Sociology Quant Camp

Introduction to R

Module 2: piping and tidyverse

Monica Alexander, Statistical Sciences and Sociology

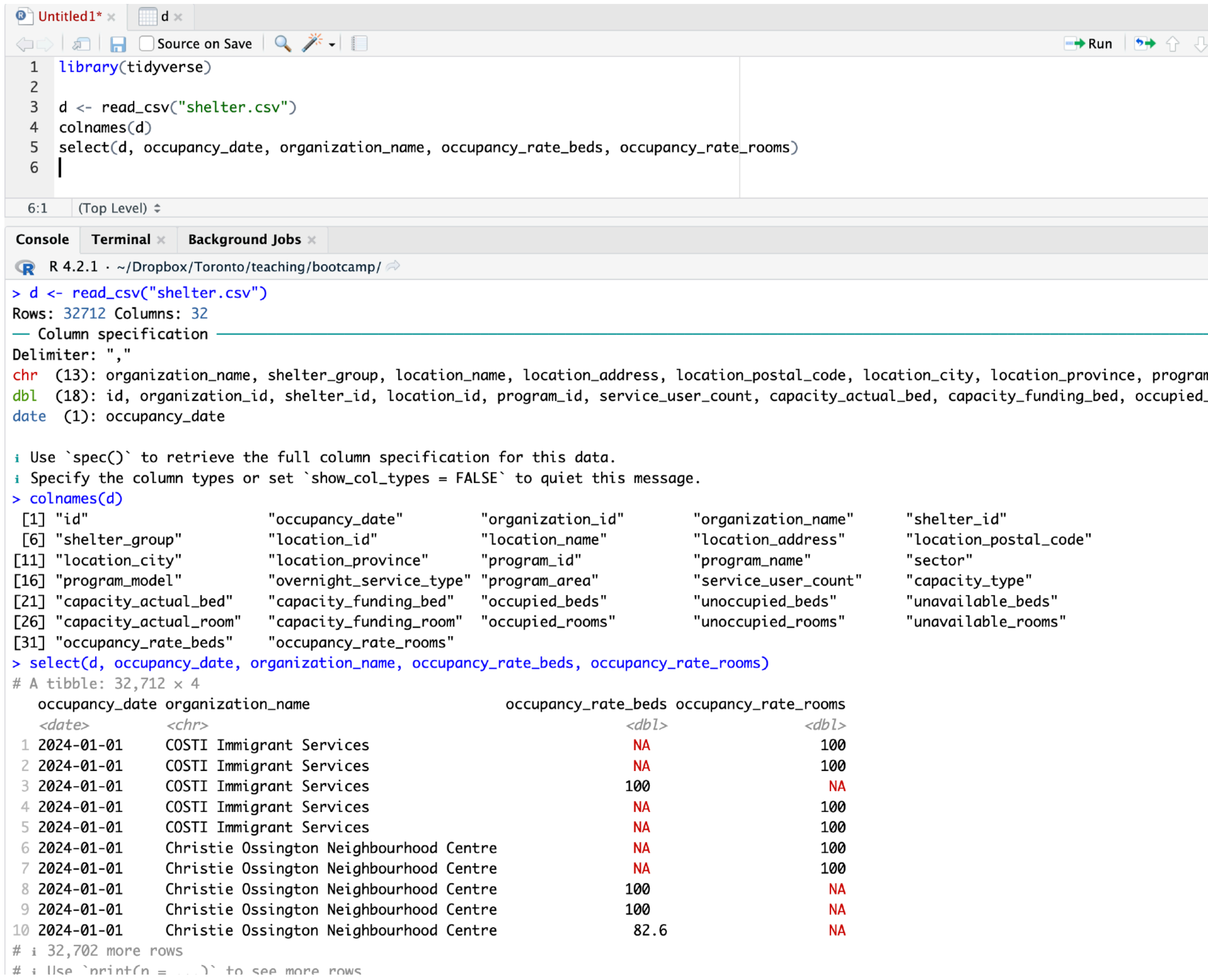
Using the tidyverse to manipulate real data

- In the previous module, we saw some functions and loaded in the `tidyverse` package
- Tidyverse has a range of functions that make it easier to manipulate real data
- Things like: adding columns, selecting columns, filtering out rows based on certain values…
- These functions have been specifically designed to work with datasets with lots of variables of different types

A first example

- Let's read in the shelter data and select some columns

- Note that `colnames()` is a useful function to see what the columns are called



```r
library(tidyverse)

d <- read_csv("shelter.csv")
colnames(d)
select(d, occupancy_date, organization_name, occupancy_rate_beds, occupancy_rate_rooms)
```

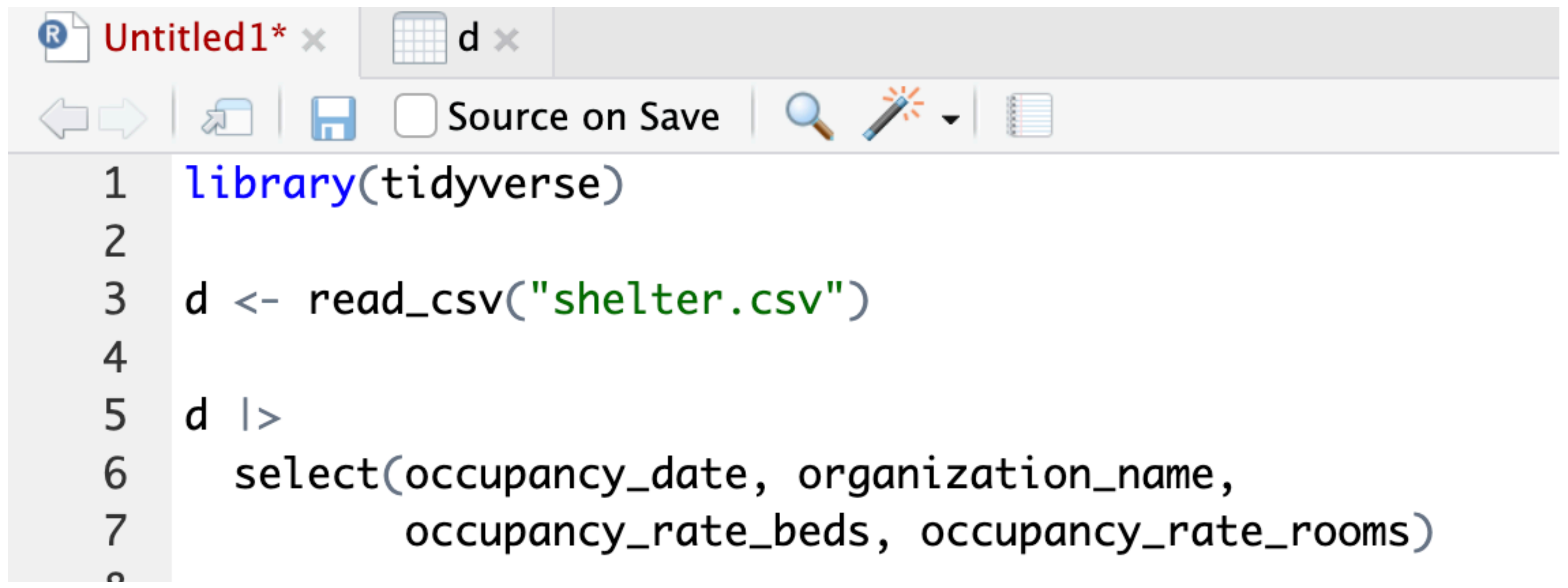
```
> d <- read_csv("shelter.csv")
Rows: 32712 Columns: 32
── Column specification ──────────────────────────────────────────────────────────
Delimiter: ","
chr  (13): organization_name, shelter_group, location_name, location_address, location_postal_code, location_city, location_province, progra
dbl  (18): id, organization_id, shelter_id, location_id, program_id, service_user_count, capacity_actual_bed, capacity_funding_bed, occupied_
date  (1): occupancy_date

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
> colnames(d)
 [1] "id"                   "occupancy_date"         "organization_id"      "organization_name"    "shelter_id"
 [6] "shelter_group"        "location_id"            "location_name"        "location_address"     "location_postal_code"
[11] "location_city"        "location_province"      "program_id"           "program_name"         "sector"
[16] "program_model"        "overnight_service_type" "program_area"         "service_user_count"   "capacity_type"
[21] "capacity_actual_bed"  "capacity_funding_bed"   "occupied_beds"        "unoccupied_beds"      "unavailable_beds"
[26] "capacity_actual_room" "capacity_funding_room"  "occupied_rooms"       "unoccupied_rooms"     "unavailable_rooms"
[31] "occupancy_rate_beds"  "occupancy_rate_rooms"
> select(d, occupancy_date, organization_name, occupancy_rate_beds, occupancy_rate_rooms)
# A tibble: 32,712 × 4
   occupancy_date organization_name                  occupancy_rate_beds occupancy_rate_rooms
   <date>         <chr>                                            <dbl>                <dbl>
 1 2024-01-01     COSTI Immigrant Services                            NA                  100
 2 2024-01-01     COSTI Immigrant Services                            NA                  100
 3 2024-01-01     COSTI Immigrant Services                           100                   NA
 4 2024-01-01     COSTI Immigrant Services                            NA                  100
 5 2024-01-01     COSTI Immigrant Services                            NA                  100
 6 2024-01-01     Christie Ossington Neighbourhood Centre             NA                  100
 7 2024-01-01     Christie Ossington Neighbourhood Centre             NA                  100
 8 2024-01-01     Christie Ossington Neighbourhood Centre            100                   NA
 9 2024-01-01     Christie Ossington Neighbourhood Centre            100                   NA
10 2024-01-01     Christie Ossington Neighbourhood Centre             82.6                 NA
# ℹ 32,702 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Demo: selecting columns

The pipe |>

- An alternative way of writing code
- Makes the code read more like a sentence
- Read the pipe as “and then”
- So here we are taking the data AND THEN selecting columns



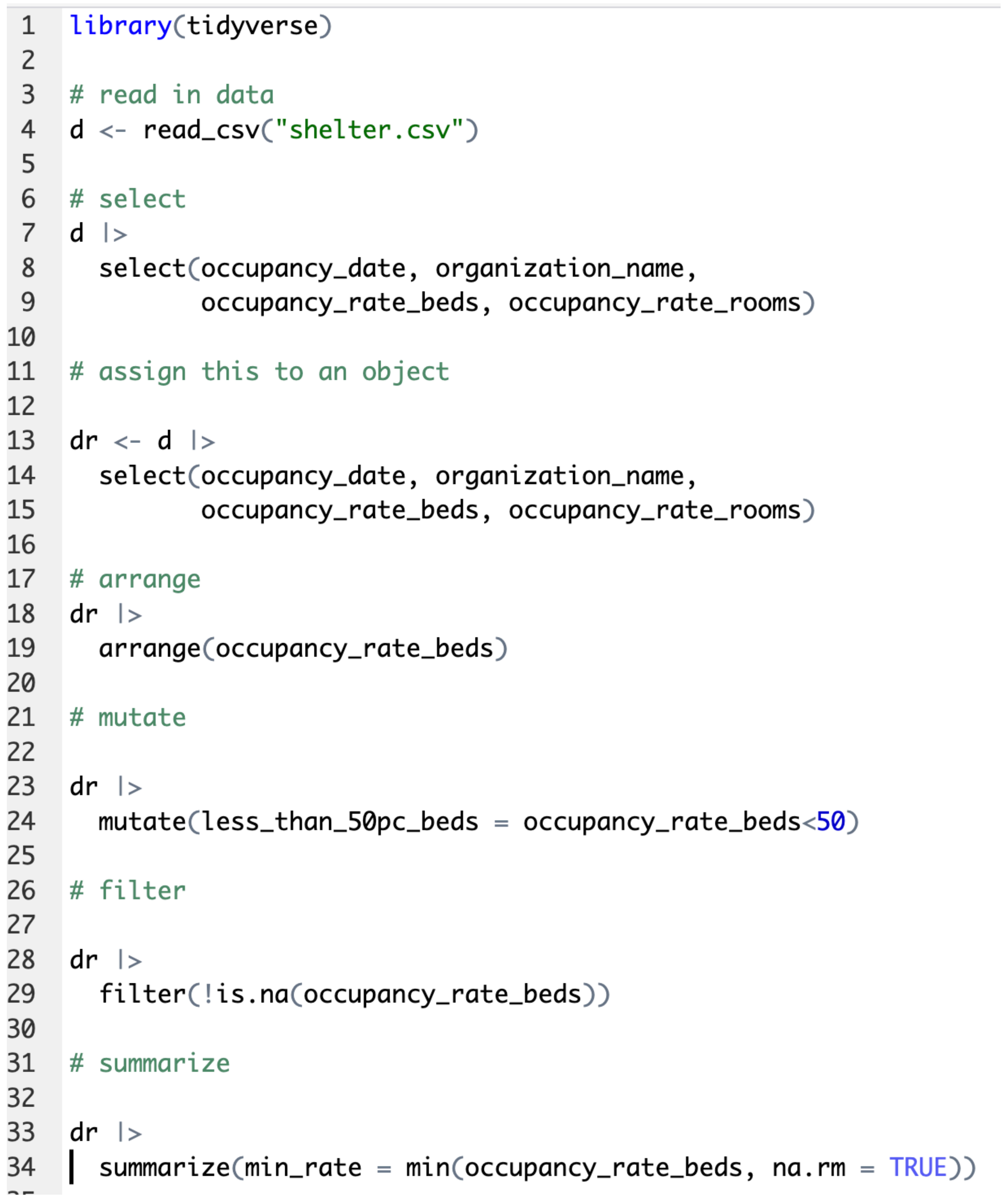
```r
library(tidyverse)

d <- read_csv("shelter.csv")

d |>
  select(occupancy_date, organization_name,
         occupancy_rate_beds, occupancy_rate_rooms)
```

Core tidyverse functions

- `select`: select columns
- `arrange`: sort/arrange by value
- `mutate`: make a new column
- `filter`: filter out certain rows
- `summarize`: produce summaries of data
- `group_by`: group the data by certain variable(s)



```r
library(tidyverse)

# read in data
d <- read_csv("shelter.csv")

# select
d |>
  select(occupancy_date, organization_name,
         occupancy_rate_beds, occupancy_rate_rooms)

# assign this to an object

dr <- d |>
  select(occupancy_date, organization_name,
         occupancy_rate_beds, occupancy_rate_rooms)

# arrange
dr |>
  arrange(occupancy_rate_beds)

# mutate

dr |>
  mutate(less_than_50pc_beds = occupancy_rate_beds<50)

# filter

dr |>
  filter(!is.na(occupancy_rate_beds))

# summarize

dr |>
  summarize(min_rate = min(occupancy_rate_beds, na.rm = TRUE))
```

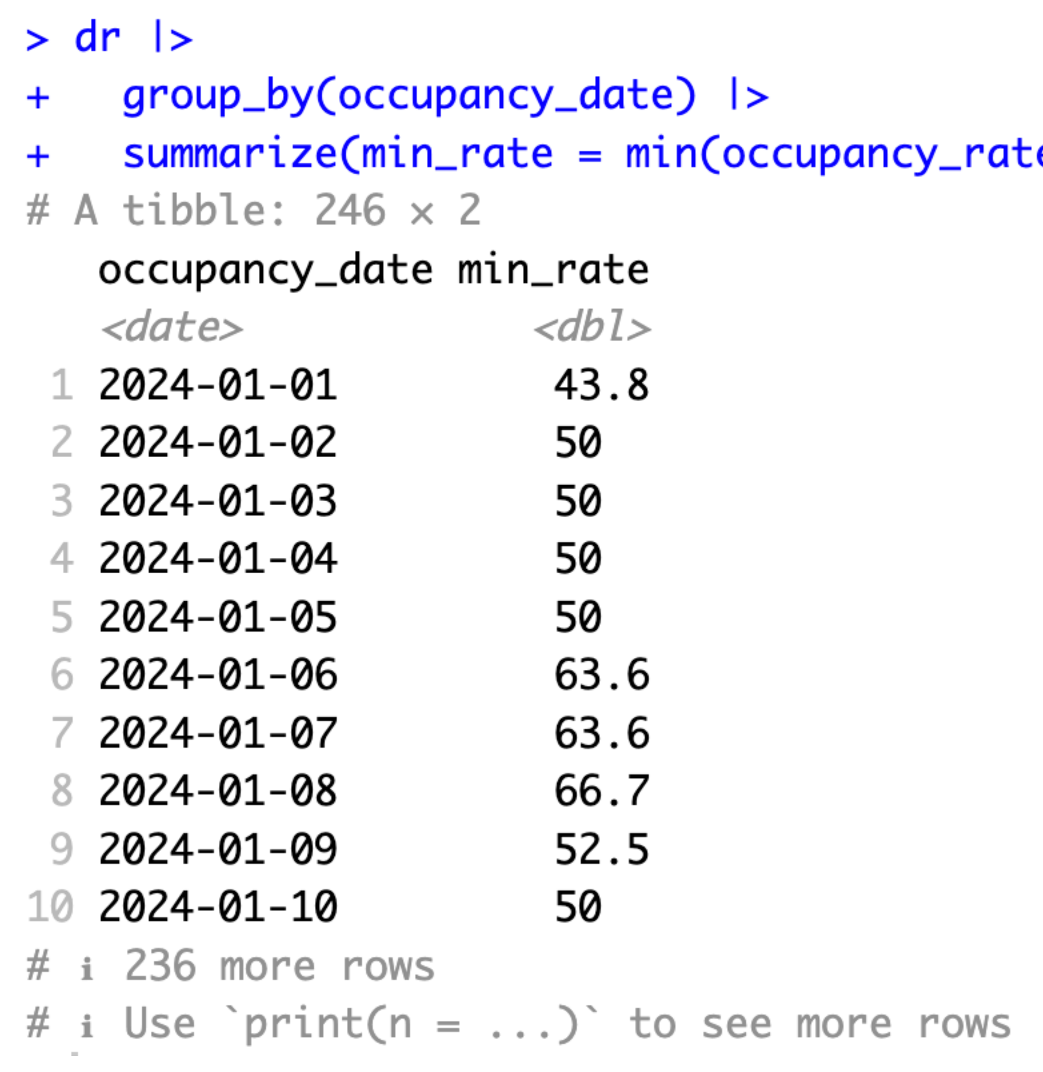
Demo: tidyverse functions

group_by

- The group_by function is extremely powerful when used in conjunction with summarize to get summaries by groups
- Note that we can thread together multiple pipes!

```
dr |>
  group_by(occupancy_date) |>
  summarize(min_rate = min(occupancy_rate_beds, na.rm = TRUE))
```

Here is the output:



```
> dr |>
+   group_by(occupancy_date) |>
+   summarize(min_rate = min(occupancy_rate_beds, na.rm = TRUE))
# A tibble: 246 × 2
   occupancy_date min_rate
   <date>            <dbl>
 1 2024-01-01         43.8
 2 2024-01-02         50
 3 2024-01-03         50
 4 2024-01-04         50
 5 2024-01-05         50
 6 2024-01-06         63.6
 7 2024-01-07         63.6
 8 2024-01-08         66.7
 9 2024-01-09         52.5
10 2024-01-10         50
# ℹ 236 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Demo: more complicated tidyverse functions

Where to get help

- Lots of good, free online sources
    - R for Data Science: https://www.tidyverse.org/learn/
    - Telling stories with data: https://tellingstorieswithdata.com/
    - Tidyverse skills for data science: https://jhudatascience.org/tidyversecourse/intro.html
- Google/Stack Overflow
- Email
- Practice, practice, practice; don’t be afraid of mistakes