

R 그래프를 위한 함수 및 옵션

카페 문서: 1118 번

R 은 그래프를 그리기 위한 강력한 능력을 가지고 있다.

예를 들어서 n 개의 짝으로 이루어진 자료를 x, y 평면에 점으로 나타내는 것을 산점도(scatterplot)라고 한다.

이 산점도를 얻기 위하여 R 프로그램에서 제공하는 plot 함수를 이용하여 그래프를 작성할 수 있다.


대표적인 고수준 그래픽 함수는 plot() 함수이다.

작성된 그래프를 지우는 방법은 plot.new()를 사용하면 된다.

항목	설명
plot(x, y) 또는 plot(y~x)	같은 길이를 가진 x, y에 대하여 그래프를 그려준다. x는 수평 축을 그린다.(x 축 자료) y는 수직 축을 그린다.(y 축 자료)
par()	이 함수는 그래픽 장치의 설정을 바꾸는 함수 로 사용된다.(레이아웃을 설정하는 함수) 2개 이상의 그래프를 하나의 화면에 그리고자 할때 사용한다. mfrow는 그래픽 장치에 그릴 그림의 갯수를 설정한다. mfrow는 행 방향으로 그래픽을 채워 나가겠다는 옵션이다. 예를 들어, 행의 개수는 1, 열의 개수는 2인 그림을 그리는 영역을 준비해준다. opar<- par(mfrow=c(1, 2)) figure margins too large 오류가 뜨면 마진을 조정해주는 옵션 다음 문장을 실행하면 된다. par('mar') par(mar=c(1,1,1,1)) mar=c(bottom, left, top, right) # default is c(5, 4, 4, 2) + 0.1 ex) par(mar=c(5, 4, 4, 4) + 0.1) 기본 plot 창을 보존하면서 계속 추가하여 그리고자 하는 경우 다음 코드를 사용한다. par(new=T)

plot 함수의 주요 옵션

plot() 함수와 관련된 주요 옵션은 다음과 같다.

항목	설명
axes	plot의 테두리(x축, y축)와 관련된 설정이다. axes=FALSE : 그림의 테두리를 그리지 않는다. ↔ axes= TRUE
type	그래프의 타입을 결정한다. type='p' # 작은 원(점)으로 그리기 type='l' # 실선으로 그리기 type='b' # 작은 원(점)점 선으로 그리기 type='o' # 원형과 실선 동시에 그리기 type='h' # 직선 type='s' # 짝은 선 type='n' # 아무것도 그리지 않는다(nothing)
pch	점 모양(plotting character)을 설정한다. 점의 표시 기호(모양)를 선택한다. 숫자(1~25)와 모양은 help(points) 명령으로 확인이 가능하다. 예시) pch=1 또는 pch='*' <p> pch = 19: solid circle, pch = 20: bullet (smaller solid circle, 2/3 the size of 19), pch = 21: filled circle, pch = 22: filled square, pch = 23: filled diamond, pch = 24: filled triangle point-up, pch = 25: filled triangle point down. </p> 
col	plot의 색상을 설정한다. 예시)'red', 'green' 또는 번호, rainbow(8) 8가지 무지개 색상
xlab, ylab	x축과 y축에 놓이는 caption/title 문구를 설정한다.
xlim, ylim	x축과 y축의 상한 값과 하한 값의 범위를 설정한다. xlim=c(0, 250) 또는 xlim=range(100)
sub	소제목을 그래프 아래 쪽에 표시한다.
main	plot의 메인 Title(창의 제목)을 설정한다.
cex	character expansion(문자의 팽창, 즉 점의 크기를 결정한다.

abline() 함수

abline 함수는 $y = a + bx$ 형태의 직선, 또는 $y = h$ 형태의 가로로 그은 직선, 또는 $x = v$ 형태의 세로로 그은 직선을 그래프로 그려 주는 함수이다.

참고 사항 : 두 점을 지나는 직선을 그릴 때는 segments() 함수를 사용하면 된다.

segments(1, 3, 2, 5)

항목	설명
a	$y = a + b * x$ 형태의 직선을 그릴 때 절편
b	$y = a + b * x$ 형태의 직선을 그릴 때 기울기
h	$y = h$ 형태의 수평선을 그릴 때 지정
v	$x = v$ 형태의 수직선을 그릴 때 지정
추가 항목	col(색상), lty(선의 종류), lwd(선의 두께) 등등
reg	선형 회귀 모델을 그릴 때 지정
사용 예시	<pre># 만약 x축과 y 축을 동시에 그리려면 다음과 같이 작성하면 된다. abline(h=0, v=0, col='black') # 격자 눈금은 다음과 같이 구현하면 된다. # seq(0, 10, 0.5)는 0부터 10까지 0.5씩 데이터를 만들어 준다. abline(h = seq(0, 10, 0.5), v=seq(1, 14, 1), lty=2, lwd=0.2)</pre>

axis() 함수

axis() 함수는 x축과 y축을 그리기 위한 함수이다.

이때 plot() 함수의 axes 옵션의 값이 false이어야 한다.

항목	설명
side	x 축은 숫자 1, y 축은 2 를 지정한다.
at	
label	축에 보여줄 문자열을 지정해준다.
las	las = 2 는 x 의 값을 90 회전시킨다.

barplot () 함수

barplot() 함수에서 사용 가능한 옵션은 다음과 같다.

barplot(height, width = 1, names.arg = NULL, legend.text = NULL, ...)

항목	설명
angle, density	막대를 칠하는 선분의 각도, 선분의 수를 지정한다.
beside	TRUE을 지정하면 각각의 값들을 별개의 막대 그래프로 그린다. FALSE이면, 모든 항목을 누적(stack) 시켜서 그린다.
col	색상을 지정한다. mycolor = rep(c('red', 'blue'), 4) col=mycolor
cex.names	y축 이름의 크기를 지정한다. 숫자가 클 수록 y 축의 이름이 커진다. cex.names = 0.8
horiz	TRUE을 지정하면 수평 막대 그래프 를 그린다. 기본 값은 수직 막대 그래프(FALSE)이다.
legend	오른쪽 상단에 범례를 그려 준다.
main	타이틀 제목을 지정한다.
names	각 막대의 레벨을 정하는 문자열 벡터를 지정한다. names.arg=names(somedata)
names.arg	x축에 보여지는 항목들의 이름을 지정한다.
space	각 막대 사이의 간격을 지정한다. 값이 클수록 막대의 굵기는 작아지고, 사이 간격은 넓어진다.
width	각 막대의 상대적인 폭을 벡터 형식으로 지정한다.
xlab, ylab	x축과 y축에 표시되는 문자열(라벨)을 지정한다.
ylim	y 축을 위한 눈금의 상하한선을 지정한다. 예시 : ylim=c(0, maxval + 10)

***boxplot () 함수**

상자 수염 그래프는 다섯 수치의 요약된 정보를 시각화하는 데 효과적인 그래프이다.
특히 데이터의 분포 정도와 이상치 발견을 목적으로 할 경우 유용하게 사용될 수 있다.

다소 복잡해 보이는 것처럼 보이지만, 최대/최소/평균 등의 값을 한 눈에 보기 편하게 분석해야 할 경우 아주 유용하게 사용되는 차트이다.

항목	설명
사용 목적/용도	데이터의 분포 정도를 파악하면서, 또한 이상치를 발견하고자 할 때 사용된다.
사용 예시	주식이나, 기온의 변화량, 강수량 등등

boxplot() 함수의 옵션

boxplot() 함수에서 사용 가능한 옵션은 다음과 같다.

항목	설명
col	박스 내부의 색상을 지정한다.
data	포물러가 적용될 데이터 프레임(또는 리스트)
formula	y ~ grp의 형식으로 y는 분포를 그릴 값, grp는 값들을 그룹 짓는 변수이다. y ~ grp는 y에 대한 상자 그림을 grp마다 그린다.
horizontal	TRUE이면 세로(수평)로, FALSE이면 가로로 상자 그림을 그린다. 아래부터 차례로 나열된다.
names	각 막대의 라벨을 지정할 문자 벡터를 지정한다.
notch	TRUE으로 지정할 경우 상자의 허리 부분을 가늘게 표시한다.(허리선 이라고 부른다.) 중위수를 강조하려는 의미로 사용된다. (FALSE)
range	박스의 끝에서 수염까지의 길이를 지정한다. 기본 값은 1.5이다.
width	박스의 폭을 지정한다.

boxplot() 함수의 반환되는 속성 항목

boxplot() 함수를 실행하여 변수에 반환할 수 있다.

이 반환된 변수가 가질 수 있는 속성들은 다음과 같은 항목들이 존재한다.

속성 항목	설명
stats	하한 값부터 상한 값까지의 수치 정보를 보여 준다.
n	도수이다.
conf	신뢰 구간(confidence level)을 의미한다.
out	이상치 정보를 보여 준다.
group	이상치(outlier) 정보가 들어 있는 그룹의 번호를 의미한다.
names	그룹의 이름을 보여 준다.

boxplot() 함수의 반환 되는 결과 그림

항목	설명
whisker	상자의 좌우 또는 상하로 뻗어 나가는 선
박스 내부의 가로선	median(중앙 값)을 의미한다.
IQR	Inter Quartile Range 제3사분위수(Q3)와 제1사분위수(Q1)의 값을 차이를 말한다.
lower whisker	최소 값, 'median - 1.5 * IQR'보다 큰 데이터 중에서 가장 작은 값
upper whisker	최대 값, 'median + 1.5 * IQR'보다 작은 데이터 중에서 가장 큰 값
outlier(이상치)	lower whisker보다 작은 데이터, upper whisker보다 큰 데이터를 말한다.

density () 함수

density() 함수에서 사용 가능한 옵션은 다음과 같다.

항목	설명
사용 형식	

dotchart() 함수

함수에서 사용 가능한 옵션은 다음과 같다.

항목	설명
color	색상을 지정한다.
lcolor	구분선(line) 색상(수평으로 긋는 선)을 지정한다.
labels	점에 대한 레이블을 표시한다.
추가 옵션	main, cex, pch, xlab, ylab 등등

***hist() 함수**

히스토그램은 특정 데이터의 빈도 수를 막대 모양으로 표시한 그래프이다.

도수 분포표와 히스토그램은 가장 많이 사용되는 통계 분석 도구로써, 데이터의 특성을 파악하는 역할을 한다.

도수 분포표는 주어진 데이터 구간으로 나누어 속한 데이터의 숫자들을 정리한 표이다.

이것을 그림 형태로 표현한 것이 히스토그램이다.

함수에서 사용 가능한 옵션은 다음과 같다.

함수의 반환 값은 'histgram' 객체로, 히스토그램의 다양한 통계 정보가 저장되어 있다.

항목	설명
x	히스토그램을 그릴 값의 벡터
breaks	나눌 구간을 지정한다.
freq	NULL이거나, TRUE이면 데이터 빈도 수에 히스토그램이 그려진다. FALSE이면 확률 밀도로 보여 준다.
기타 옵션	main, xlim, col, xlab, ylab 등등

hist() 함수의 반환 값

함수의 반환 값은 다음과 같은 정보들이 있다.

항목	설명
breaks	계급의 상하한 값을 반환해준다.
counts	도수(변량이 가지고 있는 개수)를 의미한다.
density	계급 구간의 확률 밀도를 의미한다.
mids	계급 값(계급의 중간에 있는 값)을 의미한다.

legend() 함수

legend() 함수는 해당 지점에 범례를 만들어 주는 함수이다.

항목	설명
pch	점 모양(plotting character)을 설정한다.
cex	점의 크기를 지정한다.
col	색상을 지정한다.
fill	내부에 채워질 색상을 지정한다.
lty	선의 종류를 지정한다.(line type)
lwd	line width 즉, 라인의 굵기를 의미한다.
bg	배경 색상(background color)을 지정한다.
사용 예시	<p>예시 1) legend(xpos, ypos, c('하하하', '호호호'), pch=c(3, 1))</p> <p>예시 2) legend(12, 11, col2, cex=0.8, col=colors, lty=1, lwd=2, bg="white")</p> <p>예시 3) x 좌표 19, y 좌표 71 에 무지개색으로 5 개의 범례를 표시한다.</p> <p>legend(19, 71, c('가', '나', '다', '라', '마'), cex=0.8, fill=rainbow(5))</p> <p>legend('topright', c('가', '나', '다', '라', '마'), cex=0.8, fill=rainbow(5))</p>

lines() 함수

lines()는 points()와 마찬가지로 plot()으로 새로운 그래프를 그린 뒤 선을 추가하여 그리는 목적으로 사용된다. 산점도 그래프로 그려진 내용에 대하여 선을 연결할 때에도 사용할 수 있다.

항목	설명
사용 가능 옵션	col, type, lwd
사용 예시	lines(age, col=mycolor, type='o', lwd = 2)

matplot() 함수

matplot()함수는 행렬에 대하여 컬럼 단위로 그래프를 그리기 위한 함수이다.

함수에서 사용 가능한 옵션은 다음과 같다.

참조 문서 : <https://cafe.naver.com/ugcadman/1194>

이 외에도 matlines(), matpoints() 함수등이있다.

항목	설명
x, y	그림을 그리기 위한 vectors or matrices
사용 가능한 옵션	type, lty, lwd, lend, pch, col, cex, bg, xlim, ylim

pie() 함수

pie() 함수에서 사용 가능한 옵션은 다음과 같다.

항목	설명
angle, density	pie 부분을 구성하는 각도, 수(density)를 지정한다.
clockwise	시계 방향(T)으로 회전할 지, 반시계 방향(기본 값)으로 회전할 지를 결정한다.
col	색상을 지정한다.
init.angle	시작되는 지점의 각도를 지정한다.
labels	각 pie 부분의 이름을 지정하는 문자열 벡터를 지정한다.
radius	반지름의 크기를 지정한다.
기타 옵션	main, cex 등등

points() 함수

points() 함수는 plot() 함수를 이용하여 그래프를 그린 뒤 추가적으로 특정 지점에 점을 그려 주는 함수이다.

항목	설명
사용 형식	points(x, y, pch=1)
pch	점 모양(plotting character)을 설정한다.

text() 함수

text() 함수는 그래프 내부의 임의의 위치에 글씨를 작성한다.

항목	설명
x, y	글자가 표시될 위치를 지정한다.
adj	adj 는 정렬 방식으로 기본 값은 가운데 정렬이다.(0:왼쪽 정렬) adj = 0 또는 1(adjustment)
cex	점의 크기를 결정한다.
col	보여 지는 글자의 색상을 지정한다.
labels	보여줄 문자열(글자)을 지정한다. 예시 : labels=paste(변수, '%(평균 출루율)')
사용 예시	text(x=bodywt+45, y=brainwt+45, labels=animal) text(x=bodywt+45, y=brainwt+45, labels=animal, adj=0)

title() 함수

text() 함수는 그래프 내부의 임의의 위치에 글씨를 작성한다.

항목	설명
x, y	글자가 표시될 위치를 지정한다.

히스토그램과 막대 그래프의 차이

항목	히스토그램	막대 그래프
함수 이름	hist()	barplot()
데이터의 종류	연속형 데이터	이산형 데이터
데이터 예시	키, 나이, 금액 등등	성별, 지역 등등
그래프의 차이점	막대가 서로 붙여 있다.	막대가 분리되어 있다.

Options

Allow the user to set and examine a variety of global options which affect the way in which R computes and displays its results.

도움말 보기 : help(options)

항목	설명
scipen	차트를 그릴 때 큰 숫자인 경우 e+05의 형식으로 눈금이 그려 지는 데 이것을 없애는 옵션이다. options(scipen = 25) # 25는 임의의 숫자이다. scipen의 기본 값은 0이다.
max.print	행이 많아서 max.print 경고가 나오면 다음 문장을 수행해 주면 된다. options(max.print=999999)

ggplot2 패키지

카페 문서 : 1293번

ggplot2 패키지는 기하학적 객체들(점, 선, 막대 등등)과 미적 특성(색상, 모양, 크기 등등) 연결 등을 이용하여 기본 속성만으로도 아주 실용적인 색상 조합과 더불어 미려한 그래프로 정보들을 시각화해준다.

이 패키지에 들어 있는 항목들은 plot() 함수의 확장된 버전이다.

ggplot2 패키지가 우선 설치가 되어 있어야 한다.

Hadley Wickham 에 의하여 개발이 시작되었으며, 최근 2015 년 3 월에 1.0.1 버전을 CRAN 에 공포하였다.

함수의 옵션 중에 aes 라는 옵션이 있다.

이 옵션은 aesthetic 의 줄인 말로써, '미학적 특성(색상, 모양, 크기 등등)을 살리기 위한' 여러 가지 기교, 장식 등을 말한다.

ggplot2 함수와 관련된 레퍼런스 주소는 <https://www.rdocumentation.org/packages/ggplot2/versions/3.1.0> 이다.

ggplot2 패키지를 사용하려면 ...

```
install.packages('ggplot2')
```

```
library(ggplot2)
```

ggplot2 패키지의 주요 함수

ggplot2 패키지에서는 mpg, mtcars, diamonds 등의 데이터 셋을 제공해준다.

다음과 같은 함수들이 제공되고 있다.

함수	기능	비고
qplot()	기하학적 객체와 미적 요소 매핑으로 스케일링한다.	plot() 기능을 확장했다.
ggplot()	미적 요소 매핑에 레이어 관련 함수를 추가하여 플로팅, 미적 요소를 재사용한다.	+ 연산자 이용 미적 요소를 상속한다.
ggsave()	해상도를 적용하여 다양한 형식의 이미지 파일을 저장한다.	pdf, jpg, png등의 파일을 지원한다.

qplot() 함수

기하학적 객체(점, 선, 다각형 등등)를 크기, 모양, 색상 등의 미적 요소를 매핑하여 그래프를 그려 주는 함수이다. 도수 분포를 그래프로 그려 준다.

qplot(x축~y축, data, facets, geom, stat, position, xlim, ylim, log, main, xlab, ylab, asp)

항목	설명
binwidth	막대의 폭 속성을 지정한다.
color	색상을 설정한다.
data	차트를 그리기 위한 데이터 셋
facets	측면, 양상이란 뜻으로 어떠한 측면/양상으로 보일 줄것인가를 설정한다.
fill	채워질 색상을 설정한다.
geom	geom(기하학적) 객체를 설정한다.(기본 값 : point) 2개 이상의 geom 속성을 합쳐서 사용하고자 하는 경우에는 c() 함수를 사용하면 된다. 관련 속성 : bar(막대 그래프), line(선으로 연결), point(산점도 그리기) smooth(산점도 주변에 부드러운 곡선 추가시 사용) freqpoly(다각형 폴리곤)
size	채워진 점의 크기를 지정한다.
shape	채워질 도형들의 모양을 설정한다.

xlim() 함수 옵션

x축의 상하한 값을 조정해주는 함수이다.

예시 : xlim(0, 1000) # 최대 1000까지만 표시하겠다.

ggplot() 함수

ggplot2 패키지를 이용하여 그래프를 그려 준다.

ggplot() 함수는 **그래프를 그리기 위하여 기본적인 틀**을 만들어 주는 함수로 이해하면 될 듯하다.

ggplot() 함수에 의하여 그래프가 그려지는 절차

1. 미적 요소 맵핑(aes) : x 축, y 축, color 속성
2. 통계적인 변환 과정(stat) : 통계 계산 작업
3. 기하 객체 적용(geom) : 차트 유형
4. 위치 조정 : 채우기(fill), 스택(stack), 닛지(dodge) 유형
 geom_col(position = "dodge") 옵션을 추가하면 누적 막대 그래프를 off 시킬 수 있다.
 barplot 함수의 beside = TRUE 옵션과 동일한 역할을 수행한다.

ggplot() 함수의 기본 사용 형식

ggplot(dataframe, aes(x = x 축데이터 ,y = y 축데이터)) + geom_함수

항목	설명
dataframe	처리해야 할 데이터 프레임을 명시해준다.
aes	미적 요소(aesthetics)를 맵핑해주는 함수이다. 즉, 데이터를 표현할 때 좀 더 아름답게 표현하기 위한 여러 가지 기교가 들어 가는 곳이다. 각축(axis)에 들어가는 데이터 지정하기, 점의 모양, 크기, 색상 등등을 설정할 수 있다. 사용 예시 aes(x=이름, y=점수, fill=과목)
geom_함수	geometric object 앞에서 만들어진 데이터를 렌더링 으로 표현하는 부분이다.

aes 옵션

ggplot() 함수에서 사용할 수 있는 옵션들은 다음과 같은 항목들이 있다.

항목	설명
x, y	x축에 표시될 데이터와 y축에 표시될 데이터를 의미한다. y=reorder(이름, 영어)라고 하면 영어 점수가 높은 사람의 이름을 y축에 먼저 보여 주세요. 역순은 - 옵션으로 가능하다.
color	테두리 색상을 지정할 컬럼 또는 항목을 지정한다. color=gender # gender 컬럼을 이용하여 테두리 색상을 설정하겠다.
fill	채울 색상을 지정할 컬럼 또는 항목을 지정한다.
group	그룹핑을 위한 옵션이다. geom_boxplot() 함수를 이용하여 상자 그래프를 만들 때 데이터를 그룹화할 기준을 지정한다. 두 그룹 이상에 대하여 짝은 선을 그릴 때에도 사용 가능하다. group=name이라고 한다면, 사람 이름으로 그룹핑하라는 의미이다.

ggsave() 함수

ggplot2 패키지에 의하여 그려진 그래프를 pdf, 이미지(jpg, png 등)로 저장할 수 있는 함수이다.
 이미지로 저장하는 경우 dpi 속성을 이용하여 이미지의 해상도를 적용할 수 있다.
 이미지의 폭과 너비도 지정이 가능하다.

```
ggsave(filename="kd.png",dpi=500)
```

항목	설명
filename	저장할 파일 이름
dpi	해상도(dots per inch)
plot	ggplot 함수의 결과를 저장하고 있는 이미지 객체를 지정한다.
scale	척도
width, height	너비, 높이를 설정한다.

사용 예시

```
p = ggplot(diamonds, aes(clarity))
p = p + geom_bar(aes(fill=cut), position="fill") # bar 추가
ggsave(file="C:\\RProject\\Rwork\\output\\bar.png", plot=p, width=10, height=5)
```

추가 사항

annotate()

그래프 위에 사각형(rect)이나 화살표(arrow) 등으로 특정한 영역을 강조하고자 할 때 사용하는 함수이다.

텍스트 : 화살표 : `annotate("segment", arrow=arrow())` , with grid package

음영 사각형 : `annotate("rect")`

항목	설명
사용 형식	<code>annotate(geom='모양', xmin='x축 시작', ymin='y축 시작', xmax='x축 끝', ymax='y축 끝', xend='x축 화살표 끝', yend='y축 화살표 끝', alpha='투명도', color='테두리색상', fill='채울색상', arrow='화살표')</code>
geom	모양을 설정한다. 사각형(rect), 화살표(arrow() 함수), 선(segment)

coord_fixed

1:1 의 비율로 x축과 y축을 설정하고자 할 때 사용한다.

coord_flip()

직각 좌표계를 뒤집어서 막대 그래프를 오른쪽으로 90도 회전시켜 주는 함수이다.

$x \rightarrow y$, $y \rightarrow x$ 로 회전시켜 준다.

보여 지는 라벨의 문자열이 매우 긴 경우에 유용하게 사용할 수 있다.

항목	설명
사용 형식	ggplot() 결과 객체 + coord_flip()

coord_polar()

직각

ggtitle()

그래프의 제목을 넣어 주는 함수이다.

항목	설명
사용 형식	ggtitle('메인 제목', substitute('asdf'))

labs()

그래프의 제목 및 축의 제목을 추가해주는 함수이다.

항목	설명
사용 형식	labs(x='x축 이름', y='y축 이름', title='그래프 제목') labs(title='수입 현황', y = y_caption, x = x_caption, colour = "항목")

scale_color_brewer()

색상 팔레트를 이용하여 색상을 지정하고자 할 때 사용한다.

to use color palettes from RColorBrewer package

항목	설명
사용 예시	scale_color_brewer(palette="Dark2")
palette	Dark2 Paired # Continuous colors Accent # Gradient colors

scale_color_grey()

to use grey color palettes.

항목	설명
사용 예시	scale_color_grey()

scale_color_manual()

임의의 색상으로 지정하고자 할 때 사용한다.

항목	설명
사용 예시	scale_color_manual(values=c('red', 'green', 'blue'))

scale_x_continuous

scale_x_discrete()

x축에 대하여 이산량에 대한 포지셔닝을 지정할 수 있는 함수이다.

y축은 scale_y_discrete 함수를 사용하면 된다.

항목	설명
limits	x축에 보여줄 항목의 순서를 사용자 정의 순서로 바꿀 수 있다. limits=c('young', 'middle', 'old') # young 항목을 먼저 보여 주겠다.

scale_y_continuous

축의 눈금을 임의대로 설정하는 옵션이다.

항목	설명
coma 유형	세자리마다 coma 유형을 지정한다. library(scales) # Generic plot scaling methods scale_y_continuous(labels=comma)
축의 title 지정	scale_x_continuous("xtitle의 값 지정")
축의 상/하한선 지정	scale_x_continuous(limits = c(2, 6))
주눈금 지정	scale_x_continuous(breaks = c(2, 4, 6))
주눈금 지정 및 캡션화	scale_x_continuous(breaks = c(2, 4, 6), label = c("two", "four", "six"))
눈금에 %지정	scale_y_continuous(labels = scales::percent)

Theme(테마) 함수

ggplot2에서는, 그래프의 주요 구성 요소 및 디자인들을 미리 만들어 놓고 필요할 때 즉시 반영 시킬 수 있도록 해 놓았다. 이러한 그래프의 외형을 지정할 수 있는 기능을 테마(Theme)라고 하는 데, 자주 쓰이는 여러 가지 환경들을 모아서 테마 형태로 제공한다.

주로 글자나 배경 등을 제어할 때 많이 사용한다.(축, 범례, 레이블, 격자 등등)

테마를 수정하려면 element_xxx 형태의 객체를 수정하면 된다.

element_text, element_line, element_rect 등이 있다.

Theme 관련 함수

ggplot2 패키지에는 다음과 같이 8가지의 테마가 있다.

항목	설명
theme_gray()	회색 바탕과 흰 선
theme_bw()	흰 바탕과 회색 선을 가진 테마이다.
theme_linedraw()	흰 바탕과 가늘고 검은 선
theme_light()	밝은 회색 바탕
theme_dark()	어두운 바탕
theme_minimal()	단순한 배경
theme_classic()	눈금과 안내선이 없는 기본 바탕을 지정한다.
theme_void()	가장 간결한 바탕

Theme 관련 속성과 메소드

다음과 같은 속성과 메소드들이 있다.

참조 사이트 : <https://ggplot2.tidyverse.org/reference/theme.html>

항목	설명
axis.text.x	x 축에 보여 지는 글자에 대하여 옵션들을 지정한다. 사용 예시 axis.text.x=element_text(angle=45, hjust=1, vjust=1, color='blue', size=8)
legend.position	범례의 위치를 지정한다. c(0.9, 0.9) 또는 'top'의 형식으로 표현하면 된다. top, bottom, none 등등 theme(legend.position = 'none') # 범례 없애기
legend.title	범례의 타이틀
panel.grid.major	차트에 보여지는 큰 간격의 눈금 을 설정할 때 사용한다.
panel.grid.minor	차트에 보여지는 작은 간격의 눈금 을 설정할 때 사용한다.
panel.grid.major.x=element_blank()	x축의 큰 눈금에 대한 설정하지 않겠다.

<code>panel.grid.minor.x=element_blank()</code>	x축의 작은 눈금에 대한 설정하지 않겠다.
<code>panel.grid.major.y=element_line(color='red', linetype='dashed')</code>	y 축의 큰 눈금에 빨간 색상의 점선을 그리겠다.
<code>plot.title()</code>	<code>theme(plot.title = element_text(hjust=0.5))</code> 차트 제목의 타이틀을 중앙에 배치하고자 할 때 사용한다. 반드시 <code>theme()</code> 함수의 매개 변수로 넣어 줘야 한다.

element_xxx

항목	설명
<code>element_blank()</code>	그리지 않겠다./표시하지 않겠다.
<code>element_line()</code>	선을 그리겠다. <code>color='red', linetype='dashed'</code>

theme_bw() 함수

이 함수를 추가하여 사용하면 지금 그리는 그래프에만 한하여 회색 배경 없이 격자 무늬만 표시해주는 theme를 적용하겠다는 의미이다.

예시 : `ggplot(...) + theme_bw() + theme(...)`

geom_xxx 함수

그래프에 추가적으로 그림을 그려 주는 geom_xxx 함수에는 다음과 같은 항목들이 있다.

geom_abline

절편과 기울기를 이용하여 직선의 방정식을 이용하여 사선 그래프를 그려 준다.(기울기가 있는 직선)

항목	설명
사용 형식	geom_abline(intercept, slope)
intercept	절편을 의미한다.
slope	기울기를 의미한다.

geom_area

직선

항목	설명
사용 형식	geom_abline(intercept, slope)

geom_bar

bar plot 함수와 비슷한 기능을 수행하는 함수로써, 막대 그래프를 그려주는 함수이다.

막대 그래프는 하나의 변수에서 각 값의 빈도를 파악할 때 사용하는 그래프이다.

원자료(original)를 이용하여 막대 그래프를 그릴 때는 geom_bar() 함수를 사용한다.

요약된 표(dplyr 패키지)를 이용하여 막대 그래프를 그릴 때는 geom_col() 함수를 사용한다.

항목	설명
사용 형식	geom_bar(stat, fill='green', colour='red', 등등)
stat	주어진 데이터에서 geom에 필요한 데이터를 생성한다.
fill	내부를 채울 색상을 설정한다.
colour	테두리 색상을 설정한다.
width	막대 그래프의 너비를 지정하는 옵션이다.
position	누적 그래프를 그리지 않으려면 position_dodge() 또는 position_dodge2() 항목을 설정하면 된다. 예시 : position=position_dodge2()

geom_boxplot

상자 그래프는 분포를 비교하고자 할 때 사용하는 함수이다.

주의할 사항은 aes()에서 group 옵션을 이용하여 그룹을 지을 열을 설정해줘야 한다.

항목	설명
사용 형식	

geom_col

막대 그래프를 그려주는 옵션이다.

기본 값으로 누적된 그래프를 그려준다.

주로 요약된 표(dplyr 패키지)를 이용하여 막대 그래프를 그릴 때는 이 함수를 사용한다.

항목	설명
사용 형식	geom_col(position = 'dodge')
position	'dodge'는 데이터를 누적되지 않도록 항목을 붙여서 그려준다. 'dodge2'는 dodge와 동일한데 항목을 떼어 내어 그려준다.

geom_density

밀도 그래프를 보여 준다.

항목	설명
사용 형식	geom_density(alpha=.2, fill="red")
alpha	채워지는 내부 색상의 투명도를 지정한다.
fill	내부에 채울 색상을 지정한다.

geom_hline

평행선(세로 선)을 그려 준다.

항목	설명
사용 형식	geom_hline(yintercept)

geom_histogram

도수 분포를 기둥 모양으로 표현한 히스토그램을 만들어 준다.

항목	설명
사용 형식	<code>geom_histogram(color="black", fill="white", bins = 20)</code>
alpha	내부 색상에 대한 투명도를 지정한다.(0 ~ 1.0)
binwidth	그래프의 폭(기둥)을 조정하는 옵션이다.
bins	그래프의 폭(기둥)을 조정하는 옵션이다.
color	테두리 색상을 지정한다.
fill	내부에 채울 색상을 지정한다.
position	2개 이상의 그룹을 보여주고자 할 때 사용한다. 취할 수 있는 값 identity : 공통된 영역은 누적 형태로 보여준다. dodge : 따로 따로 보여 준다.

geom_line

찍은 선 그래프를 그려 준다.

찍은 선 그래프는 점과 점을 순차적으로 연결하여 표현한 자료로써, 산점도에 비하여 변화를 관찰하기 쉽다.

항목	설명
사용 형식	

geom_point

산점도란 두 개의 변수간의 관계를 파악하기 위하여 평면에 관측점을 찍어서 그리는 방법이다.

지도/그래프에 산점도를 그려준다.

항목	설명
사용 형식	<code>geom_point(data=loc, aes(x=LON, y=LAT), size=3, alpha=0.7, color="red")</code>
aes	지도의 위도(x), 경도(y), label, color 등을 설정한다.
alpha	점의 투명도이다.
color	점의 색상을 지정한다.
data	그리기 위한 데이터
shape	모양을 지정한다. (예시 : shape=5) # shape : 1(비어 있는 원), 5(비어 있는 다이아몬드), 22(비어 있는 네모)
size	점의 크기를 지정한다. (예시 : size=6)

geom_segment

이 함수는 클리브랜드 점 그래프로 알려진 형태로 데이터를 표현해준다.

bar 그래프를 대체해서 많이 사용되는 형태이다.

항목	설명
사용 형식	<code>geom_segment(aes(yend=이름), xend=0, color='blue')</code>
yend	y 축에 보여줄 데이터
xend	x 축에 보여줄 데이터
color	색상을 설정한다.

geom_text

그래프에 글자를 입력할 때 사용하는 함수이다.

각 항목의 이름이나 값 등을 표시하고자 할때 주로 사용한다.

항목	설명
사용 형식	<code>geom_text(data=loc, aes(x = LON, y = LAT+0.001, label=보여줄문자열, vjust=세로위치, hjust=가로위치), size=3)</code>
data	그리기 위한 데이터
aes	지도의 위도(x), 경도(y), 보여줄 문구(label)를 설정한다.
size	크기이다.

geom_vline

수직선(가로 선)을 그려 준다.

항목	설명
사용 형식	<code>geom_vline(aes(xintercept=mean(키)), color="blue", linetype="dashed", size=1)</code>
color	선의 색상을 지정한다.
linetype	선의 종류를 지정한다.('dashed')
size	선의 굵기를 지정한다.

크롤링(Crawling)

웹 사이트의 특정 페이지에 접속하여 데이터를 크롤링해보도록 한다.

rvest 패키지

R로 크롤링을 할 때 가장 많이 쓰는 패키지는 rvest이다.

오라클 접속을 위한 사전 준비

```
install.packages("rvest")
library(rvest)
```

rvest 패키지와 관련된 함수 목록은 다음과 같다.

항목	설명
html_attr(node, attr)	node 항목에서 attr이라는 속성의 정보를 읽어 들인다.
html_node()	특정 태그 중에서 가장 맨 앞의 태그를 읽어 들일 때 사용하는 함수이다.
html_nodes(html, selector)	조건에 맞는 태그들을 모두 읽어 들일 때 사용하는 함수이다. html 객체에 대하여 selector을 이용하여 목록을 읽어 들인다. 목록의 갯수는 length() 함수를 이용하여 구할 수 있다.
html_nodes(x, css, xpath)	class 속성이 'searchCont'인 태그 찾기
html_text()	본문의 텍스트를 읽어 들인다.
read_html(url, encoding)	url에 명시된 HTML 페이지를 encoding 하여 읽어 들이는 함수이다.

RSelenium

R을 이용한 Selenium 실행 (Windows 10 기준)

1. Selenium을 사용하는 이유

Selenium은 GET이나 POST로 가져오기 힘든 경우 사용하면 편리하다. 예를 들어 클릭해서 로그인 후 내용을 크롤링 한다든지, 검색어를 입력해서 크롤링 하는 경우, 웹표준을 지키지 않아서 크롤링이 어려운 경우 등에 사용하면 편리하다.

2. R을 이용하여 Selenium 실행하기

*** Selenium을 사용하려면 사전에 JAVA가 설치 되어 있어야 한다.

1) 파일을 아래 링크를 통해 다운받는다. 저장은 아래 이미지와 같이 같은 폴더를 생성하여 저장한다.

selenium standalone server

gecko driver

chrome driver

<http://selenium-release.storage.googleapis.com/index.html>

<https://github.com/mozilla/geckodriver/releases/tag/v0.17.0>

<https://sites.google.com/a/chromium.org/chromedriver/>

<- 다운 받은 후 같은 폴더에 저장

2) cmd를 관리자 권한 실행한다.

검색 > 검색어 cmd 입력 > 마우스 오른쪽 버튼 클릭 > 관리자권한 실행

3) 윈도우 탐색기에서 다운받은 파일 저장폴더로 간다. 사진과 같은 위치에 오른쪽 버튼을 누른 후 경로를 복사한다.

4) cmd 창에 아래와 같이 입력한다.

cd <복사한 주소 경로> (ex) cd C:\selenium

5) 아래의 명령어를 입력한다.

```
java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-2.52.0.jar -port 4445
```

```
java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-x.x.x.jar -port 4445
```

x.x.x.jar의 x.x.x는 다운받은 selenium의 버전을 입력해주면 된다.

(ex) java -Dwebdriver.gecko.driver="geckodriver.exe" -jar selenium-server-standalone-3.5.3.jar -port 4445

아래와 비슷하게 뜨면 성공한 것.

6) 이제 위 창은 유지해둔 채로 R studio를 실행시키고 아래의 코드를 입력한다.

```
install.packages('RSelenium')
```

```
library(RSelenium)
```

```
# 포트 번호는 cmd 창에서 지정했던 포트 번호와 동일하게 작성하면 된다.
```

```
remDr <- remoteDriver(remoteServerAddr = "localhost" ,  
                      port = 4445L, # port 번호 입력  
                      browserName = "chrome")  
# browserName : 실행 브라우저 문자열을 입력한다.
```

```
remDr$open()
```

```
# 브라우저가 실행되면 성공
```

```
# open시 해당 포트 번호가 달라서 다음과 같은 오류가 발생할 수 있다.
```



```
# [1] "Connecting to remote server"
# Error in checkError(res) :
#   Undefined error in httr call. httr output: Failed to connect to localhost port 4445: Connection refused

# 접속할 사이트 입력
remDr$navigate("https://www.google.com")    # google로 연결 됨

# 출처: https://hmtb.tistory.com/5
```

항목	설명

데이터 베이스 오라클(Oracle)

R에서는 오라클 데이터베이스에 접속하여 sql 구문들을 수행할 수 있다.

오라클 접속을 위한 패키지

다음과 같은 패키지들을 설치해 주어야 한다.
또한, JDK도 미리 설치 되어 있어야 한다.

오라클 접속을 위한 사전 준비

```
install.packages('rJava')
install.packages('DBI')
install.packages('RJDBC')
```

```
Sys.setenv(JAVA_HOME='c:/program files/Java/jre1.8.0_191')
library(rJava)
library(DBI)
library(RJDBC)
```

JDBC

JDBCDriver 객체를 구해준다.

항목	설명
사용 형식	<code>drv <- JDBC(driverClass = driver, classPath = jarpath)</code>
driverClass	오라클에 접속하기 위한 드라이브 클래스(OracleDriver)를 의미한다.
classPath	ojdbc14.jar 또는 ojdbc6.jar 파일이 존재하는 full path를 의미한다.

dbConnect

접속 객체를 구해주는 함수이다.

항목	설명
사용 형식	<code>conn = dbConnect(drv, url, id, password)</code> <code>class(conn)</code> # 타입 : <code>JDBCConnection</code>
drv	JDBC 함수를 이용하여 구한 driver 객체를 의미한다.
url	데이터 베이스 출처(url)

dbGetQuery

쿼리 결과를 조회하기 위한 구문이다.

컬럼 이름이 모두 대문자로 리턴된다.

항목	설명
사용 형식	<code>query <- 'select * from test_table'</code> <code>result <- dbGetQuery(conn, query)</code> <code>result</code> # 반환 결과는 <code>data.frame</code> 이다.
conn	데이터 베이스 접속 객체이다.
query	sql 구문을 의미한다.

dbSendUpdate

쿼리 문장에 대한 DML 작업을 수행한다.

항목	설명
사용 형식	<code>query <- "insert into test_table "</code> <code>query <- paste(query, " values('kang', '1234', '강감찬', 60) ")</code> <code>dbSendUpdate(conn, query)</code>
conn	데이터 베이스 접속 객체이다.
query	sql 구문을 의미한다.

dbWriteTable

데이터 베이스에 dataframe 을 신규 테이블에 추가한다.

결과 값으로 TRUE 또는 FALSE 을 리턴한다.

항목	설명
----	----

사용 형식	<code>bool <- dbWriteTable (conn, name="asdf", value=manydata, row.names = FALSE, append=TRUE)</code>
conn	데이터 베이스 접속 객체를 의미한다.
name	테이블 이름을 지정한다.
value	저장하고자 하는 data.frame 을 지정한다.
row.names	
append	

데이터 베이스 Sqlite

표를 만들거나

sqlite 접속을 위한 사전 준비

```
install.packages('DBI')
install.packages('RSQLite', dependencies = TRUE)

Sys.setenv(JAVA_HOME='c:/program files/Java/jre1.8.0_191')
library(DBI)
library(RSQLite)
```

dbClearResult

dbSendQuery 를 사용한 경우 쿼리의 처리 결과를 제거하기 위하여 사용한다.

항목	설명
사용 형식	

dbExistsTable

해당 테이블이 존재하는 지를 판별해주는 함수이다.

항목	설명
사용 형식	<pre>bool <- dbExistsTable(conn, tablename) if(bool == TRUE) { cat('테이블 존재') }else{</pre>

	cat('테이블 존재 안함') }
--	-----------------------

dbGetQuery

select 구문을 수행하여 DB에서 데이터를 가져오는 역할을 한다.

반환 타입은 list 자료형의 data.frame 이다.

항목	설명
사용 형식	query <- 'select * from products' data <- dbGetQuery(conn, query)

dbListTables

테이블의 목록을 반환해주는 함수이다.

항목	설명
사용 형식	tblist <- dbListTables(conn) mode(tblist) # "character" class(tblist) # "character"

dbSendQuery

테이블 생성이나, 행 입력/수정/삭제 쿼리 등을 수행할 때 사용한다.

항목	설명
사용 형식	query <- "delete from products where name = '에이드'" rs <- dbSendQuery(conn, query) dbClearResult(rs)

dbWriteTable

데이터 베이스에 dataframe 을 신규 테이블에 추가한다.

결과 값으로 TRUE 또는 FALSE 을 리턴한다.

항목	설명
사용 형식	bool <- dbWriteTable(conn, 'students', alldata, row.names=F, append=TRUE) if(bool == TRUE){

	cat('yes') }else{ cat('no') }
conn	데이터 베이스 접속 객체를 의미한다.
name	테이블 이름을 지정한다.
value	저장하고자 하는 data.frame 을 지정한다.
row.names	
append	

dbSendStatement()와 dbExecute()는 dbSendQuery()와 dbGetQuery()와 유사한 기능을 가지고 있다.
 # dbExecute : dml 문장 수행 후 반영된 row 수를 리턴한다.
 # rs 는 자동으로 닫는다.

placeholder 사용

placeholder 란 실행 직전에 치환이 되어야 할 변수를 말한다.
 dbBind() 함수를 이용하여 바인딩하면 된다.

placeholder

```
query <- 'select * from students where jumsu = :x'
rs <- dbSendQuer placeholder y(conn, query)
dbBind(rs, param = list(x = seq(70, 75, 0.1)))
result <- dbFetch( rs )
result
dbClearResult(rs)
```

```
rs <- dbSendQuery(conn, query)
```

```
while ( !dbHasCompleted(rs) ){
df <- dbFetch(rs, n = 5)
print( df )
cat('-----\n')
}
dbClearResult(rs)
```

```
query <- 'select * from students where name = :x'
rs <- dbSendQuery(conn, query)
dbBind(rs, param = list(x='양종철'))
```

```
result <- dbFetch( rs )
result
dbClearResult(rs)
```

```
query <- "delete from students where name = '양종철'"
dbExecute(conn, query)
```

```
rs <- dbSendStatement(conn, 'delete from students where jumsu < :x')
dbBind(rs, param = list(x = 50))
dbGetRowsAffected(rs) # 반영된 행의 갯수를 반환해준다.
```

```
dbClearResult(rs)
```

```
dbDisconnect(conn)
```

참조 2) <https://cran.r-project.org/web/packages/DBI/DBI.pdf>

연결하기

```
conn <- dbConnect(RSQLite::SQLite(), dbname="coffee.db")
```

접속 객체 해제.

```
dbDisconnect(conn)
```

```
conn <- dbConnect(RSQLite::SQLite(), dbname='students.db')
mode(conn) # "S4"
class(conn) # "SQLiteConnection"
```

JSON 다루기

JSON 형식의 데이터를 R에서 다루기 위해서는 두 개의 package가 존재한다.

첫 번째는 rJSON package이고 두 번째는 jsonlite package이다.

jsonlite는 rJSON을 확장시켜 만든 것인데, 이 패키지를 사용해보도록 한다.

설치 방법

```
install.packages("jsonlite")
library(jsonlite)
```

관련 메소드

JSON 데이터를 처리하기 위한 메소드 목록은 다음과 같은 것들이 존재한다.

항목	설명
fromJSON	Convert R objects to/from JSON 데이터 프레임은 다루듯이 \$를 이용하여 내부 항목들을 접근할 수 있다. []를 이용하여 indexing이 가능하다. 반환 타입은 data frame이다.
toJSON	데이터를 JSON 형식의 데이터로 변환하여 준다. class 함수를 이용하면 확인 가능하다.

지원하는 함수들의 목록은 아래와 같다.

flatten: Flatten nested data frames

prettify: Prettify or minify a JSON string

rbind.pages: Combine pages into a single data frame

serializeJSON: serialize R objects to JSON

stream_in: Streaming JSON input/output

unbox: Unbox a vector or data frame

validate: Validate JSON

XML 다루기

XML은 소프트웨어나 하드웨어에 관계 없이 데이터를 저장하거나 전송하기 위한 도구, 마크업 언어이다.

일반적으로 데이터를 구조화해서 다른 프로그래밍 언어간 전송할 때 자주 사용하기도 하고 최근에는 open API 에서 데이터를 얻을 때 json 이나 xml 형태로 얻는 경우가 많다.

xml2 패키지 참조 사이트 : <https://blog.rstudio.com/2015/04/21/xml2/>

xml2 패키지 관련 주요 함수

xml2 패키지와 관련된 주요 함수 목록은 다음과 같다.

항목	설명
read_xml(url)	url(파일 또는 인터넷 주소)의 xml 문서를 읽어 온다. 반환 타입은 xml_document 객체이다.
xml_children(xml_doc)	xml_document 객체의 내부 element 들에 접근하기 위한 함수이다. 반환 타입은 xml_nodeset 객체이다.
xml_find_all(data, b)	data는 xml 데이터이고, tagpath는 찾으려는 태그의 경로이다. `./*` 의 경우는 현재 element 에서 한 단계 안쪽에 있는 모든 element 를 찾는 경로이다.
xml_name()	여러 엘리먼트의 태그명을 추출하는 함수이다.
xml_text()	content 를 추출하는 함수이다.

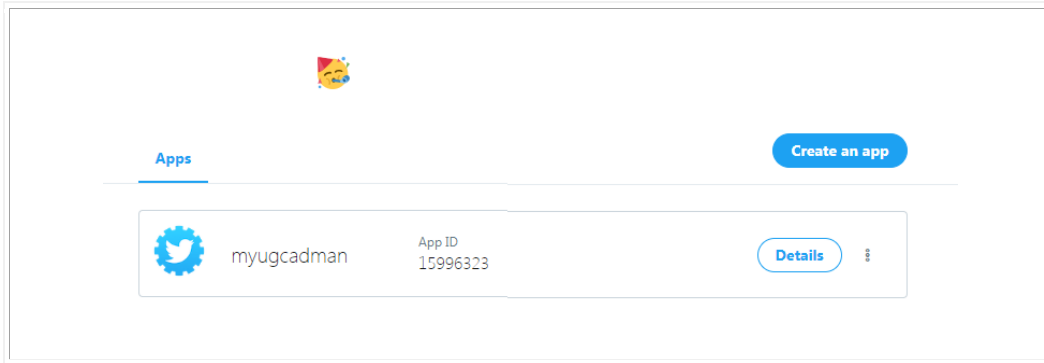
Twitter API 사용하기

트위터 API는 트위터 계정만 있으면 사용 가능하다.
만약 트위터 계정이 존재하지 않으면 우선 계정 생성부터 수행하도록 한다.
트위터는 모든 계정을 잠재적인 개발자 계정으로 취급한다.

Twitter App 생성하기

트위터 API 사용 권한을 얻기 위하여 Application Management 웹 사이트에 접속 후 로그인한다.
참조 사이트 : <https://developer.twitter.com/en/apps>

해당 사이트에 접속하여 [Create an app] 버튼을 클릭하여 다음 그림과 같이 App 를 하나 만들도록 한다.



등록 화면이 보이면 필요한 정보를 입력하고, 가장 하단의 [Create] 버튼을 클릭하여 등록한다.
항목 중에 (required)라고 명시되어 있는 곳은 필수 사항이다.

항목	설명
App name (required)	사용하고자 하는 앱의 이름을 입력한다.
Application description (required)	애플리케이션에 대한 세부 설명을 작성한다.
Website URL (required)	네이버나 다음의 url 주소를 입력하도록 한다.
Tell us how this app will be used (required)	이 앱을 어떤 방법으로 사용할 것인지를 상세하게 설명한다.

THE MEDICAL EXAMINATION OF THE CHILD

트위터 관련 패키지

다음과 같은 패키지가 사용된다.

항목	설명
twitteR	twitter 인증 관련 함수를 제공한다.
ROAuth	OAuthFactory 객체 제공을 위한 패키지이다.
base64enc	한국어 검색을 사용하기 위해서 enc2utf8() 함수를 제공한다.
RCurl	

twitteR 패키지 관련

twitter 패키지와 관련된 항목들은 다음과 같은 것들이 있다.

항목	설명
searchTwitter	제공된 단어를 이용하여 트위터에서 검색을 수행하는 함수이다.
setup_twitter_oauth()	키를 이용하여 OAuth 인증을 수행하기 위한 함수이다. setup_twitter_oauth(consumer_key, consumer_secret, access_token=NULL, access_secret=NULL)
twListToDF()	크롤링한 데이터를 데이터 프레임으로 변환해준다.

searchTwitter 함수

제공된 단어를 이용하여 특정 기간 동안의 트위터에서 검색을 수행하는 함수이다.

다음과 같은 매개 변수 목록들이 있다.

검색된 결과는 list 자료형이다.

항목	설명
lang	사용하고자 하는 language를 지정한다.
n	검색하고자 하는 결과물의 갯수를 의미한다.
searchString	검색하고자 하는 단어를 의미한다.
since	검색 시작 일자를 지정한다.(since='2018-01-01')
until	검색 종료 일자를 지정한다.

텍스트 마이닝 개요

텍스트 마이닝이란 문자로 된 데이터에서 어떤 가치가 있는 정보를 얻어 내어 분석하는 기법을 의미한다.

문장을 구성하는 어절들이 어떠한 품사로 되어 있는 지를 파악해야 하는 데, 이것을 형태소 분석이라고 한다.

텍스트 마이닝을 이용하여 SNS나 웹 사이트에 올라온 글을 분석하면 현재 가장 이슈가 되는 용어들이 어떠한 것들인지 파악할 수 있다.

비정형 데이터 처리

일반적으로 비정형 데이터는 텍스트 자료나 기존에 준비된 디지털 자료를 대상으로 미리 만들어 놓은 사전과 비교하여 단어의 빈도를 분석하는 텍스트 마이닝 방식을 주로 이용한다.

따라서 한글 단어를 처리할 수 있는 우수한 사전 기능이 무엇보다도 요구된다.

특히 비정형 데이터 처리를 위해서는 사전에 없는 단어를 추가하거나 불용어를 처리하는 별도의 함수를 정의해 놓을 필요가 있다.

토픽 분석

텍스트 데이터를 대상으로 단어를 추출한 다음, 이를 단어 사전과 비교하여 단어의 출현 빈도수를 분석하는 과정이다.

즉, 텍스트 데이터의 단어 패턴을 이용해서 내용을 파악하거나 분류 및 핵심 단어 등을 추출하여 의미가 있는 데이터로 만드는 과정을 말한다.

또한 워드 클라우드(단어 구름) 패키지를 적용하여 분석 결과를 시각화하는 과정도 여기에 포함된다.

사용 분야

관련성이 있는 단어 찾기

빈번하게 나오는 단어 찾기

단어의 논조 부정/긍정적인 단어를 파악하기

패키지	설명
KoNLP	내부에 들어 있는 사전을 이용하여 한국어에 대한 텍스트마이닝을 위한 패키지이다. Java 실행 환경을 설정해주는 rJava 패키지가 먼저 로딩되어야 한다.
참고 사항	영문 분석 패키지로는 openNLP, PKEA, Snowball 등이 있다.
tm	텍스트 전처리(텍스트 마이닝)를 위한 패키지이다. 비엔타 경제 대학의 Info Feinerer이 논문 프로젝트로 만든 패키지이다. 참조 사이트 : http://tm.r-forge.r-project.org/
wordcloud	워드 클라우드(단어 구름)

텍스트 마이닝

비정형 텍스트를 기반으로 의미 있는 정보를 추출하는 기술을 텍스트마이닝이라고 한다.

데이터마이닝과는 다른 것으로, 데이터마이닝은 구조화되고 사실적인 방대한 데이터베이스에서 관심 있는 패턴을 찾아내는 기술 분야라고 한다면 텍스트 마이닝은 구조화되지 않고 자연어로 이루어진 텍스트에서 의미를 찾아내는 기술 분야이다.

비정형 텍스트에서 텍스트 마이닝을 통해 원하는 텍스트를 추출해내고, R을 통하여 이것을 시각화할 수 있다.

워드 클라우드는 문서의 단어들을 분류하여 그 빈도를 한눈에 보기 쉽게 단어 구름 형태로 만드는 것을 말한다.

텍스트 파일 읽어 오는 방법

로컬 컴퓨터에 있는 텍스트 파일을 읽어 오는 방법들은 다음과 같은 것들이 있다.

항목	설명
<code>grep(expression, x)</code>	문자 벡터 <code>x</code> 에 <code>expression</code> 이 있으면 찾아 준다.
<code>gsub(expression, replacement, x)</code>	문자 벡터 <code>x</code> 에 <code>expression</code> 을 <code>replacement</code> 으로 치환한다. # 다음 예시는逗를 공란으로 치환하는 예시이다. <code>lyrics2 <- gsub(',', '', lyrics)</code>
<code>readLines(filename)</code>	<code>data1 <- readLines('remake.txt')</code> <code>class(data1) ; mode(data1)</code> <code>length(data1)</code> # 텍스트 파일의 라인수
<code>scan()</code>	<code>lyrics <- scan('beatles_yesterday.txt', what='character')</code> 반환 결과는 vector이다.

토픽 분석을 위한 사전 패키지 설치 및 로딩

다음과 같은 항목들을 설치 및 로딩을 하면 토픽 분석을 하기 위한 절차가 마무리 된다.

패키지 설치 및 로딩할 내역

```
# jre 경로는 상황에 따라서 다를 수 있음.
Sys.setenv(JAVA_HOME='c:/program files/Java/jre1.8.0_101')

install.packages('rJava')
install.packages(c('KoNLP', 'wordcloud'))
install.packages("tm", dependencies=TRUE)
install.packages("RColorBrewer")

library(rJava)
library(KoNLP) # 한글 사전
library(tm) # 텍스트 전처리
library(wordcloud) # 단어 구름 시각화
library(RColorBrewer)
```

KoNLP 패키지

KoNLP 패키지는 Korean National Language Process의 줄임말로 내부에 들어 있는 사전을 이용하여 한국어에 대한 텍스트 마이닝을 위한 패키지이다.

Java 실행 환경을 설정해주는 rJava 패키지가 먼저 로딩되어야 한다.

형태소(Morpheme)

형태소란 언어에서 뜻을 가진 최소의 단위를 말한다.

예를 들어서 "사람"이라는 단어가 있다면 "사"와 "람"은 별도의 뜻을 가지지 못한다.

따라서, "사람"이 하나의 형태소가 되어야 한다.

일반적으로 형태소라고 하면 하나의 단어라고 봐도 크게 무리가 없을 듯 하다.

<http://kkma.snu.ac.kr/documents/index.jsp?doc=postag>

KoNLP에는 시스템 사전, 세종 사전, NIADic 사전이 포함되어 있다.

사전마다 포함하고 있는 단어의 개수 및 사용 가능한 함수 이름은 다음과 같다.

사전 파일은 내문서\R\win-library\3.5\KoNLP_dic\current\dic_user.txt 파일이다.

사전	단어 개수	함수 이름
시스템 사전	28만 단어	useSystemDic()
세종 사전	37만 단어	useSejoingDic()
NIADic	98만 단어	useNIADic()

텍스트 마이닝의 일반적인 절차

텍스트 수집 → 분해 → 단어 추출 → 정제 → 정형 데이터 생성 → 분석 → 시각화

KoNLP 패키지	설명
buildDictionary()	사전 파일 : C:\R-3.5.3\library\KoNLP_dic\current\dic_user.txt 단어를 사전에 등록해주는 함수이다. 사용 예시 add_words = c('백두산', '남산', '철갑', '가을', '하늘', '달') mydata <- data.frame(add_words, rep('ncn', length(add_words))) # replace_usr_dic = T 옵션은 이미 들어 있는 단어를 overwrite 하겠다. buildDictionary(user_dic = mydata, replace_usr_dic = T)
extractNoun()	Hannanum analyzer(한나눔 분석기)를 사용하여 명사를 추출해주는 함수이다. 추출시 공백을 기준으로 찾아 주므로 띄워 쓰기를 잘못 하게 되면 이상한 결과를 초래할 수 있다.
get_dictionary('user_dic')	사용자가 추가한 단어들을 확인할 수 있다.(사전자 정의 사전 내용 확인)

	# 세종 사전 정보 읽어 오기 dic_df <- get_dictionary('sejong')
mergeUserDic()	세종 사전에 없는 단어를 추가한다. mergeUserDic(data.frame(c("노잼"), "ncn")) ncn은 형태소 분석상 명사(비서술성 명사)라는 뜻이다.
SimplePos09()	총 9가지 형태로 구문 분석 후 태그를 붙여서 어떤 품사인지를 알려 주는 함수이다. Do pos tagging using 9 tags uses Hannanum analyzer.
SimplePos22()	SimplePos09() 함수를 22가지로 좀더 세분화 시켜서 보여 주는 함수이다.
useSejongDic()	세종 함수를 실행하는 함수이다. 현재 등록된 단어 수를 확인할 수 있다. 새로운 단어도 추가할 수 있다.

단어 필터링 관련 함수

단어 필터링을 위해서 사용되는 함수는 다음것들이 있다.

함수명	소속 패키지	기능
is.hangul(x)	KoNLP	벡터(x)를 대상으로 한글을 추출해준다.
Filter(f, x)	base	함수(f)를 이용하여 벡터(x)에 대한 필터링을 수행하는 함수이다.
nchar(x)	base	벡터(x)를 대상으로 문자 수를 반환해주는 함수이다. Count the Number of Characters (or Bytes or Width) 사용 형식 nchar(x, type = "chars", allowNA = FALSE, keepNA = NA)

불용어란 주제 색인어로서 의미가 없는 단어들을 의미하는 데, 분석에 절대적으로 필요하지 않지만 빈도 수가 상대적으로 많은 항목들을 일컫는다.

불용어는 gsub(찾을 단어, 바꿀 단어, 찾을_곳)을 사용하면 된다.

경우에 따라서, 정규 표현식도 필요하다.

말뭉치(Corpus)란 언어학, 사회학 등 조사적 목적에 의해서 특정 집단 내에서 사용한 단어들을 모아서 정리해둔 것을 의미한다

컴퓨터의 발달로 말뭉치를 통한 데이터 수집과 분석이 용이해지면서 중요성이 부상했다.

일반적으로 분석에서는 **내가 수집해 놓은 텍스트들의 모음집(집합체)**를 의미한다.

tm 패키지

텍스트 전처리(텍스트 마이닝)를 위한 패키지이다.

documents란 tm 패키지가 처리할 수 있는 단위를 말한다.

1줄이 1개의 document가 된다.

tm 패키지의 메소드

tm 패키지에 속해 있는 메소드는 다음과 같은 것들이 있다.

tm 패키지	설명
Corpus()	벡터 데이터를 대상으로 자료집(documents)을 생성한다. 원문 해석 : 말뭉치를 계산하고, 나타낸다. 참고로 Corppus 함수는 pdf, 워드 문서 같은 문서도 읽어 들일 수 있다.
DocumentTermMatrix	말뭉치를 입력 받아서 토큰화 시킨 다음, 희소 matrix를 생성해준다.
findAssocs(tdm2, 'apple', 0.5)	특정 단어와의 상관 관계를 찾고자 하는 경우에 사용하는 함수이다. 다음 구문은 단어 apple과 0.5이상의 상관 관계를 갖는 목록을 구하는 문장이다. findAssocs(tdm2, 'apple', 0.5)
getTransformations()	불용어 처리를 위한 변형 방법을 목록으로 보여 주는 함수이다. "removeNumbers" "removePunctuation" "removeWords" "stemDocument" "stripWhitespace" 등이 있다.
inspect()	말뭉치 자료집(Corpus)에 대한 내용을 상세히 살펴 보고자 할 때 사용한다. 예를 들어 sms_corpus 객체의 내용 중에서 앞쪽 3개를 확인하려면 inspect(sms_corpus[1:3])라고 표현하면 된다.
PlainTextDocument	일반 텍스트 형식으로 만들어 주는 함수이다.
TermDocumentMatrix(말뭉치)	Term-Document 형식의 행렬로 변환해 준다. 사용 예시 tdm <- TermDocumentMatrix(cps, control=list(tokenize=함수_이름, removePunctuation=T, removeNumbers=T, wordLengths=c(2, 6), # 최대 최소 단어의 길이를 지정하는 것으로 한글은 2자리 씩 차지하기 때문에 숫자로 표현시 곱하기 2를 해야 한다. weighting=weightBin)) # weightBin은 하나의 문장에서 동일하게 반 복되는 단어의 경우 한번만 카운트 하도록 한다.
tm_map()	말뭉치로 변환해주는 기법을 mapping이라고 하는 데, tm_map() 함수는 매핑을 처리해주는 함수이다. 단어 변경, 불필요한 단어를 제거, 구뎃점 제거, 대소문자 변경하기 등의 전처리를 수행해주는 함수이다.(불용어 처리 함수)
VectorSource()	텍스트 문서를 이용하여 벡터 형태로 만들어 준다. 사용 예시 sms_corpus <- Corpus(VectorSource(sms_raw\$text)) class(sms_corpus) # [1] "SimpleCorpus" "Corpus"

tm_map() 메소드의 여러 옵션들

말뭉치를 여러 가지 방법으로 변환 시켜 주는 함수(mapping)이다.

다음과 같은 항목들이 존재한다.

tm 패키지	설명
removeNumbers	Text Document로부터 숫자를 제거해준다.
removePunctuation	Text Document로부터 구두점을 제거해준다.
removeWords	불용어를 제거한다.
stopwords('english')	영문에서 for, very, and, of, are 등 단어를 불용어로 인식하게 하는 함수이다. # 영어를 위한 기본 불용어('en')와 다른 단어들을 불용어에 추가하는 코드 newstopwords <- c(stopwords('en'), 'and', 'but', 'or')
stripWhitespace	여러 개의 공백을 하나의 공백으로 변형해준다.
tolower	소문자로 변경한다. ↔ toupper

희소 매트릭스(sparse matrix)

행은 문서(sms 메시지) 번호를 열은 용어(단어)로 이루어진 행렬을 말한다.

문서 번호	boy	girl	hello	world
document 01	0	0	0	0
document 02	0	1	0	0
document 03	2	0	0	1
document 04	0	0	3	0

2번 문서에 girl이라는 단어가 1번 나왔다.

3번 문서에 boy라는 단어가 2번, world라는 단어가 1번 나왔다.

DocumentTermMatrix 함수

말뭉치를 입력 받아서 토큰화 시킨 다음, 희소 matrix를 생성해준다.

항목	설명
사용 형식	<code>result <- DocumentTermMatrix(x, control=list())</code>
사용 예시	<pre>sms_dict <- findFreqTerms(sms_sparse_matrix_train, 5) # 5글자 이상인 것만 주 리기 # somedata 객체를 희소 행렬로 변환하되, my_dict라는 사전에 있는 단어들만 제한하여 만들어라 # my_dict는 findFreqTerms() 함수를 이용하면 생성할 수 있다. sms_train <- DocumentTermMatrix(somedata, list(dictionary = my_dict))</pre>
x	말뭉치(Corpus) 객체를 지정한다.
control	부가적인 옵션을 지정한다.

findFreqTerms() 함수

문서 용어 매트릭스에서 특정 빈도 수 이상 나타난 단어들을 문자열 벡터 형식으로 만들어 준다.

즉, 지정한 빈도 이상의 단어들만 주려 내주는 함수이다.

항목	설명
사용 형식	<pre>my_dict <- findFreqTerms(matrix, n) 라는 문장은 빈도 수가 n글자 이상의 항목들만 주 려 내라는 의미이다. my_dict는 character 타입인데, 이것을 일반적으로 사전(단어들만 모아 놓은 집합)이라고 부른다.</pre>
matrix	-
n	빈도 수를 의미한다.

워드 클라우드

wordcloud는 기본적으로 wordcloud(데이터, 빈도수)로 출력할 수 있다.
데이터의 빈도수가 클수록 글자가 크게 표현해주는 방식이다.

워드 클라우드는 캘리포니아 대학 출신의 전문 통계학자 이란 펠로우(Iran Fellow)가 개발했다.

참조 사이트 : <http://cran.r-project.org/web/packages/wordcloud/index.html>

특징

빈도 수가 큰 항목은 큰 글자로 보여 준다.

항목	설명
사용 형식	<code>wordcloud(words=names(tab.1), freq=tab.1, scale=c(5, 0.5), min.freq=1, color=rainbow(10), random.color=FALSE, random.order=FALSE, rot.per=0.25)</code>
words	출력할 단어를 지정한다.
freq	언급된 빈도 수 컬럼을 지정한다.
scale	가장 빈도가 큰 단어와 가장 빈도가 작은 단어 폰트 사이의 크기 차이를 지정한다. 즉, 단어 크기의 최대 값과 최소 값을 의미한다. <code>scale=c(10,1)</code> 즉, 비율 크기를 말한다. <code>scale=c(5, 0.5)</code>
maxwords	출력될 단어들의 최대 빈도(inf : 제한 없음)
min.freq	워드 클라우드에 나오기 위한 최소 단어 빈도 수를 지정한다. 이 값 미만의 빈도는 무시가 된다. ↔ max.freq
family	글꼴(글씨체)을 설정한다.
random.order	빈도의 크기 순서로 중앙에 배치를 할 것인가를 지정하는 옵션이다. FALSE이면 많은 빈도를 보인 단어가 중앙에 배치된다. 단, <code>set.seed(숫자)</code> 를 이용하여 랜덤에 대한 시드 배정 후 워드 클라우드를 작성하면 된다. true면 랜덤하게 배치시킨다.
random.color	이것이 TRUE이면 컬러 선택이 임의로 결정이 된다. FALSE이면 빈도 순으로 색상을 설정한다.
rot.per	수직 방향에 배치된 단어들의 상대적 비율(회전 비율)
colors	컬러(색상 팔레트)를 지정한다. 가장 작은빈도부터 큰 빈도까지의 단어 색상을 설정한다. RColorBrewer 패키지의 <code>brewer.pal()</code> 함수를 사용하게 되면 은은한 느낌의 단어 구름을 만들 수 있다. 예시 <code>palette <- brewer.pal(9, 'Set1')</code>

워드 클라우드2

Wordcloud2 패키지는 Wordcloud 패키지에 비하여 다양한 시각화를 지원한다.

패키지에 포함된 다양한 함수를 이용하여 색상이나 모양 등을 자유롭게 지정하여 그릴 수 있다.

항목	설명
사용 형식	<code>wordcloud2(data = word_table, fontFamily = '맑은 고딕', size = 4, color = 'random-light', backgroundColor = 'black')</code>
backgroundColor	배경 색상
color	색상, 'random-light' # 선택한 색상만 반복되는 워드 클라우드 <code>color=rep_len(c('red', 'blue'), nrow=(변수))</code>
figPath	명시된 이미지에 워드 클라우드를 그려준다. 배경색이 희고, 요소가 검은 색인 이미지를 추천한다.
fontFamily	글꼴, '맑은 고딕'
maxRotation	$-\pi/6$
minRotation	$-\pi/6$
minSize	시각화할 최소 빈도 수를 설정한다.
rotateRatio	회전율을 설정한다.
shape	모양/형상을 설정한다. 'circle'(기본 값), 'star', 'diamond', 'triangle' 등등
size	배수 기준 워드 클라우드 크기 변경

RColorBrewer

여러 색을 제공해주는 팔레트 같은 패키지이다.

R의 기본 컬러는 처참할 정도로 기본적으로 원색이다.

항목	설명
<code>display.brewer.all()</code>	제공해주는 모든 색상 팔레트를 보기 위해서 사용한다. 색상 세트의 이름과 개수를 보여 준다.
<code>mycolor <- brewer.pal(12, "Set3")</code>	제공되는 색상 세트 이름 "Set3"이라는 팔레트의 12가 색상을 변수 mycolor에 저장한다. 색상의 수는 팔레트의 최대치를 넘을 수 없다.
<code>display.brewer.pal(n = 5, name = 'Blues')</code>	Blues라는 색상에서 5개만 가져 오시오.