Module 4: Data Manipulation

Review: Packages

- Packages are collections of functions and data sets developed by the community.
- Two steps to use a package
  - installed with the install.packages function (only once)
  - imported with the library function (once per session)

```
install.packages("package_name")
library(package_name)
```

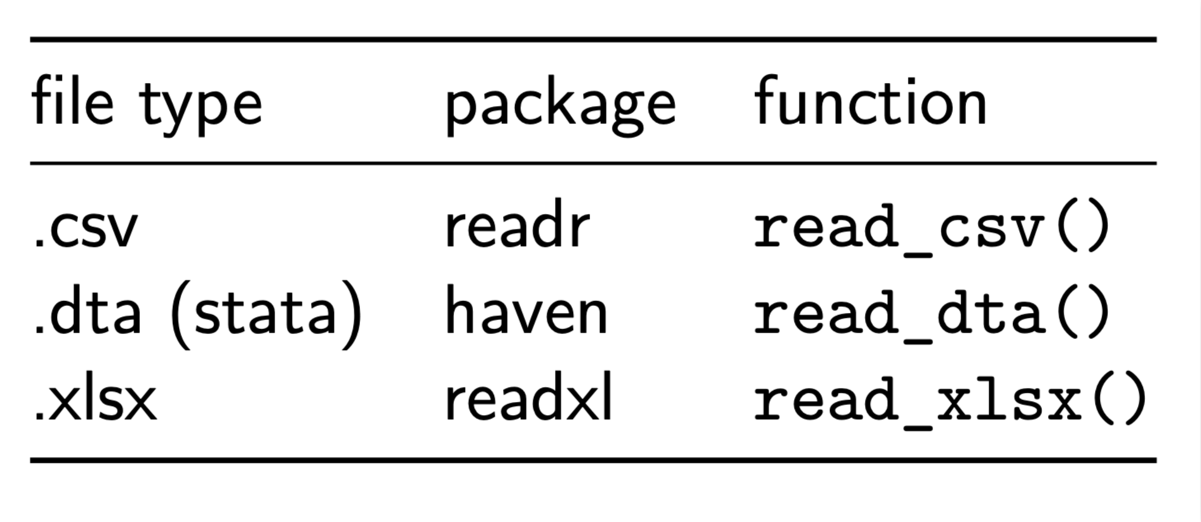
Let’s Load Tidyverse

- functions that allows us to read data into our RStudio environment, and;
- functions that allow us to manipulate our data.

```r
library(tidyverse)
```

dplyr
ggplot2
forcats
TIBBLE
readr
stringr
tidyr
purrr

Review: Reading in Data



file type	package	function
.csv	readr	`read_csv()`
.dta (stata)	haven	`read_dta()`
.xlsx	readxl	`read_xlsx()`

Loading Data from Files

1.

2.

3.

```r
getwd()
```

```
[1] "/Users/jacob/Downloads/Module 3"
```

```r
library(tidyverse)
library(haven)
housing_data =
read_dta("texas_housing_data.dta")
```

What Data do we have?

- head() and glimpse() provide ways to see part of your data.
- View() provides a more spreadsheet-like experience

```
head(housing_data)
```

```
# A tibble: 6 x 8
  city       year month sales   volume median
listings inventory
  <chr>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
<dbl>     <dbl>
1 Abilene  2000     1    72  5380000  71400
701       6.3
2 Abilene  2000     2    98  6505000  58700
746       6.6
3 Abilene  2000     3   130  9285000  58100
784       6.8
4 Abilene  2000     4    98  9730000  68600
785       6.9
```

```
5 Abilene  2000      5   141 10590000  67300
794       6.8
6 Abilene  2000      6   156 13910000  66900
780       6.6
```

Quick Glance

```
dim(housing_data)
```

```
[1] 8602    8
```

```
sapply(housing_data, median, na.rm=TRUE)
```

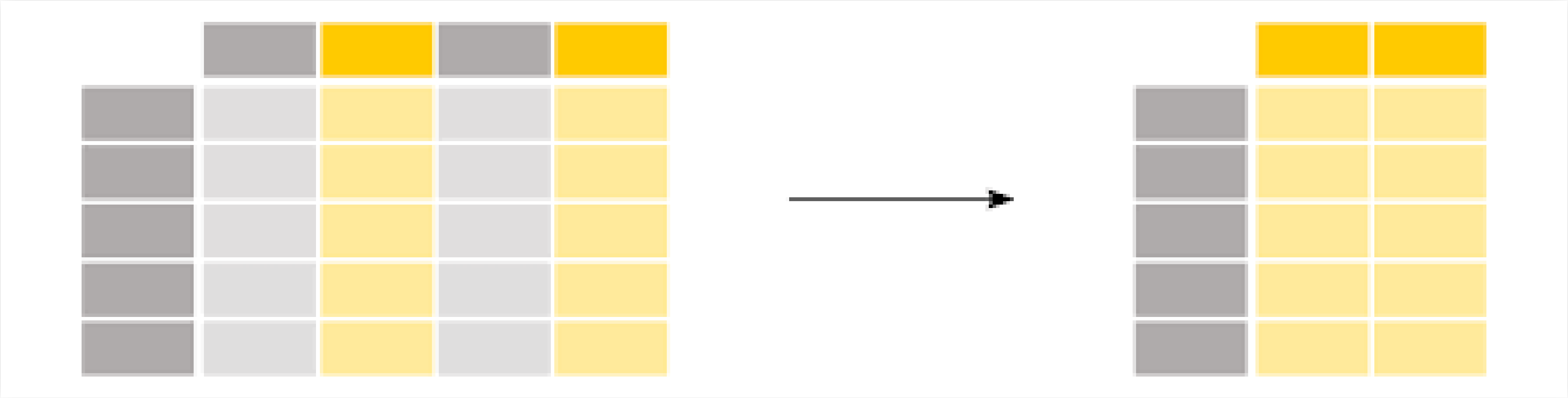
```
      city       year      month      sales
volume     median   listings
        NA     2007.0        6.0      169.0
22986824.0   123800.0     1283.0
```

```
inventory
      6.2
```

Data manipulation with dplyr

- select() to pick columns
- arrange() to order the data
- mutate() to create new columns
- filter() to get rows that meet a criteria

Selecting columns with select()



Selecting columns with select()

```
select(housing_data, city, sales, listings)
```

```
# A tibble: 8,602 x 3
   city    sales listings
   <chr>   <dbl>    <dbl>
 1 Abilene    72      701
 2 Abilene    98      746
 3 Abilene   130      784
 4 Abilene    98      785
 5 Abilene   141      794
 6 Abilene   156      780
 7 Abilene   152      742
 8 Abilene   131      765
 9 Abilene   104      771
10 Abilene   101      764
# … with 8,592 more rows
```

Selecting columns with select()

```
select(housing_data, -c(city, sales,
listings))
```

```
# A tibble: 8,602 x 5
    year month   volume median inventory
   <dbl> <dbl>    <dbl>  <dbl>     <dbl>
 1  2000     1  5380000  71400       6.3
 2  2000     2  6505000  58700       6.6
 3  2000     3  9285000  58100       6.8
 4  2000     4  9730000  68600       6.9
 5  2000     5 10590000  67300       6.8
 6  2000     6 13910000  66900       6.6
 7  2000     7 12635000  73500       6.2
 8  2000     8 10710000  75000       6.4
 9  2000     9  7615000  64500       6.5
```

```
10  2000    10  7040000  59300       6.6
# … with 8,592 more rows
```

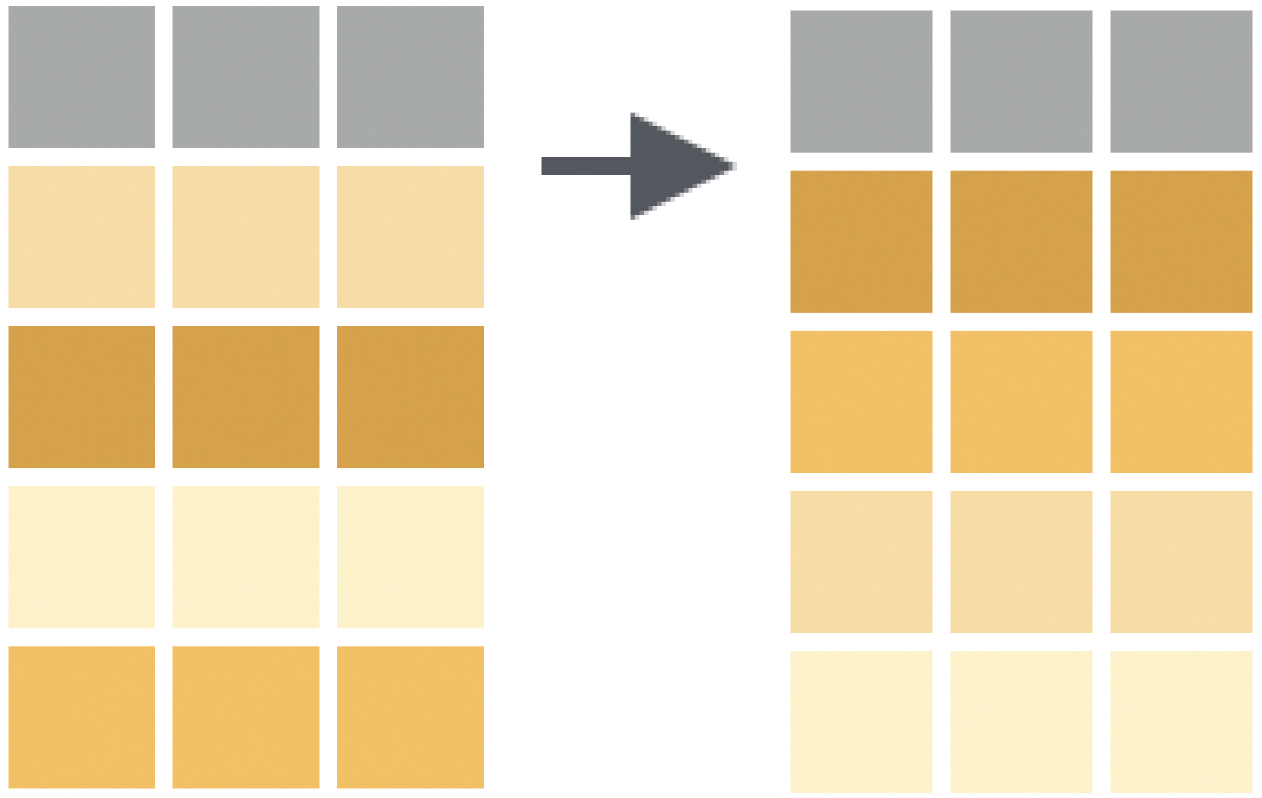
Selecting columns with select(), helpers

```
select(housing_data, city, sales, listings,
everything())
```

```
# A tibble: 8,602 x 8
   city     sales listings  year month    volume
median inventory
   <chr>    <dbl>    <dbl> <dbl> <dbl>     <dbl>
<dbl>     <dbl>
 1 Abilene     72      701  2000     1   5380000
71400       6.3
 2 Abilene     98      746  2000     2   6505000
58700       6.6
 3 Abilene    130      784  2000     3   9285000
58100       6.8
 4 Abilene     98      785  2000     4   9730000
68600       6.9
```

```
 5 Abilene   141       794  2000     5 10590000
67300       6.8
 6 Abilene   156       780  2000     6 13910000
66900       6.6
 7 Abilene   152       742  2000     7 12635000
73500       6.2
 8 Abilene   131       765  2000     8 10710000
75000       6.4
 9 Abilene   104       771  2000     9  7615000
64500       6.5
10 Abilene   101       764  2000    10  7040000
59300       6.6
# … with 8,592 more rows
```

Sort rows with arrange()



Sort rows with arrange()

```r
arrange(housing_data, year)
```

```
# A tibble: 8,602 x 8
   city     year month sales   volume median
listings inventory
   <chr>   <dbl> <dbl> <dbl>    <dbl>  <dbl>
<dbl>     <dbl>
 1 Abilene  2000     1    72  5380000  71400
701       6.3
 2 Abilene  2000     2    98  6505000  58700
746       6.6
 3 Abilene  2000     3   130  9285000  58100
784       6.8
 4 Abilene  2000     4    98  9730000  68600
785       6.9
 5 Abilene  2000     5   141 10590000  67300
```

```
794       6.8
 6 Abilene  2000     6   156 13910000  66900
780       6.6
 7 Abilene  2000     7   152 12635000  73500
742       6.2
 8 Abilene  2000     8   131 10710000  75000
765       6.4
 9 Abilene  2000     9   104  7615000  64500
771       6.5
10 Abilene  2000    10   101  7040000  59300
764       6.6
# … with 8,592 more rows
```

Sort rows with arrange()

```
arrange(housing_data, desc(year))
```

```
# A tibble: 8,602 x 8
   city         year month sales   volume median
listings inventory
   <chr>       <dbl> <dbl> <dbl>    <dbl>  <dbl>
<dbl>     <dbl>
 1 Abilene     2015     1   158 23486998 134100
801       4.4
 2 Abilene     2015     2   151 19834263 126500
767       4.1
 3 Abilene     2015     3   198 31869437 136800
821       4.4
 4 Abilene     2015     4   201 28301159 129600
891       4.7
 5 Abilene     2015     5   199 31385757 144700
```

```
919       4.8
 6 Abilene   2015     6   260 41396230 141500
965       5
 7 Abilene   2015     7   268 45845730 148700
986       5
 8 Amarillo  2015     1   204 33188726 138500
1120       4.3
 9 Amarillo  2015     2   188 34355428 149400
1084       4.2
10 Amarillo  2015     3   317 53603130 140900
1051       3.9
# … with 8,592 more rows
```

Introducing the pipe operator

%>%

Ceci est une %>%

- by default, the left-hand side is the first argument of the right-hand side function.

```
select(housing_data, city, year, sales,
volume)
```

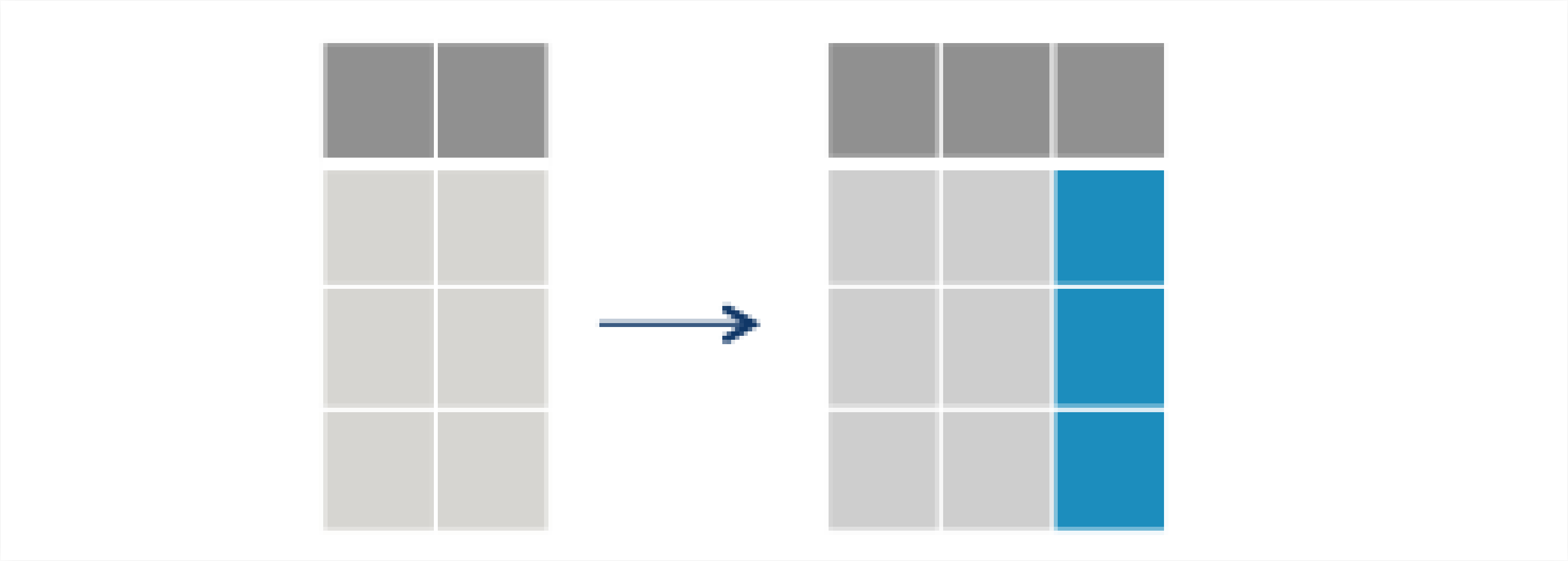
```
housing_data %>%
  select(city, year, sales, volume)
```

Ceci est une %>%

```
housing_data %>%
  select(city, year, sales, volume, median)
%>%
  arrange(desc(median))
```

```
# A tibble: 8,602 x 5
   city               year sales    volume median
   <chr>             <dbl> <dbl>     <dbl>  <dbl>
 1 Collin County      2015  1572 544545110 304200
 2 Collin County      2015  1789 614959441 300400
 3 Collin County      2015  1861 613669702 292600
 4 Collin County      2015  1391 456997967 291400
 5 Collin County      2015  1258 413242198 285800
 6 Fort Bend          2015  1341 429731131 284200
 7 Collin County  2015   938 300904769 283400
 8 Midland        2014   208  70836346 283100
 9 Fort Bend      2014  1388 437581291 282300
10 Fort Bend      2015  1372 431875327 280400
# … with 8,592 more rows
```

Creating columns with mutate()



Creating columns with mutate()

-
-

Creating columns with mutate()

```
housing_data %>%
  mutate(mean_price = volume / sales) %>%
  select(city, year, month, mean_price, sales,
volume)
```

```
# A tibble: 8,602 x 6
   city     year month mean_price sales
volume
   <chr>   <dbl> <dbl>      <dbl> <dbl>
<dbl>
 1 Abilene  2000     1     74722.    72
5380000
 2 Abilene  2000     2     66378.    98
6505000
 3 Abilene  2000     3     71423.   130
9285000
 4 Abilene  2000     4     99286.    98
9730000
 5 Abilene  2000     5     75106.   141
```

```
10590000
 6 Abilene  2000     6     89167.   156
13910000
 7 Abilene  2000     7     83125    152
12635000
 8 Abilene  2000     8     81756.   131
10710000
 9 Abilene  2000     9     73221.   104
7615000
10 Abilene  2000    10     69703.   101
7040000
# … with 8,592 more rows
```

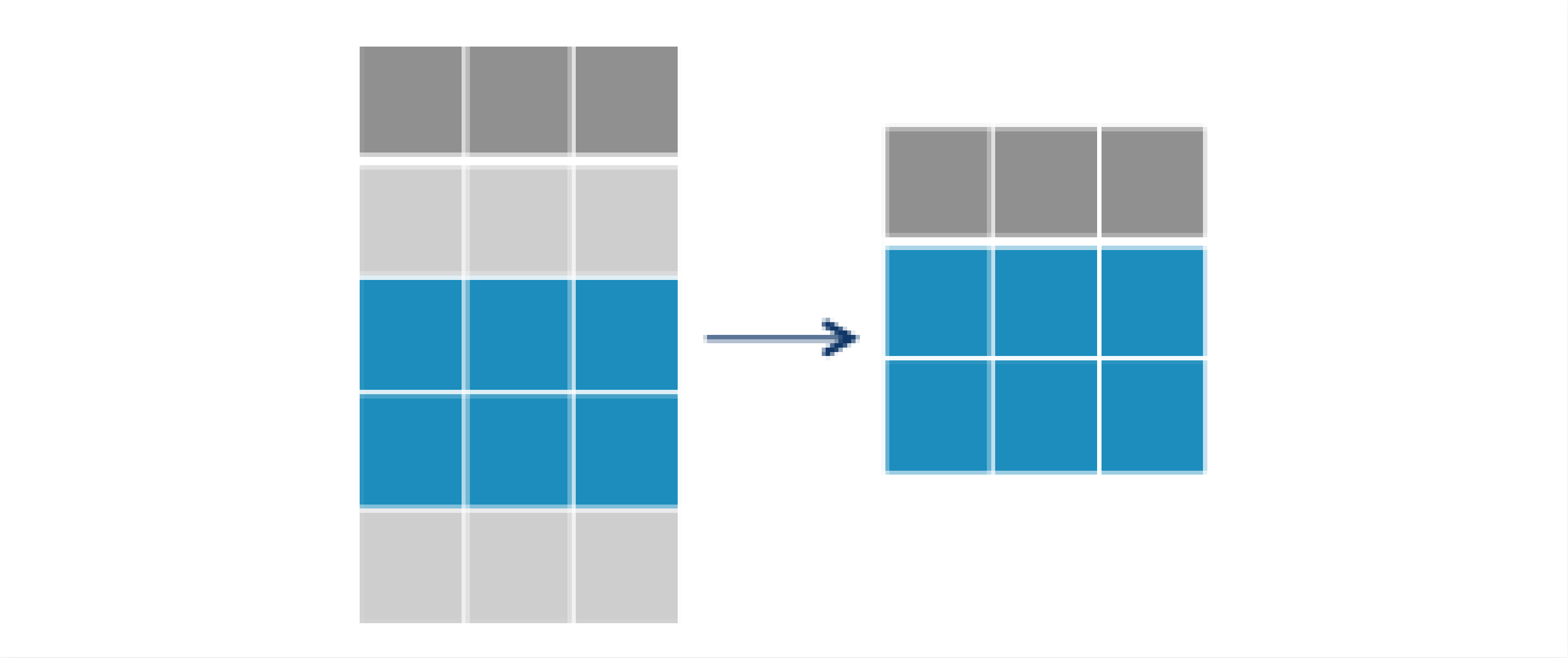
Creating columns with mutate()

```
housing_data %>%
  mutate(mean_price = volume / sales,
         sqrt_mean_price = sqrt(mean_price))
%>%
  select(city, year, month, mean_price, sales,
sqrt_mean_price)
```

```
# A tibble: 8,602 x 6
   city      year month mean_price sales
sqrt_mean_price
   <chr>    <dbl> <dbl>      <dbl> <dbl>
<dbl>
 1 Abilene  2000     1     74722.    72
273.
```

```
 2 Abilene  2000     2     66378.   98
258.
 3 Abilene  2000     3     71423.  130
267.
 4 Abilene  2000     4     99286.   98
315.
 5 Abilene  2000     5     75106.  141
274.
 6 Abilene  2000     6     89167.  156
299.
 7 Abilene  2000     7     83125   152
288.
 8 Abilene  2000     8     81756.  131
286.
 9 Abilene  2000     9     73221.  104
271.
10 Abilene  2000    10     69703.  101
264.
# … with 8,592 more rows
```

Choose rows that match a condition with filter()



Choose rows that match a condition with filter()

```
filter(housing_data, year == 2013)
```

```
# A tibble: 552 x 8
   city     year month sales   volume median
listings inventory
   <chr>   <dbl> <dbl> <dbl>    <dbl>  <dbl>
<dbl>     <dbl>
 1 Abilene  2013     1   114 15794494 125300
966        5.7
 2 Abilene  2013     2   140 16552641  94400
943        5.6
 3 Abilene  2013     3   164 19609711 102500
958        5.7
 4 Abilene  2013     4   213 27261796 113700
```

```
948       5.5
 5 Abilene  2013     5   225 31901380 130000
923       5.3
 6 Abilene  2013     6   209 29454125 127300
960       5.5
 7 Abilene  2013     7   218 32547446 140000
969       5.4
 8 Abilene  2013     8   236 30777727 120000
976       5.4
 9 Abilene  2013     9   195 26237106 127500
985       5.4
10 Abilene  2013    10   167 21781187 119000
993       5.5
# … with 542 more rows
```

```
housing_data %>%
  filter(year == 2013)
```

Choose rows that match a condition with filter()

```
housing_data %>%
  filter(year == 2013,
         city == "Houston")
```

```
# A tibble: 12 x 8
   city      year month sales     volume median
listings inventory
   <chr>    <dbl> <dbl> <dbl>      <dbl>  <dbl>
<dbl>     <dbl>
 1 Houston  2013     1  4273  852045057 149500
21364       3.7
 2 Houston  2013     2  4886 1060985674 161900
21293       3.6
```

```
 3 Houston  2013     3  6382 1479273481 172300
20909       3.5
 4 Houston  2013     4  7116 1770746764 182400
20607       3.4
 5 Houston  2013     5  8439 2121508529 186100
20526       3.3
 6 Houston  2013     6  7935 2073909387 191600
21008       3.3
 7 Houston  2013     7  8468 2168720825 187800
21497       3.3
 8 Houston  2013     8  8155 2083377894 186700
21366       3.3
 9 Houston  2013     9  6706 1638923780 180200
21207       3.2
10 Houston  2013    10  6551 1544551772 176000
20508       3.1
11 Houston  2013    11  5557 1356418081 181400
19331       2.9
12 Houston  2013    12  6380 1658872245 187500
17857       2.7
```

Choose rows that match a condition with filter()

```
housing_data %>%
  filter(year > 2013,
         city == "Houston" | city == "Austin")
```

```
# A tibble: 38 x 8
   city     year month sales     volume median
listings inventory
   <chr>   <dbl> <dbl> <dbl>      <dbl>  <dbl>
<dbl>     <dbl>
 1 Austin   2014     1  1582  426127544 213700
5118        2
 2 Austin   2014     2  1903  550882376 229400
5255        2.1
 3 Austin   2014     3  2434  717821612 235600
```

```
5512       2.2
 4 Austin  2014     4  2691  813253968 237000
5838       2.3
 5 Austin  2014     5  3178 1012123948 243900
6539       2.6
 6 Austin  2014     6  3195 1023051880 248900
7040       2.7
 7 Austin  2014     7  3151  982086356 246900
7475       2.9
 8 Austin  2014     8  3023  927019222 243800
7326       2.9
 9 Austin  2014     9  2664  813797562 238900
7072       2.8
10 Austin  2014    10  2588  796863816 239600
6791       2.7
# … with 28 more rows
```

Recap: manipulating data with dplyr

-
  - select() to pick columns
  - arrange() to order the data
  - mutate() to create new columns
  - filter() to get rows that meet a criteria
-
-