빅데이터실습 중간고사 퀴즈

[1주차 퀴즈]

1. 빅데이터의 특징이 아닌 것을 고르시오.

1. 다양성(variety)

2. 유효(valid)

3. 속도(velocity)

4. 크기(volume)

2. 빅데이터 관련 기술 중 데이터를 분산 저장하기 위한 기술을 고르시오.

1. Hadoop

2. Tensorflow

3. pandas

4. matplotlib

5. spark

3. 4차 산업 혁명의 핵심 기술이 아닌 것을 고르시오.

1. 사물인터넷(IoT)

2. 반응형 웹(Responsive Web)

3. 모바일(Mobile)

4. 클라우드 컴퓨팅(Cloud Computing)

5. 빅데이터(Big data)

4. pandas에 대한 설명으로 틀린 것은?

1. 데이터 분석을 위한 세련된 데이터 구조와 여러 도구를 제공한다.

2. 기본 파이썬에 내장되어 있으므로, 별도 설치가 필요 없어서 편리하다.

3. numpy를 기반으로 구현되어 있다.

4. jupyter, matplotlib 등 타 데이터 분석 관련 라이브러리들과 호환이 잘 된다.

5. 데이터 분석을 위한 서드파티 라이브러리이다.

5. 빅데이터의 주요 처리 과정은 데이터 수집, 저장, 처리, 분석, 시각화로 구분된다.

1. 참

2. 거짓

[3주차 퀴즈]

1. 위와 같이 10개의 데이터를 저장하고 있는 Series 타입의 변수 sr이 있다고 가정하자.

sr에서 인덱스가 B, F만 선택하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. sr[['B','F']]

2. sr.B.F

3. sr[('B','F')]

4. sr(['B','F'])

5. sr['B','F']

2. 위와 같이 10개의 데이터를 저장하고 있는 Series 타입의 변수 sr이 있다고 가정하자.

'C'부터 'E'까지 선택하는 구문이 아닌 것을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. sr[['C','D','E']]

2. sr[2:5]

3. sr[3:5]

4. sr['C':'E']

3. Series와 DataFrame의 설명 중 틀린 것을 2개 고르시오.

1. DataFrame의 컬럼 인덱스는 순서가 존재하므로, 인덱스 슬라이싱(범위 색인)이 가능하다.

2. DataFrame의 로우 인덱스는 0축(axis = 0), 컬럼 인덱스는 1축이라고도 한다.

3. Series는 동일한 데이터타입의 값들만 저장할 수 있다.

4. DataFrame은 2차원 구조이다.

5. DataFrame은 Series의 묶음이며, 컬럼 단위로 구성된다.

4. 위와 같이 데이터를 저장하고 있는 DataFrame 타입의 변수 df이 있다고 가정하자.

df에서 시와 구 컬럼만 선택하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. df[['시','구']]

2. df[('시','구')]

3. df.loc[['시','구']]

4. df.iloc[['시','구']]

5. df['시':'구']

5. 위와 같이 데이터를 저장하고 있는 DataFrame 타입의 변수 df이 있다고 가정하자.

df에서 서울시 데이터만 선택하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. df['시'] == '서울'

2. df[df['시'] == '서울']

3. df.iloc['시' == '서울']

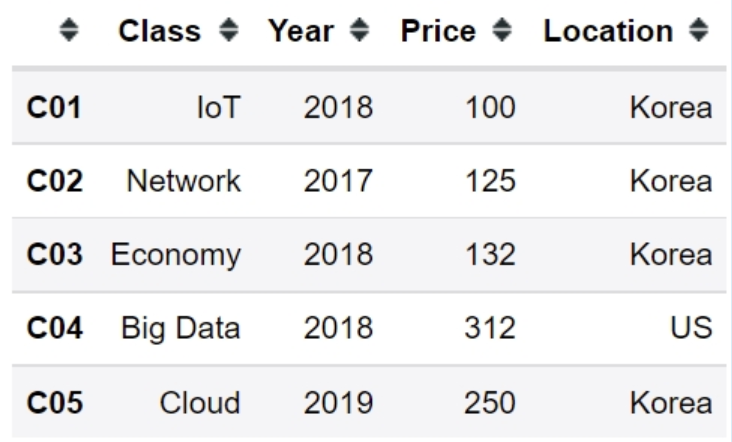
4. df == '서울시'

5. df.loc['시'] == '서울'

[4주차 퀴즈]

1. 위와 같은 데이터프레임 df가 있다고 가정하고,

Location 컬럼을 삭제하는 구문을 고르시오.



1. df.drop('Location', axis = 0)

2. df.drop('Location', level = 0)

3. df.drop('Location', axis = 1)

4. df.drop('Location', level = 1)

2. 위와 같은 sr1과 sr2가 있을 때, 원하는 결과를 얻기 위한 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. sr1.add(sr2, fill\_value = 0)

2. sr1.add(sr2, fill\_value = 1)

3. sr1.add(sr2)

4. sr1.append(sr2, fill\_value = 0)

5. sr1 + sr2

3. 다음 중 계층 색인에 대한 설명으로 틀린 것을 고르시오.

1. 계층 색인은 다중 색인(MultiIndex)라고도 한다.

2. 계층 색인에서 최상위 인덱스가 아닌 인덱스로 색인할 때에는 xs 함수를 사용한다.

3. 계층 색인으로 이루어진 인덱스의 값들은 튜플(Tuple) 타입이다.

4. 계층 색인은 컬럼 인덱스로만 사용 가능하다.

5. 계층 색인을 통해 DataFrame으로 3차원 이상의 고차원 데이터를 다룰 수 있다.

4. 위와 같은 데이터프레임 df가 있다고 가정하고,

limitStudent 컬럼(정원)을 추가하고, 값을 모두 30으로 저장하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. df.limitStudent.add= 30

2. df.limitStudent.add(30)

3. df['limitStudent'] =30

4. df.append(30)

5. df.add(30)

5. 위와 같은 데이터프레임 df가 있다고 가정하고,

Price가 200 이상이면, High, 200 미만이면 Low로 값을 할당하기 위해 아래과 같은 함수를 정의하였다. 이 함수를 활용하여 PriceLevel 컬럼을 추가하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

def get\_pLevel(X):

if X >= 200:

return 'High'

else:

return 'Low'

1. df['PriceLevel'] = df.Price.apply(get\_pLevel)

2. df['PriceLevel'] = df.apply(get\_plevel, axis = 1)

3. df['PriceLevel'] = get\_pLevel(df.Price)

4. df['PriceLevel'] = get\_pLevel(df)

5. df['PriceLevel'] = df.apply(get\_plevel)

[5주차 퀴즈]

1. 외부에서 제공해주는 데이터를 활용하는 방법과 관련된 설명 중 틀린 것을 고르시오.

1. API란 Application Programming Interface이다.

2. 웹사이트의 코드와 데이터를 직접 수집하는 것을 웹 스크래핑이라고 한다.

3. 파일을 다운로드 받아 내 PC에 저장한 후, read\_excel()이나 read\_csv() 함수 등을 활용하여 데이터를 읽어올 수 있다.

4. 웹 스크래핑을 자동화하여 정해진 규칙에 따라 여러 웹 사이트를 돌아다니면서 데이터를 수집하는 것으로 API라고 한다.

2. read\_csv() 함수의 인자명과 설명을 알맞게 짝지으세요.

encoding

인코딩 방식 지정

names

컬럼명 지정

sep

구분자 지정

header

컬럼명으로 사용할 줄(라인) 지정

skiprows

읽지 않을 줄(라인) 지정

chunksize

큰 데이터를 작은 단위로 잘라서 읽어오는 크기 지정

3. read\_excel() 함수는 엑셀 파일에서 데이터를 읽어올 때 사용하는 함수이다. 이 함수에서 sheet\_name = None으로 지정했을 때의 설명으로 맞는 것을 고르시오.

1. 마지막 시트의 데이터를 읽어오며, 결과는 사전(Dict) 타입으로 반환된다.

2. 첫번째 시트의 데이터를 읽어오며, 결과는 데이터프레임(DataFrame) 타입으로 반환된다.

3. 모든 시트의 데이터를 모두 읽어오며, 결과는 데이터프레임(DataFrame) 타입으로 반환된다.

4. 어떤 시트의 데이터도 읽어오지 않는다.

5. 모든 시트의 데이터를 모두 읽어오며, 결과는 사전(Dict) 타입으로 반환된다.

4. 웹스크래핑과 관련 없는 단어를 고르시오.

1. selenium

2. read\_html()

3. read\_database()

4. requests

5. read\_cliipboard()

[6주차 퀴즈]

1. 위와 같이 df1과 df2를 합치는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. pd.concat([df1, df2], ignore\_index = True)

2. df1.merge(df2)

3. pd.concat([df1, df2], ignore\_index = False)

4. pd.concat(df1, df2)

5. pd.merge(df1, df2)

2. 위와 같이 df1과 df2를 합치는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. pd.concat([df1, df2])

2. pd.merge(df1, df2, axis = 1)

3. df1.concat(df2)

4. pd.merge([df1, df2])

5. pd.merge(df1, df2)

3. 아래와 같이 도시/자치구별로 인구 정보를 저장한 데이터프레임 data가 있다고 가정하고,

서울 강남구 정보만 출력하는 구문을 고르시오

테이블이(가) 표시된 사진

자동 생성된 설명

1. data[(data.도시 == '서울') & (data.자치구 =='강남구')]

2. data[도시 == '서울' & 자치구 == '강남구']

3. data[자치구 == '강남구']

4. data['서울 강남구']

4. 아래와 같이 도시/자치구별로 인구 정보를 저장한 데이터프레임 data가 있다고 가정하고,

총인구수의 내림차순으로 정렬하여 출력하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. data.sort\_index('총인구', ascending = False)

2. data.sort('총인구', orderby ='descending')

3. data.sort\_values('총인구', orderby = 'desecending')

4. data.order\_by('총인구')

5. data.sort\_values('총인구', ascending = False)

5. 아래와 같이 도시/자치구별로 인구 정보를 저장한 데이터프레임 data가 있다고 가정하고,

도시/자치구별로 모든 년도의 총인구수 평균을 출력하는 구문을 고르시오.

테이블이(가) 표시된 사진

자동 생성된 설명

1. data.pivot\_table(index = ['도시','자치구'], aggfunc = 'mean', values = '총인구')

2. data.group\_table(index = ['도시','자치구'], agg = 'mean', values = '총인구')

3. data.group\_table(index = ['도시','자치구'], aggfunc = 'mean', values = '총인구')

4. data.pivot\_table(groupby = ['도시','자치구'], aggfunc = 'mean', values = '총인구')

5. data.pivot\_table(index = ['도시','자치구'], agg = 'mean', values = '총인구')