**<발표대본 - 1시간>**

**6 page (1.1 Covariance)**

X와 y 간 연관 관계의 방향은 공분산으로, 강도는 상관계수를 통해서 볼 수 있다.

(왼쪽 식) 예측변수x와 반응변수y를 x와 y의 평균에서 빼고 곱하면 그래프가 양인지 음인지에 따라 cov 값도 부호가 같다.

선형관계는 양이면 2,4사분면보다 1,3사분면에 많은 점들이 있고, 그러면 (y-y평균)은 (x-x평균)과 항상 부호가 같아서 cov는 양수가 된다.

하지만, 공분산은 양인지 음인지만 알려주고 강도를 알려주지 않는다. 왜냐면 공분산이 측정단위의 변화에 영향을 받기 때문이다. (예를 들어, 천 단위인지 만 단위 인지에 따라 값이 달라짐)

공분산의 단점을 피하기 위해서, 상관계수가 있다.

**7 page (1.1 Correlation Coefficient)**

공분산이 측정단위의 변화를 안받기 위해 표준화를 시킨다. 표준화된 x와 y 사이의 공분산은 상관계수라고 불리고, 계산방식은 ~~이다.

(캡처마지막) 이는 두 변수들의 표준편차에 대한 공분산의 비로 해석될 수도 있다.

공분산과 달리, 상관계수는 척도에 불변해서 측정단위가 달라져도 값이 달라지지 않는다.

상관관계의 두가지 큰 특징은

1. 0이면 선형관계가 없다. -> 관계가 없다는 것이 아님을 유의해야 한다.

2. symmetric 하다.

두 변수의 역할은 중요하지 않고, 오직 선형관계가 있다, 없다, 강하다, 약하다만 측정한다.

**8 page (1.1 단순선형회귀 모델)**

모수 : 베타0(상수항, 절편, x=0일 때 y의 기대값), 베타1(기울기, 베타1한 단위 변화에 대한 y의 변화)

확률변동 또는 오차

X가 y에 미치는 영향에 대한 해석과 새로운 x가 관측된 경우의 y값과 평균변화 예측, 설명변수 여러 개가 가능하다는 장점이 있다.

당연한 말이지만, 회귀분석은 상관 분석과 한 가지 중요한 점이 다르다. 상관계수는 corr(y,x)=corr(x,y) 같고 대칭적이라 x와 y 변수가 동등하게 중요하지만, 회귀분석에서는 y가 더 중요하다. 변수가 동등하게 중요하지 않다는 점이 다른 점이다.

**9,10 page (1.2 LSE)**

그래서 우리는 이제 단순회귀식을 알기 위해 모수들을 추정한다. 가장 각각의 점들을 잘 설명해주는 best fit인 회귀식을 만들어야하는 게 기본 아이디어이다.

선형 회귀 모델에서 회귀 계수를 추정하는 가장 유명한 방법인 최소 제곱법을 사용한다.

각 점으로부터 구하고자 하는 최적 직선까지의 수직거리의 제곱합을 최소로 하는 직선의 방정식을 구한다.

(9page식)그래서 이게 각 점인 y에서 최적 직선 y hat을 빼고 제곱해주는 것을 최소로 하는 식이다.

(10page) 최소로 하는 회귀 계수를 본다. 식은 ~~이고, 구하는 거는 생략하겠다.

잔차들의 합과 잔차와 x 곱한 값들의 합이 0이라는 식도 유도할 수 있다. 과정 생략

그래서 추정된 베타0, 베타1의 평균과 표준오차를 구하여 분포를 알면, 검정통계량을 구할 수 있다. 추정된 베타1의 표준오차는 ~~.

**11,12 page (1.2 가설검정)**

이제 가설검정을 하자. 베타1 = 0을 가설로 해서, y와 x 사이에 선형적인 관계가 있는지 여부를 검정한다.

H0 만들고, T값 계산해서 p-value 계산하고 신뢰구간을 확인하면 된다.

(12 page) 오차항들은 평균은 0, 분산은 시그마제곱, 서로 독립적인 확률변수로 가정한다.

각각 추정된 베타0, 베타1은 베타0과 베타1의 불편추정치이다. 분산은 ~~(맨 위 식)

추정량 분포가 우리가 모르는 미지의 모수, 시그마 제곱에 둘다 의존하는 걸 알 수 있다. 따라서 우리는 데이터로부터 시그마 제곱을 추정할 필요가 있다. 시그마제곱의 불편추정치를 구한다.

분산의 불편추정치는 ~~(두번째 식)

여기서 SSE는 잔차들의 제곱합이고 n-2는 SSE의 자유도이다.

그래서 h0가 사실이라는 가정 하에서 자유도가 (n-2)인 t-분포를 따른다.

식 확인~~ 신뢰구간 확인~~

**13,14,15 page (1.3 예측 1번&2번)**

1. 예측변수의 어떤 선택된 값 x0에 대응되는 반응변수 y의 값에 대한 예측

예측값 y0 hat은 ~~ 평균은 ~~

분산은 ~~(생략) 표준오차는 ~~

그래서 예측구간 ~

신뢰계수가 (1-알파)인 예측값에 대한 신뢰구간~~

2. X=x0으로 주어졌을 때 평균 뮤0에 대한 추정

평균반응 뮤0는 식~~ 표준오차는 ~~

신뢰계수가 (1-알파)인 뮤0에 대한 신뢰구간~~

앞의 prediction interval보다 항상 분산과 표준편차가 작다는 것을 알 수 있다. (1이 있고 없음!)

**16,17 page (1.3 Decomposition of sum of squares)**

이렇게 앞에서 추정한 회귀식을 그리면 각 점들마다 설명되는 부분과 설명이 안되는 부분이 생긴다.

(그래프) 점이 회귀식보다 위에 있으면 회귀식이 설명하지 못하는 부분이고, 점이 회귀식보다 아래에 있으면 회귀식이 설명하는 부분이다.

이를 나눠서 식을 만들면 SST=SSR+SSE 식이 나온다. SSR의 비중이 높으면 자료를 잘 설명하고 있는 것이다.

(17 page) 전체 SST 중에 SSR의 비율을 결정계수라고 한다. Correlation coefficient는 ~~.

비율이니까 당연히 0부터 1사이 값이다. 해석~~

결정계수가 1에 가까우면 x가 y의 변이 중 많은 부분을 설명한다는 것을 의미한다.

**19 page (2.1 다중선형회귀 모델)**

여러 개의 설명 변수를 가지는 다중선형회귀모형을 보겠다.

식은 ~~. 단순선형회귀의 확장, 일반화라고 할 수 있다.

**20,21 page (2.2 LSE)**

왼쪽 글~~

오차는 다음과 같다.~~

최소로 만드는 식~~

최소제곱추정량의 4가지 성질

1. 먼저 추정량 베타 hat은 베타의 불편추정량이다. 분산은 시그마제곱\*cjj으로 cjj특징으로 모든 가능한 불편추정량들 중에서 최소분산을 가져서 이 최소 제곱추정량들은 best linear unbiased estimator로 BLUE라고 불린다.

2. ~~

3. ~~

4. ~~

**22 page (2.2 해석)**

~~

**23 page (2.3 다중상관계수)**

~~

**3. Basic Code – 20분**

Import pandas as pd

Import numpy as np

dataset=pd.read\_csv(“C:/Users/김영은/Desktop/iris.csv”)

dataset

**#scatter plot**

Import matplotlib.pyplot as plt

plt.scatter(dataset.SepalLenth, dataset.SeplaWidth)

plt.title(“Scatter plot”)

plt.xlable(“ScatterLength”)

plt.ylable(“ScatterWidth”)

**#아웃라이어를 제거하여 산점도 다시 그려보기**

**#아웃라이어 : (평균) + - 3 \* (표준편차)**

def dataset\_outlier(x):

left\_out = x.mean() -3 \* x.std()

right\_out = x.mean() + 3 \* x.std()

x=x[(x>left\_out) & (x<right\_out)]

return x

import pickle

import seaborn as sns

dataset[“SepalLength”] = dataset\_outlier(dataset[“SepalLength”])

dataset[“SepalWidth”] = dataset\_outlier(dataset[“SepalWidth”])

sns.replot(x=”SepalLength”,y=”SepalWidth”,data=dataset)

**#회귀선 그리기**

sns.lineplot(x=”SepalLength”,y=”SepalWidth”,data=dataset)

**#correlation coefficient**

r = np.corrcoef(dataset.SepalLength,dataset.SepalWidth)

print( r ) #결과 : 약한 음의 상관관계 확인 가능

**#모든 변수간 산점도 확인하기 : pairplot()**

cols=['SepalLength','SepalWidth','PetalLength','PetalWidth']

dataset1=dataset[cols]

sns.pairplot(dataset1)

**#회귀분석 하기 1번 방법**

import statsmodels.formula.api as sm

model = sm.ols(formula='SepalLength~SepalWidth+PetalLength+PetalWidth',

data=dataset).fit()

print(model.summary())

#결과 : 적합된 회귀식 : SepalLength = 1.8451 + 0.6549 SepalWidth + 0.7111 PetalLength - 0.5626 PetalWidth

**#회귀분석 하기 2번 방법**

**#임의의 값에 대하여 SepalLength의 예측값 구하기**

from sklearn.linear\_model import LinearRegression

model=LinearRegression()

x=dataset[['SepalWidth','PetalLength','PetalWidth']]

y=dataset[['SepalLength']]

model.fit(x,y)

SepalWidth=3.9

PetalLength=1.7

PetalWidth=0.4

X=[SepalWidth,PetalLength,PetalWidth]

prediction=model.predict([X])

print(prediction)