Notes on Ch4: Workflow - code style

N. Lim

2025-06-30

- Good coding style is like correct punctuation; you can manage without it, but it sure makes things easier to read.
- A good way to restyle code is to use the **styler** package.

```
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    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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 become error
```

```
library(nycflights13)
```

R-rrific style for Names

- use lowercase letters, numbers, and _.
- use _ to separate words

Example:

```
short_flights <- flights |>
  filter(air_time < 60)
```

R-rrific style for Spaces

- Put spaces on either side of mathematical operators execpt for ^.

Example:

```
z <- (a + b)^2 / d
```

- Don't put spaces inside or outside parentheses for regular function calls.
- Always put a space after a comma, just like in English.

Example:

```
mean(x, na.rm = TRUE)
```

- It's ok to add extra spaces if it improves alignment.

Example:

```r
flights |>
  mutate(
    speed      = distance / air_time,
    dep_hour   = dep_time %/% 100,
    dep_minute = dep_time %% 200
  )
```

```
## # A tibble: 336,776 x 22
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # i 336,766 more rows
## # i 14 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>, speed <dbl>, dep_hour <dbl>,
## #   dep_minute <dbl>
```

R-rrific style for Pipes

- Put spaces before and after the pipe (`|>` or `%>%`)

Example:

```r
flights |>
  filter(!is.na(arr_delay), !is.na(tailnum)) |>
  count(dest)
```

```
## # A tibble: 104 x 2
##    dest      n
##    <chr> <int>
##  1 ABQ     254
##  2 ACK     264
##  3 ALB     418
##  4 ANC       8
##  5 ATL   16837
##  6 AUS    2411
##  7 AVL     261
##  8 BDL     412
##  9 BGR     358
## 10 BHM     269
## # i 94 more rows
```

- If the function you're piping ino has named arguments, put each argument on a new line.

- If the function doesn't have named arguments, keep everything on one line unless it doesn't fit, in which case you should put each argument on its own line.

Example:

```r
flights |>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )
```

```
## # A tibble: 4,044 x 3
##    tailnum  delay     n
##    <chr>    <dbl> <int>
##  1 D942DN  31.5       4
##  2 N0EGMQ   9.98    371
##  3 N10156  12.7     153
##  4 N102UW   2.94     48
##  5 N103US  -6.93     46
##  6 N104UW   1.80     47
##  7 N10575  20.7     289
##  8 N105UW  -0.267    45
##  9 N107US  -5.73     41
## 10 N108UW  -1.25     60
## # i 4,034 more rows
```

- After the first step of the pipeline, indent each line by **two spaces**.
- If you're putting each argument on its own line, indent by an extra two spaces.

Example:

```r
flights |>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )
```

```
## # A tibble: 4,044 x 3
##    tailnum  delay     n
##    <chr>    <dbl> <int>
##  1 D942DN  31.5       4
##  2 N0EGMQ   9.98    371
##  3 N10156  12.7     153
##  4 N102UW   2.94     48
##  5 N103US  -6.93     46
##  6 N104UW   1.80     47
##  7 N10575  20.7     289
##  8 N105UW  -0.267    45
##  9 N107US  -5.73     41
## 10 N108UW  -1.25     60
## # i 4,034 more rows
```

- It's ok to shirk some of these rules if your pipeline fits easily on one line.

Examples:

```r
# This fits on one line
df |> mutate(y = x + 1)
```

```r
# This is preferred because it is easier to add
# more variables in the future
df |>
  mutate(
  y = x + 1
)
```

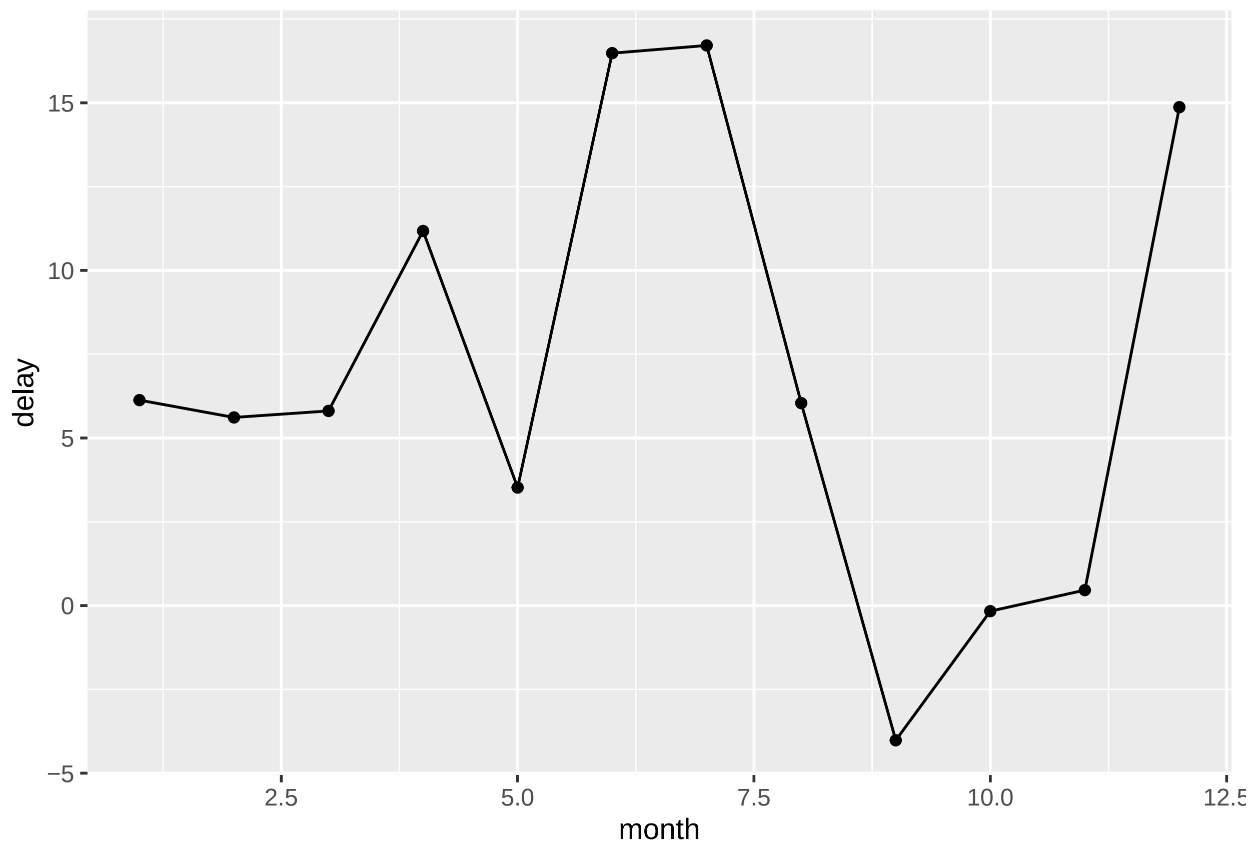
- Be wary of writing very long lines (more than 10-15 lines). It is better to break them up into smaller sub-tasks and give each task an informative name.

R-rrific style for ggplot

- Treat the + the same way as the pipe.

Example:

```r
flights |>
  group_by(month) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE)
  ) |>
  ggplot(aes(x = month, y = delay)) +
  geom_point() +
  geom_line()
```



- If you can't fit all of the arguments to a function on a single line, put each argument on its own line.

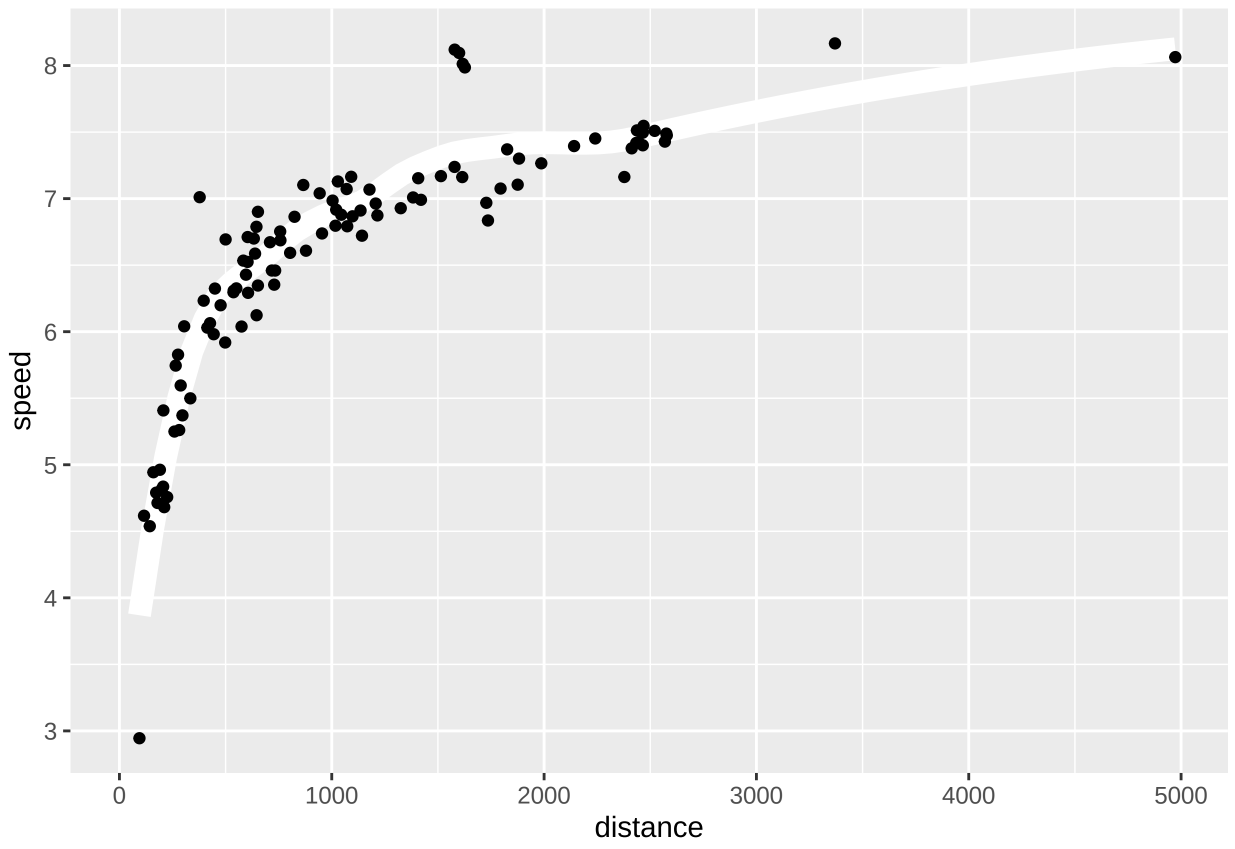
Example:

```r
flights |>
  group_by(dest) |>
  summarize(
    distance = mean(distance),
    speed = mean(distance / air_time, na.rm = TRUE)
  ) |>
  ggplot(aes(x = distance, y = speed)) +
  geom_smooth(
    method = "loess",
    span = 0.5,
    se = FALSE,
    color = "white",
    linewidth = 4
  ) +
  geom_point()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



R-rrific style for sectioning comments

```
# Load data --------------------------------------

# Plot data --------------------------------------
```

Exercises

E1. Restyle the following pipelines following the guidelines above:

—r flights|>filter(dest==“IAH”)|>group_by(year,month,day)|>summarize(n=n(), delay=mean(arr_delay,na.rm=TRUE))|>fil

flights|>filter(carrier==“UA”,dest%in%c(“IAH”,“HOU”),sched_dep_time> 0900,sched_arr_time<2000)|>group_by(flight)|>
arr_delay,na.rm=TRUE),cancelled=sum(is.na(arr_delay)),n=n())|>filter(n>10) —

Solution:

```
flights |>
  filter(dest == "IAH") |>
  group_by(year, month, day) |>
  summarize(
    n = n(),
    delay = mean(arr_delay,na.rm=TRUE)
  ) |>
  filter(n > 10)
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [12]
##     year month   day     n delay
##    <int> <int> <int> <int> <dbl>
##  1  2013     1     1    20 17.8
##  2  2013     1     2    20  7
##  3  2013     1     3    19 18.3
##  4  2013     1     4    20 -3.2
##  5  2013     1     5    13 20.2
##  6  2013     1     6    18  9.28
##  7  2013     1     7    19 -7.74
##  8  2013     1     8    19  7.79
##  9  2013     1     9    19 18.1
## 10  2013     1    10    19  6.68
## # i 355 more rows
```

```
flights |>
  filter(
    carrier == "UA",
    dest %in% c("IAH", "HOU"),
    sched_dep_time > 0900,
    sched_arr_time > 2000
  ) |>
  group_by(flight) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    cancelled = sum(is.na(arr_delay)),
    n = n()
  ) |>
  filter(n > 10)
```

```
## # A tibble: 26 x 4
##    flight  delay cancelled     n
##     <int>  <dbl>     <int> <int>
##  1    301 48.7           1    15
##  2    308 21.2           1    24
##  3    358 19             1    19
##  4    404 10.8           1    37
##  5    475  0.957         0    46
##  6    524 -1.15          0    27
##  7    652 12.9           0    17
##  8    852 11.1           0    20
##  9    891 11.3           2    35
## 10    939  7.56          1    17
## # i 16 more rows
```