01. 비지도 학습 예제 - 붓꽃

```
붓꽃 데이터는 몇 차원인가?
(1) 우리는 4차원이나 그 이상보다 2차원 데이터를 플로팅(그래프)하는 것이 쉽다.
```

Principal component analysis (PCA)

[http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html (http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html)](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html)

In [20]:

```
from sklearn.decomposition import PCA   # 모델 클래스 선택
import seaborn as sns
%matplotlib inline
```

In [25]:

```
iris = sns.load_dataset('iris')
iris.head()
print(type(iris))
X_iris = iris.drop('species', axis=1)
y_iris = iris['species']
X_iris
```

```
<class 'pandas.core.frame.DataFrame'>
```

Out[25]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4
17	5.1	3.5	1.4	0.3
18	5.7	3.8	1.7	0.3
19	5.1	3.8	1.5	0.3

	sepal_length	sepal_width	petal_length	petal_width
20	5.4	3.4	1.7	0.2
21	5.1	3.7	1.5	0.4
22	4.6	3.6	1.0	0.2
23	5.1	3.3	1.7	0.5
24	4.8	3.4	1.9	0.2
25	5.0	3.0	1.6	0.2
26	5.0	3.4	1.6	0.4
27	5.2	3.5	1.5	0.2
28	5.2	3.4	1.4	0.2
29	4.7	3.2	1.6	0.2
...	...	...	...	...
120	6.9	3.2	5.7	2.3
121	5.6	2.8	4.9	2.0
122	7.7	2.8	6.7	2.0
123	6.3	2.7	4.9	1.8
124	6.7	3.3	5.7	2.1
125	7.2	3.2	6.0	1.8
126	6.2	2.8	4.8	1.8
127	6.1	3.0	4.9	1.8
128	6.4	2.8	5.6	2.1
129	7.2	3.0	5.8	1.6
130	7.4	2.8	6.1	1.9
131	7.9	3.8	6.4	2.0
132	6.4	2.8	5.6	2.2
133	6.3	2.8	5.1	1.5
134	6.1	2.6	5.6	1.4

	sepal_length	sepal_width	petal_length	petal_width
135	7.7	3.0	6.1	2.3
136	6.3	3.4	5.6	2.4
137	6.4	3.1	5.5	1.8
138	6.0	3.0	4.8	1.8
139	6.9	3.1	5.4	2.1
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
142	5.8	2.7	5.1	1.9
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [36]:

```
model = PCA(n_components=2)             # 유지할 구성 요소수. 없으면 전체 선택
model.fit(X_iris)                       # y는 지정 안함. 데이터 학습
X_2D = model.transform(X_iris)          # 데이터를 2차원으로 변환
```

In [37]:
```
print(X_iris.head())
print(X_2D)
```

```
   sepal_length  sepal_width  petal_length  petal_width
0           5.1          3.5           1.4          0.2
1           4.9          3.0           1.4          0.2
2           4.7          3.2           1.3          0.2
3           4.6          3.1           1.5          0.2
4           5.0          3.6           1.4          0.2
[[-2.68412563  0.31939725]
 [-2.71414169 -0.17700123]
 [-2.88899057 -0.14494943]
 [-2.74534286 -0.31829898]
 [-2.72871654  0.32675451]
 [-2.28085963  0.74133045]
 [-2.82053775 -0.08946138]
 [-2.62614497  0.16338496]
 [-2.88638273 -0.57831175]
 [-2.6727558  -0.11377425]
 [-2.50694709  0.6450689 ]
 [-2.61275523  0.01472994]
 [-2.78610927 -0.235112  ]
```

```
In [23]: X_2D[: , 0]    # 첫번째 열 데이터
```

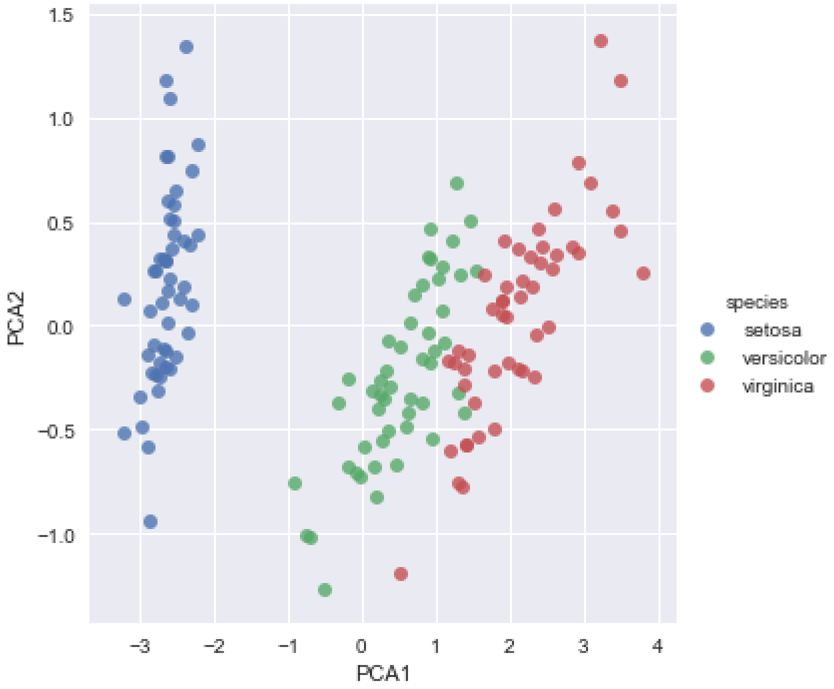
```
Out[23]: array([-2.68412563, -2.71414169, -2.88899057, -2.74534286, -2.72871654,
               -2.28085963, -2.82053775, -2.62614497, -2.88638273, -2.6727558 ,
               -2.50694709, -2.61275523, -2.78610927, -3.22380374, -2.64475039,
               -2.38603903, -2.62352788, -2.64829671, -2.19982032, -2.5879864 ,
               -2.31025622, -2.54370523, -3.21593942, -2.30273318, -2.35575405,
               -2.50666891, -2.46882007, -2.56231991, -2.63953472, -2.63198939,
               -2.58739848, -2.4099325 , -2.64886233, -2.59873675, -2.63692688,
               -2.86624165, -2.62523805, -2.80068412, -2.98050204, -2.59000631,
               -2.77010243, -2.84936871, -2.99740655, -2.40561449, -2.20948924,
               -2.71445143, -2.53814826, -2.83946217, -2.54308575, -2.70335978,
                1.28482569,  0.93248853,  1.46430232,  0.18331772,  1.08810326,
                0.64166908,  1.09506066, -0.74912267,  1.04413183, -0.0087454 ,
               -0.50784088,  0.51169856,  0.26497651,  0.98493451, -0.17392537,
                0.92786078,  0.66028376,  0.23610499,  0.94473373,  0.04522698,
                1.11628318,  0.35788842,  1.29818388,  0.92172892,  0.71485333,
                0.90017437,  1.33202444,  1.55780216,  0.81329065, -0.30558378,
               -0.06812649, -0.18962247,  0.13642871,  1.38002644,  0.58800644,
                0.80685831,  1.22069088,  0.81509524,  0.24595768,  0.16641322,
                0.46480029,  0.8908152 ,  0.23054802, -0.70453176,  0.35698149,
                0.33193448,  0.37621565,  0.64257601, -0.90646986,  0.29900084,
                2.53119273,  1.41523588,  2.61667602,  1.97153105,  2.35000592,
                3.39703874,  0.52123224,  2.93258707,  2.32122882,  2.91675097,
                1.66177415,  1.80340195,  2.1655918 ,  1.34616358,  1.58592822,
                1.90445637,  1.94968906,  3.48705536,  3.79564542,  1.30079171,
                2.42781791,  1.19900111,  3.49992004,  1.38876613,  2.2754305 ,
                2.61409047,  1.25850816,  1.29113206,  2.12360872,  2.38800302,
                2.84167278,  3.23067366,  2.15943764,  1.44416124,  1.78129481,
                3.07649993,  2.14424331,  1.90509815,  1.16932634,  2.10761114,
                2.31415471,  1.9222678 ,  1.41523588,  2.56301338,  2.41874618,
                1.94410979,  1.52716661,  1.76434572,  1.90094161,  1.39018886])
```

sns.lmplot

```
sns.lmplot(x, y,    # 입력 variable
           hue=""  # 데이터의 일부 집합을 정의
           data=iris  # 데이터 프레임
          fit_reg=False  # TRUE : 회귀 모델을 추정하고 플롯한다.
```

https://seaborn.pydata.org/generated/seaborn.lmplot.html (https://seaborn.pydata.org/generated/seaborn.lmplot.html)

In [48]:
```
iris['PCA1'] = X_2D[: ,0]  # feature 생성
iris['PCA2'] = X_2D[: ,1]  # feature 생성
sns.lmplot("PCA1", "PCA2", hue="species", data=iris, fit_reg=False);
```



붓꽃(Iris)에 대한 정보가 없음에도 2차원 표현에서 종(Species)가 매우 잘 분리되어 있다.

In [ ]: