# Import pandas

import pandas as pd

# Read 'Bronze.csv' into a DataFrame: bronze

bronze = pd.read\_csv('Bronze.csv')

# Read 'Silver.csv' into a DataFrame: silver

silver = pd.read\_csv('Silver.csv')

# Read 'Gold.csv' into a DataFrame: gold

gold = pd.read\_csv('Gold.csv')

# Print the first five rows of gold

print(gold.head())

==

# Import pandas

import pandas as pd

# Create the list of file names: filenames

filenames = ['Gold.csv', 'Silver.csv', 'Bronze.csv']

# Create the list of three DataFrames: dataframes

dataframes = []

for filename in filenames:

dataframes.append(pd.read\_csv(filename))

# Print top 5 rows of 1st DataFrame in dataframes

print(dataframes[0].head())

==

# Import pandas

import pandas as pd

# Make a copy of gold: medals

medals = gold.copy()

# Create list of new column labels: new\_labels

new\_labels = ['NOC', 'Country', 'Gold']

# Rename the columns of medals using new\_labels

medals.columns = new\_labels

# Add columns 'Silver' & 'Bronze' to medals

medals['Silver'] = silver['Total']

medals['Bronze'] = bronze['Total']

# Print the head of medals

print(medals.head())

==

# Import pandas

import pandas as pd

# Read 'monthly\_max\_temp.csv' into a DataFrame: weather1

weather1 = pd.read\_csv('monthly\_max\_temp.csv', index\_col='Month')

# Print the head of weather1

print(weather1.head())

# Sort the index of weather1 in alphabetical order: weather2

weather2 = weather1.sort\_index()

# Print the head of weather2

print(weather2.head())

# Sort the index of weather1 in reverse alphabetical order: weather3

weather3 = weather1.sort\_index(ascending=False)

# Print the head of weather3

print(weather3.head())

# Sort weather1 numerically using the values of 'Max TemperatureF': weather4

weather4 = weather1.sort\_values('Max TemperatureF')

# Print the head of weather4

print(weather4.head())

==

# Import pandas

import pandas as pd

# Reindex weather1 using the list year: weather2

weather2 = weather1.reindex(year)

# Print weather2

print(weather2)

# Reindex weather1 using the list year with forward-fill: weather3

weather3 = weather1.reindex(year).ffill()

# Print weather3

print(weather3)

==

# Import pandas

import pandas as pd

# Reindex names\_1981 with index of names\_1881: common\_names

common\_names = names\_1981.reindex(names\_1881.index)

# Print shape of common\_names

print(common\_names.shape)

# Drop rows with null counts: common\_names

common\_names = common\_names.dropna()

# Print shape of new common\_names

print(common\_names.shape)

==

# Extract selected columns from weather as new DataFrame: temps\_f

temps\_f = weather[['Min TemperatureF', 'Mean TemperatureF', 'Max TemperatureF']]

# Convert temps\_f to celsius: temps\_c

temps\_c = (temps\_f - 32) \* 5/9

# Rename 'F' in column names with 'C': temps\_c.columns

temps\_c.columns = temps\_c.columns.str.replace('F', 'C')

# Print first 5 rows of temps\_c

print(temps\_c.head())

==

import pandas as pd

# Read 'GDP.csv' into a DataFrame: gdp

gdp = pd.read\_csv('GDP.csv', parse\_dates=True, index\_col='DATE')

# Slice all the gdp data from 2008 onward: post2008

post2008 = gdp['2008-01-01':]

# Print the last 8 rows of post2008

print(post2008.tail(8))

# Resample post2008 by year, keeping last(): yearly

yearly = post2008.resample('A').last()

# Print yearly

print(yearly)

# Compute percentage growth of yearly: yearly['growth']

yearly['growth'] = yearly.pct\_change() \* 100

# Print yearly again

print(yearly)

==

# Import pandas

import pandas as pd

# Read 'sp500.csv' into a DataFrame: sp500

sp500 = pd.read\_csv('sp500.csv', parse\_dates=True, index\_col='Date')

# Read 'exchange.csv' into a DataFrame: exchange

exchange = pd.read\_csv('exchange.csv', parse\_dates=True, index\_col='Date')

# Subset 'Open' & 'Close' columns from sp500: dollars

dollars = sp500[['Open', 'Close']]

# Print the head of dollars

print(dollars.head())

# Convert dollars to pounds: pounds

pounds = dollars.multiply(exchange['GBP/USD'], axis='rows')

# Print the head of pounds

print(pounds.head())

==

# Import pandas

import pandas as pd

# Load 'sales-jan-2015.csv' into a DataFrame: jan

jan = pd.read\_csv('sales-jan-2015.csv', parse\_dates=True, index\_col='Date')

# Load 'sales-feb-2015.csv' into a DataFrame: feb

feb = pd.read\_csv('sales-feb-2015.csv', parse\_dates=True, index\_col='Date')

# Load 'sales-mar-2015.csv' into a DataFrame: mar

mar = pd.read\_csv('sales-mar-2015.csv', parse\_dates=True, index\_col='Date')

# Extract the 'Units' column from jan: jan\_units

jan\_units = jan['Units']

# Extract the 'Units' column from feb: feb\_units

feb\_units = feb['Units']

# Extract the 'Units' column from mar: mar\_units

mar\_units = mar['Units']

# Append feb\_units and then mar\_units to jan\_units: quarter1

quarter1 = jan\_units.append(feb\_units).append(mar\_units)

# Print the first slice from quarter1

print(quarter1.loc['jan 27, 2015':'feb 2, 2015'])

# Print the second slice from quarter1

print(quarter1.loc['feb 26, 2015':'mar 7, 2015'])

# Compute & print total sales in quarter1

print(quarter1.sum())

==

# Initialize empty list: units

units = []

# Build the list of Series

for month in [jan, feb, mar]:

units.append(month.Units) # or month[‘Units’]

# Concatenate the list: quarter1

quarter1 = pd.concat(units, axis='rows')

# Print slices from quarter1

print(quarter1.loc['jan 27, 2015':'feb 2, 2015'])

print(quarter1.loc['feb 26, 2015':'mar 7, 2015'])

==

# Add 'year' column to names\_1881 and names\_1981

names\_1881['year'] = 1881

names\_1981['year'] = 1981

# Append names\_1981 after names\_1881 with ignore\_index=True: combined\_names

combined\_names = names\_1881.append(names\_1981, ignore\_index=True)

# Print shapes of names\_1981, names\_1881, and combined\_names

print(names\_1981.shape)

print(names\_1881.shape)

print(combined\_names.shape)

# Print all rows that contain the name 'Morgan'

print(combined\_names.loc[combined\_names['name']=='Morgan'])

==

# Create a list of weather\_max and weather\_mean

weather\_list = [weather\_max, weather\_mean]

# Concatenate weather\_list horizontally

weather = pd.concat(weather\_list ,axis=1)

# Print weather

print(weather)

==

#Initialize an empyy list: medals

medals =[]

for medal in medal\_types:

# Create the file name: file\_name

file\_name = "%s\_top5.csv" % medal

# Create list of column names: columns

columns = ['Country', medal]

# Read file\_name into a DataFrame: medal\_df

medal\_df = pd.read\_csv(file\_name,header=0,index\_col='Country',names=columns)

# Append medal\_df to medals

medals.append(medal\_df)

# Concatenate medals horizontally: medals\_df

medals\_df = pd.concat(medals,axis='columns')

# Print medals\_df

print(medals\_df)

==

for medal in medal\_types:

file\_name = "%s\_top5.csv" % medal

# Read file\_name into a DataFrame: medal\_df

medal\_df = pd.read\_csv(file\_name, index\_col='Country')

# Append medal\_df to medals

medals.append(medal\_df)

# Concatenate medals: medals

medals = pd.concat(medals,keys=['bronze', 'silver', 'gold'])

# Print medals in entirety

print(medals)

==

# Sort the entries of medals: medals\_sorted

medals\_sorted = medals.sort\_index(level=0)

# Print the number of Bronze medals won by Germany

print(medals\_sorted.loc[('bronze','Germany')])

# Print data about silver medals

print(medals\_sorted.loc['silver'])

# Create alias for pd.IndexSlice: idx

idx = pd.IndexSlice

# Print all the data on medals won by the United Kingdom

print(medals\_sorted.loc[idx[:,'United Kingdom'], :])

==

# Concatenate dataframes: february

february = pd.concat(dataframes, axis=1, keys=['Hardware', 'Software', 'Service'])

# Print february.info()

print(february.info())

# Assign pd.IndexSlice: idx

idx = pd.IndexSlice

# Create the slice: slice\_2\_8

slice\_2\_8 = february.loc['Feb. 2, 2015':'Feb. 8, 2015', idx[:, 'Company']]

# Print slice\_2\_8

print(slice\_2\_8)

==

# Make the list of tuples: month\_list

month\_list = [('january', jan), ('february', feb), ('march', mar)]

# Create an empty dictionary: month\_dict

month\_dict = {}

for month\_name, month\_data in month\_list:

# Group month\_data: month\_dict[month\_name]

month\_dict[month\_name] = month\_data.groupby('Company').sum()

# Concatenate data in month\_dict: sales

sales = pd.concat(month\_dict)

# Print sales

print(sales)

# Print all sales by Mediacore

idx = pd.IndexSlice

print(sales.loc[idx[:, 'Mediacore'], :])

==

# Create the list of DataFrames: medal\_list

medal\_list = [bronze, silver, gold]

# Concatenate medal\_list horizontally using an inner join: medals

medals = pd.concat(medal\_list,keys=['bronze', 'silver', 'gold'],axis=1,join='inner')

# Print medals

print(medals)

==

# Resample and tidy china: china\_annual

china\_annual = china.resample('A').last().pct\_change(10).dropna()

# Resample and tidy us: us\_annual

us\_annual = us.resample('A').last().pct\_change(10).dropna()

# Concatenate china\_annual and us\_annual: gdp

gdp = pd.concat([china\_annual, us\_annual], join='inner', axis=1)

# Resample gdp and print

print(gdp.resample('10A').last())

==

# Merge revenue with managers on 'city': merge\_by\_city

merge\_by\_city = pd.merge(revenue, managers, on='city')

# Print merge\_by\_city

print(merge\_by\_city)

# Merge revenue with managers on 'branch\_id': merge\_by\_id

merge\_by\_id = pd.merge(revenue, managers, on='branch\_id')

# Print merge\_by\_id

print(merge\_by\_id)

==

# Merge revenue & managers on 'city' & 'branch': combined

combined = pd.merge(revenue, managers, left\_on='city', right\_on='branch')

# Print combined

print(combined)

==

# Add 'state' column to revenue: revenue['state']

revenue['state'] = ['TX', 'CO', 'IL', 'CA']

# Add 'state' column to managers: managers['state']

managers['state'] = ['TX', 'CO', 'CA', 'MO']

# Merge revenue & managers on 'branch\_id', 'city', & 'state': combined

combined = pd.merge(revenue, managers, on=['branch\_id', 'city', 'state'])

# Print combined

print(combined)

==

# Merge revenue and sales: revenue\_and\_sales

revenue\_and\_sales = pd.merge(revenue, sales, on=['city', 'state'], how='right')

# Print revenue\_and\_sales

print(revenue\_and\_sales)

# Merge sales and managers: sales\_and\_managers

sales\_and\_managers = pd.merge(sales, managers, left\_on=['city', 'state'], right\_on=['branch', 'state'], how='left')

# Print sales\_and\_managers

print(sales\_and\_managers)

==

# Perform the first merge: merge\_default

merge\_default = pd.merge(sales\_and\_managers, revenue\_and\_sales)

# Print merge\_default

print(merge\_default)

# Perform the second merge: merge\_outer

merge\_outer = pd.merge(sales\_and\_managers, revenue\_and\_sales, how='outer')

# Print merge\_outer

print(merge\_outer)

# Perform the third merge: merge\_outer\_on

merge\_outer\_on = pd.merge(sales\_and\_managers, revenue\_and\_sales, on=['city', 'state'], how='outer')

# Print merge\_outer\_on

print(merge\_outer\_on)

==

# Perform the first ordered merge: tx\_weather

tx\_weather = pd.merge\_ordered(austin, houston)

# Print tx\_weather

print(tx\_weather)

# Perform the second ordered merge: tx\_weather\_suff

tx\_weather\_suff = pd.merge\_ordered(austin, houston, on='date', suffixes=['\_aus', '\_hus'])

# Print tx\_weather\_suff

print(tx\_weather\_suff)

# Perform the third ordered merge: tx\_weather\_ffill

tx\_weather\_ffill = pd.merge\_ordered(austin, houston, on='date', suffixes=['\_aus', '\_hus'], fill\_method='ffill')

# Print tx\_weather\_ffill

print(tx\_weather\_ffill)

==

# Merge auto and oil: merged

merged = pd.merge\_asof(auto, oil, left\_on='yr', right\_on='Date')

# Print the tail of merged

print(merged.tail())

# Resample merged: yearly

yearly = merged.resample('A', on='Date')[['mpg', 'Price']].mean()

# Print yearly

print(yearly)

# print yearly.corr()

print(yearly.corr()) # 94.8677

==

#Import pandas

import pandas as pd

# Create file path: file\_path

file\_path = 'Summer Olympic medallists 1896 to 2008 - EDITIONS.tsv'

# Load DataFrame from file\_path: editions

editions = pd.read\_csv(file\_path, sep='\t')

# Extract the relevant columns: editions

editions = editions[['Edition', 'Grand Total', 'City', 'Country']]

# Print editions DataFrame

print(editions)

==

# Import pandas

import pandas as pd

# Create the file path: file\_path

file\_path = 'Summer Olympic medallists 1896 to 2008 - IOC COUNTRY CODES.csv'

# Load DataFrame from file\_path: ioc\_codes

ioc\_codes = pd.read\_csv(file\_path)

# Extract the relevant columns: ioc\_codes

ioc\_codes = ioc\_codes[['Country', 'NOC']]

# Print first and last 5 rows of ioc\_codes

print(ioc\_codes.head())

print(ioc\_codes.tail())

==

# Import pandas

import pandas as pd

# Create empty dictionary: medals\_dict

medals\_dict = {}

for year in editions['Edition']:

# Create the file path: file\_path

file\_path = 'summer\_{:d}.csv'.format(year)

# Load file\_path into a DataFrame: medals\_dict[year]

medals\_dict[year] = pd.read\_csv(file\_path)

# Extract relevant columns: medals\_dict[year]

medals\_dict[year] = medals\_dict[year][['Athlete', 'NOC', 'Medal']]

# Assign year to column 'Edition' of medals\_dict

medals\_dict[year]['Edition'] = year

# Concatenate medals\_dict: medals

medals = pd.concat(medals\_dict, ignore\_index=True)

# Print first and last 5 rows of medals

print(medals.head())

print(medals.tail())

==

# Construct the pivot\_table: medal\_counts

medal\_counts = medals.pivot\_table(index='Edition', values='Athlete', columns='NOC', aggfunc='count')

# Print the first & last 5 rows of medal\_counts

print(medal\_counts.head())

print(medal\_counts.tail())

==

# Set Index of editions: totals

totals = editions.set\_index('Edition')

# Reassign totals['Grand Total']: totals

totals = totals['Grand Total']

# Divide medal\_counts by totals: fractions

fractions = medal\_counts.divide(totals, axis='rows')

# Print first & last 5 rows of fractions

print(fractions.head())

print(fractions.tail())

==

# Apply the expanding mean: mean\_fractions

mean\_fractions = fractions.expanding().mean()

# Compute the percentage change: fractions\_change

fractions\_change = mean\_fractions.pct\_change()\*100

# Reset the index of fractions\_change: fractions\_change

fractions\_change = fractions\_change.reset\_index()

# Print first & last 5 rows of fractions\_change

print(fractions\_change.head())

print(fractions\_change.tail())

==

# Import pandas

import pandas as pd

# Left join editions and ioc\_codes: hosts

hosts = pd.merge(editions, ioc\_codes, how='left')

# Extract relevant columns and set index: hosts

hosts = hosts[['Edition', 'NOC']].set\_index('Edition')

# Fix missing 'NOC' values of hosts

print(hosts.loc[hosts.NOC.isnull()])

hosts.loc[1972, 'NOC'] = 'FRG'

hosts.loc[1980, 'NOC'] = 'URS'

hosts.loc[1988, 'NOC'] = 'KOR'

# Reset Index of hosts: hosts

hosts = hosts.reset\_index()

# Print hosts

print(hosts)

==

# Import pandas

import pandas as pd

# Reshape fractions\_change: reshaped

reshaped = pd.melt(fractions\_change, id\_vars='Edition', value\_name='Change')

# Print reshaped.shape and fractions\_change.shape

print(reshaped.shape, fractions\_change.shape)

# Extract rows from reshaped where 'NOC' == 'CHN': chn

chn = reshaped[reshaped.NOC=='CHN']

# Print last 5 rows of chn with .tail()

print(chn.tail())

==

# Import pandas

import pandas as pd

# Merge reshaped and hosts: merged

merged = pd.merge(reshaped, hosts)

# Print first 5 rows of merged

print(merged.head())

# Set Index of merged and sort it: influence

influence = merged.set\_index('Edition').sort\_index()

# Print first 5 rows of influence

print(influence.head())

==

# Import pyplot

import matplotlib.pyplot as plt

# Extract influence['Change']: change

change = influence['Change']

# Make bar plot of change: ax

ax = change.plot(kind='bar')

# Customize the plot to improve readability

ax.set\_ylabel("% Change of Host Country Medal Count")

ax.set\_title("Is there a Host Country Advantage?")

ax.set\_xticklabels(editions['City'])

# Display the plot

plt.show()

==