Data wrangling - base vs tidyverse vs data.table

base R operations

tidyverse operations

data.table operations

Read & write files

	read.csv() write.csv
	read_csv(), write_csv()
	fread(), fwrite()

Create data

	data.frame(x = c(1, 2), y = c("a", "b"))
	tibble(x = c(1, 2), y = c("a", "b"))
	data.table(x = c(1, 2), y = c("a", "b"))

Subset rows

```
Data prep
df <- as.data.frame(iris)
tb <- as_tibble(df)
dt <- as.data.table(df)
```

by row number	df[1:3,]
	tb %>% slice(1:3)
	dt[1:3, ,]
randomly select n rows	df[sample(nrow(df), 10),]
	tb %>% sample_n(10)
	dt[sample(.N,10)]
by variable values	df[df$Sepal.Length > 7,] df[with(df, grepl("^v", Species)),] # match a pattern in a column
	tb %>% filter(Sepal.Length > 7) tb %>% filter(str_detect(Species, "v") == TRUE)
	dt[Sepal.Length > 7,] dt[Species %like% "^v"]
sorting a table	df[order(df$Sepal.Length),] df[order(-df$Sepal.Length),] df[order(df$Species, df$Sepal.Length),]
	tb %>% arrange(Sepal.Length) tb %>% arrange(-Sepal.Length) tb %>% arrange(Species, Sepal.Length)
	dt[order(Sepal.Length),] dt[order(-Sepal.Length),] dt[order(Species, Sepal.Length),] setorder(dt, Species, Sepal.Length)
Remove duplicate rows	df[!duplicated(df),] df[!duplicated(df$Species),] # based on a variable df[!duplicated(df[,c("Species","Petal.Width")]),] # based on multiple variables
	tb %>% distinct() or distinct(tb) tb %>% distinct(Species, .keep_all= TRUE) # based on a variable tb %>% distinct(Species, Petal.Width, .keep_all= TRUE) # based on multiple variables
	unique(dt) unique(dt,by = "Species") # based on a variable unique(dt, by = c("Species", "Petal.Width")) # based on multiple variables uniqueN(dt, by = c("Species", "Petal.Width")) # return the number of unique rows

```
Using helper functions for filtering
dt[Sepal.Length %between% c(5,6)]                        # match numeric columns within a prespecified range
dt[Sepal.Width %between% list(Petal.Length,Sepal.Length)]
dt[between(Sepal.Length, 5, 6, incbounds = FALSE)]       # exclusive bounds
dt[Sepal.Length %inrange% list(3:5, 6:8)]                # between any of the intervals provided in lower,upper
dt[inrange(Sepal.Length, 3:5, 6:8, incbounds = TRUE)]    # exclusive bounds
```

Manipulate columns

```
Data prep
df <- as.data.frame(iris)
tb <- as_tibble(df)
dt <- as.data.table(df)
```

Selecting columns	df[, c(3:5)] df[, c("Petal.Width","Sepal.Width")] df[, names(df) != "Species"] df[, !names(df) %in% c("Sepal.Length", "Sepal.Width")]
	tb %>% select(3:5) tb %>% select(Petal.Width, Species) tb %>% select(Sepal.Length:Petal.Width)
	dt[, c(3:5)] dt[, .(Petal.Width, Species)]
Deleting a column	df$Sepal.Size <- NULL df[, -5]
	tb %>% select(-Species)
	dt[, Species:= NULL,]
Creating new columns	df$Sepal <- df$Sepal.Length + df$Sepal.Width
	tb %>% mutate(Sepal = Sepal.Length + Sepal.Width) # add one column tb %>% mutate(Sepal = Sepal.Length + Sepal.Width, X="x") # add multiple columns tb %>% transmute(sepal = Sepal.Length + Sepal.Width) # Drop original columns
	dt[, Sepal := Sepal.Length + Sepal.Width,] # add one column dt[, c("Sepal","X") := .(Sepal.Length + Sepal.Width, "x")]
Order columns	df[, rev(order(names(df)))]
	tb %>% select(rev(order(colnames(tb)))) tb %>% select(rev(sort(current_vars())))
	setcolorder(dt, rev(order(names(dt))))
Rename columns	colnames(df)[3:4] <- c("petal_length","petal_width")
	tb %>% rename(petal_length = Petal.Length, petal_width = Petal.Width)
	setnames(dt, c("Petal.Length","Petal.Width"), c("petal_length","petal_width"))

```
tidyselect - tidyverse helper functions for select
tb %>% select(starts_with("Sepal"))
tb %>% select(ends_with("Length"))
tb %>% select(contains("Length"))
tb %>% select(matches("al"))
tb %>% select(matches("[pt]al"))
billboard %>% select(num_range("wk", 10:15))
tb %>% select(everything()) # select all variables
```

```
base help function for all:
cols = paste0(c("Sepal","Petal"), ".Length")
cols = grep("^Sepal", names(df))
cols = grep("Length$", names(df))
cols = grep("[pt]al", names(df))
df[, cols]
tb %>% select(cols)
dt[, ..cols]
```

```
specific for tidyverse
tb <- tb %>% separate(car, c("name1","name2"), " ")
tb <- tb %>% unite("car",name1:name2, sep="_",na.rm=T)
separate_rows(tb, car,convert = TRUE, sep="_")
```

BY lixuexile@gmail.com Updated: 2022-09

Data wrangling - base vs tidyverse vs data.table

base R operations

tidyverse operations

data.table operations

Reshaping

```
Data prep
df <- mtcars[, c(1:2, 4, 9)]
df$car <- rownames(mtcars)
rownames(df) <- NULL
tb <- as_tibble(df)
dt <- as.data.table(df)
```

Reshape to long format

```
df.l <- reshape(df, idvar = "car",
                      times = names(df)[names(df) != "car"],
                      timevar = "variable", v.names="value",
                      varying = list(names(df) [names(df) != "car"]),
                      direction = "long")
```

```
tb.l <- tb %>% pivot_longer(-car,
                            names_to = "variable",
                            values_to = "value")
```

```
dt.l <- melt(dt, id.vars = c("car"),
variable.name = "variable",
               value.name = "value")
```

Reshape to wide format

```
df.w <- reshape(df.l, idvar = "car",
                timevar="variable",
                v.names="value",
                direction = "wide")
```

```
tb.w <- pivot_wider(tb.l,
                    names_from = variable,
                    values_from = value)
```

```
dt.w <- dcast(dt.l, car ~ variable,
              value.var="value")
```

sorting a table

```
df[order(df$mpg), ]
df[order(-df$mpg), ]
df[order(df$cyl, df$mpg),]
```

```
tb %>% arrange(mpg)
tb %>% arrange(-mpg)
tb %>% arrange(cyl, mpg)
```

```
dt[order(mpg), ]
dt[order(-mpg), ]
dt[order(cyl, mpg), ]
setorder(dt, cyl, mpg)
```

Group & summarize

```
Data prep
df <- mtcars[, c(1:2, 4, 9)]
df$car <- rownames(mtcars)
rownames(df) <- NULL
tb <- as_tibble(df)
dt <- as.data.table(df)
```

Summarizing all columns

```
apply(df, 2, max)
```

```
summarise_each(tb, max)
```

```
dt[ , lapply(.SD,  max), ]
```

Summarizing specific columns

```
apply(df[ , c("mpg","hp")], 2, median)
```

```
summarise(tb, mpg = median(mpg), hp = mean(hp))
```

```
dt[ , .(mpg = median(mpg), hp = mean(hp)), ]
```

Summarizing columns by group

```
data.frame(cyl = aggregate(df$mpg, list(df$cyl), mean)[,1],
           mpg = aggregate(df$mpg, list(df$cyl), mean)$x,
           hp = aggregate(df$hp, list(df$cyl), max)$x,
           n = aggregate(df$hp, list(df$cyl), length)$x)
```

```
tb %>%
    group_by(cyl)  %>%
    summarise(mpg = mean(mpg), hp = max(hp), n = n())
tb %>% group_by(cyl) %>% tally()
tb %>% count(cyl)
```

```
dt[ , .(mpg = mean(mpg), hp = max(hp), n = .N), by=cyl]
```

Combine Data Sets

```
Data prep
df.lu <- data.frame(x = c(0,1),
y = c("automatic", "manual"))
tb.lu <- as_tibble(df.lu)
dt.lu <- as.data.table(df.lu)
```

```
merge(df, df.lu, by.x = "am",by.y="x", all.x = TRUE)
rbind(df[1:10,], df[20:30,])
cbind(df[,1:3], df[,c(5,4)])
```

```
left_join(tb, tb.lu, by = c("am" = "x"))
y = data.frame(x1 = c("A","B","C"), x2 = c(1,2,3))
z = data.frame(x1 = c("B","C","D"), x2 = c(2,3,4))
intersect(y,z)
union(y,z)
setdiff(y,z)
```

```
dt[dt.lu, on = c("am" = "x")]
```

Chaining commands

```
Data prep
df <- mtcars[, c(1:2, 4, 9)]
df$car <- rownames(mtcars)
rownames(df) <- NULL
tb <- as_tibble(df)
dt <- as.data.table(df)
```

```
df$gpm <- 1/df$mpg
. <- df[ , c("cyl", "gpm")]
. <- aggregate(., list(df$cyl), median)
.$Group.1 <- NULL
.[order(-.$gpm), ]
```

```
tb %>%
    mutate(gpm = 1/mpg) %>%
    group_by(cyl) %>%
    summarise(gpm = median(gpm)) %>%
    arrange(-gpm)
```

```
dt[ , gpm := 1/mpg, ][
    order(-gpm), .(gpm = median(gpm)),
    by = cyl]
```

Summary of key functions

Environment	base	tidyverse	data.table
Supported data class(es)	data.frame	data.frame, tibble	data.table
Reading data	read.csv	read_csv	fread
Subset by column	[, ...]	select()	[,... ,]
Subset by rows	[... ,]	filter()	[... , ,]
Create new column	df$y = ...	mutate(tb, y = ...)	[, y := ...,]
Delete a column	df$y = NULL	select(tb, -y)	[, y := NULL,]
Summarize	apply(df[, y], 2, ...)	summarise()	[, ...(y),]
Grouping	aggregate()	group_by()	[, , by = ...]
Pivot to long	reshape()	pivot_longer()	melt()
Pivot to wide	reshape()	pivot_wider()	dcast()
Joining tables	merge()	left_join()	DT1[DT2, on = ...]

Manny Gimond, https://mgimond.github.io/rug_2019_12/Index.html
data.table – cheat sheet: https://raw.githubusercontent.com/rstudio/cheatsheets/main/datatable.pdf
dplyr and tidyr – cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

BY lixuexile@gmail.com Updated: 2022-09