ENVIRONMENTAL DATA ANALYTICS: M4 – DATA WRANGLING

Fall 2023

Nicholas School of the Environment - Duke University

Agenda

- Review A03-Data Exploration
- Tackling issues with knitting & working directories
- Factors – what are they?

- Data Wrangling
  - Q & A on recordings
  - Exercises

Knitting tips

- Avoid in your code:
  - install.packages()
  - View()
  - Commands that produce exceptionally long output, if possible

- Restart your R session & run code from start to finish
  - Ensures all packages are installed, objects are created in code itself

- Check working directories & relative paths

- Tidy your R code so it doesn’t extend past the page when knit

Factors

```
months <- c(1,3,2,3,1,1,5,6)
color_factor <- factor(months, levels = c(1,2,3,5,6))
```

- Created using the factor() function, which converts a character vector into a factor by assigning specific **levels** to the unique values in the vector.

- **Labels** can be associated with each level.

  ```
  ...labels=c(‘Jan’, ’Feb’, ‘March’, ‘May’, ‘June’)
  ```

- Used to represent categorical data with predefined levels or categories.

- Useful when you want to work with categorical data in a structured way, as they have an inherent order and a fixed set of possible values (levels).

- Often used in statistical modeling and analysis, as they help in specifying the categories explicitly.

M4.1

Q&A on Data Wrangling

- Datasets, “Tidy Data”
- Importing data
- Wrangling data with `dplyr`

|filter|arrange|select|mutate| ← *covered*

|slice|rename|relocate|summarize| ← *vignette*

Data transformation with dplyr : : **CHEAT SHEET**

dplyr functions work with pipes and expect **tidy data**. In tidy data:

Each **variable** is in its own **column**

&

Each **observation**, or **case**, is in its own **row**

pipes

x %>% f(y) becomes **f(x, y)**

Summarise Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

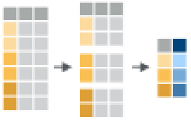
summary function

summarise(.data, ...)
Compute table of summaries.
summarise(mtcars, avg = mean(mpg))

count(.data, ..., wt = NULL, sort = FALSE, name = NULL) Count number of rows in each group defined by the variables in ... Also **tally**().
count(mtcars, cyl)

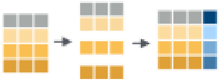
Group Cases

Use **group_by**(.data, ..., .add = FALSE, .drop = TRUE) to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



```
mtcars %>%
  group_by(cyl) %>%
  summarise(avg = mean(mpg))
```

Use **rowwise**(.data, ...) to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidyr cheat sheet for list-column workflow.



```
starwars %>%
  rowwise() %>%
  mutate(film_count = length(films))
```

ungroup(x, ...) Returns ungrouped copy of table.
ungroup(g_mtcars)

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

filter(.data, ..., .preserve = FALSE) Extract rows that meet logical criteria.
filter(mtcars, mpg > 20)

distinct(.data, ..., .keep_all = FALSE) Remove rows with duplicate values.
distinct(mtcars, gear)

slice(.data, ..., .preserve = FALSE) Select rows by position.
slice(mtcars, 10:15)

slice_sample(.data, ..., n, prop, weight_by = NULL, replace = FALSE) Randomly select rows. Use n to select a number of rows and prop to select a fraction of rows.
slice_sample(mtcars, n = 5, replace = TRUE)

slice_min(.data, order_by, ..., n, prop, with_ties = TRUE) and **slice_max**() Select rows with the lowest and highest values.
slice_min(mtcars, mpg, prop = 0.25)

slice_head(.data, ..., n, prop) and **slice_tail**() Select the first or last rows.
slice_head(mtcars, n = 5)

Logical and boolean operators to use with filter()

==	<	<=	is.na()	%in%	|	xor()
!=	>	>=	!is.na()	!	&	

See **?base::Logic** and **?Comparison** for help.

ARRANGE CASES

arrange(.data, ..., .by_group = FALSE) Order rows by values of a column or columns (low to high), use with **desc**() to order from high to low.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

ADD CASES

add_row(.data, ..., .before = NULL, .after = NULL) Add one or more rows to a table.
add_row(cars, speed = 1, dist = 1)

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

pull(.data, var = -1, name = NULL, ...) Extract column values as a vector, by name or index.
pull(mtcars, wt)

select(.data, ...) Extract columns as a table.
select(mtcars, mpg, wt)

relocate(.data, ..., .before = NULL, .after = NULL) Move columns to new position.
relocate(mtcars, mpg, cyl, .after = last_col())

Use these helpers with select() and across()
e.g. select(mtcars, mpg:cyl)

contains(match)	**num_range**(prefix, range)	:, e.g. mpg:cyl
ends_with(match)	**all_of**(x)/**any_of**(x, ..., vars)	-, e.g, -gear
starts_with(match)	**matches**(match)	**everything**()

MANIPULATE MULTIPLE VARIABLES AT ONCE

across(.cols, .funs, ..., .names = NULL) Summarise or mutate multiple columns in the same way.
summarise(mtcars, across(everything(), mean))

c_across(.cols) Compute across columns in row-wise data.
transmute(rowwise(UKgas), total = sum(c_across(1:2)))

MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).

vectorized function

mutate(.data, ..., .keep = "all", .before = NULL, .after = NULL) Compute new column(s). Also **add_column**(), **add_count**(), and **add_tally**().
mutate(mtcars, gpm = 1 / mpg)

transmute(.data, ...) Compute new column(s), drop others.
transmute(mtcars, gpm = 1 / mpg)

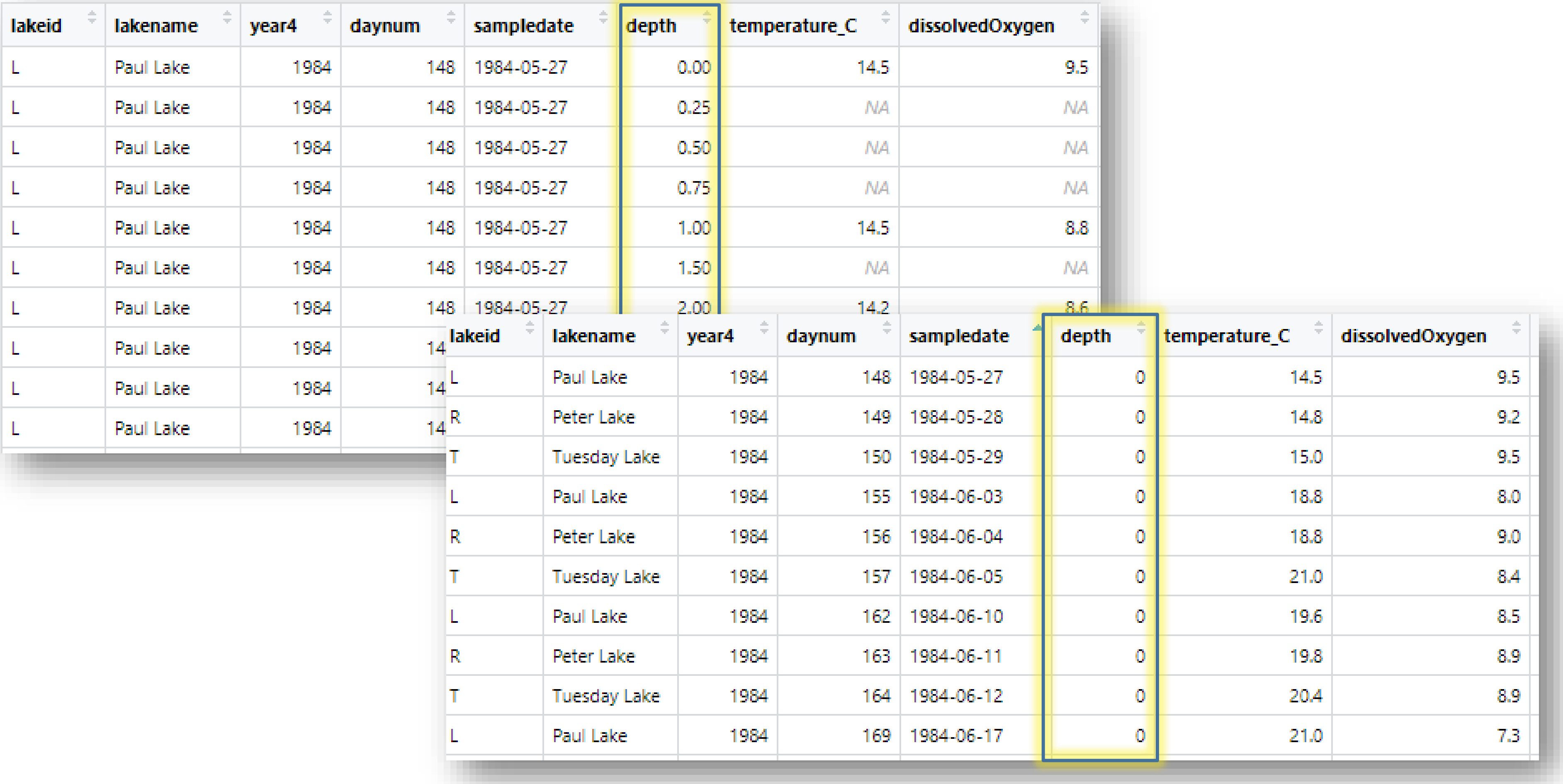
rename(.data, ...) Rename columns. Use **rename_with**() to rename with a function.
rename(cars, distance = dist)

RStudio

Q&A: **dplyr**

Filter	**Arrange**	**Select**	**Mutate**	**Pipes**	**Lubridate**

Subset rows based on a criteria



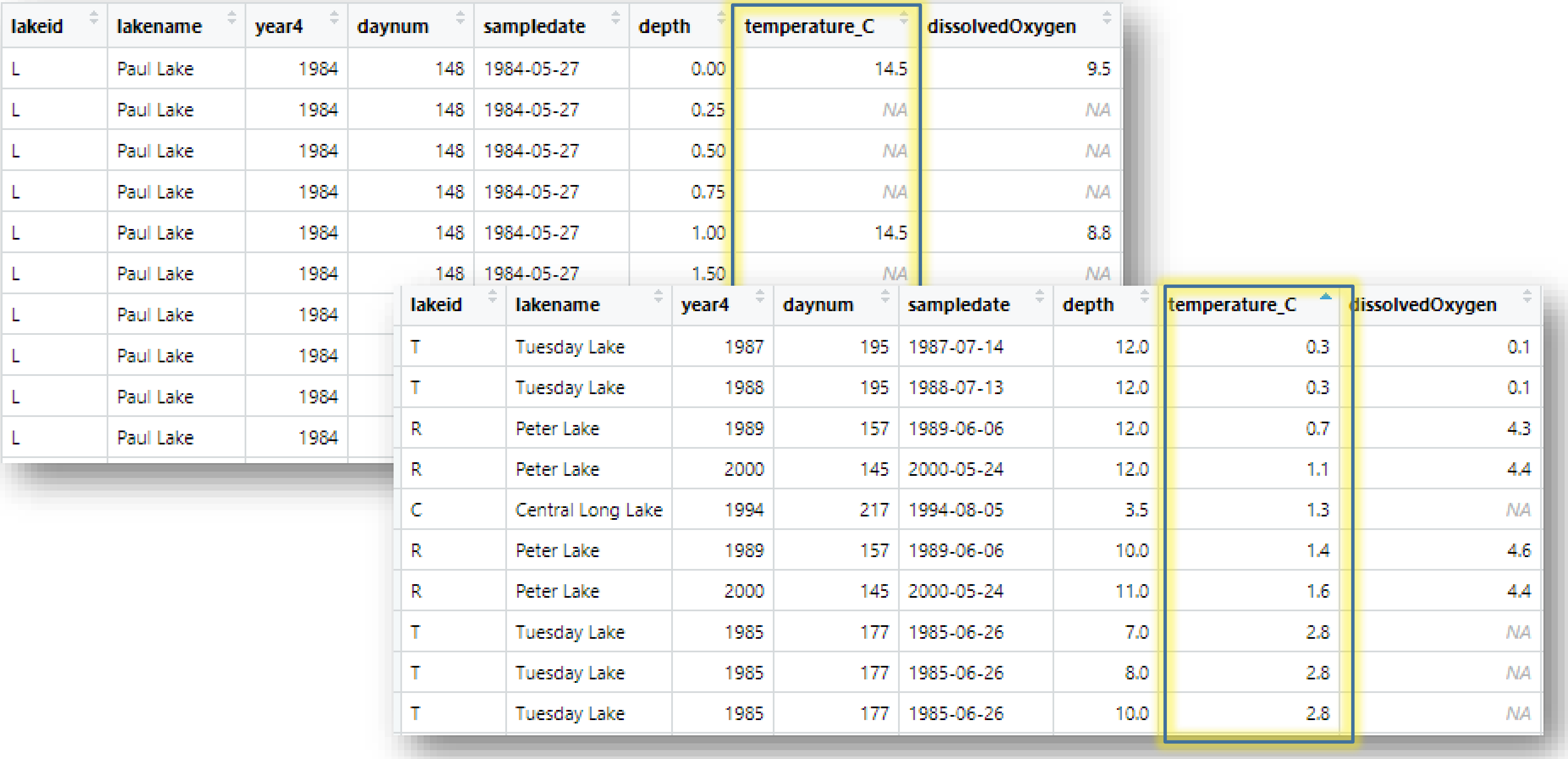
lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	2.00	14.2	8.6
L	Paul Lake	1984	14				
L	Paul Lake	1984	14				
L	Paul Lake	1984	14				

lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
L	Paul Lake	1984	148	1984-05-27	0	14.5	9.5
R	Peter Lake	1984	149	1984-05-28	0	14.8	9.2
T	Tuesday Lake	1984	150	1984-05-29	0	15.0	9.5
L	Paul Lake	1984	155	1984-06-03	0	18.8	8.0
R	Peter Lake	1984	156	1984-06-04	0	18.8	9.0
T	Tuesday Lake	1984	157	1984-06-05	0	21.0	8.4
L	Paul Lake	1984	162	1984-06-10	0	19.6	8.5
R	Peter Lake	1984	163	1984-06-11	0	19.8	8.9
T	Tuesday Lake	1984	164	1984-06-12	0	20.4	8.9
L	Paul Lake	1984	169	1984-06-17	0	21.0	7.3

Q&A: **dplyr**

Filter	**Arrange**	**Select**	**Mutate**	**Pipes**	**Lubridate**

Sort rows based on values in one or more columns…



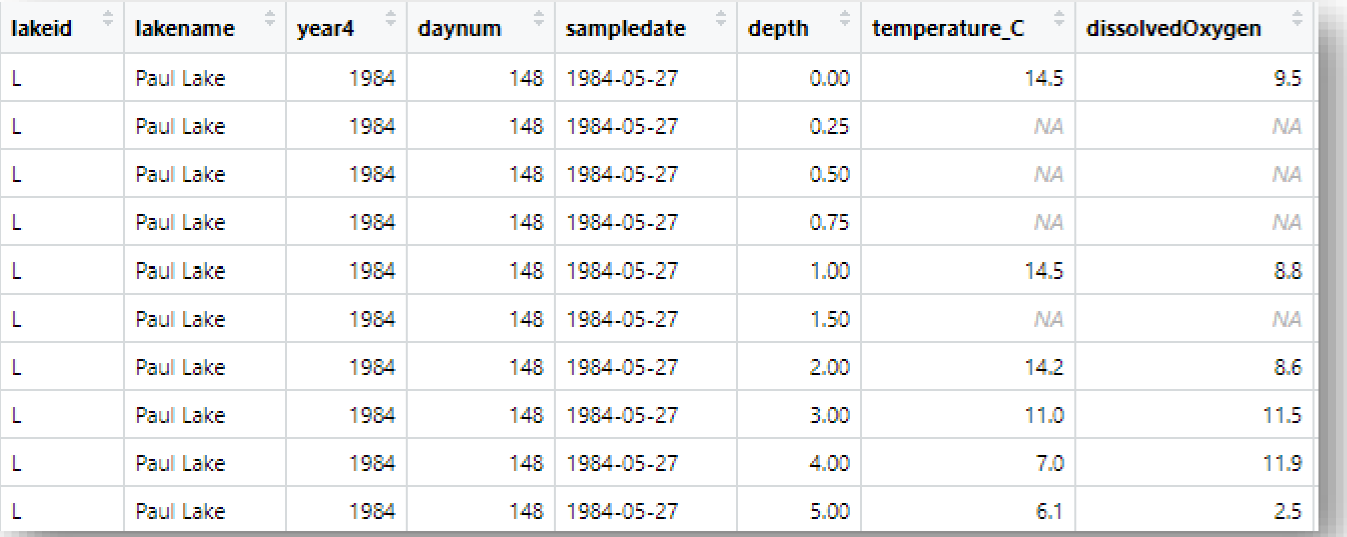
lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA
L	Paul Lake	1984					
L	Paul Lake	1984					
L	Paul Lake	1984					
L	Paul Lake	1984					

lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
T	Tuesday Lake	1987	195	1987-07-14	12.0	0.3	0.1
T	Tuesday Lake	1988	195	1988-07-13	12.0	0.3	0.1
R	Peter Lake	1989	157	1989-06-06	12.0	0.7	4.3
R	Peter Lake	2000	145	2000-05-24	12.0	1.1	4.4
C	Central Long Lake	1994	217	1994-08-05	3.5	1.3	NA
R	Peter Lake	1989	157	1989-06-06	10.0	1.4	4.6
R	Peter Lake	2000	145	2000-05-24	11.0	1.6	4.4
T	Tuesday Lake	1985	177	1985-06-26	7.0	2.8	NA
T	Tuesday Lake	1985	177	1985-06-26	8.0	2.8	NA
T	Tuesday Lake	1985	177	1985-06-26	10.0	2.8	NA

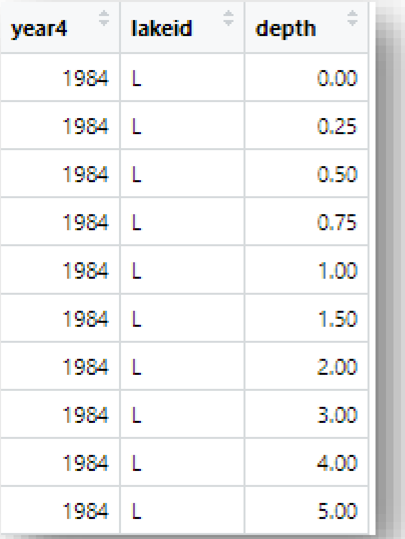
Q&A: **dplyr**

Filter | **Arrange** | **Select** | **Mutate** | **Pipes** | **Lubridate**

Subset/rearrange columns…



lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	2.00	14.2	8.6
L	Paul Lake	1984	148	1984-05-27	3.00	11.0	11.5
L	Paul Lake	1984	148	1984-05-27	4.00	7.0	11.9
L	Paul Lake	1984	148	1984-05-27	5.00	6.1	2.5

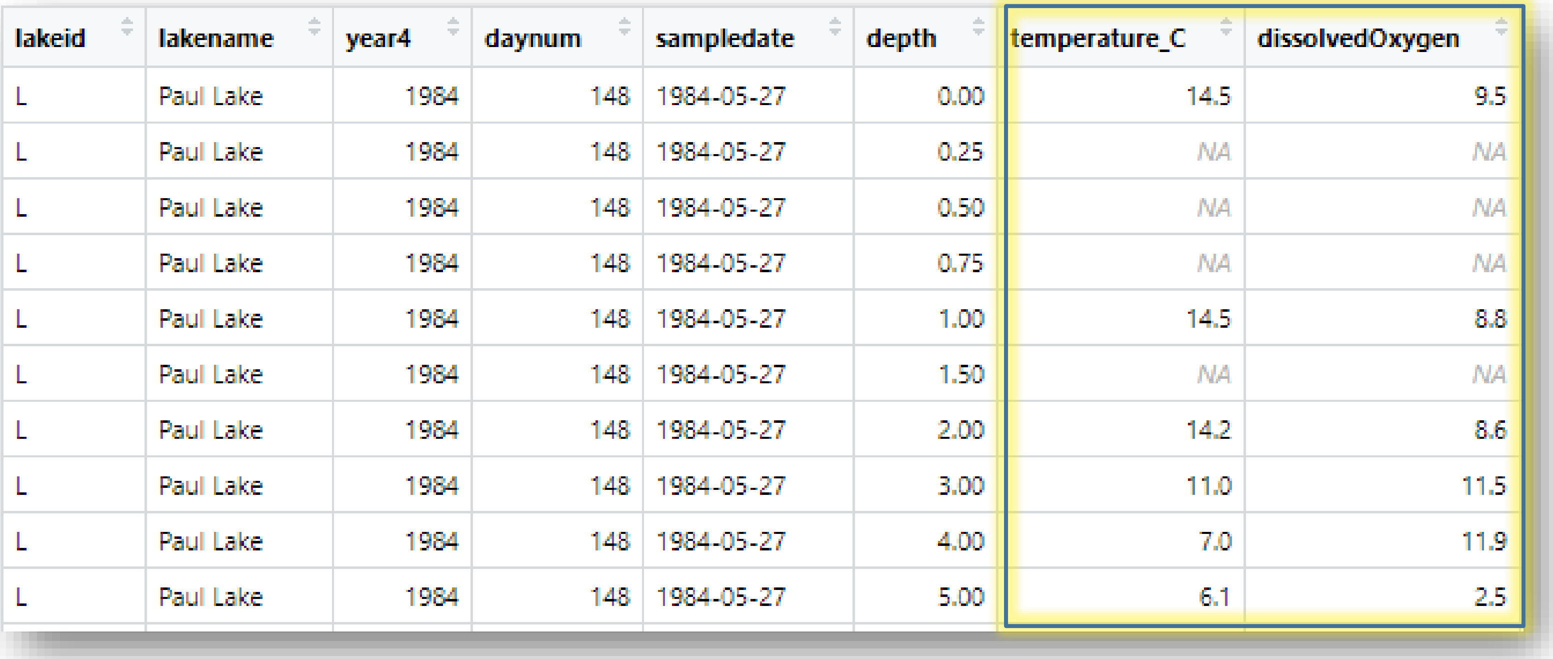


year4	lakeid	depth
1984	L	0.00
1984	L	0.25
1984	L	0.50
1984	L	0.75
1984	L	1.00
1984	L	1.50
1984	L	2.00
1984	L	3.00
1984	L	4.00
1984	L	5.00

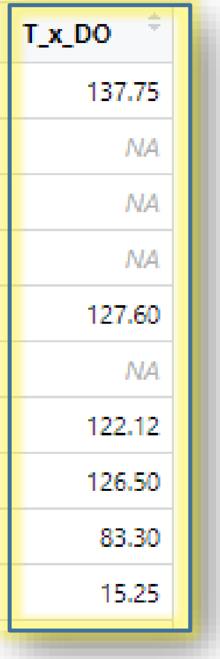
Q&A: **dplyr**

Filter	Arrange	Select	Mutate	Pipes	Lubridate

Calculate a column of new values from existing ones



lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA
L	Paul Lake	1984	148	1984-05-27	2.00	14.2	8.6
L	Paul Lake	1984	148	1984-05-27	3.00	11.0	11.5
L	Paul Lake	1984	148	1984-05-27	4.00	7.0	11.9
L	Paul Lake	1984	148	1984-05-27	5.00	6.1	2.5

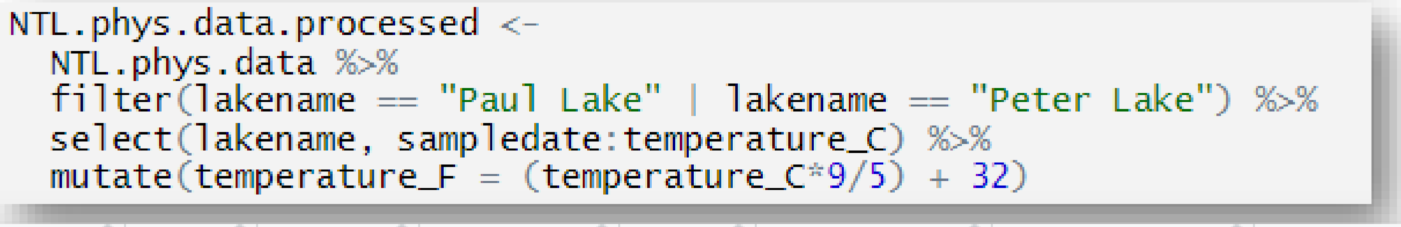


T_x_DO
137.75
NA
NA
NA
127.60
NA
122.12
126.50
83.30
15.25

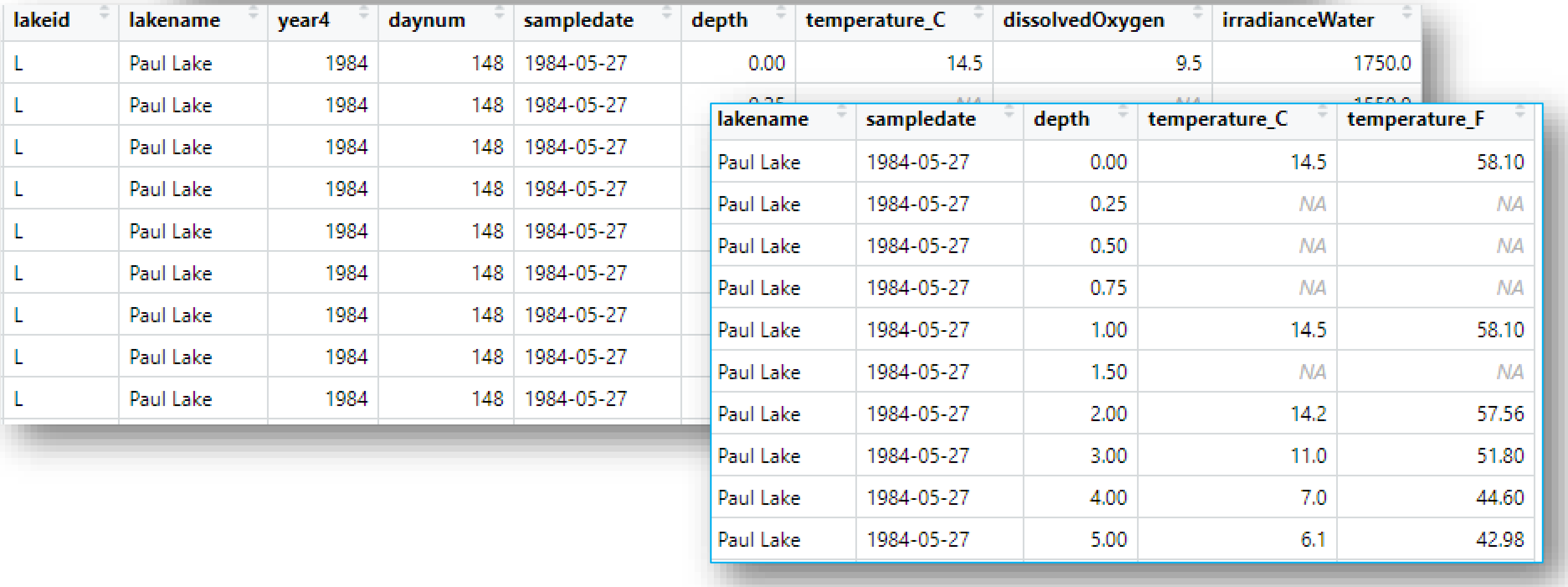
Q&A: **dplyr**

Filter	**Arrange**	**Select**	**Mutate**	**Pipes**	**Lubridate**

Perform multiple operations on a data frame…



```
NTL.phys.data.processed <-
  NTL.phys.data %>%
  filter(lakename == "Paul Lake" | lakename == "Peter Lake") %>%
  select(lakename, sampledate:temperature_C) %>%
  mutate(temperature_F = (temperature_C*9/5) + 32)
```



lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen	irradianceWater
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5	1750.0
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA	1550.0
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				
L	Paul Lake	1984	148	1984-05-27				

lakename	sampledate	depth	temperature_C	temperature_F
Paul Lake	1984-05-27	0.00	14.5	58.10
Paul Lake	1984-05-27	0.25	NA	NA
Paul Lake	1984-05-27	0.50	NA	NA
Paul Lake	1984-05-27	0.75	NA	NA
Paul Lake	1984-05-27	1.00	14.5	58.10
Paul Lake	1984-05-27	1.50	NA	NA
Paul Lake	1984-05-27	2.00	14.2	57.56
Paul Lake	1984-05-27	3.00	11.0	51.80
Paul Lake	1984-05-27	4.00	7.0	44.60
Paul Lake	1984-05-27	5.00	6.1	42.98

M4.2 – Data Wrangling II

Q&A on Data *Pipeline,* transform, grouping

- **Data pipeline:**
  - *Session set-up | Import & Explore | Wrangle*

- **More wrangling**
  - Gather (pivot-longer) & Spread (pivot-wider)
  - Joining datasets
  - Grouping & summarizing data

M4.3 – Data Wrangling III (lab)

1. Import and wrangle

- The data: https://lter.limnology.wisc.edu/about/overview
  - Nutrient data, Physical data
  - Peter and Paul Lakes (Link)

- Import, explore, wrangle
  - Subset for Peter and Paul Lakes
  - Fix dates
  - Filtering (on multiple values with `%in%`)

Exercise 1 & 2: Filtering

- Filter “NTL.phys.data” for the year 1999
  - Should get 1898 rows

- Filter for *Tuesday Lake* records from 1990 thru 1999
  - Should get 1971 rows

Exercise 3: Pipes

- Using pipes: Filter *NTL.phys.data* for:
  - Tuesday Lake
  - from 1990 through 1999
  - only for July

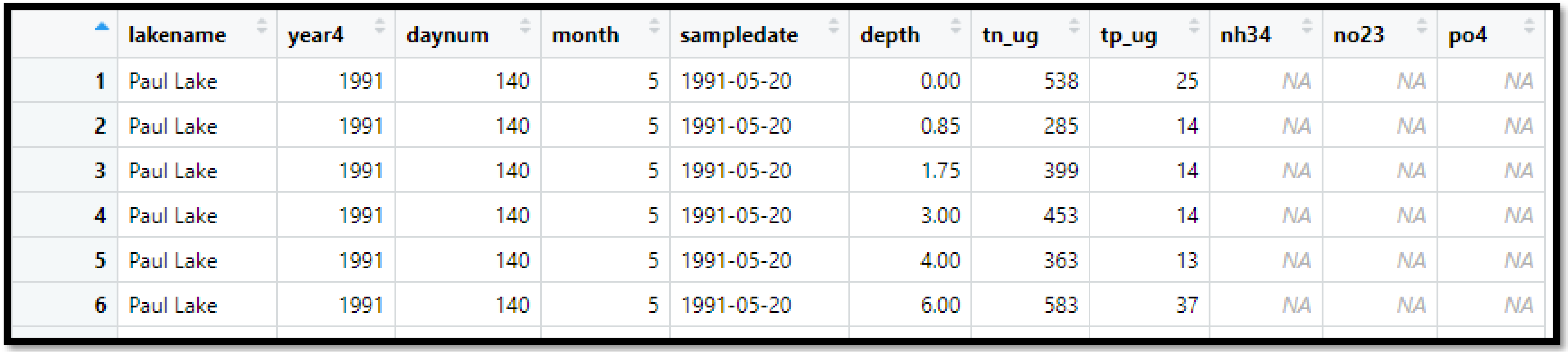
* Tip: you may want to create a new column of just the month

Exercise 4: Pipes

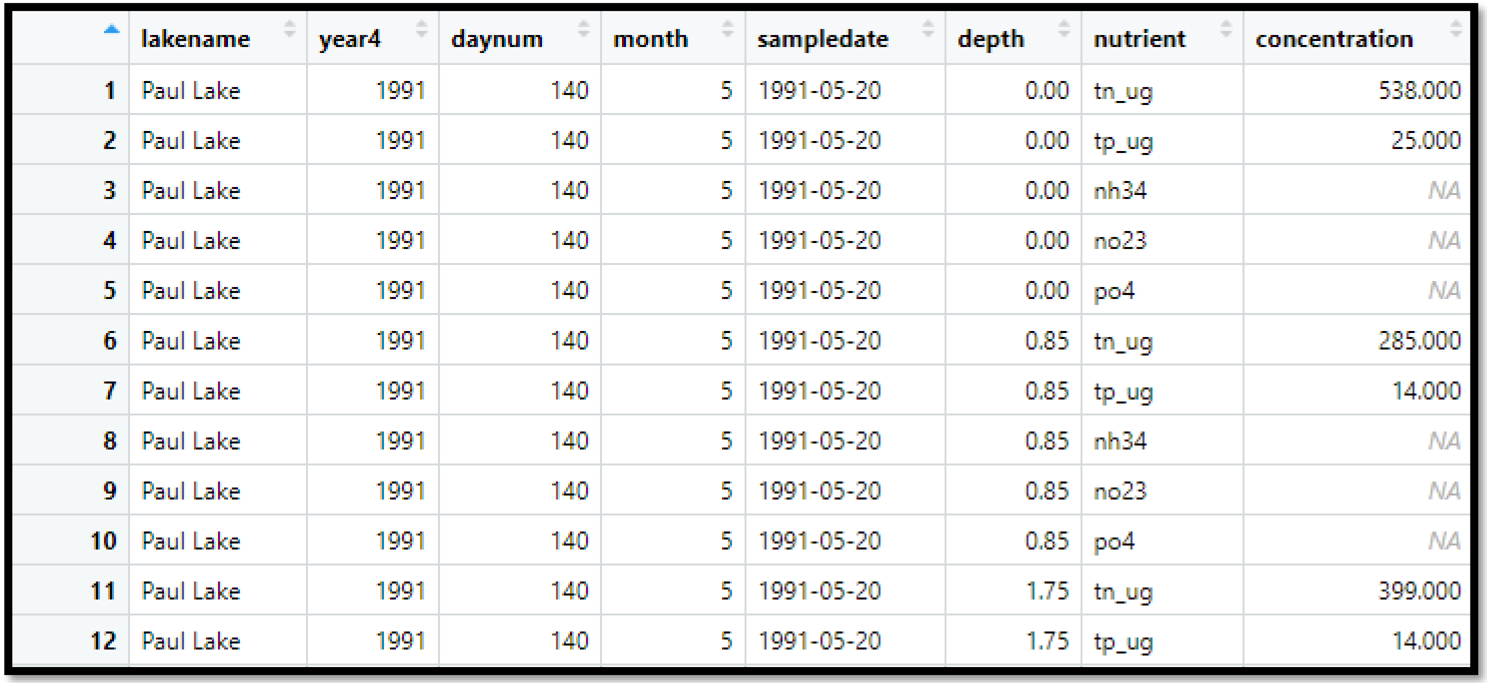
□ Using the data from part 3, pipes, and the `summarize()` function, find the mean surface temperature…

1. Need to subset for surface records…
2. Need to eliminate NAs
3. `summarize()` to compute means on a column

2. Reshape the nutrient data

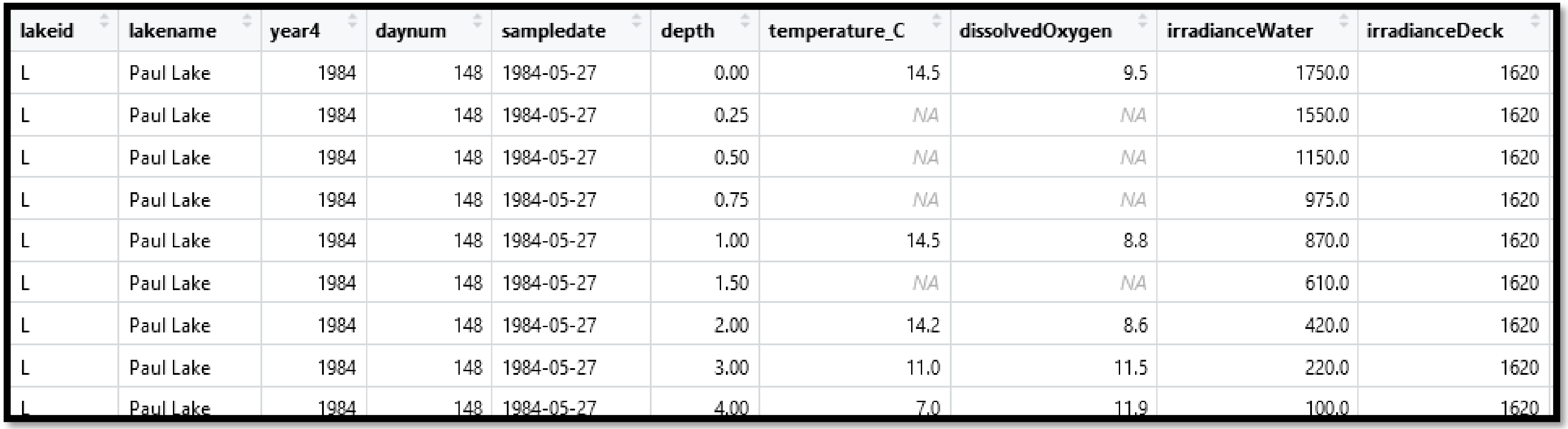


	lakename	year4	daynum	month	sampledate	depth	tn_ug	tp_ug	nh34	no23	po4
1	Paul Lake	1991	140	5	1991-05-20	0.00	538	25	NA	NA	NA
2	Paul Lake	1991	140	5	1991-05-20	0.85	285	14	NA	NA	NA
3	Paul Lake	1991	140	5	1991-05-20	1.75	399	14	NA	NA	NA
4	Paul Lake	1991	140	5	1991-05-20	3.00	453	14	NA	NA	NA
5	Paul Lake	1991	140	5	1991-05-20	4.00	363	13	NA	NA	NA
6	Paul Lake	1991	140	5	1991-05-20	6.00	583	37	NA	NA	NA

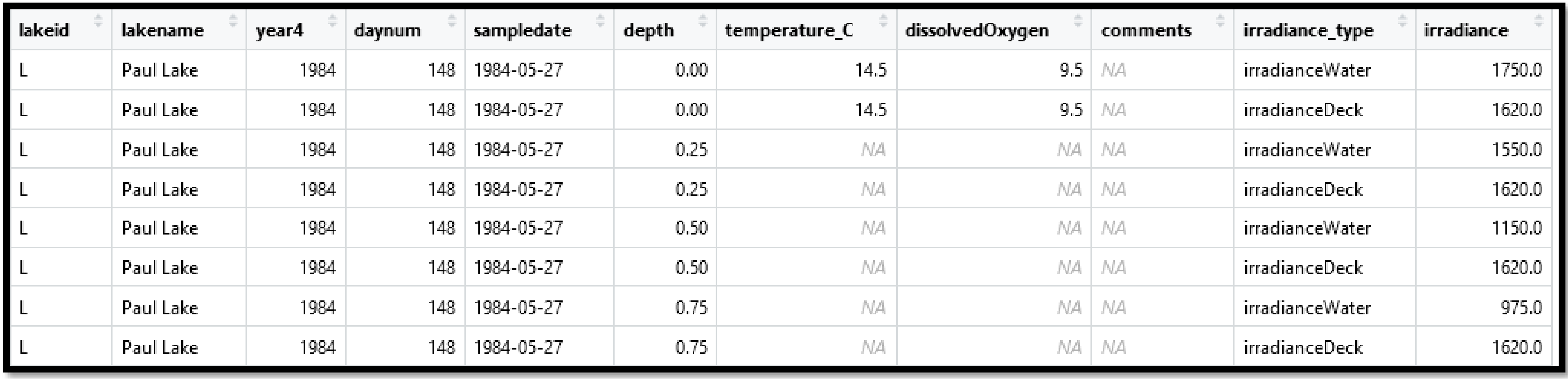


	lakename	year4	daynum	month	sampledate	depth	nutrient	concentration
1	Paul Lake	1991	140	5	1991-05-20	0.00	tn_ug	538.000
2	Paul Lake	1991	140	5	1991-05-20	0.00	tp_ug	25.000
3	Paul Lake	1991	140	5	1991-05-20	0.00	nh34	NA
4	Paul Lake	1991	140	5	1991-05-20	0.00	no23	NA
5	Paul Lake	1991	140	5	1991-05-20	0.00	po4	NA
6	Paul Lake	1991	140	5	1991-05-20	0.85	tn_ug	285.000
7	Paul Lake	1991	140	5	1991-05-20	0.85	tp_ug	14.000
8	Paul Lake	1991	140	5	1991-05-20	0.85	nh34	NA
9	Paul Lake	1991	140	5	1991-05-20	0.85	no23	NA
10	Paul Lake	1991	140	5	1991-05-20	0.85	po4	NA
11	Paul Lake	1991	140	5	1991-05-20	1.75	tn_ug	399.000
12	Paul Lake	1991	140	5	1991-05-20	1.75	tp_ug	14.000

Exercise 5: *pivot_longer()*

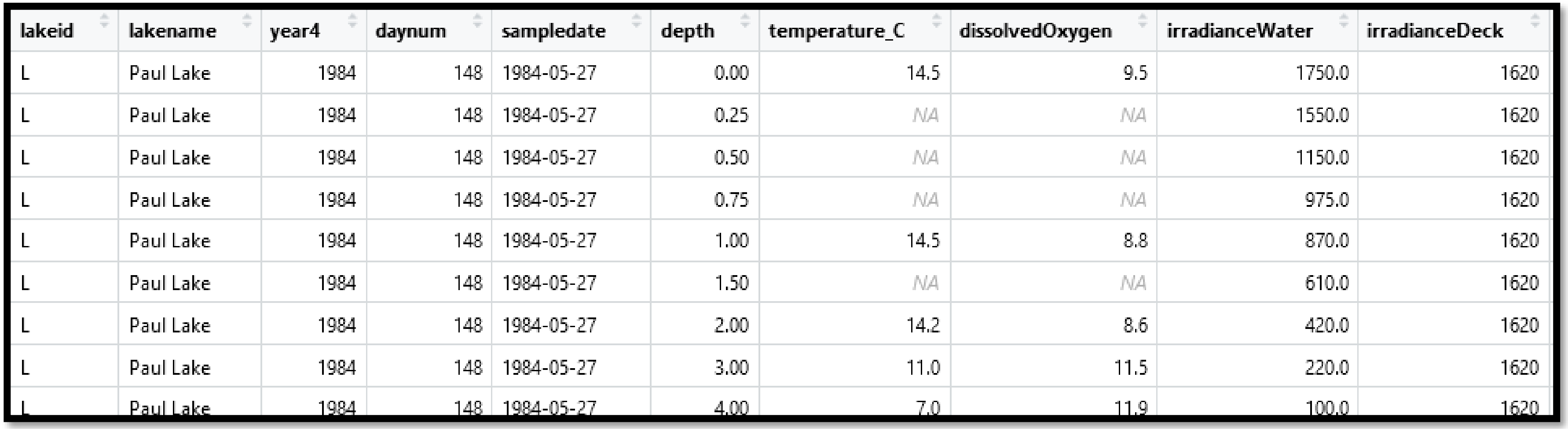


lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen	irradianceWater	irradianceDeck
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5	1750.0	1620
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA	1550.0	1620
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA	1150.0	1620
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA	975.0	1620
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8	870.0	1620
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA	610.0	1620
L	Paul Lake	1984	148	1984-05-27	2.00	14.2	8.6	420.0	1620
L	Paul Lake	1984	148	1984-05-27	3.00	11.0	11.5	220.0	1620
L	Paul Lake	1984	148	1984-05-27	4.00	7.0	11.9	100.0	1620

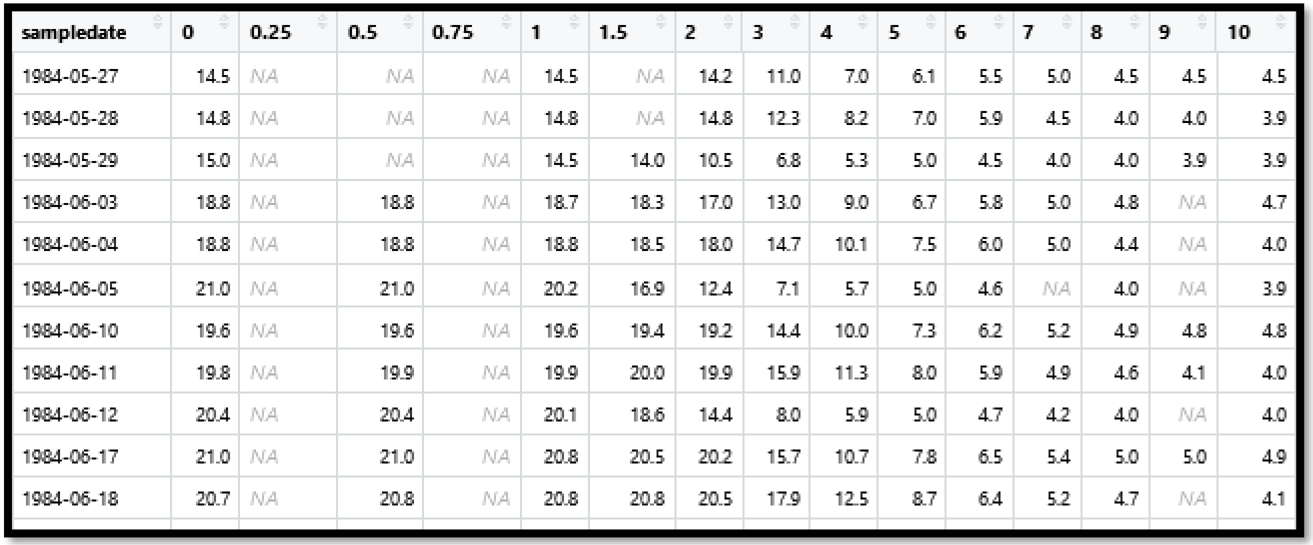


lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen	comments	irradiance_type	irradiance
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5	NA	irradianceWater	1750.0
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5	NA	irradianceDeck	1620.0
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA	NA	irradianceWater	1550.0
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA	NA	irradianceDeck	1620.0
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA	NA	irradianceWater	1150.0
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA	NA	irradianceDeck	1620.0
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA	NA	irradianceWater	975.0
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA	NA	irradianceDeck	1620.0

Exercise 5: *pivot_wider()*



lakeid	lakename	year4	daynum	sampledate	depth	temperature_C	dissolvedOxygen	irradianceWater	irradianceDeck
L	Paul Lake	1984	148	1984-05-27	0.00	14.5	9.5	1750.0	1620
L	Paul Lake	1984	148	1984-05-27	0.25	NA	NA	1550.0	1620
L	Paul Lake	1984	148	1984-05-27	0.50	NA	NA	1150.0	1620
L	Paul Lake	1984	148	1984-05-27	0.75	NA	NA	975.0	1620
L	Paul Lake	1984	148	1984-05-27	1.00	14.5	8.8	870.0	1620
L	Paul Lake	1984	148	1984-05-27	1.50	NA	NA	610.0	1620
L	Paul Lake	1984	148	1984-05-27	2.00	14.2	8.6	420.0	1620
L	Paul Lake	1984	148	1984-05-27	3.00	11.0	11.5	220.0	1620
L	Paul Lake	1984	148	1984-05-27	4.00	7.0	11.9	100.0	1620



sampledate	0	0.25	0.5	0.75	1	1.5	2	3	4	5	6	7	8	9	10
1984-05-27	14.5	NA	NA	NA	14.5	NA	14.2	11.0	7.0	6.1	5.5	5.0	4.5	4.5	4.5
1984-05-28	14.8	NA	NA	NA	14.8	NA	14.8	12.3	8.2	7.0	5.9	4.5	4.0	4.0	3.9
1984-05-29	15.0	NA	NA	NA	14.5	14.0	10.5	6.8	5.3	5.0	4.5	4.0	4.0	3.9	3.9
1984-06-03	18.8	NA	18.8	NA	18.7	18.3	17.0	13.0	9.0	6.7	5.8	5.0	4.8	NA	4.7
1984-06-04	18.8	NA	18.8	NA	18.8	18.5	18.0	14.7	10.1	7.5	6.0	5.0	4.4	NA	4.0
1984-06-05	21.0	NA	21.0	NA	20.2	16.9	12.4	7.1	5.7	5.0	4.6	NA	4.0	NA	3.9
1984-06-10	19.6	NA	19.6	NA	19.6	19.4	19.2	14.4	10.0	7.3	6.2	5.2	4.9	4.8	4.8
1984-06-11	19.8	NA	19.9	NA	19.9	20.0	19.9	15.9	11.3	8.0	5.9	4.9	4.6	4.1	4.0
1984-06-12	20.4	NA	20.4	NA	20.1	18.6	14.4	8.0	5.9	5.0	4.7	4.2	4.0	NA	4.0
1984-06-17	21.0	NA	21.0	NA	20.8	20.5	20.2	15.7	10.7	7.8	6.5	5.4	5.0	5.0	4.9
1984-06-18	20.7	NA	20.8	NA	20.8	20.8	20.5	17.9	12.5	8.7	6.4	5.2	4.7	NA	4.1