Lesson 2

Mean means Average

Variance means standard variation

Median is value that divide dataset in two equal values

Mode is value that has highest frequency

Describtive statistics provides summaries about sample as approximation of population

Correlation is statistical measure that describes strength of relationship between two variables

**Formatting table**

#plt.style.use('seaborn-whitegrid')

#plt.rc('text', usetex=True)

#plt.rc('font', family='times')

#plt.rc('xtick', labelsize=10)

#plt.rc('font', size=12)

##plt.rc('ytick', labelsize=10)

**Create dataframe**

data = {'year': [2010, 2011, 2012, 2010, 2011, 2012, 2010, 2011, 2012],

'team': ['FCBarcelona', 'FCBarcelona', 'FCBarcelona', 'RMadrid', 'RMadrid', 'RMadrid', 'ValenciaCF',

'ValenciaCF', 'ValenciaCF'],

'wins': [30, 28, 32, 29, 32, 26, 21, 17, 19],

'draws': [6, 7, 4, 5, 4, 7, 8, 10, 8],

'losses': [2, 3, 2, 4, 2, 5, 9, 11, 11]

}

football = pd.DataFrame(

data, columns=['year', 'team', 'wins', 'draws', 'losses'])

football

**Read tabular data**

edu = pd.read\_csv('educ\_figdp\_1\_Data.csv',

na\_values=':', usecols=['TIME', 'GEO', 'Value'])

edu

**View Data**

edu.head()

**print last five rows**

edu.tail()

**print columns**

edu.columns

**print index**

edu.index

**describe table**

edu.describe()

**selection**

edu[‘Value’]

**slice rows from 10 to 13th position**

edu[10:14]

**select subset of columns and rows**

edu.loc[90:94,['TIME','GEO']]

**filtering data**

edu[edu['Value'] > 6.5].tail()

**missing values**

edu[edu['Value'].isnull()].head()

**manipulate data**

edu.max(axis=0)

**sorting**

edu.sort\_values(by='Value', ascending=False, inplace=True)

edu.head()

**grouping**

group = edu[['GEO', 'Value']].groupby('GEO').mean()

group.head()

**rearranging data**

filtered\_data = edu[edu['TIME'] > 2005]

pivedu = pd.pivot\_table(filtered\_data, values='Value',

index=['GEO'], columns=['TIME'])

pivedu.head()

**ranking data**

pivedu = pivedu.drop(['Euro area (13 countries)',

'Euro area (15 countries)',

'Euro area (17 countries)',

'Euro area (18 countries)',

'European Union (25 countries)',

'European Union (27 countries)',

'European Union (28 countries)'

], axis=0)

pivedu = pivedu.rename(

index={'Germany (until 1990 former territory of the FRG)': 'Germany'})

pivedu = pivedu.dropna()

pivedu.rank(ascending=False, method='first').head()

**plotting**

fig = plt.figure(figsize=(12, 5))

totalSum = pivedu.sum(axis=1).sort\_values(ascending=False)

totalSum.plot(kind='bar', style='b', alpha=0.4,

title='Total Values for Country')

plt.savefig('Totalvalue\_Country.png', dpi=300, bbox\_inches='tight')

plt.show()

my\_colors = ['b', 'r', 'g', 'y', 'm', 'c']

ax = pivedu.plot(kind='barh', stacked=True, color=my\_colors, figsize=(12, 6))

ax.legend(loc='center left', bbox\_to\_anchor=(1, 0.5))

plt.savefig('Value\_Time\_Country.png', dpi=300, bbox\_inches='tight')

plt.show()