Kevin Markham – Pandas Video Series Summary

|  |  |
| --- | --- |
| import pandas as pd |  |
| pd.read\_table() | df = pd.read\_table('http://bit.ly/chiporders') |
|  | users = pd.read\_table('http://bit.ly/movieusers', sep='|', header=None, names=user\_cols)  #"read\_table" ile iki kullanisli bilgi; "skiprows" ve "skipfooter"  # headerlar yok ise, bir liste hazirlayip names= ile atayabiliyoruz |
| pd.read\_csv() |  |
| .head() |  |
| type() | type(df) ## type(df.clm) ## type(df[‘clm]) |
| .describe() | df.describe(include=['object']) |
| .shape |  |
| .dtypes | # head(), describe() -> methods (action oriented)  # .shape, .dtypes -> attrubutes (description like who you are) |
| .astype | df['clm\_a'] = df.clm\_a.astype(float)  df.clm\_a.str.contains('Chicken').astype(int) # Booleans |
| .index | df.index |
| Rename columns |  |
| .rename() | ufo.rename(columns = {'Colors Reported': 'Colors\_Reported', 'Shape Reported': 'Shape\_Reported'}, inplace=True) |
| .columns= | ufo.columns = ufo\_cols # “ufo\_cols” is a list |
| “names=” kyw | ufo = pd.read\_csv(file, names=ufo\_cols, header=0) |
| .str.replace() | ufo.columns = ufo.columns.str.replace(' ', '\_') |
| Remove column |  |
| .drop() | # df.drop(‘clm', axis=1, inplace=True)  # df.drop(['clm-1', 'clm-2'], axis=1, inplace=True)  # df.drop([0, 1], axis=0, inplace=True) |
| Sorting |  |
| .sort\_values() | # df.clm\_a.sort\_values() # artan sirayla verir  # df['clm\_a'].sort\_values(ascending=False) # azalan sirayla  # df.sort\_values('clm\_a', ascending=False)  # df.sort\_values([‘clm\_a, 'clm\_b']) |
| Filtering rows |  |
| df.clm\_a >= …. | dis\_long = movies.duration >= 200  movies[movies.duration >= 200]. ## daha kisa  ## bu islem for loop ile de yapilir ama gerek yok |
| & - | 🡪 conditionals for multiple filter | df[(df.clm\_a >= 200) & (df.clm\_b == 'Drama')]  ## [] icersindeki ifadelerin urunu boolean’dir |
| .isin() | df[df.clm\_a.isin(['Crime', 'Drama', 'Action'])] |
|  |  |
| (axis=1) == (axis='columns')  (axis=0) == (axis='index') | drinks.mean(axis=1).shape |
| .str.upper() / / | orders.item\_name.str.upper().head() |
| .str.contains() | orders[orders.item\_name.str.contains('Chicken')] |
| .str.replace() | orders.choice\_description.str.replace('[', '').str.replace(']', '') |
| groupby |  |
| .groupby() | df.groupby('clm\_a').clm\_b.mean()  df[df.clm\_a=='Africa'].clm\_b.mean() |
| .agg() | df.groupby('clm\_a').clm\_b.mean().agg(['count', 'mean', 'max', 'min']) |
| .plot() | df.groupby('clm\_a').mean().plot(kind='bar') |
| Exploring |  |
| .value\_counts(normalize=True) | yuzde olarak verir |
| .unique() | df.clm\_a.unique() |
| .nunique() | df.clm\_b.nunique() |
| pd.crosstab() | pd.crosstab(df.clm\_a, df.clm\_b) |
| Missing Values |  |
| .isnull() // .notnull() | df.isnull(). # NaN veriler “True” gosterilir  df.notnull() # NaN veriler “False” gosterilir  df.isnull().sum() |
| .dropna() | df.dropna(how='any', inplace=True)  df.dropna(subset=['clm\_a', 'clm\_b'], how='any')  ## herhangi bir verisi “Na Nolan satiri siler, “how=all” olur ise tum verileri NaN olan satirlari siler sadece. |
| dropna=False | df['clm\_a'].value\_counts(dropna=False)  ## missing values sayisini da verir. |
| .fillna() | df['clm\_a'].fillna(value='VARIOUS', inplace=True) |
| indexing |  |
| .loc[] |  |
| .set\_index() | df.set\_index('clm\_a', inplace=True) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |