Data Science, Artificial Intelligence, Machine Learning and Deep Learning:

Tips and tricks with R and Python

Akil Elkamel

6/11/2020

pivot_longer() - Definition

The pivot_longer() function from the tidyr package, “lengthens” data, increasing the number of rows and decreasing the number of columns. The inverse transformation is pivot_wider()

Usage:

```
pivot_longer(
  data,
  cols,
  names_to = "name",
  names_prefix = NULL,
  names_sep = NULL,
  names_pattern = NULL,
  names_ptypes = list(),
  names_transform = list(),
  names_repair = "check_unique",
  values_to = "value",
  values_drop_na = FALSE,
  values_ptypes = list(),
  values_transform = list(),
  ...
)
```

pivot_longer() - Example 1

```
# Simplest case where column names are character data
relig_income
```

```
## # A tibble: 18 x 11
##    religion `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k` `$75-100k`
##    <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
##  1 Agnostic      27        34        60        81        76       137        122
##  2 Atheist       12        27        37        52        35        70         73
##  3 Buddhist      27        21        30        34        33        58         62
##  4 Catholic     418       617       732       670       638      1116        949
##  5 Don't k~      15        14        15        11        10        35         21
##  6 Evangel~     575       869      1064       982       881      1486        949
##  7 Hindu          1         9         7         9        11        34         47
##  8 Histori~     228       244       236       238       197       223        131
##  9 Jehovah~      20        27        24        24        21        30         15
## 10 Jewish        19        19        25        25        30        95         69
## 11 Mainlin~     289       495       619       655       651      1107        939
## 12 Mormon        29        40        48        51        56       112         85
## 13 Muslim         6         7         9        10         9        23         16
## 14 Orthodox      13        17        23        32        32        47         38
## 15 Other C~       9         7        11        13        13        14         18
## 16 Other F~      20        33        40        46        49        63         46
## 17 Other W~       5         2         3         4         2         7          3
## 18 Unaffil~     217       299       374       365       341       528        407
## # ... with 3 more variables: `$100-150k` <dbl>, `>150k` <dbl>, `Don't
## #   know/refused` <dbl>
```

pivot_longer() - Example I

```
relig_income %>%
  pivot_longer(-religion, names_to = "income", values_to = "count")
```

```
## # A tibble: 180 x 3
##    religion income             count
##    <chr>    <chr>              <dbl>
##  1 Agnostic <$10k                27
##  2 Agnostic $10-20k              34
##  3 Agnostic $20-30k              60
##  4 Agnostic $30-40k              81
##  5 Agnostic $40-50k              76
##  6 Agnostic $50-75k             137
##  7 Agnostic $75-100k            122
##  8 Agnostic $100-150k           109
##  9 Agnostic >150k                84
## 10 Agnostic Don't know/refused   96
## # ... with 170 more rows
```

pivot_longer() - Example 2

```
# Slightly more complex case where columns have common prefix,
# and missing missings are structural so should be dropped.
billboard
```

```
## # A tibble: 317 x 79
##    artist track date.entered   wk1   wk2   wk3   wk4   wk5   wk6   wk7   wk8
##    <chr>  <chr> <date>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2 Pac  Baby~ 2000-02-26      87    82    72    77    87    94    99    NA
##  2 2Ge+h~ The ~ 2000-09-02      91    87    92    NA    NA    NA    NA    NA
##  3 3 Doo~ Kryp~ 2000-04-08      81    70    68    67    66    57    54    53
##  4 3 Doo~ Loser 2000-10-21      76    76    72    69    67    65    55    59
##  5 504 B~ Wobb~ 2000-04-15      57    34    25    17    17    31    36    49
##  6 98^0   Give~ 2000-08-19      51    39    34    26    26    19     2     2
##  7 A*Tee~ Danc~ 2000-07-08      97    97    96    95   100    NA    NA    NA
##  8 Aaliy~ I Do~ 2000-01-29      84    62    51    41    38    35    35    38
##  9 Aaliy~ Try ~ 2000-03-18      59    53    38    28    21    18    16    14
## 10 Adams~ Open~ 2000-08-26      76    76    74    69    68    67    61    58
## # ... with 307 more rows, and 68 more variables: wk9 <dbl>, wk10 <dbl>,
## #   wk11 <dbl>, wk12 <dbl>, wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>,
## #   wk17 <dbl>, wk18 <dbl>, wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>,
## #   wk23 <dbl>, wk24 <dbl>, wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>,
## #   wk29 <dbl>, wk30 <dbl>, wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>,
## #   wk35 <dbl>, wk36 <dbl>, wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>,
## #   wk41 <dbl>, wk42 <dbl>, wk43 <dbl>, wk44 <dbl>, wk45 <dbl>, wk46 <dbl>,
## #   wk47 <dbl>, wk48 <dbl>, wk49 <dbl>, wk50 <dbl>, wk51 <dbl>, wk52 <dbl>,
## #   wk53 <dbl>, wk54 <dbl>, wk55 <dbl>, wk56 <dbl>, wk57 <dbl>, wk58 <dbl>,
## #   wk59 <dbl>, wk60 <dbl>, wk61 <dbl>, wk62 <dbl>, wk63 <dbl>, wk64 <dbl>,
## #   wk65 <dbl>, wk66 <lgl>, wk67 <lgl>, wk68 <lgl>, wk69 <lgl>, wk70 <lgl>,
## #   wk71 <lgl>, wk72 <lgl>, wk73 <lgl>, wk74 <lgl>, wk75 <lgl>, wk76 <lgl>
```

pivot_longer() - Example 2

```
# Slightly more complex case where columns have common prefix,
# and missing missings are structural so should be dropped.
billboard
```

```
## # A tibble: 317 x 79
##    artist track date.entered   wk1   wk2   wk3   wk4   wk5   wk6   wk7   wk8
##    <chr>  <chr> <date>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2 Pac  Baby~ 2000-02-26      87    82    72    77    87    94    99    NA
##  2 2Ge+h~ The ~ 2000-09-02      91    87    92    NA    NA    NA    NA    NA
##  3 3 Doo~ Kryp~ 2000-04-08      81    70    68    67    66    57    54    53
##  4 3 Doo~ Loser 2000-10-21      76    76    72    69    67    65    55    59
##  5 504 B~ Wobb~ 2000-04-15      57    34    25    17    17    31    36    49
##  6 98^0   Give~ 2000-08-19      51    39    34    26    26    19     2     2
##  7 A*Tee~ Danc~ 2000-07-08      97    97    96    95   100    NA    NA    NA
##  8 Aaliy~ I Do~ 2000-01-29      84    62    51    41    38    35    35    38
##  9 Aaliy~ Try ~ 2000-03-18      59    53    38    28    21    18    16    14
## 10 Adams~ Open~ 2000-08-26      76    76    74    69    68    67    61    58
## # ... with 307 more rows, and 68 more variables: wk9 <dbl>, wk10 <dbl>,
## #   wk11 <dbl>, wk12 <dbl>, wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>,
## #   wk17 <dbl>, wk18 <dbl>, wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>,
## #   wk23 <dbl>, wk24 <dbl>, wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>,
## #   wk29 <dbl>, wk30 <dbl>, wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>,
## #   wk35 <dbl>, wk36 <dbl>, wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>,
## #   wk41 <dbl>, wk42 <dbl>, wk43 <dbl>, wk44 <dbl>, wk45 <dbl>, wk46 <dbl>,
## #   wk47 <dbl>, wk48 <dbl>, wk49 <dbl>, wk50 <dbl>, wk51 <dbl>, wk52 <dbl>,
## #   wk53 <dbl>, wk54 <dbl>, wk55 <dbl>, wk56 <dbl>, wk57 <dbl>, wk58 <dbl>,
## #   wk59 <dbl>, wk60 <dbl>, wk61 <dbl>, wk62 <dbl>, wk63 <dbl>, wk64 <dbl>,
## #   wk65 <dbl>, wk66 <lgl>, wk67 <lgl>, wk68 <lgl>, wk69 <lgl>, wk70 <lgl>,
## #   wk71 <lgl>, wk72 <lgl>, wk73 <lgl>, wk74 <lgl>, wk75 <lgl>, wk76 <lgl>
```

pivot_longer() - Example 2

```
billboard %>%
 pivot_longer(
   cols = starts_with("wk"),
   names_to = "week",
   names_prefix = "wk",
   values_to = "rank",
   values_drop_na = TRUE
 )
```

```
## # A tibble: 5,307 x 5
##    artist  track                   date.entered week   rank
##    <chr>   <chr>                   <date>       <chr> <dbl>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26   1        87
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26   2        82
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26   3        72
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26   4        77
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26   5        87
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26   6        94
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26   7        99
##  8 2Ge+her The Hardest Part Of ... 2000-09-02   1        91
##  9 2Ge+her The Hardest Part Of ... 2000-09-02   2        87
## 10 2Ge+her The Hardest Part Of ... 2000-09-02   3        92
## # ... with 5,297 more rows
```

pivot_longer() - Example 3

```
# Multiple variables stored in column names
who
```

```
## # A tibble: 7,240 x 60
##    country iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##    <chr>   <chr> <chr> <int>       <int>        <int>        <int>        <int>
##  1 Afghan~ AF    AFG    1980          NA           NA           NA           NA
##  2 Afghan~ AF    AFG    1981          NA           NA           NA           NA
##  3 Afghan~ AF    AFG    1982          NA           NA           NA           NA
##  4 Afghan~ AF    AFG    1983          NA           NA           NA           NA
##  5 Afghan~ AF    AFG    1984          NA           NA           NA           NA
##  6 Afghan~ AF    AFG    1985          NA           NA           NA           NA
##  7 Afghan~ AF    AFG    1986          NA           NA           NA           NA
##  8 Afghan~ AF    AFG    1987          NA           NA           NA           NA
##  9 Afghan~ AF    AFG    1988          NA           NA           NA           NA
## 10 Afghan~ AF    AFG    1989          NA           NA           NA           NA
## # ... with 7,230 more rows, and 52 more variables: new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>,
## #   new_sn_f2534 <int>, new_sn_f3544 <int>, new_sn_f4554 <int>,
## #   new_sn_f5564 <int>, new_sn_f65 <int>, new_ep_m014 <int>,
## #   new_ep_m1524 <int>, new_ep_m2534 <int>, new_ep_m3544 <int>,
## #   new_ep_m4554 <int>, new_ep_m5564 <int>, new_ep_m65 <int>,
## #   new_ep_f014 <int>, new_ep_f1524 <int>, new_ep_f2534 <int>,
## #   new_ep_f3544 <int>, new_ep_f4554 <int>, new_ep_f5564 <int>,
## #   new_ep_f65 <int>, newrel_m014 <int>, newrel_m1524 <int>,
## #   newrel_m2534 <int>, newrel_m3544 <int>, newrel_m4554 <int>,
## #   newrel_m5564 <int>, newrel_m65 <int>, newrel_f014 <int>,
## #   newrel_f1524 <int>, newrel_f2534 <int>, newrel_f3544 <int>,
## #   newrel_f4554 <int>, newrel_f5564 <int>, newrel_f65 <int>
```

pivot_longer() - Example 3

```
who %>% pivot_longer(
  cols = new_sp_m014:newrel_f65,
  names_to = c("diagnosis", "gender", "age"),
  names_pattern = "new_?(.*)_(.)(.*)",
  values_to = "count"
)
```

```
## # A tibble: 405,440 x 8
##    country     iso2  iso3   year diagnosis gender age   count
##    <chr>       <chr> <chr> <int> <chr>     <chr>  <chr> <int>
##  1 Afghanistan AF    AFG    1980 sp        m      014      NA
##  2 Afghanistan AF    AFG    1980 sp        m      1524     NA
##  3 Afghanistan AF    AFG    1980 sp        m      2534     NA
##  4 Afghanistan AF    AFG    1980 sp        m      3544     NA
##  5 Afghanistan AF    AFG    1980 sp        m      4554     NA
##  6 Afghanistan AF    AFG    1980 sp        m      5564     NA
##  7 Afghanistan AF    AFG    1980 sp        m      65       NA
##  8 Afghanistan AF    AFG    1980 sp        f      014      NA
##  9 Afghanistan AF    AFG    1980 sp        f      1524     NA
## 10 Afghanistan AF    AFG    1980 sp        f      2534     NA
## # ... with 405,430 more rows
```

pivot_longer() - Example 4

```
# Multiple observations per row
anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

pivot_longer() - Example 4

```
anscombe %>%
 pivot_longer(everything(),
   names_to = c(".value", "set"),
   names_pattern = "(.)(.)"
 )
```

```
## # A tibble: 44 x 3
##    set       x     y
##    <chr> <dbl> <dbl>
##  1 1        10  8.04
##  2 2        10  9.14
##  3 3        10  7.46
##  4 4         8  6.58
##  5 1         8  6.95
##  6 2         8  8.14
##  7 3         8  6.77
##  8 4         8  5.76
##  9 1        13  7.58
## 10 2        13  8.74
## # ... with 34 more rows
```

accumulate()

Accumulate allows you to incrementally build something. One example is that we could use accumulate() to return cumulative sums. After we do a simple summation example we'll use the accumulate function to build multiple models and to give us the AIC for each of those models.

Accumulate is a function in the purrr library, so we'd start by calling the purrr and dplyr libraries. We'll also call in the tibble library because we'll use the enframe function from the tibble package for our model building example.

```
library(purrr)
1:5 %>%
  accumulate(function(x, y) x + y)
```

```
## [1]  1  3  6 10 15
```

accumulate()

```
library(purrr)
library(tibble)
income <- read.csv("data/Income_7.csv")
models <- c("grouped_marital", "grouped_gov_work", "grouped_education", "gender") %>%
  accumulate(function(x, y) paste(x, y, sep = " + "), .init = "income_recoded ~ age") %>%
  set_names(1:length(.))
enframe(models, name = "model", value = "spec")
```

```
## # A tibble: 5 x 2
##   model spec
##   <chr> <chr>
## 1 1     income_recoded ~ age
## 2 2     income_recoded ~ age + grouped_marital
## 3 3     income_recoded ~ age + grouped_marital + grouped_gov_work
## 4 4     income_recoded ~ age + grouped_marital + grouped_gov_work + grouped_edu~
## 5 5     income_recoded ~ age + grouped_marital + grouped_gov_work + grouped_edu~
```

accumulate()

```
models %>%
  map(glm, data = income) %>%
  map(summary) %>%
  map_dbl('aic') %>%
  enframe(name = "model", value = "AIC")
```

```
## # A tibble: 5 x 2
##   model    AIC
##   <chr>  <dbl>
## 1 1     35247.
## 2 2     30678.
## 3 3     30587.
## 4 4     29663.
## 5 5     29434.
```

Train test split (R)

Train test split (Python)