[주요 라이브러리 정보]

라이브러리	내용	설명
pandas	라이브러리 찾기	dir(pd)
	pandas 중요 라이브러리	read_csv( ) df=get_dummies(df, columns=['A', B'], drop_first=True), df.A=pd.to_datetime(df.A)
	pd.DataFrame 주요함수	reset_index( ) sort_values(df.A, ascending=False) # 내림차순 groupby( ) # group 함수 idx_max( ) # max값 인덱스 찾기(idxmin), 매우유용함 nlargest( ) # n개의 큰값 찾기(nsmallest), 매우유용함

	pd
	주요함수
		pd.crosstab(df.A, df.B) # 빈도수 테이블 만들기
		pivot( ) # 데이터 값을 열이름으로 만들기
		pibot_table( )
		melt( )

	len(), count() 차이	DataFrame이나 Series의 길이를 구할 때 count()는 nan 값 개수를 제외
numpy	라이브러리 찾기	dir(np) # 많은 np 명령어를 찾을 수 있다
	numpy vs list 함수 사용	mylistnp=np.array([1,2,3,np.nan, 5]) mylist=[1,2,3,np.nan,5] mylistnp.max() # 결과는 nan, nan이 있을 경우 결과는 nan mylistnp.sum() # 결과는 nan sum(mylist) # 결과는 nan, mylist.sum() 함수는 없음

	numpy vs list 변환	xx=np.array(mylist) # list -> ndarray
		yy=list(mylistnp) # ndarray -> list

	numpy 2차원 인덱싱	a = np.array([[0, 1, 2, 3], [4, 5, 6, 7]]) # 2차원 행과열의 인덱싱 a[0, :] # 첫번째 행(전체) array([0, 1, 2, 3]) a[:, 1] # 첫 번째 열(전체) array([0, 4]) a[1,1:] # 두 번째행의 두 번째열부터 끝까지 array([5, 6, 7])
sklearn	dir 라이브러리 찾기	print(sklearn.__all__) # 모든 라이브러리 볼수있음 import sklearn dir(sklearn) # 라이브러리가 모두 안나타남 import sklearn.ensemble # 혹은 import sklearn.linear_model dir(sklearn) # 라이브러리 모두 나타남
	help 명령어	help(함수) # help 명령어로 자세한 사용법 알수있음
	sklearn.preprocessing MinMaxScaler() StandardScaler()	- 스케일러 사용은 fit(), transform(), 혹은 fit_transform() 함수사용 - train 데이터는 fit와 transform, test 데이터에는 transform만 사용함 - ndarray=fit_transform(array_like) 형식 array_like : 2차원 ndarray 혹은 df, return 값은 ndarray df의 1개의 행일 경우 : fit_tarnsform(df[['x']]로 2차원 타입으로 변환 df[cols]=fit_transform(df[cols]) # cols에 칼럼이름들을 넣는 방식이 무난 (혹은 1차원 ndarray일 경우 ndarray.reshape(-1,1)로 2차원으로 변환)
	sklearn.metrics	mean_absolute_error, mean_squared_error, median_absolute_error confusion_matrix, f1_score, r2_score, roc_auc_score, recall_score
scipy	라이브러리 찾기	dir(scipy.stats)
	주요 라이브러리	ttest, ttest_rel, chi2_contingency, chisquare, wilcoxon, levene
statsmodels	라이브러리 찾기	import statsmodels.formula.api # 여기까지는 외워야함 dir(statsmodels.formula.api) # formula 함수들 dir(statsmodels.stats.api) # stats 함수들
	주요 라이브러리	ols, logit, glm
	logit 모델, odds 비	model.summary()에서 coef 계수값 c => np.exp(c) 로 구함

[유형3] 통계검정 정리(202005)

변수형(독립변수, 종속변수)과 값의 연속성(범주형, 연속형)과 검증 함수, 예제

[용어]

연속형 변수 : 양적(Quantitative)인 데이터 : 몸무게, 매출액, 주가 등

범주형 변수 : 질적(Qualitative)인 데이터 : 성별, 지역, 만족도 등 -> 문자형으로 저장되어있음

독립변수 : 어떤 실험에서 실험자가 직접 변경하는 변수, (exog 라고도 함)

종속변수 : 독립변수의 값이 변함에 따라 달라지는 수량을 나타내는 변수, (endog 라고도 함)

[예제와 데이터 모양]

<table>
<tr><td colspan="2">독립변수 X
종속변수 y</td><td>종속변수
연속형(1,2,3...)</td><td>범주형(A,B,C)</td></tr>
<tr><td colspan="2">독립변수 없음</td><td>(모집단 1개)
ttest_1sample( ) y
df(y), 고양이 몸무게 모평균은 m이다</td><td></td></tr>
<tr><td rowspan="6">독립변수</td><td rowspan="5">범주형
(A,B,C)</td><td>검정통계량 : t 통계량
(독립변수 1개, 모집단 2개)
ttest_rel( ) y1, y2
df(y1, y2), 수면제 복용전과 후의 수면시간
ttest_ind( ) y1, y2
df(y1, y2), 고양이 수컷과 암컷의 몸무게 차이</td><td rowspan="5">교차분석
chisquare( ) - 적합성
타이타닉 남녀 사망자의 비율은 50% 씩이다. chisquare(array1, array2)
chi2_contingency( ) - 독립성, 동질성

타이타닉 class와 survived 변수는 독립이다. chi2_contingency(table)
X, y</td></tr>
<tr><td>검정통계량 : t 통계량
(독립변수 1개, 모집단 3개 이상)
f_oneway( ) y1, y2, y3
df(y1, y2, y3), 붓꽃(iris) 종별 꽃받침 폭의 차이</td></tr>
<tr><td>검정통계량 : F 통계량
(독립변수 2개이상)
ols( ), anova_lm( ) X1, X2, y
df(X1, X2, y), 자동차 변속기(a,m)과 실린더갯수에 따른 주행거리(mpg)의 차이</td></tr>
<tr><td>검정통계량 : F 통계량</td></tr>
<tr><td rowspan="2">연속형
(1,2,3...)</td><td>회귀분석
ols( )
(단순회귀) X1, y
ols(X1, y) (단순회귀)
면적(X1)에 따른 집값(y)의 예측
(다중회귀) X1, X2, ... , y
자동차 엔진크기, 무게, 연비에 따른 자동차 가격 예측</td><td>로지스틱 회귀분석
logit( ),
LogisticRegression( )
X, y
(이진) 타이타닉 생존자 예측
(3진 이상, softmax)
붓꽃(iris) 3종류 판별</td></tr>
<tr><td>평가지표 : R-squared, SSR</td><td>평가지표 : 분류지표 사용</td></tr>
</table>

[통계 라이브러리]

scipy	라이브러리 찾기	dir(scipy.stats)
	주요 라이브러리	ttest, ttest_rel, chi2_contingency, chisquare, wilcoxon, levene

statsmodels	라이브러리 찾기	import statsmodels.formula.api # 여기까지는 외워야함
		dir(statsmodels.formula.api)

	주요 라이브러리	ols, logit, glm
	logit모델 odds비	model.summary()에서 coef 계수값 c => np.exp(c) 로 구함

[1] t-test

검정 종류	설명	내용
t-test 1 표본	내용	모평균추정, sample개수=1 (X) **연속변수 개수=1개 (X)**
	라이브러리	**scipy.stats.ttest_1sample(array_like, popmean, alternative=' ')**
	검정 가설	X의 모평균은 mu이다.

	검정 알고리즘	if (shapiro(df.X)) == False
		then wilcoxon(df.X-mu, alternative='two-sided') # df.X-mu mu를 뺄셈.
		else ttest_1sample(df.X, mu);

t-test 대응표본	내용	모평균의 차이 비교 – 같은 샘플 2번 측정, sample개수=1 측정값(X1, X2) 연속변수 개수=2개 (X1, X2)
	라이브러리	**scipy.stats.ttest_rel(array_like1, array_like2, alternative=' ')** ㈜ 같은 집단이므로 등분산이라고 가정, shapiro(df.X1-df.X2)로 테스트. **(주의) array_like1(after), array_like2(before)** **before/after 순서에 따라 less/greater/two-sided 인자도 주의필요**
	검정 가설	X1, X2의 모평균은 차이가 없다.

	검정 알고리즘	if (shapiro(df.X1) & shapiro(df.X2)) == False
		then wilcoxon(X1, X2, alternative=' ')
		else ttest_rel(df.X1, df.X2, alternative='greater');

t-test 독립표본	내용	2개의 모평균의 차이 비교, sample개수=2, (X1, X2) 연속변수 개수=2개 (X1, X2)
	라이브러리	**scipy.stats.ttest_ind(array_like, array_like, equal_var=True, alternative=' ')**
	검정 가설	X1, X2의 모평균은 차이가 없다.

	검정 알고리즘	if (shapiro(X1) & shapiro(X2)) == False
		then scipy.stats.manwhitneyu(X1, X2, method='exact', alternative=...)
		# 독립표본에서는 wilcoxon 대신 사용, 샘플이 작을 때 'exact'
		else if (levene(X1, X2)) == False
		then ttest_rel(X1, X2, equal_var=False)
		else ttest_rel(X1, X2, equal_var=True);

[용어 설명]

변수 표기 - df : 데이터프레임, X는 X1하나의 속성 혹은 [X1, X2, X3, ..., y] 속성의 리스트

array_like # 리스트 등을 말함(Series도 포함)

shapiro(df.X) # 정규성검정 함수

levene(df.X1, df.X2) # 등분산검정 함수

wilcoxon( ) # 비모수 검정(non-parametric test)

[t-test 참고]

1. t-test 검정 알고리즘 그림 이해 : https://nittaku.tistory.com/459
2. wilcoxon과 manwhitney 차이 : wilcoxon은 주로 쌍표본, manwhitney는 독립표본 테스트 (아래링크엔터금지)
https://fastercapital.com/ko/content/Wilcoxon-%ED%85%8C%EC%8A%A4%ED%8A%B8-%EB%8C%80--Mann-Whitney-U-%ED%85%8C%EC%8A%A4%ED%8A%B8--%EC%B0%A8%EC%9D%B4%EC%A0%90-%EC%9D%B4%ED%95%B4.html

[2] ANOVA

검정 종류	설명	내용
ANOVA (일원 분산분석)	내용	**3개 이상의 모평균의 차이 비교,** sample개수=3 연속형 변수 3개 (X1, X2, X3) 이상 (1) array_like 3개의 연속형 변수 리스트 혹은 Series (X1, X2, X3) (2) df 에 저장할 경우 : df(X, y), y=3 # melt()를 이용하여 df(X, y)로 바꿈 연속형 변수 1개(X), 범주형 변수 1개(y)
	라이브러리	**scipy.stats.f_oneway(*samples)**
	검정 가설	귀무가설 : X1, X2, X3의 모평균은 같다. 대립가설 : 적어도 한 집단은 평균이 다르다.

	검정 알고리즘	(검정방법1)
		if (shapiro(X1) & shapiro(X2) & shapiro(X3)) == False # 정규성
		then kruskal(X1, X2, X3) # Kruscal-Wallis
		else if (levene(X1, X2, X3) == False) then welch_anova( ) # 등분산
		else f_oneway(X1, X2, X3);
		(검정방법2) - statsmodels.formula.api.ols(formula, data)
		data=df(X, y) # melt()를 이용하여 DataFrame으로 만든다.
		model=ols(formula, data).fit()
		aov_table=anova_lm(model, typ=3)

ANOVA (이원 분산분석)	내용	**종속변수에 대한 2개의 독립변수 영향 판단** 범주형 독립변수 2개(X1, X2), 연속형 종속변수 1개(y) # df(X1, X2, y), sample 개수가 X1 가지 * X2 가지가 되는 셈이다 # 정규성과 등분산 가정 : 일원분산분석과 같은 방법으로 검정
	라이브러리	**statsmodel.formula.api.ols(formula, data)** **statsmodel.stats.anova.anova_lm(*args)** **statsmodel.stats.multicomp.MultiComparison(data, groups)** **statsmodel.stats.multicomp.pairwise_tukeyhsd(endog, groups, alpha=0.05)**
	검정 가설	(귀무가설) X1, X2는 y값에 영향이 없다. X1, X2의 상호작용 효과가 없다.

	검정 알고리즘	(검정방법) statsmodels를 사용, scipy.stats에서는 지원안함
		정규성 검정(shapiro); 등분산검정(levene); # 일원분산분석과 동일
		formula=y~C(X1)+C(X2)+C(X1):C(X2)
		model=ols(formula, data).fit()
		aov_table=anova_lm(model, typ=3)
		# 사후검정1 : Turkey : (X1, X2 각각에 대하여 검증해야함)
		turkey_result = pairwise_tukeyhsd(df.X1, df.y, alpha=0.05)
		print(turkey_result.summary())
		# 사후검정2 : Bonferroni : (X1, X2 각각에 대하여 검증해야함)
		MC=MultiComparison(df.X1, groups=df.y);
		bon_result = MC.allpairtest(scipy.stats.ttest_ind, method='bonf')
		print(bin_result[0])

[ANOVA 참고]

1. 사후분석 : https://partrita.github.io/posts/post-hoc-analysis/
2. 사후검정 두방법의 주요 차이 : tukey는 집단수가 동일할때, Bonferroni는 집단수가 다를때도 사용
   => https://blog.naver.com/sub_om/221299809915
3. bonferroni 방법은 ttest_ind 함수를 사용하며 결과는 ttest를 개별 쌍으로 테스트한 것과 같다.
4. anova_lm(model, typ=2), typ=1,2,3, 기본값은 1이지만 상호작용을 고려하여 보통 3을 사용한다.,
5. 사후분석을 ttest_ind 만으로 개별집단 사후검정을 할 때 집단수가 증가하면 신뢰성이 떨어진다.
   => https://m.blog.naver.com/statsol/221472155248

[3] chisquare

검정 종류	설명	내용
교차분석 카이제곱	내용	**범주형 독립변수 2개(X1, X2), 범주형 종속변수 1개(빈도값)의 관계 분석** **# 비모수검정(변수가 범주형 데이터이거나, 연속형 변수가 정규분포가 아님)**
	라이브러리	**scipy.stats.chisquare(f_obs, f_exp, ddof=0, axis=0)** **scipy.stats.chi2_contingency(table)**
	검정 가설	(적합도검정) X1, X2의 빈도는 f_exp와 같다 (독립성검정) X1, X2 변수는 서로 독립이다.

	검정 알고리즘	**(1) 적합도검정 – 범주형 변수2개의 분포가 기대도수에 맞는지 검정**
		# 데이터가 df(X1, X2, Y)일 경우 가정
		(적합도 검정 알고리즘)
		table=df.X.value_count(); # 리스트나 시리즈로 만들기(array_like)
		f_exp=[100,100] # 기대값 리스트 예제(빈도, frequency)
		chi=chisquare(table, f_exp)
		print(chi) # p_value 출력
		
		(2) 독립성검정 – 범주형 변수2개의 분포가 독립적인지 검정
		(독립성 검정 알고리즘) - 데이터가 df 일 때(로우데이터일 경우)
		table= pd.crosstab(df.X1, df.X2) # DataFrame 빈도 table 만들기
		chi, p, df, expect = chi2_contingency(table)
		# table은 2차원 list, df 모두 가능, 결과(statistics, p_value, df, expect)
		# crosstab()을 사용하지 않으려면 value_counts() 함수로 빈도 계산
		
		(3) 동질성 검정 – 독립성 검정과 같음

[chisquare 참고]

1. 적합도는 범주가 1개일 경우이며, 독립성과 동질성은 범주가 2개일 경우이다.
2. 독립성 검정시 데이터가 df일 경우 crosstab() 함수를 사용하며,
   아닐 경우 2차원 배열에 빈도를 직접 넣어서 chi2_contingency()를 호출할 수 있다

[4] linear regression ols, glm – 종속변수(연속형)

검정 종류	설명	내용
선형회귀	내용	**연속형 독립변수 1개 이상(X1, X2, ...), 연속형 종속변수 1개(y)의 관계 분석**
	라이브러리	**statsmodels.formula.api.ols(formula, data, drop_cols=None, ...)** formual 예) price ~ bedrooms + bathrooms

	알고리즘	(0 단계) 변수간의 상관관계가 있는지 확인
		df.corr() # pearson 상관계수 구하기
		df.X1.corr(df.y)
		stats.pearsonr(df.X, df,y) # 상관계수와 pvalue 구하기
		
		(1단계)
		formula="y ~ X1" # formula="y ~ X1+X2+X3 다중선형회귀일 경우
		model=ols(formula, data).fit()
		model.symmary()
		model.params; model.pvalues
		# dir(model) 명령어로 사용가능한 함수 확인할 수 있음
		
		y_pred=model.predict(v)
		# predict, v는 DataFrame 형태 혹은 dict
		# help(model.predict) 명령어로 함수 사용법을 볼 수 있음
		
		(잔차제곱합 구하기)
		# mean squared error, 잔차제곱합
		((df.키 – model.predict(df.몸무게))**2).sum()
		model.ssr # = 잔차제곱합
		sklearn.metrics.mean_squared_error(df.y, y_pred) # 이것과 동일함.
		(예측값의 신뢰구간 구하기)
		y_pred=model.get_prediction(v) # predict, v는 dict나 DataFrame
		y_pred.summary_frame(alpha=0.05)

	검정 결과 확인	(1) 회귀분석이 통계적으로 유의한가? – F통계량의 p-value 유의수준 (2) 모형은 데이터를 얼마나 설명할수있는가? - R2이 1에 가까운지 확인 (3) 모형내 회귀계수는 유의한가? - 각 독립변수의 회귀계수의 유의수준

[회귀분석 참고]

1. statsmodels.formula.api : statsmodels.api.formula API도 있지만 오류가 난다.
2. statsmodels.api 에 있는 OLS를 사용할 때는 formula 대신 X,Y의 분리 add_constant(X) 사용
   predict 할 데이터에도 add_constant(X)를 사용하여야 한다. 나머지 함수는 사용방법이 같다.
3. 다중선형회귀의 경우 범주형 변수 X=['A', 'B', 'C']와 같은 경우 범주형 변수를 분해하여 처리해준다.
   아니면 pd.get_dummies(df, drop_first=True) 함수를 사용하여 수치변수로 바꾸면 되나 실험결과 결과는 같다.

[5] logitic regression logit, glm – 종속변수(범주형)

검정 종류	설명	내용
선형회귀	내용	**연속형 독립변수 1개 이상(X1, X2, ...), 범주형 종속변수 1개(y)의 관계 분석**
	라이브러리	**statsmodels.formula.api.logit(formula, data, drop_cols=None, ...)** formual 예) Survived ~ Pclass+C(Gender)+SibSp+Parch (범주형은 C)

	알고리즘	(0 단계) 변수간의 상관관계가 있는지 확인
		df.corr() # pearson 상관계수 구하기
		df.X1.corr(df.y)
		stats.pearsonr(df.X, df,y) # 상관계수와 pvalue 구하기
		
		(1단계)
		formula="y ~ X1+X2+X3"
		# formula="y ~ X1 단순선형회귀일 경우
		# formula="y ~ C(X1)+X2+X3 다중선형회귀, X1이 범주형일 때
		# C(X1)에서 X1이 숫자형이더라도 범주를 나타내면 C를 사용함(주의!)
		model=logit(formula, data).fit()
		model.symmary()
		model.params; model.pvalues
		# dir(model) 명령어로 사용가능한 함수 확인할 수 있음
		y_pred=model.predict(v) # predict, v는 DataFrame 형태 혹은 dict
		# help(model.predict) 명령어로 함수 사용법을 볼 수 있음
		
		# 성능 – 분류 – roc_auc_score
		
		(예측값의 신뢰구간 구하기)
		y_pred=model.get_prediction(v) # predict, v는 dict나 DataFrame
		y_pred=model.predict(v)
		y_pred.summary_frame()

	검정 결과 확인	(1) 회귀분석이 통계적으로 유의한가? – F통계량의 p-value 유의수준 (2) 모형은 데이터를 얼마나 설명할수있는가? - R2이 1에 가까운지 확인 (3) 모형내 회귀계수는 유의한가? - 각 독립변수의 회귀계수의 유의수준

[회귀분석 참고]

1. statsmodels.formula.api : statsmodels.api.formula API도 있지만 오류가 난다.

2. formula에서 범주형 변수 v는 C(v)로 만든다.

[통계검정 참고자료] 파이썬 한권으로 끝내기, SD 에듀 ㈜ 시대고시기획, 2022년06월