Week 13 – Basic advanced techniques

Joshua Rosenberg and Alex Lishinski

April 15, 2021

Welcome!

Welcome to *week 13*!

Record the meeting

Breakout rooms!

Starting with whomever has the most pets:

- What is something about R you've learned working on your final project that goes beyond what we've covered in class?

Prepare one–three responses to each of the above questions to share with the whole class!

A recap of last week (on functional programming)

- R has built in functions from packages, but you can write your own too!
- The purrr package offers tools to do multiple iterations of operations

Checking–in on final projects

Responses to feedback – we will be checking tomorrow!

Reminder – final project and recording due by May 5

Homework highlights

Model 2: Adds intrinsic goal orientation and problem solving pre-course as predictors.

Term	Est	Std Error	Stat	P Value
(Intercept)	306.243	22.523	13.597	0.000
conscient.	1.019	0.412	2.476	0.014
IntGoalOrient	0.167	0.745	0.224	0.823
Prob_solve_pre	3.243	0.665	4.881	0.000

r^2	Adj r^2	Sigma	Stat	P Value	Deg of Freedom	Log-Likelihood	AIC	BIC	Deviance of Model	DF Residual	# Obs
0.068	0.062	72.449	11.341	0	3	-2672.149	5354.298	5375.051	2440752	465	469

The additional variables now account for 6.2% of the variance in the model.

Homework highlights

Model 2: Adds intrinsic goal orientation and problem solving pre-course as predictors.

```{r}
model2 <- lm(PROJ_TOTAL ~conscient. + IntGoalOrient + Prob_solve_pre,data=data1)
m2 <- tidy(model2)
kable(tidy(model2),
      col.names = c("Term","Est","Std Error","Stat","P Value"),
      align = "c",
      digits=3)
kable(glance(model2),
      col.names = c("r^2","Adj r^2","Sigma","Stat","P Value", "Deg of Freedom", "Log-Likelihood",
                    "AIC", "BIC", "Deviance of Model", "DF Residual", "# Obs"),
      align = "c",
      digits=3)
```

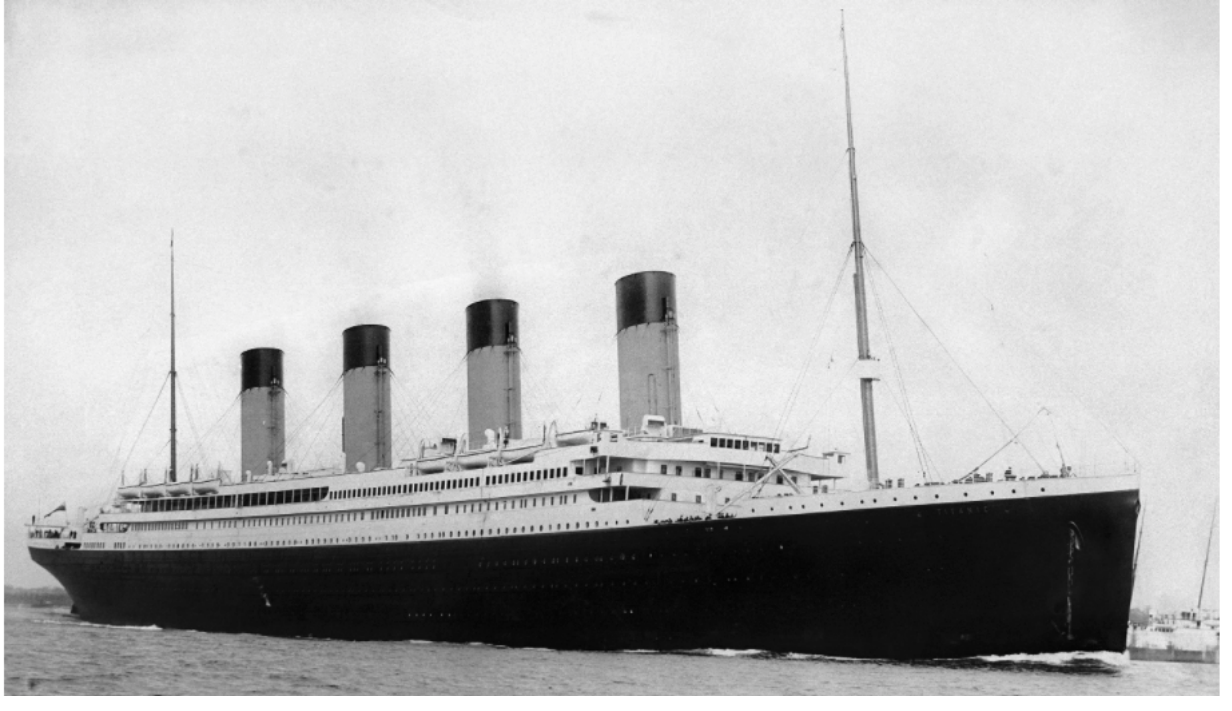
Homework highlights

Discussion

These findings are consistent with what we know about the sinking of Titanic. While I thought Embarkation may have had more effect, it did not. This is likely because at each embarkation point, passengers from all three classes boarded. Passengers also disembarked at each point. This data set is incomplete, which limits the ability to extract information about the full population of Titanic passengers. Titanic has approximately 1300 passengers on her maiden voyage, along with 885 crew. Additionally, some ticket-holders did not board the ship, and not every ticket-holder was on board for the crossing. Finally, we know one crew member sneaked off at Queenstown by hiding under the mailbags being moved ashore.

Titanic is a unique historical case that would not be repeated today. Because of Titanic's sinking, the International Convention for the Safety of Life at Sea (SOLAS) was implemented as maritime treaty. SOLAS required ships to have sufficient lifeboats for all passengers and required continuous radio watches. SOLAS has been periodically updated and is still in effect today.

Given the limited data set we have here, the best use of this data is to explore statistics and programming rather than to learn about individual's chances of survival in a shipwreck.



Chunk options

For a single chunk:

- eval = FALSE
- echo = FALSE
- warning = FALSE
- message = FALSE

For the entire document:

- knitr::opts_chunk$set(echo = FALSE, echo = FALSE, warning = FALSE, message = FALSE)

Topics for today

Record the meeting

A. Continued discussion on purrr – Iteration (or, *applying* functions)

B. Introduction to missing data procedures with MICE

(C. On creating maps using ggplot2)

A. Iteration

Iteration is another name for your computer carrying out some step(s) multiple times.

Iteration is helpful when, even after writing a function, you find yourself copying and pasting the same code (with modifications)

A. Iteration

Iteration can be helpful when you want to apply a function multiple times.

The map functions can help with this. There are different ones based on the type of object returned.

The key to understanding (and writing) functions is understanding what kind of input they take and what kind of output they return.

`map()` – Applies function to every element of a list and return a list `map2()` – Applies function to every element from 2 lists and return a list

Lists are ordered collections of any other type of object. You can have a list of vectors, a list of data frames, and a list of different types of objects.

`map_chr()` – returns a char vector of objects created `map_dbl()` – returns a numeric vector of objects created `map_dfc()` – objects created become columns of data frame `map_dfr()` – objects created become rows of a data frame

A. Iteration

Example: COVID vaccine data

NFL Memes
@NFL_Memes

Pfizer, Johnson & Johnson, Moderna

2:44 PM · Apr 11, 2021

10.5K See the latest COVID-19 information on Twitter

A. Iteration

Example: COVID vaccine data – Read in the data

One way to load data:

```
pfi <- read_csv(here("data", "COVID", "COVID-19_Vaccine_Distribution_Allocations_by_Jurisdiction_-_Pf
mod <- read_csv(here("data", "COVID", "COVID-19_Vaccine_Distribution_Allocations_by_Jurisdiction_-_Mo
jan <- read_csv(here("data", "COVID", "COVID-19_Vaccine_Distribution_Allocations_by_Jurisdiction_-_Ja
jan
```

```
## # A tibble: 378 x 3
##    Jurisdiction  `Week of Allocations` `1st Dose Allocations`
##    <chr>         <chr>                                 <dbl>
##  1 Connecticut   04/12/2021                             6400
##  2 Maine         04/12/2021                             2500
##  3 Massachusetts 04/12/2021                            12300
##  4 New Hampshire 04/12/2021                             2500
##  5 Rhode Island  04/12/2021                             2000
##  6 Vermont       04/12/2021                             1200
##  7 New Jersey    04/12/2021                            15600
##  8 New York      04/12/2021                            19800
##  9 New York City 04/12/2021                            15100
## 10 Puerto Rico   04/12/2021                             6100
## # … with 368 more rows
```

A. Iteration

Example: COVID vaccine data – Read in the data

```
vaccines <- c("Pfizer", "Moderna", "Janssen")
file_base <- "/COVID-19_Vaccine_Distribution_Allocations_by_Jurisdiction_-_"
file_ext <- ".csv"

file_names <- str_c(here("data", "COVID"), file_base, vaccines, file_ext)

vax_files <- file_names %>% map(read_csv)
names(vax_files) <- vaccines
vax_files
```

```
## $Pfizer
## # A tibble: 1,134 x 4
##    Jurisdiction  `Week of Allocations` `1st Dose Allocations` `2nd Dose Allocations`
##    <chr>         <chr>                                  <dbl>                  <dbl>
##  1 Connecticut   04/12/2021                             51480                  51480
##  2 Maine         04/12/2021                             19890                  19890
##  3 Massachusetts 04/12/2021                             97110                  97110
##  4 New Hampshire 04/12/2021                             19890                  19890
##  5 Rhode Island  04/12/2021                             15210                  15210
##  6 Vermont       04/12/2021                              9360                   9360
##  7 New Jersey    04/12/2021                            124020                 124020
##  8 New York      04/12/2021                            156780                 156780
##  9 New York City 04/12/2021                            119340                 119340
## 10 Puerto Rico   04/12/2021                             49140                  49140
## # … with 1,124 more rows
##
## $Moderna
## # A tibble: 1,071 x 4
##    Jurisdiction  `Week of Allocations` `1st Dose Allocations` `2nd Dose Allocations`
##    <chr>         <chr>                                  <dbl>                  <dbl>
##  1 Connecticut   04/12/2021                             37400                  37400
##  2 Maine         04/12/2021                             14300                  14300
##  3 Massachusetts 04/12/2021                             72100                  72100
##  4 New Hampshire 04/12/2021                             14300                  14300
##  5 Rhode Island  04/12/2021                             11300                  11300
##  6 Vermont       04/12/2021                              6700                   6700
```

A. Iteration

Example: COVID vaccine data – Rename the data

```
# write my rename function
rename_vaccine_data <- function(dat){
  dat <- dat %>% rename("State" = "Jurisdiction",
                "Week" = "Week of Allocations",
                "First Dose" = "1st Dose Allocations",
                "Second Dose" = "2nd Dose Allocations")
  dat
}

# map it to every data set in the list
vax_files %>% map(rename_vaccine_data)
```

Error: Can't rename columns that don't exist. x Column `2nd Dose Allocations` doesn't exist.

A. Iteration

Example: COVID vaccine data – Rename the data

```
rename_vaccine_data <- function(dat){
  if(ncol(dat) == 4){
    dat <- dat %>% rename("State" = "Jurisdiction",
                          "Week" = "Week of Allocations",
                          "First Dose" = "1st Dose Allocations",
                          "Second Dose" = "2nd Dose Allocations")
    dat
  } else {
    dat <- dat %>% rename("State" = "Jurisdiction",
                          "Week" = "Week of Allocations",
                          "First Dose" = "1st Dose Allocations")
    dat
  }

}

vax_files <- vax_files %>% map(rename_vaccine_data)
vax_files$Pfizer
```

```
## # A tibble: 1,134 x 4
##    State         Week       `First Dose` `Second Dose`
##    <chr>         <chr>             <dbl>         <dbl>
##  1 Connecticut   04/12/2021        51480         51480
##  2 Maine         04/12/2021        19890         19890
##  3 Massachusetts 04/12/2021        97110         97110
##  4 New Hampshire 04/12/2021        19890         19890
##  5 Rhode Island  04/12/2021        15210         15210
##  6 Vermont       04/12/2021         9360          9360
##  7 New Jersey    04/12/2021       124020        124020
##  8 New York      04/12/2021       156780        156780
##  9 New York City 04/12/2021       119340        119340
## 10 Puerto Rico   04/12/2021        49140         49140
## # … with 1,124 more rows
```

B. Multiple Imputation of missing data with MICE

`lm()` default way of dealing with missing data: listwise deletion

Better way to deal with missing data: imputation

Basic idea:

If data missing, you can fill in values that let you estimate your model on whole data

Good: Mean imputation

Better: Model based imputation

Best: Multiple imputation

B. Multiple Imputation of missing data with MICE

The MICE package: Multiple Imputation using Chained Equations

Dealing with missing data is a big idea, and the nuances can be tricky

Our goal is to show you that you need not be intimidated by approaching this task in R

If you are doing inferential models in your analyses, MI is a tool that should be in your toolkit

B. Multiple Imputation of missing data with MICE

Example: Regression model

```
ug_data <- read_csv(here("data", "undergrad_data.csv"))

vis_dat(ug_data)
```

B. Multiple Imputation of missing data with MICE

Example: Regression model

```
fit <- lm(PROJ_TOTAL ~ Prob_solve_pre + extraversion + agreeableness + conscient. + stability + openn
          data = ug_data)
```

B. Multiple Imputation of missing data with MICE

Example: Regression model

```
summary(fit)
```

```
##
## Call:
## lm(formula = PROJ_TOTAL ~ Prob_solve_pre + extraversion + agreeableness +
##     conscient. + stability + openness + Metacog. + SelfEfficacy +
##     IntGoalOrient + ExtGoalOrient, data = ug_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -334.72  -16.93   23.01   44.39   97.40
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    307.7820    35.8305   8.590  < 2e-16 ***
## Prob_solve_pre   3.3656     0.7639   4.406 1.34e-05 ***
## extraversion    -0.2442     0.3854  -0.634    0.527
## agreeableness   -0.1068     0.4867  -0.219    0.826
## conscient.       0.9207     0.4823   1.909    0.057 .
## stability        0.2047     0.3803   0.538    0.591
## openness         0.3604     0.5615   0.642    0.521
## Metacog.        -0.1107     0.5011  -0.221    0.825
## SelfEfficacy    -0.2373     0.5916  -0.401    0.689
## IntGoalOrient    0.1982     1.0673   0.186    0.853
## ExtGoalOrient    0.1893     0.8883   0.213    0.831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.46 on 416 degrees of freedom
##   (187 observations deleted due to missingness)
## Multiple R-squared:  0.07321,    Adjusted R-squared:  0.05093
## F-statistic: 3.286 on 10 and 416 DF,  p-value: 0.0004114
```

B. Multiple Imputation of missing data with MICE

The MICE package: viewing missing data pattern

```
library(mice)

md.pattern(ug_data)
```

B. Multiple Imputation of missing data with MICE

The MICE package: Generating imputed data sets with `mice()`

```
ug_data_imp <- mice(ug_data, m = 5)
```

```
##
##  iter imp variable
##    1   1  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    1   2  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    1   3  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    1   4  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    1   5  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    2   1  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    2   2  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    2   3  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    2   4  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    2   5  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    3   1  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    3   2  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    3   3  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    3   4  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    3   5  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    4   1  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    4   2  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    4   3  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    4   4  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    4   5  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    5   1  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    5   2  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    5   3  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    5   4  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
##    5   5  Prob_solve_pre  Prob_solve_post  extraversion  agreeableness  conscient.  stability  openness  Met
```

```
## Warning: Number of logged events: 276
```

B. Multiple Imputation of missing data with MICE

The MICE package: Examining missing data

```
ug_data_imp$imp$SelfEfficacy
```

```
##       1  2  3  4  5
## 3    39 48 55 47 42
## 18   32 46 41 52 20
## 35   54 56 41 55 38
## 42   56 43 42 53 41
## 90   45 47 28 35 49
## 94   34 48 45 42 40
## 100 21 44 52 29 44
## 111 53 44 35 35 41
## 124 25 31 32 44 35
## 134 42 29 45 40 22
## 137 47 53 28 55 52
## 148 44 49 28 34 50
## 166 32 34 36 27 44
## 171 47 56 46 36 45
## 176 30 54 46 46 35
## 190 31 56 42 26 26
## 193 47 55 28 48 32
## 208 47 47 27 32 41
## 225 45 32 56 32 46
## 226 48 31 25 56 17
## 236 33 49 48 54 35
## 237 28 50 48 56 36
## 238 32 44 41 49 44
## 242 46 32 29 28 41
## 246 56 32 26 45 47
## 250 56 51 43 46 39
## 257 54 54 48 41 49
## 259 16 43 30 28 43
## 260 45 44 38 32 56
## 267 46 32 48 46 42
## 283 36 32 45 26 37
## 287 49 47 55 33 48
## 299 40 48 34 35 44
```

B. Multiple Imputation of missing data with MICE

The MICE package: Extracting data sets with `complete()`

```
comp_data <- complete(ug_data_imp, 2)
comp_data
```

```
##     X1   Stud_ID PROJ_TOTAL Proj01 Proj02 Proj03 Proj04 Proj05 Proj06 Proj07 Proj08 Proj09 Proj10 Proj11 Tot
## 1    1 e75ab8254        347     14     19     13     25     27     35     40     47     44     35     48
## 2    2 07ee7a263        427     14     20     20     38     45     42     41     50     52     53     52
## 3    3 9c02eb000        393     14     18     16     37     42     44     45     41     50     37     49
## 4    4 52fb2d1ca        397     14     18     19      0     43     43     50     50     52     53     55
## 5    5 a4d76f570        316     14     20     12     36      8     34      0     41     52     49     50
## 6    6 94ea75daa        435     14     20     19     36     45     45     50     50     55     46     55
## 7    7 5e9c38d44        375     13     20     20     36     43     38     46     40     50     22     47
## 8    8 e5a5608e8        398     14     17     19     39     45     41     49     20     55     47     52
## 9    9 f2abdba22        350     14     18     18     37     44      0     45     43     51     40     40
## 10  10 21d036526        354     13     18     17     32     40     33     16     39     52     40     54
## 11  11 3fc8237b5        295     13     20     20     34     43     41     50     19     55      0      0
## 12  12 72ccca5a0        440     14     20     20     40     45     45     50     48     55     48     55
## 13  13 ca5073861         52     14     12      0     26      0      0      0      0      0      0      0
## 14  14 7e2674224        441     14     20     20     38     44     45     50     48     55     53     54
## 15  15 bc0032340        385     14     20     10     27     38     34     50     42     51     45     54
## 16  16 56bb07c2e        370     10     18     18     37     29     43     40     34     49     40     52
## 17  17 74afe8807        332     14     18     18     35     43      0     43     39     35     47     40
## 18  18 70653b198        279     14     20     18     32     43      0      0     43     51     42     16
## 19  19 c19bd5d09        406     14     20     16     37     41     45     50     38     47     45     53
## 20  20 758e0f2e4        340     15     20     20     40     32     43     44     21     50      0     55
## 21  21 2e3be0621        372     14     13     15     28     44     43     50      0     55     55     55
## 22  22 f7fc7c540        410     15     20     20     39     44     45     50     45     27     50     55
## 23  23 8a78da3d7        448     15     20     20     40     45     45     50     50     53     55     55
## 24  24 d032ddcba        408     15     20     10     25     45     45     50     49     48     48     53
## 25  25 f5fbfaa43        437     15     20     20     40     45     40     48     46     53     55     55
## 26  26 897c10248        348     15     18     14      0     37     42     44     44     55     28     51
## 27  27 006a7e31c        119     15     19     10     40      4     31      0      0      0      0      0
## 28  28 d7ebdcc64        299     13     20     20     40     39     45     14      4     34     30     40
## 29  29 2f9655115        392     13     20     17     33     39     45     49     45     52     34     45
## 30  30 1158a4bb2        446     15     20     20     38     43     45     50     50     55     55     55
## 31  31 da8783e45        438     15     20      9     40     45     45     50     50     55     54     55
## 32  32 34d07adfb        172      0     12     14     28     42     25      0     22      0      5     14
```

B. Multiple Imputation of missing data with MICE

The MICE package: fitting models with `with()`

```
fit_imp <- with(data = ug_data_imp, exp = lm( PROJ_TOTAL ~ Prob_solve_pre + extraversion + agreeablen
fit_imp
```

```
## call :
## with.mids(data = ug_data_imp, expr = lm(PROJ_TOTAL ~ Prob_solve_pre +
##     extraversion + agreeableness + conscient. + stability + openness +
##     Metacog. + SelfEfficacy + IntGoalOrient + ExtGoalOrient))
##
## call1 :
## mice(data = ug_data, m = 5)
##
## nmis :
##               X1         Stud_ID      PROJ_TOTAL          Proj01          Proj02          Proj03          Pr
##                0               0               0               0               0               0
##          Proj06          Proj07          Proj08          Proj09          Proj10          Proj11       Total
##                0               0               0               0               0               0
##           Exam2           Exam3  Prob_solve_pre Prob_solve_post    extraversion   agreeableness      consci
##                0               0              43              63              74              90
##        openness        Metacog.    SelfEfficacy   IntGoalOrient   ExtGoalOrient
##              85              67              67              67              67
##
## analyses :
## [[1]]
##
## Call:
## lm(formula = PROJ_TOTAL ~ Prob_solve_pre + extraversion + agreeableness +
##     conscient. + stability + openness + Metacog. + SelfEfficacy +
##     IntGoalOrient + ExtGoalOrient)
##
## Coefficients:
##    (Intercept)  Prob_solve_pre    extraversion   agreeableness      conscient.       stability         openn
##       260.9356          3.7502         -0.2821          0.8928          1.3440          0.2687         -0.5
##   SelfEfficacy   IntGoalOrient   ExtGoalOrient
##         1.9685         -0.2016          0.3695
##
##
```

B. Multiple Imputation of missing data with MICE

The MICE package: pooling output with `pool()`

```
summary(pool(fit_imp))
```

```
##              term     estimate  std.error   statistic         df      p.value
## 1     (Intercept) 250.84471931 45.9701657  5.45668513  42.012481 2.388848e-06
## 2  Prob_solve_pre   3.92857609  0.8719647  4.50543022 205.323239 1.111012e-05
## 3    extraversion  -0.16033222  0.4882295 -0.32839517  47.195380 7.440655e-01
## 4   agreeableness   0.67783109  0.5899823  1.14890071  50.074676 2.560554e-01
## 5      conscient.   1.70127425  0.6163776  2.76011714  32.046286 9.476173e-03
## 6       stability  -0.03809738  0.6658560 -0.05721563   9.987118 9.555020e-01
## 7        openness  -0.48674313  0.6554846 -0.74256989 102.224310 4.594451e-01
## 8        Metacog.  -1.60148596  0.6129725 -2.61265563  40.454957 1.255195e-02
## 9    SelfEfficacy   1.94410779  0.7041804  2.76080940  40.767563 8.602390e-03
## 10  IntGoalOrient   0.34587931  1.1626651  0.29748833 203.062703 7.663978e-01
## 11  ExtGoalOrient  -0.50417035  1.1836683 -0.42593888  30.774582 6.731167e-01
```

```
broom::tidy(fit)
```

```
## # A tibble: 11 x 5
##    term           estimate std.error statistic  p.value
##    <chr>             <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)     308.       35.8       8.59  1.77e-16
##  2 Prob_solve_pre    3.37      0.764     4.41  1.34e- 5
##  3 extraversion     -0.244     0.385    -0.634 5.27e- 1
##  4 agreeableness    -0.107     0.487    -0.219 8.26e- 1
##  5 conscient.        0.921     0.482     1.91  5.70e- 2
##  6 stability         0.205     0.380     0.538 5.91e- 1
##  7 openness          0.360     0.562     0.642 5.21e- 1
##  8 Metacog.         -0.111     0.501    -0.221 8.25e- 1
##  9 SelfEfficacy     -0.237     0.592    -0.401 6.89e- 1
## 10 IntGoalOrient     0.198     1.07      0.186 8.53e- 1
## 11 ExtGoalOrient     0.189     0.888     0.213 8.31e- 1
```

C. Mapping with ggplot2

A brief introduction for those interested, focused on the sf R package:

https://joshuamrosenberg.com/post/2020/06/08/basic-sf-functionality-for-analyzing-and-plotting-geospatial-data/

Data Science Certificate

https://docs.google.com/document/d/1shNL9fY-vhBKK-zJw_V_iVhAKDtZ1Wqen4maJFnswQ4/edit

Curating a data science resource

https://making-data-science-count.github.io/s21-intro-to-data-sci-methods-in-ed/assignment-curating-a-data-science-resource.html

Logistics

This week

- Homework 13: Available tomorrow by noon; **Due by Thursday, 4/22**
- Readings: https://r4ds.had.co.nz/iteration.html#the-map-functions

    ```
    https://www.statisticssolutions.com/multiple-imputation-for-missing-data/
    ```

Schedule

- The product for your final project and a recording that will serve as your presentation of it are both due by the end of the day on May 5
- *There is no portfolio!*
- Exam 3 is available now, due before class on April 22
- The revised data ethics statement is due April 29
- The curating a resource assignment is due by the same date/time as the final project – May 5

Wrapping up

In your base group's Slack channel:

- What is one thing you learned today?
- What is something you want to learn more about?
- Share your feelings in GIF form!