R Design Patterns: Base-R vs. Tidyverse

Norman Matloff
Dept. of Computer Science, University of California, Davis

August 17, 2022

This document enables the reader to see at a glance the difference between base-R and the Tidyverse in common R design settings.

All examples use R's built-in datasets. After e.g., changing a data frame, it is restored for the next example, e.g. **data(mtcars)**.

As this document is aimed at comparing base-R and the tidyverse in terms of teaching new R learners, advanced functions from either base-R or the tidyverse are excluded here.

More and more examples will be added over time.

Reading a specific cell in a data frame

```
mtcars$mpg[3]
```

```
mtcars %>% select(mpg) %>%
   filter(row_number() == 3)
```

Adding a column to a data frame

```
mtcars$hwratio
  <- mtcars$hp / mtcars$wt
```

```
mtcars
  %>% mutate(hwratio=hp/wt) -> mtcars
```

Deleting a column from a data frame

```
mtcars$carb <- NULL
```

```
mtcars %>% select(-carb) -> mtcars
```

Deleting multiple columns from a data frame

```
mtcars[c('drat','carb')]
  <- NULL
```

```
mtcars %>% select(-c(drat,carb))
  -> mtcars
```

Binary categorization on a vector

```
NileHiLow <-
   ifelse(Nile >= 1000,
     'high','low')
```

```
Nile %>% as.data.frame %>%
   mutate(
     HighLow = case_when
     (x < 1000~'low',
      x >= 1000~'high')
   ) %>%
   select(HighLow) %>%
   as.vector -> HighLow
```

The step of conversion back to a vector at the end is needed for many R packages in which vector input is required.