R Programming Practice\ Authored By: James Mba Azam\ Last Updated: 2017-04-11

70%

MANAGING MANY MODELS (By HARDLEY WICKHAM)

Summary

1. Convert models to tidy dataframes with the BROOM package
2. Use `dplyr` for working with dataframes
3. Use `purrr` for lists
4. Use `tidyr` for converting between dataframe and lists of dataframes

```
#Import data from the gapminder package (very cool package with great data)
# year1950 is a column that contains the number of years since 1950
gapminder <- gapminder %>% mutate(year1950 = year-1950)


#nested data

by_country <- gapminder%>%
  group_by(country,continent) %>%
  nest()

#country_model() fits each country data to a linear model

country_model <- function(df){
  lm(lifeExp ~ year1950, data = df)
}


models <- by_country %>%
  mutate(model=map(data, country_model))


models %>% filter(continent=="Africa")
```

```
## # A tibble: 52 × 4
##                      country continent              data    model
##                       <fctr>    <fctr>            <list>   <list>
## 1                    Algeria    Africa <tibble [12 × 5]> <S3: lm>
## 2                     Angola    Africa <tibble [12 × 5]> <S3: lm>
## 3                      Benin    Africa <tibble [12 × 5]> <S3: lm>
## 4                   Botswana    Africa <tibble [12 × 5]> <S3: lm>
## 5               Burkina Faso    Africa <tibble [12 × 5]> <S3: lm>
## 6                    Burundi    Africa <tibble [12 × 5]> <S3: lm>
## 7                   Cameroon    Africa <tibble [12 × 5]> <S3: lm>
## 8  Central African Republic    Africa <tibble [12 × 5]> <S3: lm>
## 9                       Chad    Africa <tibble [12 × 5]> <S3: lm>
## 10                   Comoros    Africa <tibble [12 × 5]> <S3: lm>
## # ... with 42 more rows
```

what can we do with a list of linear models? not very much

we can convert our data in tidy data using the broom package

what sort of data can we get from our models? In BROOM, `glance` gives the model summaries, `tidy()` gives the estimates, and `augment` gives the stats per observation

```
library("broom")

models <- models %>%
  mutate(
    glance = map(model,broom::glance),
    rsq =glance %>%map_dbl("r.squared"),
    tidy = map(model, broom::tidy),
    augment = map(model, broom::augment)
  )
models
```

```
## # A tibble: 142 × 8
##        country continent              data    model                glance
##         <fctr>    <fctr>            <list>   <list>               <list>
## 1  Afghanistan      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 2      Albania    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 3      Algeria    Africa <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 4       Angola    Africa <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 5    Argentina  Americas <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 6    Australia   Oceania <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 7      Austria    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 8      Bahrain      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 9   Bangladesh      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 10     Belgium    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## # ... with 132 more rows, and 3 more variables: rsq <dbl>, tidy <list>,
## #   augment <list>
```

```
unnest(models,data)
```

```
## # A tibble: 1,704 × 8
##        country continent       rsq  year lifeExp      pop gdpPercap
##         <fctr>    <fctr>     <dbl> <int>   <dbl>    <int>     <dbl>
## 1  Afghanistan      Asia 0.9477123  1952  28.801  8425333  779.4453
## 2  Afghanistan      Asia 0.9477123  1957  30.332  9240934  820.8530
## 3  Afghanistan      Asia 0.9477123  1962  31.997 10267083  853.1007
## 4  Afghanistan      Asia 0.9477123  1967  34.020 11537966  836.1971
## 5  Afghanistan      Asia 0.9477123  1972  36.088 13079460  739.9811
## 6  Afghanistan      Asia 0.9477123  1977  38.438 14880372  786.1134
## 7  Afghanistan      Asia 0.9477123  1982  39.854 12881816  978.0114
## 8  Afghanistan      Asia 0.9477123  1987  40.822 13867957  852.3959
## 9  Afghanistan      Asia 0.9477123  1992  41.674 16317921  649.3414
## 10 Afghanistan      Asia 0.9477123  1997  41.763 22227415  635.3414
## # ... with 1,694 more rows, and 1 more variables: year1950 <dbl>
```

```
unnest(models,glance, .drop = TRUE)# %>%View()
```

```
## # A tibble: 142 × 14
##        country continent       rsq r.squared adj.r.squared     sigma
##         <fctr>    <fctr>     <dbl>     <dbl>         <dbl>     <dbl>
## 1  Afghanistan      Asia 0.9477123 0.9477123     0.9424835 1.2227880
## 2      Albania    Europe 0.9105778 0.9105778     0.9016355 1.9830615
## 3      Algeria    Africa 0.9851172 0.9851172     0.9836289 1.3230064
## 4       Angola    Africa 0.8878146 0.8878146     0.8765961 1.4070091
```

```
## 5    Argentina  Americas 0.9955681 0.9955681     0.9951249 0.2923072
## 6    Australia   Oceania 0.9796477 0.9796477     0.9776125 0.6206086
## 7      Austria    Europe 0.9921340 0.9921340     0.9913474 0.4074094
## 8      Bahrain      Asia 0.9667398 0.9667398     0.9634138 1.6395865
## 9   Bangladesh      Asia 0.9893609 0.9893609     0.9882970 0.9766908
## 10     Belgium    Europe 0.9945406 0.9945406     0.9939946 0.2929025
## # ... with 132 more rows, and 8 more variables: statistic <dbl>,
## #   p.value <dbl>, df <int>, logLik <dbl>, AIC <dbl>, BIC <dbl>,
## #   deviance <dbl>, df.residual <int>
```

```
unnest(models,rsq)#%>%View()
```

```
## # A tibble: 142 × 8
##        country continent              data    model                glance
##         <fctr>    <fctr>            <list>   <list>               <list>
## 1  Afghanistan      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 2      Albania    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 3      Algeria    Africa <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 4       Angola    Africa <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 5    Argentina  Americas <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 6    Australia   Oceania <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 7      Austria    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 8      Bahrain      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 9   Bangladesh      Asia <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## 10     Belgium    Europe <tibble [12 × 5]> <S3: lm> <data.frame [1 × 11]>
## # ... with 132 more rows, and 3 more variables: tidy <list>,
## #   augment <list>, rsq <dbl>
```

```
unnest(models,tidy)#%>%View()
```

```
## # A tibble: 284 × 8
##        country continent       rsq        term   estimate   std.error
##         <fctr>    <fctr>     <dbl>       <chr>      <dbl>       <dbl>
## 1  Afghanistan      Asia 0.9477123 (Intercept) 29.3566375 0.698981278
## 2  Afghanistan      Asia 0.9477123    year1950  0.2753287 0.020450934
## 3      Albania    Europe 0.9105778 (Intercept) 58.5597618 1.133575812
## 4      Albania    Europe 0.9105778    year1950  0.3346832 0.033166387
## 5      Algeria    Africa 0.9851172 (Intercept) 42.2364149 0.756269040
## 6      Algeria    Africa 0.9851172    year1950  0.5692797 0.022127070
## 7       Angola    Africa 0.8878146 (Intercept) 31.7079741 0.804287463
## 8       Angola    Africa 0.8878146    year1950  0.2093399 0.023532003
## 9    Argentina  Americas 0.9955681 (Intercept) 62.2250191 0.167091314
## 10   Argentina  Americas 0.9955681    year1950  0.2317084 0.004888791
## # ... with 274 more rows, and 2 more variables: statistic <dbl>,
## #   p.value <dbl>
```

```
unnest(models,augment)#%>%View()
```

```
## # A tibble: 1,704 × 12
##        country continent       rsq lifeExp year1950  .fitted   .se.fit
##         <fctr>    <fctr>     <dbl>   <dbl>    <dbl>    <dbl>     <dbl>
## 1  Afghanistan      Asia 0.9477123  28.801        2 29.90729 0.6639995
## 2  Afghanistan      Asia 0.9477123  30.332        7 31.28394 0.5799442
## 3  Afghanistan      Asia 0.9477123  31.997       12 32.66058 0.5026799
## 4  Afghanistan      Asia 0.9477123  34.020       17 34.03722 0.4358337
## 5  Afghanistan      Asia 0.9477123  36.088       22 35.41387 0.3848726
```

```
## 6  Afghanistan      Asia 0.9477123  38.438       27 36.79051 0.3566719
## 7  Afghanistan      Asia 0.9477123  39.854       32 38.16716 0.3566719
## 8  Afghanistan      Asia 0.9477123  40.822       37 39.54380 0.3848726
## 9  Afghanistan      Asia 0.9477123  41.674       42 40.92044 0.4358337
## 10 Afghanistan      Asia 0.9477123  41.763       47 42.29709 0.5026799
## # ... with 1,694 more rows, and 5 more variables: .resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>
```

CONCLUSION

1. Store related objects in list-clumns
2. Learn functional programming to concentrate on the verb and not the object
3. Use broom to convert models to tidy data

DATA VISUALIZATION

```
library("ggplot2")
library("dplyr")
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(gridExtra)
```

```
##
```

```
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
data("economics", package = "ggplot2")
head(economics)
```

```
## # A tibble: 6 × 6
##         date   pce    pop psavert uempmed unemploy
##       <date> <dbl>  <int>   <dbl>   <dbl>    <int>
## 1 1967-07-01 507.4 198712    12.5     4.5     2944
## 2 1967-08-01 510.5 198911    12.5     4.7     2945
## 3 1967-09-01 516.3 199113    11.7     4.6     2958
## 4 1967-10-01 512.9 199311    12.5     4.9     3143
## 5 1967-11-01 518.1 199498    12.5     4.7     3066
## 6 1967-12-01 525.8 199657    12.1     4.8     3018
```

```
# Create the plot
ggplot(data = economics) + geom_line(aes(x = date, y = unemploy))
```