**[개인 과제] 데이터 분석을 위한 파이썬 및 통계 응용 코딩**

|  |  |
| --- | --- |
| **훈련과정명** | [iM뱅크] iM DiGital Banker Academy 2기 |
| **강의과목명** | 데이터 분석을 위한 파이썬 및 통계 응용 코딩 |
| **강 사 명** | 정지훈 |
| **과 제 명** | 깃허브 개발환경 설정 및 기초코딩 문제 |
| **제출 기한** | 10월 04일 오후 17:50 까지 제출  **[과제 제출시 주의사항]**  - 각 코드에 주의 사항이 있으니 확인 바랍니다. |
| **과제 설명** | 1. Github 개발환경 설정 및 깃 푸쉬 과정을 순서대로 설명할 수 있다.  - [서술형 문제 1] (10점)  2. Python 기초 문법, numpy, pandas, matplotlib, seaborn, plotly 코드를 작성할 수 있다.  - [코드 1~7] (각 10점), [코드 8] (20점)  3. 모든 답안은 20241004\_answer\_정지훈.py에서 작성한 후 제출한다.  4. ChatGPT 사용은 허락하지만, 수업 때 배운 내용이 아닌 내용으로 진행한 경우 -5점 감점  5. 20241004 폴더를 .zip파일로 묶은 후, 20241004\_정지훈.zip 형태로 만든 후 제출한다. |

[서술형 문제 1] (10점)

|  |
| --- |
| 1. Github에서 Repository를 생성한 후, virtualenv 개발 환경 설정 과정을 순차적으로 작성하세요.  2. repository 이름은 alpaco로 가정한다.  3. 가상의 파일 main.py를 만들어서 코드 작성을 한 후 alpaco에 코드를 업데이트하는 전 과정을 작성한다.  4. 계정 연동은 이미 되어 있는 것으로 가정한다.  5. virtualenv 라이브러리는 이미 설치되었다고 가정한다. |
|  |

[코드 1] (10점)

|  |
| --- |
| 1. 다음은 반복문 코드이다. |
| |  | | --- | | result = []  for i in range(10):  if i % 2 == 0:  result.append(i \* 2) |   이 코드를 List Comprehension으로 변환한다. |

[코드 2] (10점)

|  |
| --- |
| 2. 다음과 같은 Dictionary가 있다. 다음 출력값이 나오도록 코드를 작성하세요. |
| |  | | --- | | my\_dict = {'apple': 3, 'banana': 5, 'orange': 2} |   출력   |  | | --- | | apple: 3  banana: 5  orange: 2 | |

[코드 3] (10점)

|  |
| --- |
| 3. 다음과 같이 Series가 정의되어 있다. numpy.where만을 사용하여 출력값이 나오도록 코드를 작성하세요. |
| **확인**  - 주어진 Series에서 값이 30보다 크고 60보다 작은 경우에는 해당 값에 10을 더하고, 그 외의 값은 그대로 두시오.   |  | | --- | | series **=** pd**.**Series([25, 35, 45, 60, 75]) |   출력   |  | | --- | | 0 25  1 45  2 55  3 60  4 75  dtype: int64 | |

[코드 4] (10점)

|  |
| --- |
| 4. seaborn을 활용하여 iris 데이터를 불러온 후, index를 포함하여 데이터를 csv파일과 excel 파일로 내보낼 수 있다. |
| 주의사항  - 각 파일의 파일명은 code4\_jungjihoon.csv, code4\_ jungjihoon.xlsx 형태로 출력한다.  - 각 파일은 output/ 경로에 저장할 수 있도록 코드를 작성한다. 경로 오류시 부분 점수 없음 |

[코드 5] (10점)

|  |
| --- |
| 5. 다음과 같은 pandas 데이터 프레임이 있다. |
| 다음 코드는 숫자지만 모두 문자로 구성되어 있다. 03/02와 03/03 컬럼값만 int형으로 변경하여 출력한다.  - apply()와 사용자 정의함수만 사용하여 처리한다.  - 사용자 정의함수명 : rm\_comma   |  | | --- | | import pandas as pd  # 데이터프레임 생성  data = [      ["1,000", "1,100", '1,510'],      ["1,410", "1,420", '1,790'],      ["850", "900", '1,185'],  ]  columns = ["03/02", "03/03", "03/04"]  df = pd.DataFrame(data=data, columns=columns)  df.info() |   출력   |  | | --- | | <class 'pandas.core.frame.DataFrame'>  RangeIndex: 3 entries, 0 to 2  Data columns (total 3 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 03/02 3 non-null int64  1 03/03 3 non-null int64  2 03/04 3 non-null object  dtypes: int64(2), object(1)  memory usage: 200.0+ bytes | |

[코드 6] (10점)

|  |
| --- |
| 6. 다음과 같이 matplotlib을 활용하여 구현된 시각화를 seaborn 시각화 코드 변환하기. |
| **확인**  - 시각화는 출력하여 code6\_jungjihoon.png 형태로 출력한다.  - 범례 미출력시, 5점 감점  - 각 파일은 output/ 경로에 저장할 수 있도록 코드를 작성한다. 경로 오류시 부분 점수 없음   |  | | --- | | import yfinance as yf  import matplotlib.pyplot as plt  import seaborn  apple = yf.download("AAPL", start="2020-01-01", end = "2024-09-30")  fig, ax = plt.subplots()  ax.plot(apple['Open'], label = "Apple")  ax.legend()  plt.show() |   시각화 예시 |

[코드 7] (10점)

|  |
| --- |
| 7. 다음과 같이 시각화가 나오도록 코드를 작성한다. |
| **확인**  - 시각화는 출력하여 code7\_jungjihoon.png 형태로 출력한다.  - 각 파일은 output/ 경로에 저장할 수 있도록 코드를 작성한다. 경로 오류시 부분 점수 없음   |  | | --- | | import matplotlib.pyplot as plt  import seaborn as sns  # Seaborn의 tips 데이터셋 로드  tips = sns.load\_dataset("tips") |   시각화 예시 |

[코드 8] (최대 20점)

|  |
| --- |
| 8. plotly 라이브러리를 활용하여 다음과 같이 시각화가 나오도록 코드를 작성한다. |
| **확인**  - update\_layout과 update\_traces를 모두 활용하여 코드 작성 시 20점  - 둘 중 하나만 활용 시 15점  - 옵션 설정 개수는 자유  - 이미지는 출력하지 않아도 된다.   |  | | --- | | import yfinance as yf  import plotly.graph\_objects as go  # yfinance로 AAPL 주가 데이터 다운로드  apple = yf.download("AAPL", start="2024-05-01", end="2024-09-30") |   시각화 예시 |