

빅분기 작업형3 통계검정 주요함수

1. 단일표본 t검정 (단측) 표본평균 모평균 검정
`scipy.stats ttest_1samp alternative = 'greater' , 'less' , 'two-sided'`
2. 독립표본 t 검정 - 두 독립 표본의 평균을 비교 검정
`scipy.stats ttest_ind ttest_ind(group1,group2)`
`equal_var =false` 로 두면 welch t-test 가 됨.
등분산 검정 후 결정. `scipy.stats import levene levene(group1, group2)`
3. 대응표본 t 검정 - 대응되는 두 표본 (실험 전후)
`scipy.stats ttest_rel # 약이 혈압강하 효과가 있다`
`t_stats, p_val= stats.ttest_rel(bp_post,bp_pre,alternative = 'less')`
4. 일원분산분석 (ANOVA) 셋 이상의 그룹의 비교
`scipy.stats f_oneway`
`stats,p_val= f_oneway(groupA,groupB,groupC)`
5. 정규성 검정 shapiro
`scipy.stats shapiro`
`statistic, p_val = shapiro(data)`
6. 회귀모형 (상관계수)
`df.corr()`
7. 로지스틱 회귀 logit
`from statsmodels.formula.api import logit`
`formula = 'target ~ num_col1 + C(cat_col1) + num_col2:num_col3'`
`model = logit(formula,df=data).fit()`
`model.params`
8. 카이제곱 적합도 검정 (chi-square) 관측값과 기대값의 빈도 차이 비교
`from scipy.stats import chisquare`
`chisquare(obs_freq,exp_freq)`
9. 지지도/신뢰도/향상도
지지도(A,B): A와 B가 함께 팔린 거래 횟수 / 전체 거래 횟수
신뢰도(A->B): A와 B가 함께 팔린 거래 횟수 / A가 팔린 거래 횟수
향상도(A,B): 신뢰도(A->B) / 지지도(B)
10. 포아송 분포
`from scipy.stats import poisson`
`poisson.pmf(k, lambda_) pmf :정확히 k번 발생할 확률 cdf: k번 이하로 발생할 확률`
11. 독립성 검정 (chi2_contingency) # 성별과 시험합격은 독립적인가
`from scipy.stats import chi2_contingency`
`statistic, pvalue, dof,expected= stats.chi2_contingency(df)`
12. 이항분포
`from scipy.stats import binom`
`binom.cdf(k=7, n=10, p=0.7) # pmf 는 정확히 k 번 성공할 확률`
13. 점추정/구간추정
`from scipy.stats import t t.interval`
구간추정 = `t.interval(confidence, df=n_samples-1, loc=표본평균, scale=sample_std/(n_samples**0.5))`
confidence 신뢰수준 (예: 0.95 → 95% 신뢰구간)
df 자유도 (보통 n - 1)
loc 평균 (샘플 평균)
scale 표준오차 (standard error = 표준편차 / \sqrt{n})
14. 이원분산분석 Two-way ANOVA
`from statsmodels.formula.api import ols`
`model= ols('Sales ~ C(Decoration_Type) + C(Region) + C(Decoration_Type):C(Region)', data=df).fit()`

`from statsmodels.api.stats import anova_lm`
`anova_table= anova_lm(model,typ=2) # typ=2를 명시. 상호작용을 독립적으로 평가`

15. 잔차이탈도

```
from statsmodels.formula.api import logit
model = logit('purchase ~ age + income + marital_status', data=train).fit()
print(model.summary()) # Log-Likelihood: -206.22
print(round(-2 * model.llf, 2)) # # 잔차이탈도 = -2 로그 우도 (Log-Likelihood)
```

로그 우도 llf 클수록 좋음 설명 잘함

잔차이탈도 -2 * llf 작을수록 좋음 잔차 적음

16. 다중 선형회귀

```
from statsmodels.formula.api import ols
model_full = smf.ols('y ~ x1 + x2 + x3 + x4', data=data).fit()
print(model_full.summary())
```


y_pred = model_full.predict(new_data) # 새로운 데이터 포인트에 대한 예측값


residuals = model_full.resid # 잔차 분석 수행

pred_conf_int_97 = model_full.get_prediction(new_data).summary_frame(alpha=0.03)

17. 비모수 검정(wilcoxon)

```
from scipy.stats import wilcoxon
wilcoxon(df['weight'] - med, alternative='greater')
```

ttest_1samp() popmean=620 ←  기준값 직접 인자 전달

wilcoxon() 기준값 없음  → 직접 차이값을 계산해서 넣어야 함