101_wk5_iteration_with_purrr

Seung Hyun Sung

11/15/2021

DS4B 101-R: R FOR BUSINESS ANALYSIS —-

ITERATION WITH PURRR —-

```
library(readxl)
library(tidyverse)
library(tidyquant)
library(lubridate)
library(broom)

bike_orderlines_tbl <- read_rds("~/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_

glimpse(bike_orderlines_tbl)
```

```
## Rows: 15,644
## Columns: 13
## $ order_date     <dttm> 2011-01-07, 2011-01-07, 2011-01-10, 2011-01-10, 2011-0~
## $ order_id       <dbl> 1, 1, 2, 2, 3, 3, 3, 3, 3, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7~
## $ order_line     <dbl> 1, 2, 1, 2, 1, 2, 3, 4, 5, 1, 1, 2, 3, 4, 1, 2, 3, 4, 1~
## $ quantity       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1~
## $ price          <dbl> 6070, 5970, 2770, 5970, 10660, 3200, 12790, 5330, 1570,~
## $ total_price    <dbl> 6070, 5970, 2770, 5970, 10660, 3200, 12790, 5330, 1570,~
## $ model          <chr> "Jekyll Carbon 2", "Trigger Carbon 2", "Beast of the Ea~
## $ category_1     <chr> "Mountain", "Mountain", "Mountain", "Mountain", "Road",~
## $ category_2     <chr> "Over Mountain", "Over Mountain", "Trail", "Over Mounta~
## $ frame_material <chr> "Carbon", "Carbon", "Aluminum", "Carbon", "Carbon", "Ca~
## $ bikeshop_name  <chr> "Ithaca Mountain Climbers", "Ithaca Mountain Climbers",~
## $ city           <chr> "Ithaca", "Ithaca", "Kansas City", "Kansas City", "Loui~
## $ state          <chr> "NY", "NY", "KS", "KS", "KY", "KY", "KY", "KY", "KY", "~
```

1.0 PRIMER ON PURRR —-

Programmatically getting Excel files into R

```
excel_paths_tbl <- fs::dir_info("~/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_

paths_chr <- excel_paths_tbl %>% pull(path)
```

What Not To Do: Don't use for loops

```
excel_list <- list()
for(path in paths_chr){
    excel_list[[path]] <- read_excel(path)
}
```

```
## New names:
## * `` -> ...1
```

```
excel_list
```

```
## $`/Users/seunghyunsung/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/bike
## # A tibble: 97 x 4
##    bike.id model                         description                price
##      <dbl> <chr>                         <chr>                      <dbl>
##  1       1 Supersix Evo Black Inc.       Road - Elite Road - Carbon 12790
##  2       2 Supersix Evo Hi-Mod Team      Road - Elite Road - Carbon 10660
##  3       3 Supersix Evo Hi-Mod Dura Ace 1 Road - Elite Road - Carbon  7990
##  4       4 Supersix Evo Hi-Mod Dura Ace 2 Road - Elite Road - Carbon  5330
##  5       5 Supersix Evo Hi-Mod Utegra     Road - Elite Road - Carbon  4260
##  6       6 Supersix Evo Red              Road - Elite Road - Carbon  3940
##  7       7 Supersix Evo Ultegra 3        Road - Elite Road - Carbon  3200
##  8       8 Supersix Evo Ultegra 4        Road - Elite Road - Carbon  2660
##  9       9 Supersix Evo 105              Road - Elite Road - Carbon  2240
## 10      10 Supersix Evo Tiagra           Road - Elite Road - Carbon  1840
## # ... with 87 more rows
##
## $`/Users/seunghyunsung/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/bike
## # A tibble: 30 x 3
##    bikeshop.id bikeshop.name                location
##          <dbl> <chr>                        <chr>
##  1           1 Pittsburgh Mountain Machines Pittsburgh, PA
##  2           2 Ithaca Mountain Climbers     Ithaca, NY
##  3           3 Columbus Race Equipment      Columbus, OH
##  4           4 Detroit Cycles               Detroit, MI
##  5           5 Cincinnati Speed             Cincinnati, OH
##  6           6 Louisville Race Equipment    Louisville, KY
##  7           7 Nashville Cruisers           Nashville, TN
##  8           8 Denver Bike Shop             Denver, CO
##  9           9 Minneapolis Bike Shop        Minneapolis, MN
## 10          10 Kansas City 29ers            Kansas City, KS
## # ... with 20 more rows
##
## $`/Users/seunghyunsung/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/orde
## # A tibble: 15,644 x 7
##    ...1  order.id order.line order.date          customer.id product.id quantity
##    <chr>    <dbl>      <dbl> <dttm>                    <dbl>      <dbl>    <dbl>
##  1 1            1          1 2011-01-07 00:00:00           2         48        1
##  2 2            1          2 2011-01-07 00:00:00           2         52        1
##  3 3            2          1 2011-01-10 00:00:00          10         76        1
##  4 4            2          2 2011-01-10 00:00:00          10         52        1
##  5 5            3          1 2011-01-10 00:00:00           6          2        1
##  6 6            3          2 2011-01-10 00:00:00           6         50        1
##  7 7            3          3 2011-01-10 00:00:00           6          1        1
##  8 8            3          4 2011-01-10 00:00:00           6          4        1
##  9 9            3          5 2011-01-10 00:00:00           6         34        1
## 10 10           4          1 2011-01-11 00:00:00          22         26        1
## # ... with 15,634 more rows
```

What to Do: Use map()

Reading Excel Sheets

2.0 MAPPING DATA FRAMES —-

2.1 Column-wise Map —-

2.2 Map Variants —-

2.3 Row-wise Map —-

3.0 NESTED DATA —-

Unnest

Nest

Mapping Nested List Columns

4.0 MODELING WITH PURRR —-

4.1 Time Series Plot —-

- What if we wanted to approximate the 3 month rolling average with a line?

- We can use a smoother

Code comes from 04_functions_iteration/01_functional_programming

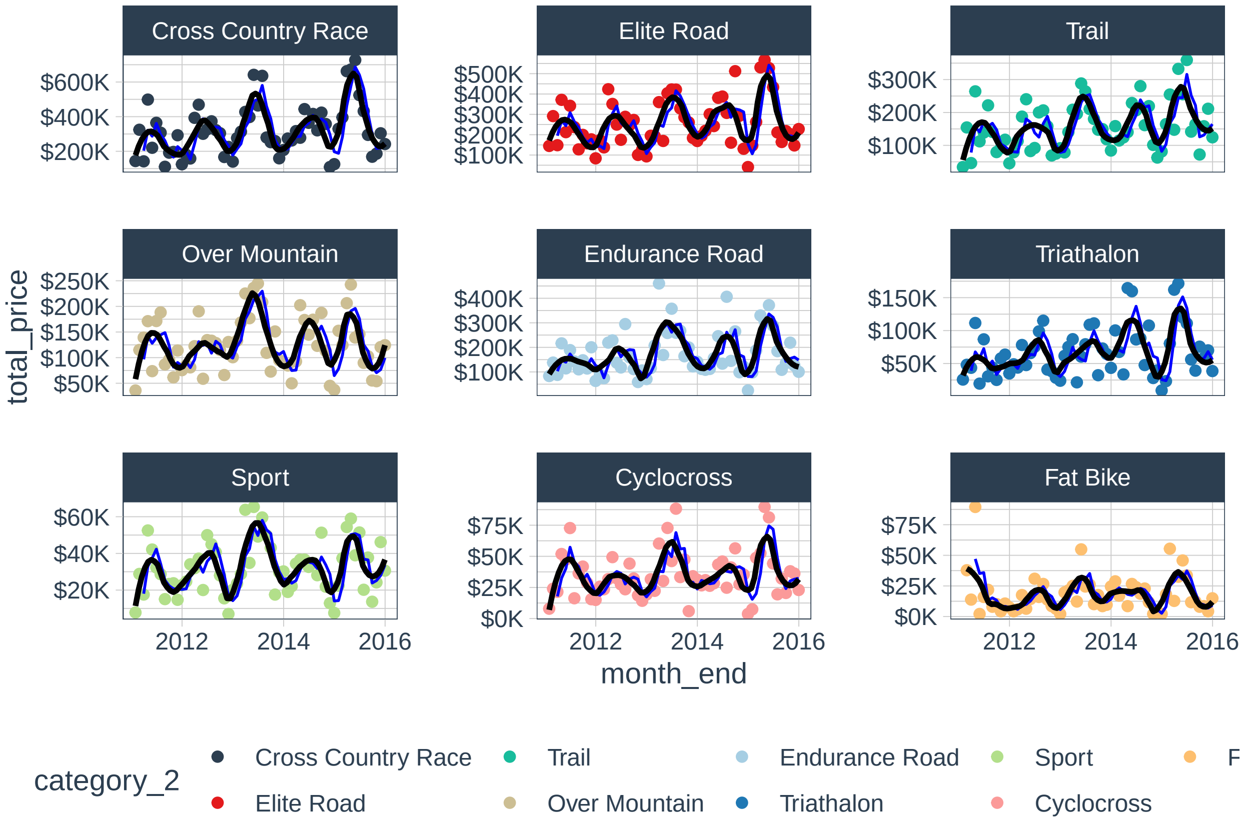
```
rolling_avg_3_tbl <- bike_orderlines_tbl %>%
    select(order_date, category_1, category_2, total_price) %>%

    mutate(order_date = ymd(order_date)) %>%
    mutate(month_end = ceiling_date(order_date, unit = "month") - period(1, unit = "days")) %>%
    group_by(category_1, category_2, month_end) %>%
    summarise(
        total_price = sum(total_price)
    ) %>%
    mutate(rolling_avg_3 = rollmean(total_price, k = 3, na.pad = TRUE, align = "right")) %>%
    ungroup() %>%

    mutate(category_2 = as_factor(category_2) %>% fct_reorder2(month_end, total_price))
```

```
## ‘summarise()‘ has grouped output by ’category_1’, ’category_2’. You can override using the ‘.groups‘
```

```
rolling_avg_3_tbl %>%

    ggplot(aes(month_end, total_price, color = category_2)) +

    # Geometries
    geom_point() +
    geom_line(aes(y = rolling_avg_3), color = "blue", linetype = 1) +
    facet_wrap(~ category_2, scales = "free_y") +

    # Add Loess Smoother
    geom_smooth(method = "loess", se = FALSE, span = 0.2, color = "black") +

    # Formatting
    theme_tq() +
    scale_color_tq() +
    scale_y_continuous(labels = scales::dollar_format(scale = 1e-3, suffix = "K"))
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



4.2 Modeling Primer —-

Data Preparation

Making a loess model

Working With Broom

Visualizing results

4.3 Function To Return Fitted Results —-

4.4 Test Function on Single Element —-

4.5 Map Function to All Categories —-

Map Functions

Visualize Results