AN ANALYSIS OF WOMEN WITH TYPE 2 DIABETES MELLITUS (T2DM) ON DIABETES SELF-CARE, DIABETES TIME MANAGEMENT, AND DIABETES DISTRESS

BY

TAYLOR BYERS, DONGJIAN HU, EMILY RUSACK, JOHN ZINZER

Actuarial Science Senior Capstone Project

Arcadia University

2019

Table of Contents

Introduction

The Centers for Disease Control and Prevention (CDC) reports that 9.4 percent of the United States' population have diabetes, either diagnosed or undiagnosed. Within the diabetes population, 90 to 95 percent are diagnosed with Type 2 Diabetes Mellitus (T2DM). Side effects of T2DM can lead to heart disease, stroke, and even death, as it is the seventh leading cause of death (Centers for Disease Control and Prevention, 2014).

Measuring glycosylated hemoglobin (HbA1c) determines how much glucose is in the blood and the severity of the diabetes the patient has. Patients with T2DM must continuously measure and maintain their HbA1c levels to reduce diabetes-related complications. This is why self-care becomes a vital skill that should be instilled in patients with T2DM. Diabetes self-care is having the ability to manage treatment guidelines and maintain overall health (Funnell, 2010).

Of the 9.4 percent of the population who have T2DM, 12.6 million of them are women. Women have a greater risk for diabetes complications than men, as well as different ethnicities having a greater risk than one another (Diabetes and Women, 2018). Without proper self-care the complications can be life-threatening, and so two factors that play a role on self-care are time management and distress.

Time management is a skill that is crucial to maintaining diabetes self-care. Arranging regular doctors appointments, testing and recording HbA1c and planning healthy nutrition are just a few examples of what diabetic individuals must keep track of. Organizing and planning daily routines becomes vital to maintain. Diabetes related distress is another factor that influences self-care. Distress is an extreme tension that can be brought upon patients, specifically women, from diabetes due to the life-changing and complex nature of the condition. Research has shown that the more distress a person with diabetes has, the higher their HbA1c levels (Fisher et al., 2009). This concludes that both time management and distress have a relationship with self-care that are essential in reducing diabetes related complications.

Literature Review

Diabetes self-care, time management and distress can be measured through three different survey and scoring methods. Self-care was analyzed in the 1950s by Dorothea Orem who developed the self-care theory. Orem's self-care theory states that self-care is a human regulatory function, defined by Orem the actions performed by individuals to regulate his or her own functioning (Orem, 1995).

The diabetes self-care was measured through a the Diabetes Self-Management Questionnaire (DSMQ), which consists of 16 items separated into 5 subscales measuring dietary control, blood glucose monitoring, physical activity, and physician contact. DSMQ was originally developed by a group of researchers in 2013 and has since been validated making it a good measurement (Schmitt, 2016).

Time management by those with diabetes was measured with the Diabetes Time Management Questionnaire (DTMQ). DTMQ is a 49 item questionnaire to assess general time management skills and those relevant to comply with a diabetes healthcare regimen. The measure was developed in 1999 by a research team. The research team tested the validity of the instrument against two other well tested instruments the Habits, Attitudes, and Knowledge Questionnaire of Diabetic Compliance (HAK) and the Diabetes Knowledge Schedule (DKS). DTMQ was proven to be a valid and reliable measure (Gafarian, 1999).

Diabetes related distress was measured with the Diabetes Distress Scale (DDS). DDS was created with consultation from patients, diabetes nurse specialists, dietitians, diabetologists, and diabetes-knowledgeable psychologists from around the country. The scale was developed in 2005 and contained four subscores analyzing emotional burden, physician distress, regimen distress, and interpersonal distress. The scale was validated after the study and proven a good measure (Polonsky, 2005).

Problem Statement

Diabetes distress is often associated with less self-care, significantly in women. There are no current studies analyzing the effect that diabetes related distress has on women's self-care. Relationships between time management and diabetes self-care has no published research either. There is a lack of information about the relationships between the three variables, hence our overarching research question through this study is to find the relationships among diabetes self-care, diabetes time management, and diabetes distress in women with T2DM.

Data Description

Two types of surveys were distributed in order to collect appropriate information for this study. An online survey was sent via e-mail across the country, and a pencil/paper version of the same survey was completed in local physician centers in Montgomery and Bucks County, Pennsylvania. There were initial questions to validate that participants were women above the age of 18, had type II diabetes Mellitus for one or more years, and were not pregnant. Each survey was comprised of three sections relating to diabetes self management, time management, and diabetes related distress, along with demographic questions at the end. The electronic survey had a few extra questions compared to the pencil/paper version. These questions included information such as email addresses, time taken on the survey, and the date it was taken. This information was not important to the study.

There were four possible responses in the diabetes self management section, which ranged from “Does not apply to me” to “Applies to me very much”. In order to rate these responses, a scale from 0 - 3 was developed, where 0 corresponds to “Does not apply to me” and 3 corresponds to “Applies to me very much.” The higher the score, the better self management the participant has. However, there were certain statements which signified poor self management. Due to this, these statements were reversed scored.

The diabetes time management section had five possible responses, which ranged from “Always” to “Never.” These responses were rated on a scale from 1 - 5, where 1 corresponds to “Always” and 5 corresponds to “Never”. For this section, the higher the score, the worse time management skills of the participant. There were also statements in this section which signified poor time management. These responses were therefore reversed scored.

Finally, the diabetes distress section had six possible responses, ranging from “Not a problem” to “A very serious problem.” These responses were rated on a scale from 1 - 6, where 1 corresponds to “Not a problem” and 6 corresponds to “A very serious problem”. Similar to the time management section, the higher the score for distress, the worse it is, or the greater distress. There was no reversed scored statements in this section.

These responses were recorded in a data dictionary. This dictionary contains the numeric values with the corresponding phrases for each of the three section. It also entails the demographic data and the overall scores for DSMQ, DDS, and DTMQ.

Data Preparation

Since there were two different versions of the survey, there were slight inconsistencies among them. The responses had spelling errors and the use of uppercase and lowercase letters was not constant with every question. These errors were corrected in order to efficiently translate the phrases into a numerical format. Using the programming language, R, all of the possible responses to the questions in the dataset were found. These responses were then mapped to the phrases that corresponded with each answer. These phrases were then transformed into numbers, which correspond to the aforementioned scales. The demographic data at the end of the surveys was unified and recorded as strings, or words, for better interpretation during analysis. Dummy columns were also created for the caregiver role of a participant, since the participants could select multiple caregiver roles, such as self, child, parent, and spouse. Dummy columns include a 0, if the participant does not have a specific caregiver role, and a 1, if the participant does have a specific caregiver role.

The surveys which were invalid, meaning that no questions were answered, were deleted from the dataset. Then, those who were ineligible because they answered false to any of the initial seven questions were also deleted. After this, we removed those who abandoned the survey after answering the eligibility questions correctly. The original dataset contained 277 participants. Of the participants, 25 were invalid, 46 were ineligible, and 13 had abandoned the survey. This reduced the dataset to 193 subjects. After careful analysis of the data, it was found that 5 of the remaining subjects were outliers due to their extreme scores. For that reason, they were excluded from the final dataset. The final amount of subjects was 188 after all of the deletions were complete.

Although most questions were answered in the remaining surveys, there were still some missing responses in the dataset. In order to accurately predict these responses, the KNN classification method was utilized. This method finds the participants who are most closely related, in terms of responses, to the participant with a missing response. The missing value is then fulfilled by the average response that the like participants recorded for that specific question. Once the KNN process had run, both datasets were complete with all information necessary.

The additional questions included in the electronic dataset which were not in the paper dataset were removed. Once each dataset had the same amount of columns, they were checked again to assure all the questions matched in each version and then they were re-numbered in a new format. The column names included the question number and the section from which the response came. The letter “R” was also added to the questions that were reverse scored. After both datasets were reconciled, they were merged together using R. From each section of the survey, we needed to convert each question from string to numeric values to be able to properly score each participant’s survey (excluding the Eligibility section since that was to only look for qualified participants and the Demographics section that was not part of the specified scoring questions).

The final step for data preparation was to calculate the scores for each section. The Diabetes Self Management Questionnaire (DSMQ) score contains the total score for all responses in relation to the total possible score for the section, multiplied by 10. This results in a score ranging from 0 - 10, where 10 is the best possible score indicating perfect self management (Schmitt, 2016). The Diabetes Time Management Questionnaire (DTMQ) score was calculated in the same manner as the DSMQ score. Thus, the results were between 1 - 10, but 10 is the worst possible score indicating no time management skills (Gafarian, 1999). Finally, the Diabetes Distress Scale (DDS) score was the calculated average of all the responses in the distress section. This score ranges from 1 - 6, where 6 is the worst possible score indicating the most possible distress (Polonsky, 2005). Scores for the subscales of the DSMQ and DDS and the were also calculated for review. These scores corresponded to the calculations of their relative sections. For the DSMQ subscores, they followed the DSMQ scoring developed by Schmitt (2016). The DDS subscores followed the DDS scoring which was developed by Polonsky (2005).

Data Exploration

All the preliminary tests and the following models were ran through SPSS and were examined at a p-value of .05. The p-value indicates the probability of rejecting the null hypothesis when it is actually true. Thus, if this value is lower than .05, then the results are significant, and the null hypothesis is rejected.

Table
Mean, Standard Deviation, Median, Range, and Cronbach's Alpha of the Diabetes Self-Management Questionnaire, Diabetes Time Management, and Diabetes Distress Scale (N = 188).

Instrument	Range	Median	Mean	SD	Alpha
Diabetes Self Care	5.8	7.29	7.11	1.40	.791
Subscale Dietary Control	10	5.83	5.55	1.98	.664
Subscale Glucose Management	10	8.0	7.83	1.85	.669
Subscale Physical Activity	10	6.66	6.01	2.74	.743
Subscale Physician Contact	7.77	10	8.97	1.76	.527
Diabetes Time Management	4.2	5.02	4.98	0.83	.892
Diabetes Distress Scale	5	1.94	2.24	1.05	.938
Subscale Emotional Burden	5	2.2	2.45	1.28	.905
Subscale Physician Distress	5	1	1.57	1.15	.920
Subscale Regimen Distress	5	2.2	2.61	1.35	.901

Subscale Interpersonal Distress	5	1.66	2.18	1.37	.779

To begin the data exploration, we looked at the breakdown of all of the total scores and subscale scores that were included in the survey. The sample's diabetes self-care reported effective self care (M = 7.11, SD = 1.40), since the score is on a 1-10 scale with 10 being the best score. For diabetes time management, it was reported an average level of time management skills (M = 4.98, SD = 0.83), where the score is on a 1-10 scale with 10 being the worst score. Lastly, diabetes distress was reported with a moderate level of distress (M = 2.24, SD = 1.05), with the the score on a 1-3 scale where 3 is the highest level of distress.

Table
Means, Standard Deviations, and bivariate correlations (Pearson's) for main study variables (N=188)

Variable	M	SD	1	2	3
1. Diabetes Self Care	7.1	1.4	-	-.605**	-.331**
2. Diabetes Time Management	4.9	0.83	-.605**	-	.394**
3. Diabetes Distress	2.2	1.05	-.331**	.394**	-

**$p < 0.01$ level

Initially a Pearson's correlation test was ran to look at the relationships between diabetes self care, diabetes self management and diabetes distress. Preliminary analysis was tested and there were no violations of homoscedasticity. Self care and time management had a strong negative relationship, where r = -0.605, meaning that as self care decreases time management decreases. Additionally, there was a medium negative correlation between self care and distress, where r = -0.331, meaning that as self care decreases, distress increases. Lastly, between time management and distress there was a medium positive correlation indicating that inefficient time management skills are associated with higher levels of stress, showing r = 0.394.

Table
Pearson's Correlations among subscales (n=188)

Variable	1	2	3	4	5	6	7	8	9
1. DSMQ Glucose Management	1	.367**	.215**	.303**	-.117	.022	-.399**	.002	-.435**
2.DSMQ Dietary Control	.367**	1	.425**	.075	-.202**	.029	-.456**	-.145*	-.439**
3.DSMQ Physical Activity	.215**	.425**	1	.047	-.116	.019	-.308**	-.085	-.449**
4.DSMQ Physician Contact	.303**	.075	.047	1	-.091	-.239**	-.215**	-.073	-.195**
5.DDS Emotional Burden	-.117	-.202**	-.116	-.091	1	.396**	.740**	.665**	.346**
6.DDS Physical Distress	.022	.029	.019	-.239**	.396**	1	.388**	.452**	.137
7.DDS Regimen Distress	-.399**	-.456**	-.308**	-.215**	.740**	.388**	1	.570**	.472**
8. DDS Interpersonal Distress	.002	-.145*	-.085	-.073	.665**	.452**	.570**	1	.246
9. DTMQ	-.435**	-.439**	-.449**	-.195**	.346**	.137	.472**	.246	1

** $p < .01$ level

There are a few statistically significant correlations between the subscores. However, the correlation with the most significance is found between the DDS subscore Regimen Distress and the DSMQ subscore Dietary Control, $p < .01$. This is the most significant change between a DSMQ subscore and a DDS subscore. With every unit increase in the Regimen Distress score, the participant's Dietary Control score decreases by .456 points.

To continue with data exploration, we then ran ANOVA tests with the DSMQ score against the demographic variables. This test was necessary in order to see if there are differences between each of the categories of demographic variables. The homogeneity of variances assumption was tested and verified for all ANOVA tests. The null hypothesis for these tests was that the means in all groups were equal.We found that the means of the DSMQ scores for employment status had a p-value of 0.002, meaning that at least one of the groups of respondents had scores different from others. The post hoc test results for this ANOVA concluded that the employment answers that differed were full time employment and part time (p-value = 0.017), full time and retired (p-value = 0.037), and full time and unemployed (p-value = 0.004). Additionally, it was found that for work environment there is significance of the means differing between answers with a p-value of .015, however by looking at the post hoc test, the differences between answers were not significant enough tested against our p-value level of .05. Lastly, we found that it was interesting that the last HbA1C level did not lead to a significant difference between the means of self management score that we thought, having a p-value of .381. We regrouped HbA1C into two categories: 6.5 and less and 6.6 and greater. This is because there was no significance between the means and we wanted to see if there was any differences in the means between each of these two categories. These two regrouping categories (6.5 and less and 6.6 and greater) were divided in that manner because according to the Centers for Disease and Control Prevention (CDC), an HbA1C level of 6.5 is the start of having diabetes. A level of 5.7 to 6.4 indicates prediabetes, and below 5.7 is a normal range (All About Your A1C, 2018). So, having this regrouping be 6.5 and below and 6.6 and above matched the CDC's ranges of diabetes. It was found that there is a slight difference between the means ($p = .05$).

Next, the DDS total score was used as the dependent variable in the ANOVA test with the factors being the demographic variables. The homogeneity of variances assumption was tested and verified for all ANOVA tests. We found that some of the means for the work environment were significantly different, indicated by a p-value of 0.010. The answers that had a significantly different score between each other were somewhat favorable and not applicable (p-value = 0.039). We looked at HbA1C against DDS total. It was not significantly different with a p-value of .735. Similarly, we then used the regrouped HbA1C as done with DSMQ and found it was not significantly different with a p-value of .198. It was also found that there was no significant difference of the means (p-value = .717) in the children caregiver role. We found this to be interesting because usually children cause more stress, and the means of the answers were about the same. It was estimated that since the majority of women in this study were older, their children were older, making the care they need less stressful to fulfill. The children caregiver role was also run in an ANOVA with DTMQ and found similar findings (p-value = .976).

MANOVA tests were then run in order to examine between which demographic groups the DSMQ subscores differed. These tests concluded that there was a statistically significant difference in the DSMQ subscores based on the participant's work environment, $p = .026$. Specifically, a participant's work environment has a statistically significant effect on the Glucose Management subscore, $p = .024$. It was also discovered that variables such as neglecting self care and diabetes status had a statistically significant effect on DSMQ subscores. This result is as expected.

Additional MANOVA tests were run with the DDS subscores against the demographic variables. These tests resulted in a statistically significant difference in the DDS subscores based on the participant's work environment, $p = .006$. Specifically, a participant's work environment has a statistically significant effect on the Physical Distress subscore, $p = .009$, and the Regimen Distress subscore, $p = .041$. Neglecting self care and diabetes status also had statistically significant effects on DDS subscore, as expected.

Data Modeling

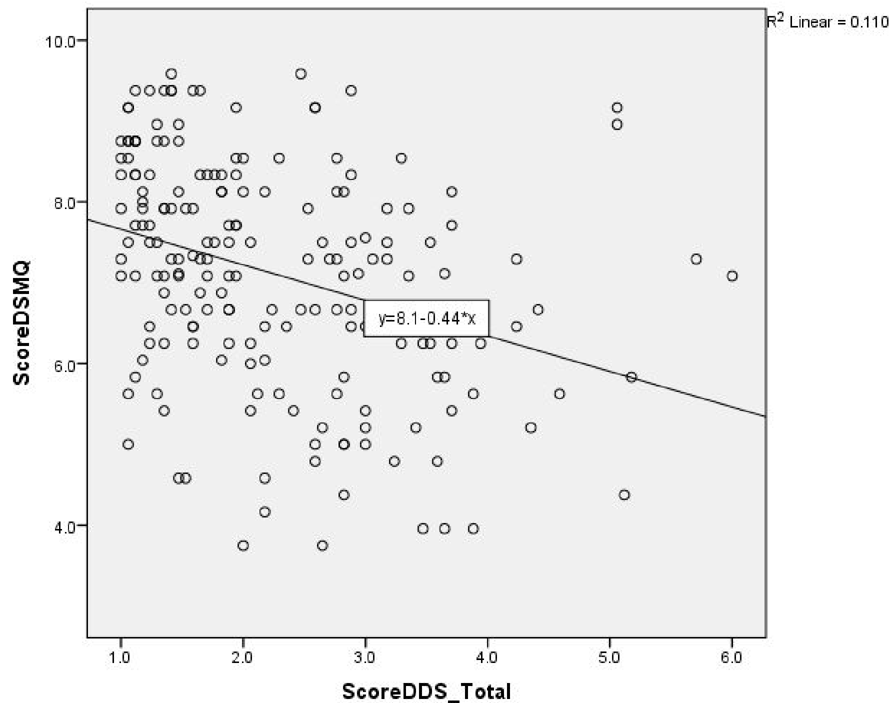
The first regression, as seen in the first table (Figure 1.1) below and the corresponding graph (Figure 1.2) to follow, was with Diabetes Self Management Score (total DSMQ) as the dependent variable with the Diabetes Distress Scale Score (total DDS) as the independent variable. The results showed a -0.44 coefficient and a significance level of 0.000. This means that there is a significant negative relationship between total DSMQ and total DDS. We see a similar output in the second linear regression model, as shown in the second table (Figure 2.1) below and the corresponding graph (Figure 2.2) to follow. In this model the total DSMQ was the dependent variable and the Diabetes Time Management Score (total DTMQ) was the independent variable. The results of this model showed a -1.022 coefficient and a significance level of 0.000. This indicates that there is a significant negative relationship between total DSMQ and total DTMQ. In the last set of models, we see the regression output the table (Figure 3.1) for Diabetes Distress Scale (total DDS) as the dependent variable and Diabetes Time Management (total DTMQ) as the independent variable below, along with the corresponding graph (Figure 3.2) to follow. The results from this regression showed a -0.500 coefficient and a significance level of 0.000. This means that there is a significant positive relationship between total DDS and total DTMQ.

Linear Regression Table - Figure 1.1

DSMQ vs DDS

Coefficient	Unstandardized B	Coefficients Std. Error	t-value	Significance
Constant	8.1	0.228	35.514	0.000
Diabetes Distress Score	-0.44	0.092	-4.783	0.000

Linear Regression Graph - Figure 1.2



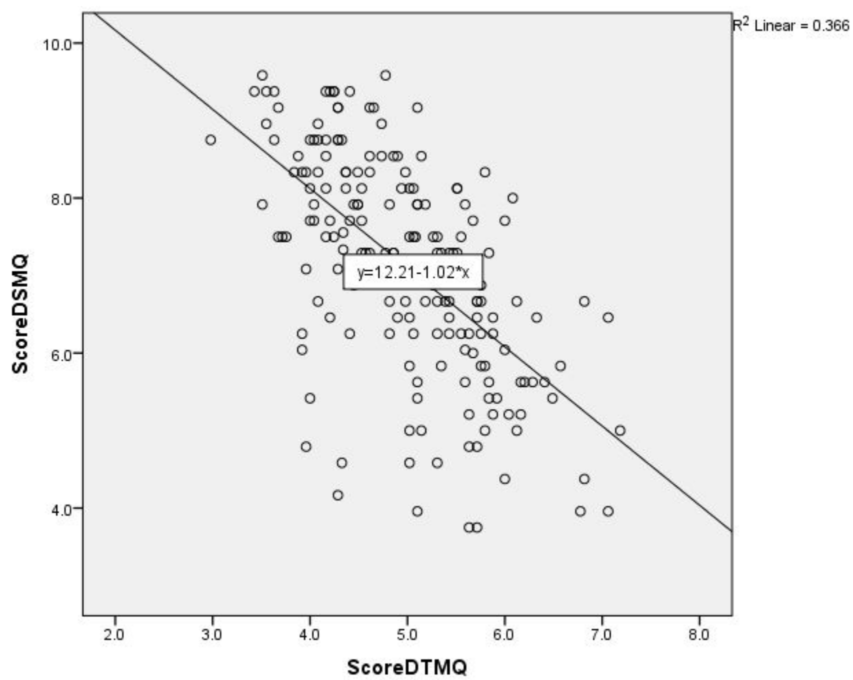
[redacted]

Linear Regression Table - Figure 2.1

DSMQ vs DTMQ

Coefficient	Unstandardized B	Coefficients Std. Error	t-value	Significance
Constant	12.209	.498	24.505	.000
Diabetes Time Management Score	-1.022	.099	-10.370	.000

Linear Regression Graph - Figure 2.2



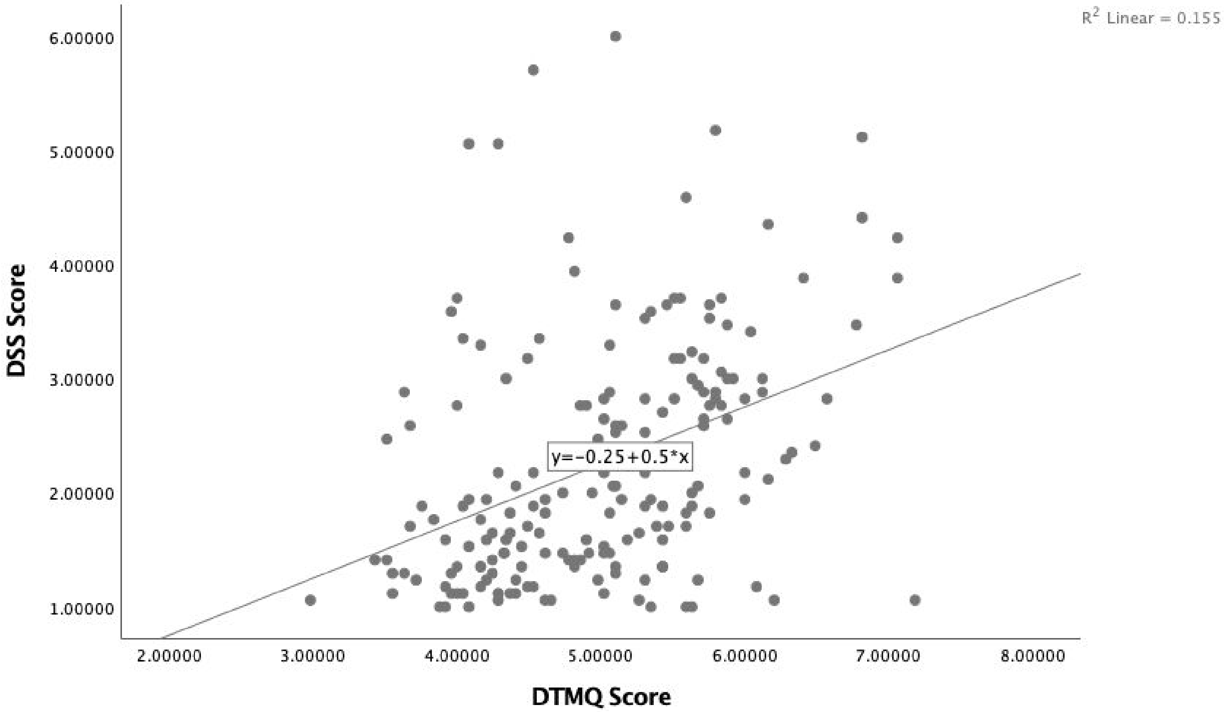
[redacted]

Linear Regression Table - Figure 3.1

DDS vs DTMQ

Coefficient	Unstandardized B	Coefficients Std. Error	t-value	Significance
Constant	-.248	.433	-.573	.568
Diabetes Time Management Score	.500	.086	5.839	.000

Linear Regression Graph - Figure 3.2



Following the simple regression models, multiple linear regression models were produced. These models include the significant demographic data which was discovered in the ANOVA and MANOVA tests during the initial data exploration. These variables include work environment, employment status, and last HbA1C.

The first multiple linear regression models that were ran tested DSMQ total score on the DDS total score and significant demographic data. DSMQ total score was used as the dependent variable with the DDS total score and Employment Status as the independent variables, including an interaction term. This resulted as a statistically significant model, $p < .05$. The model estimate was:

$$DSMQ = 8.507 - .525DDS + .413FullTimeEmployment*DDS - 1.717FullTimeEmployment$$

This indicates that when employed full time, the DSMQ score is expected to decrease .112 points with every unit increase in the DDS score. Also, when not employed full time, the DSMQ score is expected to decrease .525 points with every unit increase in the DDS score.

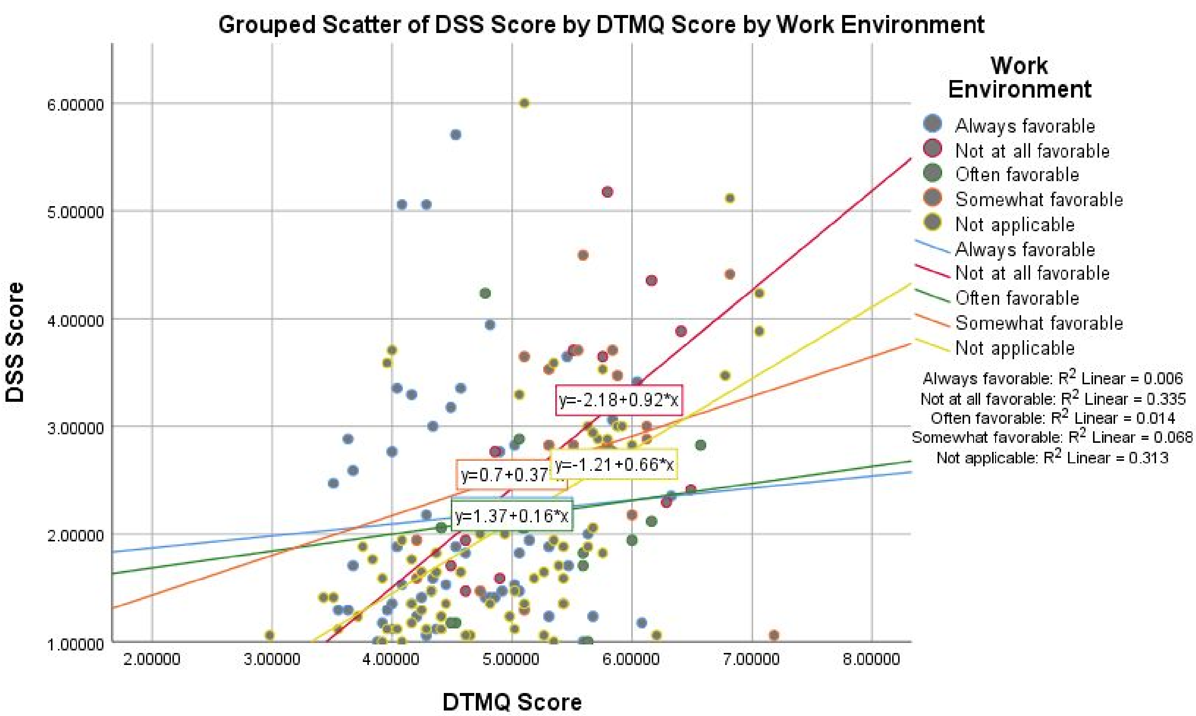
Another model which was significant, $p < .05$, included DSMQ as the response variable, A1C and DDS as the independent variable, and their interaction terms. This model was as follows:

$$DSMQ = 6.711 + .056DDS - .622HbA1C<6.5*DDS + 1.718HbA1C<6.5$$

This means that when a participant's last HbA1C was less than 6, their DSMQ score is expected to decrease .566 points with every unit increase in their DDS score. If a participant's last HbA1C was above 6, however, the DSMQ score is expected to increase .056 points with every unit increase in the DDS score. It is unusual to see self care increasing with higher levels of distress.

Models were also run testing the DSMQ total score on DTMQ total score and significant demographics. However, none of these resulted in a statistically significant model, $p > .05$.

The DDS ANOVA table, seen in the data exploration section, indicated that the variable work environment significantly affects the DDS score. Due to this, and the correlation between DDS and DTMQ, a model was run to examine DDS as the dependent variable along with DTMQ, work environment, and an interaction term as the independent variables. The effect of the predictor on the response is significant as indicated by $p < .05$, for work environment and the DTMQ score, but it was not significant, $p > .05$, for the interaction term. However, a graph of this data produced interesting results.



The red line on this graph signifies the linear regression between the DTMQ score and DDS score for participants who work in a not at all favorable environment. This line is noticeably steeper than the other work environments' regressions. This indicates that as the time management score increases, the distress score increases at a greater rate of change than other types of employment statuses.

A final model was run which included the two significantly correlated subscores, mentioned in the data exploration section, and significant demographics. These models include Dietary Control (DSMQ subscore) as the dependent variable, with the Regimen Distress (DDS Subscore), demographic variables, and interaction terms as the independent variables. The significant model with these parameters included the demographic variable of A1C. The model is as follows:

$$DietaryControl=5.672-.144RegimenDistress-.663HbA1C<6*RegimenDistress+2.018HbA1C<6$$

This model indicates that when a participant's HbA1C was less than 6, their dietary control score is expected to decrease by .807 with every unit increase in the regimen distress score. The dietary control score is also expected to decrease by .144 for every unit increase in the regimen distress score for a participant whose last A1C was above 6.

Conclusion

The overarching research question for this study was what are the relationships between and among diabetes self care, diabetes self management, and diabetes distress in women with T2DM. Initial exploration concluded that these three variables were indeed significantly correlated with each other. Furthermore, the greater distress and weaker time management skills were predicted to lead to weaker self management skills. Also, greater distress was predicted to result in weaker time management skills, and vice versa. These relationships were expected and confirmed in the preliminary analyses.

In order to further the analysis of the effect of time management and distress on self care, these relationships were tested in specific groups of people. The most significant groups were categorized by the work environment, employment status, and the last HbA1C of the participant. It was concluded that a participant's self management score will decrease more drastically when she is not employed full time, compared to when she is employed full time. Also, distress is expected to increase at a greater rate with less time management when a participant works in a not at all favorable environment compared to all other types of environments.

This analysis also included some thought-provoking findings. First, it was found that HbA1C was not significant against any main variables, which is not what was expected. HbA1C is an important part of diabetes in general, as well as self-care. HbA1C was not significantly dependent on diabetes self-care, time management, or distress as we thought it would. Another interesting finding was the children caregiver role. Usually children add more to a caregiver's schedule. It was expected that the children caregiver role would have resulted in higher stress or less time management skills, however this was not the case. There was no significant differences between a women who was or was not providing care for a child. We came to a theory that maybe this was because the women in our study were mostly older and their children would be older and would have less of an impact on these main variables towards women.

References

Center for Disease and Control Prevention (2018). All About Your A1C. Retrieved April 23, 2019, from https://www.cdc.gov/diabetes/managing/managing-blood-sugar/a1c.html

Center for Disease and Control Prevention (2018). Diabetes and Women. Retrieved from https://www.cdc.gov/diabetes/library/features/diabetes-and-women.html.

Centers for Disease Control and Prevention (2014). 2014 National diabetes statistics report. Retrieved from http://www.cdc.gov/diabetes/pubs/statsreort14.htm.

Fisher, L., Mullan, T., Skaff, M., Glasgow, R., Arean, P., & Hessler, D. (2009). Predicting distress in patients with type 2 diabetes. Diabetic Medicine, 26(6), 622-627. doi:10.1111/j.1464-5491.2009.02730.x

Funnell, Martha M et al. "National standards for diabetes self-management education." *Diabetes care vol.* 33 Suppl 1,Suppl 1 (2010): S89-96. doi:10.2337/dc10-S089.

Gafarian, Celeste T., et al. "The Diabetes Time Management Questionnaire." *The Diabetes Educator*, vol. 25, no. 4, 1999, pp. 585–592., doi:10.1177/014572179902500411.Schmitt, Andreas, et al. "Assessing Diabetes Self-Management with the Diabetes Self-Management Questionnaire (DSMQ) Can Help Analyse Behavioural Problems Related to Reduced Glycaemic Control." *Plos One*, vol. 11, no. 3, 2016, doi:10.1371/journal.pone.0150774.

Orem, D. E. (1995). Nursing concepts of practice (5th ed.). St. Louis, MO: Mosby-Year Book, Inc.

Polonsky, W. H., et al. “Assessing Psychosocial Distress in Diabetes: Development of the Diabetes Distress Scale.” *Diabetes Care*, vol. 28, no. 3, 2005, pp. 626–631., doi:10.2337/diacare.28.3.626.

Appendix

R Code:

readData.R

```
# Load data ---------------------------------------------------------------

# setwd("?")

paper <- read.csv("dataset/Paper Format Data Collection Surveys.csv",
                  stringsAsFactors = F, na.strings=c("","NA"))

ele <- read.csv("dataset/Electronic Format Data Collection Surveys.csv",
                stringsAsFactors = F, na.strings=c("","NA"))

# remove the first row
paper <- paper[-1, ]
ele <- ele[-1, ]

# Add ID
paper$ID <- paste0("p", seq(1,nrow(paper)))
ele$ID <- paste0("e",seq(1,nrow(ele)))
```

```
# Rmove ele first 17 columns
ele <- ele[, 18:ncol(ele)]

# Fix paper's col names
colnames(paper) <- c(paste0("Q", seq(1,106)), "ID")

# Check NAs -------------------------------------------------------------

# count NA
temp_paperNA <- tempFun_calculateQ1Q1OtherQStat(paper,tempFun_countNA)
temp_eleNA <- tempFun_calculateQ1Q1OtherQStat(ele,tempFun_countNA)

# count TRUE
temp_paperQ1Q7CountTure <- tempFun_calculateQ1Q1OtherQStat(paper,tempFun_countTrue)
temp_eleQ1Q7CountTure <- tempFun_calculateQ1Q1OtherQStat(ele,tempFun_countTrue)

# add 3 columns count NA, T for Q1~Q7 and count other Q Q
paper <- paper %>%
  mutate(
    temp_NAQ1Q7 = temp_paperNA$Q1Q7,
    temp_NAOtherQ = temp_paperNA$OtherQ,
```

```
    temp_trueQ1Q7 = temp_paperQ1Q7CountTure$Q1Q7
  )
ele <- ele %>%
  mutate(
    temp_NAQ1Q7 = temp_eleNA$Q1Q7,
    temp_NAOtherQ = temp_eleNA$OtherQ,
    temp_trueQ1Q7 = temp_eleQ1Q7CountTure$Q1Q7
  )

# add invalidFlag: questionnaire is not answered at all
paper <- paper %>%
  mutate(temp_invalidFlag = (temp_NAQ1Q7 > 0) & (temp_NAOtherQ == 97))
if (sum(paper$temp_NAQ1Q7>0) > sum(paper$temp_invalidFlag)){
  print("check your paper invalidFlag")
}
ele <- ele %>%
  mutate(temp_invalidFlag = (temp_NAQ1Q7 > 0) & (temp_NAOtherQ == 97))
if (sum(ele$temp_NAQ1Q7>0) > sum(ele$temp_invalidFlag)){
  print("check your ele invalidFlag")
}

# add ineligibleFlag: have at least one flase in q1~q7
```

```
temp_fun <- function(y,z){
  x <- y
  for (i in 1:length(y)){
    if (y[i] | is.na(y[i])){
      x[i] <- NA
    } else{
      x[i] <- z[i]
    }
  }
  x
}
paper <- paper %>%
  mutate(temp_ineligibleFlag = temp_fun(temp_invalidFlag, temp_trueQ1Q7 < 7))
ele <- ele %>%
  mutate(temp_ineligibleFlag = temp_fun(temp_invalidFlag, temp_trueQ1Q7 < 7))

# add abandonFlag: have more than one NA in otherQ for all eligible people
data_paper <- paper %>%
  mutate(temp_abandonFlag = temp_fun(temp_ineligibleFlag, temp_NAOtherQ > 0))

# found p56 was actually an abandonment, but was originally coded with 99
data_paper[data_paper$ID == "p56", "temp_abandonFlag"] <- T
```

```
data_ele <- ele %>%
  mutate(temp_abandonFlag = temp_fun(temp_ineligibleFlag, temp_NAOtherQ > 0))

if (option_PrintFlag){
  cat("paper count\n\n")
  print(apply(data_paper[,111:113], 2, function(x){table(x, useNA = "ifany")}))
  cat("electronic count\n\n")
  print(apply(data_ele[,111:113], 2, function(x){table(x, useNA = "ifany")}))
  #View(ele[ele$temp_ineligibleFlag,105:113])
}

if (option_HtmlFlag){
  kable(data_paper[,107:113]) %>%
    kable_styling()
}

if (option_HtmlFlag){
  kable(data_ele[,107:113]) %>%
    kable_styling()
}

if (option_UpdateDatasetFlag){
```

```
  write.csv(data_paper, "dataset/data_paper.csv", row.names = F)

  write.csv(data_ele, "dataset/data_ele.csv", row.names = F)

}
```

transformData.R

```
# paper <- read.csv("paper_del.csv")

# data_paperAndEle_fixed <- read.csv("data_paperAndEle_fixed_del.csv")

data_paper <- read.csv("dataset/data_paper.csv")

data_ele <- read.csv("dataset/data_ele.csv")


data_paperAndEle <- rbind(mutate_all(data_paper, as.character),mutate_all(data_ele,
as.character))[, 8:113]

data_paperAndEle <- mutate_at(data_paperAndEle, .funs = function(x){ifelse(is.na(x),"NAAA",
x)},

                    .vars = seq(1,100))

data_paperAndEle_fixed <- mutate_at(data_paperAndEle, tempFun_delSpace,

                     .vars = seq(1,100))

data_paperAndEle_codes <- data_paperAndEle_fixed


# Q8-Q23 ------------------------------------------------------------------

temp_z <- unique(unlist(data_paperAndEle_fixed[,1:16]))
```

```
temp_Q08Q23Names <- c(
  "Blood sugar measurement is not required as a part of my self-care",
  "Does not apply to me",
  "Applies to me to some degree",
  "Applies to me to a considerable degree",
  "Applies to me very much",
  "???<NA>99",
  NA)
temp_Q08Q23Codes <- c(
  "Not required",
  0,
  1,
  2,
  3
)

temp_Q08Q23KeywordsList <- list(
  "blood",
  "does not",
  "some degree",
  "considerable",
```

```
  "very much",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q08Q23KeywordsList)

Q08Q23_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q08Q23Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, seq(1,16),
Q08Q23_fixMapping)
temp_z <- unique(unlist(data_paperAndEle_fixed[,1:16]))

Q08Q23_codesMapping <-
  data.frame(from = temp_Q08Q23Names[1:length(temp_Q08Q23Codes)],
             to = temp_Q08Q23Codes)

temp_funFT <- function(x,y){
  for (i in 1:length(x)){
```

```
    if (x[i] %in% y$from){
      x[i] <- as.character(y$to[y$from == x[i]])
    }
  }
  x
}

data_paperAndEle_codes[,1:16] <- mutate_all(data_paperAndEle_fixed[,1:16], .funs =
                          function(x){temp_funFT(x, Q08Q23_codesMapping)})

if (option_PrintFlag){
  print(Q08Q23_fixMapping)
  print(temp_z)
  print(Q08Q23_codesMapping)
}

# Q24~Q72 ------------------------------------------------------------------

temp_z <- unique(unlist(data_paperAndEle_fixed[,17:65]))

temp_Q24Q72Names <- c(
  "Always",
```

```
  "Often",
  "Sometimes",
  "Rarely",
  "Never",
  "Blood sugar measurement is not required as a part of my self-care",
  "???<NA>99",
  NA)
temp_Q24Q72Codes <- c(
  1,
  2,
  3,
  4,
  5,
  "Not required"
)

temp_Q24Q72KeywordsList <- list(
  "always",
  "often",
  "sometimes",
  "rarely",
  "never",
```

```
  c("does", "not", "blood"),
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q24Q72KeywordsList)

Q24Q72_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q24Q72Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, seq(17,65),
Q24Q72_fixMapping)
temp_z <- unique(unlist(data_paperAndEle_fixed[,17:65]))

Q24Q72_codesMapping <-
  data.frame(from = temp_Q24Q72Names[1:length(temp_Q24Q72Codes)],
             to = temp_Q24Q72Codes)



data_paperAndEle_codes[,17:65] <- mutate_all(data_paperAndEle_fixed[,17:65], .funs =
```

```
        function(x){temp_funFT(x, Q24Q72_codesMapping)})


if (option_PrintFlag){
  print(Q24Q72_fixMapping)
  print(temp_z)
  print(Q24Q72_codesMapping)
}


# Q73~Q89 ----------------------------------------------------------------
temp_z <- unique(unlist( data_paperAndEle_fixed[,66:82]))


temp_Q73Q89Names <- c(
  "Not a problem",
  "A Slight problem",
  "A moderate problem",
  "Somewhat serious problem",
  "A serious problem",
  "A very serious problem",
  "???<NA>99",
  NA
)
```

```
temp_Q73Q89Codes <- c(
  1,
  2,
  3,
  4,
  5,
  6
)

temp_Q73Q89KeywordsList <- list(
  "not",
  "slight",
  "moderate",
  c("somewhat serious", "somewhat a serious problem", "somewhat series"),
  c("a serious problem", "a serious probelm"),
  c("very serious", "very"),
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q73Q89KeywordsList)
```

```
Q73Q89_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q73Q89Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, seq(66,82),
Q73Q89_fixMapping)
temp_z <- unique(unlist(data_paperAndEle_fixed[,66:82]))

Q73Q89_codesMapping <-
  data.frame(from = temp_Q73Q89Names[1:length(temp_Q73Q89Codes)],
             to = temp_Q73Q89Codes)

data_paperAndEle_codes[,66:82] <- mutate_all(data_paperAndEle_fixed[,66:82], .funs =
                              function(x){temp_funFT(x, Q73Q89_codesMapping)})

if (option_PrintFlag){
  print(Q73Q89_fixMapping)
  print(temp_z)
  print(Q73Q89_codesMapping)
}
```

```
# Q90 ---------------------------------------------------

unique(data_paperAndEle_fixed[,83])

temp_z <- data_paperAndEle_fixed[,83]


for (i in 1:length(temp_z)){

  if (temp_z[i] == "99 (not answered)"){

    temp_z[i] <- "???<NA>99"

  } else if (temp_z[i] == "naaa"){

    temp_z[i] <- NA

  }

}

data_paperAndEle_fixed[,83] <- temp_z

data_paperAndEle_codes[,83] <- temp_z


if (option_PrintFlag){

  unique(temp_z)

}


# Q91 ---------------------------


temp_z <- unique(unlist(data_paperAndEle_fixed[,84]))

temp_Q91Names <- c(
```

```
  "I am not currently employed",
  "I am employed part time",
  "I am employed full time",
  "I am retired",
  "I prefer not to answer",
  "???<NA>99",
  NA
)

temp_Q91Codes <- c(
  1,
  2,
  3,
  4,
  0
)

temp_Q91KeywordsList <- list(
  "i am not currently employed",
  "i am employed part time",
  "i am employed full time",
  "i am retired",
```

```
  "i prefer not to answer",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q91KeywordsList)

Q91_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q91Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 84,
Q91_fixMapping)
temp_z <- unique(unlist(data_paperAndEle_fixed[,84]))

Q91_codesMapping <-
  data.frame(from = temp_Q91Names[1:length(temp_Q91Codes)],
             to = temp_Q91Codes)

data_paperAndEle_codes[,84] <-  temp_funFT(data_paperAndEle_fixed[,84],
Q91_codesMapping)
```

```
if (option_PrintFlag){
  print(Q91_fixMapping)
  print(temp_z)
  print(Q91_codesMapping)
}

# Q92 ---------------------------------------------------------------------
temp_z <- unique(unlist(data_paperAndEle_fixed[,85]))

temp_Q92Names <- c(
  "Not at all favorable",
  "Somewhat favorable",
  "Often favorable",
  "Always favorable",
  "Does not apply to me",
  "???<NA>99",
  NA
)

temp_Q92Codes <- c(
  1,
```

```
  2,
  3,
  4,
  0
)

temp_Q92KeywordsList <- list(
  "not at all",
  "somewhat",
  "often",
  "always",
  c("does not apply to me", "not applicable to me"),
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q92KeywordsList)

Q92_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q92Names,x)}))
```

```
data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 85,
Q92_fixMapping)
temp_z <- unique(unlist(data_paperAndEle_fixed[,85]))

Q92_codesMapping <-
  data.frame(from = temp_Q92Names[1:length(temp_Q92Codes)],
             to = temp_Q92Codes)

data_paperAndEle_codes[,85] <-  temp_funFT(data_paperAndEle_fixed[,85],
Q92_codesMapping)

if (option_PrintFlag){
  print(Q92_fixMapping)
  print(temp_z)
  print(Q92_codesMapping)
}

# Q93 -------------------------------------------------------------------
temp_z <- unlist(data_paperAndEle_fixed[,86])
for (i in 1:length(temp_z)){
  if (!is.na(temp_z[i])){
```

```
    if (temp_z[i] == "naaa"){
      temp_z[i] <- NA
    }
  }
}
temp_fun <- function(x){
  output <- vector(length = length(temp_z))
  for (i in 1:length(temp_z)){
    if (is.na(temp_z[i])){
      output[i] <- NA
    } else if (str_detect(temp_z[i],"99")){
      output[i] <- "???<NA>99"
    } else if (str_detect(temp_z[i],x)){
      output[i] <- 1
    } else {
      output[i]< - 0
    }
  }
  output
}

data_paperAndEle_fixed[,86] <- temp_z
```

```
data_paperAndEle_codes[,86] <- temp_z


data_paperAndEle_codes$`Caregiver Role (self)` <- temp_fun("self")

data_paperAndEle_codes$`Caregiver Role (children)` <- temp_fun("children")

data_paperAndEle_codes$`Caregiver Role (partner)` <- temp_fun("partner")

data_paperAndEle_codes$`Caregiver Role (parent)` <- temp_fun("parent")


# Q94 -------------------------------------------------------------------

temp_z <- unique(unlist(data_paperAndEle_fixed[,87]))


temp_Q94Names <- c(
  "rarely",
  "about half the time",
  "most of the time",
  "Does not apply to me",
  "???<NA>99",
  NA
)


temp_Q94Codes <- c(
  3,
  2,
```

```
  1,
  0
)

temp_Q94KeywordsList <- list(
  c("not at all", "rarely"),
  c("sometimes", "often", "about half the time"),
  c("always", "most of the time"),
  c("does not apply to me", "not apply"),
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q94KeywordsList)

Q94_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q94Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 87,
Q94_fixMapping)
```

```
Q94_codesMapping <-
  data.frame(from = temp_Q94Names[1:length(temp_Q94Codes)],
             to = temp_Q94Codes)

data_paperAndEle_codes[,87] <-  temp_funFT(data_paperAndEle_fixed[,87],
Q94_codesMapping)

if (option_PrintFlag){
  print(Q94_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,87])))
  print(Q94_codesMapping)
}

# Q95 ------------------------------------------------------------------

temp_z <- data_paperAndEle_fixed[,88]
for (i in 1:length(temp_z)){
  if (temp_z[i] == "15 years"){
    temp_z[i] <- 15
  } else if (temp_z[i] == "naaa"){
    temp_z[i] <- NA
```

```
  }
}
data_paperAndEle_fixed[,88] <- temp_z
data_paperAndEle_codes[,88] <- temp_z

if (option_PrintFlag){
  print(temp_z)
}

# Q96 ---------------------------------------------------------------

temp_z <- unique(unlist(data_paperAndEle_fixed[,89]))

temp_Q96Names <- c(
  "I do not have any other health conditions",
  "I have 1 other health condition",
  "I have 2 other health conditions",
  "I have 3 or more other health conditions",
  "I am not sure if i have any other health conditions",
  "???<NA>99",
  NA
)
```

```
temp_Q96Codes <- c(
  1,
  3,
  4,
  5,
  2
)

temp_Q96KeywordsList <- list(
  "do not have",
  "1",
  "2",
  "3",
  "i am not sure",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q96KeywordsList)

Q96_fixMapping <-
```

```
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q96Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 89,
Q96_fixMapping)

Q96_codesMapping <-
  data.frame(from = temp_Q96Names[1:length(temp_Q96Codes)],
             to = temp_Q96Codes)

data_paperAndEle_codes[,89] <-  temp_funFT(data_paperAndEle_fixed[,89],
Q96_codesMapping)

if (option_PrintFlag){
  print(Q96_codesMapping)
  print(Q96_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,89])))
}

# Q97 -------------------------------------------------------------------
temp_z <- unique(unlist(data_paperAndEle_fixed[,90]))
```

```
temp_Q97Names <- c(
  "less than 6.0",
  "6.1 to 6.5",
  "6.6 to 7.0",
  "7.1 to 7.5",
  "7.6 to 8.0",
  "greater than 8.0",
  "Not sure",
  "???<NA>99",
  NA
)

temp_Q97Codes <- c(
  1,
  2,
  3,
  4,
  5,
  6,
  0
)
```

```
temp_Q97KeywordsList <- list(
  "less than 6.0",
  "6.1 to 6.5",
  "6.6 to 7.0",
  "7.1 to 7.5",
  "7.6 to 8.0",
  "greater than 8.0",
  "i don't know my a1c",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q97KeywordsList)

Q97_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q97Names,x)}))
```

```
data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 90,
Q97_fixMapping)

Q97_codesMapping <-
  data.frame(from = temp_Q97Names[1:length(temp_Q97Codes)],
             to = temp_Q97Codes)

data_paperAndEle_codes[,90] <-  temp_funFT(data_paperAndEle_fixed[,90],
Q97_codesMapping)

if (option_PrintFlag){
  print(Q97_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,90])))
  print(Q97_codesMapping)
}

# Q98 ------------------------------------------------------------------
temp_z <- unique(unlist(data_paperAndEle_fixed[,91]))

temp_Q98Names <- c(
  "Oral or non-insulin medications only",
  "Oral medications and insulin",
```

```
  "Insulin pens/injections only",
  "Insulin pump only",
  "???<NA>99",
  NA
)

temp_Q98Codes <- c(
  1,
  2,
  3,
  4
)

temp_Q98KeywordsList <- list(
  c("oral or non-insulin medications","oral medications only"),
  "oral medications and insulin",
  c("insulin pens/injections only", "insulin pens or injections only"),
  "insulin pump only",
  "99",
  "naaa"
)
```

```
temp_zTran <- tempFun_createWordList(temp_z, temp_Q98KeywordsList)

Q98_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q98Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 91,
Q98_fixMapping)

Q98_codesMapping <-
  data.frame(from = temp_Q98Names[1:length(temp_Q98Codes)],
             to = temp_Q98Codes)



data_paperAndEle_codes[,91] <-  temp_funFT(data_paperAndEle_fixed[,91],
Q98_codesMapping)

if (option_PrintFlag){
  print(Q98_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,91])))
  print(Q98_codesMapping)
```

```
}

# Q99 ------------------------------------------------------------------
temp_z <- unique(unlist(data_paperAndEle_fixed[,92]))

temp_Q99Names <- c(
  "my diabetes is getting worse",
  "my diabetes is staying the same",
  "my diabetes is getting better",
  "???<NA>99",
  NA
)

temp_Q99Codes <- c(
  1,
  2,
  3
)

temp_Q99KeywordsList <- list(
  "worse",
  "same",
```

```
  c("better", "beter"),
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q99KeywordsList)

Q99_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q99Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 92,
Q99_fixMapping)




Q99_codesMapping <-
  data.frame(from = temp_Q99Names[1:length(temp_Q99Codes)],
             to = temp_Q99Codes)
```

```
data_paperAndEle_codes[,92] <-  temp_funFT(data_paperAndEle_fixed[,92],
Q99_codesMapping)

if (option_PrintFlag){
  print(Q99_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,92])))
  print(Q99_codesMapping)
}

# Q99 ------------------------------------------------------------------
temp_z <- unique(unlist(data_paperAndEle_fixed[,93]))

temp_Q100Names <- c(
  "less than $30,000" ,
  "$30,001 to $50,000",
  "$50,001 to $70,000",
  "$70,001 to $100,000",
  "greater than $100,001",
  "I prefer not to answer",
  "???<NA>99",
  NA
)
```

```
temp_Q100Codes <- c(
  1,
  2,
  3,
  4,
  5,
  0
)

temp_Q100KeywordsList <- list(
  "less than $30,000" ,
  c("$30,001 to $50,000", "$30,001 - $50,000"),
  c("$50,001 to $70,000", "$50,001 - $70,000"),
  c("$70,001 to $100,000", "$70,001 - $100,00"),
  "greater than $100,001",
  "i prefer not to answer",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q100KeywordsList)
```

```
Q100_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q100Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 93,
Q100_fixMapping)

Q100_codesMapping <-
  data.frame(from = temp_Q100Names[1:length(temp_Q100Codes)],
             to = temp_Q100Codes)

data_paperAndEle_codes[,93] <-  temp_funFT(data_paperAndEle_fixed[,93],
Q100_codesMapping)

if (option_PrintFlag){
  print(Q100_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,93])))
  print(Q100_codesMapping)
}
```

```
# Q101 ----------------------------------------------------------------

temp_z <- unique(unlist(data_paperAndEle_fixed[,94]))


temp_Q101Names <- c(
  "Less than high school",
  "High school graduate (or equivalent)",
  "Some college (no degree)",
  "Associate's degree",
  "Bachelor's degree",
  "Graduate or professional degree",
  "Doctorate (phd)",
  "???<NA>99",
  NA
)


temp_Q101Codes <- c(
  1,
  2,
  3,
  4,
  5,
  6,
```

```
  7
)

temp_Q101KeywordsList <- list(
  "less than high school",
  "high school graduate",
  c("some college (no degree)", "some college, no degree"),
  "associate's degree",
  "bachelor's degree",
  "professional degree",
  "phd",
  "99",
  "naaa"
)

temp_zTran <- tempFun_createWordList(temp_z, temp_Q101KeywordsList)

Q101_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q101Names,x)}))
```

```
data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 94,
Q101_fixMapping)

Q101_codesMapping <-
  data.frame(from = temp_Q101Names[1:length(temp_Q101Codes)],
             to = temp_Q101Codes)

data_paperAndEle_codes[,94] <-  temp_funFT(data_paperAndEle_fixed[,94],
Q101_codesMapping)

if (option_PrintFlag){
  print(Q101_fixMapping)
  print(unique(unlist(data_paperAndEle_fixed[,94])))
  print(Q101_codesMapping)
}

# Q102 -------------------------------------------------------------------
temp <- data_paperAndEle_fixed[,95]
data_paperAndEle_fixed[str_detect(temp,","),95] <- ","

temp_z <- unique(unlist(data_paperAndEle_fixed[,95]))
temp_Q102Names <- c(
```

```
  "Mixed",
  "Indian",
  "Native",
  "African",
  "Latino",
  "Asian",
  "White",
  "Other",
  "???<NA>99",
  NA
  )
temp_Q102Codes <- c(
  "Mixed",
  "Indian",
  "Native",
  "African",
  "Latino",
  "Asian",
  "White",
  "Other"
)
temp_Q102KeywordsList <- list(
```

```
  ",",
  "indian",
  "native",
  "african",
  "latino",
  "asian",
  "white",
  "other",
  "99",
  "naaa")


temp_zTran <- tempFun_createWordList(temp_z, temp_Q102KeywordsList)


Q102_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q102Names,x)}))


data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 95,
Q102_fixMapping)
```

```
Q102_codesMapping <-
  data.frame(from = temp_Q102Names[1:length(temp_Q102Codes)],
             to = temp_Q102Codes)


data_paperAndEle_codes[,95] <-  temp_funFT(data_paperAndEle_fixed[,95],
Q102_codesMapping)


if (option_PrintFlag){
  print(Q102_fixMapping)
  print(unique(unlist(data_paperAndEle_codes[,95]))) # Here I used the code
  print(Q102_codesMapping)
}


# Q103 ----------------------------------------------------------


temp_z <- unlist(data_paperAndEle_fixed[,96])
for (i in 1:length(temp_z)){
  if (str_detect(temp_z[i], "rural")){
    temp_z[i] <- "rural"
  } else if (str_detect(temp_z[i], "urban")) {
    temp_z[i] <- "urban"
  } else if (str_detect(temp_z[i], "naaa")) {
```

```
    temp_z[i] <- NA
  }
}
data_paperAndEle_fixed[,96] <- temp_z
data_paperAndEle_codes[,96] <- temp_z

if (option_PrintFlag){
  print(unique(temp_z))
}



# Q104 ------------------------------------------------------------------

temp_z <- unique(unlist(str_to_lower(data_paperAndEle[,97])))
temp_Q104Codes <- c("Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
"Connecticut", "Delaware",
            "Florida", "Georgia", "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas",
"Kentucky", "Louisiana",
            "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota", "Mississippi",
"Missouri", "Montana", "Nebraska",
```

```
"Nevada", "New Hampshire", "New Jersey", "New Mexico", "New York", "North
Carolina", "North Dakota", "Ohio", "Oklahoma",
"Oregon", "Pennslyvania", "Rhode Island", "South Carolina", "South Dakota",
"Tennessee", "Texas", "Utah", "Vermont",
"Virginia", "Washington", "West Virginia", "Wisconsin", "Wyoming", "refer","not
answered")
temp_Q104KeywordsList <- c(str_to_lower(temp_Q104Codes), "99", "naaa")
temp_Q104Names <- c(temp_Q104Codes[1:(length(temp_Q104KeywordsList)-2)], "Prefer not
to answer", "not answered (paper version)", "???<NA>99", NA)
temp_zTran <- tempFun_createWordList(temp_z, temp_Q104KeywordsList)
Q104_fixMapping <-
  data.frame(from = temp_z,
             to = sapply(temp_zTran[[1]],
                         function(x){tempFun_NANumber(temp_Q104Names,x)}))

data_paperAndEle_fixed <- tempFun_replaceWords(data_paperAndEle_fixed, 97,
Q104_fixMapping)


Q104_codesMapping <-
  data.frame(from = as.character(temp_Q104Names[1:length(temp_Q104Codes)]),
             to = as.character(temp_Q104Codes))
```

```
data_paperAndEle_codes[,97] <-  temp_funFT(data_paperAndEle_fixed[,97],
Q104_codesMapping)


if (option_PrintFlag){
  print(unique(unlist(data_paperAndEle_fixed[,97])))
  print(Q104_codesMapping)
}


# Colnames -----------------------------------------------------------


#rename all columns
colnames(data_paperAndEle_fixed)[1:16] = paste0("DSMQ", as.character(seq(1, 16))) #rename
as DSMQ
colnames(data_paperAndEle_fixed)[17:65] = paste0("DTMQ", as.character(seq(1, 49)))
#rename as DTMQ
colnames(data_paperAndEle_fixed)[66:82] = paste0("DDS", as.character(seq(1, 17))) #rename
as DDS
colnames(data_paperAndEle_fixed)[83] = "Age"
colnames(data_paperAndEle_fixed)[84] = "Employment Status"
colnames(data_paperAndEle_fixed)[85] = "Work Environment"
colnames(data_paperAndEle_fixed)[86] = "Caregiver Role"
colnames(data_paperAndEle_fixed)[87] = "Neglect Self-Care"
```

```
colnames(data_paperAndEle_fixed)[88] = "Years with Diabetes"

colnames(data_paperAndEle_fixed)[89] = "Other Health Conditions"

colnames(data_paperAndEle_fixed)[90] = "Last A1C"

colnames(data_paperAndEle_fixed)[91] = "Diabetes Medication"

colnames(data_paperAndEle_fixed)[92] = "Current Diabetes State"

colnames(data_paperAndEle_fixed)[93] = "Income"

colnames(data_paperAndEle_fixed)[94] = "Education"

colnames(data_paperAndEle_fixed)[95] = "Race/Ethnicity"

colnames(data_paperAndEle_fixed)[96] = "Setting Lived In"

colnames(data_paperAndEle_fixed)[97] = "State"

colnames(data_paperAndEle_fixed)[98] = "Survey Method"

colnames(data_paperAndEle_fixed)[99] = "Location of Paper Survey"

colnames(data_paperAndEle_fixed)[100] = "ID"


colnames(data_paperAndEle_codes)[1:100] = colnames(data_paperAndEle_fixed)[1:100]


data_paperAndEle_fixed <- data_paperAndEle_fixed[,c(seq(100,106),seq(1,99))]

data_paperAndEle_codes <- data_paperAndEle_codes[,c(seq(100,106),seq(1,99),seq(107,110))]


if (option_UpdateDatasetFlag){

  write.csv(data_paperAndEle_fixed,"dataset/data_paperAndEle_fixed.csv", row.names = F)

  write.csv(data_paperAndEle_codes,"dataset/data_paperAndEle_codes.csv", row.names = F)
```

```
}

# Rmove all the abandonFlag
temp_ <- as.logical(data_paperAndEle_fixed$temp_abandonFlag)
data_paperAndEle_fixed_deleted <- data_paperAndEle_fixed[!ifelse(is.na(temp_),T, temp_),
          c(1, seq(8,106))]
temp_ <- as.logical(data_paperAndEle_codes$temp_abandonFlag)
data_paperAndEle_codes_deleted <- data_paperAndEle_codes[!ifelse(is.na(temp_),T, temp_),
          c(1, seq(8,110))]

if (option_UpdateDatasetFlag){
  write.csv(data_paperAndEle_fixed_deleted, "dataset/data_paperAndEle_fixed_deleted.csv",
row.names = F)
  write.csv(data_paperAndEle_codes_deleted, "dataset/data_paperAndEle_codes_deleted.csv",
row.names = F)
}

# Treat "???<>" as NA
temp_fun <- function(x){
  for (i in 1:length(x)){
    suppressWarnings(temp<-as.numeric(x[i]))
    if(str_detect(x[i],"\\?\\?\\?<")){
```

```
      x[i] <- NA
    } else if (!is.na(temp)) {
      x[i] <- temp
    }
  }
  x
}
data_paperAndEleReversed <- tempFun_reverse(mutate_all(data_paperAndEle_codes_deleted,
temp_fun))

# Replace all 99 by NA
data_paperAndEleReversed <- data_paperAndEleReversed %>%
  mutate_at(function(x){ifelse(str_detect(as.character(x), "99"), NA, as.character(x))},
            .vars = seq(2, ncol(data_paperAndEleReversed)))
# Check there are no "99" anymore and they are NA instead
if (option_PrintFlag){
  any(str_detect(unlist(data_paperAndEleReversed), "99"), na.rm = T)
}

if (option_UpdateDatasetFlag){
  write.csv(data_paperAndEleReversed, paste0("dataset/data_paperAndEleReversed_",
option_version,".csv"), row.names = F)
```

```
}


if (option_HtmlFlag){
  kable(data_paperAndEle_fixed) %>%
    kable_styling() %>%
    scroll_box(width = "100%", height = "1000px")
}


if (option_HtmlFlag){
  kable(data_paperAndEle_codes_deleted) %>%
    kable_styling() %>%
    scroll_box(width = "100%", height = "1000px")
}


if (option_HtmlFlag){
  kable(data_paperAndEleReversed) %>%
    kable_styling() %>%
    scroll_box(width = "100%", height = "1000px")
}



# Reverse ----------------------------------------------------------------
```

```
################################################################################
if (option_CleanEnvFlag){
  temp_spaceDatasetSaved <- "data_|tempFun_|option_"
  rm(list = ls()[!grepl(temp_spaceDatasetSaved, ls())])
}
```

imputeData.R

```
data_paperAndEleReversed <- read.csv(paste0("dataset/data_paperAndEleReversed_",
option_version, ".csv"))

data_paperAndEleReversed_imputed <- data_paperAndEleReversed

# KNN impute
set.seed(1)
data_paperAndEleReversed_imputed[,c(seq(2,86), seq(88,95), seq(101,104))]<-
  knn.impute(as.matrix(data_paperAndEleReversed[,c(seq(2,86),seq(88,95), seq(101,104))]), k =
10)
data_paperAndEleReversed_imputed <- as.data.frame(data_paperAndEleReversed_imputed)
```

```
for (i in 1:100){
  for (j in 1:nrow(data_paperAndEleReversed_imputed)){
    if (!is.na(data_paperAndEleReversed[j, i])){
      if (data_paperAndEleReversed[j, i] == "Not required"){
        data_paperAndEleReversed_imputed[j, i] <- NA
      }
    }
  }
}


temp_fun <- function(x){
  output <- x
  for (i in 1:length(x)){
    if (!(is.na(x[i]) | is.na(as.numeric(x[i])))){
      output[i] <- round(as.numeric(x[i]))
    }
  }
  output
}

s<-data_paperAndEleReversed_imputed[,96]
```

```
s[is.na(s)] <- "White"
data_paperAndEleReversed_imputed[,96]  <- s

if (option_UpdateDatasetFlag){
  write.csv(data_paperAndEleReversed_imputed,
            paste0("dataset/data_paperAndEleReversed_imputed_", option_version, ".csv"),
row.names = F)
}
```

addScore.R

```
fun_Calculate_Score <- function(x, col, method){
  output <- x
  temp <- x[,col]
  if (method == "DSMQ"){
    temp_funModel <- function(x){
      output <- vector(length = nrow(x))
      for (i in 1:nrow(x)){
        output[i] <- mean(as.numeric(x[i, ]),na.rm =T)/3*10
      }
      output
```

```
  }
  output$ScoreDSMQ <-
    temp_funModel(temp)
  output$ScoreDSMQ_Dietary_Control <-
    temp_funModel(temp[,c(2,5,9,13)])
  output$ScoreDSMQ_Glucose_Management <-
    temp_funModel(temp[,c(1,4,6,10,12)])
  output$ScoreDSMQ_Physical_Activity <-
    temp_funModel(temp[,c(8,11,15)])
  output$ScoreDSMQ_Physician_Contact <-
    temp_funModel(temp[,c(3,7,14)])
} else if (method == "DDS") {
  temp_funModel <- function(x){
    output <- vector(length = nrow(x))
    for (i in 1:nrow(x)){
      output[i] <- mean(as.numeric(x[i, ]),na.rm =T)
    }
    output
  }
  output$ScoreDDS_Total <-
    temp_funModel(temp)
  output$ScoreDDS_Emotional_Burden <-
```

```
    temp_funModel(temp[,c(2,4,7,10,14)])
   output$ScoreDDs_Physical_Distress <-
    temp_funModel(temp[,c(1,5,11,15)])
   output$ScoreDDS_Regimen_Distress <-
    temp_funModel(temp[,c(3,6,8,12,16)])
   output$ScoreDDS_Interpersonal_Distress <-
    temp_funModel(temp[,c(9,13,17)])
  } else if  (method == "DTMQ") {
   temp_funModel <- function(x){
    output <- vector(length = nrow(x))
    for (i in 1:nrow(x)){
     output[i] <- mean(as.numeric(x[i, ]),na.rm =T)/5*10
    }
    output
   }
   output$ScoreDTMQ <- temp_funModel(temp)
  }
  output
}

data_paperAndEleReversed_imputed
<-read.csv(paste0("dataset/data_paperAndEleReversed_imputed_", option_version, ".csv"))
```

```
data_paperAndEleReversedDSMQ_imputed <-
fun_Calculate_Score(data_paperAndEleReversed_imputed, seq(2,17), method = "DSMQ")
data_paperAndEleReversedDSMQ_imputed <-
fun_Calculate_Score(data_paperAndEleReversedDSMQ_imputed, seq(18,66), method =
"DTMQ")
data_paperAndEleReversedScoreAdded_imputed <-
fun_Calculate_Score(data_paperAndEleReversedDSMQ_imputed, seq(67,83), method =
"DDS")

if (option_UpdateDatasetFlag){
  write.csv(data_paperAndEleReversedScoreAdded_imputed,
paste0("dataset/data_paperAndEleReversedScoreAdded_imputed_", option_version, ".csv"),
row.names = F)
}

if (option_HtmlFlag){
  kable(data_paperAndEleReversedScoreAdded_imputed[,c(1,105:114)]) %>%
    kable_styling() %>%
    scroll_box(width = "100%", height = "1000px")
}
```

convertDemo.R

```
setwd("~/Documents/Teaching/MA490 Spring 2019/Diabetes Self-Management")
# source('r/r_cleanData/pre.R')

temp_data <- data_paperAndEleReversedScoreAdded_imputed
names(temp_data)



# Q91 - Employment Status ------------------------------------------------
temp <- Q91_codesMapping
codesMapping_employment <- data.frame(from = temp$to,
                                      to = c( "unemployed", "part time",
                                              "full time", "retired",
                                              "No answer"))

temp_df <- tempFun_replaceWords(temp_data, "Employment.Status",
                                codesMapping_employment)

temp_data <- temp_df

# Q92 - Work Environment ------------------------------------------------
```

```
temp <- Q92_codesMapping
codesMapping_environment <- data.frame(from = temp$to,
                                       to = c("Not at all favorable", "Somewhat favorable",
                                              "Often favorable",
                                              "Always favorable", "Not applicable"))

temp_df <- tempFun_replaceWords(temp_data, "Work.Environment",
                                codesMapping_environment)

temp_data <- temp_df



# Q94 - Neglect Self-care ------------------------------------------------
temp <- Q94_codesMapping
codesMapping_neglect <- data.frame(from = temp$to,
                                   to = c("rarely", "half the time",
                                          "most of the time", "Not applicable"))

temp_df <- tempFun_replaceWords(temp_data, "Neglect.Self.Care",
                                codesMapping_neglect)
```

```
temp_data <- temp_df



# Q96 - Other Health Conditions -----------------------------------------------
temp <- Q96_codesMapping
codesMapping_health <- data.frame(from = temp$to,
                                  to = c("None",
                                         "1",
                                         "2",
                                         "3 or more",
                                         "Not sure"))

temp_df <- tempFun_replaceWords(temp_data, "Other.Health.Conditions",
                                codesMapping_health)

temp_data <- temp_df



# Q97 - Last A1C -----------------------------------------------
temp <- Q97_codesMapping
codesMapping_A1C <- data.frame(from = temp$to,
                               to = c("< 6.0", "[6.1, 6.5]", "[6.6, 7.0]",
```

```
                    "[7.1, 7.5]", "[7.6, 8.0]", "> 8.0",
                    "don't know"))

temp_df <- tempFun_replaceWords(temp_data, "Last.A1C",
                   codesMapping_A1C)

temp_data <- temp_df

# Q98 - Diabetes Medication -----------------------------------------------
temp <- Q98_codesMapping
codesMapping_medication <- data.frame(from = temp$to,
                   to = c("Oral/non-insulin only",
                      "Oral med and insulin",
                      "Insulin pens/injections only",
                      "Insulin pump only"))

temp_df <- tempFun_replaceWords(temp_data, "Diabetes.Medication",
                   codesMapping_medication)

temp_data <- temp_df
```

```
# Q99 - Current Diabetes State -----------------------------------------------

temp <- Q99_codesMapping

codesMapping_dstate <- data.frame(from = temp$to,

                                  to = c("worse", "the same",

                                         "better"))


temp_df <- tempFun_replaceWords(temp_data, "Current.Diabetes.State",

                                codesMapping_dstate)


temp_data <- temp_df



# Q100 - Income -----------------------------------------------

temp <- Q100_codesMapping

codesMapping_income <- data.frame(from = temp$to,

                                  to = c("< $30,000", "[$30,001, $50,000]",

                                         "[$50,001, $70,000]", "[$70,001, $100,000]",

                                         "> $100,001", "no answer"))


temp_df <- tempFun_replaceWords(temp_data, "Income",

                                codesMapping_income)
```

```
temp_data <- temp_df



# Q101 - Education -----------------------------------------------
temp <- Q101_codesMapping
codesMapping_education <- data.frame(from = temp$to,
                                     to = c("below high school",
                                            "High school",
                                            "Some college (no degree)",
                                            "Associate's degree",
                                            "Bachelor's degree",
                                            "Graduate/professional degree",
                                            "Doctorate"))

temp_df <- tempFun_replaceWords(temp_data, "Education",
                                 codesMapping_education)

temp_data <- temp_df

# Save data --------------------------------------------------------------
data_paperAndEleReversedScoreAdded_imputed_converted <- temp_data
```

```
if (option_UpdateDatasetFlag | T){
  write.csv(temp_data,
          paste0("dataset/data_paperAndEleReversedScoreAdded_imputed_", option_version,
"_01.csv"), row.names = F)
}
```