



고급시각화 분석



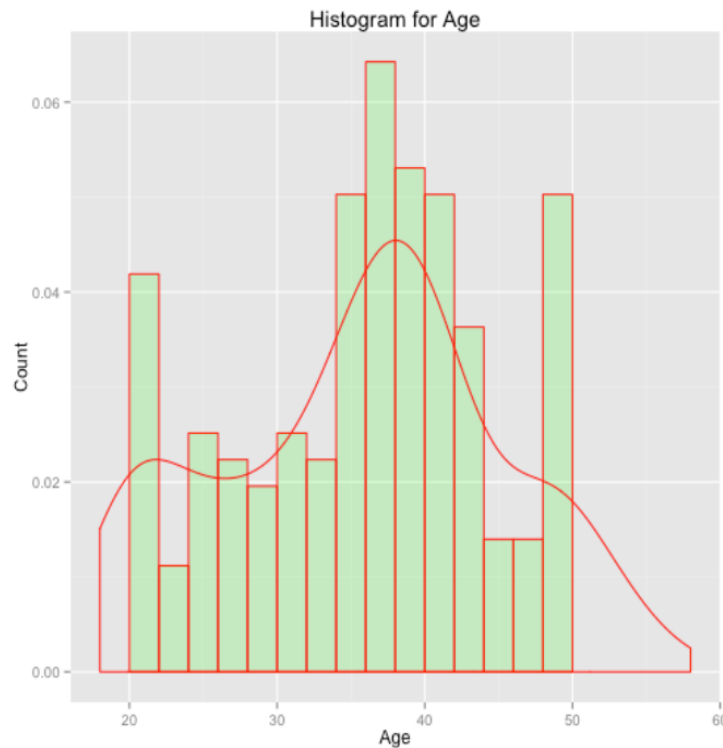


주요 패키지 :

- **graphics**
 - 막대차트, 파이차트, 산점도, 히스토 그램
 - **lattice(격자)**
 - 서로 상관 있는 확률적 반응 변수의 시각화
 - 범주 별로 독립된 패널을 격자처럼 배치
 - **ggplot2**
 - 기하학적 객체에 미적 특징을 적용하여 시각화
 - 그래프와 사용 간에 상호작용 기능 제공
 - **ggmap**
 - 지도를 기반으로 위치, 영역, 시간과 공간에 따른 차이 변화
-

다양한 시각화 도구:

- `histogram(param1, data=데이터셋)`



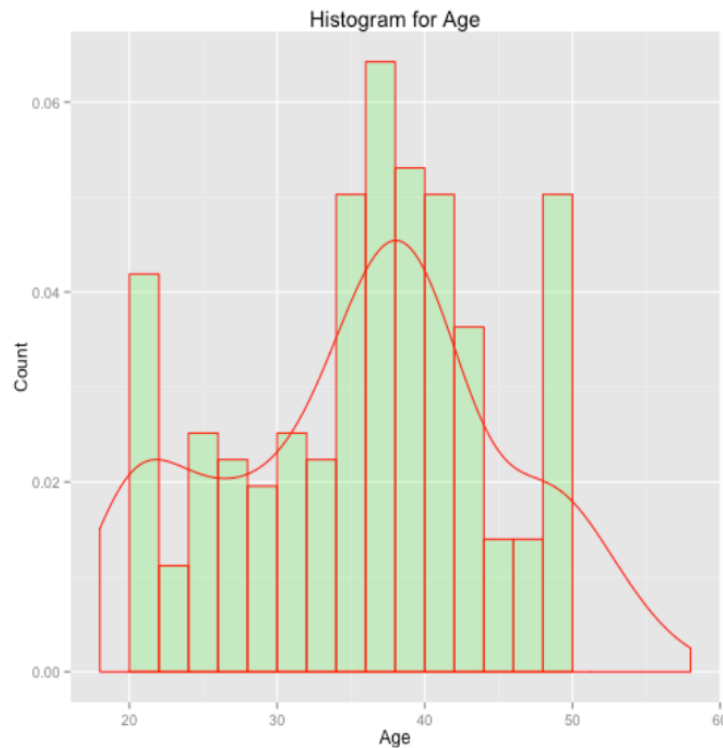


- 데이터 셋 가져오기
 - data(VADeaths)
 - View(VADeaths)
 - str(VADeaths) #
 - mode(VADeaths)
 - class(VADeaths)
 - 데이터 리모델링(함수처리를 위해)
 - matrix --> data.frame 변환
 - df <- as.data.frame(VADeaths)
 - class(df); View(df)
 - matrix --> data.table 변환
 - dft <- as.data.frame.table(VADeaths)
 - class(dft); View(dft)
-



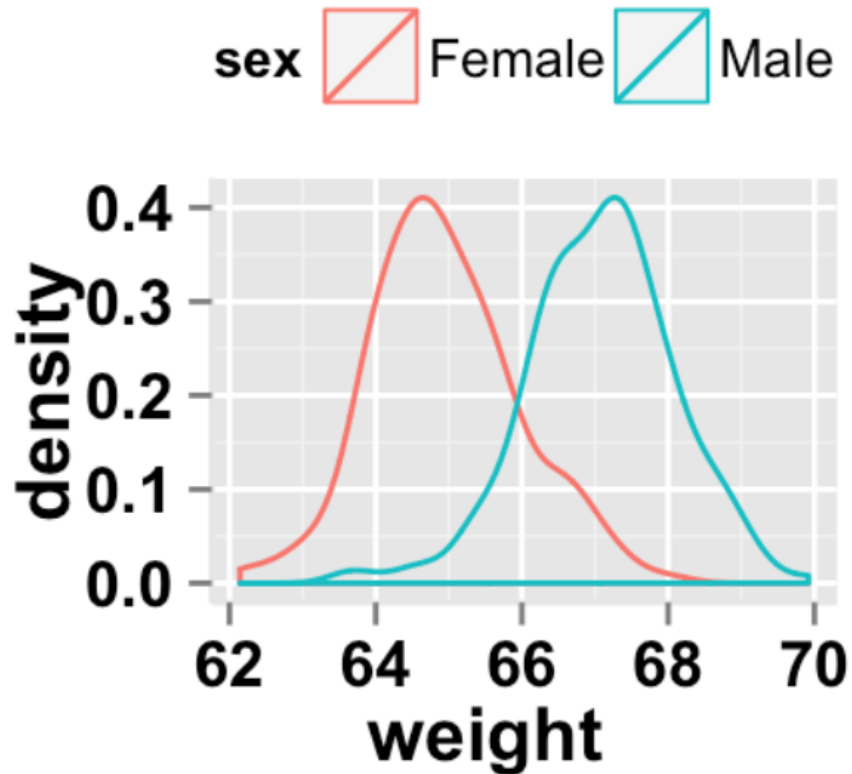
다양한 시각화 도구:

- `histogram(param1, data=데이터셋)`



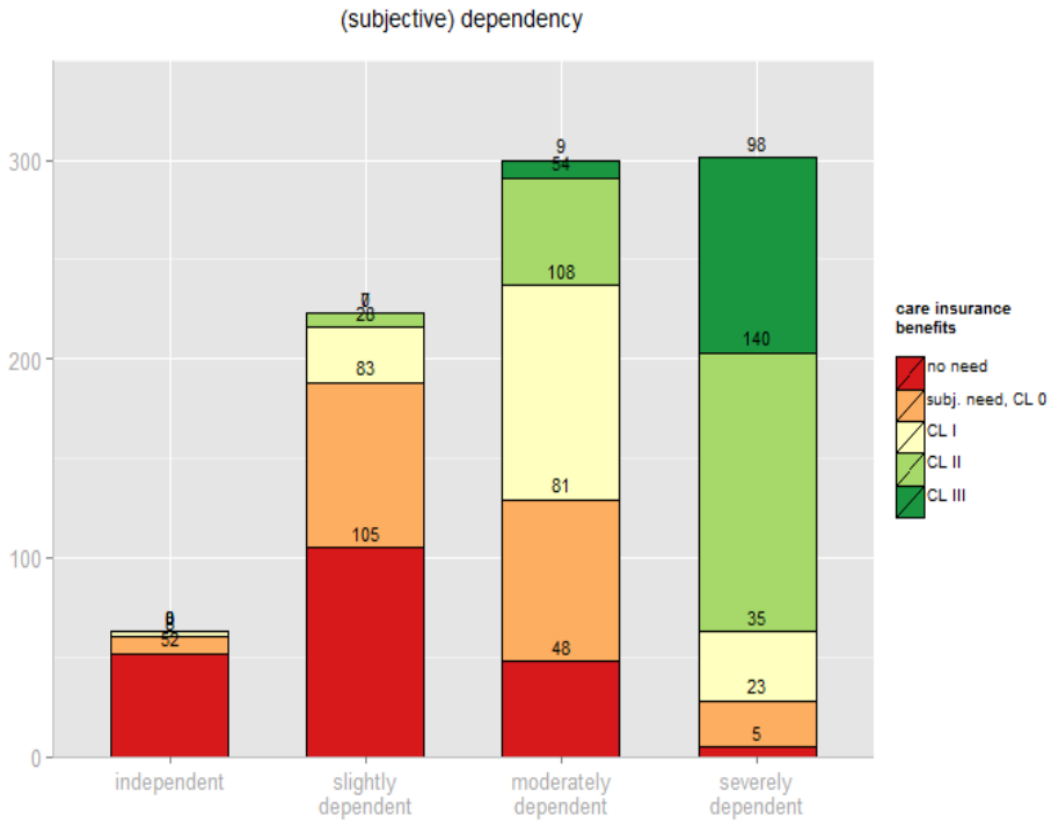
다양한 시각화 도구:

- `densityplot()`



다양한 시각화 도구 :

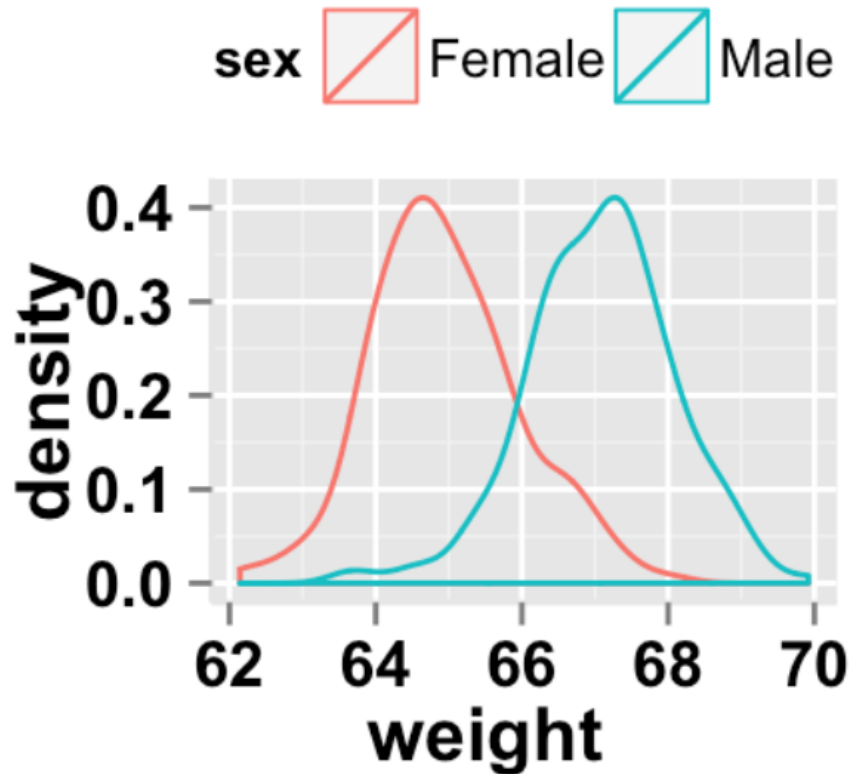
■ barchart()



- **barchart** : 막대 그래프
 - 형식)
 - `barchart(y-x축 | 변수조건, dataframe, layout)`
 - `install.packages("lattice")`
 - `library(lattice)`
 - `barchart(Var1~Freq| Var2, data=dft, layout=c(4,1))`

다양한 시각화 도구:

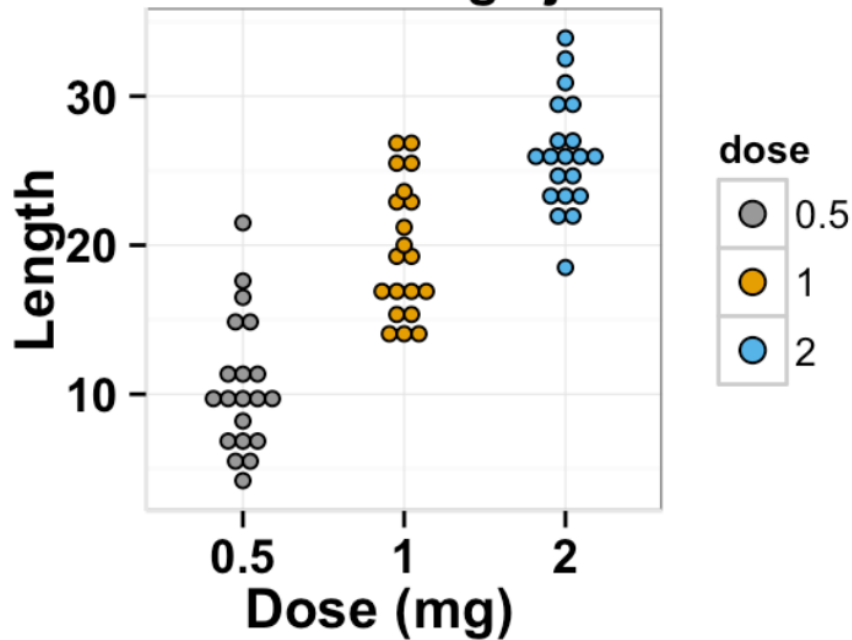
- `densityplot()`



다양한 시각화 도구 :

- `dotplot()`

Plot of teeth length according
to vitamin C/Orange juice dose

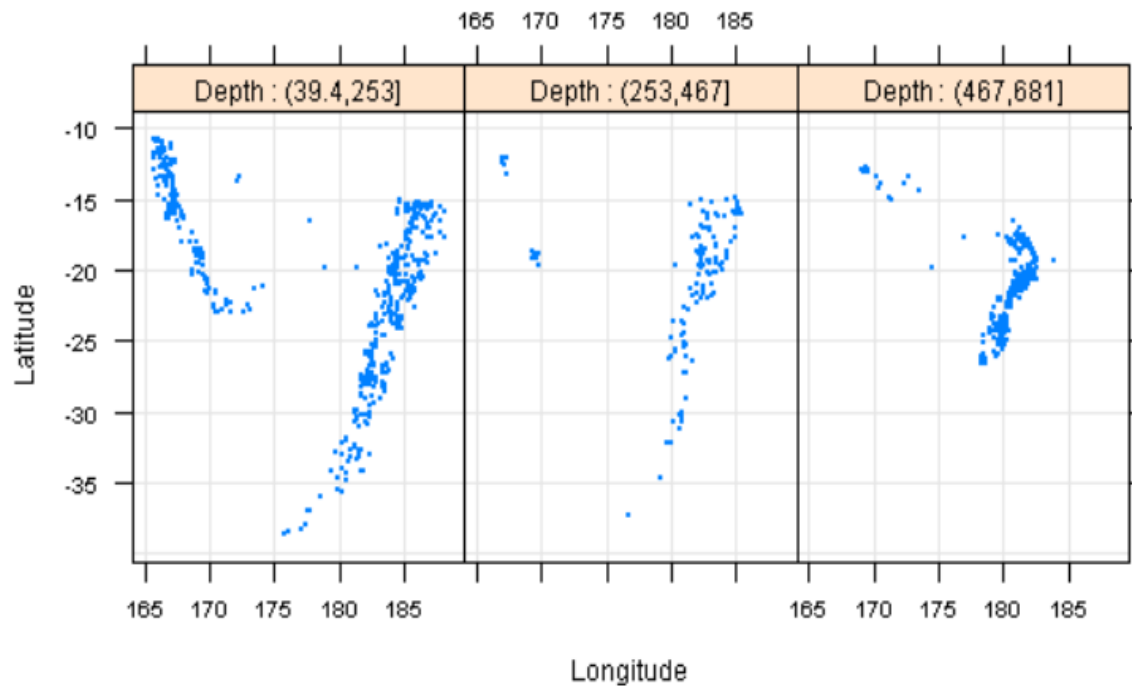




- 형식
 - `dotplot(y~x축|조건, dataframe, layout)`
 - `dotplot(Var1~Freq|Var2, dft)`
- Var2변수 단위(그룹화)로 점을 연결 하여 플로팅
 - `dotplot(Var1~Freq, data=dft, groups=Var2, type="o", auto.key=list(space="right", points=T, lines=T))`

다양한 시각화 도구 :

- xyplot()





■ 형식

- `xyplot(y축~x축 | 조건변수, dataframe or list)`
 - 대기오염..
 - `library(datasets)`
 - `str(airquality); View(airquality)`
 - `airquality`의 `Ozone(y)`, `Wind(x)` 산점도 플로팅
 - `xyplot(Ozone~Wind, data=airquality)`
 - `Month`변수 단위로 플로팅
 - `xyplot(Ozone~Wind|Month, data=airquality)`
 - `xyplot(Ozone~Wind|Month, data=airquality, layout=c(5,1))`
 - `xyplot(Ozone~Wind|factor(Month), data=airquality, layout=c(5,1))`
-



기본예제 :

- 추가 옵션..
 - pch : 점의 모양
 - 속성을 추가 처리..
 - 신규변수지정 <- xyplot(.....)
 - **update**(신규변수지정, 추가속성(속성="속성값"))
 - 예제
 - head(quakes) # quakes 데이터 확인
 - xyplot(lat~long, data=quakes, pch=".")
 - 추가변수 필요
 - **tplot** <-xyplot(lat~long, data=quakes, pch=".")
 - 그래프에 제목 추가
 - **tplot2** <- **update**(**tplot**, main="1964년 이후 태평양에서 발생한 지진위치")

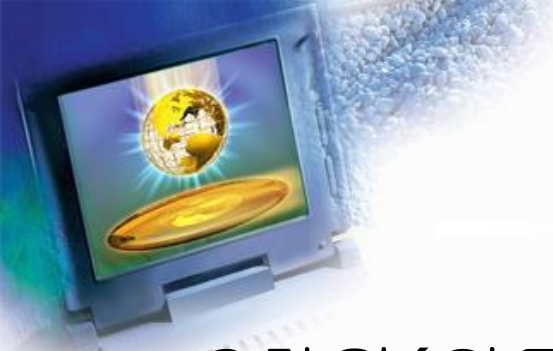
확인예제 :

- quakes를 기준으로 mag, stations을 x,y축으로 xyplot() 처리하고, 조건변수(depth를 4구간으로 나누어서)를 추가하여 화면레이아웃(2행2열)에 플로팅하세요..
 - `xyplot(mag~stations|factor(depth%%6), data=quakes, pch=".", layout(3,2))`
 - `max(quakes$depth)` 680, `min(quakes$depth)` 40, 한구간당 $640/4 = 160$
 - `quakes$part[quakes$dept >= 40 && quakes$dept < 200] = "1구간"`
 - `quakes$part[quakes$dept >= 200 && quakes$dept < 360] = "2구간"`
 - `xyplot(stations~mag|factor(part), data=quakes, pch="o", layout=c(2,2))`



equal.count() :

- 개념
 - 지정된 범위 대상 영역구분과 카운팅
 - 형식 : `equal.count(data, number, overlap)`
 - ex) 지진의 깊이를 3영역으로 구분하여 카운팅
 - `depthgroup <- equal.count(quakes$depth, number=3, overlap=0)`
 - `depthgroup` 변수 기준으로 플로팅
 - `xyplot(lat~long|depthgroup, data=quakes, main="Fiji Earthquakes(depthgroup)", ylab="latitude", xlab="longitude", pch="@", col="red")`
-



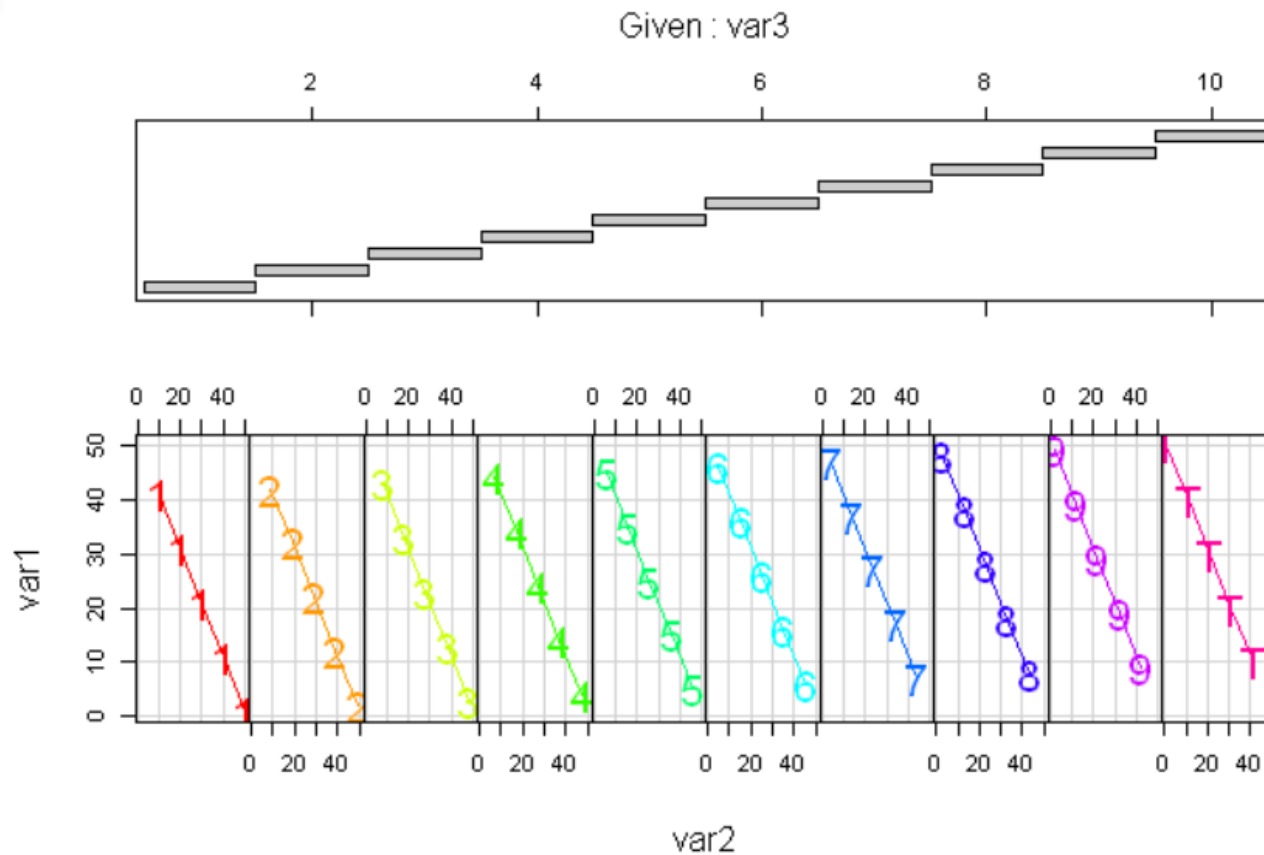
- 3차원(위도, 경도, 깊이) 산점도 그래프
 - `cloud(1속성~2속성*3속성, data=프레임,list, zlim=3속성이 나타날범위지정, xlab="1속성레이블", ylab="2속성레이블", zlab="3속성레이블")`
 - `cloud(depth~lat*long, data=quakes, zlim=rev(range(quakes$depth)), xlab="경도", ylab="위도", zlab="깊이")`

- student.csv에 국어,영어, 수학 를 기준으로 cloud처리하세요..
 - `student<-read.csv("student01.csv",header = T)`
 - `names(student)<-c("grade","part","no","name","kor","eng","math")`
 - `cloud(kor~eng*math, data=student, zlim=rev(range(student$kor)), xlab="영어점수",ylab="수학점수", zlab="국어점수", pch="o")`



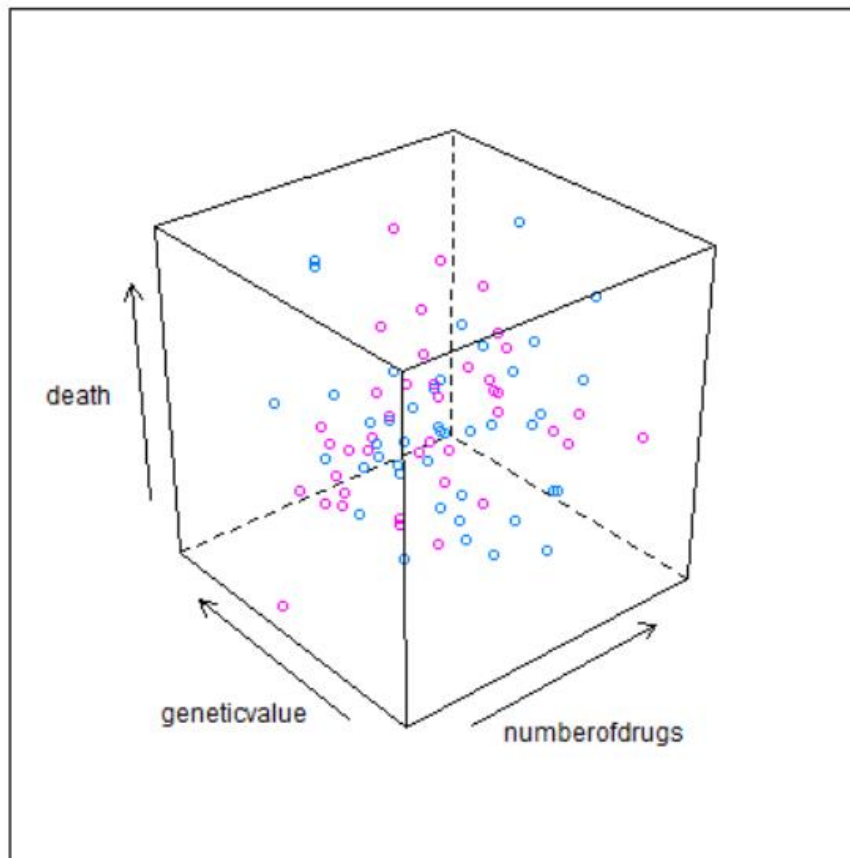
다양한 시각화 도구 :

■ coplot



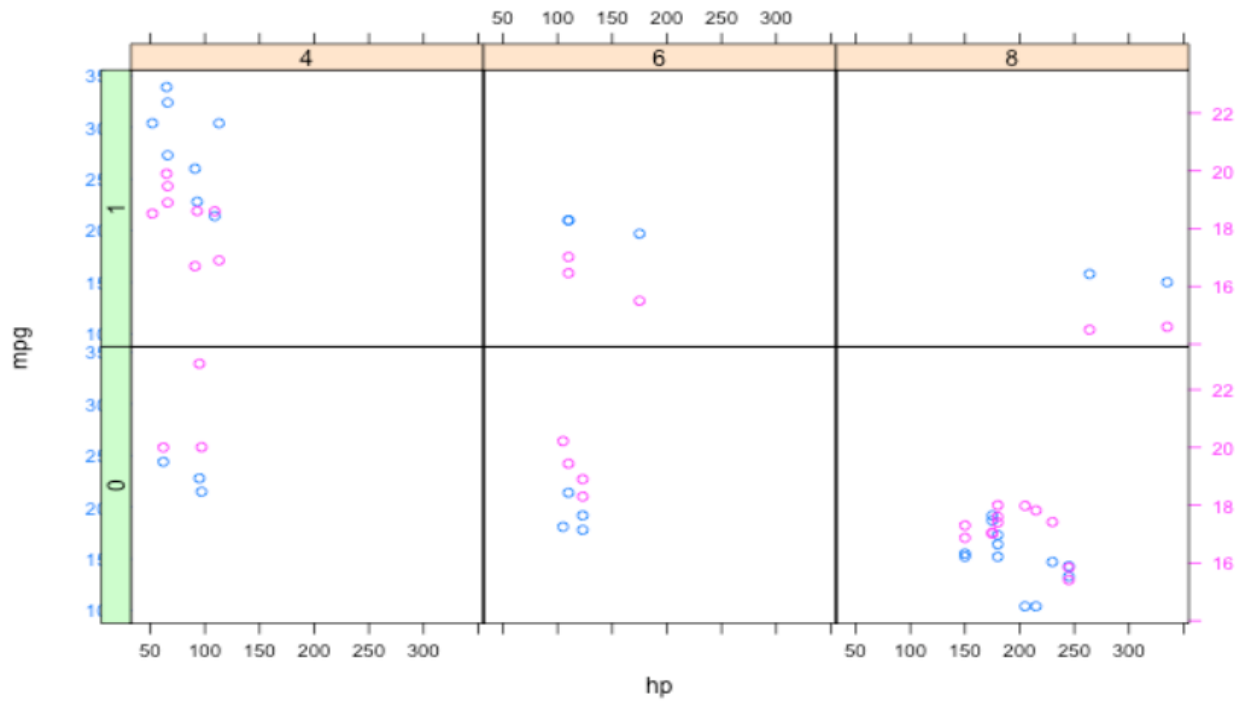
다양한 시각화 도구:

- cloud



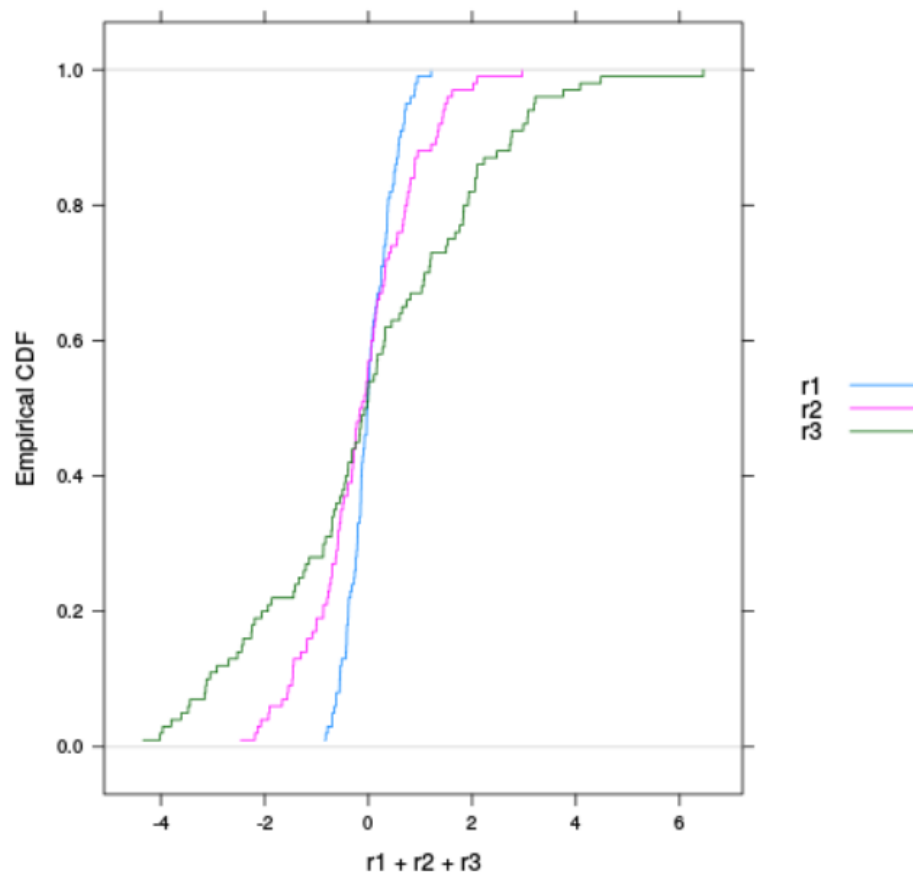
다양한 시각화 도구:

- doubleYScale



다양한 시각화 도구:

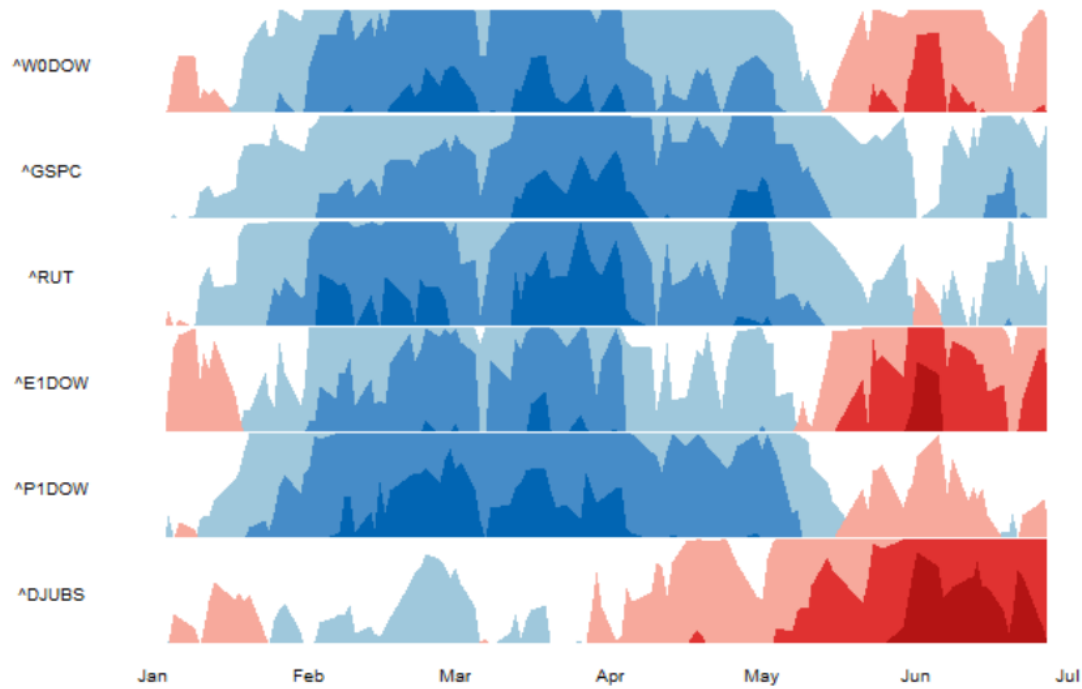
- ecdfplot



다양한 시각화 도구 :

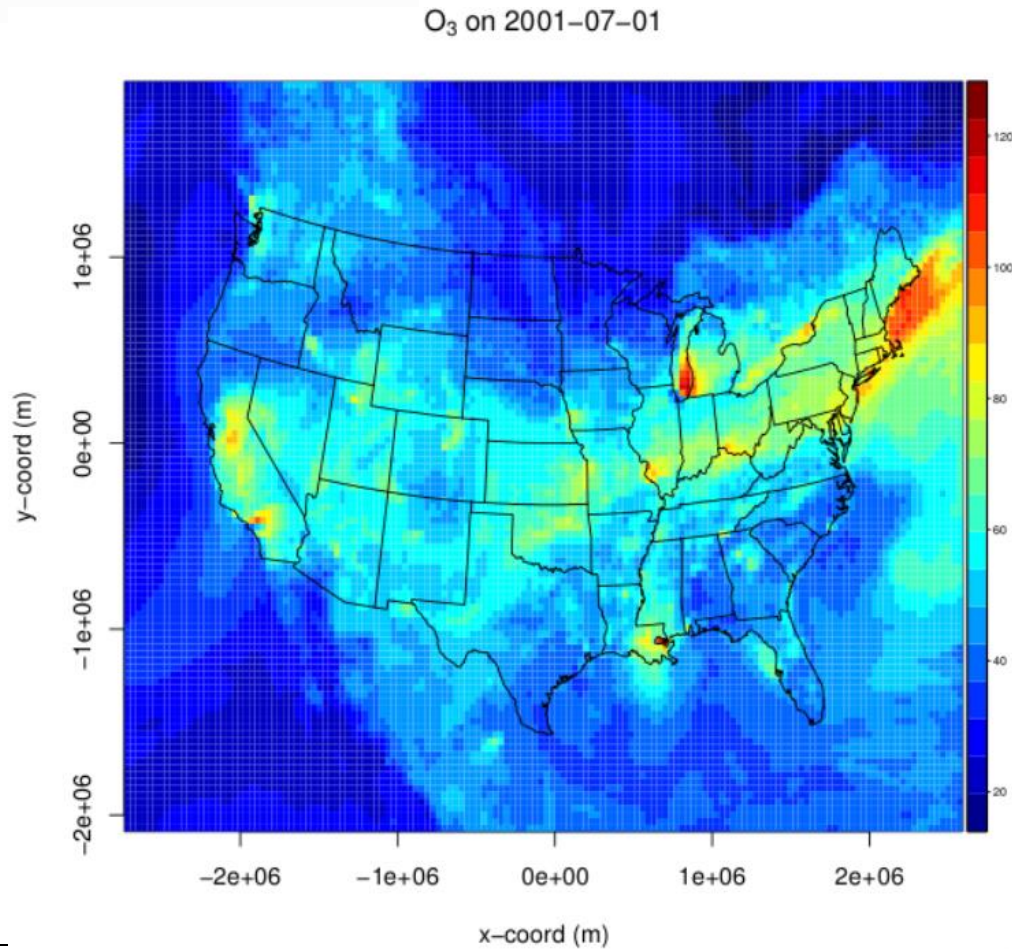
- horizonplot

World Indexes Change Since 2011



다양한 시각화 도구:

- mapplot





ggplot 패키지 :

- 필요한 패키지 설치..
 - `install.package("ggplot2")`
 - `library(ggplot2)` # 메모리로딩
 - ggplot2 패키지 제공 데이터 셋
 - `data(diamonds)`
 - `data(mtcars)`
 - `data(mpg)`
 - `str(mpg)` # map 데이터셋 구조 보기..
 - `summary(mpg)` # 요약 통계량
 - `table(mpg$drv)` # 구동방식 빈도수
-



qplot() 함수 :

- 1개 변수 대상 : 속이 짝찬 막대 모양의 세로막대 그래프
 - `qplot(hwy, data=mpg)`
- fill 옵션 : hwy 변수를 대상을 drv변수에 색 채우기(누적 막대그래프)
 - `qplot(hwy, data=mpg, fill=drv)`
- binwidth 옵션 : 막대 폭 지정 옵션
 - `qplot(hwy, data=mpg, fill=drv, binwidth=2)`
- facets 옵션 : drv변수 값으로 컬럼단위와 행단위로 패널 생성
 - 컬럼단위 패널 생성 `facets=~기준변수`
 - `qplot(hwy, data=mpg, fill=drv, facets=~ drv, binwidth=2)`
 - 행단위 패널 생성 `facets=기준변수~.`
 - `qplot(hwy, data=mpg, fill=drv, facets=drv~., binwidth=2)`



qplot() 함수 :

- 2변수 대상 기본 - 속이 꽉찬 점 모양과 점의 크기는 1를 갖는 산점도 그래프
 - `qplot(displ, hwy, data=mpg)`
 - mpg데이터셋의 `displ`(엔진크기)과 `hwy` 변수 이용
 - `qplot(displ, hwy, data=mpg, color=drv)`



qplot() 함수 :

- 색상, 크기, 모양 적용
 - ggplot2 패키지에서 제공..
 - `str(mtcars); view(mtcars)`
 - `qplot()`
 - color: 색상 `factor(carb)`
 - size : 크기 `size=qsec`
 - shape : 모양 `shape=factor(cyl)`
 - `qplot(wt, mpg, data=mtcars, size=qsec, color=factor(carb), shape=factor(cyl))`
-



qplot() 함수 확인예제 :

- qplot함수를 이용해서, student 데이터셋, 두개의 변수 (국어, 영어)
 - 크기 : 총계 조절
 - 색상 : 학년
 - 모양 : 반
 - `qplot(kor, eng, data=student, size=kor+eng+math, color=factor(grade), shape=factor(part))`
-



qplot() 함수 :

- diamonds 데이터셋
 - price : 가격, carat: 무게, cut : 품질, color: 색상의 품질, clarity : 선명도, x:길이, y:폭
- 여러옵션..
 - geom : 포인트(point), 평활(smooth), line, freqploy를 통해서 모양, 막대그래프(bar)
 - 한가지, 또는 벡터로 두가지를 혼합처리 c("point", "smooth")
 - fill : 레이아웃 색 cut 변수로..
- `qplot(wt, mpg, data=mtcars, size=qsec)`
- `qplot(wt, mpg, data=mtcars, geom=c("point", "smooth"))`
- `qplot(mpg, wt, data=mtcars, color=factor(cyle), geom="point") + geom_line()`



qplot() 확인예제 :

- student으로 qplot(), 국어, 영어점수 기준으로 평활, 점선옵션으로 화면을 출력하세요(색상은 반별)..
 - `qplot(kor, eng, data=student, geom=c("point","smooth"), color=factor(part))`



ggplot() 함수 :

- `aes(x축변수, y축변수, color색상)` 함수 ==> `ggplot()` 함수에서 **미적 요소** 를 처리할 내용
 - `+(연산자)` : 기존 `ggplot()`의 변수에서 추가할 옵션을 설정.
 - 기존변수 + `geom_point()/geom_line()/geom_step()` <point, line, step 추가>
 - `p<-ggplot(diamonds, aes(carat, price, color=cut))`
 - `p+ geom_point()` # point 추가
-



ggplot() :

- ggsave() 함수 : 디스크에 plot 이미지를 저장..
 - ggplot() 생성..
 - ggsave(file="절대경로/XXX.pdf" , dpi=해상도)
 - 가장 **최근** 그래프 저장!!!!
 - ex)
ggsave(file="c:/Rwork/output/damond_price.jpg", dpi=72)



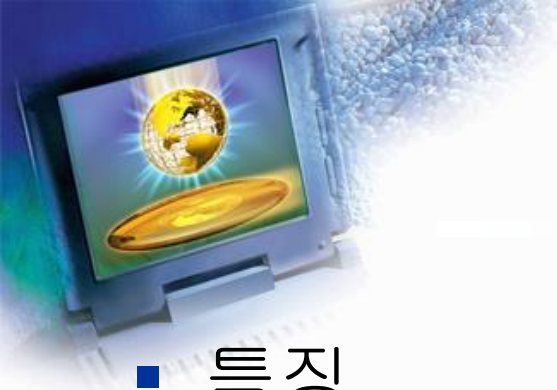
ggmap 패키지 :

- 지도 관련 패키지 설치..
 - `library(ggplot2)`
 - `install.packages("ggmap")`
 - ggmap과 ggplot2(우선설치) 관련 패키지
 - `library(ggmap)`
 - 1. `get_gogglemap()` 함수
 - 지도위치정보 가져오기
 - `gc <- geocode("seoul, korea", source="google")` # geolocation API 이용
 - `center <- as.numeric(gc)` ## 위도 경도
-



ggmap 패키지 :

- 지도 정보 생성하기..
 - `map <- get_googlemap(center= 위치정보, language="한글 encoding 방식", color="색상 bw", scale="해상도 2(1280*1280픽셀)`
 - `map <- get_googlemap(center = center, language="ko-KR", color="bw", scale=2)`



공간시각화!! :

■ 특징

- 지도를 기반으로 하기 때문에 위치, 영역, 시간과 공간에 따른 차이 및 변화에 대한 것
 - 위치 : 위도 및 경도, 지도에 버블로 표현
 - 영역 : 데이터에 따른 색상으로 표현
 - 시.공간 : 레이어 형태로 추가하여 시각화
-



google 지도 불러오기 :

- `library(ggplot2)`
 - `library(ggmap)`
 - 위치정보를 변수로 할당
 - `map <- get_googlemap("Jeonju", zoom=14, maptype="satellite", scale=2)`
 - 지역과 `zoom` 정도를 변경해서 확인하세요
 - 화면으로 플로팅
 - 장치 허용크기 표시(`extent="device"`)
 - `ggmap(map, size=c(600,600), extent="device")`
-



데이터를 통한 공간시각화 :

- 레이어 기법
 - 위치 정보 데이터 가져오기..
 - 맵을 플로딩
 - 위치 정보와 맵을 연결(point)
 - 위치 정보의 **text**와 해당 맵에서 표현하기
-



데이터를 통한 공간시각화 :

- 레이어 기법
 - 위치 정보 데이터 가져오기..
 - 맵을 플로딩
 - 위치 정보와 맵을 연결(point)
 - 위치 정보의 **text**와 해당 맵에서 표현하기
-



레이어 기법 :

- 데이터 가져오기
 - `loc <- read.csv("c:/Rwork/Part-III/seouloffice.csv", header=T)`
 - loc 구청명, 위도LON, 경도LAT
 - `View(loc)`
- 지도를 플로팅하기..
 - `get_map("중심지역", 확대비율, 지도유형)`
 - `kor <- get_map("seoul", zoom=11, maptype="roadmap")`
 - maptype: roadmap, satellite, terrain, hybrid
 - `ggmap(kor) #실제 맵에 플로팅`



레이어 기법 :

- 레이어1 : 지도 --> 레이어2 : 지도 위에 포인트
 - `kor.map <- ggmap(kor) + geom_point(data=loc, aes(x=LON,y=LAT), size=3)`
 - 지도 플로팅 + 포인트(◆) 추가
 - `geom_point(data=데이터셋, aes(x=X축위치 LON이라는 변수의 데이터값`
- 레이어2 --> 레이어3 : 지도의 포인트 위에 텍스트(구청명) 표시
 - `kor.map + geom_text(data=loc, aes(x=LON, y=LAT+0.01, label=구청명), size=3)`
 - 레이어2 + `geom_text(data=데이터셋, aes(x=데이터셋변수1, y=데이터셋변수2_포인트상단표시를 위해+@@@, label=데이터셋변수3, size=크기)`

- Part-II\university.csv
 - 대학교관련 위치정보 loading 후, 변수명 변경
학교명 --> name, LAT->xpt, LON->ypt
 - 해당 위치를 구글맵으로 표시하고, label도 표시



다양한 지도 유형 :

- 파일을 통한 데이터 로딩
 - #2015년도 6월 기준 대한민국 인구수.
 - `pop <- read.csv("c:/Rwork/Part-II/population201506.csv",header=T)`
- 지정한 변수의 데이터 프레임으로 변경..
 - `region <- pop$지역명`
 - `lon <- pop$LON # 위도`
 - `lat <- pop$LAT # 경도`
 - `house <- pop$세대수`
 - vector에서 프레임 만들기..
 - `df <- data.frame(region, lon, lat, house)`



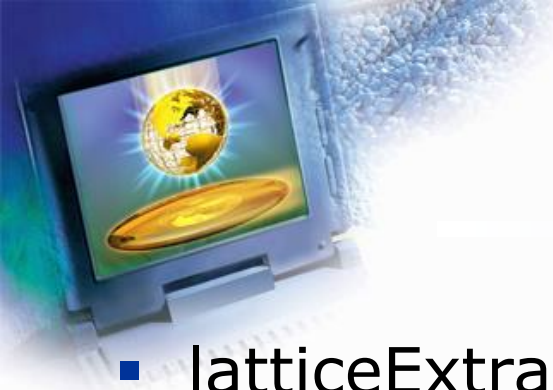
다양한 지도유형 :

- `maptype="terrain/satellite/roadmap/hybrid"`
 - `map1 <- get_map("daegu", zoom=11, maptype="terrain")`
 - `map2 <- ggmap(map1)`
 - `map3`
 - `aes` : 좌표 `x=lon`, `y=lat`, `colour=house`, `size=house`
 - `map3 <- map2 + geom_point(aes(x=lon,y=lat,colour=house, size=house),data=df)`
 - `map3 + geom_text(data=df, aes(x=lon+0.01, y=lat+0.18, label=region), size=3)`
 - 밀도 곡선 처리..
 - `map3 + geom_density2d()`

- 다음 조건에 맞게 **quakes** 데이터 셋의 수심(**depth**)과 리히터규모(**mag**)가 동일한 패널에 지진의 발생지를 산점도로 시각화 하시오. **View(quakes)**
 - lat:위도, long:경도, depth:수심, mag(리히트규모), stations(관측소)
 - 수심 3개 영역으로 범주화
 - 리히터규모 2개 영역으로 범주화
 - 수심과 리히터규모가 3행 2열 구조의 패널로 산점도 그래프 그리기 **xyplot(범위1~범위2|조건, ...)**
 - **lattice** 패키지의 **equal.count()**와 **xyplot()** 함수 이용



- 수심 3개 영역으로 범주화
 - 수심 : `quakes$depth`
 - `depthgroup <- equal.count(quakes$depth, number=3, overlap=0)`
 - `depthgroup`
- 리히터규모 2개 영역으로 범주화
 - `quakes$mag`
 - `magnitudegroup<-equal.count(quakes$mag, number=2, overlap=0)`
- 수심과 리히터규모가 3행 2열 구조의 패널로 산점도 그래프 그리기
 - `xyplot(lat~long | magnitudegroup* depthgroup, data=quakes, main="지진", ylab="latitude", xlab="longitude", pch="@", col=c("red","blue"))`



- **latticeExtra** 패키지에서 제공되는 **SeatacWeather** 데이터 셋에서 월 별로 최저기온과 최고기온을 **선 그래프** 로 플로팅 하시오.
 - **lattice** 패키지 **xylot()** 함수 이용..
 - **type="l" ==> 엘(line)**
 - **options(repos = c(CRAN = "https://cran.ma.imperial.ac.uk/", + CRANextra = "https://mirrors.ebi.ac.uk/CRAN/"))**
 - **install.packages("latticeExtra")**
 - **library(latticeExtra)**
 - **View(SeatacWeather)**
 - **xplot(min.temp + max.temp~day|month, data=SeatacWeather, type="l", layout=c(3,1))**

- diamonds 데이터 셋을 대상으로 x축에 **carat**변수, y축에 **price**변수를 지정하고, **clarity**변수를 선 색으로 지정하여 미적 요소 맵핑 객체를 생성한 후 산점도 그래프 주변에 부드러운 곡선이 추가되도록 레이아웃을 추가하시오.
 - `qplot()` 활용으로 처리시,
 - `qplot(carat, price, data=diamonds, geom=c("point","smooth"),color=clarity)`
 - `ggplot()` 활용
 - `show01<-ggplot(diamonds, aes(carat,price,color=clarity))`
 - `show01<-show01+geom_point()`

연습예제 :





이 장이 끝나면 :

정리:





참고자료 :



감사합니다
