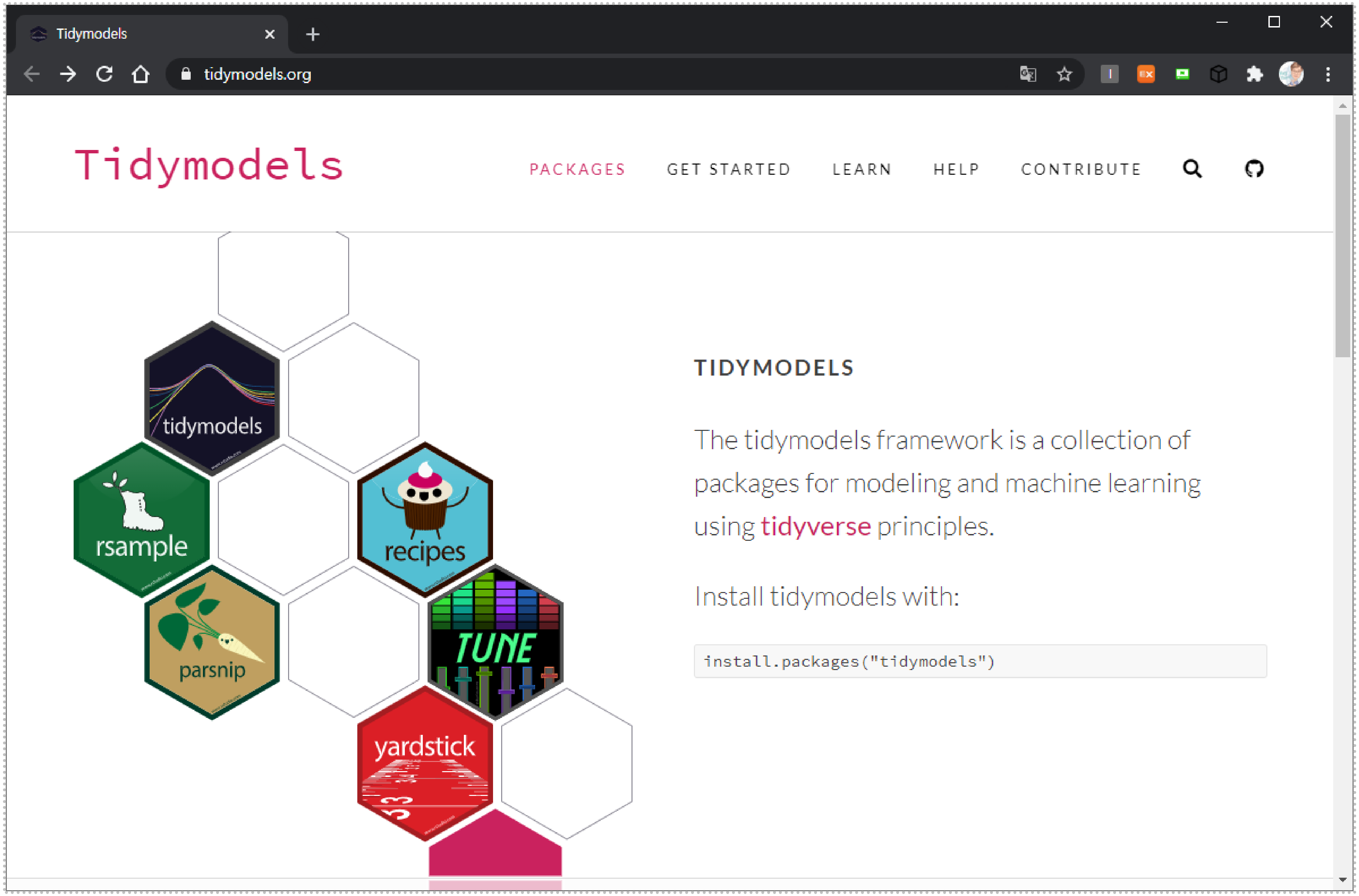
Ch11.군집분석 실습

교재



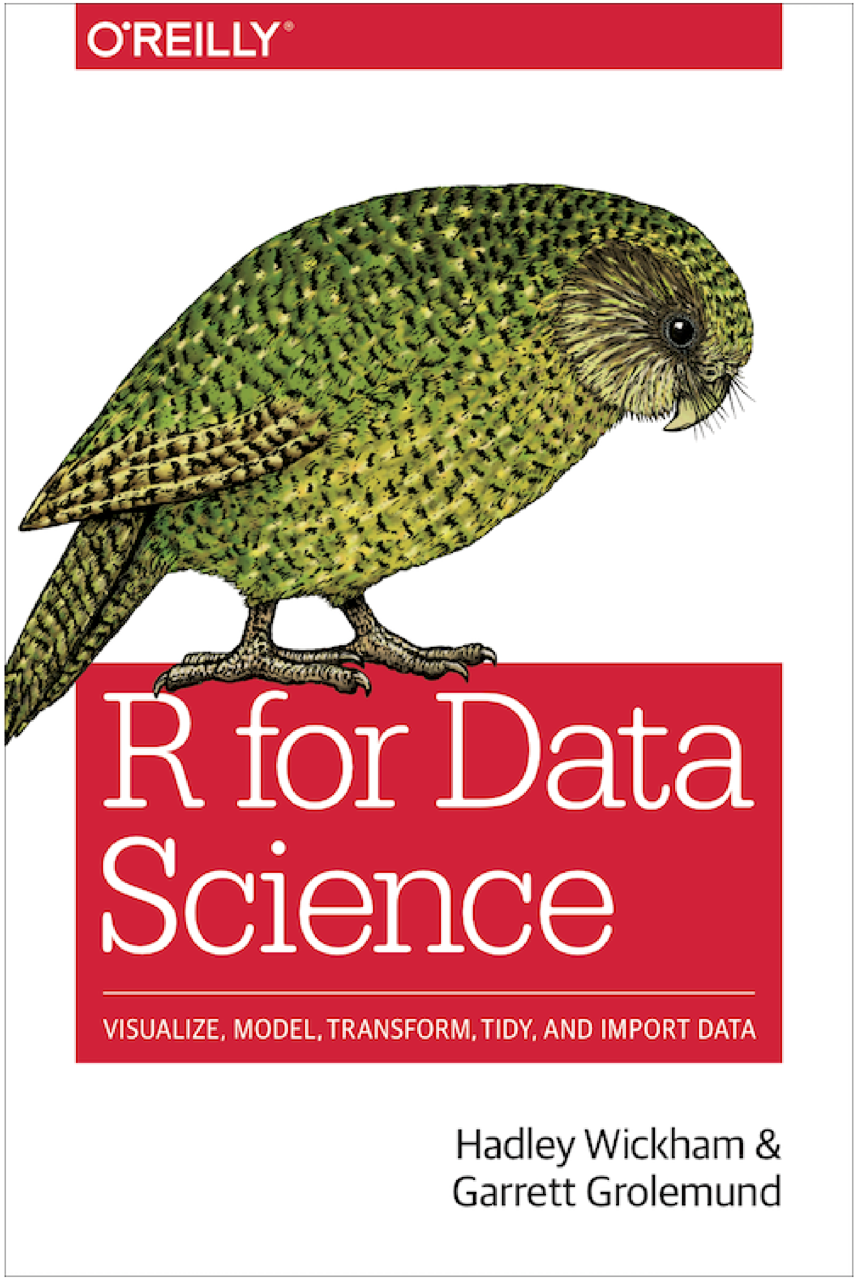
TIDYMODELS

The tidymodels framework is a collection of packages for modeling and machine learning using tidyverse principles.

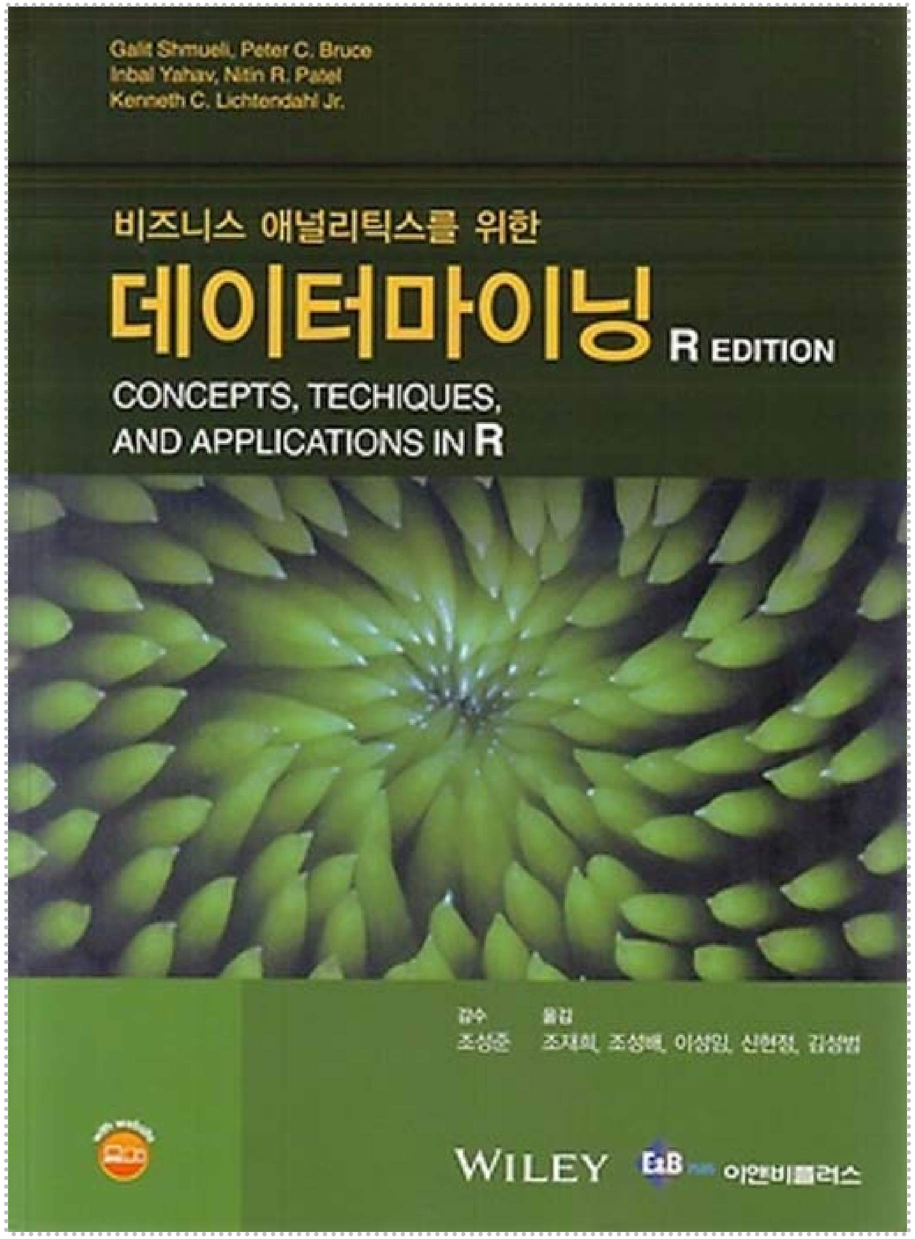
Install tidymodels with:

```
install.packages("tidymodels")
```

교재



https://r4ds.had.co.nz/


Galit Shmueli, Peter C. Bruce
Inbal Yahav, Nitin R. Patel
Kenneth C. Lichtendahl Jr.
비즈니스 애널리틱스를 위한
데이터마이닝 R EDITION
CONCEPTS, TECHIQUES,
AND APPLICATIONS IN R
감수
조성준
옮김
조재희, 조성배, 이성임, 신현정, 김성범
WILEY
E&B 이앤비플러스

학습목표

- R 프로그래밍 실습
- 실습: K-평균 군집화

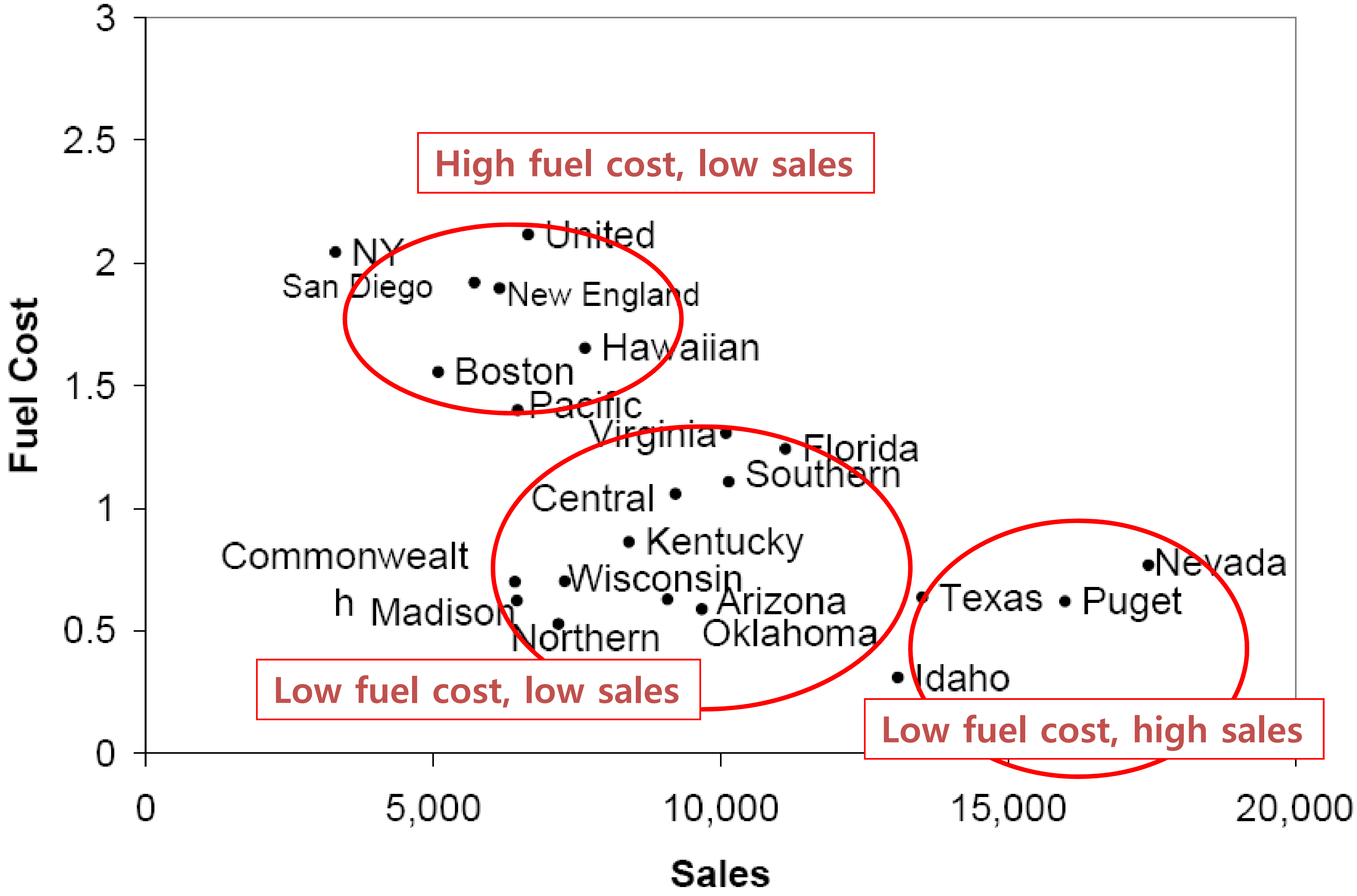
전력회사 사례

데이터형식

- 공공전력회사 사례
  - 22개 전력회사
  - 유사한 전력회사로 군집

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost
Arizona	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53	3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56	0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0	0.768
New England	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56	9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

군집분석 사례

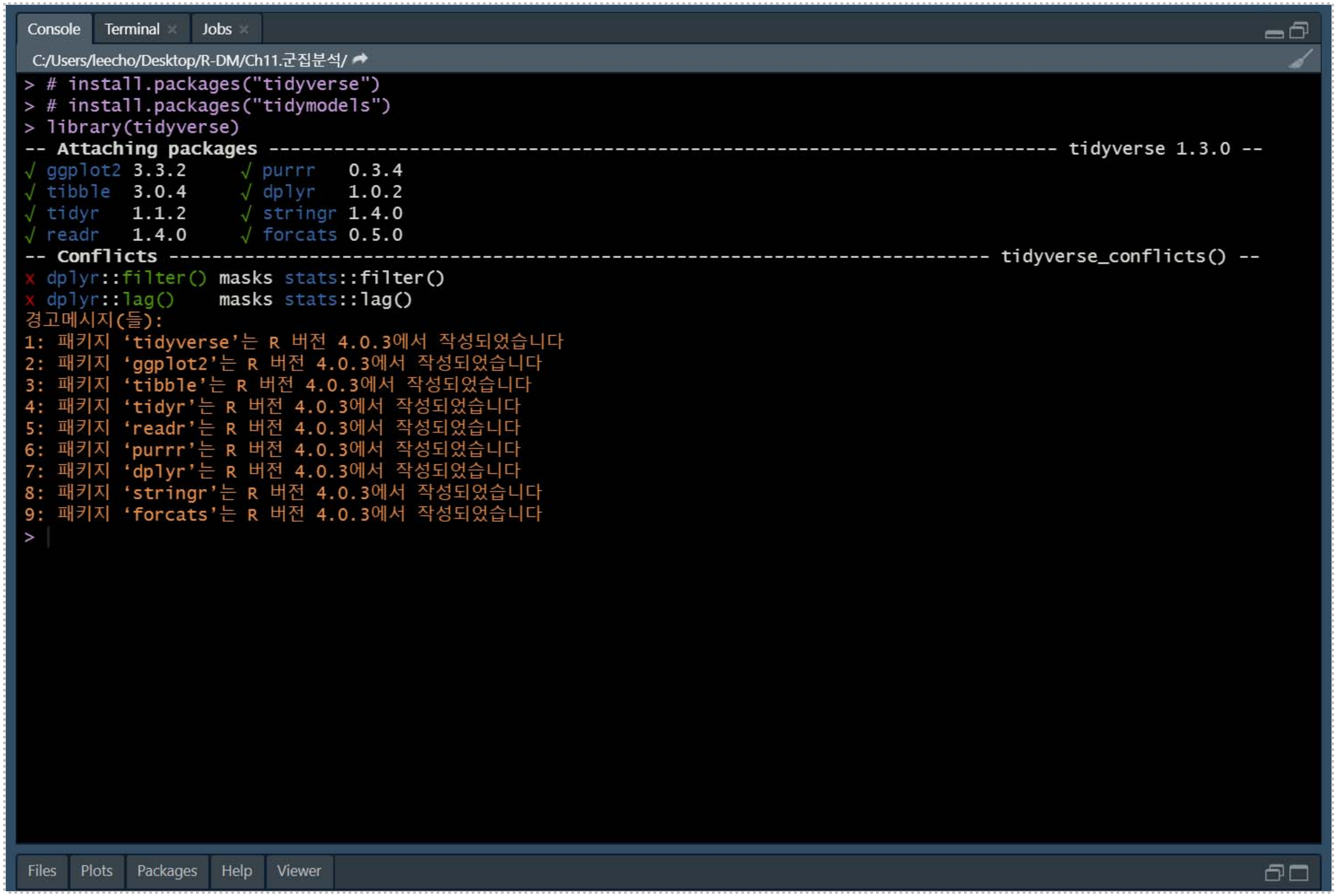


High fuel cost, low sales
Low fuel cost, low sales
Low fuel cost, high sales
NY
San Diego
United
New England
Hawaiian
Boston
Pacific
Virginia
Florida
Southern
Central
Kentucky
Commonwealth
Madison
Wisconsin
Arizona
Oklahoma
Northern
Texas
Puget
Nevada
Idaho
3
2.5
2
1.5
1
0.5
0
0
5,000
10,000
15,000
20,000
Fuel Cost
Sales

Package 설치

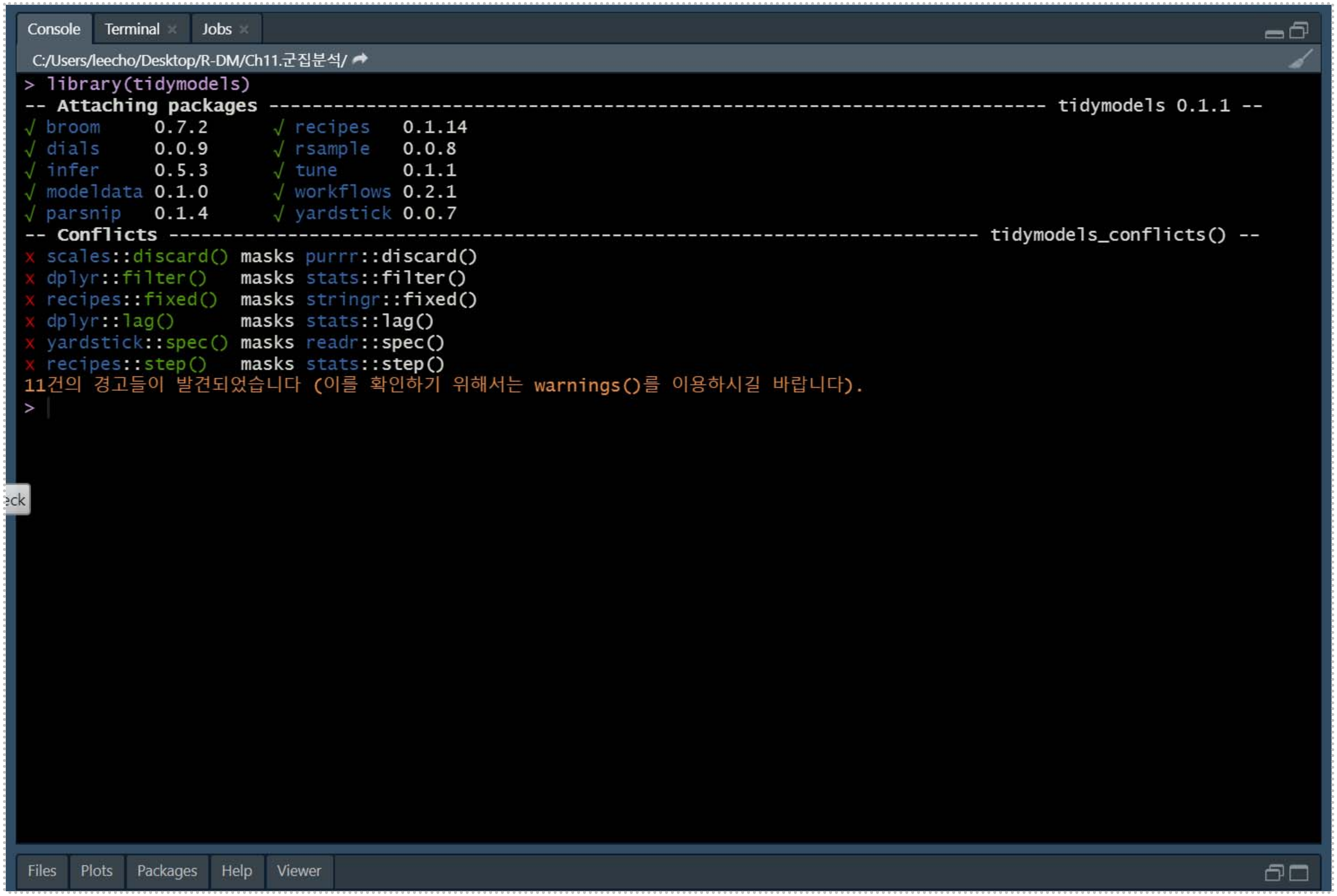
```
# https://www.tidymodels.org/learn/statistics/k-means/
# 02.K-평균 군집화: Table 15.9


############################################################
# tidyverse: ggplot2, purrr, tibble  3.0.3,          #
#          dplyr, tidyr, stringr, readr, forcats    #
############################################################

# install.packages("tidyverse")
# install.packages("tidymodels")
library(tidyverse)
library(tidymodels)
```

Package 설치



```
> # install.packages("tidyverse")
> # install.packages("tidymodels")
> library(tidyverse)
-- Attaching packages --------------------------------------------------------- tidyverse 1.3.0 --
√ ggplot2 3.3.2     √ purrr   0.3.4
√ tibble  3.0.4     √ dplyr   1.0.2
√ tidyr   1.1.2     √ stringr 1.4.0
√ readr   1.4.0     √ forcats 0.5.0
-- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
경고메시지(들):
1: 패키지 'tidyverse'는 R 버전 4.0.3에서 작성되었습니다
2: 패키지 'ggplot2'는 R 버전 4.0.3에서 작성되었습니다
3: 패키지 'tibble'는 R 버전 4.0.3에서 작성되었습니다
4: 패키지 'tidyr'는 R 버전 4.0.3에서 작성되었습니다
5: 패키지 'readr'는 R 버전 4.0.3에서 작성되었습니다
6: 패키지 'purrr'는 R 버전 4.0.3에서 작성되었습니다
7: 패키지 'dplyr'는 R 버전 4.0.3에서 작성되었습니다
8: 패키지 'stringr'는 R 버전 4.0.3에서 작성되었습니다
9: 패키지 'forcats'는 R 버전 4.0.3에서 작성되었습니다
> 
```

Package 설치

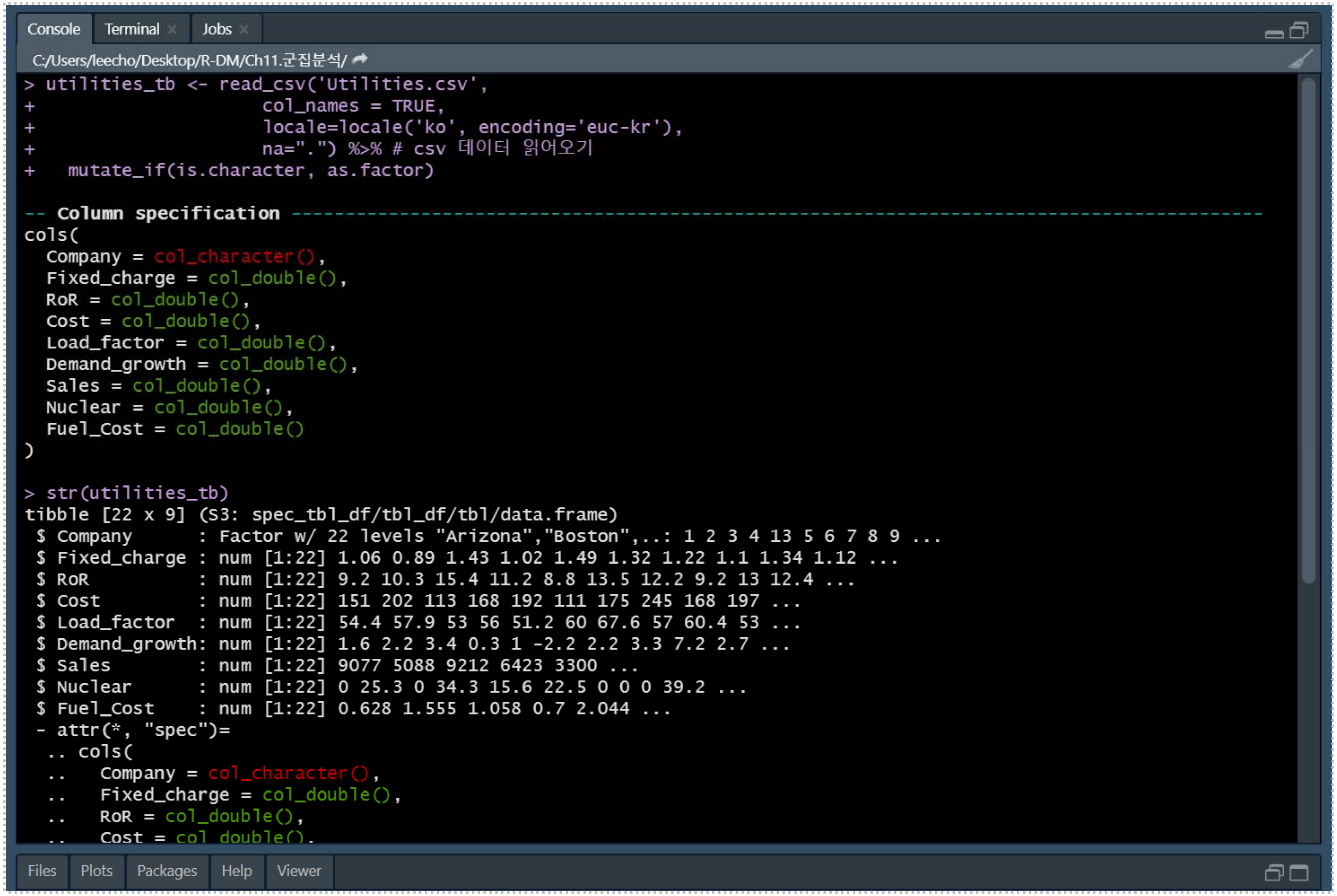


```
> library(tidymodels)
-- Attaching packages ------------------------------------------------------------------ tidymodels 0.1.1 --
√ broom     0.7.2      √ recipes   0.1.14
√ dials     0.0.9      √ rsample   0.0.8
√ infer     0.5.3      √ tune      0.1.1
√ modeldata 0.1.0      √ workflows 0.2.1
√ parsnip   0.1.4      √ yardstick 0.0.7
-- Conflicts ------------------------------------------------------------------ tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
11건의 경고들이 발견되었습니다 (이를 확인하기 위해서는 warnings()를 이용하시길 바랍니다).
> 
```

01.데이터 불러오기

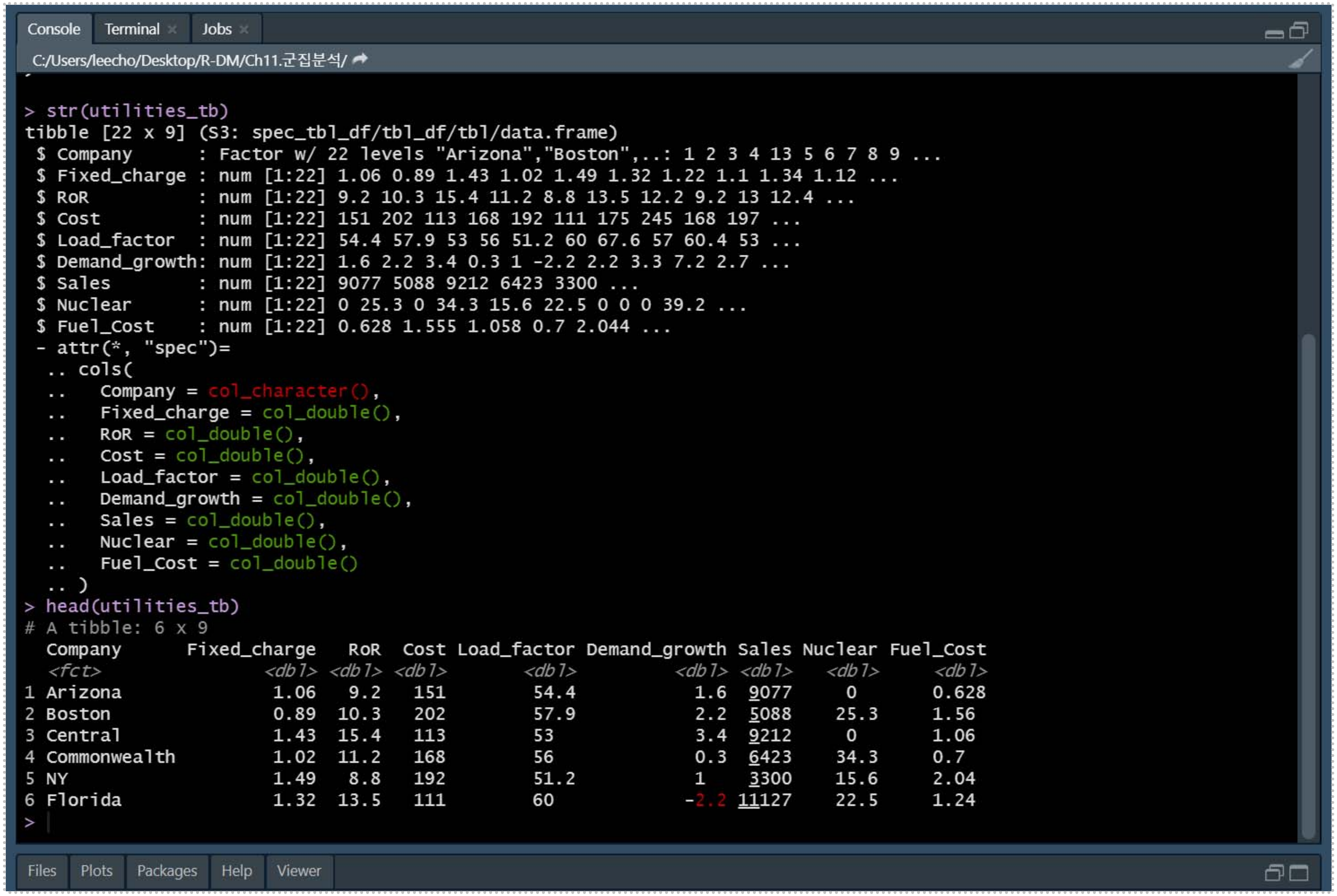
```
# 01.데이터 불러오기
book_tb <- read_csv('CharlesBookClub.csv',
                    col_names = TRUE,
                    locale=locale('ko', encoding='euc-kr'),
                    na=".") %>% # csv 데이터 읽어오기
  mutate_if(is.character, as.factor)

str(book_tb)
head(book_tb)
```

01.데이터 불러오기



```
> utilities_tb <- read_csv('Utilities.csv',
+                          col_names = TRUE,
+                          locale=locale('ko', encoding='euc-kr'),
+                          na=".") %>% # csv 데이터 읽어오기
+   mutate_if(is.character, as.factor)

-- Column specification ---------------------------------------------------------------------------
cols(
  Company = col_character(),
  Fixed_charge = col_double(),
  RoR = col_double(),
  Cost = col_double(),
  Load_factor = col_double(),
  Demand_growth = col_double(),
  Sales = col_double(),
  Nuclear = col_double(),
  Fuel_Cost = col_double()
)

> str(utilities_tb)
tibble [22 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Company      : Factor w/ 22 levels "Arizona","Boston",..: 1 2 3 4 13 5 6 7 8 9 ...
 $ Fixed_charge : num [1:22] 1.06 0.89 1.43 1.02 1.49 1.32 1.22 1.1 1.34 1.12 ...
 $ RoR          : num [1:22] 9.2 10.3 15.4 11.2 8.8 13.5 12.2 9.2 13 12.4 ...
 $ Cost         : num [1:22] 151 202 113 168 192 111 175 245 168 197 ...
 $ Load_factor  : num [1:22] 54.4 57.9 53 56 51.2 60 67.6 57 60.4 53 ...
 $ Demand_growth: num [1:22] 1.6 2.2 3.4 0.3 1 -2.2 2.2 3.3 7.2 2.7 ...
 $ Sales        : num [1:22] 9077 5088 9212 6423 3300 ...
 $ Nuclear      : num [1:22] 0 25.3 0 34.3 15.6 22.5 0 0 0 39.2 ...
 $ Fuel_Cost    : num [1:22] 0.628 1.555 1.058 0.7 2.044 ...
 - attr(*, "spec")=
  .. cols(
  ..   Company = col_character(),
  ..   Fixed_charge = col_double(),
  ..   RoR = col_double(),
  ..   Cost = col_double(),
```

01.데이터 불러오기

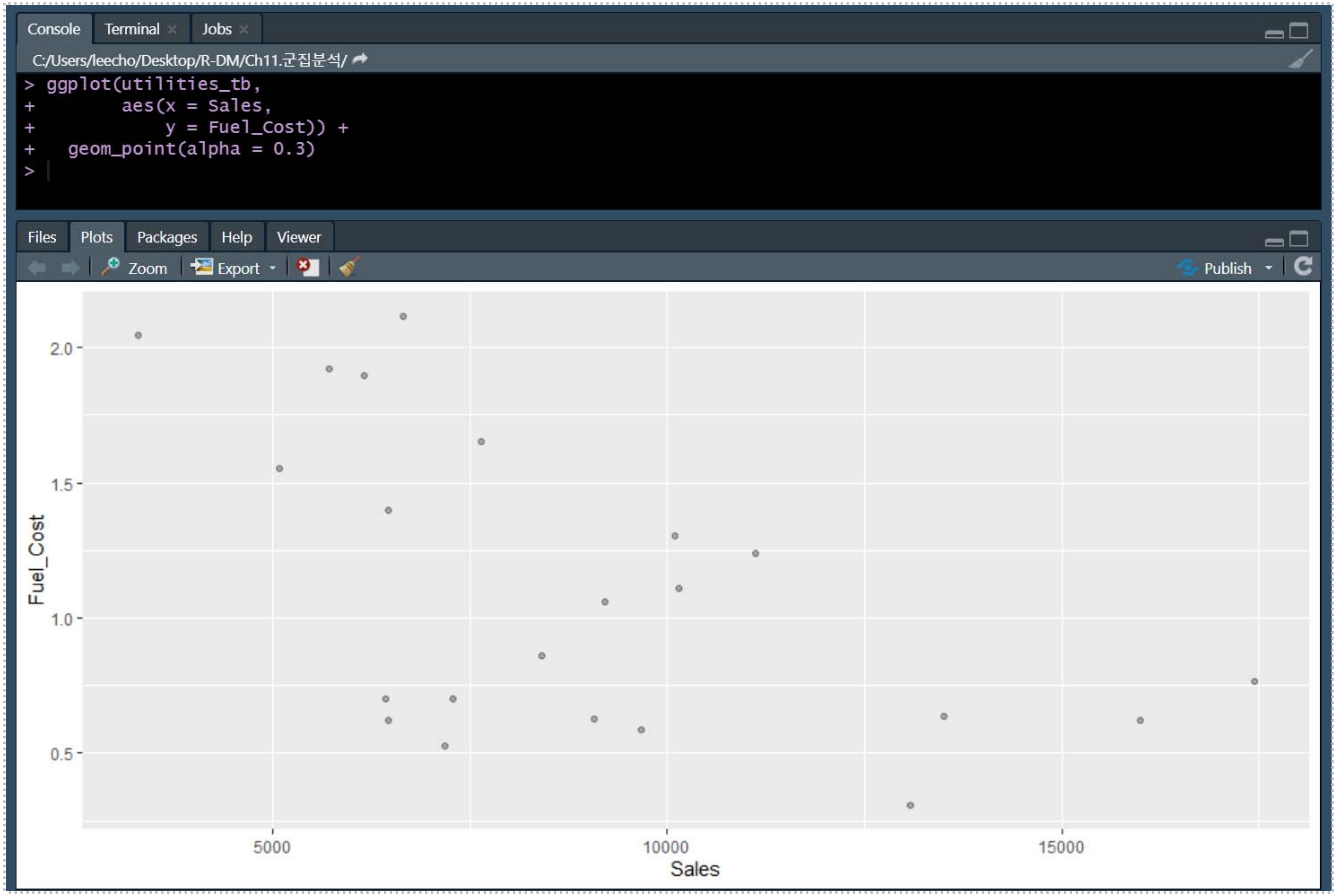


```
> str(utilities_tb)
tibble [22 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Company       : Factor w/ 22 levels "Arizona","Boston",..: 1 2 3 4 13 5 6 7 8 9 ...
 $ Fixed_charge  : num [1:22] 1.06 0.89 1.43 1.02 1.49 1.32 1.22 1.1 1.34 1.12 ...
 $ RoR           : num [1:22] 9.2 10.3 15.4 11.2 8.8 13.5 12.2 9.2 13 12.4 ...
 $ Cost          : num [1:22] 151 202 113 168 192 111 175 245 168 197 ...
 $ Load_factor   : num [1:22] 54.4 57.9 53 56 51.2 60 67.6 57 60.4 53 ...
 $ Demand_growth: num [1:22] 1.6 2.2 3.4 0.3 1 -2.2 2.2 3.3 7.2 2.7 ...
 $ Sales         : num [1:22] 9077 5088 9212 6423 3300 ...
 $ Nuclear       : num [1:22] 0 25.3 0 34.3 15.6 22.5 0 0 0 39.2 ...
 $ Fuel_Cost     : num [1:22] 0.628 1.555 1.058 0.7 2.044 ...
 - attr(*, "spec")=
  .. cols(
  ..   Company = col_character(),
  ..   Fixed_charge = col_double(),
  ..   RoR = col_double(),
  ..   Cost = col_double(),
  ..   Load_factor = col_double(),
  ..   Demand_growth = col_double(),
  ..   Sales = col_double(),
  ..   Nuclear = col_double(),
  ..   Fuel_Cost = col_double()
  .. )
> head(utilities_tb)
# A tibble: 6 x 9
  Company      Fixed_charge   RoR  Cost Load_factor Demand_growth Sales Nuclear Fuel_Cost
  <fct>               <dbl> <dbl> <dbl>       <dbl>         <dbl> <dbl>   <dbl>     <dbl>
1 Arizona              1.06   9.2   151        54.4           1.6  9077     0       0.628
2 Boston               0.89  10.3   202        57.9           2.2  5088    25.3     1.56 
3 Central              1.43  15.4   113        53             3.4  9212     0       1.06 
4 Commonwealth         1.02  11.2   168        56             0.3  6423    34.3     0.7  
5 NY                   1.49   8.8   192        51.2           1    3300    15.6     2.04 
6 Florida              1.32  13.5   111        60            -2.2 11127    22.5     1.24 
> 
```

01.데이터 불러오기

```
# 데이터 분포 확인

ggplot(utilities_tb,
      aes(x = Sales,
          y = Fuel_Cost)) +
  geom_point(alpha = 0.3)
```

01.데이터 불러오기

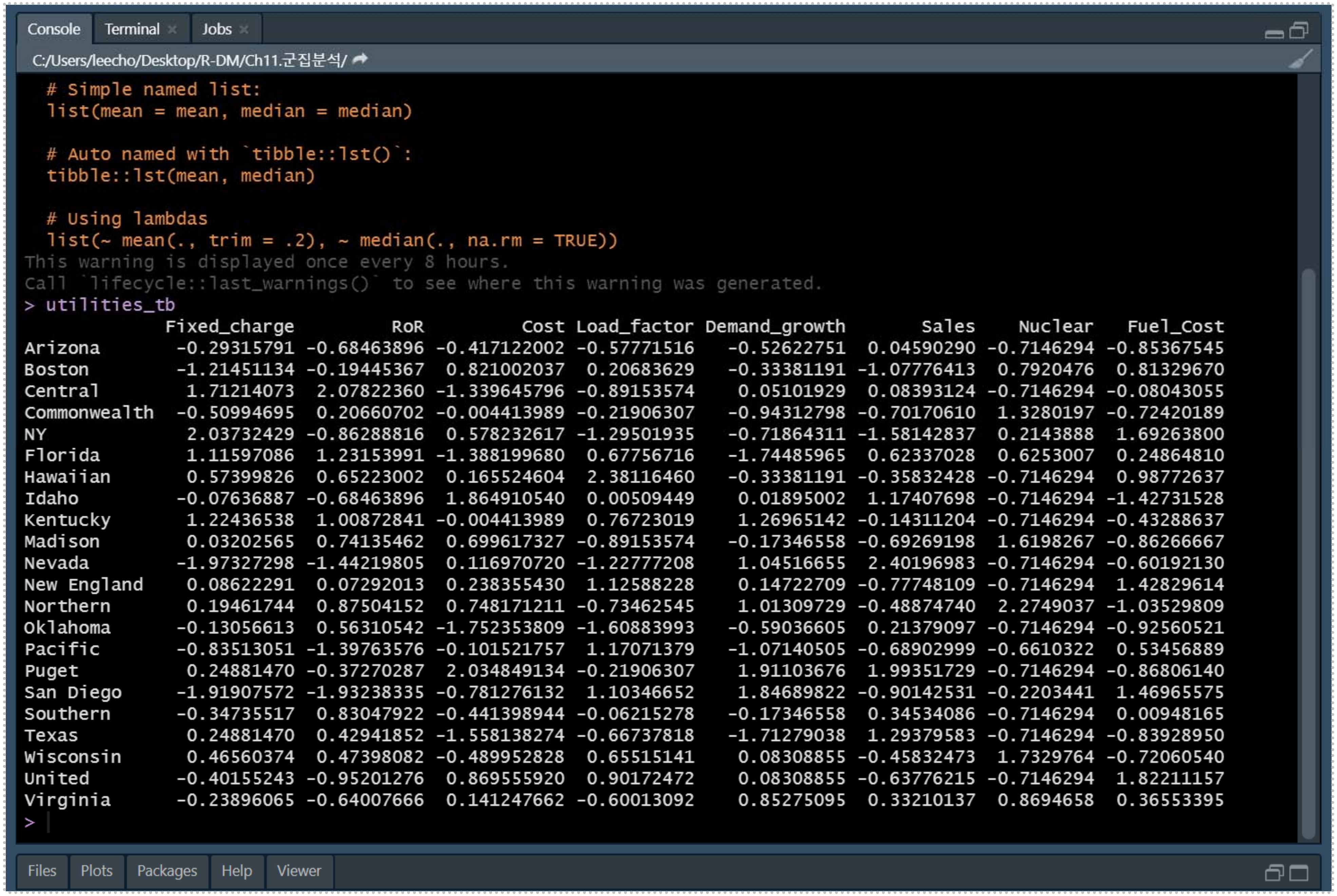


```
> ggplot(utilities_tb,
+        aes(x = Sales,
+            y = Fuel_Cost)) +
+   geom_point(alpha = 0.3)
>
```

02.데이터 정규화

```
# 02.데이터 정규화
# 데이터 정규화: mutate_if, 수치형 변수만 정규화
# 회사이름을 row 이름으로 변경

utilities_tb <-
  utilities_tb %>%
  mutate_if(is.numeric, funs(scale(.))) %>%
  column_to_rownames(var = "Company")

utilities_tb
```

02.데이터 정규화

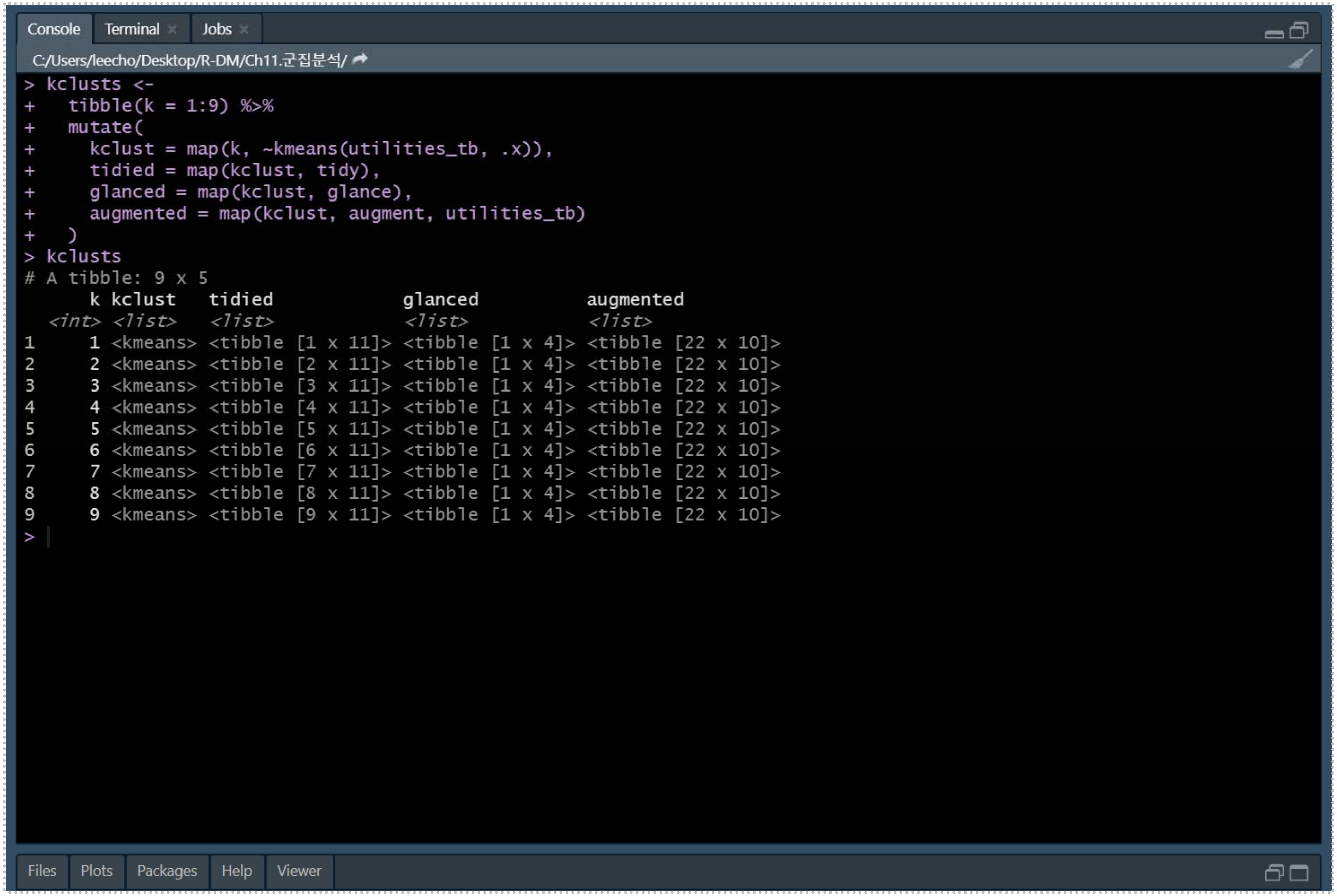


```
  # Simple named list:
  list(mean = mean, median = median)

  # Auto named with `tibble::lst()`:
  tibble::lst(mean, median)

  # Using lambdas
  list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was generated.
> utilities_tb
             Fixed_charge         RoR         Cost Load_factor Demand_growth       Sales    Nuclear   Fuel_Cost
Arizona       -0.29315791 -0.68463896 -0.417122002 -0.57771516   -0.52622751  0.04590290 -0.7146294 -0.85367545
Boston        -1.21451134 -0.19445367  0.821002037  0.20683629   -0.33381191 -1.07776413  0.7920476  0.81329670
Central        1.71214073  2.07822360 -1.339645796 -0.89153574    0.05101929  0.08393124 -0.7146294 -0.08043055
Commonwealth  -0.50994695  0.20660702 -0.004413989 -0.21906307   -0.94312798 -0.70170610  1.3280197 -0.72420189
NY             2.03732429 -0.86288816  0.578232617 -1.29501935   -0.71864311 -1.58142837  0.2143888  1.69263800
Florida        1.11597086  1.23153991 -1.388199680  0.67756716   -1.74485965  0.62337028  0.6253007  0.24864810
Hawaiian       0.57399826  0.65223002  0.165524604  2.38116460   -0.33381191 -0.35832428 -0.7146294  0.98772637
Idaho         -0.07636887 -0.68463896  1.864910540  0.00509449    0.01895002  1.17407698 -0.7146294 -1.42731528
Kentucky       1.22436538  1.00872841 -0.004413989  0.76723019    1.26965142 -0.14311204 -0.7146294 -0.43288637
Madison        0.03202565  0.74135462  0.699617327 -0.89153574   -0.17346558 -0.69269198  1.6198267 -0.86266667
Nevada        -1.97327298 -1.44219805  0.116970720 -1.22777208    1.04516655  2.40196983 -0.7146294 -0.60192130
New England    0.08622291  0.07292013  0.238355430  1.12588228    0.14722709 -0.77748109 -0.7146294  1.42829614
Northern       0.19461744  0.87504152  0.748171211 -0.73462545    1.01309729 -0.48874740  2.2749037 -1.03529809
Oklahoma      -0.13056613  0.56310542 -1.752353809 -1.60883993   -0.59036605  0.21379097 -0.7146294 -0.92560521
Pacific       -0.83513051 -1.39763576 -0.101521757  1.17071379   -1.07140505 -0.68902999 -0.6610322  0.53456889
Puget          0.24881470 -0.37270287  2.034849134 -0.21906307    1.91103676  1.99351729 -0.7146294 -0.86806140
San Diego     -1.91907572 -1.93238335 -0.781276132  1.10346652    1.84689822 -0.90142531 -0.2203441  1.46965575
Southern      -0.34735517  0.83047922 -0.441398944 -0.06215278   -0.17346558  0.34534086 -0.7146294  0.00948165
Texas          0.24881470  0.42941852 -1.558138274 -0.66737818   -1.71279038  1.29379583 -0.7146294 -0.83928950
Wisconsin      0.46560374  0.47398082 -0.489952828  0.65515141    0.08308855 -0.45832473  1.7329764 -0.72060540
United        -0.40155243 -0.95201276  0.869555920  0.90172472    0.08308855 -0.63776215 -0.7146294  1.82211157
Virginia      -0.23896065 -0.64007666  0.141247662 -0.60013092    0.85275095  0.33210137  0.8694658  0.36553395
> 
```

03.최적 군집수 찾기

```
# 03.최적 군집수 찾기
# 최적 군집수를 찾는 엘보우(Elbow) 차트

# 군집 9개

kclusts <-
  tibble(k = 1:9) %>%
  mutate(
    kclust = map(k, ~kmeans(utilities_tb, .x)),
    tidied = map(kclust, tidy),
    glanced = map(kclust, glance),
    augmented = map(kclust, augment, utilities_tb)
  )

kclusts
```

03.최적 군집수 찾기

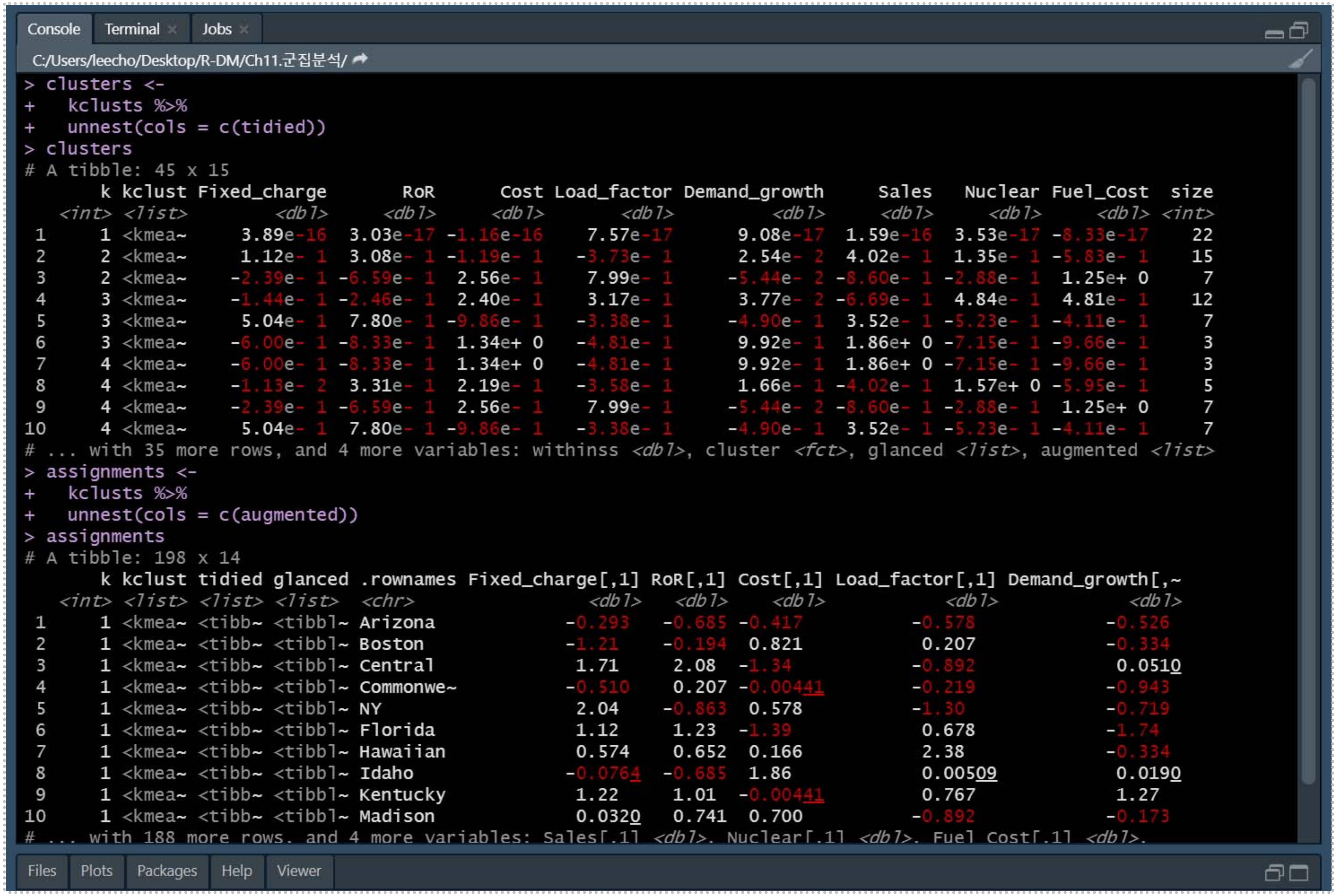


```
> kclusts <-
+   tibble(k = 1:9) %>%
+   mutate(
+     kclust = map(k, ~kmeans(utilities_tb, .x)),
+     tidied = map(kclust, tidy),
+     glanced = map(kclust, glance),
+     augmented = map(kclust, augment, utilities_tb)
+   )
> kclusts
# A tibble: 9 x 5
      k kclust   tidied             glanced          augmented
  <int> <list>   <list>             <list>           <list>
1     1 <kmeans> <tibble [1 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
2     2 <kmeans> <tibble [2 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
3     3 <kmeans> <tibble [3 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
4     4 <kmeans> <tibble [4 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
5     5 <kmeans> <tibble [5 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
6     6 <kmeans> <tibble [6 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
7     7 <kmeans> <tibble [7 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
8     8 <kmeans> <tibble [8 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
9     9 <kmeans> <tibble [9 x 11]> <tibble [1 x 4]> <tibble [22 x 10]>
>
```

03.최적 군집수 찾기

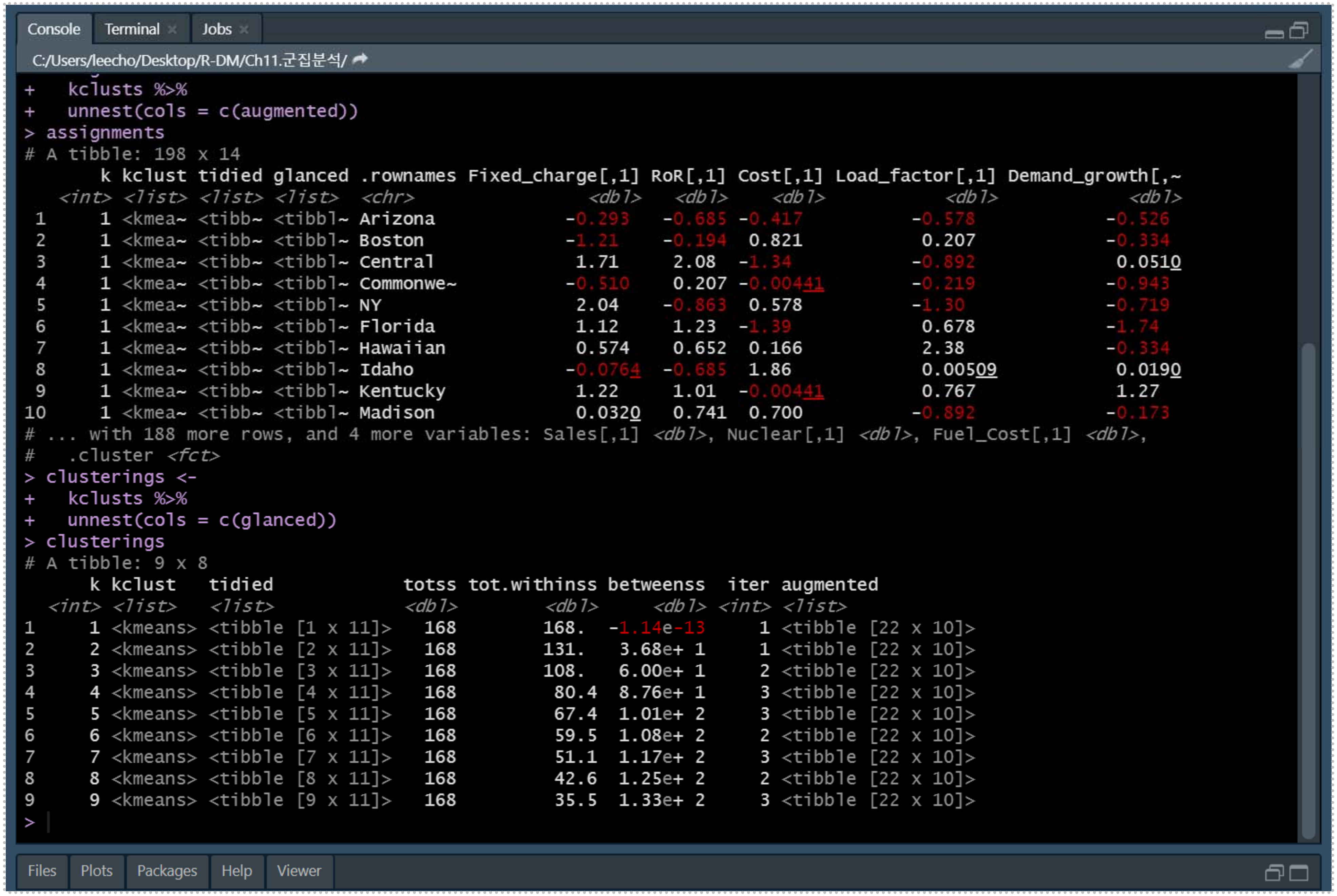
```
clusters <-
  kclusts %>%
  unnest(cols = c(tidied))

assignments <-
  kclusts %>%
  unnest(cols = c(augmented))

clusterings <-
  kclusts %>%
  unnest(cols = c(glanced))
```

03.최적 군집수 찾기



```
> clusters <-
+   kclusts %>%
+   unnest(cols = c(tidied))
> clusters
# A tibble: 45 x 15
       k kclust Fixed_charge       RoR      Cost Load_factor Demand_growth     Sales   Nuclear Fuel_Cost  size
   <int> <list>        <dbl>     <dbl>     <dbl>       <dbl>         <dbl>     <dbl>     <dbl>     <dbl> <int>
 1     1 <kmea~     3.89e-16  3.03e-17 -1.16e-16    7.57e-17      9.08e-17  1.59e-16  3.53e-17 -8.33e-17    22
 2     2 <kmea~     1.12e- 1  3.08e- 1 -1.19e- 1   -3.73e- 1      2.54e- 2  4.02e- 1  1.35e- 1 -5.83e- 1    15
 3     2 <kmea~    -2.39e- 1 -6.59e- 1  2.56e- 1    7.99e- 1     -5.44e- 2 -8.60e- 1 -2.88e- 1  1.25e+ 0     7
 4     3 <kmea~    -1.44e- 1 -2.46e- 1  2.40e- 1    3.17e- 1      3.77e- 2 -6.69e- 1  4.84e- 1  4.81e- 1    12
 5     3 <kmea~     5.04e- 1  7.80e- 1 -9.86e- 1   -3.38e- 1     -4.90e- 1  3.52e- 1 -5.23e- 1 -4.11e- 1     7
 6     3 <kmea~    -6.00e- 1 -8.33e- 1  1.34e+ 0   -4.81e- 1      9.92e- 1  1.86e+ 0 -7.15e- 1 -9.66e- 1     3
 7     4 <kmea~    -6.00e- 1 -8.33e- 1  1.34e+ 0   -4.81e- 1      9.92e- 1  1.86e+ 0 -7.15e- 1 -9.66e- 1     3
 8     4 <kmea~    -1.13e- 2  3.31e- 1  2.19e- 1   -3.58e- 1      1.66e- 1 -4.02e- 1  1.57e+ 0 -5.95e- 1     5
 9     4 <kmea~    -2.39e- 1 -6.59e- 1  2.56e- 1    7.99e- 1     -5.44e- 2 -8.60e- 1 -2.88e- 1  1.25e+ 0     7
10     4 <kmea~     5.04e- 1  7.80e- 1 -9.86e- 1   -3.38e- 1     -4.90e- 1  3.52e- 1 -5.23e- 1 -4.11e- 1     7
# ... with 35 more rows, and 4 more variables: withinss <dbl>, cluster <fct>, glanced <list>, augmented <list>
> assignments <-
+   kclusts %>%
+   unnest(cols = c(augmented))
> assignments
# A tibble: 198 x 14
       k kclust tidied glanced .rownames Fixed_charge[,1] RoR[,1] Cost[,1] Load_factor[,1] Demand_growth[,~
   <int> <list> <list> <list>  <chr>                <dbl>   <dbl>    <dbl>           <dbl>            <dbl>
 1     1 <kmea~ <tibb~ <tibbl~ Arizona             -0.293  -0.685 -0.417           -0.578           -0.526
 2     1 <kmea~ <tibb~ <tibbl~ Boston              -1.21   -0.194  0.821            0.207           -0.334
 3     1 <kmea~ <tibb~ <tibbl~ Central              1.71    2.08  -1.34            -0.892            0.0510
 4     1 <kmea~ <tibb~ <tibbl~ Commonwe~           -0.510   0.207 -0.00441         -0.219           -0.943
 5     1 <kmea~ <tibb~ <tibbl~ NY                   2.04   -0.863  0.578           -1.30            -0.719
 6     1 <kmea~ <tibb~ <tibbl~ Florida              1.12    1.23  -1.39             0.678           -1.74
 7     1 <kmea~ <tibb~ <tibbl~ Hawaiian             0.574   0.652  0.166            2.38            -0.334
 8     1 <kmea~ <tibb~ <tibbl~ Idaho               -0.0764 -0.685  1.86             0.00509          0.0190
 9     1 <kmea~ <tibb~ <tibbl~ Kentucky             1.22    1.01  -0.00441          0.767            1.27
10     1 <kmea~ <tibb~ <tibbl~ Madison              0.0320  0.741  0.700           -0.892           -0.173
# ... with 188 more rows, and 4 more variables: Sales[,1] <dbl>, Nuclear[,1] <dbl>, Fuel_Cost[,1] <dbl>,
```

03.최적 군집수 찾기

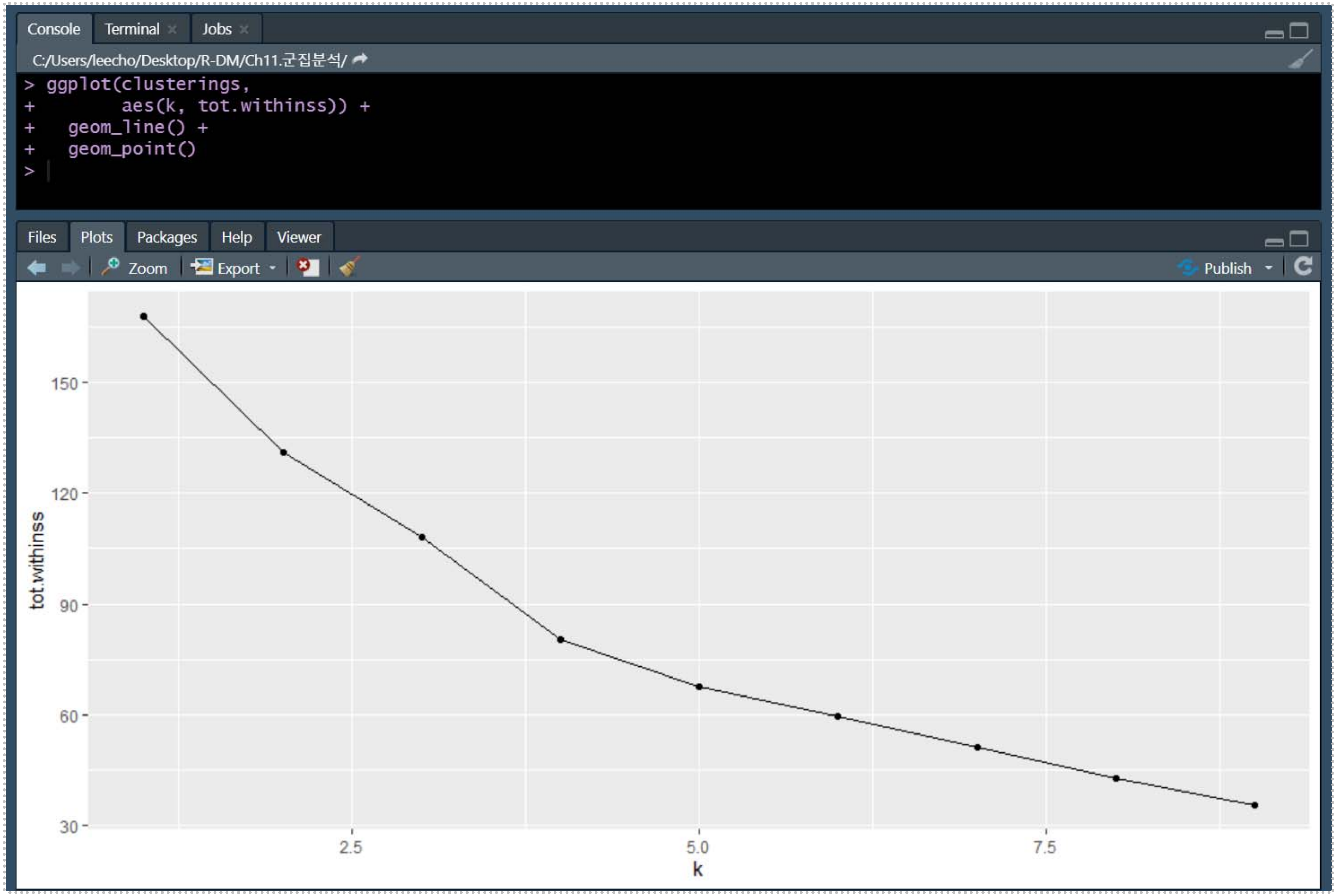


```
+   kclusts %>%
+   unnest(cols = c(augmented))
> assignments
# A tibble: 198 x 14
       k kclust tidied glanced .rownames Fixed_charge[,1] RoR[,1] Cost[,1] Load_factor[,1] Demand_growth[,~
   <int> <list> <list> <list>  <chr>                <dbl>   <dbl>    <dbl>           <dbl>            <dbl>
 1     1 <kmea~ <tibb~ <tibbl~ Arizona            -0.293   -0.685 -0.417          -0.578            -0.526
 2     1 <kmea~ <tibb~ <tibbl~ Boston             -1.21    -0.194  0.821           0.207            -0.334
 3     1 <kmea~ <tibb~ <tibbl~ Central             1.71     2.08  -1.34           -0.892             0.0510
 4     1 <kmea~ <tibb~ <tibbl~ Commonwe~          -0.510    0.207 -0.00441        -0.219            -0.943
 5     1 <kmea~ <tibb~ <tibbl~ NY                  2.04    -0.863  0.578          -1.30             -0.719
 6     1 <kmea~ <tibb~ <tibbl~ Florida             1.12     1.23  -1.39            0.678            -1.74
 7     1 <kmea~ <tibb~ <tibbl~ Hawaiian            0.574    0.652  0.166           2.38             -0.334
 8     1 <kmea~ <tibb~ <tibbl~ Idaho              -0.0764  -0.685  1.86            0.00509           0.0190
 9     1 <kmea~ <tibb~ <tibbl~ Kentucky            1.22     1.01  -0.00441         0.767             1.27
10     1 <kmea~ <tibb~ <tibbl~ Madison             0.0320   0.741  0.700          -0.892            -0.173
# ... with 188 more rows, and 4 more variables: Sales[,1] <dbl>, Nuclear[,1] <dbl>, Fuel_Cost[,1] <dbl>,
#   .cluster <fct>
> clusterings <-
+   kclusts %>%
+   unnest(cols = c(glanced))
> clusterings
# A tibble: 9 x 8
      k kclust   tidied            totss tot.withinss betweenss  iter augmented
  <int> <list>   <list>            <dbl>        <dbl>     <dbl> <int> <list>
1     1 <kmeans> <tibble [1 x 11]>   168        168.  -1.14e-13     1 <tibble [22 x 10]>
2     2 <kmeans> <tibble [2 x 11]>   168        131.   3.68e+ 1     1 <tibble [22 x 10]>
3     3 <kmeans> <tibble [3 x 11]>   168        108.   6.00e+ 1     2 <tibble [22 x 10]>
4     4 <kmeans> <tibble [4 x 11]>   168         80.4  8.76e+ 1     3 <tibble [22 x 10]>
5     5 <kmeans> <tibble [5 x 11]>   168         67.4  1.01e+ 2     3 <tibble [22 x 10]>
6     6 <kmeans> <tibble [6 x 11]>   168         59.5  1.08e+ 2     2 <tibble [22 x 10]>
7     7 <kmeans> <tibble [7 x 11]>   168         51.1  1.17e+ 2     3 <tibble [22 x 10]>
8     8 <kmeans> <tibble [8 x 11]>   168         42.6  1.25e+ 2     2 <tibble [22 x 10]>
9     9 <kmeans> <tibble [9 x 11]>   168         35.5  1.33e+ 2     3 <tibble [22 x 10]>
>
```

03.최적 군집수 찾기

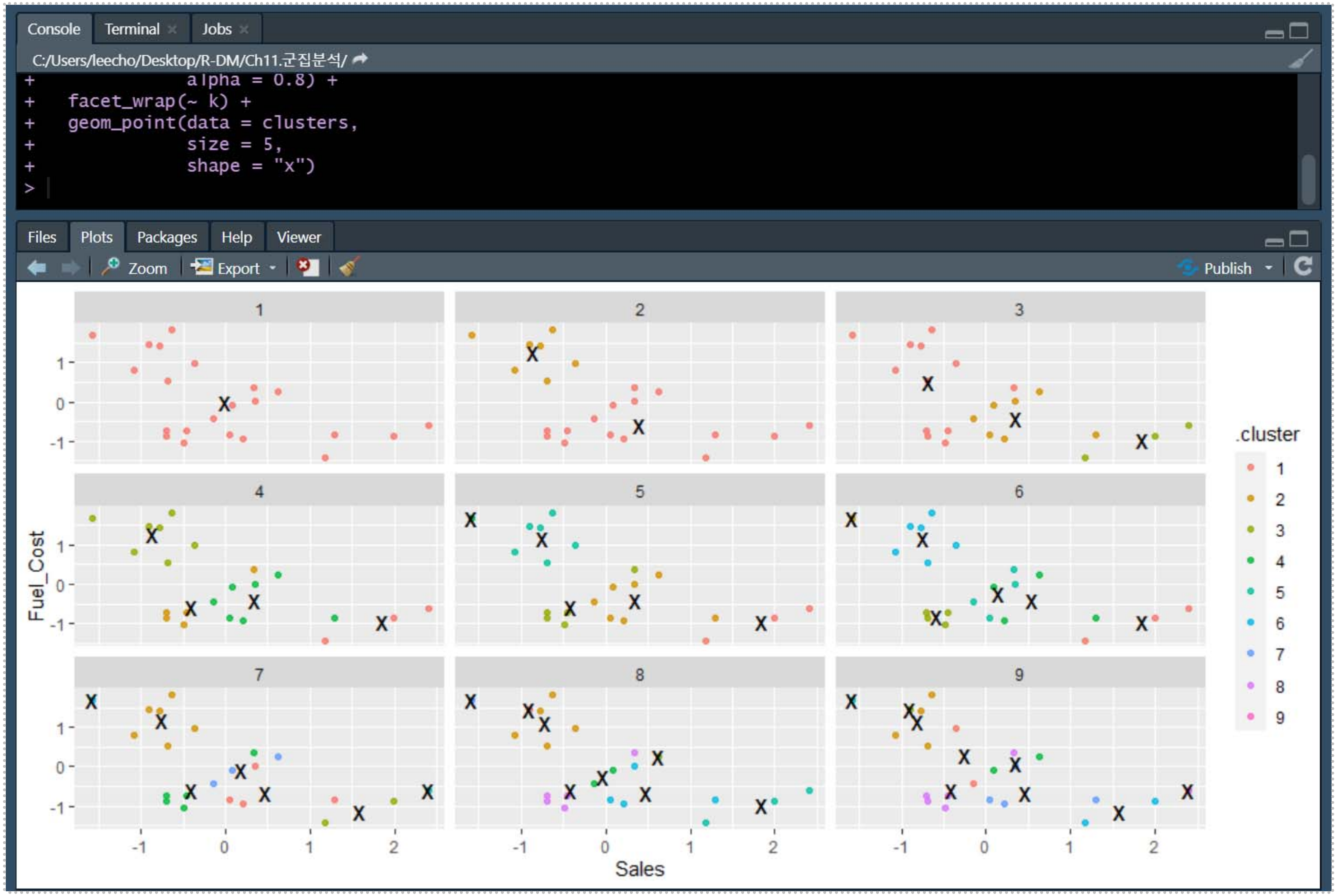
```
# 엘보우(Elbow) 챠트

ggplot(clusterings,
       aes(k, tot.withinss)) +
  geom_line() +
  geom_point()

# 군집별 그래프

ggplot(assignments,
       aes(x = Sales,
           y = Fuel_Cost)) +
  geom_point(aes(color = .cluster),
             alpha = 0.8) +
  facet_wrap(~ k) +
  geom_point(data = clusters,
             size = 5,
             shape = "x")
```

03.최적 군집수 찾기



```
> ggplot(clusterings,
+        aes(k, tot.withinss)) +
+   geom_line() +
+   geom_point()
>
```

03.최적 군집수 찾기

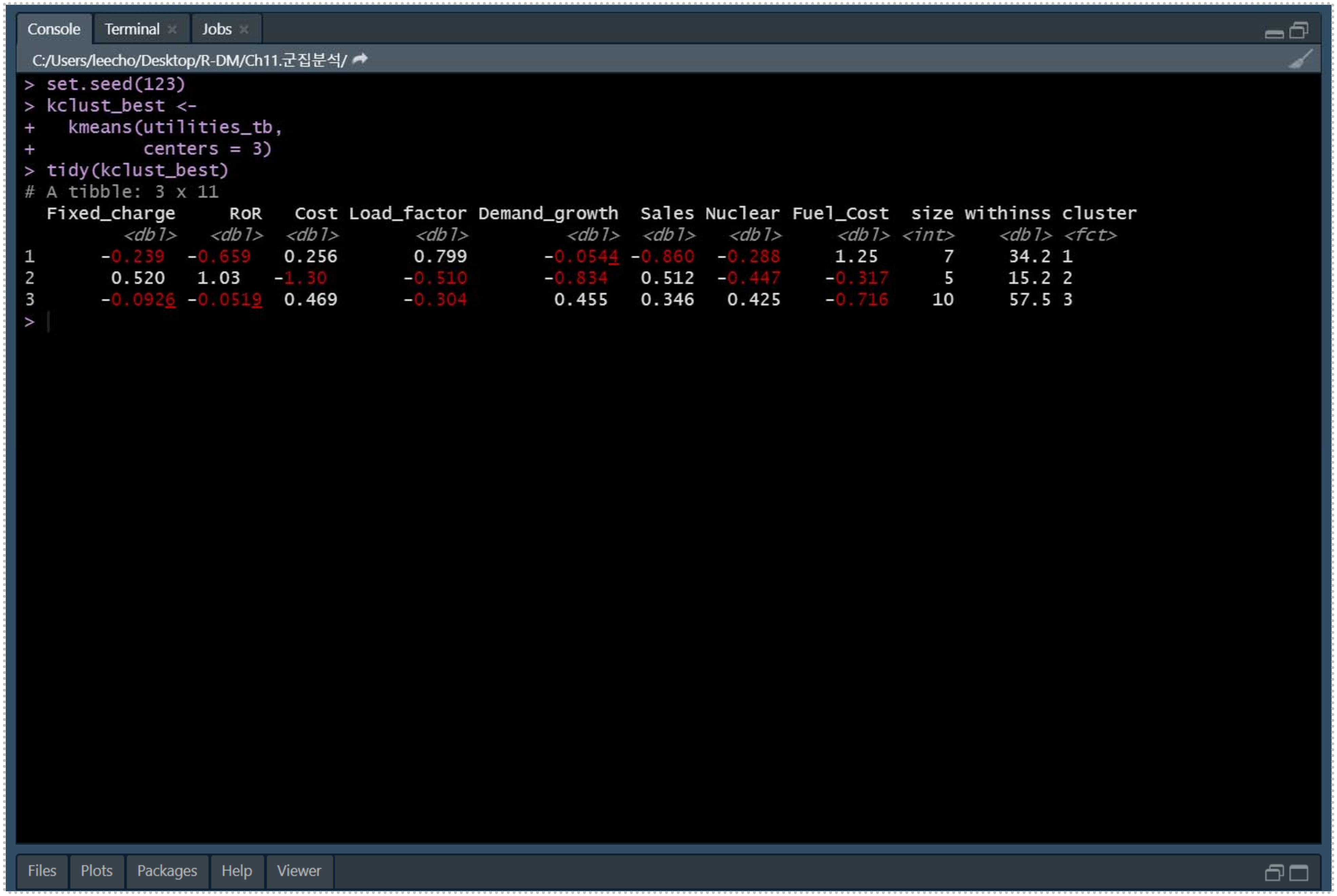


```
+             alpha = 0.8) +
+   facet_wrap(~ k) +
+   geom_point(data = clusters,
+             size = 5,
+             shape = "x")
> 
```

04.best K-mean clustering

```
#  best model 구축

set.seed(123)

kclust_best <-
  kmeans(utilities_tb,
         centers = 3)

# 군집분석 결과 확인

tidy(kclust_best)
```

04.best K-mean clustering

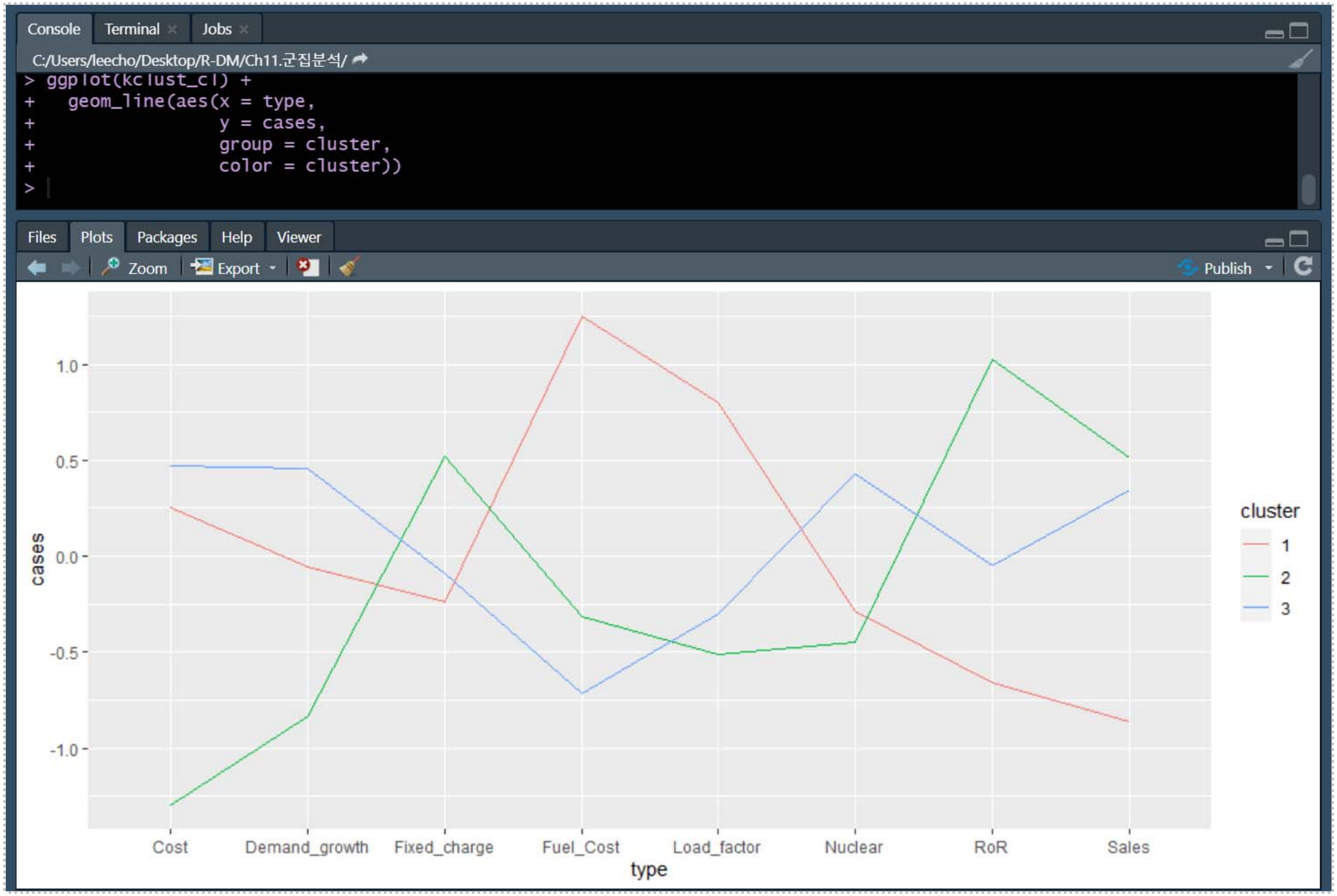


```
> set.seed(123)
> kclust_best <-
+   kmeans(utilities_tb,
+          centers = 3)
> tidy(kclust_best)
# A tibble: 3 x 11
  Fixed_charge     RoR   Cost Load_factor Demand_growth  Sales Nuclear Fuel_Cost  size withinss cluster
         <dbl>   <dbl>  <dbl>       <dbl>         <dbl>  <dbl>   <dbl>     <dbl> <int>    <dbl> <fct>  
1      -0.239  -0.659   0.256       0.799       -0.0544 -0.860  -0.288      1.25      7     34.2 1      
2       0.520   1.03   -1.30       -0.510       -0.834   0.512  -0.447     -0.317     5     15.2 2      
3      -0.0926 -0.0519  0.469      -0.304        0.455   0.346   0.425     -0.716    10     57.5 3      
> 
```

05.군집별 특성 파악

```
kclust_cl <-
  tidy(kclust_best) %>%
  select(-c(size, withinss))

kclust_cl <-
  kclust_cl %>%
  pivot_longer(c("Fixed_charge", #c("1999, 2000")에러남
                 "Cost",
                 "RoR",
                 "Load_factor",
                 "Demand_growth",
                 "Sales",
                 "Nuclear",
                 "Fuel_Cost"),
               names_to = "type",
               values_to = "cases")

ggplot(kclust_cl) +
  geom_line(aes(x = type,
```

05.군집별 특성 파악



```
> ggplot(kclust_cl) +
+   geom_line(aes(x = type,
+                 y = cases,
+                 group = cluster,
+                 color = cluster))
>
```

참고자료

동영상 및 참고교재

- International
  - Tidymodels, https://www.tidymodels.org/
  - R for Data Science, https://r4ds.had.co.nz/
  - Statistical Inference via Data Science, https://www.tidymodels.org/books/moderndive/
  - Data Mining for Business Analytics_ Concepts, Techniques, and Applications in R, Shmueli et al., WILEY
- Domestic
  - K-MOOC
    - 경영데이터마이닝, 김종우, 한양대학교
    - 빅데이터의 세계, 원리와 응용, 신경식, 이화여자대학교
    - 빅데이터와 텍스트마이닝, 이신행, 세종대학교
  - 데이터마이닝 R edition, 조재희 외, 이앤비플러스
  - 김종우, 김선태, 경영을 위한 데이터마이닝, 한경사, 2009.
  - 경영정보시스템, 김우주 외, 시그마프레스