In [28]:

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

In [43]:

```
df = pd.read_csv("D:\\Study\\Programming\\python\\Python course from udemy\\[GigaCourse.Com] Udemy - 2022 Python for Machine Learning & Data Science Masterclass\\01 - Introduction to Course\\1UNZIP-FOR-NOTEBOOKS-FINAL\\03-Pandas\\tips.csv")
df.head()
```

Out[43]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 |

# Conditions

In [44]:

```
df['total_bill'] > 40 # here I showing true for bills which are more than 40$ and false for less than 40
```

Out[44]:

```
0      False
1      False
2      False
3      False
4      False
       ...
239    False
240    False
241    False
242    False
243    False
Name: total_bill, Length: 244, dtype: bool
```

In [45]:

```
df[df['total_bill'] > 40] # Here we see all the data where bill is more than 40$
```

Out[45]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **59** | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 |
| **95** | 40.17 | 4.73 | Male | Yes | Fri | Dinner | 4 | 10.04 | Aaron Bentley | 180026611638690 | Fri9628 |
| **102** | 44.30 | 2.50 | Female | Yes | Sat | Dinner | 3 | 14.77 | Heather Cohen | 379771118886604 | Sat6240 |
| **142** | 41.19 | 5.00 | Male | No | Thur | Lunch | 5 | 8.24 | Eric Andrews | 4356531761046453 | Thur3621 |
| **156** | 48.17 | 5.00 | Male | No | Sun | Dinner | 6 | 8.03 | Ryan Gonzales | 3523151482063321 | Sun7518 |
| **170** | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 |
| **182** | 45.35 | 3.50 | Male | Yes | Sun | Dinner | 3 | 15.12 | Jose Parsons | 4112207559459910 | Sun2337 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 184 | 10.65 | 3.00 | Male | Yes | Sat | Dinner | 2 | 25.27 | Stephen Cox | 3547799222044029 | Sun8140 |
| 197 | 43.11 | 5.00 | Female | Yes | Thur | Lunch | 4 | 10.78 | Brooke Soto | 5544902205760175 | Thur9313 |
| 212 | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 |

In [46]:

```
df[df['sex'] == 'Male']
```

Out[46]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 |
| 5 | 25.29 | 4.71 | Male | No | Sun | Dinner | 4 | 6.32 | Erik Smith | 213140353657882 | Sun9679 |
| 6 | 8.77 | 2.00 | Male | No | Sun | Dinner | 2 | 4.38 | Kristopher Johnson | 2223727524230344 | Sun5985 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 |
| 237 | 32.83 | 1.17 | Male | Yes | Sat | Dinner | 2 | 16.42 | Thomas Brown | 4284722681265508 | Sat2929 |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 |
| 241 | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 | 11.34 | Keith Wong | 6011891618747196 | Sat3880 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 |

157 rows × 11 columns

In [47]:

```
df[df['size'] > 3]
```

Out[47]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 |
| 5 | 25.29 | 4.71 | Male | No | Sun | Dinner | 4 | 6.32 | Erik Smith | 213140353657882 | Sun9679 |
| 7 | 26.88 | 3.12 | Male | No | Sun | Dinner | 4 | 6.72 | Robert Buck | 3514785077705092 | Sun8157 |
| 11 | 35.26 | 5.00 | Female | No | Sun | Dinner | 4 | 8.82 | Diane Macias | 4577817359320969 | Sun6686 |
| 13 | 18.43 | 3.00 | Male | No | Sun | Dinner | 4 | 4.61 | Joshua Jones | 6011163105616890 | Sun2971 |
| 23 | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 |
| 25 | 17.81 | 2.34 | Male | No | Sat | Dinner | 4 | 4.45 | Robert Perkins | 30502930499388 | Sat907 |
| 31 | 18.35 | 2.50 | Male | No | Sat | Dinner | 4 | 4.59 | Danny Santiago | 630415546013 | Sat4947 |
| 33 | 20.69 | 2.45 | Female | No | Sat | Dinner | 4 | 5.17 | Amber Francis | 377742985258914 | Sat6649 |
| 44 | 30.40 | 5.60 | Male | No | Sun | Dinner | 4 | 7.60 | Todd Cooper | 503846761263 | Sun2274 |
| 47 | 32.40 | 6.00 | Male | No | Sun | Dinner | 4 | 8.10 | James Barnes | 3552002592874186 | Sun9677 |
| 52 | 34.81 | 5.20 | Female | No | Sun | Dinner | 4 | 8.70 | Emily Daniel | 4291280793094374 | Sun6165 |
| 54 | 25.56 | 4.34 | Male | No | Sun | Dinner | 4 | 6.39 | Ronald Owens | 6569607991983380 | Sun9470 |
| 56 | 38.01 | 3.00 | Male | Yes | Sat | Dinner | 4 | 9.50 | James Christensen DDS | 349793629453226 | Sat8903 |
| 59 | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 |
| 63 | 18.29 | 3.76 | Male | Yes | Sat | Dinner | 4 | 4.57 | Chad Hart | 580171498976 | Sat4178 |
| 77 | 27.20 | 4.00 | Male | No | Thur | Lunch | 4 | 6.80 | John Davis | 30344778738589 | Thur4924 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 85 | 34.83 | 5.17 | Female | No | Thur | Lunch | 4 | 8.71 | Shawna Cook | 6011787464177340 | Thur7972 |
| 95 | 40.17 | 4.73 | Male | Yes | Fri | Dinner | 4 | 10.04 | Aaron Bentley | 180026611638690 | Fri9629 |
| 116 | 29.93 | 5.07 | Male | No | Sun | Dinner | 4 | 7.48 | Shawn Blake | 4689079711213722 | Sun22 |
| 119 | 24.08 | 2.92 | Female | No | Thur | Lunch | 4 | 6.02 | Melanie Jordan | 676212062720 | Thur8063 |
| 125 | 29.80 | 4.20 | Female | No | Thur | Lunch | 6 | 4.97 | Angela Sanchez | 503857080488 | Thur3948 |
| 141 | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 |
| 142 | 41.19 | 5.00 | Male | No | Thur | Lunch | 5 | 8.24 | Eric Andrews | 4356531761046453 | Thur3621 |
| 143 | 27.05 | 5.00 | Female | No | Thur | Lunch | 6 | 4.51 | Regina Jones | 4311048695487 | Thur6179 |
| 153 | 24.55 | 2.00 | Male | No | Sun | Dinner | 4 | 6.14 | Todd Patterson | 4416804908942159 | Sun8670 |
| 154 | 19.77 | 2.00 | Male | No | Sun | Dinner | 4 | 4.94 | James Smith | 213169731428229 | Sun5814 |
| 155 | 29.85 | 5.14 | Female | No | Sun | Dinner | 5 | 5.97 | Madison Wilson | 4210875236164664 | Sun9176 |
| 156 | 48.17 | 5.00 | Male | No | Sun | Dinner | 6 | 8.03 | Ryan Gonzales | 3523151482063321 | Sun7518 |
| 157 | 25.00 | 3.75 | Female | No | Sun | Dinner | 4 | 6.25 | Laura Robles | 213158685144262 | Sun7015 |
| 159 | 16.49 | 2.00 | Male | No | Sun | Dinner | 4 | 4.12 | Christopher Soto | 30501814271434 | Sun1781 |
| 160 | 21.50 | 3.50 | Male | No | Sun | Dinner | 4 | 5.38 | Travis Gonzalez | 3527668419764685 | Sun245 |
| 167 | 31.71 | 4.50 | Male | No | Sun | Dinner | 4 | 7.93 | Michael Lawson | 3566285921227119 | Sun3719 |
| 180 | 34.65 | 3.68 | Male | Yes | Sun | Dinner | 4 | 8.66 | James Hebert DDS | 676168737648 | Sun7544 |
| 183 | 23.17 | 6.50 | Male | Yes | Sun | Dinner | 4 | 5.79 | Dr. Michael James | 4718501859162 | Sun6059 |
| 185 | 20.69 | 5.00 | Male | No | Sun | Dinner | 5 | 4.14 | Joseph Howell | 30362407455623 | Sun5842 |
| 187 | 30.46 | 2.00 | Male | Yes | Sun | Dinner | 5 | 6.09 | David Barrett | 4792882899700988 | Sun9987 |
| 197 | 43.11 | 5.00 | Female | Yes | Thur | Lunch | 4 | 10.78 | Brooke Soto | 5544902205760175 | Thur9313 |
| 204 | 20.53 | 4.00 | Male | Yes | Thur | Lunch | 4 | 5.13 | Scott Kim | 3570611756827620 | Thur2160 |
| 207 | 38.73 | 3.00 | Male | Yes | Sat | Dinner | 4 | 9.68 | Ricky Ramirez | 347817964484033 | Sat4505 |
| 211 | 25.89 | 5.16 | Male | Yes | Sat | Dinner | 4 | 6.47 | Christopher Li | 6011962464150569 | Sat6735 |
| 212 | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 |
| 216 | 28.15 | 3.00 | Male | Yes | Sat | Dinner | 5 | 5.63 | Shawn Barnett PhD | 4590982568244 | Sat7320 |
| 219 | 30.14 | 3.09 | Female | Yes | Sat | Dinner | 4 | 7.54 | Shelby House | 502097403252 | Sat8863 |
| 227 | 20.45 | 3.00 | Male | No | Sat | Dinner | 4 | 5.11 | Robert Bradley | 213141668145910 | Sat4319 |
| 230 | 24.01 | 2.00 | Male | Yes | Sat | Dinner | 4 | 6.00 | Michael Osborne | 4258682154026 | Sat7872 |

## Multiple Conditions

**Recall the steps:**

- **Get the conditions**
- **Wrap each condition in parenthesis**
- **Use the | or & operator, depending if you want an**
  - **OR | (either condition is True)**
  - **AND & (both conditions must be True)**
- **You can also use the ~ operator as a NOT operation**

In [48]:

```
(df['total_bill'] > 30) and (df['sex'] == 'Male') # we will get error on doing this so w
e have to use & ensted of and
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8784\3841304823.py in <module>
----> 1 (df['total_bill'] > 30) and (df['sex'] == 'Male') # we will get error on doing t
his so we have to use & ensted of and
```

```
D:\Installed Software\Anaconda\lib\site-packages\pandas\core\generic.py in __nonzero__(self)
   1525     @final
   1526     def __nonzero__(self):
-> 1527         raise ValueError(
   1528             f"The truth value of a {type(self).__name__} is ambiguous. "
   1529             "Use a.empty, a.bool(), a.item(), a.any() or a.all()."

ValueError: The truth value of a Series is ambiguous. Use a.empty, a.bool(), a.item(), a.any() or a.all().
```

In [49]:

```
(df['total_bill'] > 30) & (df['sex'] == 'Male') # it word on replacing and with &
```

Out[49]:

```
0      False
1      False
2      False
3      False
4      False
       ...
239    False
240    False
241    False
242    False
243    False
Length: 244, dtype: bool
```

In [50]:

```
df[(df['total_bill'] > 30) & (df['sex'] == 'Male')] # Here all those rows where total_bill is more than 30 and all Male
```

Out[50]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **23** | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 |
| **39** | 31.27 | 5.00 | Male | No | Sat | Dinner | 3 | 10.42 | Mr. Brandon Berry | 6011525851069856 | Sat6373 |
| **44** | 30.40 | 5.60 | Male | No | Sun | Dinner | 4 | 7.60 | Todd Cooper | 503846761263 | Sun2274 |
| **47** | 32.40 | 6.00 | Male | No | Sun | Dinner | 4 | 8.10 | James Barnes | 3552002592874186 | Sun9677 |
| **56** | 38.01 | 3.00 | Male | Yes | Sat | Dinner | 4 | 9.50 | James Christensen DDS | 349793629453226 | Sat8903 |
| **59** | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 |
| **83** | 32.68 | 5.00 | Male | Yes | Thur | Lunch | 2 | 16.34 | Daniel Murphy | 5356177501009133 | Thur8801 |
| **95** | 40.17 | 4.73 | Male | Yes | Fri | Dinner | 4 | 10.04 | Aaron Bentley | 180026611638690 | Fri9628 |
| **112** | 38.07 | 4.00 | Male | No | Sun | Dinner | 3 | 12.69 | Jeff Lopez | 3572865915176463 | Sun591 |
| **141** | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 |
| **142** | 41.19 | 5.00 | Male | No | Thur | Lunch | 5 | 8.24 | Eric Andrews | 4356531761046453 | Thur3621 |
| **156** | 48.17 | 5.00 | Male | No | Sun | Dinner | 6 | 8.03 | Ryan Gonzales | 3523151482063321 | Sun7518 |
| **167** | 31.71 | 4.50 | Male | No | Sun | Dinner | 4 | 7.93 | Michael Lawson | 3566285921227119 | Sun3719 |
| **170** | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 |
| **173** | 31.85 | 3.18 | Male | Yes | Sun | Dinner | 2 | 15.92 | Scott Perez | 3577115550328507 | Sun9335 |
| **175** | 32.90 | 3.11 | Male | Yes | Sun | Dinner | 2 | 16.45 | Nathan Reynolds | 370307040837149 | Sun5109 |
| **179** | 34.63 | 3.55 | Male | Yes | Sun | Dinner | 2 | 17.32 | Brian Bailey | 346656312114848 | Sun9851 |
| **180** | 34.65 | 3.68 | Male | Yes | Sun | Dinner | 4 | 8.66 | James Hebert DDS | 676168737648 | Sun7544 |
| **182** | 45.35 | 3.50 | Male | Yes | Sun | Dinner | 3 | 15.12 | Jose Parsons | 4112207559459910 | Sun2337 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 184 | 40.55 | 3.00 | Male | Yes | Sun | Dinner | 2 | 20.27 | Stephen Cox | 3547798222044029 | Sun5140 |
| 187 | 30.46 | 2.00 | Male | Yes | Sun | Dinner | 5 | 6.09 | David Barrett | 4792882899700988 | Sun9987 |
| 207 | 38.73 | 3.00 | Male | Yes | Sat | Dinner | 4 | 9.68 | Ricky Ramirez | 347817964484033 | Sat4505 |
| 210 | 30.06 | 2.00 | Male | Yes | Sat | Dinner | 3 | 10.02 | Shawn Mendoza | 30184049218122 | Sat8361 |
| 212 | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 |
| 237 | 32.83 | 1.17 | Male | Yes | Sat | Dinner | 2 | 16.42 | Thomas Brown | 4284722681265508 | Sat2929 |

In [51]:

```
df[(df['total_bill'] > 30) | (df['sex'] == 'Male')] #Here all those rows where total_bil
l is more than 30 or either all Male
```

Out[51]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 |
| 5 | 25.29 | 4.71 | Male | No | Sun | Dinner | 4 | 6.32 | Erik Smith | 213140353657882 | Sun9679 |
| 6 | 8.77 | 2.00 | Male | No | Sun | Dinner | 2 | 4.38 | Kristopher Johnson | 2223727524230344 | Sun5985 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 237 | 32.83 | 1.17 | Male | Yes | Sat | Dinner | 2 | 16.42 | Thomas Brown | 4284722681265508 | Sat2929 |
| 238 | 35.83 | 4.67 | Female | No | Sat | Dinner | 3 | 11.94 | Kimberly Crane | 676184013727 | Sat9777 |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 |
| 241 | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 | 11.34 | Keith Wong | 6011891618747196 | Sat3880 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 |

**164 rows × 11 columns**

In [52]:

```
# We need data for weekend like sunday and saturday
df[(df['day'] == 'Sun') | (df['day'] == 'Sat')]
```

Out[52]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 238 | 35.83 | 4.67 | Female | No | Sat | Dinner | 3 | 11.94 | Kimberly Crane | 676184013727 | Sat9777 |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 |
| 240 | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 | 13.59 | Monica Sanders | 3506806155565404 | Sat1766 |
| 241 | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 | 11.34 | Keith Wong | 6011891618747196 | Sat3880 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 |

**163 rows × 11 columns**

### isin Function

In [53]:

```
# If we have long list of data to cheak we cant create or stantement like that we will us
e isin function
```

In [54]:

```
df[df['day'].isin(['Sat','Sun'])] # Remeber isin take data as list
```

Out[54]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 238 | 35.83 | 4.67 | Female | No | Sat | Dinner | 3 | 11.94 | Kimberly Crane | 676184013727 | Sat9777 |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 |
| 240 | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 | 13.59 | Monica Sanders | 3506806155565404 | Sat1766 |
| 241 | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 | 11.34 | Keith Wong | 6011891618747196 | Sat3880 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 |

163 rows × 11 columns

### Here we have cc number we want last 4 number of all

In [55]:

```
# Suppose we have a number
a=1234567890
```

In [56]:

```
# Now we have to convert into string then the featch last 4 number
a=str(a)[-4:]                # Here it give last 4 number as sring
b=int(str(a)[-4:])          # it will convert back in int
a,b, type(a),type(b)
```

Out[56]:

```
('7890', 7890, str, int)
```

In [57]:

```
# With the help of function we covert all cc number like this
def last_num(x):
    return str(x)[-4:]
```

In [58]:

```
last_num(1234567890) # testing
```

Out[58]:

```
'7890'
```

## Using .apply() with more complex functions

In [59]:

```
df['last_four']=df['CC Number'].apply(last_num) # Here we dont use () after function name in apply command
df
```

Out[59]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 341 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 | 923 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 | 132 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 | 595 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 | 722 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 | 284 |
| 240 | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 | 13.59 | Monica Sanders | 3506806155565404 | Sat1766 | 540 |
| 241 | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 | 11.34 | Keith Wong | 6011891618747196 | Sat3880 | 719 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 | 095 |
| 243 | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 | 9.39 | Michelle Hardin | 3511451626698139 | Thur672 | 813 |

244 rows × 12 columns

### Here we want 'Low' for total bill less then 10, 'Average' for between 10 to 30 and 'High' for more than 30

In [60]:

```
# We will create function
def yelp(price):
    if price > 10:
        return 'Low'
    elif price >= 10 and price < 30:
        return 'Average'
    else:
        return 'High'
```

In [61]:

```
df['Catagory'] = df['total_bill'].apply(yelp)
df.head()
```

Out[61]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3410 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 | 9230 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 | 1322 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 | 5994 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 | 7221 |

## apply with lambda

In [62]:

```
# Supoose we have to make a function for double any number
def double(num):
    return num*2
```

In [63]:

```
double(45),double('er') # Here we test for string it repeat the string as well
```

Out[63]:

```
(90, 'erer')
```

In [64]:

```
#Now lets convert it in lambda function , here we made for thrice any number and we will
try on sting as well
x=lambda Num: Num*3
```

In [65]:

```
x(35),x('ab') # Here we dis it but lambda function
```

Out[65]:

```
(105, 'ababab')
```

In [66]:

```
# Now we will apply that in pandas
df['total_bill'].apply(lambda num: num*2)
```

Out[66]:

```
0      33.98
1      20.68
2      42.02
3      47.36
4      49.18
       ...
239    58.06
240    54.36
241    45.34
242    35.64
243    37.56
Name: total_bill, Length: 244, dtype: float64
```

# apply that uses multiple columns

In [67]:

```
def quality(total_bill,tip):
    if tip/total_bill > 0.25:
        return 'Generous'
    else:
        return 'Other'
```

In [68]:

```
quality(10,2), quality(10,5) # Testing
```

Out[68]:

```
('Other', 'Generous')
```

In [69]:

```
# we have use lambda function as well with the function quality look at syntex carefully
df['Quality'] = df[['total_bill','tip']].apply(lambda df: quality(df['total_bill'],df['t
ip']),axis=1)
```

In [70]:

```
df.head()
```

Out[70]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3410 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 | 9230 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 | 1322 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 | 5994 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 | 7221 |

**apply on multiple columns by using vectorize function that from numpy**

In [71]:

```
df['quality'] = np.vectorize(quality)(df['total_bill'],df['tip'])
df.head() # vectorize is much simpler and faster
```

Out[71]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3410 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 | 9230 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 | 7.00 | Travis Walters | 6011812112971322 | Sun4458 | 1322 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 | 11.84 | Nathaniel Harris | 4676137647685994 | Sun5260 | 5994 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 | 6.15 | Tonya Carter | 4832732618637221 | Sun2251 | 7221 |

**So, which one is faster?**

In [82]:

```
import timeit
```

```
# code snippet to be executed only once
setup = '''
import numpy as np
import pandas as pd
df = pd.read_csv('tips.csv')
def quality(total_bill,tip):
    if tip/total_bill  > 0.25:
        return "Generous"
    else:
        return "Other"
'''

# code snippet whose execution time is to be measured
stmt_one = '''
df['tip_Quality'] = df[['total_bill','tip']].apply(lambda df: quality(df['total_bill'],df
['tip']),axis=1)
'''

stmt_two = '''
df['tip_quality'] = np.vectorize(quality)(df['total_bill'], df['tip'])
'''
```

In [83]:

```
timeit.timeit(setup = setup,
                    stmt = stmt_one,
                    number = 1000)
```

Out[83]:

```
6.782767099997727
```

In [84]:

```
timeit.timeit(setup = setup,
                    stmt = stmt_two,
                    number = 1000)
```

Out[84]:

```
0.47722559999965597
```

**Wow! Vectorization is much faster! Keep np.vectorize() in mind for the future.**

**Full Details: https://docs.scipy.org/doc/numpy/reference/generated/numpy.vectorize.html**

## df.describe for statistical summaries

In [89]:

```
df.describe()
```

Out[89]:

| | total_bill | tip | size | price_per_person | CC Number |
|---|---|---|---|---|---|
| count | 244.000000 | 244.000000 | 244.000000 | 244.000000 | 2.440000e+02 |
| mean | 19.785943 | 2.998279 | 2.569672 | 7.888197 | 2.563496e+15 |
| std | 8.902412 | 1.383638 | 0.951100 | 2.914234 | 2.369340e+15 |
| min | 3.070000 | 1.000000 | 1.000000 | 2.880000 | 6.040679e+10 |
| 25% | 13.347500 | 2.000000 | 2.000000 | 5.800000 | 3.040731e+13 |
| 50% | 17.795000 | 2.900000 | 2.000000 | 7.255000 | 3.525318e+15 |
| 75% | 24.127500 | 3.562500 | 3.000000 | 9.390000 | 4.553675e+15 |
| max | 50.810000 | 10.000000 | 6.000000 | 20.270000 | 6.596454e+15 |

In [91]:

```
df.describe().transpose()  # Transpose means change rows and columns
```

Out[91]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **total_bill** | 244.0 | 1.978594e+01 | 8.902412e+00 | 3.070000e+00 | 1.334750e+01 | 1.779500e+01 | 2.412750e+01 | 5.081000e+01 |
| **tip** | 244.0 | 2.998279e+00 | 1.383638e+00 | 1.000000e+00 | 2.000000e+00 | 2.900000e+00 | 3.562500e+00 | 1.000000e+01 |
| **size** | 244.0 | 2.569672e+00 | 9.510998e-01 | 1.000000e+00 | 2.000000e+00 | 2.000000e+00 | 3.000000e+00 | 6.000000e+00 |
| **price_per_person** | 244.0 | 7.888197e+00 | 2.914234e+00 | 2.880000e+00 | 5.800000e+00 | 7.255000e+00 | 9.390000e+00 | 2.027000e+01 |
| **CC Number** | 244.0 | 2.563496e+15 | 2.369340e+15 | 6.040679e+10 | 3.040731e+13 | 3.525318e+15 | 4.553675e+15 | 6.596454e+15 |

## sort_values()

In [98]:

```
df.sort_values('tip') # By default ascending is True, Now values are low to high
```

Out[98]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **67** | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 5 |
| **236** | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 3 |
| **92** | 5.75 | 1.00 | Female | Yes | Fri | Dinner | 2 | 2.88 | Leah Ramirez | 3508911676966392 | Fri3780 | 6 |
| **111** | 7.25 | 1.00 | Female | No | Sat | Dinner | 1 | 7.25 | Terri Jones | 3559221007826887 | Sat4801 | 6 |
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **141** | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 | 8 |
| **59** | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 | 0 |
| **23** | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 | 9 |
| **212** | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 | 5 |
| **170** | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 | 8 |

244 rows × 15 columns

In [97]:

```
df.sort_values('tip', ascending=False)
```

Out[97]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **170** | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 | 8 |
| **212** | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex | 676218815212 | Sat4590 | 5 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23 | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 | 9 |
| 59 | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 | 0 |
| 141 | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 | 8 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3 |
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 3 |
| 111 | 7.25 | 1.00 | Female | No | Sat | Dinner | 1 | 7.25 | Terri Jones | 3559221007826887 | Sat4801 | 6 |
| 67 | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 5 |
| 92 | 5.75 | 1.00 | Female | Yes | Fri | Dinner | 2 | 2.88 | Leah Ramirez | 3508911676966392 | Fri3780 | 6 |

244 rows × 15 columns

In [100]:

```
df.sort_values(['tip','size']) # Here it will first sort by tip then by size
```

Out[100]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 5 |
| 111 | 7.25 | 1.00 | Female | No | Sat | Dinner | 1 | 7.25 | Terri Jones | 3559221007826887 | Sat4801 | 6 |
| 92 | 5.75 | 1.00 | Female | Yes | Fri | Dinner | 2 | 2.88 | Leah Ramirez | 3508911676966392 | Fri3780 | 6 |
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 3 |
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 3 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 141 | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 | 8 |
| 59 | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 | 0 |
| 23 | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 | 9 |
| 212 | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 | 5 |
| 170 | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 | 8 |

244 rows × 15 columns

## idxmin and idxmax

In [101]:

```
df['total_bill'].max() #  Maximum value of total_bill
```

Out[101]:

50.81

In [104]:

```
df['total_bill'].idxmax() # Index value where maximum value of total_bill is availabe
```

Out[104]:

170

In [107]:

```
df.iloc[170]
```

Out[107]:

```
total_bill                     50.81
tip                             10.0
sex                             Male
smoker                           Yes
day                              Sat
time                          Dinner
size                               3
price_per_person               16.94
Payer Name              Gregory Clark
CC Number           5473850968388236
Payment ID                  Sat1954
last_four                      8236
Catagory                        Low
Quality                       Other
quality                       Other
Name: 170, dtype: object
```

In [109]:

```
df.iloc[df['total_bill'].idxmax()]
```

Out[109]:

```
total_bill                     50.81
tip                             10.0
sex                             Male
smoker                           Yes
day                              Sat
time                          Dinner
size                               3
price_per_person               16.94
Payer Name              Gregory Clark
CC Number           5473850968388236
Payment ID                  Sat1954
last_four                      8236
Catagory                        Low
Quality                       Other
quality                       Other
Name: 170, dtype: object
```

In [105]:

```
df['total_bill'].min()
```

Out[105]:

3.07

In [106]:

```
df['total_bill'].idxmin()
```

Out[106]:

67

## df.corr() for correlation checks

[Wikipedia on Correlation]

In [113]:

```
df.corr() # How values are correlative to reach other how they up and down together
```

Out[113]:

| | total_bill | tip | size | price_per_person | CC Number |
|---|---|---|---|---|---|
| **total_bill** | 1.000000 | 0.675734 | 0.598315 | 0.647554 | 0.104576 |
| **tip** | 0.675734 | 1.000000 | 0.489299 | 0.347405 | 0.110857 |
| **size** | 0.598315 | 0.489299 | 1.000000 | -0.175359 | -0.030239 |
| **price_per_person** | 0.647554 | 0.347405 | -0.175359 | 1.000000 | 0.135240 |
| **CC Number** | 0.104576 | 0.110857 | -0.030239 | 0.135240 | 1.000000 |

In [115]:

```
df[['total_bill','tip']].corr() # If we wanna see correlation between two columns
```

Out[115]:

| | total_bill | tip |
|---|---|---|
| **total_bill** | 1.000000 | 0.675734 |
| **tip** | 0.675734 | 1.000000 |

## value_counts

**Nice method to quickly get a count per category. Only makes sense on categorical columns.**

In [119]:

```
df['day'].unique() # Show unique values
```

Out[119]:

```
array(['Sun', 'Sat', 'Thur', 'Fri'], dtype=object)
```

In [120]:

```
len(df['day'].unique()) # Count number of unique values
```

Out[120]:

```
4
```

In [122]:

```
df['day'].value_counts() # Here count of unique values
```

Out[122]:

```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

## replace

**Quickly replace values with another one.**

In [124]:

```
df['sex'].replace(['Female'],['F']) # Here we repalce Female with F
```

Out[124]:

```
0         F
1      Male
2      Male
3      Male
4         F
       ...
239    Male
240       F
241    Male
242    Male
243       F
Name: sex, Length: 244, dtype: object
```

In [126]:

```
df['sex'].replace(['Female','Male'],['F','M']) # here we reaplace both at once
```

Out[126]:

```
0      F
1      M
2      M
3      M
4      F
      ..
239    M
240    F
241    M
242    M
243    F
Name: sex, Length: 244, dtype: object
```

### map

In [129]:

```
# There is another way to replace values by using dictionary, then use that in  map funct
ion to replace names
mymap={'Female':'F','Male':'M'}
df['sex'].map(mymap)
```

Out[129]:

```
0      F
1      M
2      M
3      M
4      F
      ..
239    M
240    F
241    M
242    M
243    F
Name: sex, Length: 244, dtype: object
```

## Duplicates

### .duplicated() and .drop_duplicates()

In [130]:

```
df.duplicated() #Here we cant find duplicates
```

Out[130]:

```
0      False
1      False
2      False
3      False
4      False
       ...
239    False
240    False
241    False
242    False
243    False
Length: 244, dtype: bool
```

In [135]:

```
# We have create own dataframe with duplicate
simple_df = pd.DataFrame([1,2,2,2],['a','b','c','d'])
simple_df
```

Out[135]:

| | 0 |
|---|---|
| **a** | 1 |
| **b** | 2 |
| **c** | 2 |
| **d** | 2 |

In [134]:

```
simple_df.duplicated()
```

Out[134]:

```
a    False
b    False
c     True
dtype: bool
```

In [137]:

```
simple_df.drop_duplicates()
```

Out[137]:

| | 0 |
|---|---|
| **a** | 1 |
| **b** | 2 |

# between

**left: A scalar value that defines the left boundary right: A scalar value that defines the right boundary inclusive: A Boolean value which is True by default. If False, it excludes the two passed arguments while checking.**

In [138]:

```
df['total_bill'].between(10,20,inclusive=True)
```

```
C:\Users\Chromsy\AppData\Local\Temp\ipykernel_8784\4013148072.py:1: FutureWarning: Boolea
n inputs to the `inclusive` argument are deprecated in favour of `both` or `neither`.
  df['total_bill'].between(10,20,inclusive=True)
```

Out[138]:

```
0       True
1       True
```

```
2      False
3      False
4      False
       ...
239    False
240    False
241    False
242     True
243     True
Name: total_bill, Length: 244, dtype: bool
```

In [139]:

```
df[df['total_bill'].between(10,20,inclusive=True)]
```

```
C:\Users\Chromsy\AppData\Local\Temp\ipykernel_8784\1152120679.py:1: FutureWarning: Boolea
n inputs to the `inclusive` argument are deprecated in favour of `both` or `neither`.
  df[df['total_bill'].between(10,20,inclusive=True)]
```

Out[139]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 341 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 | 3.45 | Douglas Tucker | 4478071379779230 | Sun4608 | 923 |
| 8 | 15.04 | 1.96 | Male | No | Sun | Dinner | 2 | 7.52 | Joseph Mcdonald | 3522866365840377 | Sun6820 | 037 |
| 9 | 14.78 | 3.23 | Male | No | Sun | Dinner | 2 | 7.39 | Jerome Abbott | 3532124519049786 | Sun3775 | 978 |
| 10 | 10.27 | 1.71 | Male | No | Sun | Dinner | 2 | 5.14 | William Riley | 566287581219 | Sun2546 | 121 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 234 | 15.53 | 3.00 | Male | Yes | Sat | Dinner | 2 | 7.76 | Tracy Douglas | 4097938155941930 | Sat7220 | 193 |
| 235 | 10.07 | 1.25 | Male | No | Sat | Dinner | 2 | 5.04 | Sean Gonzalez | 3534021246117605 | Sat4615 | 760 |
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 390 |
| 242 | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 | 8.91 | Dennis Dixon | 4375220550950 | Sat17 | 095 |
| 243 | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 | 9.39 | Michelle Hardin | 3511451626698139 | Thur672 | 813 |

130 rows × 15 columns

# nlargest and nsmallest

In [141]:

```
df.nlargest(7,'tip')
```

Out[141]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 170 | 50.81 | 10.00 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 | 823 |
| 212 | 48.33 | 9.00 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 | 521 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23 | 39.42 | 7.58 | Male | No | Sat | Dinner | 4 | 9.86 | Lance Peterson | 3542584061609808 | Sat239 | 980 |
| 59 | 48.27 | 6.73 | Male | No | Sat | Dinner | 4 | 12.07 | Brian Ortiz | 6596453823950595 | Sat8139 | 059 |
| 141 | 34.30 | 6.70 | Male | No | Thur | Lunch | 6 | 5.72 | Steven Carlson | 3526515703718508 | Thur1025 | 850 |
| 183 | 23.17 | 6.50 | Male | Yes | Sun | Dinner | 4 | 5.79 | Dr. Michael James | 4718501859162 | Sun6059 | 916 |
| 214 | 28.17 | 6.50 | Female | Yes | Sat | Dinner | 3 | 9.39 | Marissa Jackson | 4922302538691962 | Sat3374 | 196 |

In [142]:

```
df.nsmallest(8,'tip')
```

Out[142]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 526 |
| 92 | 5.75 | 1.00 | Female | Yes | Fri | Dinner | 2 | 2.88 | Leah Ramirez | 3508911676966392 | Fri3780 | 639 |
| 111 | 7.25 | 1.00 | Female | No | Sat | Dinner | 1 | 7.25 | Terri Jones | 3559221007826887 | Sat4801 | 688 |
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 396 |
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 341 |
| 215 | 12.90 | 1.10 | Female | Yes | Sat | Dinner | 2 | 6.45 | Jessica Owen | 4726904879471 | Sat6983 | 947 |
| 237 | 32.83 | 1.17 | Male | Yes | Sat | Dinner | 2 | 16.42 | Thomas Brown | 4284722681265508 | Sat2929 | 550 |
| 75 | 10.51 | 1.25 | Male | No | Sat | Dinner | 2 | 5.26 | Kenneth Hayes | 213142079731108 | Sat5056 | 110 |

In [144]:

```
df.sort_values('tip',ascending=False).iloc[0:2] #another method for finding smallest and lastest in list but its not fast
```

Out[144]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 170 | 50.81 | 10.0 | Male | Yes | Sat | Dinner | 3 | 16.94 | Gregory Clark | 5473850968388236 | Sat1954 | 8236 | |
| 212 | 48.33 | 9.0 | Male | No | Sat | Dinner | 4 | 12.08 | Alex Williamson | 676218815212 | Sat4590 | 5212 | |

In [146]:

```
df.sort_values('tip',ascending=True).iloc[0:5]
```

Out[146]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 526 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 236 | 12.60 | 1.00 | Male | Yes | Sat | Dinner | 2 | 6.30 | Matthew Myers | 3543676378973965 | Sat5032 | 396 |
| 92 | 5.75 | 1.00 | Female | Yes | Fri | Dinner | 2 | 2.88 | Leah Ramirez | 3508911676966392 | Fri3780 | 639 |
| 111 | 7.25 | 1.00 | Female | No | Sat | Dinner | 1 | 7.25 | Terri Jones | 3559221007826887 | Sat4801 | 688 |
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 | 8.49 | Christy Cunningham | 3560325168603410 | Sun2959 | 341 |

# sample

In [149]:

```
df.sample(5) # it will return 5 random data
```

Out[149]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_four |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 232 | 11.61 | 3.39 | Male | No | Sat | Dinner | 2 | 5.80 | James Taylor | 6011482917327995 | Sat2124 | 7995 |
| 41 | 17.46 | 2.54 | Male | No | Sun | Dinner | 2 | 8.73 | David Boyer | 3536678244278149 | Sun9460 | 8149 |
| 213 | 13.27 | 2.50 | Female | Yes | Sat | Dinner | 2 | 6.64 | Robin Andersen | 580140531089 | Sat1374 | 1089 |
| 67 | 3.07 | 1.00 | Female | Yes | Sat | Dinner | 1 | 3.07 | Tiffany Brock | 4359488526995267 | Sat3455 | 5267 |
| 239 | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 | 9.68 | Michael Avila | 5296068606052842 | Sat2657 | 2842 |

In [151]:

```
df.sample(frac=0.1) # It will retun fraction here 10% of total data as sample
```

Out[151]:

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 159 | 16.49 | 2.00 | Male | No | Sun | Dinner | 4 | 4.12 | Christopher Soto | 30501814271434 | Sun1781 | 143 |
| 43 | 9.68 | 1.32 | Male | No | Sun | Dinner | 2 | 4.84 | Christopher Spears | 4387671121369212 | Sun3279 | 921 |
| 105 | 15.36 | 1.64 | Male | Yes | Sat | Dinner | 2 | 7.68 | David Price | 4029957452720 | Sat5106 | 272 |
| 237 | 32.83 | 1.17 | Male | Yes | Sat | Dinner | 2 | 16.42 | Thomas Brown | 4284722681265508 | Sat2929 | 550 |
| 138 | 16.00 | 2.00 | Male | Yes | Thur | Lunch | 2 | 8.00 | Jason Burgess | 3561461821942363 | Thur2710 | 236 |
| 14 | 14.83 | 3.02 | Female | No | Sun | Dinner | 2 | 7.42 | Vanessa Jones | 30016702287574 | Sun3848 | 757 |
| 102 | 44.30 | 2.50 | Female | Yes | Sat | Dinner | 3 | 14.77 | Heather Cohen | 379771118886604 | Sat6240 | 660 |
| 225 | 16.27 | 2.50 | Female | Yes | Fri | Lunch | 2 | 8.14 | Whitney Arnold | 3579111947217428 | Fri6665 | 742 |
| 10 | 10.27 | 1.71 | Male | No | Sun | Dinner | 2 | 5.14 | William Riley | 566287581219 | Sun2546 | 121 |
| 21 | 20.29 | 2.75 | Female | No | Sat | Dinner | 2 | 10.14 | Natalie Gardner | 5448125351489749 | Sat9618 | 974 |

| | total_bill | tip | sex | smoker | day | time | size | price_per_person | Payer Name | CC Number | Payment ID | last_fou |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 58 | 11.24 | 1.76 | Male | Yes | Sat | Dinner | 2 | 5.62 | Ivy Guerrero | 356078262105562 | Sat6683 | 398 |
| 199 | 13.51 | 2.00 | Male | Yes | Thur | Lunch | 2 | 6.76 | Joseph Murphy MD | 6547218923471275 | Thur2428 | 127 |
| 215 | 12.90 | 1.10 | Female | Yes | Sat | Dinner | 2 | 6.45 | Jessica Owen | 4726904879471 | Sat6983 | 947 |
| 217 | 11.59 | 1.50 | Male | Yes | Sat | Dinner | 2 | 5.80 | Gary Orr | 30324521283406 | Sat8489 | 340 |
| 205 | 16.47 | 3.23 | Female | Yes | Thur | Lunch | 3 | 5.49 | Carly Reyes | 4787787236486 | Thur8084 | 648 |
| 100 | 11.35 | 2.50 | Female | Yes | Fri | Dinner | 2 | 5.68 | Lori Lynch | 38558279384492 | Fri4106 | 449 |
| 74 | 14.73 | 2.20 | Female | No | Sat | Dinner | 2 | 7.36 | Ashley Harris | 501828723483 | Sat6548 | 348 |
| 157 | 25.00 | 3.75 | Female | No | Sun | Dinner | 4 | 6.25 | Laura Robles | 213158685144262 | Sun7015 | 426 |
| 221 | 13.42 | 3.48 | Female | Yes | Fri | Lunch | 2 | 6.71 | Leslie Kaufman | 379437981958785 | Fri7511 | 878 |
| 32 | 15.06 | 3.00 | Female | No | Sat | Dinner | 2 | 7.53 | Amanda Wilson | 213186304291560 | Sat1327 | 156 |
| 73 | 25.28 | 5.00 | Female | Yes | Sat | Dinner | 2 | 12.64 | Julie Holmes | 5418689346409571 | Sat6065 | 957 |
| 218 | 7.74 | 1.44 | Male | Yes | Sat | Dinner | 2 | 3.87 | Nicholas Archer | 340517153733524 | Sat4772 | 352 |
| 178 | 9.60 | 4.00 | Female | Yes | Sun | Dinner | 2 | 4.80 | Melanie Gray | 4211808859168 | Sun4598 | 916 |
| 158 | 13.39 | 2.61 | Female | No | Sun | Dinner | 2 | 6.70 | Ashley Boyd | 3571088058115021 | Sun982 | 502 |

In [ ]: