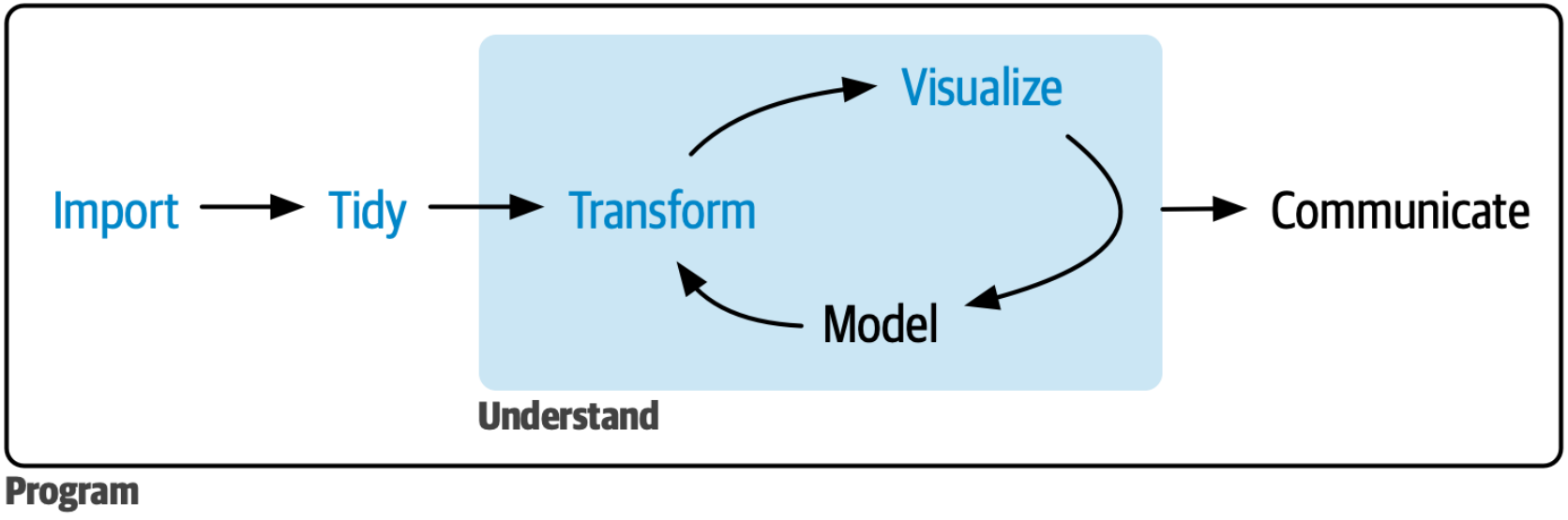
https://r4ds.hadley.nz/ Chapter 3-9

CSCI 297b, Spring 2023

May 2, 2023

The Big Picture



The dplyr package and the nycflights13 dataset

```
library(nycflights13)
library(tidyverse)
```

the nycflights13 dataset

```
> glimpse(flights)
Rows: 336,776
Columns: 19
$ year           <int> 2013, 2013, 2013, 2013,...
$ month          <int> 1, 1, 1, 1, 1, 1, 1, 1,...
$ day            <int> 1, 1, 1, 1, 1, 1, 1, 1,...
$ dep_time       <int> 517, 533, 542, 544, 554...
$ sched_dep_time <int> 515, 529, 540, 545, 600...
$ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5...
$ arr_time       <int> 830, 850, 923, 1004, 81...
$ sched_arr_time <int> 819, 830, 850, 1022, 83...
$ arr_delay      <dbl> 11, 20, 33, -18, -25, 1...
$ carrier        <chr> "UA", "UA", "AA", "B6",...
$ flight         <int> 1545, 1714, 1141, 725, ...
$ tailnum        <chr> "N14228", "N24211", "N6...
$ origin         <chr> "EWR", "LGA", "JFK", "J...
$ dest           <chr> "IAH", "IAH", "MIA", "B...
$ air_time       <dbl> 227, 227, 160, 183, 116...
$ distance       <dbl> 1400, 1416, 1089, 1576,...
$ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6,...
$ minute         <dbl> 15, 29, 40, 45, 0, 58, ...
$ time_hour      <dttm> 2013-01-01 05:00:00, 2...
```

The dplyr package

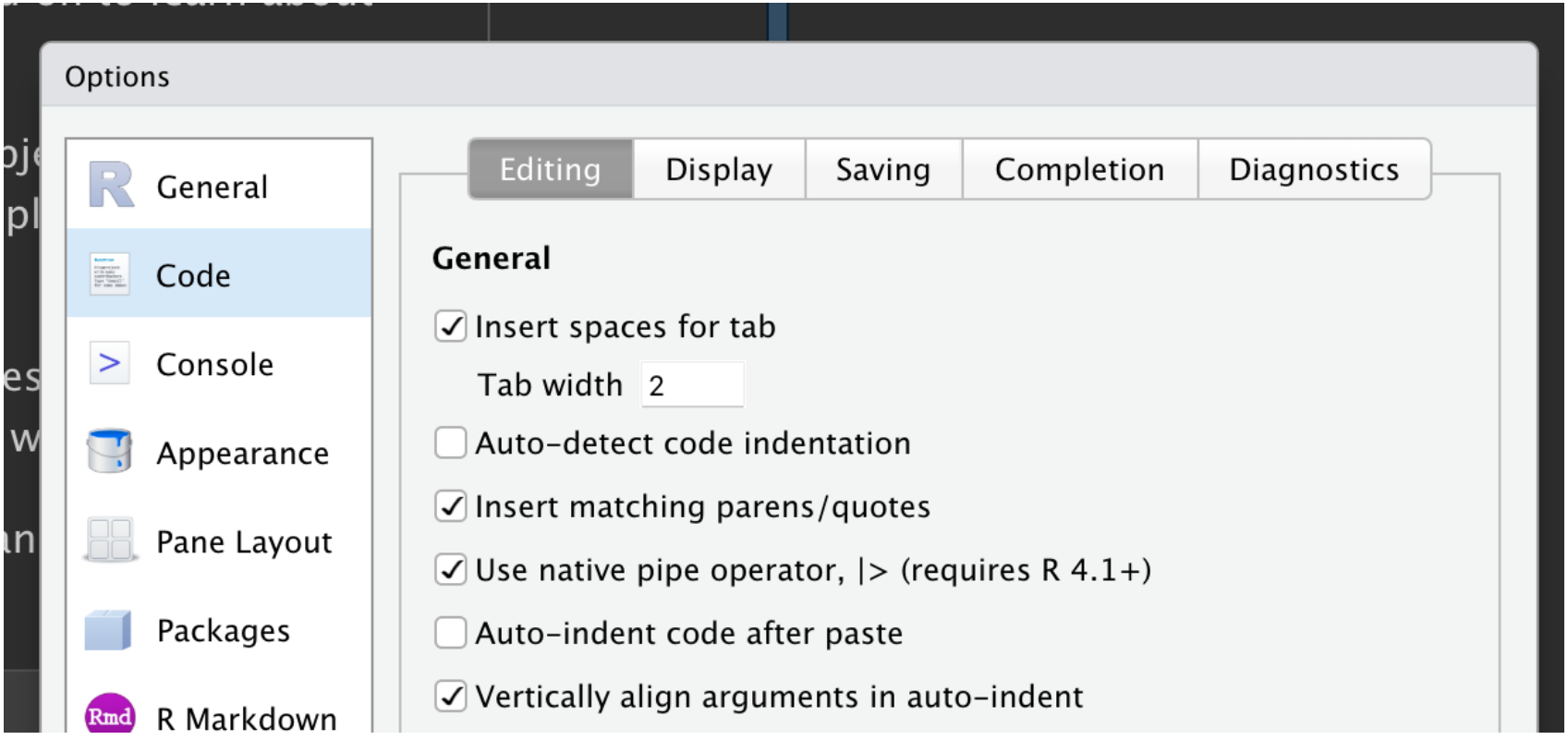
- The first argument is always a data frame.
- The subsequent arguments typically describe which columns to operate on, using the variable names (without quotes).
- The output is always a new data frame.
- Each verb operates on either
  - rows,
  - columns,
  - groups, or
  - tables

The pipe

```
x |> f(y)       ⇔  f(x, y)
x |> f(y) |> g(z)  ⇔  g(f(x, y), z)
```

```
flights |>
  filter(dest == "IAH") |>
  group_by(year, month, day) |>
  summarize(
    arr_delay = mean(arr_delay, na.rm = TRUE)
  )
```

Global options



- Enable "Use native pipe operator."
- This will enable you to produce the pipe with `ctrl-shift-M`

filter

```
flights |>
  filter(dep_delay > 120)
#> # A tibble: 9,723 × 19
#>    year month   day dep_time sched_dep_time dep_delay arr
#>   <int> <int> <int>    <int>          <int>     <dbl>
#> 1  2013     1     1      848           1835       853
#> 2  2013     1     1      957            733       144
#> 3  2013     1     1     1114            900       134
#> 4  2013     1     1     1540           1338       122
#> 5  2013     1     1     1815           1325       290
#> 6  2013     1     1     1842           1422       260
#> # i 9,717 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, fl
```

arrange

```
flights |>
  arrange(year, month, day, dep_time)
#> # A tibble: 336,776 × 19
#>    year month   day dep_time sched_dep_time dep_delay ar
#>   <int> <int> <int>    <int>          <int>     <dbl>
#> 1  2013     1     1      517            515         2
#> 2  2013     1     1      533            529         4
#> 3  2013     1     1      542            540         2
#> 4  2013     1     1      544            545        -1
#> 5  2013     1     1      554            600        -6
#> 6  2013     1     1      554            558        -4
#> # i 9,717 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, fl
```

distinct

```
# Find all unique origin and destination pairs
flights |>
  distinct(origin, dest)
#> # A tibble: 224 × 2
#>   origin dest
#>   <chr>  <chr>
#> 1 EWR    IAH
#> 2 LGA    IAH
#> 3 JFK    MIA
#> 4 JFK    BQN
#> 5 LGA    ATL
#> 6 EWR    ORD
#> # i 218 more rows
```

count

```
flights |>
  count(origin, dest, sort = TRUE)
#> # A tibble: 224 × 3
#>   origin dest      n
#>   <chr>  <chr> <int>
#> 1 JFK    LAX   11262
#> 2 LGA    ATL   10263
#> 3 LGA    ORD    8857
#> 4 JFK    SFO    8204
#> 5 LGA    CLT    6168
#> 6 EWR    ORD    6100
#> # i 218 more rows
```

Do exercise 6

mutate

```
flights |>
  mutate(
    gain = dep_delay - arr_delay,
    speed = distance / air_time * 60,
    .before = 1
  )
#> # A tibble: 336,776 × 21
#>    gain speed  year month   day dep_time sched_dep_time c
#>   <dbl> <dbl> <int> <int> <int>    <int>          <int>
#> 1    -9  370.  2013     1     1      517            515
#> 2   -16  374.  2013     1     1      533            529
#> 3   -31  408.  2013     1     1      542            540
#> 4    17  517.  2013     1     1      544            545
#> 5    19  394.  2013     1     1      554            600
#> 6   -16  288.  2013     1     1      554            558
#> # i 336,770 more rows
#> # i 12 more variables: sched_arr_time <int>, arr_delay <d
```

select

```
flights |>
  select(year, month, day)

flights |>
  select(year:day)

flights |>
  select(!year:day)

flights |>
  select(where(is.character))
```

- `starts_with("abc")`: matches names that begin with “abc”.
- `ends_with("xyz")`: matches names that end with “xyz”.
- `contains("ijk")`: matches names that contain “ijk”.
- `num_range("x", 1:3)`: matches x1, x2 and x3.

rename

```
flights |>
  rename(tail_num = tailnum)
#> # A tibble: 336,776 × 19
#>    year month   day dep_time sched_dep_time dep_delay ar
#>   <int> <int> <int>    <int>          <int>     <dbl>
#> 1  2013     1     1      517            515         2
#> 2  2013     1     1      533            529         4
#> 3  2013     1     1      542            540         2
#> 4  2013     1     1      544            545        -1
#> 5  2013     1     1      554            600        -6
#> 6  2013     1     1      554            558        -4
```

relocate

```
flights |>
  relocate(time_hour, air_time)
#> # A tibble: 336,776 × 19
#>   time_hour           air_time  year month   day dep_time
#>   <dttm>                 <dbl> <int> <int> <int>    <int>
#> 1 2013-01-01 05:00:00      227  2013     1     1      517
#> 2 2013-01-01 05:00:00      227  2013     1     1      533
#> 3 2013-01-01 05:00:00      160  2013     1     1      542
#> 4 2013-01-01 05:00:00      183  2013     1     1      544
#> 5 2013-01-01 06:00:00      116  2013     1     1      554
#> 6 2013-01-01 05:00:00      150  2013     1     1      554
```

Do exercise 7