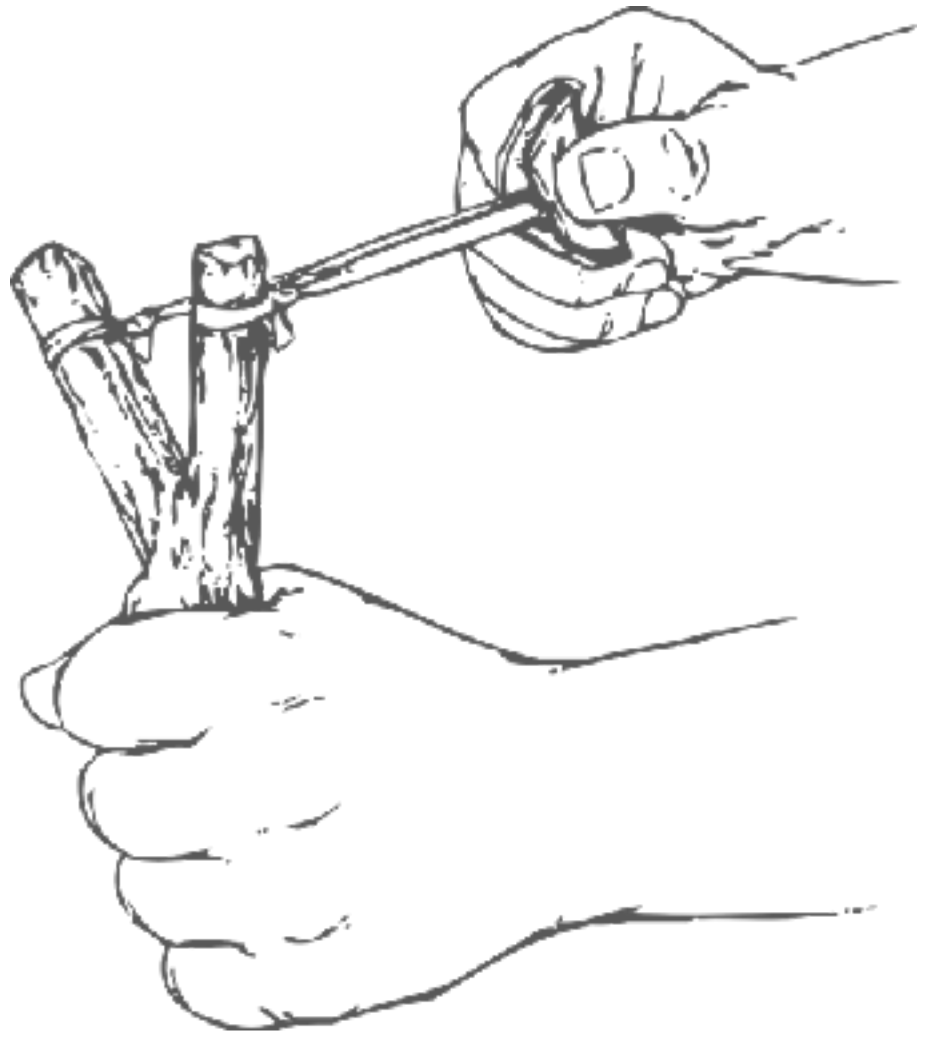
INTRODUCTION TO DATA ANALYSIS

DATA WRANGLING

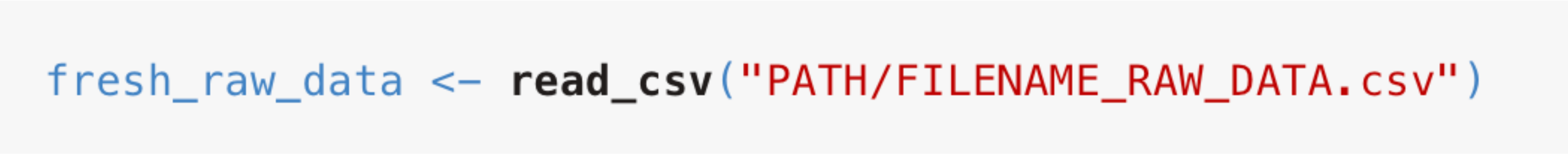
LEARNING GOALS



- be able to read from and write data to files
- understand notion of tidy data
- be able to solve common problems of data preprocessing

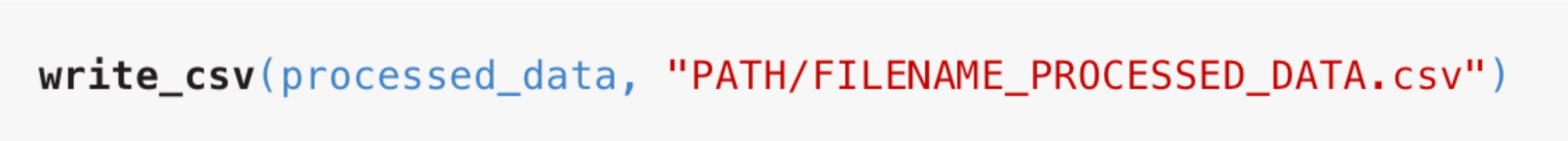
DATA I/O

- use functions for readr package
- preferred data format is CSV (in this course)
- read data from file



```
fresh_raw_data <- read_csv("PATH/FILENAME_RAW_DATA.csv")
```

- write data to file

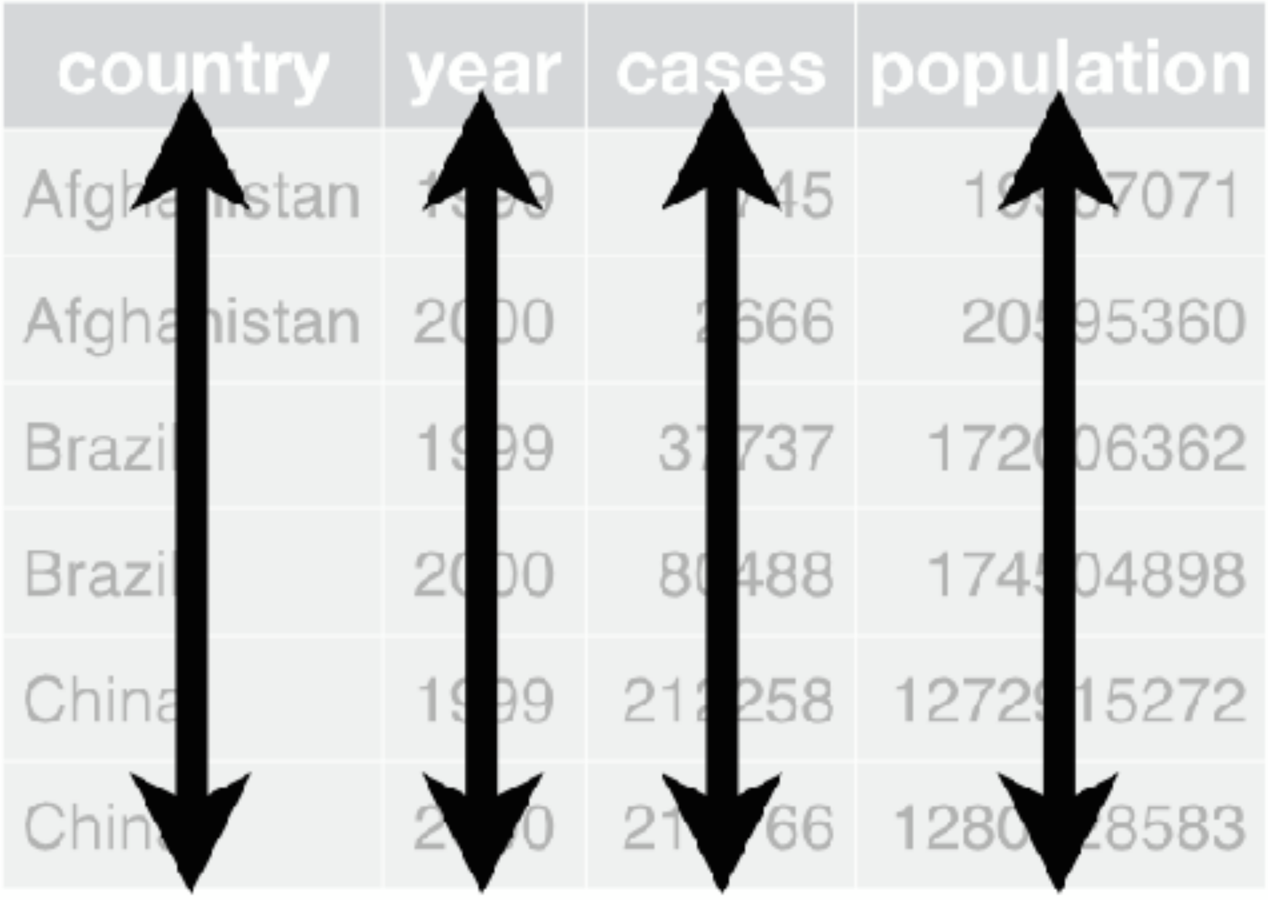


```
write_csv(processed_data, "PATH/FILENAME_PROCESSED_DATA.csv")
```

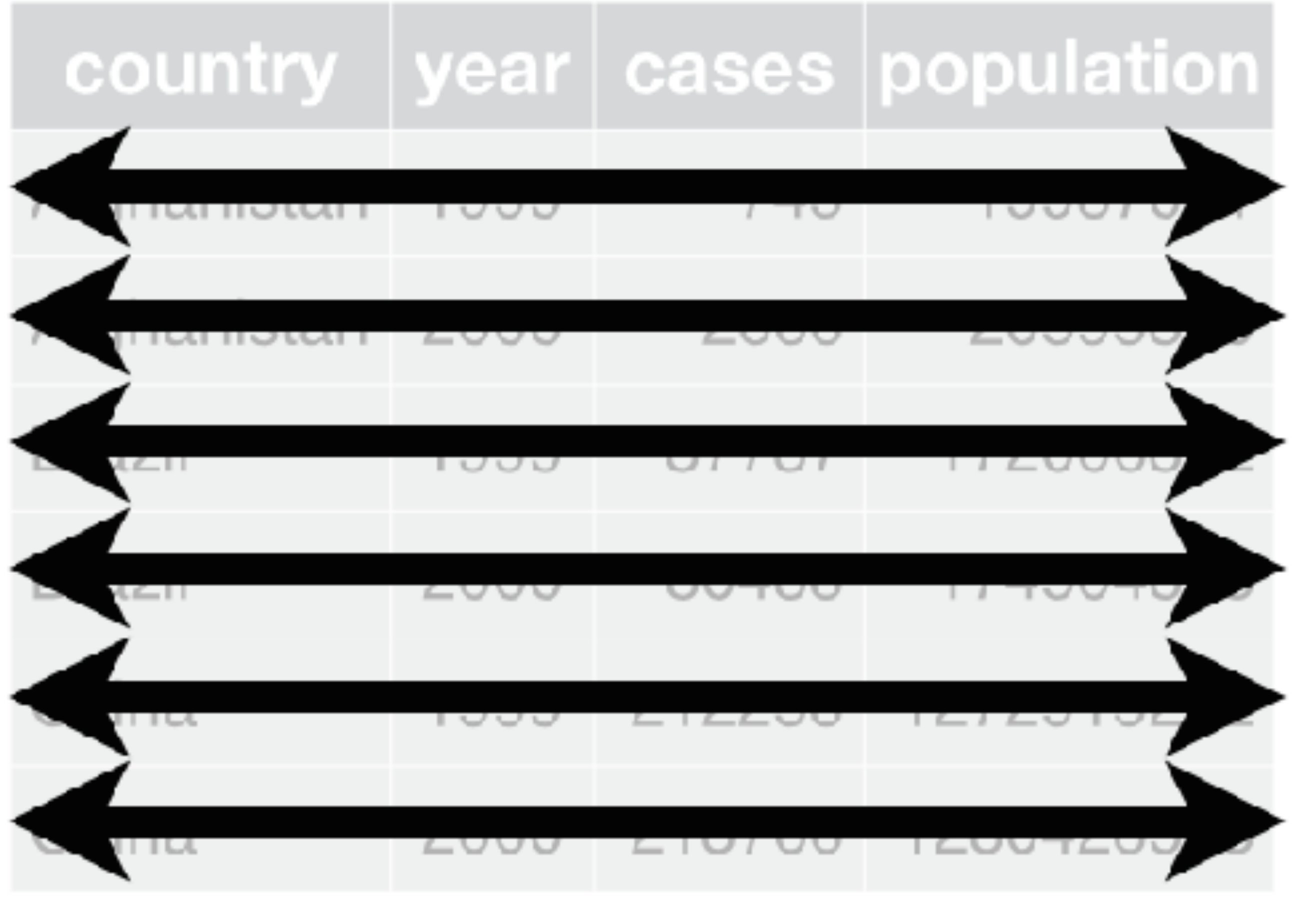
TIDY DATA

- data is tidy data if it satisfies three constraints:
  1. each variable forms a column
  2. each observation forms a row
  3. each type of observational unit forms a table

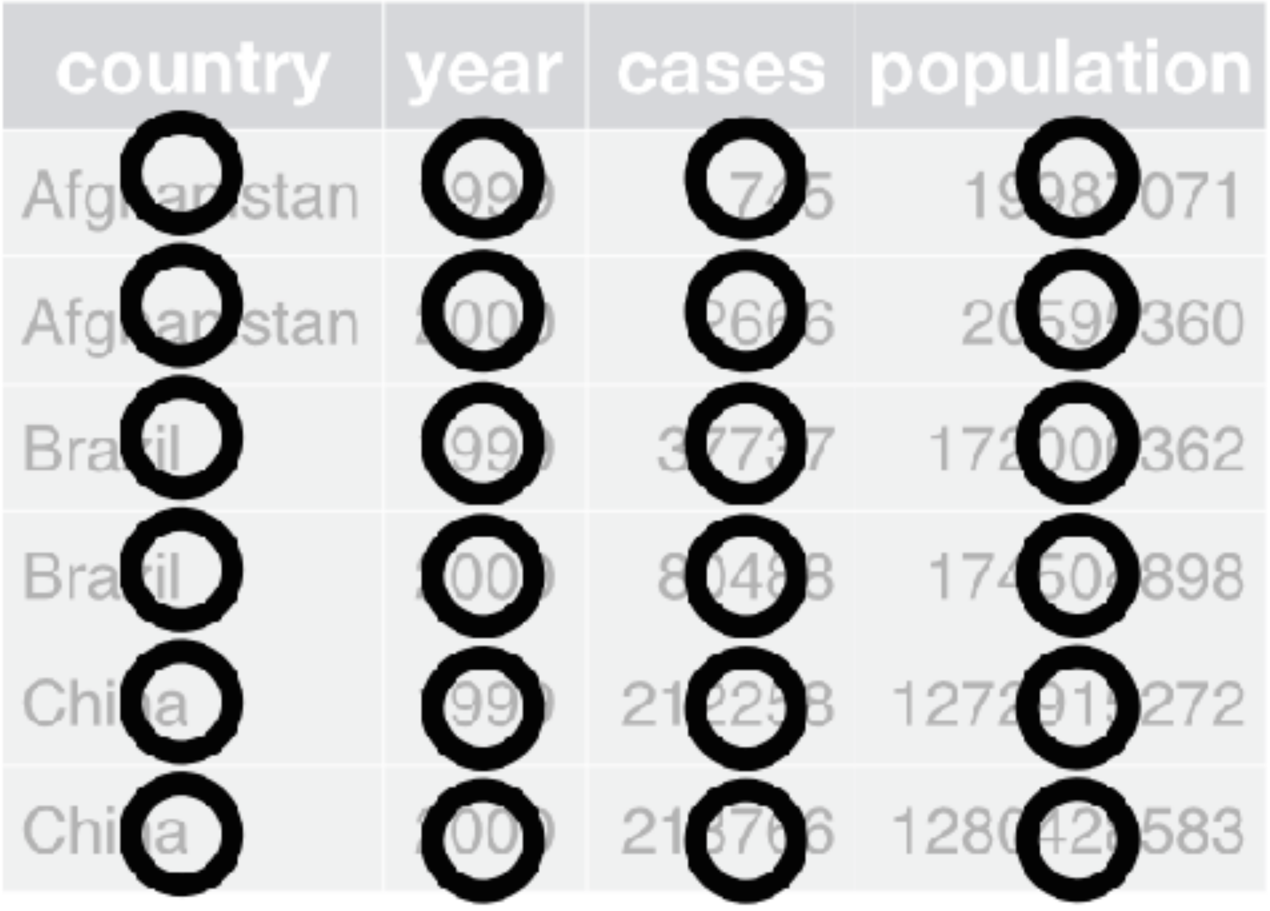
- data which is not tidy is messy
- data that satisfies 1 & 2 is almost tidy



variables

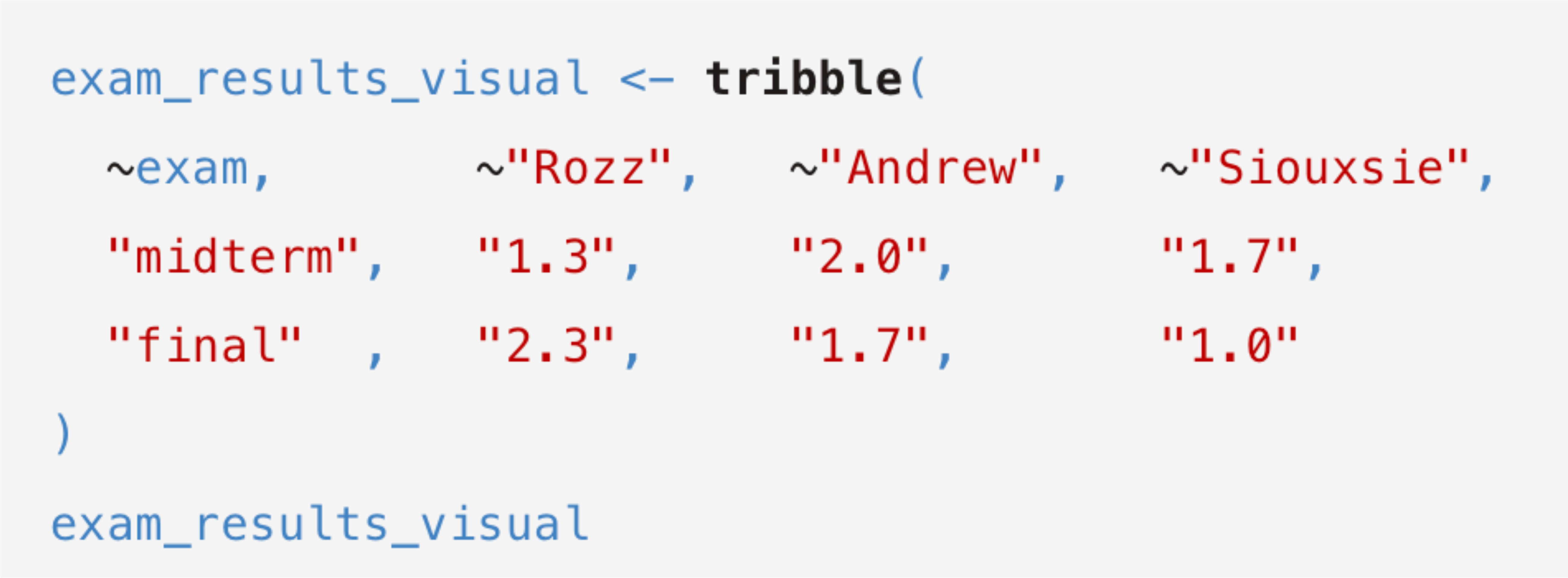


observations

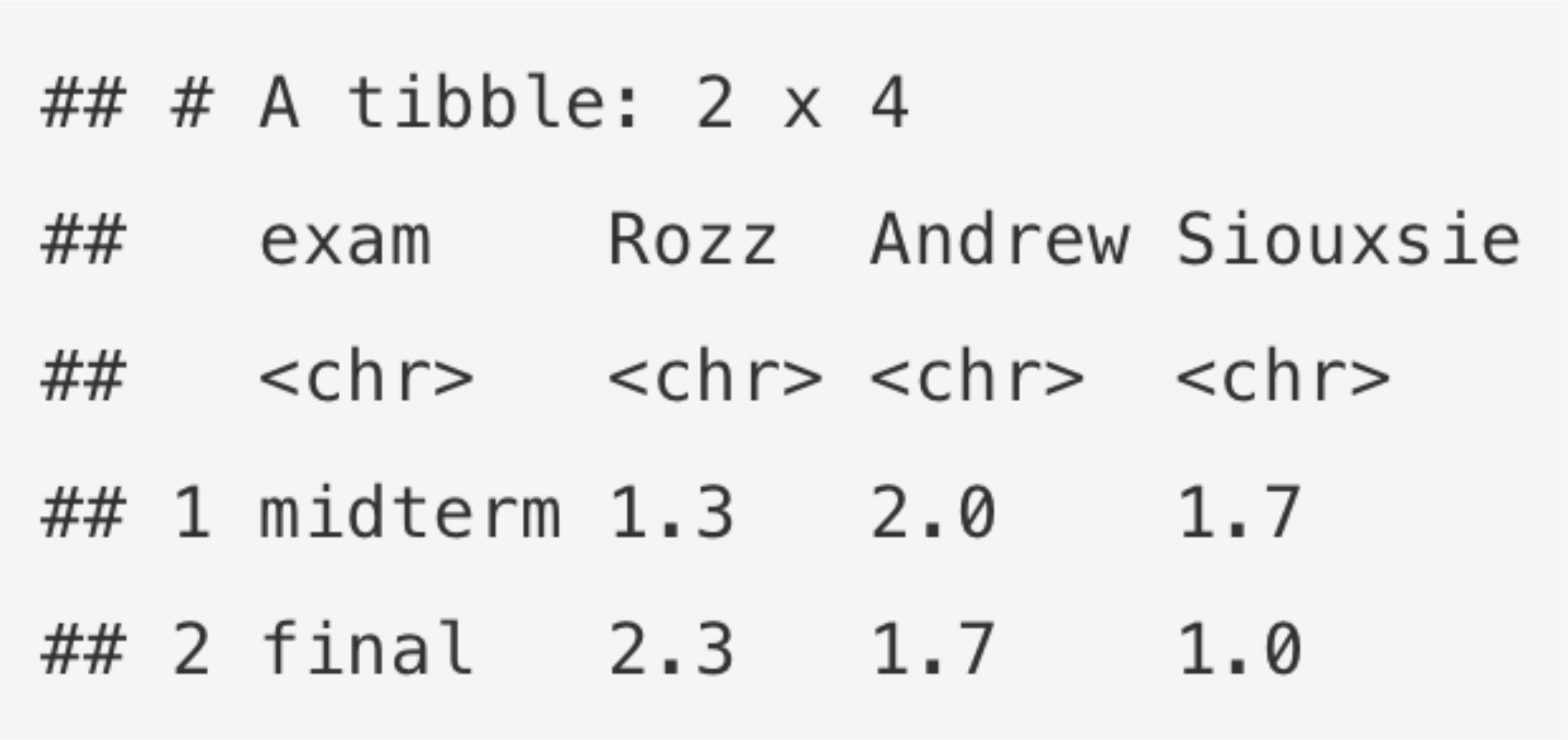


values

VISUALLY APPETIZING BUT MESSY DATA

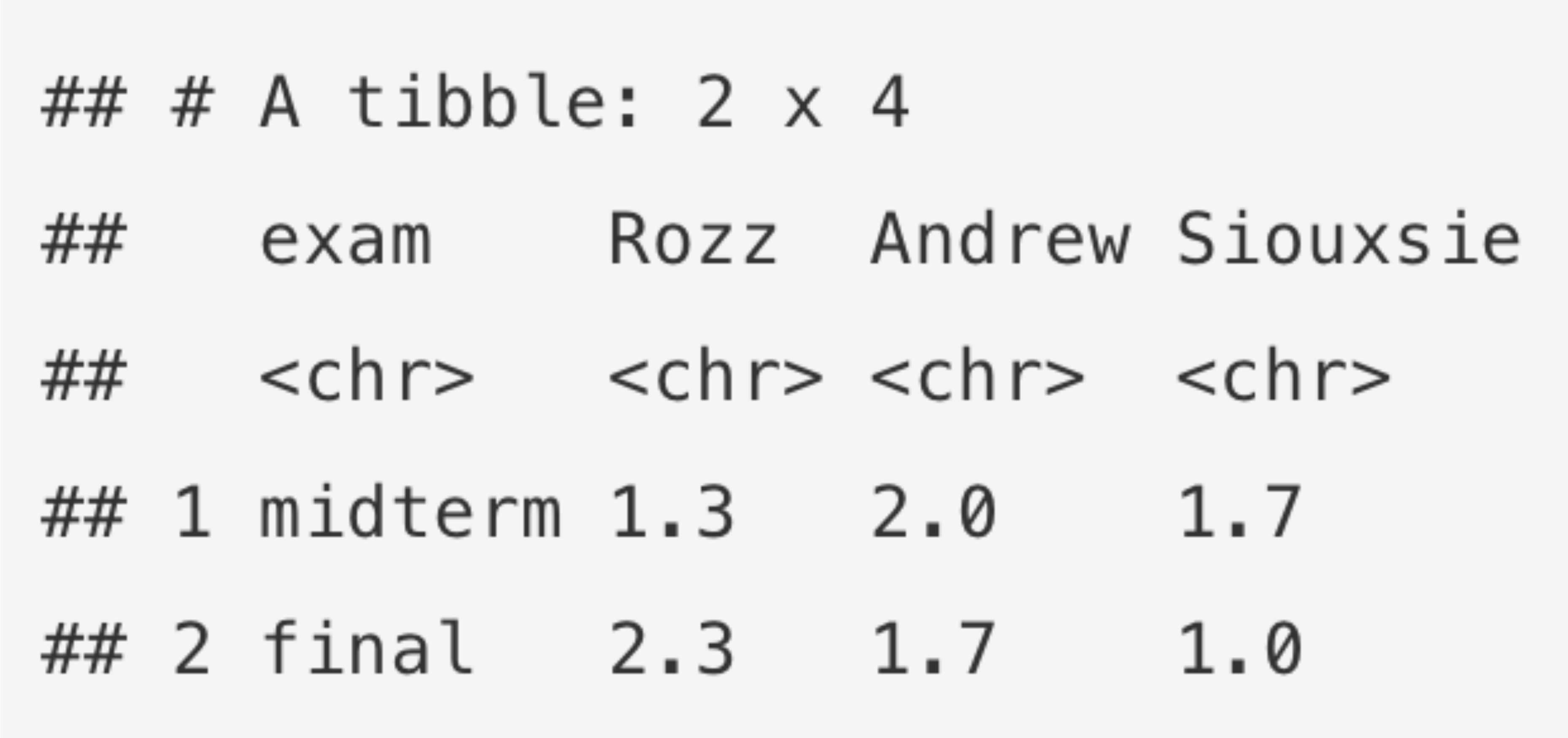


```r
exam_results_visual <- tribble(
  ~exam,       ~"Rozz",   ~"Andrew",   ~"Siouxsie",
  "midterm",   "1.3",     "2.0",       "1.7",
  "final"  ,   "2.3",     "1.7",       "1.0"
)
exam_results_visual
```



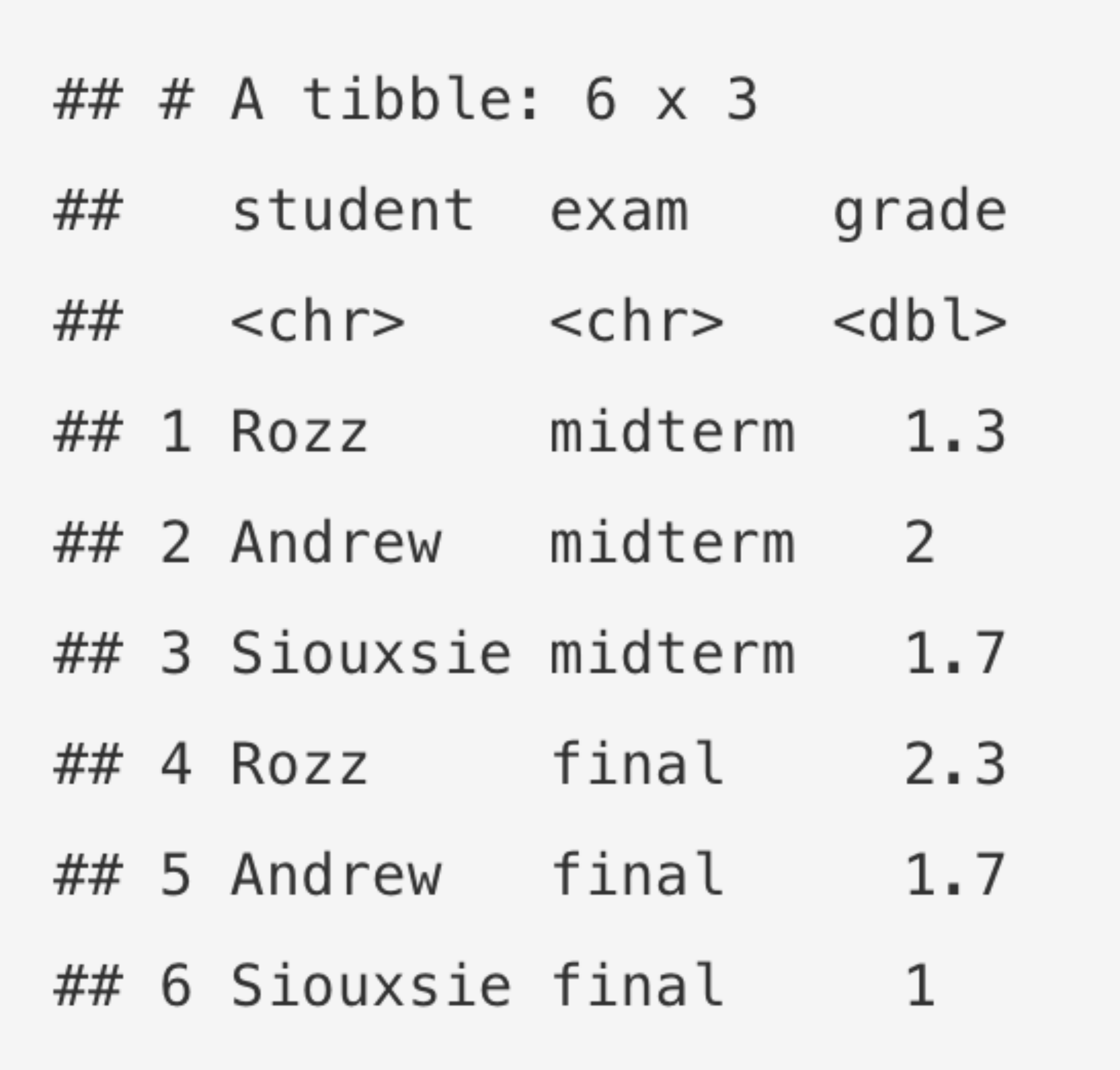
```
## # A tibble: 2 x 4
##   exam    Rozz  Andrew Siouxsie
##   <chr>   <chr> <chr>  <chr>
## 1 midterm 1.3   2.0    1.7
## 2 final   2.3   1.7    1.0
```

MESSY DATA



```
## # A tibble: 2 x 4
##   exam    Rozz  Andrew Siouxsie
##   <chr>   <chr> <chr>  <chr>
## 1 midterm 1.3   2.0    1.7
## 2 final   2.3   1.7    1.0
```

TIDY DATA



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

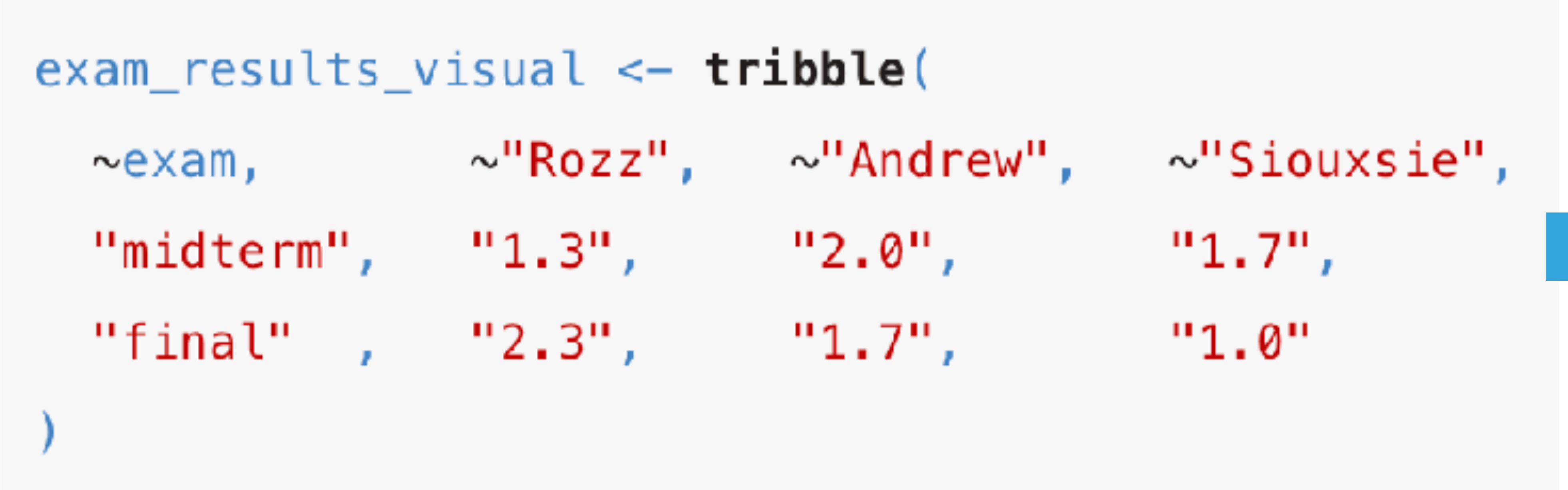
EXCURSION: MESSINESS FROM REDUNDANCY

```
## # A tibble: 6 x 4
##   student  stu_number exam    grade
##   <chr>    <chr>      <chr>   <dbl>
## 1 Rozz     666        midterm   1.3
## 2 Andrew   1969       midterm   2
## 3 Siouxsie 3.14       midterm   1.7
## 4 Rozz     666        final     2.3
## 5 Andrew   1969       final     1.7
## 6 Siouxsie 3.14       final     1
```

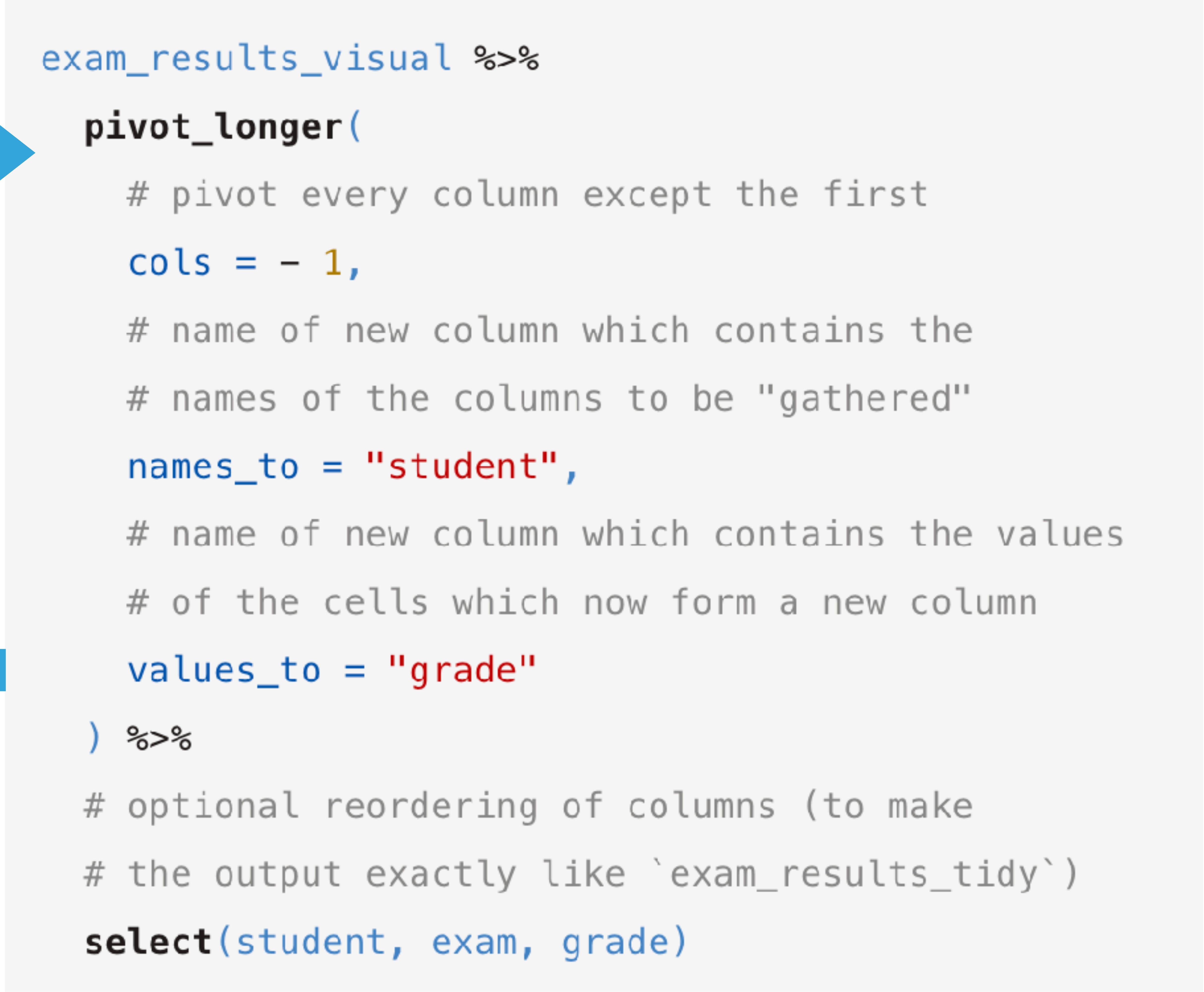
```
# same as before
exam_results_tidy <- tribble(
  ~student,    ~exam,     ~grade,
  "Rozz",      "midterm", 1.3,
  "Andrew",    "midterm", 2.0,
  "Siouxsie",  "midterm", 1.7,
  "Rozz",      "final",   2.3,
  "Andrew",    "final",   1.7,
  "Siouxsie",  "final",   1.0
)
# additional table with student numbers
student_numbers <- tribble(
  ~student,    ~student_number,
  "Rozz",      "666",
  "Andrew",    "1969",
  "Siouxsie",  "3.14"
)
```

```
full_join(exam_results_tidy, student_numbers, by = "student")
```

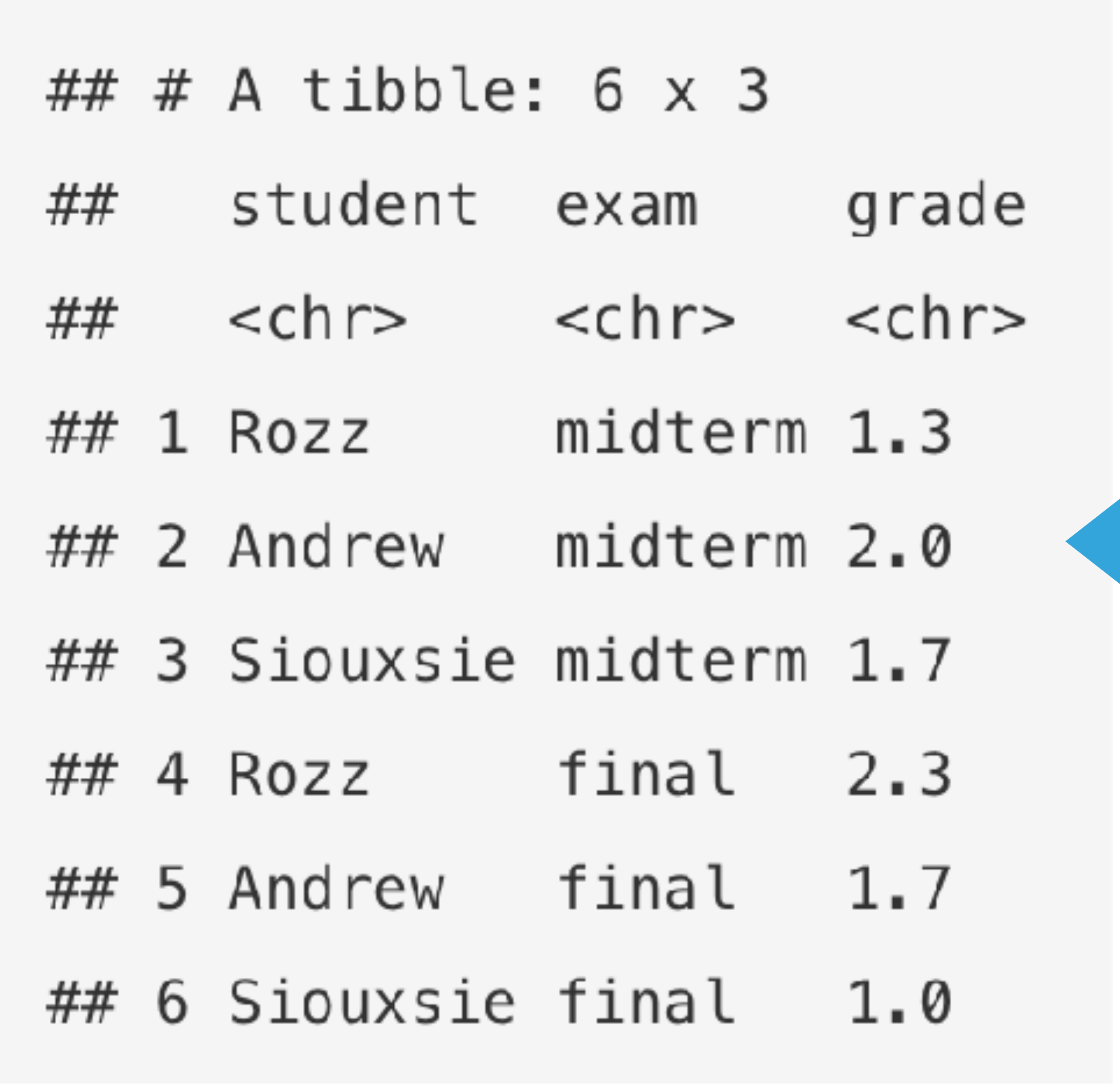
PIVOTING: LONGER



```r
exam_results_visual <- tribble(
  ~exam,      ~"Rozz",  ~"Andrew",  ~"Siouxsie",
  "midterm",  "1.3",    "2.0",      "1.7",
  "final"  ,  "2.3",    "1.7",      "1.0"
)
```

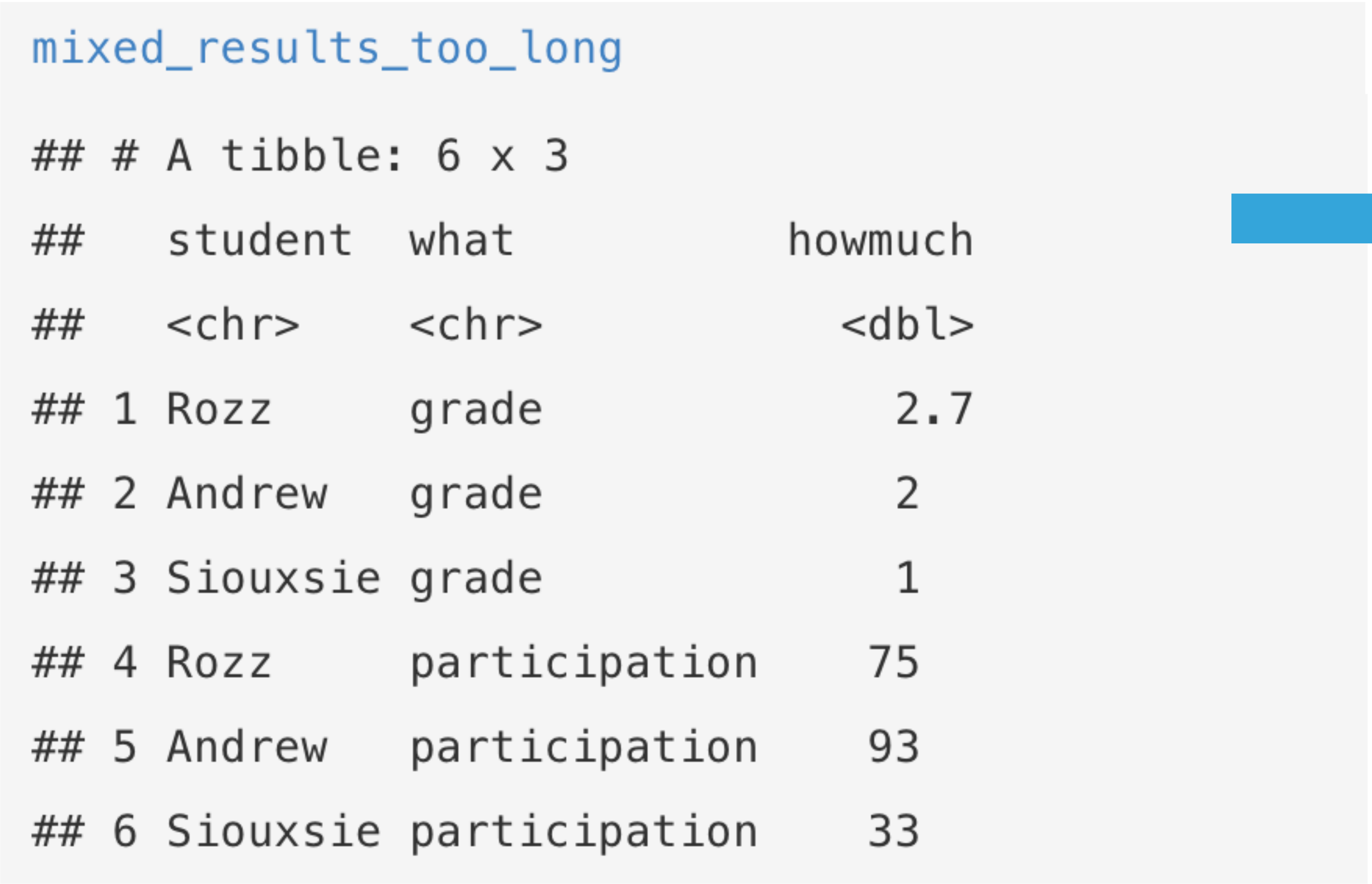


```r
exam_results_visual %>%
  pivot_longer(
    # pivot every column except the first
    cols = - 1,
    # name of new column which contains the
    # names of the columns to be "gathered"
    names_to = "student",
    # name of new column which contains the values
    # of the cells which now form a new column
    values_to = "grade"
  ) %>%
  # optional reordering of columns (to make
  # the output exactly like `exam_results_tidy`)
  select(student, exam, grade)
```



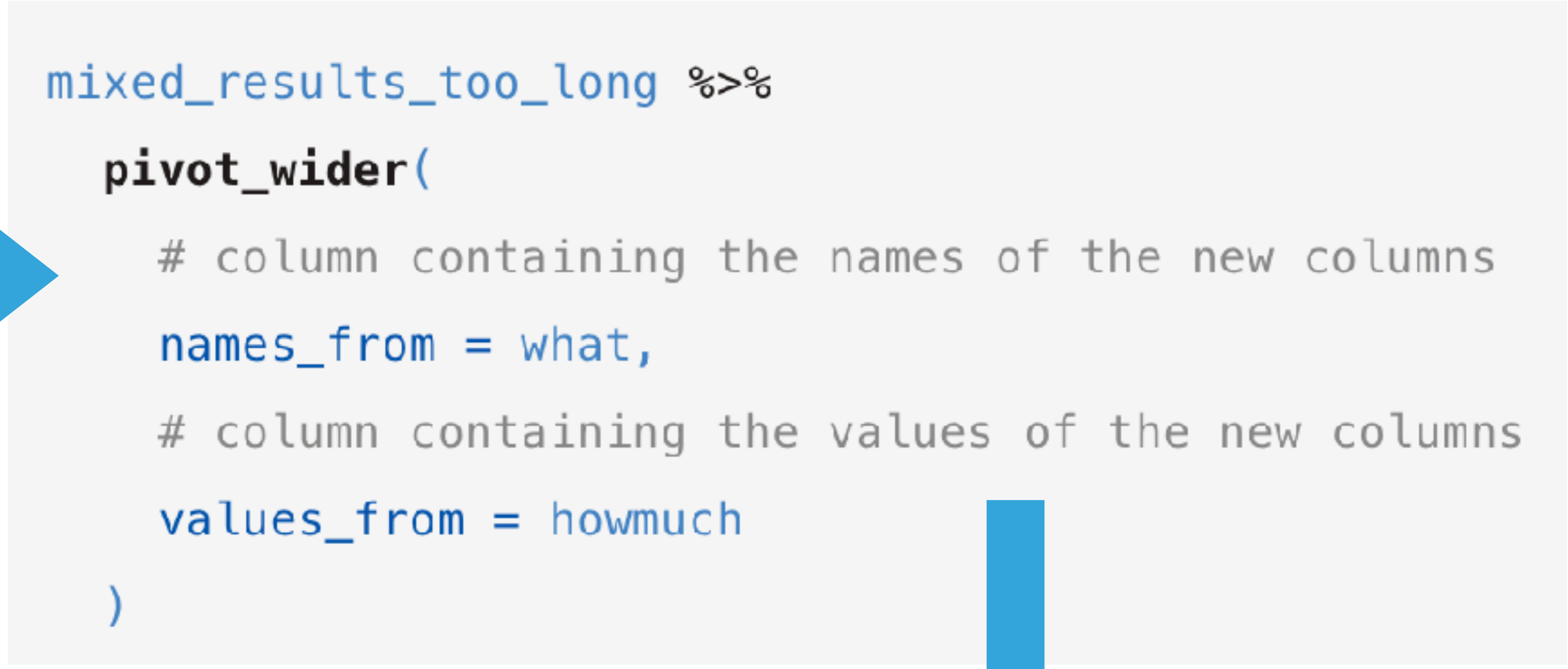
```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <chr>
## 1 Rozz     midterm 1.3
## 2 Andrew   midterm 2.0
## 3 Siouxsie midterm 1.7
## 4 Rozz     final   2.3
## 5 Andrew   final   1.7
## 6 Siouxsie final   1.0
```

PIVOTING: WIDER

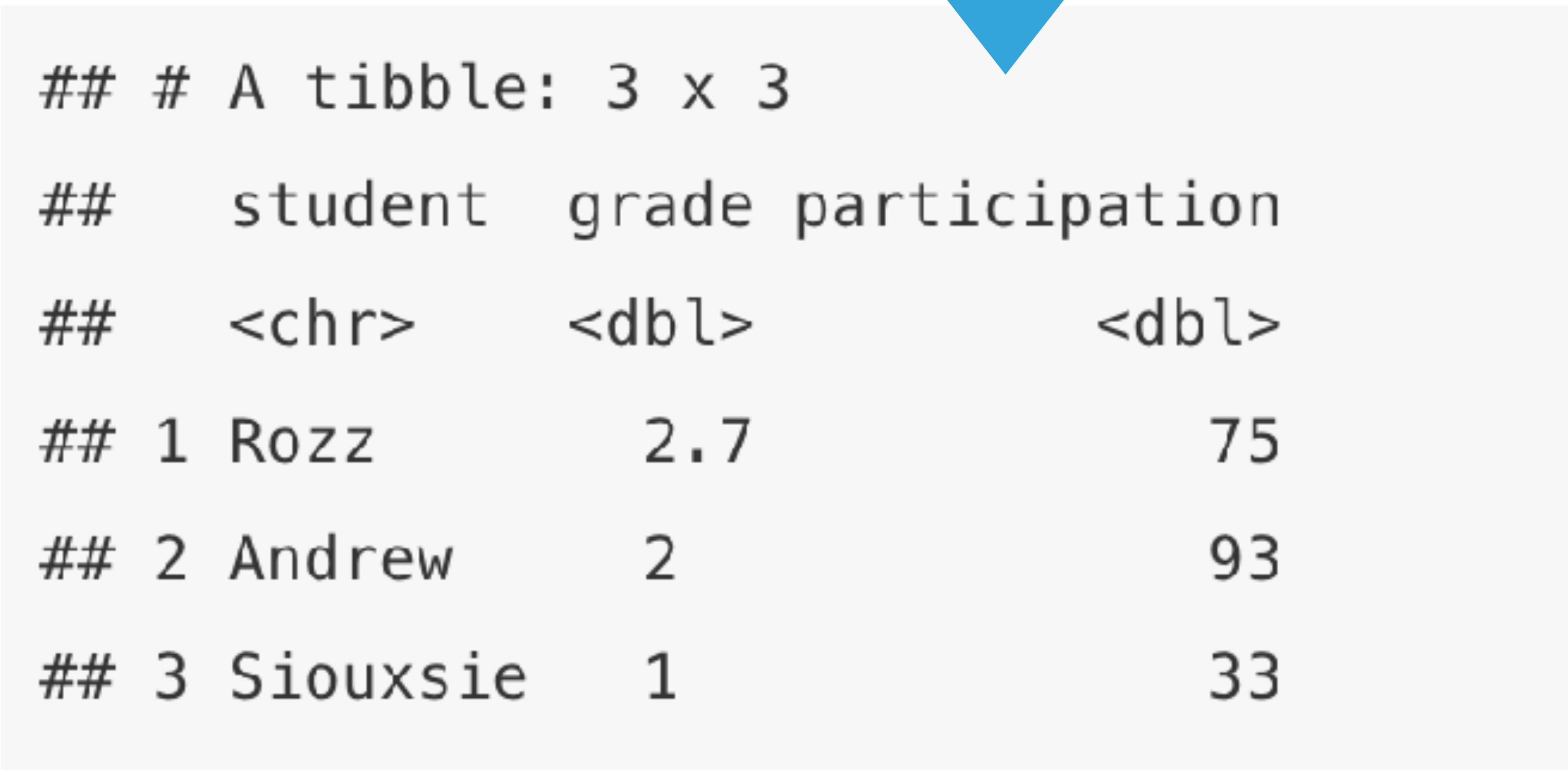


```
mixed_results_too_long

## # A tibble: 6 x 3
##   student  what          howmuch
##   <chr>    <chr>           <dbl>
## 1 Rozz     grade             2.7
## 2 Andrew   grade             2  
## 3 Siouxsie grade             1  
## 4 Rozz     participation    75  
## 5 Andrew   participation    93  
## 6 Siouxsie participation    33  
```

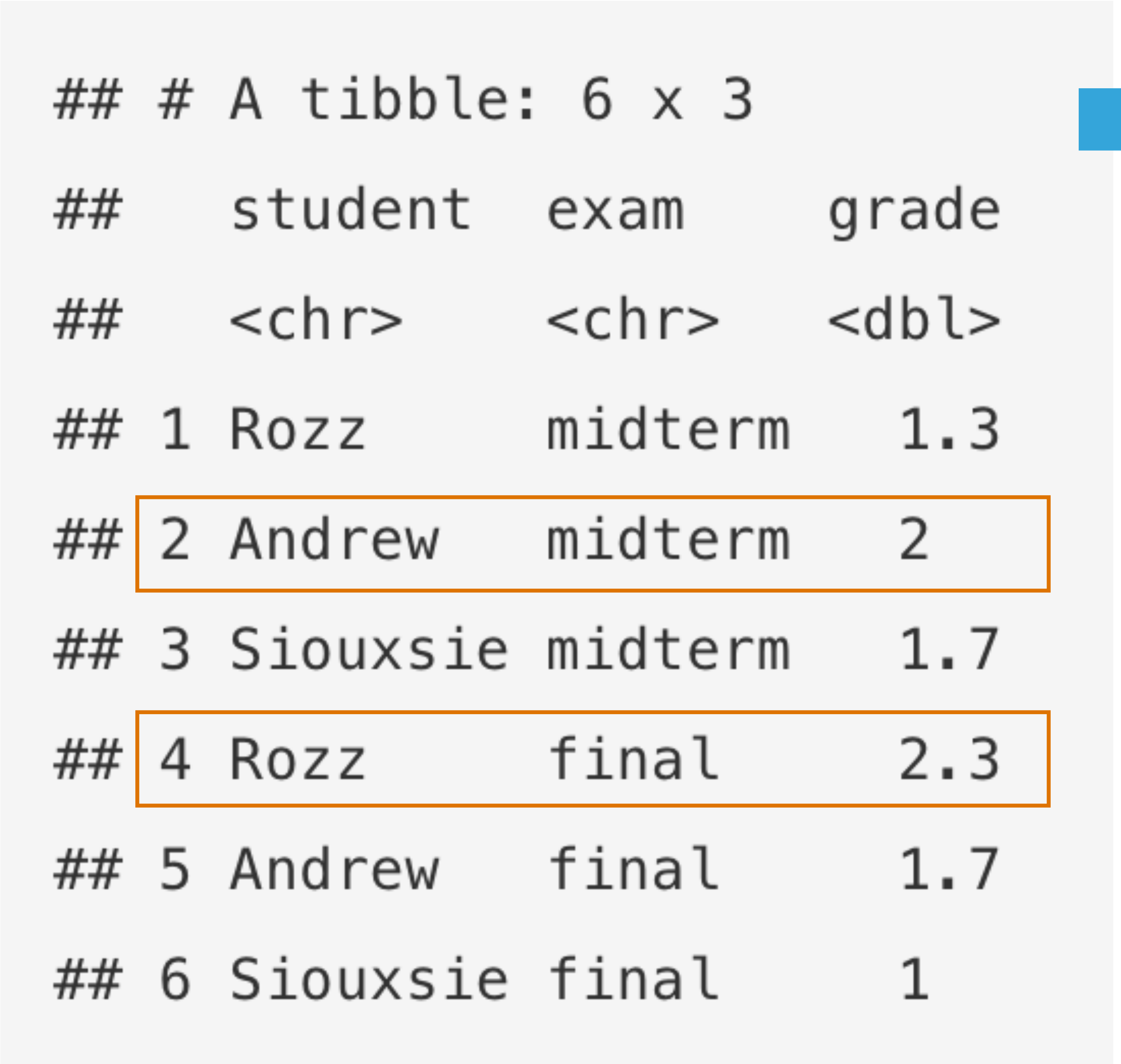


```
mixed_results_too_long %>%
  pivot_wider(
    # column containing the names of the new columns
    names_from = what,
    # column containing the values of the new columns
    values_from = howmuch
  )
```

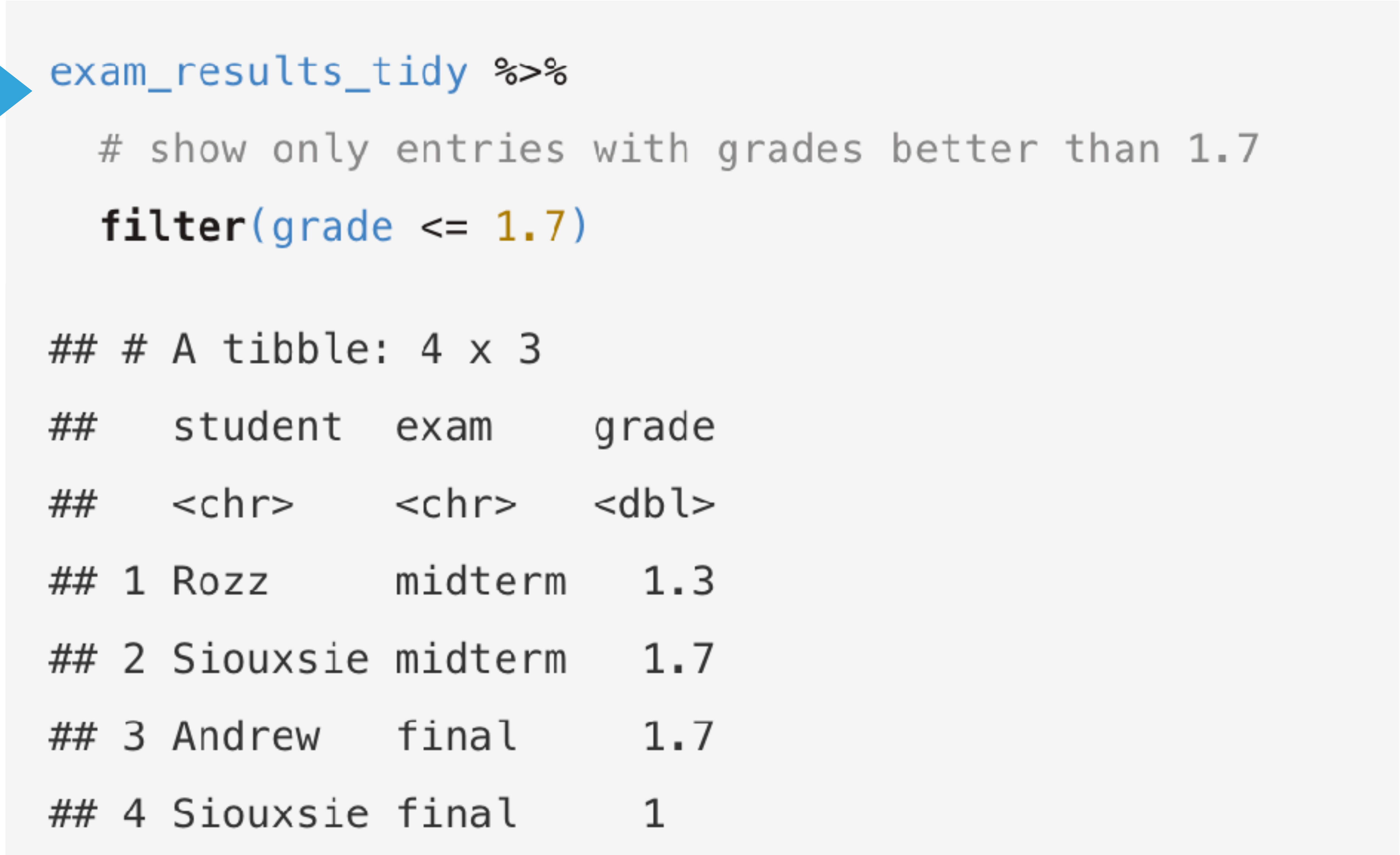


```
## # A tibble: 3 x 3
##   student  grade participation
##   <chr>    <dbl>         <dbl>
## 1 Rozz       2.7            75
## 2 Andrew     2              93
## 3 Siouxsie   1              33
```

FILTERING ROWS



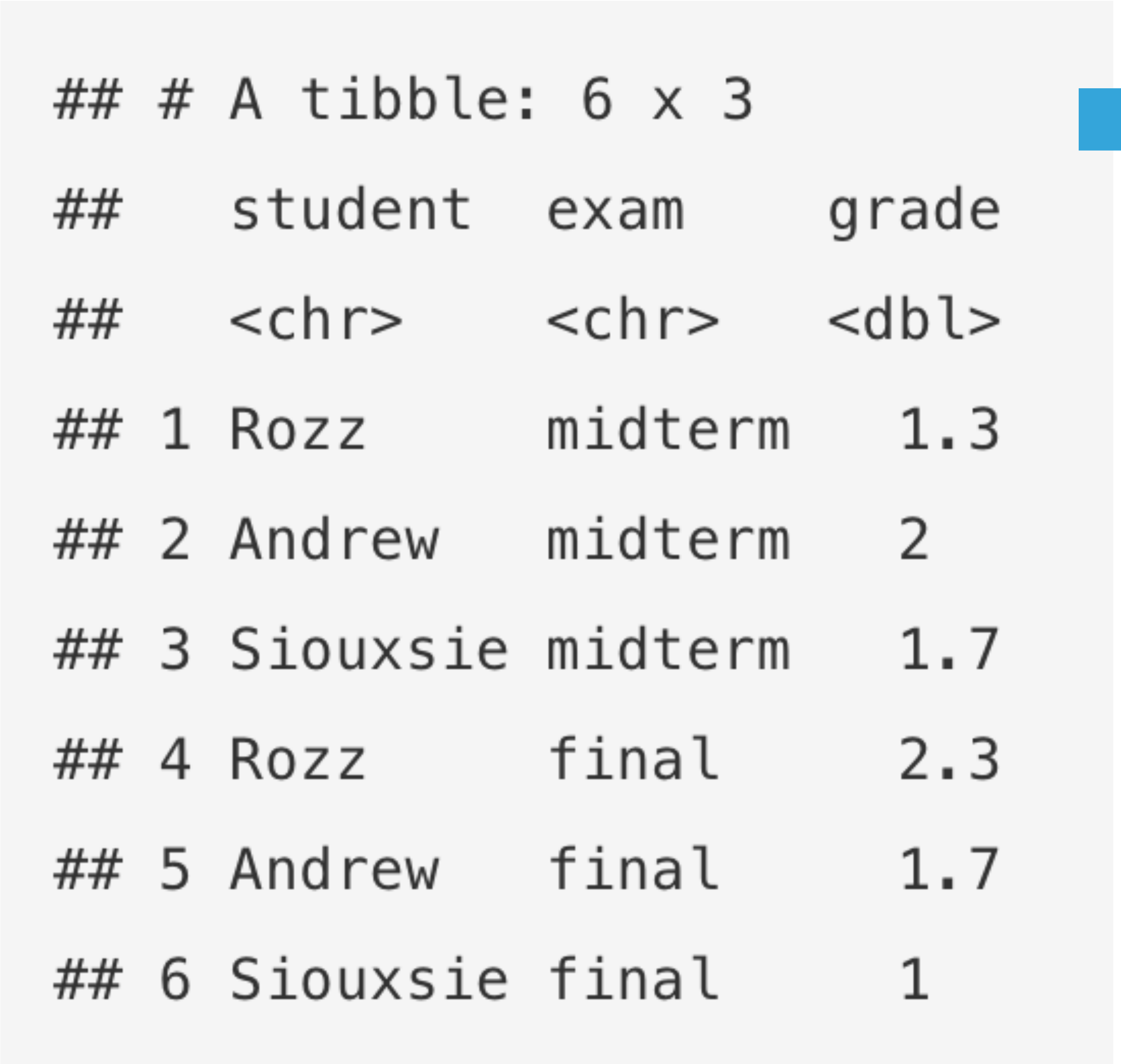
```
## # A tibble: 6 x 3
##   student exam    grade
##   <chr>   <chr>   <dbl>
## 1 Rozz    midterm   1.3
## 2 Andrew  midterm   2
## 3 Siouxsie midterm  1.7
## 4 Rozz    final     2.3
## 5 Andrew  final     1.7
## 6 Siouxsie final    1
```



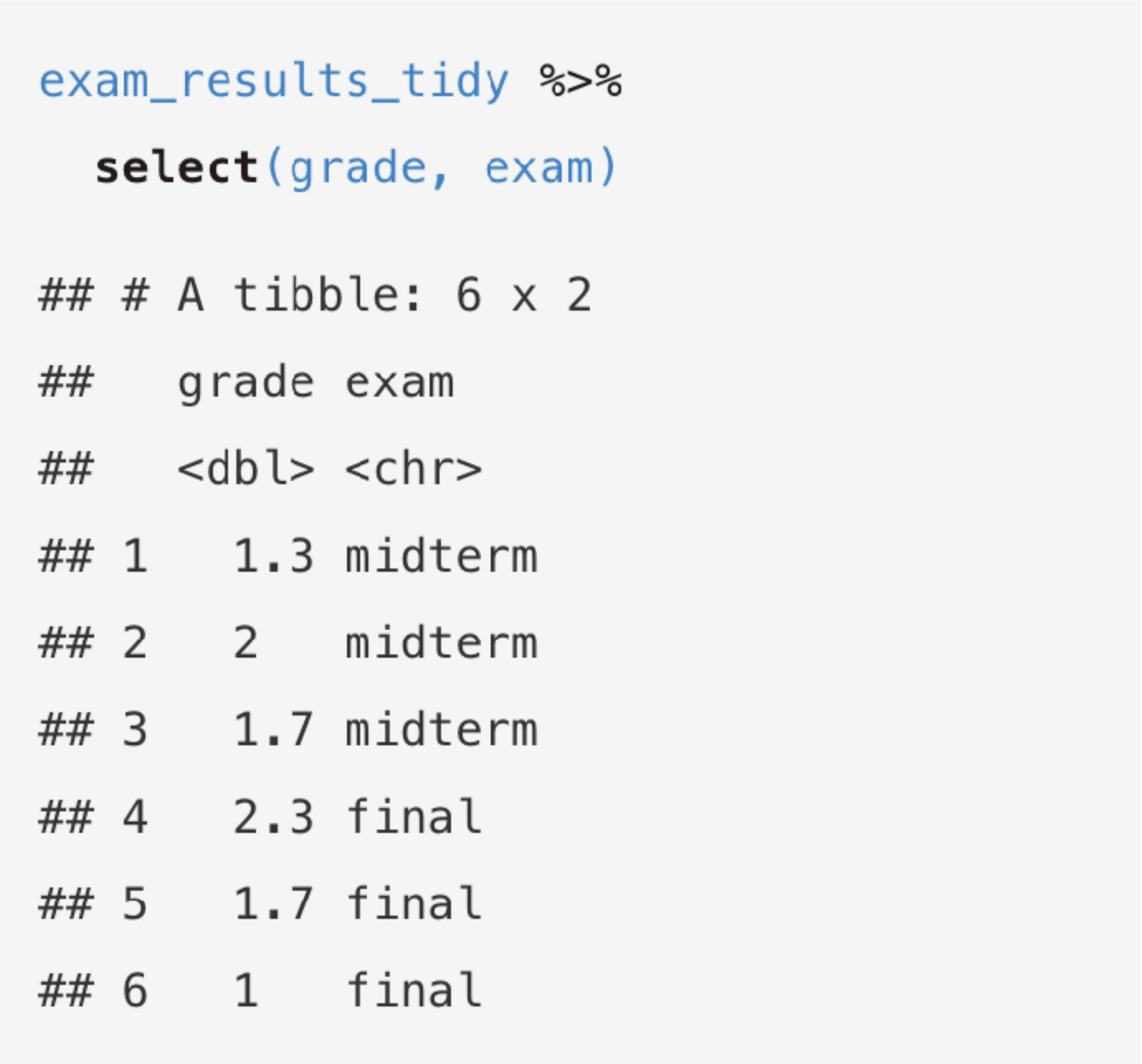
```
exam_results_tidy %>%
  # show only entries with grades better than 1.7
  filter(grade <= 1.7)

## # A tibble: 4 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Siouxsie midterm   1.7
## 3 Andrew   final     1.7
## 4 Siouxsie final     1
```

SELECTING COLUMNS



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2  
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1  
```

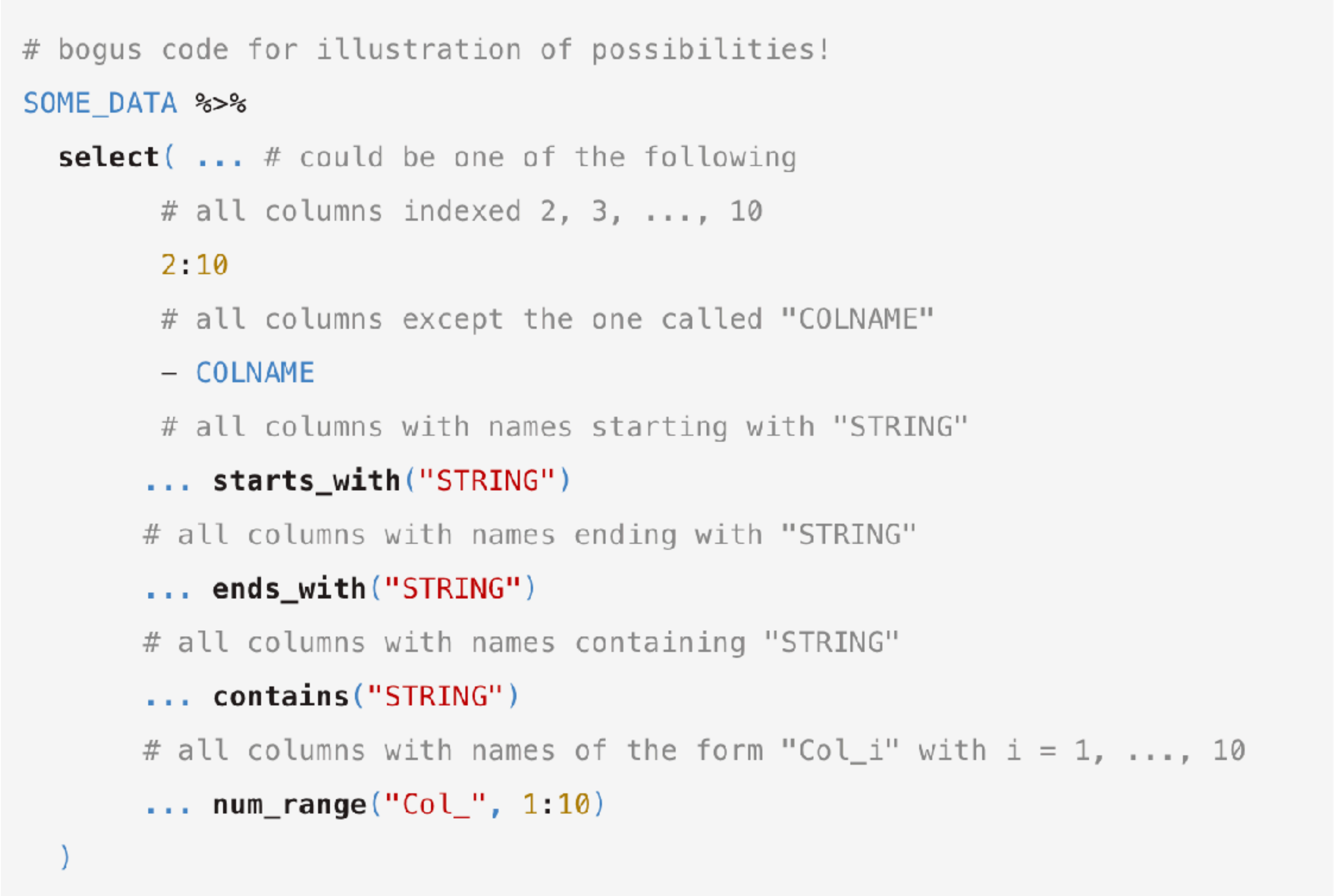


```
exam_results_tidy %>%
  select(grade, exam)

## # A tibble: 6 x 2
##   grade exam   
##   <dbl> <chr>  
## 1   1.3 midterm
## 2   2   midterm
## 3   1.7 midterm
## 4   2.3 final  
## 5   1.7 final  
## 6   1   final  
```

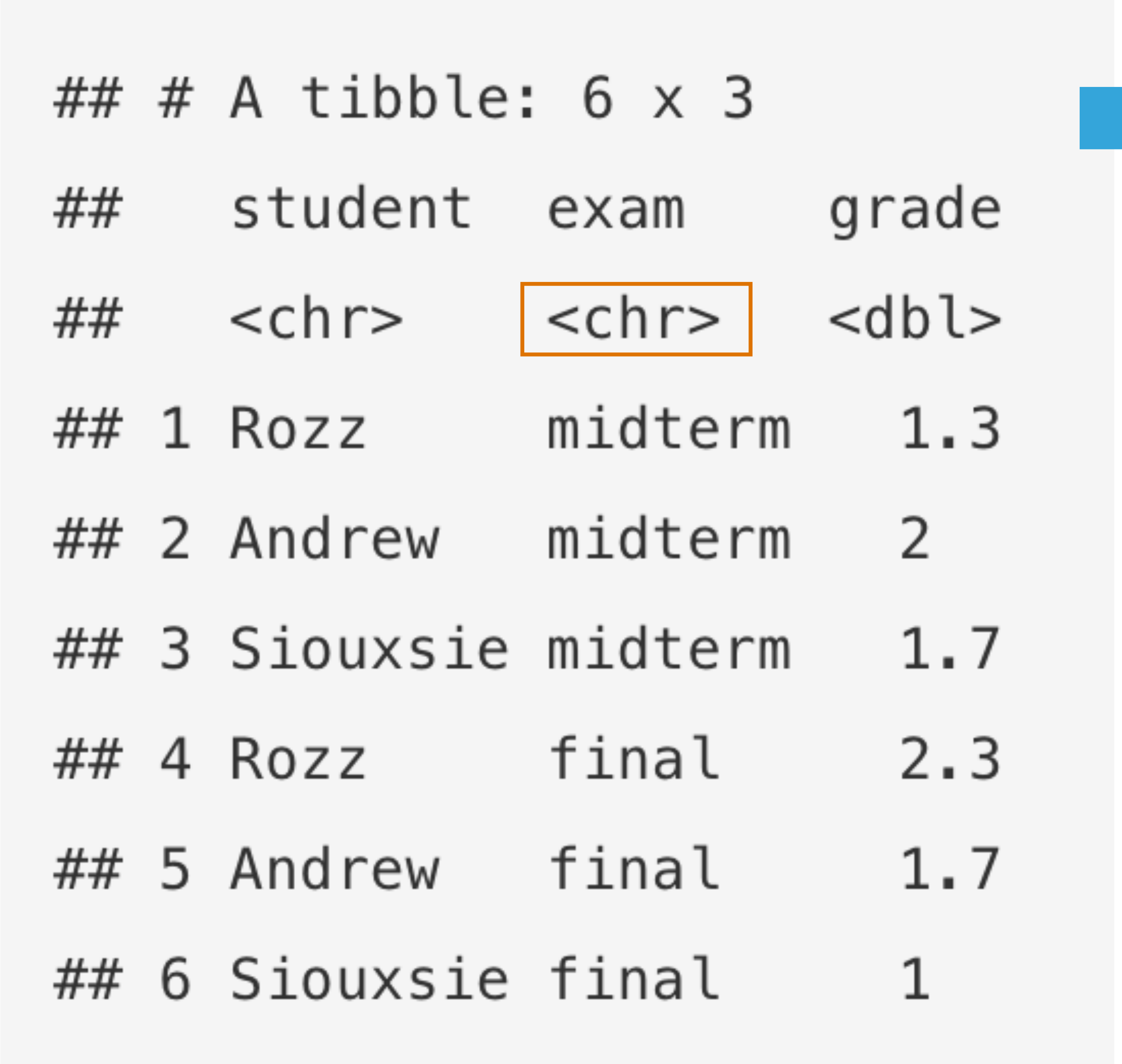
TIDY SPECIFICATION OF COLUMNS TO SELECT

- from tidyselect package

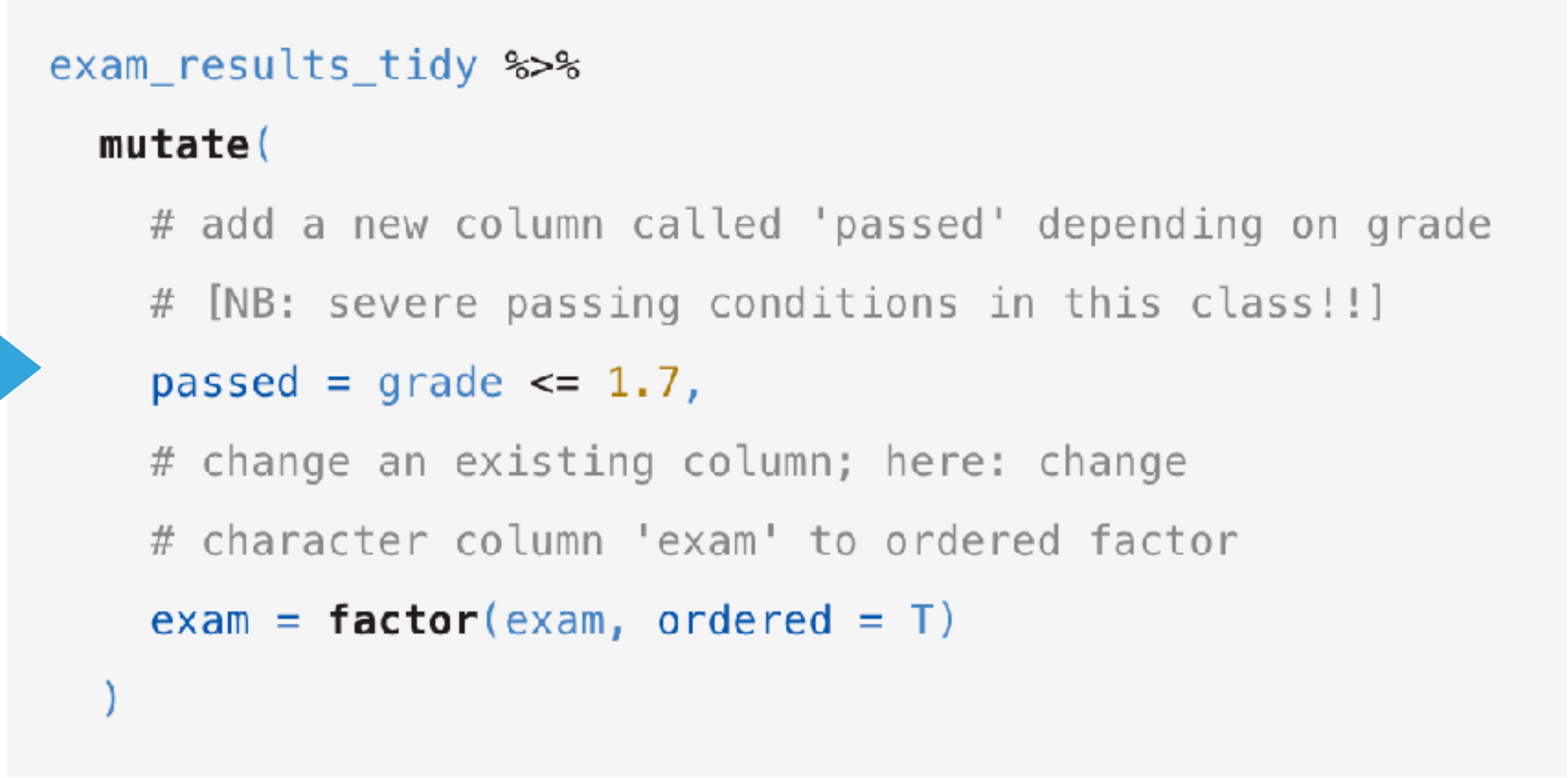


```
# bogus code for illustration of possibilities!
SOME_DATA %>%
  select( ... # could be one of the following
        # all columns indexed 2, 3, ..., 10
        2:10
        # all columns except the one called "COLNAME"
        - COLNAME
        # all columns with names starting with "STRING"
       ... starts_with("STRING")
       # all columns with names ending with "STRING"
       ... ends_with("STRING")
       # all columns with names containing "STRING"
       ... contains("STRING")
       # all columns with names of the form "Col_i" with i = 1, ..., 10
       ... num_range("Col_", 1:10)
  )
```

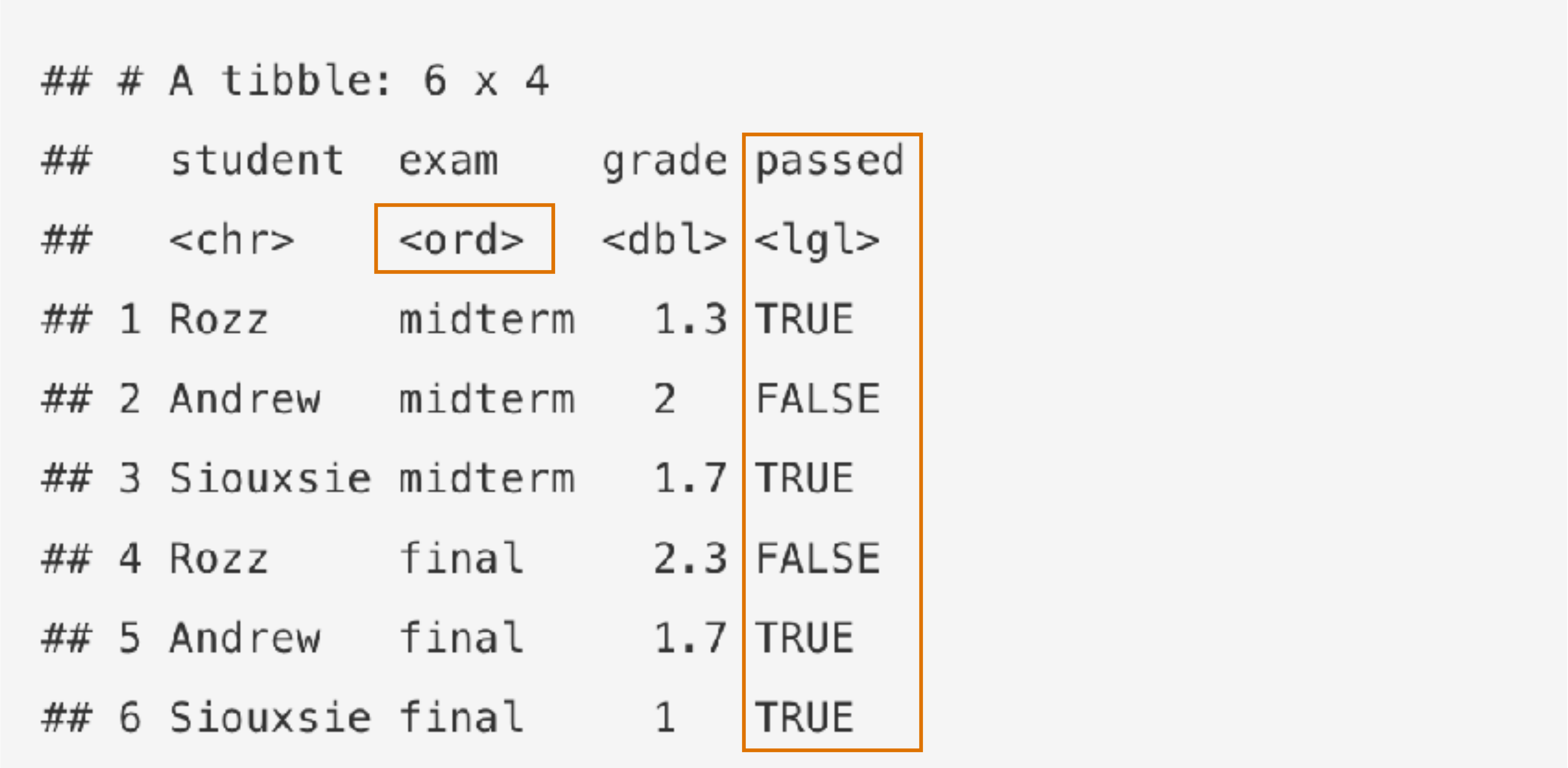
ADDING OR CHANGING COLUMNS



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

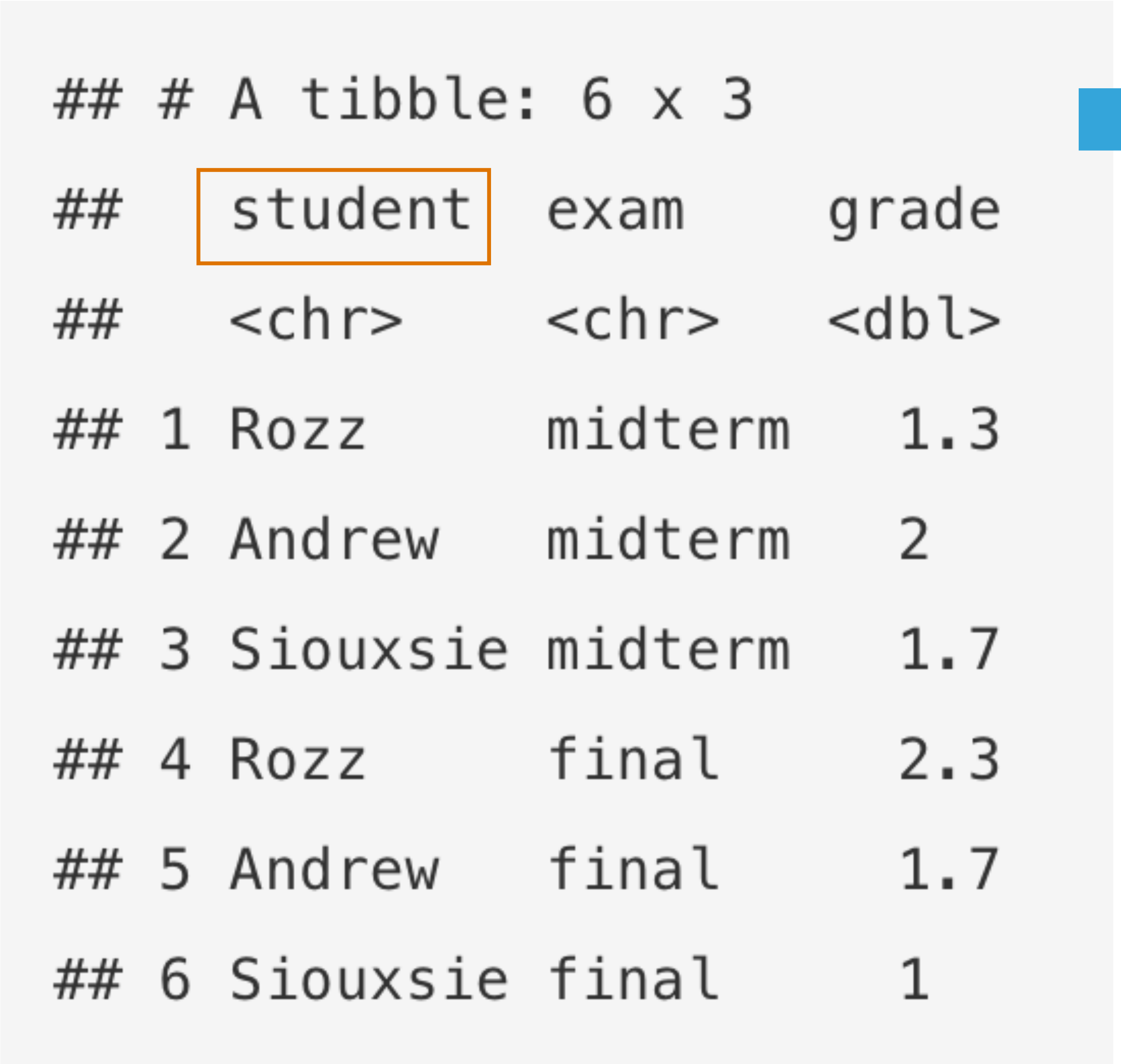


```
exam_results_tidy %>%
  mutate(
    # add a new column called 'passed' depending on grade
    # [NB: severe passing conditions in this class!!]
    passed = grade <= 1.7,
    # change an existing column; here: change
    # character column 'exam' to ordered factor
    exam = factor(exam, ordered = T)
  )
```

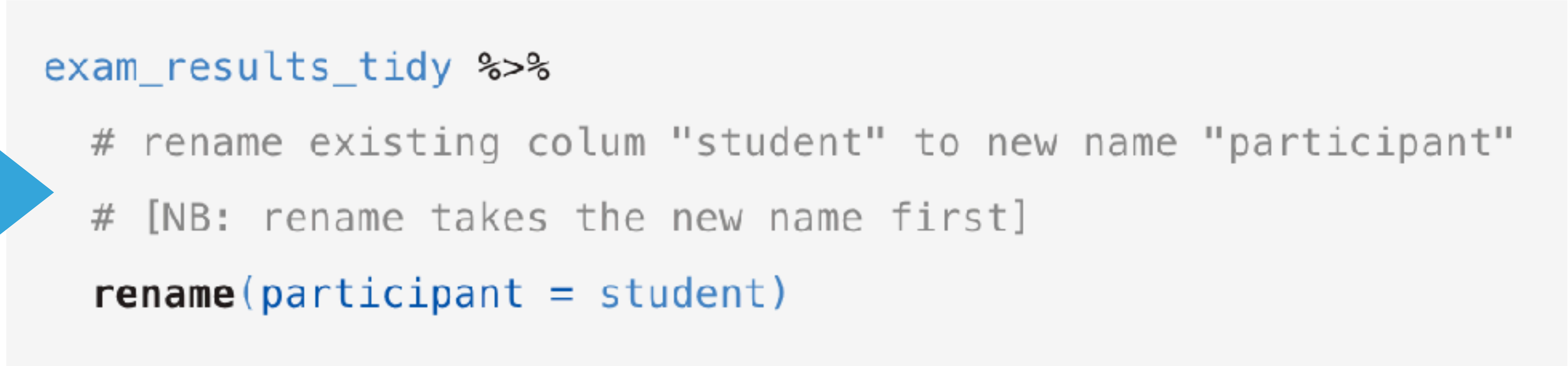


```
## # A tibble: 6 x 4
##   student  exam    grade passed
##   <chr>    <ord>   <dbl> <lgl>
## 1 Rozz     midterm   1.3 TRUE
## 2 Andrew   midterm   2   FALSE
## 3 Siouxsie midterm   1.7 TRUE
## 4 Rozz     final     2.3 FALSE
## 5 Andrew   final     1.7 TRUE
## 6 Siouxsie final     1   TRUE
```

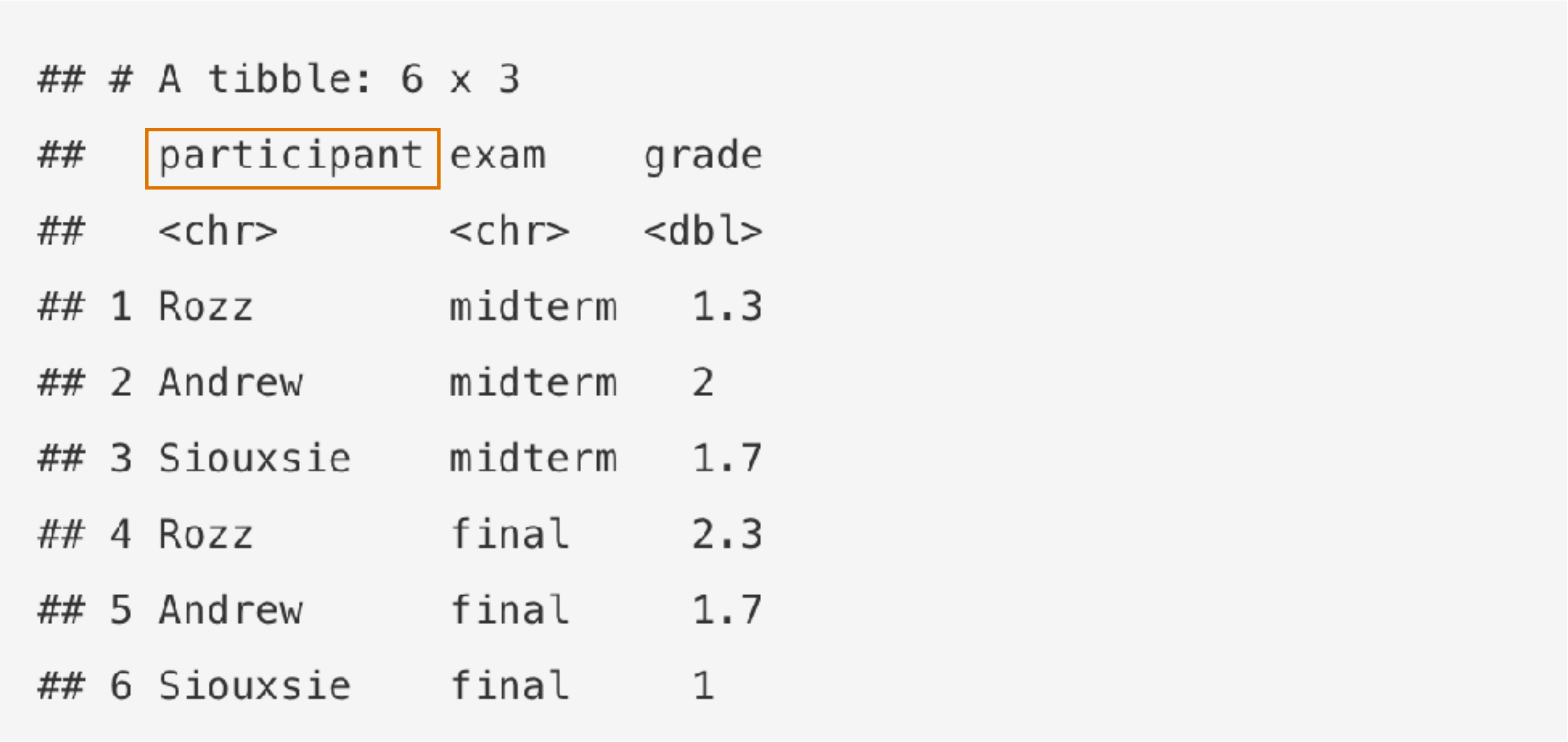
RENAMING COLUMNS



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

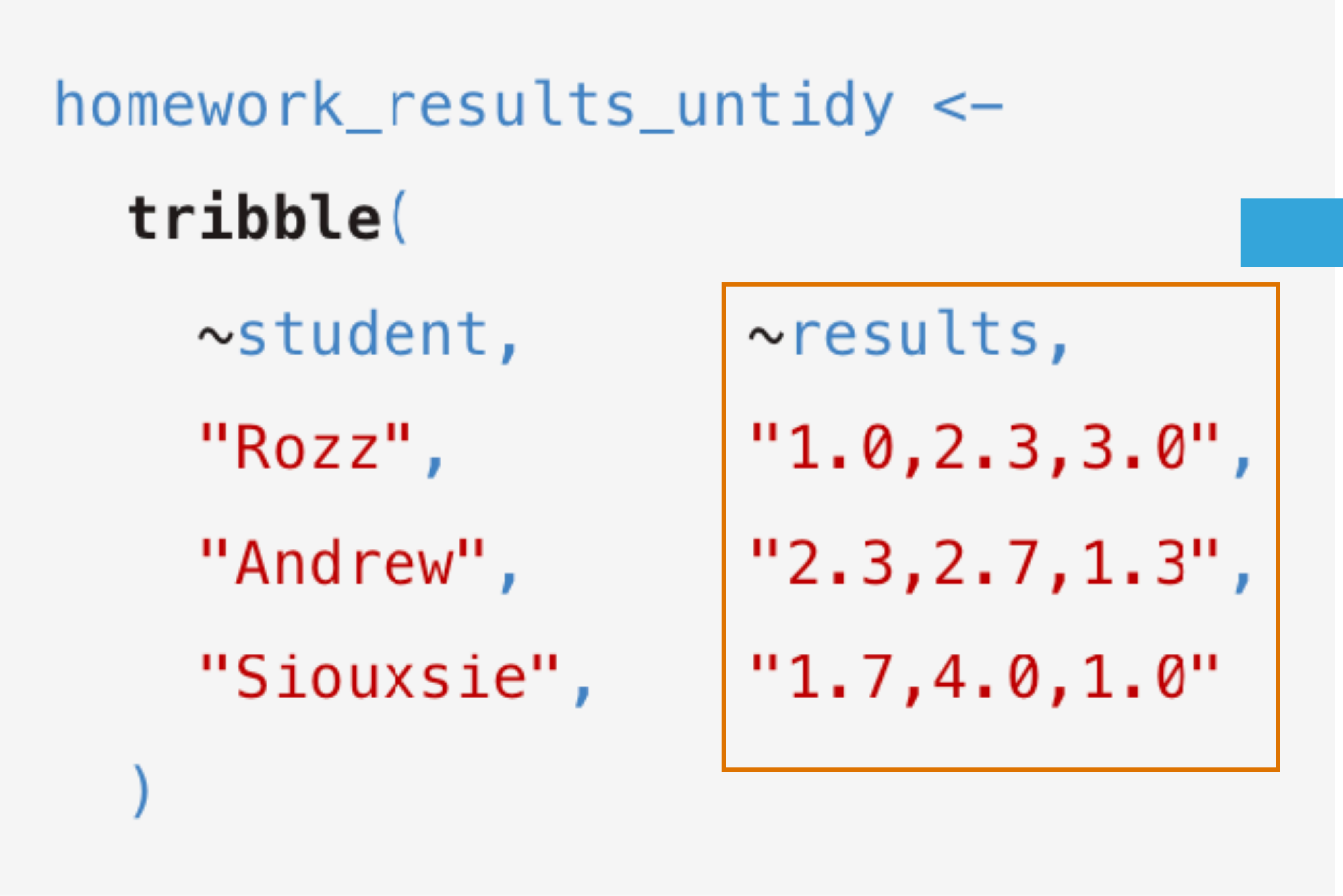


```
exam_results_tidy %>%
  # rename existing colum "student" to new name "participant"
  # [NB: rename takes the new name first]
  rename(participant = student)
```

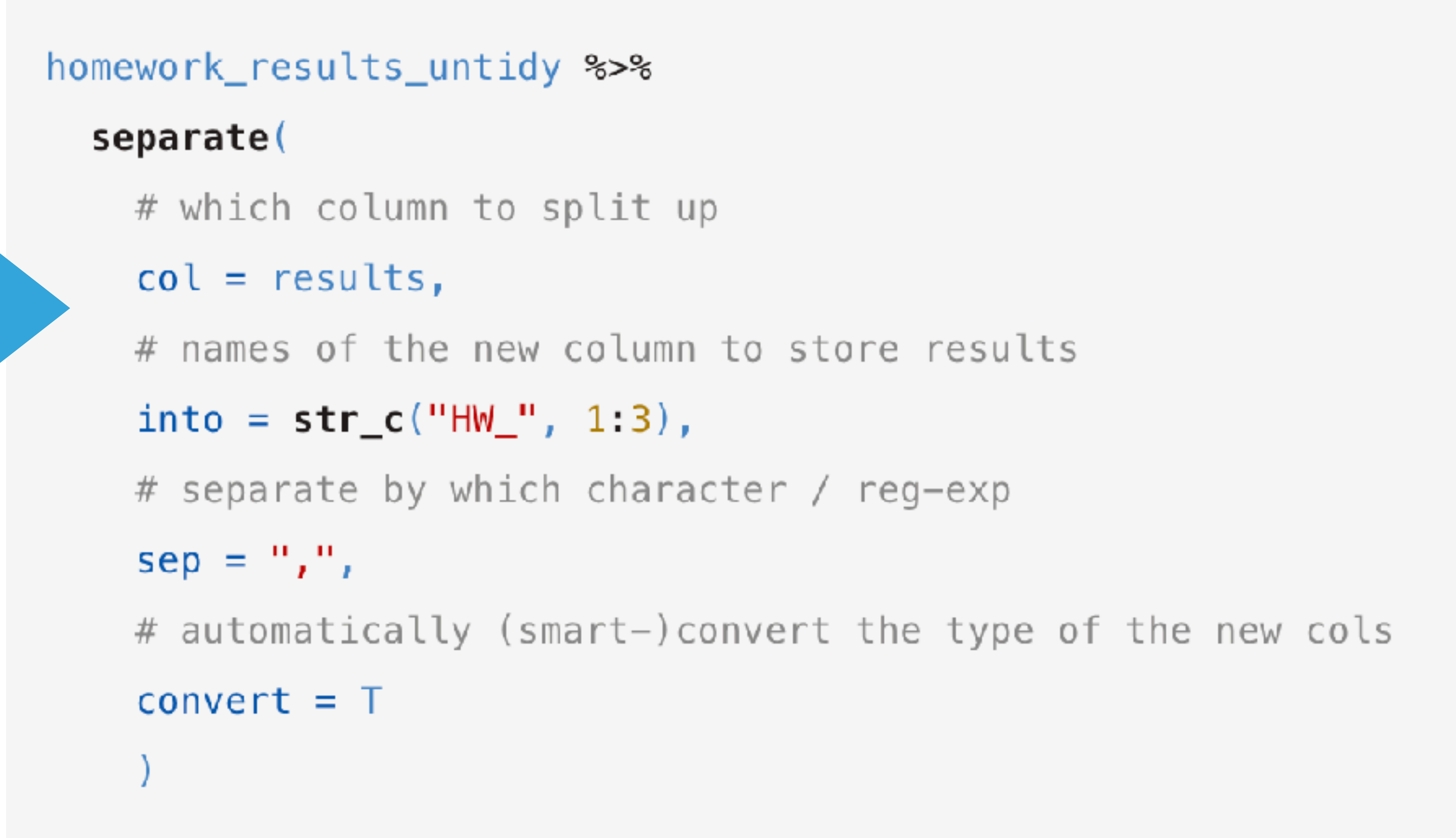


```
## # A tibble: 6 x 3
##   participant exam    grade
##   <chr>       <chr>   <dbl>
## 1 Rozz        midterm   1.3
## 2 Andrew      midterm   2
## 3 Siouxsie    midterm   1.7
## 4 Rozz        final     2.3
## 5 Andrew      final     1.7
## 6 Siouxsie    final     1
```

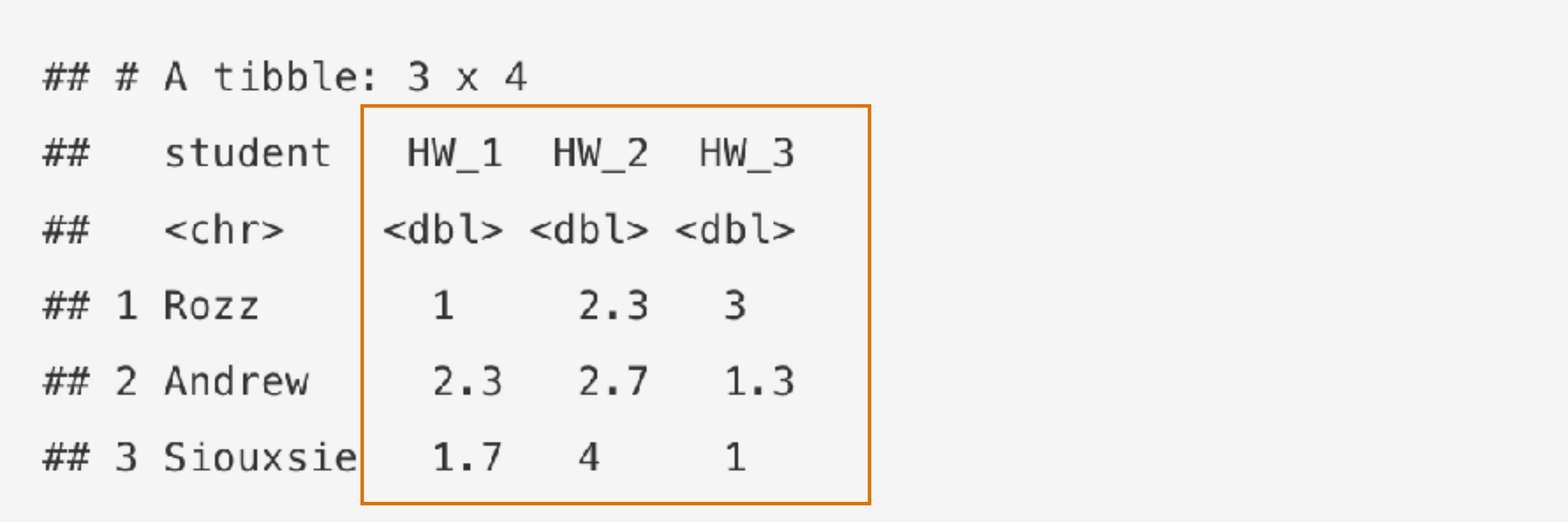
SPLITTING COLUMNS



```r
homework_results_untidy <-
  tribble(
    ~student,      ~results,
    "Rozz",        "1.0,2.3,3.0",
    "Andrew",      "2.3,2.7,1.3",
    "Siouxsie",    "1.7,4.0,1.0"
  )
```

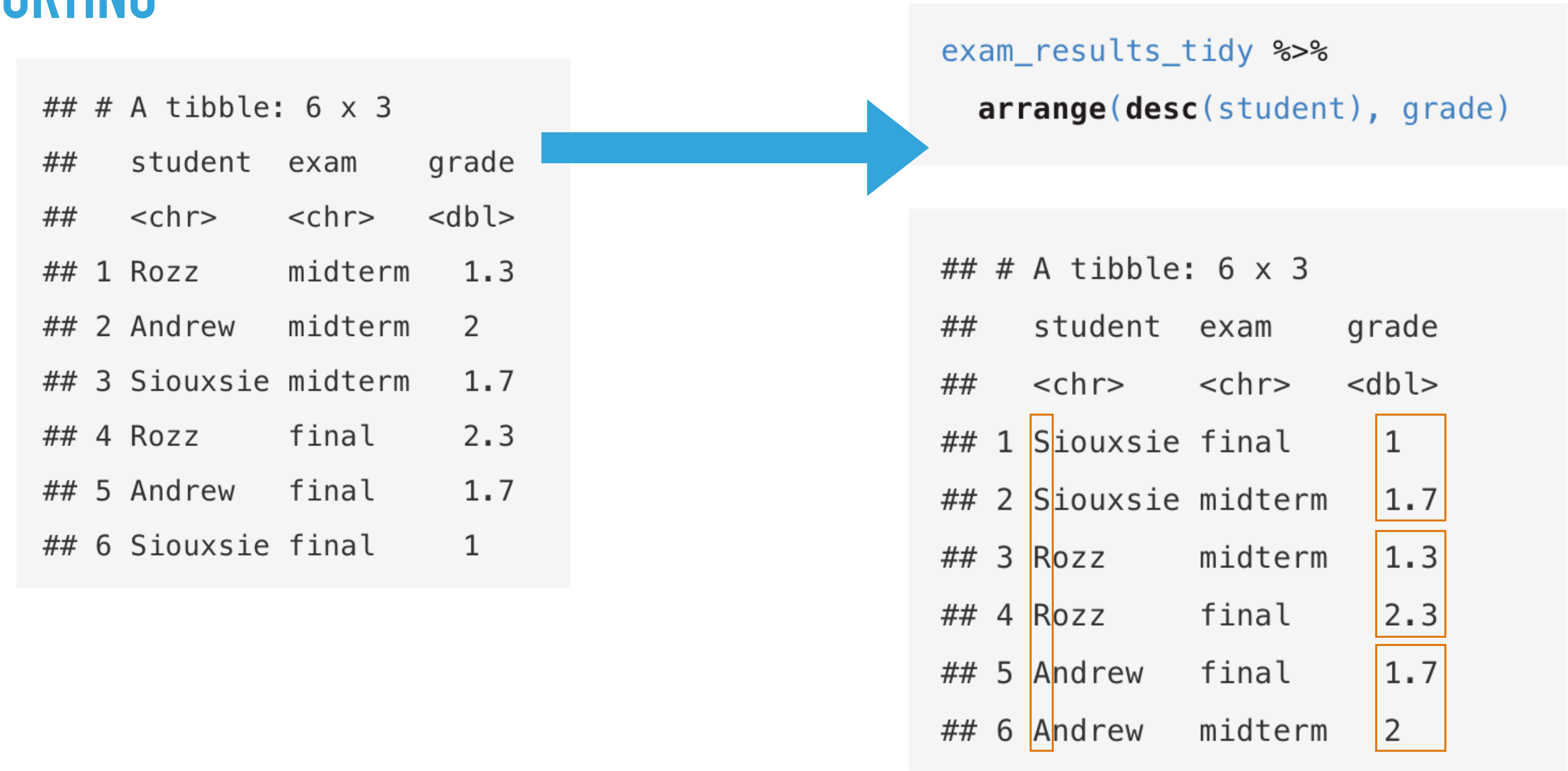


```r
homework_results_untidy %>%
  separate(
    # which column to split up
    col = results,
    # names of the new column to store results
    into = str_c("HW_", 1:3),
    # separate by which character / reg-exp
    sep = ",",
    # automatically (smart-)convert the type of the new cols
    convert = T
    )
```



```
## # A tibble: 3 x 4
##   student   HW_1  HW_2  HW_3
##   <chr>    <dbl> <dbl> <dbl>
## 1 Rozz       1     2.3   3
## 2 Andrew     2.3   2.7   1.3
## 3 Siouxsie   1.7   4     1
```

SORTING

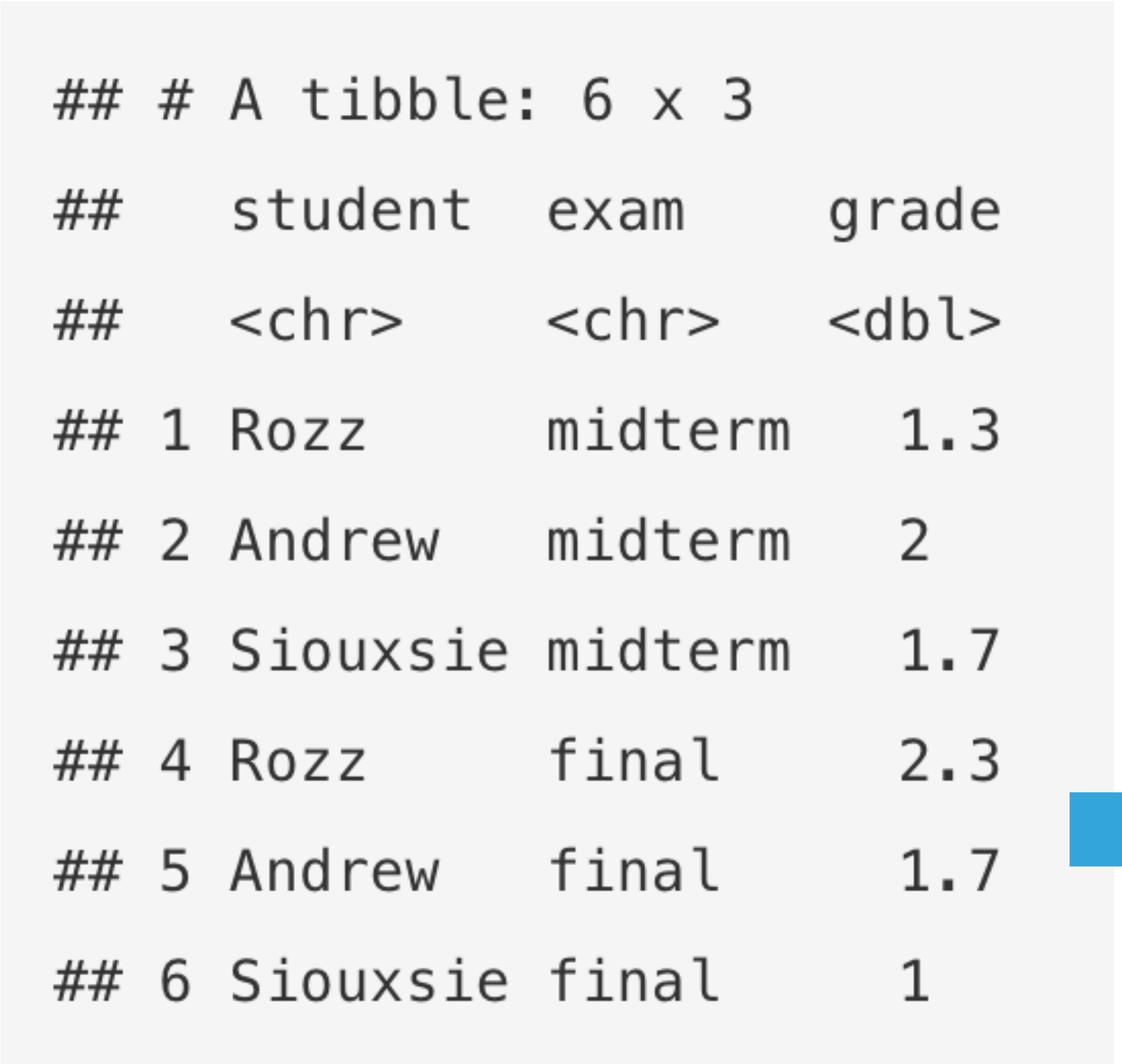


```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

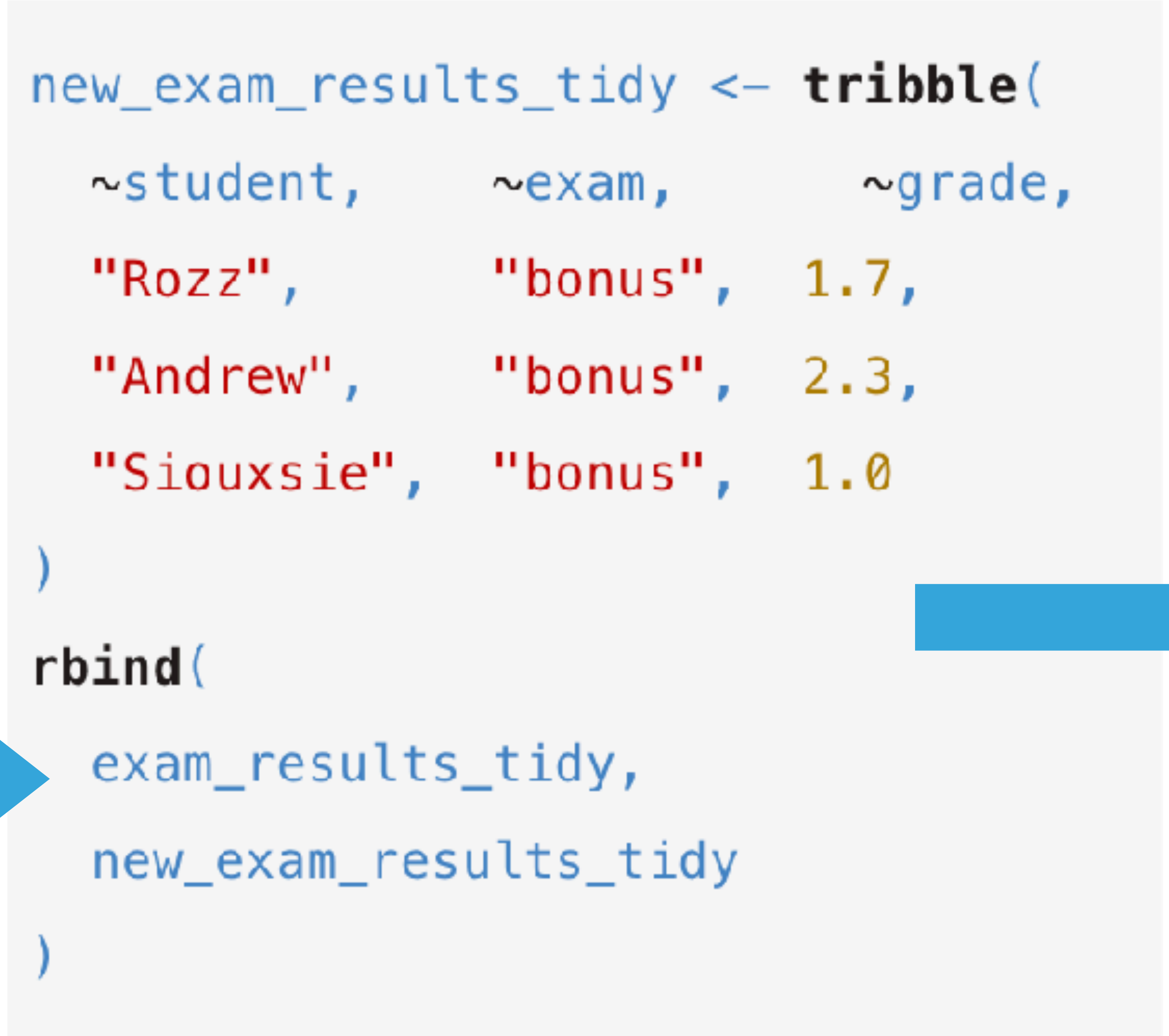
```
exam_results_tidy %>%
  arrange(desc(student), grade)
```

```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Siouxsie final     1
## 2 Siouxsie midterm   1.7
## 3 Rozz     midterm   1.3
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Andrew   midterm   2
```

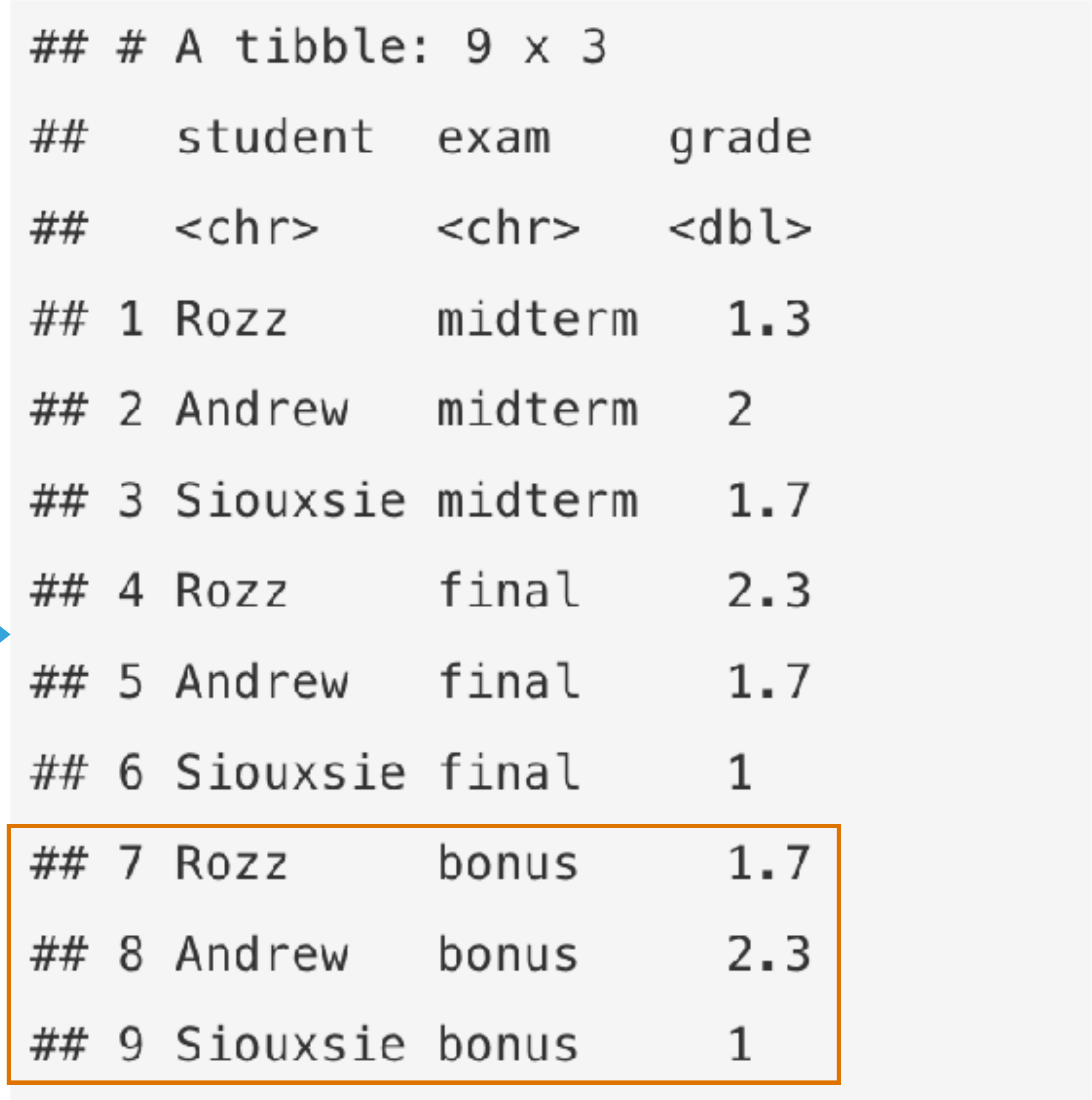
COMBINING DATA



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

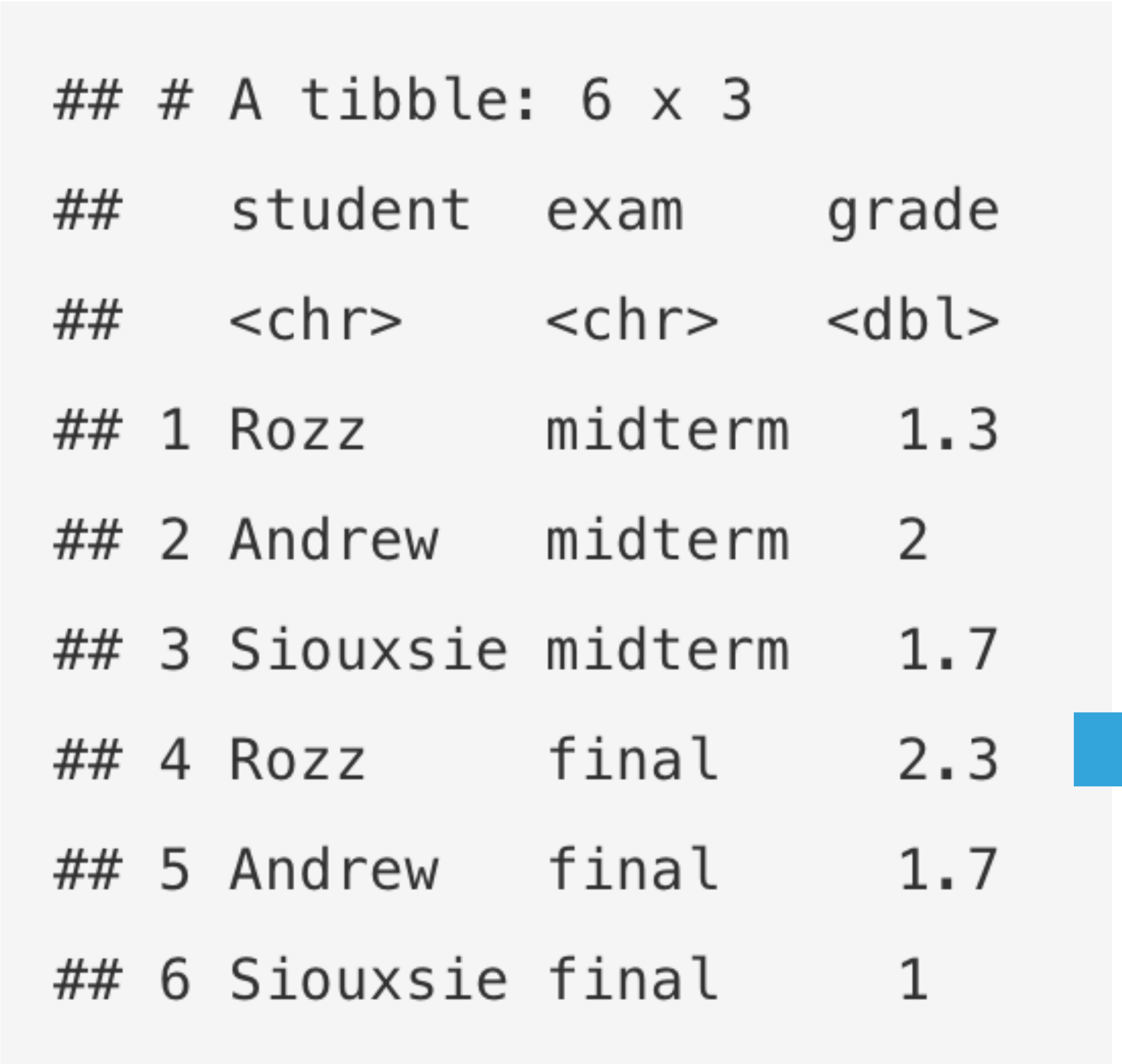


```
new_exam_results_tidy <- tribble(
  ~student,     ~exam,      ~grade,
  "Rozz",       "bonus",    1.7,
  "Andrew",     "bonus",    2.3,
  "Siouxsie",   "bonus",    1.0
)
rbind(
  exam_results_tidy,
  new_exam_results_tidy
)
```

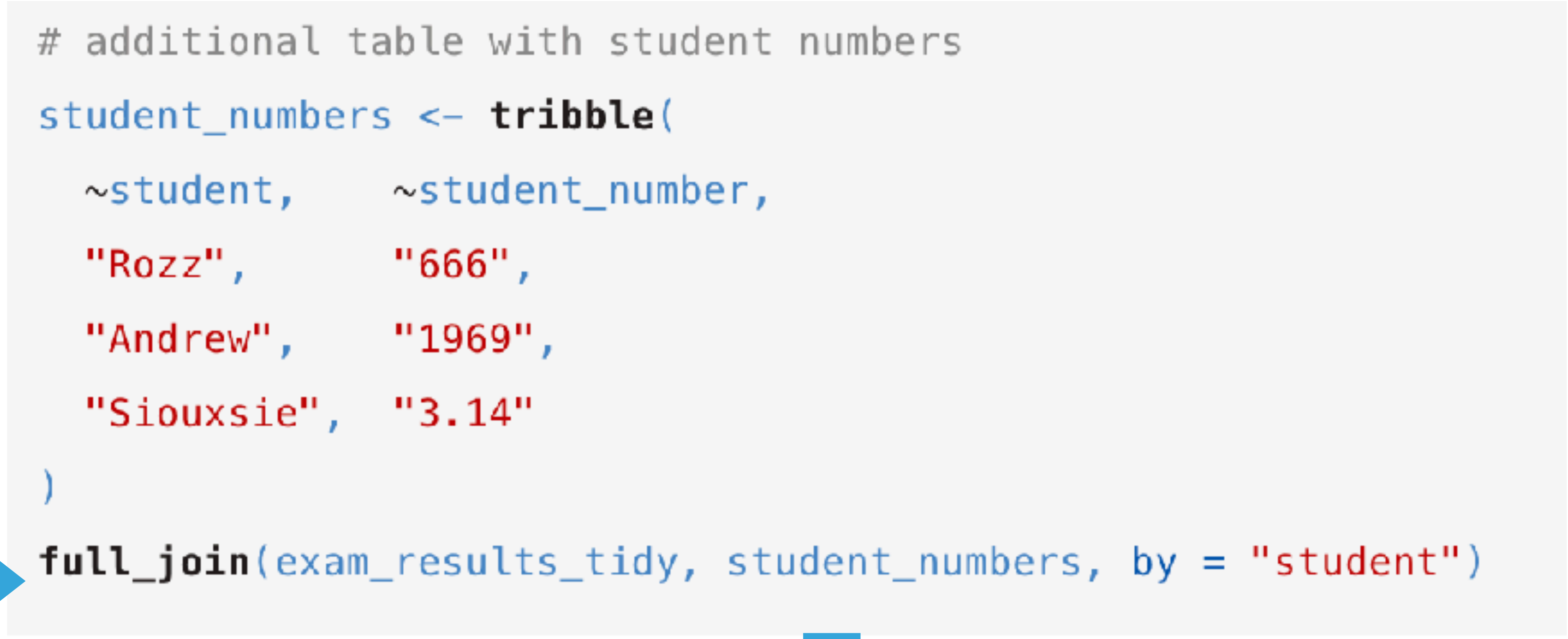


```
## # A tibble: 9 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
## 7 Rozz     bonus     1.7
## 8 Andrew   bonus     2.3
## 9 Siouxsie bonus     1
```

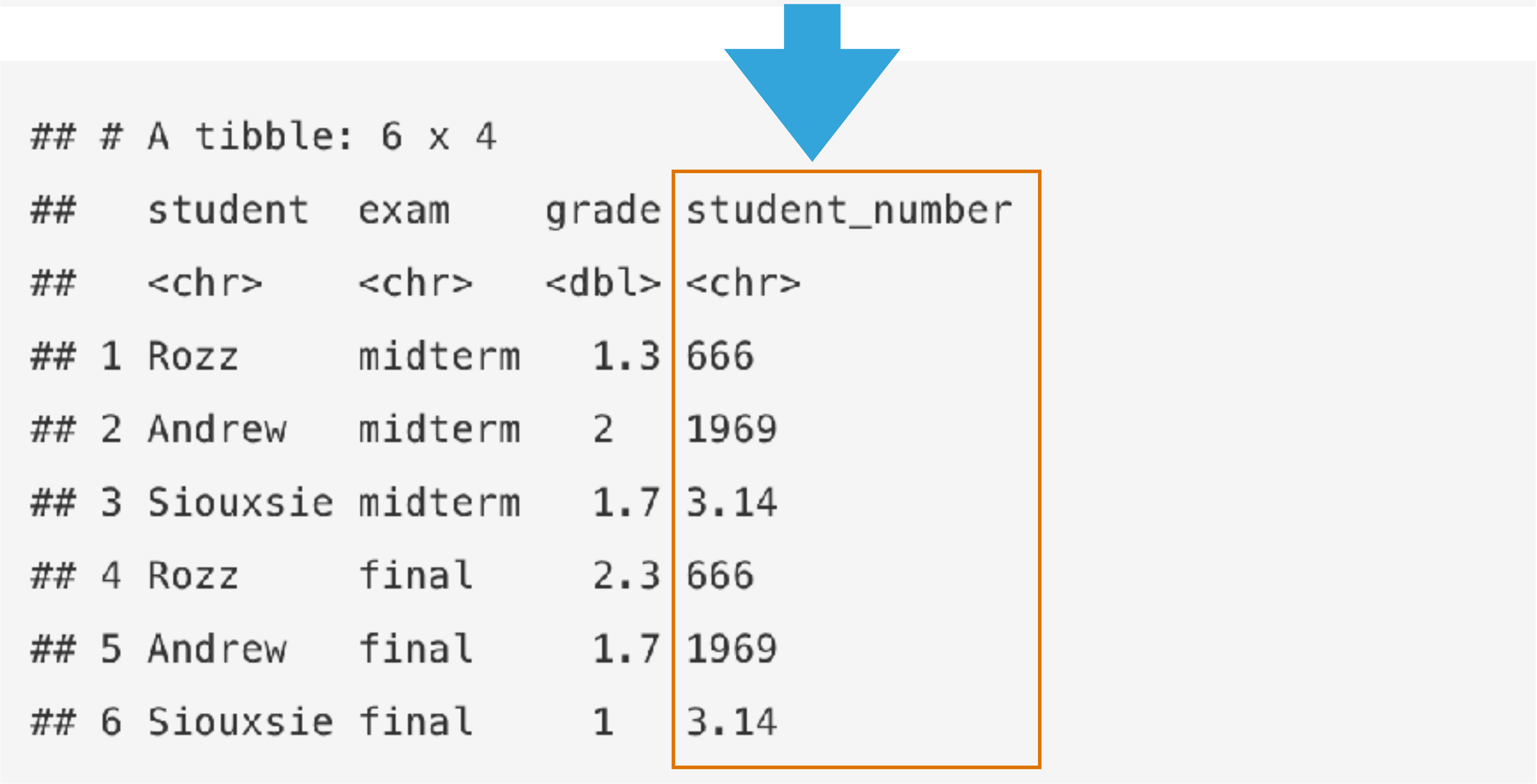
COMBINING DATA



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```



```
# additional table with student numbers
student_numbers <- tribble(
  ~student,    ~student_number,
  "Rozz",      "666",
  "Andrew",    "1969",
  "Siouxsie",  "3.14"
)
full_join(exam_results_tidy, student_numbers, by = "student")
```



```
## # A tibble: 6 x 4
##   student  exam    grade student_number
##   <chr>    <chr>   <dbl> <chr>
## 1 Rozz     midterm   1.3 666
## 2 Andrew   midterm   2   1969
## 3 Siouxsie midterm   1.7 3.14
## 4 Rozz     final     2.3 666
## 5 Andrew   final     1.7 1969
## 6 Siouxsie final     1   3.14
```

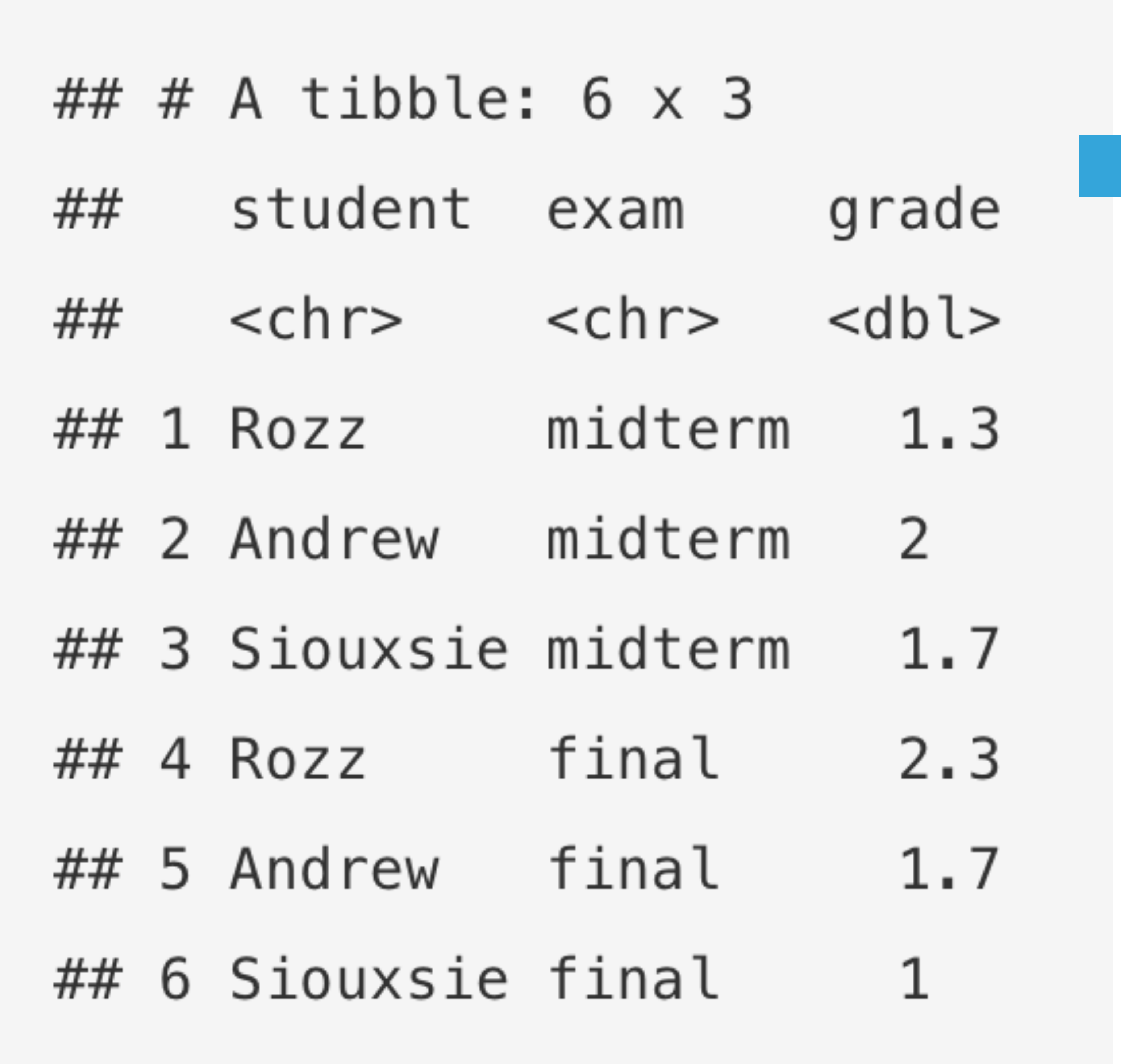
GROUPED OPERATIONS: SUMMARISE

```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2  
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1  
```

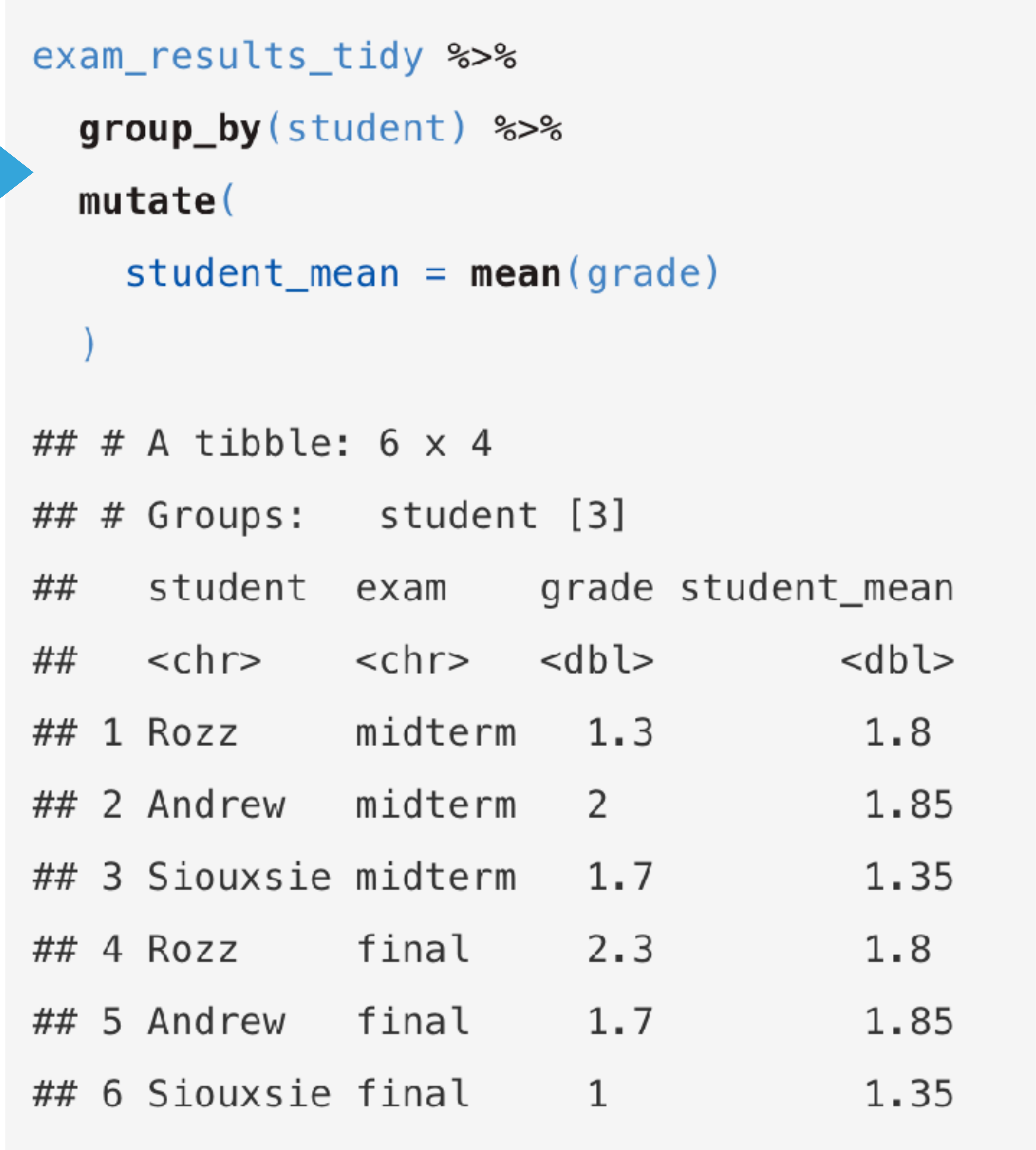
```
exam_results_tidy %>%
  group_by(student) %>%
  summarise(
    student_mean = mean(grade)
  )

## # A tibble: 3 x 2
##   student  student_mean
##   <chr>           <dbl>
## 1 Andrew           1.85
## 2 Rozz             1.8 
## 3 Siouxsie         1.35
```

GROUPED OPERATIONS: MUTATE



```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2  
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1  
```



```
exam_results_tidy %>%
  group_by(student) %>%
  mutate(
    student_mean = mean(grade)
  )

## # A tibble: 6 x 4
## # Groups:   student [3]
##   student  exam    grade student_mean
##   <chr>    <chr>   <dbl>        <dbl>
## 1 Rozz     midterm   1.3         1.8 
## 2 Andrew   midterm   2           1.85
## 3 Siouxsie midterm   1.7         1.35
## 4 Rozz     final     2.3         1.8 
## 5 Andrew   final     1.7         1.85
## 6 Siouxsie final     1           1.35
```

CASE STUDY: THE KING OF FRANCE

- presupposition:
  - piece of information required to be true for a sentence to make sense; not-at-issue content
  - examples:
    - “The King of France is bald”
    - “When did you stop beating your wife?”
    - “Make America great again!”

MATERIALS

- 5 critical conditions:

C0. The king of France is bald.

C1. France has a king, and he is bald.

C6. The King of France isn’t bald.

C9. The King of France, he did not call Emmanuel Macron last night.

C10. Emmanuel Macron, he did not call the King of France last night.

MATERIALS

- 5 vignettes:

V1. The King of France is bald.

V2. The Emperor of Canada is fond of sushi.

V3. The Pope’s wife is a lawyer.

V4. The Belgian rainforest provides a habitat for many species.

V5. The volcanoes of Germany dominate the landscape.

MATERIALS

- 5 “background check” questions:

BC1. France has a king.

BC2. The Pope is currently not married.

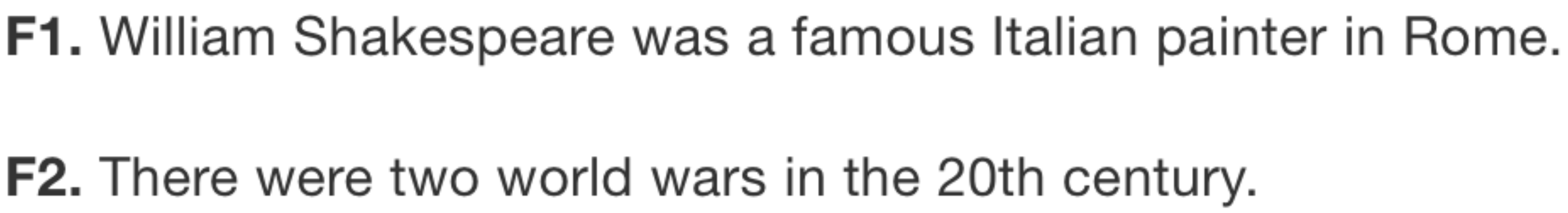
BC3. Canada is a democracy.

BC4. Belgium has rainforests.

BC5. Germany has volcanoes.

MATERIALS

- 110 filler sentences (also acting as controls)



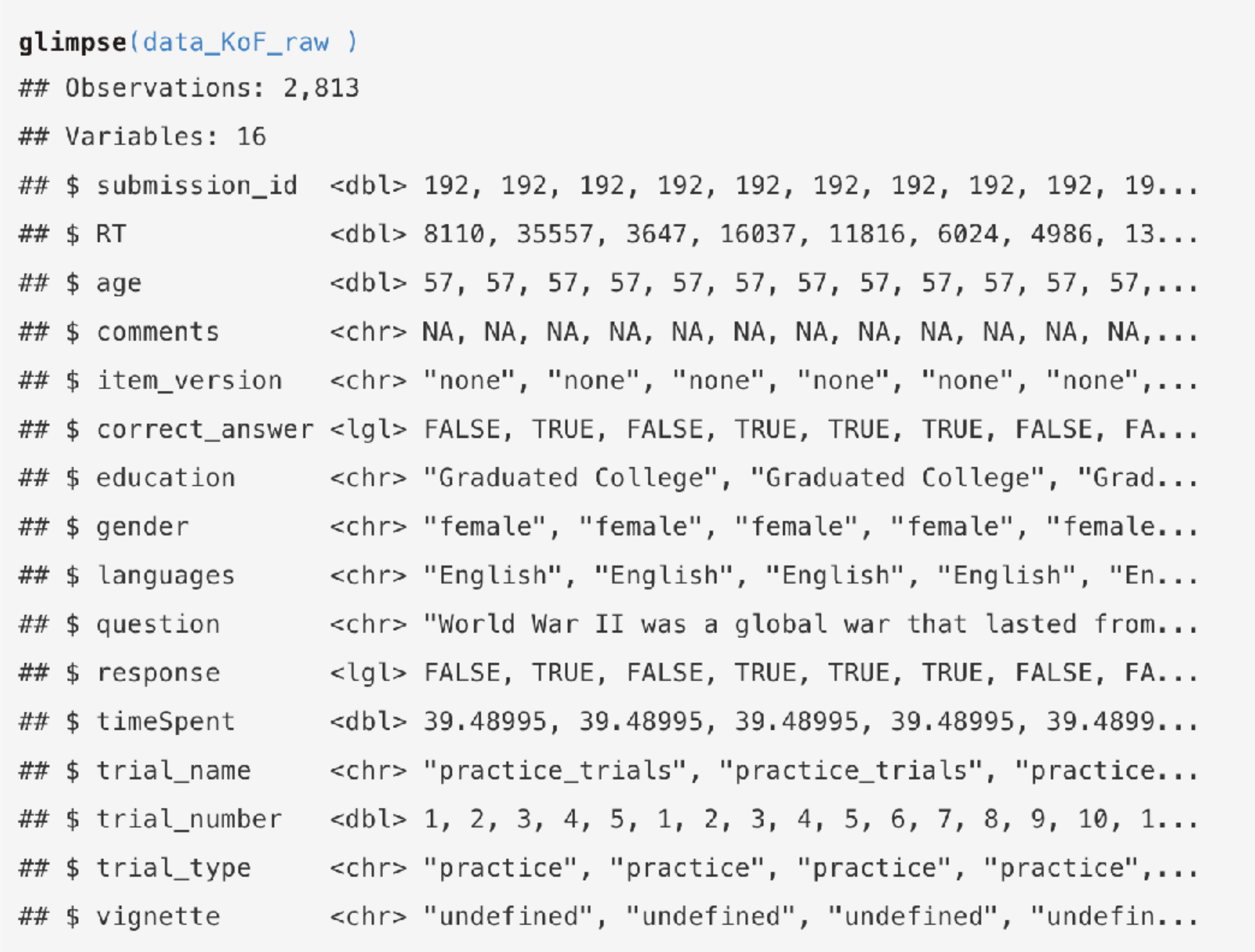
F1. William Shakespeare was a famous Italian painter in Rome.

F2. There were two world wars in the 20th century.

PARTICIPANTS & PROCEDURE

- participants:
  - N=97 recruited via Prolific
- procedure:
  - five initial practice trials (similar to fillers but disjoint)
  - main trials consisted of:
    - 5 critical trials
      - one for each vignette & one for each condition
      - completely at random
    - all 5 "background check" questions (*after* critical trials)
    - 14 random fillers

RAW DATA



```
glimpse(data_KoF_raw )
## Observations: 2,813
## Variables: 16
## $ submission_id  <dbl> 192, 192, 192, 192, 192, 192, 192, 192, 192, 19...
## $ RT             <dbl> 8110, 35557, 3647, 16037, 11816, 6024, 4986, 13...
## $ age            <dbl> 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57,...
## $ comments       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ item_version   <chr> "none", "none", "none", "none", "none", "none",...
## $ correct_answer <lgl> FALSE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE, FA...
## $ education      <chr> "Graduated College", "Graduated College", "Grad...
## $ gender         <chr> "female", "female", "female", "female", "female...
## $ languages      <chr> "English", "English", "English", "English", "En...
## $ question       <chr> "World War II was a global war that lasted from...
## $ response       <lgl> FALSE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE, FA...
## $ timeSpent      <dbl> 39.48995, 39.48995, 39.48995, 39.48995, 39.4899...
## $ trial_name     <chr> "practice_trials", "practice_trials", "practice...
## $ trial_number   <dbl> 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1...
## $ trial_type     <chr> "practice", "practice", "practice", "practice",...
## $ vignette       <chr> "undefined", "undefined", "undefined", "undefin...
```

ANY COMMENTS?

```
data_KoF_raw %>% pull(comments) %>% unique
```

```
1."I hope I was right most of the time!"
2."My level of education is Some Highschool, not finished. So I couldn't
 input what was correct, so I'm leaving a comment here."
3."It was interesting, and made re-read questions to make sure they weren't
 tricks. I hope I got them all correct."
4."Worked well"
5."A surprisingly tricky study! Thoroughly enjoyed completing it, despite
 several red herrings!!"
6."Thank you for the opportunity."
7."this was challenging"
8."I'm not good at learning history so i might of made couple of mistakes. I
 hope I did well. :)"
9."Interesting survey - thanks!"
10."Regarding the practice question - I'm aware that Alexander Bell invented
 the telephone, but in reality, it was a collaborative effort by a team of
 people"
11."Fun study!"
12."Fun stuff"
```

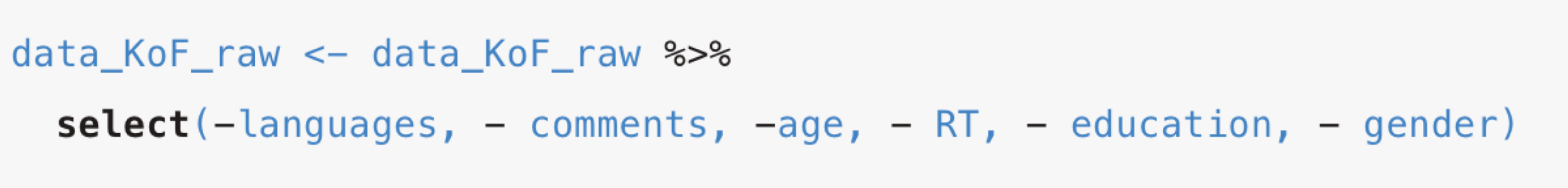
NATIVE LANGUAGES

```
data_KoF_raw %>% pull(languages) %>% unique
```

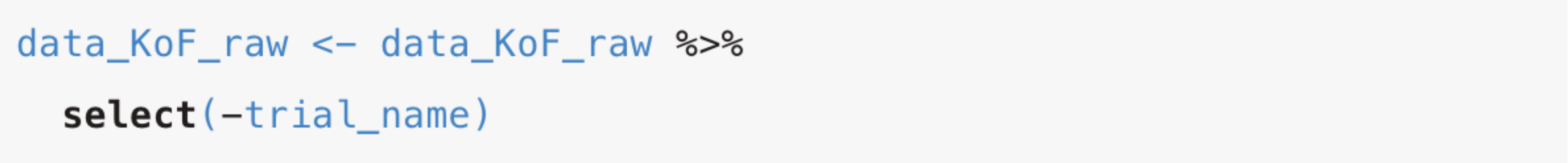
```
##  [1] "English"             "english"                "English, Italian"
##  [4] "English/ ASL"        "English and Polish"     "Chinese"
##  [7] "English, Mandarin"   "Polish"                 "Turkish"
## [10] NA                    "English, Sarcasm"       "English, Portuguese"
```

🥰

REMOVE IRRELEVANT COLUMNS

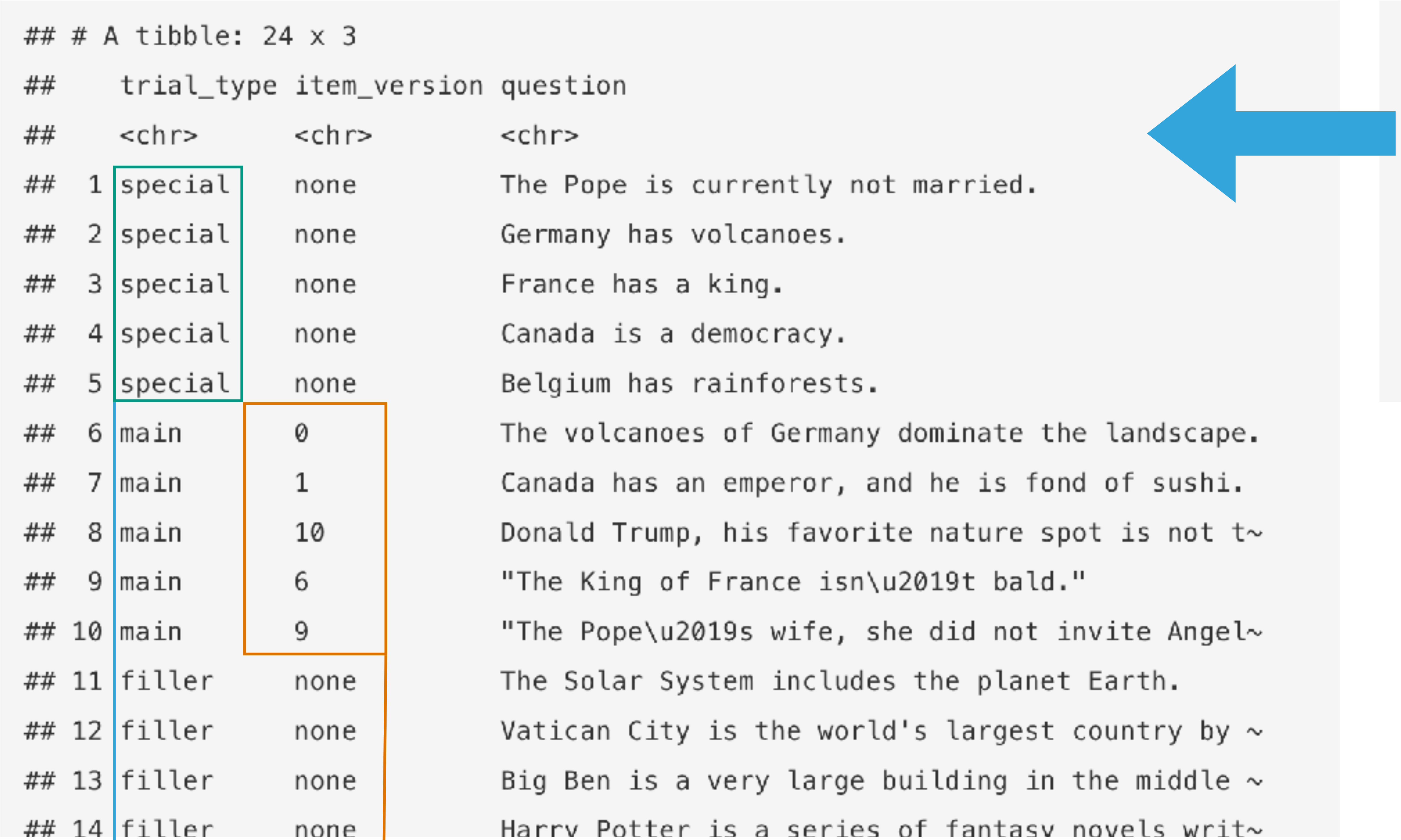


```
data_KoF_raw <- data_KoF_raw %>%
  select(-languages, - comments, -age, - RT, - education, - gender)
```

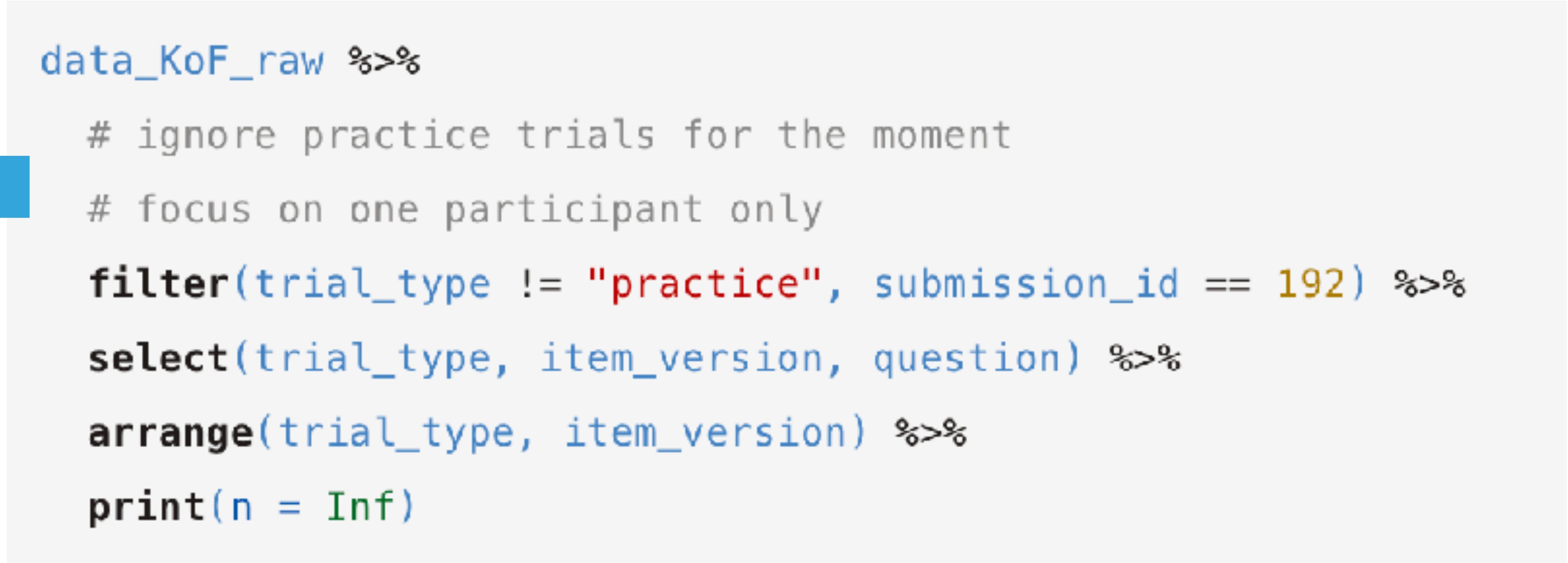


```
data_KoF_raw <- data_KoF_raw %>%
  select(-trial_name)
```

UNHELPFUL DISTRIBUTION OF INFORMATION



```
## # A tibble: 24 x 3
##    trial_type item_version question
##    <chr>      <chr>        <chr>
##  1 special    none         The Pope is currently not married.
##  2 special    none         Germany has volcanoes.
##  3 special    none         France has a king.
##  4 special    none         Canada is a democracy.
##  5 special    none         Belgium has rainforests.
##  6 main       0            The volcanoes of Germany dominate the landscape.
##  7 main       1            Canada has an emperor, and he is fond of sushi.
##  8 main       10           Donald Trump, his favorite nature spot is not t~
##  9 main       6            "The King of France isn\u2019t bald."
## 10 main       9            "The Pope\u2019s wife, she did not invite Angel~
## 11 filler     none         The Solar System includes the planet Earth.
## 12 filler     none         Vatican City is the world's largest country by ~
## 13 filler     none         Big Ben is a very large building in the middle ~
## 14 filler     none         Harry Potter is a series of fantasy novels writ~
```



```
data_KoF_raw %>%
  # ignore practice trials for the moment
  # focus on one participant only
  filter(trial_type != "practice", submission_id == 192) %>%
  select(trial_type, item_version, question) %>%
  arrange(trial_type, item_version) %>%
  print(n = Inf)
```

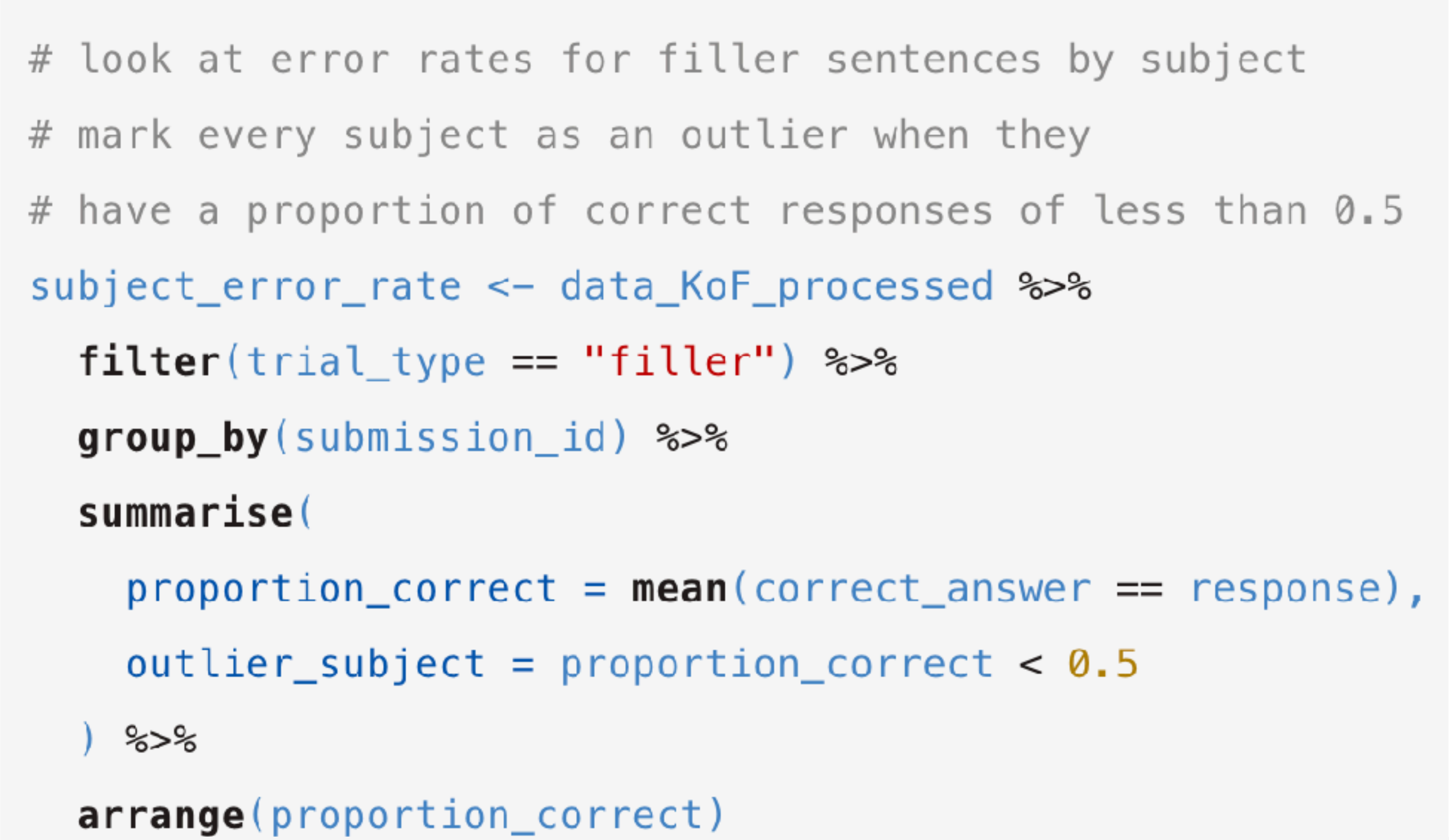
type of critical experimental condition

“background check” question

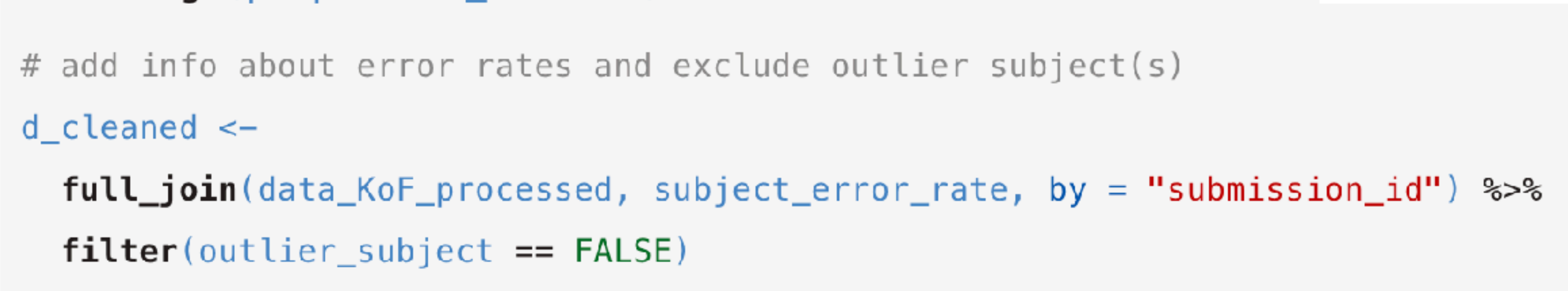
CREATING AN INFORMATIVE `CONDITION` COLUMN

```
data_KoF_processed <-  data_KoF_raw %>%
  # discard practice trials
  filter(trial_type != "practice") %>%
  mutate(
    # add a 'condition' variable
    condition = case_when(
      trial_type == "special" ~ "background check",
      trial_type == "main" ~ str_c("Condition ", item_version),
      TRUE ~ "filler"
    ) %>%
      # make the new 'condition' variable a factor
      factor(
        ordered = T,
        levels = c(
          str_c("Condition ", c(0, 1, 6, 9, 10)),
          "background check", "filler"
        )
      )
  )
```

CLEANING BY-PARTICIPANT



```r
# look at error rates for filler sentences by subject
# mark every subject as an outlier when they
# have a proportion of correct responses of less than 0.5
subject_error_rate <- data_KoF_processed %>%
  filter(trial_type == "filler") %>%
  group_by(submission_id) %>%
  summarise(
    proportion_correct = mean(correct_answer == response),
    outlier_subject = proportion_correct < 0.5
  ) %>%
  arrange(proportion_correct)
```



```r
# add info about error rates and exclude outlier subject(s)
d_cleaned <-
  full_join(data_KoF_processed, subject_error_rate, by = "submission_id") %>%
  filter(outlier_subject == FALSE)
```

CLEANING BY-TRIAL

```
# exclude every critical trial whose 'background' test question was answered wrongly
d_cleaned <-
  d_cleaned %>%
  # select only the 'background question' trials
  filter(trial_type == "special") %>%
  # is the background question answered correctly?
  mutate(
    background_correct = correct_answer == response
  ) %>%
  # select only the relevant columns
  select(submission_id, vignette, background_correct) %>%
  # right join lines to original data set
  right_join(d_cleaned, by = c("submission_id", "vignette")) %>%
  # remove all special trials, as well as main trials with incorrect background check
  filter(trial_type == "main" & background_correct == TRUE)
```

FINAL EXAM

- Friday February 7 2020 ::: 4-8pm
- 66/E33 & 66/E34
- no class at noon on that day

HOMEWORK

- [voluntarily] do small experiment (see email on StudIP)
- work on HW1
  - to be submitted next Friday before noon
- put exam date in your agenda!