Fall 2018 Final Exam

QBS 181: Data Wrangling

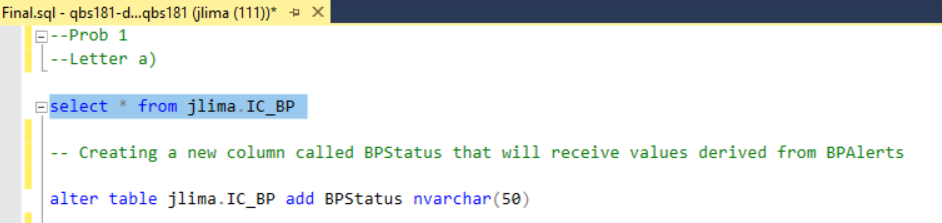
Due: 2PM, Monday, 11/19/2018

Instructions

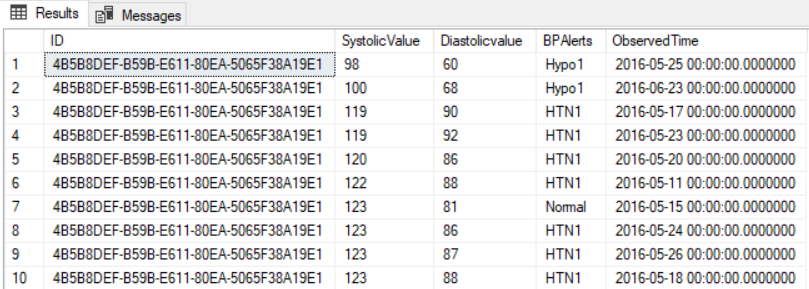
* For this final exam, you MUST work individually and NO collaboration is allowed. The Dartmouth Honor Principles apply (https://www.dartmouth.edu/student-handbook/).
* For each question, print random 10 rows to show these changes. Please also include the code with comments or descriptions explaining the steps
* For Canvas submission, Please name files which makes it easily understandable. For example: *last\_first\_qbs181\_final.pdf*
* In addition to submitting the final on Canvas, you are required to deposit your code for this exam and previous assignments on GitHub (see question 4 for detail). Please also include a README.md file to help readers understand the contents of your GitHub repository
* Include the link to your GitHub repository in your report while submitting the answers.

Questions

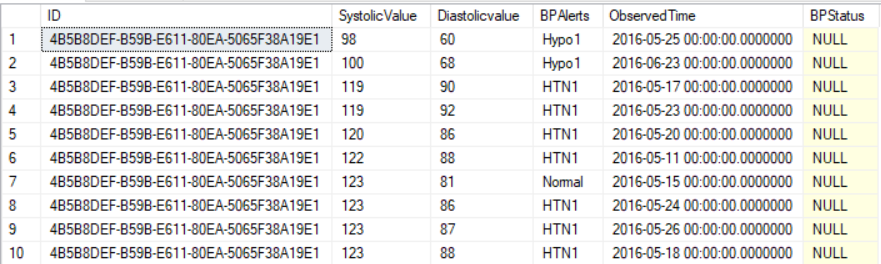
1. Consider the following blood pressure dataset (IC\_BP.csv). Perform the following operations
2. Convert BP alerts to BP status



Original table:

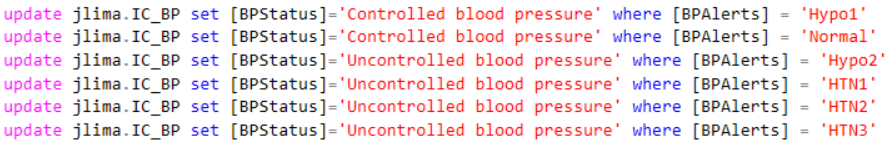


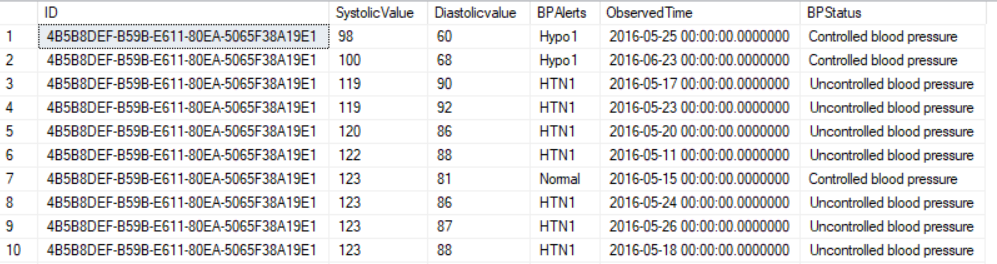
New Column added:



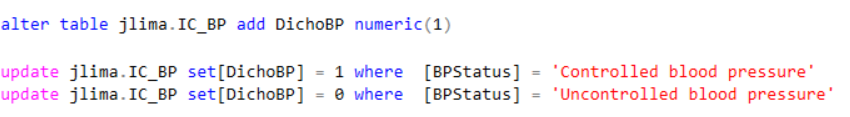
1. Define Hypotension-1 & Normal as Controlled blood pressure Hypotension-2, Hypertension-1, Hypertension-2 & Hypertension-3 as Uncontrolled blood pressure: Controlled & Uncontrolled blood pressure as 1 or 0 (Dichotomous Outcomes)

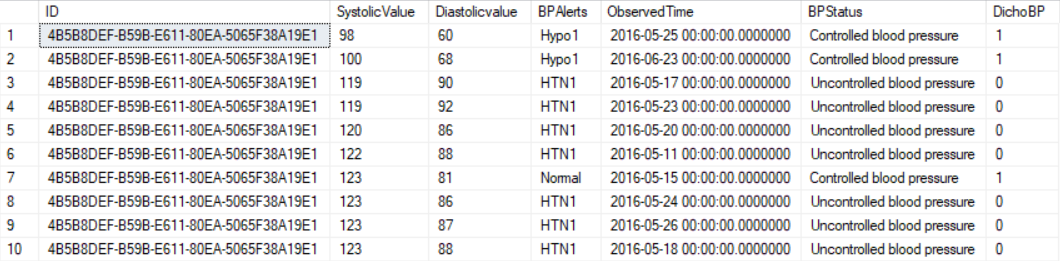
Populating the new column with the corresponding categories:





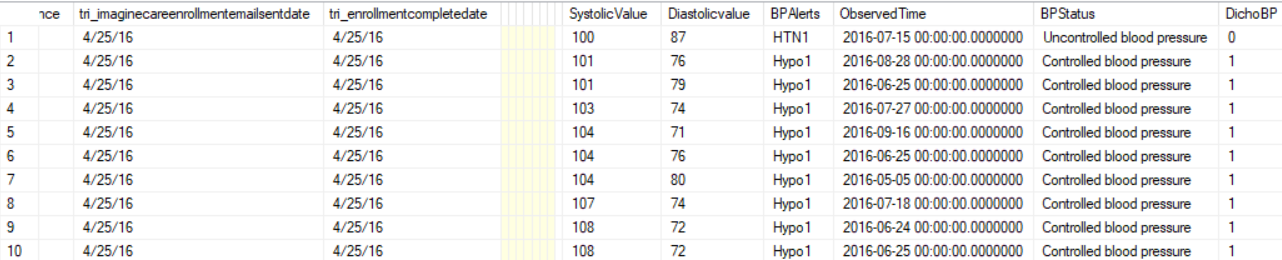
Creating and populating a new variable to receive the Dichotomous Outcomes values derived from the “BPStatus” column





1. Merge this table with demographics (SQL table) to obtain their enrollment dates

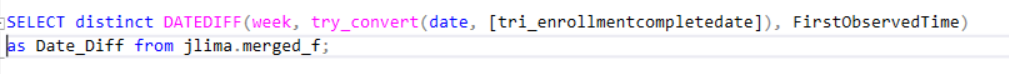


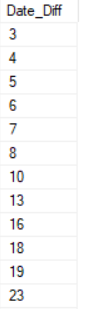


1. Create a 12-week interval of averaged scores of each customer

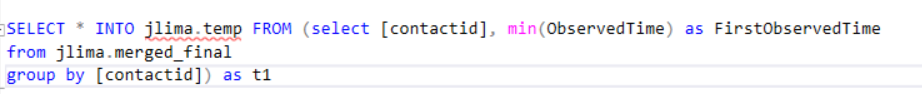
I initially tried to work with the date difference between variables “tri\_enrollmentcompletedate” and “Observed time”.

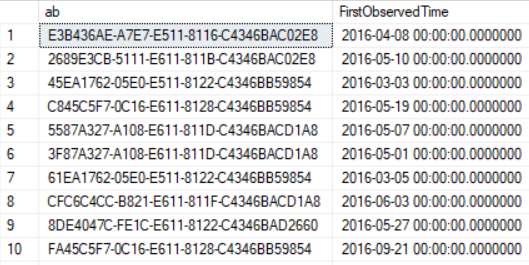
Since many people have their first measurement only several weeks after they enroll online that gave me bad results. That is the reason I chose to only work with date differences based on “ObservedTime”.





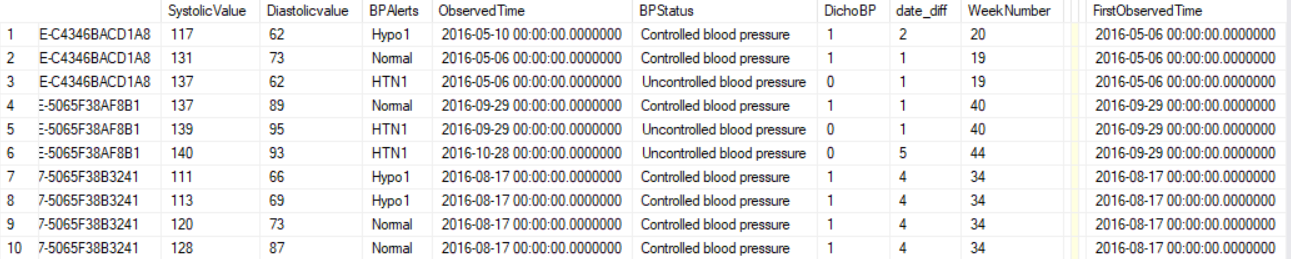
Creating a temporary table to contain the values of patient ID and first Observed Time (which equals to the first time of measurement of the blood pressure)





Merging the tables together so I have my start point for counting time on FirstObservedTime:

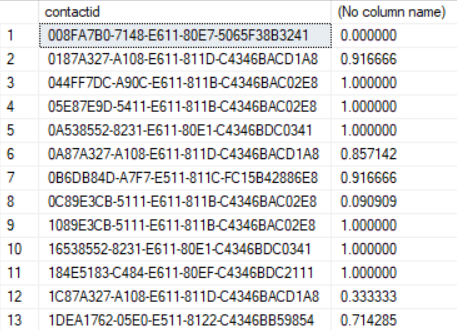




Creating the averages from the weekly intervals:

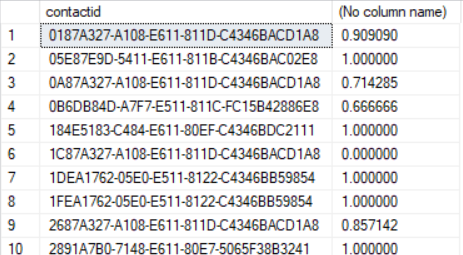
Average from the start point (minimum observed time) to 1 week later for each patient





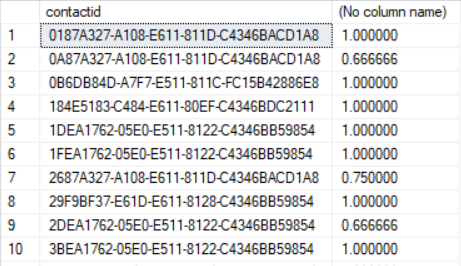
Average from week 1 from week 2 for each patient:





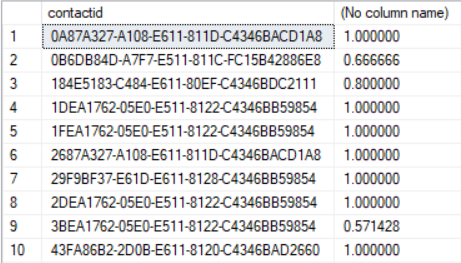
Average from week 2 from week 3 for each patient:





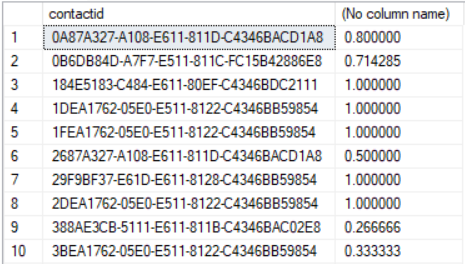
Average from week 3 from week 4 for each patient:





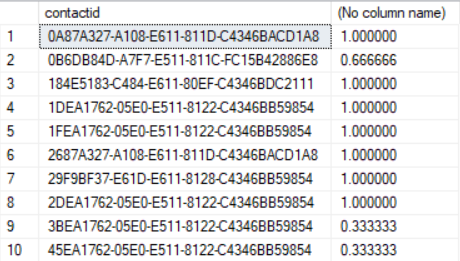
Average from week 4 from week 5 for each patient:





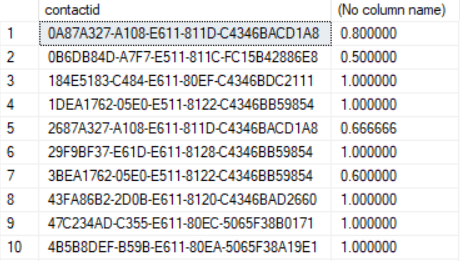
Average from week 5 from week 6 for each patient:





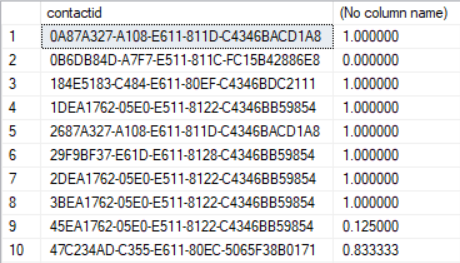
Average from week 6 from week 7 for each patient:





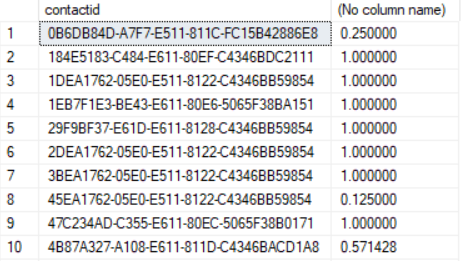
Average from week 7 from week 8 for each patient:





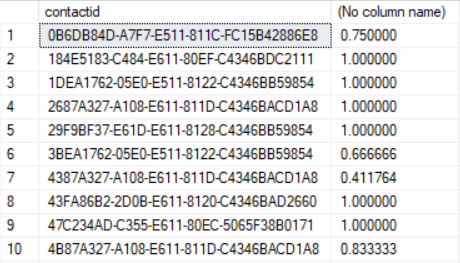
Average from week 8 from week 9 for each patient:





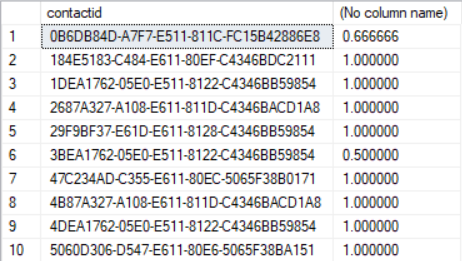
Average from week 9 from week 10 for each patient:





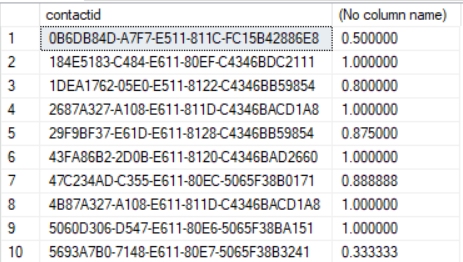
Average from week 10 from week 11 for each patient:





Average from week 11 from week 12 for each patient:

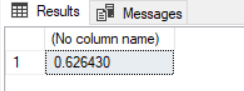




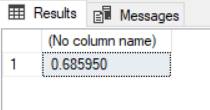
1. Compare the scores from baseline (first week) to follow-up scores (12 weeks)

I will compare the average off all the patient scores on the first interval (week 0-1) to the average of all the scores for the last interval (week 11-12):





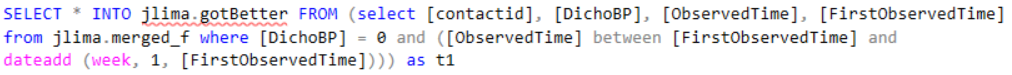


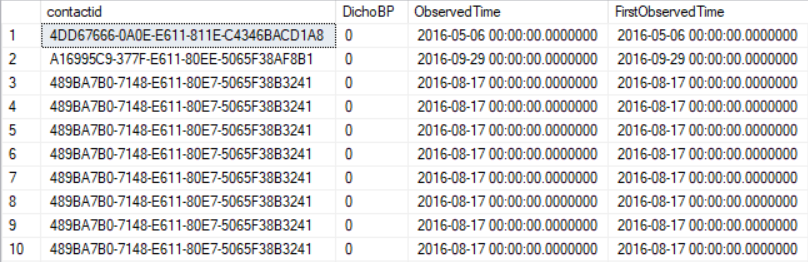


The average scores for all patients increased from the beginning of the program towards the end. That means that on average the number of measurements were better (more people with 1 = Controlled Blood Pressure)

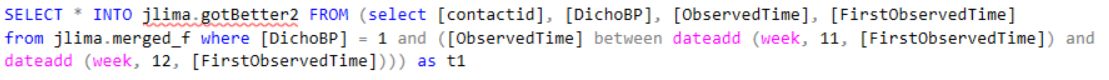
1. How many customers were brought from uncontrolled regime to controlled regime after 12 weeks of intervention?

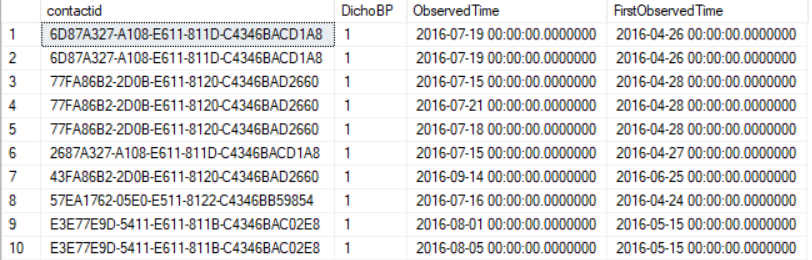
Making a new table with the patients that were marked as uncontrolled blood pressure (=0) at the first time interval (week 0-1):



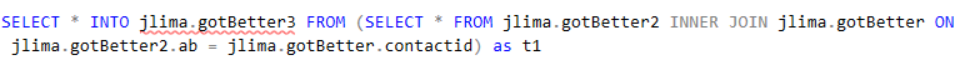


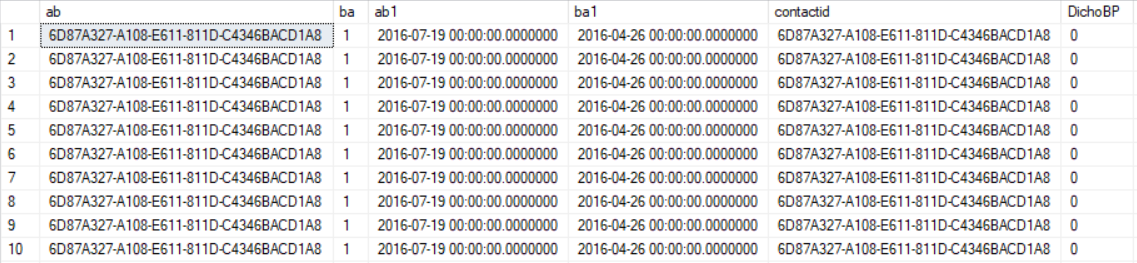
Making a new table with the patients that were marked as controlled blood pressure (=1) at the last time interval (week 11-12):





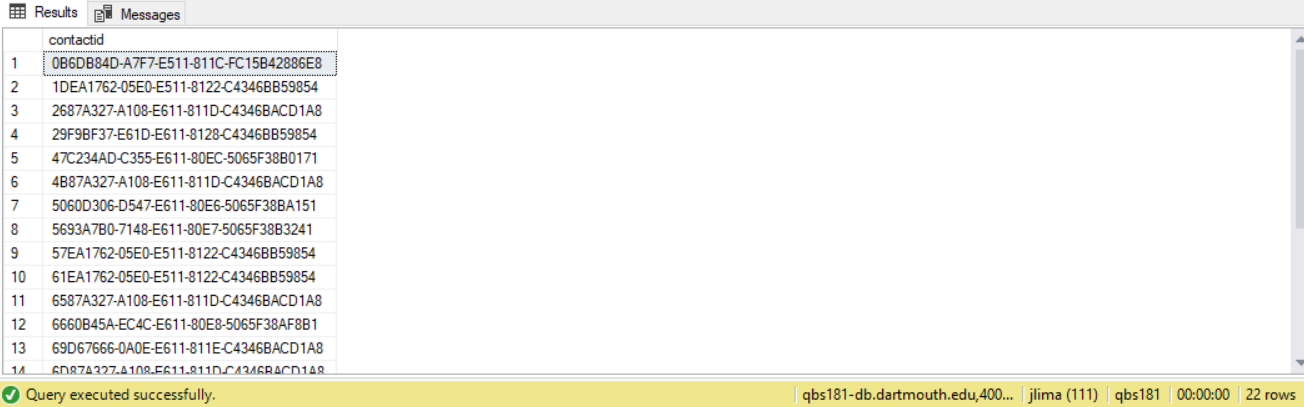
Merging the two tables together so everyone that had uncontrolled blood pressure at the first time interval and controlled blood pressure at the last time interval are in the same table:





Selecting the distinct IDs that are present in the merged table:





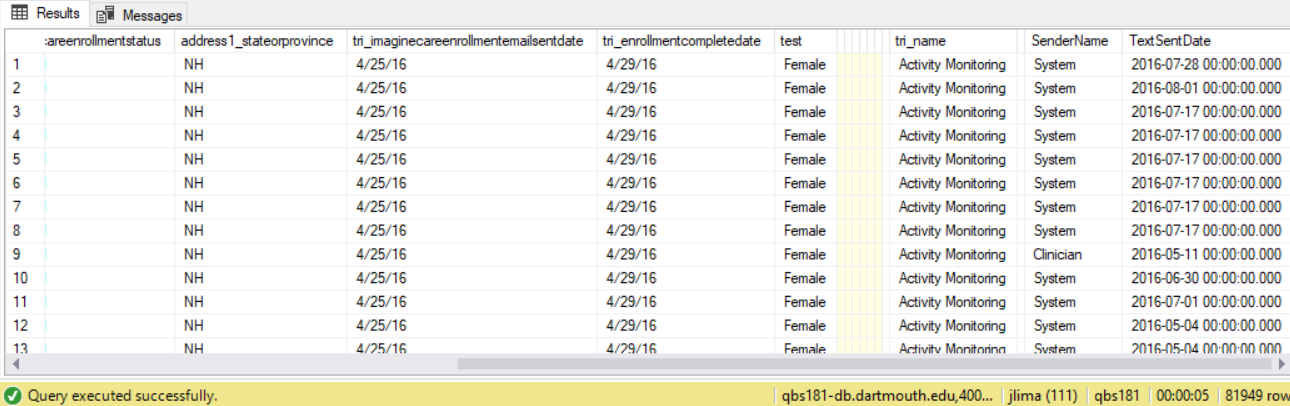
We can see there are 22 rows which means 22 people were uncontrolled (=0) in the first period (week 0-1) and became controlled (=1) in the last period considered (week 11-12).

1. Merge the tables Demographics, Chronic Conditions and TextMessages.

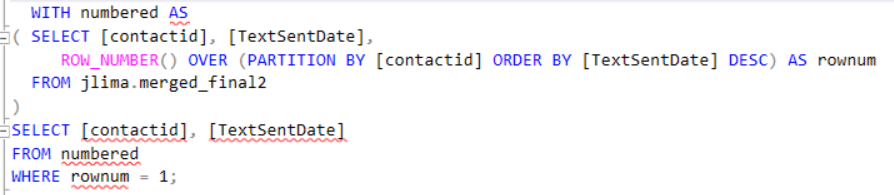
Obtain the final dataset such that we have 1 Row per ID by choosing on the latest date when the text was sent (if sent on multiple days)

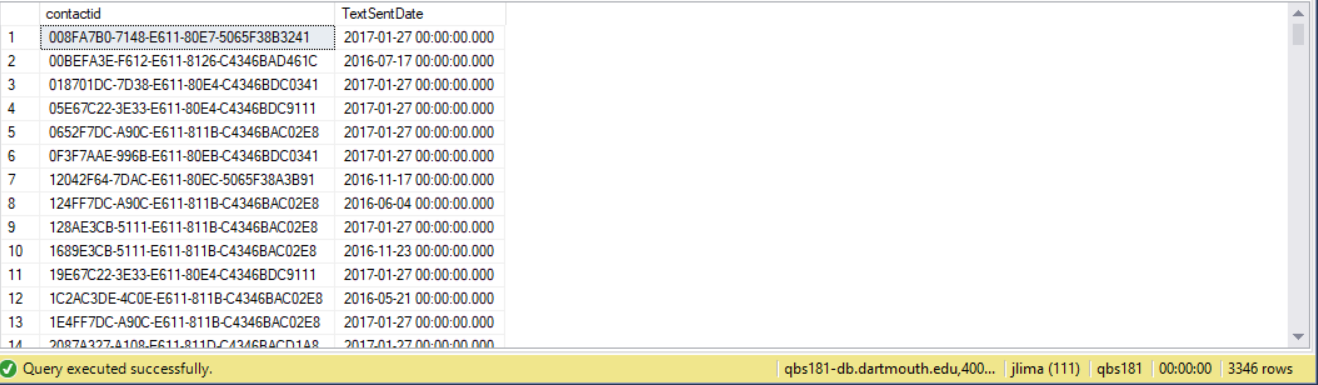
Merging tables Demographics, Chronic Conditions and TextMessages:





Choosing the latest date when the text was sent so I have 1 Row per ID:





1. Repeat Question 2 in R. The answer should reflect use of ODBC drivers to connect to SQL raw tables and import data into the R environment before the wrangling process. Hint: You might want to use tidyr/dplyr packages

Importing the data using the RODBC package:

library("RODBC")  
myconn<- odbcConnect("dartmouth1", "jlima", "jlima@qbs181")  
demographics <- sqlQuery(myconn, "select \* from [dbo].[Demographics]")  
ChronicConditions <- sqlQuery(myconn, "select \* from [dbo].[ChronicConditions]")  
Text <- sqlQuery(myconn, "select \* from [dbo].[Text]")

Merging tables Demographics, Chronic Conditions and TextMessages:

library(tidyverse)

## -- Attaching packages ------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ---------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

colnames(ChronicConditions)[1] <- "contactid"  
str(ChronicConditions)

## 'data.frame': 11465 obs. of 2 variables:  
## $ contactid: Factor w/ 9709 levels "0023FCAF-2616-E611-8128-C4346BB59854",..: 1242 1548 1626 1774 2308 2693 2925 3001 3380 3765 ...  
## $ tri\_name : Factor w/ 5 levels "Activity Monitoring",..: 5 3 5 5 5 5 5 5 5 5 ...

merged\_final <- inner\_join(demographics, ChronicConditions, by = "contactid")

## Warning: Column `contactid` joining factors with different levels, coercing  
## to character vector

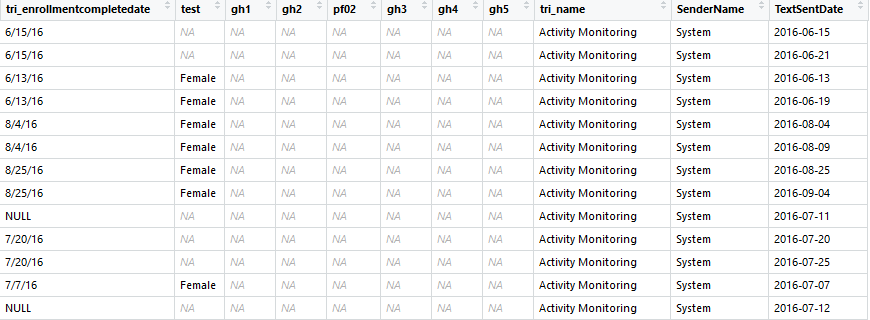
colnames(Text)[1] <- "contactid"  
merged\_final <- inner\_join(merged\_final, Text, by = "contactid")

## Warning: Column `contactid` joining character vector and factor, coercing  
## into character vector

View(merged\_final)  
str(merged\_final)

## 'data.frame': 81949 obs. of 18 variables:  
## $ contactid : chr "B917A194-E92C-E611-80E2-5065F38BA151" "B917A194-E92C-E611-80E2-5065F38BA151" "CF17A194-E92C-E611-80E2-5065F38BA151" "CF17A194-E92C-E611-80E2-5065F38BA151" ...  
## $ gendercode : Factor w/ 4 levels "1","167410000",..: 4 4 3 3 3 3 3 3 4 1 ...  
## $ tri\_age : num 48 48 37 37 36 36 63 63 28 62 ...  
## $ parentcontactidname : Factor w/ 1 level "NULL": 1 1 1 1 1 1 1 1 1 1 ...  
## $ tri\_imaginecareenrollmentstatus : num 1e+08 1e+08 1e+08 1e+08 1e+08 ...  
## $ address1\_stateorprovince : Factor w/ 43 levels "AK","AL","AZ",..: 39 39 39 39 39 39 39 39 24 39 ...  
## $ tri\_imaginecareenrollmentemailsentdate: Factor w/ 132 levels "1/12/17","1/18/17",..: 87 87 87 87 105 105 105 105 105 105 ...  
## $ tri\_enrollmentcompletedate : Factor w/ 248 levels "1/10/17","1/11/17",..: 137 137 135 135 214 214 206 206 248 170 ...  
## $ test : Factor w/ 1 level "Female": NA NA 1 1 1 1 1 1 NA NA ...  
## $ gh1 : logi NA NA NA NA NA NA ...  
## $ gh2 : logi NA NA NA NA NA NA ...  
## $ pf02 : logi NA NA NA NA NA NA ...  
## $ gh3 : logi NA NA NA NA NA NA ...  
## $ gh4 : logi NA NA NA NA NA NA ...  
## $ gh5 : logi NA NA NA NA NA NA ...  
## $ tri\_name : Factor w/ 5 levels "Activity Monitoring",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ SenderName : Factor w/ 3 levels "Clinician","Customer",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ TextSentDate : POSIXct, format: "2016-06-15" "2016-06-21" ...

Result from View(merged\_final):



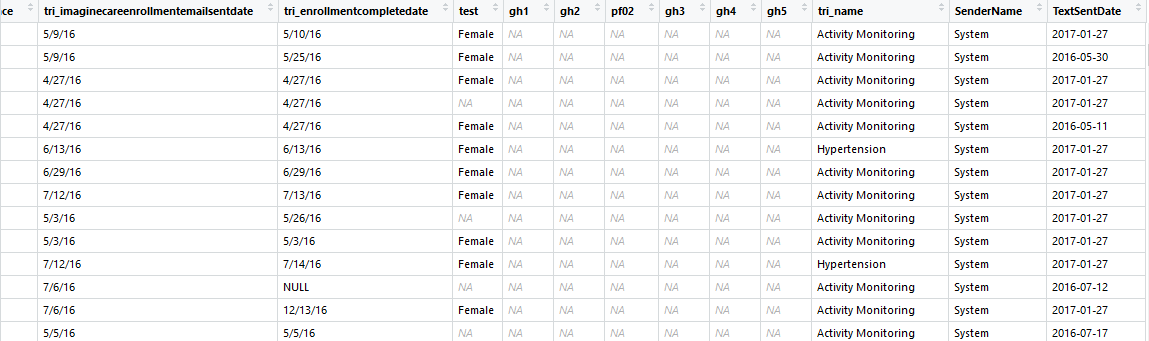
Choosing the latest date when the text was sent so I have 1 Row per ID:

The final result has the same number of rows of the table in SQL (3346)

selected\_merged\_final <- group\_by (merged\_final, contactid)  
selected\_merged\_final <- slice(selected\_merged\_final, which.max(TextSentDate))  
View(selected\_merged\_final)  
str(selected\_merged\_final)

## Classes 'grouped\_df', 'tbl\_df', 'tbl' and 'data.frame': 3346 obs. of 18 variables:  
## $ contactid : chr "0023FCAF-2616-E611-8128-C4346BB59854" "0046C5F7-0C16-E611-8128-C4346BB59854" "004FF7DC-A90C-E611-811B-C4346BAC02E8" "0050F7DC-A90C-E611-811B-C4346BAC02E8" ...  
## $ gendercode : Factor w/ 4 levels "1","167410000",..: 3 3 3 1 3 3 3 3 1 3 ...  
## $ tri\_age : num 21 38 40 44 40 47 50 25 48 37 ...  
## $ parentcontactidname : Factor w/ 1 level "NULL": 1 1 1 1 1 1 1 1 1 1 ...  
## $ tri\_imaginecareenrollmentstatus : num 1.67e+08 1.67e+08 1.67e+08 1.67e+08 1.67e+08 ...  
## $ address1\_stateorprovince : Factor w/ 43 levels "AK","AL","AZ",..: 24 24 24 24 24 24 15 24 24 39 ...  
## $ tri\_imaginecareenrollmentemailsentdate: Factor w/ 132 levels "1/12/17","1/18/17",..: 71 71 50 50 50 75 84 91 67 67 ...  
## $ tri\_enrollmentcompletedate : Factor w/ 248 levels "1/10/17","1/11/17",..: 101 117 96 96 96 135 149 162 118 122 ...  
## $ test : Factor w/ 1 level "Female": 1 1 1 NA 1 1 1 1 NA 1 ...  
## $ gh1 : logi NA NA NA NA NA NA ...  
## $ gh2 : logi NA NA NA NA NA NA ...  
## $ pf02 : logi NA NA NA NA NA NA ...  
## $ gh3 : logi NA NA NA NA NA NA ...  
## $ gh4 : logi NA NA NA NA NA NA ...  
## $ gh5 : logi NA NA NA NA NA NA ...  
## $ tri\_name : Factor w/ 5 levels "Activity Monitoring",..: 1 1 1 1 1 5 1 1 1 1 ...  
## $ SenderName : Factor w/ 3 levels "Clinician","Customer",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ TextSentDate : POSIXct, format: "2017-01-27" "2016-05-30" ...  
## - attr(\*, "vars")= chr "contactid"  
## - attr(\*, "drop")= logi TRUE  
## - attr(\*, "indices")=List of 3346  
## ..$ : int 0  
## ..$ : int 1  
## ..$ : int 2  
## ..$ : int 3  
## ..$ : int 4  
## ..$ : int 5  
## ..$ : int 6  
## ..$ : int 7  
## ..$ : int 8  
## ..$ : int 9  
## ..$ : int 10  
## ..$ : int 11  
## ..$ : int 12  
## ..$ : int 13  
## ..$ : int 14  
## ..$ : int 15  
## ..$ : int 16  
## ..$ : int 17  
## ..$ : int 18  
## ..$ : int 19  
## ..$ : int 20  
## ..$ : int 21  
## ..$ : int 22  
## ..$ : int 23  
## ..$ : int 24  
## ..$ : int 25  
## ..$ : int 26  
## ..$ : int 27  
## ..$ : int 28  
## ..$ : int 29  
## ..$ : int 30  
## ..$ : int 31  
## ..$ : int 32  
## ..$ : int 33  
## ..$ : int 34  
## ..$ : int 35  
## ..$ : int 36  
## ..$ : int 37  
## ..$ : int 38  
## ..$ : int 39  
## ..$ : int 40  
## ..$ : int 41  
## ..$ : int 42  
## ..$ : int 43  
## ..$ : int 44  
## ..$ : int 45  
## ..$ : int 46  
## ..$ : int 47  
## ..$ : int 48  
## ..$ : int 49  
## ..$ : int 50  
## ..$ : int 51  
## ..$ : int 52  
## ..$ : int 53  
## ..$ : int 54  
## ..$ : int 55  
## ..$ : int 56  
## ..$ : int 57  
## ..$ : int 58  
## ..$ : int 59  
## ..$ : int 60  
## ..$ : int 61  
## ..$ : int 62  
## ..$ : int 63  
## ..$ : int 64  
## ..$ : int 65  
## ..$ : int 66  
## ..$ : int 67  
## ..$ : int 68  
## ..$ : int 69  
## ..$ : int 70  
## ..$ : int 71  
## ..$ : int 72  
## ..$ : int 73  
## ..$ : int 74  
## ..$ : int 75  
## ..$ : int 76  
## ..$ : int 77  
## ..$ : int 78  
## ..$ : int 79  
## ..$ : int 80  
## ..$ : int 81  
## ..$ : int 82  
## ..$ : int 83  
## ..$ : int 84  
## ..$ : int 85  
## ..$ : int 86  
## ..$ : int 87  
## ..$ : int 88  
## ..$ : int 89  
## ..$ : int 90  
## ..$ : int 91  
## ..$ : int 92  
## ..$ : int 93  
## ..$ : int 94  
## ..$ : int 95  
## ..$ : int 96  
## ..$ : int 97  
## ..$ : int 98  
## .. [list output truncated]  
## - attr(\*, "group\_sizes")= int 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "biggest\_group\_size")= int 1  
## - attr(\*, "labels")='data.frame': 3346 obs. of 1 variable:  
## ..$ contactid: chr "0023FCAF-2616-E611-8128-C4346BB59854" "0046C5F7-0C16-E611-8128-C4346BB59854" "004FF7DC-A90C-E611-811B-C4346BAC02E8" "0050F7DC-A90C-E611-811B-C4346BAC02E8" ...  
## ..- attr(\*, "vars")= chr "contactid"  
## ..- attr(\*, "drop")= logi TRUE

View(selected\_merged\_final):



1. Set up a public GitHub repository to share your code. If you didn’t attend the Research Computing’s Git/BASH workshop, or no longer familiar with it, please use online resources (e.g. YouTube tutorials) to re-familiarize yourself with Git/GitHub.
   * Create an account on GitHub, if you haven’t done so. For this final, you are required to use the GitHub and make the repository public.
   * Create a new repository called “Data\_Wrangling\_Project\_and\_Tasks”. This repository should be a public, stand-alone repository.
   * Create sub-directories named “project1”, “project2”, “project3”, “midterm\_project”, “final\_project”, etc.
   * Copy your SQL and R code for each task into their respective repository.
     + Please include R and SQL code only.
     + Important: Do NOT include data, intermediate results, or your
   * Include a single *README.md* file for the entire repository, containing:
     + Description of the purpose of the repository.
     + A statement (1 sentence) that you have been given the permission to make your work public
     + A description of each sub-directory (you do not need to explain every file; just every task
   * Provide the link of your GitHub repository.

Example: <https://github.com/ydavidchen/data_wrangling_projects>

Github Link:

https://github.com/jorflima/Data\_Wrangling\_Project\_and\_Tasks

