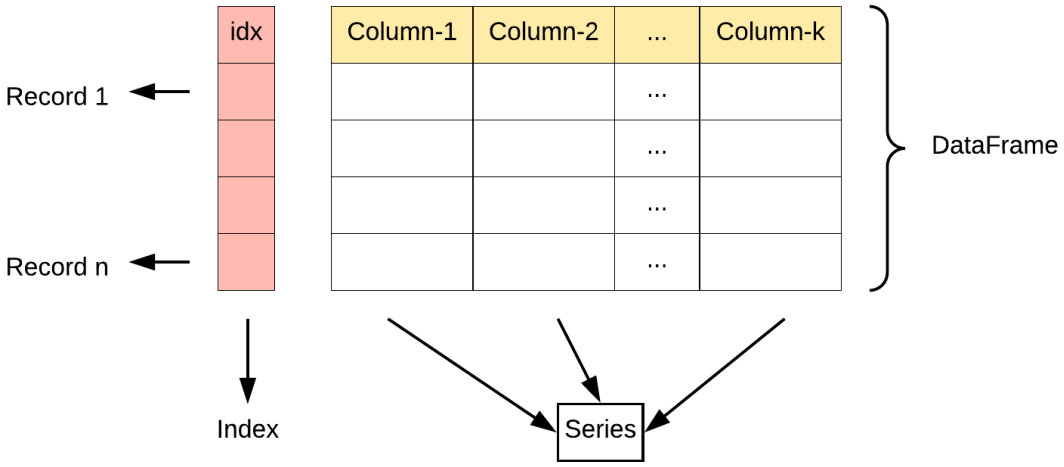
Chapter 0

- 基于Numpy和Matplotlib
- Pandas 的主要数据结构是 Series （一维数据）与 DataFrame（二维数据）
  - Series可以认为是一维数组(矩阵)，只有行索引
  - DataFrame可以认为是二维数组(矩阵)，有行和列索引



Chapter 1

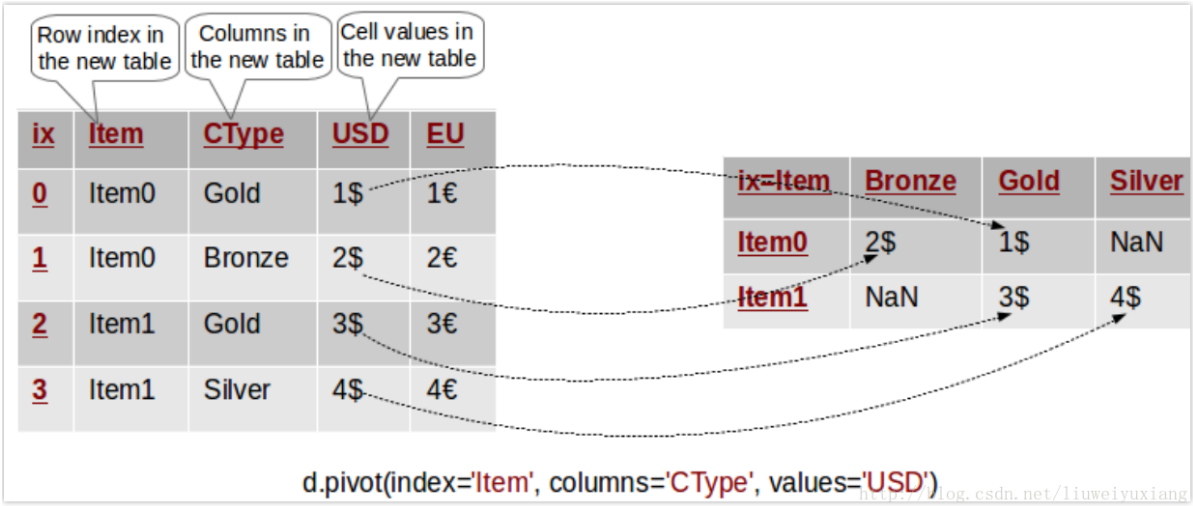
- 查看DataFrame
  https://campus.datacamp.com/courses/data-manipulation-with-pandas/transforming-dataframes?ex=2
- Sorting: **df.sort_values(["性别","地区"], ascending = [True, False])**
- Subsetting
  - subsetting columns
    https://campus.datacamp.com/courses/data-manipulation-with-pandas/transforming-dataframes?ex=6
  - subsetting rows: create logical conditions to filter rows
    - **df["地区"].isin(["东部","西部"])**
    - **df.loc[("东部", "上海"):("中部", "河南")]**
    - **df[df["性别"]=="女"]**
  - subsetting rows and columns
    - **df.loc[:, ["地区","性别"]]**
    - **df.iloc[[2:4, 0:3]]**
      https://campus.datacamp.com/courses/data-manipulation-with-pandas/transforming-dataframes?ex=7
- Adding
  - Adding columns: **df.["bmi"] = ...**

- Dropping duplicates
  - **df.drop_duplicates(subset = ["列1","列2"], keep='first', inplace=True)**
- Missing values
  - **df.isna()**
  - **df.isna().any()**
  - **df.isna().sum()**
  - **df.dropna()**
  - **df.fillna(0)**

Chapter 2

- Summary statistics
  - 连续变量
    - mean: **df.["体重"].mean()**
    - 其他描述性统计 .min() .max() .median() .mode() .var() .std() .sum()
    - 同时计算多个统计指标可以用.agg()
    - **df.["体重"].agg(["mean", "median"])**
    - **df.agg(func=None, axis=0, args,kwargs)**
  - 类别变量
    - **df["性别"].value_counts(normalize=False, sort=True, ascending=False, bins=None, dropna=True)**
- 分组计算描述性统计
  - **df.groupby("性别")["体重"].mean()**
  - **df.groupby("性别")["体重"].agg([min, max, sum])**
  - **df.groupby(["性别","地区"]).mean()**
  - **df.groupby(["性别","地区"]).agg([np.mean, np.median])**
- 用透视表计算描述性统计
  - The **pivot_table () function** is used to create a spreadsheet-style pivot table as a DataFrame
  - 一个分组：
    - **df.pivot_table(values="体重",index="性别")**
    - value --> columns want to summarize
    - index --> columns want to groupby
    - **df.pivot_table(values="体重",index="性别", aggfunc=np.median)**
    - **df.pivot_table(values="体重",index="性别", aggfunc=[np.mean, np.median])**
  - 一个以上分组：
    - **df.pivot_table(values="体重",index="性别",column="地区", full_value=0, margins=True)**

Chapter 3

- Indexing
- https://campus.datacamp.com/courses/data-manipulation-with-pandas/slicing-and-indexing-dataframes?ex=2
- Slicing: We can select specific ranges of our data in both the row and column directions using either label or integer-based indexing.
- loc：works on labels in the index.
- iloc：works on the positions in the index (so it only takes integers).
  - df.iloc[row slicing, column slicing]
  - df.iloc[0:10, :]
  - df.loc[0:10, :]
  - df.iloc[:, 0:10]
  - df.loc[:, 0:10]
- 透视表选区
  - pt = df.pivot_table(values=, index=, column=)
  - 

  - pt.loc[]
  - pt.mean(axis="index") --> calculate the statistic across rows
  - pt.mean(axis="column") --> calculate the statistic across columns

Chapter 4

- Visualizing
- https://campus.datacamp.com/courses/data-manipulation-with-pandas/creating-and-visualizing-dataframes?ex=2
  - df.hist(bins=5)
  - df.plot(kind="hist")
  - df.plot(x="", y="", kind="line", rot=45)
  - df.plot(x="", y="", kind="scatter")
  - df.isna().any().plot(kind="bar")
- Create DataFrame
- https://campus.datacamp.com/courses/data-manipulation-with-pandas/creating-and-visualizing-dataframes?ex=11

- list of dictionary
- dictionary of list
- import CSV file
    - **pd.read_csv()**