

CS 4561
HW 1 - Pt 2
Robert Whitmer

The function I have chosen to test for this assignment is **wellness()**, which can be found in [views/myworkouts.pug](#) in our public repository. For ease, I will paste the code on this document as an addendum, although it is more readable on github. This function was written by myself, and it's main use is to take workout data for a specific athlete and place information from that data into the wellness chart. There are 4 other charts with nearly identical functions, but the assignment said to pick a function, so I have selected this one. The function checks the dates of the workouts and if the workout is within that last 7 days, it's wellness rating is placed in the correct slot of the 7-wide array. Ideally, if the user has workouts in each of the last 7 days, the chart will correctly show each data point for those days' workouts. If a workout is missing, that data point should default to 0 and say invalid date. There are three decisions (ifs) in the code.

The main outer decision essentially says: **if**(# of workouts > 0) then do the work
Else display chart mentioning lack of data

This decision can easily be tested by having one test case where workouts are present and one test case without any workouts in that timeframe.

The next decision essentially is used inside the for loop to check if the date is within 7 days. This will be tested simply based on if a workout correctly shows in that range, as covered above.

The last decision in the code essentially checks to see if a calculated difference of dates is negative. If it is, then that means we are at a boundary of a month (which was a hard bug to catch) and the slot number will need to be multiplied by -1, essentially taking the absolute value in order to correctly place it in the correct spot. Unfortunately, this is where I will miss out on 100% decision coverage as I cannot test this until the first few days of some month as the function requires the ability to take the current date and manually placing a fake date would defeat the purpose of testing the automated nature of it. If you manage to wait a few days to grade these, then I will technically have 100% decision coverage.

TEST CASES

As such, my test cases will essentially involve adding/removing workouts for the recent week and checking to see if the chart displays correctly. Due to the manual nature of our application, there is no way to use a testing framework that wouldn't involve an insane amount of work. I will display my test for which days I will add workouts in the following format:

6 days ago	5 days ago	4 days ago	3 days ago	2 days ago	1 days ago	Today
------------	------------	------------	------------	------------	------------	-------

A test will be successful if the chart looks as intended for the appropriate data entered.

Test 1 (most common case, handles first two decisions passing)

Workout	Workout	Workout	Workout	Workout	Workout	Workout
---------	---------	---------	---------	---------	---------	---------

Test 2 (edge case)

No Workout	Workout	Workout	Workout	Workout	Workout	Workout
------------	---------	---------	---------	---------	---------	---------

Test 3 (edge case)

Workout	Workout	Workout	Workout	Workout	Workout	No workout
---------	---------	---------	---------	---------	---------	------------

Test 4 (handles first decision failing)

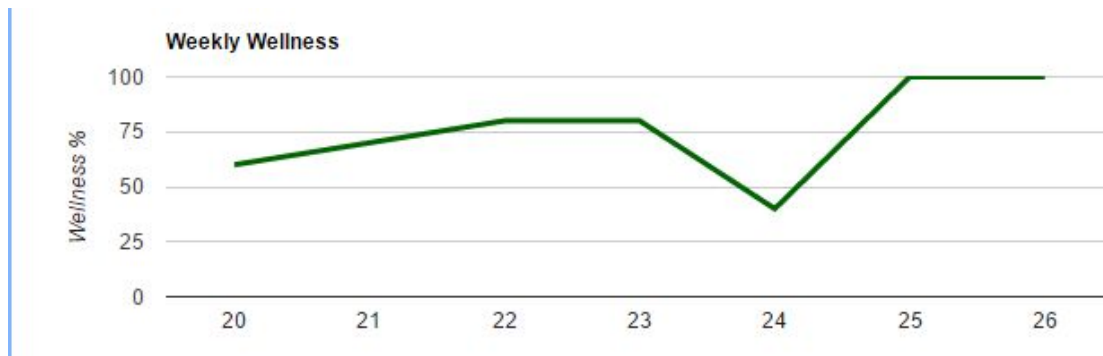
No Workout	No Workout	No Workout	No Workout	No Workout	No Workout	No Workout
------------	------------	------------	------------	------------	------------	------------

TEST RESULTS

The following will show the data entered into the user's account and then the chart that is displayed for each test case. The "Percent Health" field should match with the wellness score on the chart for each date.

Test 1 - Pass

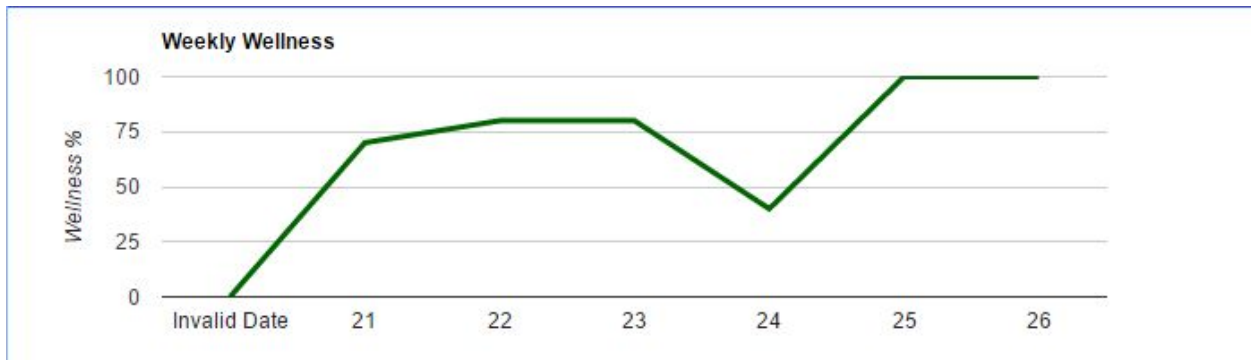
Date	Hours Slept	Illness	Injury	Percent Health	Cycle	RPE	Time	Distance	Notes	
Thu Jan 26 2017 00:00:01 GMT+0000 (UTC)	00	chronic soreness	rib	100	0	20	30	30	Hello my friends.	<div>EDIT</div> <div>REMOVE</div>
Wed Jan 25 2017 00:00:01 GMT+0000 (UTC)	07	n/a	n/a	100	0	12	20	2		<div>EDIT</div> <div>REMOVE</div>
Tue Jan 24 2017 00:00:01 GMT+0000 (UTC)	14	insomnia	shin	40	0	8	41	2		<div>EDIT</div> <div>REMOVE</div>
Mon Jan 23 2017 00:00:01 GMT+0000 (UTC)	08	upper respiratory	n/a	80	0	8	20	2		<div>EDIT</div> <div>REMOVE</div>
Sun Jan 22 2017 00:00:01 GMT+0000 (UTC)	06	upper respiratory	n/a	80	0	13	30	2		<div>EDIT</div> <div>REMOVE</div>
Sat Jan 21 2017 00:00:01 GMT+0000 (UTC)	07	upper respiratory	n/a	70	0	12	25	2		<div>EDIT</div> <div>REMOVE</div>
Fri Jan 20 2017 00:00:01 GMT+0000 (UTC)	04	insomnia	n/a	60	0	7	15	1		<div>EDIT</div> <div>REMOVE</div>



This test passed, as you can see all of the correct data was placed into the chart with the correct dates listed. The first decision's positive case has therefore also passed, as well as the second decision's positive and negative cases, as they must have occurred correctly as well.

Test 2 - Pass

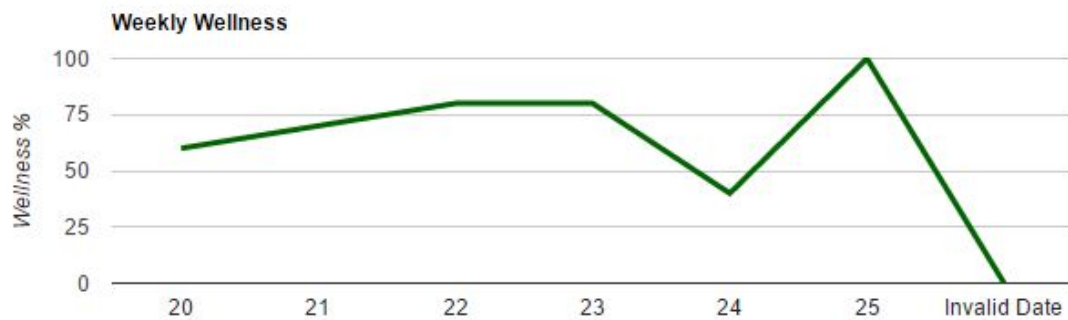
Date	Hours Slept	Illness	Injury	Percent Health	Cycle	RPE	Time	Distance	Notes	
Thu Jan 26 2017 00:00:01 GMT+0000 (UTC)	00	chronic soreness	rib	100	0	20	30	30	Hello my friends.	<div>EDIT</div> <div>REMOVE</div>
Wed Jan 25 2017 00:00:01 GMT+0000 (UTC)	07	n/a	n/a	100	0	12	20	2		<div>EDIT</div> <div>REMOVE</div>
Tue Jan 24 2017 00:00:01 GMT+0000 (UTC)	14	insomnia	shin	40	0	8	41	2		<div>EDIT</div> <div>REMOVE</div>
Mon Jan 23 2017 00:00:01 GMT+0000 (UTC)	08	upper respiratory	n/a	80	0	8	20	2		<div>EDIT</div> <div>REMOVE</div>
Sun Jan 22 2017 00:00:01 GMT+0000 (UTC)	06	upper respiratory	n/a	80	0	13	30	2		<div>EDIT</div> <div>REMOVE</div>
Sat Jan 21 2017 00:00:01 GMT+0000 (UTC)	07	upper respiratory	n/a	70	0	12	25	2		<div>EDIT</div> <div>REMOVE</div>



This test passed, you can see the one date we left out is displayed as 0 with an “Invalid Date” tag.

Test 3 - Pass

Date	Hours Slept	Illness	Injury	Percent Health	Cycle	RPE	Time	Distance	Notes	
Wed Jan 25 2017 00:00:01 GMT+0000 (UTC)	07	n/a	n/a	100	0	12	20	2		<div>EDIT</div> <div>REMOVE</div>
Tue Jan 24 2017 00:00:01 GMT+0000 (UTC)	14	insomnia	shin	40	0	8	41	2		<div>EDIT</div> <div>REMOVE</div>
Mon Jan 23 2017 00:00:01 GMT+0000 (UTC)	08	upper respiratory	n/a	80	0	8	20	2		<div>EDIT</div> <div>REMOVE</div>
Sun Jan 22 2017 00:00:01 GMT+0000 (UTC)	06	upper respiratory	n/a	80	0	13	30	2		<div>EDIT</div> <div>REMOVE</div>
Sat Jan 21 2017 00:00:01 GMT+0000 (UTC)	07	upper respiratory	n/a	70	0	12	25	2		<div>EDIT</div> <div>REMOVE</div>
Fri Jan 20 2017 00:00:01 GMT+0000 (UTC)	07	n/a	n/a	60	0	15	53	3		<div>EDIT</div> <div>REMOVE</div>



This test passed, you can see the one date we left out is displayed as 0 with an “Invalid Date” tag.

Test 4 - Somewhat Failure

Date	Hours Slept	Illness	Injury	Percent Health	Cycle	RPE	Time	Distance	Notes	
Thu Jan 19 2017 00:00:01 GMT+0000 (UTC)	06	n/a	n/a	90	0	17	100	10		<div>EDIT</div> <div>REMOVE</div>

(No data from this week)



This test somewhat failed. Ideally, the chart should've just displayed one string that says "No workouts for this week", but this behavior still looks nice and gets the point acrossed. I will probably look into tweaking this a bit to let the user know that there are no current workouts as well. This was the negative decision of the first if, so the outer else.

The entire test suite had a success rate of 75%, but one could argue that while the 4th test case did not display intended behavior, it is actually a more ideal outcome, albeit accidentally. With regard to the decision coverage, there were 3 decisions to be made, so 6 different sides of the decisions to test. 4 of those were testable (3 of which passed) and the other 2 could be tested reasonably (and I'm fairly sure it works as I've tested at the beginning of this month) in about 5 days.

Addendum

```
function wellness() {
    // Create the data table.
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'day');
    data.addColumn('number', 'hours');
    var oneWeekAgo = new Date();
    oneWeekAgo.setDate(oneWeekAgo.getDate() - 7);
    console.log(oneWeekAgo);
    var d = [{data_w};

    if(Object.keys(d).length > 0){

        var oneWeekAgo_1 = new Date();
        oneWeekAgo_1.setDate(oneWeekAgo + 1);
        var oneWeekAgo_2 = new Date();
        oneWeekAgo_2.setDate(oneWeekAgo + 1);
        var oneWeekAgo_3 = new Date();
        oneWeekAgo_3.setDate(oneWeekAgo + 1);
        var oneWeekAgo_4 = new Date();
        oneWeekAgo_4.setDate(oneWeekAgo + 1);
        var oneWeekAgo_5 = new Date();
        oneWeekAgo_5.setDate(oneWeekAgo + 1);
        var oneWeekAgo_6 = new Date();
        oneWeekAgo_6.setDate(oneWeekAgo + 1);
        var oneWeekAgo_7 = new Date();
        oneWeekAgo_7.setDate(oneWeekAgo + 1);

        var rows =
[[oneWeekAgo_1.toString(),0.0],[oneWeekAgo_2.toString(),0.0],[oneWeekAgo_3.toString(),0.0],[oneWeekAgo_4.toSt
ring(),0.0],[oneWeekAgo_5.toString(),0.0],[oneWeekAgo_6.toString(),0.0],[oneWeekAgo_7.toString(),0.0]];
        console.log(JSON.stringify(d));
        for(var i = 0; i < d.length; i++) {
            var obj = d[i];
            console.log("DATE: ");
            console.log(obj.date);
            var dt = new Date(Date.parse(obj.date));
            console.log(dt+1);
            console.log(oneWeekAgo);
            if(dt.getDate() >= oneWeekAgo.getDate()){

                var slot = dt.getDate() - oneWeekAgo.getDate();
                if(slot < 0)
                    slot = slot * (-1);
                console.log(dt.getDate() + 1);
                var dt_fixed = dt.getDate() + 1;
                rows[slot][0] = dt_fixed.toString();
                rows[slot][1] = parseInt(obj.percent_health, 10);
                //rows[dt.getDay()][1] = parseInt(obj.sleep, 10);
```

```

    }
  }
}
else{
  var dt = new Date();
  dt.setDate(dt.getDate());
  var rows = [["No workouts this week", 0]];
}
console.log("ROWS");
console.log(rows);
data.addRows(rows);
// Set chart options
var options = {'title':'Weekly Wellness',
'width':700,
'height':200,
legend: 'none',
lineWidth: 3,
backgroundColor: '#ffffff',
vAxis: {
title: "Wellness %",
viewWindowMode:'explicit'
},
colors: ['#006400'],
};
// Instantiate and draw our chart, passing in some options.
var chart = new google.visualization.LineChart(document.getElementById('wellnesschart'));
chart.draw(data, options);
}

```