**# plot() 함수로 색깔 점 찍기**

**plot(0,0, pch=16, cex=9, col='black')**

0,0 = x,y축 좌표

pch = 점의 종류

cex = 점의 크기

col = 색상

**# rgb( ) 함수**

**rgb( 0/255, 70/255, 42/255)**

red, green, blue를 의미하며 각 자리의 숫자는 해당 색의 강도를 의미하고 255가 100% 기준으로 100을 의미함.

**"#RRGGBB" HEX코드 활용**

**plot(0,0, pch=16, cex=10, col='#00462A')**

col 부분의 '#00462A'는 HEX코드 http://colorbrewer2.org/ 를 통해서 확인가능하고, 상황에

맞게 어떤 코드가 색 배합이 적합한지 확인가능함.

색상이름 참고

http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

**# RColorBrewer 패키지의 활용**

1) install.packages('RColorBrewer') : 패키지 설치

2) library(RColorBrewer) : 패키지 열기

display.brewer.all() : 패키지 내 모든 색상조합 확인

brewer.pal(9, 'Set1') : brewer.pal에서 ‘Set1’이라는 팔레트를 선택 후 20개이면 순서대로 9개를 뽑아온다.

* Brewer. Pal의 종류를 모아놓은 사이트 리서치 필요

**## ggplot의 강점**

ggplot만 알아도 시각화는 해결이 가능하다.

R의 기본 내장함수인 plot, barchart, histogram보다 훨씬 짜임새가 있음으로 package를 자주사용한다.

코드는 어렵지만, 한줄한줄 직독직해하면 전부 가능하다.

**## ggplot2 패키지를 활용한 시각화**

1) ggplot2 패키지 설치, 불러오기

1.1 install.packages('ggplot2')

1.2 library(ggplot2)

\* ggplot 이 아닌 ggplot2로 기재

2) 데이터 요약/처리를 위한 패키지도 불러오기

2.1 library(dplyr)

2.2 library(tidyr)

**## gapminder 패키지**

학습용 데이터를 담아놓은 패키지

**## ggplot 이용하여 차트 작성 형식**

ggplot(data1) +

aes(x = gdpPercap) + **#x축 지정**

aes(y = lifeExp) + **#y축 지정**

geom\_point() + **#나타낼 그림**

aes(color = continent) **#색 지정**

* 변수 기반 색을 칠하는게 아닐 경우에는 aes를 사용하지 않는다.

1) data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color=continent)) + geom\_point()

2) data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp)) + geom\_point(aes(color=continent))

3) data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp)) + geom\_point(color = "red")

1), 2)은 변수 기준으로 묶은 수식이 2개로 나눠짐 (결과는 동일)

3) 변수기준으로 묶인게 아님으로 변수 기반 색을 칠하는게 아닌 결과가 나온다.

**## 불가능한 예시들**

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = "red")) + geom\_point()

-> 한 변수에 따른 색으로 채워줘

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, fill = continent)) + geom\_point()

- 산점도는 color를 쓰고, bar plot 이나 box plot은 fill을 쓴다. >

## 모양 지정

ggplot(data1) +

aes(x = gdpPercap) + **#x축 지정**

aes(y = lifeExp) + **#y축 지정**

geom\_point() + **#나타낼 그림**

aes(color = continent) + **#색 지정**

aes(shape = continent) **#모양 지정**

**같은표현**

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent)) + geom\_point()

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent)) + geom\_point(aes(shape = continent))

특정모양 지정

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent)) + geom\_point(shape = 10)

* Shape=10이라는 조건을 기재하면 변수 기준으로 묶는게 아님을 인지 (중요!)

shape 변경을 통해 모양 변경가능 (변수 할당하면 기본 shape로 적용)

원래는 shape=continent면 continent 내의 값에 따라 다른 모양이 나오는데, shape=3이게 되면 하나의 모양으로 continent 내의 값을 모두 적용한다.

## 투명도 지정

ggplot(data1) +

aes(x = gdpPercap) + **#x축 지정**

aes(y = lifeExp) + **#y축 지정**

geom\_point() + **#나타낼 그림**

aes(color = continent) + **#색 지정**

aes(shape = continent) + **#모양 지정**

aes(size = pop) + **#크기 지정**

aes(alpha = lifeExp) **#투명도**

* **CSSA :** geom\_point() 뒤에 기본 요소들

같은표현

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent, size = pop, alpha = lifeExp)) + geom\_point()

특정 크기 지정

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent, size = pop)) + geom\_point(alpha = 0.5)

## 연습문제 1.2

insurance %>% ggplot(aes(x=age, y=charges, color=bmi, shape=smoker)) + geom\_point()

insurance %>% ggplot(aes(x=age, y=charges, color=bmi, shape=bmi)) + geom\_point()

* 모양 부분이 범주형이 아닌 연속형의 데이터이면 에러가 난다. (

**## 막대그래프**

ggplot(data1) +

aes(x = continent) + **# x축 지정**

geom\_bar() + **# 막대그래프 그리기**

aes(fill = continent) + **# 전체색**

scale\_fill\_brewer(palette='Set1') **#팔레트 사용하기**

* **팔레트 사용하게 되면 막대그래프 색상 배치가 더욱 전문적이게 된다.**

**## 막대그래프 주의사항**

ggplot(data1) +

aes(x = continent) + **# x축 지정**

geom\_bar() + **# 막대그래프 그리기**

aes(color = continent) **# 개별색**

* 막대그래프는 color이 아닌 fill로 사용! (주의)

결과적으로는 bar의 겉 테두리만 색칠됨.

## 막대그래프에서 x와 y를 모두 지정해주고 싶으면?

geom\_bar(stat = "identity") 사용하기 > 이건 barplot일 경우에만 사용이 가능하다. (인터넷으로 다시 체크해볼 필요가 있음)

ggplot(data1) +

aes(x = continent) + **# x축 지정**

aes(y = lifeExp) + **# y축 지정**

geom\_bar(stat = "identity") + **# 막대그래프 x,y축**

aes(fill = continent) # **전체색**

data1 %>% ggplot(aes(x = continent, y = lifeExp, fill=continent)) + geom\_bar(stat = "identity")

동일한 식 : 파일에는 없지만 직접 해보는 습관을 들여야 함.

* geom\_bar(stat = "count") x축만 지정할 때는 디폴트 값
* geom\_bar(stat = "identity")는 x,y축을 둘 다 지정하면 무조건이다. 꼭꼭

## 주의 주의 또 주의

ggplot(data1) +

aes(x = continent) + # x축 지정

aes(y = lifeExp) + # y축 지정

geom\_bar(stat = "identity") + # 막대그래프 x,y축

aes(color = continent) # 전체색

**##### 데이터 전처리와 막대 차트 그리기**

**# continent 마다 평균을 그리고 싶으면??**

data1 %>%

group\_by(continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot() +

aes(x = continent) +

aes(y = mean) +

geom\_bar(stat = "identity") +

aes(fill = continent) +

aes(alpha = 0.7)

**%>% 와 + 구분이 확실해야 한다.**

**%>% 와 + 마지막에 사용 여부를 확실히 해야한다.**

테이터 원본 용량이 크면 데이터를 전처리 한 상태에서 하는게 좋다 (속도적인 방면)

## 특정변수로 구분해서 그리고 싶다면?

gapminder %>%

filter(year %in% c(2002,2007)) %>%

group\_by(continent,year) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot() +

aes(x = continent) +

aes(y = mean) +

geom\_bar(stat = "identity") +

aes(color = continent) +

aes(fill = continent) +

facet\_grid(~year)

## 전체연도 기준으로 그리고 싶다면?

gapminder %>%

group\_by(continent,year) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot() +

aes(x = continent) +

aes(y = mean) +

geom\_bar(stat = "identity") +

aes(color = continent) +

aes(fill = continent) +

facet\_grid(~year)

* group\_by(continent,year) %>% : 두 개의 변수를
* facet\_grid(~year) : 나눠서 그림을 그린다.

1)이 없는 경우에는 데이터 전체(1952~2007년)이면 (1952~2007년)에 대한 전체연도를 group\_by를 하여 진행

facet\_grid(~year)이 없으면 데이터가 누적되서 나온다 (직접해볼것)

나눌 수 잇는 기준에 대해서 year을 넣어서 나눠준다.

연습문제 2 (2개) 한번 더 풀어볼 것

## x = 범주형

gapminder %>%

ggplot(aes(x=continent, y= lifeExp, fill= continent)) +

geom\_boxplot(alpha = 0.5)

x=continent 는 범주형임으로

# 주의! 요약을 한 데이터를 사용하지 않는다!

gapminder %>%

group\_by(continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot(aes(x=continent, y= mean, fill= continent)) + geom\_boxplot()

* 요약을 한 후에 펼치는 boxplot으로 사용하면 안된다. 결과값은 줄 처럼 나오고 사용할 그림들의 특징을 확실히 파악하고 있어야 함.

gapminder %>%

ggplot(aes(x=continent, y=lifeExp, fill= continent)) + geom\_boxplot()

* boxplot은 위와 같이 사용하면 정상적으로 결과값이 나온다.

### 히스토그램

gapminder %>%

ggplot(aes(x=lifeExp)) +

geom\_histogram()

* histogram만 사용한 케이스

gapminder %>%

ggplot(aes(x=lifeExp)) +

geom\_histogram() +

facet\_grid(~continent)

* histogram + facet\_grid 같이 사용한 케이스

막대그래프를 붙인개념(히스토그램) : 확대 축소에 따라 데이터 분포가 달라보일 수가 있다.

boxplot이 더 좋긴한데 histogram이 직관적이라서 현업에서 소통할 때 가장 편리함.

**## 선 그래프**

gapminder %>%

group\_by(year) %>%

dplyr::summarise(sum = sum(lifeExp)) %>%

ggplot(aes(x=year,y=sum)) + geom\_line()

* 시간의 변화에 대한 데이터에 자주 쓰인다.

**## 여러 그룹을 그리고 싶을 경우**

gapminder %>%

group\_by(year,continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot(aes(x=year, y=mean , group=continent ,color= continent)) + geom\_line()

* line 그래프에서는 어떤 그룹기준으로 그림을 그릴꺼라는 제시가 필요하다. 그룹 설정을 꼭 해야한다.

mutate(dataframe, 새로운 column명 = 기존 columns을 조합한 수식

**## ggplot 추가**

**### 테마변경하기**

1. **Classic Theme**

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_classic()

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_classic()

1. **BW Theme**

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_bw()

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw()

**## 패키지를 통한 다양한 테마 변경**

library(ggthemes) : 라이브러리 설치

Graph + theme\_bw() + ggtitle("Theme\_bw")

theme\_bw(), theme\_classic(), theme\_dark(), theme\_light(), theme\_linedraw(), theme\_minimal(), theme\_test(), theme\_void(),

**## 범례제목 수정 (2가지)**

**형식 : labs(fill = XX)**

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_bw() +

labs(fill = "범례 제목 수정(fill)")

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw() + labs(fill = "범례 제목 수정(fill)")

**## 범례 테두리 설정**

Graph + theme(legend.position = "top")

Legend.postion = “top”, “bottom”, c(0.9, 0.7), “none”

**## 축 변경**

**이산형 :** deiscreate()

**연속형 :** continuous()

범주형(또는 이산형)으로 된 축의 항목 순서를 변경하고 싶을 때에는 scale\_x\_discrete() 또는 scale\_y\_discrete() 함수를 사용하면 된다.

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_x\_discrete(labels = c("하","중","상")) +

scale\_y\_continuous(breaks = seq(0,8000,by = 1000))

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw() + scale\_x\_discrete(labels = c("하","중","상")) + scale\_y\_continuous(breaks = seq(0,8000, by=1000))

x축은 상중하로 나눠짐

y축은 0~8000까지 1000단위로 나눠짐

**## 범례 부분 채우기 (low,medium,high) > (하, 중, 상)**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_x\_discrete(labels = c("하","중","상")) +

scale\_y\_continuous(breaks = seq(0,8000,by = 1000)) +

scale\_fill\_discrete(labels = c("하","중","상"))

* 범례 부분의 컬럼 부분을 수정한다.

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw() + scale\_x\_discrete(labels = c("하","중","상")) + scale\_y\_continuous(breaks = seq(0,8000, by=1000)) + scale\_fill\_discrete(labels = c("하","중","상"))

**## ylim**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

ylim(0,5000)

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() +ylim(0,5000)

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

ylim(0,13000)

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw() + ylim(0,13000)

## **색변경**

**색깔을 하나하나 지정을 해주고 싶을 때 쓰는 문장**

**1)**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan'))

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + theme\_bw() + scale\_fill\_manual(values = c('red', 'royalblue', 'tan'))

**2)**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary), alpha = 0.4) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan'))

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar(alpha = 0.4) + theme\_bw() + scale\_fill\_manual(values = c('red', 'royalblue', 'tan'))

**### 글자크기, 각도 수정**

**coord\_flip() : 대칭 그래프 (대칭해서 적용한다)**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary), alpha = 0.4) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan')) +

coord\_flip()

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar(alpha=0.4) + scale\_fill\_manual(values = c('red', 'royalblue', 'tan')) + coord\_flip()

**## legend.position = 'none', 포지션(범례) 이 없게 나와줘**

**## axis.text.x(y) = element\_text(), x축과 y축에 대한 수정 사항**

**## legend.text = element\_text(), 축 변수 부분 수정**

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan')) +

coord\_flip() +

theme(legend.position = 'none',

axis.text.x = element\_text(size = 15,angle = 90),

axis.text.y = element\_text(size = 15),

legend.text = element\_text(size = 15))

HR %>% ggplot(aes(x=salary, fill=salary)) + geom\_bar() + scale\_fill\_manual(values = c('red', 'royalblue', 'tan')) + coord\_flip() +

theme(legend.position = 'none', axis.text.x = element\_text(size=15, angle=90), axis.text.y = element\_text(size=15), legend.text = element\_text(size=15))

**## 그래프에 평행선, 수직선, 대각선을 그릴 수 있는 명령어**

ggplot(NULL) +

geom\_vline(xintercept = 10,

col = 'royalblue', size = 2) +

geom\_hline(yintercept = 10, linetype = 'dashed',

col = 'royalblue', size = 2) +

geom\_abline(intercept = 0, slope = 1, col = 'red',

size = 2) +

theme\_bw()

참조선 그리기

geom\_vline: 세로

geom\_hline: 가로

xintercept = 10 (10에 대한 절편)

linetype = 'dashed' (라인 유형)

intercept = 0, slope = 1

절편 : 0

기울기 : 1

**# 데이터 요약**

agg2 = insurance %>%

mutate(bmi\_grp = cut(bmi,

breaks=c(0,30,35,40,100),

labels=c('G1','G2','G3','G4'))) %>%

group\_by(region, bmi\_grp) %>%

summarise(Q90 = quantile(charges, 0.9))

**상위 10%에 대한 값을 구하고 싶다 라는 의미 (quantile = 분위수)**

**## 분위수**

quantile(iris$Sepal.Width,0.5) #중위수

quantile(iris$Sepal.Width,0.7) #70%

quantile( , q) : q\*100% 값 계산

**## 히트맵 작성방식**

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile()

**## 색상지정**

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile() +

scale\_fill\_gradient(low='white', high='#FF6600')

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile() +

scale\_fill\_distiller(palette='YlGnBu')

### 분석가 면접에 대한 기준

- 접근법 (정답은 없다. 그래서 접근법이 중요하다)

- 기술적인 것은 2순위, 과정 및 풀어서 이야기하는 방법이 굉장히 중요하다.

- 자율적으로 분석한 것에 대해서 서술해봐라

- associate / 아빠 > 기저기 > 맥주 (장바구니 분석)

- 정확도에 치중하지마라. 나는 어떻게 접근했는지가 중요하다 (컨퍼런스, 유투브에서 참고가능)

시각화 패키지 종류 정리 필요