Kate Barron
SI 601 Final Project: Effects of Holidays and Weekends on Blood Glucose Measures in the United States

Motivation

Diabetes is a condition in which the body either does not produce insulin (the hormone that helps transmit glucose from the bloodstream to the body's cells) or does not use it properly.[1][2] Prolonged high or low blood glucose levels can lead to serious complications, including kidney disease, heart disease and stroke.[3] There are many known factors which cause blood glucose levels to rise or fall, such as too little/too much food; too little/too much physical activity; dehydration; stress; short-term and long-term pain; and more.[4] For the purposes of this project, I want to explore whether variables in calendar data—such as weekend vs. weekday, or holiday—affect blood glucose measures. My expectation is that average blood glucose measures increase on holidays and weekends, since these occasions provide opportunities to deviate from diet/exercise routines.

Data Sources

I pulled a blood glucose measures data set from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Diabetes). The data set consisted of 70 tab separated values files, with the following fields:

- Date in MM-DD-YYYY format
- Time in XX:YY format
- Code
- Value

...where "code" refers to some special circumstance (although I ended up not exploring these values in the final analysis). For example:

33 = Regular insulin dose
34 = NPH insulin dose
35 = UltraLente insulin dose

The blood glucose measures were collected from an automatic electronic recording device and paper records. Because paper records only provided the time slots breakfast, lunch, dinner, bedtime, timestamps were assigned as 08:00, 12:00, 18:00, and 22:00, respectively. There were approximately 29330 records from the years 1988, 1989,1990 and 1991 in this dataset.

The holiday dataset was pulled from https://holidayapi.com/. To access the data, I made GET requests using the required parameters 'key' (API key), 'country' (US), 'year', and the optional parameters 'month' and 'day.' Data was returned in JSON format of the following structure:

```
{
 "status": 200,
 "holidays": [{
   "name": "Independence Day",
   "date": "2015-07-04"
   "observed": "2015-07-03"
   "public": true,
 }]
}
```

...where **status** signifies whether the GET request were successful ("200" indicates success) and the internal structure "**holidays**" includes the *name* of the holiday, its calendar *date*, the date it is *observed*, and whether it is *publicly* recognized (true/false). If there is no holiday on the requested day/month/year, the internal "holidays" structure returns an empty list. I retrieved records for all unique appearing in the diabetes data set--approximately 1141 records over the years 1988,1989,1990 and 1991.

Data Manipulation Methods

I created a script (create_diabetes_set.py) to read through the file paths for all 70 blood glucose measure data files; appended these paths to a list; and then looped through said list, writing their data to a master file called diabetes_concatenated.txt.

Next, I created a different script (createcal.py) to manipulate all 29330 records in the diabetes_concatenated.txt. For each record in diabetes_concatenated.txt, I created a dictionary of with the keys 'month,' 'day,' 'year,' 'time,' 'code,' and 'measure.' I used regular expressions to pull corresponding values out of said record. (The 'measure' regular expression was tricky to develop, as measures included between one and three digits, and occasionally included decimal points).

I then used the datetime module to identify the weekday corresponding to a record's numerical day (key, 'weekday'). All of these instrutions were placed in a try/except structure, because blood glucose measures were not formatted consistently. For instance, some measures included alphabetical characters, but because I could not interpret these entries, I decided to drop their full records from my analysis. In total, 70 records were lost and 29260 were maintained. All newly created dictionaries were appended to a list called my_diabetes_list.

(I should also note that while browsing the data, some of the blood glucose measurements seemed rather suspicious: for instance, there were measures of zero (which should not be possible), or else dangerously low (below 80 mg/dL)[5]. However, since I'm not a domain expert, I decided to keep all records with numerical values as measures.)

Using the dictionaries in my_diabetes_list, I created yet another list of dictionaries called api_list. The idea was to create a set of unique dictionaries which could be used to call the holiday API (with keys 'country', 'key' (API), 'day', 'month' and 'year'). Although country and API key would remain the same among these new dictionaries, I needed to pull month, day and year from the dictionaries in my_diabetes_list. Because there were repeat dates in my_diabetes_list, I used an "if not in" statement to append unique dictionaries to api_list (1140 dictionaries were created in total).

(Originally, I had tried to call on the holiday API for every record in my_diabetes_list--which was close to 30,000 records. This method appeared to be too stressful on my computer, as the program would crash after around 3,000 API calls).

Later, I used the api_list dictionaries to call on the holiday API, and appended the results to each one. To do so, I encoded each dictionary in api_list (using urllib.urlencode), appended the encoded data to https://holidayapi.com/v1/holidays?, and called/read the full url using urllib.urlopen and .read(). When appending the API results, I decoded the incoming JSON using json.loads.

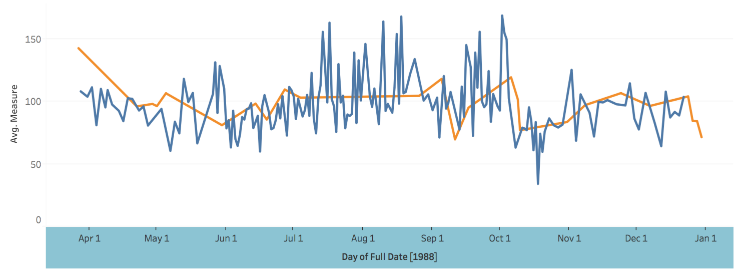
Ultimately, I needed to add these holiday API results to the dictionaries in my_diabetes_list. As such, where month, day and year matched between the dictionaries in my_diabetes_list and api_list, I created a 'results' key for the my_diabetes_list dictionaries (for which the api_list 'results' would be added as a value). I also took this opportunity to add two new key-value pairs to each my_diabetes_list dictionary: 'holiday' (with values True or False, depending on the 'results') and 'weekend' (also with values True or False, depending on the 'weekday'). I added these key-value pairs because I found they might be simpler ways of approaching my data analysis, at least at the start. I then wrote all of the my_diabetes_list data to a csv.

Analysis and Visualization

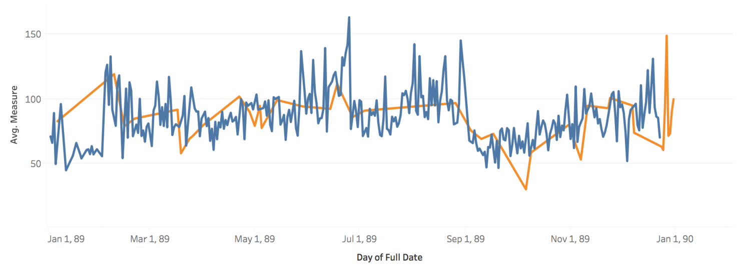
I decided to use Tableau to create data visualizations. I quickly realized that the most efficient way to compare blood glucose measures was to examine the average blood glucose measure on a given date. As such, I created a graphic that looked at average blood glucose measures by month for the years 1988-1991, coloring the lines by holiday or non-holiday.

However, I found this graphic a little difficult to read. I thought it would be better to map blood glucose measures by day, so as to get a more precise idea of holiday vs. non-holiday measures. These graphics were a little more helpful, though still somewhat odd to interpret, because average blood glucose measures seemed to spike and dip on a nearly daily basis. Holiday blood glucose measures seemed to follow the same kinds of dips and spikes, although they usually didn't dip as low as non-holiday blood glucose measures. In 1989, 1990 and 1991, holiday blood glucose measures seemed to consistently increase around Christmas and the New Year.

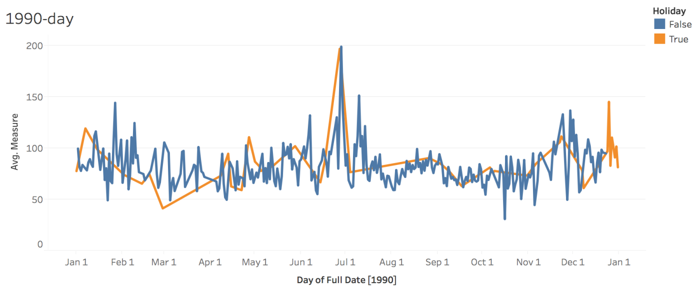
1988-day



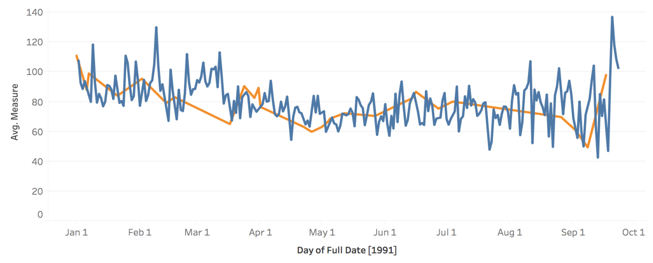
1989-day



1990-day

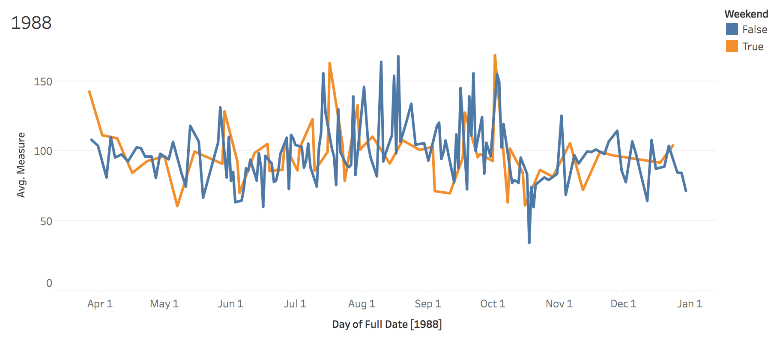


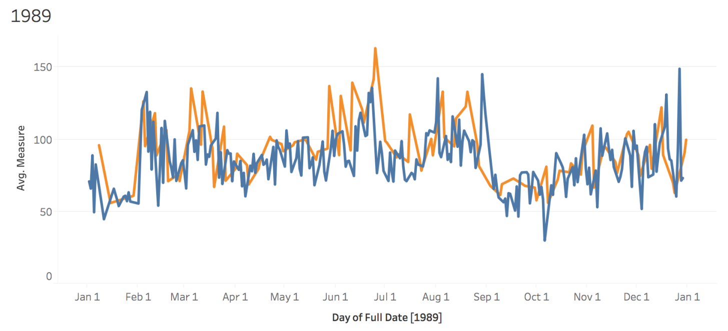
1991-day

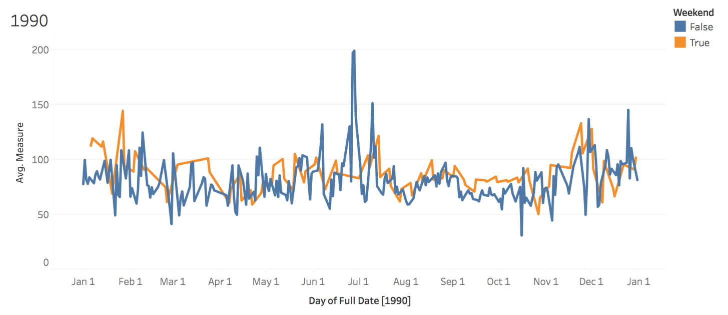


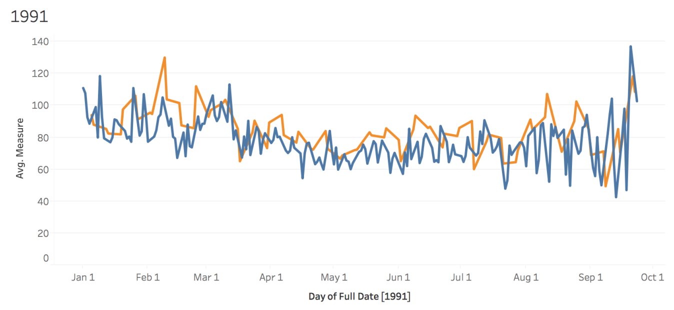
When I replaced the holiday variable with the weekend variable, I was surprised by how much more apparent increased blood glucose levels were on weekends. The orange lines (indicating weekend) were clearly higher than the blue lines (non-weekends), perhaps with the exception of the 1988 data. Here, in August and September, weekday blood glucose levels spiked several times.

In total, appears that holidays and weekends tend to correlate with higher blood glucose measures than non-holidays and weekdays; but that weekend blood glucose levels are markedly higher than weekday levels. Thus, even though weekends are ostensibly less "special" than holidays, they create more consistent risk for greater blood glucose levels. This reality is important in considering advisement for diabetes patients self-care.









[1] http://www.diabetes.org/diabetes-basics/type-1/?loc=util-header_type1
[2] http://www.diabetes.org/diabetes-basics/type-2/?loc=util-header_type2
[3] http://www.diabetes.org/living-with-diabetes/complications/
[4] http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/factors-affecting-blood-glucose.html?referrer=https://www.google.com/
[5] http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/checking-your-blood-glucose.html?referrer=https://www.google.com/