Data Visualization

우석진(명지대 경제/응용데이터사이언스스)

개요 1

- R에 내장된 그래픽 기능을 사용하는 것도 좋지만
- ggplot2 패키지를 이용하는 것이 보통이다
- 따라서 중복 투자 없이 효율적으로 R을 사용하기 위해서는 ggplot2를 처음부터 사용하는 것이 좋다.

개요 2

- 기본적인 구조는 다음과 같다.

1. 먼저, 데이터가 있어야 한다.
2. 자료 중 어떤 변수를 사용할지를 시각화 시킬지 결정해야 한다
   - 이 과정을 mapping 이라고 한다
3. 어떤 모양을 통해 시각화를 할 것인지를 정해야 한다.
   - 이를 geom(etrics) 이라고 부른다.
4. 이렇게 그려진 그림에 축, 스케일, 색 팔레트, 범례 등을 설정해주면 된다.

개요 3

tidy data	• 자료
mapping	• x축, y축 • fill, colour 등
geom	• bar, point, line 등 • text, rug, density, smooth, jitter 등
axis & scale	• coord_cartesian 등 • scale_x_continuous 등
label & guides	• xlab, ylab, lab 등 • theme, guies 등

tidy data

- ggplot으로 그림을 편하게 그리기 위해서는 R에서는 tidy data라고 부르는 형태의 자료가 필요하다.
- Stata에서는 자료의 형태를 wide-form 혹은 long-form 으로 부른다.
- wide-form은 가로로 뚱뚱한 자료 형태이다.

tidy data 2

region	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
서울특별시	1.275	1.111	1.006	1.014	1.015	0.932	0.980	1.068	1.010	0.962	1.015	1.014	1.059	0.968	0.983	1.001	0.940	0.836
부산광역시	1.235	1.103	0.975	0.988	0.953	0.887	0.915	1.024	0.980	0.940	1.045	1.078	1.135	1.049	1.090	1.139	1.095	0.976
대구광역시	1.378	1.216	1.076	1.116	1.087	1.001	1.011	1.137	1.072	1.029	1.109	1.146	1.217	1.127	1.169	1.216	1.186	1.067
인천광역시	1.473	1.324	1.185	1.213	1.158	1.075	1.116	1.257	1.186	1.143	1.214	1.232	1.301	1.195	1.212	1.216	1.144	1.007
광주광역시	1.636	1.421	1.264	1.278	1.203	1.105	1.152	1.262	1.198	1.137	1.223	1.234	1.295	1.170	1.199	1.207	1.168	1.053
대전광역시	1.501	1.330	1.207	1.221	1.181	1.107	1.158	1.274	1.215	1.156	1.205	1.261	1.315	1.234	1.250	1.277	1.192	1.075
울산광역시	1.633	1.423	1.242	1.280	1.241	1.186	1.242	1.403	1.338	1.308	1.369	1.393	1.481	1.391	1.437	1.486	1.418	1.261
세종특별자치시													1.597	1.435	1.354	1.893	1.821	1.668
경기도	1.628	1.437	1.305	1.321	1.280	1.183	1.239	1.361	1.285	1.226	1.309	1.314	1.355	1.226	1.241	1.272	1.194	1.069
강원도	1.600	1.413	1.317	1.279	1.261	1.188	1.202	1.356	1.253	1.248	1.313	1.338	1.374	1.249	1.248	1.311	1.237	1.123
충청북도	1.583	1.426	1.294	1.270	1.272	1.195	1.233	1.398	1.319	1.317	1.402	1.428	1.485	1.365	1.363	1.414	1.358	1.235
충청남도	1.698	1.532	1.361	1.358	1.357	1.267	1.356	1.506	1.444	1.408	1.479	1.496	1.571	1.442	1.421	1.480	1.395	1.276
전라북도	1.595	1.426	1.275	1.274	1.239	1.184	1.213	1.380	1.305	1.279	1.374	1.405	1.440	1.320	1.329	1.352	1.251	1.151
전라남도	1.750	1.566	1.391	1.389	1.360	1.290	1.337	1.542	1.449	1.445	1.537	1.568	1.642	1.518	1.497	1.549	1.466	1.325
경상북도	1.578	1.402	1.232	1.253	1.203	1.173	1.208	1.369	1.313	1.274	1.377	1.434	1.489	1.379	1.408	1.464	1.396	1.256
경상남도	1.586	1.417	1.272	1.290	1.266	1.189	1.254	1.434	1.368	1.323	1.413	1.446	1.503	1.367	1.409	1.437	1.358	1.227
제주특별자치도	1.783	1.564	1.394	1.438	1.365	1.310	1.372	1.489	1.386	1.378	1.463	1.487	1.598	1.427	1.481	1.477	1.432	1.305

tidy data 3

- 아래 그림은 위 자료를 long-form, 이른바 tidy data로 전환한 것
- year 변수, region 변수, 옆에 합계출산율 fertility 의 값이 들어간다.

tidy data 4

year	region	fertility
2000	강원도	1.600
2001	강원도	1.413
2002	강원도	1.317
2003	강원도	1.279
2004	강원도	1.261
2005	강원도	1.188
2006	강원도	1.202
2007	강원도	1.356
2008	강원도	1.253
2009	강원도	1.248
2010	강원도	1.313
2011	강원도	1.338
2012	강원도	1.374
2013	강원도	1.249
2014	강원도	1.248
2015	강원도	1.311
2016	강원도	1.237
2017	강원도	1.123
2000	경기도	1.628
2001	경기도	1.437
2002	경기도	1.305
2003	경기도	1.321
2004	경기도	1.280

tidy data 만들기

- 필요한 라이브러리인 tidyverse를 장착하자
- 엑셀 자료인 fertility.xlsx 를 읽어 들이자

tidy data 만들기 2

- head()함수를 통해서 보면, fertility가 wide-form 자료임을 알 수 있다.

```
# A tibble: 3 × 5
  region      `2000` `2001` `2002` `2003`
  <chr>        <dbl>  <dbl>  <dbl>  <dbl>
1 서울특별시    1.27   1.11  1.01   1.01
2 부산광역시    1.24   1.10  0.975  0.988
3 대구광역시    1.38   1.22  1.08   1.12
```

tidy data 만들기 3

- ggplot으로 그림을 편하기 그리기 위해서는 tidy data로 전환을 해주어야 한다.

tidy data 만들기 4

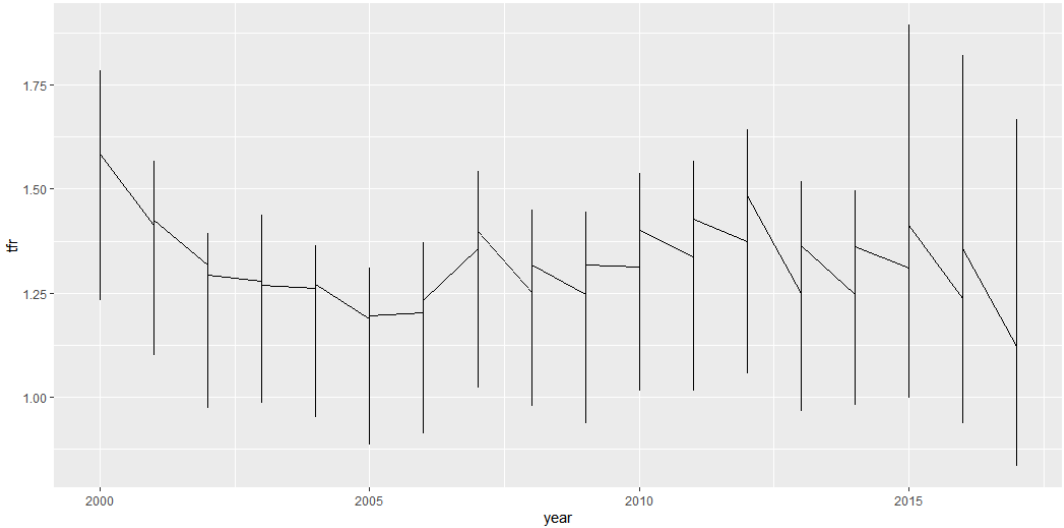
```
# A tibble: 5 × 3
  region  year   tfr
  <chr>   <dbl> <dbl>
1 강원도  2000  1.6
2 강원도  2001  1.41
3 강원도  2002  1.32
4 강원도  2003  1.28
5 강원도  2004  1.26
```

tidy data 만들기 5

- 그러면 아래와 같이 long-form, 즉 tidy data로 전환할 수 있다.
- 이제 우리는 tidy data를 가지고 있습니다.
- 모든 data는 이렇게 tidy form으로 관리를 해줘야 합니다(엑셀에서도 마찬가지)

추세선

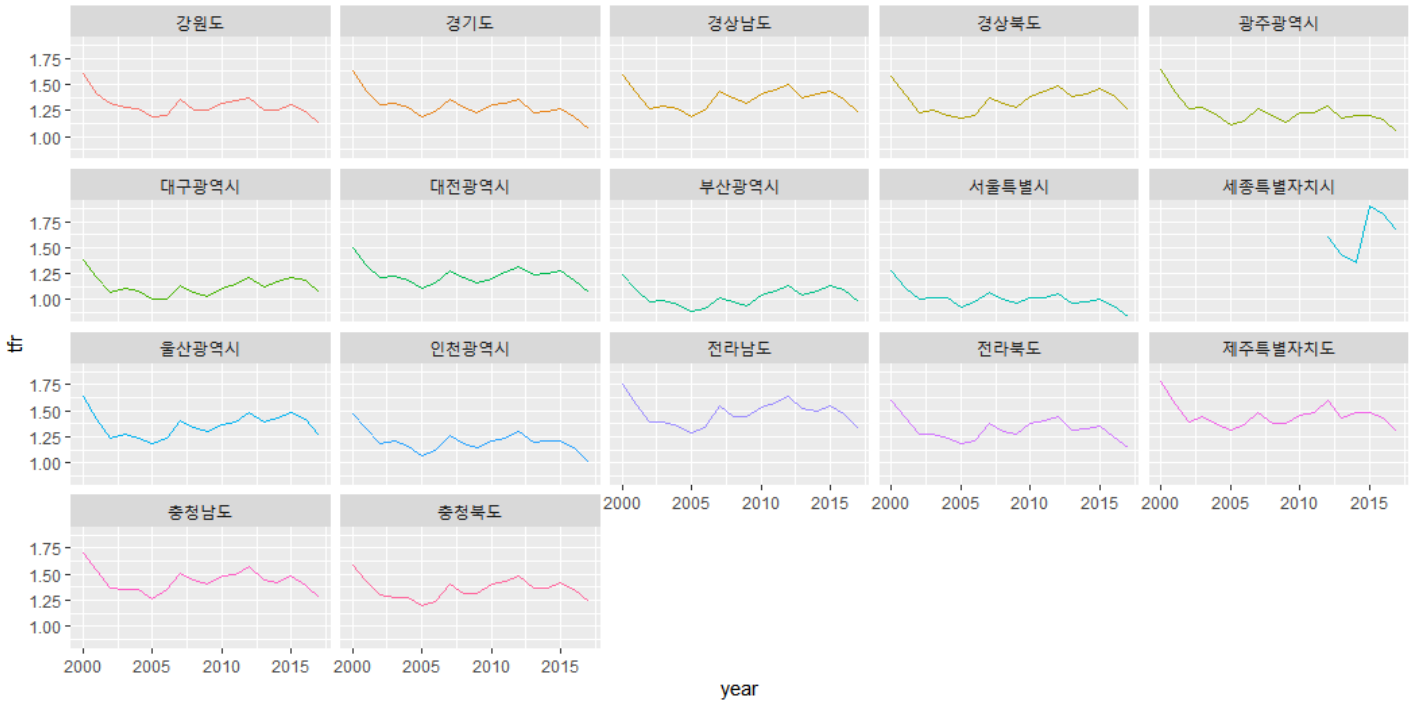
우리가 x축에는 연도, y축에는 합계출산율을 내려고 함



추세선 2

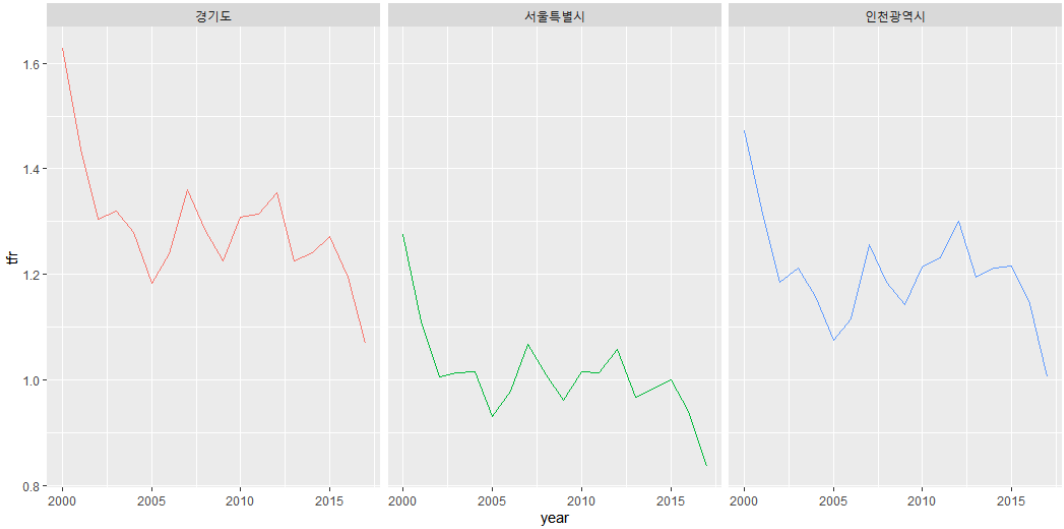
- 한 해에 많은 지역이 대응되고 있기 때문

facet_wrap


강원도
경기도
경상남도
경상북도
광주광역시
대구광역시
대전광역시
부산광역시
서울특별시
세종특별자치시
울산광역시
인천광역시
전라남도
전라북도
제주특별자치도
충청남도
충청북도
1.75
1.50
1.25
1.00
2000
2005
2010
2015
tfr
year

추세선 3

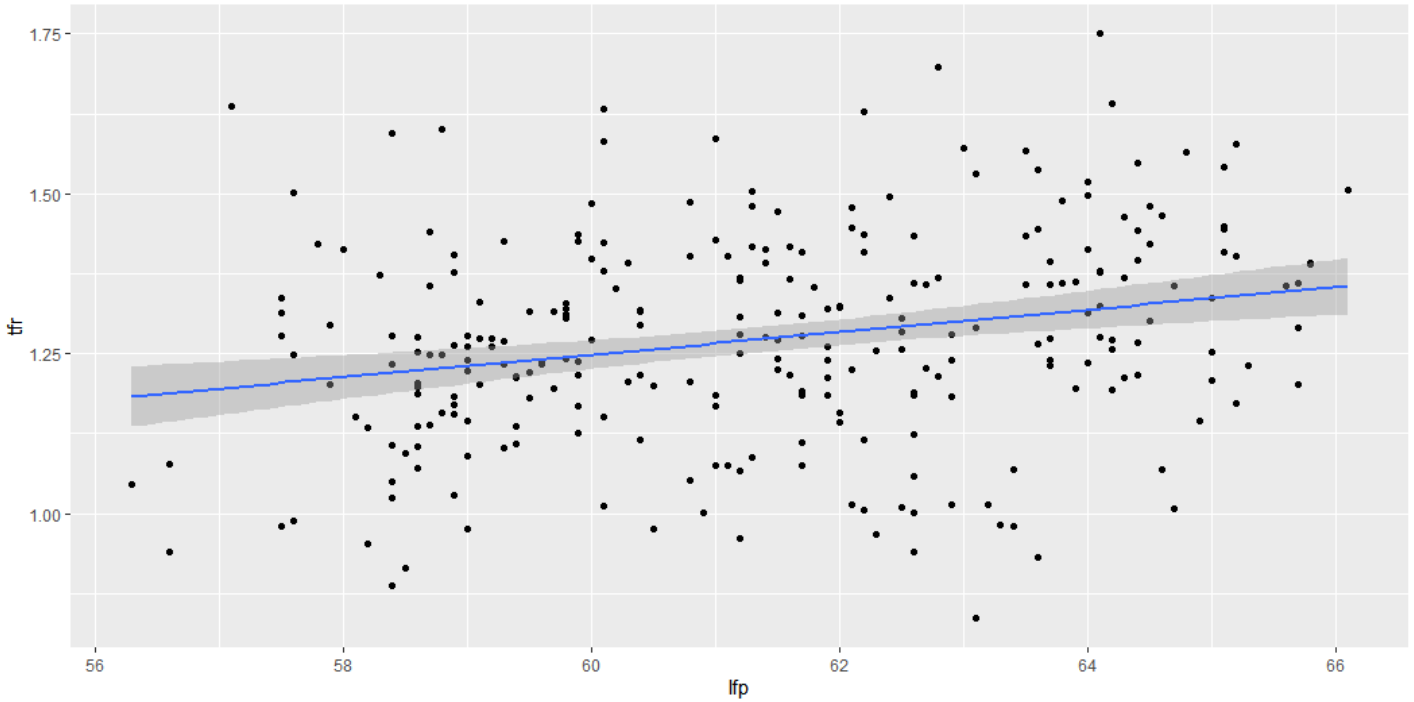
서울, 경기, 인천 만 뽑아서 그리는 경우

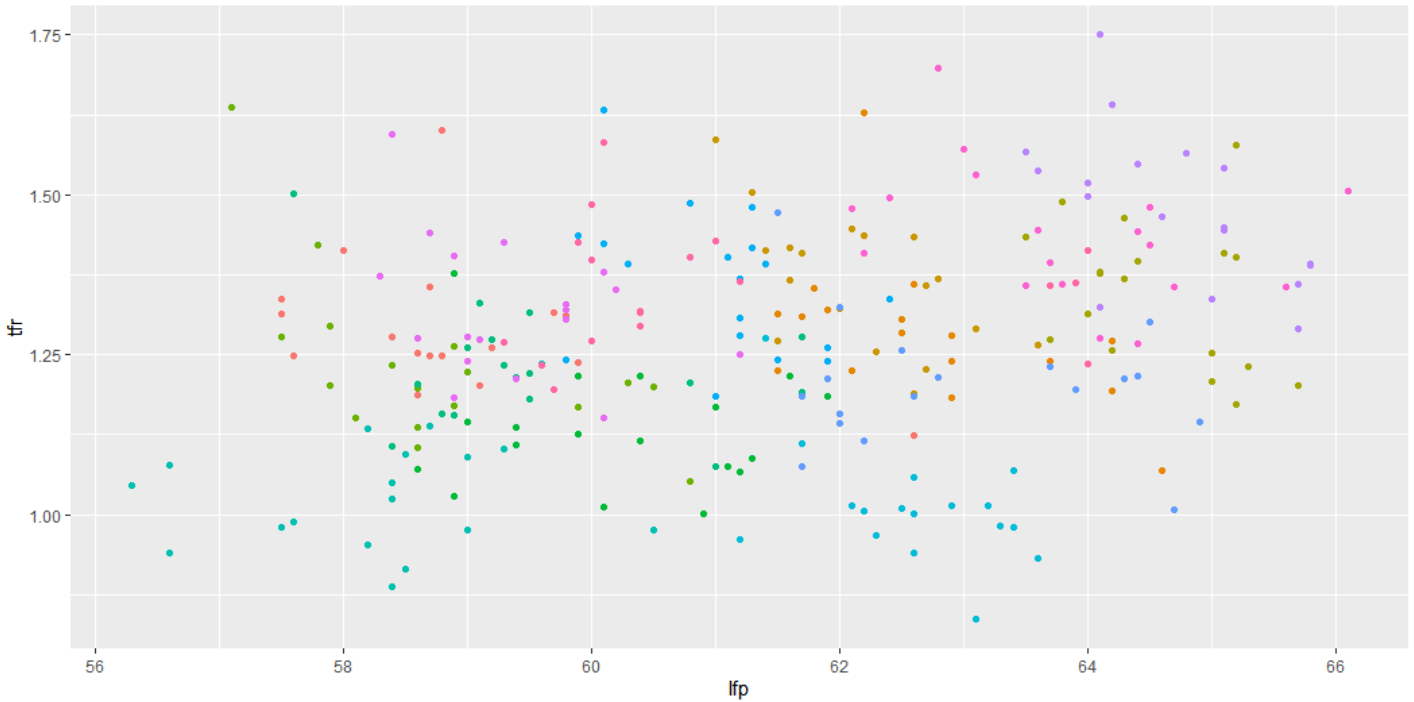


출산율과 경제활동참가율

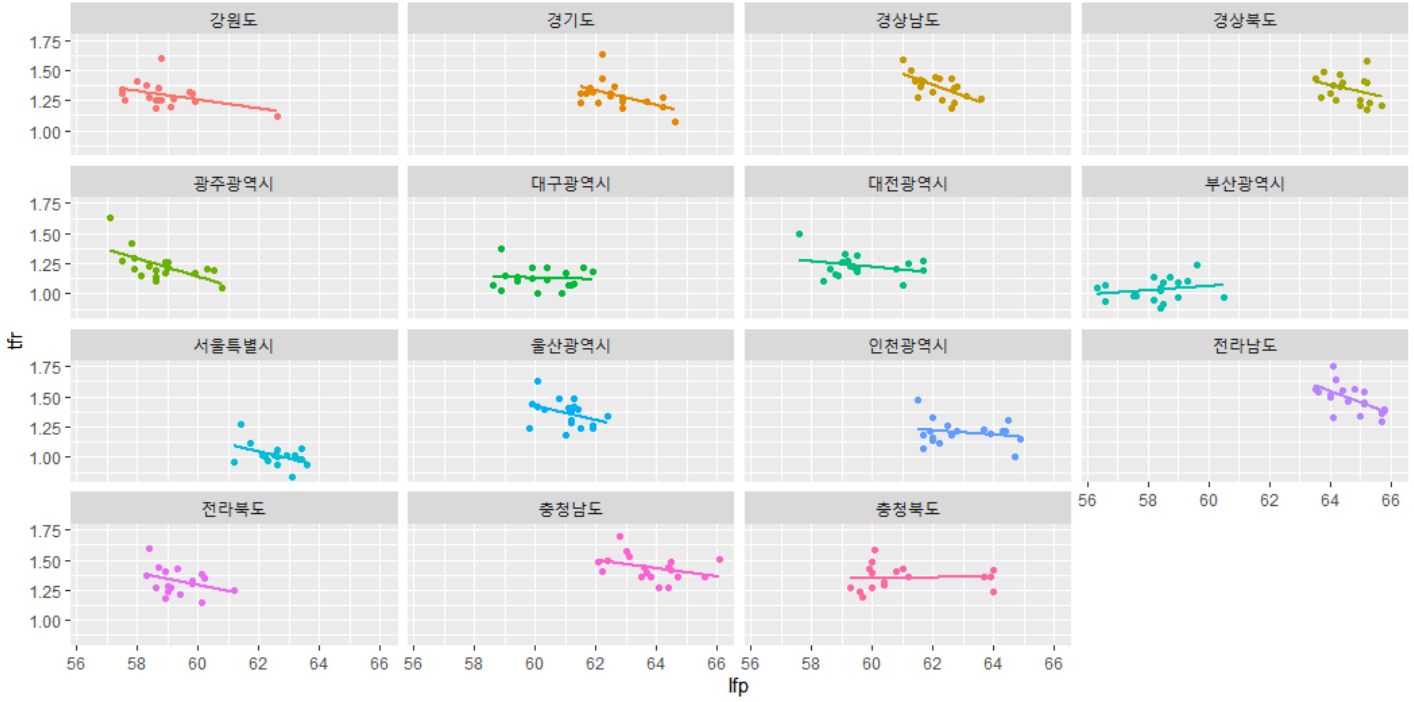
- 새로운 자료를 하나 로딩하자.
- 연도별, 지역별 합계출산율, 경제활동참가율 자료임

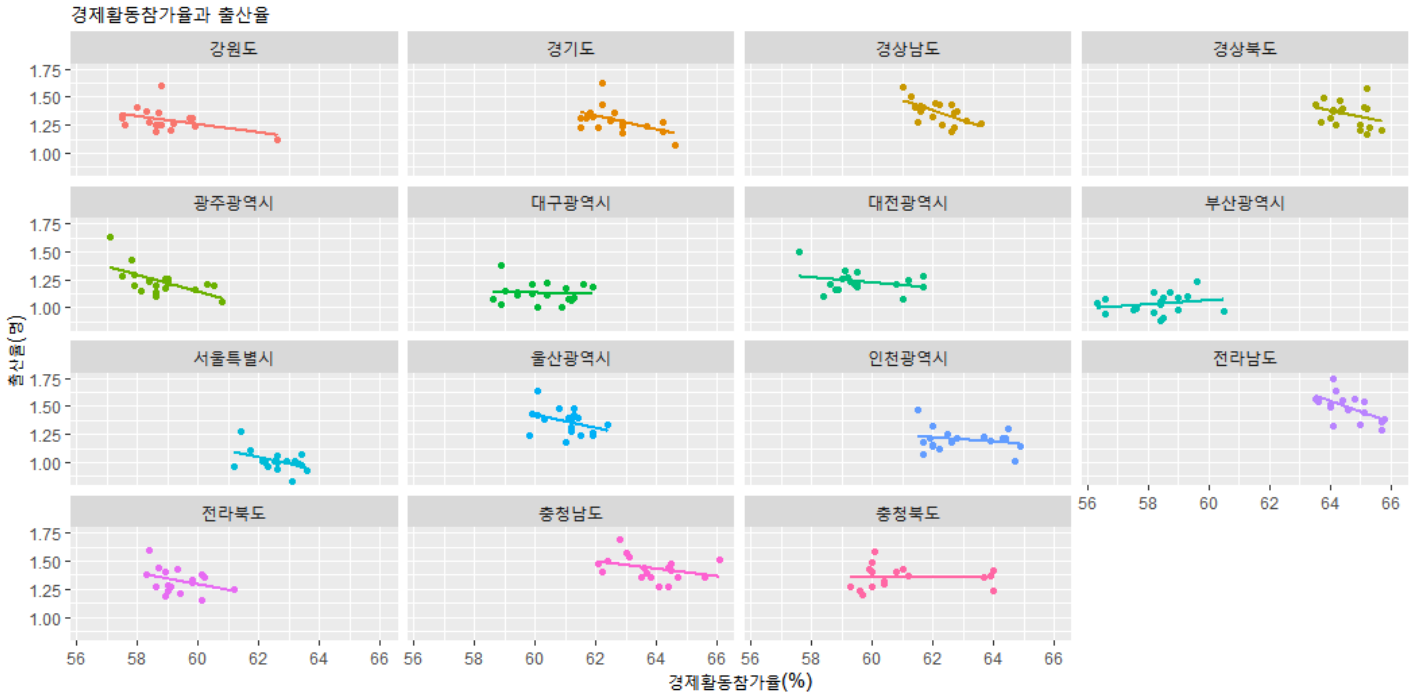
산포도

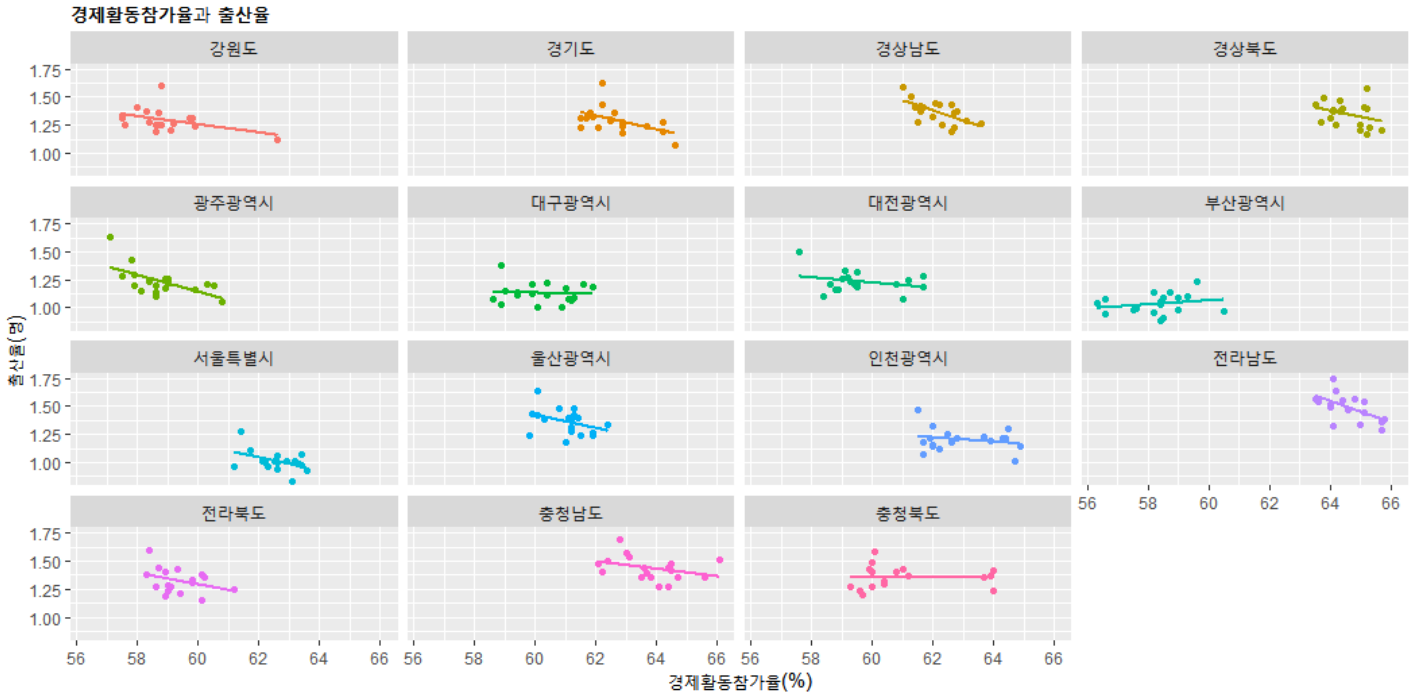



1.75
1.50
1.25
1.00
tfr
56
58
60
62
64
66
lfp

Simpson’s paradox




경제활동참가율과 출산율
강원도
경기도
경상남도
경상북도
광주광역시
대구광역시
대전광역시
부산광역시
서울특별시
울산광역시
인천광역시
전라남도
전라북도
충청남도
충청북도
출산율(명)
1.75
1.50
1.25
1.00
56
58
60
62
64
66
경제활동참가율(%)

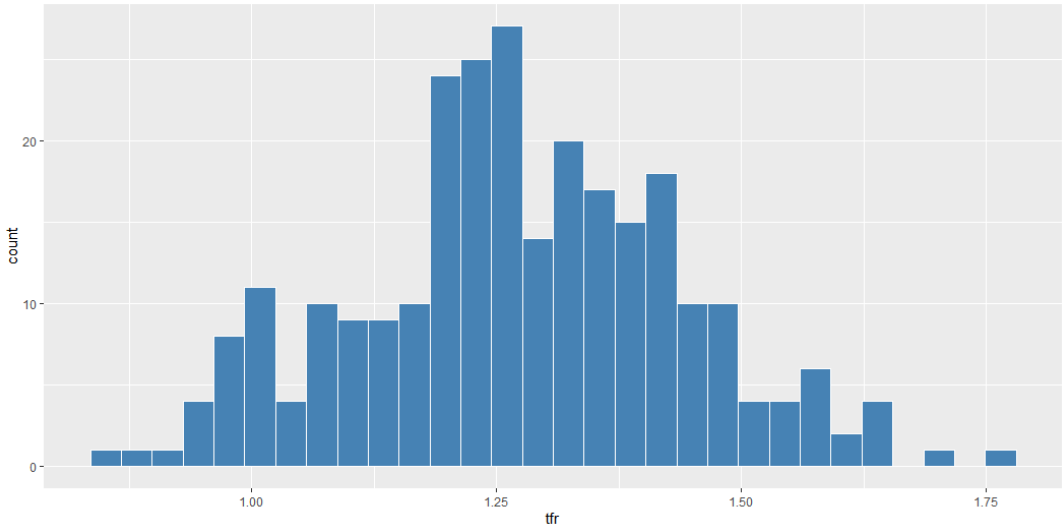

경제활동참가율과 출산율
강원도
경기도
경상남도
경상북도
광주광역시
대구광역시
대전광역시
부산광역시
서울특별시
울산광역시
인천광역시
전라남도
전라북도
충청남도
충청북도
출산율(명)
1.75
1.50
1.25
1.00
56
58
60
62
64
66
경제활동참가율(%)

히스토그램

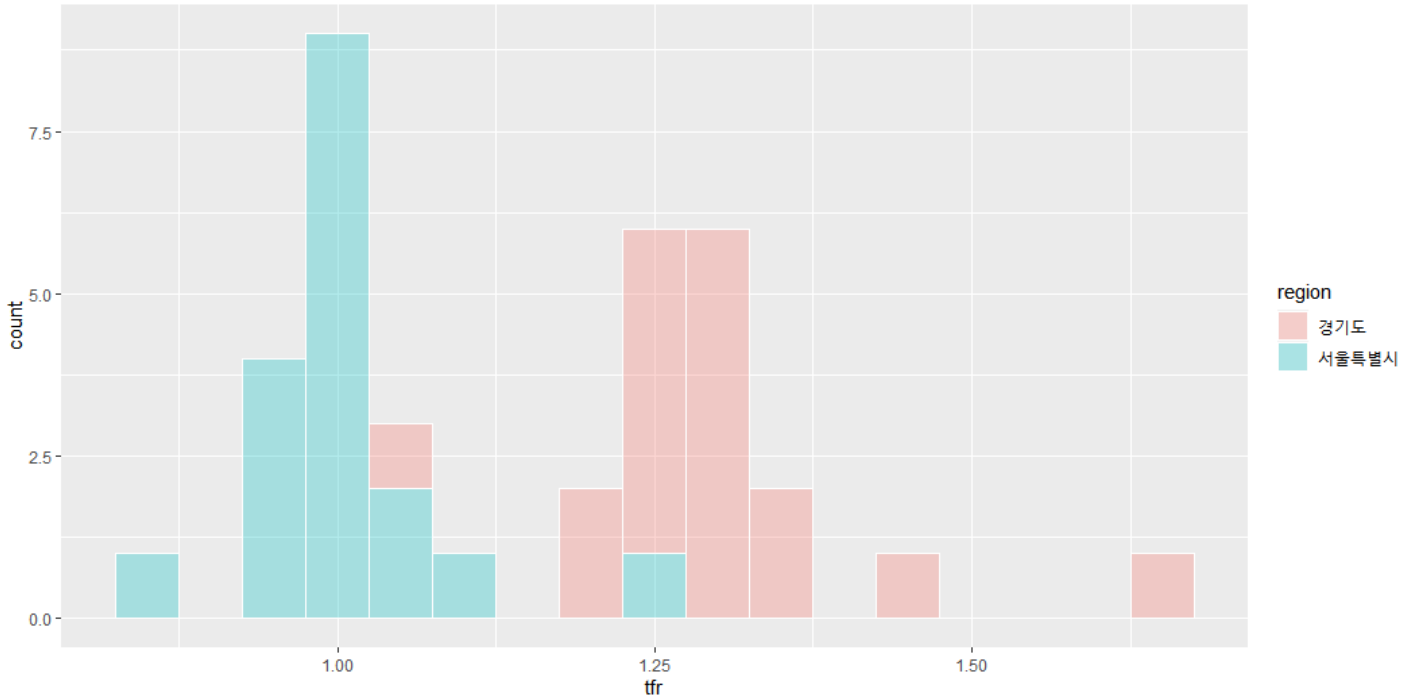
히스토그램

확률변수의 분포를 살펴보는 가장 편한 방법은 히스토그램을 그리는 것이다.

ggplot에서는 geom_histogram을 사용하게 된다.



특정지역의 히스토그램



요약통계량

요약통계량

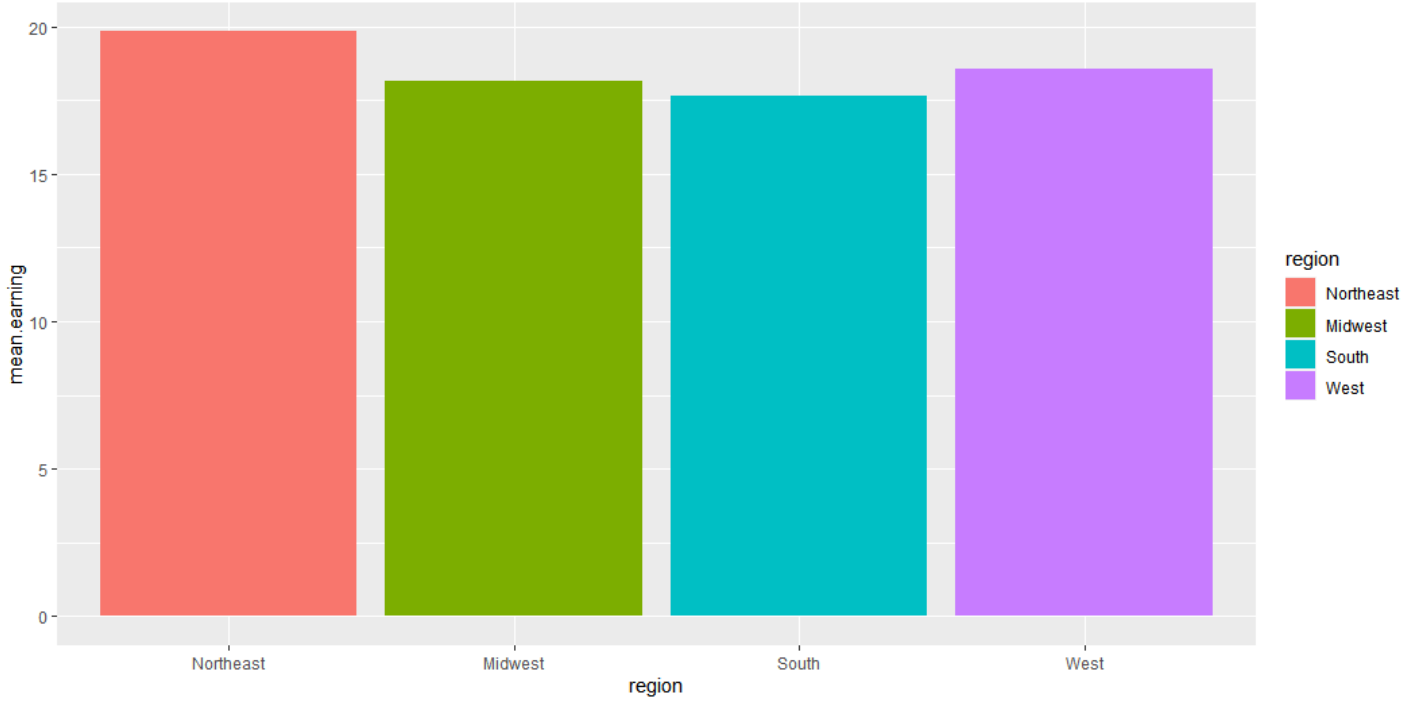
- ggplot의 편리한 점 중에 하나는 dplyr 패키지의 pipe 연산과 연계해서 요약통계량을 그릴 수 있다는 점
- 실습을 위해 AER 패키지의 CPSSW8 자료를 불러 들이자.

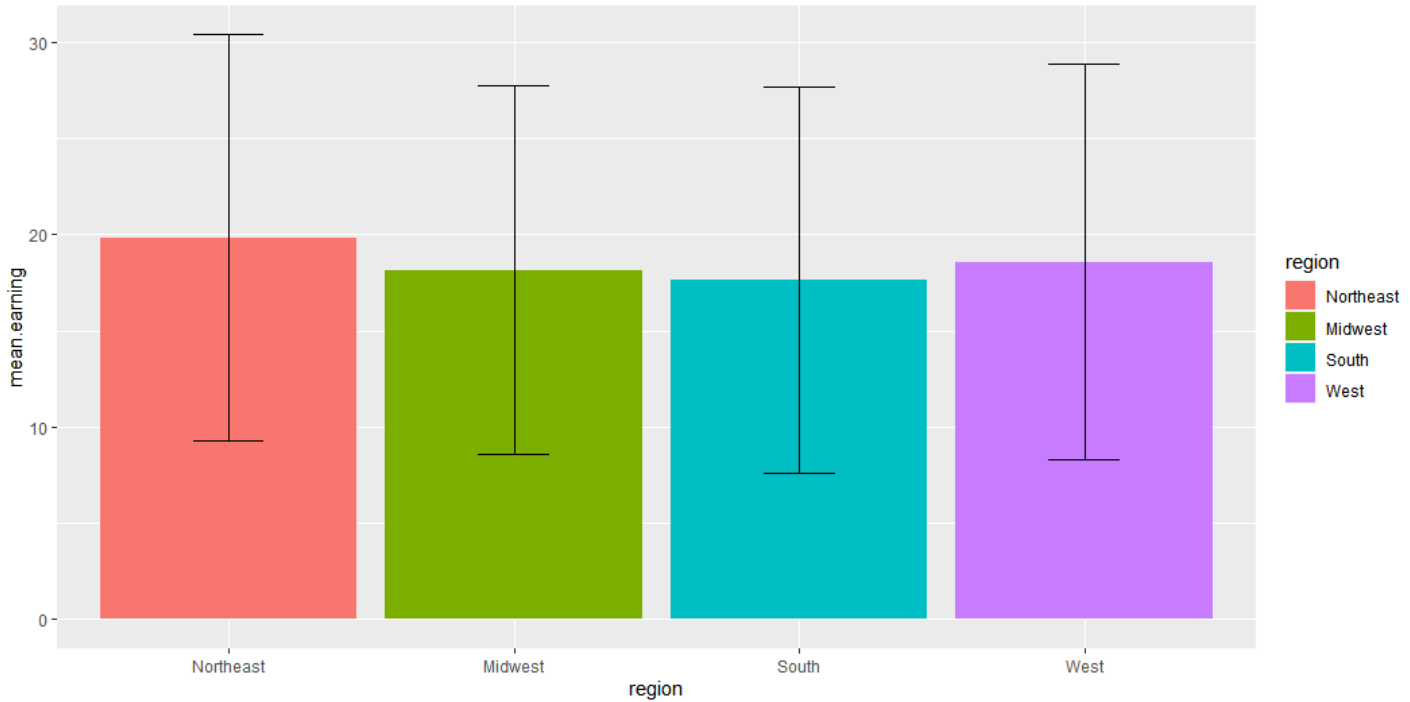
요약통계량 2

- 지역 region별 소득 수준을 살펴보자.

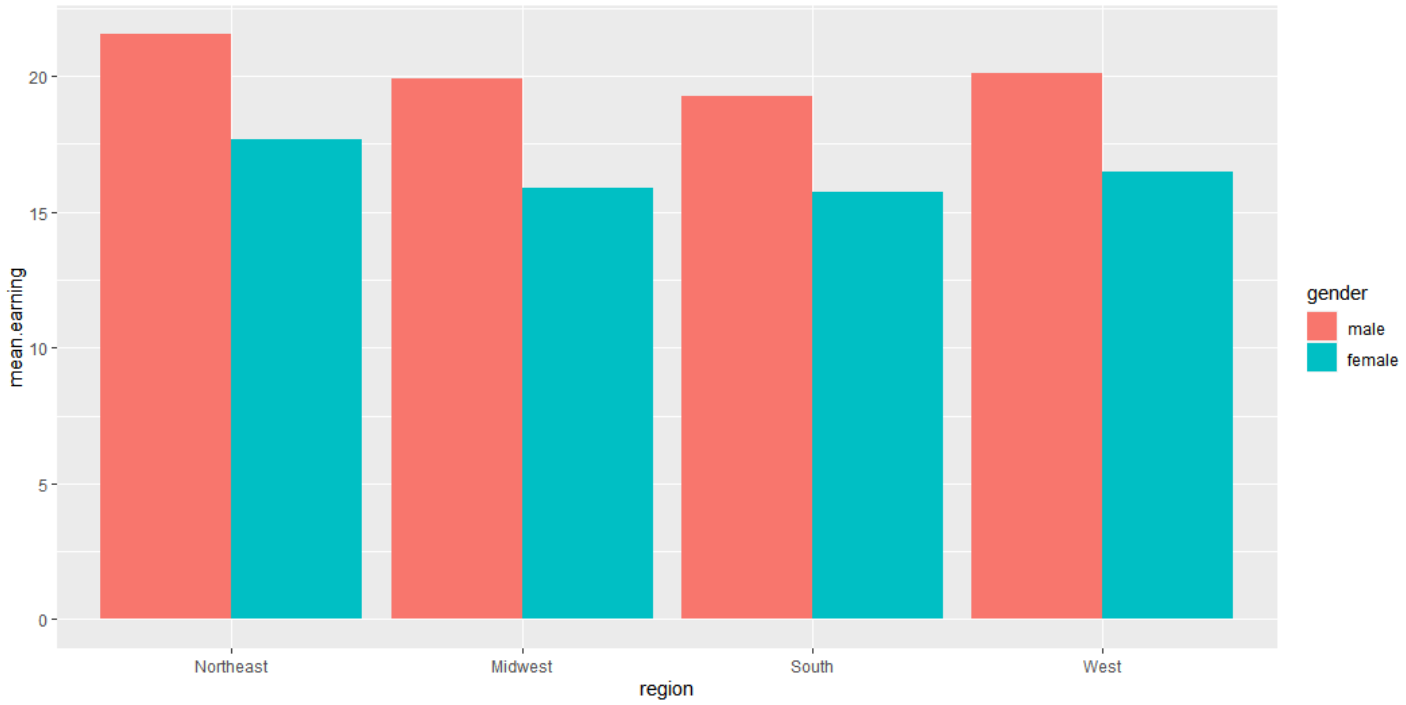
```
# A tibble: 3 × 4
  region         N mean.earning sd.earning
  <fct>      <int>        <dbl>       <dbl>
1 Northeast 12371         19.8        10.6
2 Midwest   15136         18.1         9.59
3 South     18963         17.6        10.0
```

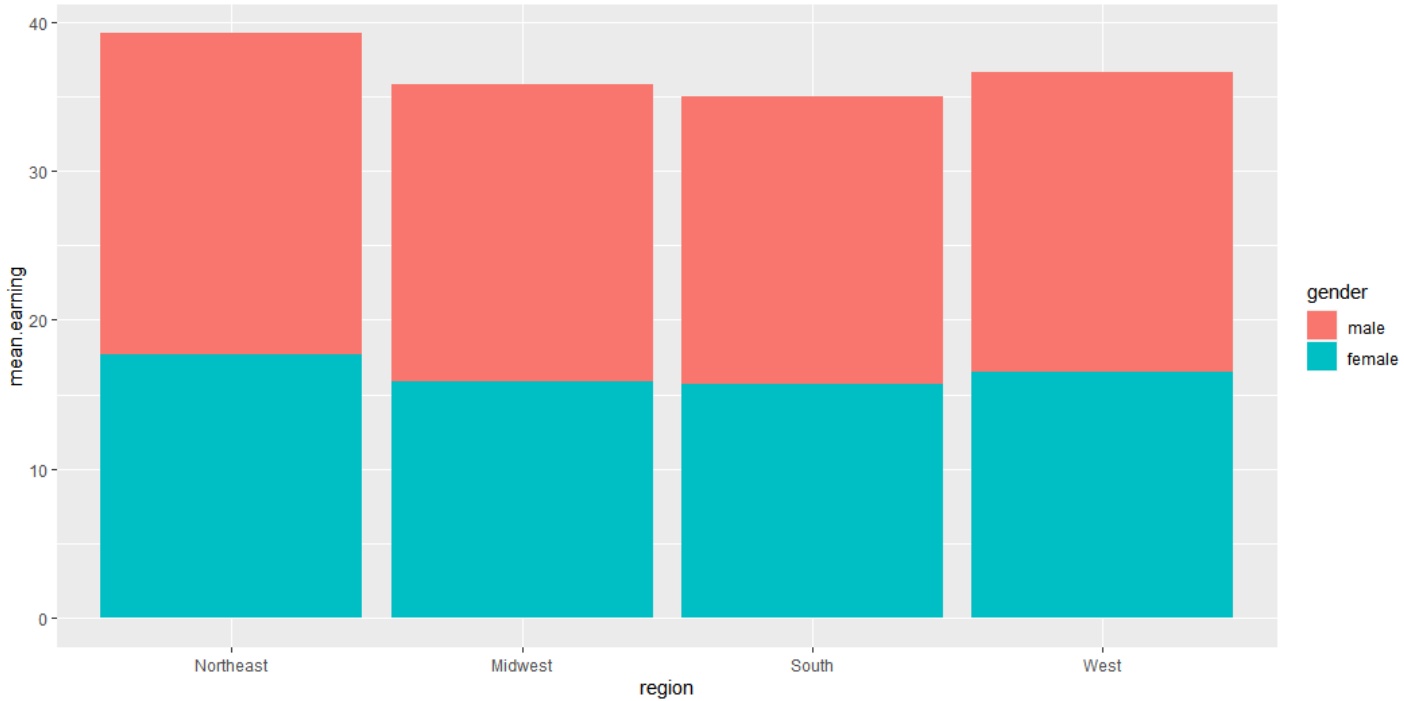
요약통계량 (bar graph,지역별)




30
20
10
0
mean.earning
Northeast
Midwest
South
West
region
region
Northeast
Midwest
South
West

요약통계량 (bar graph,지역x성 별)




40
30
20
10
0
mean.earning
Northeast
Midwest
South
West
region
gender
male
female

bar graphs applications

- 지역별 인플레이션 데이터

```
inf <- readxl::read_xlsx("inflation.xlsx")
```

tidy 형태로 전환

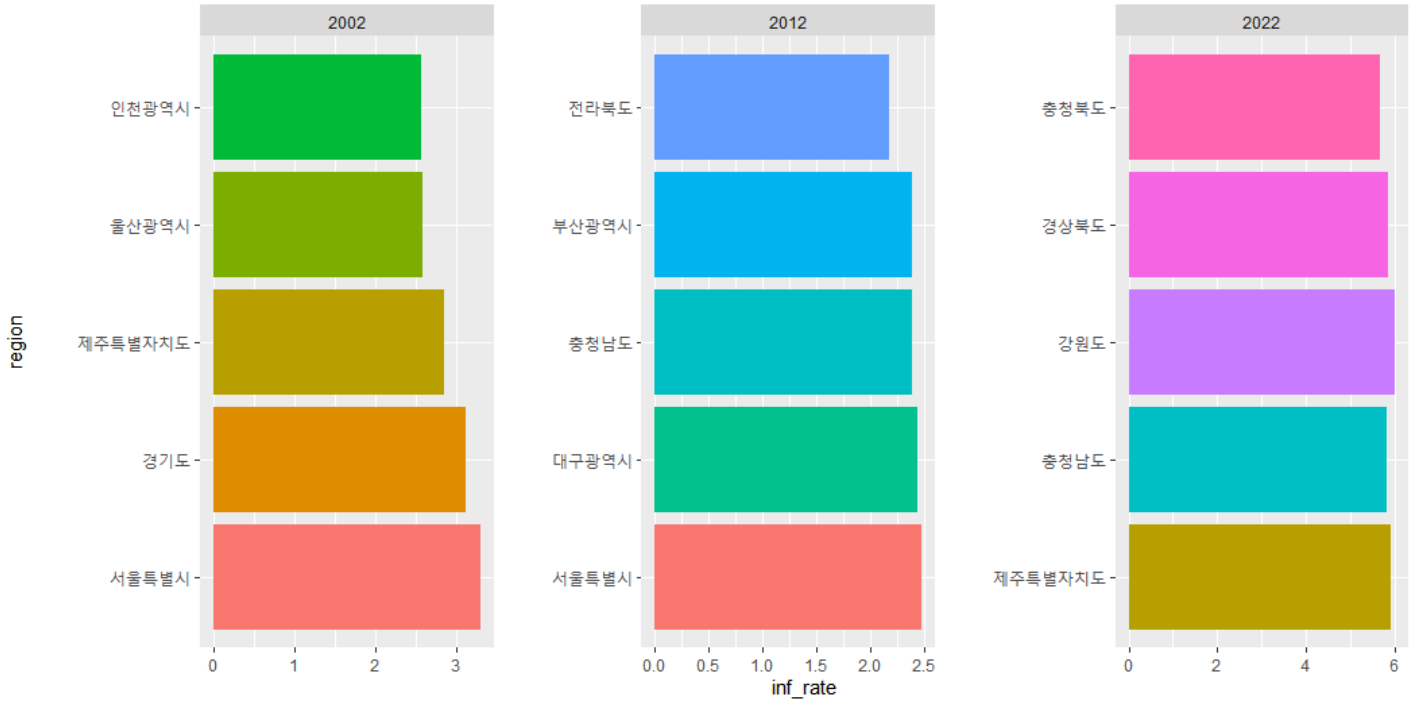
- tidy 형태로 전환

```
inf.tidy <- inf %>%
pivot_longer(cols = -region, names_to = "year", values_to = "inf")
```

```
inf.tidy %>%
mutate(inf_rate = (inf-
dplyr::lag(inf))/dplyr::lag(inf) *100 ) %>%
select(year, region, inf_rate) %>%
filter(year %in% c(2002, 2012, 2022)) %>%
group_by(year) %>%
slice_max(inf_rate, n = 5) %>%
arrange(year, inf_rate) %>%   ungroup()-> fig
```

```
library("forcats")
fig %>%
group_by(year) %>%
arrange(-inf_rate) %>%
mutate(region = fct_inorder(region)) %>%
ggplot(aes(x = region, y = inf_rate, fill = region)) +
geom_bar(stat = "identity") +
facet_wrap(~year, scales="free") +
theme(legend.position = "none") +
coord_flip()
```

inflation 높은 지역 5개 지역



ggcharts

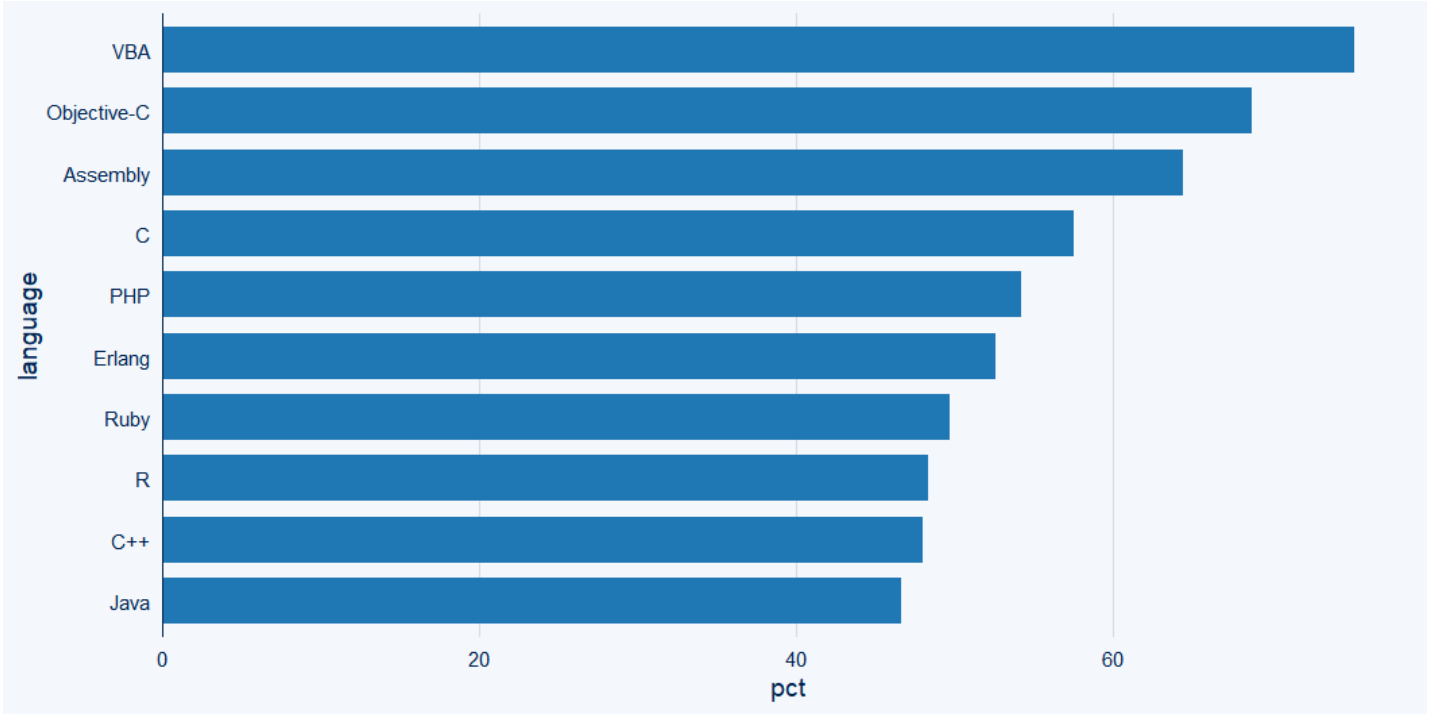
- 필요한 라이브러리 로딩

데이터 입력

```
df_lang <- tibble::tribble(  ~language, ~pct,
"VBA", 75.2,
"Objective-C", 68.7,
"Assembly", 64.4,
"C", 57.5,
"PHP", 54.2,
"Erlang", 52.6,
"Ruby", 49.7,
"R", 48.3,
"C++", 48.0,
"Java", 46.6)
```

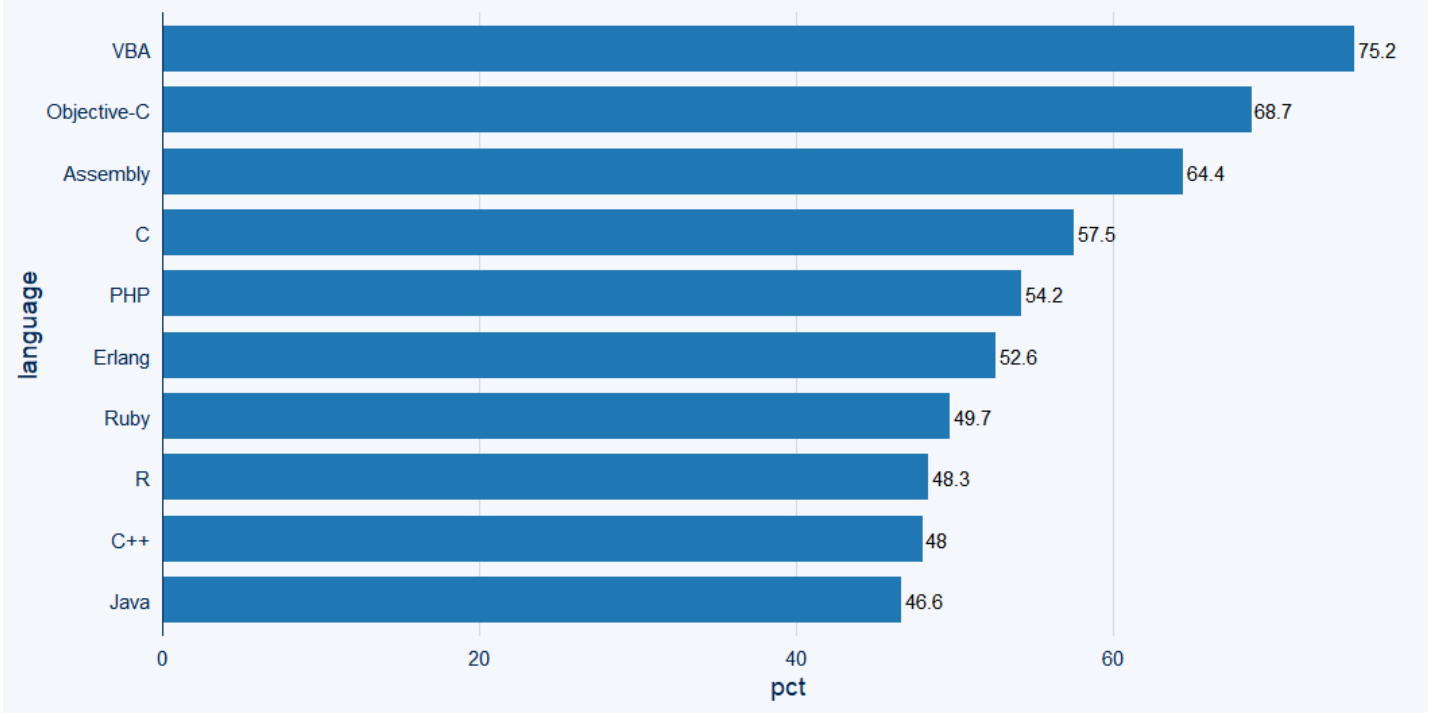
```
chart <- df_lang %>%
bar_chart(x = language, y = pct) %>%
print()
```

bar chart


VBA
Objective-C
Assembly
C
PHP
Erlang
Ruby
R
C++
Java
language
0
20
40
60
pct

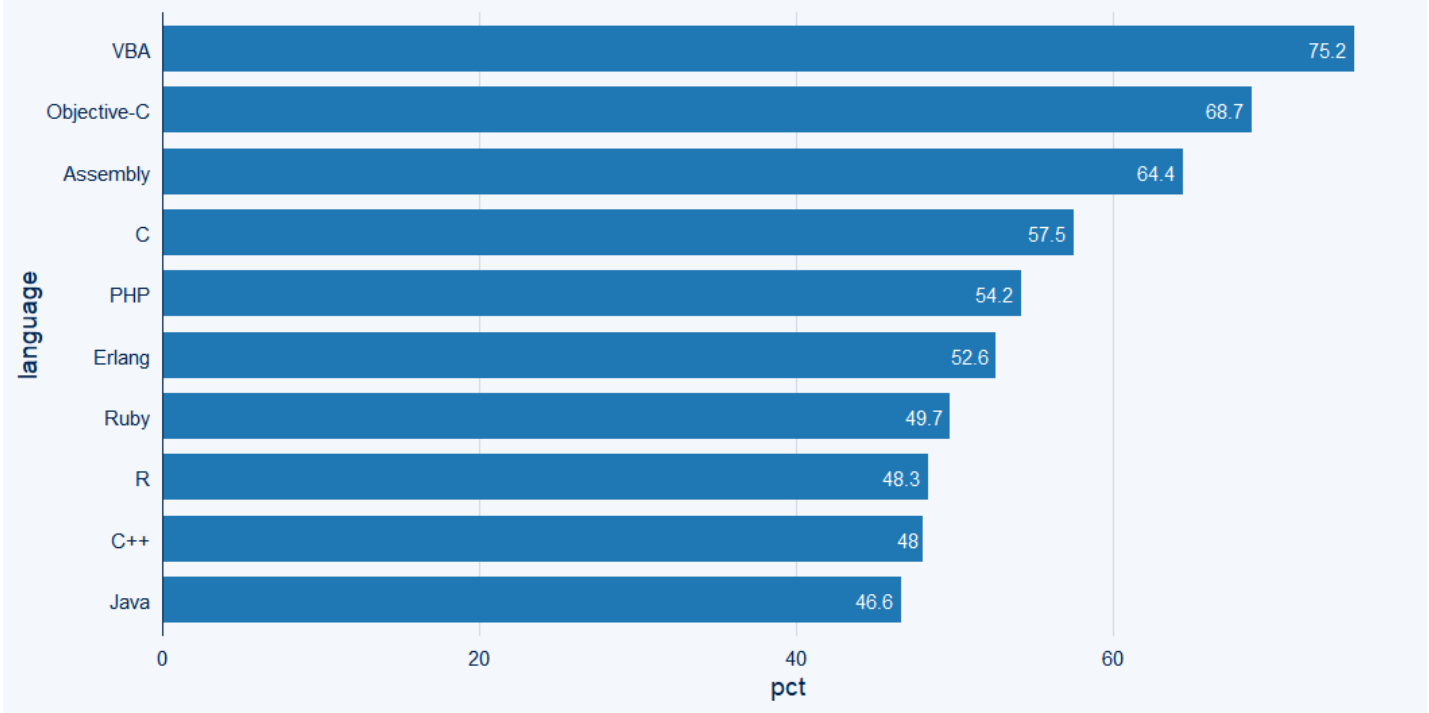
```
chart +
geom_text(aes(x = language, y = pct, label = pct, hjust = -0.1))
```

값-레이블 달기


VBA
Objective-C
Assembly
C
PHP
Erlang
Ruby
R
C++
Java
language
75.2
68.7
64.4
57.5
54.2
52.6
49.7
48.3
48
46.6
0
20
40
60
pct

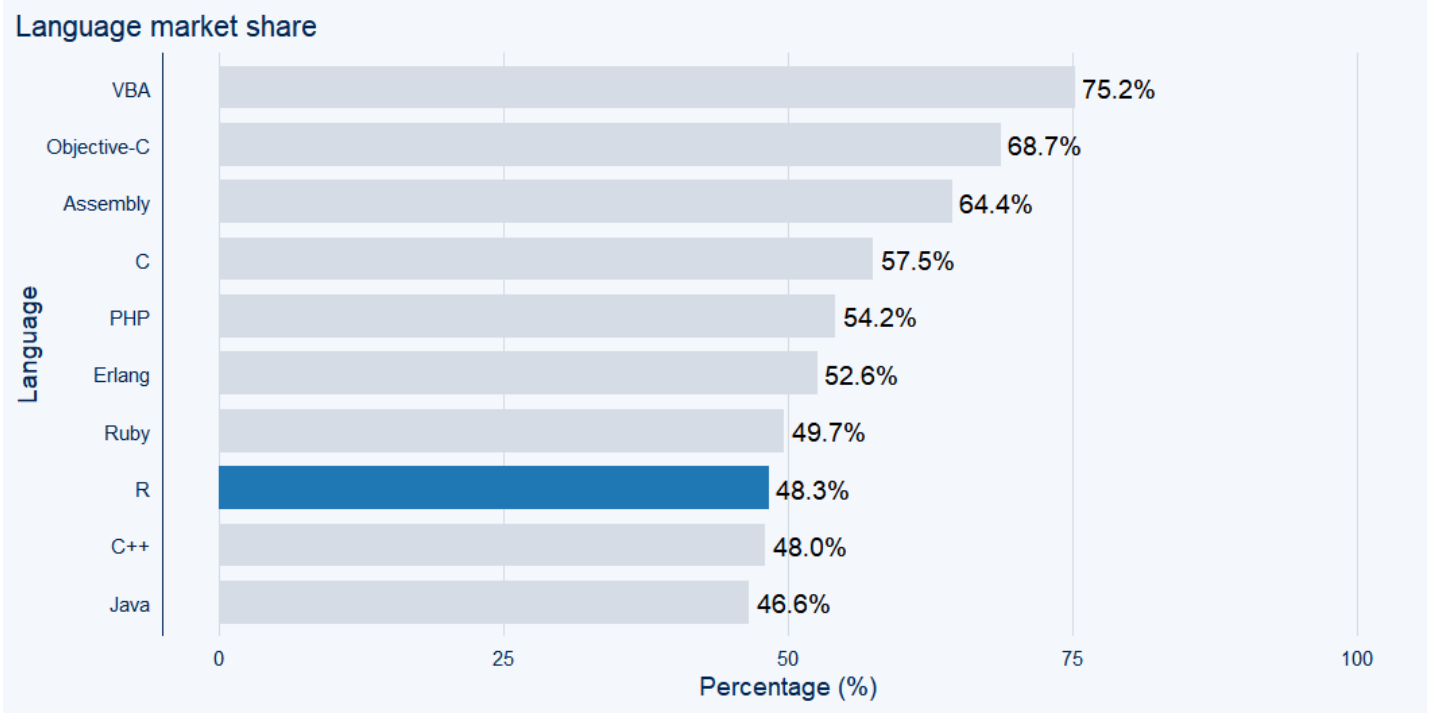
```
chart +  geom_text(aes(label = pct,
hjust = 1.2),
color = "white"
)
```

위치조정

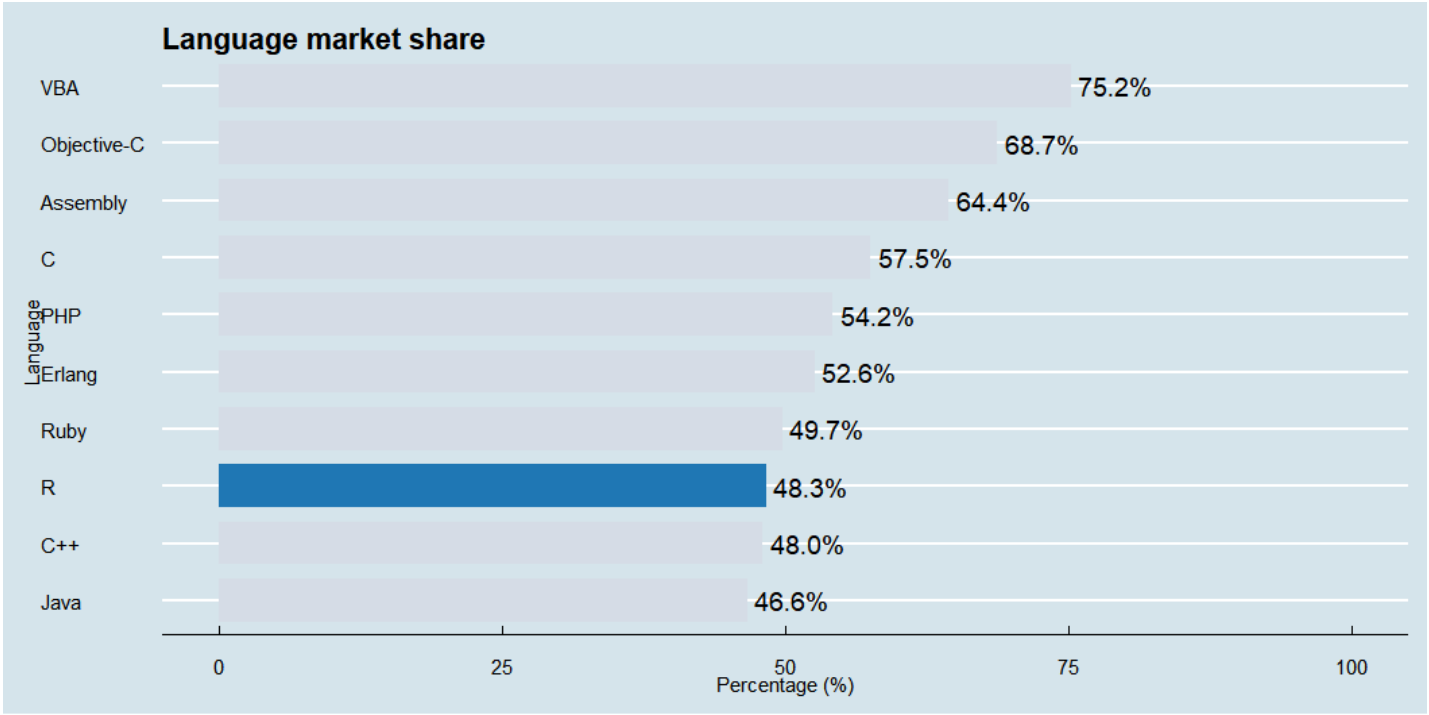

language
VBA
Objective-C
Assembly
C
PHP
Erlang
Ruby
R
C++
Java
75.2
68.7
64.4
57.5
54.2
52.6
49.7
48.3
48
46.6
0
20
40
60
pct

```
df_lang %>%
mutate(label = sprintf("%1.1f%%", pct)) %>%
bar_chart(x = language, y = pct, highlight = "R") +
geom_text(aes(label = label, hjust = -0.1), size = 5)
+
scale_y_continuous(    limits = c(0, 100)    ) +
labs(
title = "Language market share",
x = "Language",
y = "Percentage (%)"
)
```

완성된 챠트


Language market share
VBA
Objective-C
Assembly
C
PHP
Erlang
Ruby
R
C++
Java
75.2%
68.7%
64.4%
57.5%
54.2%
52.6%
49.7%
48.3%
48.0%
46.6%
Language
0
25
50
75
100
Percentage (%)

완성된 챠트 2


Language market share
VBA
Objective-C
Assembly
C
PHP
Erlang
Ruby
R
C++
Java
75.2%
68.7%
64.4%
57.5%
54.2%
52.6%
49.7%
48.3%
48.0%
46.6%
Language
0
25
50
75
100
Percentage (%)