5.1 연속형 독립변수가 하나인 모델 (단순회귀)

5.1.1 분석 준비

```
# 수치 계산에 사용하는 라이브러리

import numpy as np
import pandas as pd
import scipy as sp
from scipy import stats

# 그래프를 그리기 위한 라이브러리
from matplotlib import pyplot as plt
import seaborn as sns
sns.set()

# 선형모델을 추정하는 라이브러리 (경고가 나올 수도)
import statsmodels. formula.api as smf
import statsmodels.api as sm
# 표시 자릿수 지정
%precision 3

#그래프를 주피터 노트북에 그리기 위한 설정
%matplotlib inline
```

경고가 나와도 그대로 진행 가능

5.1.2 데이터 읽어 들이기와 표시

```
beer = pd.read_csv("5-1-1-beer.csv")
print(beer.head())
```

가공의 맥주 매상 데이터 읽어 들이기.

그래프 그리기 - 데이터의 특징을 알 수 있다.

```
sns.jointplot(x = "temperature", y = "beer", data = beer, color = 'black')
```

(그래프 나옴)

5.1.3 모델 구축

$$맥주\ 매상 \sim \mathcal{N}(\beta_0 + \beta_1 \times 기온, \sigma^2)$$

- (종속변수 - 맥주 매상, 독립변수 - 기온)을 사용한 정규선형모델
- 독립변수가 한 개밖에 없으므로, 기온이 모델에 들어가는지 판단만 하면 됨.

- 파라미터 추정으로는 식에 있는 계수 2개를 추정. 시그마 제곱은 장애모수이므로 고려 x

모델을 구축함으로써 얻는 장점

1. 현상을 해석할 수 있게 된다.
    - 계수 beta1이 0이 아니다 - 맥주 매상은 기온의 영향을 받는다
    - 계수 의 부호를 안다 - 기온이 오르면 맥주 매상이 올라갈지 떨어질지 판단 가능
    - 계수 검정 대신 AIC를 이용한 모델 선택을 이용해도 됨. - 맥주 매상을 예측하려면 기온이 필요하다는 해석 가능
2. 예측이 가능하다
    - 계수와 기온을 알면 맥주 매상의 기댓값을 계산할 수 있게 됨

5.1.4 statsmodels를 이용한 모델링

[정규선형모델 구축]

- 통계모델 추정을 위해 import statsmodels. formula. api as smf를 이용해서 statsmodels를 임포트한다.

```
Im_model = smf.ols(formula = "beer ~ temperature", data = beer).fit()
```

- smf.ols 함수 사용 (ols는 범용최소제곱법의 약자)
- 모집단분포가 정규분포임을 가정했을 때, 최대우도법의 결과는 최소제곱법의 결과와 일치
- formula - 모델의 구조 지정
    - "beer ~ temperature"로 지정함으로써 종속변수는 beer, 독립변수는 temperature인 모델 지정
    - formula를 바꿈으로써 다양한 모델 추정 가능
- formula와 대상이 되는 데이터프레임을 지정하는 것으로 모델에 대한 설정 종료
- 마지막으로 fit()을 호출 - 이것으로 파라미터 추정까지 자동으로 끝남

5.1.5 추정 결과 표시와 계수 설정

summary 함수를 이용해 추정 결과 표시

```
Im_model.summary()
```

coef 은 계숫값

이 다음부터 순서대로 계수의 표준오차, t값, 귀무가설을 '계수의 값이 0'이라고 했을 때의 p값, 95% 신뢰구간에서 하측신뢰한계와 상측신뢰한계

- p값은 매우 작아서 반올림하여 0이 됨.
- 기온에 대한 계수는 0과 다르다고 판단 가능
- 기온이 맥주 매상에 영향을 끼친다는 것을 알 수 있음
- 기온이 오르면 좋은지 내려가면 좋은지는 계숫값을 보면 알 수 있음 - 0.7654로 양수 - 기온이 오르면 맥주 매상도 오른다는 의미

이 정도의 해석은 산포도를 본 시점에는 어느 정도 알 수 있다. 하지만 독립변수가 많아지는 등 복잡한 모델이 되는 경우네는 모델을 구축하고 그 계수를 보는 것이 해석하기 쉽다.

5.1.6 summary 함수의 출력 내용 설명

세 번째는 뒤에 나온다

첫 번째 표에 대하여

- **Dep.Variable** : 종속변수의 이름. Dep은 Depended의 약자로, 종속변수라는 의미입니다.
- **Model, Method** : 범용최소제곱법을 사용했다는 설명
- **Date, Time** : 모델을 추정한 일시
- **No. Observations** : 샘플사이즈
- **Df Residuals** : 샘플사이즈에서 추정된 파라미터 수를 뺀 것
- **Df Model** : 사용된 독립변수의 수
- **Covariance Type** : 공분산 타입. 특별히 지정하지 않으면 nonrobust가 됩니다.
- **R-squared, Adj. R-squared** : 결정계수와 자유도 조정이 끝난 결정계수. 결정계수는 5.1.12절에서 설명하겠습니다.
- **F-statistic, Prob (F-statistic)** : 분산분석 결과. 분산분석은 5.2절에서 설명하겠습니다.
- **Log-Likelihood** : 최대로그우도
- **AIC** : 아카이케 정보 기준
- **BIC** : 베이즈 정보 기준. 정보 기준의 일종이지만 이 책에서는 사용하지 않습니다. 세세한 부분은 사용하는 라이브러리나 버전에 따라 달라질 수 있습니다. 샘플사이즈, 결정계수, AIC 등을 참조하는 것만으로도 충분합니다.

5.1.7 AIC를 이용한 모델 선택

AIC를 이용한 모델 선택

독립변수가 1개 밖에 없기 때문에 **Null모델의 AIC**와 **기온이라는 독립변수가 들어간 모델의 AIC**를 비교하는 작업

1. Null모델 구축 : 독립변수가 없을 때는 "beer ~ 1"이라고 함수에 파라미터를 넘긴다.

```
null_model = smf.ols("beer ~ 1", data = beer).fit()
```

```
null_model.aic
```

2. 독립변수가 있는 모델

```
Im_model.aic
```

독립변수가 있는 모델 쪽이 더 작은 AIC를 가지고 있음.

때문에 기온이라는 독립변수가 있는 쪽이 예측 정확도가 높아지는 것이 아닐까 하는 판단 가능 - 맥주 매상 예측 모델에는 기온이라는 독립변수가 필요함!

AIC 계산 방법

```
AIC = -2*(최대로그우도 - 추정된 파라미터 수)
```

추정된 모델의 로그우도

```
lm_model.llf
```

추정된 파라미터 수를 바로 알면 좋겠지만 이에 대한 정보는 모델에 포함되어 있지 않으므로 모른데요

사용된 독립변수의 수

```
lm_model.df_model
```

실제로 절편도 추정되었기 때문에 여기에 1을 더하면 추정된 파라미터 수를 구할 수 있다.

최종 AIC

```
-2*(lm_model.llf - (lm_model.df_model + 1))
```

(주의)

그런데 추정된 파라미터 수에는 몇 가지 유형이 있습니다.

이번에는 장애모수를 파라미터 수에 포함시키지 않았지만 이를 포함한 AIC를 구하는 경우도 존재 - 210.909

R언어 등 다른 소프르웨어에서는 장애모수가 포함되어 있기도 함

AIC는 그 값의 크고 작음에 의미가 있는 지표 - (절댓값은 의미가 없음)

같은 유형으로 계산됐다면 괜찮지만 다른 소프트웨어나 라이브러리에서 계산된 AIC와의 비교는 피해야 한다.

5.1.8 회귀직선

회귀직선

모델에 의한 종속변수의 추측값을 직선으로 표시한 것

(비선형모델의 경우는 회귀곡선)

5.1.9 seaborn을 이용한 회귀직선 그래프 그리기

```
sns.lmplot(x = "temperature", y = "beer", data = beer, scatter_kws =
{"color":"blue"}, line_kws = {"color":"black"})
```

이것은 산포도에 회귀직선을 덧그린 그래프

- 산포도의 디자인은 scatter_kws, 회귀직선의 디자인은 line_kws 으로 지정
- 음영부분 : 회귀직선의 95% 신뢰구간

5.1.10 모델을 이용한 예측

모델의 계수를 추정할 수 있으므로 이를 사용하면 예측 할 수 있다.

이를 위해 추정된 모델에 **predict**함수 적용

파라미터에 아무것도 넘기지 않으면 훈련 데이터를 사용한 값이 그대로 출력됨.

```
lm_model.predict()
```

기온값을 지정해서 예측할 수도 있다.

파라미터로 데이터프레임을 넘긴다. 이번에는 기온이 0도일 때의 맥주 매상의 기댓값을 계산

```
lm_model.predict(pd.DataFrame({"temperature":[0]}))
```

그러나 이번에 추정한 모델은 아래와 같았다

$$\text{맥주 매상} \sim \mathcal{N}(\beta_0 + \beta_1 \times \text{기온}, \sigma^2)$$

모델의 예측값, 즉 정규분포에서 기댓값은 ~~으로 계산된다.

그러므로 기온이 0도일 때는 beta0과 같아진다.

확인해보자면

```
lm_model.params
```

*Intercept*가 beta0으로 예측값과 일치한다.

다음은 기온이 20도일 때의 맥주 매상의 기댓값이다.

```
lm_model.predict(pd.DataFrame({"temperature":[20]}))
```

이 값은 beta0 + beta1*20와 같다.

```
beta0 = lm_model.params[0]
beta1 = lm_model.params[1]
temperature = 20

beta0 + beta1*temperature
```

5.1.11 잔차 계산

마지막으로 모델의 평가 방법에 대해....

원래는 예측을 하기 전에 모델의 평가를 해두면 좋다.

모델의 평가는 주로 **잔차를 체크**해서 한다.

정규선형모델의 경우, 잔차가 '평균이 0인 정규분포'를 따르는 것이므로 모델이 그 분포를 따르고 있는지 체크함.

잔차는 다음과 같이 계산해서 얻는다

```
resid = lm_model.resid
resid.head(3)
```

공부가 목적이므로 잔차를 따로.....계산해보자

잔차 계산식은

$$residuals = y - \hat{y}$$

여기서

$$\hat{y} = \beta_0 + \beta_1 \times 기온$$

이 값은

```
y_hat = beta0 + beta1*beer.temperature
y_hat.head(3)
```

실젯값에서 예측값을 빼면 잔차가 된다.

```
(beer.beer - y_hat).head(3)
```

5.1.12 결정계수

R - squared - 가지고 있는 데이터에 대해, 모델을 적용했을 때의 적합도를 평가한 지표

아래와 같이 계산할 수 있다.

y는 종속변수, y^은 모델에 의한 추측치(예측치), .는 y의 평균값

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}-\mu)^2}{\sum_{i=1}^{N}(y-\mu)^2}$$

- (모델에 의한 추측치 = 종속변수의 실젯값)이면 R^2은 1이 됨

결정계수 파이썬으로 계산/ 아래와 같이 코드를 작성해도 무방

식을 변형해서 결정계수의 분모는 아래와 같이 분해할 수 있다.

$$\sum_{i=1}^{N}(y-\mu)^2 = \sum_{i=1}^{N}(\hat{y}-\mu)^2 + \sum_{i=1}^{N} residuals^2$$

종속변숫값의 변동크기를

(모델로 설명 가능한 변동 + 그렇지 않은 잔차제곱합)으로 분해 가능

- 이 때문에 결정계수는 *전체 변동폭의 크기에 대한 모델로 설명 가능한 변동폭의 비율*로 해석 가능

5.1.13 수정된 결정계수

독립변수의 수가 늘어나는 것에 대해 페널티를 적용한 결정계수

- 독립변수의 수가 늘어나면 결정계수는 큰 값이 된다.
- 결정계수가 높아지면 과학습을 일으키기 때문에 조정이 필요하다.

아래의 식. s는 독립변수의 수

$$R^2 = 1 - \frac{\sum_{i=1}^{N} residuals^2 \Big/ (N-s-1)}{\sum_{i=1}^{N}(y-\mu)^2 \Big/ (N-1)}$$

5.1.14 잔차 그래프

잔차의 특징을 보는 가장 간단한 방법 - *잔차의 히스토그램을 그리는 것*

이것을 보고 정규분포의 특징을 갖고 있는지 확인해보자

```
sns.distplot(resid, color = 'blue')
```

이를 보면 좌우대칭으로 **정규분포**를 따르는 것처럼 보인다.

(x축 - 적합도, y축 - 잔차)인 산포도를 그려보자

이 산포도가 완전 랜덤이며 상관이 없다는 것을 확인한다. 매우 큰 잔차가 나오지 않는 것도 확인한다.

```
sns.jointplot(lm_model.fittedvalues, resid, joint_kws=
{"color":"black"},marginal_kws={"color":"blue"})
```

자세하게 검정 순서를 기억하고 있지 않아도 이런 그래프를 보는 것만으로도 명확한 문제점을 깨닫게 됩니다.

5.1.15 Q-Q 플롯

이론상의 분위점과 실제 데이터의 분위점을 산포도 그래프로 그린 것. (Q = Quantile)

- 이번에는 모든 데이터에 대한 분위점을 구한다. - 데이터가 100개 있다면 1%씩 100개의 분위점 존재
- 한편, 정규분포의 퍼센트 포인트를 사용하면 이론상의 분위점을 얻을 수 있다.
- 이론상의 분위점과 실제 데이터의 분위점을 구해서 그 둘을 비교하는 것으로 잔차가 정규분포에 근접하는지 아닌지 '시각적'으로 판단 가능.

이것은 sm.qqlot 함수를 사용해서 그릴 수 있다.

line = "s"라고 파라미터를 넘김으로써 잔차가 정규분포를 따르면 이 선상에 위치한다는 기준을 표시하게 된다.

```
fig = sm.qqplot(resid, line = "s")
```

직접 만들어 보겠습니다!!

5.1.16 summary 함수의 출력으로 보는 잔차 체크

Omnibus:	0.587	Durbin-Watson:	1.960
Prob(Omnibus):	0.746	Jarque-Bera (JB):	0.290
Skew:	−0.240	Prob(JB):	0.865
Kurtosis:	2.951	Cond. No.	52.5

Prob는 잔차의 정규성에 대한 검정 결과

- 귀무가설 : 잔차가 정규분포를 따른다.
- 대립가설 : 잔차가 정규분포와 다르다.

p값이 0.05다 큰지 확인하자.

그러나 검정의 비대칭성이 있으므로 크다고 해도 정규분포라고 주장할 수 없다.

정규분포와 다른지 여부를 판단

- 왜도 : 히스토그램의 좌우비대칭 방향과 그 정도를 측정하는 지표
  - 0보다 크면 오른쪽 자락이 길어진다.
  - 정규분포는 좌우 대칭이기 때문에 왜도는 0

$$Skew = E\left(\frac{(x-\mu)^3}{\sigma^3}\right)$$

- 첨도 : 히스토그램 중심부의 뾰족함을 측정하는 지표
  - 값이 클수록 히스토그램의 *가운데 부분*이 뾰족해짐
  - $Kurtosis = E\left(\frac{(x-\mu)^4}{\sigma^4}\right)$

Durbin - Watson : 잔차의 자기상관을 체크하는 지표

- 이것이 2 전후라면 문제 없다고 판단
- 시계열 데이터를 대상으로 분석하는 경우 반드시 확인 필요
- 잔차에 자기상관이 있으면 게수의 t검정 결과를 신뢰할 수 없음
  - 이 문제를 **보여주기 위한 회귀**라고 부른다.