Implementing Blocked Randomization in Online Survey Experiments

Charlotte Cavaille*
Georgetown University

Contents

*Email: cc1933@georgetown.edu. I am grateful to Kosuke Imai, Dan L. Chen and Rich Nielsen for their help. Diag Davenport has provided stellar research assistance and should be fully credited for developing the remote platform which runs the `seqblock` R function. Readers should feel free to email with suggestions on how to improve this memo or to flag mistakes. This project was supported by a School of Foreign Service Summer Research Grant.

1 Blocking in Survey Experiments: An Overview

Pure random assignment guarantees that, on average, the treatment and control groups have identical characteristics. However, this might not be true on any particular random allocation. One solution is to increase the sample size to the point where this risk becomes negligible. Another solution is to achieve *ex ante* balance on key variables, something called blocking[1] (Fisher 1935). Blocks are most often defined using combinations of discrete variables (e.g. partisanship and gender, yielding 6 blocks). Allocation to treatment and control occurs by randomly assigning units within a block. Another approach identifies blocks of J "most similar" units and randomly allocates each unit to one of the J treatment conditions. When $J = 2$, this form of blocked randomization is best known as pair-wise matching. In this memo, I call the former approach 'discrete blocking' and the latter 'continuous blocking'.

Discrete blocking is the most commonly-used blocking procedure. Existing studies most often rely on a few readily available economic and socio-demographic variables. Yet, such pre-treatment covariates are not always very predictive of the outcome of interest. In other words, guaranteeing balance on these covariates might not do that much for inference. One solution is to use as many available variables as possible. With discrete blocking, researchers face "a dimensionality curse" (Moore and Moore 2013): the number of blocks increases exponentially with the number of blocking covariates and the number of covariate levels. This is the point at which researchers might switch to high-dimensional multivariate blocking, i.e. continuous blocking. Another solution is to collect information on variables that are highly predictive of the outcome of interest (e.g. partisanship if the outcome is immigration policy preferences). Ideally, researchers are able to combine both continuous blocking and detailed baseline information on variables predictive of the outcome. Blocking on covariates that do not affect the outcome is discouraged.

Blocking is especially attractive with studies relying on small sample sizes relative to the number of treatment conditions. Yet, even studies with large sample sizes might still benefit from using discrete or continuous blocking. Indeed, blocking minimizes sources of variation that are not related to the experimental manipulation. In other words, blocking maximizes the "signal" relative to the "noise." Blocking appears especially relevant for research on political attitudes and behavior. In this line of work, the study of cognitive mechanisms "in context" (Falleti and Lynch 2009) requires that researchers disentangle the relative contribution of individual factors, contextual factors and their interaction. When the number of factors to manipulate increases, researchers' budgets rarely do. In addition, in many of these studies, the manipulation is brief and, for good reasons, only generates small and temporary effect sizes. When the signal is weak, minimizing noise becomes very important. In this situation, blocking, especially when done on carefully chosen pre-treatment covariates, can help maximize power.

Sample size and power aside, many political science survey experiments would, at a minimum, benefit from blocking on important predictors of political outcomes such as partisanship (Achen and Bartels 2016) or subjective ideology, especially if combined with demographic variables like education or race. Indeed, by

[1] Some readers might have also encountered the term "stratified randomization" or "stratification."

blocking on these variables, researchers can minimize post-hoc covariate adjustments too common in online behavioral experiments, with partisanship being the usual *ex post* go to (Freeder, Lenz and Turney 2018; Gelman and Loken 2014).

Despite the well known benefits of blocking, the following assessment made a decade ago by Horiuchi, Imai and Taniguchi (2007) still holds: “experimental designs used by political scientists remain elementary ... (S)imple randomization of treatments remains the single most commonly used method in the Discipline”. This is especially true of studies using online survey experiment. A (cursory) Google Scholar search covering the 2007-2018 period returned only seven papers using a randomized blocking design for a survey experiment.

Several reasons explain the current situation. First, survey experiments are overwhelmingly run through online recruitment platforms. Most commonly used are crowd-sourcing platforms such as MTurk or research and marketing firms that maintain large online panels. As a result, the cost of collecting survey data is relatively low. When data collection is expensive, such as in a field experiment, researchers often only get one “shot” and are more likely to pay attention to imbalance and power. Because many field experiments are in development economics, there are few opportunities for cross-fertilization across disciplines (i.e. development economics - political behavior) and area studies (i.e. developing - developed countries) despite existing contributions by Horiuchi, Imai and Taniguchi (2007); Imai, King and Stuart (2007); Imai et al. (2009); Moore (2012); Moore and Moore (2013).

Another reason has to do with technical constraints specific to online recruitment platforms. In most online surveys, subjects “trickle” into the survey and are then randomized to treatment conditions. In other words, the researcher only has information on the units that have already arrived and the unit that has just appeared, but not future units. A second obstacle is the need to collect information on participants either *ex ante* or dynamically. Indeed, information on individuals who decide to answer a survey is not always available to researchers. The few studies identified through the above mentioned Google Scholar search rely on designs that do not face such constraints. For example, Nyhan and Reifler (2014) run a survey experiment on a random sample of Congress members, a fixed population, known *a priori*, with baseline characteristics made publicly available through online profiles and media coverage.

Techniques to address these issues include sequential covariate-adaptive randomization, which addresses “trickling in” (Moore and Moore 2013) and partnering with survey firms that have previously collected information on panel members. However, these solutions are not well known to political scientists involved in survey research. This is especially the case for studies focusing on post-industrial democracies. Again, the costs of data collection and the lack of cross-fertilization across experimental fields seem to be the main cause.

Overall, a lack of understanding of the benefits of randomized blocking and how it can be implemented in the context of online survey experiments have limited its use in survey research. The goal of this memo is to address this gap. I describe how to implement discrete and continuous blocking in the context of online survey experiments. All the solutions described below are easily implementable using common tools such as Qualtrics and the R programming language. These solutions require only minimal coding skills thanks to R packages developed by other scholars (Moore and Schnakenberg 2012; Leeper 2015). The memo is written with the most commonly used respondent recruitment platforms (RRP hereafter) in mind (e.g. MTurk, SSI, Prolific, Lucid, YouGov, GfK/IPSOS or NORC). It is also designed to address the common practical considerations rarely discussed in methodological papers regarding when and how to use blocked randomization. Readers interested in the theory behind blocking should refer themselves to the bibliographies in Moore (2012); Horiuchi, Imai and Taniguchi (2007) and Bruhn and McKenzie (2009). Duflo, Glennerster and Kremer (2007) provide a very accessible overview of randomization in experimental research.

Next, the memo describes the main decision parameters and then goes on to describe three ideal-typical cases: one where budget constraint is minimal, the case where budget is limited (especially true in the case of a pilot) and the more common ”somewhere in between” case.

2 General Principles for Implementing Blocking in Online Experiments

Broadly speaking, there are two possible approaches to blocking: either assign treatment “all at once” or do so one unit at a time, based on the moment a respondent “shows up” to take the survey. The first approach I will call ‘immediate’ blocking (A) and the second approach, ‘sequential’ blocking (B) (Moore and Moore 2013).

Immediate blocking requires first generating a target group of respondents, then assigning each individual in this group to treatment conditions *ex ante*, and finally inviting them to take the version of the survey they have been assigned to. The target group can be generated in two ways, either through a first screening survey (A.a) or by partnering with an RPP that relies on a pre-existing (and well maintained) online panel (A.b). In the latter case, the target group is a sample of individuals in the panel pre-identified by the RRP as recipients of the survey mailers. Researchers should try to figure out early on in the process how the RRP samples from the online panel. Based on my own experience, it appears that this target sample is somewhat of a moving target itself, changing dynamically depending on who has already answered the survey. In such a situation, some of the options mentioned below might not be available. Researchers should keep in mind that when negotiating a sales contract, they can potentially ask for this recruitment procedure to be changed to adapt to a blocked randomization procedure. I recommend this be put in writing in the final sales document.

Sequential blocking focuses on individuals who start the survey (minimizing, by design, non-response rate). Assignment will thus be done dynamically as respondents “trickle in.” Sequential blocking roughly takes one of two forms: non-adaptive and covariate-adaptive. In the covariate-adaptive version (B.a) assignment probabilities change based on the covariate profiles of units already in the different treatment groups. In practice, this procedure seeks to create similar covariate distributions in the treatment groups by biasing the current unit’s treatment assignment in favor of treatment condition with fewer units with similar profiles. This approach is especially helpful when researchers are left in the dark regarding what the final sample of respondents will look like. For example, when using a platform like MTurk, research have limited information on whom is going to ultimately take the survey. Instead, researchers only have information on prior units who have already taken the survey and the unit that has just arrived. Again, they have no information on future units.

When researchers have enough information on the (expected) final sample of respondents, then one can implement a non-adaptive sequential blocking procedure using discrete variables (B.b). For example, in the ideal case mentioned below, one can partner with an RRP using an online panel *representative* of the US population. In this case, researchers can implement non-adaptive discrete blocking without having to worry that some blocks might have too few units.[2] Individuals who enter the survey are assigned to a block. Using Qualtrics' branching and complete randomization features, researchers can easily implement complete randomization within a block. Descriptive statistics provided by the firm regarding its online panel and different groups' response rates can further help pick the discrete variables to block on. Note that, by design, non-adaptive blocking using continuous variables is impossible in a sequential set up.

With approach B, information on pre-treatment covariates can be done either by 1) running a screening survey before the final survey, 2) partnering with a survey firms that has already collected information on respondents or 3) using survey time at the beginning of the survey, before treatment assignment, to collect this information. Which combination of B.a - B.b and 1, 2 or 3 researchers end up choosing depends on the RRP they pick, the amount of uncertainty regarding who is going to ultimately take the survey and a few additional technical considerations discussed below.

3 The Ideal Case: Continuous Blocking, Key Baseline Covariates, High Response Rate

Ideally, two conditions are met. First, researchers have rich information on pre-treatment covariates for all the units to be assigned to different treatment conditions. Second, individuals invited to take the survey will both take the survey and finish it. In other words, the response rate will be high and drop out rate low, guaranteeing that the final dataset does not have too many missing values on the outcome of interest. In this ideal situation, researchers can assign units using immediate non-adaptive continuous blocking. The rich information on baseline covariates can also be used to correct for missing data due to non-responses.

This ideal outcome can be achieved when partnering with an RPP that relies on a probability-based panel to recruit respondents (examples of such firms are GfK/IPSOS or NORC). To deliver the number of respondents agreed upon in the contract, these firms randomly select a sample of individuals from their representative online panel. To address non-response and minimize time necessary for data collection, this sample is larger than the final number of respondents agreed upon in the contract. High response rates can mean that the difference between this first target sample and the final sample of respondents can be quite

[2] At the inference stage, researchers need to add block fixed effects. If some blocks do not have enough observations then existing unit will be in practice dropped from the analysis. A randomization procedure that results in discarded data is less than ideal.

small. GfK/IPSOS provides weights to correct for the differences between the first, representative, target sample and the final, no longer representative sample of completed surveys.

Partnering with GfK/IPSOS, the researcher can request, for free, all the demographic covariates of respondents selected in the first target wave. One can then assign *ex ante* each respondent in this first target wave (approach A) to a treatment condition using all this information in the context of immediate non-adaptive continuous blocking. Note that the basic demographic data can be supplemented by additional data available for a fee. This data is very rich. In the case of a recent study (September 2018), my co-authors and I purchased prior information on partisanship, vote and ideology. The partisanship information proved very reliable. Absent any restriction on budget, I highly recommend having a look at these covariates which cover not only politics but also consumption, healthcare and many other topics.

Once assignment is done, the researcher can share with GfK/IPSOS a file with at least two columns: respondents' unique identifier and the treatment condition they have been assigned to. The firm will then share this information back with researchers by embedding it in the URL respondents click on to take the survey. Qualtrics has simple built-in features that enable to record the treatment value and then present the respondent with the version of the survey he or she has been assigned to.

I consider this solution "ideal" because researchers have a lot of baseline information and are relying on it to minimize variance that was not induced by the manipulation. The assignment procedure is under the researcher's full control. He or she has time to implement a computationally-demanding randomization procedure and can even double check that balance was indeed achieved relying on both blocking covariates (balance should be achieved by design) and non-blocking covariates. Yet, I would like to highlight two concerns to keep in mind.

First, this solution is ideal when response rates are high, namely when the first target sample does not differ too much from the final sample. In my own experience, there is some uncertainty surrounding the final response rate. GfK/IPSOS has rich information on past response rates so should be able to provide support on this issue.[3] In our own recent experience, GfK/IPSOS proved very conservative in its estimate. For a target sample of 4,000, GfK/IPSOS contacted 7,800 individuals (a response rate of a little over 50%). They estimated a survey completion time of 10 to 14 days. In practice it took only 78 hours to reach 4,000, indicating that the sample size/survey completion time trade-off could have been addressed somewhat differently.

The second issue has to do with the implementation of immediate continuous blocking when there is uncertainty about completion rates. Indeed, take the simple case with one control group and one treatment group: any time one respondent in any given pair does not take the survey, the data collected on the other matching respondent needs to be dropped.[4] To minimize the number of missing observations, one can

[3] I highly recommend to get an engineer on the phone as sales personnel is not always very well versed in the concept if blocked randomization.

[4] Indeed, the final analysis ought to include fixed effects for each matched pair. If one of the unit is missing then there is no variance to rely on for inference.

increase the number of time the survey is ”in the field” (and decrease the size of the first target group accordingly). If researchers are measuring outcomes sensitive to the political context, then too much time spent collecting the data can introduce heterogeneity that is not always easy to model.

To address this issue, one can opt for immediate discrete blocking (approach A.b). Blocking on discrete covariates, and thus using fewer covariates, need not be a concern if one is blocking using important covariates highly predictive of the outcome of interest. Researchers might also consider approach B.a, namely sequential covariate-adaptive continuous blocking.[5] The latter case, however, faces possible technical hurdles that we discuss in the next section.

4 The Most Useful Case: Sequential Continuous Blocking for Small Pilots

The benefits of blocking are especially large for studies with small sample sizes. This is usually the case at the pilot stage. Most researchers running pilots use flexible and cheap platforms such as MTurk. This section is written with such platform in mind.

Run a Screening Survey and Use Immediate Blocking

One solution is for researchers to first run a screening survey to collect information on baseline covariates. In a second step, researchers identify a sample of respondents from this first wave to re-contact. Using the information from the screening survey, they then assign individuals in this target group to treatment conditions using immediate discrete or continuous blocking. These individuals can be easily recontacted “en mass” using Thomas Leeper’s `TurkeR` R package (Leeper 2015). In my experience, if one focuses on Turkers who have accomplished a lot of HITs, response rates are surprisingly very high (over 70 % , even after 3 months). Yet given uncertainty on response rates and the cost of dropping observations in a small sample, discrete blocking on carefully chosen pre-treatment covariates appears the safest bet.

A Sequential Covariate-Adaptive Randomization design

A second solution is for researchers to implement a covariate-adaptive design. First, researchers collect information on key covariates either through a prior screening survey or at the beginning of the survey.[6] Then, using this information, researchers can rely on Qualtric’s Web Service feature to run a covariate-adaptive assignment algorithm remotely using an API (more below). One useful algorithm is the `seqblock` function available in the `blocktools` R package developed by Moore and Schnakenberg (2012).

For each respondent that “enters” the survey, this R function creates “a measure of multivariate distance

[5] Indeed, GfK/IPSOS can similarly “pass on”, embedded in the URL, information on multiple baseline covariates.

[6] Researchers might already have this information through a prior survey run on MTurk.

between a current unit and one or multiple prior, already-assigned unit(s). Then, average distance between current unit and each treatment condition is calculated, and random assignment is biased toward conditions more dissimilar to the current unit" (Moore and Schnakenberg 2012: 20). This can be done with both discrete variables and continuous variables.

Yet, researchers should be aware that assignment time increases for each additional entrants. This will not be a concern for pilots, which usually have less than 500 respondents to assign. For researchers who might be interested in using covariate-adaptive randomization for large surveys, I have estimated that in a survey of 4,000 respondents, the 4,000th respondent takes 5 seconds on average to assign. This estimation was done by running `seqblock` on a Macbook Pro and might not be fully reflective of how the algorithm runs online.[7] To maintain an homogeneous survey experience across survey respondents, I recommend using `seqblock` for small pilot surveys of less than 1000 respondents. I also recommend doing test runs to check what the experience of a new survey taker might be conditional on the number of existing survey responses already provided (described below)

For readers interested in this solution, Diag Davenport[8] describes below how to deploy the API and embed it into one's own qualtric survey.

Github and Herokou In order to edit and use the API's software, it is strongly recommended that researchers use Github and Herokou to deploy the application:

- What is Github? Github is a hosting service for open-source software projects. It enables storing, editing, and sharing of computer code using a file storage system called "repositories".
- What is Heroku? Heroku is a platform that allows users to run computer code in the cloud, instead of on their personal machines, which enables rapid scaling (up and down) of computing resources in order to meet demand.

The first step is to create a Github account. Go to github.com and follow the instructions there. Once an account is made,'fork" the API's repository (`https://github.com/diagdavenport/rpy2-heroku`) to create a copy under your own account, i.e. under your control. Now you can edit the code base in this newly created repository. This code base will be the source code for your API. The file you want to pay attention to here is called `diag.py`.

Next you will need to set up resources in order to have the API run in the cloud and be available on the fly:

- Go to heroku.com and create an account.

[7] I would like to thank Rich Nielsen for helping me on this issue

[8] https://diagdavenport.com/

- Click the icon in the top right corner to access “Account settings”.
- In “Account settings” click the “Applications” tab.
- Under ”Third-party services”, connect Heroku to your Github account.
- Go back to the homepage (by clicking the Heroku logo on the top left corner) and create a new app by click ”New” towards the top right.
- Select ”Create new app”
- After specifying an available name, under ”Deployment method”, select Github and link the app to the forked repository in Github. This step sets up your API to update and deploy whenever new code is ”pushed” to the Github repository.
- From here forward, you can check if the app is running, updating, or failing under the Activity tab.

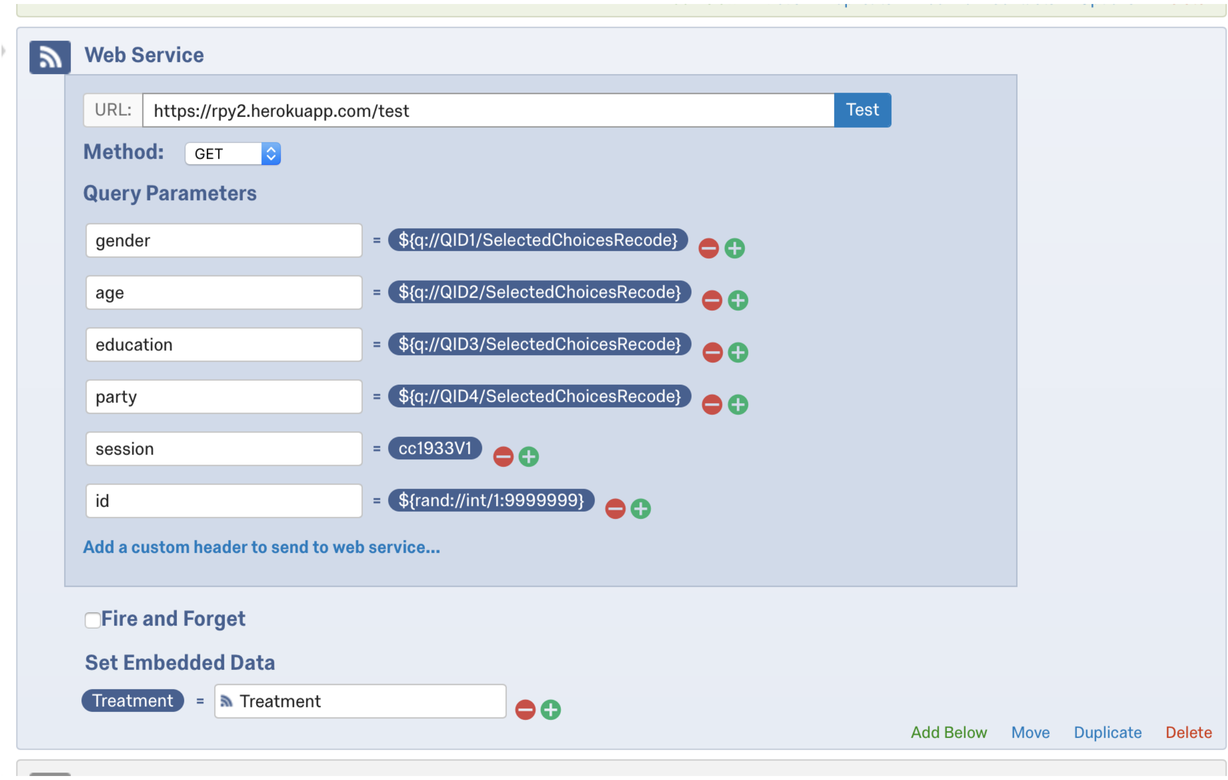
The API code is a mix of Python language and the R code necessary to run the `seqblock` function. The information for the `seqblock` function is available in the manual for the R package `blocktools`.

Qualtrics Web Interface Figure 1 provides a screen shot of a web interface set up to implement blocking on four discrete variables. Information was collected at the beginning of the survey (QID1 through QID4). The embedded variable used to record the assigned treatment condition is called `treatment`. Additional variables are used to record the user identifier (`id`) and the name of the survey wave (`session`). The name of the session should be changed at the beginning of each data collection. It is used to identify a respondent as being the first to take the survey. Make sure to also set the method to `GET`.

Figure 2 provides a screen shot of the baseline code, which will need to be modified depending on the blocking variables research plan on using (please see the `blocktools` user manual).[9] The first block of code captures the variables recorded at the beginning of the survey that will be subsequently used for blocking. The second block of code defines the vectors to input into the `seqblock` function. One vector records the names of the blocking variables and another records the values assigned to a given respondent for each variable. Vectors differ depending on whether variables are discrete or continuous (see manual). Here the example only shows blocking on discrete variables. Note the condition on line 34: in this example, partisanship is coded in Qualtrics to vary from 11 to 37, i.e. the number of characters is equal to 2. When this condition is not met, this means that there are no recorded value for a given respondent and an error message will be recorded. One should consider carefully how to deal with respondents with missing values on key blocking variables.

Assuming the key partisanship variable is not empty, then the function ‘f’ runs. The first step in the function examines whether the value recorded in `session` has already been previously recorded. If it has

[9] Notice towards the bottom, the additional path added ”/test”, which matches the url in the web interface: “https://rpy2.herokuapp.com/test.”



Web Service

URL: https://rpy2.herokuapp.com/test

Method: GET

Query Parameters

gender = ${q://QID1/SelectedChoicesRecode}

age = ${q://QID2/SelectedChoicesRecode}

education = ${q://QID3/SelectedChoicesRecode}

party = ${q://QID4/SelectedChoicesRecode}

session = cc1933V1

id = ${rand://int/1:9999999}

☐ Fire and Forget

Set Embedded Data

Treatment = Treatment

Figure 1: Qualtrics Web Interface to Run `blocktools`

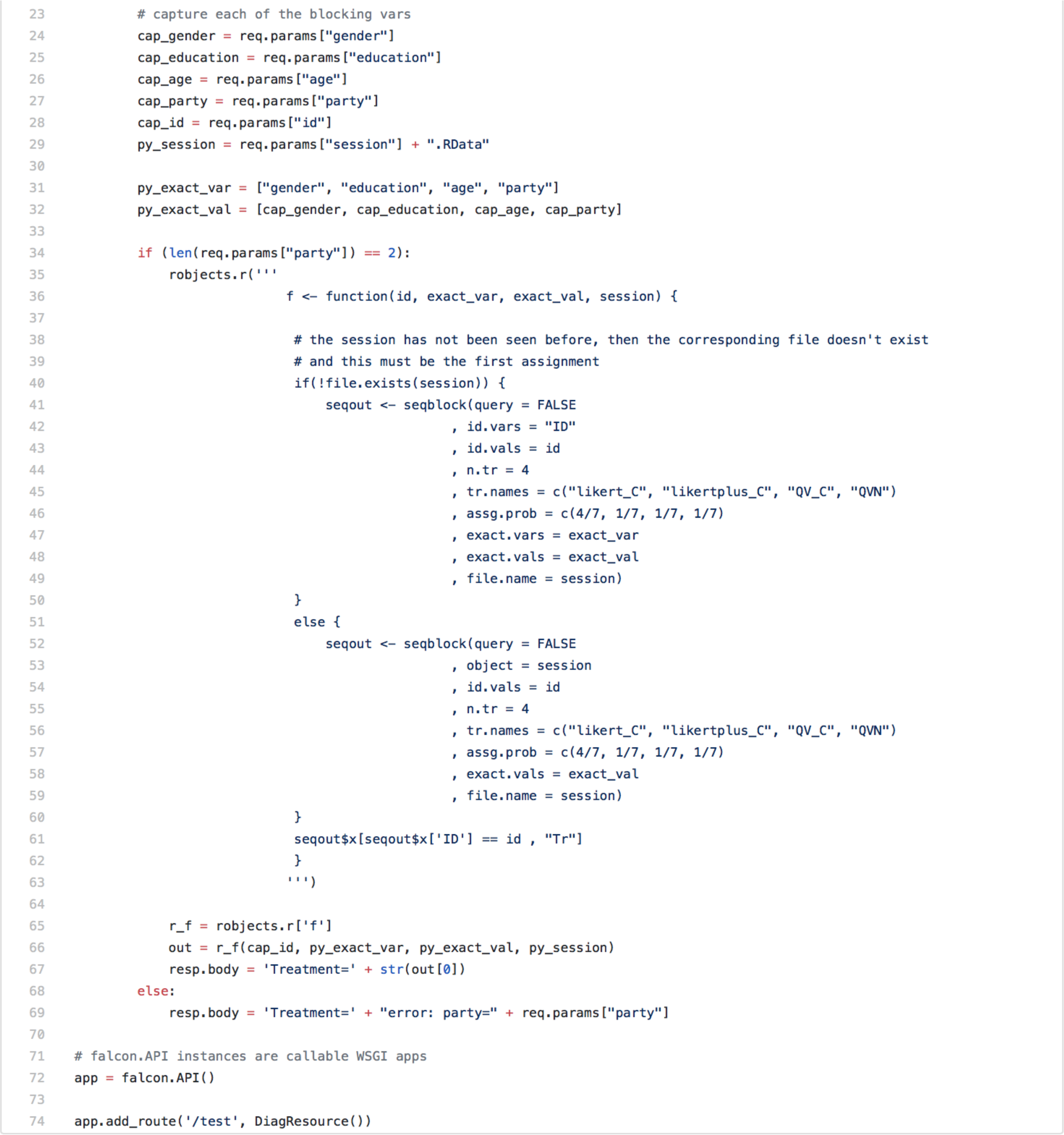
not then this means that this observation is the first one to "show up." This first observation is allocated to a treatment condition using a separate algorithm. For subsequent observations, the allocation will depend on prior units in the survey, this is accomplished by the second step in the code, lines 52-59. Note that in this example, there are four treatment groups, with different assignment probabilities.

As noted above, when the survey is large, the algorithm might take a few seconds to assign new units. To examine how the experience of a new survey respondent varies with the number of previously recorded responses, one can use the "test responses" feature in Qualtrics. This feature allows researchers to generate a large number of responses (wargning: researchers should make sure to set a new session name). Once responses have been generated, one can take the survey to see how long it takes for a unit to be assigned.

Beyond MTurk

MTurk, while flexible, is no longer as attractive with regards to price.[10] Other alternatives are firms like SSI or Lucid, which maintain an online panel of respondents interested in taking surveys. These firms usually guarantee a final sample which match the US population on basic demographics. These firms also have previously collected baseline information on its respondents.

[10] This is especially true if researchers follow minimum wage guidelines, which is unfortunately not always the case.



```
        # capture each of the blocking vars
        cap_gender = req.params["gender"]
        cap_education = req.params["education"]
        cap_age = req.params["age"]
        cap_party = req.params["party"]
        cap_id = req.params["id"]
        py_session = req.params["session"] + ".RData"

        py_exact_var = ["gender", "education", "age", "party"]
        py_exact_val = [cap_gender, cap_education, cap_age, cap_party]

        if (len(req.params["party"]) == 2):
            robjects.r('''
                           f <- function(id, exact_var, exact_val, session) {

                            # the session has not been seen before, then the corresponding file doesn't exist
                            # and this must be the first assignment
                            if(!file.exists(session)) {
                                seqout <- seqblock(query = FALSE
                                                , id.vars = "ID"
                                                , id.vals = id
                                                , n.tr = 4
                                                , tr.names = c("likert_C", "likertplus_C", "QV_C", "QVN")
                                                , assg.prob = c(4/7, 1/7, 1/7, 1/7)
                                                , exact.vars = exact_var
                                                , exact.vals = exact_val
                                                , file.name = session)
                            }
                            else {
                                seqout <- seqblock(query = FALSE
                                                , object = session
                                                , id.vals = id
                                                , n.tr = 4
                                                , tr.names = c("likert_C", "likertplus_C", "QV_C", "QVN")
                                                , assg.prob = c(4/7, 1/7, 1/7, 1/7)
                                                , exact.vals = exact_val
                                                , file.name = session)
                            }
                            seqout$x[seqout$x['ID'] == id , "Tr"]
                            }
                           ''')

            r_f = robjects.r['f']
            out = r_f(cap_id, py_exact_var, py_exact_val, py_session)
            resp.body = 'Treatment=' + str(out[0])
        else:
            resp.body = 'Treatment=' + "error: party=" + req.params["party"]

# falcon.API instances are callable WSGI apps
app = falcon.API()

app.add_route('/test', DiagResource())
```

Figure 2: Python Code for the API using `blocktools`

Source: Diag Davenport

Depending on how the firm generates the final sample of respondents, there are several possible ways to implement blocking. I recommend figuring out early on in the process how exactly final sample of completed survey responses are generated. The best solution is often to request talking to one of the panel engineers before discussing a quote with the salesperson. This will save everyone time. In my experience, doing so before the contract is signed requires polite insistence.

My understanding is that the sampling procedures used can sometimes make it risky to implement immediate blocked randomization (i.e. approach A.a and A.b described above). Indeed, it is unclear how cooperative the firm will be in releasing prior information on individuals who will be contacted to answer the survey. In addition, it appears that many more individuals are contacted than ultimately answer the survey. Additional procedures to make sure the final sample matches the census on key covariates can introduce additional uncertainty (the target sample changes over time). Provided sample sizes are not too large, one can rely on the sequential covariate-adaptive blocking procedure described above (B.a). If researchers are concerned with possible issues regarding the smooth running of the API, then the solution described in the next section might be a safer option.

5 The Most-Likely Case: Discrete Blocking, Key Baseline Covariates, Low Response Rate

Most researchers are likely to use the cheaper marketing panels to collect data on a large number of respondents. I recommend collecting information on pre-treatment variables highly predictive of the outcome of interest. This can be done at the beginning of the survey. If collecting this information functions as a treatment in itself, then researchers should consider the covariates the RRPs have already collected and check whether these variables can be shared dynamically with the Qualtrics platform. Researchers should make sure to inquire about the reliability of this data, especially the number of missing values. After running a few tests, researchers can opt for the sequential co-variate adaptive procedure described above.

Approach B.b, i.e. sequential non-adaptive blocking using discrete variables is less technologically demanding and can accomplish wonders with the right covariates. Researchers should always request descriptive statistics about the panel and response rates (e.g. blocking on ethnicity if the panel has no Latino respondents might not be very useful). The general principle of sequential non-adaptive blocking using discrete variables is as follow:

Step 1: Identify the block to which a new respondent starting the survey belongs.

Step 2: Generate a branch for each block using Qualtrics branching feature.

Step 3: Within each branch/block implement **complete** randomization (tick the ”Evenly Present Elements” box).

When the number of blocks is large, the branching for each block can be a little overwhelming to do manually. Qualtrics’ engineers can take care of this if researchers uses Qualtrics as an RRP. Step 1 is where there is some variation:

- (V1) Some RRP generate samples using something akin to blocks (also called “stratas”). If the firm has the capabilities to share a variable identifying these stratas, then researchers can, at a minimum, implement complete randomization within a strata, hopefully at no additional cost. Again, I recommend talking to a technician to better understand what these stratas are and whether they can help with balancing.

- (V2) As already mentioned, many of these firms also provide additional covariates for a fee. If a researcher wants more specific blocks than broad demographic ones, then he or she can buy such covariates and partner with the RRP for one of the following two solutions:
  - Ask the firm to sample respondents within blocks and pass on a variable that records the block to which a respondent belongs.
  - Ask the firm to pass on the values of the blocking variables and use Qualtrics embedded data features to identify a respondent's block.
- (V3) Another solution is to start the survey with questions designed to collect information on the blocking variables, and then use Qualtrics embedded data features to identify a respondent's block.

References

Achen, Christopher H and Larry M Bartels. 2016. *Democracy for realists: Why elections do not produce responsive government*. Princeton University Press.

Bruhn, Miriam and David McKenzie. 2009. "In pursuit of balance: Randomization in practice in development field experiments." *American economic journal: applied economics* 1(4):200–232.

Duflo, Esther, Rachel Glennerster and Michael Kremer. 2007. "Using randomization in development economics research: A toolkit." *Handbook of development economics* 4:3895–3962.

Falleti, Tulia G and Julia F Lynch. 2009. "Context and causal mechanisms in political analysis." *Comparative Political Studies* 42(9):1143–1166.

Fisher, Ronald Aylmer. 1935. "The design of experiments.".

Freeder, Sean, Gabriel S Lenz and Shad Turney. 2018. "The Importance of Knowing 'What Goes With What'." *Journal of Politics* .

Gelman, Andrew and Eric Loken. 2014. "The statistical crisis in science." *American scientist* 102(6):460.

Horiuchi, Yusaku, Kosuke Imai and Naoko Taniguchi. 2007. "Designing and analyzing randomized experiments: Application to a Japanese election survey experiment." *American Journal of Political Science* 51(3):669–687.

Imai, Kosuke, Gary King, Clayton Nall et al. 2009. "The essential role of pair matching in cluster-randomized experiments, with application to the Mexican universal health insurance evaluation." *Statistical Science* 24(1):29–53.

Imai, Kosuke, Gary King and Elizabeth A Stuart. 2007. "Misunderstandings among Experimentalists and Observationalists about Causal Inference1.".

Leeper, Thomas J. 2015. "MTurkR: Access to Amazon Mechanical Turk Requester API via R." *R package version 0* 6(5.1).

Moore, Ryan T. 2012. "Multivariate continuous blocking to improve political science experiments." *Political Analysis* 20(4):460–479.

Moore, Ryan T and K Schnakenberg. 2012. "blockTools: Blocking, Assignment, and Diagnosing Interference in Randomized Experiments. R package version 0.5-7.".

Moore, Ryan T and Sally A Moore. 2013. "Blocking for sequential political experiments." *Political Analysis* 21(4):507–523.

Nyhan, Brendan and Jason Reifler. 2014. "The Effect of Fact-Checking on Elites: A Field Experiment on U.S. State Legislators." *American Journal of Political Science,* 59(3):628–640.