To: Incoming Pre-Docs

From: Alvin Christian

Date: 06/18/2019

RE: Pre-Doctoral Memo

Summary

This memo provides a brief overview of the typical pre-doctoral full-time research assistant (RA) position in the social sciences. It provides information on what to expect, how to be a successful RA, and how to get the most out of this position. It is primarily aimed towards those coming straight out of undergrad or a master's program, but working professionals transitioning over to such a position may also find it helpful.

What is a pre-doc and why should I do one?

A pre-doctoral position is a 1-2 year position where you work closely with a professor or research team as a full-time research assistant. The position is an increasingly important steppingstone to grad school. It is targeted at those coming straight from undergrad/a master's program and for young professionals who've been working for a few years and want to transition to academia. It is a great opportunitiy to make yourself more attractive to grad schools, gain valuable skills through mentorship, and receive a recommendation letter from someone in your chosen field.[1]

What to expect?

Experiences will vary by institutions and professors. However, the main function remains the same—your job is to help your boss publish journal articles.

Your day to day will vary depending on the projects you're working on and on what stage of the research process they're in. Unlike projects in school that last a term or two max, a lightning fast research project for professional academics probably won't be any shorter than a year!

Expect most of your time to be spent on data cleaning, processing, and analyzing.[2] For projects still in their early stages, you may find yourself doing a lot of reading and writing—you'll read papers to familiarize yourself with the field and write research proposals, analysis plans, or literature reviews. You may also find yourself conducting focus groups and fieldwork, designing and administering surveys, managing undergrad research assistants, and even drafting articles. Even if you don't end up pursuing academia you will gain a breadth of experience that will be valuable in various professions.

[1] Although a recommendation letter is an end-goal, you should not do one of these positions just for the sake of getting a recommendation letter. Professors will be less likely to work with you if they think you only want a recommendation letter. I'd recommend approaching these positions through the apprenticeship lens—you're there to learn how to conduct high-quality research from an expert in your field of interest.

[2] The program used is going to depend on your boss, but Stata is the most widely used.

How to be successful on the job

Many students come into these positions straight out of undergrad and enter with a student mindset. That is a mistake! Do not approach this position like a student—you're an employee and a professional, so act like it. Here are some tips to help with the transition from student to professional.

Manage up

This refers to managing your manager. Your boss will be busy and have limited time to spend with you—don't take this personally. Your goal should be to make their job easier. You do this by anticipating their needs. Before I create any deliverable for my boss, I ask myself the following questions:

- What are they asking for?
- What do they know?
- What do they need to know?

Communicate often and effectively

Communication is key in any job. Your boss is likely juggling a handful of projects, RAs, co-authors, children, etc. Help them stay above water by letting them know what you've been doing, what you're currently doing, and what you will be doing.

I recommend having a weekly check-in with your boss. Block off 30-60 minutes a week to chat with your boss to discuss the work you've been doing, the work you want to be doing, questions you have, and grad school related stuff. Some tips to go along with that:

- Don't go into check-ins unprepared. Create a brief agenda of things to talk about and email it to your boss ahead of the meeting. This will structure your conversation and keep it focused. It will also give your boss an opportunity to prepare.
- If your boss emails you to do something, reply with a quick "on it". This small but important practice that lets them know you're doing what they're asking. They will appreciate this.
- Don't bury the lede. Relay the most important information as quickly and succinctly as possible. Do this when talking in person, over the phone, in email, or with any memos or write-ups. For memos, I like to start with a *summary* section that summarizes the entire memo in a few sentences.
- Ask questions! If you don't understand something or if you're not sure how to do something, just ask. You'll save everyone time in the long-run, not to mention these positions are typically offered with the idea of RA growth in mind.

Be brief

Communication is important but how do you best communicate? As with writing, brevity is everything.

- Be brief over email. If you can't keep your email short and focused, it means you and your boss should be discussing the issue over the phone or in person.

- Keep memos and write-ups as lean as possible. Exclude any information from the body that isn't necessary. You can put this information in footnotes or an appendix that your boss can refer to if they're interested. When possible, provide tables and figures because they're easier to look at.

Be organized and learn time management

Your success as an RA is almost wholly dependent on your time management and organizational skills. It's likely that you'll work on multiple projects at different development stages. It's imperative you learn how to divvy up your time and stay organized.

- If you don't know what you should be prioritizing, ask your boss. After being assigned tasks, I like to confirm how they should be prioritized. *So it looks like you want me to do X, Y, and Z. Since Z is time sensitive, I'll focus on that before tackling X and Y. Let me know if that doesn't seem right to you. Thanks!*

- Stay organized—I work with two screens and on my second screen I like to keep a To Do list open that is organized by project that I regularly update. I also like to list important deadlines in Google calendar. Some people like to block out sections of their day based on their tasks. *Between 9-12 I'll clean data for project A and between 1-4 I'll work on the figures for project B*.

- Organize files and folders for different projects in a logical and parallel way. I like to create a folder for each project and the folders inside each project are always the same—I include a folder for documents, do files, data, and output. I even further subdivide some of these folders by data source. Doing this for all my projects keeps everything organized and allows me to quickly find important files. Similarly, make sure to name all files intuitively. *Will someone with no project knowledge be able to find what they're looking for?*

Documentation is key

Document everything. You will easily get lost in 1,000 lines of code and forget what you were doing a week ago. It's imperative that you leave a paper trail, both in your code and elsewhere.

- I create data memos for each data source I work with. Memos include information about where the data is collected, any data issues, and information on variable creation. In particular, I emphasize documenting missing rules and sample definitions as I find that's where people make the most mistakes.

Coding tips

Jesse Shapiro and Matthew Gentzkow have developed a superb guide for coding that everyone should follow, so I won't dwell on this section. At the end of the day, everyone has their own preferences, so I would closely follow the coding style of your boss. They're comfortable with their rules and they work for them. With that said, here are some tips that I find useful.

Documentation

As noted above, I heavily document my code. I usually create a document block at the beginning of my code with information on the program, the purpose of the file, an outline of what I'm doing, and important notes. In the actual code itself, I write notes detailing exactly what I'm doing and why, making references to any relevant dates, meetings, or emails. I'd say 30-40% of my code is just documentation (this is probably overkill for most people but works for me.).

Consistent names

Create simple, parallel, and intuitive naming structures for files and variables. It doesn't matter how you do it as long as it's consistent. For example, I name my variables like this: *dataset prefix_variable name* (e.g., the variable Y4CA_M_StdScore = year 4 child assessment math standard score). This is helpful when you're merging data as you'll always know the data source of each variable. When it comes to naming things, just be intuitive and consistent.

Sample

One thing that I think trips people up, especially newcomers, is defining your sample. It's extremely important you are explicit and understand exactly who is in your sample what your sample size should be. For example, it's easy to say your sample is "new teachers in Tennessee." But how is it defined? Do you include teachers with experience from out of state that are new to the district? Is the sample flag you're using capturing every possible case? Are there weird cases involving missing values? Know your sample inside and out; in particular, make sure for any output you create that the sample size makes sense (e.g., if the sample size for first year teachers is larger than experienced teachers, you might be doing something wrong).

Flexible code

One common pitfall I see many people running into is writing code for some sort of output that they think is one-off. That is almost never the case. Your boss will ask you to run the same set of analyses or create the same output and figures for various subgroups or samples. I recommend when you first write code for any sort of analysis to make it as flexible as possible.

- Use macros—it's likely that any regression you do will use some base set of control variables. Create a global macro for these variables, so when running regressions you can easily update all regressions by just updating the global macro.

- Use loops whenever possible. They save time in the long-run and are easier to debug.

- I've recently started writing programs in Stata and find they save me a lot of time in the long-run.

How to get the most out of your RAship

This position is more a mentorship position than a job. You should think about how you can take advantage of the resources, time, and people available to you during your stint as a RA.

Develop new skills

Use the position to do things you haven't done before. This position will be a great opportunity for you to get applied experience doing a variety of things that most won't do until grad school. This includes learning and applying fancy methods, learning how to use different programs, getting your hands dirty doing fieldwork, etc. Take advantage of the position and use it to build human capital.

Your job is going to be much more flexible than your typical 9 to 5, so take advantage of that flexibility and take any additional courses you need to be a competitive grad school applicant. If you're working at an university, it's likely they will waive most if not all class fees as an employee.

Network

If you're working at a university make an effort to meet with people you don't directly work with. There will be tons of people inside and outside your field who are doing amazing work. Either ask your boss to introduce you or simply cold email them explaining you are interested in their work and would love to grab a coffee to hear about their work. In my experience, academics are generally pretty down to earth and would love to chat with you about their background and experiences. I've learned so much about my research field, grad school, and more during just an hour-long coffee conversation.

Write

Writing is an academic's bread and butter. This is a skill that takes time to cultivate and you improve only by writing, editing, and writing some more. I'd recommend anyone in a pre-doc position to get as much writing experience on the job as possible. The more writing you do on the job, the better prepared you will be for grad school and beyond. Let your boss know that this is something you want to develop. It shouldn't be hard to convince them to pass on some of the writing responsibilities to you. For example, it's easy to make the case for you to draft the data or methods section of a paper because these sections are formulaic and because you probably know the intricacies of the data better than them.

Work on project management skills

Grad school is great at teaching someone to be methodologically rigorous but doesn't teach one how to manage large projects. Observe how your boss organizes and manages their research projects. Your boss may be great at this, in which case learn from them and try to emulate them. If your boss isn't great at this, try to understand why. It's possible that how your boss manages their project works for them, but it may not work for you. That's ok—take time to reflect on how you think and what you need to do to keep yourself motivated and organized.

Explore and discover your research interests

The purpose of this job is twofold: 1) determine if grad school is right for you and 2) determine which grad program is right for you. Take time to explore your research interests—talk to people in different fields, read diverse papers, and attend various seminars. What you wanted to study as an undergrad may not be great for you now—and that's ok! I thought I wanted to do health research as an undergrad, but decided to focus on education because I found it more fulfilling and a better fit.[3] Reflect on why you are at this job and why you are studying what you're studying.

Miscellaneous

Don't overwork/over-commit and work for free

Remember, you are not a student—you are an employee and should not work for free. If something is taking longer than expected, let your boss know and work with them to develop a plan of action. You should not work 60 hours/week if you're only being paid for 40 hours/week—it's not fair to yourself.

Be respectful and kind

There isn't really more to this. The people you're working with now are likely the same people you'll be working with the rest of your career. It will go a long way if when people think about you they recall you as nice and easy to be around.

[3] In addition to being brilliant, the education researchers I've met are some of the kindest people I know—would recommend joining the field.

Have fun

You’re at a unique point in life with this position. You are no longer a poor college student and will have disposable income. You’re also not a grad student with obligations to do problems sets or readings. Take this time to do the things that you enjoy—exercise, travel, and find fun hobbies. For example, I throw most of my money and time away on Yeezys and Tinder dates.[4] You may not have the income or time to do some of these things as a grad student. Enjoy yourself!

[4] Don’t swipe right on colleagues; that’s not cool.