

Яблонцева А.Д.

Хакасский государственный университет им. Н.Ф. Катанова, Россия, г. Абакан

РАСЧЕТ ЛИНЕЙНОЙ РЕГРЕССИИ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА R В СТИЛЕ ПАКЕТОВ: PLYR И PURRR

Аннотация

В статье приводятся разные техники работы по написанию программного кода, для реализации работы по множественному применению функций, с использованием декларативного описания пакетом `plyr` [1], а также в функциональном стиле с использованием `purrr` [2]. Приводится описание программного кода, а также рендеринг итоговых результатов для разных подходов программирования.

Ключевые слова: функциональное программирование, фрейм данных, линейная регрессия
Keywords: functional programming, data frame, linear regression

*Научный руководитель Голубничий А.А.
старший преподаватель кафедры ПОВТиАС,
ХГУ им. Н.Ф. Катанова, Россия, г. Абакан*

Пакет `purrr` расширяет набор инструментов функционального программирования R, предоставляя полный и согласованный набор инструментов для работы с функциями и векторами. Основной особенностью функций пакета `purrr` является возможность и удобство работы со списками, в то время как `dplyr` содержит множество инструментов для работы с фреймами данных.

Для визуализации особенностей и разницы работы данных пакетов будем использовать функцию построения модели линейной регрессии `lm` [3] для сгенерированных случайным образом данных (рисунок 1).

```
set.seed(20210813)
n_groups <- 10
group_sizes <- rpois(n_groups, 10)
n <- sum(group_sizes)
fake_data <- tibble(id = 1:n, group = rep(1:n_groups, times = group_sizes),
                    predictor = runif(n, 0, 1))
group_intercept <- rnorm(n_groups, 0, 1)
fake_data$response <- fake_data$predictor * 10 +
  group_intercept[fake_data$group] + rnorm(n)
```

Рисунок 1 – Генерация данных для построения модели линейной регрессии

При реализации через пакет `plyr` (предшественник пакета `dplyr`) процесс выглядит следующим образом: `dlply` переводит данные из фрейма данных, разделяя его по группам, к списку линейных моделей, затем `ldply` переводит из списка моделей в набор данных коэффициентов (рисунок 2).

```
fit_model <- function(data) lm(response ~ predictor, data)
models <- dlply(fake_data, "group", fit_model)
result <- ldply(models, tidy)
```

Рисунок 2 – Построение наборов линейных моделей в логике пакета `plyr`

Такую же логику, но в более упрощенном виде позволяет выполнить функция `tidy` пакета `broom`, которая извлекает ту же информацию, что и объект из `summary(lm)`, но в виде фрейма данных.

`ldply` добавляет разделенные метки в столбец группы, чтобы было известно, какие строки из какого подмножества (рисунок 3).

```
> head(result)
  group      term estimate std.error statistic    p.value
1     1 (Intercept) -0.4615214 0.6379759 -0.7234151 0.4900333163
2     1 predictor  10.7251023 1.1725741  9.1466308 0.0000164527
3     2 (Intercept)  1.2792440 1.4263598  0.8968592 0.4204932352
4     2 predictor  10.2290565 2.0417362  5.0099796 0.0074377762
5     3 (Intercept) -0.4109177 0.7811463 -0.5260445 0.6213463888
6     3 predictor  10.2430185 1.3135394  7.7980292 0.0005555841
```

Рисунок 3 – Рендеринг результата построения линейных моделей

При использовании `purrr`, можно получить точно такую же таблицу. Единственное отличие состоит в том, что мы получаем из него более корректный фрейм данных, вместо базового R `data.frame` (рисунок 4).

```
> models <- map(split(fake_data, fake_data$group), fit_model)
> result <- map_df(models, tidy, .id = "group")
> result
# A tibble: 20 x 6
   group term      estimate std.error statistic    p.value
  <chr> <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1 1     (Intercept) -0.462      0.638     -0.723    0.490
2 1     predictor  10.7       1.17      9.15     0.0000165
3 2     (Intercept)  1.28       1.43      0.897    0.420
4 2     predictor  10.2       2.04      5.01     0.00744
5 3     (Intercept) -0.411      0.781     -0.526    0.621
6 3     predictor  10.2       1.31      7.80     0.000556
```

Рисунок 4 – Фрейм данных (tibble) созданный с помощью пакета `purrr`

Базовая функция `split` позволяет нам разбивать данные на подмножества на основе значений переменной, которая в данном случае является групповой переменной. Результатом этой функции является список фреймов данных, по одному для каждой группы. Далее, используем `map()` для применения функции к каждому элементу этого списка и также применяем функцию `tidy` к каждому элементу этого списка моделей. Поскольку нам необходимо, чтобы результатом был один единственный фрейм данных, объединяющий выходные данные каждого элемента, мы используем `map_df`, который объединит результаты. Аргумент `.id` указывает добавление столбца группы, который указывает, из какого элемента списка моделей берется каждая строка. Это необходимо для идентификации коэффициентов. Однако, чтобы не убирать линейную модель из фрейма данных, подальше от ее группы, маркирующей любые потенциальные ковариаты на уровне группы, так как это приводит к некоторому смещению элементов. С помощью столбцов списка возможно хранить группы во фрейме данных, сохраняя вместе подмножества данных и любые объекты R, которые мы сгенерировали из них. Для реализации данной логики необходимо использовать функцию `nest` для создания фрейма данных, где каждая строка является группой, а столбец «data», содержащий данные для этого подмножества. Затем в этот фрейм данных добавляются два столбца списка: наш список моделей, а затем наш список фреймов данных с коэффициентами. Наконец, извлекаются оценки в новый фрейм данных с помощью `unnest`.

Результатом является тот же самый фрейм данных коэффициентов и статистики, также содержащий подмножества данных, линейные модели и коэффициенты (рисунок 5).

```
> fake_data %>%
+   group_by(group) %>%
+   nest(data = c(id, predictor, response)) %>%
+   mutate(model = map(data, fit_model),
+           estimates = map(model, tidy)) %>%
+   unnest(estimates)
# A tibble: 20 x 8
# Groups:   group [10]
   group data          model term      estimate std.error statistic    p.value
  <int> <list>         <list> <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1     1 1 <tibble [10 x 3... <lm> (Intercep... -0.462      0.638    -0.723    4.90e-1
2     1 1 <tibble [10 x 3... <lm> predictor    10.7       1.17     9.15     1.65e-5
3     2 2 <tibble [6 x 3]> <lm> (Intercep... 1.28       1.43     0.897    4.20e-1
4     2 2 <tibble [6 x 3]> <lm> predictor    10.2       2.04     5.01     7.44e-3
5     3 3 <tibble [7 x 3]> <lm> (Intercep... -0.411      0.781    -0.526    6.21e-1
6     3 3 <tibble [7 x 3]> <lm> predictor    10.2       1.31     7.80     5.56e-4
```

Рисунок 5 – Вложение данных в столбцы списка

Колонка списков сохраняет объекты, которые принадлежат друг другу, так что нет риска нарушить порядок. Также можно запускать модели и извлекать коэффициенты в одном вызове `mutate`. Пакет `lme4` используется для соотношения переменного пересечения модели, которая ставит один и тот же коэффициент на наклон между ответом и предиктором в каждой группе, но позволяет использовать `intercept` для варьирования между группами, предполагая, что они взяты из одного и того же нормального распределения. Мы помещаем коэффициенты из линейных моделей, подходящих для каждой группы отдельно, и линейной смешанной модели вместе в одном фрейме данных, чтобы иметь возможность построить их. Линейные смешанные модели имеют одинаковый наклон в каждой группе, а точки пересечения приближаются к среднему значению. Результат построения приведен на рисунке 6.

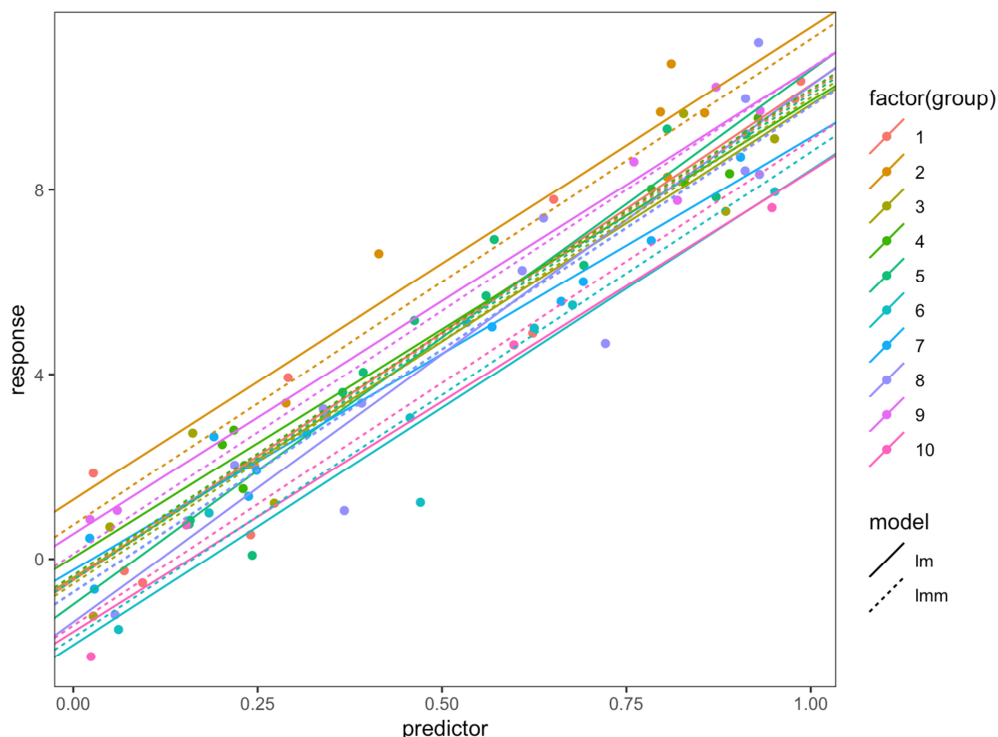


Рисунок 6 – Линейная смешанная модель, построенная на общем наборе данных

Заключение

Реализация линейной регрессии, на сгенерированном наборе данных, показывает возможность использовать разный стиль написания программного кода. Имея в своем распоряжении элегантные способы преобразования фреймов данных и суммирования моделей, мы делаем этот процесс более плавным.

Литература

1. CRAN – Package plyr [Электронный ресурс] URL: <https://CRAN.R-project.org/package=plyr> (дата обращения: 13.08.2021).
2. CRAN – Package purrr [Электронный ресурс] URL: <https://CRAN.R-project.org/package=purrr> (дата обращения: 13.08.2021).
3. Основы линейной регрессии [Электронный ресурс] <http://statistica.ru/theory/osnovy-lineynoy-regressii/> (дата обращения: 13.08.2021).