In [1]:
```
import pandas as pd
```

In [3]:
```
orders = pd.read_table('http://bit.ly/chiporders')

# "read_table" ile "read_csv" arasindaki temel fark seperator. birinci sinde "tab", ikincisinde "commo"dir.
```

In [4]:
```
orders.head()
```

Out[4]:

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98

In [9]:
```
user_cols=['user_id', 'age', 'gender', 'occupation', 'zip_code']
users = pd.read_table('http://bit.ly/movieusers', sep='|', header=None, names=user_cols)

#"read_table" ile iki kullanisli bilgi; "skiprows" ve "skipfooter"
```

In [10]:
```
users.head()
```

Out[10]:

	user_id	age	gender	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213

How do I select a pandas Series from a DataFrame?

In [11]:
```
import pandas as pd
```

In [12]:
```
ufo = pd.read_csv('http://bit.ly/uforeports')
```

In [13]:
```
type(ufo)
```

Out[13]: pandas.core.frame.DataFrame

In [15]:
```
ufo.head()
```

Out[15]:

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00

In [18]:
```
type(ufo['City'])  # ufo['City'] == ufo.City   (ikincisi tabiki daha k
ullanilisli)
```

Out[18]: pandas.core.series.Series

In [20]:
```
ufo.City.head()
```

Out[20]:
```
0                  Ithaca
1             Willingboro
2                 Holyoke
3                 Abilene
4    New York Worlds Fair
Name: City, dtype: object
```

In [ ]:
```
# 3 durumda "noktali" secenegi degil 'square brackets' kullanmak zorun
dayiz
#1 series header'da bosluk varsa
#2 series header herhangi bir "built-in attribute" ismi ile ayni ise (
shape gibi)
#3 yeni bir series olusturuyor isek
```

In [22]:
```
'ab'+'cd'
```

Out[22]: 'abcd'

In [26]:
```
ufo['Location'] = ufo.City + ',' + ufo.State
```

In [27]:
```
ufo.head()
```

Out[27]:

	City	Colors Reported	Shape Reported	State	Time	Location
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00	Ithaca,NY
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00	Willingboro,NJ
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00	Holyoke,CO
3	Abilene	NaN	DISK	KS	6/1/1931 13:00	Abilene,KS
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00	New York Worlds Fair,NY

Why do some pandas commands end with parentheses, and other commands don't?

In [28]:
```
import pandas as pd
```

In [30]:
```
movies = pd.read_csv('http://bit.ly/imdbratings')
```

In [32]:
```
movies.head()
```

Out[32]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....

In [33]:
```
movies.describe()
```

Out[33]:

	star_rating	duration
count	979.000000	979.000000
mean	7.889785	120.979571
std	0.336069	26.218010
min	7.400000	64.000000
25%	7.600000	102.000000
50%	7.800000	117.000000

50%	7.800000	117.000000
75%	8.100000	134.000000
max	9.300000	242.000000

In [34]:
```
movies.shape
```

Out[34]:
```
(979, 6)
```

In [36]:
```
movies.dtypes
```

Out[36]:
```
star_rating       float64
title              object
content_rating     object
genre              object
duration            int64
actors_list        object
dtype: object
```

In [ ]:
```
# head(), describe() -> methods (action oriented)
# .shape, .dtypes -> attrubutes (description like who you are)
```

In [42]:
```
movies.describe(include=['object'])
```

Out[42]:

	title	content_rating	genre	actors_list
count	979	976	979	979
unique	975	12	16	969
top	The Girl with the Dragon Tattoo	R	Drama	[u'Daniel Radcliffe', u'Emma Watson', u'Rupert...
freq	2	460	278	6

In [ ]:
```
# "."dan sonra tab tusuna basildiginda girilebilecek komutlar cikiyor.
# parantez icersinde herhangi bir yerde "shift+tab" basildiginda ise b
uraya girilebilecek "arguments"lar (4 kademeli).
```

How do I rename columns in a pandas DataFrame?

In [44]:
```
ufo = pd.read_csv('http://bit.ly/uforeports')
```

In [45]:
```
ufo.columns
```

Out[45]:
```
Index(['City', 'Colors Reported', 'Shape Reported', 'State', 'Time']
, dtype='object')
```

In [ ]:
```
# 1st Method
```

In [46]:
```
ufo.rename(columns = {'Colors Reported': 'Colors_Reported', 'Shape Rep
orted': 'Shape_Reported'}, inplace=True)
```

In [ ]:
```
# "inplace=True" argument'i, sonucu goruntulemekle kalmaz, degerleri d
egistirir
```

In [47]:
```
ufo.columns
```

Out[47]:
```
Index(['City', 'Colors_Reported', 'Shape_Reported', 'State', 'Time']
, dtype='object')
```

In [ ]:
```
# 2nd Method
```

In [49]:
```
ufo_cols = ['city', 'colors reported', 'shape reported', 'state', 'tim
e']
```

In [50]:
```
ufo.columns = ufo_cols
```

In [51]:
```
ufo.head()
```

Out[51]:

	city	colors reported	shape reported	state	time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00

In [52]:
```
# 3rd Method (differnt way of second method)
```

In [69]:
```
ufo = pd.read_csv('http://bit.ly/uforeports', names=ufo_cols, header=0
)
```

In [70]:
```
# 4th Method (sadece '_' gibi standart bir eklemek yapmak istiyorsak)
```

In [71]:
```
# ufo.columns = ufo.columns.str.replace(' ', '_')  ##################
# burada problem var
```

In [72]:
```
ufo.columns
```

Out[72]:
```
Index(['city', 'colors reported', 'shape reported', 'state', 'time']
, dtype='object')
```

How do I remove columns from a pandas DataFrame?

In [73]:
```
ufo = pd.read_csv('http://bit.ly/uforeports')
```

In [74]:
```
ufo.drop('Colors Reported', axis=1, inplace=True)
```

In [76]:
```
ufo.head()
```

Out[76]:

	City	Shape Reported	State	Time
0	Ithaca	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	OTHER	NJ	6/30/1930 20:00
2	Holyoke	OVAL	CO	2/15/1931 14:00
3	Abilene	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	LIGHT	NY	4/18/1933 19:00

In [77]:
```
ufo.drop(['City', 'State'], axis=1, inplace=True)
```

In [79]:
```
ufo.drop([0, 1], axis=0, inplace=True)
```

In [80]:
```
ufo.head()
```

Out[80]:

	Shape Reported	Time
2	OVAL	2/15/1931 14:00
3	DISK	6/1/1931 13:00
4	LIGHT	4/18/1933 19:00
5	DISK	9/15/1934 15:30
6	CIRCLE	6/15/1935 0:00

How do I sort a pandas DataFrame or Series?

In [81]:
```
import pandas as pd
```

In [82]:
```
movies = pd.read_csv('http://bit.ly/imdbratings')
```

In [ ]:
```
movies.title # == movies['title']" -> orjinal siralamayi verir
```

In [ ]:
```
movies.title.sort_values()  #yukselen/artan sirayla verir
```

In [87]:
```
movies['title'].sort_values(ascending=False).head()  # azalan sirayla
verir
```

Out[87]:
```
864               [Rec]
526                Zulu
615          Zombieland
677              Zodiac
955    Zero Dark Thirty
Name: title, dtype: object
```

In [88]:
```
type(movies.title.sort_values())
```

Out[88]:
```
pandas.core.series.Series
```

In [89]:
```
movies.sort_values('title', ascending=False).head()
```

Out[89]:

	star_rating	title	content_rating	genre	duration	actors_list

864	7.5	[Rec]	R	Horror	78	[u'Manuela Velasco', u'Ferran Terraza', u'Jorg...
526	7.8	Zulu	UNRATED	Drama	138	[u'Stanley Baker', u'Jack Hawkins', u'Ulla Jac...
615	7.7	Zombieland	R	Comedy	88	[u'Jesse Eisenberg', u'Emma Stone', u'Woody Ha...
677	7.7	Zodiac	R	Crime	157	[u'Jake Gyllenhaal', u'Robert Downey Jr.', u'M...
955	7.4	Zero Dark Thirty	R	Drama	157	[u'Jessica Chastain', u'Joel Edgerton', u'Chri...

In [ ]:
```
# tum tabloyu 'title' siralamasina bagli olarak gosterdi
```

In [ ]:
```
movies.sort_values(['content_rating', 'duration'])  # verdigimiz siray
la sort ederb
```

How do I filter rows of a pandas DataFrame by column value?

In [90]:
```
import pandas as pd
```

In [91]:
```
movies = pd.read_csv('http://bit.ly/imdbratings')
```

In [96]:
```
booleans = []
for length in movies.duration:
    if length >= 200:
        booleans.append(True)
    else:
        booleans.append(False)
```

In [97]:
```
booleans[0:5]
```

Out[97]:
```
[False, False, True, False, False]
```

In [98]:
```
len(booleans)
```

Out[98]:
```
979
```

In [99]:
```
is_long = pd.Series(booleans)
```

In [100]:
```
is_long.head()
```

Out[100]:
```
0    False
1    False
2     True
3    False
4    False
dtype: bool
```

In [101]:
```
movies[is_long]
```

Out[101]:

	star_rating	title	content_rating	genre	duration	actors_list
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
7	8.9	The Lord of the Rings: The Return of the King	PG-13	Adventure	201	[u'Elijah Wood', u'Viggo Mortensen', u'Ian McK...
17	8.7	Seven Samurai	UNRATED	Drama	207	[u'Toshir\xf4 Mifune', u'Takashi Shimura', u'K...
78	8.4	Once Upon a Time in America	R	Crime	229	[u'Robert De Niro', u'James Woods', u'Elizabet...
85	8.4	Lawrence of Arabia	PG	Adventure	216	[u"Peter O'Toole", u'Alec Guinness', u'Anthony...
142	8.3	Lagaan: Once Upon a Time in India	PG	Adventure	224	[u'Aamir Khan', u'Gracy Singh', u'Rachel Shell...
157	8.2	Gone with the Wind	G	Drama	238	[u'Clark Gable', u'Vivien Leigh', u'Thomas Mit...
204	8.1	Ben-Hur	G	Adventure	212	[u'Charlton Heston', u'Jack Hawkins', u'Stephe...
445	7.9	The Ten Commandments	APPROVED	Adventure	220	[u'Charlton Heston', u'Yul Brynner', u'Anne Ba...
						[u'Kenneth Branagh',

476	7.8	Hamlet	PG-13	Drama	242	u'Julie Christie', u'Dere...
630	7.7	Malcolm X	PG-13	Biography	202	[u'Denzel Washington', u'Angela Bassett', u'De...
767	7.6	It's a Mad, Mad, Mad, Mad World	APPROVED	Action	205	[u'Spencer Tracy', u'Milton Berle', u'Ethel Me...

In [ ]:
```
# this is the long wat to get our result
# "for loop" kullanmadan daha kisa bir komutla ayni sonucu alabiliriz
```

In [102]:
```
dis_long = movies.duration >= 200
dis_long.head()
```

Out[102]:
```
0    False
1    False
2     True
3    False
4    False
Name: duration, dtype: bool
```

In [104]:
```
# daha da kisa olarak tek satirda da ayni sonucu aliriz
```

In [ ]:
```
movies[movies.duration >= 200]
```

In [106]:
```
# bu kosulu saglayan satirlarda sadece belirli bir bilgiyi (column/ser
ies) gormek istiyorsak
```

In [108]:
```
movies[movies.duration >= 200].genre # or " ...200]['genre']"
```

Out[108]:
```
2          Crime
7      Adventure
17         Drama
78         Crime
85     Adventure
142    Adventure
157        Drama
204    Adventure
445    Adventure
476        Drama
630    Biography
```

```
767       Action
Name: genre, dtype: object
```

How do I apply multiple filter criteria to a pandas DataFrame?

In [110]:
```
# coklu kosul olusturmak istiyorsam "conditional" kullanmaliyim
# birinci husus, "and" yerine "&" (ampersand), "or" yerine "|" (pipe/v
ertical bar) kullanilir
# ikinci husus, her kosulu ayri parantezler icersinde ifade etmeliyiz
```

In [111]:
```
movies[(movies.duration >= 200) & (movies.genre == 'Drama')]
```

Out[111]:

	star_rating	title	content_rating	genre	duration	actors_list
17	8.7	Seven Samurai	UNRATED	Drama	207	[u'Toshir\xf4 Mifune', u'Takashi Shimura', u'K...
157	8.2	Gone with the Wind	G	Drama	238	[u'Clark Gable', u'Vivien Leigh', u'Thomas Mit...
476	7.8	Hamlet	PG-13	Drama	242	[u'Kenneth Branagh', u'Julie Christie', u'Dere...

In [112]:
```
# koseli parantez icersindeki kosullarin urunleri boolean, programa sa
tirlari gostermek icin
```

In [114]:
```
((movies.duration >= 200) & (movies.genre == 'Drama')).head()
```

Out[114]:
```
0    False
1    False
2    False
3    False
4    False
dtype: bool
```

In [116]:
```
# coklu "or" condition olusturmak istersek, uzun yol;
# movies[(movies.genre == 'Crime') | (movies.genre == 'Drama') | (movi
es.genre == 'Action')]
# kisa yolu asagidaki sekilde:
```

In [118]:
```
movies[movies.genre.isin(['Crime', 'Drama', 'Action'])].head()
```

Out[118]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino',

						u James Caan]
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....

Answers to questions

In [119]:
```
# What about reading from csv file only two columns and ignore others?
```

In [124]:
```
ufo = pd.read_csv('http://bit.ly/uforeports', usecols=['City', 'State'
]) #[0, 4]
ufo.columns
```

Out[124]:
```
Index(['City', 'State'], dtype='object')
```

In [125]:
```
# one way to speed up reading a csv file. aslinda neye benzedigini anl
aminin kisa yolu olarak ilk birkac satiri okuma
```

In [127]:
```
ufo = pd.read_csv('http://bit.ly/uforeports', nrows=3)
ufo
```

Out[127]:

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00

In [128]:
```
# How do DataFrame and Series work with regard to selecting individual
entries and iteration?
```

In [129]:
```
for c in ufo.City:
    print(c)
```

```
Ithaca
Willingboro
Holyoke
```

In [136]:
```
for index, row in ufo.iterrows():
    print(index, row.City, row.State)
```

```
0 Ithaca NY
1 Willingboro NJ
2 Holyoke CO
```

In [137]:
```
# What's the best way to drop every mom-numeric column from a DataFram
e?
```

In [138]:
```
drinks = pd.read_csv('http://bit.ly/drinksbycountry')
```

In [139]:
```
drinks.dtypes
```

In [139]:
```
drinks.dtypes
```

Out[139]:
```
country                          object
beer_servings                     int64
spirit_servings                   int64
wine_servings                     int64
total_litres_of_pure_alcohol    float64
continent                        object
dtype: object
```

In [141]:
```
import numpy as np
drinks.select_dtypes(include=[np.number]).dtypes
```

Out[141]:
```
beer_servings                     int64
spirit_servings                   int64
wine_servings                     int64
total_litres_of_pure_alcohol    float64
dtype: object
```

How do I use the axis parameter in pandas?

In [143]:
```
drinks.head()
```

Out[143]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcoh
0	Afghanistan	0	0	0	0.0
1	Albania	89	132	54	4.9
2	Algeria	25	0	14	0.7
3	Andorra	245	138	312	12.4
4	Angola	217	57	45	5.9

In [ ]:
```
drinks.drop('continent', axis=1)   # column cikartarak goruntuler, 'in
place=True' eklersek, kalici olarak siler
```

In [ ]:
```
drinks.drop(2, axis=0) # row cikartir
```

In [145]:
```
drinks.mean()  # burada default olarak "axis=0"dir, burada sutunlardan
asagi dogru calistir diyoruz
```

Out[145]:
```
beer_servings                   106.160622
spirit_servings                  80.994819
wine_servings                    49.450777
total_litres_of_pure_alcohol      4.717098
dtype: float64
```

In [150]:
```
drinks.mean(axis=0)
```

Out[150]:
```
beer_servings                   106.160622
spirit_servings                  80.994819
wine_servings                    49.450777
total_litres_of_pure_alcohol      4.717098
dtype: float64
```

In [148]:
```
# axis=0 derken aslinda asagiya dogru hareket et, tum verileri isleyim
bir satira dusur diyorum burdaki komutta
# yani "axis=0" iken dikey(vertical), "axis=1" iken yatay/horizantal h
esaplama yapar
# so, "axis" arguments decides the direction of the movement of operat
ion.
```

In [151]:
```
drinks.mean(axis=1).head()
```

Out[151]:
```
0      0.000
1     69.975
2      9.925
3    176.850
4     81.225
dtype: float64
```

In [152]:
```
drinks.mean(axis=1).shape  # (axis=1) == (axis='columns')
```

Out[152]:
```
(193,)
```

In [153]:
```
drinks.mean(axis=0).shape  # (axis=0) == (axis='index')
```

Out[153]:
```
(4,)
```

How do I use string methods in pandas?

In [154]:
```
'hello'.upper()
```

Out[154]:
```
'HELLO'
```

In [155]:
```
import pandas as pd
```

In [158]:
```
orders = pd.read_table('http://bit.ly/chiporders')
```

In [159]:
```
orders.head()
```

Out[159]:

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39

			Green Chili Salsa		
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98

In [ ]:
```
# pandas'ta str method kullanilmak istendiginde ".str" ifadesi eklenmelidir.
```

In [161]:
```
orders.item_name.str.upper().head()
```

Out[161]:
```
0             CHIPS AND FRESH TOMATO SALSA
1                                     IZZE
2                         NANTUCKET NECTAR
3    CHIPS AND TOMATILLO-GREEN CHILI SALSA
4                              CHICKEN BOWL
Name: item_name, dtype: object
```

In [162]:
```
orders.item_name.str.contains('CHICKEN').head()
```

Out[162]:
```
0    False
1    False
2    False
3    False
4    False
Name: item_name, dtype: bool
```

In [163]:
```
# onceki bolumlerde gectigi gibi, olusturulan bu boolean series, argument olarak kullanilabilir
```

In [166]:
```
orders = pd.read_table('http://bit.ly/chiporders')
```

In [170]:
```
orders[orders.item_name.str.contains('Chicken')].head()
```

Out[170]:

	order_id	quantity	item_name	choice_description	item_price
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
5	3	1	Chicken Bowl	[Fresh Tomato Salsa (Mild), [Rice, Cheese, Sou...	$10.98
11	6	1	Chicken Crispy Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables,...	$8.75
12	6	1	Chicken Soft Tacos	[Roasted Chili Corn Salsa, [Rice, Black Beans,...	$8.75
13	7	1	Chicken Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Rice,...	$11.25

In [171]:
```
# burada onemli olan komutlari ezberlemek degil, nereden bulacagini bilmek, ki o da basit
# API Reference'da "string handling" basligi altinda "pandas series" icin kullanilabilecek tum "str" komutlari var.
```

In [174]:
```
orders.choice_description.str.replace('[', '').head()
```

Out[174]:
```
0                                                   NaN
1                                           Clementine]
2                                                Apple]
3                                                   NaN
4    Tomatillo-Red Chili Salsa (Hot), Black Beans, ...
Name: choice_description, dtype: object
```