



NANODEGREE PROGRAM SYLLABUS

# Programming for Data Science with R



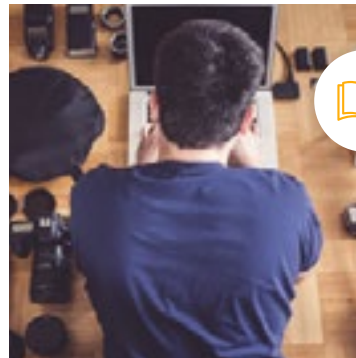
# Overview

Learn the programming fundamentals required for a career in data science. By the end of the program, you will be able to use R, SQL, Command Line, and Git.

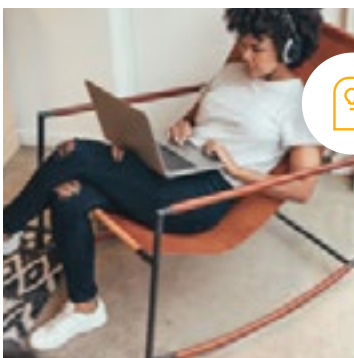
IN COLLABORATION WITH



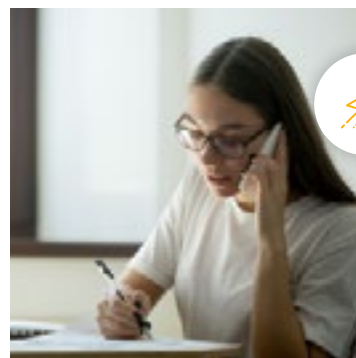
**Estimated Time:**  
3 Months at  
10hrs/week



**Prerequisites:**  
None



**Flexible Learning:**  
Self-paced, so  
you can learn on  
the schedule that  
works best for you



**Need Help?**  
[udacity.com/advisor](https://www.udacity.com/advisor)  
Discuss this program  
with an enrollment  
advisor.

# Course 1: Introduction to SQL

Learn SQL fundamentals such as JOINS, Aggregations, and Subqueries. Learn how to use SQL to answer complex business problems.

## Course Project

### Investigate a Database

In this project, you'll work with a relational database while working with PostgreSQL. You'll complete the entire data analysis process, starting by posing a question, running appropriate SQL queries to answer your questions and finishing by sharing your findings.

## LEARNING OUTCOMES

### LESSON ONE

#### Basic SQL

- Write common SQL commands including SELECT, FROM, and WHERE
- Use logical operators like LIKE, AND, and OR

### LESSON TWO

#### SQL Joins

- Write JOINS in SQL, as you are now able to combine data from multiple sources to answer more complex business questions
- Understand different types of JOINS and when to use each type

### LESSON THREE

#### SQL Aggregations

- Write common aggregations in SQL including COUNT, SUM, MIN, and MAX
- Write CASE and DATE functions, as well as work with NULLs

### LESSON FOUR

#### Advanced SQL Queries

- Use subqueries, also called CTEs, in a number of different situations
- Use other window functions including RANK, NTILE, LAG, LEAD new functions along with partitions to complete complex tasks

# Course 2: Introduction to R Programming

In this part, you'll learn to represent and store data using R data types and variables, and use conditionals and loops to control the flow of your programs. You'll harness the power of complex data structures like lists, sets, dictionaries, and tuples to store collections of related data. You'll define and document your own custom functions, write scripts, and handle errors. You will also learn to use two powerful R libraries - Numpy, a scientific computing package, and Pandas, a data manipulation package.

## Course Project : Explore US Bikeshare Data

You will use R to answer interesting questions about bikeshare trip data collected from three US cities. You will write code to collect the data, compute descriptive statistics, and create an interactive experience in the terminal that presents the answers to your questions.

### LEARNING OUTCOMES

#### LESSON ONE

##### Introduction to R

- Understand common use cases of R and why it's popular
- Install & Setup R Environment
- Learn with basic syntax associated with R
- Understand how you can get help when writing R code

#### LESSON TWO

##### Syntax & Data Types

- Work with data structures available in R including scalars, factors, vectors arrays, lists, and dataframes
- Manipulate, compare, and perform fundamental operations associated with each of the data structures

#### LESSON THREE

##### Control Flow & Functions

- Write conditional expressions using if statements and boolean expressions
- Use loops and other built-in functions to iterate over and manipulate data
- Define your own custom functions

#### LESSON FOUR

##### Data Visualizations & EDA

- Make beautiful visualizations using the ggplot2 library
- Create commonly used data visualizations for each data type including histograms, scatter plots, and box plots
- Improve your data visualizations using facets
- Create reference variables using appropriate scope
- Use the popular diamonds dataset to put your R skills to work

# Course 3: Introduction to Version Control

Learn how to use version control and share your work with other people in the data science industry.

## Course Project : Post your work on Github

In this project, you will learn important tools that all programmers use. First, you'll get an introduction to working in the terminal. Next, you'll learn to use git and Github to manage versions of a program and collaborate with others on programming projects. In this project you will add a completed project on GitHub, work with branches, edit a README file and project files, merge branches, stage and commit your changes to your project GitHub repository.

### LEARNING OUTCOMES

#### LESSON ONE

##### Shell Workshop

- The Unix shell is a powerful tool for developers of all sorts. Get a quick introduction to the basics of using it on your computer.

#### LESSON TWO

##### Purpose & Terminology

- Learn why developers use version control and discover ways you use version control in your daily life
- Get an overview of essential Git vocabulary
- Configure Git using the command line

#### LESSON THREE

##### Create a Git Repo

- Create your first Git repository with git init
- Copy an existing Git repository with git clone
- Review the current state of a repository with the powerful git status

#### LESSON FOUR

##### Review a Repo's History

- Review a repo's commit history git log
- Customize git log's output using command line flags in order to reveal more (or less) information about each commit
- Use the git show command to display just one commit

**LEARNING OUTCOMES****LESSON FIVE****Add commits to a Repo**

- Master the Git workflow and make commits to an example project
- Use git diff to identify parts of a file that changed in a commit
- Learn how to mark files as “untracked” using .gitignore

**LESSON SIX****Tagging, Branching, and Merging**

- Tagging, Branching, and Merging
- Organize your commits with tags and branches
- Jump to particular tags and branches using git checkout
- Learn how to merge together changes on different branches and crush those pesky merge conflicts

**LESSON SEVEN****Undoing Changes**

- Learn how and when to edit or delete an existing commit
- Use git commit's --amend flag to alter the last commit
- Use git reset and git revert to undo and erase commits

**LESSON EIGHT****Working with Remotes**

- Create remote repositories on GitHub
- Learn how to pull and push changes to the remote repo

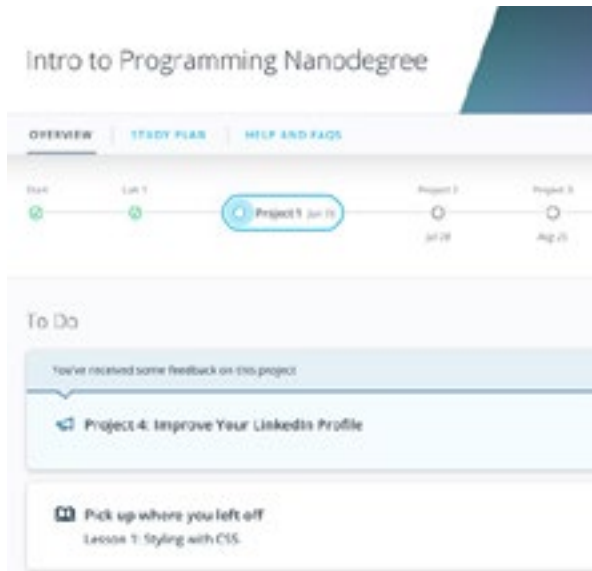
**LESSON NINE****Working on Another Repository**

- Learn how to fork another developer's project
- Use GitHub to contribute to a public project

**LESSON TEN****Staying in Sync with a Remote Repository**

- Learn how to sync new changes to a forked remote repository, retrieve and sync updates
- Create pull requests and squash commits with git rebase

# Our Classroom Experience



## REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

## KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

## STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

## WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

## QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

## CUSTOM STUDY PLANS

Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

## PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

## Learn with the Best



### Josh Bernhard

DATA SCIENTIST

Josh has been sharing his passion for data for nearly a decade at all levels of university, and as Lead Data Science Instructor at Galvanize. He's used data science for work ranging from cancer research to process automation.



### Juno Lee

DATA SCIENCE INSTRUCTOR

As a data scientist, Juno built a recommendation engine to personalize online shopping experiences, computer vision and natural language processing models to analyze product data, and tools to generate insight into user behavior.



### Derek Steer

CEO AT MODE

Derek is the CEO of Mode Analytics. He developed an analytical foundation at Facebook and Yammer and is passionate about sharing it with future analysts. He authored SQL School and is a mentor at Insight Data Science..



### Richard Kalehoff

INSTRUCTOR

Richard is a Course Developer with a passion for teaching. He has a degree in computer science, and first worked for a nonprofit doing everything from front end web development, to backend programming, to database and server management.



## Learn with the Best



### Karl Krueger

INSTRUCTOR

Before joining Udacity, Karl was a Site Reliability Engineer (SRE) at Google for eight years, building automation and monitoring to keep the world's busiest web services online.

# All Our Nanodegree Programs Include:



## EXPERIENCED PROJECT REVIEWERS

### REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve



## TECHNICAL MENTOR SUPPORT

### MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions



## PERSONAL CAREER SERVICES

### CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

# Frequently Asked Questions

## PROGRAM OVERVIEW

### WHY SHOULD I ENROLL?

The Programming for Data Science Nanodegree program offers you the opportunity to learn the most important programming languages used by data scientists today. Get your start into the fascinating field of data science and learn R, SQL, terminal, and git with the help of experienced instructors.

### WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This is an introductory program that is not designed to prepare you for a specific job. However, as a graduate of this program, you will be proficient in the programming skills used in many data analysis and data science roles, including R, SQL, Terminal, and Git.

### HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you are interested in taking the first step into the field of Data Science this course is for you. This course will quickly teach you the foundational data science programming tools (R, SQL, Git). This course requires no pre-requisite knowledge so you can get started now. Having mastered these in demand tools, you will be able to tackle real world data analysis problems.

Learning to program R and SQL, the main programming languages used by data scientists and analysts, is the core of this program. By learning these foundational programming skills, you will be ready to advance your career in data.

### WHAT IS THE DIFFERENT BETWEEN THE PYTHON AND THE R TRACK?

We offer two tracks to this program, one teaching Python and one teaching R. These are both popular among data scientists. You can enroll in one or the other, not both.

Both tracks cover the same fundamental concepts, but use a different programming language. The SQL, command line, and Git curriculum is the same in both tracks. This includes the first and third projects, which are the same between the two tracks.

The programming course and project are different between the two tracks. One course relies on Python, while the other relies on R. The projects for the two courses rely on the same dataset and skills, but they differ in the approach and final deliverable. [Learn more about the Programming for Data Science with Python Nanodegree](#) program.

### WHAT IS THE SCHOOL OF DATA SCIENCE, AND HOW DO I KNOW WHICH PROGRAM TO CHOOSE?

Udacity's School of Data consists of several different Nanodegree programs, each of which offers the opportunity to build data skills, and advance your



## FAQs Continued

career. These programs are organized around career roles like Business Analyst, Data Analyst, Data Scientist, and Data Engineer.

The School of Data currently offers three clearly-defined career paths in Business Analytics, Data Science, and Data Engineering. Whether you are just getting started in data, are looking to augment your existing skill set with in-demand data skills, or intend to pursue advanced studies and career roles, Udacity's School of Data has the right path for you! Visit [How to Choose the Data Science Program That's Right for You](#) to learn more.

### ENROLLMENT AND ADMISSION

#### DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

#### WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

There are no prerequisites for enrolling aside from basic computer skills and English proficiency. You should feel comfortable performing basic operations on your computer like opening files and folders, opening applications, and copying and pasting.

### TUITION AND TERM OF PROGRAM

#### HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Programming for Data Science with R Nanodegree program is comprised of content and curriculum to support three (3) projects. We estimate that students can complete the program in three (3) months, working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided, and if you do not pass the project, you will be asked to resubmit the project until it passes.

#### HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the [Terms of Use](#) and [FAQs](#) for other policies regarding the terms of access to our Nanodegree programs.



## FAQs Continued

### **CAN I SWITCH MY START DATE? CAN I GET A REFUND?**

Please see the Udacity Nanodegree program [FAQs](#) for policies on enrollment in our programs.

### **I HAVE GRADUATED FROM THE PROGRAMMING FOR DATA SCIENCE WITH R NANODEGREE PROGRAM AND I WANT TO KEEP LEARNING. WHERE SHOULD I GO FROM HERE?**

Check out our [Data Analyst Nanodegree](#) program to take the programming skills you have learned and apply them to real world Data Analyst business problems. Working on these more complex projects will deepen your knowledge of coding and make you a more attractive candidate to be hired as an data analyst.

You could also consider the [Data Engineer Nanodegree](#) program, which focuses on data models, data warehouses, and data pipelines.

### SOFTWARE AND HARDWARE

#### **WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?**

For this Nanodegree program, you will need a 64-bit computer and access to the Internet.

