Insight Interview Prep Guide

*This guide is a curated list of topics found in a majority of past interviews. Interviews typically cover a wide range of these topics but then will go in depth for those that are the most relevant for the specific role. These topics are split into three sections: basic technical skills, applying those skills to answer open ended business questions, and cultural fit. The most commonly emphasized topics have been* ***bolded****.*

*Understanding the material is only a small portion of the interview. Effectively communicating your knowledge and how you would contribute to the business goals is a critical portion of the interview. Previous fellows have found that practice interviews, whiteboarding, and data challenges are the best way to prepare for real interviews.*

[Textbook Questions](#_1fob9te)

[SQL](#_2et92p0)

[CS Fundamentals](#_3dy6vkm)

[Machine Learning](#_4d34og8)

[Stats and Probability](#_17dp8vu)

[Business Case Studies](#_26in1rg)

[Translate Business Problem to Data Problem](#_35nkun2)

[Actionable Insight from Data](#_44sinio)

[Design a Data Product](#_z337ya)

[Build a Machine Learning Algorithm](#_1y810tw)

[Design an Experiment](#_2xcytpi)

[Lingo/Topics to know](#_3whwml4)

[Tools to use](#_qsh70q)

[Cultural Questions](#_1pxezwc)

# Textbook Questions

*Communication is key during these questions. Make sure you:*

* ***Ask questions to clarify*** */ Make sure you understand the question*
* *Explain answers at a high level (algorithm) before going into details (coding)*
* *Get something functional first then iterate*
* *Are very comfortable whiteboarding*
* *Keep interviewers engaged even while whiteboarding*

***Remember that it’s more about how you think and reason than getting the right answer***

## **SQL**

*Resources*

* [*Mode SQL school*](http://sqlschool.modeanalytics.com/) ; [*SQLZoo*](http://sqlzoo.net/wiki/Main_Page) *;* [*Stanford Lagunita course*](https://lagunita.stanford.edu/courses/DB/SQL/SelfPaced/about) *for learning/practice*
* [*10 steps to a complete understanding of SQL*](https://blog.jooq.org/2016/03/17/10-easy-steps-to-a-complete-understanding-of-sql/) *and the brief* [*command listing at w3schools*](http://www.w3schools.com/sql/) *for syntax.*
* [*Leetcode*](https://oj.leetcode.com/) *and* [*Hackerrank*](https://www.hackerrank.com/) *for practice interview questions*

Topics

* Main Query control
  + **SELECT**
    - Nested
  + **AS**
  + **JOIN**
    - [Venn diagram visualization](https://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins)
    - Self joins
  + **WHERE**
  + **GROUP BY**
    - **Aggregate functions** (MIN/MAX, AVG, COUNT, FIRST/LAST)
  + **HAVING**
    - Difference with GROUP BY
  + **ORDER BY**
  + LIMIT
* Other functions
  + Window Functions
  + **IF**
  + **CASE**
  + COALESCE
  + PIVOT
  + UNION, UNION ALL
* Style
  + Indentation is important. Write on whiteboard with proper indentation and you’ll show you write clean SQL.
  + Capitalize keywords. Table names or variables, lowercase.
* Pros and cons of MySQL, PostgreSQL, SQLite, Oracle, NoSQL, etc.
  + [Resource for relational DB comparisons](https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems)
* Basic principles of MapReduce/Hadoop/[Hive](https://cwiki.apache.org/confluence/display/Hive/LanguageManual)

Example problems:

* Given some table(s), be able to whiteboard solutions to any type of SQL query that involves the above functions, e.g. “How many unique users visited the site in August?”
* Be able to translate Pandas functions into SQL (and vice versa!)
* “Explain the difference between UNION and UNION ALL”
* “How do you get the second-largest value from a column of a table that may have repeated values?”
* “Given a table on Facebook posts, comments, etc., make a frequency distribution of all the posts that have comments”

## **CS Fundamentals**

*Resources*

* [*MIT OCW videos*](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/video-lectures/)
* *‘*[*Problem Solving with Algorithms and Data Structures in Python*](http://interactivepython.org/runestone/static/pythonds/index.html)*’*
* [*https://www.geeksforgeeks.org/data-structures/*](https://www.geeksforgeeks.org/data-structures/)
* *Cracking the Coding Interview (Library)*
* *Programming Interviews Exposed (*[*Dropbox library*](https://www.dropbox.com/sh/qc5is7h0sr5nfp3/AAAQUInUTmuL-zoghCaeVK_ha?dl=0&lst=)*)*
* [*Leetcode*](https://oj.leetcode.com/) *and* [*Hackerrank*](https://www.hackerrank.com/) *for practice interview questions*

Topics

* **Pandas in Python**
  + Be able to duplicate all SQL functions
* O(N) notation/Time complexity of data structures and algorithms
* Data structures
  + [Hash table](http://interactivepython.org/runestone/static/pythonds/SortSearch/searching.html) ([on coursera](https://class.coursera.org/algo-004/lecture/preview)): hash functions
  + Stack/Queue/Deque
  + [Linked list](http://interactivepython.org/courselib/static/pythonds/BasicDS/ImplementinganUnorderedListLinkedLists.html) (O(n) vs O(1) implementations)
  + [Heap](http://interactivepython.org/runestone/static/pythonds/Trees/heap.html): constructs in O(n) time
  + Treesand Binary search trees (BST)
* [Recursion](http://interactivepython.org/runestone/static/pythonds/index.html#recursion) and [Master theorem](http://en.wikipedia.org/wiki/Master_theorem)
* Algorithms: *see* [*coursera Algorithms course*](https://class.coursera.org/algs4partI-003/)
  + Sorting algorithms: Quick, Merge, etc.
  + Graph algorithms: depth first vs breadth first, Dijkstra’s
  + [**Dynamic programming**](https://class.coursera.org/algo2-2012-001/lecture/preview)**:** memoization, knapsack problem, Fibonacci sequence
* Math brain teasers
  + Combinatorics
    - e.g. n choose k, two eggs problem, five fastest horses
  + recurrence relations
  + last chapter of coding interview questions book

Example problems:

* Traverse a binary search tree in order without using recursion
* Write a function which takes a base-10 number and a positive integer k > 1 and prints out the number in base k.
* Compute the median of a set of numbers which is too large to fit into memory (Hint: use a histogram)
* Solve the 0-1 (as opposed to unbounded) knapsack problem
* Given an array of numbers find the subarray which has the greatest sum in linear time.
* reverse a string in-place (without consuming any extra memory)
* Implement left-join in Python
* find the palindromes (substrings which read the same forward and backward) in a given string

## **Machine Learning**

*Resources*

* [*Stanford CS 229 Machine Learning (taught by Andrew Ng)*](http://cs229.stanford.edu/materials.html)
* [*CMU/10-701*](http://alex.smola.org/teaching/cmu2013-10-701x/index.html)
* [*shape of data blog*](http://shapeofdata.wordpress.com/)
* [*LearnDataScience on github*](http://nborwankar.github.io/LearnDataScience/)
* *Algorithm cheat sheets:* [*scikit learn*](http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html)

Topics

* What is machine learning?(in your own words)
* **Know the pros, cons, and assumptions of each algorithm**
* Supervised Learning
  + **Linear Regression** (with derivation)
  + **Logistic Regression** (with derivation)
  + **Decision Trees**
  + **Random Forest**, Gradient Boosted Trees
  + SVM
  + Naive Bayes
  + Neural Nets (basic concepts)
* Unsupervised Learning
  + **Clustering - K-means**
  + SVD, [**PCA**](http://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/)
  + Kernel density estimation
  + Gaussian mixture models
  + Outlier detection (one class SVM)
  + Latent Dirichlet Allocation (LDA)
* Recommendation Systems
  + Collaborative Filtering (user-user, item-item)
  + Validation methods
* Building ML systems
  + **Model selection**
  + **Feature selection**
  + **Regularization and overfitting**
  + **Validation and testing schemes (k-fold, stratified k-fold, CV, …)**
    - **Metrics (F1, precision, recall)**
  + Dealing with missing data (imputation, robust models)
  + Imbalanced data
  + Optimization schemes: Gradient Descent, Conjugate Gradient, BFGS, etc.
  + Evaluating model success (accuracy, precision/recall, ROC curve)
  + Combining models - Ensembles, Boosting

Example problems:

* Talk about models that can do nonlinear decision boundaries for classification problems
* Explain regularization. Difference between L1 and L2 and when they would be used.
* Talk about ways to reduce the dimensionality of a dataset and discuss the pros and cons

## **Stats and Probability**

*Resources*

* [***Khan Academy***](https://www.khanacademy.org/math/probability)
* [*openintro stats*](http://www.openintro.org/stat/down/OpenIntroStatSecond.pdf)
* [*statistics done wrong*](http://www.refsmmat.com/statistics/)
* *Introduction to Statistical Learning (Library)*

Topics

* **Probability and** [**Bayes theorem**](http://georgemdallas.wordpress.com/2013/07/13/how-to-build-an-anti-aircraft-missile-probability-bayes-theorem-and-the-kalman-filter/)
  + Central limit theorem and law of large numbers
  + Monty Hall problem
  + Frequentist vs Bayesian viewpoints
* **Modeling regressions**
  + multiple, logistic
  + Feature selection
* **Descriptive statistics**
  + Mean, median, stdev, variance
  + Confidence intervals, p values
* **Hypothesis testing**
  + **T-test**
  + **ANOVA**
  + chi square: test for normality
  + R2 (Pearson’s correlation coefficient, squared)
  + Kolmogorov-Smirnov tests and QQ plots
* **Experimentation/inference**
  + Power analysis
  + Bias/variance
  + A/B Testing, A/A testing
  + Sampling methods (random, stratified, etc.)
  + Multiple Comparisons Correction
    - [Bonferroni correction](http://en.wikipedia.org/wiki/Bonferroni_correction)
  + Metrics
    - Type 1, type 2 errors
* Resampling
  + bootstrap (random sampling), bootstrap for confidence interval on median
  + jack-knife (all possible subsets)

Sample questions:

* Design a uniform random number generator between 1-10 using two 6-sided dice.
* What is the expectation value of days until an individual is chosen for something if we have a total of 1000 people and we add new people to the pool every day?
* Can you explain what hypothesis testing means? What’s a p value?
* How do you know if the data you have is appropriate to model in a regression? How do you test it? How do you know if your data is normally distributed? What do you do if it’s not? How do you test if your data is distributed the same as another system?

# Business Case Studies

*These questions make up a MAJOR portion of interviews. They are very open ended and test both your product sense and how you work as a data scientist on a high level. Because of their open ended nature, the listed questions are* ***not meant to be comprehensive****, but rather a starting point. A company deep dive is absolutely necessary to perform well on these questions.*

A typical outline of these questions follows:

1. Translate Business Problem to Data Problem
2. Solve the Data Problem and Relate Back to Business (using one or more of the following methods)
   1. Actionable Insight from Data
   2. Design a Data Product
   3. Build a Machine Learning Algorithm
   4. Design an Experiment

## **Translate Business Problem to Data Problem**

* What is the goal of the company?
  + What is the company business model?
  + Who are the customers?
    - How are they acquired? Retained?
  + What is the use case of the product? What is the key value add?
  + What are the key metrics for success?
  + What are the pain points?
* What is the data like?
  + How was it collected?
  + What biases & caveats are there?
  + What information is available?
    - What kind of schema would it use?
  + How is/can this data be used for the product?
    - How can it address the pain points?
  + What new data (internal/external) would be useful to obtain?

## **Actionable Insight from Data**

*Check out the Lean Analytics Book in* [*Dropbox*](https://www.dropbox.com/sh/qc5is7h0sr5nfp3/AAAQUInUTmuL-zoghCaeVK_ha?dl=0&lst=)

* How can you analyze the data?
  + Simple first pass
    - Visual trends / outliers
    - Counts / Group By / Histograms
  + More advanced
    - Regression relationships
    - Unsupervised Clustering
* What conclusions/relationships can you find from a graph?
* What actions/decisions are made as a result?
* How can you measure success/impact of those actions/decisions?

## **Design a Data Product**

* What is the current platform?
  + Can you picture / describe the current interface?
  + What data products are up and running and how do they work?
  + Why were they build and how have they helped the bottom line?
* What would you change? Completely new product? Changes to existing product?
  + How would you build out new/changed product?
  + Which algorithm? Why?
  + Which pain points are addressed?
* How can you evaluate success?
  + Design experiment
  + Bring it back to a key metric and/or company business model

## **Build a Machine Learning Algorithm**

* Classification or Regression? Clustering? On what?
* What data is available or would you ideally like to have?
  + What biases exist from the data sources?
  + **Feature engineering**
    - Define precisely (Number of Facebook friends instead of “popularity”)
    - **Normalization** (How exactly?)
  + Feature selection / importance
* Which ML Algorithm?
  + **Why?** (Remember all the pros and cons of each of your options)
  + How does the ML algorithm work?
  + What caveats exist and how will it scale?
    - Hyperparameters
* How can you evaluate success and iterate?
  + Cross validation and testing
  + Grid search

## **Design an Experiment**

* What is the hypothesis to be tested?
* What are the controls?
* Is the experiment worth running in the first place?
* How exactly will the experiment be run?
  + What effect size is needed for us to make decision (power analysis!)?
  + How many samples are needed to see the effect size as significant?
  + Would you peek at the data? In which cases?
    - How does this affect the experiment?
  + When/how would you stop?
* What if you need to stop early?
* What would the results look like?
* What biases do you need to worry about?
  + How do you correct for them?
  + What caveats are in place for those that you can’t correct for?
* What are the action items for each possible result of the experiment?
* What can you do if the experiment “fails”?

## **Lingo/Topics to know**

* Churn, retention
  + Survival Analysis, Customer lifetime value (LTV)
  + Engagement, DAU (daily active users), MAU
* **Funnel analysis**
* KPI (Key Performance Indicators)
* User segmentation
* B2B, SaaS, two sided marketplace
* Lift
* Pain Points
* Cohort analysis
* Correlation does not imply causation!
* A/B testing - [Udacity’s course](https://www.udacity.com/course/ab-testing--ud257)
  + A/A testing
  + Power analysis

## **Tools to use**

*This is a list of tools/languages that you can use for take home or on site data challenges. Sometimes, specific languages/packages will be asked for. Make sure you’re familiar with the* ***tools relevant to the roles*** *you’re interviewing for.*

* Python
  + if you don’t know it yet, then [learn python the hard way](http://learnpythonthehardway.org/book/)
  + ipython/Jupyter, ipython/Jupyter notebook
  + pip: install, freeze
  + Packages:
    - numerics: numpy, scipy, [pandas](http://pandas.pydata.org/pandas-docs/stable/10min.html) (see also Wes McKinney lectures [1](http://www.youtube.com/watch?v=w26x-z-BdWQ) & [2](http://www.youtube.com/watch?v=0unf-C-pBYE))
    - plotting: pylab/matplotlib, [altair](https://github.com/ellisonbg/altair), [daft](http://daft-pgm.org/) for probabilistic graphical models
    - machine learning: scikit-learn
    - [data science toolkit (dstk)](http://www.datasciencetoolkit.org/)
    - database/storage: [dataset](https://dataset.readthedocs.org/en/latest/), MySQLdb, sqlite, shove, shelve, pickle/cPickle
    - web: urllib/httplib, requests, scrapy
  + Clean code/style: Use pep8/autopep8/pylint/pyflakes or the Google python style guide
* [R](http://www.r-project.org/) for statistical analysis
  + plotting: ggplot2, lattice
  + SQL-like data manipuluation: dplyr
  + [Rpy](http://scienceoss.com/rpy-statistics-in-r-from-python/)
* Web Development
  + [Flask](http://flask.pocoo.org/)/[gunicorn](http://gunicorn.org/)/[Twistd](http://flask.pocoo.org/docs/deploying/wsgi-standalone/), HTML/CSS, Javascript
  + Handy modules/scripts: Twitter [bootstrap](http://getbootstrap.com/)/[typeahead](http://twitter.github.io/typeahead.js/), [jQuery](http://jquery.com/)/[React](http://facebook.github.io/react/), [purecss](http://purecss.io), [d3.js](http://d3js.org/), [moment.js](http://momentjs.com/), [leaflet.js](http://leafletjs.com/), meteor.js/angular.js/ember.js/backbone.js
  + Amazon Web Services (AWS) esp. [EC2 elastic compute cloud](http://aws.amazon.com/ec2/) free tier
  + DNS reg: use [namecheap.com](http://www.namecheap.com/) or similar, prior bad experience w/godaddy
* Presentation sharing: [Google slides, slid.es](http://slid.es/), deck.js, remark
* Version control -- git/github:
  + commands: init, commit, push, branch, clone, pull, merge, status, diff
  + basics of .gitconfig and .gitignore

# Cultural Questions

*Every interaction (email/phone/onsite) is a chance for* ***both*** *the interviewers and you to evaluate whether you all want to spend 40+ hours each week together. Once you pass the technical bar, cultural fit is the last thing that can determine whether or not you receive the offer. These questions range from being focused on you to being focused on the company.*

* Tell me about your background, your academic research
  + Prepare to answer questions for any part of your demo and ***resume***
* Tell me about a time when… (STAR - Situation, Task, Action, Result) - Practice these!
  + Didn’t get along with a coworker
  + Overcame a problem
* Why data science? (this is not an opportunity to shit on academia!)
* What roles are you interested in?
* What is your preferred workflow? Preferred management style?
* Why are you interested in this company? Why do you want to work here?
* **What are you going to bring to the company?** (Talk about this even if not asked!)
  + Read first couple of chapters of [Sell Yourself in any Interview, by Oscar Adler](http://www.amazon.com/Sell-Yourself-Any-Interview-Techniques/dp/0071549099) (on bookshelf)!
* **Come prepared with questions about the company that you care about!**
  + MAKE A LIST BEFOREHAND
  + **Having no questions implies a lack of interest**
  + Lookup roles of interviewers beforehand