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()

purrr::map : designed for iteration

Super powerful!!

Anonymous function & functional operation

- anonymous function: An anonymous function is a function that is not stored in a program file, but is associated with a variable whose data type is function_handle . Anonymous functions can accept multiple inputs and return one output.
- In comparison to functional operation, anonymous function is little more customisable and less typing.
- For anonymous function must remember to place (.)

```
excel_list_2 <- paths_chr %>%
    map(read_excel) %>%
    # naming the each list of the data frame
    setNames(paths_chr)
```

```
## New names:
## * ‘‘ -> ...1
```

```
# Different variance!

# Method 1. Function specified with function()
paths_chr %>%
    map(function(x) read_excel(path = x)) %>%
    setNames(paths_chr)
```

```
## New names:
## * ‘‘ -> ...1
```

```
## $`/Users/seunghyunsung/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/bikes
## # 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/bikes
## # 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
```

```r
# Method 2. anonymous function
paths_chr %>%
    map(~read_excel(.)) %>%
    setNames(paths_chr)
```

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

```
## $`/Users/seunghyunsung/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/bikes
## # 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/bikes
## # 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
```

Reading Excel Sheets

```
excel_sheets("~/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw/bikes.xlsx") %
    map(~ read_excel(path = "~/Desktop/University_business_science/DS4B_101/00_data/bike_sales/data_raw
```

```
## [[1]]
## # 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
```

2.0 MAPPING DATA FRAMES —-

2.1 Column-wise Map —-

- Map functions apply a function iteractively to each element of a list or vector.
- date frame is actually a list!!

```
# bike_orderlines_tbl %>% as.list()

bike_orderlines_tbl %>% is.list()
```

```
## [1] TRUE
```

```
bike_orderlines_tbl %>%
    map(~class(.)[1]) %>% unlist()
```

```
##     order_date       order_id     order_line       quantity          price
##      "POSIXct"      "numeric"      "numeric"      "numeric"      "numeric"
##    total_price          model     category_1     category_2 frame_material
##      "numeric"    "character"    "character"    "character"    "character"
##  bikeshop_name           city          state
##    "character"    "character"    "character"
```

```
bike_orderlines_tbl %>%
    select(where(is.numeric)) %>%
    map(~mean(.)) %>% unlist()
```

```
##    order_id  order_line    quantity       price total_price
##  997.953081    8.471619    1.289440 3521.110969 4540.547814
```

2.2 Map Variants —-

- map: list
- map_chr: character vector
- map_dbl: double(numeric) vector
- map_dfc: data frame (column bind)
- map_int: integer vector

- map_lgl: logical vector
- walk: triggers side effects, returns the input invisibly

```
# Charcter map
bike_orderlines_tbl %>%
    # these are named character vector
    map_chr(~class(.)[1])
```

```
##     order_date       order_id     order_line       quantity          price
##      "POSIXct"      "numeric"      "numeric"      "numeric"      "numeric"
##    total_price          model     category_1     category_2 frame_material
##      "numeric"    "character"    "character"    "character"    "character"
##  bikeshop_name           city          state
##    "character"    "character"    "character"
```

```
# Data Frame map
bike_orderlines_tbl %>%
    map_df(~ class(.)[1])
```

```
## # A tibble: 1 x 13
##   order_date order_id order_line quantity price   total_price model   category_1
##   <chr>      <chr>    <chr>      <chr>    <chr>   <chr>       <chr>   <chr>
## 1 POSIXct    numeric  numeric    numeric  numeric numeric     charac~ character
## # ... with 5 more variables: category_2 <chr>, frame_material <chr>,
## #   bikeshop_name <chr>, city <chr>, state <chr>
```

```
# Data Frame map + gather
bike_orderlines_tbl %>%
    map_df(~ class(.)[1]) %>%
    gather()
```

```
## # A tibble: 13 x 2
##    key            value
##    <chr>          <chr>
##  1 order_date     POSIXct
##  2 order_id       numeric
##  3 order_line     numeric
##  4 quantity       numeric
##  5 price          numeric
##  6 total_price    numeric
##  7 model          character
##  8 category_1     character
##  9 category_2     character
## 10 frame_material character
## 11 bikeshop_name  character
## 12 city           character
## 13 state          character
```

```
# Observation length map
bike_orderlines_tbl %>%
    map_df(~length(.)) %>%
    gather(key = variable, value = length)
```

```
## # A tibble: 13 x 2
##    variable       length
##    <chr>           <int>
##  1 order_date      15644
##  2 order_id        15644
##  3 order_line      15644
##  4 quantity        15644
##  5 price           15644
##  6 total_price     15644
##  7 model           15644
##  8 category_1      15644
##  9 category_2      15644
## 10 frame_material  15644
## 11 bikeshop_name   15644
## 12 city            15644
## 13 state           15644
```

```
# mean value map
bike_orderlines_tbl %>%
    map_df(~mean(.)) %>%
    gather(key = variable, value = mean)
```

```
## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning in mean.default(.): argument is not numeric or logical: returning NA

## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
## # A tibble: 13 x 2
##    variable                mean
##    <chr>                  <dbl>
##  1 order_date     1377841483.
##  2 order_id             998.
##  3 order_line             8.47
##  4 quantity               1.29
##  5 price               3521.
##  6 total_price         4541.
##  7 model                NA
##  8 category_1           NA
##  9 category_2           NA
## 10 frame_material       NA
## 11 bikeshop_name        NA
## 12 city                   NA
## 13 state                  NA
```

```
# NA value map
bike_orderlines_tbl %>%
    map_df(~sum(is.na(.))/length(.)) %>%
    gather(key = variable, value = na)
```

```
## # A tibble: 13 x 2
##    variable          na
##    <chr>          <dbl>
##  1 order_date         0
##  2 order_id           0
##  3 order_line         0
##  4 quantity           0
##  5 price              0
##  6 total_price        0
##  7 model              0
##  8 category_1         0
##  9 category_2         0
## 10 frame_material     0
## 11 bikeshop_name      0
## 12 city               0
## 13 state              0
```

2.3 Row-wise Map —-

- keeping excel file organised as tibble
- This is an alternative way to read all the file from the directory
- This is the concept of nesting. Very powerful when it is utilised into modelling. Topic of the next section!

```
excel_tbl <- excel_paths_tbl %>%
    select(path) %>%
    mutate(data = path %>% map(read_excel))
```

```
## New names:
## * ‘‘ -> ...1
```

```
excel_tbl
```

```
## # A tibble: 3 x 2
##   path                                                              data
##   <fs::path>                                                        <list>
## 1 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [97 x 4~
## 2 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [30 x 3~
## 3 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [15,644~
```

3.0 NESTED DATA —-

Unnest

unnest: unnests a nested data frame converting tibbles burried within list-columns to a single level tibble

- .id = “ID”: assign id number with respect to the individual tibbles nested.
- Very important for nesting it back!!
- Similarly to gather and spread: where mutate row number was a key to return back gather than spread,

```
excel_tbl
```

```
## # A tibble: 3 x 2
##   path                                                       data
##   <fs::path>                                                 <list>
## 1 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [97 x 4~
## 2 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [30 x 3~
## 3 /Users/seunghyunsung/Desktop/University_business_science/DS4~ <tibble [15,644~
```

```
excel_tbl$data
```

```
## [[1]]
## # 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
##
## [[2]]
## # 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
##
## [[3]]
## # 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
```

```
# pull second data
excel_tbl$data[[2]]
```

```
## # 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
```

```
# unnests nested data frame
# brings all the data, expanded the tibbles organised into single data frame
excel_tbl_unnested <- excel_tbl %>%
    unnest_legacy(data, .id = "ID")
```

```
## New names:
## * ...1 -> ...9
```

```
## New names:
## * ...9 -> ...10
```

```
# these data frames originally nested contains different information(features) hence it unnests into si
#   %>% View()
```

Nest

```
excel_tbl_nested <- excel_tbl_unnested %>%
    group_by(ID, path) %>%
    nest()
```

Mapping Nested List Columns

```
# nested excel data
excel_tbl$data[[1]]
```

```
## # 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
```

```
# nested -> unnested -> nested back
excel_tbl_nested$data[[1]] %>%
    # select deals with columns: all the columns that
    # is not all in NA will be droped
    select_if(~!is.na(.) %>% all())
```

```
## # 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
```

```
# Quick example: all()
# contains 5 NA and 3
x <- c(rep(NA_real_, 5), 3)
is.na(x)
```

```
## [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
```

```
is.na(x) %>% all()
```

```
## [1] FALSE
```

```
# contains only NA
y <- rep(NA_real_, 5)
is.na(y)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE
```

```
is.na(y) %>% all()  # Is Y all NA vectors? FALSE
```

```
## [1] TRUE
```

```
!is.na(y) %>% all() # Is Y all not NA vectors? TRUE
```

```
## [1] FALSE
```

Method 1: Creating a function outside of purrr::map()

```
# step 1: create a function that can be mapped to one element
select_non_na_columns <- function(data){
    data %>%
        select_if(~!is.na(.) %>% all())
}

# step 2: Extract an element, and test the function
excel_tbl_nested$data[[1]] %>%
    select_non_na_columns()
```

```
## # 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
```

```
# Step 3: Use mutate() + map()
excel_tbl_nested_fixed <- excel_tbl_nested %>%
    # Remember this nested tibble (tibble inside the tibble) are row operation
    # Hence the mutate function works beautifully
    # Here: create new nested set of tibbles (map)
    mutate(data_fixed = data %>% map(select_non_na_columns))

# Step 4: Check
excel_tbl_nested_fixed$data_fixed[[1]]
```

```
## # 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
```

4.0 MODELING WITH PURRR —-

- Apply modeling functions at scale

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‘
```

The 3 month moving (rolling) average looks choppy: it was to get the idea of the trend.

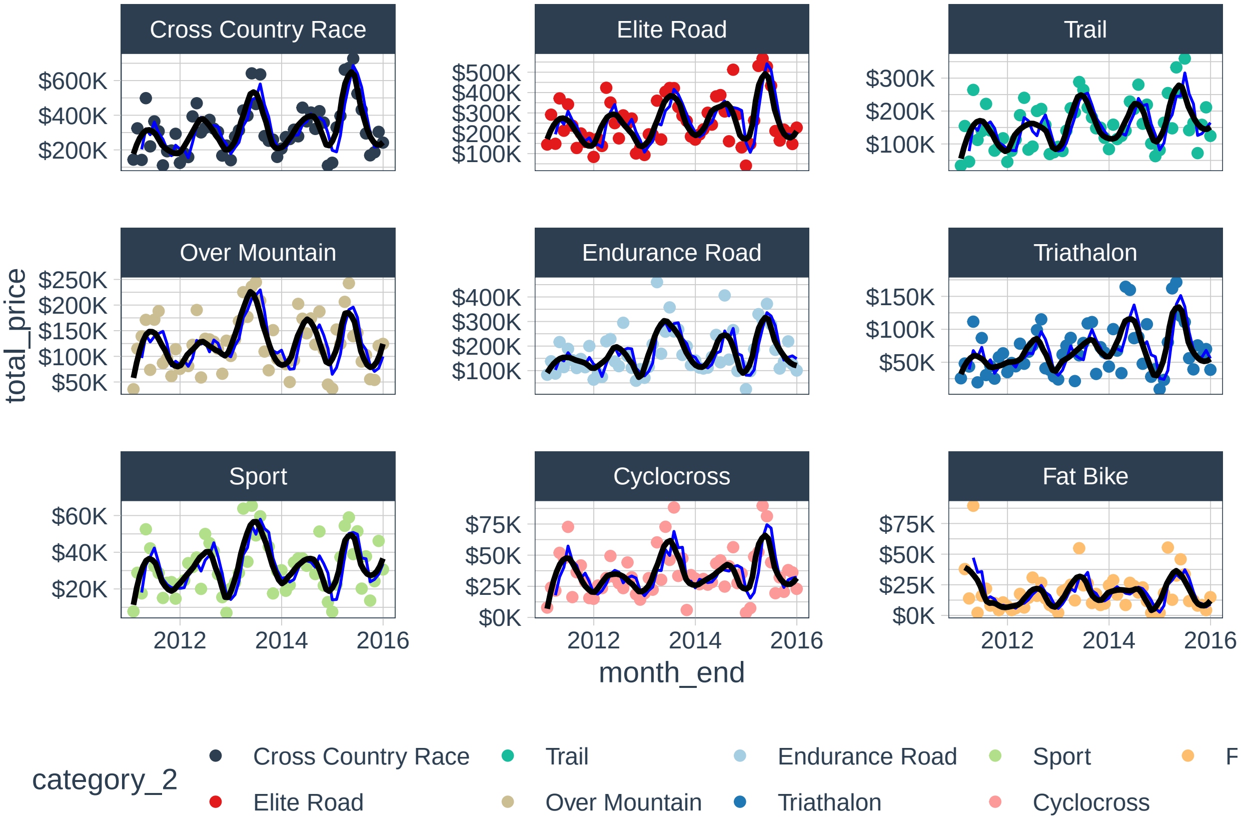
- In statistics, a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. It is also called a moving mean (MM) or rolling mean and is a type of finite impulse response filter.
- 3 month MA are not centered or aligned + there are missing points: There are down sides.
    - usually align to the right hence does not follow the trend appropriately.

Often we would like to use smoother other than 3 month rolling average.

```
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
    # [1] The smoother does not actually follow the trend, must adjust the span argument!
    # geom_smooth(method = "loess", se = FALSE) +
    geom_smooth(method = "loess", se = FALSE, span = 0.2, colour = "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).
```

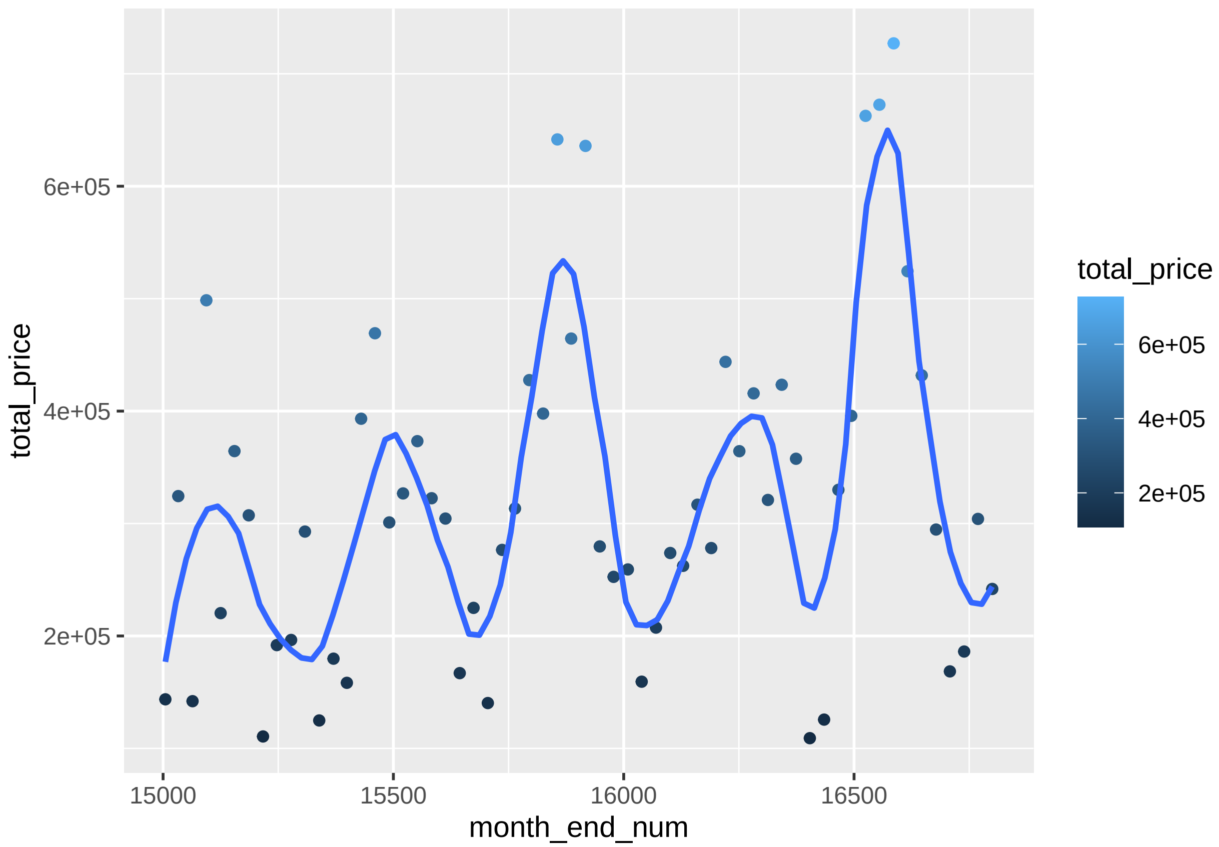


4.2 Modeling Primer —-

Data Preparation

```
sales_by_m_cross_country_tbl <- rolling_avg_3_tbl %>%
  filter(category_2 == "Cross Country Race") %>%

  select(month_end, total_price) %>%
  # smoother does not work with date data
  mutate(month_end_num = as.numeric(month_end))

sales_by_m_cross_country_tbl %>%
  ggplot(aes(x = month_end_num, y = total_price)) +
  geom_point(aes(colour = total_price)) +
  geom_smooth(method = "loess", se = FALSE, span = 0.2)
```

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



Making a loess model

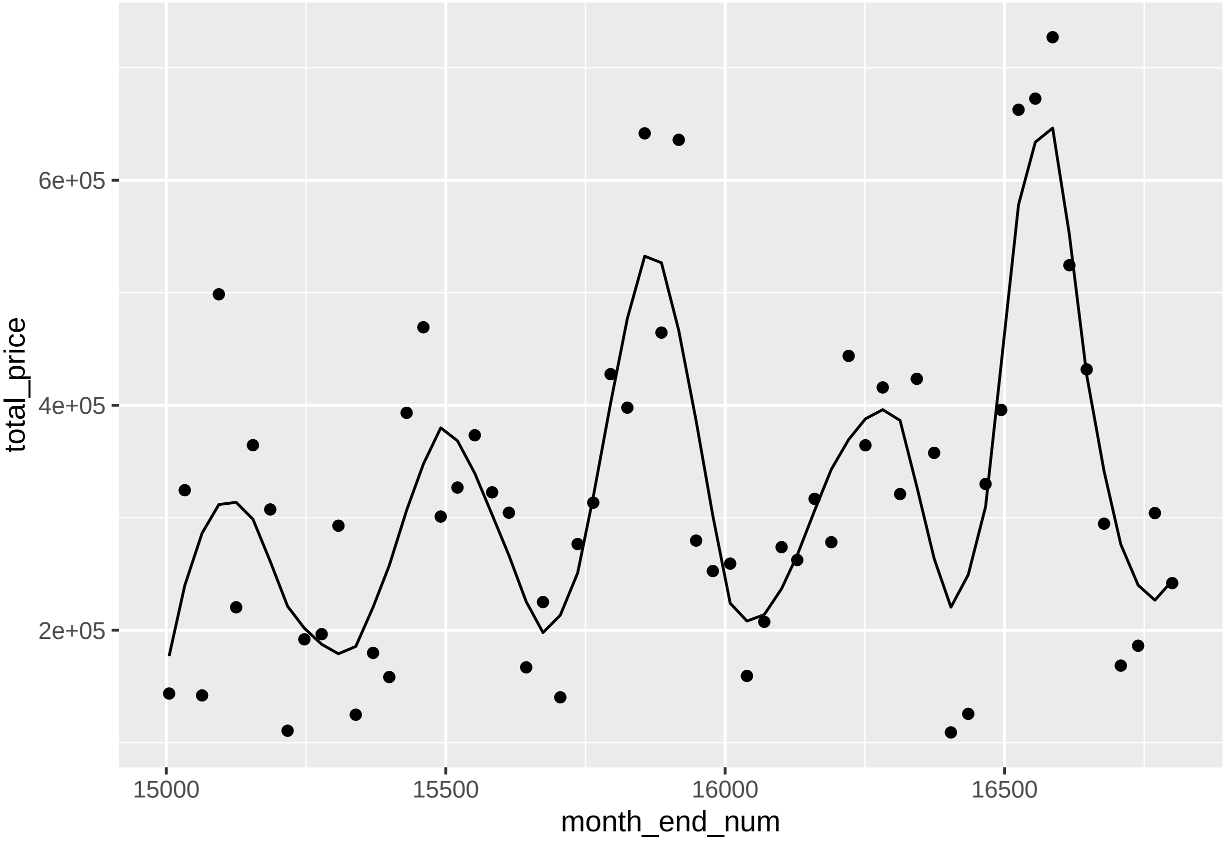
- Smoothing data using local regression
- Fit a polynomial surface determined by one or more numerical predictors, using local fitting.

```
?loess

fit_loess_cross_country <- sales_by_m_cross_country_tbl %>%
  # notice here this is not tidy function, the "data" is second argument of loess function
  # hence, you will get error without data = . argument.
  loess(total_price ~ month_end_num, data = ., span = 0.2)
```

Working With Broom

- broom has three useful functions stored
  - augment(): retuns model fitted values, residuals, and standard erors in data frame format
  - tidy() :
  - glance():

```
# we now obtained the smooth data points using loess + broom::augment function
fit_loess_cross_country %>%
  # fitted, standard error, residuals from model
  broom::augment() %>%
  # Visualising result
  ggplot(aes(x = month_end_num, y = total_price)) +
  geom_point() +
  geom_line(aes(y = .fitted))
```



4.3 Step 1: Function To Return Fitted Results —-

Pro Tip: When making functions, save some testable data as each argument so you can interactively test the function while you build it.

```
# group_by {category_1, category_2} gives 9 total categories
rolling_avg_3_tbl %>%
  distinct(category_1, category_2)
```

```
## # A tibble: 9 x 2
##   category_1 category_2
##   <chr>      <fct>
## 1 Mountain   Cross Country Race
## 2 Mountain   Fat Bike
## 3 Mountain   Over Mountain
## 4 Mountain   Sport
## 5 Mountain   Trail
## 6 Road       Cyclocross
## 7 Road       Elite Road
## 8 Road       Endurance Road
## 9 Road       Triathalon
```

```
rolling_avg_3_tbl_nested <- rolling_avg_3_tbl %>%
  group_by(category_1, category_2) %>%
  nest()

rolling_avg_3_tbl_nested$data[[1]]
```

```
## # A tibble: 60 x 3
##    month_end  total_price rolling_avg_3
##    <date>           <dbl>         <dbl>
##  1 2011-01-31      143660           NA
##  2 2011-02-28      324400           NA
##  3 2011-03-31      142000       203353.
##  4 2011-04-30      498580       321660
##  5 2011-05-31      220310       286963.
##  6 2011-06-30      364420       361103.
##  7 2011-07-31      307300       297343.
##  8 2011-08-31      110600       260773.
##  9 2011-09-30      191870       203257.
## 10 2011-10-31      196440       166303.
## # ... with 50 more rows
```

```
### Pro Tip: here
data <- rolling_avg_3_tbl_nested$data[[1]]

tidy_loess <- function(data, span = 0.2){

  data_formatted <- data %>%
    select(month_end, total_price) %>%
    mutate(month_end_num = as.numeric(month_end))

  fit_loess <- loess(formula = total_price ~ month_end_num,
                     data    = data_formatted,
                     span    = 0.2)
  output_tbl <- fit_loess %>%
    broom::augment() %>%
    select(.fitted)

  return(output_tbl)
}

tidy_loess(data)
```

```
## # A tibble: 60 x 1
##    .fitted
##      <dbl>
##  1 176998.
##  2 239802.
##   3 286279.
##   4 311685.
##   5 313621.
##   6 298642.
##   7 261073.
##   8 221223.
##   9 201690.
##  10 187415.
## # ... with 50 more rows
```

4.4 Step 2: Test Function on Single Element —-

```
# test whether the tidy_loess() function operates well with nested tibble
rolling_avg_3_tbl_nested$data[[6]] %>%
  tidy_loess()
```

```
## # A tibble: 60 x 1
##    .fitted
##      <dbl>
##  1   6996.
##  2  21804.
##  3  34076.
##  4  42266.
##  5  46788.
##  6  47804.
##  7  43823.
##  8  37541.
##  9  31132.
## 10  25190.
## # ... with 50 more rows
```

4.5 Step 3: Map Function to All Categories —-

Map Functions

```
loess_tbl_nested <- rolling_avg_3_tbl_nested %>%
  mutate(fitted = data %>% map(tidy_loess))

loess_tbl_nested$fitted[[1]]
```

```
## # A tibble: 60 x 1
##    .fitted
##      <dbl>
##  1 176998.
##  2 239802.
##  3 286279.
##  4 311685.
##  5 313621.
##  6 298642.
##  7 261073.
##  8 221223.
##  9 201690.
## 10 187415.
## # ... with 50 more rows
```

```
loess_tbl_nested %>%
  unnest()
```

```
## Warning: ‘cols‘ is now required when using unnest().
## Please use ‘cols = c(data, fitted)‘
```

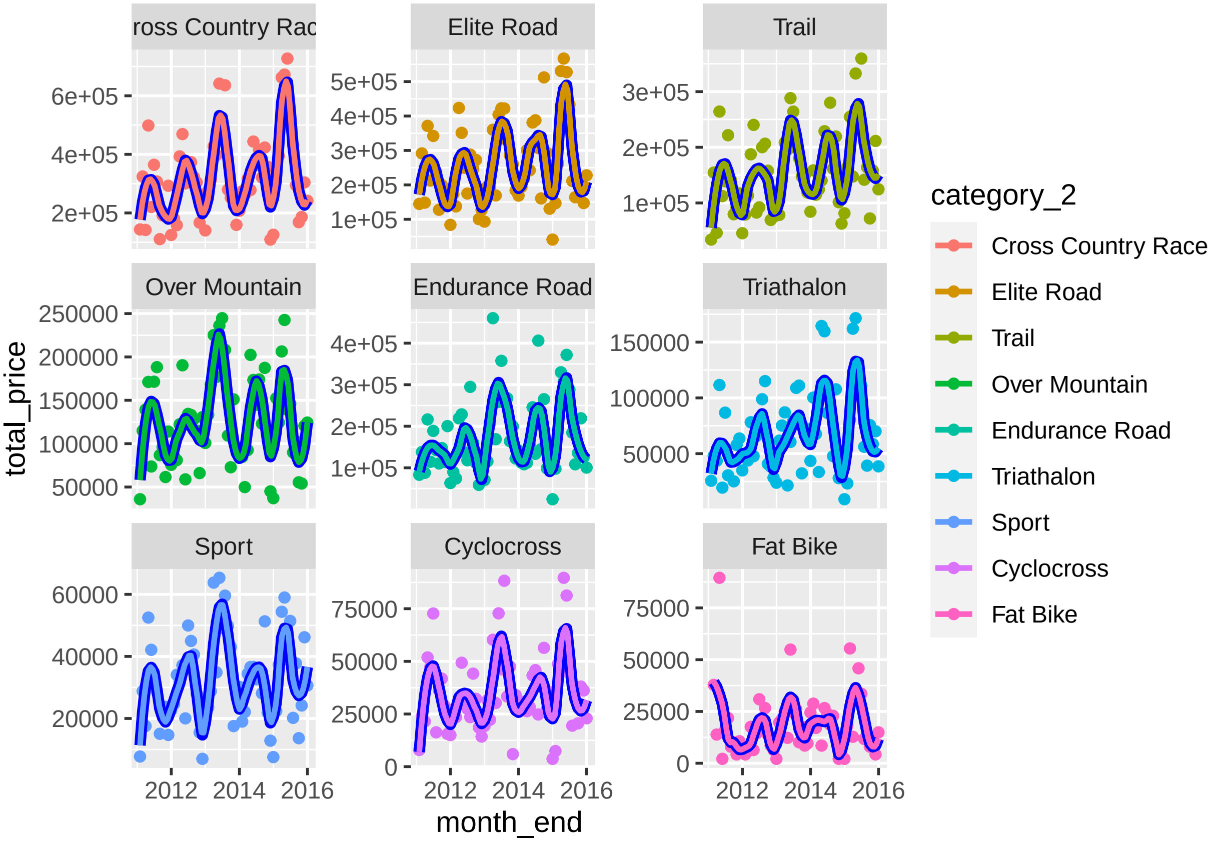
```
## # A tibble: 538 x 6
## # Groups:   category_1, category_2 [9]
##    category_1 category_2         month_end  total_price rolling_avg_3 .fitted
##    <chr>      <fct>              <date>           <dbl>         <dbl>   <dbl>
##  1 Mountain   Cross Country Race 2011-01-31      143660            NA  176998.
##  2 Mountain   Cross Country Race 2011-02-28      324400            NA  239802.
##  3 Mountain   Cross Country Race 2011-03-31      142000       203353.  286279.
##  4 Mountain   Cross Country Race 2011-04-30      498580       321660   311685.
##  5 Mountain   Cross Country Race 2011-05-31      220310       286963.  313621.
##  6 Mountain   Cross Country Race 2011-06-30      364420       361103.  298642.
##  7 Mountain   Cross Country Race 2011-07-31      307300       297343.  261073.
##  8 Mountain   Cross Country Race 2011-08-31      110600       260773.  221223.
##  9 Mountain   Cross Country Race 2011-09-30      191870       203257.  201690.
## 10 Mountain   Cross Country Race 2011-10-31      196440       166303.  187415.
## # ... with 528 more rows
```

Visualize Results

```
loess_tbl_nested %>%
  unnest() %>%
  ggplot(aes(x = month_end, total_price, colour = category_2)) +

  # Geometries
  geom_point() +
  geom_line(aes(y = .fitted), colour = "blue", size = 2) +
  geom_smooth(method = "loess", span = 0.2, se = FALSE) +
  facet_wrap(~category_2, scales = "free_y")
```

```
## Warning: ‘cols‘ is now required when using unnest().
## Please use ‘cols = c(data, fitted)‘
```

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


ross Country Rac
Elite Road
Trail
Over Mountain
Endurance Road
Triathalon
Sport
Cyclocross
Fat Bike
6e+05
4e+05
2e+05
5e+05
4e+05
3e+05
2e+05
1e+05
3e+05
2e+05
1e+05
250000
200000
150000
100000
50000
4e+05
3e+05
2e+05
1e+05
150000
100000
50000
60000
40000
20000
75000
50000
25000
0
75000
50000
25000
0
2012
2014
2016
total_price
month_end
category_2
Cross Country Race
Elite Road
Trail
Over Mountain
Endurance Road
Triathalon
Sport
Cyclocross
Fat Bike