

그래프 그리기 (2) - 선 그래프

선 그래프는 시간에 따라 달라지는 데이터를 표현하는 시각화 자료입니다. 예를 들면 환율이나 주가지수 등의 경제지표가 시간에 따라 어떻게 변하는지를 표현할 수 있습니다.

이와 같이 시간에 따라 나열된 데이터를 시계열 데이터(Time Series Data) 라고 하는데, 시계열 데이터를 표현하기 때문에 선 그래프를 다른 말로 "시계열 그래프"(Time Series Chart) 라고도 합니다.

#01. 그래프 생성 준비하기

1) 필요한 패키지 로드

```
# 한국 서버를 통해 라이브러리 로드
REPO_URL <- "https://cran.seoul.go.kr/"

# 그래프 패키지
if (!require(ggplot2)) install.packages("ggplot2", repos=REPO_URL)
library(ggplot2)

# 폰트 설정 패키지
if (!require(extrafont)) install.packages("extrafont", repos=REPO_URL)
library(extrafont)

# 데이터 전처리 패키지
if (!require("dplyr"))      install.packages("dplyr", repos=REPO_URL)
library(dplyr)
```

2) 한글 사용을 위한 폰트 로드

font_import() 함수와 fonttable() 함수를 사용하여 설치된 폰트를 검색하는 과정은 시스템당 최초 1회만 수행하면 된다.

만약 출력 결과물의 한글이 깨져 보인다면 위의 두 과정을 수행한 후 아래의 loadfonts()를 실행

```
# mac의 경우 `device="win"` 생략
loadfonts(device="win")
```

▶ 출력결과

```
NanumGothic already registered with windowsFonts().
NanumGothicExtraBold already registered with windowsFonts().
NanumGothic Light already registered with windowsFonts().
```

3) 그래프 기본 크기 및 불필요한 경고 메시지 끄기

```
options(repr.plot.width=20, repr.plot.height=10, warn=-1)
```

#02. 단일 선 그래프

1) 그래프 생성에 사용할 샘플 데이터 준비

2000~2018년도 신생아 수 집계 데이터 (출처: 국가통계포털)

```
년도별_신생아_수 <- read.csv("http://itpaper.co.kr/demo/r/baby.csv",
                              stringsAsFactors=F, fileEncoding="euc-kr")

년도별_신생아_수
```

▶ 출력결과

A data.frame: 19 × 2

년도	신생아수
<int>	<dbl>
2000	640089
2001	559934
2002	496911
2003	495036
2004	476958
2005	438707
2006	451759
2007	496822
2008	465892
2009	444849
2010	470171
2011	471265
2012	484550
2013	436455
2014	435435
2015	438420
2016	406243
2017	357771
2018	326822

2) 기본 선 그래프 생성

구문형식

`geom_line()` 함수에 x,y축 정보와 색상, 종류, 굵기 등을 설정한다.

```
ggplot(data=데이터프레임) + geom_line(aes(x=x축컬럼, y=y축컬럼),
                                          color=색상값,
                                          linetype=1~4,
                                          size=굵기)
```

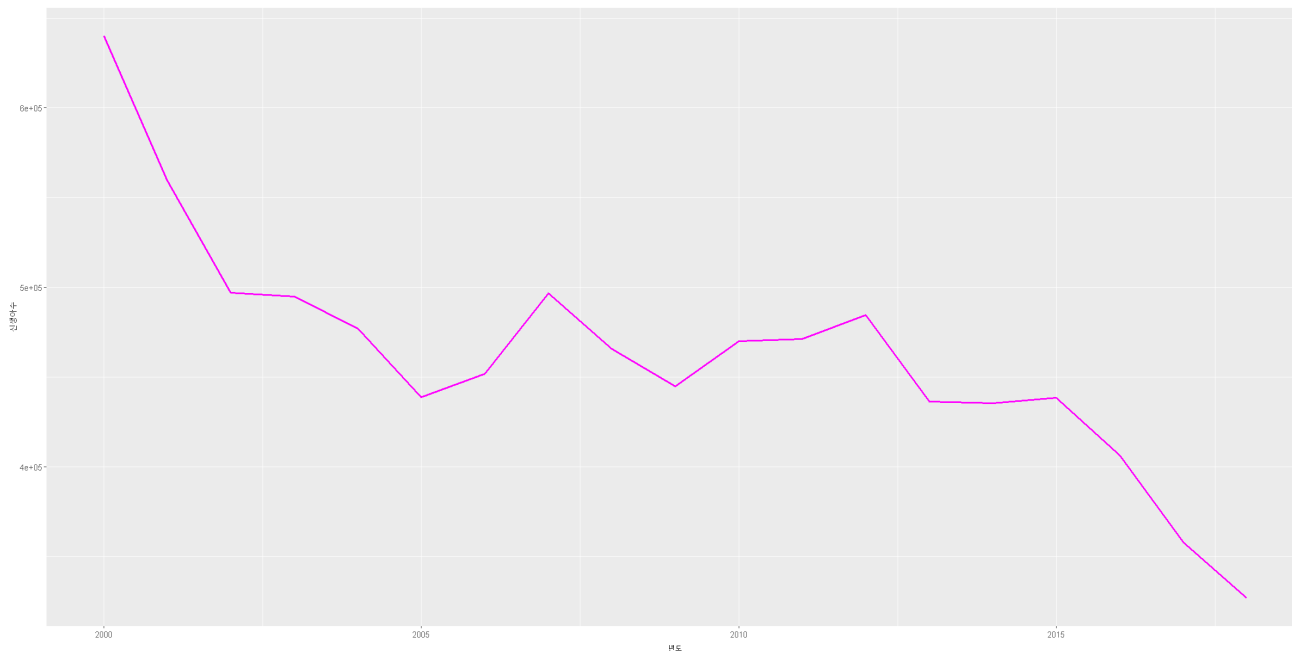
linetype 값의 종류

- 1: 직선
- 2: 끊긴선
- 3: 점선
- 4: 끊긴선 + 점선 교차 표시

기본 그래프 표시하기

```
# 그래프 표시하기
ggplot(data=년도별_신생아_수) +
  geom_line(aes(x=년도, y=신생아수),
            color="#ff00ff",
            linetype = 1,
            size=1)
```

▶ 출력결과



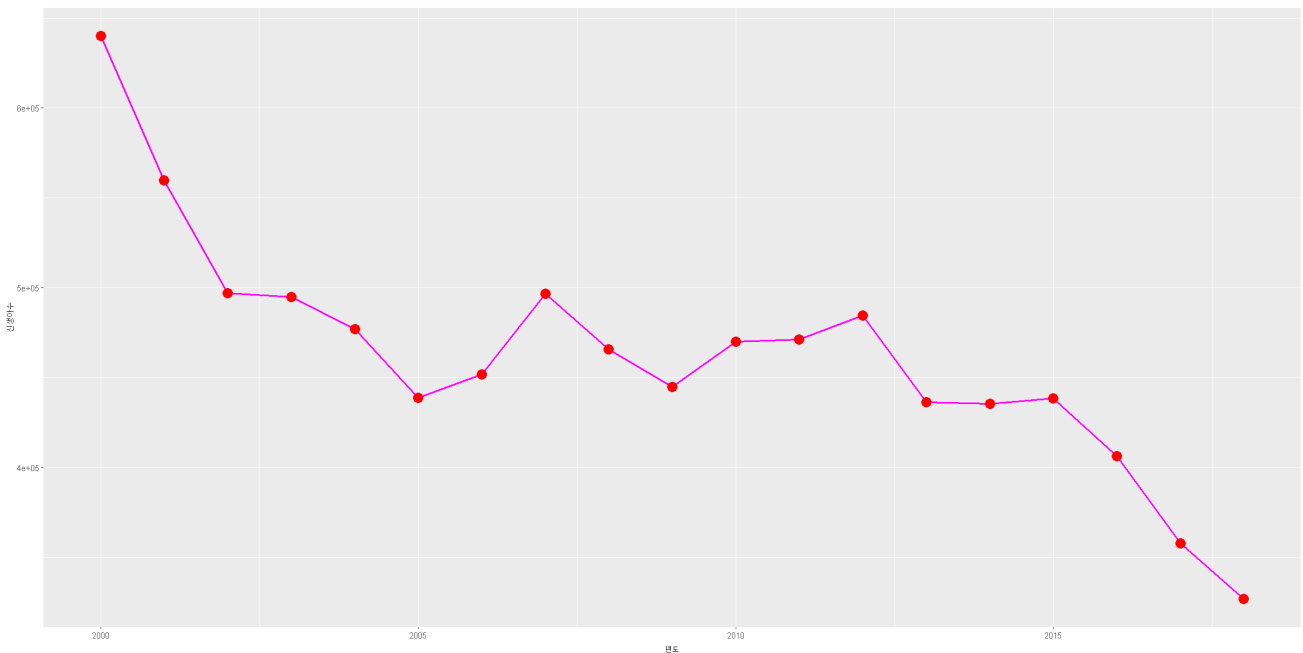
그래프에 마커 추가하기

`geom_point()` 함수를 사용한다.

범주를 사용한다면 별도의 옵션 함수를 지정하여 범주를 개별적으로 설정할 수 도 있다.

```
ggplot(data=년도별_신생아_수) +
  geom_line(aes(x=년도, y=신생아수),
            color="#ff00ff",
            linetype = 1,
            size=1) +
  geom_point(aes(x=년도, y=신생아수),
             size=5,
             color="red")
```

▶ 출력결과



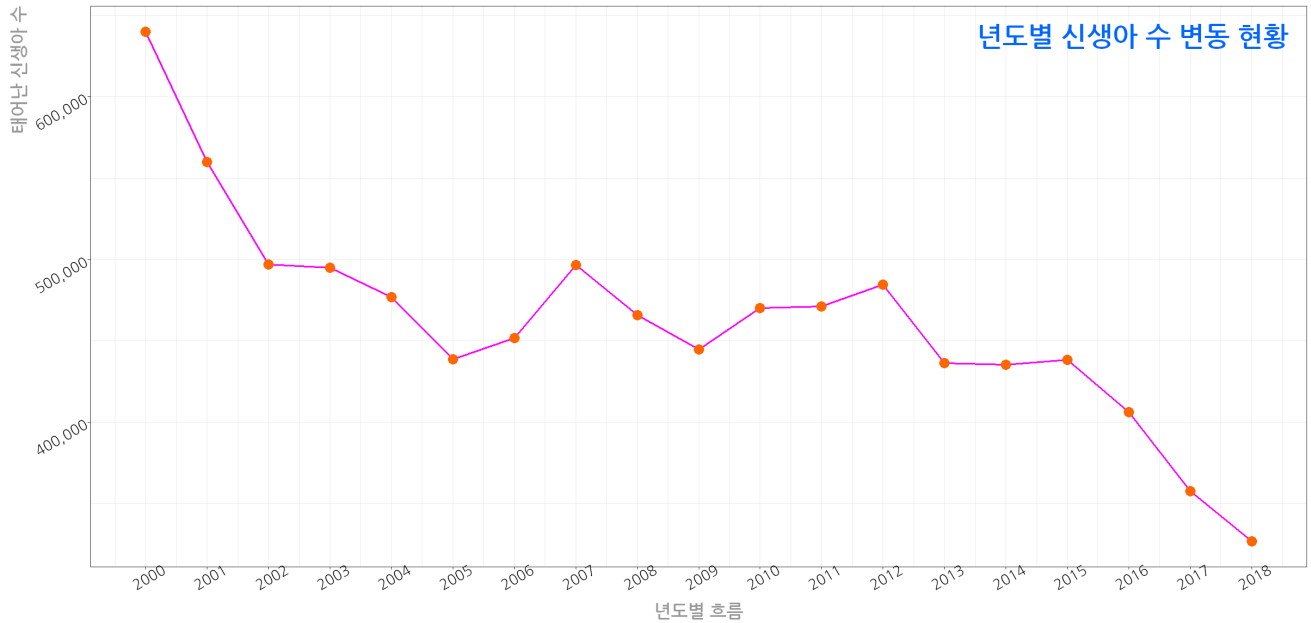
그래프 옵션 적용하기

```
# 데이터프레임 설정
ggplot(data=년도별_신생아_수) +
  # 선 그래프 x,y축 및 색상, 종류, 굵기 설정
  geom_line(aes(x=년도, y=신생아수), color="#ff00ff", linetype = 1, size=1) +
  # 마커 표시하기
  geom_point(aes(x=년도, y=신생아수), size=5, color="#ff6600") +
  # 배경을 흰색으로 설정
  theme_bw() +
  # 그래프 타이틀 설정
  ggtitle("년도별 신생아 수 변동 현황") +
  # x축 제목 설정
  xlab("년도별 흐름") +
  # y축 제목 설정
  ylab("태어난 신생아 수") +
  # x축 간격 설정 -> seq(x,y): x~y까지 1씩 증가하는 벡터를 생성하여 반환한다.
  scale_x_continuous(breaks=seq(min(년도별_신생아_수$년도),
                                max(년도별_신생아_수$년도))) +

  # y축 범위 설정
  scale_y_continuous(limits=c(min(년도별_신생아_수$신생아수),
                               max(년도별_신생아_수$신생아수)),
                    labels = scales::comma) +

  # 각 텍스트의 색상, 크기, 각도, 글꼴 설정
  theme(plot.title=element_text(family="NanumGothic", color="#0066ff", size=30,
                                face="bold", hjust=0.98, vjust=-8),
        axis.title.x=element_text(family="NanumGothic", color="#999999", size=20,
                                face="bold"),
        axis.title.y=element_text(family="NanumGothic", color="#999999", size=20,
                                face="bold", hjust=1),
        axis.text.x=element_text(family="NanumGothic", color="#000000", size=16,
                                angle=30),
        axis.text.y = element_text(family="NanumGothic", color="#000000", size=16,
                                angle=30))
```

▶ 출력결과



2) 다중 선 그래프

예제를 위한 데이터셋 준비

2005~2018년도 월별 전국 교통사고 현황 데이터 (출처: 국가통계포털)

```
년도별_교통사고 <- read.csv("http://itpaper.co.kr/demo/r/traffic.csv",
                             stringsAsFactors=F, fileEncoding="euc-kr")
년도별_교통사고
```

▶ 출력결과

A data.frame: 168 x 5

년도	월	발생건수	사망자수	부상자수
<int>	<int>	<int>	<int>	<int>
2005	1	15494	504	25413
2005	2	13244	431	21635
2005	3	16580	477	25550
2005	4	17817	507	28131
2005	5	19085	571	29808
2005	6	18092	476	28594
2005	7	18675	528	29984
2005	8	19035	562	31603
2005	9	18759	577	29831
2005	10	19757	639	31597
2005	11	19129	574	30337
2005	12	18504	530	29750
2006	1	14971	420	24533
2006	2	14270	373	22903
2006	3	16767	465	26013

년도	월	발생건수	사망자수	부상자수
<int>	<int>	<int>	<int>	<int>
2006	4	17948	469	28725
2006	5	19140	531	30279
2006	6	17435	455	27032
2006	7	18634	516	29978
2006	8	18794	585	30882
2006	9	19293	580	30186
2006	10	19100	651	30715
2006	11	19877	701	31270
2006	12	17516	581	27713
2007	1	14914	468	23975
2007	2	14696	446	23717
2007	3	18166	476	28811
2007	4	18055	460	28555
2007	5	19264	516	30532
2007	6	18310	538	28662
...
2016	7	18955	358	28586
2016	8	18398	336	28017
2016	9	17883	375	26761
2016	10	19918	440	29635
2016	11	19234	416	28520
2016	12	18869	408	28192
2017	1	16970	353	26099
2017	2	14832	280	22323
2017	3	17047	295	25046
2017	4	17717	293	26530
2017	5	18502	366	27268
2017	6	18047	315	26454
2017	7	18158	357	27362
2017	8	18682	353	28162
2017	9	19891	419	29371
2017	10	18863	420	28698
2017	11	19377	379	28472
2017	12	18249	355	27044
2018	1	17026	304	25438
2018	2	16208	275	24630

년도	월	발생건수	사망자수	부상자수
<int>	<int>	<int>	<int>	<int>
2018	3	17022	310	25015
2018	4	17992	303	26643
2018	5	18636	309	27834
2018	6	18082	266	26574
2018	7	18699	315	28104
2018	8	18335	357	27749
2018	9	18371	348	27751
2018	10	19738	373	28836
2018	11	19029	298	28000
2018	12	18010	323	26463

년도별 교통사고 합계 구하기

년도별로 집계

```
교통사고_집계 <- 년도별_교통사고 %>%
  group_by(년도) %>%
  summarise(발생건수=sum(발생건수, na.rm=TRUE),
            사망자수=sum(사망자수, na.rm=TRUE),
            부상자수=sum(부상자수, na.rm=TRUE))

교통사고_집계
```

▶ 출력결과

A tibble: 14 × 4

년도	발생건수	사망자수	부상자수
<int>	<int>	<int>	<int>
2005	214171	6376	342233
2006	213745	6327	340229
2007	211662	6166	335906
2008	215822	5870	338962
2009	231990	5838	361875
2010	226878	5505	352458
2011	221711	5229	341391
2012	223656	5392	344565
2013	215354	5092	328711
2014	223552	4762	337497
2015	232035	4621	350400
2016	220917	4292	331720
2017	216335	4185	322829
2018	217148	3781	323037

값의 종류 별로 그래프 그리기

`geom_line()` 함수를 여러 번 사용하여 다중 선 그래프를 표현한다.

여기서는 전체 그래프 생성 결과를 별도의 변수에 할당하여 출력하고 있다.

```
# 그래프 생성
graph <- ggplot(data=교통사고_집계) +
  # 선 그래프 x,y축 및 색상, 종류, 굵기 + 마커
  geom_line(aes(x=년도, y=발생건수), color="#ff6600", linetype=1, size=1.5) +
  geom_line(aes(x=년도, y=사망자수), color="#ff00ff", linetype=2, size=1) +
  geom_line(aes(x=년도, y=부상자수), color="#0000ff", linetype=3, size=1) +
  # 마커
  geom_point(aes(x=년도, y=발생건수), size=5, color="#ff6600") +
  geom_point(aes(x=년도, y=사망자수), size=3, color="#ff00ff") +
  geom_point(aes(x=년도, y=부상자수), size=3, color="#0000ff") +
  # 배경을 흰색으로 설정
  theme_bw() +
  # 그래프 타이틀 설정
  ggtitle("년도별 교통사고 추이") +
  # x축 제목 설정
  xlab("년도") +
  # y축 제목 설정
  ylab("교통사고(건) / 사망자, 부상자(명)") +
  # x축 간격 설정 -> seq(x,y): x~y까지 1씩 증가하는 벡터를 생성하여 반환한다.
  scale_x_continuous(breaks=seq(min(교통사고_집계$년도),
                                max(교통사고_집계$년도))) +
  # y축 간격 및 데이터에 대한 세자리 콤마 적용
  scale_y_continuous(breaks=seq(0, 400000, 50000), labels=scales::comma) +
  # 각 텍스트의 색상, 크기, 각도, 글꼴 설정
  theme(plot.title=element_text(family="NanumGothic", color="#0066ff",
                                size=30, face="bold", hjust=0.98, vjust=-8),
        axis.title.x=element_text(family="NanumGothic", color="#999999",
                                size=20, face="bold"),
        axis.title.y=element_text(family="NanumGothic", color="#999999",
                                size=20, face="bold", hjust=1),
        axis.text.x=element_text(family="NanumGothic", color="#000000",
                                size=16, angle=30),
        axis.text.y=element_text(family="NanumGothic", color="#000000",
                                size=16, angle=30))

graph
```

▶ 출력결과

