"*If you cannot measure it, you cannot improve it."*

-- Lord Kelvin

|  |  |
| --- | --- |
| *Concepts to cover*   * What is Fitbit? * Getting started with Fitbit and HealthVault * HealthVault PowerShell | *To do*   * Change the awakenings data for insomnia |

# Data is a powerful behavior change tool. The act of simply tracking changes one’s perception of that activity. Summarizing the data over time provides a yard stick by which to measure. The act of tracking activity overtime uncovers patterns in behavior and provides definitive answers to self-experimentation questions. The structured data in HealthVault provides such an opportunity. Moreover the HealthVault ecosystem offers a variety of applications and devices to assist in this endeavor. In this chapter we will explore how a consumer can use various devices to track critical health measures. We will also use common tools to explore the data stored by devices in to Microsoft HealthVault. We’ll capture and view some data, then use a PowerShell plugin to extract selected data to a CSV format and manipulate the data in that format.

Fitbit is being used in this chapter just to illustrate the ways you can use data from all kinds of devices, so long as they provide a gateway to HealthVault. If you’re not using Fitbit, I encourage you to download the sample Fitbit sleep data included as part of this book’s examples, and follow along.

# How Fitbit Tracks sleep

Fitbit is a pedometer on steroids thatenables you to monitor a number of aspects of daily living. This chapter concentrates on sleep, Fitbit has been very popular with users trying to understand and change their sleep patterns. Fitbit provides an arm band (Figure 2-1) that tracks whether you’re awake or asleep based on your activity level. Alternatively, the user can select an on/off mode to indicate whether she’s asleep.

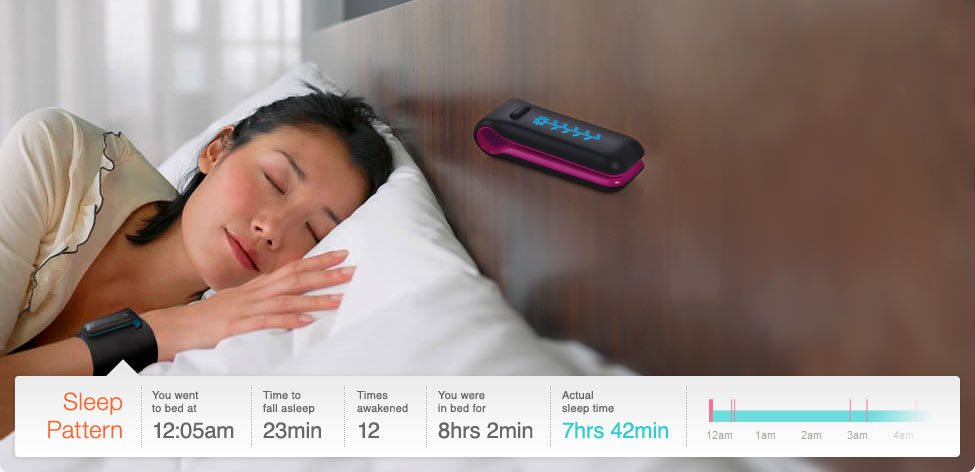


Fig 2-1. Fitbit being used to track sleep.

Fitbit also provides a base station that wirelessly uploads information from the device to the Fitbit web-service. Not having to worry about uploading information is a great value-add provided by this product.

## Sending Data to HealthVault

Fitbit enables users to sync their data automatically with HealthVault. Once you have a Fitbit account, you can choose the Share stats page (Figure 2-2) which becomes available after clicking on the account settings.

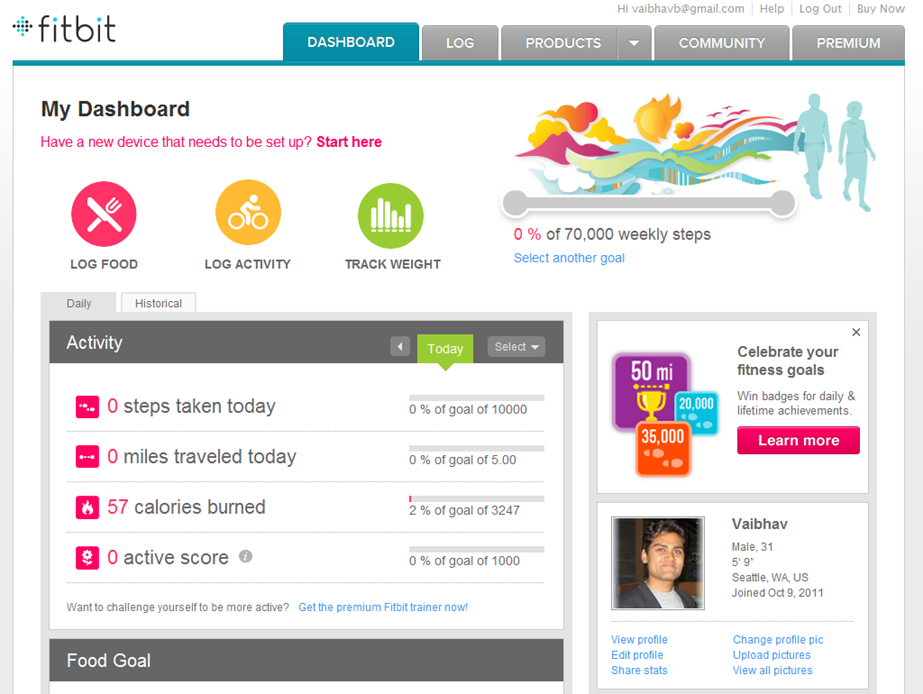
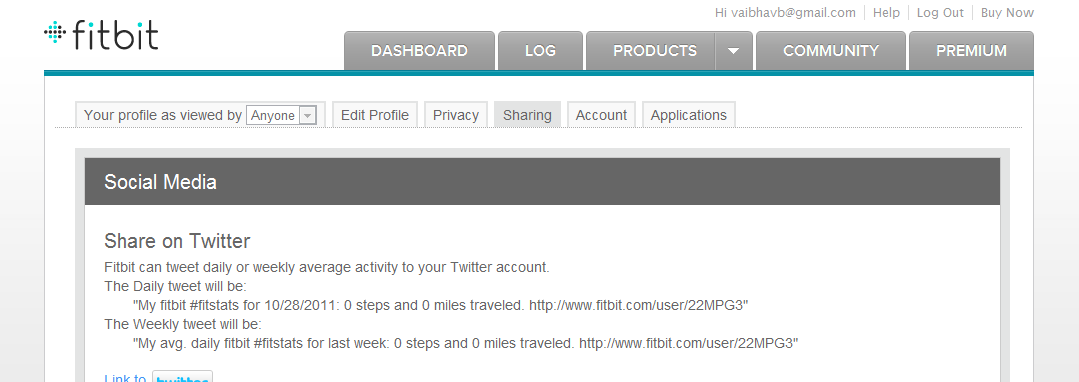


Fig 2-2. Fitbit Share stats feature

The Share Stats page, among other services, enables a link to HealthVault (Figure 2-3).



..

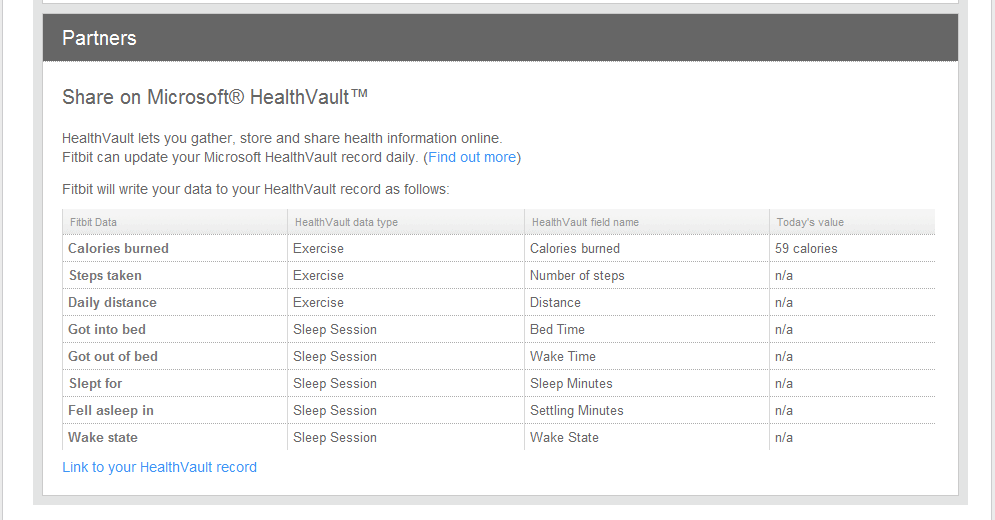


Fig 2-3. Connecting Fitbit with Microsoft HealthVault

Any application connecting to HealthVault has to get consent from the user for kinds of data it will be reading from or writing to Microsoft HealthVault. The user control is a two-step process. In the first step the user chooses the context of the record being authorized (Figure 2-4). As Figure 2-4 shows, in my case I have the option of using the application for my or my Mom’s record. In the second step the user grants access to specific health data being shared with the application (Figure 2-5). As Figure 2-5 shows, the Fitbit application wants to access to my Exercise, Sleep Session and other health information. We will learn in more detail about the user authentication and authorization system in chapter 3.

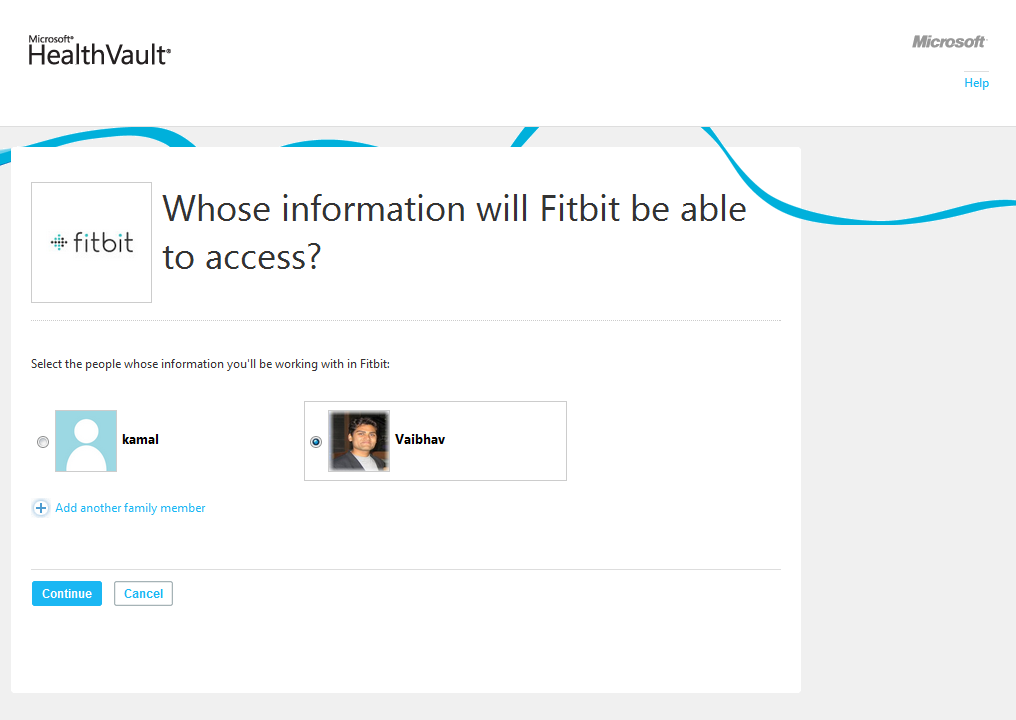


Fig 2-4. Choosing the context of a HealthVault record to work with an application

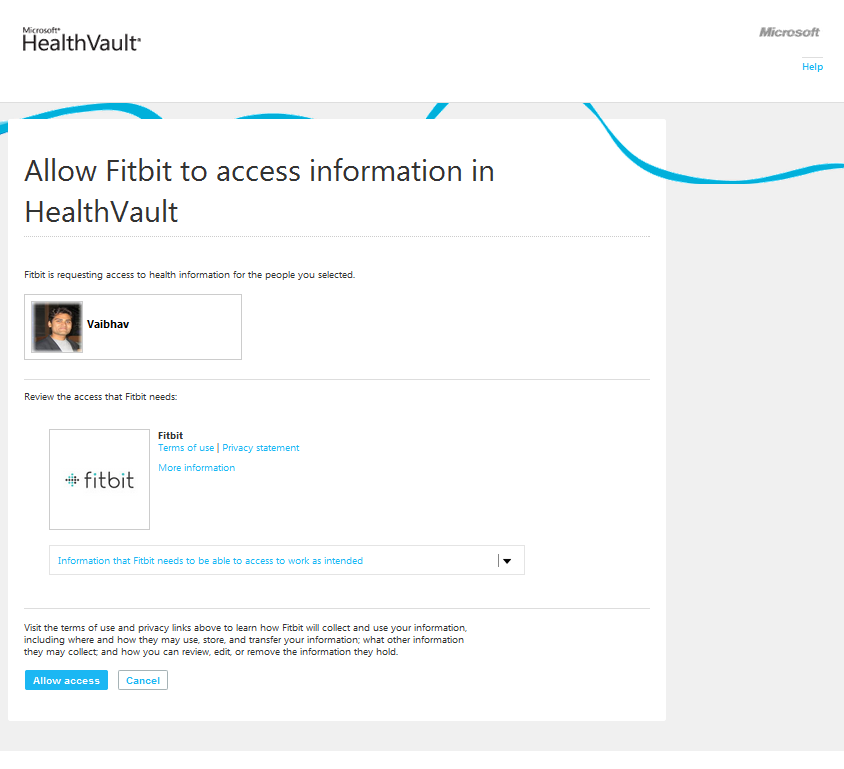


Fig 2-5. Authorizing an application to access a user’s Health data.

Clicking on the “Information that Fitbit needs to be able to access to work as intended”, you will notice that Fitbit wants to access a user’s Custom Data, Fitness, Measurements, and Personal Profile, as shown in Fig 2-6. In the line below the heading you will notice – Application Specific Information, Exercise, Sleep Session, Personal Contact Information and Personal Demographic Information, these are granular HealthVault data types. HealthVault has about 80+ granular data types which form the building blocks for various kinds of health information (Fitness, Measurement etc.). The data types are optimized to work with different devices and health systems. We will learn more about HealthVault data types and vocabularies in Chapter 4.

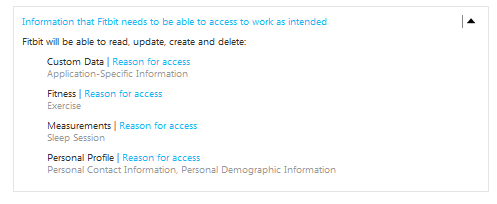


Fig 2-6. Granular details of the HealthVault data accessed by Fitbit

## Understanding the Data Model

Fitbit collects pedometer and sleep data. When the device syncs its data to HealthVault’s granular types it stores data as detailed in Table 2.1.

|  |  |  |
| --- | --- | --- |
| **Fitbit Data** | **HealthVault Data type** | **HealthVault Field Name** |
| Calories Burned | Exercise | Calories burned |
| Steps Taken | Exercise | Number of steps |
| Daily Distance | Exercise | Distance |
| Got in to bed | Sleep Session | Bed Time |
| Got out of bed | Sleep Session | Wake Time |
| Slept for | Sleep Session | Sleep Minutes |
| Fell asleep in | Sleep Session | Settling Minutes |
| Wake State | Sleep Session | Wake State |

Table 2-1. Fitbit HealthVault data mapping

As a user we are interested in tracking all the information about sleep as collected by Fitbit. As you will note from Table 2-1, we should look at the HealthVault Sleep Session data type for tracking sleep, and HealthVault Exercise data type for tracking Fitbit pedometer data.

# Exploring the HealthVault data

You can look at the data stored from Fitbit in the HealthVault user interface (sometimes referred as HealthVault Shell), as shown in Figure 2-7.

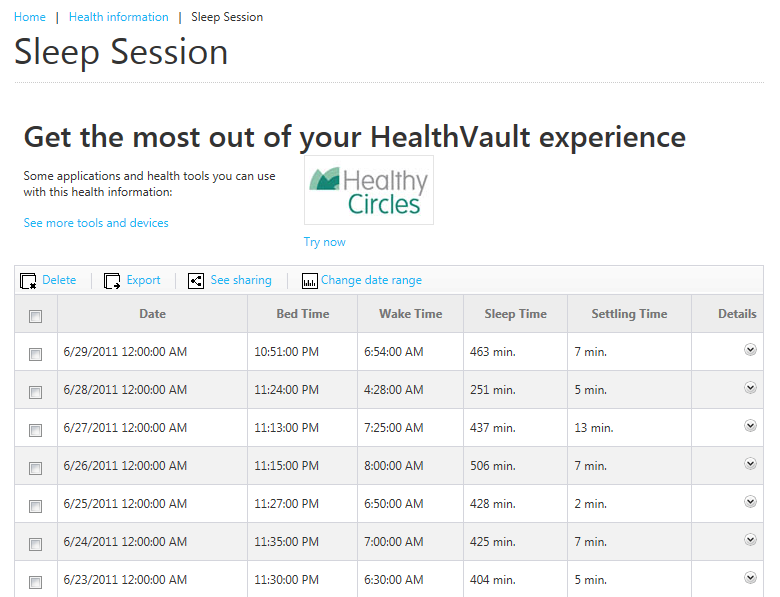


Fig 2-7. Sleep Session in HealthVault

Viewing information through the HealthVault user interface is convenient but a user cannot retrieve the entire information by exporting the data. As a power user and a quantifier I would like the data to be available to me to do some data-noodling. For that purpose, I would get this data in command-line format using the HealthVault PowerShell plugin (HvPosh). You can find the details of installing and extending this plug-in at <https://github.com/vaibhavb/HvPosh>. PowerShell can export data to a standard CSV format that can be consumed by a variety of other tools, simple or advanced, that let you do calculations and generate charts.

Once you have installed the PowerShell, load HealthVault’s plugin in to Windows PowerShell using

**Powershell> import-module HvPosh**

Then grant access to HealthVault Powershell interface using the following command line. Note this command will walk you through the same record picking interface and authentication and authorization interface as we used earlier for the Fitbit application.

**Powershell> Grant-HVPermission**

Once you have access to HealthVault within PowerShell, you can start using the utility from the command line and extract information pertinent to Sleep Session:

**Powershell>Get-Things -item Sleep-Session | format-table**

The results for my sample data are shown in Figure 2-8.

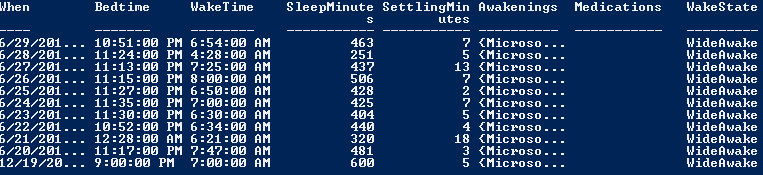


Figure 2-8. Structured data from Sleep Session as retrieved by PowerShell

--  
If you don’t have a Fitbit and want to follow along you can import the above data from the file Sleep-Data.xml available in the code associated with Chapter 3. Following is the command to import this data.

**Powershell>Import-HvDataXml –File Sleep-Data.xml**

--

You can understand the data further by using ways to look at this data by exploring the individual properties. Fig 2-9 shows how you can select particular properties of a HealthVault data type using PowerShell Select-object command.

**Powershell>** **get-things sleep | select-object When, Bedtime, WakeTime | format-table**

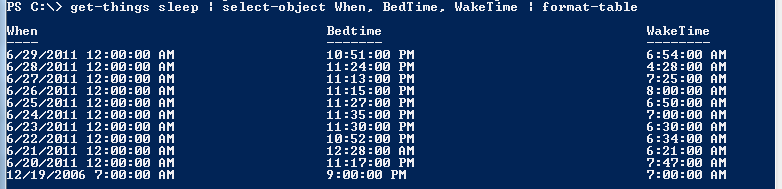


Figure 2-9. Three columns of structured data from Sleep Session as retrieved by PowerShell

In fact, I want to be able to explore detailed of awakenings. This is particularly relevant in knowing patterns of sleep disturbances.

**Powershell>** **get-things sleep | select-object -expandproperty Awakenings | format-table**

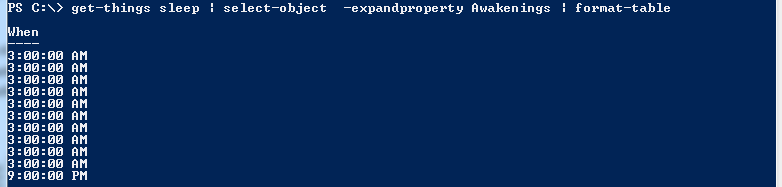


Fig 2-10. Understanding the pattern of Awakenings

Now you can enable some data crunching by exporting this data to a CSV file and switching the data analysis to Microsoft Excel or Google Spread sheets.

**Powershell>** **get-things sleep | select-object When, SleepMinutes, SettlingMinutes | export-csv SleepData.csv**

This creates the file SleepData.csv with the selected data.

# Analyzing the HealthVault Data

Once you have all the data in CSV file, you can open it in Excel (Figure 2-11) and analyze sleep patterns. You will notice that the spreadsheet has data for each sleep session specifying when that session occurred, the total sleep time in minutes and the time it took to get to bed termed as *settlingminutes*. I want to understand this data better so I create a sleep pattern X-Y scatter plot for this information [Fig 2-14].

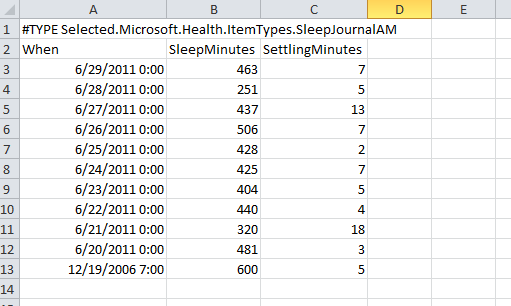


Fig 2-11. Sleep session data in an excel spread sheet

As Fig 2-12, reveals that for the duration of this week the median sleep has been around 400 minutes (i.e. around 6.5 hours), and as the data clearly shows the days when it took the longest to get to sleep the duration of sleep has been lower. So a good indicator of not been able to get to sleep in 10 minutes is a lower and poor quality of sleep.

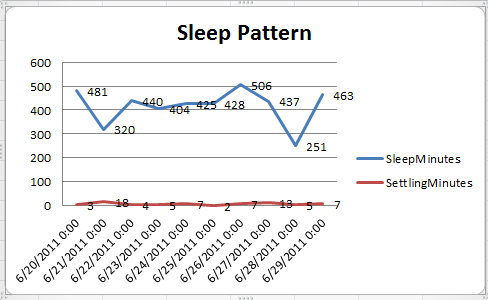


Fig 2-12. Sleep Pattern analysis

In fact for this duration I also want to understand the patterns around awakenings. So using powershell we generate another CSV file which focuses on awakenings.

**Powershell>** **get-things sleep | select-object effectivedate -expandproperty awakenings | Export-Csv d:\sleep-date-aw.csv**

We can open the file in Excel and visualize how the awakenings are triggered. It’s very obvious that the most awakening are for duration of 10 minutes around 3am. I know that is because the workshop in neighborhood is doing an early project and the noise around that time wakes me up.

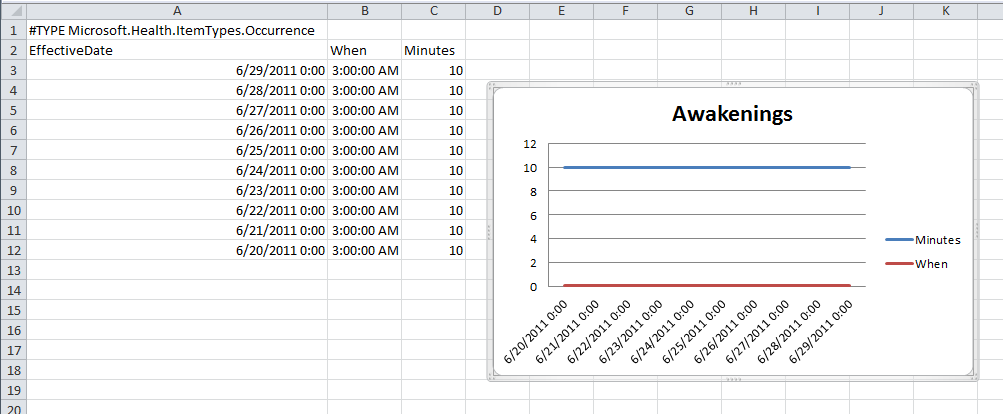


Fig 2-13. Awakenings Pattern Analysis

We can even take it a step further and correlate the sleep information with other types of data. With Fitbit we get the activity data and one can try and associated the steps information on the existing sleep data.

Using Powershell HealthVault plugin we can grab the appropriate fitness data from HealthVault.

**Powershell>** **Get-Things exercise | where-object {$\_.Activity.Name -eq "walk"} | select-object -expandproperty Activity | export-csv pedometer.csv**

Adding the pedometer data to what we obtained in Fig 2-12, gives us a way to correlate physical activity to sleep, as shown in Fig 2-14.

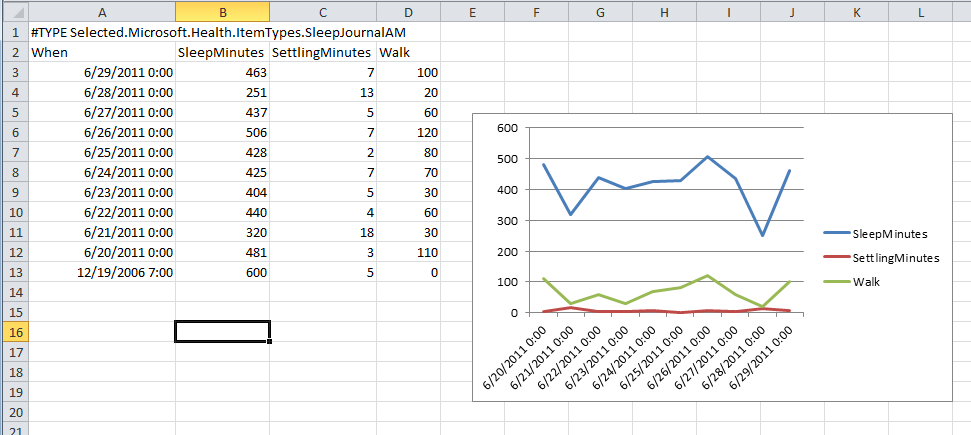


Fig 2.14 Correlating Sleep with Walking Exercise

Note that we scaled the steps information from pedometer to play nice on the graph by dividing it by 100. Using the above information one can say that possibly the days on which sleep was lower was possibly owing to lack of exercise on the other had the settling time was high on those days as well. So may be as a behavioral change one can mandate to get at least 5000 steps of walking to ascertain a good sleep.

This might change in long run, but that is the joy of learning from data and motivating a behavior shift! The associated website with this book, enablingquantifiedself.com, has a repository of spreadsheets giving inspiration for additional self-experimentation scenarios.

In upcoming chapters we will learn how we can automate some of the work we have done in this chapter with HealthVault application programming interface.