분석 예제 데이터

LPGA 데이터 상금 순위 40등, 41등 이하 선수로 구별

> Target 집단 : 상금상위, 상금하위 판별변수 : 경기력 변수 6개

```
import pandas as pd
df=pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/lpga2008.csv')
df.columns=['golfer','drive','fairway','green','putting','sand_no','sand_save','money','play_no']
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 9 columns):
golfer       157 non-null object
drive        157 non-null float64
fairway      157 non-null float64
green        157 non-null float64
putting      157 non-null float64
sand_no      157 non-null float64
sand_save    157 non-null float64
money        157 non-null int64
play_no      157 non-null int64
dtypes: float64(6), int64(2), object(1)
memory usage: 11.1+ KB
```

선수 이름 행인덱스 만들기

```
df['Name']=df['golfer'].str.split(',',n=1, expand=True)[0]
df.set_index('Name',inplace=True)
df.head(3)
```

Name	golfer	drive	fairway	green	putting	sand_no	sand_save	money	play_no
Ahn	Ahn, Shi Hyun	249.4	64.6	61.2	27.44	1.10	34.5	6063	50
Alfredsson	Alfredsson, Helen	253.8	62.7	68.2	29.36	0.66	38.8	19343	74
Ammaccapane									

상금 순위변수 만들고 집단변수 만들기

```
df['rank']=df['money'].rank(method='min',ascending=False)
df['group']=['up' if x<=40 else 'dn' for x in df['rank']]
df.head(3)
```

Name	golfer	drive	fairway	green	putting	sand_no	sand_save	money	play_no	rank	group
Ahn	Ahn, Shi Hyun	249.4	64.6	61.2	27.44	1.10	34.5	6063	50	49.0	dn
Alfredsson	Alfredsson, Helen	253.8	62.7	68.2	29.36	0.66	38.8	19343	74	5.0	up
Ammaccapane											

BOX M-test []

> 집단 간 공분산 동등성 검증

> 공분산 동일하면 선형 판별분석, 이분산이면 이차판별분석 실시

> 파이썬에는 이를 제공하지 않음

```
import rpy2
%load_ext rpy2.ipython
```

```
The rpy2.ipython extension is already loaded. To reload it, use:
  %reload_ext rpy2.ipython
```

```
%%R
lpga<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/lpga2008.csv',fileEncoding="utf-8")
lpga$상금그룹[rank(-lpga$상금)<=40]<-'상'
lpga$상금그룹[rank(-lpga$상금)>40]<-'하'
head(lpga)
```

```
                골퍼 평균_비거리 페어웨이_안착율 그린_적중률 평균_퍼팅수
1     Ahn, Shi Hyun       249.4            64.6        61.2       27.44
2 Alfredsson, Helen       253.8            62.7        68.2       29.36
3 Ammaccapane, Dina       246.3            70.2        64.6       30.20
4       Bader, Beth       249.1            64.1        61.2       29.78
5       Bae, Kyeong       244.0            62.4        60.7       28.38
6     Baena, Marisa       254.2            64.7        60.9       29.21
  샌드_회수 샌드_세이브  상금 참가_라운드수 상금그룹
1      1.10        34.5  6063           50       하
2      0.66        38.8 19343           74       상
3      0.74        40.5  1873           50       하
4      1.12        41.1  1212           65       하
5      1.02        43.9  2555           65       하
6      1.27        33.3  2282           52       하
```

```
%%R
#install.packages('heplots')
library(heplots)
boxM(lpga[,2:7],lpga$상금그룹)
```

```
        Box's M-test for Homogeneity of Covariance Matrices

data:  lpga[, 2:7]
Chi-Sq (approx.) = 59.804, df = 21, p-value = 1.367e-05
```

> 등분산 가정이 만족하지 않아 이차 판별분석을 적용해야(정분류 비율이 높음)

> 왜 그런지 알아보기 위하여 선형 판별분석도 실시하였다.

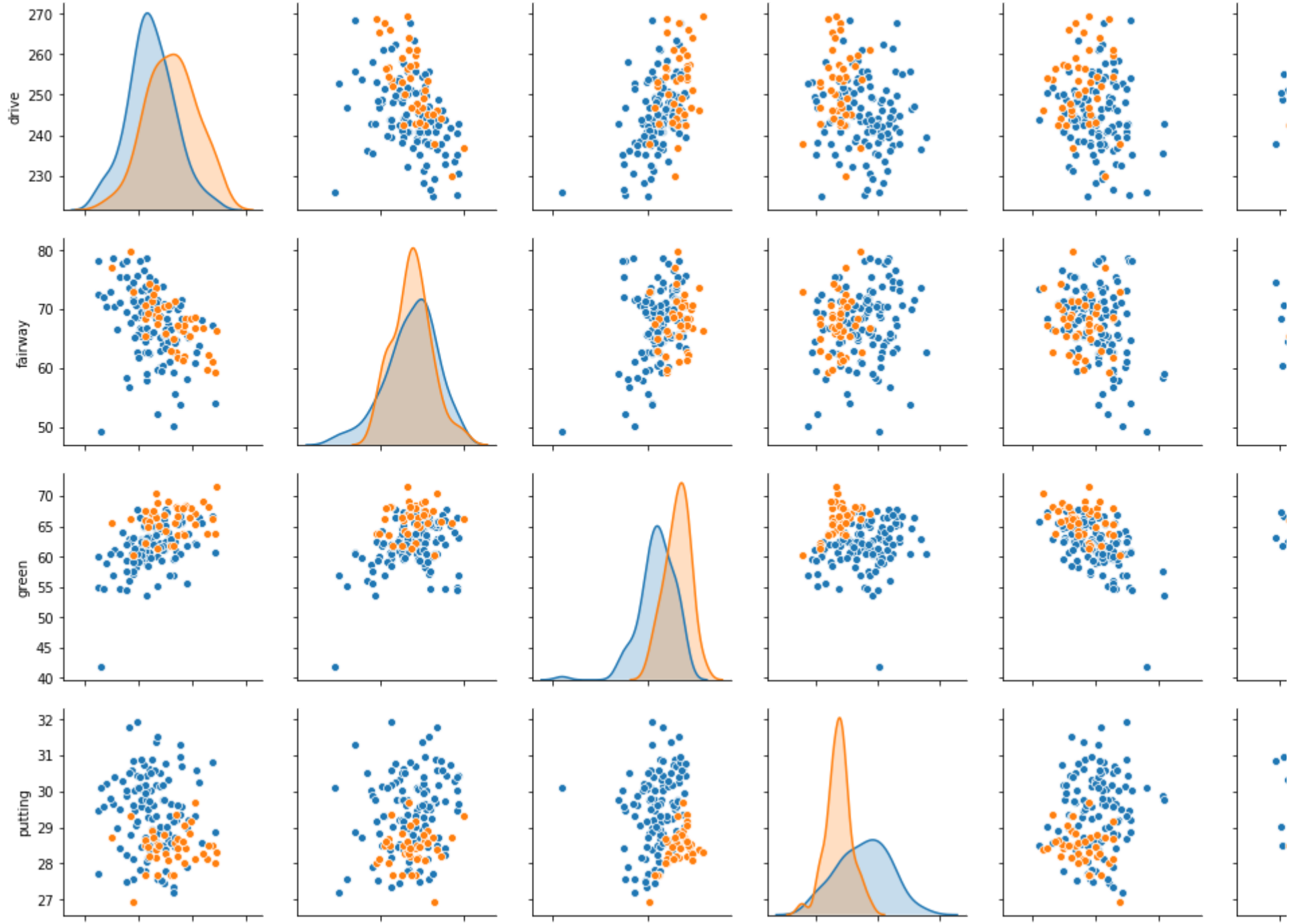
Scatter plot(판별변수) by Group

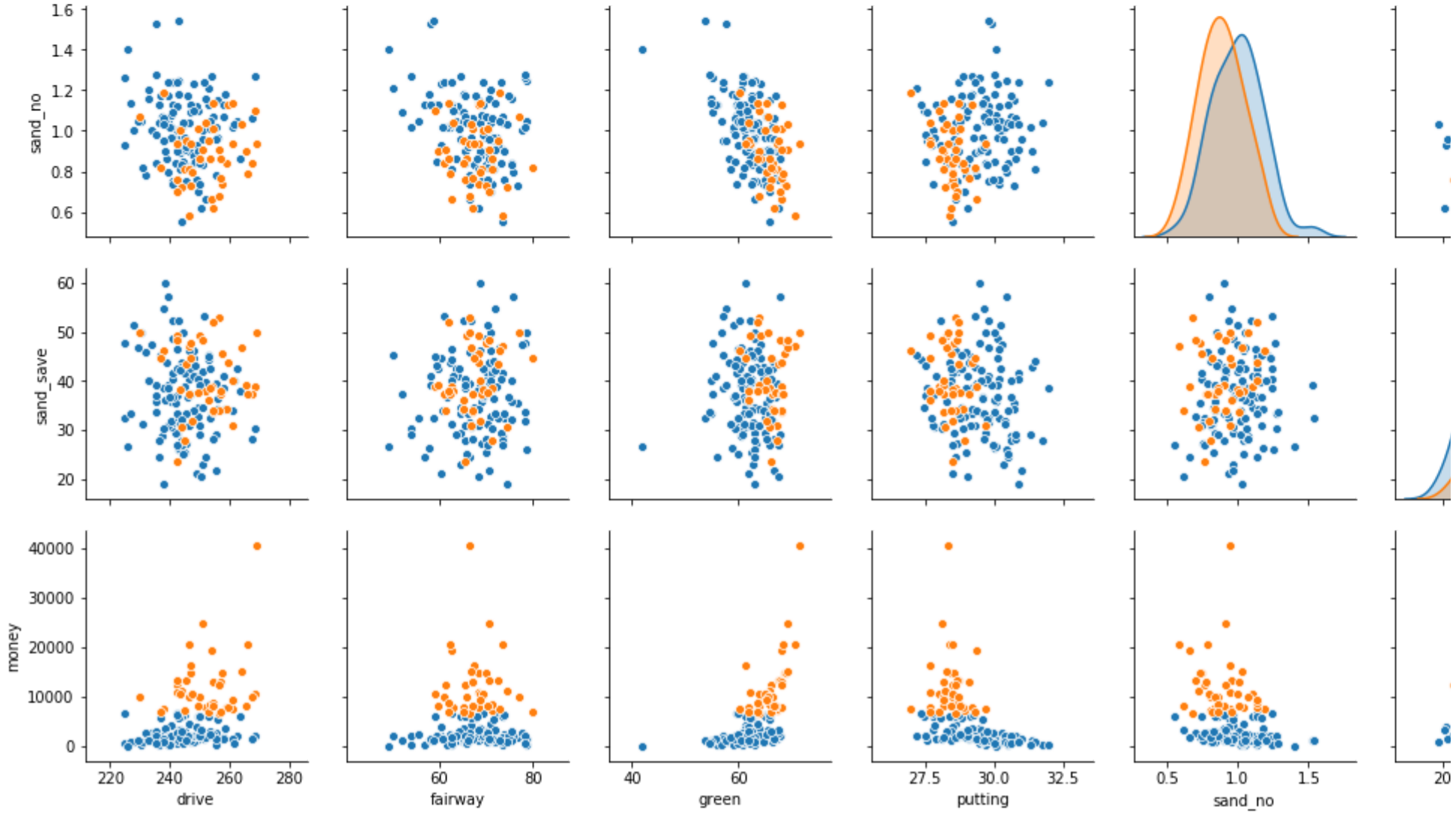
```
df0=df.iloc[:,[1,2,3,4,5,6,7,10]] #판별변수와 집단변수만 subset
df0.head(3)
```

Name	drive	fairway	green	putting	sand_no	sand_save	money	group
Ahn	249.4	64.6	61.2	27.44	1.10	34.5	6063	dn
Alfredsson	253.8	62.7	68.2	29.36	0.66	38.8	19343	up
Ammaccapane	246.3	70.2	64.6	30.20	0.74	40.5	1873	dn

```
import seaborn as sns
sns.pairplot(df0, hue='group')
```

```
<seaborn.axisgrid.PairGrid at 0x7f560682b898>
```





Fisher Qurdratic Discriminant analysis [피셔 2차 판별분석]

> QDA() 함수에 의해 선형판별분석 결과가 저장된다

```
import numpy as np
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as QDA
X=df0.iloc[:,1:7].values
y=df0.iloc[:,7].values
qda=QDA()
X_lda=qda.fit(X, y)
```

집단 소속확률

```
qda.predict_proba(X)[0:3] #Estimate probability
```

```
array([[5.71105788e-01, 4.28894212e-01],
       [7.43414964e-45, 1.00000000e+00],
       [9.99975694e-01, 2.43062683e-05]])
```

> 첫번째 선수 Ahn, Shi Hyun : DN(알파벳 상 먼저 나옴) 집단 속할 확률 = 57.1%, UP집단 속할 확률 42.9%
>
> 그러므로 Ahn, Shi Hyun 피셔 2차판별규칙에 의한 판별 결과는 DN(상금 하위) 선수이다.
>
> 판별규칙에 의한 판별결과는 lda.predict(X)에 저장되어 있음

추정판별집단 데이터프레임 : 판별예측결과는 np.array 형식으로 저장되어 있어 데이터프레임으로 변환하고 행 인덱스로 원 데이터의 행인덱스[선수 이름]을 가져왔음.

```
y_pred=pd.DataFrame(qda.predict(X))   #Predict class labels for samples in X.
y_pred.columns=['group_qda']
y_pred.set_index(df0.index,inplace=True)
y_pred.head(3)
```

Name	group_qda
Ahn	dn
Alfredsson	up
Ammaccapane	dn

[원데이터+판별집단] 합치기 - 정분류 교차표

```
df_qda=pd.concat([df0,y_pred],axis=1)
df_qda['DA_result']=df_qda.group+'-'+df_qda.group_qda
qda_table=pd.crosstab(df_qda.group,df_qda.group_qda)
qda_table
```

group	group_qda dn	up
dn	109	8
up	0	40

```
qda_table.apply(lambda r: r/r.sum(), axis=1)  #정분류
```

group	group_qda dn	up
dn	0.931624	0.068376
up	0.000000	1.000000

> 피셔 선형 판별규칙 정분류 - dn집단 정분류 93.2%, up집단 정분류 100%로 완벽함

새로운 선수 집단 판별하기

```
new=pd.DataFrame([260,70,65,28,1.5,40]).T
new_X=new.values
print('추정집단확률',qda.predict_proba(new_X),'추정판별집단',qda.predict(new_X))
```

```
추정집단확률 [[1. 0.]] 추정판별집단 ['dn']
```

> 비거리=260, 페어웨이=70%, ..., 샌드세이브=40% 인 선수는 dn(상금 하위) 선수로 판별(dn에 속할 확률은 거의 100%임)

Fisher Linear Discriminant (참고용)

> 분류 결과표 보면 : dn 정분류 100%이나 up 집단 정분류 비율은 75%로 낮음

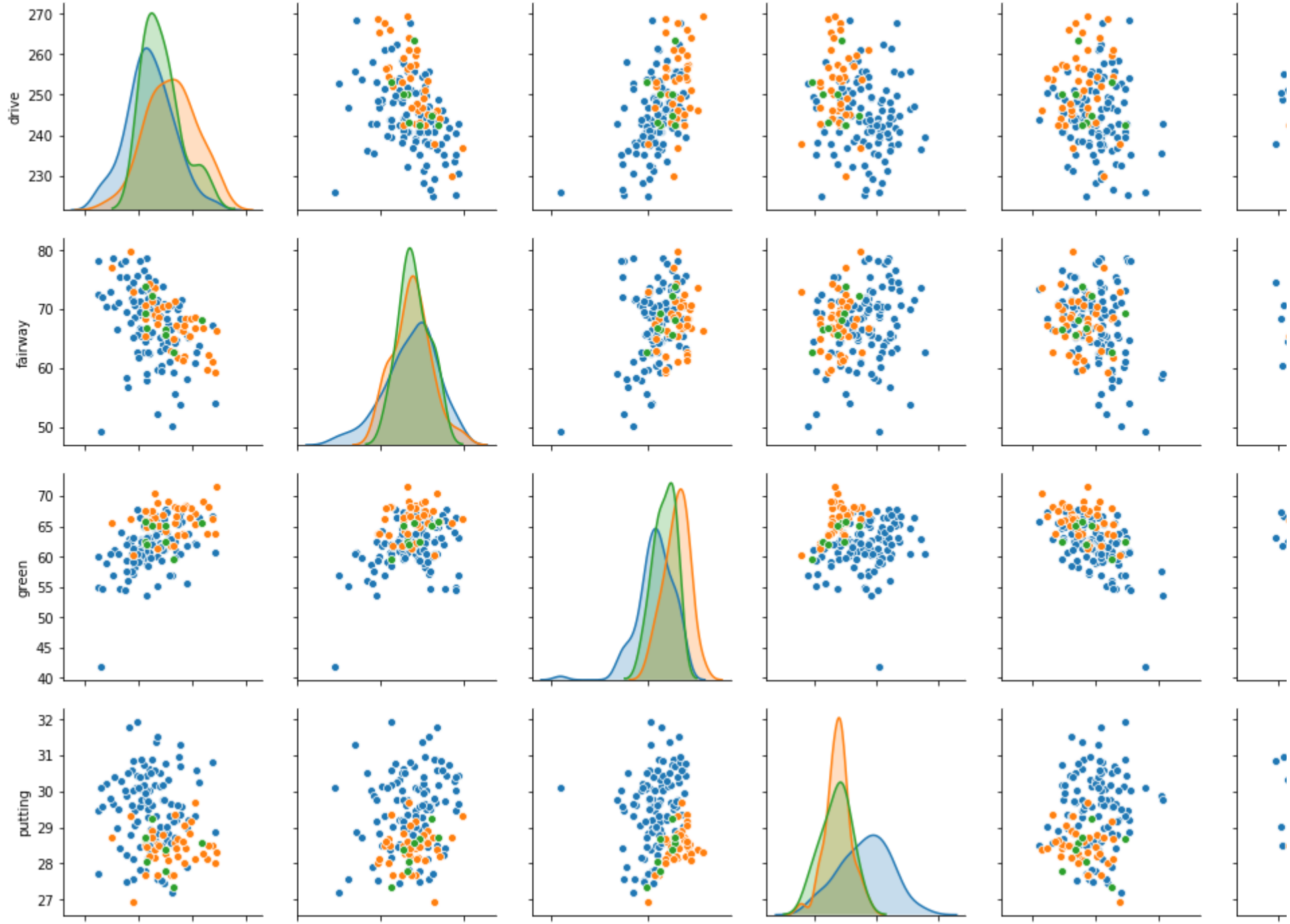
```
import numpy as np
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
X=df0.iloc[:,1:7].values
y=df0.iloc[:,7].values
qda=LDA()
X_qda=lda.fit(X, y)
y_pred2=pd.DataFrame(lda.predict(X))   #Predict class labels for samples in X.
y_pred2.columns=['group_lda']
y_pred2.set_index(df0.index,inplace=True)
df_lda=pd.concat([df0,y_pred2],axis=1)
df_lda['DA_result']=df_lda.group+'-'+df_lda.group_lda
lda_table=pd.crosstab(df_lda.group,df_lda.group_lda)

lda_table.apply(lambda r: r/r.sum(), axis=1)  #정분류
```

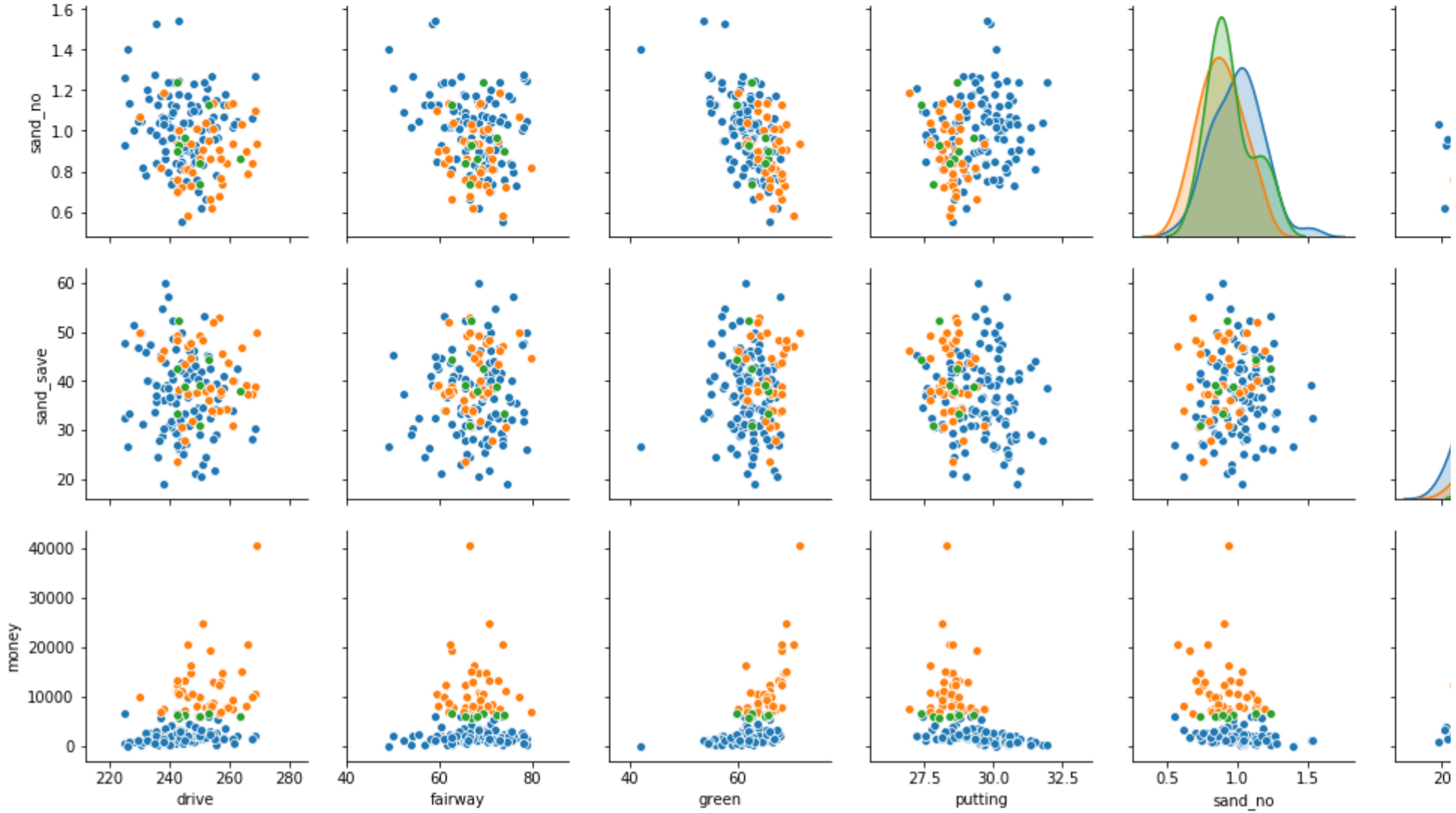
group_lda	dn	up
group		
dn	1.000	0.000
up	0.225	0.775

이차 판별분석 결과 시각화

```
1  import seaborn as sns
2  sns.pairplot(df_qda, hue='DA_result')
```

```
<seaborn.axisgrid.PairGrid at 0x7f5601f51198>
```





판별변수 (평균, 표준편차) by 판별결과

```
df_qda.groupby(['DA_result']).mean()
```

DA_result	drive	fairway	green	putting	sand_no	sand_save	money
dn-dn	244.454128	67.5000	61.715596	29.533211	1.003119	36.840367	2093.137615
dn-up	248.812500	68.1875	63.562500	28.352500	0.951250	39.925000	6299.625000
up-up	252.440000	67.6675	66.102500	28.440000	0.887250	40.592500	12127.025000

주성분 활용 판별결과 보기

```
# Standardizing the features
from sklearn.preprocessing import StandardScaler
df_s=StandardScaler().fit_transform(df_qda.iloc[:,0:6])
# PCA
from sklearn.decomposition import PCA
pca=PCA(0.8) #80% rule pca=PCA(0.8)
df_pca=pca.fit_transform(df_s) #PC variables
```

```
df_loading=pd.DataFrame(pca.components_.T)  #loading values
df_loading
```

	0	1	2	3
0	-0.399759	0.579306	0.154754	-0.321000
1	-0.195232	-0.706543	-0.121244	0.009798
2	-0.665787	-0.070567	0.101261	-0.277574
3	0.144779	-0.358622	0.598731	-0.582617
4	0.581238	0.169826	0.035767	-0.387920
5	0.002420	-0.052635	-0.768986	-0.574362

```
df_pca=pd.DataFrame(df_pca,columns=['ParOn(-)','Power(+)','RiskManage(-)','PC4'])
```

```
df_pca.set_index(df_qda.index,inplace=True)  #주성분점수 데이터프레임 행 인덱스 원 데이터 사용
pca_df=pd.concat([df_qda,df_pca],axis=1)
pca_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 157 entries, Ahn to Yoo
Data columns (total 14 columns):
drive            157 non-null float64
fairway          157 non-null float64
green            157 non-null float64
putting          157 non-null float64
sand_no          157 non-null float64
sand_save        157 non-null float64
money            157 non-null int64
group            157 non-null object
group_qda        157 non-null object
DA_result        157 non-null object
ParOn(-)         157 non-null float64
Power(+)         157 non-null float64
RiskManage(-)    157 non-null float64
PC4              157 non-null float64
dtypes: float64(10), int64(1), object(3)
memory usage: 23.4+ KB
```

```
pca_df0=pca_df.iloc[:,[9,10,11,12,13]]
import seaborn as sns
sns.pairplot(pca_df0, hue='DA_result')
```

```
<seaborn.axisgrid.PairGrid at 0x7f55e554f2b0>
```