Python Pandas – I expect to do at least couple of the tasks.

Run following code:

import pandas as pd

def get\_employees\_df():

  return pd.read\_csv(

      "https://gist.githubusercontent.com/kevin336/acbb2271e66c10a5b73aacf82"

        "ca82784/raw/e38afe62e088394d61ed30884dd50a6826eee0a8/employees.csv"

  )

def get\_departments\_df():

  dep\_df = pd.read\_csv(

      "https://gist.githubusercontent.com/kevin336/5ea0e96813aa88871c20d315b5"

        "bf445c/raw/d8fcf5c2630ba12dd8802a2cdd5480621b6a0ea6/departments.csv"

  )

  dep\_df = dep\_df.rename(columns={"DEPARTMENT\_ID": "DEPARTMENT\_IDENTIFIER"})

  return dep\_df

employees = get\_employees\_df()

departments = get\_departments\_df()

Tasks:

# 1. Please calculate the average, median, lower and upper quartiles of an employees' salaries.

print("average",employees['SALARY'].mean(),"|","median",employees['SALARY'].median(),"|","lower\_quartile",employees['SALARY'].quantile(0.25),"|","upper\_quartile",employees['SALARY'].quantile(0.75))

# 2. Please calculate the average salary per department. Please include the department name in the results.

avg\_salary\_per\_department = employees.groupby('DEPARTMENT\_ID')['SALARY'].mean().reset\_index()

print("Average Salary per Department:\n", avg\_salary\_per\_department)

# 3. Please create a new column named `SALARY\_CATEGORY` with value "low" when the salary is lower than average and "high" if is it higher or equal.

import numpy as np

average\_salary = employees['SALARY'].mean()

conditions = [

(employees['SALARY'] < average\_salary),

(employees['SALARY'] >= average\_salary)

]

results = ['low', 'high']

employees['SALARY\_CATEGORY'] = np.select(conditions, results)

print(employees)

# 4. Please create another column named `SALARY\_CATEGORY\_AMONG\_DEPARTMENT` with value "low" when the employee salary is lower than average in his / her department and "high" in the other case.

# 5. Please filter the dataframe `employees` to include only the rows where `DEPARTMENT\_ID` equals to 20. Assign the result to new variable.

employees\_dept\_20 = employees[employees['DEPARTMENT\_ID'] == 20]

print(employees\_dept\_20)

# 6. Please increase the salary by 10% for all employees working at the department 20.

def increase\_salary(row):

if row['DEPARTMENT\_ID'] == 20:

return row['SALARY'] \* 1.10

return row['SALARY']

employees['SALARY'] = employees.apply(increase\_salary, axis=1)

print(employees)

# 7. Please check if any of the `PHONE\_NUMBER` column values are empty.

empty\_phone\_numbers = employees[employees['PHONE\_NUMBER'].isnull()]

print(empty\_phone\_numbers)