week 3

Melanie Beebe

Reading Delimited Data

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
#alt, use readr instead of tidyverse
#can read csv from online link
bike_details <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/bikeDetails.csv")
```

```
Rows: 1061 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): name, seller_type, owner
dbl (4): selling_price, year, km_driven, ex_showroom_price

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.
```

```
bike_details
```

```
# A tibble: 1,061 x 7
   name        selling_price  year seller_type owner km_driven ex_showroom_price
   <chr>               <dbl> <dbl> <chr>       <chr>     <dbl>              <dbl>
 1 Royal Enfi~        175000  2019 Individual  1st ~       350                 NA
 2 Honda Dio           45000  2017 Individual  1st ~      5650                 NA
 3 Royal Enfi~        150000  2018 Individual  1st ~     12000             148114
 4 Yamaha Faz~         65000  2015 Individual  1st ~     23000              89643
 5 Yamaha SZ ~         20000  2011 Individual  2nd ~     21000                 NA
 6 Honda CB T~         18000  2010 Individual  1st ~     60000              53857
 7 Honda CB H~         78500  2018 Individual  1st ~     17000              87719
 8 Royal Enfi~        180000  2008 Individual  2nd ~     39000                 NA
 9 Hero Honda~         30000  2010 Individual  1st ~     32000                 NA
10 Bajaj Disc~         50000  2016 Individual  1st ~     42000              60122
# i 1,051 more rows
```

tibbles do not coerce to a vector when you subset one column using [], need to instead call as dataframe or use $

```
as.data.frame(bike_details)[1:10 ,1]
```

```
 [1] "Royal Enfield Classic 350"
 [2] "Honda Dio"
 [3] "Royal Enfield Classic Gunmetal Grey"
 [4] "Yamaha Fazer FI V 2.0 [2016-2018]"
 [5] "Yamaha SZ [2013-2014]"
 [6] "Honda CB Twister"
 [7] "Honda CB Hornet 160R"
 [8] "Royal Enfield Bullet 350 [2007-2011]"
 [9] "Hero Honda CBZ extreme"
[10] "Bajaj Discover 125"
```

```
#OR
bike_details$name[1:10]
```

```
 [1] "Royal Enfield Classic 350"
 [2] "Honda Dio"
 [3] "Royal Enfield Classic Gunmetal Grey"
 [4] "Yamaha Fazer FI V 2.0 [2016-2018]"
```

```
 [5] "Yamaha SZ [2013-2014]"
 [6] "Honda CB Twister"
 [7] "Honda CB Hornet 160R"
 [8] "Royal Enfield Bullet 350 [2007-2011]"
 [9] "Hero Honda CBZ extreme"
[10] "Bajaj Discover 125"
```

```
#better
bike_details[1:10, ] |>
  pull(name)
```

```
 [1] "Royal Enfield Classic 350"
 [2] "Honda Dio"
 [3] "Royal Enfield Classic Gunmetal Grey"
 [4] "Yamaha Fazer FI V 2.0 [2016-2018]"
 [5] "Yamaha SZ [2013-2014]"
 [6] "Honda CB Twister"
 [7] "Honda CB Hornet 160R"
 [8] "Royal Enfield Bullet 350 [2007-2011]"
 [9] "Hero Honda CBZ extreme"
[10] "Bajaj Discover 125"
```

another csv example

```
library(readr)
air_quality_data <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/AirQuality.csv")
```

```
New names:
Rows: 9471 Columns: 18
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(2): Date, Time dbl (14): ...1, CO(GT), PT08.S1(CO), NMHC(GT), C6H6(GT),
PT08.S2(NMHC), NOx(... lgl (2): ...17, ...18
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.
* `` -> `...1`
* `...16` -> `...17`
* `...17` -> `...18`
```

```
air_quality_data
```

```
# A tibble: 9,471 x 18
    ...1 Date       Time     `CO(GT)` `PT08.S1(CO)` `NMHC(GT)` `C6H6(GT)`
   <dbl> <chr>      <chr>       <dbl>         <dbl>      <dbl>      <dbl>
 1     1 10/03/2004 18.00.00      2.6          1360        150       11.9
 2     2 10/03/2004 19.00.00      2            1292        112        9.4
 3     3 10/03/2004 20.00.00      2.2          1402         88        9
 4     4 10/03/2004 21.00.00      2.2          1376         80        9.2
 5     5 10/03/2004 22.00.00      1.6          1272         51        6.5
 6     6 10/03/2004 23.00.00      1.2          1197         38        4.7
 7     7 11/03/2004 00.00.00      1.2          1185         31        3.6
 8     8 11/03/2004 01.00.00      1            1136         31        3.3
 9     9 11/03/2004 02.00.00      0.9          1094         24        2.3
10    10 11/03/2004 03.00.00      0.6          1010         19        1.7
# i 9,461 more rows
# i 11 more variables: `PT08.S2(NMHC)` <dbl>, `NOx(GT)` <dbl>,
#   `PT08.S3(NOx)` <dbl>, `NO2(GT)` <dbl>, `PT08.S4(NO2)` <dbl>,
#   `PT08.S5(O3)` <dbl>, T <dbl>, RH <dbl>, AH <dbl>, ...17 <lgl>, ...18 <lgl>
```

```
#not columns have '' because not standard
air_quality_data$`CO(GT)`[1:10]
```

```
 [1] 2.6 2.0 2.2 2.2 1.6 1.2 1.2 1.0 0.9 0.6
```

reading in a fixed width field

```
library(readr)
#fixed width, columns lined up and only spaces used, no tabs
#read_fwf("https://www4.stat.ncsu.edu/~online/datasets/cigarettes.txt")
#Error: `file` must be a regular file, not a connection
#error unexpected, some change in code
#solution
#look at original file and copy first lint to determine widths
#Alpine           14.1 0.86     0.9853 13.6
#1-17, 18-22, 23-31, 32-38, 39-42,
#widths are 17, 5, 8, 6, 3  paste in fwf_widths
#first row is not data so want to skip the first row
read_fwf("https://www4.stat.ncsu.edu/~online/datasets/cigarettes.txt",
```

```
        fwf_widths(c(17, 5, 9, 7, 4), c("brand", "tar", "nicotine", "weight",
                                        "co")), skip = 1)
```

```
Rows: 23 Columns: 5
-- Column specification --------------------------------------------------------

chr (1): brand
dbl (4): tar, nicotine, weight, co

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.
```

```
# A tibble: 23 x 5
   brand          tar nicotine weight    co
   <chr>        <dbl>    <dbl>  <dbl> <dbl>
 1 Alpine        14.1     0.86  0.985  13.6
 2 Benson        16       1.06  1.09   16.6
 3 CamelLights    8       0.67  0.928  10.2
 4 Carlton        4.1     0.4   0.946   5.4
 5 Chesterfield  15       1.04  0.888  15
 6 GoldenLights   8.8     0.76  1.03    9
 7 Kent          12.4     0.95  0.922  12.3
 8 Kool          16.6     1.12  0.937  16.3
 9 L&M           14.9     1.02  0.886  15.4
10 LarkLights    13.7     1.01  0.964  13
# i 13 more rows
```

```
#use above step to check column types for data validation
```

delimited raw data with character delimiter and no column names

```
#in function read_delim default column names is TRUE so add names
ump_data <- read_delim("https://www4.stat.ncsu.edu/~online/datasets/umps2012.txt",
                       delim = ">",
                       col_names = c("Year", "Month", "Day", "Home", "Away", "HPUmpire")
)
```

```
Rows: 2359 Columns: 6
-- Column specification --------------------------------------------------------
```

```
Delimiter: ">"
chr (3): Home, Away, HPUmpire
dbl (3): Year, Month, Day

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.
```

```
ump_data
```

```
# A tibble: 2,359 x 6
    Year Month   Day Home  Away  HPUmpire
   <dbl> <dbl> <dbl> <chr> <chr> <chr>
 1  2012     4    12 MIN   LAA   D.J. Reyburn
 2  2012     4    12 SD    ARI   Marty Foster
 3  2012     4    12 WSH   CIN   Mike Everitt
 4  2012     4    12 PHI   MIA   Jeff Nelson
 5  2012     4    12 CHC   MIL   Fieldin Culbreth
 6  2012     4    12 LAD   PIT   Wally Bell
 7  2012     4    12 TEX   SEA   Doug Eddings
 8  2012     4    12 COL   SF    Ron Kulpa
 9  2012     4    12 DET   TB    Mark Carlson
10  2012     4    13 NYY   LAA   Mike DiMuro
# i 2,349 more rows
```

data comments

first 3 columns are stored as numeric (dbl) but correspond to a date, let's fix this

```
library(lubridate)
#create column to store date (ymd is year month day which is data format)
ump_data$date <- ymd("2012-01-01")
head(ump_data)
```

```
# A tibble: 6 x 7
   Year Month   Day Home  Away  HPUmpire         date
  <dbl> <dbl> <dbl> <chr> <chr> <chr>            <date>
1  2012     4    12 MIN   LAA   D.J. Reyburn     2012-01-01
2  2012     4    12 SD    ARI   Marty Foster     2012-01-01
3  2012     4    12 WSH   CIN   Mike Everitt     2012-01-01
4  2012     4    12 PHI   MIA   Jeff Nelson      2012-01-01
5  2012     4    12 CHC   MIL   Fieldin Culbreth 2012-01-01
6  2012     4    12 LAD   PIT   Wally Bell       2012-01-01
```

```
for (i in 1:nrow(ump_data)){
  ump_data$date[i] <- ymd(paste(ump_data$Year[i], ump_data$Month[i], ump_data$Day[i], sep = '
}
head(ump_data)
```

```
# A tibble: 6 x 7
   Year Month   Day Home  Away  HPUmpire         date
  <dbl> <dbl> <dbl> <chr> <chr> <chr>            <date>
1  2012     4    12 MIN   LAA   D.J. Reyburn     2012-04-12
2  2012     4    12 SD    ARI   Marty Foster     2012-04-12
3  2012     4    12 WSH   CIN   Mike Everitt     2012-04-12
4  2012     4    12 PHI   MIA   Jeff Nelson      2012-04-12
5  2012     4    12 CHC   MIL   Fieldin Culbreth 2012-04-12
6  2012     4    12 LAD   PIT   Wally Bell       2012-04-12
```

Text files

use read_file() or read_lines()

Reading Excel Data

```
library(readxl)
dry_bean_data <- read_excel("Dry_Bean_Dataset.xlsx")
dry_bean_data
```

```
# A tibble: 13,611 x 17
    Area Perimeter MajorAxisLength MinorAxisLength AspectRatio Eccentricity
   <dbl>     <dbl>           <dbl>           <dbl>       <dbl>        <dbl>
 1 28395      610.            208.            174.        1.20        0.550
 2 28734      638.            201.            183.        1.10        0.412
 3 29380      624.            213.            176.        1.21        0.563
 4 30008      646.            211.            183.        1.15        0.499
 5 30140      620.            202.            190.        1.06        0.334
 6 30279      635.            213.            182.        1.17        0.520
 7 30477      670.            211.            184.        1.15        0.489
 8 30519      630.            213.            183.        1.17        0.514
 9 30685      636.            214.            183.        1.17        0.514
10 30834      632.            217.            181.        1.20        0.554
```

```
# i 13,601 more rows
# i 11 more variables: ConvexArea <dbl>, EquivDiameter <dbl>, Extent <dbl>,
#   Solidity <dbl>, Roundness <dbl>, Compactness <dbl>, ShapeFactor1 <dbl>,
#   ShapeFactor2 <dbl>, ShapeFactor3 <dbl>, ShapeFactor4 <dbl>, Class <chr>
```

Reading From a Particular Sheet

```
#We can pull in data from a specific sheet with the name or via integers
#(or NULL for 1st)
citation_dry_bean_data <- read_excel("Dry_Bean_Dataset.xlsx",
                             sheet = excel_sheets("Dry_Bean_Dataset.xlsx")[2])
citation_dry_bean_data
```

```
# A tibble: 0 x 1
# i 1 variable:
#   Citation Request :
KOKLU, M. and OZKAN, I.A., (2020), "Multiclass Classification of Dry Beans Using Computer Vis
```

Notice that didn't read in correctly! There is only one entry there (the 1st cell, 1st column) and it is currently being treated as the column name. Similar to the read_csv() function we can use col_names = FALSE here (thanks coherent ecosystem!!).

```
citation_dry_bean_data <- read_excel("Dry_Bean_Dataset.xlsx",
                             sheet = excel_sheets("Dry_Bean_Dataset.xlsx")[2],
                             col_names = FALSE)
```

```
New names:
* `` -> `...1`
```

```
citation_dry_bean_data
```

```
# A tibble: 1 x 1
  ...1
  <chr>
1 "Citation Request :\r\nKOKLU, M. and OZKAN, I.A., (2020), "Multiclass Classif~
```

```
cat(dplyr::pull(citation_dry_bean_data, 1))
```

```
Citation Request :
KOKLU, M. and OZKAN, I.A., (2020), "Multiclass Classification of Dry Beans Using Computer Vis
```

Reading Only Specific Cells

```
dry_bean_range <- read_excel("Dry_Bean_Dataset.xlsx",
                   range = cell_cols("A:B")
                   )
dry_bean_range
```

```
# A tibble: 13,611 x 2
    Area Perimeter
   <dbl>     <dbl>
 1 28395      610.
 2 28734      638.
 3 29380      624.
 4 30008      646.
 5 30140      620.
 6 30279      635.
 7 30477      670.
 8 30519      630.
 9 30685      636.
10 30834      632.
# i 13,601 more rows
```

Manipulating with dplyr

Going back to air quality data

```
air_quality_data
```

```
# A tibble: 9,471 x 18
    ...1 Date       Time     `CO(GT)` `PT08.S1(CO)` `NMHC(GT)` `C6H6(GT)`
   <dbl> <chr>      <chr>       <dbl>         <dbl>      <dbl>      <dbl>
 1     1 10/03/2004 18.00.00      2.6          1360        150       11.9
 2     2 10/03/2004 19.00.00      2            1292        112        9.4
 3     3 10/03/2004 20.00.00      2.2          1402         88        9
 4     4 10/03/2004 21.00.00      2.2          1376         80        9.2
 5     5 10/03/2004 22.00.00      1.6          1272         51        6.5
 6     6 10/03/2004 23.00.00      1.2          1197         38        4.7
 7     7 11/03/2004 00.00.00      1.2          1185         31        3.6
```

```
 8     8 11/03/2004 01.00.00      1            1136         31        3.3
 9     9 11/03/2004 02.00.00      0.9          1094         24        2.3
10    10 11/03/2004 03.00.00      0.6          1010         19        1.7
# i 9,461 more rows
# i 11 more variables: `PT08.S2(NMHC)` <dbl>, `NOx(GT)` <dbl>,
#   `PT08.S3(NOx)` <dbl>, `NO2(GT)` <dbl>, `PT08.S4(NO2)` <dbl>,
#   `PT08.S5(O3)` <dbl>, T <dbl>, RH <dbl>, AH <dbl>, ...17 <lgl>, ...18 <lgl>
```

manipulate to clean it up

```
#view data (equivalent to looking in environment)
View(air_quality_data)
#notice columns 1 and the last two ... columns aren't useful
air_quality_data |>
  select(-starts_with("..."))
```

```
# A tibble: 9,471 x 15
   Date       Time  `CO(GT)` `PT08.S1(CO)` `NMHC(GT)` `C6H6(GT)` `PT08.S2(NMHC)`
   <chr>      <chr>    <dbl>         <dbl>      <dbl>      <dbl>            <dbl>
 1 10/03/2004 18.0~      2.6          1360        150       11.9             1046
 2 10/03/2004 19.0~      2            1292        112        9.4              955
 3 10/03/2004 20.0~      2.2          1402         88        9                939
 4 10/03/2004 21.0~      2.2          1376         80        9.2              948
 5 10/03/2004 22.0~      1.6          1272         51        6.5              836
 6 10/03/2004 23.0~      1.2          1197         38        4.7              750
 7 11/03/2004 00.0~      1.2          1185         31        3.6              690
 8 11/03/2004 01.0~      1            1136         31        3.3              672
 9 11/03/2004 02.0~      0.9          1094         24        2.3              609
10 11/03/2004 03.0~      0.6          1010         19        1.7              561
# i 9,461 more rows
# i 8 more variables: `NOx(GT)` <dbl>, `PT08.S3(NOx)` <dbl>, `NO2(GT)` <dbl>,
#   `PT08.S4(NO2)` <dbl>, `PT08.S5(O3)` <dbl>, T <dbl>, RH <dbl>, AH <dbl>
```

create some new variables

```
#need to first rename columns so they are standard
air_quality_data |>
  select(-starts_with("...")) |>
  rename("co_gt" = 'CO(GT)', "pt_08_s1_co" = 'PT08.S1(CO)',
```

```
         "nmhc_gt" = 'NMHC(GT)', "c6h6_gt" = 'C6H6(GT)',
         "pt_08_s2_nmhc" = 'PT08.S2(NMHC)', "nox_gt" = 'NOx(GT)',
         "pt_08_s3_nox" = 'PT08.S3(NOx)', "no2_gt" = 'NO2(GT)',
         "pt_08_s4_no2" = 'PT08.S4(NO2)', "pt_08_s5_o3" = 'PT08.S5(O3)') |>
  filter(co_gt != -200) |>
  mutate(mean_co_gt = mean(co_gt, na.rm = TRUE)) |>
  View()
#he thinks the -200 are in co_gt are missing values, so added filter to remove
```

add mean for all numeric columns

```
#need to specify names if you don't want columns replaced, using {'col} in
#documentation for across
air_quality_data |>
  select(-starts_with("...")) |>
  rename("co_gt" = 'CO(GT)', "pt_08_s1_co" = 'PT08.S1(CO)',
         "nmhc_gt" = 'NMHC(GT)', "c6h6_gt" = 'C6H6(GT)',
         "pt_08_s2_nmhc" = 'PT08.S2(NMHC)', "nox_gt" = 'NOx(GT)',
         "pt_08_s3_nox" = 'PT08.S3(NOx)', "no2_gt" = 'NO2(GT)',
         "pt_08_s4_no2" = 'PT08.S4(NO2)', "pt_08_s5_o3" = 'PT08.S5(O3)') |>
  filter(co_gt != -200) |>
  mutate(across(where(is.numeric), mean, .names = "mean_{.col}")) |>
  View()
```

```
#can also return multiple functions
air_quality_data |>
  select(-starts_with("...")) |>
  rename("co_gt" = 'CO(GT)', "pt_08_s1_co" = 'PT08.S1(CO)',
         "nmhc_gt" = 'NMHC(GT)', "c6h6_gt" = 'C6H6(GT)',
         "pt_08_s2_nmhc" = 'PT08.S2(NMHC)', "nox_gt" = 'NOx(GT)',
         "pt_08_s3_nox" = 'PT08.S3(NOx)', "no2_gt" = 'NO2(GT)',
         "pt_08_s4_no2" = 'PT08.S4(NO2)', "pt_08_s5_o3" = 'PT08.S5(O3)') |>
  filter(co_gt != -200) |>
  mutate(across(where(is.numeric), list(mean = mean, sd = sd),
                .names = "{.col}_{.fn}")) |>
  View()
```

add in grouping functionality

```
air_quality_data |>
  select(-starts_with("...")) |>
  rename("co_gt" = 'CO(GT)', "pt_08_s1_co" = 'PT08.S1(CO)',
         "nmhc_gt" = 'NMHC(GT)', "c6h6_gt" = 'C6H6(GT)',
         "pt_08_s2_nmhc" = 'PT08.S2(NMHC)', "nox_gt" = 'NOx(GT)',
         "pt_08_s3_nox" = 'PT08.S3(NOx)', "no2_gt" = 'NO2(GT)',
         "pt_08_s4_no2" = 'PT08.S4(NO2)', "pt_08_s5_o3" = 'PT08.S5(O3)') |>
  filter(co_gt != -200) |>
  group_by(Date) |>
  mutate(across(where(is.numeric), list(mean = mean, sd = sd),
                .names = "{.col}_{.fn}")) |>
  View()
```

Manipulating data with tidyr

```
#in wide form
temps_data <- read_table(file = "https://www4.stat.ncsu.edu/~online/datasets/cityTemps.txt")
```

```
-- Column specification --------------------------------------------------------
cols(
  city = col_character(),
  sun = col_double(),
  mon = col_double(),
  tue = col_double(),
  wed = col_double(),
  thr = col_double(),
  fri = col_double(),
  sat = col_double()
)
```

```
head(temps_data)
```

```
# A tibble: 6 x 8
  city        sun   mon   tue   wed   thr   fri   sat
  <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
```

```
1 atlanta      81    87    83    79    88    91    94
2 baltimore    73    75    70    78    73    75    79
3 charlotte    82    80    75    82    83    88    93
4 denver       72    71    67    68    72    71    58
5 ellington    51    42    47    52    55    56    59
6 frankfort    70    70    72    70    74    74    79
```

```
#convert to long form
library(tidyr)
temps_data |>
  pivot_longer(cols = 2:8,
               names_to = "day",
               values_to = "temp")
```

```
# A tibble: 42 x 3
   city      day    temp
   <chr>     <chr> <dbl>
 1 atlanta   sun      81
 2 atlanta   mon      87
 3 atlanta   tue      83
 4 atlanta   wed      79
 5 atlanta   thr      88
 6 atlanta   fri      91
 7 atlanta   sat      94
 8 baltimore sun      73
 9 baltimore mon      75
10 baltimore tue      70
# i 32 more rows
```

convert to wide form

```
library(dplyr)
library(Lahman)
batting_tbl <- as_tibble(Batting)
#subset data for just pirates, select hits and year columns, and pivot that data
#set wider so that we have the year across the top (names_from), the players as
#the rows, and the entries as the hits (values_from)
batting_tbl |>
  filter(yearID %in% 2018:2020, teamID == "PIT") |>
  select(playerID, yearID, H) |>
  pivot_wider(names_from = yearID, values_from = "H")
```

```
# A tibble: 96 x 4
   playerID  `2018` `2019` `2020`
   <chr>      <int>  <int>  <int>
 1 anderta01      0     NA     NA
 2 archech01      2      4     NA
 3 belljo02     131    146     44
 4 bostich01      0     NA     NA
 5 braulst01      3     14      0
 6 burdini01      0      0      0
 7 cervefr01     86     21     NA
 8 crickky01      0      0      0
 9 diazel01      72     73     NA
10 dickeco01    151     40     NA
# i 86 more rows
```

```
#not missing values, if we want to drop do this
batting_tbl |>
  filter(yearID %in% 2018:2020, teamID == "PIT") |>
  select(playerID, yearID, H) |>
  pivot_wider(names_from = yearID, values_from = "H") |>
  drop_na()
```

```
# A tibble: 17 x 4
   playerID  `2018` `2019` `2020`
   <chr>      <int>  <int>  <int>
 1 belljo02     131    146     44
 2 braulst01      3     14      0
 3 burdini01      0      0      0
 4 crickky01      0      0      0
 5 felizmi01      0      0      0
 6 fraziad01     88    154     48
 7 holmecl01      0      0      0
 8 kelake01       0      0      0
 9 moranco01    115    129     44
10 musgrjo01      5      8      0
11 neverdo01      0      0      0
12 newmake01     19    152     35
13 osunajo01     24     69     16
14 polangr01    117     37     24
15 rodriri05      0      0      0
16 stallja01      8     50     31
17 willitr01      5      6      0
```

```r
#Let's also remove those with 0 hits:
batting_tbl |>
  filter(yearID %in% 2018:2020, teamID == "PIT", H > 0) |>
  select(playerID, yearID, H) |>
  pivot_wider(names_from = yearID, values_from = "H") |>
  drop_na()
```

```
# A tibble: 7 x 4
  playerID  `2018` `2019` `2020`
  <chr>      <int>  <int>  <int>
1 belljo02     131    146     44
2 fraziad01     88    154     48
3 moranco01    115    129     44
4 newmake01     19    152     35
5 osunajo01     24     69     16
6 polangr01    117     37     24
7 stallja01      8     50     31
```

column manipulation with tidyr

```r
chicago_data <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/Chicago.csv")
```

```
Rows: 1461 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): city, date, season
dbl (8): X, death, temp, dewpoint, pm10, o3, time, year

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.
```

```r
chicago_data
```

```
# A tibble: 1,461 x 11
      X city  date      death  temp dewpoint  pm10    o3  time season  year
  <dbl> <chr> <chr>     <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <chr>  <dbl>
1  3654 chic  1/1/1997    137  36      37.5  13.1   5.66  3654 winter  1997
2  3655 chic  1/2/1997    123  45      47.2  41.9   5.53  3655 winter  1997
```

```
 3  3656 chic  1/3/1997    127  40      38    27.0   6.29  3656 winter  1997
 4  3657 chic  1/4/1997    146  51.5    45.5  25.1   7.54  3657 winter  1997
 5  3658 chic  1/5/1997    102  27      11.2  15.3  20.8   3658 winter  1997
 6  3659 chic  1/6/1997    127  17       5.75  9.36 14.9   3659 winter  1997
 7  3660 chic  1/7/1997    116  16       7    20.2  11.9   3660 winter  1997
 8  3661 chic  1/8/1997    118  19      17.8  33.1   8.68  3661 winter  1997
 9  3662 chic  1/9/1997    148  26      24    12.1  13.4   3662 winter  1997
10  3663 chic  1/10/1997   121  16       5.38 24.8  10.4   3663 winter  1997
# i 1,451 more rows
```

```
#change dates
chicago_data |>
  separate_wider_delim(cols = date,
                       delim = "/",
                       names = c("Month", "Day", "Year"),
                       cols_remove = FALSE)
```

```
# A tibble: 1,461 x 14
      X city  Month Day   Year  date     death  temp dewpoint  pm10    o3  time
  <dbl> <chr> <chr> <chr> <chr> <chr>    <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
 1  3654 chic  1     1     1997  1/1/1997   137  36      37.5  13.1   5.66  3654
 2  3655 chic  1     2     1997  1/2/1997   123  45      47.2  41.9   5.53  3655
 3  3656 chic  1     3     1997  1/3/1997   127  40      38    27.0   6.29  3656
 4  3657 chic  1     4     1997  1/4/1997   146  51.5    45.5  25.1   7.54  3657
 5  3658 chic  1     5     1997  1/5/1997   102  27      11.2  15.3  20.8   3658
 6  3659 chic  1     6     1997  1/6/1997   127  17       5.75  9.36 14.9   3659
 7  3660 chic  1     7     1997  1/7/1997   116  16       7    20.2  11.9   3660
 8  3661 chic  1     8     1997  1/8/1997   118  19      17.8  33.1   8.68  3661
 9  3662 chic  1     9     1997  1/9/1997   148  26      24    12.1  13.4   3662
10  3663 chic  1     10    1997  1/10/19~   121  16       5.38 24.8  10.4   3663
# i 1,451 more rows
# i 2 more variables: season <chr>, year <dbl>
```

combine two columns for display purposes

```
chicago_data |>
  unite(col = "season_date", season, date, sep = ": ") |>
  select(season_date, everything())
```

```
# A tibble: 1,461 x 10
```

```
   season_date           X city  death  temp dewpoint  pm10    o3  time  year
   <chr>             <dbl> <chr> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
 1 winter: 1/1/1997   3654 chic    137  36      37.5  13.1   5.66  3654  1997
 2 winter: 1/2/1997   3655 chic    123  45      47.2  41.9   5.53  3655  1997
 3 winter: 1/3/1997   3656 chic    127  40      38    27.0   6.29  3656  1997
 4 winter: 1/4/1997   3657 chic    146  51.5    45.5  25.1   7.54  3657  1997
 5 winter: 1/5/1997   3658 chic    102  27      11.2  15.3  20.8   3658  1997
 6 winter: 1/6/1997   3659 chic    127  17       5.75  9.36 14.9   3659  1997
 7 winter: 1/7/1997   3660 chic    116  16       7    20.2  11.9   3660  1997
 8 winter: 1/8/1997   3661 chic    118  19      17.8  33.1   8.68  3661  1997
 9 winter: 1/9/1997   3662 chic    148  26      24    12.1  13.4   3662  1997
10 winter: 1/10/1997  3663 chic    121  16       5.38 24.8  10.4   3663  1997
# i 1,451 more rows
```

Databases and basic SQL

connecting to databases

```
library(DBI)
#con <- dbConnect(data_base_type_goes_here_usually_requires_a_package,
#  host = "hostname.website",
#  user = "username",
#  password = rstudioapi::askForPassword("DB password")
#)
#This code tells R where the connection exists (host) and, if you need to login to gain acces
```

databases

• RSQLite::SQLite() for RSQLite • RMySQL::MySQL() for RMySQL • RPostgreSQL::PostgreSQL() for RPostgreSQL • odbc::odbc() for Open Database Connectivity • bigrquery::bigquery() for google's bigQuery

querying a table

use tbl() to reference a table in the database

disconnecting

dbDisconnect(con)

Practice

```
library(DBI)
con <- dbConnect(RSQLite::SQLite(), "lahman.db")
#view all tables in database
dbListTables(con)
```

```
 [1] "AllstarFull"         "Appearances"         "AwardsManagers"
 [4] "AwardsPlayers"       "AwardsShareManagers" "AwardsSharePlayers"
 [7] "Batting"             "BattingPost"         "CollegePlaying"
[10] "Fielding"            "FieldingOF"          "FieldingOFsplit"
[13] "FieldingPost"        "HallOfFame"          "HomeGames"
[16] "LahmanData"          "Managers"            "ManagersHalf"
[19] "Parks"               "People"              "Pitching"
[22] "PitchingPost"        "Salaries"            "Schools"
[25] "SeriesPost"          "Teams"               "TeamsFranchises"
[28] "TeamsHalf"           "battingLabels"       "fieldingLabels"
[31] "pitchingLabels"
```

```
#or
DBI::dbListTables(con)
```

```
 [1] "AllstarFull"         "Appearances"         "AwardsManagers"
 [4] "AwardsPlayers"       "AwardsShareManagers" "AwardsSharePlayers"
 [7] "Batting"             "BattingPost"         "CollegePlaying"
[10] "Fielding"            "FieldingOF"          "FieldingOFsplit"
[13] "FieldingPost"        "HallOfFame"          "HomeGames"
[16] "LahmanData"          "Managers"            "ManagersHalf"
[19] "Parks"               "People"              "Pitching"
[22] "PitchingPost"        "Salaries"            "Schools"
[25] "SeriesPost"          "Teams"               "TeamsFranchises"
[28] "TeamsHalf"           "battingLabels"       "fieldingLabels"
[31] "pitchingLabels"
```

access specific table

```
tbl(con, "Pitching")
```

```
# Source:   table<`Pitching`> [?? x 30]
# Database: sqlite 3.46.0 [C:\Users\kimel\Documents\ST558test\lahman.db]
   playerID  yearID stint teamID lgID      W     L     G    GS    CG   SHO    SV
   <chr>      <int> <int> <chr>  <chr> <int> <int> <int> <int> <int> <int> <int>
 1 bechtge01   1871     1 PH1    NA        1     2     3     3     2     0     0
 2 brainas01   1871     1 WS3    NA       12    15    30    30    30     0     0
 3 fergubo01   1871     1 NY2    NA        0     0     1     0     0     0     0
 4 fishech01   1871     1 RC1    NA        4    16    24    24    22     1     0
 5 fleetfr01   1871     1 NY2    NA        0     1     1     1     1     0     0
 6 flowedi01   1871     1 TRO    NA        0     0     1     0     0     0     0
 7 mackde01    1871     1 RC1    NA        0     1     3     1     1     0     0
 8 mathebo01   1871     1 FW1    NA        6    11    19    19    19     1     0
 9 mcbridi01   1871     1 PH1    NA       18     5    25    25    25     0     0
10 mcmuljo01   1871     1 TRO    NA       12    15    29    29    28     0     0
# i more rows
# i 18 more variables: IPouts <int>, H <int>, ER <int>, HR <int>, BB <int>,
#   SO <int>, BAOpp <dbl>, ERA <dbl>, IBB <int>, WP <int>, HBP <int>, BK <int>,
#   BFP <int>, GF <int>, R <int>, SH <int>, SF <int>, GIDP <int>
```

get data into R (above doesn't, need to tell R)

```
tbl(con, "Pitching") |>
  select(ends_with("ID")) |>
  filter(yearID == 2010) |>
  collect()
```

```
# A tibble: 684 x 4
   playerID  yearID teamID lgID
   <chr>      <int> <chr>  <chr>
 1 aardsda01   2010 SEA    AL
 2 abadfe01    2010 HOU    NL
 3 accarje01   2010 TOR    AL
 4 aceveal01   2010 NYA    AL
 5 acostma01   2010 NYN    NL
 6 adamsmi03   2010 SDN    NL
```

```
 7 affelje01   2010 SFN    NL
 8 albaljo01   2010 NYA    AL
 9 alberma01   2010 BAL    AL
10 ambrihe01   2010 CLE    AL
# i 674 more rows
```

```
#get out some SQL code from our dplyr code
tbl(con, "Pitching") |>
  select(ends_with("ID")) |>
  filter(yearID == 2010) |>
  show_query()
```

```
<SQL>
SELECT `playerID`, `yearID`, `teamID`, `lgID`
FROM `Pitching`
WHERE (`yearID` = 2010.0)
```

```
#or write straight SQL code
tbl(con, sql(
"SELECT `playerID`, `yearID`, `teamID`, `lgID`
FROM `Pitching`
WHERE (`yearID` = 2010.0)")
)
```

```
# Source:   SQL [?? x 4]
# Database: sqlite 3.46.0 [C:\Users\kimel\Documents\ST558test\lahman.db]
   playerID  yearID teamID lgID
   <chr>      <int> <chr>  <chr>
 1 aardsda01   2010 SEA    AL
 2 abadfe01    2010 HOU    NL
 3 accarje01   2010 TOR    AL
 4 aceveal01   2010 NYA    AL
 5 acostma01   2010 NYN    NL
 6 adamsmi03   2010 SDN    NL
 7 affelje01   2010 SFN    NL
 8 albaljo01   2010 NYA    AL
 9 alberma01   2010 BAL    AL
10 ambrihe01   2010 CLE    AL
# i more rows
```

```
#disconnect
dbDisconnect(con)
```

Reading from a database (video)

Sample Query on website:

– This query shows a list of the daily top Google Search terms. SELECT refresh_date AS Day, term AS Top_Term, – These search terms are in the top 25 in the US each day. rank, FROM `bigquery-public-data.google_trends.top_terms` WHERE rank = 1 – Choose only the top term each day. AND refresh_date >= DATE_SUB(CURRENT_DATE(), INTERVAL 2 WEEK) – Filter to the last 2 weeks. GROUP BY Day, Top_Term, rank ORDER BY Day DESC – Show the days in reverse chronological order.

```
#need bigrquery package
library(DBI)
library(tidyverse)
con <- dbConnect(
  bigrquery::bigquery(),
  project = "bigquery-public-data",
  dataset = "google_trends",
  billing = "st558-424916"
)
#NOTES, can see Google Trends Demo Query and select Open This Query to see
#data sets,google_trends is there,there is a sample query (SQL), can run it
#within the web interface
#Code below lists tables availablw within goggle_trends
dbListTables(con)
```

```
! Using an auto-discovered, cached token.

  To suppress this message, modify your code or options to clearly consent to
  the use of a cached token.

  See gargle's "Non-interactive auth" vignette for more details:

  <https://gargle.r-lib.org/articles/non-interactive-auth.html>

i The bigrquery package is using a cached token for 'mbeebe@ncsu.edu'.
```

```
[1] "international_top_rising_terms" "international_top_terms"
[3] "top_rising_terms"               "top_terms"
```

```
# choose something in google_trends, can click in browser and see variables, or
#allternatiely to see variables use (names didn't work, update in R I guess)
#tbl(con, "top_terms") |>
#  names()
tbl(con, "top_terms") |>
  colnames()
```

```
[1] "term"         "week"         "score"        "rank"         "refresh_date"
[6] "dma_name"     "dma_id"
```

```
#get desired data, want dates greater than 5/14
my_data <- tbl(con, "top_terms") |>
  select(refresh_date, term, rank, dma_id, dma_name ) |>
  rename("Day" = "refresh_date", "Top_Term" = "term") |>
  filter(rank == 1, Day > lubridate::as_date("2024-05-14"), dma_id ==500) |>
  collect()
my_data
```

```
# A tibble: 4,451 x 5
   Day        Top_Term         rank dma_id dma_name
   <date>     <chr>           <int>  <int> <chr>
 1 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 2 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 3 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 4 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 5 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 6 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 7 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 8 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
 9 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
10 2024-05-26 Grayson Murray      1    500 Portland-Auburn ME
# i 4,441 more rows
```

Look at my_data in environment to see what dma_id is. It is Portland_Auburn ME

SQL Style Joins

inner join

batting and pitching have player ID, stint, lgID in common

SEE CODE IN SQL JOINS NOTES