Learn how to get a 6-figure job in data science and analytics

This document is [somewhat] short, to the point, and effective.
When you're finished with it you WILL be more competitive for high-paying data jobs.

Hello fellow job seekers!

I'm happy you made it. My name is Peter. About a year ago, I was in your shoes. I wanted one of those high-paying jobs in data science and analytics, but after sending application after application, I wasn't getting that "you're hired" call I wanted. So I changed my tactics, and within weeks I had multiple offers from some of the biggest tech companies in the world.

You're maybe thinking, "Well, Peter probably had way more experience than me, or some fancy connections, etc., etc." That's not true. When I started applying, I was less than a year out of undergrad, had zero connections, and didn't really know what data science was. Instead, I figured out how to present the best possible version of myself to recruiters and hiring managers. Now, I get to answer interesting and challenging business questions with data every day.

Through my own application experience as well as evaluating and interviewing applicants interested in joining my company's data science and analytics team, I've learned two, big, BIG things:

1. The data science and analytics field is broken and confused. Since this is still a new and emerging field, recruiters and hiring managers are still trying to figure out what to look for and how to hire for these roles—if you are able to present yourself as a technically-competent worker who can clearly present results to non-technical people, you will be more competitive than 90% of applicants
2. Recruiters and hiring managers aren't even sure what to call these positions—most positions are some variation of data analyst, product analyst, data scientist, research analyst, quantitative analyst, marketing analyst, quantitative scientist, etc., etc. Read the descriptions—but don't completely trust them. I ended up with a job offer for a position that was looking for someone with a master's degree and 4+ years of experience. I had neither of those things. Sure, some positions will only consider advanced degrees, or PhDs, or 5+ years of experience—but still apply even if you don't check off all the boxes the company is looking for

Ready to get started? I've broken this guide into the following sections:

5. What is the career trajectory for someone in data science and analytics?
6. Still want more help? Contact me!

What is a data analyst/scientist? What skills do you need?

There are a lot of resources on the Internet explaining the difference between a data analyst and a data scientist. In my experience, there is a lot of overlap between the two—but usually, a data analyst job has fewer programming/statistics requirements than a data scientist job. Note: I said USUALLY.

In order to be competitive for the average data analyst/scientist job, you need the following skills and you need to be able to demonstrate these skills throughout the interview process:

- **SQL ability**—if you know zero SQL, or need a refresher, follow this *FREE* SQL tutorial on Mode Analytics. I knew *ZERO* SQL before I started applying; if you follow this tutorial (you can easily finish it in 3-4 days) you will be able to comfortably answer even the most challenging SQL interview questions
- **Python/R skills**—again, Mode Analytics has a fantastic Python tutorial. Other than this tutorial, I recommend you try your own programming projects—find a public dataset and get to work! In the next section, I'll explain how to effectively start and showcase your personal programming projects
- **Analytical thinking ability**—yes, you guessed it. Mode Analytics has great questions for improving your analytical thinking ability. During my application process I was asked variations of the problems from this tutorial. One of the biggest issues I see when interviewing applicants is that they have strong technical skills, but very little business sense. Spend time practicing how to think through and discuss business questions, and you'll stand out in the interview process. Now, I won't mention which company--but one of the Silicon Valley tech giants sends this tutorial to their applicants before their onsite interview as a friendly reminder for what to prepare for

You may have noticed that I only recommended one tutorial for each type of skill. **This document is not meant to be an exhaustive list for the "best programming courses to get hired in data science."** You can check out this link for additional course recommendations. Obviously, one course is not enough to make you a programming expert. But let me say this: find your own data, struggle with what questions you want to answer, get frustrated and infuriated while combing through StackOverflow, quit for awhile and feel sorry for yourself and binge-watch Bob's Burgers, and come back and figure out the error in your code—that's the best way to learn programming and data science!

What you need to do before you apply

This section is all about how to make your application stand out to the recruiters sifting through the piles and piles of resumes and cover letters and applications. **Want to know a hard truth?** Your resume probably needs work. As does your cover letter. As does the rest of your application. But the good news? So do everyone else's. Follow these steps, and your application WILL stand out. This section is broken into the following:

1. GitHub advice + personal projects template
2. Resume advice + resume template
3. Cover letter advice + cover letter template
4. Website advice + website template

GitHub advice + personal projects template

Already have a GitHub? Awesome. Just pin the repos you want people to see and add clear and concise READMEs that explain what the project is. Don't have a GitHub? Confused what "pin the repo" means? Create a GitHub account, read this introduction, then come back here. I'll wait.

Okay. So now you have a GitHub account, and are at least familiar with it. GitHub is a fantastic place to demonstrate your programming ability to hiring managers. But some of you are probably worried and saying to yourself, "Oh no, I don't have any work to show," or "All the coding I know is from a Mode Analytics tutorial" or "My code is messy and I would be embarrassed for companies to see it."

But worry not! Think about a data/programming challenge that interests you, write some code, and display it on GitHub. Not sure what types of projects to work on? These are examples of some projects I worked on to populate my GitHub account:

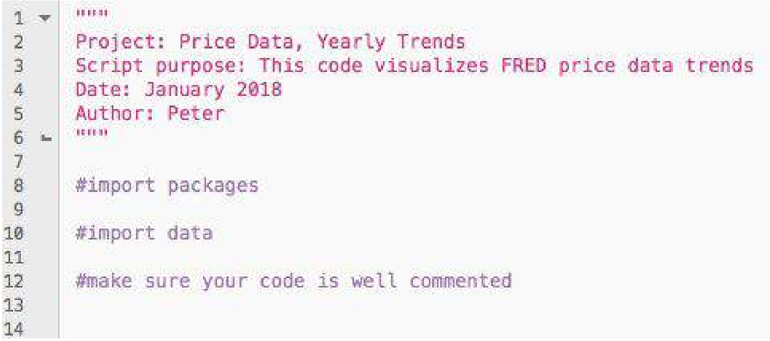
- Built a web scraping tool in Python using the BeautifulSoup package to retrieve NBA player statistics
- Experimented with R's ggplot2 to create aesthetically pleasing charts using data from the Federal Reserve's Economic Database

Now, do the following:

1. Literally just Google "free public data" and find some data that you would be interested in working with—and figure out what question you want to answer or what visualization you want to create
2. Start coding! But make sure your code is clean and well commented—see the screenshot after this list for a clear way to write a description for your code. If

you’re worried about whether your code looks messy, then contact me! I offer one-on-one help and my contact info is at the end of this guide

3. Upload your code to GitHub and make sure to add a README file explaining your motivation and what your project is
4. Save a copy of the chart/results you make (later in this guide I’ll discuss how to present your personal projects on your website)
5. Be prepared to discuss your projects during your interviews! On two separate occasions, I had interviewers bring up my personal projects after viewing them on GitHub and my website. One interviewer commented on how I “clearly had the technical skills needed” and another interview panel was interested in having me explain my price index project because they were impressed by the visualizations I constructed. Moral of the story: companies look at (and are impressed) when you have a portfolio to show. Not only does it look good, but it gives you an opportunity to practice coding and have a project to discuss when you’re interviewing—which can be especially helpful for some applicants who have little/no work experience



```
"""
Project: Price Data, Yearly Trends
Script purpose: This code visualizes FRED price data trends
Date: January 2018
Author: Peter
"""

#import packages

#import data

#make sure your code is well commented


```

Resume advice + resume template

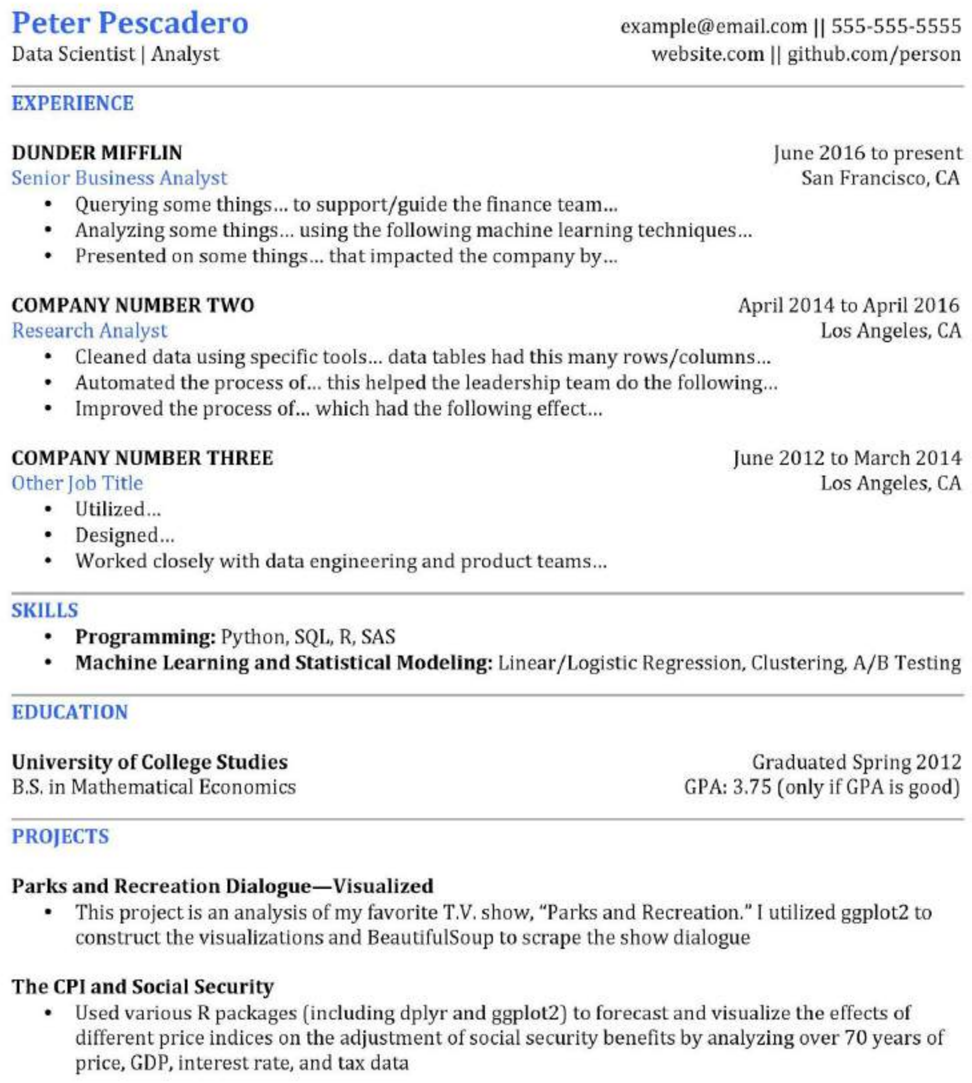
In this section I’ll discuss things to include and avoid in your resume, a template for building an effective/informative/eye-catching resume, as well as a link to a Word document version of that template so you can fill in your own information and experience.

When putting together your resume, there are a couple things to make sure you include as well as things to make sure to avoid.

- Always keep your resume to one page—there’s that “recruiters only look at your resume for six seconds” statistic. I’m not a recruiter—but that statistic is probably true. People skim. Make sure important information is bold/highlighted/easily noticeable—information like previous positions and places of employment, skills, and education (again, depending on if you’re applying as a “new grad” or not)

- You don’t need 5+ multi-line bullets explaining your experience—people just won’t read it. Just have 2-3 bullets starting with action words (like analyzed, queried, automated, constructed, etc.) and ending with how your analysis/results had an impact (i.e. helped leadership team do this, supported content team in deciding which services to move behind the pay wall, etc.). An important note: stay away from using hard numbers when explaining how your analysis had an impact unless you can really prove those numbers. Case in point: one candidate wrote how his work saved his company two million dollars. When asked about it, his reasoning for the cost savings was some weird combination of average project length times average employee salary with some strange assumptions—it was clear he wasn’t prepared to explain it in person, and if you’re not prepared to explain/defend something, don’t include it in your resume. Another example—watch Season 7 Episode 26 Minute 0 Seconds 30 (Search Committee Part 2) of The Office. Don’t be Darryl.
- Don’t include your picture or a list of your personal interests. At this point in the process, nobody cares and it just adds clutter to your resume
- Don’t include those “ratings bars” in your resume. How you rate your own technical skills (or your soft skills) means nothing to a recruiter or hiring manager, and you’re probably lying anyway. I came across a resume for someone that rated himself as “intermediate” at sociability—which struck me as odd.

Now, without further ado—here is a resume template that is easy to read and makes additional content easily discoverable (i.e. links to your GitHub/website/LinkedIn at the top). Again, the link to the Word document version is HERE.



Peter Pescadero
Data Scientist | Analyst

example@email.com || 555-555-5555
website.com || github.com/person

EXPERIENCE

DUNDER MIFFLIN — June 2016 to present
Senior Business Analyst — San Francisco, CA

- Querying some things... to support/guide the finance team...
- Analyzing some things... using the following machine learning techniques...
- Presented on some things... that impacted the company by...

COMPANY NUMBER TWO — April 2014 to April 2016
Research Analyst — Los Angeles, CA

- Cleaned data using specific tools... data tables had this many rows/columns...
- Automated the process of... this helped the leadership team do the following...
- Improved the process of... which had the following effect...

COMPANY NUMBER THREE — June 2012 to March 2014
Other Job Title — Los Angeles, CA

- Utilized...
- Designed...
- Worked closely with data engineering and product teams...

SKILLS

- **Programming:** Python, SQL, R, SAS
- **Machine Learning and Statistical Modeling:** Linear/Logistic Regression, Clustering, A/B Testing

EDUCATION

University of College Studies — Graduated Spring 2012
B.S. in Mathematical Economics — GPA: 3.75 (only if GPA is good)

PROJECTS

Parks and Recreation Dialogue—Visualized

- This project is an analysis of my favorite T.V. show, "Parks and Recreation." I utilized ggplot2 to construct the visualizations and BeautifulSoup to scrape the show dialogue

The CPI and Social Security

- Used various R packages (including dplyr and ggplot2) to forecast and visualize the effects of different price indices on the adjustment of social security benefits by analyzing over 70 years of price, GDP, interest rate, and tax data

Cover letter advice + cover letter template

There is a lot of conflicting information out there about how important cover letters are. I believe that a "great" cover letter does not help much in the application process, but a "bad" cover letter can kill your chances. There is not some perfect formula to writing a cover letter, but there are a few things you need to consider:

- You need the cover letter to be short
- You need to sound like you're a human
- You need to demonstrate that you can communicate clearly—grammatical errors are a BIG red flag

So with that, here is my cover letter template:

Dear [HIRING MANAGER NAME],

I am applying for your open [JOB TITLE] position. Based on the job description, I think this position looks both fun and challenging—and [COMPANY NAME] sounds like a great place for me to further my career! I believe that to be a successful [JOB TITLE] you need to have a strong combination of technical skills and storytelling ability—check out my resume and website for examples of my relevant data science work.

If you have a moment, I'd love to ask you more questions about [COMPANY NAME] and the types of projects I might be working on if hired. And if you have any questions about my resume, please feel free to reach out over email or call me at my phone number in the signature.

Hope to hear from you soon,
Peter Pescadero
555-555-5555
example@gmail.com

If you want to add 1-2 sentences about some project you worked on, or slightly tailor the cover letter to the specific company/position, be my guest. But—make sure it is clear, concise, INTERESTING, and well written.

Website advice + website template

In this section, I'll discuss a few things to consider when building your professional website, as well as provide an outline for creating an effective website.

Now, which website builder to use: I don't know much about the difference between WordPress/Squarespace/whatever else—I just used WordPress. It was straightforward to set up, easy to build and edit, and cheap. But if you have some preference or know something I don't—use whatever website builder you want. For example, I know some people who love GitHub Pages for building their websites—but I haven't used it and can't speak to its worth and ease of use. Quick note: if you can, try your hardest to make your website URL as close to your first and last name as possible. Be searchable.

You don't need to spend a ton of time and effort building some fancy, intricate website—all you need are a couple of simple pages:

- **About me**—I'd recommend that you have your "about me" section as your home page. Keep this section short—basically just try to sound like a human.

Write a couple of sentences about yourself, add a picture or two if you want, and make sure to include links to your GitHub and LinkedIn profile as well as your contact info. An example of an effective "about me" section is:

  - I am currently a Business Analyst at Dunder Mifflin. I previously worked at Stark Industries and received my bachelor's degree in economics from The University of College Studies. I love data, statistics, and watching Netflix with my dog, Buddy (pictured below!) You can contact me at example@email.com, or find me on LinkedIn and GitHub

- **Resume**—this is an easy one. Once your resume is finished, just upload it as a PDF. On WordPress, it's pretty simple to have a PDF as a standalone page on your website, so when someone clicks on the resume tab they can instantly see your full resume
- **Projects**—this is where you can show your personal projects, research projects, and polished class assignments (if you're a new graduate). I believe that this is the most important page on your website. The best projects to display are ones that can be succinctly presented—meaning, you have a well-constructed plot or table and a clear description of the project that is only a few sentences to a paragraph or so in length. Also—don't forget to include a link to your code! Here is an example of an effective way to present your projects:
  - <u>Title</u>: *"Parks and Recreation" Dialogue—Visualized in Two Charts*
  - <u>Visualization</u>: (Your charts go in this section.) *Which characters had the most screen time? Were Leslie Knope and Co really THAT mean to Jerry? (Make sure the titles are interesting and clearly convey what the plot/table is about—it is good practice to make sure that a plot/chart/table can stand alone; someone should be able to look at it and, without reading the description, know what it is about and what it is trying to convey)*
  - <u>Description</u>: *This project is an analysis of my favorite T.V. show, "Parks and Recreation." When looking at total words spoken by character, it is no surprise that Leslie and Tom occupy the top spots—but it is surprising that Ron Swanson (whose ideal conversation is no conversation) rounds out the top three! Also, when analyzing the sentiment of how the characters talk to Jerry, Tom again comes out on top as the most negative towards Jerry—although pretty much all the characters speak poorly about Park and Rec's longstanding employee. To complete this analysis, I utilized ggplot2 to construct the visualizations and BeautifulSoup to scrape the show dialogue. Click here to take a look at my code and data!*
- **Recommendations**—I believe this is a great section to talk about technical skills you have from online classes and books and maybe not from professional experience. I had taken a few online classes and tutorials—so I took the most "impactful" ones and basically wrote a review for each with a link to each course. An example of what you could write is:
  - Below I have listed some of the books and online courses that I would recommend! Feel free to read my descriptions and thoughts on each book/class!

  - Random Python class—this is a great resource for intermediate Python users who are interested in getting more exposure to using Python to complete data science tasks. You'll get the chance to use Pandas for data crunching, matplotlib for data visualization, and NumPy for numeric computation
- **Blog**—I consider this section optional. I have seen a number of data scientist/analyst blogs—some good and some bad. If you can write well and enjoy having a blog, more power to you—having a well-written and interesting blog about data science and analytics can be an effective way to demonstrate your softer skills like communication, as well as help network and get your name out in the industry. However, when I was applying I did not have a blog. To me, I felt like it was more work than it was worth—I'd rather spend the time improving my technical skills or doing something completely not data-related. Also, if you are considering starting a blog, make sure your writing is organized and clear. Grammatical errors, disorganized/jumbled posts, and unclear or wrong explanations of data analytics/science topics are obviously red flags

I'll wrap up this section with this: if you have a website, companies and hiring managers will at least glance at it. A website can be a fantastic way to show off the projects you've worked on—during one of my in-person interviews, the hiring team mentioned that they saw my price index project on my website, thought the plots I created and my explanation of the importance of price indices was interesting, and wanted to know more. By having a portfolio of my work on my website, I got a chance to showcase my data analysis skills and talk about a topic that I was extremely knowledgeable and passionate about.

How to apply and how to prepare for interviews and data challenges

Okay, now we're getting to the good stuff. You have an awesome resume, a jaw-dropping portfolio, and a Pulitzer Prize winning cover letter. What's next?

For most positions, the application process is relatively straightforward: after submitting your application, the recruiter will reach out and you'll generally have a phone screen with the recruiter. Then, you'll often be asked to complete a data task as part of the technical screen. If you pass that, you'll usually have a more technical conversation with the hiring manager, followed by the final stage: the onsite interview.

In this section, I'll discuss the following:

1. Where to look for job postings
2. How to follow up after submitting your application
3. How to prepare for the phone screen with the recruiter
4. How to impress the hiring team with the technical data assignment
5. Common interview questions and how to answer them

Where to look for job postings

This will be a pretty short section. The best places to look for job postings are LinkedIn, Glassdoor, AngelList, and Indeed, generally in that order. Note—if you end up using AngelList, you will need to put together a profile similar to LinkedIn. There isn't a magical one-stop shop for all listings, but if you scour these sites you will definitely see the bulk of the job postings in data science and analytics.

How to follow up after submitting your application

I consider the advice in this section somewhat optional. I would not do this for all the jobs you apply to, but following up for the select few jobs you are really interested in can make a difference.

Okay. So you submitted your application but you haven't heard anything back. For job openings you are particularly interested in, I would do the following:

1. Find the hiring manager's email. This is generally pretty easy—you'll just have to do some digging. You can find emails on LinkedIn, or use something like hunter. If you can't find the hiring manager's email, try to find someone on the data team
2. Once you have the email address, write a thoughtful email and attach your resume. Keep the email short, but interesting. Say what job you recently applied for, what you like about the company, how you believe you can make an impact, and end with a soft ask. Something like:
   a. Subject line—*Question about joining the Dunder Mifflin data team*
   b. Body—*Dear Michael Scott, I recently applied for Dunder Mifflin's open data science analyst position. I admire Dunder Mifflin's commitment to being the people person's paper people—and I'd love to be part of the team. I love all things data—from helping a social media marketing start-up make data-driven decisions or using data to help a subscription box company predict when users will churn, I have the skills to help Dunder Mifflin use data to make better business decisions. So with that being said, would you have a chance to talk to see if there might be a place for me at Dunder Mifflin? Best, Peter Pescadero*
3. Continue to follow up until you hear a response

Another strategy that works is to message another data analyst/scientist on the team (either email or on LinkedIn), and say something like:

Hey Peter!

Hope you had a good weekend. I noticed that there was a job opening at your company. I'd love to connect and learn more about your journey into data science and how you like it so far at Dunder Mifflin. Let me know if you're free this week to chat over coffee.

I would recommend you send this email out before you apply—if they never respond, apply to the job anyway. However, if they do respond, there's a good chance you can give your resume to them and they can forward your resume to the recruiter.

I particularly like this strategy—for a few reasons. One, most people are nice—and it's hard to turn down free coffee (quick note: buy their coffee for them). If/when you meet, keep the conversation light; ask them about their experience, what advice they have for you, and what they like about their company.

How to prepare for a phone screen with the recruiter

This stage of the application process is also always pretty straightforward. The recruiter at this point is trying to check a few things:

- You can clearly speak about the stuff on your resume—the recruiter might ask you to explain a project you worked on that used data
- The recruiter will most likely ask you what your programming background is—generally, s/he is looking for keywords like SQL, Python packages like maplotlib, Pandas, and NumPy, etc.
- Always have a few reasons for why you are leaving your current job and why you're interested in this position/company (in a later section, I'll go into more detail about how to answer some standard interview questions). Also, always have a few questions to ask the recruiter as well—should be common sense, but definitely something that is overlooked when preparing for recruiter phone screens
- The recruiter might ask about what salary range you're interested in—see a later section on how to answer the salary range question
- Be friendly and sound interested and excited. Recruiters are also trying to gauge if you'd be a "good fit." It's easier to check off this "good fit" box when you're friendly and having an engaging conversation with the recruiter

Remember: the recruiter probably does not know much about the position you're applying for—meaning, they're not going to ask you technical statistics questions, or questions about SQL/Python/R. They might have a list of questions they're given to ask, like "have you worked with window functions?" or "what type of statistical modeling techniques have you used?" but the point of these is to make sure you have the skills the hiring manager is looking for.

Lastly, have an open (email) dialogue with the recruiter. If they set up a time to have you talk to the hiring manager, or come in for an onsite interview, ask them if they have any advice or anything that might be helpful to prepare for. As the application process goes into the later and later stages, recruiters want to get positions filled, and there is a good chance they will offer at least some advice. The more information you can have at each step of the process, the better.

How to impress the hiring team with the technical data assignment

Oh, the famed data task. In this section, I have a list of DOs and DON'Ts for you, as well as a conceptual discussion on how to ace the data assignments (based on real assignments from some of the biggest tech companies!) with advice for how I would approach the questions, how I would approach writing the code, and how I would present my results.

In this technical screening step, there are usually two types of tasks:

1. Timed programming challenge—I do not have much advice for how to prepare for these. Most companies (that I know of) do not use this as a step in the application process, but if you find yourself having to complete one of these, just do your best and don't stress about finishing in the allotted time—just make sure what you can finish is correct and a good representation of your programming skills
2. Data analysis challenge—this is the most frequently used technical screen. It usually involves a dataset and some questions to show off your programming skills as well as your ability to analyze and synthesize results. **This section is focused on this type of technical challenge**

The data analysis challenge is used to evaluate the following:

1. Can you demonstrate the technical skills you discussed in your resume?
2. Are you able to handle and clean messy data?
3. Is your code clean, well-written, and well-documented?
4. Are you able to clearly communicate and present your results?

The data assignment phase is where you can initially stand out in the interview process. If you follow my advice, you can set yourself apart from the other applicants with a strong data assignment. Other than the resume/cover letter step, this is the one phase of the application process you have complete control over. You can generally spend as much time on it as you want, troubleshoot your code, look at resources online if you need to, and think through how you want to approach the task and how you want to present your work.

So with that said, here are some DOs and DON'Ts for how to be successful in the data assignment stage:

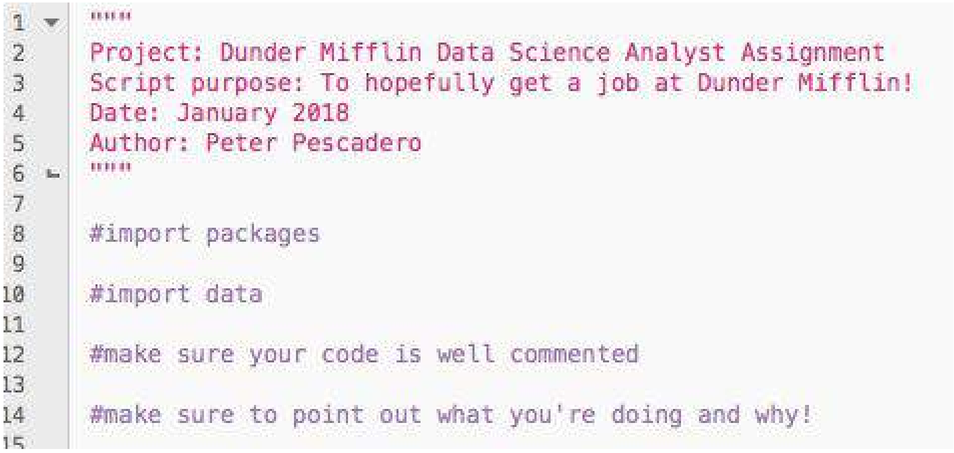
- DO: if you're really interested in the job, finish the assignment. Not sure why this is the case—but a fair number of the assignments I review are incomplete. If you really want to make a strong impression, finish the assignment
- DO: make it extremely easy for the team to review your assignment. In 99.99% of cases, all you need to send back are two files: one text-file of your code, and one PDF with the questions you had to answer in bold, and your answers/visualizations/results below
- DON'T: send a Jupyter notebook, html file, etc. You run the risk of the formatting being off and it will look disjointed and clunky. Just think about this: if you get hired, your job would be to take a question, use data to answer that question, and present and/or communicate your findings to non-technical people. You're not going to send someone on the marketing team a Jupyter notebook where they have to sift through the code
- DON'T: send your code without a write-up of your results/findings. The hiring team will not replicate your results; they probably won't even look at it. You're wrong if you think people will spend the time debugging your code, making sure the directories are correct, all the packages are installed to run, etc. Again, you're not going to send an executive a text file saying "make sure to install the following packages to review my findings." So don't do that in your data task

The people looking at your assignments are interested in two things: clean code and clear storytelling ability—so make sure to demonstrate that in your assignments!

Now, I'm going to describe two database schemas that are similar to what might be given to you. I'll discuss potential questions you could be asked to answer using the data or questions you should consider answering if the prompt is vague, how you should approach answering those questions, and potential "Easter eggs" and things to look out for and consider while doing your data cleaning and analysis.

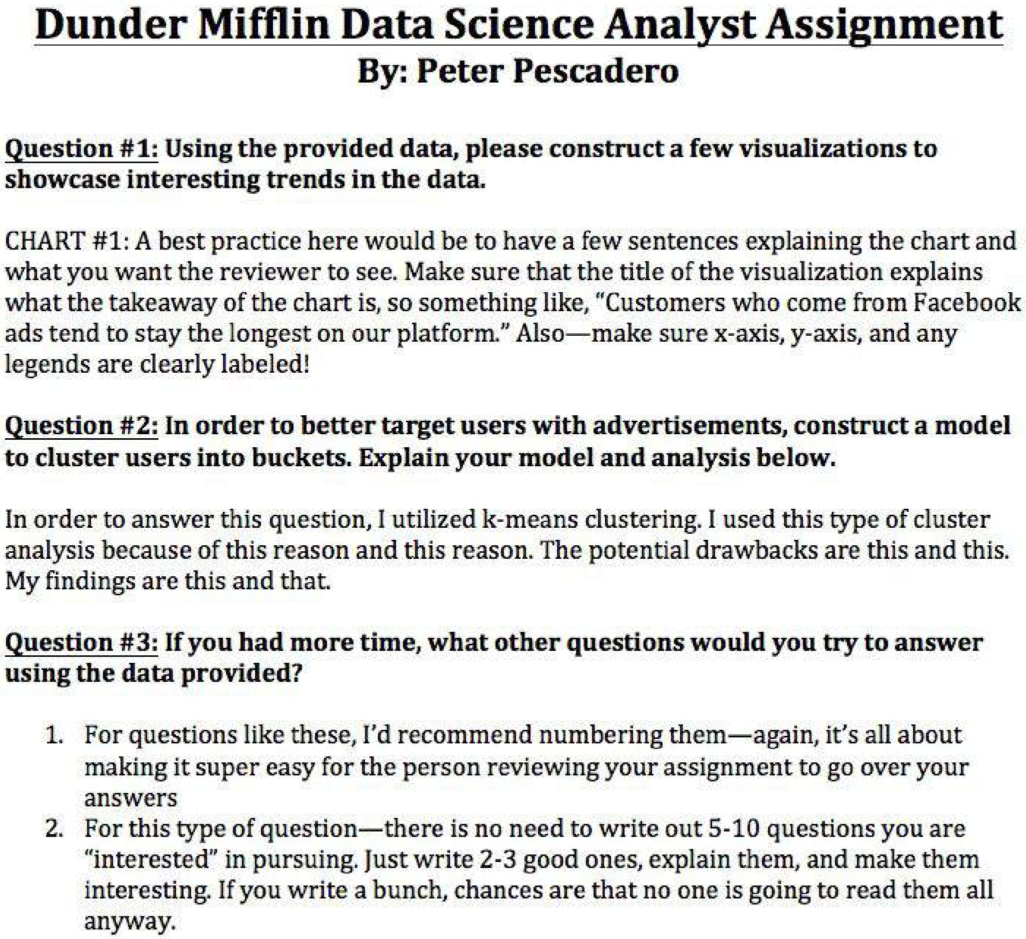
But first, let's quickly discuss the two files to send once you've completed the data assignment: a text-file with your code, and a PDF of your analysis and results:

For the text-file, keep it simple, clean, and well-documented. Format it like the example for the personal project code; see the screenshot below for a reminder:



```
"""
Project: Dunder Mifflin Data Science Analyst Assignment
Script purpose: To hopefully get a job at Dunder Mifflin!
Date: January 2018
Author: Peter Pescadero
"""

#import packages

#import data

#make sure your code is well commented

#make sure to point out what you're doing and why!
```

For the PDF showing your analysis, keep it organized and easy for whoever is reviewing it to know which question you are trying to answer. See the screenshot below for an example of how to organize and format this page:



Dunder Mifflin Data Science Analyst Assignment

By: Peter Pescadero

Question #1: Using the provided data, please construct a few visualizations to showcase interesting trends in the data.

CHART #1: A best practice here would be to have a few sentences explaining the chart and what you want the reviewer to see. Make sure that the title of the visualization explains what the takeaway of the chart is, so something like, "Customers who come from Facebook ads tend to stay the longest on our platform." Also—make sure x-axis, y-axis, and any legends are clearly labeled!

Question #2: In order to better target users with advertisements, construct a model to cluster users into buckets. Explain your model and analysis below.

In order to answer this question, I utilized k-means clustering. I used this type of cluster analysis because of this reason and this reason. The potential drawbacks are this and this. My findings are this and that.

Question #3: If you had more time, what other questions would you try to answer using the data provided?

1. For questions like these, I'd recommend numbering them—again, it's all about making it super easy for the person reviewing your assignment to go over your answers
2. For this type of question—there is no need to write out 5-10 questions you are "interested" in pursuing. Just write 2-3 good ones, explain them, and make them interesting. If you write a bunch, chances are that no one is going to read them all anyway.

Alright, alright, alright. Now I'm going to walk you through the two mock data assignments. I'll describe a dataset and discuss a few things to consider.

Mock Data Assignment #1:
(Note: there is no actual data in this section, just a conceptual exercise)

<u>Column name</u>	<u>Column description</u>
Customer_id	Unique alphanumeric ID assigned to customers when they install the application
Acquisition_channel	Channel where customer came from before installing application
Install_date	Date of application install
Account_creation_date	Date of account creation
Action_date	Date when user performs an action on platform
Action_type	String description of action performed by user on platform

In most cases, the dataset(s) you are given require at least a little bit of cleaning. This can be anything from converting variable types, dealing with unnecessary whitespaces, things like that. Not to make you paranoid, but it appears to be relatively common for companies to have a few “Easter eggs” in the data—either in the data cleaning stage or the analysis stage of the assignment. So, it is important to keep this in mind and be on the lookout for things like null or missing values, incorrectly spelled data, and certain IDs acting like “bots” that will screw with your results. When you find these types of things, make sure to document them either in your code comments or in your write-up.

Oftentimes, companies will ask super general questions in the data assignment to gauge what insights you can gather from the data; questions like: what trends do you see in the data? What insights from the data are actionable? If there are multiple datasets, how do the datasets differ? What are the most interesting findings? Can you use a model to predict X, Y, or Z?

After cleaning and checking the above dataset for any errors, there are a number of “low-hanging fruit” visualizations you can put together:

- Bar chart showing number of customers by acquisition channel
- Histogram showing length of time it takes users to go from installing the application to creating an account (the thinking here is that the user experience is like a funnel—and users go from acquisition channel to installation to account creation—a long installation to account creation time suggests that there is confusion or a pain-point for users). Similarly, you can make a funnel showing what percent of users go from acquisition channel to installation to account creation

- Plot showing number of actions by day—you can also divide users into cohorts based on when they installed the application—do different cohorts tend to behave differently?

Obviously, this list isn't exhaustive—but it should give you a good idea of some simple visualizations you can put together to show companies that you can clean data and present it in an aesthetically-pleasing way.

You can also suggest using the data for prediction—for example, based on when and what action a user takes, can you predict when and what their next action will be?

And lastly—even if they don't ask, it is also good practice to have a short section at the end with any questions or concerns you have about the data and possible avenues for further analysis. You want to make the most out of the opportunity to showcase that you are vigilant and concerned about data integrity (meaning, you make sure the data is correct/accurate/error-free), and you are thinking about other ways to analyze and glean insights from the data.

Mock Data Assignment #2:
(Note: there is no actual data in this section, just a conceptual exercise)

Column name	Column description
Activity_date	Activity date of an action on platform
User_id	Unique alphanumeric ID assigned to user
Device_type	Type of device
Time_on_page	Length of time user stayed on a specific page
Content_genre	String description of content genre
Page_title	Title of page
Content_length	Length of content on page (i.e., number of words)

Similar to the mock data assignment #1, here are a few visualization ideas:

- Histogram showing lengths of time users tend to stay on a page—what about segmenting this by content_genre or content_length?
- Bubble chart showing most popular daily content
- Donut chart showing site activity by device type

Can you cluster or segment users based on what content they consume? In addition, if this was a publishing platform, can you tag a user as an "entertainment" or "sports" reader? Does the length of the page title have any effect on how "popular" content is? Does it get more visits if the title is longer/shorter?

Try to push yourself to come up with at least one visualization that you know other applicants will most likely not have—you want to stand out and demonstrate your critical thinking and creativity. You want whoever is reviewing your assignment to think to themselves, “Huh, that’s pretty cool.”

Common interview questions and how to answer them

Before I begin, let me reiterate: this Mode Analytics tutorial has great business “case study” questions. During some of my interviews, I was asked variations of the problems from this tutorial. Now, a short list of questions to prepare for:

1. **Why are you interested in [said company]?**
   a. You just need to have a cohesive answer about why you're interested in a specific company—do you use their product? Are you excited about the chance to use their unique data to answer interesting business questions? Have you heard amazing things about their culture?
2. **Can you discuss a project you worked on that used messy data (or had to join various datasets together)?**
   a. Usually, someone on the data science and analytics team will ask this question. A data scientist/analyst spends a ton of time dealing with messy data and figuring out the best way to join/merge data-tables. You need to be able to talk clearly about a time you had to handle 2/3/4+ large datasets: what issues did you run into and how did you deal with them? What datasets did you have to begin with, and what dataset were you interested in putting together?
3. **Can you discuss a project you worked on that used statistical modeling?**
   a. Again, usually someone on the data science and analytics team will ask this question. You need to have a clear and concise answer for a time you developed a model to answer a business question. Why did you choose that specific model? What assumptions did you have to make? What questions were you trying to answer?
4. **Can you discuss a project you worked on that used data?**
   a. This is a variation/more general version of the previous two questions. Basically, you need to be able to talk intelligently about a time you used data to answer a question. Ideally, you can discuss an example project in two ways: one, with a more technical interviewer that gets more in the nitty-gritty of why you made certain decisions, what data issues you had, what programming languages/packages you used, etc. And two, you also need to be able to explain an example project to a non-technical interviewer—like a product manager or an executive: what question were you trying to answer? How did you arrive at that conclusion? You need to be able to explain a data project, from planning to the communication of results, in an interesting and non-technical way.

5. **When was a time you had to explain your analysis to a non-technical audience?**
    a. This is pretty self-explanatory. The bulk of your job as a data scientist/analyst is taking complex analyses and explaining them in interesting and clear ways. Being able to talk about a time you effectively did this is critical during the interview process.
6. **When was a time you had to explain poor results/findings with tact and thoughtfulness?**
    a. This happens quite a bit. As a data scientist/analyst, you will end up analyzing and evaluating a new product feature, initiative, marketing strategy, etc. that ends up failing. You need to be able to demonstrate that you're not afraid to be honest, but giving "bad" news is definitely a challenge. You need to say that you understand that a lot of time and effort goes into a new feature/initiative/strategy, and that being mindful of that is important. Your job isn't to drop data bombs and call out people and teams—it's to be a resource and to help the company reach its goals.
7. **The product team is interested in testing a new feature for returning users from the United States. Plan an experiment, describe your analysis, and discuss how you would communicate results (in case you're interested, here are two cool but somewhat technical resources for how to think about designing experiments: one and two.) And one last reminder: this Mode Analytics tutorial has an awesome case study on A/B experiments! There are a number of important things to discuss with this question:**
    a. Make sure to discuss what metrics are important to evaluate
        i. Discuss that it is a common mistake with A/B testing to not define what metrics you are trying to move before you start an experiment. Also, discuss that it is important to not just throw in every possible metric and evaluate everything—when you're designing experiments, you need to know how to react when an experiment finishes—so you need to decide what success looks like
    b. Calculating the sample size
        i. Why is calculating the sample size important? You need to have an appropriate sample size for experiments in order to get enough power for valid results. You also don't want to undershoot—and have to run an experiment again
        ii. You can discuss Type I errors (false positive) and Type II errors (false negative)
        iii. You can discuss that when you're designing an experiment, (in most cases) you can either run the experiment for longer/shorter, or show the feature to more/less users in order to show the experiment effect to the correct sample size

        iv. You can point out that when calculating the sample size, you need to only look at historical metrics for returning users from the United States

    c. While the experiment is running

        i. You can discuss that it is important to not run general experiments during strange times or holidays—i.e., best to not run an experiment during a holiday because there is a greater chance the data might be noisy, and starting an experiment over a weekend is risky if there are any errors

        ii. Also regarding errors—oftentimes, the analytics team is responsible for high level monitoring of experiments while they are running to make sure nothing is broken. This can involve reviewing error logs, raw logs, etc. Making a quick mention of that is a good idea as well

        iii. Do not suggest p-hacking—or any variation of it. Instead, mention that it is important to wait until the experiment has finished before reviewing/analyzing results

    d. Analyzing the experiment results

        i. Suggest filtering the analysis somehow. For example, the experiment was looking at returning U.S. users. How did various cohorts respond? Did users who have been around for 1+ years behave similarly to users who have only been around for 1-2 months?

        ii. Obviously, you need to discuss which statistical tests to use to determine whether the results are significant. Is the data binary (then chi-square, proportion test, etc.) or continuous (t-test)? If you're confused by this stuff, spend 20-30 minutes reading through some basic stats stuff online.

        iii. Lastly, spend some time discussing that it is important to interpret the results—do the experiment results suggest anything about LTV? Or user behavior? Or potentially a product feature issue? You need to bring it back to what's most important: using data and statistics to answer a business question

8. **An often-used analytics dashboard is showing a strange drop in a key KPI. How would you go about investigating this? There are a number of ways you can answer this, I'll highlight a couple things you can discuss below:**

    a. There might actually just be an error in the data—like a possible change in how something is reported or other data engineering issue

    b. Maybe the KPI is opt-ins, and there is a worrying drop in opt-ins over the past 48 hours. Was there a change in the opt-in or upsell screen? You can discuss talking with the product or marketing team to make sure there weren't any changes you were not aware of before you dive more deeply into the data

    c. Discuss how else you can filter the analysis. Is the drop in opt-ins for all users? Or only U.S. users on mobile? Is the drop just because of a holiday or seasonality? Is traffic from a specific source down, and that might be causing the dip? These are the types of things to think about and discuss—and again, take a look at this Mode Analytics tutorial for a similar case study

9. **Based on your experience with our company, suggest a feature change. This evaluates a few things: one, that you can talk intelligently about the company and their product (if they're product-based), and two, that you can talk intelligently about how to use data analysis to make feature improvements. Below are a few things to consider when answering this question:**
    a. Not sure if this is even necessary to say—but obviously don't say one of two extremes: that there is nothing to improve, "I love your company and your product, I can't think of anything I'd want to change!" or insinuate that there are a lot of improvements you think are necessary
    b. Once you bring up a potential feature change, make sure to discuss how you would test that feature and what metrics are important to consider
    c. When asked this type of question, you have the opportunity to discuss what you like and dislike about the product and show that you've at least spent time using or thinking about the product
10. **You have a table with the following schema: activity_date, device_type, source, content_type. Answer the following questions:**
    a. What type of dashboard can you develop with these metrics?
        i. There are a ton of dashboard/visualization ideas you can discuss: from most popular content by device or by source, a daily chart of top content, visualizations to see where users are coming from, etc. In addition, is there a way to cluster by content type?
    b. How can we make sure to feature the most popular content?
        i. Is certain content more popular based on the type of device a user is on, where they are coming from, or when they are visiting the site? You can discuss the possibility of featuring content based on these
    c. What other data would be helpful to include with the above table?
        i. Just a couple of ideas: geo/region_id/state, content_position, activity_starttime/activity_endtime, search_term
11. **What questions do you have for me?**
    a. Make sure you have at least a few questions prepared. If you're talking to a product manager, ask them about the product roadmap—what are they working on? What are they especially excited about? If you're talking to someone on the data team, ask them about how the team works, or data issues and problems they've faced.

How do I negotiate? Should I negotiate?

To start: yes. You should always negotiate. Tech companies in particular EXPECT you to negotiate. You can almost always get at least another $5k or 5% increase in base salary—and oftentimes, a lot more.

In this section, I'll give you a few short recommendations for how to negotiate; however, I would recommend taking a look at this post for a fantastic piece on how to negotiate well.

First thing to consider: in most cases, a recruiter will ask you what your salary range is—my recommendation is to not give them a number or a range. Say something along the lines of "I'm open to any competitive offer." If they push you on giving them a number, I would counter with something like "I really don't have a set range—it really depends on the entire compensation package." I get it—this is an awkward conversation—but don't let an awkward conversation get between you and a 5k/10k/20k jump in salary. You won't regret having this awkward conversation, you will regret losing out on thousands of dollars. Ultimately, you want them to give you a number first—if that number is in your range, great, if not, now is the time to tell them that the number isn't what you're looking for.

Now, what about once you actually have an offer? Remember: it is not selfish or wrong to negotiate—it is expected. First, aim high so there is room to compromise—and at this point, give them a number and not a range. Explain that you're really excited about the company and the position, and bring up an example or two about what you'd bring to the company. Even if you don't have another offer to use in the negotiation—that's not a problem. Consider saying something simple but effective, like: "I'm really excited about the chance to join Dunder Mifflin—however, I am far along in the process with a few other companies, and an additional $10k would make this decision a lot easier for me."

Still want more advice? Again, take a look at this post.

What is the career trajectory for someone in data science/analytics?

One of the things I really enjoy about data analyst/scientist roles is where they can take you. On one hand, you can continue honing your technical skills and move into even more of a technical data science role. Or, you can move from being a data analyst/scientist to a manager position.

At my current company, a former data science analyst took on more and more technical work and ultimately transitioned to an engineering role, while a different former data science analyst transitioned to a leadership position on our data science and analytics team.

Still want more help? Contact me!

If you found this guide helpful but still want more guidance, contact me! I offer:

- Resume, cover letter, and website feedback
- Interview preparation sessions
- Data task advice (but no, I won't do your data task for you)
- SQL training

Shoot me an email at: **petersguides@gmail.com** and let's get you hired.

Also, if you found this guide helpful and want to share it, email me and I can get you an affiliate link so you can make money every time someone buys through your link.

I'll also leave you with this: don't be discouraged if it feels like things are moving slowly for you—there are a lot of factors that you simply won't be able to account for in the recruiting process. The best thing you can do is present the best possible version of yourself, learn from every rejection, figure out what you are deficient in, and keep applying.

Best of luck!
-Peter