빅데이터 분석기사 실기 정리 문서

**Chap1. Pandas Practice**

▶ Check DataFrame Information

1. df.info()

- 각 Column의 Data type ( dtypes ) 정보.

- Null 값의 개수

- DataFrame Memory usage.

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  Index: 3500 entries, 0 to 3499  Data columns (total 9 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 총구매액 3500 non-null int64  1 최대구매액 3500 non-null int64  2 환불금액 1205 non-null float64  3 주구매상품 3500 non-null object  4 주구매지점 3500 non-null object  5 내점일수 3500 non-null int64  6 내점당구매건수 3500 non-null float64  7 주말방문비율 3500 non-null float64  8 구매주기 3500 non-null int64  dtypes: float64(3), int64(4), object(2)  memory usage: 273.4+ KB |

Table 1. df.info() 결과

2. df.describe()

- 수치형 데이터에 대한 요약 통계 제공 ( mean, std, min, quantile, max )

- 옵션(include=’object’)을 통해 문자열(object) 자료도 열람 가능.

|  |
| --- |
| 총구매액 최대구매액 ... 주말방문비율 구매주기  count 3.500000e+03 3.500000e+03 ... 3500.000000 3500.000000  mean 9.191925e+07 1.966424e+07 ... 0.307246 20.958286  std 1.635065e+08 3.199235e+07 ... 0.289752 24.748682  min -5.242152e+07 -2.992000e+06 ... 0.000000 0.000000  25% 4.747050e+06 2.875000e+06 ... 0.027291 4.000000  50% 2.822270e+07 9.837000e+06 ... 0.256410 13.000000  75% 1.065079e+08 2.296250e+07 ... 0.448980 28.000000  max 2.323180e+09 7.066290e+08 ... 1.000000 166.000000  [8 rows x 7 columns] |

Table 2. df.describe() 결과

|  |
| --- |
| 주구매상품 주구매지점  count 3500 3500 # 각 Column별 Row 개수  unique 42 24 # count 개수의 Row 중 고유한 값의 도메인 개수  top 기타 본 점 # 전체에서 가장 많이 존재하는 값  freq 595 1077 # top 값의 빈도(출현) 수 |

Table 3. df.describe( include=’object’ ) 결과

3. df.shape

- DataFrame의 행과 열을 튜플 형태로 확인. (ex. (20, 5) 🡪 20행 5열. (13, 10) 🡪 13행 10열. )

4. df.head(N)

- 첫 Row를 포함한 N개의 Row만 출력 (N 입력이 없으면 기본값 5 )

|  |
| --- |
| 총구매액 최대구매액 환불금액 주구매상품 ... 내점일수 내점당구매건수 주말방문비율 구매주기  cust\_id ...  0 68282840 11264000 6860000.0 기타 ... 19 3.894737 0.527027 17  1 2136000 2136000 300000.0 스포츠 ... 2 1.500000 0.000000 1  2 3197000 1639000 NaN 남성 캐주얼 ... 2 2.000000 0.000000 1  3 16077620 4935000 NaN 기타 ... 18 2.444444 0.318182 16  4 29050000 24000000 NaN 보석 ... 2 1.500000 0.000000 85  [5 rows x 9 columns] |

Table 4. df.head() 또는 df.head(5)의 결과

5. df.tail(N)

- 마지막 Row를 포함한 하위 5개의 Row만 출력 (N 입력이 없으면 기본값 5)

6. df.columns

- DataFrame의 Column 이름을 출력한다.

|  |
| --- |
| Index(['총구매액', '최대구매액', '환불금액', '주구매상품', '주구매지점', '내점일수', '내점당구매건수', '주말방문비율',  '구매주기'],  dtype='object') |

Table 5. df.columns 결과 (Column명 리스트를 볼 수 있다).

7. df.index

- DataFrame의 Index를 출력한다.

- pd.read\_csv() 함수에서 ‘index\_col=’ 옵션을 통해 인덱스를 설정할 수 있다.

ㆍ1개의 Column으로만 인덱스 설정 시 index\_col=0 또는 index\_col=’열이름1’ 처럼 적용 가능하다.

ㆍ2개 이상의 Column으로 구성하려면 리스트 형태로 입력한다 (index\_col = [0, 1] 또는 index\_col = [‘열1’, ‘열2’]

|  |
| --- |
| Index([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  ...  3490, 3491, 3492, 3493, 3494, 3495, 3496, 3497, 3498, 3499],  dtype='int64', name='cust\_id', length=3500) |

Table 6. ‘cust\_id’ Column을 Index로 구성한 DataFrame의 df.index 결과

8. df.dtypes

- DataFrame의 각 Column별 데이터 타입을 출력한다 (df.info() 결과에 포함된 값 중 하나).

|  |
| --- |
| 총구매액 int64  최대구매액 int64  환불금액 float64  주구매상품 object  주구매지점 object  내점일수 int64  내점당구매건수 float64  주말방문비율 float64  구매주기 int64  dtype: object |

Table 7. df.dtypes 결과 (‘object’는 문자열)

9. df.isnull()과 df.isnull().sum()

- df.isnull() 함수는 데이터프레임을 구성하는 각 열에서 Null(None or NaN)값이 있는 경우를 확인해

[‘True’, ‘False’]의 Table을 출력하는 함수.

|  |
| --- |
| 총구매액 최대구매액 환불금액 주구매상품 주구매지점 내점일수 …  cust\_id  0 False False False False False False …  1 False False False False False False ...  2 False False **True** False False False ...  3 False False **True** False False False ...  4 False False **True** False False False ...  ... ... ... ... ... ... ... ... ... |

Table 8. df.isnull() 결과 Table

- 특정 Column의 Null 여부만 확인하려면 df[‘열이름’].isnull() 방식으로 작성한다.

- 출력 결과에 .sum() 함수를 추가하여 Null 값이 존재하는 전체 Row 수를 Count할 수 있다.

|  |  |
| --- | --- |
| # df[‘환불금액’].isnull() 명령 결과  cust\_id  0 False  1 False  2 True  3 True  4 True  ...  3495 True  3496 False  3497 True  3498 True  3499 False  Name: 환불금액, Length: 3500, dtype: bool | # df.isnull().sum() 결과  총구매액 0  최대구매액 0  환불금액 2295  주구매상품 0  주구매지점 0  내점일수 0  내점당구매건수 0  주말방문비율 0  구매주기 0  dtype: int64 |

Table 9. 특정 Column의 isnull() 결과, 결측값(Null)이 존재하는 Row 수 계산 결과

10. df.nunique()

- 각 Column의 고유값의 수(Number of Unique value)를 출력한다.

|  |
| --- |
| 총구매액 3396  최대구매액 2576  환불금액 1114  주구매상품 42  주구매지점 24  내점일수 147  내점당구매건수 1107  주말방문비율 1142  구매주기 135  dtype: int64 |

Table 10. df.nunique() 결과

11. df.corr()

- 각 Column끼리의 상관관계를 계산하여 출력한다.

- 중요) ‘수치형’ 자료만 계산이 가능하므로 ‘범주형’ 자료의 경우 계산에서 제외시켜야 오류가 발생하지 않는다.

|  |
| --- |
| # 전처리 과정  df\_new = df.select\_dtypes( include= [‘float’, ‘int’] ) # float, int형의 데이터타입 Column만 가져와 새 df 생성.  print(df\_new.corr()) |
| 총구매액 최대구매액 환불금액 내점일수 내점당구매건수 주말방문비율 구매주기  총구매액 1.000000 **0.700080** 0.419734 **0.659084** 0.090022 0.014396 -0.212944  최대구매액 0.700080 1.000000 0.410562 0.374147 0.018980 0.022277 -0.115837  환불금액 0.419734 0.410562 1.000000 0.270290 -0.063114 -0.062397 -0.211125  내점일수 0.659084 0.374147 0.270290 1.000000 0.225264 -0.010325 -0.293200  내점당구매건수 0.090022 0.018980 -0.063114 0.225264 1.000000 0.007659 -0.091151  주말방문비율 0.014396 0.022277 -0.062397 -0.010325 0.007659 1.000000 0.003372  구매주기 -0.212944 -0.115837 -0.211125 -0.293200 -0.091151 0.003372 1.000000 |

Table 11. Column 간의 상관계수 결과 테이블

- 상관계수의 값은 -1 ~ 1 사이의 값을 가지며 다음과 같은 성격이 있다.

ㆍ -1 : 완벽한 음의 상관 ( 한 변수의 값이 증가할 때 다른 변수의 값은 감소).

ㆍ 0 : 상관관계가 없음 ( 두 변수의 선형적 관계가 없다 ).

ㆍ 1 : 완벽한 양의 상관 ( 한 변수의 값이 증가할 때 다른 변수의 값도 증가).

12. df[‘특정 열’].value\_counts()

- 특정 Column의 값의 빈도를 계산해서 출력한다.

- 항목 분류 등의 ‘범주형’ 변수의 Column에서 분석하기 용이하다.

|  |
| --- |
| # df[‘주구매상품’].value\_counts() |
| 주구매상품  기타 595  가공식품 546 … 일부 생략  골프 82  스포츠 69  일용잡화 64  모피/피혁 57  육류 57  남성 캐주얼 55  소형가전 2  악기 2  Name: count, dtype: int64 |

Table 12. ‘주구매상품’ Column의 데이터 빈도수 확인

▶ 사분위 수 (Quantile, Quartile) 정리.

- 사분위 수 계산은 DataFrame에서 quantile() 함수를 사용해 적용한다.

ㆍ Parameter로 분위수 값 ( 1사분: 0.25, 3사분: 0.75 )을 지정해 출력한다.

ㆍ 2분위 수의 경우 ‘중앙값’이므로 median() 함수를 적용해도 된다.

ㆍ 4분위 수의 경우 ‘최대값’이므로 max() 함수를 적용해도 된다.

- 하위 25% : 전체 데이터의 25%는 1사분위 수 이하의 값을 갖는다.

- 상위 25% (하위 75%) : 전체 데이터의 25%는 3사분위 수 이상의 값을 갖는다.

또는 전체 데이터의 75%는 3사분위 수 이하의 값을 갖는다.

- IQR (InterQuartileRange) : 사분위 수 범위 🡪 3사분 - 1사분 (Q3-Q1)

- 이상치(Outlier) 상/하한선 : Q3+1.5\*IQR (상한선). Q3-1.5\*IQR (하한선)

ㆍ상/하한선을 넘는 값의 경우 이상치로 간주할 수 있다.

|  |
| --- |
| # 임시 데이터 프레임 만들기 data = [1,1,2,3,4,5,6,6,6,7,8,9,9] dtmp = pd.DataFrame(data, columns=['value'])  print(f"사분위 수 0% : {dtmp['value'].**quantile(0)**}") print(f"min() 결과: {dtmp['value'].**min()**}") print(f"사분위 수 하위 25% : {dtmp['value'].**quantile(0.25)**}") print(f"median() 결과: {dtmp['value'].**median()**}") print(f"사분위 수 하위 50% : {dtmp['value'].**quantile(0.5)**}") print(f"사분위 수 하위 75% (상위 25%) : {dtmp['value'].**quantile(0.75)**}") print(f"max() 결과: {dtmp['value'].**max()**}") print(f"사분위 수 하위 100% : {dtmp['value'].**quantile(1.0)**}") |
| 사분위 수 0% : 1.0  min() 결과: 1  사분위 수 하위 25% : 3.0  median() 결과: 6.0  사분위 수 하위 50% : 6.0  사분위 수 하위 75% (상위 25%) : 7.0  max() 결과: 9  사분위 수 하위 100% : 9.0 |

Table 13. 사분위 수 테스트 코드와 실행 결과

▶ 데이터 정렬과 인덱싱

- sort\_values( [정렬할 기준 Columns], ascending= [True, False], inplace= [True, False] )

ㆍ 특정 Column을 기준으로 DataFrame 정렬하는 함수

- DataFrame 인덱스 활용

ㆍloc (Location) : 특정 열에 대한 ‘이름’, 특정 행에 대한 ‘조건’을 적용해 데이터에 접근할 수 있다.

ㆍiloc (Integer Location) : 특정 행과 열을 인덱스 번호인 ‘정수’ 값으로만 접근할 수 있다.

|  |
| --- |
| # 0~4번 인덱스의 [0, 1, 3, 4]번 째 Column에 대한 데이터프레임 행 가져오기.  # df.iloc[ 행시작:행끝+1, “Column인덱스번호”]로 구성되며 ‘ : ‘, ‘ [] ‘ 등의 구분 기호를 통해 인덱스를 다룬다.  # 정수 값의 인덱스 번호로만 파라미터를 구성할 수 있으므로 ‘조건’ ‘이름’ 등은 적용할 수 없다.   * **df.iloc[0:5, [0,1,3,4]]** |
| 총구매액 최대구매액 주구매상품 주구매지점  cust\_id  0 68282840 11264000 기타 강남점  1 2136000 2136000 스포츠 잠실점  2 3197000 1639000 남성 캐주얼 관악점  3 16077620 4935000 기타 광주점  4 29050000 24000000 보석 본 점 |

Table 14. iloc 데이터 접근법을 통해 인덱스 번호 기준의 데이터프레임 출력

|  |
| --- |
| # loc에서는 인덱스 슬라이싱 시 행 번호 ‘그대로’를 적용하므로 ‘행끝+1’이 아닌 ‘행끝’ 값 그대로를 입력한다.   * df.loc[0:4, [‘총구매액’, ‘최대구매액’, ‘주구매상품’, ‘주구매지점’]]   🡪 위의 출력 결과와 동일한 결과를 얻는다. |
| # 특정 행만 추출하기 위한 ‘조건식’을 추가해서 loc에 반영할 수 있다.  # ex. 30000000 이상의 ‘총구매액’을 갖는 Row만 추출하고, ‘총구매액’과 ‘주구매상품’, ‘주구매지점’ Column 확인.   * printdf = df.loc[df['총구매액'] >= 30000000, ['총구매액', '주구매상품', '주구매지점']] print(printdf[0:5])   총구매액 주구매상품 주구매지점  cust\_id  0 68282840 기타 강남점  7 514570080 명품 본 점  8 688243360 기타 본 점  10 245110760 디자이너 부산본점  13 103217000 시티웨어 분당점 |

Table 15. loc 데이터 접근법을 통해 ‘조건식’ 및 Column명 기준의 데이터프레임 출력

|  |
| --- |
| # ‘주구매지점’ 기준의 오름차순과 내림차순 결과의 데이터프레임 0~4번 인덱스 행 출력  ㆍ ascending= True 🡪 오름차순 정렬. <-> False 🡪 내림차순  ㆍ inplace= False 🡪 원본 데이터프레임은 변경 안 함.  True 🡪 원본 데이터프레임도 같이 변경.   * tmpdf = df.sort\_values(['주구매지점'], ascending=True, inplace=False) * print(tmpdf.iloc[0:4, [3, 4, 5, 6]]) # 특정 Column의 값들만 보기   주구매상품 주구매지점 내점일수 내점당구매건수  cust\_id  0 기타 강남점 19 3.894737  169 건강식품 강남점 3 1.666667  2697 시티웨어 강남점 1 2.000000  1750 피혁잡화 강남점 3 2.333333 |
| # 주의) loc를 이용해 출력하려고 행 파라미터에 0:3을 입력하면 큰 문제가 발생한다.  ㆍ loc는 인덱스 슬라이싱이 ‘라벨’ 기준이기 때문에 0:3은 첫 행에서 두 번째 행까지가 아니라  “0번 인덱스의 행부터 3번 인덱스의 행까지’로 인식한다.   * print(tmpdf.loc[0:4, ['주구매상품', '주구매지점', '내점일수', '내점당구매건수']])   주구매상품 주구매지점 내점일수 내점당구매건수  cust\_id  **0** 기타 강남점 19 3.894737  169 건강식품 강남점 3 1.666667  2697 시티웨어 강남점 1 2.000000  1750 피혁잡화 강남점 3 2.333333  451 농산물 강남점 3 1.333333  ... ... ... ... ...  3441 수산품 본 점 19 1.947368  3495 골프 본 점 1 2.000000  3493 농산물 본 점 68 3.852941  3489 기타 본 점 6 1.833333  **4**  보석 본 점 2 1.500000 |

Table 16. 데이터프레임 정렬과 인덱스 활용 주의 예시

▶ 결측값에 대한 처리

1. dropna()

- 결측값이 있는 데이터 위치를 기준으로 ‘행’, ‘열’ 기준의 제거.

- **axis=0** (행 단위 제거). **axis=1** (열 단위 제거)

ㆍ **axis=’index’** 🡪 행 단위 제거. **axis=’column’** 🡪 열 단위 제거.

- **how=’any’** 🡪 기본값. 1개라도 결측값이 있으면 기준(행/열) 단위로 제거.

**how=’all’** 🡪 기준(행/열) 단위로 전부 결측값이면 제거.

- **subset=[‘열1’, ‘열2’]** 🡪 특정 열에만 결측값이 있는지 검사.

- inplace= [True 또는 False] 🡪 처리 결과를 원본 데이터프레임에 반영할 것인지 여부.

|  |
| --- |
| # 특정 Column의 결측값이 있는 데이터 추출   * nadf = df[df['환불금액'].isnull() == True] 🡪 2295개의 Row가 나왔다고 한다.   # 결측값이 있는 Row를 ‘행’ 단위로 제거 후의 데이터프레임 확인   * nodf = df.dropna(axis=0, subset=['환불금액'], inplace=False) * nodf.info()   - 원본 df는 3,500개의 Row가 있었으므로 ‘환불금액’ Column에 결측값이 존재했던 2,295개의 Row가 삭제되어  총 1,205개의 Row가 출력된 것을 확인할 수 있다.  <class 'pandas.core.frame.DataFrame'>  Index: **1205** entries, 0 to 3499  Data columns (total 9 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 총구매액 1205 non-null int64  1 최대구매액 1205 non-null int64  2 환불금액 1205 non-null float64  3 주구매상품 1205 non-null object  4 주구매지점 1205 non-null object  5 내점일수 1205 non-null int64  6 내점당구매건수 1205 non-null float64  7 주말방문비율 1205 non-null float64  8 구매주기 1205 non-null int64  dtypes: float64(3), int64(4), object(2)  memory usage: 94.1+ KB |

Table 17. 결측값에 대한 dropna() 함수 처리 결과