## Rep\_ Trial task

Big Five Personality Traits identified personality with 5 primary factors, known as \*Dimensions\* or \*Traits\*.

The 5 traits shortened for “OCEAN”' are:

* Openness to experience
* Conscientiousness
* Extroversion
* Agreeableness
* Neuroticism

Each dimension encompasses a large portion of personality-related terms. Known as \*Sub-Dimension\* / \*Sub-Traits\*.

Dimensions can present a high level interpretation of the person personality. Likewise, Sub-dimension can present a lower level more

Identified personality traits.

Dimension / Sub-dimension traits, clarified through a percentile value. Which is a way of expressing how personality scored relative to others on the same trait.

5 Trait percentiles can be estimated using Sub-traits questionnaire with 5 category answers. Each answer category has a Sub-percentile itself:

* category0 ⇾ 0% or 0
* category1 ⇾ 25% or 1
* category2 ⇾ 50% or 2
* category3 ⇾ 75% or 3
* category4 ⇾ 100% or 4

The mean percentile of all Sub-trait related to one of the 5 Trait, is the percentile value. We can normalize it to Z-Score equivalent

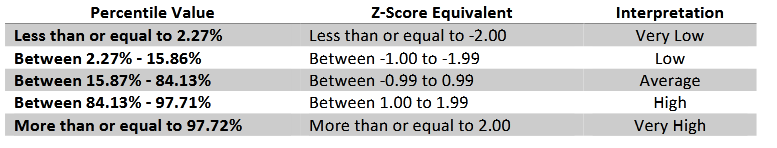
]…-2,…,2…[ using:

norm = 4 \* (value - min)/(max - min) -2

Or



while Xmin = 2.27% , Xmax = 97.72% based on the following table:



=============================================================

### Coding: (code can be found in the shared colab notebook:

<https://colab.research.google.com/drive/102IbIxnaYxBl5xBCEUYlCq4nGjWmnLwx?usp=sharing>

1. Find a suitable BIG 5 explanation that allows you to convert their suvey answers into the standard BIG 5 characteristics (extraversion, agreeableness, openness, conscientiousness, neuroticism) and ideally programatically generate these ratings from columns A to AX.

Changing directory to project folder

## Import needed Libraries, Import data from google sheet shared url

#After changing directory to project folder

# Import needed analysis libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

#import Regex library for text search and filtering

import re

# import google authenticate to import google Credentials

from google.colab import auth

auth.authenticate\_user()

from oauth2client.client import GoogleCredentials

# import gsread to import the data from the shared google sheet

import gspread

# store google Credentials in local variable which have open\_by\_url feature that can open shared sheet

gc = gspread.authorize(GoogleCredentials.get\_application\_default())

# The gc object has a function that we can use called open\_by\_url()

wb = gc.open\_by\_url('https://docs.google.com/spreadsheets/d/1rIYnJpVMHLy\_4\_YaBmj7ElhXOBntMkv0gMvAPw-mFV8/edit#gid=0')

sheet = wb.worksheet('Data')

data = sheet.get\_all\_values()

## prepare and moderatize data for analysis

# change the list of values in data to organized DataFrame

q\_data = pd.DataFrame(data)

# make first row column names and take the data less the header row.

q\_data = q\_data.rename(columns=q\_data.iloc[0])[1:].reset\_index()

# delete the old index new created column

q\_data = q\_data.drop('index',axis=1)

# have a summary look on the dataframe makeing sure no missing data

q\_data.info()

print('-=-='\*25)

q\_data.head(2)

print('Check answers categories options. which we need to replace with scores:\n')

#check available option to replace with score

list\_Ans\_Cat = set()

for x in q\_data.columns[0:1]:

list\_Ans\_Cat.update([x for x in q\_data[x]])

list\_Ans\_Cat

#create a reference dictionary for replace process

scores\_val = {'Very Inaccurate':0,

'Moderately Inaccurate':1,

'Neither Accurate Nor Inaccurate':2,

'Moderately Accurate':3,

'Very Accurate':4}

#apply and check output

q\_data.replace(scores\_val,inplace=True)#.astype(int)

print('All categorical answers replaced with equivalent values successfully:')

q\_data.head(2)

list(q\_data.columns)

# prepare dictionary to replace columns with their realated 5 traits

traits\_cat = {'[Am the life of the party.]':'extraversion',

'[Feel little concern for others.]':'rev\_agreeableness',

'[Am always prepared.]':'conscientiousness',

'[Get stressed out easily.]':'neuroticism',

'[Have a rich vocabulary.]':'openness',

"[Don't talk a lot.]":'rev\_extraversion',

'[Am interested in people.]':'agreeableness',

'[Leave my belongings around.]':'rev\_conscientiousness',

'[Am relaxed most of the time.]':'rev\_neuroticism',

'[Have difficulty understanding abstract ideas.]':'openness',

'[Feel comfortable around people.]':'extraversion',

'[Insult people.]':'rev\_agreeableness',

'[Pay attention to details.]':'conscientiousness',

'[Worry about things.]':'neuroticism',

'[Have a vivid imagination.]':'openness',

'[Keep in the background.]':'rev\_extraversion',

"[Sympathize with others' feelings.]":'agreeableness',

'[Make a mess of things.]':'rev\_conscientiousness',

'[Seldom feel blue.]':'neuroticism',

'[Am not interested in abstract ideas.]':'rev\_openness',

'[Start conversations.]':'extraversion',

"[Am not interested in other people's problems.]":'rev\_agreeableness',

'[Get chores done right away.]':'conscientiousness',

'[Am easily disturbed.]':'neuroticism',

'[Have excellent ideas.]':'openness',

'[Have little to say.]':'rev\_extraversion',

'[Have a soft heart.]':'agreeableness',

'[Often forget to put things back in their proper place.]':'rev\_conscientiousness',

'[Get upset easily.]':'neuroticism',

'[Do not have a good imagination.]':'rev\_openness',

'[Talk to a lot of different people at parties.]':'extraversion',

'[Am not really interested in others.]':'rev\_agreeableness',

'[Like order.]':'conscientiousness',

'[Change my mood a lot.]':'neuroticism',

'[Am quick to understand things.]':'openness',

"[Don't like to draw attention to myself.]":'rev\_extraversion',

'[Take time out for others.]':'agreeableness',

'[Shirk my duties.]':'rev\_conscientiousness',

'[Have frequent mood swings.]':'neuroticism',

'[Use difficult words.]':'openness',

"[Don't mind being the center of attention.]":'extraversion',

"[Feel others' emotions.]":'agreeableness',

'[Follow a schedule.]':'conscientiousness',

'[Get irritated easily.]':'neuroticism',

'[Spend time reflecting on things.]':'openness',

'[Am quiet around strangers.]':'rev\_extraversion',

'[Make people feel at ease.]':'agreeableness',

'[Am exacting in my work.]':'conscientiousness',

'[Often feel blue.]':'neuroticism',

'[Am full of ideas.]':'openness',

'Link to report generated from their online content':'Link to report generated from their online content',

'What is your sex?':'What is your sex?',

'What age bracket do you fall in?':'What age bracket do you fall in?',

'What is your country of residence?':'What is your country of residence?'}

# replace dataframe columns name using the created dictionary

q\_data.rename(columns=traits\_cat, inplace=True)

print('All columns are with Traits/rev\_traits name: \n')

q\_data.columns

# reverse score for rev traits columns

rev\_scores = {0:4,1:3,2:2,3:1,4:0}

col\_Unify = {'openness':'Openness','rev\_openness':'Openness',

'conscientiousness':'Conscientiousness','rev\_conscientiousness':'Conscientiousness',

'extraversion':'Extraversion','rev\_extraversion':'Extraversion',

'agreeableness':'Agreeableness','rev\_agreeableness':'Agreeableness',

'neuroticism':'Emotional range','rev\_neuroticism':'Emotional range'

}

# identify reverse columns names

rev\_columns = []

for cx in q\_data.columns:

if re.search('rev\_\*',cx.lower()):

rev\_columns.append(cx)

rev\_columns = set(rev\_columns)

# reverse score under rev\_traits columns

for x in rev\_columns:

q\_data[x] = q\_data[x].replace(rev\_scores)

#Unify columns

q\_data.rename(columns=col\_Unify, inplace=True)

# create new dataframe with percentile instead of answer categories

scor\_to\_perc = {0:0,1:25,2:50,3:75,4:100}

Perc\_q\_data = q\_data.replace(scor\_to\_perc).copy()

#create a users name list and modify dataframe index by it

User\_names = ['Evojanus','Deborahvilla9','andreatappe','bayrakmeltem','athena-thompson-techlever','belindawilson','simongphillips','pc40','alexayhat','DrLeighYourVetOnline']

Perc\_q\_data.index = User\_names

print('making sure only 5 Traits are there:\n\n')

set(Perc\_q\_data.columns)

print('reverse dataframe row/columns to have each user in a separate column, and Traits Average percentile as rows')

# reverse dataframe row/columns to have each users as a column

User\_Traits\_data = Perc\_q\_data[['Agreeableness',

'Conscientiousness',

'Emotional range',

'Extraversion',

'Openness']].T.reset\_index()

#calculate users Trait percentile

User\_Traits\_data = User\_Traits\_data.groupby('index').mean()

#create new columns for users Traits rows and we will us column name from User\_names in preivious cell

# first we calculate z\_score then normalize it to create row values

for x in User\_names:

User\_Traits\_data['%s\_row'%x] = 4\*((User\_Traits\_data[x]-2.27)/(97.72-2.27))-2

User\_Traits\_data['%s\_row'%x] = (User\_Traits\_data['%s\_row'%x]-User\_Traits\_data['%s\_row'%x].min())/(User\_Traits\_data['%s\_row'%x].max()-User\_Traits\_data['%s\_row'%x].min())

# create Interpretation columns based on np.select with a list of conditions and values for Interpretation Trait based on percentile

for x in User\_names:

temp = []

User\_Traits\_data['%s\_Interpretation'%x] = " "

for perc in User\_Traits\_data[x]:

temp.append(np.select([ (perc <= 2.27),

(perc > 2.27) & (perc <= 15.87),

(perc > 15.87) & (perc <= 84.13),

(perc > 84.13) & (perc <= 97.72),

(perc > 97.72)

] , ['Very Low', 'Low', 'Average', 'High', 'Very High']))

User\_Traits\_data['%s\_Interpretation'%x] = temp

#re-name columns

User\_Traits\_data.columns = ['Evojanus(%)', 'Deborahvilla9(%)', 'andreatappe(%)', 'bayrakmeltem(%)', 'athena-thompson-techlever(%)', 'belindawilson(%)', 'simongphillips(%)',

'pc40(%)', 'alexayhat(%)', 'DrLeighYourVetOnline(%)', 'Evojanus\_row',

'Deborahvilla9\_row', 'andreatappe\_row', 'bayrakmeltem\_row',

'athena-thompson-techlever\_row', 'belindawilson\_row',

'simongphillips\_row', 'pc40\_row', 'alexayhat\_row',

'DrLeighYourVetOnline\_row', 'Evojanus\_Interpretation',

'Deborahvilla9\_Interpretation', 'andreatappe\_Interpretation',

'bayrakmeltem\_Interpretation', 'athena-thompson-techlever\_Interpretation',

'belindawilson\_Interpretation', 'simongphillips\_Interpretation',

'pc40\_Interpretation', 'alexayhat\_Interpretation',

'DrLeighYourVetOnline\_Interpretation']

# check new columns applied successfully

User\_Traits\_data

print('Users Overall dimension plots')

for idx in User\_names:

sns.catplot(data=User\_Traits\_data,

x='%s(%%)'%idx,

y=['Agreeableness','Conscientiousness','Emotional range','Extraversion','Openness'],

kind='bar',

height=4,

aspect=3)

plt.title('\n\nOverall dimension plots for %s\n'%idx,fontsize=15)

plt.xlabel('Percentile Rank')

plt.ylabel('Personality Dimension')

for i, x in enumerate(User\_Traits\_data['%s(%%)'%idx]):

plt.text(x,i,x)

plt.autoscale()

plt.show()

temp = pd.DataFrame()

temp = User\_Traits\_data[['%s(%%)'%idx,'%s\_row'%idx,'%s\_Interpretation'%idx]]

temp.columns = ['Percentile','Raw','Interpretation']

print('%s values Interpretation:'%idx)

print(temp)

1. Compare ONLY agreeableness and concenciousness for this trial task you generated above to the values from their report generated from their public online content in column AY.

# to read pdf file

!pip install tabula-py

from tabula import read\_pdf

from tabula import convert\_into

# to import file from drive

!pip install PyDrive

from pydrive.auth import GoogleAuth

from pydrive.drive import GoogleDrive

#Authenticate and create the PyDrive client

gauth = GoogleAuth()

gauth.credentials = GoogleCredentials.get\_application\_default()

drive = GoogleDrive(gauth)

# using Regex filter link to extract files ids

pdf\_files\_ids = []

for x in q\_data['Link to report generated from their online content']:

print(re.search('\d.+/',x).group(0)[0:-1])

pdf\_files\_ids.append(re.search('\d.+/',x).group(0)[0:-1])

# using the list of ids import the files into colab with specific names in User\_names list

# after creating the pdf file read and save it using tabula-py to list

df\_all = []

for i,x in enumerate(pdf\_files\_ids):

downloaded = drive.CreateFile({'id':x})

downloaded.GetContentFile('%s\_report.pdf'%User\_names[i])

df\_all.append(read\_pdf('%s\_report.pdf'%User\_names[i], pages ='all'))

# now we have all report tabulas in the list

print('print Evojanus 5 Traits Interpretation table which is the first table \'0\'\nEvojanus index is 0 in the list(in the User names list)\n')

df\_all[0][0]

#extrat all and only the first 5 Traits tables from the df\_all list and save them in organized DataFrames to compare

df\_all\_5Trait = []

for x in df\_all:

df\_all\_5Trait.append(pd.DataFrame(x[0]))

# prepare dataframe by ignoring first row after making it as columns names

# also make the 5 Traits column as index. remembering to delete old index column

for i in range(len(df\_all\_5Trait)):

df\_all\_5Trait[i] = df\_all\_5Trait[i].rename(columns=df\_all\_5Trait[i].iloc[0])[1:].reset\_index().set\_index('Trait')

df\_all\_5Trait[i] = df\_all\_5Trait[i].drop('index',axis=1)

# delete % character from Percentile column and convert its values type to float

for User\_df in df\_all\_5Trait:

for i,x in enumerate(User\_df['Percentile']):

User\_df['Percentile'].iloc[i] = re.sub('%$','',x)

User\_df[['Percentile','Raw']] = User\_df[['Percentile','Raw']].astype(float)

# make sure all done succesfully with nothing missing

df\_all\_5Trait[0].info()

print('-=-='\*15,'\n\nAll users 5 Traits Interpretation tables are prepared, cleaned and ready...!\nExample first two tables:\n')

df\_all\_5Trait[0:2]

# we can get the Agreeableness, Conscientiousness as the first two values if sorted

df\_all\_5Trait[0].sort\_index()[0:2]#['Percentile'][0]

# create a comparison DataFrame

comparison\_df = pd.DataFrame(data=None,columns=['Agree\_rep','Agree\_ans','Agree\_compare','Consci\_rep','Consci\_ans','Consci\_compare'])

# fill with data and make comparison\_df based on the subtraction of the two Agreeableness percintile and Conscientiousness percintile.

for i, x in enumerate(User\_names):

comparison\_df = comparison\_df.append({'Agree\_rep':df\_all\_5Trait[i].sort\_index()[0:2]['Percentile'][0],

'Agree\_ans':User\_Traits\_data['%s(%%)'%x].sort\_index()[0:2][0],

'Consci\_rep':df\_all\_5Trait[i].sort\_index()[0:2]['Percentile'][1],

'Consci\_ans':User\_Traits\_data['%s(%%)'%x].sort\_index()[0:2][1]},ignore\_index=True)

comparison\_df['Agree\_compare']=np.abs(comparison\_df['Agree\_rep'] - comparison\_df['Agree\_ans'])

comparison\_df['Consci\_compare']=np.abs(comparison\_df['Consci\_rep'] - comparison\_df['Consci\_ans'])

# create category columns based on filtering and categorizing the comparison\_df columns

comparison\_df['Agree\_compare\_cat']= [np.select([(diff <= 5),

(diff > 5) & (diff <= 20),

(diff > 20) & (diff <= 40),

(diff > 40)

] , ['conforming', 'Convergent', 'far', 'Opposite']) for diff in comparison\_df['Agree\_compare']]

comparison\_df['Consci\_compare\_cat']= [np.select([(diff <= 5),

(diff > 5) & (diff <= 20),

(diff > 20) & (diff <= 40),

(diff > 40)

] , ['conforming', 'Convergent', 'far', 'Opposite']) for diff in comparison\_df['Consci\_compare']]

comparison\_df['Agree\_compare\_cat'] = comparison\_df['Agree\_compare\_cat'].astype(str)

comparison\_df['Consci\_compare\_cat'] = comparison\_df['Consci\_compare\_cat'].astype(str)

# add User names as index and create a User names columns for ploting

comparison\_df['User\_names'] = User\_names

comparison\_df.set\_index([pd.Index(User\_names)])

comparison\_df1 = comparison\_df.groupby(['User\_names','Agree\_compare\_cat','Consci\_compare\_cat']).mean()[['Agree\_compare','Consci\_compare']]

comparison\_df1.sort\_values(by='Agree\_compare')

comparison\_df1 = comparison\_df.groupby(['User\_names','Agree\_compare\_cat','Consci\_compare\_cat']).mean()[['Agree\_compare','Consci\_compare']]

ax = comparison\_df1.plot(kind='barh', figsize=(12, 8),zorder=2, width=0.5)

# Set x-axis label

ax.set\_xlabel("Traits Percinile Subtraction differences", labelpad=20, weight='bold', size=12)

# Set y-axis label

ax.set\_ylabel("User Traits Report/Ans comparison", labelpad=20, weight='bold', size=12)

ax.set\_yticklabels(comparison\_df1.index, Fontsize=12, weight='bold' )

plt.title('\n\nOverall Users comparison status (conforming,Convergent,far,Opposite)\n',fontsize=15)

plt.show()

sns.catplot(data=comparison\_df,

x='Agree\_compare',

y='User\_names',

kind='bar',

height=5, # make the plot 5 units high

aspect=2)

plt.title('\n\nOverall Users comparison Agreeableness,Conscientiousness repot/answers subtraction values\n',fontsize=15)

sns.catplot(data=comparison\_df,

x='Consci\_compare',

y='User\_names',

kind='bar',

height=5, # make the plot 5 units high

aspect=2)

plt.title('\n\nOverall Users comparison Agreeableness,Conscientiousness repot/answers subtraction values\n',fontsize=15)

plt.show()

\*\*Some Extra analytics of data\*\*

Extra\_df = User\_Traits\_data.T[0:10]

Extra\_df['Gender'] = list(Perc\_q\_data['What is your sex?'])

Extra\_df['country'] = list(Perc\_q\_data['What is your country of residence?'])

Extra\_df['Age\_group'] = list(Perc\_q\_data['What age bracket do you fall in?'])

#assure the numric column type to be float

Extra\_df[['Agreeableness', 'Conscientiousness', 'Emotional range', 'Extraversion','Openness']] = Extra\_df[['Agreeableness', 'Conscientiousness', 'Emotional range', 'Extraversion','Openness']].astype(float)

Extra\_df.info()

print(' Obviously we have lake of data as all Males are in only two age group\nPlus to no males from Albania nor Italy, Moreover no female from United States\n')

Extra\_df.groupby(['Gender','country','Age\_group']).mean()#.plot(kind='barh',figsize=(15,8))

# Extra\_df.groupby

# plt.figure(figsize=(10,5))

print('Below charts indecate that Agreeableness, Emotional range and Openness is higher in females then male\n ')

Extra\_df.groupby(['Gender']).mean().plot(kind='barh',figsize=(18,5))

for g in ['Agreeableness', 'Conscientiousness', 'Emotional range', 'Extraversion','Openness']:

sns.catplot(data=Extra\_df, x='Gender', y=g, kind='bar', height=3,aspect=1)

# Extra\_df.groupby

print('Below charts indecate that:\nAgreeableness is highest in age group 45-49 and 25-29\nConscientiousness is highest in 25-29 age group and lowest in 18-24 and 30-34\nEmotional range is highest in 18-24 and lowest in 30-34 \nExtraversion is high on all age groups\nOpenness is higher in females then male\n')

Extra\_df.groupby(['Age\_group']).mean().plot(kind='barh',figsize=(15,8))

for g in ['Agreeableness', 'Conscientiousness', 'Emotional range', 'Extraversion','Openness']:

sns.catplot(data=Extra\_df, x='Age\_group', y=g, kind='bar', height=3,aspect=1)

plt.show()

# Extra\_df.groupby

print('Below charts indecate that:\nAgreeableness and Conscientiousness is highest in Australia\nEmotional range is highest in Albania and lowest in United state\nExtraversion is highest in Australia and lowest in Italy\nOpenness is highest in Albania\n')

Extra\_df.groupby(['country']).mean().plot(kind='barh',figsize=(15,8))

for g in ['Agreeableness', 'Conscientiousness', 'Emotional range', 'Extraversion','Openness']:

sns.catplot(data=Extra\_df, x='country', y=g, kind='bar', height=3,aspect=1)

plt.show()