Py 이론적 확률분포 3

I. χ^2-분포

II. t-분포

III. F-분포

Ⅰ. χ^2-분포

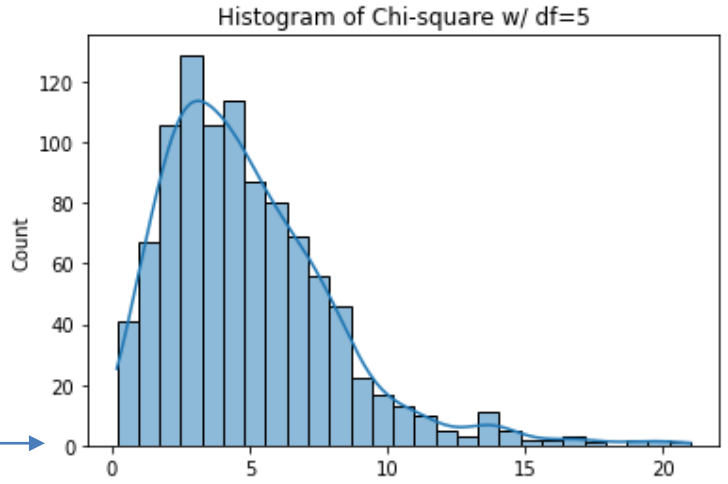
1. 확률분포

- 표준정규분포의 제곱의 합 X=$\sum_{i=1}^{n} Z_i^2$이 자유도가 n인 χ^2-분포에 따름
- X~χ^2(n)
- X가 X~χ^2(n)이라고 할 때, X의 평균은 n, 분산은 2n

b1-ch4-10.py

```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
random.seed(12345)
r=1000
#np.random.seed(12)
z_1 = random.normal(loc=0.0, scale=1.0, size=r)
#np.random.seed(34)
z_2 = random.normal(loc=0.0, scale=1.0, size=r)
#np.random.seed(56)
z_3 = random.normal(loc=0.0, scale=1.0, size=r)
#np.random.seed(78)
z_4 = random.normal(loc=0.0, scale=1.0, size=r)
#np.random.seed(910)
z_5 = random.normal(loc=0.0, scale=1.0, size=r)
chi_5 = z_1**2+z_2**2+z_3**2+z_4**2+z_5**2
mean = np.mean(chi_5)
variance = np.var(chi_5, ddof=1)
print("Mean of chi-distrinution w/ df=5 is :",round(mean,5))
print("Variance of chi-distrinution w/ df=5 is :",round(variance,5))
sns.histplot(data=chi_5, x=None, kde=True).set(title='Histogram of Chi-square w/ df=5')
plt.show()
```

```
Mean of chi-distrinution w/ df=5 is : 4.94847
Variance of chi-distrinution w/ df=5 is : 9.94462
```



- 자유도에 따라 χ^2-분포의 모양이 달라지는데 자유도가 클수록 정규분포와 근사

b1-ch4-11.py

```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
np.random.seed(12345)
r=10000
chi1 = random.chisquare(df=5, size=r)
sns.histplot(data=chi1, x=None).set(title='Histogram of Chi-square w/ df=5')
plt.show()
chi2 = random.chisquare(df=10, size=r)
sns.histplot(data=chi2, x=None).set(title='Histogram of Chi-square w/ df=10')
plt.show()
chi3 = random.chisquare(df=20, size=r)
sns.histplot(data=chi3, x=None).set(title='Histogram of Chi-square w/ df=20')
plt.show()
chi4 = random.chisquare(df=30, size=r)
sns.histplot(data=chi4, x=None).set(title='Histogram of Chi-square w/ df=30')
plt.show()
np.mean(chi1)
np.mean(chi2)
np.mean(chi3)
np.mean(chi4)
print("Mean of Chi-square Distribution w/ df=5 is : ", np.mean(chi1))
print("Mean of Chi-square Distribution w/ df=10 is : ", np.mean(chi2))
print("Mean of Chi-square Distribution w/ df=20 is : ", np.mean(chi3))
print("Mean of Chi-square Distribution w/ df=30 is : ", np.mean(chi4))
```

```
Mean of Chi-square Distribution w/ df=5 is :  4.966797648108684
Mean of Chi-square Distribution w/ df=10 is :  10.032103035129143
Mean of Chi-square Distribution w/ df=20 is :  19.92401803752546
Mean of Chi-square Distribution w/ df=30 is :  30.01999670525099
```

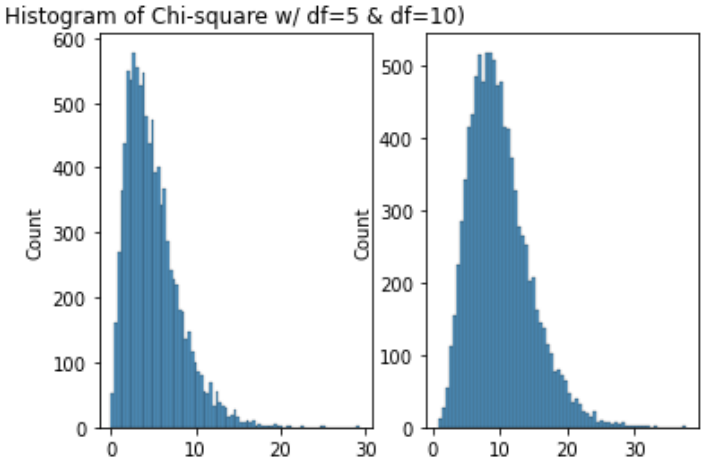
- 자유도에 따라 χ^2-분포의 모양이 달라지는데 자유도가 클수록 정규분포와 근사

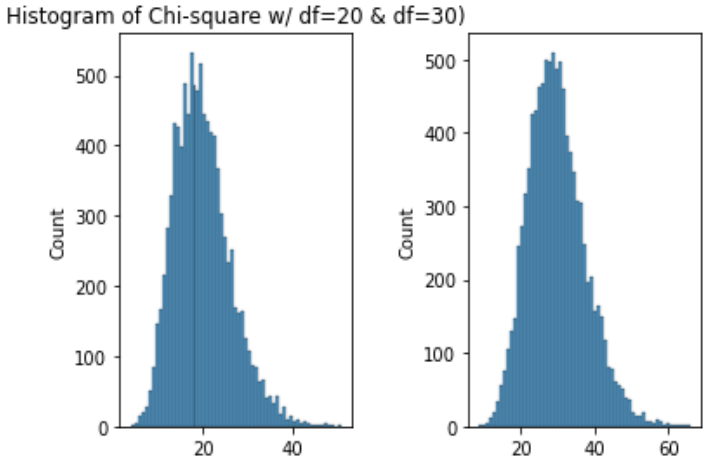
b1-ch4-11.py

(앞에서 계속)

```
np.var(chi1, ddof=1)
np.var(chi2, ddof=1)
np.var(chi3, ddof=1)
np.var(chi4, ddof=1)
print("Variance of Chi-square Distribution w/ df=5 is : ", np.var(chi1, ddof=1))
print("Variance of Chi-square Distribution w/ df=10 is : ", np.var(chi2, ddof=1))
print("Variance of Chi-square Distribution w/ df=20 is : ", np.var(chi3, ddof=1))
print("Variance of Chi-square Distribution w/ df=30 is : ", np.var(chi4, ddof=1))

fig = plt.figure(figsize=(14,7))
# 두 개의 그래프를 한 페이지에 그림
fig, axs = plt.subplots(ncols=2)
fig, ays = plt.subplots(ncols=2)

sns.histplot(data=chi1, x=None, ax=axs[0]).set(title='Histogram of Chi-square w/ df=5 & df=10)')
sns.histplot(data=chi2, x=None, ax=axs[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
#plt.show()
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/chi-1.png')

sns.histplot(data=chi3, x=None, ax=ays[0]).set(title='Histogram of Chi-square w/ df=20 & df=30)')
sns.histplot(data=chi4, x=None, ax=ays[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
plt.show()
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/chi-2.png')
```

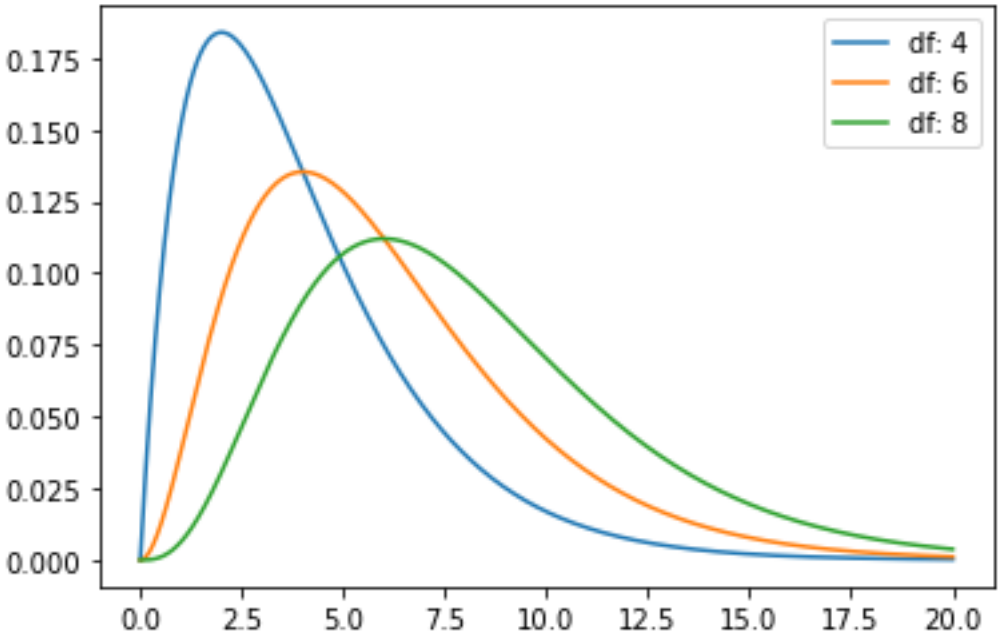
```
Variance of Chi-square Distribution w/ df=5 is :  9.78390973692184
Variance of Chi-square Distribution w/ df=10 is :  20.411965163444886
Variance of Chi-square Distribution w/ df=20 is :  40.075730426388795
Variance of Chi-square Distribution w/ df=30 is :  59.76935320106778
```





b1-ch4-12.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2

#x-axis ranges from 0 to 20 with .001 steps
x = np.arange(0, 20, 0.001)

#define multiple Chi-square distributions

plt.plot(x, chi2.pdf(x, df=4), label='df: 4')
plt.plot(x, chi2.pdf(x, df=6), label='df: 6')
plt.plot(x, chi2.pdf(x, df=8), label='df: 8')

#add legend to plot
plt.legend()
```



2. 확률분포표

b1-ch4-13.py

```
import pandas as pd
import numpy as np
from scipy.stats.distributions import chi2 as chi2dist
#df = 31
df = 11
chi_1 = np.empty(df)
chi_2 = np.empty(df)
chi_3 = np.empty(df)
chi_4 = np.empty(df)
chi_5 = np.empty(df)
chi_6 = np.empty(df)
# repeat r times:
for j in  range(df):
 chi_1[j] = chi2dist.ppf(0.01, df=j)
 j=j+1
for j in  range(df):
 chi_2[j] = chi2dist.ppf(0.05, df=j)
 j=j+1
for j in  range(df):
 chi_3[j] = chi2dist.ppf(0.1, df=j)
 j=j+1
for j in  range(df):
 chi_4[j] = chi2dist.ppf(0.9, df=j)
 j=j+1
for j in  range(df):
 chi_5[j] = chi2dist.ppf(0.95, df=j)
 j=j+1
for j in  range(df):
 chi_6[j] = chi2dist.ppf(0.99, df=j)
 j=j+1
chi_square = pd.DataFrame({'p=0.99':chi_1,'p=0.95':chi_2,'p=0.9':chi_3,'p=0.1':chi_4,'p=0.05':chi_5,'p=0.01':chi_6})
chi_sq = chi_square.dropna(how='all') # to drop if all values in the row are NaN
round(chi_sq, 3)
print("Chi-square Distribution Table with 'p=0.99, ..., p=0.01' : ", f'\n{round(chi_sq, 3)}\n')
```

자유도	P=0.99	0.95	0.90	0.10	0.05	0.01
1	0.000157	0.00393	0.0158	2.706	3.841	6.635
2	0.0201	0.103	0.211	4.605	5.991	9.210
3	0.115	0.352	0.584	6.251	7.815	11.341
4	0.297	0.711	1.064	7.779	9.488	13.277
5	0.554	1.145	1.610	9.236	11.070	15.086
6	0.872	1.635	2.204	10.645	12.592	16.812
7	1.239	2.167	2.833	12.017	14.067	18.475
8	1.646	2.733	3.490	13.362	15.507	20.090
9	2.088	3.325	4.168	14.684	16.919	21.666
10	2.558	3.940	4.865	15.987	18.307	23.209

```
Chi-square Distribution Table with 'p=0.99, ..., p=0.01' :
    p=0.99  p=0.95  p=0.9   p=0.1  p=0.05  p=0.01
1    0.000   0.004  0.016   2.706   3.841   6.635
2    0.020   0.103  0.211   4.605   5.991   9.210
3    0.115   0.352  0.584   6.251   7.815  11.345
4    0.297   0.711  1.064   7.779   9.488  13.277
5    0.554   1.145  1.610   9.236  11.070  15.086
6    0.872   1.635  2.204  10.645  12.592  16.812
7    1.239   2.167  2.833  12.017  14.067  18.475
8    1.646   2.733  3.490  13.362  15.507  20.090
9    2.088   3.325  4.168  14.684  16.919  21.666
10   2.558   3.940  4.865  15.987  18.307  23.209
```

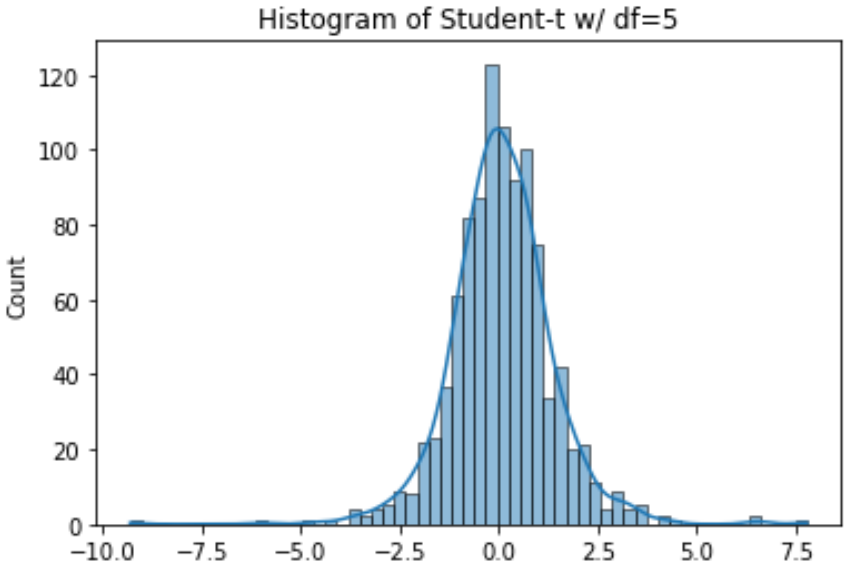
1. 확률분포

- 서로 독립적인 표준정규분포와 χ^2-분포에 의해 t-분포가 도출

  X~t(n-1)

b1-ch4-14.py

```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
r=1000
np.random.seed(1)
z = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(12)
z_1 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(34)
z_2 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(56)
z_3 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(78)
z_4 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(910)
z_5 = random.normal(loc=0.0, scale=1.0, size=r)
chi_5 = z_1**2+z_2**2+z_3**2+z_4**2+z_5**2
sqchi_5 = np.sqrt(chi_5/5)
t_5 = z / sqchi_5
sns.histplot(data=t_5, x=None, kde=True).set(title='Histogram of Student-t w/ df=5')
plt.show()
```



- 자유도에 따라 t-분포의 모양이 달라지는데 자유도가 30 이상이면 표준정규분포와 거의 같아짐

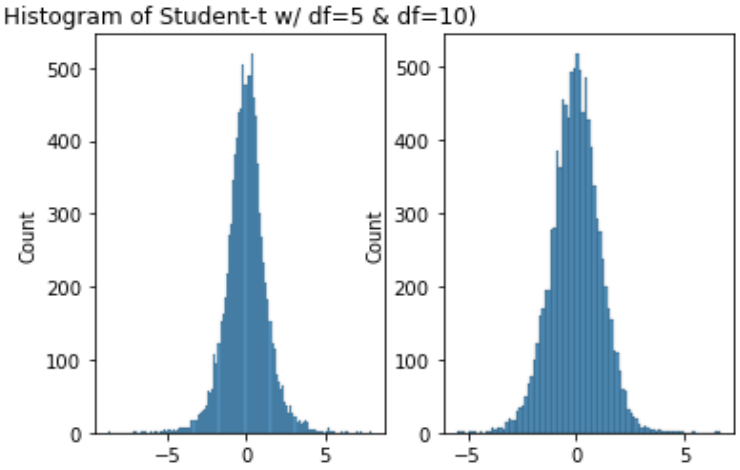
b1-ch4-15.py

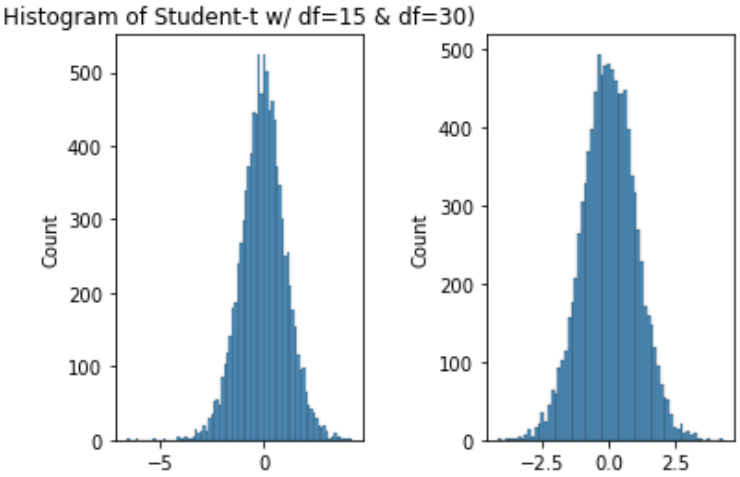
```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
np.random.seed(12345)
r=10000
t1 = random.standard_t(df=5, size=r)
sns.histplot(data=t1, x=None).set(title='Histogram of Student-t w/ df=5')
plt.show()
t2 = random.standard_t(df=10, size=r)
sns.histplot(data=t2, x=None).set(title='Histogram of Student-t w/ df=10')
plt.show()
t3 = random.standard_t(df=15, size=r)
sns.histplot(data=t3, x=None).set(title='Histogram of Student-t w/ df=15')
plt.show()
t4 = random.standard_t(df=30, size=r)
sns.histplot(data=t4, x=None).set(title='Histogram of Student-t w/ df=30')
plt.show()
fig = plt.figure(figsize=(14,7))
```

b1-ch4-15.py

(앞에서 계속)

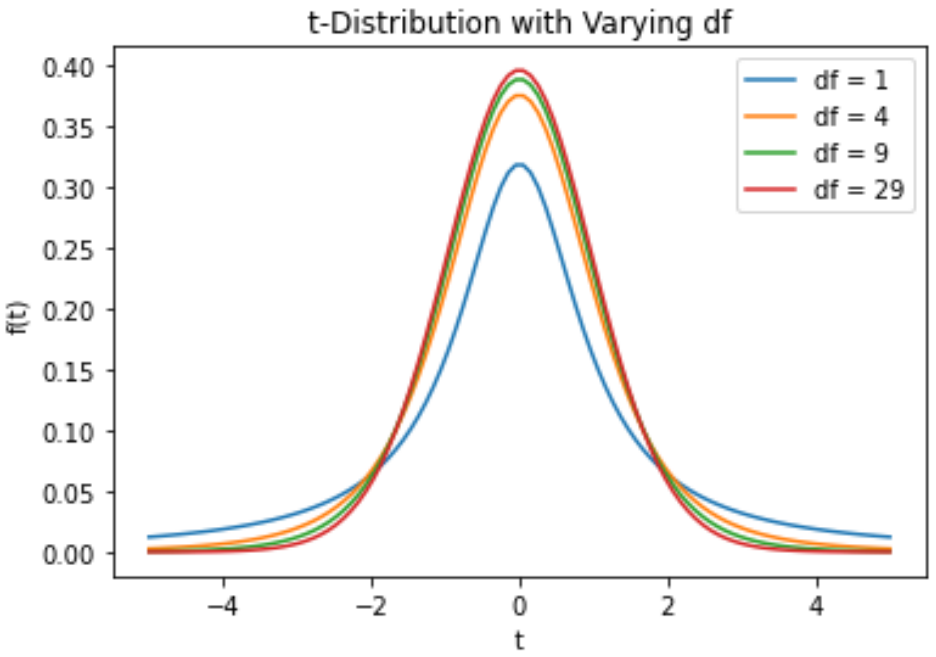
```
# 두 개의 그래프를 한 페이지에 그림
fig, axs = plt.subplots(ncols=2)
fig, ays = plt.subplots(ncols=2)
sns.histplot(data=t1, x=None, ax=axs[0]).set(title='Histogram of Student-t w/ df=5 & df=10)')
sns.histplot(data=t2, x=None, ax=axs[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/t-1.png')
sns.histplot(data=t3, x=None, ax=ays[0]).set(title='Histogram of Student-t w/ df=15 & df=30)')
sns.histplot(data=t4, x=None, ax=ays[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/t-2.png')
plt.show()
```





b1-ch4-16.py

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import t

x = np.linspace(-5, 5, 100)
degrees_of_freedom = [1, 4, 9, 29]  # Varying degrees of freedom

# Plotting T-distribution curves for different degrees of freedom
for df in degrees_of_freedom:
    y = t.pdf(x, df)  # Using default location and scale parameters (0 and 1)
    plt.plot(x, y, label=f"df = {df}")

plt.xlabel('t')
plt.ylabel('f(t)')
plt.title('t-Distribution with Varying df')
plt.legend()
plt.show()
```



2. 확률분포표

b1-ch4-17.py

```
import pandas as pd
import numpy as np
from scipy.stats.distributions import t as tdist
df = 10
t_1 = np.empty(df)
t_2 = np.empty(df)
t_3 = np.empty(df)
t_4 = np.empty(df)
t_5 = np.empty(df)
# repeat r times:
for j in  range(df):
  t_1[j] = tdist.ppf(0.9, df=j)
  j=j+1
for j in  range(df):
  t_2[j] = tdist.ppf(0.95, df=j)
  j=j+1
for j in  range(df):
  t_3[j] = tdist.ppf(0.975, df=j)
  j=j+1
for j in  range(df):
  t_4[j] = tdist.ppf(0.99, df=j)
  j=j+1
for j in  range(df):
  t_5[j] = tdist.ppf(0.995, df=j)
  j=j+1
```

b1-ch4-17.py

(앞에서 계속)

```
t_dist = pd.DataFrame({'p=0.1':t_1,'p=0.05':t_2,'p=0.025':t_3,'p=0.01':t_4,'p=0.005':t_5})
t = t_dist.dropna(how='all') # to drop if all values in the row are NaN
round(t, 3)
print("t Distribution Table with 'p=0.9, ..., p=0.995' : ", f'\n{round(t, 3)}\n')
```

v \ p	0.1	0.05	0.025	0.01	0.005
1	3.078	6.314	12.706	31.821	63.657
2	1.886	2.920	4.303	6.965	9.923
3	0.1638	2.353	3.182	4.541	5.841
4	1.533	2.132	2.776	3.747	4.604
5	1.476	2.015	2.571	3.365	4.032
6	1.440	1.943	2.447	3.143	3.707
7	1.415	1.895	2.365	2.998	3.499
8	1.397	1.860	2.306	2.896	3.355
9	1.383	1.833	2.262	2.821	3.250

```
t Distribution Table with 'p=0.9, ..., p=0.995' :
   p=0.1  p=0.05  p=0.025  p=0.01  p=0.005
1  3.078   6.314   12.706  31.821   63.657
2  1.886   2.920    4.303   6.965    9.925
3  1.638   2.353    3.182   4.541    5.841
4  1.533   2.132    2.776   3.747    4.604
5  1.476   2.015    2.571   3.365    4.032
6  1.440   1.943    2.447   3.143    3.707
7  1.415   1.895    2.365   2.998    3.499
8  1.397   1.860    2.306   2.896    3.355
9  1.383   1.833    2.262   2.821    3.250
```

1. 확률분포

- 두 개의 독립적인 χ^2-분포에 의해 F-분포가 도출

  X~F(n_1-1, n_2-1)

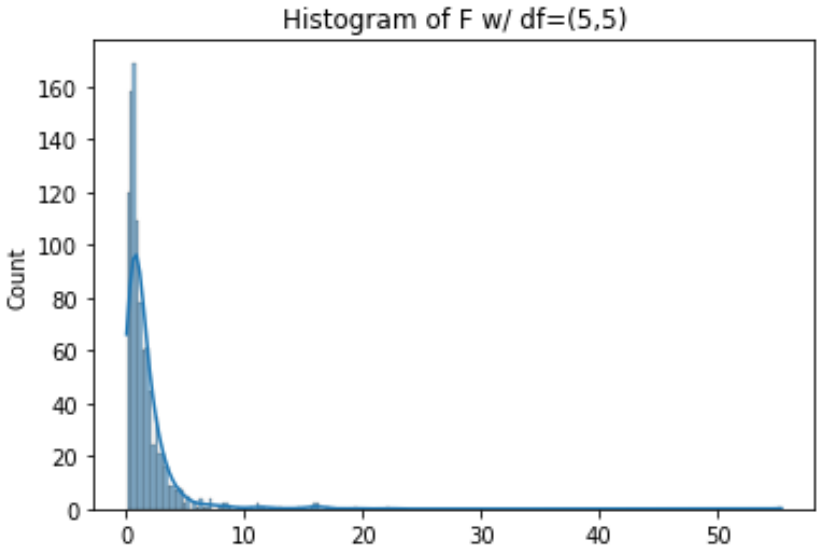
b1-ch4-18.py

```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
r=1000
np.random.seed(12)
z_1 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(34)
z_2 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(56)
z_3 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(78)
z_4 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(910)
z_5 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(21)
z_6 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(43)
z_7 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(65)
z_8 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(87)
z_9 = random.normal(loc=0.0, scale=1.0, size=r)
np.random.seed(109)
z_10 = random.normal(loc=0.0, scale=1.0, size=r)
```

b1-ch4-18.py

(앞에서 계속)

```
chi_5_1 = z_1**2+z_2**2+z_3**2+z_4**2+z_5**2
chi_5_2 = z_6**2+z_7**2+z_8**2+z_9**2+z_10**2

f_5_5 = (chi_5_1/5)/(chi_5_2/5)

sns.histplot(data=f_5_5, x=None, kde=True).set(title='Histogram of F w/ df=(5,5)')
plt.show()
```



- 분자 및 분모의 자유도에 같아지면서 커질수록 좌우대칭 분포와 비슷하게 됨

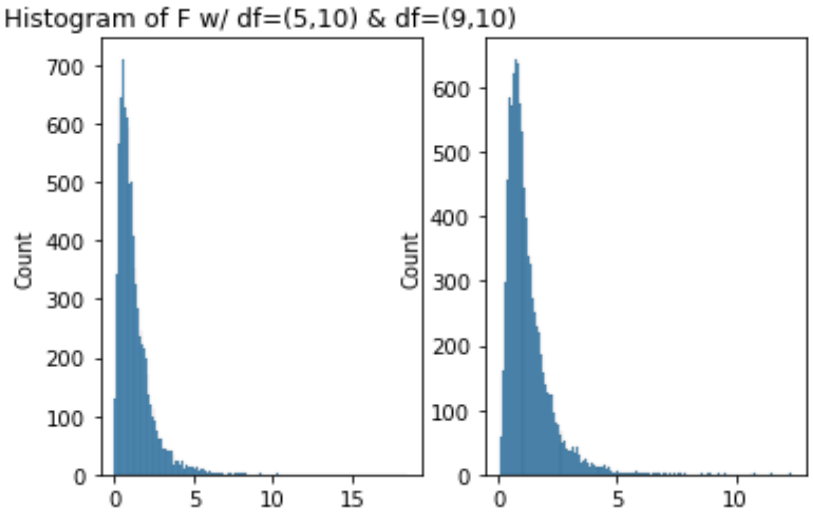
b1-ch4-19.py

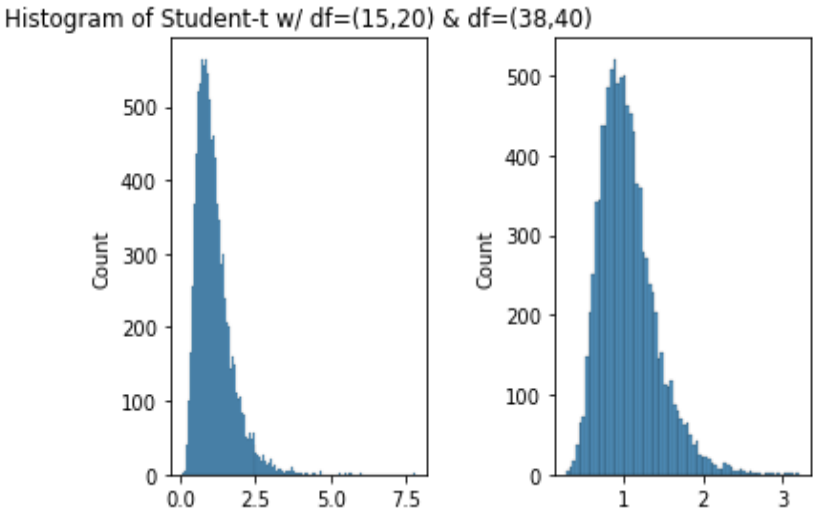
```
import numpy as np
import seaborn as sns # seaborn package를 이용
import matplotlib.pyplot as plt # matplotlib.pyplot package를 이용
from numpy import random
# set the random seed:
np.random.seed(12345)
r=10000
f1 = random.f(5, 10,size=r)
sns.histplot(data=f1, x=None).set(title='Histogram of F w/ df=(5,10)')
plt.show()
f2 = random.f(9, 10, size=r)
sns.histplot(data=f2, x=None).set(title='Histogram of F w/ df=(9,10)')
plt.show()
f3 = random.f(15,20, size=r)
sns.histplot(data=f3, x=None).set(title='Histogram of F / df=(15,20)')
plt.show()
f4 = random.f(38,40, size=r)
sns.histplot(data=f4, x=None).set(title='Histogram of F w/ df=(38,40)')
plt.show()
fig = plt.figure(figsize=(14,7))
```

b1-ch4-19.py

(앞에서 계속)

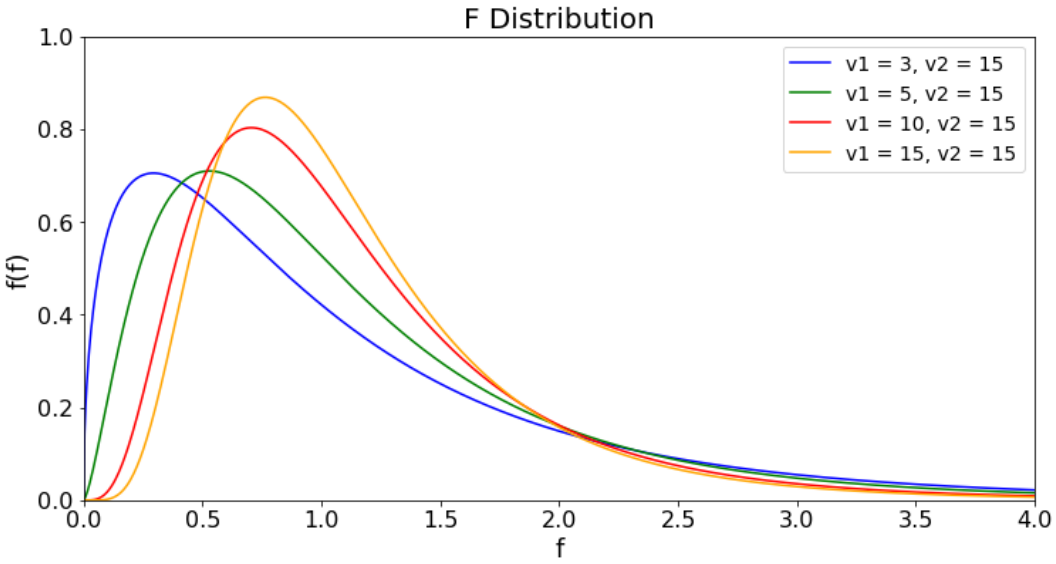
```
# 두 개의 그래프를 한 페이지에 그림
fig, axs = plt.subplots(ncols=2)
fig, ays = plt.subplots(ncols=2)
sns.histplot(data=f1, x=None, ax=axs[0]).set(title='Histogram of F w/ df=(5,10) & df=(9,10)')
sns.histplot(data=f2, x=None, ax=axs[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/f-1.png')
sns.histplot(data=f3, x=None, ax=ays[0]).set(title='Histogram of Student-t w/ df=(15,20) & df=(38,40)')
sns.histplot(data=f4, x=None, ax=ays[1])
fig.subplots_adjust(wspace=0.5) # 우측그림의 좌우 간격을 조정
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/f-2.png')
plt.show()
```





b1-ch4-20.py

```
import numpy as np
from scipy.stats import f
import matplotlib.pyplot as plt
# get x values
x = np.linspace(0, 4.5, 1000)
# get F-Distributions
f1 = f(3, 15, 0)
f2 = f(5, 15, 0)
f3 = f(10, 15, 0)
f4 = f(15, 15, 0)
# plot the distributions
plt.figure(figsize=(12, 6))
plt.plot(x, f1.pdf(x), label = 'v1 = 3, v2 = 15', color = 'blue')
plt.plot(x, f2.pdf(x), label = 'v1 = 5, v2 = 15', color = 'green')
plt.plot(x, f3.pdf(x), label = 'v1 = 10, v2 = 15', color = 'red')
plt.plot(x, f4.pdf(x), label = 'v1 = 15, v2 = 15', color = 'orange')
plt.xlim(0, 4)
plt.ylim(0.0, 1)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel('f', fontsize=18)
plt.ylabel('f(f)', fontsize=18)
plt.title("F Distribution",fontsize=20)
plt.legend(fontsize=14)
#plt.savefig('C:/BOOK/PyBasics/PyStat/code/f_dist.png')
plt.show()
```



2. 확률분포표

b1-ch4-21.py

```
import pandas as pd
import numpy as np
from scipy.stats.distributions import f as fdist
df = 11
f_1 = np.empty(df)
f_2 = np.empty(df)
f_3 = np.empty(df)
f_4 = np.empty(df)
f_5 = np.empty(df)
f_6 = np.empty(df)
f_7 = np.empty(df)
f_8 = np.empty(df)
f_9 = np.empty(df)
f_10 = np.empty(df)
# repeat r times:
for j in  range(df):
  f_1[j] = fdist.ppf(0.95, 1, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_2[j] = fdist.ppf(0.95, 2, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_3[j] = fdist.ppf(0.95, 3, j)
  j=j+1
```

b1-ch4-21.py

```
(옆에서 계속)
f_4[j] = fdist.ppf(0.95, 4, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_5[j] = fdist.ppf(0.95, 5, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_6[j] = fdist.ppf(0.95, 6, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_7[j] = fdist.ppf(0.95, 7, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_8[j] = fdist.ppf(0.95, 8, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_9[j] = fdist.ppf(0.95, 9, j)
  j=j+1
# repeat r times:
for j in  range(df):
  f_10[j] = fdist.ppf(0.95, 10, j)
  j=j+1
```

b1-ch4-21.py

(안ㅍ에서 계속)

```
f_dist = pd.DataFrame({'v2=1':f_1,'v2=2':f_2,'v2=3':f_3,'v2=4':f_4,'v2=5':f_5,'v2=6':f_6,'v2=7':f_7,'v2=8':f_8,'v2=9':f_9,'v2=10':f_10})
f = f_dist.dropna(how='all') # to drop if all values in the row are NaN
round(f, 2)
print("F Distribution Table at 5% significancr level with 'v2=1, ..., v2=10' : ", f'\n{round(f, 2)}\n')
```

v_2 \ v_1	1	2	3	4	5	6	7	8	9	10
1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	240.54	241.88
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.76
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96
5	6.61	5.79	5.41	5.19	5.05	4.95	4.48	4.82	4.77	4.74
6	5.99	4.74	7.35	4.12	3.94	3.87	3.79	3.73	3.68	3.64
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64
8	5.32	4346	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98

```
F Distribution Table at 5% significancr level with 'v2=1, ..., v2=10' :
      v2=1    v2=2    v2=3    v2=4  ...    v2=7    v2=8    v2=9   v2=10
1   161.45  199.50  215.71  224.58  ...  236.77  238.88  240.54  241.88
2    18.51   19.00   19.16   19.25  ...   19.35   19.37   19.38   19.40
3    10.13    9.55    9.28    9.12  ...    8.89    8.85    8.81    8.79
4     7.71    6.94    6.59    6.39  ...    6.09    6.04    6.00    5.96
5     6.61    5.79    5.41    5.19  ...    4.88    4.82    4.77    4.74
6     5.99    5.14    4.76    4.53  ...    4.21    4.15    4.10    4.06
7     5.59    4.74    4.35    4.12  ...    3.79    3.73    3.68    3.64
8     5.32    4.46    4.07    3.84  ...    3.50    3.44    3.39    3.35
9     5.12    4.26    3.86    3.63  ...    3.29    3.23    3.18    3.14
10    4.96    4.10    3.71    3.48  ...    3.14    3.07    3.02    2.98
```