- The final method to create a DataFrame uses a dictionary containing Series, where the keys contain the column names.

Example 14.13 *s1 = Series(arange(0.0,5))*
s2 = Series(arange(1.0,3))
DataFrame({'one': s1, 'two': s2})

- Excel data can be read by read_excel.

Example 14.14 *from pandas import read_ excel*
gdp_ korea = read_ excel('gdp_ korea','Sheet1')
gdp_ korea['Q1-2011','Q2-2011']
gdp_ korea.iloc[0:3, 0:3]
gdp=gdp_ korea.iloc[0,:]

- drop(), dropna() and drop_duplicates() can all be used to drop rows or columns fromaDataFrame.
- fillna() fills NaN or other null values with other values.
- T and transpose are identical – both swap rows and columns of a DataFrame.

Example 14.15 *np.transpose(gdp_ korea)*

- sort_values and sort_index provide methods to sort a DataFrame. sort_values sorts the contents of the DataFrame along either axis using the contents of a single column.

Example 14.16 *gdp_ korea=np.transpose(gdp_ korea)*
gdp_ korea.sort_ index(ascending=0)

- pivot reshapes a table using column values.
- stack and unstack transforma DataFrame to a Series (stack) and back to a DataFrame (unstack).
- append appends rows of another DataFrame to the end of an existing DataFrame.
- concat is a core function which concatenates two or more DataFrames.
- reindex changes the labels while null-filling any missing values, which is useful for selecting subsets of a DataFrame or re-ordering rows.
- merge and join provide operations for merging the DataFrames using row labels or the contents of columns.
- apply executes a function along the columns or rows of a DataFrame.
- pivot_table provides a method to summarize data by groups.

- groupby produces a DataFrameGroupBy object which is a groupedDataFrame, and is usefulwhenaDataFrame has columns containing group data (e.g., sex or race in cross-sectional data).
- filter allows groups to be selected based on some function.

14.2 Statistical Functions

- count returns number of non-null values.
- describe provides a summary of the Series or DataFrame.
- Growth rates are computed using pct_change.

15 Custom Function and Modules

Functions are declared using the def keyword, and the value produced is returned using the return keyword.

Exercise 15.17 *def square(x):*
*return x**2*

 # call the function
x=2
y=square(x)
print(x,y)

Exercise 15.18 *import numpy as np*
def lp_ norm(x,y):
d = x-y
p=2
*return sum(abs(d)**p)**(1/p)*