# HW 5

## Enter your name and EID here

**This homework is due on Mar 1, 2020 at 11:59pm. Please submit as a pdf file on Canvas.**

*For all questions, include the R commands/functions that you used to find your answer. Answers without supporting code will not receive credit.*

> **How to submit this assignment**
>
> All homework assignments will be completed using R Markdown. These `.Rmd` files consist of >text/syntax (formatted using Markdown) alongside embedded R code. When you have completed the assignment (by adding R code inside codeblocks and supporting text outside of the codeblocks), create your document as follows:
>
> - Click the "Knit" button (above) to create an .html file
> - Open the html file in your internet browser to view
> - Go to `File > Print` and print your .html file to a .pdf
> - (or knit to PDF)
> - Upload the .pdf file to Canvas

---

**Question 1:**

**WARNING: In this assignment, you will be performing computationally expensive operations on large datasets. You are strongly encouraged to use your own local version of RStudio (also available on any campus computer) rather than the servers, especially if you are a procrastinator. If the server gets slow, try it on a different one (educomp01, educomp02, educomp03, or educomp04). You are advised to begin working on this well before the due date. IF YOUR COMPUTER IS RUNNING SLOWLY when trying to figure out the reshape, YOU ARE STRONGLY ENCOURAGED to take a subset of the wide merged data (e.g., first 500 rows) and test your code on that. Once you get everything working, replace your test sample with the entire dataset and run it one final time (it will likely take several minutes to knit the entire assignment: I would allow a half an hour just in case).**

**Read in the two files `background.csv` and `records.csv` (see code chunk below)**

**In these two datasets, I simulate a real-world scenario of the sort I dealt with regularly as an institutional data analyst. The `background` dataset contains an ID column that identifies each unique student ($\approx 150,000$ from 2000 to 2018), along with background/demographic variables about each student (the data is fake, but the variables and many features of the data are true-to-life). `fseut` is the first semester a student enrolled at UT; `derivation` is based on a university race/ethnicity/nationality category; `SES` is a measure of socioeconomic status based on educational attainment and family income, averaged for both parents (1=lowest SES**

**category, 10=highest SES category). `AP` and `CBE` indicate transfer credits from those exams. `SAT` is an SAT-equivalent score (ACT converted if applicable).**

**The `records` dataset is a wide file that contains, for each of eight possible years, a unique students' hours undertaken, hours passed, hours failed, grade points, and gpa. You would be wise to familiarize yourself with what these two datasets look like before diving into the assignment, especially `records` (i.e., you will almost certainly save yourself time).**

**1 (4 pts) How many IDs are in `background` that do not appear in `records`? How many IDs are in `records` that do not appear in `background`? How many IDs do the two datasets have in common? If there were supposed to be 150000 students total, how many students are missing entirely from these data (i.e., their IDs appear neither in the background data or student records)?**

```
library(tidyverse)

## On the server? Uncomment and run these

#bg<-read.csv("/stor/work/SDS348_Fall2019/Data/background.csv")
#rec<-read.csv("/stor/work/SDS348_Fall2019/Data/records.csv")

## Not on the server? Uncomment and run these

#bg<-read.csv("https://drive.google.com/uc?export=download&id=1iDZjouO3o2KmO3EJg8tdqjKXyQ3XE7FA")
#rec<-read.csv("https://drive.google.com/uc?export=download&id=1PhQ51JED5ZVR6Qp85Ds5GK2cg55IQzjr")

#You are encouraged to poke around: to get some sense of the data, try
#head(bg)
#names(rec)[-1]%>%matrix(nrow=18,byrow=F)
```

```
# your code here
```

*your answer here, 1-2 sentences*

**2 (a) (1 pt) Perform a full-join on this data and save it as `fulldata` (1 pt).**

```
# your code here
```

**2 (b) (8 pt) Now, tidy this data. Create a new dataset (call it `longdat`). Each student-year-semester is an observation, so I want a column for year order (called `order`: first, second, third, etc.; need to use separate), a column for `semester` (recoded with semester names rather than numbers: "9"="fall", "6"="summer", "2"="spring"; need to use separate), a column called `ccyys` (e.g., 20089, 20092, etc; you will need to create this variable name because it will be NA after separating), and columns for hrs.undertaken, hrs.fail, hrs.pass, grade.points, and gpa. There should be 17 columns total: `ID, fseut, derivation, female, SES, SAT, AP, CBE, graduated, order, semester, ccyys, hrs.undertaken, hrs.fail, hrs.pass, grade.points, gpa`. You will need to use pivot_longer(), separate(), and probably also pivot_wider(). DO NOT PRINT YOUR FINAL OUTPUT: instead, pipe it into `glimpse()`.**

```
## you might consider getting your code running with just the first 500 rows of your merged dataset (an

# first500 <- mergedat %>% slice(1:500)
```

```
#your code here
#and here
#and here
#...

#longdat%>%glimpse()
```

**3 (a) (1 pt) Take the resulting tidy dataset and emove all rows containing NAs (use this na-free dataset from here on unless otherwise noted). How many rows were lost?**

```
#your code here
```

*your answer here, 1-2 sentences*

**3 (b) (1 pt) Notice there is no single variable that uniquely identifies a row. Use `unite(...,remove=F)` to add a new variable `unique` that combines `ccyys` and `ID` into a unique identifier. Show that it is in fact unique (i.e., that there are no duplicates in this column).**

```
#your code here
```

**3 (c) (1 pt) Create a new variable called `year` by copying `ccyys` and then removing the fifth digit using `separate()`, or just by using `separate(..., remove=F)` without explicitly copying `ccyys`. The goal is 2008 instead of 20089, 2009 instead of 20092, etc. Keep the last number (9, 2, or 6) around in a column caled semester2 (this variable will make your life easier shortly). Pipe your output into `select(ID,ccyys,year,semester,semester2,ccyys)%>%glimpse()`**

```
#your code here
```

**3 (d) (2 pts) Again, after removing the NAs, create a new column with each student's *cumulative GPA* (call it `cum_gpa`) as of each semester (make sure data is sorted correctly before calculating cumulative statistics). Note that this is not as a simple as averaging the GPAs from each semester (think about an average of averages versus a weighted average). I would probably save this as something else rather than overwriting in case anything goes wrong. Pipe your output into `select(ID,ccyys,gpa,cum_gpa) %>% arrange(ID) %>% glimpse()`**

```
#your code here
```

**3 (e) (1 pt) What proportion of students took at least one summer class? You are advised to use `semester2` rather than `semester` to summarize etc. (it takes much less time).**

```
#your code here
```

*your answer here, 1-2 sentences*

**3 (f) (1 pt) What is the record/maximum for number of semesters attended without graduating? Which student ID has this distinction?**

```
#your code here
```

*your answer here, 1-2 sentences*

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.2.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.4.0   stringr_1.4.0   dplyr_0.8.3     purrr_0.3.3
##  [5] readr_1.3.1     tidyr_1.0.0     tibble_2.1.3    ggplot2_3.2.1
##  [9] tidyverse_1.3.0 knitr_1.28
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5 xfun_0.13        haven_2.2.0      lattice_0.20-35
##  [5] colorspace_1.4-1 vctrs_0.2.1      generics_0.0.2   htmltools_0.3.6
##  [9] yaml_2.2.0       rlang_0.4.2      pillar_1.4.2     glue_1.3.1
## [13] withr_2.1.2      DBI_1.0.0        dbplyr_1.4.2     modelr_0.1.5
## [17] readxl_1.3.1     lifecycle_0.1.0  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0 rvest_0.3.5      evaluate_0.14    broom_0.5.2
## [25] Rcpp_1.0.2       scales_1.0.0     backports_1.1.4  jsonlite_1.6
## [29] fs_1.3.1         hms_0.5.3        digest_0.6.20    stringi_1.4.3
## [33] grid_3.4.4       cli_1.1.0        tools_3.4.4      magrittr_1.5
## [37] lazyeval_0.2.2   crayon_1.3.4     pkgconfig_2.0.2  zeallot_0.1.0
## [41] xml2_1.2.2       reprex_0.3.0     lubridate_1.7.4  assertthat_0.2.1
## [45] rmarkdown_2.1    httr_1.4.1       rstudioapi_0.10  R6_2.4.0
## [49] nlme_3.1-131     compiler_3.4.4
```

```
## [1] "2020-05-13 10:51:31 CDT"
```

```
##                                      sysname
##                                      "Linux"
##                                      release
##                           "4.15.0-99-generic"
##                                      version
## "#100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020"
##                                     nodename
##                   "educcomp03.ccbb.utexas.edu"
##                                      machine
##                                     "x86_64"
##                                        login
##                                    "unknown"
##                                         user
##                                     "ndr432"
```

```
##                                 effective_user
##                                       "ndr432"
```