**Module 3**

Here are a few helpful downloads for this module:

* [Video Transcripts](https://student.emeritus.org/courses/4765/files/2911992?wrap=1)
* [Download Video Transcripts](https://student.emeritus.org/courses/4765/files/2911992/download?download_frd=1)
* [Quick Reference Guide](https://student.emeritus.org/courses/4765/files/2911994?wrap=1)

**plotly**

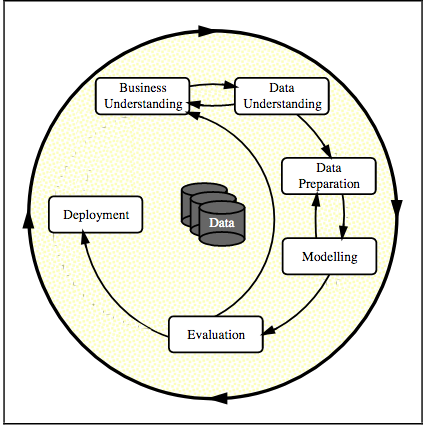
pip install --trusted-host pypi.org --trusted-host files.pythonhosted.org plotly

Done! Successfully installed plotly-5.6.0 tenacity-8.0.1

Cross-Industry Standard Process for Data Mining (CRISP-DM)

The CRISP-DM framework is important to data scientists for a variety of reasons, including:

* The methodology includes several processes that take care of simple data mining tasks
* It promotes best practices and facilitates the replication of projects
* It provides a uniform framework for planning and managing projects
* It is a cross-industry standard to be used by any data science project, regardless of its domain



df.sample(5)

df[df["column name"].isin(name of list)] or df.query(‘Entity in @list\_of\_countries’)

ans\_4 = df.query('minutes > 30 and points > 25')

df.query('team.str.contains("k")', engine = "python")

ans1 = gapminder.groupby('year')[['lifeExp']].agg('mean')

ans2 = gapminder.groupby('continent')[['gdpPercap']].agg('median')

ans3 = gapminder.groupby('continent')[['gdpPercap']].agg(['mean','median','std'])

ans4 = gapminder.groupby(gapminder['pop'] > 500\_000\_000)[['lifeExp']].agg('mean')

ans5a = gapminder.query('continent in ("Americas", "Europe")')

ans5b = ans5a.groupby(['continent', 'country'])[['lifeExp']].agg('mean')

churn\_pct = churn\_df['churn'].value\_counts(normalize = True)

**Issues**

1. Activity 3.2: Problem 8: fig.write\_image('images/plotly\_hist.png') requires missing kaleido package:

$ pip install -U kaleido. Skip

1. Activity 3.3: advanced, not straight forward
2. Activity 3.4 Problem 3 goal is not clear, Problem 4 has wrong input, Problem 5 is misleading
3. Activity 3.6 Instructions are not clear to keep what columns or not
4. Activity 3.8 Problem 8, plot is misleading, not clear instructions, Problem 9 is misleading too. Grading is not done, scores 0.
5. Activity 3.8 fails to grade!

**Jupiter environment update:**

#import sys

#!{sys.executable} -m pip install -U kaleido

**Install plotly**

CERTIFICATE\_VERIFY\_FAILED

import ssl

ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context

!jupyter kernelspec list

gapminder['lifeExp'].plot(kind='hist', bins=15, edgecolor='black', title='Histogram of Life Expectency')

sns.displot(gapminder,kind='kde',x='lifeExp', hue='continent', multiple="stack")

sns.boxplot(data=gapminder, x='lifeExp', y='gdpPercap', hue='continent')

#create seaborn boxplots by group

sns.boxplot(x='variable', y='value', data=df\_melted).set(title='Points by Team')

#modify axis labels

plt.xlabel('Team')

plt.ylabel('Points')

**Quizes**

In Python, what is the correct syntax for loading a dataframe from a csv file? : df=pd.read\_csv(filename)

*You are correct! The answer “*df=pd.read\_csv(filename)*” is correct because the statement calls pandas (pd) as well as the function [read\_csv()] and passes the filename inside the function parentheses.*

The function head() is used to show the last rows of the dataframe. : False

*That is correct! The answer “False” is correct because*head()*is used to show the top rows of the dataframe.*

In Python, what will df.sample(10) return? : Ten random rows of the dataframe df

*You are correct! The answer “*Ten random rows of the dataframe df*” is correct because the function*sample()*is used to print random rows of the data, and when given a parameter ‘10’, it will return ten random rows.*

For the dataframe shown in Video 3.2, you want to show only the rows of data from China. Which would be the correct syntax? : df[df[“Entity”]==”China”]

Dataframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Entity** | **Code** | **Year** | **GDP (constant 2010 US$)** |
| **6856** | Saint Lucia | LCA | 2009 | 1.404756e+09 |
| **1479** | Canada | CAN | 1983 | 8.031318e+11 |
| **6277** | Papua New Guinea | PNG | 1962 | 2.579601e+09 |
| **6073** | Oman | OMN | 1974 | 7.085347e+09 |
| **8279** | Uganda | UGA | 1992 | 5.751489e+09 |

*You are correct! The answer “*df[df[“Entity”]==”China”]*” is correct because the column Entity has the name of the countries, and the filtering syntax is correct.*

In Python, the function df.query() is used to apply filters on a dataframe. - True

*You are correct! The answer “True” is correct because the*query()*function is used to query the columns of a dataframe with a Boolean expression to filter the data.*

In Python, given the list

list\_of\_countries=[“UK”,”USA”,”China”,”Egypt”], what is the output of the statement

df.query(‘Entity in @list\_of\_countries’)? : Return all the entries within the dataframe where Entity is in list\_of\_countries

*You are correct! The answer “*Return all the entries within the dataframe where Entity is in list\_of\_countries*” is correct because the statement will return all the data in the dataframe that meets the filter criteria that the Entity column equals to any of the country names from the list list\_of\_countries.*

Which statement do you need to use in Python for the dataframe df to build a bar plot between the columns Entity and GDP? : df.plot(x=”Entity”,y=”gdp”,kind=”bar”)

*You are correct! The answer “*df.plot(x=”Entity”,y=”gdp”,kind=”bar”)*” is correct because the correct syntax to build a plot with the required x- and y-axes and plot are provided correctly.*

What is the function that rotates the x-axis labels of a plot built with Seaborn? : xticks(rotation)

*You are correct! The answer “*xticks(rotation)*” is correct because the function is used for the rotation of the x-axis labels of a plot.*

Which of the following is the function to plot a bar chart using Plotly? : fig=px.bar(df,x=”Entity”,y=”gdp”,color=”Entity”)

*You are correct! The answer “*fig=px.bar(df,x=”Entity”,y=”gdp”,color=”Entity”)*” is correct because this is the function for the Python library Plotly to draw a bar chart.*

Which function alters font size in a plot built with Plotly? : fig.update\_layout(font\_size=””)

!!!Ambigues!

*You are correct! The answer “*fig.update\_layout(font\_size=””)*” is correct because this is the function used to alter font size in a chart built by the Python library Plotly.*

The constructor “Title” in the Python statement

fig=px.kbar(df,x=”EntityPPp”,color=”Entity”,Title=”XYZ”)

is used to set the title of the legends in the chart. : False

*You are correct! The answer “False” is correct because the constructor “Title” is used to set the title of the whole chart, not the legends.*

The Python dataframe function groupby(’Entity’) allows splitting data into separate groups to perform computations for better analysis. : True

*You are correct! The answer “True” is correct because the function*groupby(’Entity’)*recognizes data so that rows with the same entity are clustered together.*

The agg(sum) followed by groupby(‘Entity’) sums all the values in the columns where the group of entities is matched. : True

*You are correct! The answer “True” is correct because the*agg(sum) *followed by* groupby(‘Entity’) sums*the values in all the columns of a dataframe for each group of Entity.*

What will be the output of the last column for df.groupby(‘Entity’).agg(sum)” when Entity is equal to China? : 86.88

**groupby("Entity")**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| China |  | 1960 |  | 1 |
|  |  |  |  |  |
| India |  | 1960 |  | 1 |
|  |  |  |  |  |
| USA |  | 1960 |  | 1 |
|  |  |  |  |  |
| China |  | 1990 |  | 6.48 |
|  |  |  |  |  |
| India |  | 1990 |  | 3.41 |
|  |  |  |  |  |
| USA |  | 1990 |  | 2.94 |
|  |  |  |  |  |
| China |  | 2017 |  | 79.4 |
|  |  |  |  |  |
| India |  | 2017 |  | 19.2 |
|  |  |  |  |  |
| USA |  | 2017 |  | 5.62 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| China |  | 1960 |  | 1 |
|  |  |  |  |  |
| China |  | 1990 |  | 6.48 |
|  |  |  |  |  |
| China |  | 2017 |  | 79.4 |
|  |  |  |  |  |

**agg(sum)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| China |  | 5961 |  |  |

Group of answer choices

*You are correct! The answer “86.88” is correct because for all the rows where Entity is equal to China, the sum of the last column can be calculated as 1 + 6.48 + 79.4 = 86.88, which is the correct answer.*

The function df.sort\_values(“Column”) returns the random values of a column. : False

*You are correct! The answer “False” is correct because the function is used to sort a dataframe in ascending or descending order on a passed column.*

For a dataframe df, given a column name gdp, what would be the statement in Python to sort the dataframe on gdp? : df.sort\_values(“gdp”)

*You are correct! The answer “*df.sort\_values(“gdp”)*” is correct because this is the correct syntax for applying a sorting function on the column gdp.*

A function in Python is a named section of a code that performs a specific task. : True

*You are correct! The answer “True” is correct because a function is a specific section of a code that involves taking some input, manipulating the input, and returning an output.*

Suppose you have the Python statement

df.groupby(“column”).agg(sum). Instead of using .agg(sum), what would be the built-in function to aggregate on a summation? : .sum()

*You are correct! The answer “*.sum()*” is correct because this is the built-in function used for aggregation on a summation.*

Given this dataframe, what is the Python statement to get the total GDP per year? : df.groupby(“Year”).agg(sum)

!!!Ambigues! Choices

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Entity** | **Code** | **Year** | **GDP (constant 2010 US$)** | **gdp** |
| **0** | Afghanistan | AFG | 2002 | 8.013233e+09 | 8.013233 |
| **1** | Afghanistan | AFG | 2003 | 8.689884e+09 | 8.689884 |
| **2** | Afghanistan | AFG | 2004 | 8.781610e+09 | 8.781610 |
| **3** | Afghanistan | AFG | 2005 | 9.762979e+09 | 9.762979 |
| **4** | Afghanistan | AFG | 2006 | 1.030523e+10 | 10.305228 |
| **...** | ... | ... | ... | ... | ... |
| **8864** | Zimbabwe | ZWE | 2013 | 1.418193e+10 | 14.181927 |
| **8865** | Zimbabwe | ZWE | 2014 | 1.448359e+10 | 14.483588 |
| **8866** | Zimbabwe | ZWE | 2015 | 1.472830e+10 | 14.728302 |
| **8867** | Zimbabwe | ZWE | 2016 | 1.481899e+10 | 14.818986 |
| **8868** | Zimbabwe | ZWE | 2017 | 1.532981e+10 | 15.329811 |

*You are correct! The answer “*df.groupby(“Year”).agg(sum)*” is correct because the statement groups the data on the column Year and aggregates the sum for each year tuple on all remaining columns.*

If dataframe df1 has an index Entity, and dataframe df2 has an index Year, will the statement df1/df2 be valid? : No

!!!Ambigues! The statement is true, semantics wrong!

*You are correct! The answer “No” is correct because the two dataframes have separate indexes that do not match. Hence, the division of the two dataframes will return NaN values, which is not a valid result.*

The Python function set\_index(“Column”) changes the original dataframe. : False

*You are correct! The answer “False” is correct because the function set\_index() does not actually change the original dataframe, just creates a copy.*

Suppose you have a dataframe df1 that has the columns year and gdp, with the index set as year. You also have a dataframe df2 that has columns labeled Year, Entity, and gdp.

To get the fraction of the GDP that each entity generates per year, what should be the index of df2? : set\_index([“Entity”,”Year”])

*You are correct! The answer “*set\_index([“Entity”,”Year”])*” is correct because setting the index at Year and Entity will generate a result for the fraction of the GDP that each entity generates per year.*

The Python function dropna() is used to drop all the rows that have NaN values. : True

*You are correct! The answer “True” is correct because in the pandas dataframe, the*dropna()*function is used to remove rows and columns with null/NaN values.*

Given this dataframe and the Python statement px.line(df,x=”Year”,y=”gdp”,color=”?”), what should be inside the quotes for the color parameter to get separate colored lines for each country? : Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Entity** | **Code** | **Year** | **GDP (constant 2010 US$)** | **gdp** |
| **0** | Afghanistan | AFG | 2002 | 8.013233e+09 | 8.013233 |
| **1** | Afghanistan | AFG | 2003 | 8.689884e+09 | 8.689884 |
| **2** | Afghanistan | AFG | 2004 | 8.781610e+09 | 8.781610 |
| **3** | Afghanistan | AFG | 2005 | 9.762979e+09 | 9.762979 |
| **4** | Afghanistan | AFG | 2006 | 1.030523e+10 | 10.305228 |
| **...** | ... | ... | ... | ... | ... |
| **8864** | Zimbabwe | ZWE | 2013 | 1.418193e+10 | 14.181927 |
| **8865** | Zimbabwe | ZWE | 2014 | 1.448359e+10 | 14.483588 |
| **8866** | Zimbabwe | ZWE | 2015 | 1.472830e+10 | 14.728302 |
| **8867** | Zimbabwe | ZWE | 2016 | 1.481899e+10 | 14.818986 |
| **8868** | Zimbabwe | ZWE | 2017 | 1.532981e+10 | 15.329811 |

*You are correct! The answer “Entity” is correct because the constructor “color” should be divided into Entity to show separate color codes for each country.*

In Python, the function filter() returns a sequence from the iterable elements for which the function returns True. : True

*You are correct! The answer “True” is correct because Python's*filter()*is a built-in function that allows you to process an iterable and extract those items that satisfy a given condition.*

Given the dataframe and the Python statement df.groupby(“Entity”).filter(max\_gdp\_ratio\_gt\_10), which of the following entities of the data will be in the output? : China, India

**def max\_gdp\_ratio\_gt\_10(s):** **max\_gdp\_ratio = max(s["gdp\_ratio"])** **return max\_gdp\_ratio > 10**

**Dataframe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| China |  | 1960 |  | 1 |
|  |  |  |  |  |
| India |  | 1960 |  | 1 |
|  |  |  |  |  |
| USA |  | 1960 |  | 1 |
|  |  |  |  |  |
| China |  | 1990 |  | 6.48 |
|  |  |  |  |  |
| India |  | 1990 |  | 3.41 |
|  |  |  |  |  |
| USA |  | 1990 |  | 2.94 |
|  |  |  |  |  |
| China |  | 2017 |  | 79.4 |
|  |  |  |  |  |
| India |  | 2017 |  | 19.2 |
|  |  |  |  |  |
| USA |  | 2017 |  | 5.62 |

*You are correct! The answer “China” and “India” are correct because in the dataframe, both countries have a GDP ratio greater than ten for at least one row of data.*

The Python function set\_index(“Column”) changes the original dataframe. : False

The agg(sum) followed by groupby(‘Entity’) sums all the values in the columns where the group of entities is matched. : True

Suppose you have a dataframe df1 that has the columns year and gdp, with the index set as year. You also have a dataframe df2 that has columns labeled Year, Entity, and gdp.

To get the fraction of the GDP that each entity generates per year, what should be the index of df2? : set\_index([“Entity”,”Year”])

In Python, what will df.sample(10) return? : Ten random rows of the dataframe df

Suppose you have the Python statement df.groupby(“column”).agg(sum). Instead of using .agg(sum), what would be the built-in function to aggregate on a summation? : .sum()

The function df.sort\_values(“Column”) returns the random values of a column. : Flase

A function in Python is a named section of a code that performs a specific task. : True

For a dataframe df, given a column name gdp, what would be the statement in Python to sort the dataframe on gdp? : df.sort\_values(“gdp”)

In Python, what is the correct syntax for loading a dataframe from a csv file? : df=pd.read\_csv(filename)

If dataframe df1 has an index Entity, and dataframe df2 has an index Year, will the statement df1/df2 be valid? :

**Discussion Activity**

**3.1**

<http://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf>

Data projects we deal almost everyday, the methodology outlined in the paper sounded a mix of software development and project management lifecycle which made me automatically think of bunch of questions related to each phase as below:

**Business Understanding**

Describes what form of business area it is implementing, what are business objectives, success criteria? What information should be provided to help out. Should be it be reactive, proactive? Which business units should be involved, what products are affected in large scale?

**Data Understanding**

For the specific business area, do we have data? Where can we get data if missing, how do we onboard? What specific data elements is needed? What are the quality measures of this data?

**Data Preparation**

After listing what data elements is needed, what transformation logic is needed to form data, data mapping rules? Split, merge, normalization, transform, data standardization steps?

**Modeling**

How to define business understanding by Laying out prepared data, what sequences of data is defining business needs? What actions is needed to detect business need and what corrective actions should it take? What are the business rules to execute when there is a match or mismatch?

**Evaluation**

How closely modeling is serving to the business needs, feedback to business understanding if any shortfalls, or any new requirements arise

**Deployment**

Rolling to production if satisfies business needs or within acceptable threshold per business success criteria, where are all targeted deployment areas, use cases? Are they all covered?

Obviously, the process is standardization of project management just for data mining, however, it is a broader scale can be applied to any project management.

**3.2**

* Attach your visualization
* Explain the method you used to create the visualization
* Share what library you used
* Describe the results you found and what they tell you about the dataset you chose

Exploring

Get gapminder dataset

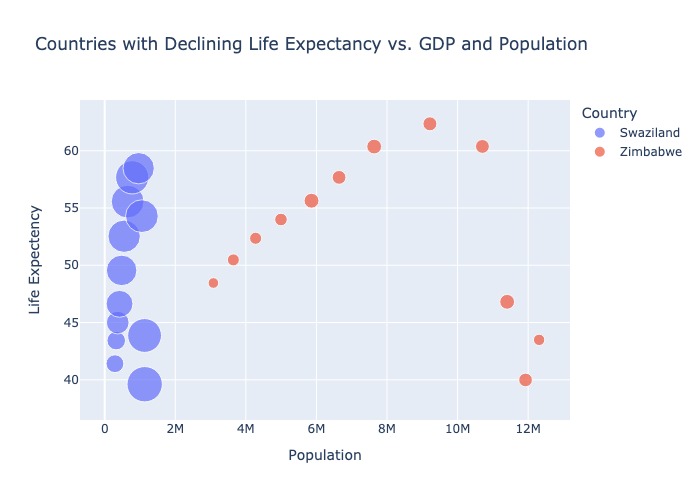
Show which countries having life expectancy decreasing over time

Correlate this finding to poverty! Look up if GDP is decreasing too!

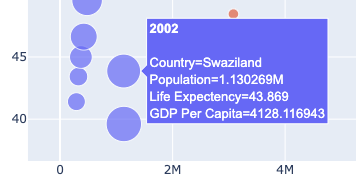
I found the gapminder dataset very intriguing, it contains year, continent, country, life expectancy, population and location. I analyzed data to see if there are any trends in country’s declining life expectancy with other features over time, I located 2 countries following the suit: Swaziland and Zimbabwe, both are in Africa.

px.scatter(gapminder.query('country in @list\_of\_countries'), x='pop', y='lifeExp', color='country', size='gdpPercap', size\_max=25, hover\_name='year', labels=dict(pop="Population", gdpPercap="GDP Per Capita", lifeExp="Life Expectency", country="Country"), title="Countries with Declining Life Expectancy vs. GDP and Population")

Visualization



I used plotly library to draw above scatter plot with hovering functionality to reveal all attributes for highlighting those features among declining life expectancy in two countries. The size of bubbles show how big their GDP per capita is, the x-axis is Population and y-axis is Life Expectancy. It is interactive displaying a pop up like:



Although, there is subtle or no correlation among those numeric features, there is an intriguing factor per country by looking at the x-axis and its hovering detail: For Swaziland, life expectancy started declining after the country’s population reached at 1M people and for Zimbabwe after reaching 10M people, perhaps these thresholds highlighting a point that each country’s infrastructure has a limit on how many people a country can serve effectively because the fluctuations in GDP per capita does not seem have an affect on life expectancy.

 ————— o —————