최고의 냉동만두를 찾아서

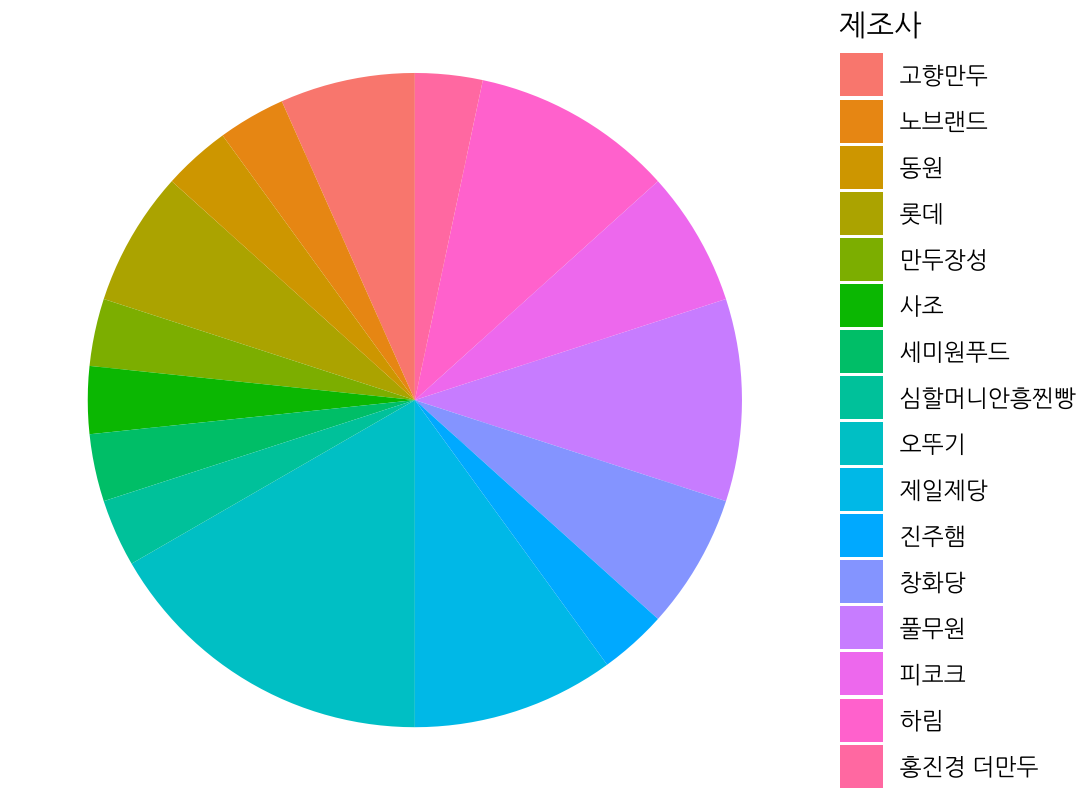
2025-02-16

```
library(tidyverse)
library(tidymodels)
library(showtext)
library(kableExtra)

showtext_auto()

font_add_google("Nanum Gothic", "NanumGothic")
dumpling <- read.csv("https://raw.githubusercontent.com/cloeyeons/dumpling/refs/heads/main/dumpling_data
```

1. 기본 분포도 및 주요 정보 조사

1-1. 점유율

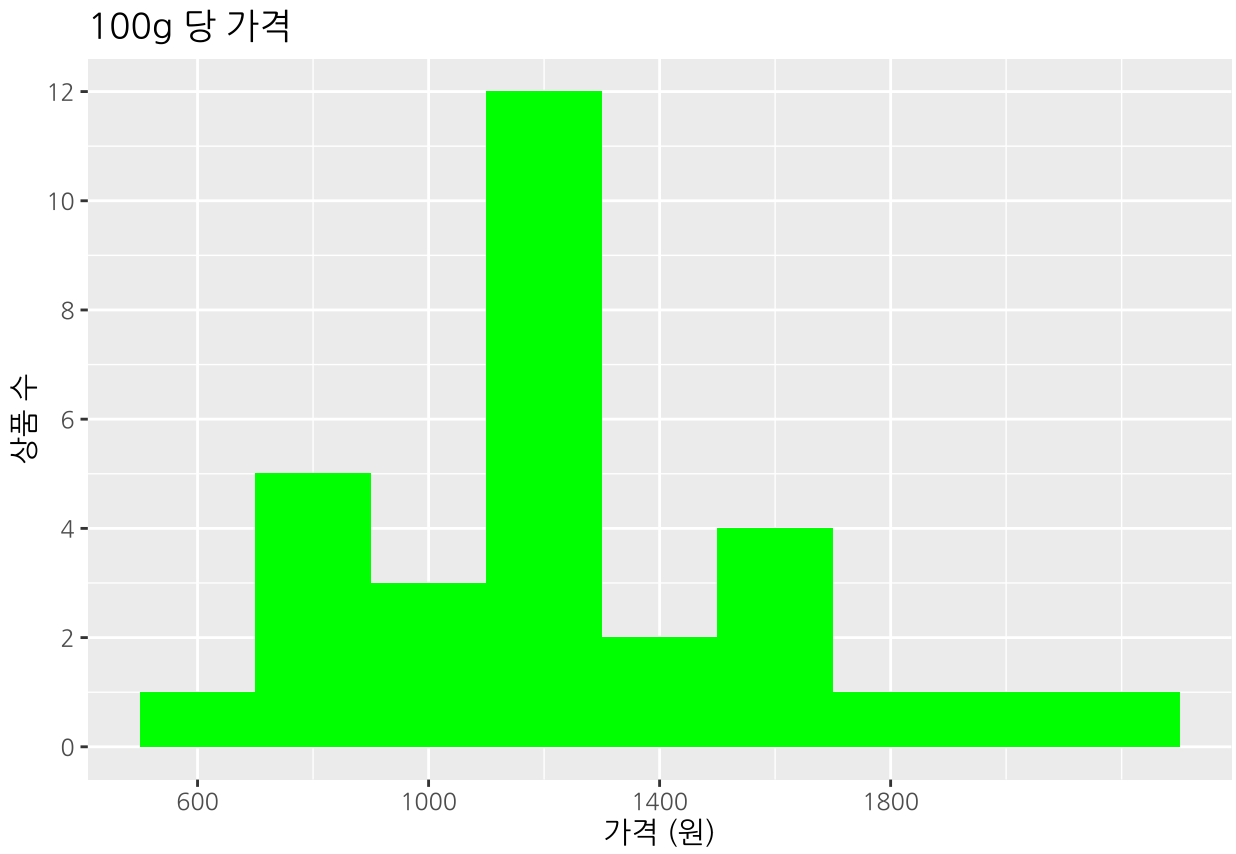
```
brand_count <- dumpling %>%
  count(comp) %>%
  arrange(desc(n)) %>%
  rename(frequency = n)

ggplot(brand_count, aes(x = "", y = frequency, fill = comp)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "        ",
       fill = "  ") +
  theme_void() +
  theme(text = element_text(family = "NanumGothic"))
```



오뚜기 사의 만두 5종이 누적 냉동만두 판매량 상위 30위에 랭크되며 약 17%로 가장 높은 점유율을 보여주었습니다.

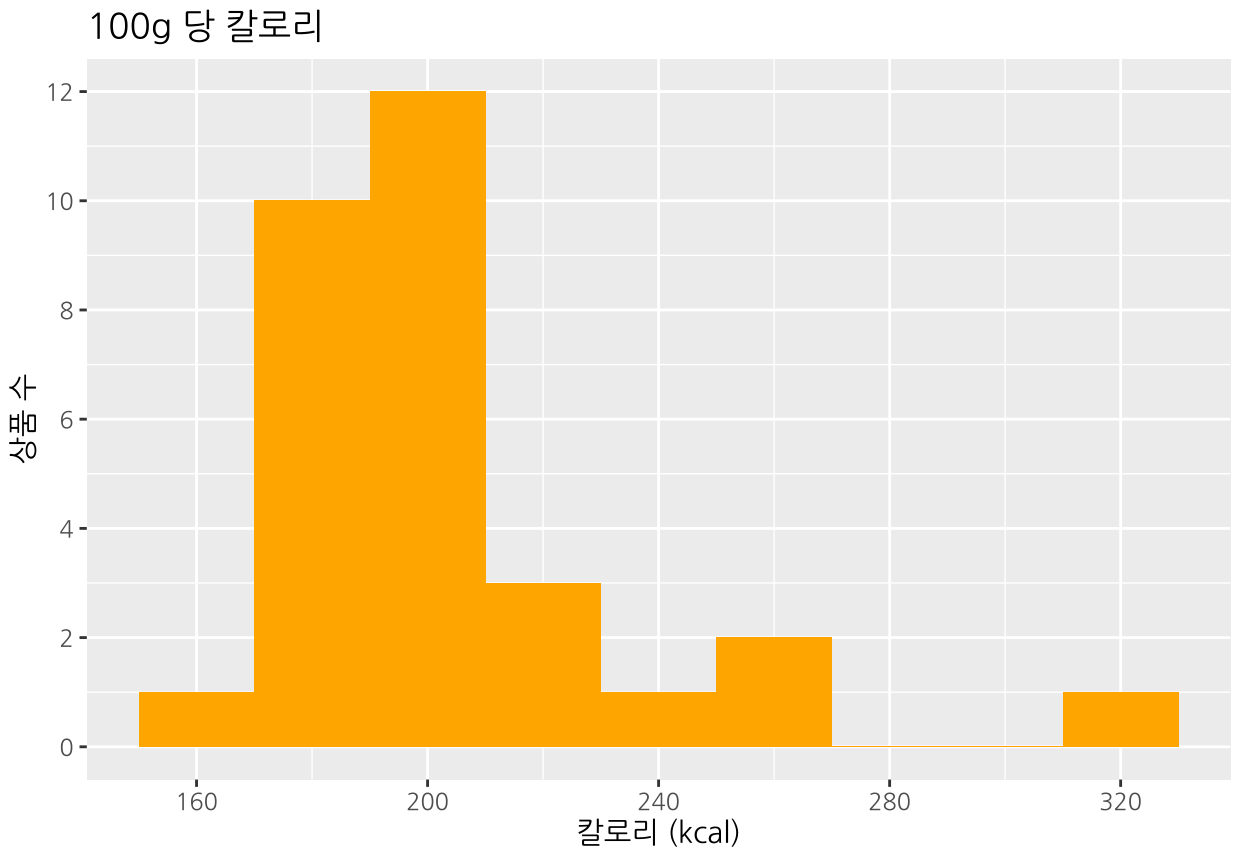
1-2. 100g 당 가격 분포

```
ggplot(dumpling, aes(x = price)) +
  geom_histogram(binwidth = 200, fill = "green") +
  labs (x = "   ( )",
        y = "   ",
        title = "100g    ") +
  scale_x_continuous(breaks = seq(600, max(dumpling$price), by = 400))  +
  scale_y_continuous(breaks = seq(0, 30, by = 2)) +
  theme(text = element_text(family = "NanumGothic"))
```



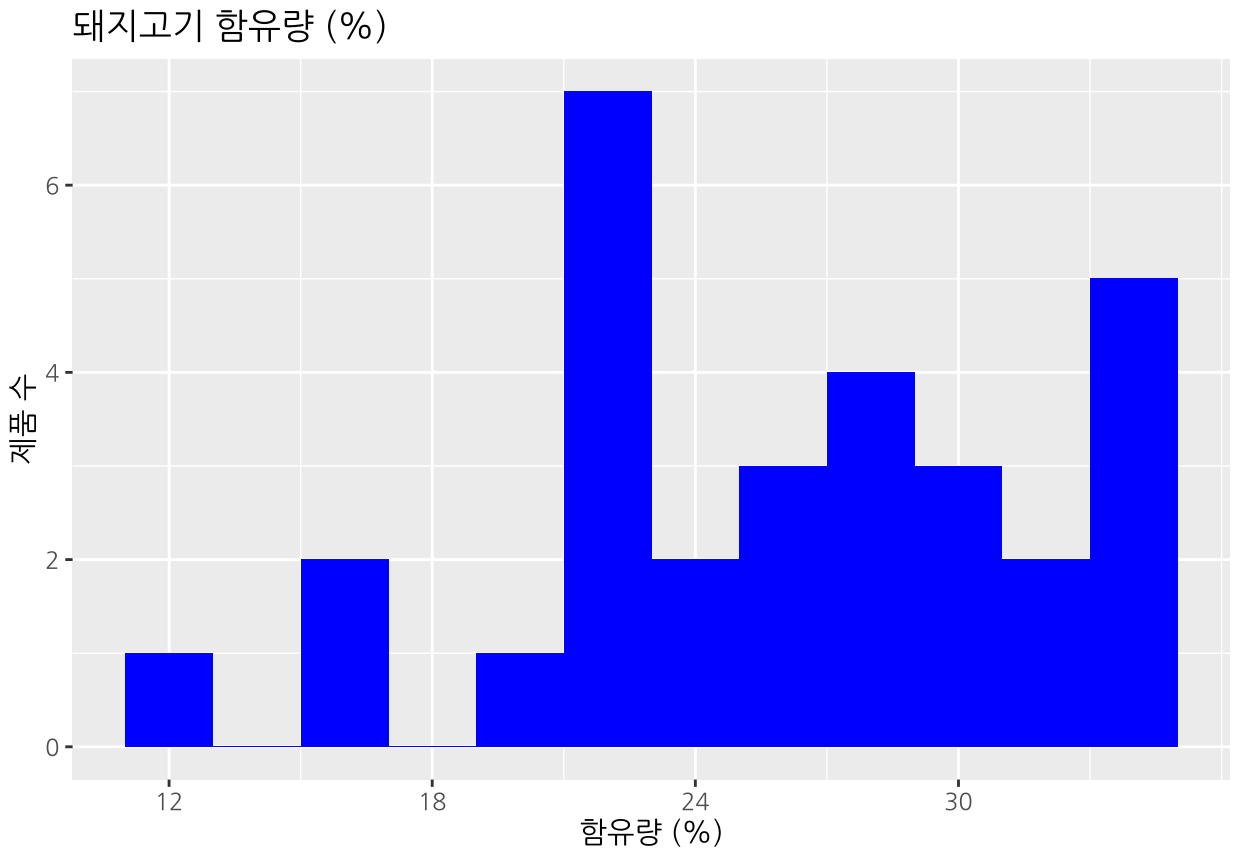
1-3. 100g 당 칼로리 분포

```
ggplot(dumpling, aes(x = calories)) +
  geom_histogram(binwidth = 20, fill = "orange") +
  labs (x = "   (kcal)",
        y = "   ",
        title = "100g     ") +
  scale_x_continuous(breaks = seq(0, max(dumpling$calories), by = 40)) +
  scale_y_continuous(breaks = seq(0, 30, by = 2)) +
  theme(text = element_text(family = "NanumGothic"))
```



1-4. 돼지고기 함유량

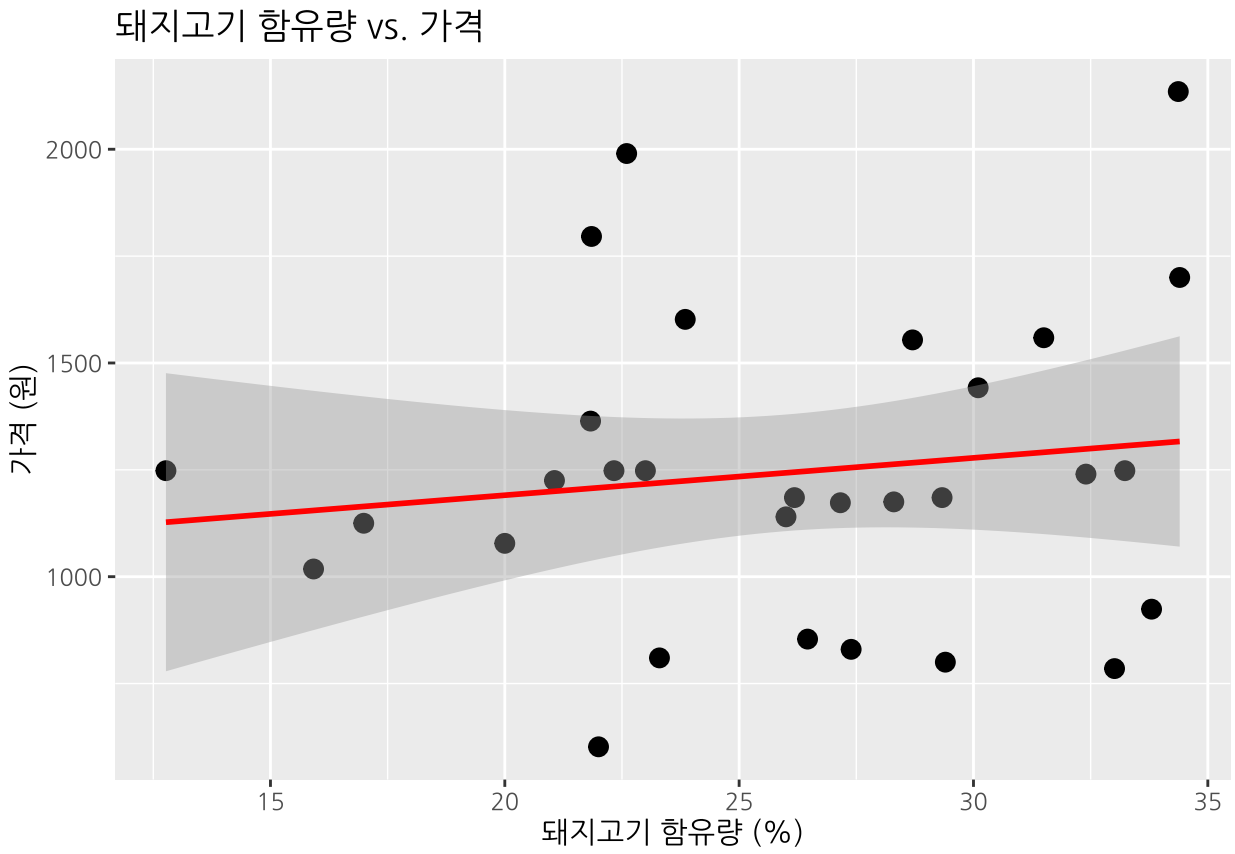
```
ggplot(dumpling, aes(x = porkCont)) +
  geom_histogram(binwidth = 2, fill = "blue") +
  labs (x = "   (%)",
        y = "   ",
        title = "      (%)") +
  scale_x_continuous(breaks = seq(0, max(dumpling$porkCont), by = 6)) +
   scale_y_continuous(breaks = seq(0, 30, by = 2)) +
  theme(text = element_text(family = "NanumGothic"))
```



2. 연관성 조사

2-1. 돼지고기 함유량 vs. 가격

```
ggplot(dumpling, aes(x = porkCont, y = price)) +
  geom_point(color = "black", size = 3) +
  geom_smooth(method = "lm", color = "red") +
  labs (x = "      (%)",
        y = "  ( )",
        title = "      vs.  ") +
  theme(text = element_text(family = "NanumGothic"))
```



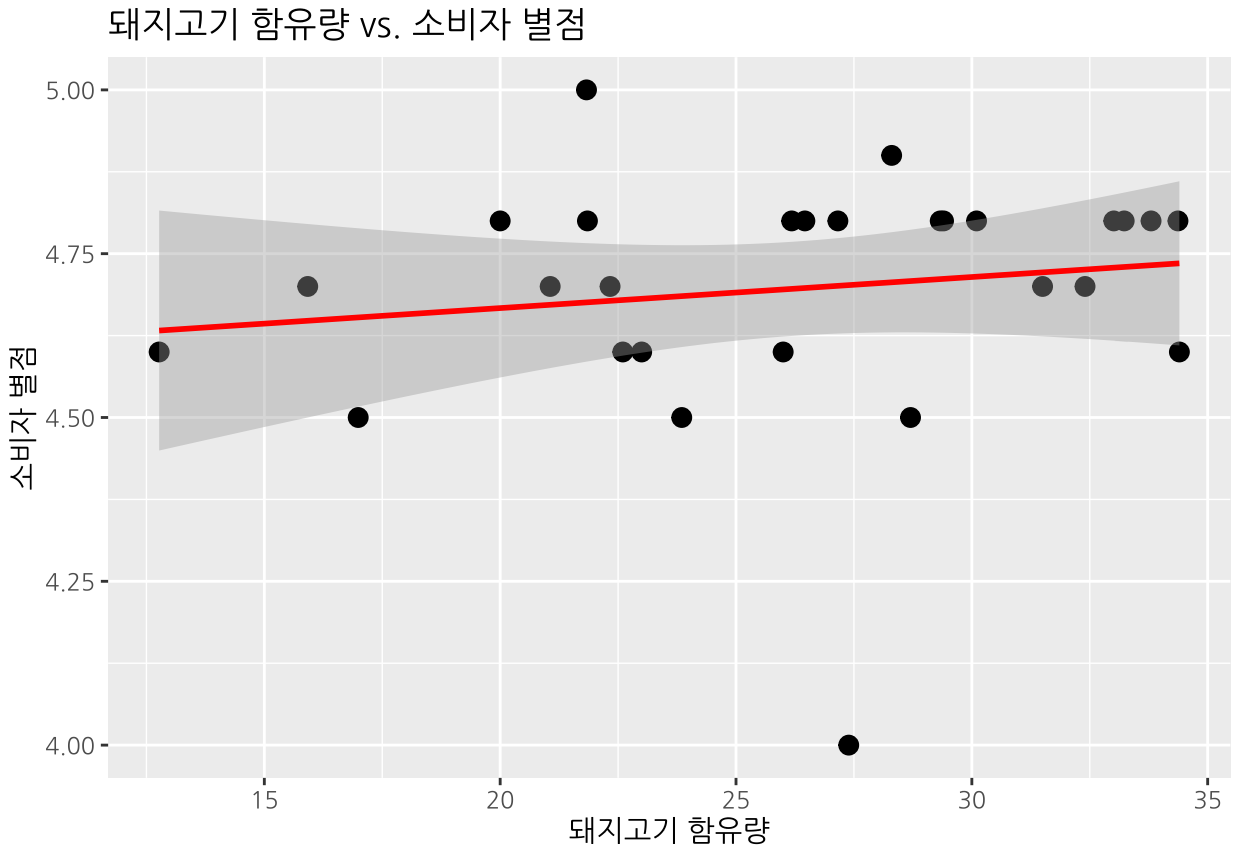

```
pp_corr <- cor.test(dumpling$porkCont, dumpling$price)
print(pp_corr)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dumpling$porkCont and dumpling$price
## t = 0.73609, df = 28, p-value = 0.4678
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2341094  0.4744967
## sample estimates:
##       cor
## 0.1377806
```

2-2. 돼지고기 함유량, 가격 vs. 소비자 별점

```
cleaned_data <- dumpling[complete.cases(dumpling$porkCont, dumpling$rating),]

ggplot(cleaned_data, aes(x = porkCont, y = rating)) +
  geom_point(color = "black", size = 3) +
  geom_smooth(method = "lm", color = "red") +
  labs (x = "     ",
```

```
       y = "    ",
       title = "      vs.     ") +
  theme(text = element_text(family = "NanumGothic"))
```

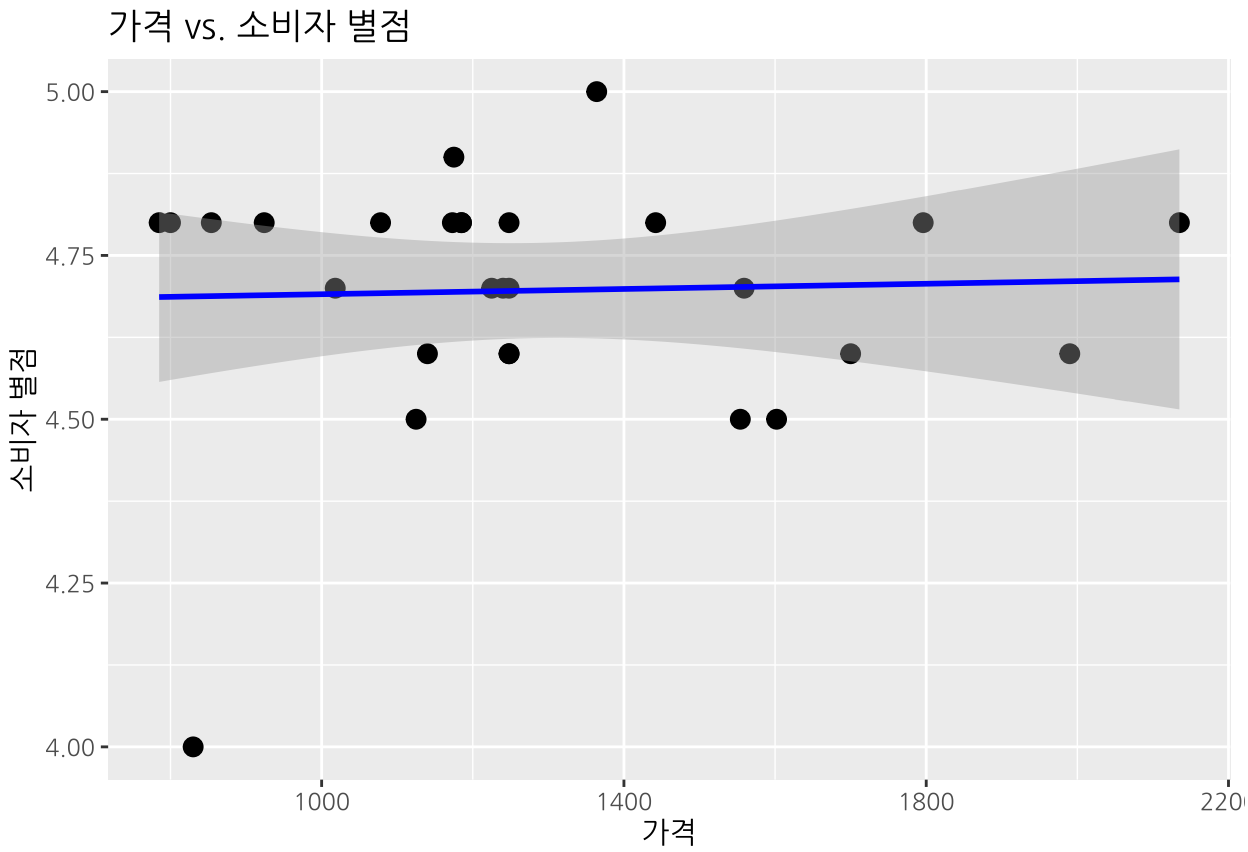



```
pr_corr <- cor.test(cleaned_data$porkCont, cleaned_data$rating)
print(pr_corr)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cleaned_data$porkCont and cleaned_data$rating
## t = 0.77765, df = 26, p-value = 0.4438
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2355600  0.4959477
## sample estimates:
##       cor
## 0.1507666
```

```
ggplot(cleaned_data, aes(x = price, y = rating)) +
  geom_point(color = "black", size = 3) +
  geom_smooth(method = "lm", color = "blue") +
  labs (x = " ",
        y = "    ",
```

```
       title = "   vs.      ") +
  theme(text = element_text(family = "NanumGothic"))
```




```
pr_corr <- cor.test(cleaned_data$porkCont, cleaned_data$rating)
print(pr_corr)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cleaned_data$porkCont and cleaned_data$rating
## t = 0.77765, df = 26, p-value = 0.4438
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2355600  0.4959477
## sample estimates:
##       cor
## 0.1507666
```

```
interaction_model <- lm(rating ~ price * porkCont, data = cleaned_data)
summary(interaction_model)
```

```
##
## Call:
## lm(formula = rating ~ price * porkCont, data = cleaned_data)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69379 -0.08372  0.06378  0.09342  0.31902
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.358e+00  7.629e-01   5.712 6.94e-06 ***
## price           1.695e-04  5.855e-04   0.290    0.775
## porkCont        1.167e-02  2.656e-02   0.439    0.664
## price:porkCont -5.463e-06  2.016e-05  -0.271    0.789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1921 on 24 degrees of freedom
## Multiple R-squared:  0.02635,    Adjusted R-squared:  -0.09536
## F-statistic: 0.2165 on 3 and 24 DF,  p-value: 0.884
```

3. 종합 순위

```
dumpling_rank <- dumpling %>%
  mutate(
    sales_rank = rank(rank, ties.method = "first"),
    price_rank = rank(price, ties.method = "first"),
    calories_rank = rank(calories, ties.method = "first"),
    pork_content_rank = rank(-porkCont, ties.method = "first"))

ranked_summary <- dumpling_rank %>%
  select(comp,
         product,
         sales_rank,
         price_rank,
         calories_rank,
         pork_content_rank)

ranked_summary <- ranked_summary %>%
  mutate(overall_rank_raw = rowSums(select(., sales_rank, price_rank, calories_rank, pork_content_rank)))
  mutate(overall_rank = rank(overall_rank_raw, ties.method = "first")) %>%
  arrange(overall_rank)

ranking_table <- ranked_summary %>%
  select(overall_rank, comp, product, sales_rank, price_rank, calories_rank, pork_content_rank) %>%
  mutate(
    comp = stringr::str_wrap(comp, width = 20),
    product = stringr::str_wrap(product, width = 30)
  )

knitr::kable(ranking_table,
             format = "latex",
             booktabs = TRUE,
             caption = "    ",
```

```
            col.names = c("    ", "  ", "  ", "   ", "   ", "    (   )", "      "),
            align = c("c", "l", "l", "c", "c", "c", "c"),
            escape = TRUE) %>%
 kable_styling(latex_options = c("hold_position", "scale_down"),
               font_size = 9)
```

Table 1: 종합 순위표

종합 순위	제조사	제품명	판매순위	가격순위	칼로리순위 (낮은 순)	돼지고기함유량순위
1	풀무원	얇은피 꽉찬속 고기만두	1	18	1	4
2	하림	더미식 육즙고기교자	2	17	4	6
3	노브랜드	얇은피 고기만두	3	2	20	5
4	제일제당	비비고 깻잎고기만두	5	14	7	10
5	피코크	생돈육왕만두	11	6	10	15
6	오뚜기	엑스오 교자 고기	10	7	24	3
7	오뚜기	맛있는 찹쌀군만두 고기	12	3	29	9
8	하림	더미식 땡초고기교자	13	24	5	11
9	하림	더미식 고기군만두	14	27	14	1
10	롯데	셰푸드 쫄깃롤만두	25	5	15	13
11	동원	속이 비치는 얇은피 고기만두	19	15	9	16
12	창화당	참만두	26	13	8	12
13	오뚜기	맛있는 왕교자고기	6	12	28	14
14	롯데	식사이론 대파고기만두	22	11	11	17
15	오뚜기	엑스오 수제 손만두 고기듬뿍	8	23	23	8
16	고향만두	얇은피 왕교자 고기만두	23	9	3	27
17	제일제당	비비고 진한고기만두	4	30	27	2
18	진주햄	포자고기만두	28	4	13	19
19	제일제당	비비고 청양고기만두	9	29	6	21
20	고향만두	고기깻잎 지짐만두	16	19	12	20
21	세미원푸드	이화맛집고기손만두	27	1	18	23
22	만두장성	시장부추고기만두	15	16	16	26
23	피코크	속이 꽉 찬 고기 킹만두	20	8	17	29
24	홍진경 더만두	더큰만두 고기맛	29	26	2	18
25	오뚜기	엑스오 감자떡만두 고기	21	25	26	7
26	풀무원	고기지짐만두	7	28	21	24
27	풀무원	한식교자 고기한상	18	21	22	22
28	창화당	고기궁중만두	24	22	19	25
29	사조	대림 고기손만두	17	20	25	30
30	심할머니안흥찐빵	감자고기손만두	30	10	30	28