CHECKLIST

Data Career Skills

UDACITY

Welcome

Welcome to your ultimate skills checklist for getting your first data science job! There's never been a better time to pursue a career is this rapidly-growing field, and between your passion for data, and our wide-ranging learning options, you can be on your way to a new job in as little as a few months!

Our data-focused Nanodegree programs are designed to guide you along a project-based curriculum so that you learn the skills you need to succeed in a data career. We have worked closely with industry experts—including both hiring and curriculum partners—to ensure the up-to-the-minute relevance of our curriculum, and this skills list is an actionable distillation of everything you need to know to advance your data career.

In this guide, you'll find the ultimate skills checklist for getting a job as a data analyst or data scientist, as well as other resources to help you along the way.

Congratulations on taking an important step forward towards building a rewarding career in data!!

Data Career Skills Checklist: What We'll Cover

Here's a breakdown of the skills you need to learn to be a data analyst or data scientist. Take some time to review this list—how many boxes can you check off? For more details on these skills, and to discover additional learning resources, navigate to the corresponding pages listed.

Data Analyst Skills Checklist

Programming and Tools 4

- □ Spreadsheets
- □ SQL
- □ R and/or Python (numpy, pandas, matplotlib, scipy)
- □ Jupyter Notebooks

Statistics 6

- □ Descriptive
- □ Inferential
- □ A/B Test Analysis

Mathematics 7

- □ Algebra
- □ Notation

Data Wrangling 8

- □ Data cleaning, blending, transforming, formatting
- □ Relational Databases (SQL)

Communication and Data Visualization 9

- □ Programming (matplotlib, ggplot, etc)
- □ Dashboarding (Tableau, Excel, PowerBI, etc)

Data Intuition 10

- □ Asking the right questions
- □ Becoming subject matter experts
- □ Business and product thinking
- □ PRACTICE!

Additional Skills to Land Your First Data Scientist Job

Machine Learning 11

- □ Supervised Learning
- □ Unsupervised Learning

Software Skills 13

- □ Code testing and debugging
- □ Version control (Git)
- □ Model deployment
- □ Data at Scale (Hadoop or Spark)

Advanced Mathematics 13

- □ Linear Algebra
- □ Calculus

Experiment Design 14

Programming and Tools

Data analysts and scientists use a variety of tools and programming languages in their everyday work. You'll use these tools to query and retrieve data from databases, transform and summarize data, or build machine learning algorithms.

You should be able analyze data in one or more programming languages, and have a good grasp of the landscape of the most commonly used data science libraries and packages. Excel and SQL are a good place to start. For more complicated analysis, Python and R are good programming languages to begin with because of their popularity and community support. Here at Udacity, we heavily favor Python because it's a more general use language, meaning it has a lot more functionality outside of data science.

○ 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.

○ SQL

The majority of company data lives in relational databases, and querying that data using SQL is something data analysts and scientist do everyday. Data science managers often cite insufficient SQL skills as a reason for not hiring a candidate.

○ Data visualization tools (like Tableau)

Most companies have a data visualization tool used for building dashboards to report company performance. Tableau is the most popular, but there are many others with similar capabilities

○ 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, multidimensional 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 Skills to Stand Out

○ 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 (supersedes 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

○ 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
- http://ipython.org/notebook.html
- http://en.wikipedia.org/wiki/IPython#Notebook
- 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.

○ Seaborn

A Python visualization library based on matplotlib with a high-level interface

Statistics

At least a basic understanding of statistics is vital as a data analyst. For example, you may be asked 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 which techniques to use in a given situation.

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.

Descriptive Statistics

○ Mean, median, mode

○ Data distributions

- Standard normal
- Exponential/Poisson
- Binomial
- Chi-square

○ Standard deviation and variance

Inferential Statistics

○ Hypothesis testing

- P-values
- Confidence Intervals

○ Test for significance

- Z-test, t-test, Mann-Whitney U
- Chi-squared and ANOVA testing

○ Regression

- Linear Regression
- Logistic Regression

Mathematics

At a basic level, you should be comfortable with 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 √(1/3g)*

○ Solve for missing values in Algebra equations:
14 = 2x + 29

○ How does the 1/2 value change the shape of this graph?

○ Interpreting mathematical notation:

$$y = \sum_{i=1}^{3} c_i x^i = c_1 x^1 + c_2 x^2 + c_3 x^3$$

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. While not as exciting as building advanced 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
    - String manipulations
    - Parsing common file formats such as csv and xml files
    - Regular Expressions
    - Mathematical transformations, such as converting non-normal distribution to normal with log-10 transformation
- ○ SQL: querying relational databases, such as Oracle, SQL Server, PostgreSQL, or mySQL

Communication and Data Visualization

As a data analyst or data scientist, 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 Tableau, ggplot, matplotlib, and seaborn. 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 and Tools
  - **TABLEAU**
  - **PYTHON:** matplotlib, seaborne
  - **R:** ggplot
- 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

○ Data Storytelling

Data analysts and scientists should know how to present an engaging narrative that empowers the audience to take action. Data analysts should be aware of the type of audience they are presenting to and craft the presentation to that type of audience.

Data Intuition: Thinking like a data scientist

Your manager or co-workers, 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 Standing Out

○ 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.

Data Scientist: Additional Skills and Knowledge

Data Scientist has been deemed the sexiest job of the 21st century, and demand for advanced data skills has never been higher. In addition to all the skills listed above for data analysts, data scientists should have deeper programming, machine learning, statistics, and mathematics knowledge and skills. If you want to become a data scientist, here's what else you should know!

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. As a data analyst, you likely won't use machine learning beyond regression, but data scientists should master a variety of machine learning techniques. 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

○ Optional Machine Learning Skills to Stand Out

- **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

- **Deep Learning**

  Deep learning is a type of machine learning that uses a cascade of multiple layers to contribute to feature extraction and transformation. Deep learning can be applied to both supervised and unsupervised ML algorithms and has been transformational in the fields of computer vision, natural language process, robotics, and many others.

  - Neural Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Generative Adversarial Networks

Software Skills

Data scientists should be strong programmers, and should be able to debug and test complex programs. This is particularly important if you're interviewing at a smaller company. You'll be responsible for handling a lot of data logging, and potentially the development of data-driven products.

- Debugging
- Testing
- Version control (Git)
- SQL: editing and updating relational databases

○ Optional Skills to Stand Out

- Spark or Hadoop: languages for working with data at scale
- NoSQL database systems, such as MongoDB

Advanced Math

A solid grasp of multivariable calculus and linear algebra help data scientists understand statistical concepts at a deeper level. 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

Experiment 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
- Confidence level
- SMART experiments: Specific, Measurable, Actionable, Realistic, Timely
- Bayesian Statistics
- Bootstrapping
- Simulation

What Next? Learning Resources

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

Whether you're starting at the beginning in terms of checking off boxes, or already possess many of the skills already, you can certainly pat yourself on the back for taking a big step forward by reading this guide!

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

We invite you to check out Udacity's Nanodegree programs for a structured way to help you learn all these skills, with the support of experts, mentors, and fellow students. You can think of this skills checklist as a blueprint, and the Nanodegree programs as an action plan. Whether you're a beginner, or already have some data analysis experience, Udacity has programs that will enable you to grow your data skills, and secure rewarding employment in the field.

Stay on Top of the Industry

Here are some great weekly newsletters to help you stay on top of industry trends, events, and new. We suggest picking two and reading them every week.

- The Analytics Dispatch
  The Analytics Dispatch is a weekly email about data, data science, and analytics curated by the team at Mode Analytics, an enterprise analytics company and a Udacity partner!
- Data Elixir
  Curated by ex-NASA data scientist Lon Riesberg, Data Elixir is a weekly newsletter of curated data science news and resources from around the web.
- O'Reilly Data Newsletter
  O'Reilly's network of experts and innovators share their knowledge and expertise on the in a variety of topic areas related to technology. O'Reilly has a great blog, weekly newsletter, and podcast that discuss a variety of interesting topics in big data and data science.

- Data Science Weekly

  Data Science Weekly, curated by Hannah Brooks and Sebastian Gutierrez, shares recent news, articles and jobs related to Data Science. You can sign up for the newsletter, or enjoy archives that date back to 2013.

- Analytics Vidhya

  Analytics Vidhya is an online community, created by Kunal Jain, dedicated to the study of analytics. Its resources include training, tips and tricks, case studies, and interviews with analytics leaders.

Open Data Sources

- data.world

  data.world is designed for data and the people who work with data. From professional projects to open data, data.world helps you host and share your data, collaborate with others, and capture context and conclusions as you work. You can post data, access data through an online workspace, and see what other people have done with it. It a great place to go to start building your data science portfolio

- kaggle

  Known for their machine learning competition, kaggle has a vast amount of open source datasets that you can use to practice and build out your portfolio. Also, you can try your hand at the machine learning competitions on actual company datasets and problems.

○ Other Sources

Here are a few collections of data sources that you can also use as you look for interesting datasets to use for your portfolio projects.

- Analytics Vidhya: 25+ websites to find datasets for data science projects
- Dataquest: 18 places to find data for data science projects
- KDnuggets: Datasets for Data Mining and Data Science

More Focus Learning Resources

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/