```
import pandas as pd

url = "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/tips.csv"
df = pd.read_csv(url)

print(df.head())
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

```
# 2. Show first 10 rows
print("First 10 rows:")
print(df.head(10))
print()
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
5       25.29  4.71    Male     No  Sun  Dinner     4
6        8.77  2.00    Male     No  Sun  Dinner     2
7       26.88  3.12    Male     No  Sun  Dinner     4
8       15.04  1.96    Male     No  Sun  Dinner     2
9       14.78  3.23    Male     No  Sun  Dinner     2
```

```
# 3. Display columns and data types
print("Columns and data types:")
print(df.dtypes)
print()
```

```
Columns and data types:
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

```
# 4. Show average tip given per day
avg_tip_per_day = df.groupby("day")["tip"].mean()
print("Average tip per day:")
print(avg_tip_per_day)
print()
```

```
Average tip per day:
day
Fri     2.734737
Sat     2.993103
Sun     3.255132
Thur    2.771452
Name: tip, dtype: float64
```

```
# 5. Find max tip amount and who gave it
max_tip_row = df.loc[df["tip"].idxmax()]
print("Max tip details:")
print(max_tip_row)
print()
```

```
Max tip details:
total_bill     50.81
tip             10.0
sex             Male
smoker           Yes
day              Sat
time          Dinner
size               3
Name: 170, dtype: object
```

```
# 6. Add Tip_Percent column
df["Tip_Percent"] = (df["tip"] / df["total_bill"]) * 100
```

```
# 7. Show only rows where tip percentage > 20
high_tip_percent = df[df["Tip_Percent"] > 20]
print("Rows where tip percentage > 20:")
print(high_tip_percent)
print()
```

```
Rows where tip percentage > 20:
     total_bill   tip     sex smoker   day    time  size  Tip_Percent
6          8.77  2.00    Male     No   Sun  Dinner     2    22.805017
9         14.78  3.23    Male     No   Sun  Dinner     2    21.853857
14        14.83  3.02  Female     No   Sun  Dinner     2    20.364127
17        16.29  3.71    Male     No   Sun  Dinner     3    22.774708
18        16.97  3.50  Female     No   Sun  Dinner     3    20.624632
20        17.92  4.08    Male     No   Sat  Dinner     2    22.767857
42        13.94  3.06    Male     No   Sun  Dinner     2    21.951220
46        22.23  5.00    Male     No   Sun  Dinner     2    22.492128
51        10.29  2.60  Female     No   Sun  Dinner     2    25.267250
63        18.29  3.76    Male    Yes   Sat  Dinner     4    20.557682
67         3.07  1.00  Female    Yes   Sat  Dinner     1    32.573290
81        16.66  3.40    Male     No  Thur   Lunch     2    20.408163
87        18.28  4.00    Male     No  Thur   Lunch     2    21.881838
88        24.71  5.85    Male     No  Thur   Lunch     2    23.674626
93        16.32  4.30  Female    Yes   Fri  Dinner     2    26.348039
100       11.35  2.50  Female    Yes   Fri  Dinner     2    22.026432
108       18.24  3.76    Male     No   Sat  Dinner     2    20.614035
109       14.31  4.00  Female    Yes   Sat  Dinner     2    27.952481
110       14.00  3.00    Male     No   Sat  Dinner     2    21.428571
115       17.31  3.50  Female     No   Sun  Dinner     2    20.219526
124       12.48  2.52  Female     No  Thur   Lunch     2    20.192308
139       13.16  2.75  Female     No  Thur   Lunch     2    20.896657
140       17.47  3.50  Female     No  Thur   Lunch     2    20.034345
149        7.51  2.00    Male     No  Thur   Lunch     2    26.631158
172        7.25  5.15    Male    Yes   Sun  Dinner     2    71.034483
174       16.82  4.00    Male    Yes   Sun  Dinner     2    23.781213
178        9.60  4.00  Female    Yes   Sun  Dinner     2    41.666667
181       23.33  5.65    Male    Yes   Sun  Dinner     2    24.217745
183       23.17  6.50    Male    Yes   Sun  Dinner     4    28.053517
185       20.69  5.00    Male     No   Sun  Dinner     5    24.166264
191       19.81  4.19  Female    Yes  Thur   Lunch     2    21.150934
194       16.58  4.00    Male    Yes  Thur   Lunch     2    24.125452
200       18.71  4.00    Male    Yes  Thur   Lunch     3    21.378942
214       28.17  6.50  Female    Yes   Sat  Dinner     3    23.074192
221       13.42  3.48  Female    Yes   Fri   Lunch     2    25.931446
222        8.58  1.92    Male    Yes   Fri   Lunch     1    22.377622
228       13.28  2.72    Male     No   Sat  Dinner     2    20.481928
232       11.61  3.39    Male     No   Sat  Dinner     2    29.198966
239       29.03  5.92    Male     No   Sat  Dinner     3    20.392697
```

```
# 8. Group by gender and find average total_bill
avg_bill_by_gender = df.groupby("sex")["total_bill"].mean()
print("Average total bill by gender:")
print(avg_bill_by_gender)
```

```
Average total bill by gender:
sex
Female    18.056897
Male      20.744076
Name: total_bill, dtype: float64
```

```
import pandas as pd

url = "https://raw.githubusercontent.com/ybifoundation/Dataset/main/EmployeeAttrition.csv"
df = pd.read_csv(url)

# Display first few rows
print(df.head())
```

```
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7
```

```
   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  ...                         1            80                 0
1  ...                         4            80                 1
2  ...                         2            80                 0
3  ...                         3            80                 0
4  ...                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

```
# 1. Display first 8 rows and total number of columns
print("First 8 rows:")
print(df.head(8))
print("\nTotal number of columns:", df.shape[1]) # 1 means column by index
print("-" * 50)
```

```
First 8 rows:
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development
5   32        No  Travel_Frequently       1005  Research & Development
6   59        No      Travel_Rarely       1324  Research & Development
7   30        No      Travel_Rarely       1358  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7
5                 2          2  Life Sciences              1               8
6                 3          3        Medical              1              10
7                24          1  Life Sciences              1              11

   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  ...                         1            80                 0
1  ...                         4            80                 1
2  ...                         2            80                 0
3  ...                         3            80                 0
4  ...                         4            80                 1
5  ...                         3            80                 0
6  ...                         1            80                 3
7  ...                         2            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2
5                  8                      2               2               7
6                 12                      3               2               1
7                  1                      2               3               1

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2
5                  7                        3                     6
6                  0                        0                     0
7                  0                        0                     0

[8 rows x 35 columns]

Total number of columns: 35
```

```
# 2. Employees working in each Department
dept_count = df["Department"].value_counts()  #df["Department"] selects the Department column from the DataFrame.
#.value_counts(): Counts how many times each unique department name appears
```

```
print("Employees per Department:")
print(dept_count)
print("-" * 50)
```

```
Employees per Department:
Department
Research & Development    961
Sales                     446
Human Resources            63
Name: count, dtype: int64
--------------------------------------------------
```

```
# 3. Average MonthlyIncome and YearsAtCompany
avg_income_years = df[["MonthlyIncome", "YearsAtCompany"]].mean()
print("Average MonthlyIncome and YearsAtCompany:")
print(avg_income_years)
print("-" * 50)
```

```
Average MonthlyIncome and YearsAtCompany:
MonthlyIncome     6502.931293
YearsAtCompany       7.008163
dtype: float64
--------------------------------------------------
```

```
# 4. Employees with Attrition = Yes and OverTime = Yes
attrition_overtime = df[
    (df["Attrition"] == "Yes") & (df["OverTime"] == "Yes")
]
print("Employees with Attrition = Yes and OverTime = Yes:")
print(attrition_overtime)
print("-" * 50)
```

```
1396                24          4  Life Sciences              1
1442                 1          4        Medical              1
1461                28          3      Marketing              1

      EmployeeNumber  ...  RelationshipSatisfaction StandardHours  \
0                  1  ...                         1            80
2                  4  ...                         2            80
14                19  ...                         2            80
26                33  ...                         2            80
34                45  ...                         1            80
...              ...  ...                       ...           ...
1375            1939  ...                         1            80
1395            1967  ...                         3            80
1396            1968  ...                         2            80
1442            2027  ...                         2            80
1461            2055  ...                         2            80

      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                    0                  8                      0
2                    0                  7                      3
14                   0                  6                      4
26                   0                 10                      5
34                   1                  6                      2
...                ...                ...                    ...
1375                 0                  8                      2
1395                 0                 10                      4
1396                 0                 15                      2
1442                 3                  4                      3
1461                 1                 20                      3

     WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \

1461                            3                    0
```

```
1461                          2                    0

[127 rows x 35 columns]
--------------------------------------------------
```

```
# 5. Average MonthlyIncome for each JobRole
avg_income_jobrole = df.groupby("JobRole")["MonthlyIncome"].mean()
print("Average MonthlyIncome per JobRole:")
print(avg_income_jobrole)
print("-" * 50)
```

```
Average MonthlyIncome per JobRole:
JobRole
Healthcare Representative     7528.763359
Human Resources               4235.750000
Laboratory Technician         3237.169884
Manager                      17181.676471
Manufacturing Director        7295.137931
Research Director            16033.550000
Research Scientist            3239.972603
Sales Executive               6924.279141
Sales Representative          2626.000000
Name: MonthlyIncome, dtype: float64
--------------------------------------------------
```

```
# 6. Max, Min, and Average Age
age_stats = {
    "Max Age": df["Age"].max(),
    "Min Age": df["Age"].min(),
    "Average Age": df["Age"].mean()
}
print("Age Statistics:")
print(age_stats)
print("-" * 50)
```

```
Age Statistics:
{'Max Age': 60, 'Min Age': 18, 'Average Age': np.float64(36.923809523809524)}
--------------------------------------------------
```

```
# 7. EducationField with highest average salary
edu_salary = df.groupby("EducationField")["MonthlyIncome"].mean()
highest_edu_salary = edu_salary.idxmax()
print("EducationField with highest average salary:")
print(highest_edu_salary)
print("-" * 50)
```

```
EducationField with highest average salary:
Marketing
--------------------------------------------------
```

```
# 8. Employees with MonthlyIncome > 15000 and PerformanceRating = 4
high_income_performance = df[
    (df["MonthlyIncome"] > 15000) & (df["PerformanceRating"] == 4)
]
print("Employees with MonthlyIncome > 15000 and PerformanceRating = 4:")
print(high_income_performance)
```

```
1166              3                3                  3
```

```
1166             3              2                   2
1184             3             10                   8
1185             3             14                  10
1301             2             16                   9

      YearsSinceLastPromotion  YearsWithCurrManager
105                         2                     2
194                        11                     8
235                         1                     9
445                         7                     7
609                         1                     0
714                         1                     3
746                         5                    10
804                         2                     1
861                        15                     9
867                         2                     2
918                        11                    10
936                         0                     0
1009                        1                     5
1076                        1                    12
1096                        7                     7
1116                        2                    13
1129                        0                     1
1166                        2                     2
1184                        4                     7
1185                        6                    11
1301                       14                    14

[21 rows x 35 columns]
```

```
# 9. Add IncomeCategory column
def income_category(income):
    if income < 5000:
        return "Low"
    elif 5000 <= income <= 10000:
        return "Medium"
    else:
        return "High"

df["IncomeCategory"] = df["MonthlyIncome"].apply(income_category)

print("IncomeCategory column added:")
print(df[["MonthlyIncome", "IncomeCategory"]].head())
print("-" * 50)
```

```
IncomeCategory column added:
   MonthlyIncome IncomeCategory
0           5993         Medium
1           5130         Medium
2           2090            Low
3           2909            Low
4           3468            Low
--------------------------------------------------
```

```
# 10. Percentage of employees who left for each IncomeCategory
attrition_percentage = (
    df.groupby("IncomeCategory")["Attrition"]
      .apply(lambda x: (x == "Yes").mean() * 100)
)

print("Attrition percentage by IncomeCategory:")
print(attrition_percentage)
print("-" * 50)
```

```
Attrition percentage by IncomeCategory:
IncomeCategory
High       8.896797
Low       21.762350
Medium    11.136364
Name: Attrition, dtype: float64
--------------------------------------------------
```

```
# 11. Group by Department and Gender and count employees
dept_gender_count = df.groupby(["Department", "Gender"]).size()

print("Employee count by Department and Gender:")
print(dept_gender_count)
print("-" * 50)
```

```
Employee count by Department and Gender:
Department              Gender
Human Resources         Female     20
                        Male       43
Research & Development  Female    379
                        Male      582
Sales                   Female    189
```

```
                        Male      257
dtype: int64
--------------------------------------------------
```

```
# 12. Top 10 employees with highest MonthlyIncome
top_10_income = df.sort_values("MonthlyIncome", ascending=False).head(10)

print("Top 10 employees by MonthlyIncome:")
print(top_10_income[["EmployeeNumber", "MonthlyIncome", "JobRole", "Department"]])
print("-" * 50)
```

```
Top 10 employees by MonthlyIncome:
      EmployeeNumber  MonthlyIncome            JobRole              Department
190              259          19999            Manager  Research & Development
746             1035          19973  Research Director  Research & Development
851             1191          19943            Manager  Research & Development
165              226          19926            Manager  Research & Development
568              787          19859            Manager  Research & Development
918             1282          19847            Manager                   Sales
749             1038          19845            Manager                   Sales
1242            1740          19833            Manager                   Sales
898             1255          19740  Research Director  Research & Development
956             1338          19717            Manager         Human Resources
--------------------------------------------------
```

```
# 13. Average PerformanceRating and YearsAtCompany for OverTime = Yes
overtime_stats = df[df["OverTime"] == "Yes"][["PerformanceRating", "YearsAtCompany"]].mean()

print("Average PerformanceRating and YearsAtCompany (OverTime = Yes):")
print(overtime_stats)
```

```
Average PerformanceRating and YearsAtCompany (OverTime = Yes):
PerformanceRating    3.156250
YearsAtCompany       6.894231
dtype: float64
```