UDACITY

Ultimate Skills Checklist for Your First Data Analyst Job

As personal device usage explodes and billions of users get online, there has been a veritable explosion of data that is being collected. However, the ability to analyze that data and make sense out of it is not improving at the same rate.

In my career leading data science teams at Yahoo!, Google, Groupon, and Udacity, I've experienced firsthand the lack of qualified professionals who can analyze data and find useful patterns in it.

> **"My hiring needs have *always* exceeded qualified candidates, which is why I'm thrilled to see this skills checklist."**

These are exactly the skills I look for in the data analysts I have hired when growing data teams at Yahoo!, Google, Groupon, and Udacity.

With better data, companies improve user experience in various ways - better search results (Google), recommending better products (Amazon, Netflix), showing interesting content in your news feed (Facebook), optimizing site design, and building the right features for their products, among other things.

The data analysis skills needed to do these things are described in this guide. Best of luck and happy learning!

Nitin Sharma
VP of Engineering and Data Science
Udacity

Welcome

Welcome to your ultimate skills checklist for getting your first job as a data analyst! You're standing at a unique and exciting time in the birth of a new field - data science career opportunities are expanding by leaps and bounds, and so are your options for learning.

Having choices is always a good thing. But sometimes it's helpful to have a guide, so we're here to help you cut the noise.

We recently developed the first-ever Data Analyst Nanodegree, which guides students along a project-based curriculum to learn the skills they need to get their first job in data. We learned a TON from talking to employers to make sure our skills list is cutting edge, and we can't wait to pass this skills list on to *you*.

In this guide, you'll find the ultimate skills checklist for getting a job as a data analyst, as well as resources where you can get started.

Congratulations on taking a step towards using data in your career! Read on for the ultimate data skills checklist and recommended resources.

Data Analyst Skills Checklist: What We'll Cover

Here's a breakdown of the skills you need to learn to be a data analyst. Take some time to review this list - how many boxes can you check off?

For more detail on these skills and for learning resources, navigate to the corresponding pages listed.

Programming

Programming will be an integral part of your everyday work. This is one key skill that will separate you from a traditional business analyst or statistician. At any given date, you may need to write programs to query and retrieve data from databases. Or you may need to write programs to run your data set on machine learning algorithms. Therefore you should be able to program well in one or more programming languages, and have a good grasp of the landscape of the most commonly used data science libraries and packages. Both Python are R are good programming languages to start with because of their popularity and community support.

- ☐ **R programming language**: a special purpose programming language and software environment for statistical computing and graphics. Know these R packages:
  - ☐ **ggplot2**: a plotting system for R, based on the grammar of graphics
  - ☐ **dplyr** (or plyr): a set of tools for efficiently manipulating datasets in R (supercedes plyr)
  - ☐ **ggally**: a helper to ggplot2, which can combine plots into a plot matrix, includes a parallel coordinate plot function and a function for making a network plot
  - ☐ **ggpairs:** another helper to ggplot2, a GGplot2 Matrix
  - ☐ **reshape2**: “Flexibly reshape data: a reboot of the reshape package”, using melt and cast
- ☐ **Python programming language**: Python is a high level programming language with many useful packages written for it. Know these Python packages:
  - ☐ **numpy**: an optimized python library for numerical analysis, specifically: large, multi-dimensional arrays and matrices
  - ☐ **pandas**: an optimized python library for data analysis including dataframes inspired by R
  - ☐ **matplotlib**: a 2D plotting library for python, includes the pyplot interface which provides a MATLAB-like interface (see ipython notebooks and seaborn below)
  - ☐ **scipy**: a library for scientific computing and technical computing
  - ☐ **scikit-learn**: machine learning library built on NumPy, SciPy, and matplotlib

- ☐ optional:
  - ☐ **ipython**: an improved interactive shell for python with introspection, rich media, additional shell syntax, tab completion, and richer history
  - ☐ **ipython notebooks**: a web-based interactive computational environment
  - ☐ **anaconda**: a python package manager for science, math, engineering, data analysis with the intent of simplifying and maintaining compatibility between library versions. Also useful for getting started with ipython notebooks.
  - ☐ **ggplot**: and (in-progress) port of R's ggplot2 which premised upon a grammar of graphics
  - ☐ **seaborn**: a Python visualization library based on matplotlib with a high-level interface

- ☐ **Spreadsheet tools (like Excel)** - These tools visually present data into rows and columns allowing for easy data manipulation. Many organization analyze and communicate data through spreadsheets.
  - ☐ Create dashboards and pivot table reports to share for business analysts

Additional Skills for Udaciousness

- ☐ **Javascript and HTML for D3.js** - thse are web development languages which turn static visualizations into interactive visualizations to create online dashboards and reports. Javascript packages include:
  - ☐ **D3.js**
  - ☐ **AJAX implementation** - nice to know
  - ☐ **jQuery** - nice to know
- ☐ **C/C++ or Java** - Low-level programming languages that help turn development high-level code such as (Python and R) into efficient production-level ready code for deployment

Statistics

At least a basic understanding of statistics is vital as a data analyst. For example, your boss may ask you to run an A/B test, and understanding of statistics will help you interpret the data that you've collected. You should be familiar with statistical tests, distributions, maximum likelihood estimators, etc. One of the more important aspects of your statistics knowledge will be understanding when different techniques are (or aren't) a valid approach.

Descriptive and Inferential statistics

One of the most important concepts to understand in statistics is that of sampling. That is, when you collect any data, you are often only seeing a subset of all possible data that could be collected on that topic. The collected data is known as a *sample*, and the larger space from which the data is drawn is typically called a *population*. Quantitative measures that describe properties of a sample are referred to as *descriptive statistics* - they describe the data at hand in a compact and useful form. We often wish to *infer* properties of the larger population just by looking at our sample - these predictive measures are known as *inferential statistics*.

- ☐ Mean, median, mode
- ☐ Data distributions
  - ☐ Standard normal
  - ☐ Exponential/Poisson
  - ☐ Binomial
  - ☐ Chi-square
- ☐ Standard deviation and variance
- ☐ Hypothesis testing
  - ☐ P-values
- ☐ Test for significance
  - ☐ Z-test, t-test, Mann-Whitney U
  - ☐ Chi-squared and ANOVA testing

Experimental design

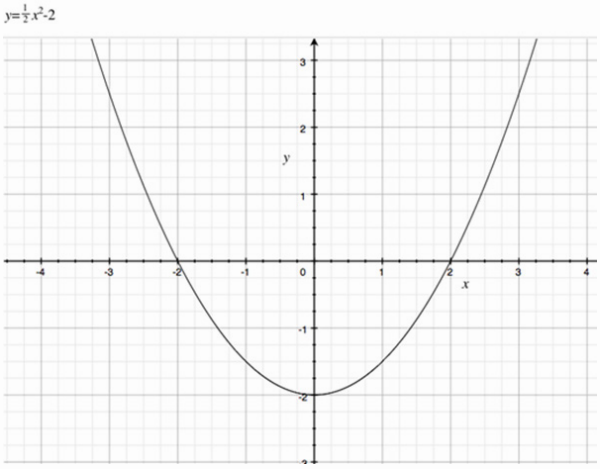
Properly laying out an experiment helps ensure that conclusions we draw from the observed results are not misleading. Experimental design is the systematic process of choosing different parameters that can affect an experiment, in order to make results *valid* and *significant*. This may include deciding how many samples need to be collected, how different factors should be interleaved, being cognizant of ordering effects, etc. Formal terms used to describe experiments are useful in succinctly and unambiguously conveying design parameters.

- ☐ A/B Testing
- ☐ Controlling variables and choosing good control and testing groups
- ☐ Sample Size and Power law
- ☐ Hypothesis Testing, test hypothesis
- ☐ Confidence level
- ☐ SMART experiments: Specific, Measurable, Actionable, Realistic, Timely

Mathematics

At a basic level, you should be comfortable with college algebra. Specifically, you should be able to translate word problems into mathematical expressions, manipulate algebraic expressions and solve equations, and graph different types of functions and understand the relationship between a function's graph and its equation.

- ☐ Translate numbers and concepts into a mathematical expression: 4 times the square-root of one-third of a gallon of water (expressed as g): 4 √(1/3 g)
- ☐ Solve for missing values in Algebra equations: 14 = 2x + 29
- ☐ How does the 1/2 value change the shape of this graph?



Additional Skills for Udaciousness

On a more Udacious level, it will be good to have a solid grasp of multivariable calculus and linear algebra. These two areas of math make up the basic foundation to understand machine learning and to effectively manipulate data efficiently in your data models.

- ☐ Linear algebra and Calculus
- ☐ Matrix manipulations. Dot product is crucial to understand.
- ☐ Eigenvalues and eigenvectors -- Understand the significance of these two concepts
- ☐ Multivariable derivatives and integration in Calculus

Machine Learning

Machine learning is incredibly powerful if you are working with large amounts of data, and you want to make predictions or calculated suggestions based on these data. You won't need to invent new machine learning algorithms, but you should know the most common machine learning algorithms, from dimensionality reduction to supervised and unsupervised techniques.

Some examples are principal component analysis, neural networks, support vector machines, and k-means clustering. You may not need to know the theory and implementation details behind these algorithms. But you should know the pros and cons of these algorithms, as well as when you should (and shouldn't) apply these algorithms.

- ☐ **Supervised Learning**

  Supervised learning is useful in cases where a property (usually known as *label*) is available for a certain dataset (*training set*), but is missing and needs to be predicted for other instances (a *test set* of such instances is used to measure and refine the effectiveness of the learning algorithm). Note that the label can be a numeric value, in which case the difference between what is predicted and the corresponding actual value constitutes an error measure.

  - ☐ Decision trees
  - ☐ Naive Bayes classification
  - ☐ Ordinary Least Squares regression
  - ☐ Logistic regression
  - ☐ Neural networks
  - ☐ Support vector machines
  - ☐ Ensemble methods

- [ ] **Unsupervised Learning**

  Sometimes the goal is not to predict the value of a specific property. Instead, we are faced with the challenge of discovering implicit relationships in a given dataset. The most common example of this is grouping or *clustering* items based on their similarities and differences. In such cases, the dataset does not define any groups, and as a result, items are not pre-assigned. Hence the dataset is called *unlabeled* (here, cluster assignment could be thought of as a label) and the corresponding learning process is known as *unsupervised*.

  - [ ] Clustering Algorithms
  - [ ] Principal Component Analysis (PCA)
  - [ ] Singular Value Decomposition (SVD)
  - [ ] Independent Component Analysis (ICA)

- [ ] **Reinforcement Learning**

  Certain situations fall between these two extremes, i.e. there is some form of feedback available for each predictive step or action, but no precise label or error measure. A classic formulation of this category of learning problems would involve some form of reward or *reinforcement* being given for each correct action. A reinforcement learning agent can thus keep generating actions while it learns, continually refining its internal model to make better choices.

  - [ ] Q-Learning
  - [ ] TD-Learning
  - [ ] Genetic Algorithms

Data Wrangling

A less celebrated part of doing data science is manually collecting and cleaning data so it can be easily explored and analyzed later. This process is otherwise known as “data wrangling” or “data munging” in the data science community. Though not as glamorous as building cool machine learning models, data wrangling is a task that data scientists can spend up to 50-80% of their time doing.

So why do you need to wrangle data? Often times, the data you’re analyzing is going to be messy and/or difficult to work with. Because of this, it’s really important to know how to deal with imperfections in data. This will be most important at small companies where you’re an early data hire, or data-driven companies where the product is not data-related (particularly because the latter has often grown quickly with not much attention to data cleanliness). Nevertheless, this skill is important for everyone to have no matter where you work.

- ☐ **Python:** ideal for wrangling data
  - ☐ Learn about Python String library for string manipulations
  - ☐ Parsing common file formats such as csv and xml files
  - ☐ Regular Expressions
  - ☐ Mathematical transformations
    - ☐ Convert non-normal distribution to normal with log-10 transformation
- ☐ **Database systems (SQL-based and NO SQL based)** - Databases act as a central hub to store information
  - ☐ Relational databases such as PostgreSQL, mySQL, Netezza, Oracle, etc.
  - ☐ Optional: Hadoop, Spark, MongoDB
- ☐ **SQL**: (Structured Query Language) is a special-purpose programming language for relational database management system (RDBMS)

Communication and Data Visualization

As a Data Analyst, your job is to not only interpret the data but to also effectively communicate your findings to other stakeholders, so they can make data-informed decisions. Many stakeholders will not be interested in the technical details behind your analysis. That's why it's very important for you to be able to communicate and present your findings in a way that is easy to understand for your audience, both technical and non-technical. It can be immensely helpful to be familiar with data visualization tools like ggplot, matplotlib, seaborn and d3.js. It is important to not just be familiar with the tools necessary to visualize data, but also the principles behind visually encoding data and communicating information.

- ☐ **Data visualization and communications** - Knowing how to present the data in the most consumable way is crucial to communicating the message
  - ☐ Understand visual encoding and communicating what you want the audience to take away from your visualizations
  - ☐ Programming
    - ☐ matplotlib, ggplot, seaborne, d3.js,
  - ☐ Presenting data and convincing people with your data
    - ☐ Know the context of the business situation at hand with regards to your data
    - ☐ Make sure to think 5 steps ahead and predict what their questions will be and where your audience will challenge your assumptions and conclusions
    - ☐ Give out pre-reads to your presentations and have pre-alignment meetings with interested parties before the actual meeting

Additional Skills for Udaciousness

- ☐ **Crafting a story in presentations** - Data analysts should know how to present an engaging presentation that empowers the audience to take action. Data analysts should be aware of the type of audience she is presenting to and craft the presentation to that type of audience.

Data Intuition (Thinking like a data scientist)

Your boss or coworkers, such as other engineers or product managers, may want you to address important questions with data-informed insights. But you may not have enough time to address all of their questions or analyze all of the data. Therefore, it is important for you to have intuition about what things are important, and what things aren't.

For example, understanding what methods should you use or when do approximations make sense? This will help you avoid dead ends and focus on the important questions or bits of data that you have to analyze. The best way to develop this intuition is to work through as many data sets as you can. Working through data analysis competitions like Kaggle can help you develop this kind of intuition.

- ☐ **Ask the right questions** - The data analyst must be aware of the "question behind the question" - what are the exact business questions and issues that is driving the need to analyze data?
  - ☐ Recognize what things are important and what things are not important

Additional Skills for Udaciousness

- ☐ Project management involves organizing one's team and managing communications and expectations across multiple departments and parties on any data analyst project
  - ☐ Communicate effectively with stakeholders including:
    - ☐ Executives and project sponsor
    - ☐ Project leads
    - ☐ Product managers
    - ☐ Engineering, Sales, Information Technology
- ☐ **Subject Matter knowledge in area of analysis** - This skill is developed through experience working in an industry. Each dataset is different and comes with certain assumptions and industry knowledge. For example, a data analyst specializing in stock market data would need time to develop knowledge in analyzing transactional data for restaurants.

What Next?
Learning Resources

You made it to the end of the checklist - congratulations!

Whether you were able to check off many skills, or whether you're going to start tackling the checklist from the very beginning, pat yourself on the back for taking a big step by reading this guide.

As we mentioned at the beginning of this guide, we're here to help you cut the noise when it comes to navigating your learning choices.

We invite you to check out our Data Analyst Nanodegree for a structured program to help you learn all these skills, with the support of Coaches and fellow students:

In the Data Analyst Nanodegree, you'll work your way through five projects designed to teach you data science fundamentals - as you build a portfolio that will demonstrate your new skills to employers. You can think of this skills checklist as a blueprint, and the nanodegree as an action plan.

The nanodegree is a new type of credential designed to prepare you for a career, and it's a big commitment at a minimum of 10 hours a week for 9 to 12 months.

If you are looking for a learning plan with lower time commitment, or if you're looking to fill a specific gap in your skill set, check out our individual courses:

Intro to Data Science - What does a data scientist do? In this course, we will survey the main topics in data science so you can understand the skills that are needed to become a data scientist!

Data Wrangling with MongoDB - Data Scientists spend most of their time cleaning data. In this course, you'll learn to convert and manipulate messy data to extract what you need.

Data Analysis with R - Data is everywhere and so much of it is unexplored. Learn how to investigate and summarize data sets using R and eventually create your own analysis.

Intro to Machine Learning - This class teaches you the end-to-end process of investigating data through a machine learning lens, and you'll apply what you've learned to a real-world data set.

Data Visualization - Learn the fundamentals of data visualization and apply design and narrative concepts to create your own visualization.

If you're looking for even more specialized resources, we've got you covered! Check out these tutorials for individual items from our skill checklist:

R programming language: a special purpose programming language and software environment for statistical computing and graphics (cf. http://www.r-project.org, http://en.wikipedia.org/wiki/R_(programming_language)). Know these R packages:

- ☐ **ggplot2**: a plotting system for R, based on the grammar of graphics
  - ☐ http://ggplot2.org/
- ☐ **dplyr** (or plyr): a set of tools for efficiently manipulating datasets in R (supercedes plyr)
- ☐ **ggally**: a helper to ggplot2, which can combine plots into a plot matrix, includes a parallel coordinate plot function and a function for making a network plot
  - ☐ http://cran.r-project.org/web/packages/GGally/index.html

- ☐ **ggpairs:** another helper to ggplot2, a GGplot2 Matrix
  - ☐ http://www.inside-r.org/packages/cran/GGally/docs/ggpairs
  - ☐ http://cran.r-project.org/web/packages/GGally/GGally.pdf
- ☐ **reshape2**: "Flexibly reshape data: a reboot of the reshape package", using melt and cast
  - ☐ http://cran.r-project.org/web/packages/reshape2/index.html

Python programming language: Python is a high level programming language with many useful packages written for it

- ☐ **Python packages** ("modules")
  - ☐ **numpy**: an optimized python library for numerical analysis, specifically: large, multi-dimensional arrays and matrices. Found in Introduction to Data Science
    - ☐ http://www.numpy.org/
    - ☐ http://en.wikipedia.org/wiki/NumPy
  - ☐ **pandas**: an optimized python library for data analysis including dataframes inspired by R. Found in Introduction to Data Science
    - ☐ http://pandas.pydata.org/
    - ☐ http://en.wikipedia.org/wiki/Pandas_(software)
  - ☐ **matplotlib**: a 2D plotting library for python, includes the pyplot interface which provides a MATLAB-like interface (see ipython notebooks and seaborn below). Found in Introduction to Data Science
    - ☐ http://matplotlib.org/
    - ☐ http://en.wikipedia.org/wiki/Matplotlib
  - ☐ **scipy**: a library for scientific computing and technical computing. Found in Introduction to Data Science
    - ☐ http://www.scipy.org/
    - ☐ http://en.wikipedia.org/wiki/SciPy
  - ☐ **scikit-learn**: machine learning library built on NumPy, SciPy, and matplotlib. Mentioned in Introduction to Machine Learning
    - ☐ http://scikit-learn.org/stable/
    - ☐ http://en.wikipedia.org/wiki/Scikit-learn
  - ☐ optional:
    - ☐ **ipython**: an improved interactive shell for python with introspection, rich media, additional shell syntax, tab completion, and richer history
      - ☐ http://ipython.org/
      - ☐ http://en.wikipedia.org/wiki/IPython

- ☐ **ipython notebooks**: a web-based interactive computational environment
  - ☐ http://ipython.org/notebook.html
  - ☐ http://en.wikipedia.org/wiki/IPython#Notebook
  - ☐ hosting: http://nbviewer.ipython.org/
- ☐ **anaconda**: a python package manager for science, math, engineering, data analysis with the intent of simplifying and maintaining compatibility between library versions. Also useful for getting started with ipython notebooks.
  - ☐ http://continuum.io/downloads
- ☐ **ggplot**: and (in-progress) port of R's ggplot2 which premised upon a grammar of graphics
  - ☐ http://ggplot.yhathq.com
- ☐ **seaborn**: a Python visualization library based on matplotlib with a high-level interface
  - ☐ http://web.stanford.edu/~mwaskom/software/seaborn/

Here are some good data science resources and communities to keep your finger on the pulse of this growing field:

Our good friends

- ☐ The Open Source Data Science Masters
- ☐ Learn Data Science with iPython Notebooks

Books

- ☐ Doing Data Science: Straight Talk from the Frontline
- ☐ Elements of Statistical Learning
- ☐ Pattern Recognition and Machine Learning
- ☐ Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython
- ☐ Data Points: Visualizations That Means Something
- ☐ Interactive Data Visualization for the Web

Newsletters

- ☐ Data Science Weekly

Communities

- ☐ Datatau
- ☐ Cross Validated
- ☐ Reddit Machine Learning Subreddit

Datasets

- ☐ Kaggle Competitions
- ☐ 6 Dataset Lists Curated by Data Scientists

UDACITY