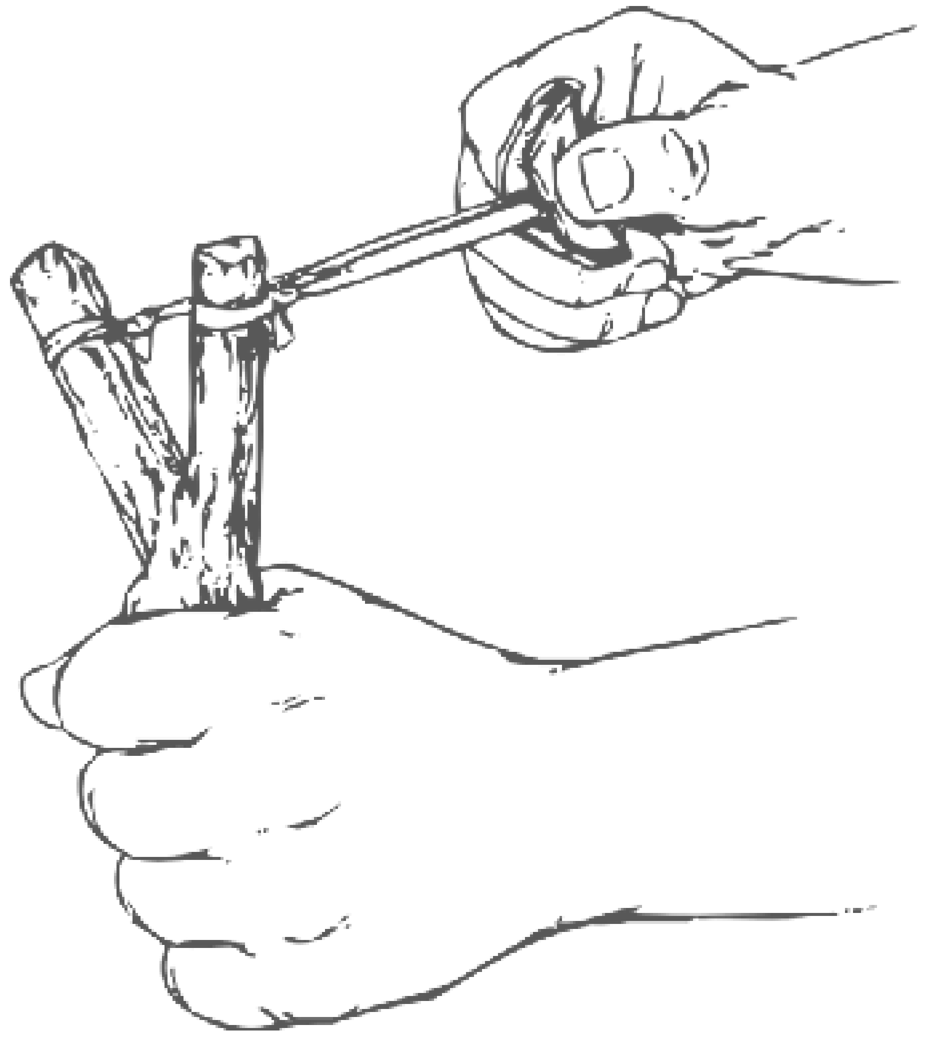
DATA ANALYSIS

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 Input/Output

- 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")
```

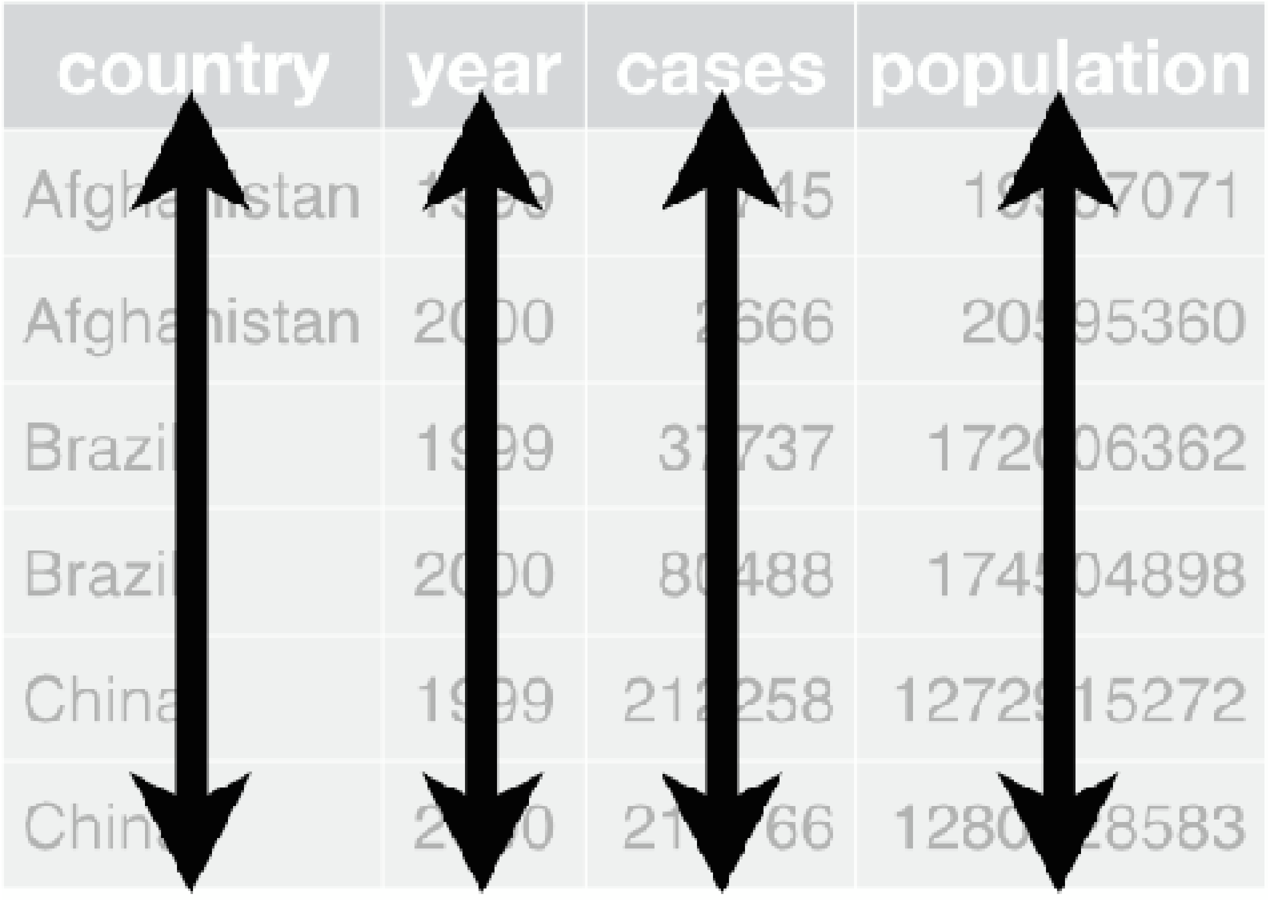
- write data to file

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

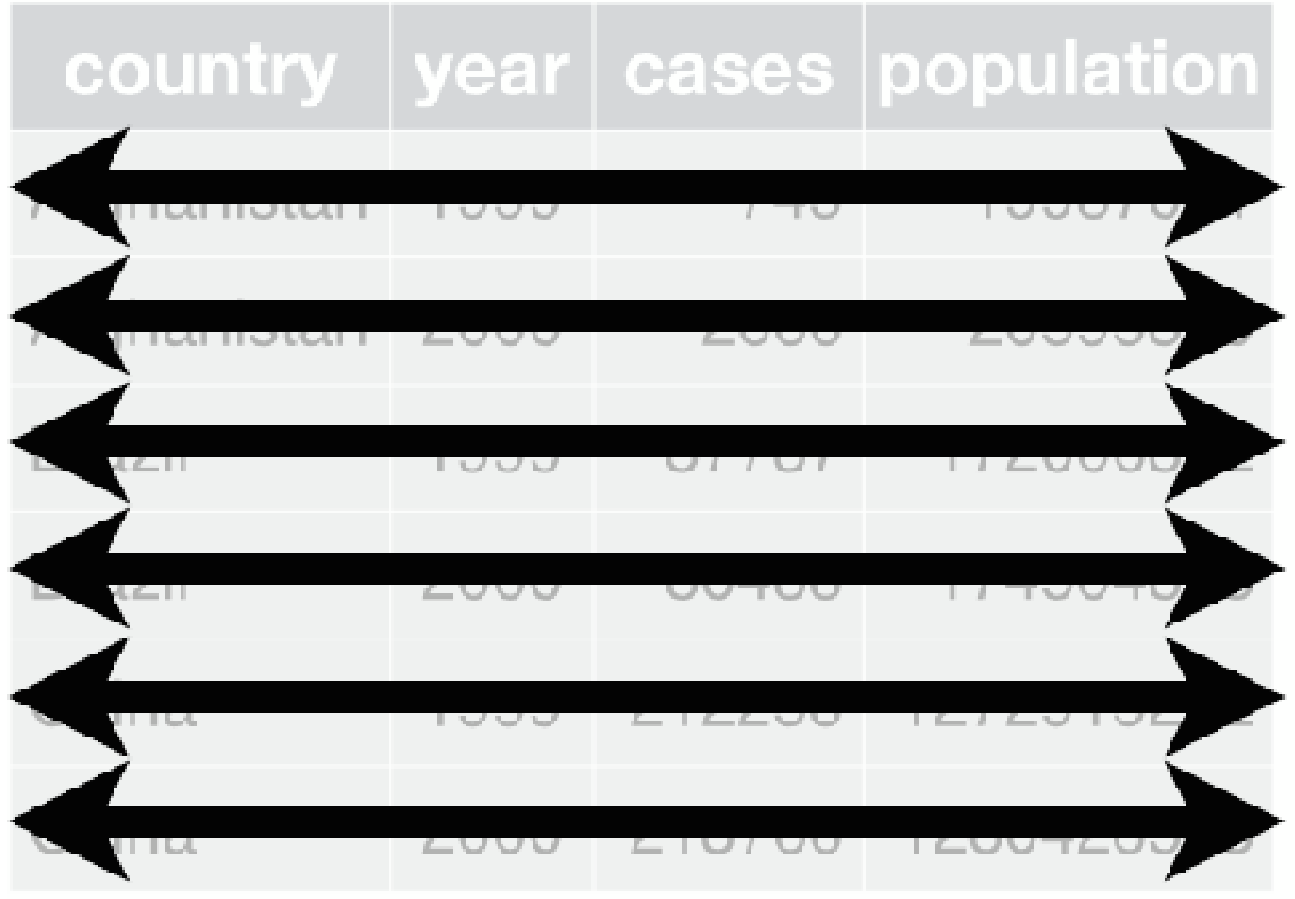
TIDY DATA

- 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

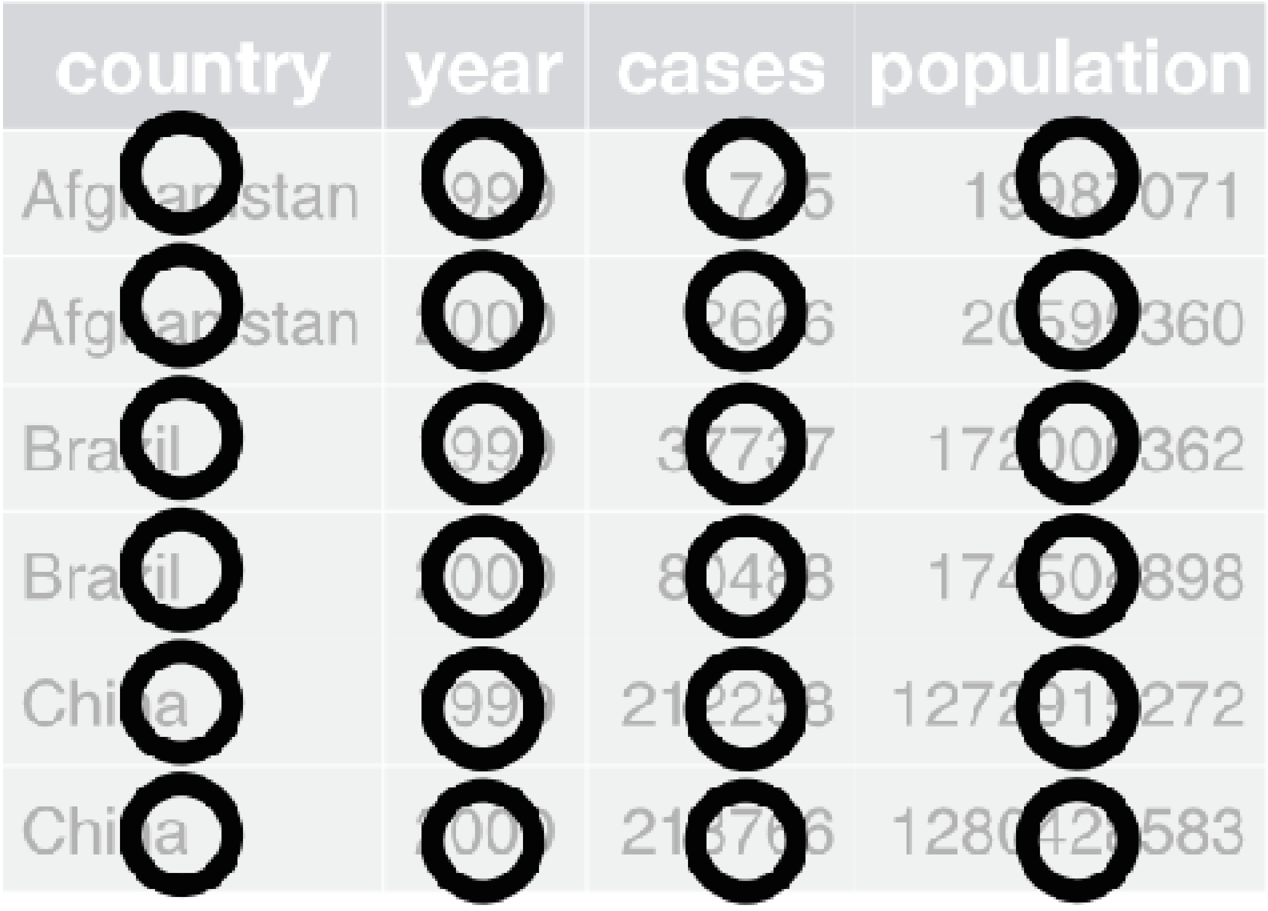
- data which is not tidy is messy
- data that satisfies 1 & 2 is almost tidy



variables



observations



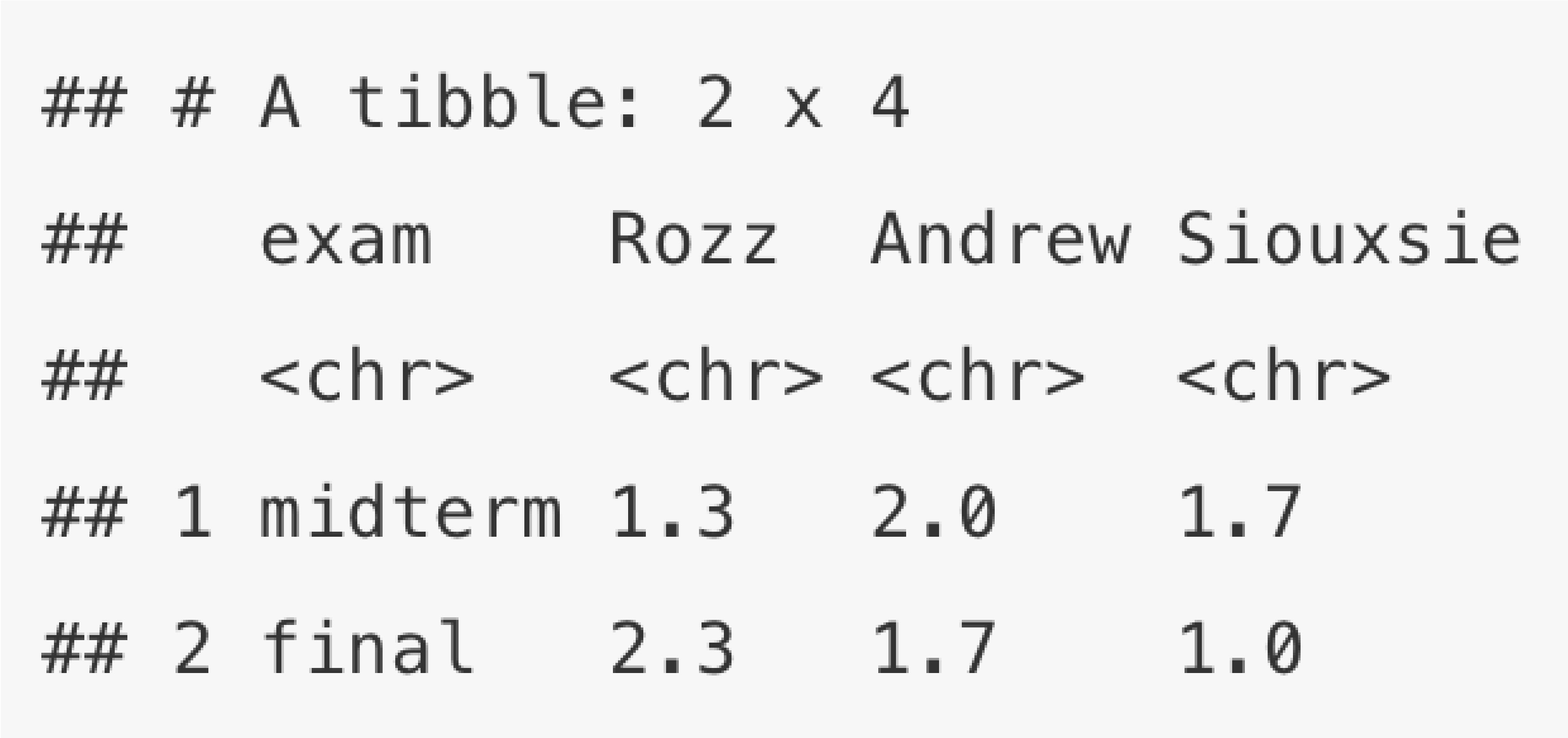
values

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
```

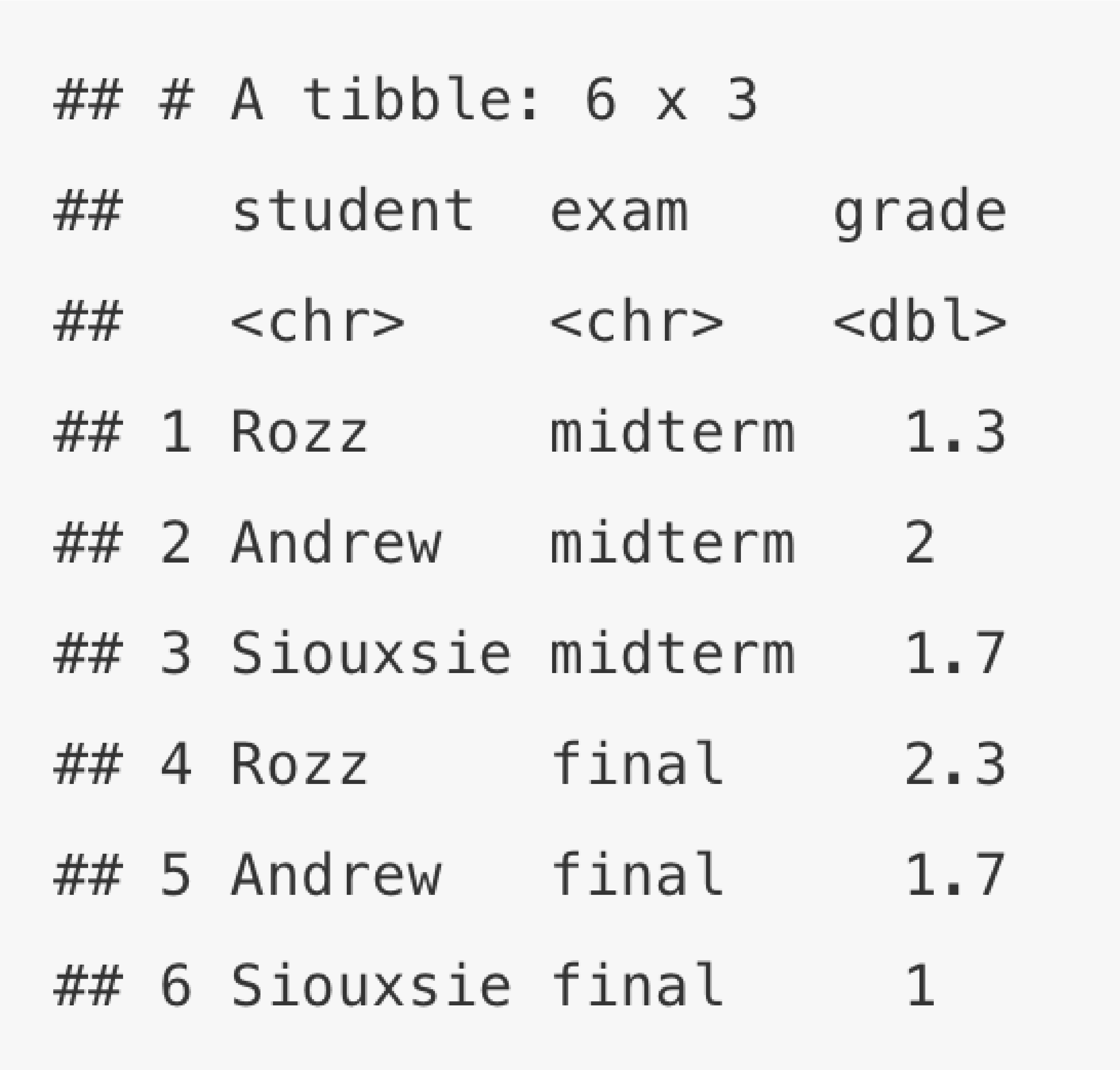
```
## # A tibble: 2 x 4
##   exam    Rozz  Andrew Siouxsie
##   <chr>   <chr> <chr>  <chr>
## 1 midterm 1.3   2.0    1.7
## 2 final   2.3   1.7    1.0
```

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")
```

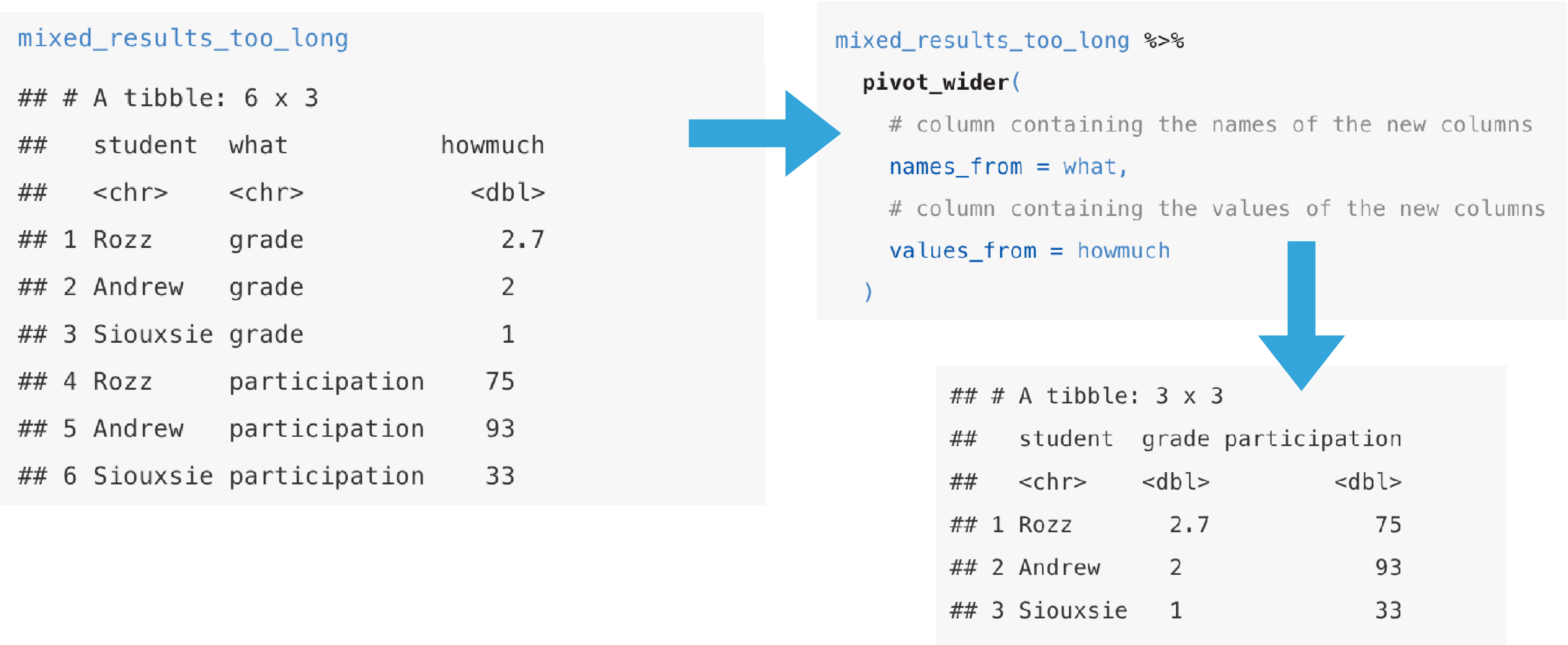
PIVOTING: LONGER

```
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 %>%
  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
```

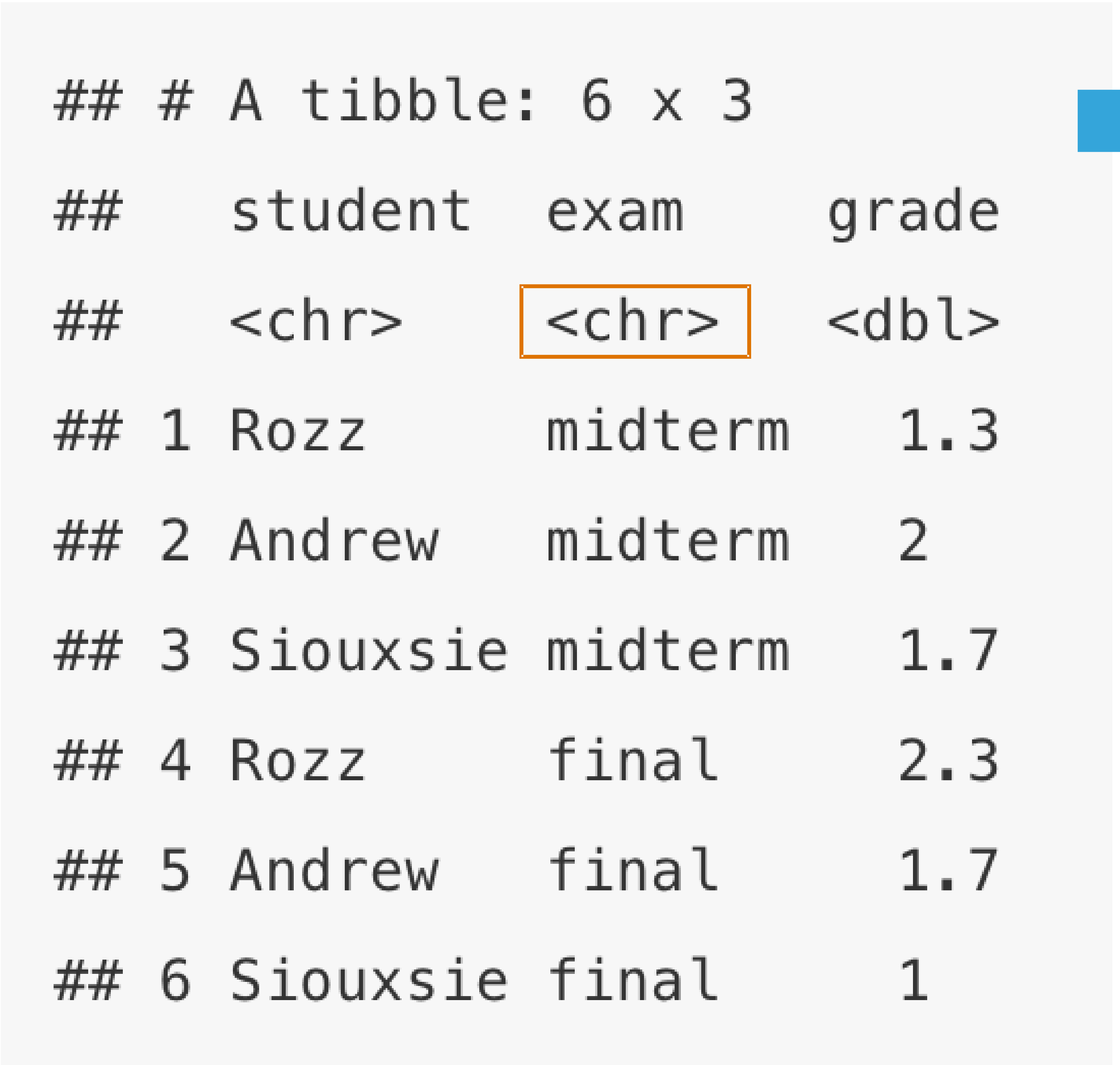
```
exam_results_tidy %>%
  select(grade, exam)

## # 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

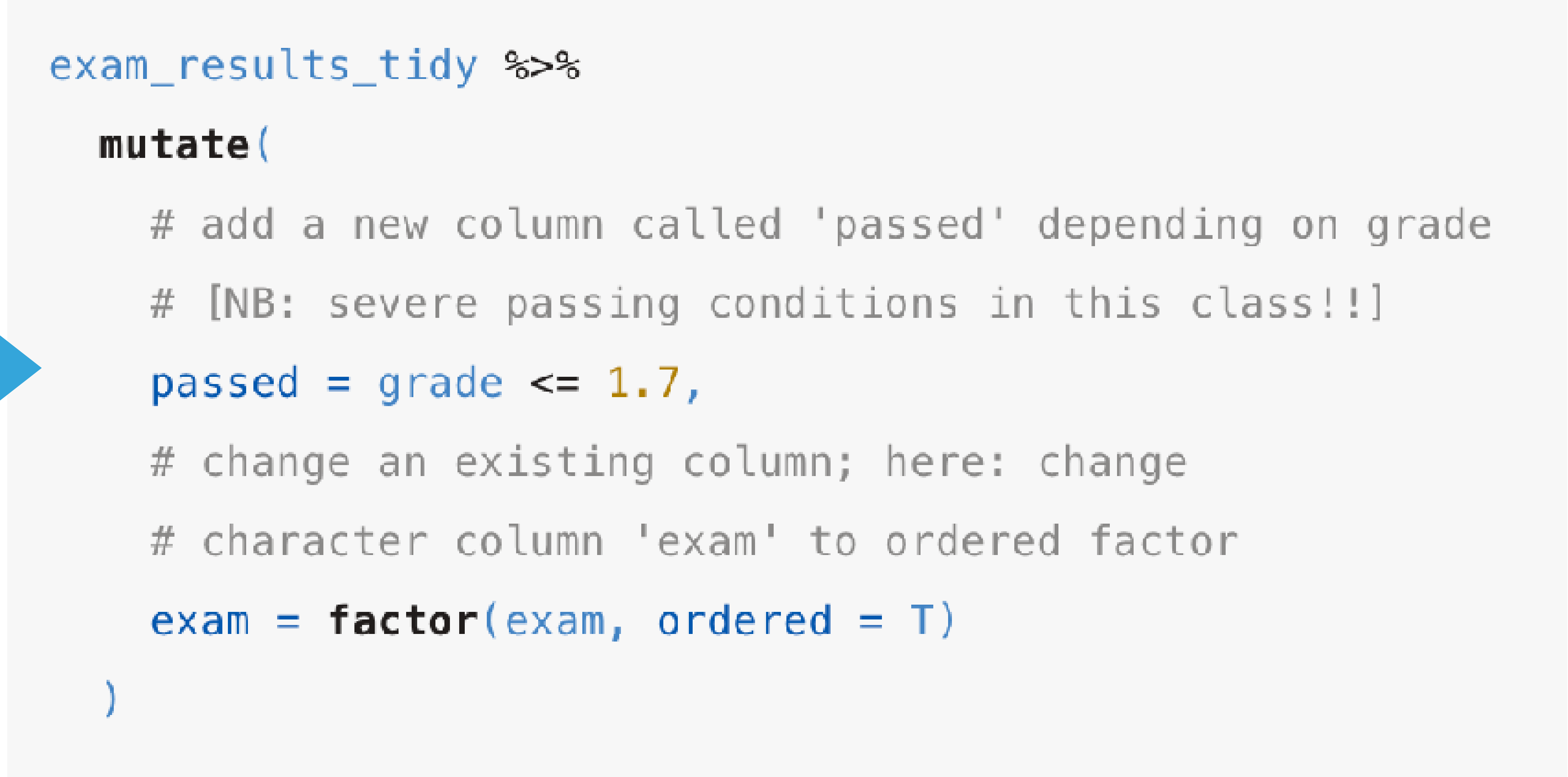
- from tidyselect package

```
# 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)
  )
```

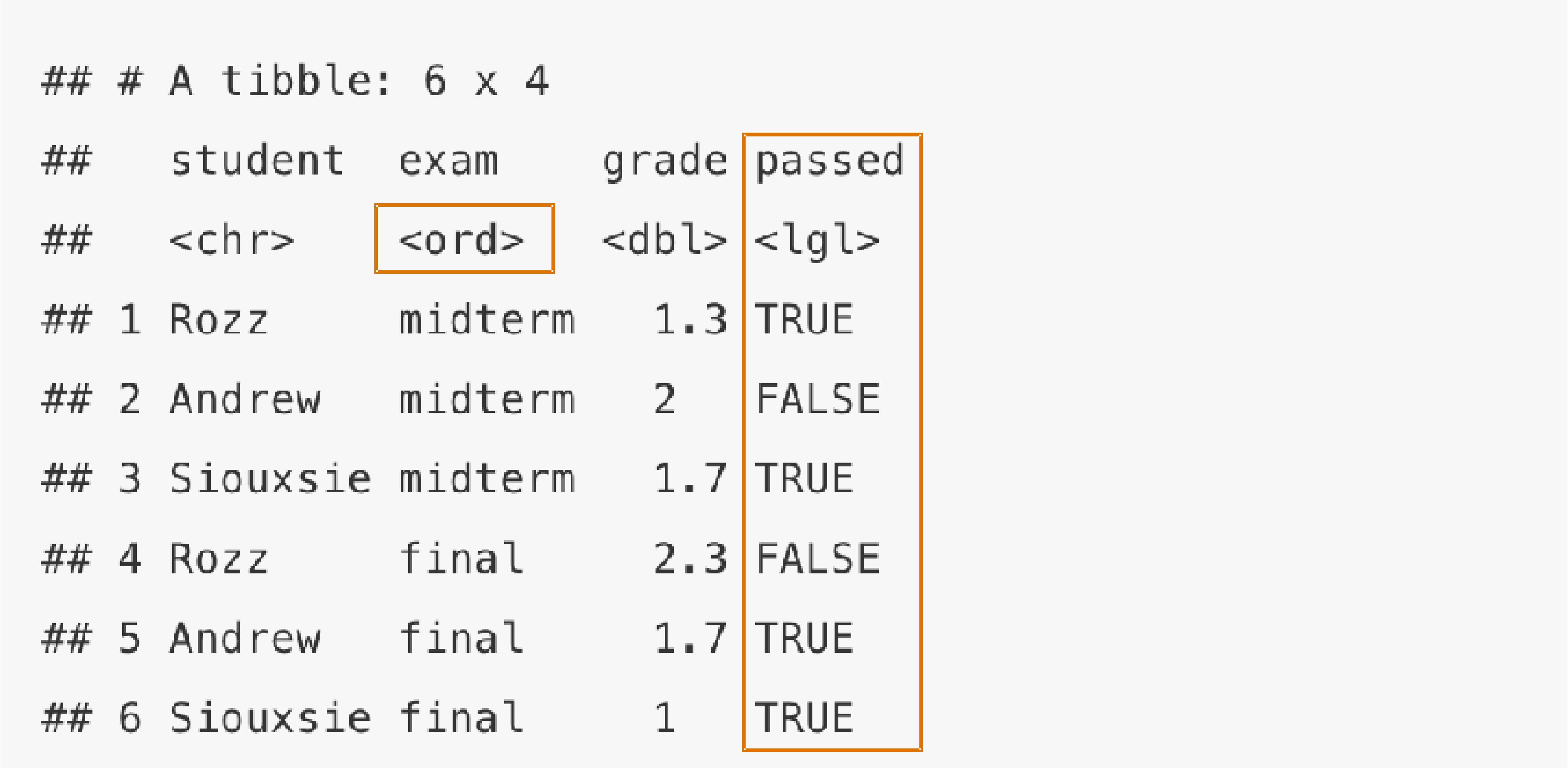
ADDING OR CHANGING COLUMNS



```
## # 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
```

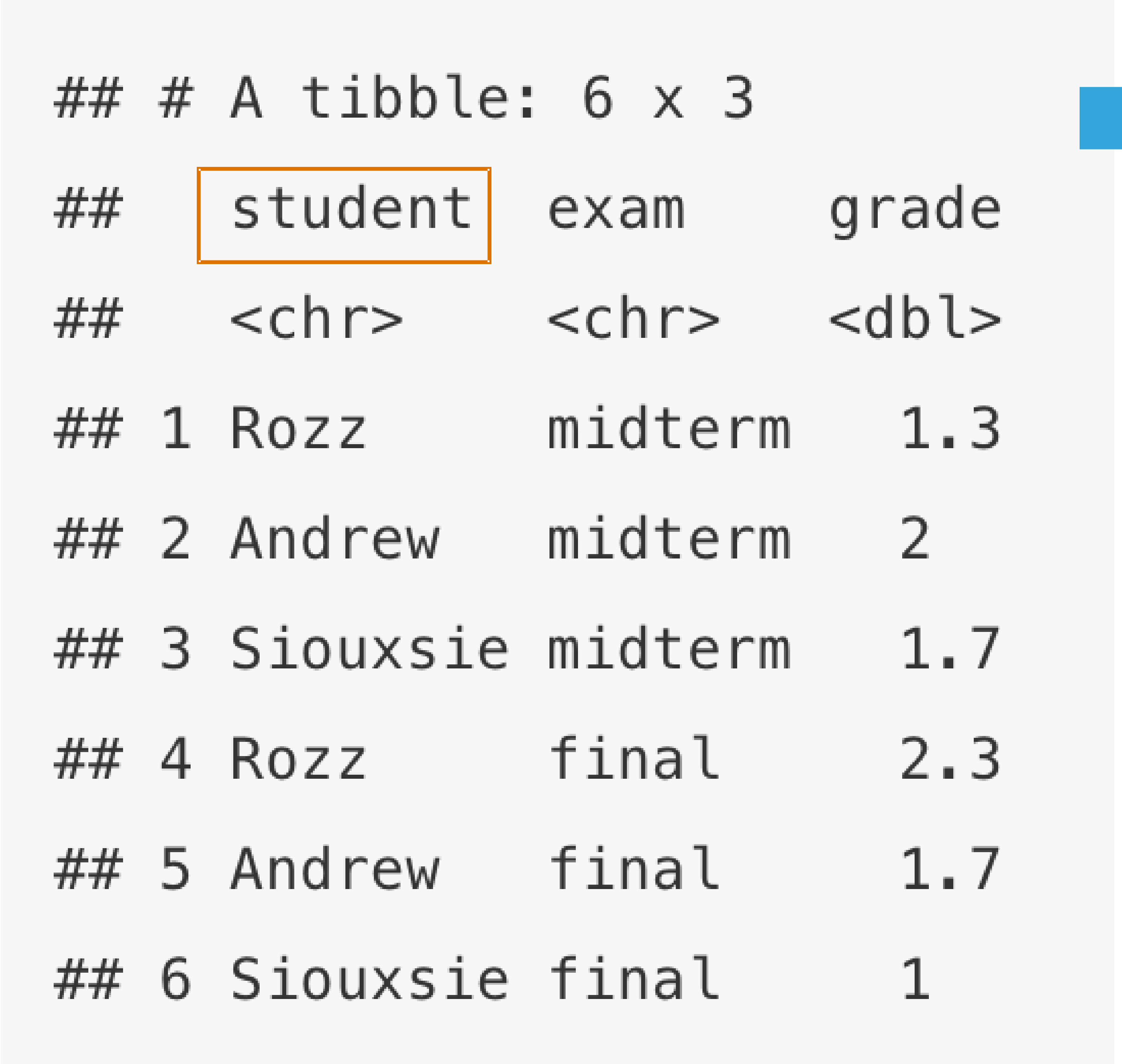


```
exam_results_tidy %>%
  mutate(
    # add a new column called 'passed' depending on grade
    # [NB: severe passing conditions in this class!!]
    passed = grade <= 1.7,
    # change an existing column; here: change
    # character column 'exam' to ordered factor
    exam = factor(exam, ordered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

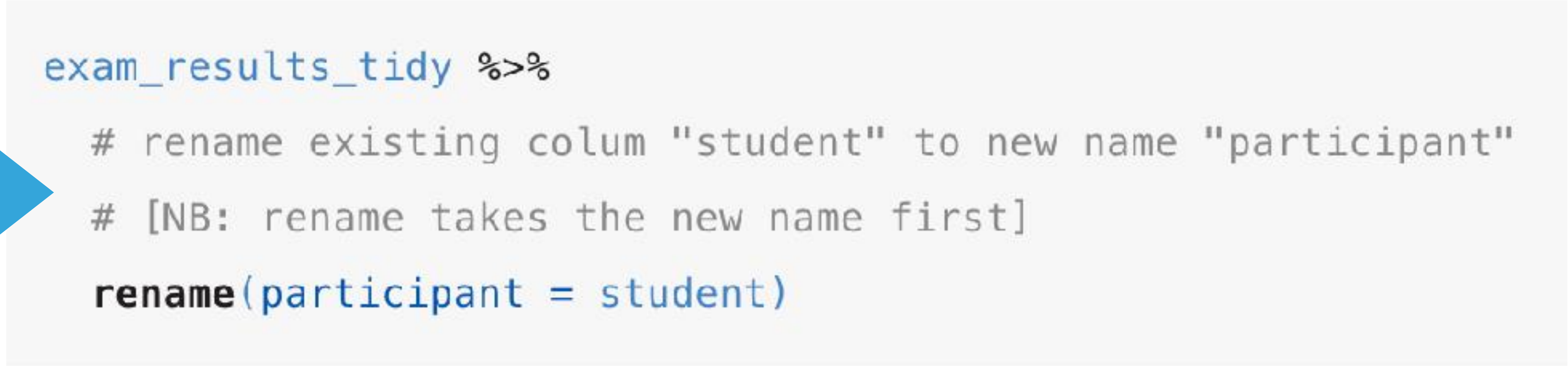


```
## # A tibble: 6 x 4
##   student  exam    grade passed
##   <chr>    <ord>   <dbl> <lgl>
## 1 Rozz     midterm   1.3 TRUE
## 2 Andrew   midterm   2   FALSE
## 3 Siouxsie midterm   1.7 TRUE
## 4 Rozz     final     2.3 FALSE
## 5 Andrew   final     1.7 TRUE
## 6 Siouxsie final     1   TRUE
```

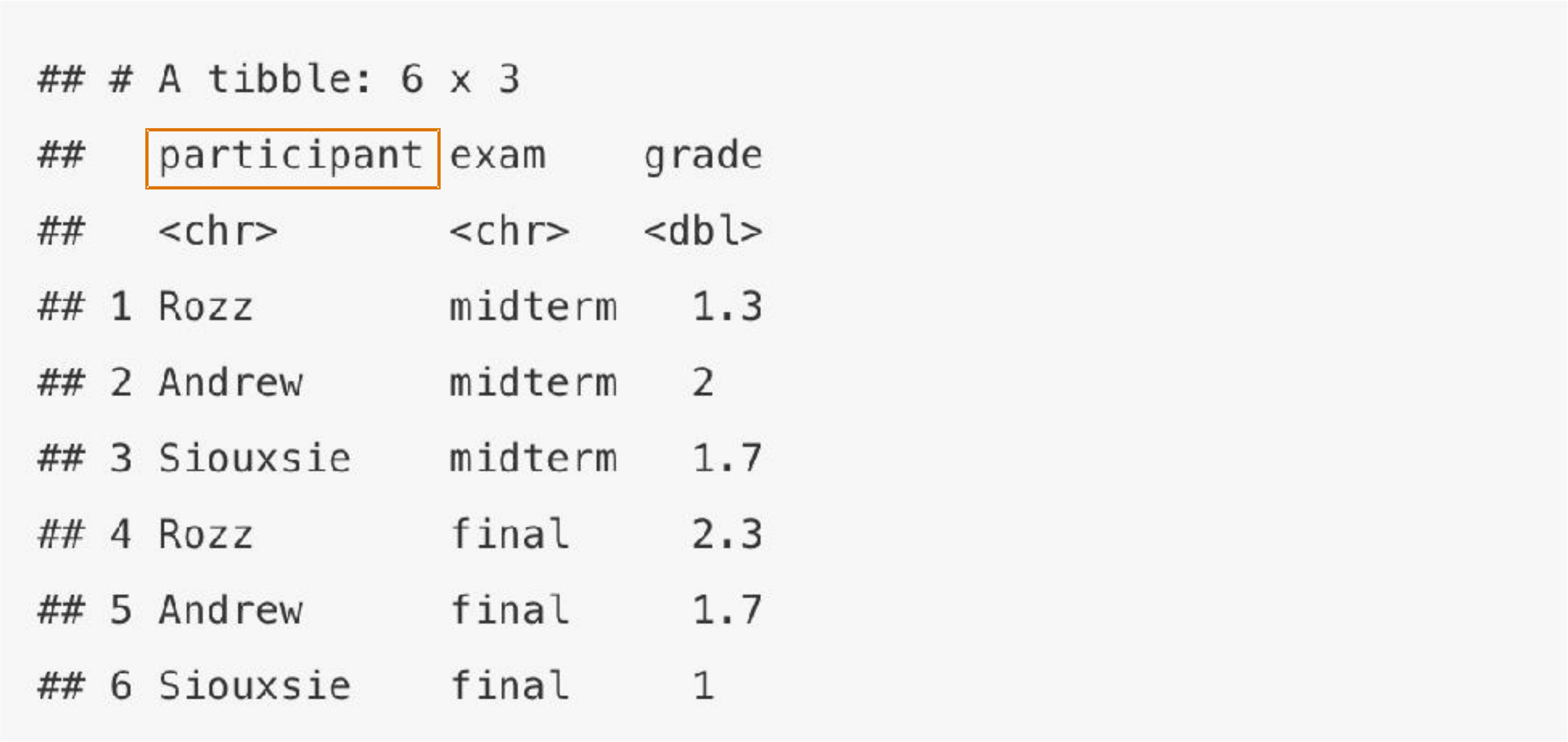
RENAMING COLUMNS



```
## # 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

```
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"
  )
```

```
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
```

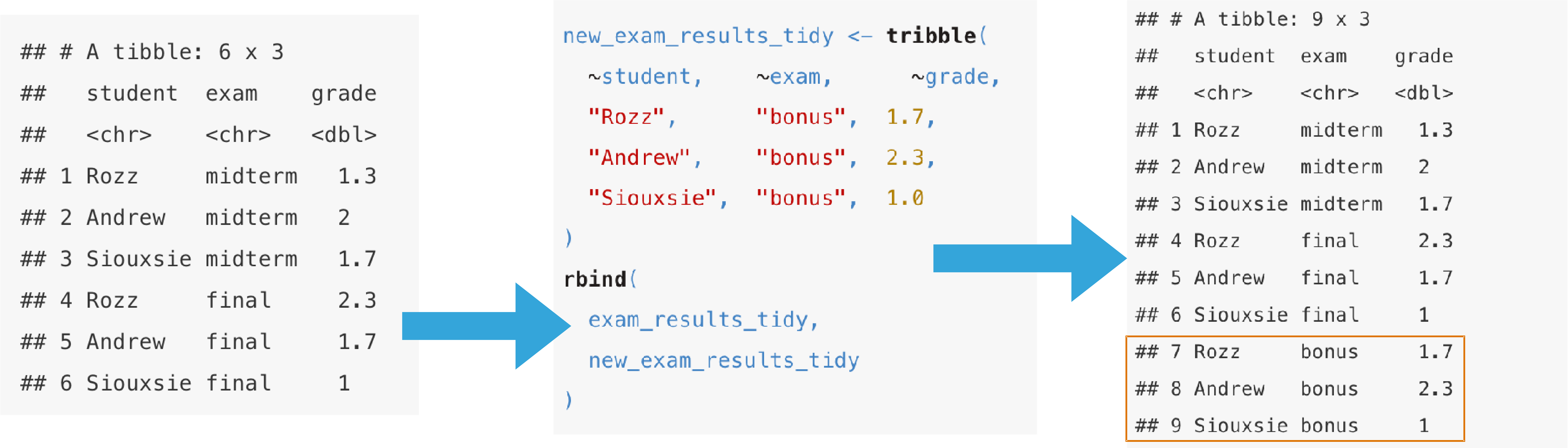
SORTING

```
## # 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: SUMMARIZE

```
## # 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

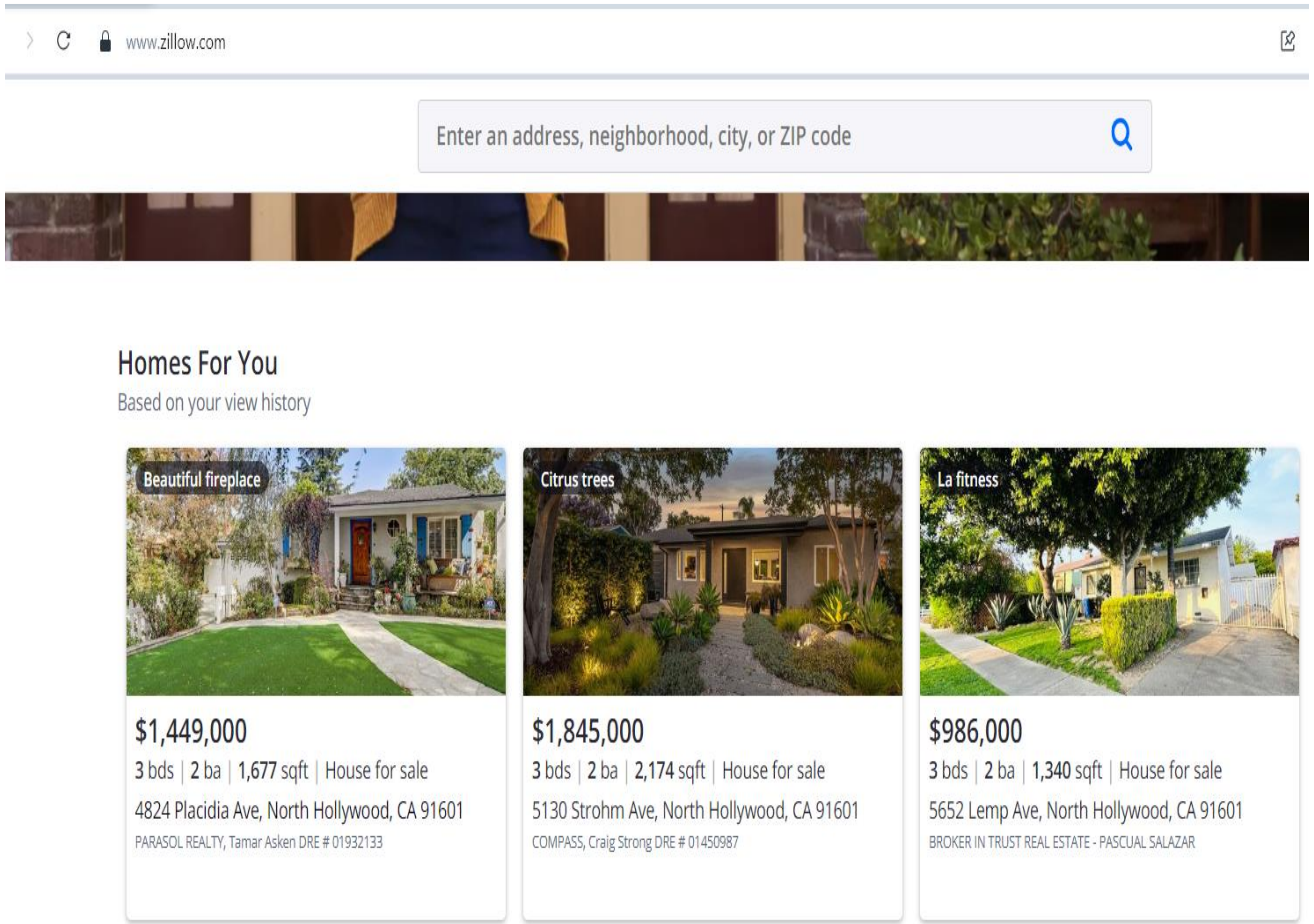
```
## # A tibble: 6 x 3
##   student  exam    grade
##   <chr>    <chr>   <dbl>
## 1 Rozz     midterm   1.3
## 2 Andrew   midterm   2
## 3 Siouxsie midterm   1.7
## 4 Rozz     final     2.3
## 5 Andrew   final     1.7
## 6 Siouxsie final     1
```

```
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: Zillow Housing Data

- introduction:
  - provides housing price data at the metro, city and zipcode levels
- website:
  - www.zillow.com



www.zillow.com

Enter an address, neighborhood, city, or ZIP code

Homes For You

Based on your view history

Beautiful fireplace

$1,449,000

3 bds | 2 ba | 1,677 sqft | House for sale

4824 Placidia Ave, North Hollywood, CA 91601

PARASOL REALTY, Tamar Asken DRE # 01932133

Citrus trees

$1,845,000

3 bds | 2 ba | 2,174 sqft | House for sale

5130 Strohm Ave, North Hollywood, CA 91601

COMPASS, Craig Strong DRE # 01450987

La fitness

$986,000

3 bds | 2 ba | 1,340 sqft | House for sale

5652 Lemp Ave, North Hollywood, CA 91601

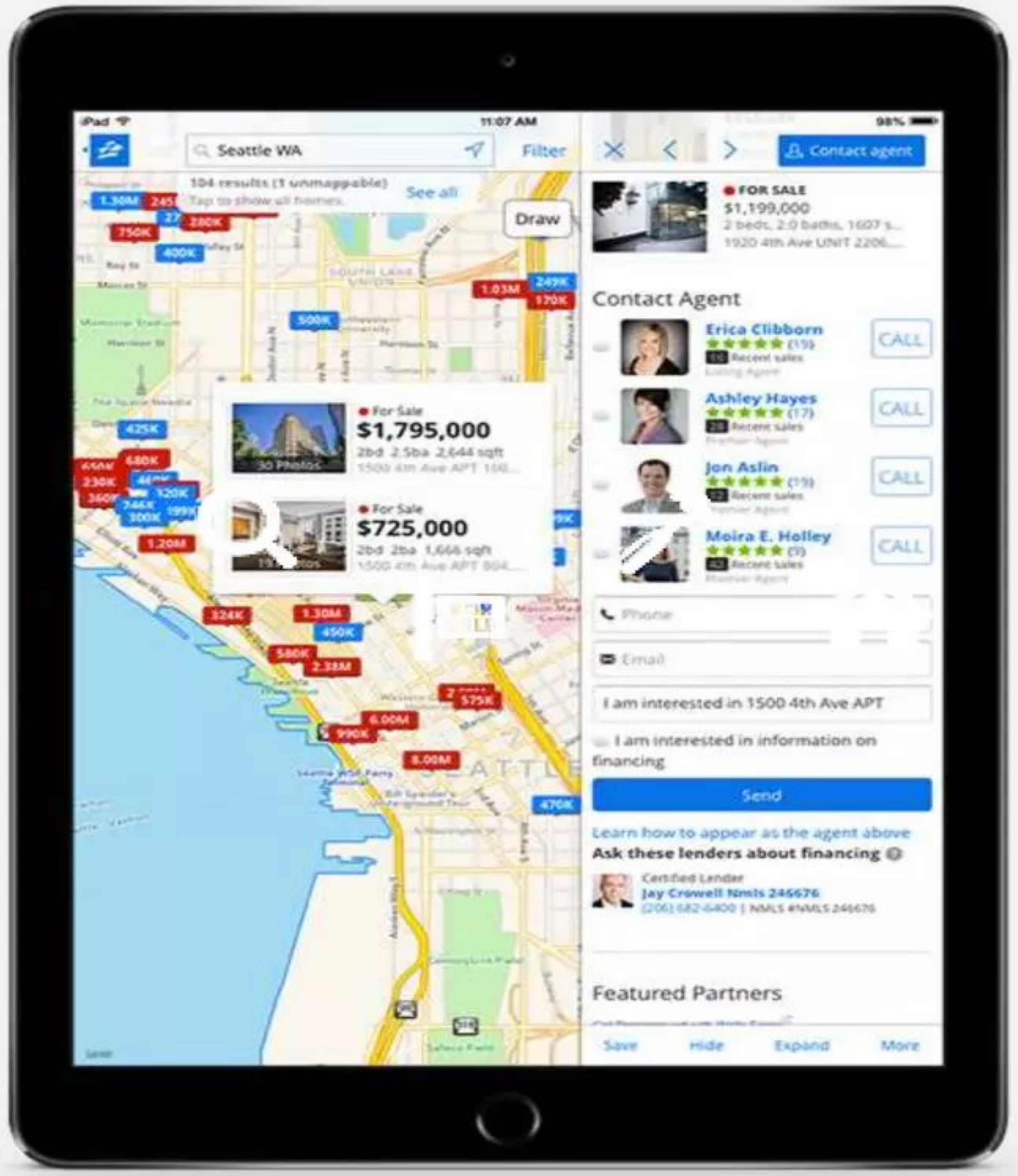
BROKER IN TRUST REAL ESTATE - PASCUAL SALAZAR

Zillow's marketplace

BUYERS & SELLERS

RENTERS

HOMEOWNERS



For Sale
$1,795,000
2bd 2.5ba 2,644 sqft

For Sale
$725,000
2bd 2ba 1,666 sqft

FOR SALE
$1,199,000

Contact Agent

Erica Clibborn CALL

Ashley Hayes CALL

Jon Aslin CALL

Moira E. Holley CALL

Phone

Email

I am interested in 1500 4th Ave APT

I am interested in information on financing

Send

Learn how to appear as the agent above

Ask these lenders about financing

Certified Lender
Jay Crowell Nmls 246676
(206) 682-6400 | NMLS #NMLS 246676

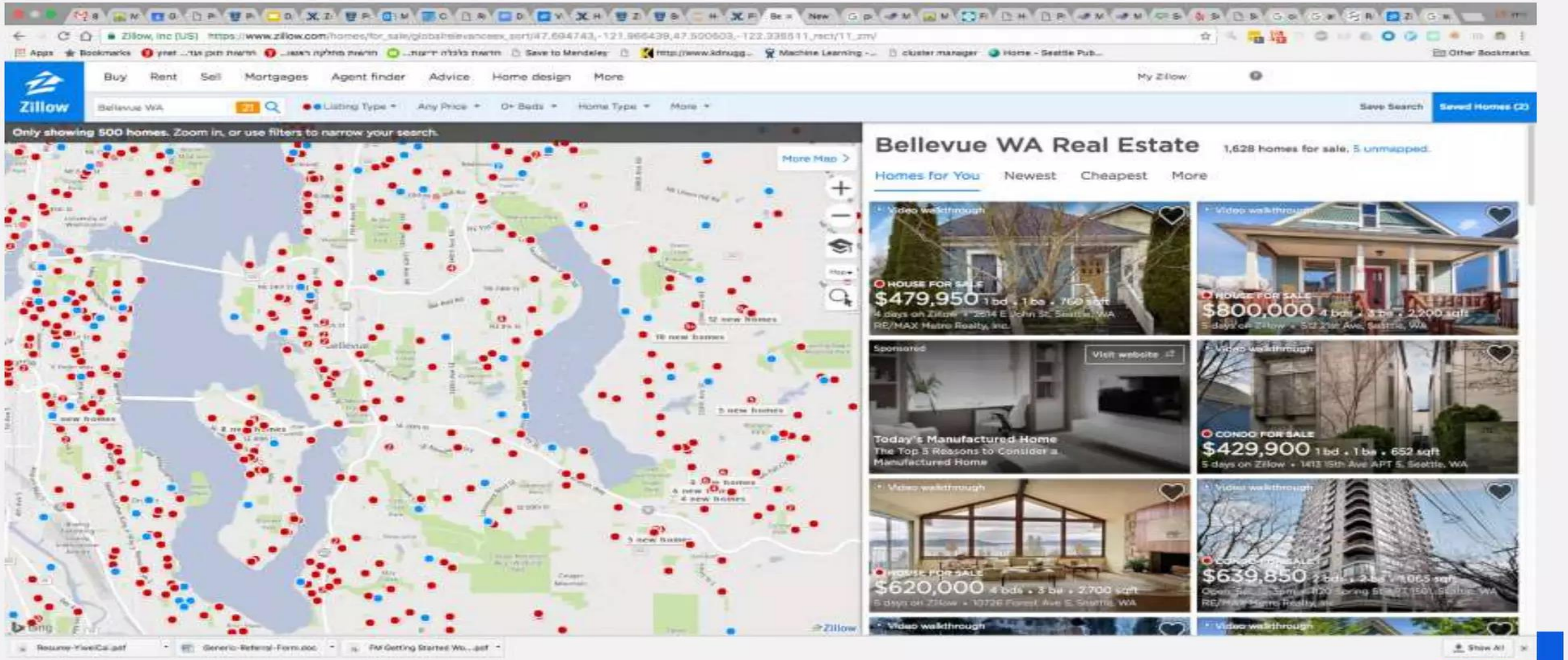
Featured Partners

REAL ESTATE AGENTS

PROPERTY MANAGERS & LANDLORDS

MORTGAGE PROVIDERS

For buyers



Only showing 500 homes. Zoom in, or use filters to narrow your search.

Bellevue WA Real Estate 1,628 homes for sale. 5 unmapped.

HOUSE FOR SALE
$479,950 1 bd · 1 ba · 760 sqft
4 days on Zillow · 2814 E John St, Seattle, WA
RE/MAX Metro Realty, Inc.

HOUSE FOR SALE
$800,000 4 bds · 3 ba · 2,200 sqft
5 days on Zillow · 512 21st Ave, Seattle, WA

Today's Manufactured Home
The Top 5 Reasons to Consider a Manufactured Home

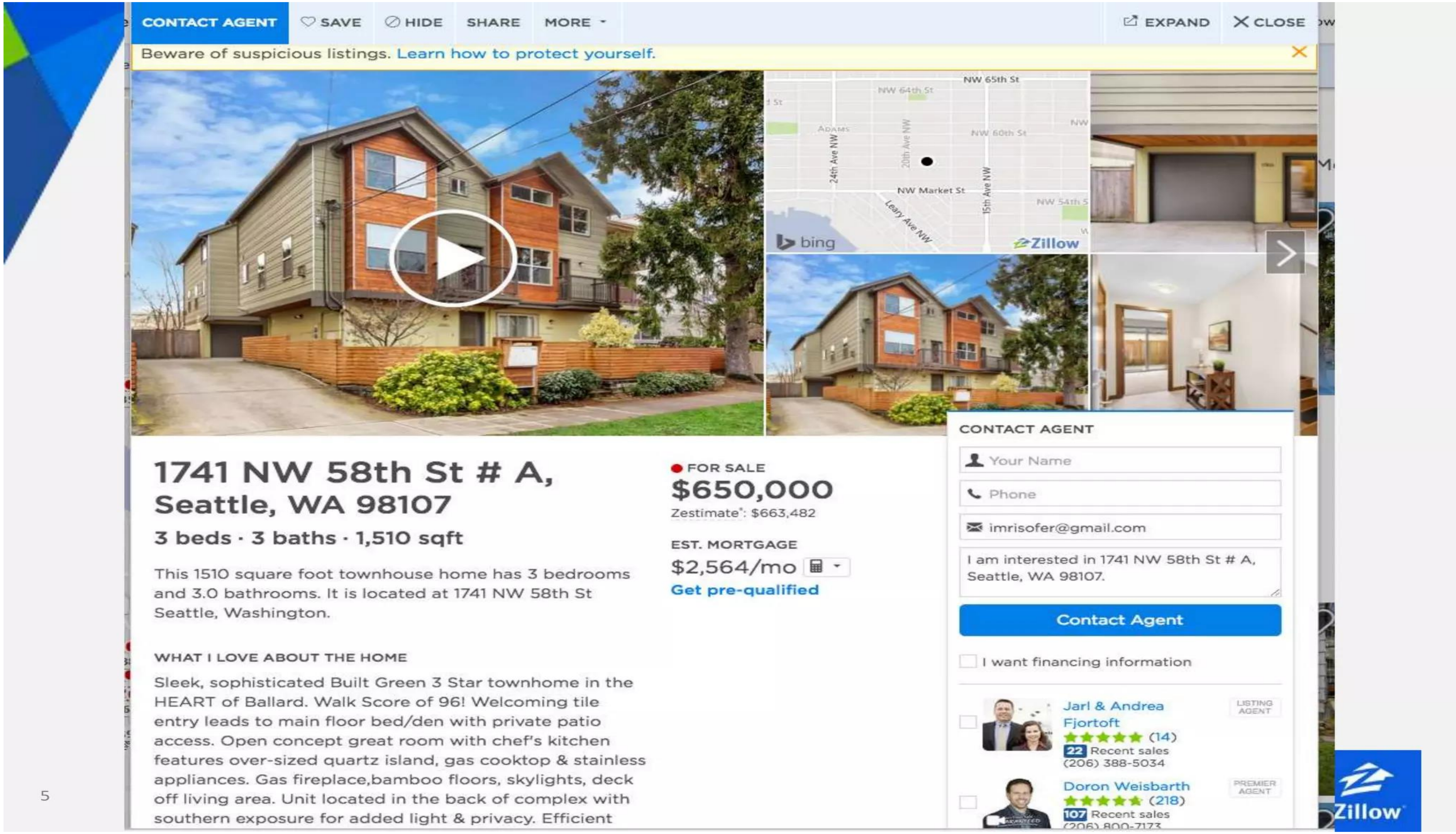
CONDO FOR SALE
$429,900 1 bd · 1 ba · 652 sqft
5 days on Zillow · 1413 15th Ave APT 5, Seattle, WA

HOUSE FOR SALE
$620,000 4 bds · 3 ba · 2,700 sqft
5 days on Zillow · 10726 Forest Ave S, Seattle, WA

CONDO FOR SALE
$639,850 2 bds · 2 ba · 1,065 sqft
Open: Sat 1-3pm · 1120 Spring St APT 1501, Seattle, WA
RE/MAX Metro Realty, Inc.

Resume-YiweiCai.pdf
Generic-Referral-Form.doc
PM Getting Started Wo....pdf

DATA ANALYSIS



Beware of suspicious listings. Learn how to protect yourself.

1741 NW 58th St # A, Seattle, WA 98107

3 beds · 3 baths · 1,510 sqft

● FOR SALE
$650,000
Zestimate®: $663,482

EST. MORTGAGE
$2,564/mo
Get pre-qualified

This 1510 square foot townhouse home has 3 bedrooms and 3.0 bathrooms. It is located at 1741 NW 58th St Seattle, Washington.

WHAT I LOVE ABOUT THE HOME

Sleek, sophisticated Built Green 3 Star townhome in the HEART of Ballard. Walk Score of 96! Welcoming tile entry leads to main floor bed/den with private patio access. Open concept great room with chef's kitchen features over-sized quartz island, gas cooktop & stainless appliances. Gas fireplace,bamboo floors, skylights, deck off living area. Unit located in the back of complex with southern exposure for added light & privacy. Efficient

CONTACT AGENT

Your Name
Phone
imrisofer@gmail.com
I am interested in 1741 NW 58th St # A, Seattle, WA 98107.

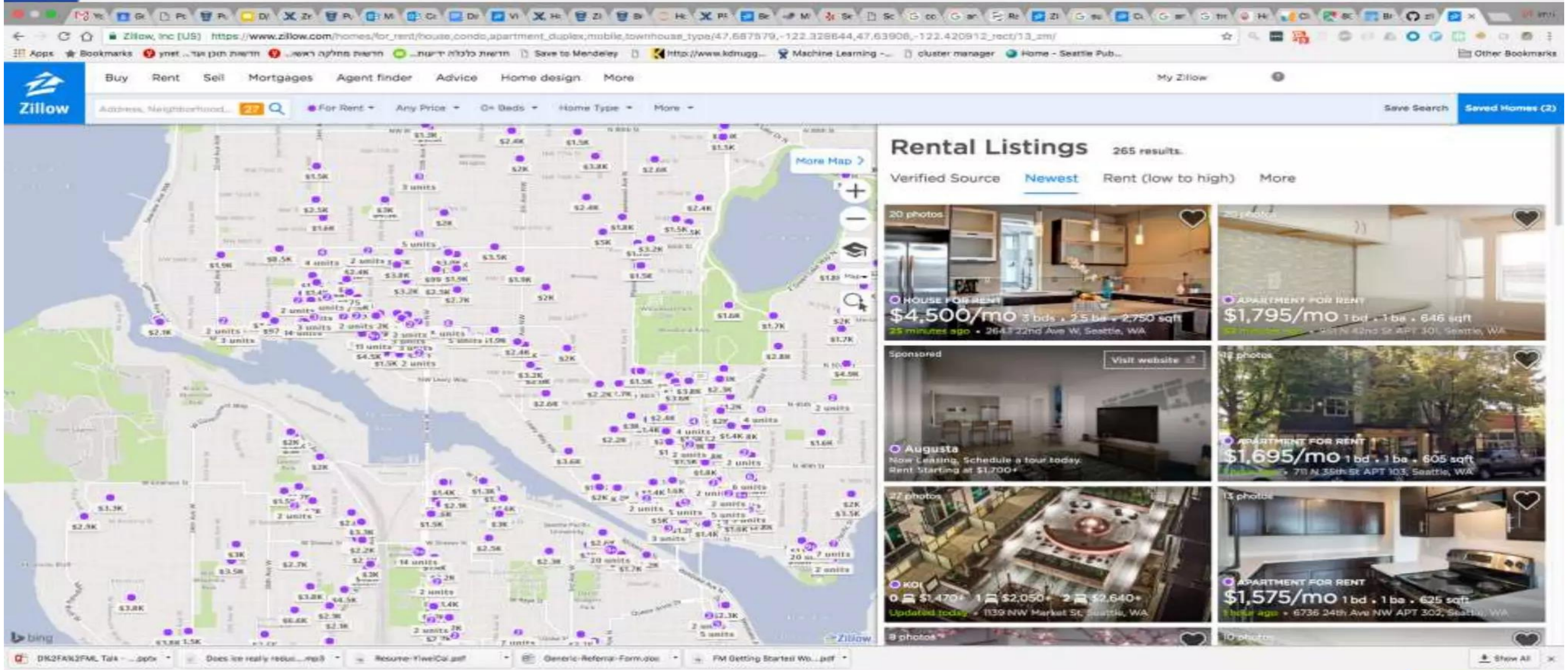
Contact Agent

□ I want financing information

□ Jarl & Andrea Fjortoft — LISTING AGENT
★★★★★ (14)
22 Recent sales
(206) 388-5034

□ Doron Weisbarth — PREMIER AGENT
★★★★★ (218)
107 Recent sales
(206) 800-7173

For renters



Rental Listings

265 results

Verified Source | Newest | Rent (low to high) | More

20 photos
HOUSE FOR RENT
$4,500/mo 3 bds · 2.5 ba · 2,750 sqft
25 minutes ago · 2643 22nd Ave W, Seattle, WA

APARTMENT FOR RENT
$1,795/mo 1 bd · 1 ba · 646 sqft
· N 42nd St APT 301, Seattle, WA

Sponsored
Visit website
Augusta
Now Leasing, Schedule a tour today.
Rent Starting at $1,700+

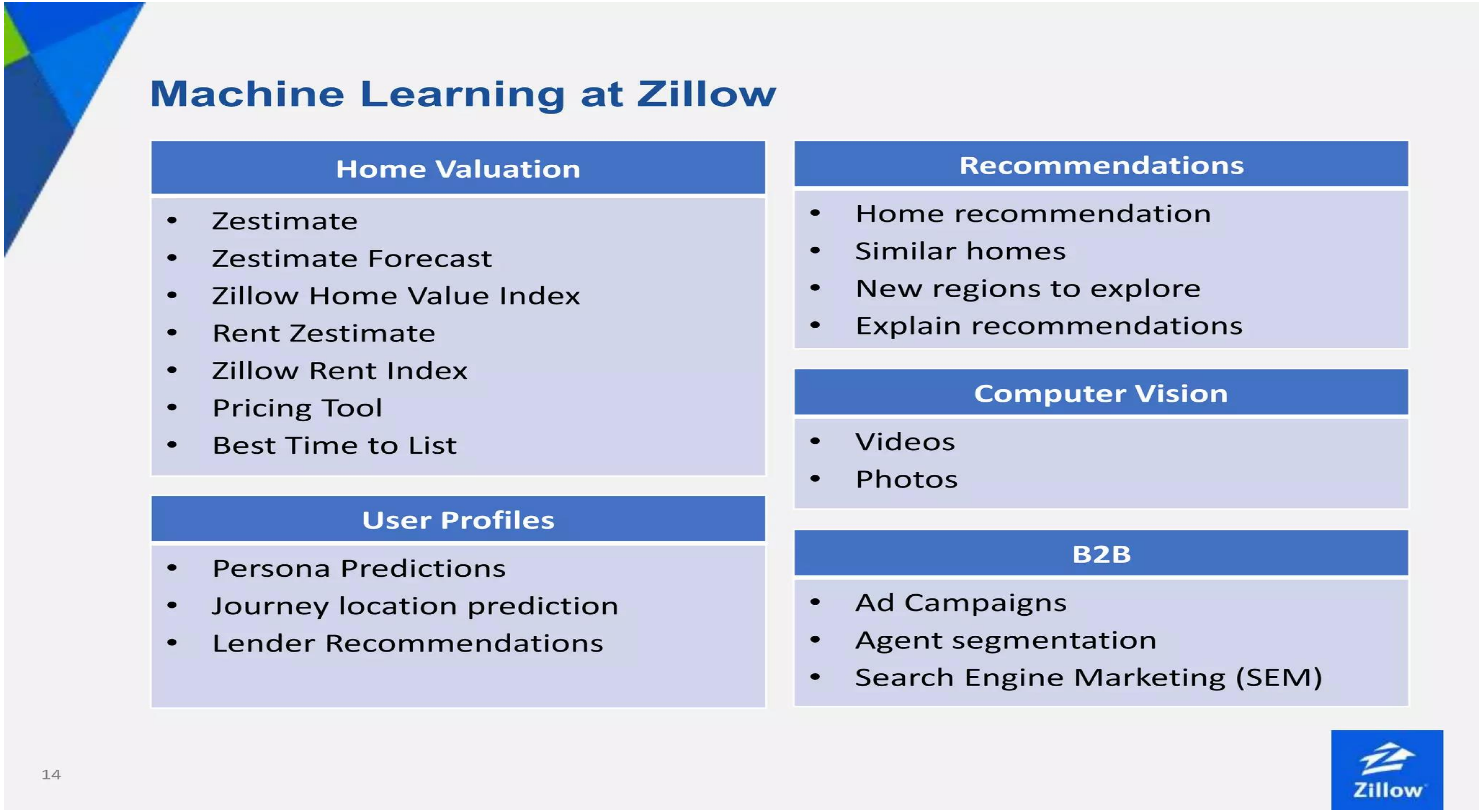
APARTMENT FOR RENT
$1,695/mo 1 bd · 1 ba · 605 sqft
· 711 N 35th St APT 103, Seattle, WA

27 photos
KOI
0 $1,470+ 1 $2,050+ 2 $2,640+
Updated today · 1139 NW Market St, Seattle, WA

13 photos
APARTMENT FOR RENT
$1,575/mo 1 bd · 1 ba · 625 sqft
1 hour ago · 6736 24th Ave NW APT 302, Seattle, WA

8 photos
10 photos

Data Science at Zillow



Machine Learning at Zillow

Home Valuation

- Zestimate
- Zestimate Forecast
- Zillow Home Value Index
- Rent Zestimate
- Zillow Rent Index
- Pricing Tool
- Best Time to List

User Profiles

- Persona Predictions
- Journey location prediction
- Lender Recommendations

Recommendations

- Home recommendation
- Similar homes
- New regions to explore
- Explain recommendations

Computer Vision

- Videos
- Photos

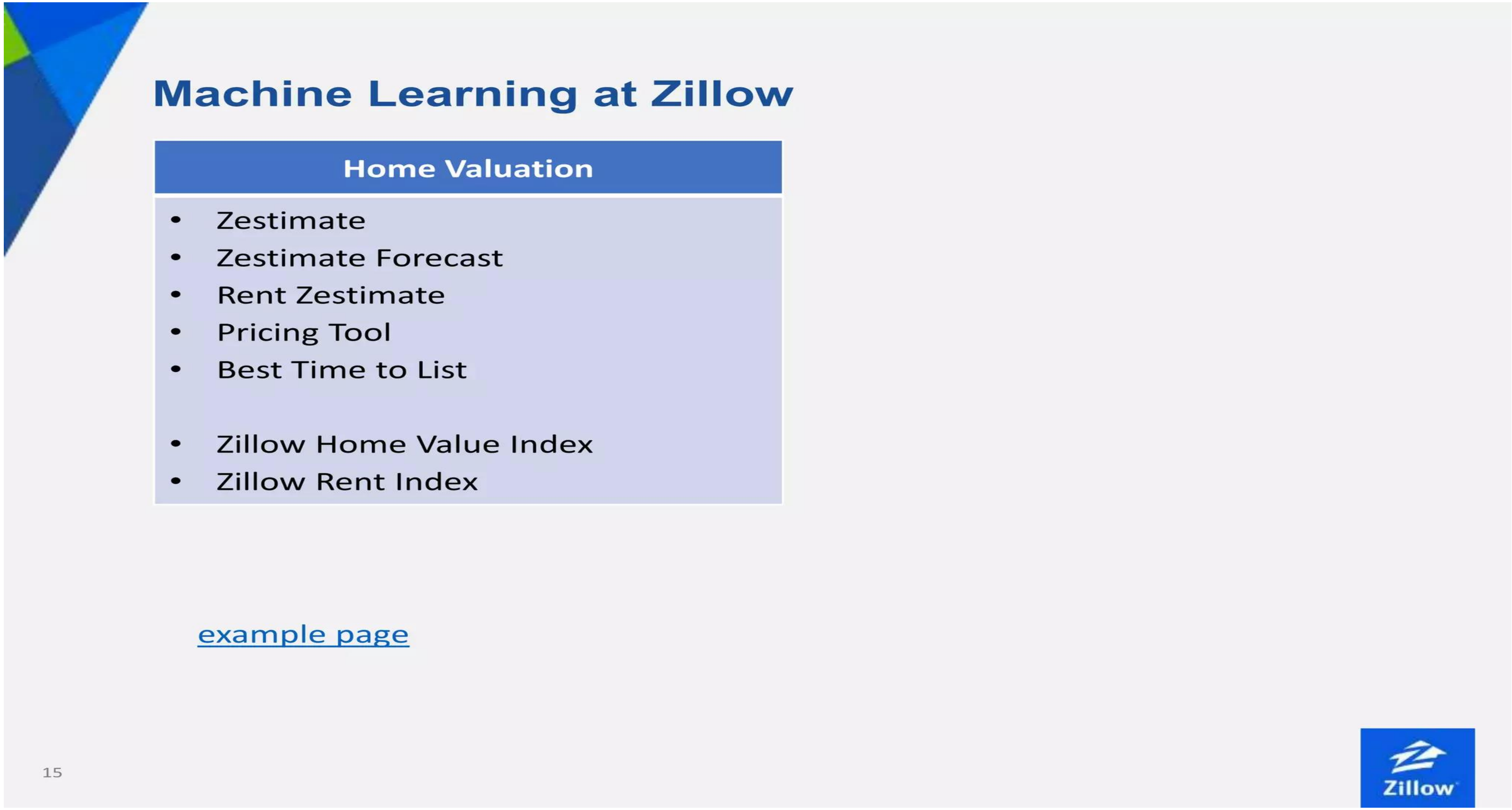
B2B

- Ad Campaigns
- Agent segmentation
- Search Engine Marketing (SEM)

14

Zillow

Data Science at Zillow



Machine Learning at Zillow

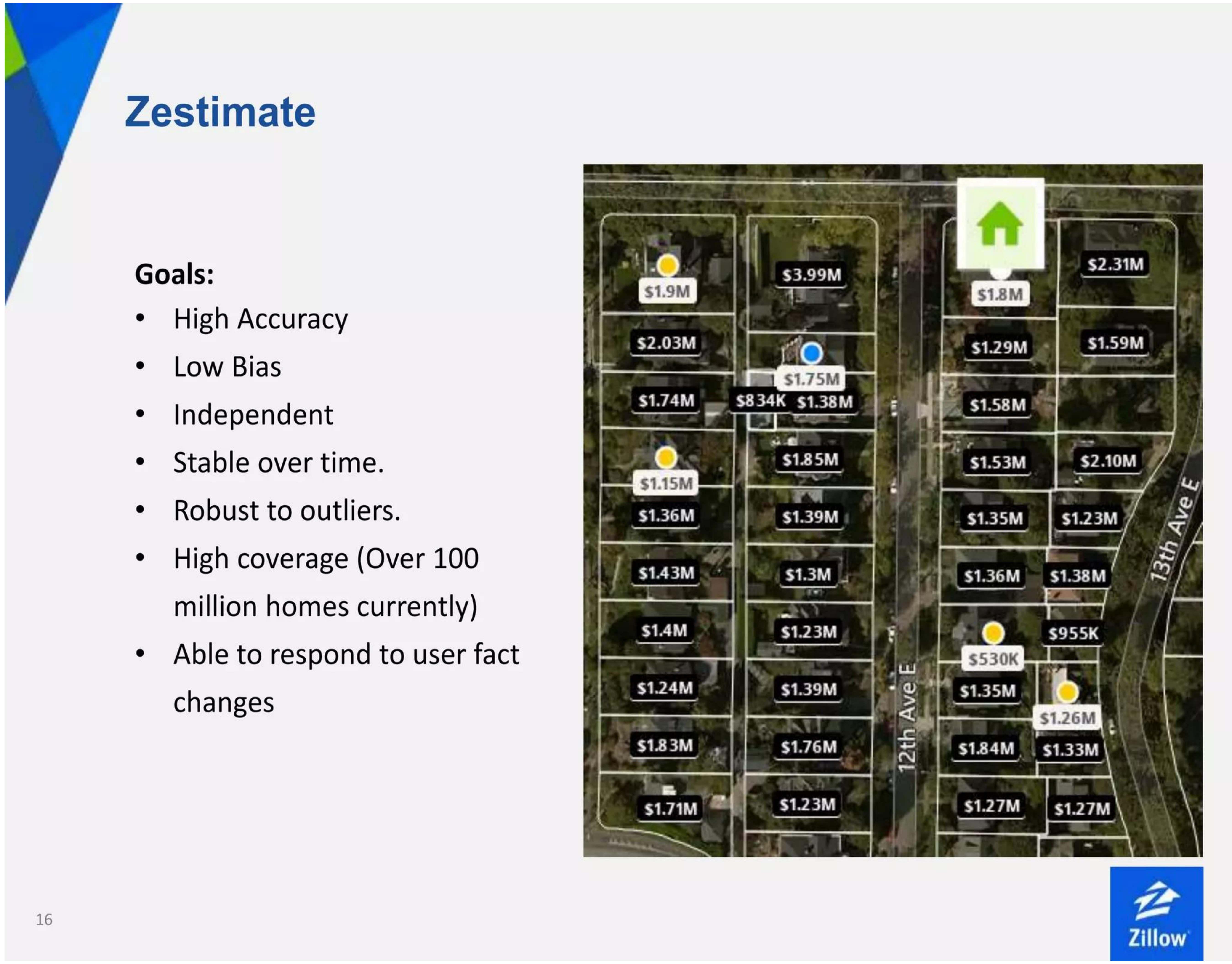
Home Valuation
• Zestimate • Zestimate Forecast • Rent Zestimate • Pricing Tool • Best Time to List • Zillow Home Value Index • Zillow Rent Index

example page

15

Data Science at Zillow

Zillow list of properties that are not on market and provide the estimates on those properties which they commonly referes to as Zestimate

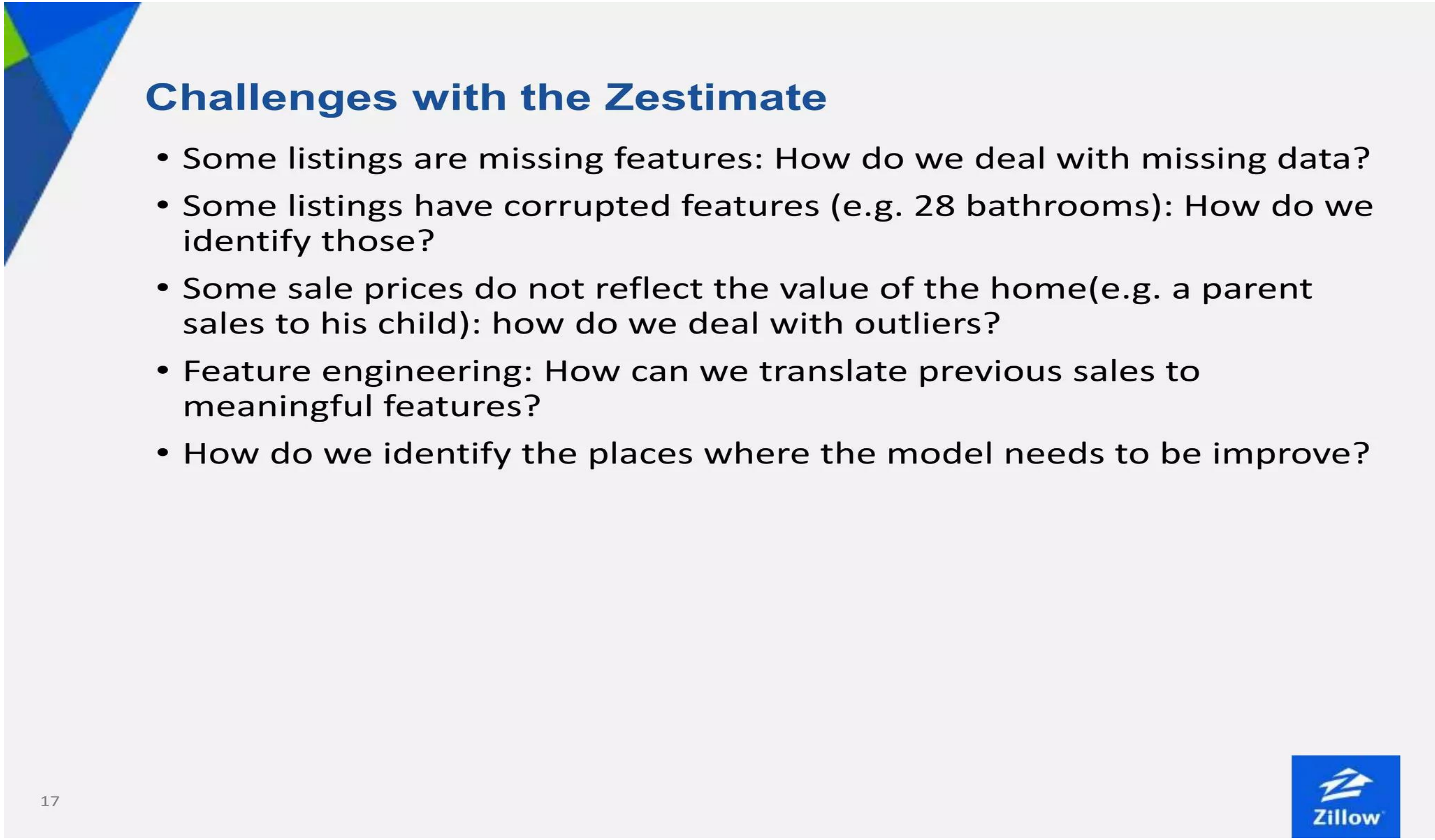


Zestimate

Goals:

- High Accuracy
- Low Bias
- Independent
- Stable over time.
- Robust to outliers.
- High coverage (Over 100 million homes currently)
- Able to respond to user fact changes

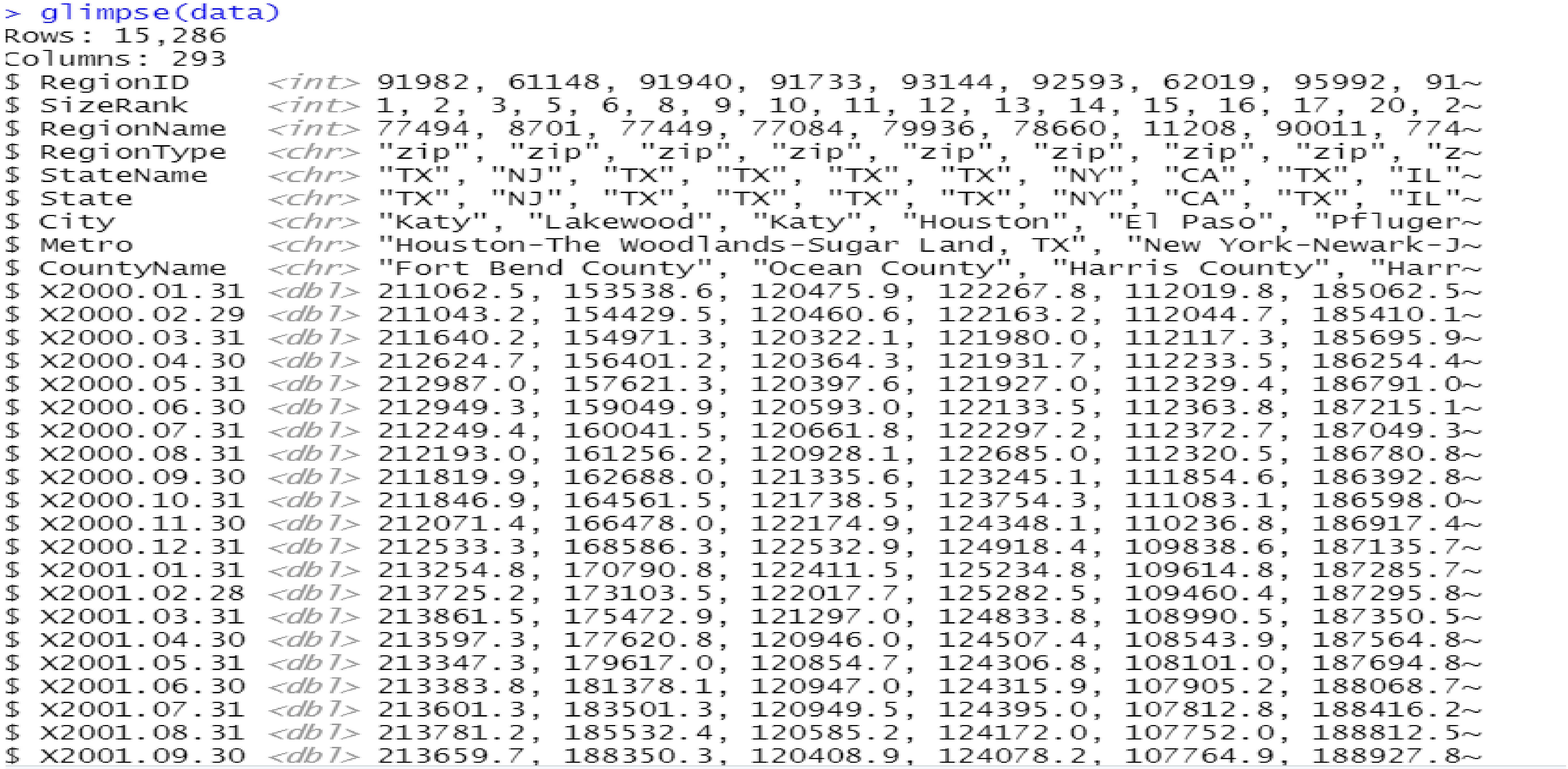
Data Science at Zillow



Challenges with the Zestimate

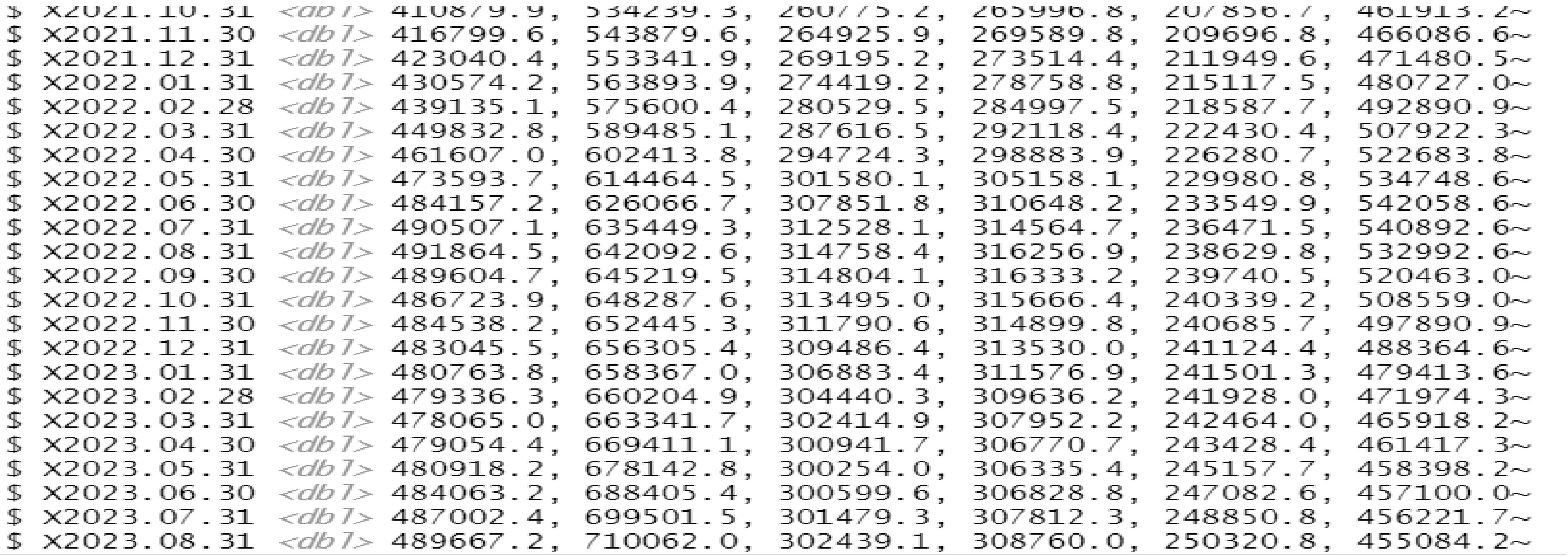
- Some listings are missing features: How do we deal with missing data?
- Some listings have corrupted features (e.g. 28 bathrooms): How do we identify those?
- Some sale prices do not reflect the value of the home(e.g. a parent sales to his child): how do we deal with outliers?
- Feature engineering: How can we translate previous sales to meaningful features?
- How do we identify the places where the model needs to be improve?

RAW DATA



```
> glimpse(data)
Rows: 15,286
Columns: 293
$ RegionID     <int> 91982, 61148, 91940, 91733, 93144, 92593, 62019, 95992, 91~
$ SizeRank     <int> 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 2~
$ RegionName   <int> 77494, 8701, 77449, 77084, 79936, 78660, 11208, 90011, 774~
$ RegionType   <chr> "zip", "zip", "zip", "zip", "zip", "zip", "zip", "zip", "z~
$ StateName    <chr> "TX", "NJ", "TX", "TX", "TX", "TX", "NY", "CA", "TX", "IL"~
$ State        <chr> "TX", "NJ", "TX", "TX", "TX", "TX", "NY", "CA", "TX", "IL"~
$ City         <chr> "Katy", "Lakewood", "Katy", "Houston", "El Paso", "Pfluger~
$ Metro        <chr> "Houston-The Woodlands-Sugar Land, TX", "New York-Newark-J~
$ CountyName   <chr> "Fort Bend County", "Ocean County", "Harris County", "Harr~
$ X2000.01.31 <dbl> 211062.5, 153538.6, 120475.9, 122267.8, 112019.8, 185062.5~
$ X2000.02.29 <dbl> 211043.2, 154429.5, 120460.6, 122163.2, 112044.7, 185410.1~
$ X2000.03.31 <dbl> 211640.2, 154971.3, 120322.1, 121980.0, 112117.3, 185695.9~
$ X2000.04.30 <dbl> 212624.7, 156401.2, 120364.3, 121931.7, 112233.5, 186254.4~
$ X2000.05.31 <dbl> 212987.0, 157621.3, 120397.6, 121927.0, 112329.4, 186791.0~
$ X2000.06.30 <dbl> 212949.3, 159049.9, 120593.0, 122133.5, 112363.8, 187215.1~
$ X2000.07.31 <dbl> 212249.4, 160041.5, 120661.8, 122297.2, 112372.7, 187049.3~
$ X2000.08.31 <dbl> 212193.0, 161256.2, 120928.1, 122685.0, 112320.5, 186780.8~
$ X2000.09.30 <dbl> 211819.9, 162688.0, 121335.6, 123245.1, 111854.6, 186392.8~
$ X2000.10.31 <dbl> 211846.9, 164561.5, 121738.5, 123754.3, 111083.1, 186598.0~
$ X2000.11.30 <dbl> 212071.4, 166478.0, 122174.9, 124348.1, 110236.8, 186917.4~
$ X2000.12.31 <dbl> 212533.3, 168586.3, 122532.9, 124918.4, 109838.6, 187135.7~
$ X2001.01.31 <dbl> 213254.8, 170790.8, 122411.5, 125234.8, 109614.8, 187285.7~
$ X2001.02.28 <dbl> 213725.2, 173103.5, 122017.7, 125282.5, 109460.4, 187295.8~
$ X2001.03.31 <dbl> 213861.5, 175472.9, 121297.0, 124833.8, 108990.5, 187350.5~
$ X2001.04.30 <dbl> 213597.3, 177620.8, 120946.0, 124507.4, 108543.9, 187564.8~
$ X2001.05.31 <dbl> 213347.3, 179617.0, 120854.7, 124306.8, 108101.0, 187694.8~
$ X2001.06.30 <dbl> 213383.8, 181378.1, 120947.0, 124315.9, 107905.2, 188068.7~
$ X2001.07.31 <dbl> 213601.3, 183501.3, 120949.5, 124395.0, 107812.8, 188416.2~
$ X2001.08.31 <dbl> 213781.2, 185532.4, 120585.2, 124172.0, 107752.0, 188812.5~
$ X2001.09.30 <dbl> 213659.7, 188350.3, 120408.9, 124078.2, 107764.9, 188927.8~
```

⋮



```
$ X2021.10.31 <dbl> 410879.9, 534239.3, 260775.2, 265996.8, 207856.7, 461913.2~
$ X2021.11.30 <dbl> 416799.6, 543879.6, 264925.9, 269589.8, 209696.8, 466086.6~
$ X2021.12.31 <dbl> 423040.4, 553341.9, 269195.2, 273514.4, 211949.6, 471480.5~
$ X2022.01.31 <dbl> 430574.2, 563893.9, 274419.2, 278758.8, 215117.5, 480727.0~
$ X2022.02.28 <dbl> 439135.1, 575600.4, 280529.5, 284997.5, 218587.7, 492890.9~
$ X2022.03.31 <dbl> 449832.8, 589485.1, 287616.5, 292118.4, 222430.4, 507922.3~
$ X2022.04.30 <dbl> 461607.0, 602413.8, 294724.3, 298883.9, 226280.7, 522683.8~
$ X2022.05.31 <dbl> 473593.7, 614464.5, 301580.1, 305158.1, 229980.8, 534748.6~
$ X2022.06.30 <dbl> 484157.2, 626066.7, 307851.8, 310648.2, 233549.9, 542058.6~
$ X2022.07.31 <dbl> 490507.1, 635449.3, 312528.1, 314564.7, 236471.5, 540892.6~
$ X2022.08.31 <dbl> 491864.5, 642092.6, 314758.4, 316256.9, 238629.8, 532992.6~
$ X2022.09.30 <dbl> 489604.7, 645219.5, 314804.1, 316333.2, 239740.5, 520463.0~
$ X2022.10.31 <dbl> 486723.9, 648287.6, 313495.0, 315666.4, 240339.2, 508559.0~
$ X2022.11.30 <dbl> 484538.2, 652445.3, 311790.6, 314899.8, 240685.7, 497890.9~
$ X2022.12.31 <dbl> 483045.5, 656305.4, 309486.4, 313530.0, 241124.4, 488364.6~
$ X2023.01.31 <dbl> 480763.8, 658367.0, 306883.4, 311576.9, 241501.3, 479413.6~
$ X2023.02.28 <dbl> 479336.3, 660204.9, 304440.3, 309636.2, 241928.0, 471974.3~
$ X2023.03.31 <dbl> 478065.0, 663341.7, 302414.9, 307952.2, 242464.0, 465918.2~
$ X2023.04.30 <dbl> 479054.4, 669411.1, 300941.7, 306770.7, 243428.4, 461417.3~
$ X2023.05.31 <dbl> 480918.2, 678142.8, 300254.0, 306335.4, 245157.7, 458398.2~
$ X2023.06.30 <dbl> 484063.2, 688405.4, 300599.6, 306828.8, 247082.6, 457100.0~
$ X2023.07.31 <dbl> 487002.4, 699501.5, 301479.3, 307812.3, 248850.8, 456221.7~
$ X2023.08.31 <dbl> 489667.2, 710062.0, 302439.1, 308760.0, 250320.8, 455084.2~
```

Thank you for your listening!

“One who never asks
Either knows everything or nothing”
Malcolm S. Forbes