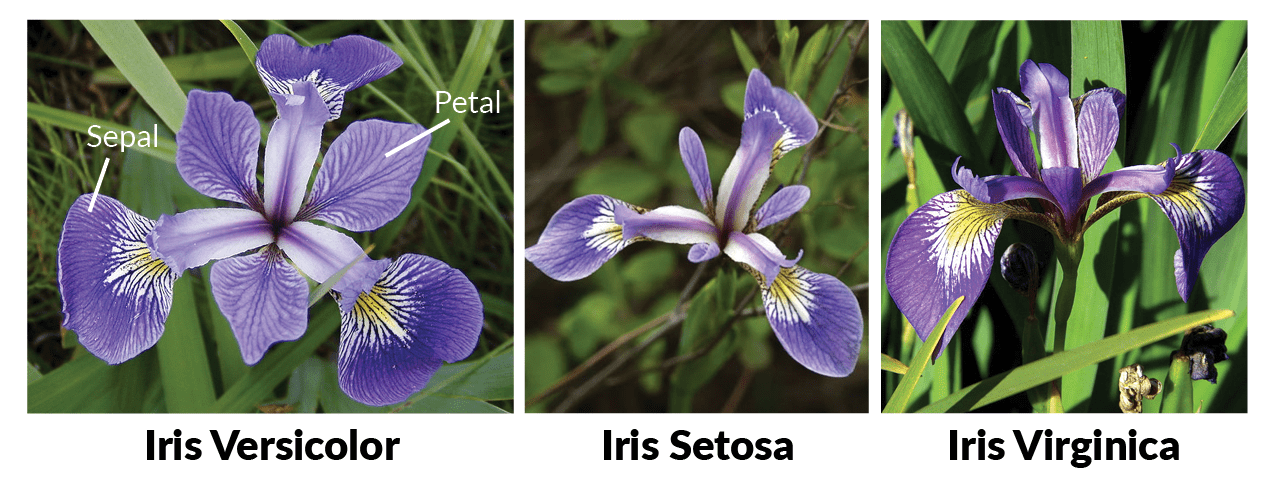
**배경설명:**

Iris 데이터 세트에는 iris 3종 (setosa, virginica 및 versicolor) 50 개 샘플의 4 가지 특징 (꽃받침과 꽃잎의 길이 및 너비)이 포함되어 있습니다. 이 관측치들은 종을 분류하는 선형 판별 모델을 만드는데 사용되었고, 데이터 세트는 데이터 마이닝, 분류 및 클러스터링 예제와 알고리즘 테스트에 자주 사용됩니다.

다음은 세 가지 꽃 종의 사진입니다.

본격적인 분석 전에 데이터 구조를 살펴보는게 필요합니다.

**library**(datasets)

**str**(iris)

구조가 파악되었으니, 각 class 별로 다음 명령어로 데이터를 살펴보겠습니다.

**subset**(iris, Species == "setosa")[1:5,]

**subset**(iris, Species == "versicolor")[1:5,]

**subset**(iris, Species == "virginica")[1:5,]

결과 데이터를 살펴보면, setosa class의 Petal.Length가 다른 class의 Petal.Length보다 짧은 것을 알 수 있습니다.

그러므로, 이를 확인하고자, 아래 code를 수행하면 Petal.Length가 2보다 작은 class는 모두 setosa 임을 알 수 있습니다.

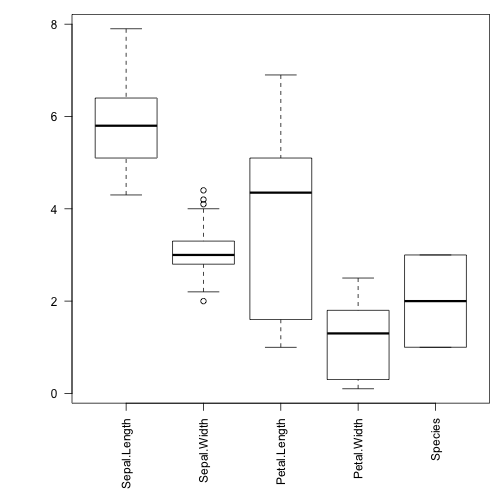
**subset**(iris, Petal.Length < 2)[,"Species"]

그리고, 데이터에 대해 좀 더 알기 위해 우리는 iris 데이터프레임의 각 변수에 대한 기본적인 통계량을 아래와 같이 간단히 파악할 수 있다.

**summary**(iris)

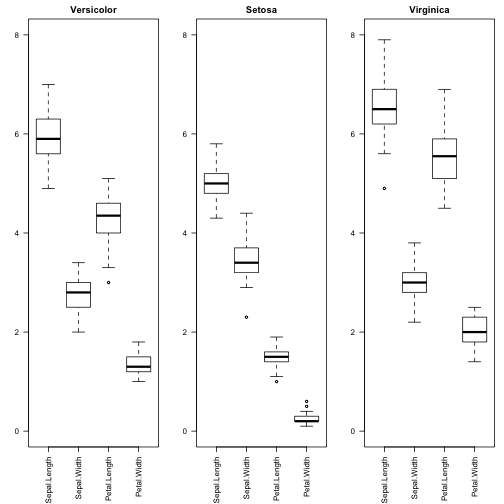
summary() 함수 결과는 많은 정보를 제공하지만, 통계값들을 시각적으로 보는 것이 더 좋을 때가 있다.

**Q1. 다음과 같은 그래프를 생성하는 코드를 제시하시오. par ()를 사용할 수 있습니다.**



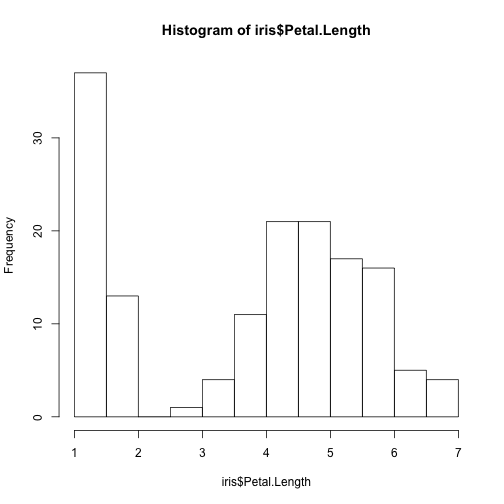
Boxplot을 위와 같이 해석할 줄 알아야 합니다. 이를 통해 각 속성 값의 분포를 대략적으로 추정할 수 있습니다. 그러나, 각 class에 대한 label이 있으므로 각 class를 고려하여 값의 분포를 보는 것이 더 합리적 일 것입니다. 다음 Q2 문제입니다.

**Q2. 다음과 같이 class 별 그래프를 생성하는 코드를 제시하시오.**



히스토그램을 각 속성별로 아래와 같이 그려보는 것도 데이터를 이해하는데 도움이 됩니다.

**hist**(iris$Petal.Length)



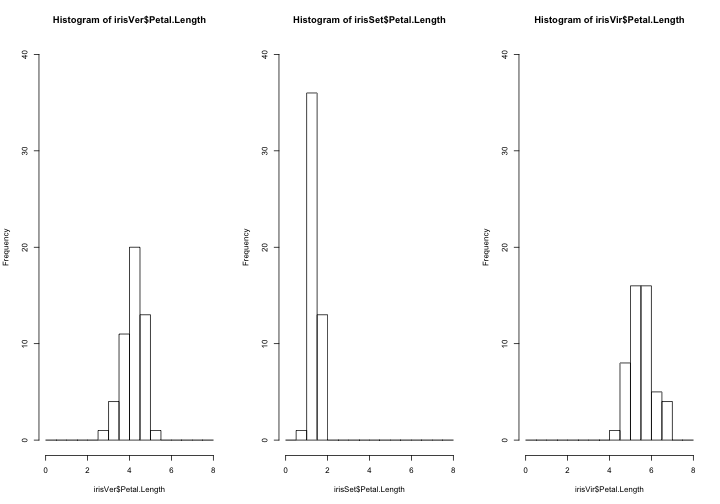
각 class에 대해 특정 속성을 히스토그램으로 표현 후 서로 비교해보면 데이터를 더 잘 이해할 수 있습니다.

**par**(mfrow=**c**(1,3))

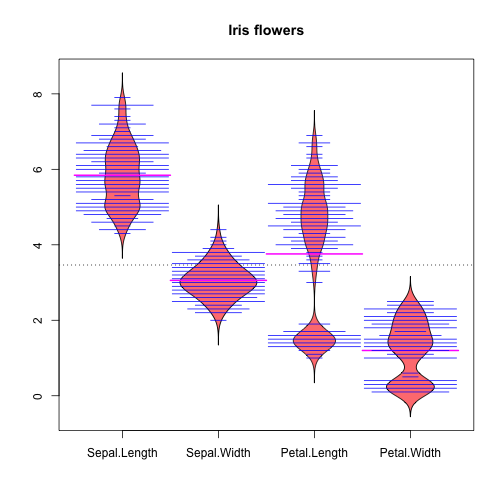
**hist**(irisVer$Petal.Length,breaks=**seq**(0,8,l=17),xlim=**c**(0,8),ylim=**c**(0,40))

**hist**(irisSet$Petal.Length,breaks=**seq**(0,8,l=17),xlim=**c**(0,8),ylim=**c**(0,40))

**hist**(irisVir$Petal.Length,breaks=**seq**(0,8,l=17),xlim=**c**(0,8),ylim=**c**(0,40))



**Q3. 아래는 우리가 강의시간에 학습하지 않은 Bean plot이란 그래프입니다. 다음과 같은 그래프를 생성하는 코드를 제시하시오. “beanplot” package를 설치해야 합니다.**



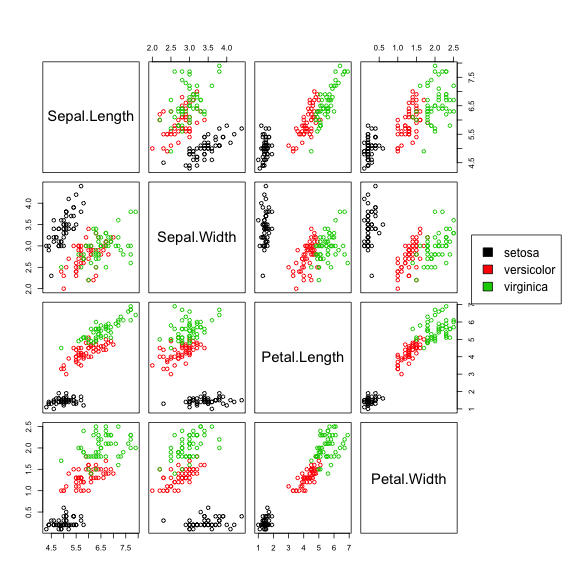
**Q4. 한 변수가 다른 변수와 상관관계가 있음을 다음과 같은 코드를 사용해 확인할 수 있습니다. Class 별 상관관계는 class를 색을 달리함으로써 확인할 수 있습니다. 아래와 같은 그래프를 생성하는 코드를 제시하시오.**

corr <- **cor**(iris[,1:4])

**round**(corr,3)

산점도는 시각화의 좋은 도구이며, 상관관계 여부를 식별하는데 도움을 줍니다.

**pairs**(iris[,1:4])



**Q5. 상관관계나 일반적으로 값의 시각화를 위해 데이터 프레임의 값을 도시하는 다른 방법은 “parallel coordinate plot”을 그리는 것입니다. 아래와 같은 그래프를 생성하는 코드를 제시하시오. “MASS” library의 parcoord() 함수를 사용할 수 있습니다.**

