts()
```

```
cylinders  origin
8          usa       103
6          usa        74
4          usa        72
           japan      69
           europe     63
6          japan       6
3          japan       4
6          europe      4
5          europe      3
dtype: int64
```

Problem 6: Provide unique values of column origin

```
car[["origin"]].value_counts()
```

```
origin
usa       249
japan      79
europe     70
dtype: int64
```

```
car["origin"].unique()
```

```
array(['usa', 'japan', 'europe'], dtype=object)
```

```
car["origin"].nunique()
```

```
3
```

Problem 7: Sort value car dataframe as per displacement column

```
car.displacement
```

```
0      307.0
1      350.0
2      318.0
3      304.0
4      302.0
       ...
393    140.0
394     97.0
395    135.0
396    120.0
397    119.0
Name: displacement, Length: 398, dtype: float64
```

```
# Now sorting the values

car.sort_values("displacement")
```

```
      mpg  cylinders  displacement  horsepower  weight
acceleration  \
117  29.0          4          68.0        49.0    1867          19.5

71   19.0          3          70.0        97.0    2330          13.5

111  18.0          3          70.0        90.0    2124          13.5

334  23.7          3          70.0       100.0    2420          12.5

131  32.0          4          71.0        65.0    1836          21.0

..    ...        ...           ...         ...     ...           ...

94   13.0          8         440.0       215.0    4735          11.0

6    14.0          8         454.0       220.0    4354           9.0

95   12.0          8         455.0       225.0    4951          11.0

8    14.0          8         455.0       225.0    4425          10.0

13   14.0          8         455.0       225.0    3086          10.0


     model_year  origin                         name
117          73  europe                     fiat 128
71           72   japan               mazda rx2 coupe
111          73   japan                    maxda rx3
```

```
334          80   japan                  mazda rx-7 gs
131          74   japan            toyota corolla 1200
..          ...     ...                            ...
94           73     usa  chrysler new yorker brougham
6            70     usa               chevrolet impala
95           73     usa       buick electra 225 custom
8            70     usa               pontiac catalina
13           70     usa         buick estate wagon (sw)

[398 rows x 9 columns]
```

Problem 8: Sort value of car dataframe as per displacement column in descending order.

```
car.sort_values("displacement",  ascending = False)
```

```
      mpg  cylinders  displacement  horsepower  weight  
acceleration  \
8    14.0          8         455.0       225.0    4425          10.0   

95   12.0          8         455.0       225.0    4951          11.0   

13   14.0          8         455.0       225.0    3086          10.0   

6    14.0          8         454.0       220.0    4354           9.0   

7    14.0          8         440.0       215.0    4312           8.5   

..    ...        ...           ...         ...     ...           ...   

131  32.0          4          71.0        65.0    1836          21.0   

111  18.0          3          70.0        90.0    2124          13.5   

71   19.0          3          70.0        97.0    2330          13.5   

334  23.7          3          70.0       100.0    2420          12.5   

117  29.0          4          68.0        49.0    1867          19.5   


     model_year  origin                      name  
8            70     usa          pontiac catalina  
95           73     usa  buick electra 225 custom  
13           70     usa   buick estate wagon (sw)  
6            70     usa          chevrolet impala  
7            70     usa         plymouth fury iii  
..          ...     ...                       ...  
131          74   japan       toyota corolla 1200  
111          73   japan                 maxda rx3  
```

```
71           72   japan           mazda rx2 coupe
334          80   japan             mazda rx-7 gs
117          73  europe                  fiat 128

[398 rows x 9 columns]
```

Problem 9: Sort value of car dataframe as per displacement and weight columns in descending order

```
car.sort_values(["displacement", "weight"], ascending = False)
```

```
      mpg  cylinders  displacement  horsepower  weight
acceleration  \
95   12.0          8         455.0       225.0    4951          11.0

8    14.0          8         455.0       225.0    4425          10.0

13   14.0          8         455.0       225.0    3086          10.0

6    14.0          8         454.0       220.0    4354           9.0

94   13.0          8         440.0       215.0    4735          11.0

..    ...        ...           ...         ...     ...           ...

53   31.0          4          71.0        65.0    1773          19.0

334  23.7          3          70.0       100.0    2420          12.5

71   19.0          3          70.0        97.0    2330          13.5

111  18.0          3          70.0        90.0    2124          13.5

117  29.0          4          68.0        49.0    1867          19.5


     model_year  origin                          name
95           73     usa      buick electra 225 custom
8            70     usa              pontiac catalina
13           70     usa       buick estate wagon (sw)
6            70     usa              chevrolet impala
94           73     usa  chrysler new yorker brougham
..          ...     ...                           ...
53           71   japan           toyota corolla 1200
334          80   japan                 mazda rx-7 gs
71           72   japan               mazda rx2 coupe
111          73   japan                     maxda rx3
117          73  europe                      fiat 128
```

```
[398 rows x 9 columns]
```

Problem 10: Summary statistics of all columns

```
car.describe(include = "all")
```

```
               mpg   cylinders  displacement  horsepower       weight  
\
count   398.000000  398.000000    398.000000  392.000000   398.000000   

unique         NaN         NaN           NaN         NaN          NaN   

top            NaN         NaN           NaN         NaN          NaN   

freq           NaN         NaN           NaN         NaN          NaN   

mean     23.514573    5.454774    193.425879  104.469388  2970.424623   

std       7.815984    1.701004    104.269838   38.491160   846.841774   

min       9.000000    3.000000     68.000000   46.000000  1613.000000   

25%      17.500000    4.000000    104.250000   75.000000  2223.750000   

50%      23.000000    4.000000    148.500000   93.500000  2803.500000   

75%      29.000000    8.000000    262.000000  126.000000  3608.000000   

max      46.600000    8.000000    455.000000  230.000000  5140.000000   


        acceleration  model_year origin        name  
count     398.000000  398.000000    398         398  
unique           NaN         NaN      3         305  
top              NaN         NaN    usa  ford pinto  
freq             NaN         NaN    249           6  
mean       15.568090   76.010050    NaN         NaN  
std         2.757689    3.697627    NaN         NaN  
min         8.000000   70.000000    NaN         NaN  
25%        13.825000   73.000000    NaN         NaN  
50%        15.500000   76.000000    NaN         NaN  
75%        17.175000   79.000000    NaN         NaN  
max        24.800000   82.000000    NaN         NaN  
```

Problem 11: Transpose of dataframe

```
car.T
```

```
                                    0                  1    \
mpg                               18.0               15.0   
cylinders                            8                  8   
displacement                     307.0              350.0   
horsepower                       130.0              165.0   
weight                            3504               3693   
acceleration                      12.0               11.5   
model_year                          70                 70   
origin                             usa                usa   
name          chevrolet chevelle malibu  buick skylark 320   

                             2              3            4    \
mpg                       18.0           16.0         17.0   
cylinders                    8              8            8   
displacement             318.0          304.0        302.0   
horsepower               150.0          150.0        140.0   
weight                    3436           3433         3449   
acceleration              11.0           12.0         10.5   
model_year                  70             70           70   
origin                     usa            usa          usa   
name          plymouth satellite  amc rebel sst  ford torino   

                           5                 6                  7    \
mpg                     15.0              14.0               14.0   
cylinders                  8                 8                  8   
displacement           429.0             454.0              440.0   
horsepower             198.0             220.0              215.0   
weight                  4341              4354               4312   
acceleration            10.0               9.0                8.5   
model_year                70                70                 70   
origin                   usa               usa                usa   
name          ford galaxie 500  chevrolet impala  plymouth fury iii   

                           8                   9    ...  \
mpg                     14.0                15.0  ...   
cylinders                  8                   8  ...   
displacement           455.0               390.0  ...   
horsepower             225.0               190.0  ...   
weight                  4425                3850  ...   
acceleration            10.0                 8.5  ...   
model_year                70                  70  ...   
origin                   usa                 usa  ...   
name          pontiac catalina  amc ambassador dpl  ...   

                                     388             389  
390  \
mpg                                 26.0            22.0  
32.0   
cylinders                              4               6  
4   
```

```
displacement                       156.0           232.0
144.0
horsepower                          92.0           112.0
96.0
weight                              2585            2835
2665
acceleration                        14.5            14.7
13.9
model_year                            82              82
82
origin                               usa             usa
japan
name          chrysler lebaron medallion  ford granada l  toyota
celica gt

                            391               392              393
394  \
mpg                        36.0              27.0             27.0
44.0
cylinders                     4                 4                4
4
displacement              135.0             151.0            140.0
97.0
horsepower                 84.0              90.0             86.0
52.0
weight                     2370              2950             2790
2130
acceleration               13.0              17.3             15.6
24.6
model_year                   82                82               82
82
origin                      usa               usa              usa
europe
name          dodge charger 2.2  chevrolet camaro  ford mustang gl  vw
pickup

                        395          396         397
mpg                    32.0         28.0        31.0
cylinders                 4            4           4
displacement          135.0        120.0       119.0
horsepower             84.0         79.0        82.0
weight                 2295         2625        2720
acceleration           11.6         18.6        19.4
model_year               82           82          82
origin                  usa          usa         usa
name          dodge rampage  ford ranger  chevy s-10

[9 rows x 398 columns]
```

Problem 1: Import Tips dataset and store as the pandas dataframe with the name tips.

```
import pandas as pd
```

```
tips =
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/Tips
%20Payment%20Data.csv")
```

Problem 2: Display the first 5 rows of the tips dataframe.

```
tips.head()
```

```
   Total Bill   Tip  Gender Smoker  Day    Time  Size  Bill Per Person
\
0       16.99  1.01  Female     No  Sun  Dinner     2             8.49

1       10.34  1.66    Male     No  Sun  Dinner     3             3.45

2       21.01  3.50    Male     No  Sun  Dinner     3             7.00

3       23.68  3.31    Male     No  Sun  Dinner     2            11.84

4       24.59  3.61  Female     No  Sun  Dinner     4             6.15


           Payer Name         CC Number Payment ID
0  Christy Cunningham  3560325168603410    Sun2959
1      Douglas Tucker  4478071379779230    Sun4608
2      Travis Walters  6011812112971320    Sun4458
3    Nathaniel Harris  4676137647685990    Sun5260
4        Tonya Carter  4832732618637220    Sun2251
```

Problem 3: Calculate percentage of tip to total bill.

Formula : (tip/Total Bill) * 100

```
tips["Tip"]/tips["Total Bill"]*100
```

```
0       5.944673
1      16.054159
2      16.658734
3      13.978041
4      14.680765
         ...
239    20.392697
240     7.358352
241     8.822232
242     9.820426
243    15.974441
Length: 244, dtype: float64
```

Problem 4: Create a new column of percentage tip

```
tip_percentage = tips["Tip"]/tips["Total Bill"]*100
```

```
tip_percentage
```

```
0       5.944673
1      16.054159
2      16.658734
3      13.978041
4      14.680765
         ...
239    20.392697
240     7.358352
241     8.822232
242     9.820426
243    15.974441
Length: 244, dtype: float64
```

Problem 5: Inserting tips_percentage col in tips dataframe

```
tips["tip_percentage"] = tips["Tip"]/tips["Total Bill"]*100
```

```
tips.head()
```

```
   Total Bill   Tip  Gender Smoker  Day    Time  Size  Bill Per Person  
\
0       16.99  1.01  Female     No  Sun  Dinner     2             8.49   

1       10.34  1.66    Male     No  Sun  Dinner     3             3.45   

2       21.01  3.50    Male     No  Sun  Dinner     3             7.00   

3       23.68  3.31    Male     No  Sun  Dinner     2            11.84   

4       24.59  3.61  Female     No  Sun  Dinner     4             6.15   


           Payer Name         CC Number Payment ID  tip_percentage  
0  Christy Cunningham  3560325168603410    Sun2959        5.944673  
1      Douglas Tucker  4478071379779230    Sun4608       16.054159  
2      Travis Walters  6011812112971320    Sun4458       16.658734  
3    Nathaniel Harris  4676137647685990    Sun5260       13.978041  
4        Tonya Carter  4832732618637220    Sun2251       14.680765  
```

Problem 6: Round upto one decimal place the tip_percentage column values.

```
tips["tip_percentage"] = tips["tip_percentage"].round(1)
```

```
tips.head()
```

```
   Total Bill   Tip  Gender Smoker  Day    Time  Size  Bill Per Person
\
0       16.99  1.01  Female     No  Sun  Dinner     2             8.49

1       10.34  1.66    Male     No  Sun  Dinner     3             3.45

2       21.01  3.50    Male     No  Sun  Dinner     3             7.00

3       23.68  3.31    Male     No  Sun  Dinner     2            11.84

4       24.59  3.61  Female     No  Sun  Dinner     4             6.15


           Payer Name         CC Number Payment ID  tip_percentage
0  Christy Cunningham  3560325168603410    Sun2959             5.9
1      Douglas Tucker  4478071379779230    Sun4608            16.1
2      Travis Walters  6011812112971320    Sun4458            16.7
3    Nathaniel Harris  4676137647685990    Sun5260            14.0
4        Tonya Carter  4832732618637220    Sun2251            14.7
```

Problem 7: Drop column Payer Number.

```
tips = tips.drop(["Payer Name"], axis = 1)
```

```
tips.head()
```

```
   Total Bill   Tip  Gender Smoker  Day    Time  Size  Bill Per Person
\
0       16.99  1.01  Female     No  Sun  Dinner     2             8.49

1       10.34  1.66    Male     No  Sun  Dinner     3             3.45

2       21.01  3.50    Male     No  Sun  Dinner     3             7.00

3       23.68  3.31    Male     No  Sun  Dinner     2            11.84

4       24.59  3.61  Female     No  Sun  Dinner     4             6.15


          CC Number Payment ID  tip_percentage
0  3560325168603410    Sun2959             5.9
1  4478071379779230    Sun4608            16.1
2  6011812112971320    Sun4458            16.7
3  4676137647685990    Sun5260            14.0
4  4832732618637220    Sun2251            14.7
```

Problem 8: Index tips dataframe as per Payment ID

```
tips.set_index("Payment ID")
```

```
            Total Bill   Tip  Gender Smoker   Day    Time  Size  \
Payment ID
Sun2959          16.99  1.01  Female     No   Sun  Dinner     2
Sun4608          10.34  1.66    Male     No   Sun  Dinner     3
Sun4458          21.01  3.50    Male     No   Sun  Dinner     3
Sun5260          23.68  3.31    Male     No   Sun  Dinner     2
Sun2251          24.59  3.61  Female     No   Sun  Dinner     4
...                ...   ...     ...    ...   ...     ...   ...
Sat2657          29.03  5.92    Male     No   Sat  Dinner     3
Sat1766          27.18  2.00  Female    Yes   Sat  Dinner     2
Sat3880          22.67  2.00    Male    Yes   Sat  Dinner     2
Sat17            17.82  1.75    Male     No   Sat  Dinner     2
Thur672          18.78  3.00  Female     No  Thur  Dinner     2

            Bill Per Person         CC Number  tip_percentage
Payment ID
Sun2959                8.49  3560325168603410             5.9
Sun4608                3.45  4478071379779230            16.1
Sun4458                7.00  6011812112971320            16.7
Sun5260               11.84  4676137647685990            14.0
Sun2251                6.15  4832732618637220            14.7
...                     ...               ...             ...
Sat2657                9.68  5296068606052840            20.4
Sat1766               13.59  3506806155565400             7.4
Sat3880               11.34  6011891618747190             8.8
Sat17                  8.91     4375220550950             9.8
Thur672                9.39  3511451626698130            16.0

[244 rows x 10 columns]
```

```
tips.head()
```

```
   Total Bill   Tip  Gender Smoker  Day    Time  Size  Bill Per Person
\
0       16.99  1.01  Female     No  Sun  Dinner     2             8.49

1       10.34  1.66    Male     No  Sun  Dinner     3             3.45

2       21.01  3.50    Male     No  Sun  Dinner     3             7.00

3       23.68  3.31    Male     No  Sun  Dinner     2            11.84

4       24.59  3.61  Female     No  Sun  Dinner     4             6.15


          CC Number Payment ID  tip_percentage
0  3560325168603410    Sun2959             5.9
1  4478071379779230    Sun4608            16.1
2  6011812112971320    Sun4458            16.7
```

```
3  4676137647685990    Sun5260           14.0
4  4832732618637220    Sun2251           14.7
```

Problem 9: Change index tips dataframe as per Payment ID

```
tips = tips.set_index("Payment ID")
```

```
tips.head()
```

```
            Total Bill   Tip  Gender Smoker  Day    Time  Size  \
Payment ID
Sun2959          16.99  1.01  Female     No  Sun  Dinner     2
Sun4608          10.34  1.66    Male     No  Sun  Dinner     3
Sun4458          21.01  3.50    Male     No  Sun  Dinner     3
Sun5260          23.68  3.31    Male     No  Sun  Dinner     2
Sun2251          24.59  3.61  Female     No  Sun  Dinner     4

            Bill Per Person         CC Number  tip_percentage
Payment ID
Sun2959                8.49  3560325168603410             5.9
Sun4608                3.45  4478071379779230            16.1
Sun4458                7.00  6011812112971320            16.7
Sun5260               11.84  4676137647685990            14.0
Sun2251                6.15  4832732618637220            14.7
```

Eg for Locating row by payment id.

```
tips.loc["Sun4608"]
```

```
Total Bill                    10.34
Tip                            1.66
Gender                         Male
Smoker                           No
Day                             Sun
Time                         Dinner
Size                              3
Bill Per Person                3.45
CC Number          4478071379779230
tip_percentage                 16.1
Name: Sun4608, dtype: object
```

Problem 10: Reset index of tips dataframe to row index

```
tips = tips.reset_index()
```

```
tips.head()
```

```
  Payment ID  Total Bill   Tip  Gender Smoker  Day    Time  Size  \
0    Sun2959       16.99  1.01  Female     No  Sun  Dinner     2
1    Sun4608       10.34  1.66    Male     No  Sun  Dinner     3
2    Sun4458       21.01  3.50    Male     No  Sun  Dinner     3
3    Sun5260       23.68  3.31    Male     No  Sun  Dinner     2
```

```
4    Sun2251       24.59  3.61  Female     No  Sun  Dinner     4

   Bill Per Person         CC Number  tip_percentage
0             8.49  3560325168603410             5.9
1             3.45  4478071379779230            16.1
2             7.00  6011812112971320            16.7
3            11.84  4676137647685990            14.0
4             6.15  4832732618637220            14.7
```

```
# PRoblem 1, Importing pandas and importing MPG data set.

import pandas as pd

car =
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/MPG.cs
v")
```

Problem 2: Print car data *frame*

```
car

      mpg  cylinders  displacement  horsepower  weight  
acceleration  \
0    18.0          8         307.0       130.0    3504          12.0   

1    15.0          8         350.0       165.0    3693          11.5   

2    18.0          8         318.0       150.0    3436          11.0   

3    16.0          8         304.0       150.0    3433          12.0   

4    17.0          8         302.0       140.0    3449          10.5   

..    ...        ...           ...         ...     ...           ...   

393  27.0          4         140.0        86.0    2790          15.6   

394  44.0          4          97.0        52.0    2130          24.6   

395  32.0          4         135.0        84.0    2295          11.6   

396  28.0          4         120.0        79.0    2625          18.6   

397  31.0          4         119.0        82.0    2720          19.4   


     model_year  origin                       name  
0            70     usa  chevrolet chevelle malibu  
1            70     usa          buick skylark 320  
2            70     usa         plymouth satellite  
3            70     usa              amc rebel sst  
4            70     usa                ford torino  
..          ...     ...                        ...  
393          82     usa            ford mustang gl  
394          82  europe                  vw pickup  
395          82     usa              dodge rampage  
396          82     usa                ford ranger  
397          82     usa                 chevy s-10  
```

```
[398 rows x 9 columns]
```

Problem 3: Print rows of choice.

```
car.head(10)
```

```
    mpg  cylinders  displacement  horsepower  weight  acceleration  \
0  18.0          8         307.0       130.0    3504          12.0   
1  15.0          8         350.0       165.0    3693          11.5   
2  18.0          8         318.0       150.0    3436          11.0   
3  16.0          8         304.0       150.0    3433          12.0   
4  17.0          8         302.0       140.0    3449          10.5   

   model_year origin                       name  
0          70    usa  chevrolet chevelle malibu  
1          70    usa          buick skylark 320  
2          70    usa         plymouth satellite  
3          70    usa              amc rebel sst  
4          70    usa                ford torino  
```

Problem 4: Inspect Last 5 Rows

```
car.tail()
```

```
      mpg  cylinders  displacement  horsepower  weight  
acceleration  \
393  27.0          4         140.0        86.0    2790            15.6   

394  44.0          4          97.0        52.0    2130            24.6   

395  32.0          4         135.0        84.0    2295            11.6   

396  28.0          4         120.0        79.0    2625            18.6   

397  31.0          4         119.0        82.0    2720            19.4   


     model_year  origin             name  
393          82     usa  ford mustang gl  
394          82  europe        vw pickup  
395          82     usa    dodge rampage  
396          82     usa      ford ranger  
397          82     usa       chevy s-10  
```

Problem 5:View all rows.

```
pd.options.display.max_rows = 400
```

```
car
```

```
      mpg  cylinders  displacement  horsepower  weight
acceleration  \
0    18.0          8         307.0       130.0    3504          12.0

1    15.0          8         350.0       165.0    3693          11.5

2    18.0          8         318.0       150.0    3436          11.0

3    16.0          8         304.0       150.0    3433          12.0

4    17.0          8         302.0       140.0    3449          10.5

5    15.0          8         429.0       198.0    4341          10.0

6    14.0          8         454.0       220.0    4354           9.0

7    14.0          8         440.0       215.0    4312           8.5

8    14.0          8         455.0       225.0    4425          10.0

9    15.0          8         390.0       190.0    3850           8.5

10   15.0          8         383.0       170.0    3563          10.0

11   14.0          8         340.0       160.0    3609           8.0

12   15.0          8         400.0       150.0    3761           9.5

13   14.0          8         455.0       225.0    3086          10.0

14   24.0          4         113.0        95.0    2372          15.0

15   22.0          6         198.0        95.0    2833          15.5

16   18.0          6         199.0        97.0    2774          15.5

17   21.0          6         200.0        85.0    2587          16.0

18   27.0          4          97.0        88.0    2130          14.5

19   26.0          4          97.0        46.0    1835          20.5

20   25.0          4         110.0        87.0    2672          17.5

21   24.0          4         107.0        90.0    2430          14.5
```

```
22   25.0          4         104.0        95.0    2375          17.5
23   26.0          4         121.0       113.0    2234          12.5
24   21.0          6         199.0        90.0    2648          15.0
25   10.0          8         360.0       215.0    4615          14.0
26   10.0          8         307.0       200.0    4376          15.0
27   11.0          8         318.0       210.0    4382          13.5
28    9.0          8         304.0       193.0    4732          18.5
29   27.0          4          97.0        88.0    2130          14.5
30   28.0          4         140.0        90.0    2264          15.5
31   25.0          4         113.0        95.0    2228          14.0
32   25.0          4          98.0         NaN    2046          19.0
33   19.0          6         232.0       100.0    2634          13.0
34   16.0          6         225.0       105.0    3439          15.5
35   17.0          6         250.0       100.0    3329          15.5
36   19.0          6         250.0        88.0    3302          15.5
37   18.0          6         232.0       100.0    3288          15.5
38   14.0          8         350.0       165.0    4209          12.0
39   14.0          8         400.0       175.0    4464          11.5
40   14.0          8         351.0       153.0    4154          13.5
41   14.0          8         318.0       150.0    4096          13.0
42   12.0          8         383.0       180.0    4955          11.5
43   13.0          8         400.0       170.0    4746          12.0
44   13.0          8         400.0       175.0    5140          12.0
45   18.0          6         258.0       110.0    2962          13.5
```

46	22.0	4	140.0	72.0	2408	19.0
47	19.0	6	250.0	100.0	3282	15.0
48	18.0	6	250.0	88.0	3139	14.5
49	23.0	4	122.0	86.0	2220	14.0
50	28.0	4	116.0	90.0	2123	14.0
51	30.0	4	79.0	70.0	2074	19.5
52	30.0	4	88.0	76.0	2065	14.5
53	31.0	4	71.0	65.0	1773	19.0
54	35.0	4	72.0	69.0	1613	18.0
55	27.0	4	97.0	60.0	1834	19.0
56	26.0	4	91.0	70.0	1955	20.5
57	24.0	4	113.0	95.0	2278	15.5
58	25.0	4	97.5	80.0	2126	17.0
59	23.0	4	97.0	54.0	2254	23.5
60	20.0	4	140.0	90.0	2408	19.5
61	21.0	4	122.0	86.0	2226	16.5
62	13.0	8	350.0	165.0	4274	12.0
63	14.0	8	400.0	175.0	4385	12.0
64	15.0	8	318.0	150.0	4135	13.5
65	14.0	8	351.0	153.0	4129	13.0
66	17.0	8	304.0	150.0	3672	11.5
67	11.0	8	429.0	208.0	4633	11.0
68	13.0	8	350.0	155.0	4502	13.5
69	12.0	8	350.0	160.0	4456	13.5

```
70   13.0          8         400.0       190.0    4422         12.5
71   19.0          3          70.0        97.0    2330         13.5
72   15.0          8         304.0       150.0    3892         12.5
73   13.0          8         307.0       130.0    4098         14.0
74   13.0          8         302.0       140.0    4294         16.0
75   14.0          8         318.0       150.0    4077         14.0
76   18.0          4         121.0       112.0    2933         14.5
77   22.0          4         121.0        76.0    2511         18.0
78   21.0          4         120.0        87.0    2979         19.5
79   26.0          4          96.0        69.0    2189         18.0
80   22.0          4         122.0        86.0    2395         16.0
81   28.0          4          97.0        92.0    2288         17.0
82   23.0          4         120.0        97.0    2506         14.5
83   28.0          4          98.0        80.0    2164         15.0
84   27.0          4          97.0        88.0    2100         16.5
85   13.0          8         350.0       175.0    4100         13.0
86   14.0          8         304.0       150.0    3672         11.5
87   13.0          8         350.0       145.0    3988         13.0
88   14.0          8         302.0       137.0    4042         14.5
89   15.0          8         318.0       150.0    3777         12.5
90   12.0          8         429.0       198.0    4952         11.5
91   13.0          8         400.0       150.0    4464         12.0
92   13.0          8         351.0       158.0    4363         13.0
93   14.0          8         318.0       150.0    4237         14.5
```

94	13.0	8	440.0	215.0	4735	11.0
95	12.0	8	455.0	225.0	4951	11.0
96	13.0	8	360.0	175.0	3821	11.0
97	18.0	6	225.0	105.0	3121	16.5
98	16.0	6	250.0	100.0	3278	18.0
99	18.0	6	232.0	100.0	2945	16.0
100	18.0	6	250.0	88.0	3021	16.5
101	23.0	6	198.0	95.0	2904	16.0
102	26.0	4	97.0	46.0	1950	21.0
103	11.0	8	400.0	150.0	4997	14.0
104	12.0	8	400.0	167.0	4906	12.5
105	13.0	8	360.0	170.0	4654	13.0
106	12.0	8	350.0	180.0	4499	12.5
107	18.0	6	232.0	100.0	2789	15.0
108	20.0	4	97.0	88.0	2279	19.0
109	21.0	4	140.0	72.0	2401	19.5
110	22.0	4	108.0	94.0	2379	16.5
111	18.0	3	70.0	90.0	2124	13.5
112	19.0	4	122.0	85.0	2310	18.5
113	21.0	6	155.0	107.0	2472	14.0
114	26.0	4	98.0	90.0	2265	15.5
115	15.0	8	350.0	145.0	4082	13.0
116	16.0	8	400.0	230.0	4278	9.5
117	29.0	4	68.0	49.0	1867	19.5

```
118  24.0          4         116.0        75.0    2158          15.5
119  20.0          4         114.0        91.0    2582          14.0
120  19.0          4         121.0       112.0    2868          15.5
121  15.0          8         318.0       150.0    3399          11.0
122  24.0          4         121.0       110.0    2660          14.0
123  20.0          6         156.0       122.0    2807          13.5
124  11.0          8         350.0       180.0    3664          11.0
125  20.0          6         198.0        95.0    3102          16.5
126  21.0          6         200.0         NaN    2875          17.0
127  19.0          6         232.0       100.0    2901          16.0
128  15.0          6         250.0       100.0    3336          17.0
129  31.0          4          79.0        67.0    1950          19.0
130  26.0          4         122.0        80.0    2451          16.5
131  32.0          4          71.0        65.0    1836          21.0
132  25.0          4         140.0        75.0    2542          17.0
133  16.0          6         250.0       100.0    3781          17.0
134  16.0          6         258.0       110.0    3632          18.0
135  18.0          6         225.0       105.0    3613          16.5
136  16.0          8         302.0       140.0    4141          14.0
137  13.0          8         350.0       150.0    4699          14.5
138  14.0          8         318.0       150.0    4457          13.5
139  14.0          8         302.0       140.0    4638          16.0
140  14.0          8         304.0       150.0    4257          15.5
141  29.0          4          98.0        83.0    2219          16.5
```

142	26.0	4	79.0	67.0	1963	15.5
143	26.0	4	97.0	78.0	2300	14.5
144	31.0	4	76.0	52.0	1649	16.5
145	32.0	4	83.0	61.0	2003	19.0
146	28.0	4	90.0	75.0	2125	14.5
147	24.0	4	90.0	75.0	2108	15.5
148	26.0	4	116.0	75.0	2246	14.0
149	24.0	4	120.0	97.0	2489	15.0
150	26.0	4	108.0	93.0	2391	15.5
151	31.0	4	79.0	67.0	2000	16.0
152	19.0	6	225.0	95.0	3264	16.0
153	18.0	6	250.0	105.0	3459	16.0
154	15.0	6	250.0	72.0	3432	21.0
155	15.0	6	250.0	72.0	3158	19.5
156	16.0	8	400.0	170.0	4668	11.5
157	15.0	8	350.0	145.0	4440	14.0
158	16.0	8	318.0	150.0	4498	14.5
159	14.0	8	351.0	148.0	4657	13.5
160	17.0	6	231.0	110.0	3907	21.0
161	16.0	6	250.0	105.0	3897	18.5
162	15.0	6	258.0	110.0	3730	19.0
163	18.0	6	225.0	95.0	3785	19.0
164	21.0	6	231.0	110.0	3039	15.0
165	20.0	8	262.0	110.0	3221	13.5

166	13.0	8	302.0	129.0	3169	12.0
167	29.0	4	97.0	75.0	2171	16.0
168	23.0	4	140.0	83.0	2639	17.0
169	20.0	6	232.0	100.0	2914	16.0
170	23.0	4	140.0	78.0	2592	18.5
171	24.0	4	134.0	96.0	2702	13.5
172	25.0	4	90.0	71.0	2223	16.5
173	24.0	4	119.0	97.0	2545	17.0
174	18.0	6	171.0	97.0	2984	14.5
175	29.0	4	90.0	70.0	1937	14.0
176	19.0	6	232.0	90.0	3211	17.0
177	23.0	4	115.0	95.0	2694	15.0
178	23.0	4	120.0	88.0	2957	17.0
179	22.0	4	121.0	98.0	2945	14.5
180	25.0	4	121.0	115.0	2671	13.5
181	33.0	4	91.0	53.0	1795	17.5
182	28.0	4	107.0	86.0	2464	15.5
183	25.0	4	116.0	81.0	2220	16.9
184	25.0	4	140.0	92.0	2572	14.9
185	26.0	4	98.0	79.0	2255	17.7
186	27.0	4	101.0	83.0	2202	15.3
187	17.5	8	305.0	140.0	4215	13.0
188	16.0	8	318.0	150.0	4190	13.0
189	15.5	8	304.0	120.0	3962	13.9

190	14.5	8	351.0	152.0	4215	12.8
191	22.0	6	225.0	100.0	3233	15.4
192	22.0	6	250.0	105.0	3353	14.5
193	24.0	6	200.0	81.0	3012	17.6
194	22.5	6	232.0	90.0	3085	17.6
195	29.0	4	85.0	52.0	2035	22.2
196	24.5	4	98.0	60.0	2164	22.1
197	29.0	4	90.0	70.0	1937	14.2
198	33.0	4	91.0	53.0	1795	17.4
199	20.0	6	225.0	100.0	3651	17.7
200	18.0	6	250.0	78.0	3574	21.0
201	18.5	6	250.0	110.0	3645	16.2
202	17.5	6	258.0	95.0	3193	17.8
203	29.5	4	97.0	71.0	1825	12.2
204	32.0	4	85.0	70.0	1990	17.0
205	28.0	4	97.0	75.0	2155	16.4
206	26.5	4	140.0	72.0	2565	13.6
207	20.0	4	130.0	102.0	3150	15.7
208	13.0	8	318.0	150.0	3940	13.2
209	19.0	4	120.0	88.0	3270	21.9
210	19.0	6	156.0	108.0	2930	15.5
211	16.5	6	168.0	120.0	3820	16.7
212	16.5	8	350.0	180.0	4380	12.1
213	13.0	8	350.0	145.0	4055	12.0

214	13.0	8	302.0	130.0	3870	15.0
215	13.0	8	318.0	150.0	3755	14.0
216	31.5	4	98.0	68.0	2045	18.5
217	30.0	4	111.0	80.0	2155	14.8
218	36.0	4	79.0	58.0	1825	18.6
219	25.5	4	122.0	96.0	2300	15.5
220	33.5	4	85.0	70.0	1945	16.8
221	17.5	8	305.0	145.0	3880	12.5
222	17.0	8	260.0	110.0	4060	19.0
223	15.5	8	318.0	145.0	4140	13.7
224	15.0	8	302.0	130.0	4295	14.9
225	17.5	6	250.0	110.0	3520	16.4
226	20.5	6	231.0	105.0	3425	16.9
227	19.0	6	225.0	100.0	3630	17.7
228	18.5	6	250.0	98.0	3525	19.0
229	16.0	8	400.0	180.0	4220	11.1
230	15.5	8	350.0	170.0	4165	11.4
231	15.5	8	400.0	190.0	4325	12.2
232	16.0	8	351.0	149.0	4335	14.5
233	29.0	4	97.0	78.0	1940	14.5
234	24.5	4	151.0	88.0	2740	16.0
235	26.0	4	97.0	75.0	2265	18.2
236	25.5	4	140.0	89.0	2755	15.8
237	30.5	4	98.0	63.0	2051	17.0

238	33.5	4	98.0	83.0	2075	15.9
239	30.0	4	97.0	67.0	1985	16.4
240	30.5	4	97.0	78.0	2190	14.1
241	22.0	6	146.0	97.0	2815	14.5
242	21.5	4	121.0	110.0	2600	12.8
243	21.5	3	80.0	110.0	2720	13.5
244	43.1	4	90.0	48.0	1985	21.5
245	36.1	4	98.0	66.0	1800	14.4
246	32.8	4	78.0	52.0	1985	19.4
247	39.4	4	85.0	70.0	2070	18.6
248	36.1	4	91.0	60.0	1800	16.4
249	19.9	8	260.0	110.0	3365	15.5
250	19.4	8	318.0	140.0	3735	13.2
251	20.2	8	302.0	139.0	3570	12.8
252	19.2	6	231.0	105.0	3535	19.2
253	20.5	6	200.0	95.0	3155	18.2
254	20.2	6	200.0	85.0	2965	15.8
255	25.1	4	140.0	88.0	2720	15.4
256	20.5	6	225.0	100.0	3430	17.2
257	19.4	6	232.0	90.0	3210	17.2
258	20.6	6	231.0	105.0	3380	15.8
259	20.8	6	200.0	85.0	3070	16.7
260	18.6	6	225.0	110.0	3620	18.7
261	18.1	6	258.0	120.0	3410	15.1

262	19.2	8	305.0	145.0	3425	13.2
263	17.7	6	231.0	165.0	3445	13.4
264	18.1	8	302.0	139.0	3205	11.2
265	17.5	8	318.0	140.0	4080	13.7
266	30.0	4	98.0	68.0	2155	16.5
267	27.5	4	134.0	95.0	2560	14.2
268	27.2	4	119.0	97.0	2300	14.7
269	30.9	4	105.0	75.0	2230	14.5
270	21.1	4	134.0	95.0	2515	14.8
271	23.2	4	156.0	105.0	2745	16.7
272	23.8	4	151.0	85.0	2855	17.6
273	23.9	4	119.0	97.0	2405	14.9
274	20.3	5	131.0	103.0	2830	15.9
275	17.0	6	163.0	125.0	3140	13.6
276	21.6	4	121.0	115.0	2795	15.7
277	16.2	6	163.0	133.0	3410	15.8
278	31.5	4	89.0	71.0	1990	14.9
279	29.5	4	98.0	68.0	2135	16.6
280	21.5	6	231.0	115.0	3245	15.4
281	19.8	6	200.0	85.0	2990	18.2
282	22.3	4	140.0	88.0	2890	17.3
283	20.2	6	232.0	90.0	3265	18.2
284	20.6	6	225.0	110.0	3360	16.6
285	17.0	8	305.0	130.0	3840	15.4

286	17.6	8	302.0	129.0	3725	13.4
287	16.5	8	351.0	138.0	3955	13.2
288	18.2	8	318.0	135.0	3830	15.2
289	16.9	8	350.0	155.0	4360	14.9
290	15.5	8	351.0	142.0	4054	14.3
291	19.2	8	267.0	125.0	3605	15.0
292	18.5	8	360.0	150.0	3940	13.0
293	31.9	4	89.0	71.0	1925	14.0
294	34.1	4	86.0	65.0	1975	15.2
295	35.7	4	98.0	80.0	1915	14.4
296	27.4	4	121.0	80.0	2670	15.0
297	25.4	5	183.0	77.0	3530	20.1
298	23.0	8	350.0	125.0	3900	17.4
299	27.2	4	141.0	71.0	3190	24.8
300	23.9	8	260.0	90.0	3420	22.2
301	34.2	4	105.0	70.0	2200	13.2
302	34.5	4	105.0	70.0	2150	14.9
303	31.8	4	85.0	65.0	2020	19.2
304	37.3	4	91.0	69.0	2130	14.7
305	28.4	4	151.0	90.0	2670	16.0
306	28.8	6	173.0	115.0	2595	11.3
307	26.8	6	173.0	115.0	2700	12.9
308	33.5	4	151.0	90.0	2556	13.2
309	41.5	4	98.0	76.0	2144	14.7

```
310  38.1          4          89.0        60.0    1968          18.8
311  32.1          4          98.0        70.0    2120          15.5
312  37.2          4          86.0        65.0    2019          16.4
313  28.0          4         151.0        90.0    2678          16.5
314  26.4          4         140.0        88.0    2870          18.1
315  24.3          4         151.0        90.0    3003          20.1
316  19.1          6         225.0        90.0    3381          18.7
317  34.3          4          97.0        78.0    2188          15.8
318  29.8          4         134.0        90.0    2711          15.5
319  31.3          4         120.0        75.0    2542          17.5
320  37.0          4         119.0        92.0    2434          15.0
321  32.2          4         108.0        75.0    2265          15.2
322  46.6          4          86.0        65.0    2110          17.9
323  27.9          4         156.0       105.0    2800          14.4
324  40.8          4          85.0        65.0    2110          19.2
325  44.3          4          90.0        48.0    2085          21.7
326  43.4          4          90.0        48.0    2335          23.7
327  36.4          5         121.0        67.0    2950          19.9
328  30.0          4         146.0        67.0    3250          21.8
329  44.6          4          91.0        67.0    1850          13.8
330  40.9          4          85.0         NaN    1835          17.3
331  33.8          4          97.0        67.0    2145          18.0
332  29.8          4          89.0        62.0    1845          15.3
333  32.7          6         168.0       132.0    2910          11.4
```

334	23.7	3	70.0	100.0	2420	12.5
335	35.0	4	122.0	88.0	2500	15.1
336	23.6	4	140.0	NaN	2905	14.3
337	32.4	4	107.0	72.0	2290	17.0
338	27.2	4	135.0	84.0	2490	15.7
339	26.6	4	151.0	84.0	2635	16.4
340	25.8	4	156.0	92.0	2620	14.4
341	23.5	6	173.0	110.0	2725	12.6
342	30.0	4	135.0	84.0	2385	12.9
343	39.1	4	79.0	58.0	1755	16.9
344	39.0	4	86.0	64.0	1875	16.4
345	35.1	4	81.0	60.0	1760	16.1
346	32.3	4	97.0	67.0	2065	17.8
347	37.0	4	85.0	65.0	1975	19.4
348	37.7	4	89.0	62.0	2050	17.3
349	34.1	4	91.0	68.0	1985	16.0
350	34.7	4	105.0	63.0	2215	14.9
351	34.4	4	98.0	65.0	2045	16.2
352	29.9	4	98.0	65.0	2380	20.7
353	33.0	4	105.0	74.0	2190	14.2
354	34.5	4	100.0	NaN	2320	15.8
355	33.7	4	107.0	75.0	2210	14.4
356	32.4	4	108.0	75.0	2350	16.8
357	32.9	4	119.0	100.0	2615	14.8

358	31.6	4	120.0	74.0	2635	18.3
359	28.1	4	141.0	80.0	3230	20.4
360	30.7	6	145.0	76.0	3160	19.6
361	25.4	6	168.0	116.0	2900	12.6
362	24.2	6	146.0	120.0	2930	13.8
363	22.4	6	231.0	110.0	3415	15.8
364	26.6	8	350.0	105.0	3725	19.0
365	20.2	6	200.0	88.0	3060	17.1
366	17.6	6	225.0	85.0	3465	16.6
367	28.0	4	112.0	88.0	2605	19.6
368	27.0	4	112.0	88.0	2640	18.6
369	34.0	4	112.0	88.0	2395	18.0
370	31.0	4	112.0	85.0	2575	16.2
371	29.0	4	135.0	84.0	2525	16.0
372	27.0	4	151.0	90.0	2735	18.0
373	24.0	4	140.0	92.0	2865	16.4
374	23.0	4	151.0	NaN	3035	20.5
375	36.0	4	105.0	74.0	1980	15.3
376	37.0	4	91.0	68.0	2025	18.2
377	31.0	4	91.0	68.0	1970	17.6
378	38.0	4	105.0	63.0	2125	14.7
379	36.0	4	98.0	70.0	2125	17.3
380	36.0	4	120.0	88.0	2160	14.5
381	36.0	4	107.0	75.0	2205	14.5

```
382  34.0          4         108.0        70.0    2245          16.9

383  38.0          4           91.0        67.0    1965          15.0

384  32.0          4           91.0        67.0    1965          15.7

385  38.0          4           91.0        67.0    1995          16.2

386  25.0          6         181.0       110.0    2945          16.4

387  38.0          6         262.0        85.0    3015          17.0

388  26.0          4         156.0        92.0    2585          14.5

389  22.0          6         232.0       112.0    2835          14.7

390  32.0          4         144.0        96.0    2665          13.9

391  36.0          4         135.0        84.0    2370          13.0

392  27.0          4         151.0        90.0    2950          17.3

393  27.0          4         140.0        86.0    2790          15.6

394  44.0          4           97.0        52.0    2130          24.6

395  32.0          4         135.0        84.0    2295          11.6

396  28.0          4         120.0        79.0    2625          18.6

397  31.0          4         119.0        82.0    2720          19.4

     model_year  origin                                  name
0            70     usa             chevrolet chevelle malibu
1            70     usa                     buick skylark 320
2            70     usa                    plymouth satellite
3            70     usa                         amc rebel sst
4            70     usa                           ford torino
5            70     usa                      ford galaxie 500
6            70     usa                      chevrolet impala
7            70     usa                     plymouth fury iii
8            70     usa                      pontiac catalina
9            70     usa                    amc ambassador dpl
10           70     usa                   dodge challenger se
11           70     usa                    plymouth 'cuda 340
12           70     usa                 chevrolet monte carlo
```

13	70	usa	buick estate wagon (sw)
14	70	japan	toyota corona mark ii
15	70	usa	plymouth duster
16	70	usa	amc hornet
17	70	usa	ford maverick
18	70	japan	datsun pl510
19	70	europe	volkswagen 1131 deluxe sedan
20	70	europe	peugeot 504
21	70	europe	audi 100 ls
22	70	europe	saab 99e
23	70	europe	bmw 2002
24	70	usa	amc gremlin
25	70	usa	ford f250
26	70	usa	chevy c20
27	70	usa	dodge d200
28	70	usa	hi 1200d
29	71	japan	datsun pl510
30	71	usa	chevrolet vega 2300
31	71	japan	toyota corona
32	71	usa	ford pinto
33	71	usa	amc gremlin
34	71	usa	plymouth satellite custom
35	71	usa	chevrolet chevelle malibu
36	71	usa	ford torino 500
37	71	usa	amc matador
38	71	usa	chevrolet impala
39	71	usa	pontiac catalina brougham
40	71	usa	ford galaxie 500
41	71	usa	plymouth fury iii
42	71	usa	dodge monaco (sw)
43	71	usa	ford country squire (sw)
44	71	usa	pontiac safari (sw)
45	71	usa	amc hornet sportabout (sw)
46	71	usa	chevrolet vega (sw)
47	71	usa	pontiac firebird
48	71	usa	ford mustang
49	71	usa	mercury capri 2000
50	71	europe	opel 1900
51	71	europe	peugeot 304
52	71	europe	fiat 124b
53	71	japan	toyota corolla 1200
54	71	japan	datsun 1200
55	71	europe	volkswagen model 111
56	71	usa	plymouth cricket
57	72	japan	toyota corona hardtop
58	72	usa	dodge colt hardtop
59	72	europe	volkswagen type 3
60	72	usa	chevrolet vega
61	72	usa	ford pinto runabout
62	72	usa	chevrolet impala

63	72	usa	pontiac catalina
64	72	usa	plymouth fury iii
65	72	usa	ford galaxie 500
66	72	usa	amc ambassador sst
67	72	usa	mercury marquis
68	72	usa	buick lesabre custom
69	72	usa	oldsmobile delta 88 royale
70	72	usa	chrysler newport royal
71	72	japan	mazda rx2 coupe
72	72	usa	amc matador (sw)
73	72	usa	chevrolet chevelle concours (sw)
74	72	usa	ford gran torino (sw)
75	72	usa	plymouth satellite custom (sw)
76	72	europe	volvo 145e (sw)
77	72	europe	volkswagen 411 (sw)
78	72	europe	peugeot 504 (sw)
79	72	europe	renault 12 (sw)
80	72	usa	ford pinto (sw)
81	72	japan	datsun 510 (sw)
82	72	japan	toyouta corona mark ii (sw)
83	72	usa	dodge colt (sw)
84	72	japan	toyota corolla 1600 (sw)
85	73	usa	buick century 350
86	73	usa	amc matador
87	73	usa	chevrolet malibu
88	73	usa	ford gran torino
89	73	usa	dodge coronet custom
90	73	usa	mercury marquis brougham
91	73	usa	chevrolet caprice classic
92	73	usa	ford ltd
93	73	usa	plymouth fury gran sedan
94	73	usa	chrysler new yorker brougham
95	73	usa	buick electra 225 custom
96	73	usa	amc ambassador brougham
97	73	usa	plymouth valiant
98	73	usa	chevrolet nova custom
99	73	usa	amc hornet
100	73	usa	ford maverick
101	73	usa	plymouth duster
102	73	europe	volkswagen super beetle
103	73	usa	chevrolet impala
104	73	usa	ford country
105	73	usa	plymouth custom suburb
106	73	usa	oldsmobile vista cruiser
107	73	usa	amc gremlin
108	73	japan	toyota carina
109	73	usa	chevrolet vega
110	73	japan	datsun 610
111	73	japan	maxda rx3
112	73	usa	ford pinto

113	73	usa	mercury capri v6
114	73	europe	fiat 124 sport coupe
115	73	usa	chevrolet monte carlo s
116	73	usa	pontiac grand prix
117	73	europe	fiat 128
118	73	europe	opel manta
119	73	europe	audi 100ls
120	73	europe	volvo 144ea
121	73	usa	dodge dart custom
122	73	europe	saab 99le
123	73	japan	toyota mark ii
124	73	usa	oldsmobile omega
125	74	usa	plymouth duster
126	74	usa	ford maverick
127	74	usa	amc hornet
128	74	usa	chevrolet nova
129	74	japan	datsun b210
130	74	usa	ford pinto
131	74	japan	toyota corolla 1200
132	74	usa	chevrolet vega
133	74	usa	chevrolet chevelle malibu classic
134	74	usa	amc matador
135	74	usa	plymouth satellite sebring
136	74	usa	ford gran torino
137	74	usa	buick century luxus (sw)
138	74	usa	dodge coronet custom (sw)
139	74	usa	ford gran torino (sw)
140	74	usa	amc matador (sw)
141	74	europe	audi fox
142	74	europe	volkswagen dasher
143	74	europe	opel manta
144	74	japan	toyota corona
145	74	japan	datsun 710
146	74	usa	dodge colt
147	74	europe	fiat 128
148	74	europe	fiat 124 tc
149	74	japan	honda civic
150	74	japan	subaru
151	74	europe	fiat x1.9
152	75	usa	plymouth valiant custom
153	75	usa	chevrolet nova
154	75	usa	mercury monarch
155	75	usa	ford maverick
156	75	usa	pontiac catalina
157	75	usa	chevrolet bel air
158	75	usa	plymouth grand fury
159	75	usa	ford ltd
160	75	usa	buick century
161	75	usa	chevroelt chevelle malibu
162	75	usa	amc matador

163	75	usa	plymouth fury
164	75	usa	buick skyhawk
165	75	usa	chevrolet monza 2+2
166	75	usa	ford mustang ii
167	75	japan	toyota corolla
168	75	usa	ford pinto
169	75	usa	amc gremlin
170	75	usa	pontiac astro
171	75	japan	toyota corona
172	75	europe	volkswagen dasher
173	75	japan	datsun 710
174	75	usa	ford pinto
175	75	europe	volkswagen rabbit
176	75	usa	amc pacer
177	75	europe	audi 100ls
178	75	europe	peugeot 504
179	75	europe	volvo 244dl
180	75	europe	saab 99le
181	75	japan	honda civic cvcc
182	76	europe	fiat 131
183	76	europe	opel 1900
184	76	usa	capri ii
185	76	usa	dodge colt
186	76	europe	renault 12tl
187	76	usa	chevrolet chevelle malibu classic
188	76	usa	dodge coronet brougham
189	76	usa	amc matador
190	76	usa	ford gran torino
191	76	usa	plymouth valiant
192	76	usa	chevrolet nova
193	76	usa	ford maverick
194	76	usa	amc hornet
195	76	usa	chevrolet chevette
196	76	usa	chevrolet woody
197	76	europe	vw rabbit
198	76	japan	honda civic
199	76	usa	dodge aspen se
200	76	usa	ford granada ghia
201	76	usa	pontiac ventura sj
202	76	usa	amc pacer d/l
203	76	europe	volkswagen rabbit
204	76	japan	datsun b-210
205	76	japan	toyota corolla
206	76	usa	ford pinto
207	76	europe	volvo 245
208	76	usa	plymouth volare premier v8
209	76	europe	peugeot 504
210	76	japan	toyota mark ii
211	76	europe	mercedes-benz 280s
212	76	usa	cadillac seville

213	76	usa	chevy c10
214	76	usa	ford f108
215	76	usa	dodge d100
216	77	japan	honda accord cvcc
217	77	usa	buick opel isuzu deluxe
218	77	europe	renault 5 gtl
219	77	usa	plymouth arrow gs
220	77	japan	datsun f-10 hatchback
221	77	usa	chevrolet caprice classic
222	77	usa	oldsmobile cutlass supreme
223	77	usa	dodge monaco brougham
224	77	usa	mercury cougar brougham
225	77	usa	chevrolet concours
226	77	usa	buick skylark
227	77	usa	plymouth volare custom
228	77	usa	ford granada
229	77	usa	pontiac grand prix lj
230	77	usa	chevrolet monte carlo landau
231	77	usa	chrysler cordoba
232	77	usa	ford thunderbird
233	77	europe	volkswagen rabbit custom
234	77	usa	pontiac sunbird coupe
235	77	japan	toyota corolla liftback
236	77	usa	ford mustang ii 2+2
237	77	usa	chevrolet chevette
238	77	usa	dodge colt m/m
239	77	japan	subaru dl
240	77	europe	volkswagen dasher
241	77	japan	datsun 810
242	77	europe	bmw 320i
243	77	japan	mazda rx-4
244	78	europe	volkswagen rabbit custom diesel
245	78	usa	ford fiesta
246	78	japan	mazda glc deluxe
247	78	japan	datsun b210 gx
248	78	japan	honda civic cvcc
249	78	usa	oldsmobile cutlass salon brougham
250	78	usa	dodge diplomat
251	78	usa	mercury monarch ghia
252	78	usa	pontiac phoenix lj
253	78	usa	chevrolet malibu
254	78	usa	ford fairmont (auto)
255	78	usa	ford fairmont (man)
256	78	usa	plymouth volare
257	78	usa	amc concord
258	78	usa	buick century special
259	78	usa	mercury zephyr
260	78	usa	dodge aspen
261	78	usa	amc concord d/l
262	78	usa	chevrolet monte carlo landau

263	78	usa	buick regal sport coupe (turbo)
264	78	usa	ford futura
265	78	usa	dodge magnum xe
266	78	usa	chevrolet chevette
267	78	japan	toyota corona
268	78	japan	datsun 510
269	78	usa	dodge omni
270	78	japan	toyota celica gt liftback
271	78	usa	plymouth sapporo
272	78	usa	oldsmobile starfire sx
273	78	japan	datsun 200-sx
274	78	europe	audi 5000
275	78	europe	volvo 264gl
276	78	europe	saab 99gle
277	78	europe	peugeot 604sl
278	78	europe	volkswagen scirocco
279	78	japan	honda accord lx
280	79	usa	pontiac lemans v6
281	79	usa	mercury zephyr 6
282	79	usa	ford fairmont 4
283	79	usa	amc concord dl 6
284	79	usa	dodge aspen 6
285	79	usa	chevrolet caprice classic
286	79	usa	ford ltd landau
287	79	usa	mercury grand marquis
288	79	usa	dodge st. regis
289	79	usa	buick estate wagon (sw)
290	79	usa	ford country squire (sw)
291	79	usa	chevrolet malibu classic (sw)
292	79	usa	chrysler lebaron town @ country (sw)
293	79	europe	vw rabbit custom
294	79	japan	maxda glc deluxe
295	79	usa	dodge colt hatchback custom
296	79	usa	amc spirit dl
297	79	europe	mercedes benz 300d
298	79	usa	cadillac eldorado
299	79	europe	peugeot 504
300	79	usa	oldsmobile cutlass salon brougham
301	79	usa	plymouth horizon
302	79	usa	plymouth horizon tc3
303	79	japan	datsun 210
304	79	europe	fiat strada custom
305	79	usa	buick skylark limited
306	79	usa	chevrolet citation
307	79	usa	oldsmobile omega brougham
308	79	usa	pontiac phoenix
309	80	europe	vw rabbit
310	80	japan	toyota corolla tercel
311	80	usa	chevrolet chevette
312	80	japan	datsun 310

313	80	usa	chevrolet citation
314	80	usa	ford fairmont
315	80	usa	amc concord
316	80	usa	dodge aspen
317	80	europe	audi 4000
318	80	japan	toyota corona liftback
319	80	japan	mazda 626
320	80	japan	datsun 510 hatchback
321	80	japan	toyota corolla
322	80	japan	mazda glc
323	80	usa	dodge colt
324	80	japan	datsun 210
325	80	europe	vw rabbit c (diesel)
326	80	europe	vw dasher (diesel)
327	80	europe	audi 5000s (diesel)
328	80	europe	mercedes-benz 240d
329	80	japan	honda civic 1500 gl
330	80	europe	renault lecar deluxe
331	80	japan	subaru dl
332	80	europe	vokswagen rabbit
333	80	japan	datsun 280-zx
334	80	japan	mazda rx-7 gs
335	80	europe	triumph tr7 coupe
336	80	usa	ford mustang cobra
337	80	japan	honda accord
338	81	usa	plymouth reliant
339	81	usa	buick skylark
340	81	usa	dodge aries wagon (sw)
341	81	usa	chevrolet citation
342	81	usa	plymouth reliant
343	81	japan	toyota starlet
344	81	usa	plymouth champ
345	81	japan	honda civic 1300
346	81	japan	subaru
347	81	japan	datsun 210 mpg
348	81	japan	toyota tercel
349	81	japan	mazda glc 4
350	81	usa	plymouth horizon 4
351	81	usa	ford escort 4w
352	81	usa	ford escort 2h
353	81	europe	volkswagen jetta
354	81	europe	renault 18i
355	81	japan	honda prelude
356	81	japan	toyota corolla
357	81	japan	datsun 200sx
358	81	japan	mazda 626
359	81	europe	peugeot 505s turbo diesel
360	81	europe	volvo diesel
361	81	japan	toyota cressida
362	81	japan	datsun 810 maxima

```
363          81     usa                         buick century
364          81     usa                 oldsmobile cutlass ls
365          81     usa                       ford granada gl
366          81     usa                chrysler lebaron salon
367          82     usa                    chevrolet cavalier
368          82     usa              chevrolet cavalier wagon
369          82     usa             chevrolet cavalier 2-door
370          82     usa            pontiac j2000 se hatchback
371          82     usa                        dodge aries se
372          82     usa                       pontiac phoenix
373          82     usa                  ford fairmont futura
374          82     usa                        amc concord dl
375          82  europe                   volkswagen rabbit l
376          82   japan                    mazda glc custom l
377          82   japan                      mazda glc custom
378          82     usa                plymouth horizon miser
379          82     usa                        mercury lynx l
380          82   japan                      nissan stanza xe
381          82   japan                          honda accord
382          82   japan                        toyota corolla
383          82   japan                           honda civic
384          82   japan                    honda civic (auto)
385          82   japan                         datsun 310 gx
386          82     usa                 buick century limited
387          82     usa     oldsmobile cutlass ciera (diesel)
388          82     usa            chrysler lebaron medallion
389          82     usa                        ford granada l
390          82   japan                      toyota celica gt
391          82     usa                     dodge charger 2.2
392          82     usa                      chevrolet camaro
393          82     usa                       ford mustang gl
394          82  europe                             vw pickup
395          82     usa                         dodge rampage
396          82     usa                           ford ranger
397          82     usa                            chevy s-10
```

Problem 6:How many missing values?

```
car.isna().sum()
```

```
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model_year      0
origin          0
name            0
dtype: int64
```

Problem 7:Drop all missing values.

```
car = car.dropna()
```

```
car.isna().sum()
```

```
mpg             0
cylinders       0
displacement    0
horsepower      0
weight          0
acceleration    0
model_year      0
origin          0
name            0
dtype: int64
```

Problem 8:Description of the data frame.

```
car.describe()
```

```
              mpg   cylinders  displacement  horsepower      
weight  \
count  392.000000  392.000000    392.000000  392.000000   392.000000   

mean    23.445918    5.471939    194.411990  104.469388  2977.584184   

std      7.805007    1.705783    104.644004   38.491160   849.402560   

min      9.000000    3.000000     68.000000   46.000000  1613.000000   

25%     17.000000    4.000000    105.000000   75.000000  2225.250000   

50%     22.750000    4.000000    151.000000   93.500000  2803.500000   

75%     29.000000    8.000000    275.750000  126.000000  3614.750000   

max     46.600000    8.000000    455.000000  230.000000  5140.000000   


       acceleration  model_year  
count    392.000000  392.000000  
mean      15.541327   75.979592  
std        2.758864    3.683737  
min        8.000000   70.000000  
25%       13.775000   73.000000  
50%       15.500000   76.000000  
75%       17.025000   79.000000  
max       24.800000   82.000000  
```

Problem 9:Data type in each column

```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 392 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           392 non-null    float64
 1   cylinders     392 non-null    int64  
 2   displacement  392 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        392 non-null    int64  
 5   acceleration  392 non-null    float64
 6   model_year    392 non-null    int64  
 7   origin        392 non-null    object 
 8   name          392 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 30.6+ KB
```

Problem 10:Shape of dataframe

```
car.shape
```

```
(392, 9)
```

Indexing and Slicing

Problem 1: Import Titanic dataset and store as the pandas dataframe with name titanic

```
import pandas as pd
```

```
titanic =
pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/Titani
c.csv")
```

Problem 2: Print the info of titanic dataframe

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   pclass     1309 non-null   int64  
 1   survived   1309 non-null   int64  
 2   name       1309 non-null   object 
 3   sex        1309 non-null   object 
 4   age        1046 non-null   float64
 5   sibsp      1309 non-null   int64  
 6   parch      1309 non-null   int64  
 7   ticket     1309 non-null   object 
 8   fare       1308 non-null   float64
 9   cabin      295 non-null    object 
 10  embarked   1307 non-null   object 
 11  boat       486 non-null    object 
 12  body       121 non-null    float64
 13  home.dest  745 non-null    object 
dtypes: float64(3), int64(4), object(7)
memory usage: 143.3+ KB
```

Problem 3: Print the column labels

```
titanic.columns
```

```
Index(['pclass', 'survived', 'name', 'sex', 'age', 'sibsp', 'parch',
'ticket',
       'fare', 'cabin', 'embarked', 'boat', 'body', 'home.dest'],
      dtype='object')
```

Problem 4: Select passengers name column

```
titanic.name
```

```
0                          Allen, Miss. Elisabeth Walton
1                         Allison, Master. Hudson Trevor
2                           Allison, Miss. Helen Loraine
```

```
3                  Allison, Mr. Hudson Joshua Creighton
4       Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
                             ...
1304                               Zabour, Miss. Hileni
1305                              Zabour, Miss. Thamine
1306                          Zakarian, Mr. Mapriededer
1307                                Zakarian, Mr. Ortin
1308                                 Zimmerman, Mr. Leo
Name: name, Length: 1309, dtype: object
```

Problem 5: Select passengers name column as pandas series and save as name

```
name = titanic["name"]
```

```
name
```

```
0                         Allen, Miss. Elisabeth Walton
1                        Allison, Master. Hudson Trevor
2                          Allison, Miss. Helen Loraine
3                  Allison, Mr. Hudson Joshua Creighton
4       Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
                             ...
1304                               Zabour, Miss. Hileni
1305                              Zabour, Miss. Thamine
1306                          Zakarian, Mr. Mapriededer
1307                                Zakarian, Mr. Ortin
1308                                 Zimmerman, Mr. Leo
Name: name, Length: 1309, dtype: object
```

```
type(name)
```

```
pandas.core.series.Series
```

```
name.shape
```

```
(1309,)
```

Problem 6: Select passengers name column and save as pandas dataframe

```
name = titanic[["name"]]
```

```
name
```

```
                                                 name
0                       Allen, Miss. Elisabeth Walton
1                      Allison, Master. Hudson Trevor
2                        Allison, Miss. Helen Loraine
3                Allison, Mr. Hudson Joshua Creighton
4     Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
...                                               ...
1304                             Zabour, Miss. Hileni
1305                            Zabour, Miss. Thamine
1306                        Zakarian, Mr. Mapriededer
```

```
1307                              Zakarian, Mr. Ortin
1308                               Zimmerman, Mr. Leo

[1309 rows x 1 columns]
```

Note: The extracted column now has the label 'name'and also the name is now a dataframe.

```
type(name)
```

```
pandas.core.frame.DataFrame
```

Problem 7: Select 100th row and all columns with iloc function

```
titanic.iloc[100,:]
```

```
pclass                                                  1
survived                                                1
name         Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")
sex                                                  male
age                                                  49.0
sibsp                                                   1
parch                                                   0
ticket                                           PC 17485
fare                                              56.9292
cabin                                                 A20
embarked                                                C
boat                                                    1
body                                                  NaN
home.dest                                  London / Paris
Name: 100, dtype: object
```

Problem 8: Select 100th row with loc function

```
titanic.loc[100, :]
```

```
pclass                                                  1
survived                                                1
name         Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")
sex                                                  male
age                                                  49.0
sibsp                                                   1
parch                                                   0
ticket                                           PC 17485
fare                                              56.9292
cabin                                                 A20
embarked                                                C
boat                                                    1
body                                                  NaN
home.dest                                  London / Paris
Name: 100, dtype: object
```

Problem 9: Select all rows with column label name and fare column with iloc function

```
titanic.iloc[:, [2,8]]
```

```
                                                 name      fare
0                       Allen, Miss. Elisabeth Walton  211.3375
1                      Allison, Master. Hudson Trevor  151.5500
2                        Allison, Miss. Helen Loraine  151.5500
3                Allison, Mr. Hudson Joshua Creighton  151.5500
4     Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  151.5500
...                                               ...       ...
1304                             Zabour, Miss. Hileni   14.4542
1305                            Zabour, Miss. Thamine   14.4542
1306                        Zakarian, Mr. Mapriededer    7.2250
1307                              Zakarian, Mr. Ortin    7.2250
1308                               Zimmerman, Mr. Leo    7.8750

[1309 rows x 2 columns]
```

Problem 10: Select all rows with loc function and column label name and fare

```
titanic.loc[:, ["name","fare"]]
```

```
                                                 name      fare
0                       Allen, Miss. Elisabeth Walton  211.3375
1                      Allison, Master. Hudson Trevor  151.5500
2                        Allison, Miss. Helen Loraine  151.5500
3                Allison, Mr. Hudson Joshua Creighton  151.5500
4     Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  151.5500
...                                               ...       ...
1304                             Zabour, Miss. Hileni   14.4542
1305                            Zabour, Miss. Thamine   14.4542
1306                        Zakarian, Mr. Mapriededer    7.2250
1307                              Zakarian, Mr. Ortin    7.2250
1308                               Zimmerman, Mr. Leo    7.8750

[1309 rows x 2 columns]
```

Problem 11: Select row number 5oth, 25th, 15th and column label passenger class, fare, age, with both loc and iloc function .

```
#Syntax:  Dataframe.iloc[[. , . , .], ["","",""]]
```

```
titanic.loc[[50, 25, 15], ["pclass", "fare", "age"]]
```

```
    pclass      fare   age
50       1  512.3292  58.0
25       1   26.0000  25.0
15       1   25.9250   NaN
```

```
titanic.iloc[[50,25,15], [0,8,4]]
```

```
    pclass      fare   age
50       1  512.3292  58.0
```

```
25        1   26.0000  25.0
15        1   25.9250   NaN
```

Problem 12: Select rows from 10th to 12th and column label passenger class, fare, age with both loc and iloc function.

```
titanic.loc[10:25, ["pclass","fare","age"]]       #Extracting starting
from 10 and ending at 25. using ':'
```

```
    pclass      fare   age
10       1  227.5250  47.0
11       1  227.5250  18.0
12       1   69.3000  24.0
13       1   78.8500  26.0
14       1   30.0000  80.0
15       1   25.9250   NaN
16       1  247.5208  24.0
17       1  247.5208  50.0
18       1   76.2917  32.0
19       1   75.2417  36.0
20       1   52.5542  37.0
21       1   52.5542  47.0
22       1   30.0000  26.0
23       1  227.5250  42.0
24       1  221.7792  29.0
25       1   26.0000  25.0
```

```
titanic.iloc[10:26, [0,8,4]]            #In iloc function last index
is not included! It has [....)
```

```
    pclass      fare   age
10       1  227.5250  47.0
11       1  227.5250  18.0
12       1   69.3000  24.0
13       1   78.8500  26.0
14       1   30.0000  80.0
15       1   25.9250   NaN
16       1  247.5208  24.0
17       1  247.5208  50.0
18       1   76.2917  32.0
19       1   75.2417  36.0
20       1   52.5542  37.0
21       1   52.5542  47.0
22       1   30.0000  26.0
23       1  227.5250  42.0
24       1  221.7792  29.0
```

Problem 13: Select rows from 10th to 15th and columns from passenger class to age with both loc and iloc function.

```
titanic.loc[10:15, "pclass" : "age" ]
```

```
    pclass  survived                                                    
name  \
10       1         0                              Astor, Col. John 
Jacob   
11       1         1  Astor, Mrs. John Jacob (Madeleine Talmadge 
Force)   
12       1         1                        Aubart, Mme. Leontine 
Pauline   
13       1         1                         Barber, Miss. Ellen 
"Nellie"   
14       1         1                 Barkworth, Mr. Algernon Henry 
Wilson   
15       1         0                                    Baumann, Mr. John 
D   

       sex   age  
10    male  47.0  
11  female  18.0  
12  female  24.0  
13  female  26.0  
14    male  80.0  
15    male   NaN  
```

```
titanic.iloc[10:16, 0:5]
```

```
    pclass  survived                                                    
name  \
10       1         0                              Astor, Col. John 
Jacob   
11       1         1  Astor, Mrs. John Jacob (Madeleine Talmadge 
Force)   
12       1         1                        Aubart, Mme. Leontine 
Pauline   
13       1         1                         Barber, Miss. Ellen 
"Nellie"   
14       1         1                 Barkworth, Mr. Algernon Henry 
Wilson   
15       1         0                                    Baumann, Mr. John 
D   

       sex   age  
10    male  47.0  
11  female  18.0  
12  female  24.0  
13  female  26.0  
14    male  80.0  
15    male   NaN  
```

Problem 14: Select all passengers with age equal to and more than 35 years.

```
titanic[(titanic["age"]>= 35)]
```

```
      pclass  survived
name  \
5          1         1                              Anderson, Mr.
Harry
6          1         1                Andrews, Miss. Kornelia
Theodosia
7          1         0                           Andrews, Mr. Thomas
Jr
8          1         1    Appleton, Mrs. Edward Dale (Charlotte
Lamson)
9          1         0                          Artagaveytia, Mr.
Ramon
...      ...       ...                                              ..
.
1286       3         1  Whabee, Mrs. George Joseph (Shawneene Abi-
Saab)
1287       3         0                 Widegren, Mr. Carl/Charles
Peter
1290       3         1                 Wilkes, Mrs. James (Ellen
Needs)
1298       3         0                        Wittevrongel, Mr.
Camille
1301       3         0                             Youseff, Mr.
Gerious

         sex   age  sibsp  parch    ticket     fare cabin embarked
boat  \
5       male  48.0      0      0     19952  26.5500   E12         S
3
6     female  63.0      1      0     13502  77.9583    D7         S
10
7       male  39.0      0      0    112050   0.0000   A36         S
NaN
8     female  53.0      2      0     11769  51.4792  C101         S
D
9       male  71.0      0      0  PC 17609  49.5042   NaN         C
NaN
...      ...   ...    ...    ...       ...      ...   ...      ...  ..
.
1286  female  38.0      0      0      2688   7.2292   NaN         C
C
1287    male  51.0      0      0    347064   7.7500   NaN         S
NaN
1290  female  47.0      1      0    363272   7.0000   NaN         S
NaN
1298    male  36.0      0      0    345771   9.5000   NaN         S
NaN
1301    male  45.5      0      0      2628   7.2250   NaN         C
```

```
NaN

       body           home.dest
5       NaN        New York, NY
6       NaN          Hudson, NY
7       NaN         Belfast, NI
8       NaN  Bayside, Queens, NY
9      22.0  Montevideo, Uruguay
...     ...                 ...
1286    NaN                 NaN
1287    NaN                 NaN
1290    NaN                 NaN
1298    NaN                 NaN
1301  312.0                 NaN

[345 rows x 14 columns]
```

Problem 15: Select all passengers with age equal to and more than 35 years and column with label passenger class to age.

```
titanic.loc[(titanic["age"]>= 35), "pclass":"age"]
```

```
      pclass  survived                                                
name  \
5          1         1                              Anderson, Mr. 
Harry
6          1         1                Andrews, Miss. Kornelia 
Theodosia
7          1         0                           Andrews, Mr. Thomas 
Jr
8          1         1    Appleton, Mrs. Edward Dale (Charlotte 
Lamson)
9          1         0                          Artagaveytia, Mr. 
Ramon
...      ...       ...                                                 ..
.
1286       3         1  Whabee, Mrs. George Joseph (Shawneene Abi-
Saab)
1287       3         0               Widegren, Mr. Carl/Charles 
Peter
1290       3         1               Wilkes, Mrs. James (Ellen 
Needs)
1298       3         0                      Wittevrongel, Mr. 
Camille
1301       3         0                           Youseff, Mr. 
Gerious

         sex   age
5       male  48.0
6     female  63.0
```

```
7       male  39.0
8     female  53.0
9       male  71.0
...      ...   ...
1286  female  38.0
1287    male  51.0
1290  female  47.0
1298    male  36.0
1301    male  45.5

[345 rows x 5 columns]
```

Problem 16: Select all female passengers with age equal to and more than 35 years.

```
titanic.loc[(titanic["sex"] == "female") & (titanic["age"]>=35)]
```

```
      pclass  survived
name  \
6          1         1                 Andrews, Miss. Kornelia
Theodosia
8          1         1     Appleton, Mrs. Edward Dale (Charlotte
Lamson)
17         1         1   Baxter, Mrs. James (Helene DeLaudeniere
Chaput)
21         1         1  Beckwith, Mrs. Richard Leonard (Sallie
Monypeny)
23         1         1                             Bidois, Miss.
Rosalie
...      ...       ...                                               .
..
1158       3         0          Rosblom, Mrs. Viktor (Helena
Wilhelmina)
1211       3         0   Skoog, Mrs. William (Anna Bernhardina
Karlsson)
1261       3         1                            Turkula, Mrs.
(Hedwig)
1286       3         1   Whabee, Mrs. George Joseph (Shawneene Abi-
Saab)
1290       3         1                  Wilkes, Mrs. James (Ellen
Needs)

         sex   age  sibsp  parch    ticket      fare    cabin embarked
boat  \
6     female  63.0      1      0     13502   77.9583       D7        S
10
8     female  53.0      2      0     11769   51.4792     C101        S
D
17    female  50.0      0      1  PC 17558  247.5208  B58 B60        C
6
21    female  47.0      1      1     11751   52.5542      D35        S
5
```

```
23    female  42.0      0      0  PC 17757  227.5250      NaN          C
4
...      ...   ...    ...    ...       ...       ...      ...      ...
...
1158  female  41.0      0      2    370129   20.2125      NaN          S
NaN
1211  female  45.0      1      4    347088   27.9000      NaN          S
NaN
1261  female  63.0      0      0      4134    9.5875      NaN          S
15
1286  female  38.0      0      0      2688    7.2292      NaN          C
C
1290  female  47.0      1      0    363272    7.0000      NaN          S
NaN

      body            home.dest
6      NaN           Hudson, NY
8      NaN  Bayside, Queens, NY
17     NaN         Montreal, PQ
21     NaN         New York, NY
23     NaN                  NaN
...    ...                  ...
1158   NaN                  NaN
1211   NaN                  NaN
1261   NaN                  NaN
1286   NaN                  NaN
1290   NaN                  NaN

[125 rows x 14 columns]
```