The Design and Implementation of a Phone Experiment

Ryan D. Enos and Anthony Fowler

May 7, 2011

1 Introduction

After the November 2010 election, we closely followed the tight race for the state legislative seat in Worcester County's 6th district. After several recounts, the Republican candidate led by 1 vote, but one disputed vote in favor of the Democrat went to court. The election was declared a tie, and the state House called for a special election on May 10th.

Before the November election, we conducted surveys to reassess the calculus of voting. Riker and Ordeshook (1968) provided the canonical model of turnout as follows:

$$R = pB + D - C \tag{1}$$

An individual's returns to voting depend on the probability that her vote will shift the result of the election (p), how much she cares about the result (B), her extrinsic utility from the act of voting (D for duty), and the cost of voting (C). Surprisingly, citizens perceptions of p and reported levels of B are unrelated to turnout, suggesting that pB fails to factor into the typical turnout decision.

The special election in Worcester's 6th district provides a rare opportunity to test for the effect of p on turnout, because the probability that one vote will be pivotal is much higher than normal. Moreover, the low salience of the race means that few citizens may be aware of this fact. In our experiment, we will randomly inform some individuals that p is particularly high ("pivotal condition"). Other respondents will be assigned to a condition that simply reminds them of the election ("reminder condition"). The difference in turnout between these two conditions tests whether this increases their chances of voting.

2 Previous Experiments on Turnout

Before designing our experiment, we reviewed the vast literature on get-out-the-vote experiments `http://gotv.research.yale.edu/`. We also contacted Donald P. Green at Yale for advice on fielding this experiment. We want to reach voters in the most effective means possible. If we detect no effect, we want to attribute this to a minimal effect of p, and rule out the possibility that our experimental treatment was weak or ineffective.

One of our primary concerns is that many citizens in the district already know that the last election was tied. We first considered conducting a mail

experiment and a separate survey to assess the extent to which the public was informed about the election. Then, we decided that we could do both at once by administering a phone experiment but also asking some quick questions. Additionally, this approach gives us additional inferential leverage because we potentially restrict our analysis to individuals who were previously uninformed about the election or we can interact our experimental treatment with prior levels of knowledge.

Experiments conducted through professional phone banks are typically ineffective (Gerber and Green 2000), but volunteer phone calls (Nickerson 2006; Nickerson, Friedricks, and King 2006) can mobilize voters. According to Nickerson, phone calls are effective when the caller is personal, conversational, and informal. For this reason, we have asked Harvard students to volunteer in helping us make the calls. Callers should relax, introduce themselves by name, mention that they are Harvard students, and convey the relevant information without reading directly from the script.

3 Power Calculations

Before investing the time and money to conduct the experiment, we want to know whether we can detect a reasonably sized effect. Considering the size of the district, the time constraints, and the number of volunteer callers, we hope to make about 6000 calls (3000 phone numbers in each of the two experimental groups and 300 phone numbers in a survey condition). Using this simple back-of-the-envelope formula, we can determine the power of our test. How small of an effect can we detect. In technical language, what is the smallest effect for which could reject the null hypothesis that the effect is zero at the .05?

$$Effect/[St.Dev * sqrt(2/n)] > 2 \tag{2}$$

$Effect$ is the expected size of the experimental effect. $St.Dev.$ is the standard deviation in the outcome variable, in this case turnout in the special election. N is the sample size in each treatment group (setting them both to be the same size to maximize efficiency). Setting n to 3000, and the standard deviation to .5 (the largest possible standard deviation for a binary outcome variable), this simple calculation suggests that we can detect an effect of .026, or 2.6%. This is a conservative approximation because turnout

will probably be lower than 50% in this election (therefore the standard deviation will be less than .5), and we will be able to reduce the variance of our estimates by stratifying and conditioning. Nonetheless, this conservative power calculation gives us some confidence to move forward with the experiment. If the minimum detectable effect were 5% or more, there is little that could be learned from our experiment.

4 Data

To collect the list of registered voters in the district along with their phone numbers, vote histories, and other characteristics, we contacted two major vendors of political data, Catalist and Aristotle. These companies typically sell data to candidates, parties, PACs, interest groups, etc. but they are more than happy to work with academics. We purchased a file of approximately 19,000 registered voters at 10,000 different unique phone numbers from Catalist for approximately $1000. To confirm the validity of phone numbers, we conducted robo-calls (callfire.com) which automatically called every number on the list and confirmed whether each number was valid and recorded which numbers were answered by a live person. Approximately 46% of calls were answered, 52% were valid but unanswered (busy, voicemail, not answered), and 2% were invalid. We removed the invalid numbers to increase the efficiency of our calling efforts. We also removed phone numbers with more than 4 registered voters to improve the ease of identifying the person who answers each call.

5 Stratification

We could simply randomly assign each phone number to a treatment condition, but this could lead to two different problems. As we saw with the power calculation, a simple random assignment is inefficient, only allowing us to detect a 2.6% effect. Also, even with random assignment, there could by chance be slight imbalances between the treatment conditions. We can improve both balance and efficiency, by stratifying. This means that we will divide phone numbers into similar groups and randomize within each group. Typically, the more groups (strata) and more covariates used to stratify, the greater balance and efficiency. However, in this case, not every phone will be

answered, so we have to be careful. If nobody from the treatment (or control) group within a particular stratum answered the phone, then we would have to exclude the entire stratum from our analysis. To avoid this, we decided that every stratum should have at least 100 phone numbers. As a result, we could only stratify on a few important covariates.

Based on the literature, we know that the best predictor of turnout in the special election is likely to be previous turnout (if you voted in the past, you are likely to keep voting). We divided phone numbers into three categories based on the previous voting behavior of individuals at that numbers: (1) numbers where at least one person voted in a low salience special election in 2009, (2) numbers where at least one person voted in the 2010 general election but no one voted in the 2009 special election, (3) numbers where nobody voted in either election. In doing this, we are assuming that the special election in 2009 is a particularly good predictor of turnout in this election.

Next, we divided phone numbers by the number of voters listed at each number. We again divided numbers into three categories: (1) 1 registered voter, (2) 2 registered voters, (3) 3 or 4 voters (remember that we dropped numbers with more than 4).

Then, we decided that the town could be an important predictor of turnout in this election. In this district, roughly one-third of the voters live in Charlton, one-third live in Southbridge, and one-third live in a number of smaller towns. We divided the phone numbers into three categories accordingly.

Lastly, we decided to stratify based on our data collected from robo-calls. It is important that similar numbers of individuals in our treatment groups answer the phone, so separated the numbers where a live person answered from all others.

As a result of these categorizations, an individual phone number will fall into one of 54 unique categories (i.e. 1 registered voter, voted in 2010, Southbridge, failed to answer robo-call). Most of these strata had more than 100 phone numbers, but some fell short. To ensure that each stratum was of sufficient size, we combined smaller strata that shared everything in common except their answers of the robo-call. In one case, we also had to combine strata from different towns. This resulted in 44 unique strata all with at least 100 phone numbers. All subsequent randomization was conducted separately within strata.

6 Randomization

All randomization was conducted in Stata. Stata has a command [uniform()] which randomly generates real numbers between 0 and 1 according to a uniform distribution. If you have a data set in Stata, you can always randomly order it with the following code:

```
gen random = uniform()
sort random
drop random
```

Then, you can use this random ordering to assign observations into treatment groups. Since we randomized within each stratum, we did this separately for each one. For example, if the stratum labeled as number 1110 had 102 phone numbers in it, the code would look like this:

```
    treatment = 0
g rand = uniform() if stratum ==  1110
scalar stratum_size = 102
scalar big = round(stratum_size/3)
scalar small = round(stratum_size/30)
sort rand
replace treatment = 1 if _n < big + 1
replace treatment = 2 if _n > big & _n < big*2 + 1
replace treatment = 3 if _n > big*2 & _n < big*2 + small + 1
drop rand
```

This code created 4 different treatment groups from each stratum: (1) our reminder condition, one-third of the population, (2) our pivotal condition, one-third of the sample, (3) our survey condition, one-thirtieth of the sample, and (4) our no-contact condition.

This randomization scheme ensures that our two primary groups of interest will be exactly the same size and be identical in terms of the stratification variables. However, there could still be imbalance on some other important variables such as age, race, or turnout in other elections. To improve balance on other covariates, we generated 500 different randomizations and evaluated them based on the balance of other variables across the two primary treatment groups. Stata allows you to do this with ease, using the command, forvalues. If we wanted to generate 500 different versions of the randomization above, we could do so with this code:

```
forvalues i = 1/500{
g treatment‘i’ = 0
qui:g rand = uniform() if stratum ==  1110
scalar stratum_size = 102
scalar big = round(stratum_size/3)
scalar small = round(stratum_size/30)
sort rand
qui:replace treatment‘i’ = 1 if _n < big + 1
qui:replace treatment‘i’ = 2 if _n > big & _n < big*2 + 1
qui:replace treatment‘i’ = 3 if _n > big*2 & _n < big*2 + small + 1
drop rand
}
```

This code generated 500 new variables (treatment1, treatment2, etc.). The qui: that preceded some of the lines simply suppresses Stata's visual display of the operation and improves the computational speed of this process.

7 Placebo Tests

Having completed the randomization and checked for balance, we can conduct some additional tests that provide a better indication of our statistical power. We conducted several placebo tests, estimating the (fake) effect of our treatment on turnout in previous elections. Comparing our two experimental groups of interest, we can estimate the effect of being in the pivotal treatment on turnout in a 2008 special election employing OLS with stratum fixed-effects. Effectively, this procedure calculates the average turnout level of each stratum, subtracts the stratum average from each person's turnout, and then calculates the difference in these values between the two treatment groups. As expected, the estimated placebo effect was zero. More importantly, the standard error across several different tests was less than .01, suggesting that we can confidently identify any effect larger than 2%.

8 Experiment implementation

With our list of subjects randomized and checked for balance, it is time to conduct our experiment. This required us to recruit callers and to set up the

call infrastructure.

8.1 caller recruitment

As mentioned above, based on a review of the literature, we decided that Harvard undergraduate and graduate students would be the most effective callers. We recruited these callers after writing the script for calls and calculating the length of typical call by including ring time and the estimated time of the conversation. Based on the time of an average call and the contact rate we were able to determine from the robo-calls, we calculated the number of completed calls we would be able to make an hour. With this hourly rate of contact, we recruited enough callers to be able to complete our targeted number of calls.

The scripts used for our treatments are located in Appendix 1.

8.2 call infrastructure

We needed office space and phones to complete the calls so we consulted with the Harvard Institute for Quantitative Social Science to use the space and phones. We then needed a standardized way to collect the data recorded in the calls. When callers record the outcome of their calls, we could have record the data with paper and pencil, but the disadvantage of this method is the inefficiency and potential for error involved in then having to digitalize the data. In order to allow our callers to enter data and simultaneously have aggregated to a single source we created an webpage that combined data entry and the call scripts for our treatments. The webpages were created using Python cgi scripts and are located on our personal webspace. We password protected the pages to protect the data as required by the Harvrd IRB.

The cgi scripts consist of two programs, the first creates the webpage with the call scripts and html forms that are used to enter the data. The script was designed to take information from the list of subjects and output the information to the webpage so that callers would know the names of the household members and other important information, such as the phone number for each call. The cgi script "call_page.cgi" is in Appendix 2.

We then needed a script to take the data entered in the original script and write it to a file. This script "save.cgi" is in the Appendix 2.

9 Appendix 1: call scripts

Reminder Treatment

Hello, this is XXXX from Harvard University. We are calling registered voters to provide information about an upcoming election in your town.

Am I speaking with XXXXXXXX?
Did you know that there is a special election coming up?

[IF YES] Do you know when it is?

We just want to remind you that there's a special election on Tuesday, May 11th to fill the seat of your representative in the Massachusetts State House. For more information on the election you can visit the website of the Secretary of the Commonwealth.

Goodbye

Pivotal Treatment

Hello, this is XXXX from Harvard University. We are calling registered voters to provide information about an upcoming election in your town.

Am I speaking with XXXXXXXX?
Did you know that there is a special election coming up?

[IF YES] Do you know when it is?

We just want to remind you that there's a special election on Tuesday, May 11th to fill the seat of your representative in the Massachusetts State House. For more information on the election you can visit the website of the Secretary of the Commonwealth.

The reason that there is a special election is that the last election ended in an exact tie. Had one more or one less person voted in the last election, your candidate would have won. The special election on Tuesday is likely to be close again, so there is a high chance that your vote could make a difference.

Goodbye

Hello, this is XXXX from Harvard University.
We are conducting a survey of registered voters in your town and I would like just a few minutes of your time.

This survey is for research purposes only and all answers are anonymous.

Are you willing to answer a few questions about elections in your town?

[YES - CONTINUE TO SURVEY]
[NO - Thank you, goodbye.]
Thankyou for participating. I am about to ask you a series of questions about the election. There are lots of reasons that people might not pay attention to politics and elections, for example they might be too busy with work and family or they might just not be interested because they know that it does not affect them.

1. Did you know that there is a special election coming up? [NO, SKIP TO QUESTION 4. YES, PROCEED TO QUESTION 2]
2. Do you know when it is?
3. Do you know who is running in this election?
4. The election will be on Tuesday, May 10 for the seat that represents you in the Massachusetts State House. There is not usually an election this time of year. Do you know why we need to have a special election now?

Thank you for participating.

For more information on the election you can visit the website of the Secretary of the Commonwealth.

Goodbye

10 Appendix 2: python code for webpages and data aggregation

10.1 call_page.cgi

```
#!/usr/bin/env python

import time
import cgi, cgitb
import csv
from os import environ


##THE FILE WITH THE SUBJECTS, RANDOMIZED
inf = open('Treatment File   Random.csv', 'rU')
incsv = csv.reader(inf)

contents = list()

counter = 0

###CREATE COOKIE FOR CALLERS NAME
if environ.has_key('HTTP_COOKIE'):
    cookie = environ['HTTP_COOKIE']
    caller = str(cookie)
    caller = cookie.split('=')[1]
else:
    caller = 'enter your name'


###TIME OF THE CALL, NEEDED TO APPEND TO OUTDATA
callTime = time.ctime( time.time() )
callTime = str(callTime)

###CYCLE THROUGH THE FILE OF SUBJECTS, TAKE THE FIRST ROW AND WRITE IT TO
    THE HTML PAGE, TAKE THE REST AND REWRITE THE FILE WITH THE NEW DATA
for row in incsv:
    counter = counter + 1
    if counter == 1:
        link = row
        treatment = row[1]
        number = row[0]
        areaCode = str(number[0:3])
        prefix = str(number[3:6])
        suffix = str(number[6:10])
        numberFormat = areaCode+'-'+prefix+'-'+suffix

        first1 = row[2]
        first2 = row[3]
        first3 = row[4]
        first4 = row[5]

        last1 = row[6]
        last2 = row[7]
```

```
        last3 = row[8]
        last4 = row[9]

        age1 = row[10]
        age2 = row[11]
        age3 = row[12]
        age4 = row[13]

        gender1 = row[14]
        gender2 = row[15]
        gender3 = row[16]
        gender4 = row[17]

###START OF THE HTML CODE, INTERRUPTED BY PYTHON COMMANDS THAT USE BOOLEAN
    LOGIC TO SPECIFY CONDITIONS BASED ON THE RANDOMIZATION
###STRING METHODS ARE USED TO ADD PYTHON OBJECTS INTO THE HTML WHERE THE '%s
    ' APPEARS
        print 'Content-type: text/html'
        print # Prints an empty line, to end the headers
        #print link
        print """

        <html>
            <head>
                <title> Call Page </title>
            <style type='text/css'>
                    .textborder {border: 1px solid blue; padding: 2px;}
                    .horiz { float: left; padding: 0 15px; list-style: none}
                    .script {color: red; font-size: large}
                    </style>

            </head>
            <body>
            <h1>
                     <center><span style='color:red'> %s </span></center>
                    </h1>
            <hr>
            <center>
                <!--SUBJECT TALBE-->
                    <table border="1">
                    <tr>
                        <th>Person 1</th>
                 """ % numberFormat
        if first2 != '':
            print """
                        <th>Person 2</th>
                    """
        if first3 != '':
            print """
                        <th>Person 3</th>
                    """
        if first4 != '':
            print """
                        <th>Person 4</th>
                    """
        print """
                        </tr>
```

```
            <tr>
              <td> %s %s (%s,%s) </td>   <!--person1-->
          """ % (first1,last1,age1,gender1)
if first2 != '':
    print """
              <td> %s %s (%s,%s) </td>   <!--person2-->
            """ % (first2,last2,age2,gender2)
if first3 != '':
    print """
              <td> %s %s (%s,%s) </td>   <!--person3-->
            """ % (first3,last3,age3,gender3)
if first4 != '':
    print """
              <td> %s %s (%s,%s) </td>   <!--person4-->
            """ % (first4,last4,age4,gender4)
print """
                </tr>
                </table>
            <!--END SUBJECT TALBE-->

        </center>

    <hr>

            <form action = 'save.cgi'>
                <input type='hidden' name = 'number' value = '%s' />
                <input type='hidden' name = 'treatment' value = '%s'
                    />
                <input type='hidden' name = 'callTime' value = '%s'
                    />

                <fieldset>
                    <legend>call outcome</legend>
                        <ul>
                            <li class = 'horiz'>
                                <label for 'answered'>
                                    <input type='radio' name = '
                                        outcome' id='answered' value
                                        ='answered' /> Answered
                                    </label>
                                </li>
                            <li class = 'horiz'>
                                <label for 'machine'>
                                    <input type='radio' name = '
                                        outcome' id='machine' value
                                        ='machine' /> Answering
                                        Machine/Voice Mail
                                    </label>
                                </li>
                            <li class = 'horiz'>
                                <label for 'busy'>
                                    <input type='radio' name = '
                                        outcome' id='busy' value='
                                        busy' /> Busy Signal
                                    </label>
                                </li>
                            <li class = 'horiz'>
```

```
                                    <label for 'no_answer'>
                                        <input type='radio' name = '
                                            outcome' id='no_answer'
                                            value='no_answer' /> No
                                            Answer
                                        </label>
                                    </li>
                                </ul>
                            </fieldset>
                    """ % (number,treatment,callTime)

if treatment == "1" or treatment == "2":
    print """
                    <div class='textborder'>
                    <p class = 'script'> Hello, this is %s from Harvard
                        University. We are calling registered voters to
                         provide information about an upcoming election
                        in your town. </p>
                    <p class = 'script'> Am I speaking with [RESPONDENT
                        NAME]? </p>
                    """ % caller

if treatment == "3":
    print """
                    <div class='textborder'>
                   <p class = 'script'>Hello, this is %s from Harvard
                        University. </p>
                    <p class = 'script'> We are conducting a survey of
                        registered voters in your town and I would like
                        just a few minutes of your time.</p>
                    <p class = 'script'> Am I speaking with [RESPONDENT
                        NAME]? </p>
                """ % caller

    ####RESPODENT BUTTONS########
print """
                        <fieldset>
                            <ul>
                                <li class = 'horiz'>
                                    <label for 'NA'>
                                      <input type='radio' name = '
                                          contact' id='NA' value='NA'
                                          checked/> N/A
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'person1'>
                                        <input type='radio' name = '
                                            contact' id='person1' value
                                            ='person1' /> %s
                                        </label>
                                    </li>
                    """ % first1
if first2 != '':
    print """
                                <li class = 'horiz'>
                                    <label for 'person2'>
```

```
                                    <input type='radio' name = '
                                        contact' id='person2' value
                                        ='person2' /> %s
                                    </label>
                                </li>
                    """ % first2
if first3 != '':
    print """
                                <li class = 'horiz'>
                                    <label for 'person3'>
                                        <input type='radio' name = '
                                            contact' id='person3' value
                                            ='person3' /> %s
                                    </label>
                                </li>
                    """ % first3
if first4 != '':
    print """
                                <li class = 'horiz'>
                                    <label for 'person4'>
                                        <input type='radio' name = '
                                            contact' id='person4' value
                                            ='person4' /> %s
                                    </label>
                                </li>
                    """ % first4
print """
                                <li class = 'horiz'>
                                    <label for 'other'>
                                        <input type='radio' name = '
                                            contact' id='other' value='
                                            other' /> Another Person
                                    </label>
                                </li>
                            </ul>
                        </fieldset>
                    """
    ####END RESPODENT BUTTONS########
if treatment == "3":
    print """
                <p class = 'script'> This survey is for research
                    purposes only and all answers are anonymous and
                    it will take just a few minutes of your time. </
                    p>
                <p class = 'script'> Are you willing to answer a few
                     questions about elections in your town? </p>
                <p class = 'script'> <span style='color:blue'>[YES -
                     CONTINUE TO SURVEY]</span> </p>
                <p class = 'script'> <span style='color:blue'>[NO]
                    </span> Thank you, goodbye. </p>
                <p class = 'script'> Thankyou for participating. I
                    am about to ask you a series of questions about
                    the election. There are lots of reasons that
                    people might not pay attention to politics and
                    elections, for example they might be too busy
                    with work and family or they might just not be
                    interested because they know that it does not
```

```
                affect them. </p>
                """

if treatment == "1" or treatment == "2":
    print """
                <p class = 'script'> Did you know that there is a
                    special election coming up? </p>
                    <fieldset>
                            <ul>
                                <li class = 'horiz'>
                                    <label for 'NA'>
                                        <input type='radio' name = '
                                            knows_election' id='NA'
                                            value='NA' checked/> N/A
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'yes'>
                                        <input type='radio' name = '
                                            knows_election' id='yes'
                                             value='yes' /> Yes
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'no'>
                                        <input type='radio' name = '
                                            knows_election' id='no'
                                            value='no' /> No
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'refused'>
                                        <input type='radio' name = '
                                            knows_election' id='
                                            refused' value='refused'
                                             /> refused
                                        </label>
                                    </li>
                                </ul>
                        </fieldset>
                <p class = 'script'> <span style='color:blue'>[IF
                    YES]</span> Do you know when it is? </p>
                        <fieldset>
                            <ul>
                                <li class = 'horiz'>
                                    <label for 'NA'>
                                        <input type='radio' name = '
                                            election_date' id='NA'
                                            value='NA' checked/> N/A
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'yes'>
                                        <input type='radio' name = '
                                            election_date' id='yes'
                                            value='yes' /> Yes
                                        </label>
```

```
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'no'>
                                        <input type='radio' name = '
                                            election_date' id='no'
                                            value='no' /> No
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'vague'>
                                        <input type='radio' name = '
                                            election_date' id='vague
                                            ' value='vague' /> vague
                                        </label>
                                    </li>
                                <li class = 'horiz'>
                                    <label for 'refused'>
                                        <input type='radio' name = '
                                            election_date' id='
                                            refused' value='refused'
                                             /> refused
                                        </label>
                                    </li>
                                </ul>
                            </fieldset>
                    <p class = 'script'> We just want to remind you that
                         there's a special election on Tuesday, May 11th
                         to fill the seat of your representative in the
                        Massachusetts State House. For more information
                         on the election you can visit the website of
                        the Secretary of the Commonwealth. </p>
                """
if treatment == "2":
    print """
                    <p class = 'script'> <i> The reason that there is a
                         special election is that the last election ended
                         in an exact tie. Had one more or one less
                        person voted in the last election, your
                        candidate would have won. The special election
                        on Tuesday is likely to be close again, so there
                         is a high chance that your vote could make a
                        difference. </i> </p>
                """
if treatment == "3":
    print """
                    <p class = 'script'> 1. Did you know that there is a
                         special election coming up? <span style='color:
                        blue'>[NO, SKIP TO QUESTION 4. YES, PROCEED TO
                        QUESTION 2]</span> </p>
                        <fieldset>
                            <ul>
                                <li class = 'horiz'>
                                    <label for 'NA'>
                                        <input type='radio' name = '
                                            knows_election' id='NA'
                                            value='NA' checked/> N/A
                                        </label>
```

```
                    </li>
                <li class = 'horiz'>
                    <label for 'yes'>
                        <input type='radio' name = '
                            knows_election' id='yes'
                             value='yes' /> Yes
                    </label>
                </li>
                <li class = 'horiz'>
                    <label for 'no'>
                        <input type='radio' name = '
                            knows_election' id='no'
                            value='no' /> No
                    </label>
                </li>
                <li class = 'horiz'>
                    <label for 'refused'>
                        <input type='radio' name = '
                            knows_election' id='
                            refused' value='refused'
                             /> refused
                    </label>
                </li>
            </ul>
        </fieldset>
<p class = 'script'> 2. Do you know when it is? </p>
            <fieldset>
            <ul>
                <li class = 'horiz'>
                    <label for 'NA'>
                        <input type='radio' name = '
                            election_date' id='NA'
                            value='NA' checked/> N/A
                    </label>
                </li>
                <li class = 'horiz'>
                    <label for 'yes'>
                        <input type='radio' name = '
                            election_date' id='yes'
                            value='yes' /> Yes
                    </label>
                </li>
                <li class = 'horiz'>
                    <label for 'no'>
                        <input type='radio' name = '
                            election_date' id='no'
                            value='no' /> No
                    </label>
                </li>
                <li class = 'horiz'>
                    <label for 'refused'>
                        <input type='radio' name = '
                            election_date' id='
                            refused' value='refused'
                             /> refused
                    </label>
                </li>
```

```
                </ul>
            </fieldset>
<p class = 'script'> 3. Do you know who is running
    in this election? </p>
        <fieldset>
            <ul>
            <li class = 'horiz'>
            <label for 'NA'>
                    <input type='radio' name = '
                        candidates' id='NA'
                        value='NA' checked/> N/A
                    </label>
                </li>
            <li class = 'horiz'>
                    <label for 'both'>
                        <input type='radio' name = '
                        candidates' id='both'
                        value='yes' /> both
                    </label>
                </li>
            <li class = 'horiz'>
                    <label for 'alicea'>
                        <input type='radio' name = '
                        candidates' id='alicea'
                        value='alicea_only' />
                        Geraldo Alicea
                    </label>
                </li>
            <li class = 'horiz'>
                    <label for 'durant'>
                        <input type='radio' name = '
                        candidates' id='durant'
                        value='durant_only' />
                        Peter Durant
                    </label>
                </li>
            <li class = 'horiz'>
                    <label for 'durant'>
                        <input type='radio' name = '
                        candidates' id='neither'
                         value='neither' />
                        neither
                    </label>
                </li>
            <li class = 'horiz'>
                    <label for 'refused'>
                        <input type='radio' name = '
                        candidates' id='refused'
                         value='refused' />
                        refused
                    </label>
                </li>
            </ul>
        </fieldset>
<p class = 'script'> 4. The election will be on
    Tuesday, May 10 for the seat that represents you
     in the Massachusetts State House. There is not
```

```
        usually an election this time of year. Do you
        know why we need to have a special election now?
         </p>
        <fieldset>
                <ul>
                    <li class = 'horiz'>
                        <label for 'NA'>
                            <input type='radio' name = '
                                why' id='NA' value='NA'
                                checked/> N/A
                            </label>
                        </li>
                    <li class = 'horiz'>
                        <label for 'yes'>
                            <input type='radio' name = '
                                why' id='yes' value='yes
                                ' /> yes (election was a
                                 tie)
                            </label>
                        </li>
                    <li class = 'horiz'>
                        <label for 'no'>
                            <input type='radio' name = '
                                why' id='no' value='no'
                                /> no
                            </label>
                        </li>
                    <li class = 'horiz'>
                        <label for 'partial'>
                            <input type='radio' name = '
                                why' id='partial' value
                                ='partial' /> partial
                                knowledge
                            </label>
                        </li>
                    <li class = 'horiz'>
                        <label for 'unclear'>
                            <input type='radio' name = '
                                why' id='unclear' value
                                ='unclear' /> unclear
                            </label>
                        </li>
                    <li class = 'horiz'>
                        <label for 'refused'>
                            <input type='radio' name = '
                                why' id='refused' value
                                ='refused' /> refused
                            </label>
                        </li>
                    </ul>
                    </fieldset>
            <p class = 'script'> Thank you for participating. </
                p>
            <p class = 'script'> For more information on the
                election you can visit the website of the
                Secretary of the Commonwealth. </p>
"""
```

```
        print """
                        <p class = 'script'> Goodbye </p>
                         </div>

                         <label for 'notes'> Notes
                             <textarea name ='notes' id = 'notes' rows='6'
                                  cols='50'> </textarea>
                             </label>

                         <label for 'caller'> Caller
                             <input type='text' id = 'caller' name = 'caller'
                                  value = %s>
                                 </label>
                         <center>
                          <p>
                             <input type = 'submit'  value = 'click here when
                                  call is complete' style='background-color:
                                 #99FF99;' />
                             </p>
                             </center>

                         </form>

                </body>
            </html>
            """ % caller

    else:
        contents.append(row)


inf.close()

###WRITE OUT THE NEW FILE FOR THE NEXT CALL

fout = file('Treatment File - Random.csv',"w")
outcsv = csv.writer(fout,lineterminator="\n")

for item in contents:
    outcsv.writerow(item)

fout.close()
```

10.2 save.cgi

```
#!/usr/bin/env python

import cgi
import csv

###WRITE THE INFORMATION FROM THE FORMS TO A NEW FILE
form = cgi.FieldStorage()
outrecord = open('call_outcomes.csv',"a")
outrecordCsv = csv.writer(outrecord, lineterminator = '\n')

##input from the forms on the previous page.  Assign them a name and give a
```

```
    name to the default
number = form.getvalue('number','no_data')
treatment = form.getvalue('treatment','no_data')
callTime = form.getvalue('callTime','no_data')
outcome = form.getvalue('outcome', 'no_data')
notes = form.getvalue('notes', 'no_data')
contact = form.getvalue('contact','no_data')
knows_election = form.getvalue('knows_election','no_data')
election_date = form.getvalue('election_date','no_data')
survey_agree = form.getvalue('survey_agree','no_data')
candidates = form.getvalue('candidates','no_data')
why = form.getvalue('why','no_data')
caller = form.getvalue('caller','no_data')

###COOKIE WITH THE CALLERS NAME
print "Set-Cookie:caller=%s;\r\n" % caller

###APPEND TO THE NEW FILE
output = [number,treatment,callTime,caller,outcome,contact,notes,
    knows_election,election_date,survey_agree,candidates,why]
outrecordCsv.writerow(output)
outrecord.close()

##SEND THE CALLER BACK TO THE ORIGINAL CALL PAGE
print("Location:call_page.cgi")
```