```
#
# Purdue University Global
#
# IN402 - Modeling and Predictive Analysis
#
# Unit 3 Assignment / Module 3 Part 1 Competency Assessment
#
# Predicting Gender-Based Salary Gap
#
# Jupyter Notebook Code
#

# [1] ********************************************************************************************************

# Data import and wrangling using multiple tools:

# Import all necessary initial libraries, including pandas, numpy, matplotlib, and seaborn

# For ignoring warning

import sys

# Ignoring warnings

if not sys.warnoptions:
    import warnings

warnings.simplefilter("ignore")

# [2] ********************************************************************************************************

import pandas as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import time

sns.set_style("whitegrid") # preferred seaborn style

# [3] ********************************************************************************************************

# Comment the line below if you are not using Jupyter. Leave uncommented if you are using PyCharm.

%matplotlib inline

# [4] ********************************************************************************************************

# Import and explore the quality of the dataset. What do you notice about the data?

df = pd.read_csv('/home/codio/workspace/data/IN402/data.csv')

df.dtypes

# [5] ********************************************************************************************************

# In the paper, describe the data source and how are you going to use the libraries

# Conduct exploratory data analysis.

# Examine the quality of the data

# what does the data look like? Use .head() method to explore first few rows

df.head() # (NOTE: if coding in PyCharms use print(df.head()) instead)

# [6] ********************************************************************************************************

# What does the data look like? Use .tail() method to explore last few rows

df.tail()

# [7] ********************************************************************************************************

# Check the structure/datatypes of each variable; are there any missing values?

# Identify using .info() method and remove (if any) using .dropna() method

# Are there any duplicate values? to detect use .duplicated() method

df[df.duplicated(keep=False)]

# [8] ********************************************************************************************************

# Check the descriptive statistics on numeric variables using .describe() method

df.describe()

# [9] ********************************************************************************************************

# Based on the initial observation, generate a Null hypothesis.

# Wrangle the data

# Create dummy variable for gender to allow the usage in the regression (1 for male and 0 for female)

df = pd.get_dummies(df, columns=['gender', 'edu'])

df.head() # (NOTE: if coding in PyCharms use print(df.head()) instead)

# [10] ********************************************************************************************************

# Group ages into 5 age groups

# Create new variable for natural log rate of base pay

# Create initial plots using matplotlib, seaborn and/or plotly to further understand the data

# Create scatterplots of the relationships between the features.

# In the paper, describe the initial state of the data, its quality, the wrangling techniques
# you've applied to transform the data, and why you needed to do that.

# Run the multiple regression

# Import all necessary libraries to run the regression, including statsmodel, sklearn.

import statsmodels.formula.api as sm

# [11] ********************************************************************************************************

# Write a code for the model

model = sm.ols(data=df, formula = "basePay ~ gender_Female + gender_Male +age+ seniority")

# [12] ********************************************************************************************************

# Fit the model into the data

result = model.fit()

result.summary()

# ********************************************************************************************************
```