

## 2. 히스토그램(histogram) 작성하기

도수분포표(frequency table)란 주어진 자료를 몇 개의 구간으로 나누고, 나누어진 각 구간에 속하는 자료의 개수(도수)를 정리한 표이다. 히스토그램(histogram)은 도수분포표를 시각적으로 표현한 막대그래프이다. 히스토그램을 그리면, 수집한 데이터가 종모양(정규분포)을 모양을 이루고 있는지, 특정 구간에 빈도가 몰려 있는 비대칭적인 분포를 이루고 있는지, 또는 이상치가 들어 있는지를 한눈에 파악할 수 있다. 다시 요약하자면, 도수분포표와 히스토그램은 우리가 수집한 데이터를 요약하여 전반적인 생김새(분포)를 파악할 수 있는 자료표현 방법이다.

### [예제] R에 내장된 지진 발생 데이터 세트(quakes)

1964년 이후 피지 근방에서 발생한 진도 4.0 이상의 지진이 발생한 1000개의 위치를 제공하는 데이터 셋이다.

```
library(MASS)
```

```
str(quakes)
```

```
View(quakes) # V는 대문자에 유의하자.
```

```
'data.frame': 1000 obs. of 5 variables:
```

```
$ lat      : num  -20.4 -20.6 -26 -18 -20.4 ... (위도)
```

```
$ long     : num   182 181 184 182 182 ... (경도)
```

```
$ depth    : int   562 650 42 626 649 195 82 194 211 622 ... (km)
```

```
$ mag      : num   4.8 4.2 5.4 4.1 4 4 4.8 4.4 4.7 4.3 ... (Richter Magnitude 리히터 크기, 규모)
```

```
$ stations : int    41 15 43 19 11 12 43 15 35 19 ... Number of stations reporting
```

```
head(quakes, 200)
```

```
hist( ) 함수
```

```
hist(객체(data), breaks= # 막대 개수 조절
```

```

main='****',      # 제목
xlab='****',      # x축 레이블
ylab='****',      # y축 레이블
col='****',      # 막대 색
border='****',    # 막대 테두리색
las=2,           # x축 글씨 방향(0~3)
)

```

**hist** (quakes\$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포', xlab='진도', ylab='발생 횟수')

**right=F** # 계급의 break 포인트 중 오른쪽을 포함할지에 대한 여부 TRUE이면 (a, b]가 되며 FALSE이면 [a, b)가 됨

#### ■ 히스토그램 꾸미기(색상)

**colors( )** 함수를 실행하면 R 그래프에 사용할 수 있는 색의 이름 표시

```
[1] "white" ..... [657] "yellowgreen"
```

**hist** (quakes\$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포', xlab='진도', ylab='발생 횟수', col="grey", border="white")

**hist** (quakes\$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포', xlab='진도', ylab='발생 횟수', col="66EC66", border="66EC66")

**hist** (quakes\$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포', xlab='진도', ylab='발생 횟수', col="#66EC66", border="#5A5AFF")

**참고.** R에서 그래프 등을 그릴 때 색상을 지정하는 방법은 여러 가지가 있다. 물론 경우에 따라 복잡한 색상의 정의가 필요하거나 하다면 **rgb( )** 함수나 **hcl( )**, **hsv( )** 같은 함수를 사용할 수도 있지만, 그래도 가장 간단하고 많이 쓰이는 방법은 역시 R에서 미리 정의된 색상 이름을 사용하는 것이다.

① 'blue', 'red', 'yellow'와 같이 색의 이름을 지정한다.

② '#0000FF'와 같이 색상별로 지정된 코드를 이용한다.

`colours( )` 함수를 사용하면 미리 정의된 색상 이름들을 보여주지만 번거롭다. 아래의 색상표를 참고하면 도움이 될 것이다.

<https://www.stat.auckland.ac.nz/~ihaka/courses/787/R-colours-a4.pdf>

<https://wepplication.github.io/tools/colorPicker/>

<http://www.hipenpal.com/tool/html-color-charts-rgb-color-table-in-korean.php>

### # 막대의 색을 각각 지정하기

```
hist (quakes$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포',  
xlab='진도', ylab='발생 횟수', col=c('blue','red','green','yellow','orange'))
```

### # 팔레트에서 색을 선택하여 사용하기

- 팔레트 기능은 다양한 색을 미리 준비해놓고 그곳에서 필요한 개수만큼의 색을 가져와 사용하는 것

```
hist (quakes$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포',  
xlab='진도', ylab='발생 횟수', col=rainbow(5))
```

# 레인보우 팔레트에서 5개의 색을 선택 / `rainbow(13)`



R 자동으로 색 설정해주는 함수들...

R에 자동으로 설치되는 `grDevices` 패키지에서 제공되는 함수는

`rainbow( )`, `heat.colors( )`, `terrain.colors( )`, `topo.colors( )`, `cm.colors( )`

등을 제공한다.

### ▣ `las` 값에 따른 `x`축의 그룹 이름의 출력 방향

- 0 : 축 방향(기본값)
- 1 : 수평 방향(축 방향과 상관없음)
- 2 : 축을 기준으로 수직 방향
- 3 : 수직 방향(축 방향과 상관없음)

```
hist (quakes$mag, breaks=seq(4, 6.5, by=0.5), right=F, main='지진의 진도 분포',  
xlab='진도', ylab='발생 횟수', col="#66EC66", border="#5A5AFF", las=1)
```

[실습과제] MASS 패키지에 내장된 **cars** 데이터 프레임의 **dist** 변수를 선택하여 히스토그램으로 작성하자.

```
library(MASS)
```

```
str(cars)
```

```
View(cars)
```

```
'data.frame': 50 obs. of 2 variables:
```

```
$ speed: num 4 4 7 7 8 9 10 10 10 11 ... (주행속도)
```

```
$ dist : num 2 10 4 22 16 10 18 26 34 17 ... (제동거리)
```

※ **cars** 데이터 셋은 자동차의 **주행속도(speed)**에 따른 **제동거리(dist)** 데이터를 저장하고 있는 데이터 프레임

### 3. 막대그래프(Bar Chart)

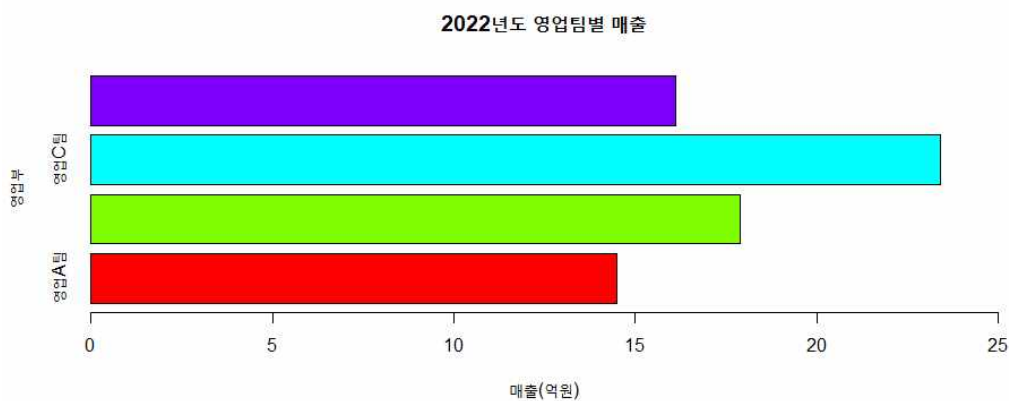
막대그래프는 각 범주(그룹)에서 도수의 크기를 막대로 그려 나타낸다. 막대그래프를 이용하면 각 범주간의 도수를 비교하는데 용이하다. 그러나 각 범주 또는 몇 개의 범주가 전체에서 차지하는 비율을 파악하려면 원형그래프(Pie Chart)가 적당하다.

막대그래프는 그룹별로 집계된 데이터를 표현하는 도구이기 때문에 막대그래프를 작성하기 위해서는 먼저 그룹별로 데이터를 집계하는 작업이 필요하다.

```
x <- c(14.5, 17.9, 23.4, 16.1)
```

```
label <- c('영업A팀', '영업B팀', '영업C팀', '영업D팀')
```

```
barplot( x, horiz=T, width=5,  
         xlim=c(0, 25),  
         col=rainbow(length(x)),  
         xlab='매출(억원)', ylab='영업부',  
         names.arg=label,  
         main='2022년도 영업팀별 매출'  
       )
```



horiz 속성은 막대를 가로로 하고 width(폭)의 크기는 5

```
xlim=c(0, 25) # x축 눈금(범위)
```

```
names.arg # 막대 아래에 벡터(값)의 명칭 넣기
```

```
barplot( x,
```

```
ylim=c(0, 25),
col=rainbow(length(x)),
xlab='영업부', ylab='매출(억원)',
names.arg=label,
main='2022년도 영업팀별 매출' )
```



[실습과제] MASS 패키지에 내장된 **Cars93** 데이터프레임의 **차종형태(Type)** 변수를 막대 그래프로 작성하자.

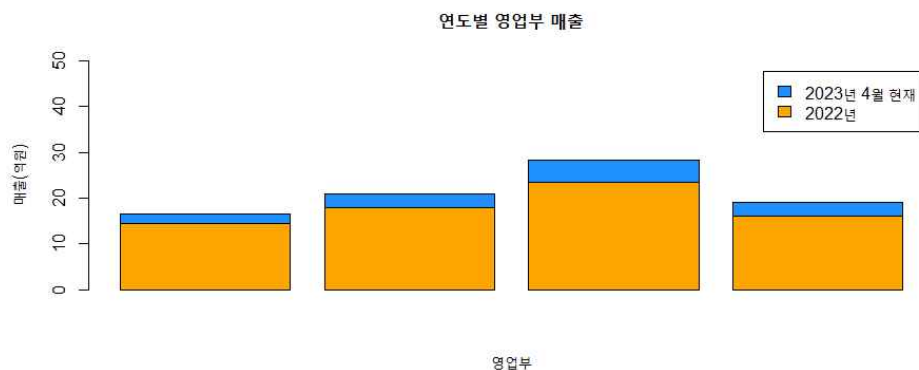
```
library(MASS)
str(Cars93)
View(Cars93)
```

(1) **table( )** 함수 사용하여 도수분포표(frequency distribution table) R-코드를 이용하여 작성하기

(2) 직접 막대그래프로 작성하기

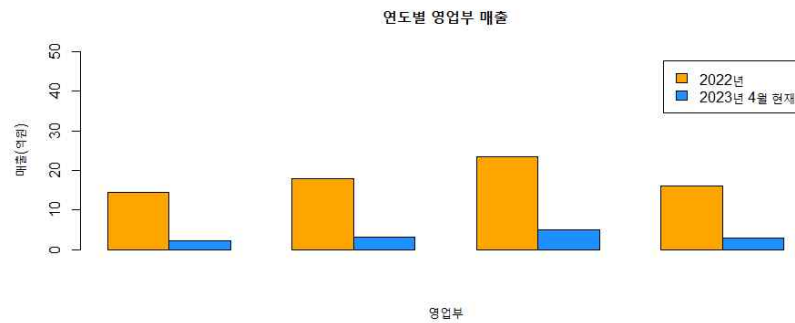
## ■ 중첩 그룹의 barchart

```
x <- c(14.5, 17.9, 23.4, 16.1)
y <- c(2.1, 3.1, 4.9, 2.9)
chart <- rbind(x, y)
barplot(chart,
  main='연도별 영업부 매출',
  xlab='영업부', ylab='매출(억원)',
  ylim=c(0, 50),
  col=c('Orange', 'Dodger Blue'),
  legend.text=c('2022년', '2023년 4월 현재')
)
# col=c('Orange', 'Dodger Blue') 연도별로 색을 다르게 지정하기
# legend.text=c('2022년', '2023년 4월 현재') 범례
```



# 연도별 자료를 각각의 막대로 표현하기

```
barplot(chart,
  main='연도별 영업부 매출',
  xlab='영업부', ylab='매출(억원)',
  ylim=c(0, 50),
  col=c('Orange', 'Dodger Blue'),
  legend.text=c('2022년', '2023년 4월 현재'),
  beside=TRUE
)
```



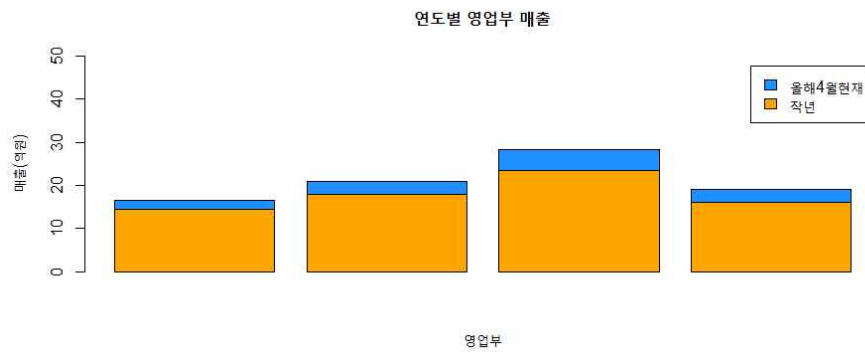
## 참고

```
barplot(chart,
  main='연도별 영업부 매출',
  xlab='영업부', ylab='매출(억원)',
  ylim=c(0, 50),
  col=c('Orange', 'Dodger Blue'),
  legend.text=T
)
```

```
작년 <- c(14.5, 17.9, 23.4, 16.1)
올해4월현재 <- c(2.1, 3.1, 4.9, 2.9)
chart <- rbind(작년, 올해4월현재)
```

```
barplot(chart,
  main='연도별 영업부 매출',
  xlab='영업부', ylab='매출(억원)',
  ylim=c(0, 50),
  col=c('Orange', 'Dodger Blue'),
  legend.text=T
)
```





[실습과제] '21년도 연령별 스토킹 피해 경험'을 막대그래프로 작성하자.<sup>1)</sup>

연령	있음(%)	없음(%)
19~29세	4.1	95.9
30~39세	3.1	96.9
40~49세	3.2	96.8
50~59세	1.9	98.1
60세 이상	1.4	98.6

1)자료출처 KOSIS [https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT\\_2KAAG01](https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_2KAAG01)

#### 4. 원형그래프(Pie Chart)

원형그래프란 먼저 원을 그리고 이 원을 상대도수에 비례하여 중심각을 나누어 마치 Pie 를 조각으로 나눈 것과 같은 형태의 그림이다. 즉, 원 안에 데이터값이 차지하는 비율을 넓이로 나타낸 그래프이다.

```
data( )      # R에 내장되어 있는 데이터들의 목록을 확인하기
data(Cars93)  # R에서 기본 제공하는 내장 데이터 가져오는 방법
str(Cars93)
table(Cars93$Type)
```

이 예시에 대한 Pie Chart는 `pie( )` 함수를 사용한다.

```
pie(table(Cars93$Type))
```

```
table(Cars93$Cylinders)
x <- table(Cars93$Cylinders)
pie(x, main='실린더 수', col=rainbow(length(x)),
    paste(round(x/sum(x)*100), '%'), radius = 1.2 )
```

# radius = 파이의 크기, 0.8(기본값)

#### ■ 3차원 원그래프 작성하기

```
install.packages('plotrix')          # 3차원 pie chart
library(plotrix)
pie3D(x, main='실린더 수', explode=0.2) # explode 속성으로 조각을 분리 가능 / 파
이 간 간격
```