QUANT TRADING & RESEARCH

Quantitative Finance Interview Playbook

Applications, projects & technical preparation for quant trading and research.

- Application evidence
- 40 worked problems
- Trading & research cases
- Two complete mock rounds

WALLSTREETPLAYBOOK.ORG 42 PAGES · SEPTEMBER 2026

Contents

Start here: a route through the book

Prepare to explain what you can do, then demonstrate it under questions. A strong application makes your work easy to assess. A strong interview answer makes your reasoning easy to follow.

This playbook serves candidates preparing for entry-level quantitative trading and quantitative research roles. You should be comfortable with algebra, introductory probability and basic Python. Research candidates will benefit from introductory statistics, calculus and linear algebra. If conditional probability or a Python loop is unfamiliar, use the prerequisite resources in the final chapter before attempting a timed mock.

The book has two tracks because the jobs overlap without being interchangeable. Trading preparation emphasizes decisions under uncertainty, numerical fluency, conditional values and risk. Research preparation emphasizes inference, model evaluation, data timing, coding and the defense of a research choice. A particular team may test both. Read its current posting and follow the recruiter's instructions.

Applications, projects and interview preparation belong together. A project you cannot explain is weak application evidence; interview preparation that never reaches a relevant employer cannot produce an interview. The first chapters connect your target role to work you can show, without promising that a project, referral or school will get you hired.

What this edition contains

Original practice problems with worked solutions, an application evidence matrix, annotated resume examples, a trading simulation case, a reproducible research audit, two mock rounds and a preparation plan. All candidate stories, datasets, prices and trading situations are fictional teaching examples. None is a customer result or a question attributed to an employer. The numerical cases simplify real markets deliberately.

This is an independent publication by Wall Street Playbook. No employer endorsement, proprietary question access or placement guarantee is implied. It is not a complete course in stochastic calculus, derivatives pricing, machine learning or low-latency C++ development. Those specialisms require additional preparation when a posting asks for them. The code uses Python's standard library unless stated otherwise; a PDF is not an executable notebook.

YOUR IMMEDIATE PROBLEM	START WITH	PRODUCE
I am not getting interviews	Role selection, evidence, projects and applications	A role-specific application and one inspectable project
I have a trading interview	Diagnostic, probability, numerical reasoning and trading case	A spoken solution and a decision log

YOUR IMMEDIATE PROBLEM	START WITH	PRODUCE
I have a research interview	Diagnostic, inference, modeling, Python and research case	A reproducible analysis and a defense of its limitations
I solve familiar questions only	Mock rounds and follow-up prompts	A corrected first attempt, then a fresh variation

How to practice

Attempt a prompt before reading the solution. Write the assumption that makes your answer valid. Say the reasoning aloud. After checking, label the error: misunderstood question, missing concept, arithmetic, coding, unsupported inference or communication. Rework the failed step on a different example. Re-reading a correct answer is useful review, but it does not establish that you can produce one.

Employer descriptions were checked on September 28, 2026. They illustrate preparation priorities, not a universal recruiting calendar. Check current eligibility, location, degree requirements, assessment rules and deadlines on the actual role before applying. This guide does not claim to be a 2028 recruiting edition.

1. A first problem: when a good signal misleads

A useful quant answer separates the information observed from the thing being estimated. Start with this complete example before choosing a track.

Prompt

A fictional feed is in a degraded state on 10% of days. An alert fires on 80% of degraded days and 20% of normal days. Today's alert fires. What is the probability that the feed is degraded? You may purchase a diagnostic that reveals the state perfectly for 0.5 units. Pausing a normal feed costs 2 units; leaving a degraded feed active costs 12 units. Pausing a degraded feed and leaving a normal feed active cost zero. What should you do after the alert?

State whether the percentages describe the same population and whether the cost is incurred once for this decision. For the exercise, assume both are true. The diagnostic has no delay or other cost. These assumptions are part of the model, not facts about production systems.

Solution: use counts before symbols

Imagine 1,000 days. There are 100 degraded days, of which 80 trigger alerts. There are 900 normal days, of which 180 trigger alerts. Among 260 alerts, only 80 come from degraded days. The required probability is 80/260 = 4/13, approximately 30.77%. The 80% alert sensitivity answers a different question.

```
P(degraded | alert) = (0.10 × 0.80) / (0.10 × 0.80 + 0.90 × 0.20) = 4/13.
```

Without a diagnostic, keeping the feed active has expected loss 12 × 4/13 = 48/13, approximately 3.69. Pausing has expected loss 2 × 9/13 = 18/13, approximately 1.38. Pause is the lower expected-loss action. A perfect diagnostic lets you choose the zero-loss action in each state, so its expected loss is its price, 0.5. Buy it under the stated assumptions.

The value of perfect information here is 18/13, the expected loss of the best uninformed action minus the zero expected loss with perfect information. It is not 48/13: that comparison uses an action you already know is inferior. The diagnostic creates net expected benefit 18/13 - 0.5 = 23/26, approximately 0.885.

Follow-up: change the base rate

If degradation occurs on only 1% of days, the posterior becomes 0.008 / (0.008 + 0.198) = 4/103, approximately 3.88%. Keeping the feed active now costs 48/103, approximately 0.466, in expectation; pausing costs 198/103, approximately 1.922. Keep it active. A perfect diagnostic costing 0.5 is now too expensive on expected-loss grounds.

Follow-up: derive the decision boundary

Let p be the posterior probability of degradation after all available information. Pause when 2(1 - p) < 12p, or p > 1/7. At equality you are indifferent on expected loss. This boundary is about costs and the posterior, not the raw alert rate. Different downtime penalties, diagnostic delays or correlated failures can change the decision.

What a strong spoken answer sounds like

> “I need the probability of degradation given an alert, not the alert's sensitivity. I will compare degraded-alert days with all alert days. That gives 4/13. I then compare the expected losses of the available actions. Pausing costs about 1.38 in expectation, so a perfect diagnostic at 0.5 is worthwhile if there is no delay. If the underlying failure rate changes, I must update the posterior before reusing that decision.”

A trading interviewer can extend this into the value of information before quoting. A research interviewer can extend it into calibration, base-rate drift or the cost of classification errors. Neither extension requires memorizing the original numbers.

Your variation

Keep the 10% base rate but lower the false-alert probability to 5%. Work out the posterior and the best action without a diagnostic. Answer: 0.08 / 0.125 = 0.64. Pausing costs 0.72; remaining active costs 7.68. A perfect diagnostic is worth at most 0.72 before its fee. Explain why fewer false alerts can make pausing more compelling while reducing the value of a perfect diagnostic relative to the original alert case.

2. Choose the work before the job title

“Quant” can mean designing predictive models, making prices, measuring risk, implementing execution systems or valuing derivatives. Do not send the same application to all five and hope that a quantitative degree explains the fit. Read the responsibilities as a list of deliverables.

ROLE EMPHASIS	WORK TO UNDERSTAND	USEFUL EVIDENCE
Quantitative trading	Decisions, prices, inventory, execution and risk under uncertainty	A simulation with conditional values, position limits and an honest decision review
Quantitative research	Hypotheses, data, statistical models and evaluation	A reproducible experiment with a baseline, untouched evaluation data and failure analysis
Research engineering or quant development	Reliable research systems, implementation and performance	Tested software, explicit complexity, profiling and maintainable interfaces
Pricing or model validation	Mathematical models, implementation, calibration and model risk	Relevant probability, numerical methods and a model-validation exercise

The last two rows help you recognize a scope mismatch. This book gives useful foundations but does not substitute for systems interview preparation or an advanced pricing curriculum. A role's degree and experience requirements matter more than the appeal of its title.

What current employer guidance actually supports

Jane Street's trading overview describes probability, statistics, problem solving, coding and data analysis, and says its trading interviews do not require prior finance knowledge. Citadel's campus research overview identifies programming, research, algorithms and discussion of past work. G-Research describes mathematical and statistical assessment alongside programming or machine learning. Two Sigma's public technical demonstrations emphasize explaining how a solution is reached. These are examples from those employers, not a claim about every firm. Sources and dates appear at the end.

Build a role evidence matrix

For each real posting, record the exact URL, date checked, location, role level, eligibility, requested skills and application deadline if published. Mark missing information “unconfirmed.” Save your own short interpretation separately from the employer's wording. Do not turn a third-party calendar into a verified deadline.

REQUIREMENT FROM A FICTIONAL POSTING	EVIDENCE AVAILABLE	GAP AND NEXT ACTION
Python data analysis	Course project with clean functions and tests	Add instructions and reproduce the results from an empty environment
Statistical reasoning	Explained a regression in a seminar	Prepare residual analysis and an alternative explanation
Research ownership	Contributed preprocessing to a group project	Separate personal contribution from the team's model and conclusions
Specific graduation window	Degree completion date	Check the actual window; do not assume an internship is open to all students

Use three decisions: apply now, clarify first or develop the missing prerequisite. A hard eligibility mismatch is different from a skill you can strengthen. Do not self-reject because you lack an invented universal credential, but do not describe a mandatory qualification as optional.

Two candidate routes

Fictional mathematics undergraduate: strong probability, little software experience. A small, tested simulation with a clear assumptions log is more useful than a sprawling trading bot copied from a tutorial. Prepare the numerical and probability track while learning to write and test simple functions. Select actual student roles whose requirements match the degree stage.

Fictional experimental scientist: strong inference and reproducible analysis, unfamiliar with market data. Lead with hypothesis design, noisy measurements and rigorous evaluation. Build one exercise that addresses timestamps, delayed availability and non-stationarity. Do not rename laboratory output “alpha” or imply that scientific research automatically transfers to a profitable strategy.

A decision to make today

Choose one primary track and one adjacent track only when there is genuine overlap in your evidence. Write a three-sentence explanation: what work attracts you, what you have already done that resembles it, and what you are learning next. “I like finance and mathematics” does not identify the work.

3. Build evidence that earns a closer look

An application cannot prove every skill, but it can make the next evaluation easier. The reviewer should be able to find your relevant education, contribution, technical tools and results without decoding a project title. A link should lead to something that works and that you are allowed to share.

A small project with a complete chain of evidence

Use this sequence: question, data, baseline, method, evaluation, result, limitation and next experiment. A project is unfinished when the result is impressive but its inputs or evaluation are unclear. A negative result can be useful evidence if the experiment was well designed and you understand why it failed.

PROJECT	MINIMUM CREDIBLE VERSION	WHAT IT DOES NOT ESTABLISH
Trading simulation	Specify a hidden value process, order behavior, quoting rule, inventory limit and transaction cost. Compare with a fixed-quote baseline across several seeds.	That the simulation describes a real venue or that its profits can be traded
Forecasting experiment	Use a permitted dataset, a time-aware split, a simple baseline, a prespecified metric and documented feature availability.	That a small improvement survives execution costs, regime change or repeated selection
Streaming analysis tool	Implement a rolling statistic, test boundary cases and compare complexity with a simple reference implementation.	That research Python is production-ready low-latency infrastructure

Project brief: a forecast that can fail honestly

Question: can information available at the end of period t improve a forecast of a quantity observed in period t + 1, relative to the most recent value? Begin with synthetic data so you can verify the mechanism and timing. If you later use public data, record its license, release timestamp, revisions and collection method. A current historical download may contain information that was unavailable at the time.

Write the baseline and evaluation rule before comparing alternatives. Keep a chronological development period and a final period that is not used to choose features or settings. Report the number of alternatives tried. If a candidate model loses to the baseline, keep the result and explain the next test. Do not remove weak periods merely because they weaken the story.

Deliver four files or sections: a short README with exact run instructions; code that recreates the figures; a data dictionary with timing and units; and a one-page report showing the baseline, candidate result and limitations. Where data cannot be redistributed, provide a lawful acquisition instruction and a small synthetic fixture for tests.

Project brief: trading decisions under selective fills

Use a terminal value with two states. Let some counterparties observe the state and others trade randomly. Compare a quote based on the unconditional value with one based on the value conditional on a fill. Add an inventory cap and log rejected orders. The trading case later in this book supplies a worked starting model. Your contribution is an extension you can explain and test, not copying the case into a repository.

Useful extensions include unequal buy and sell arrival rates, imperfect private information, a finite horizon and a cost per inventory unit. Change one mechanism at a time. State which conclusions come from assumptions you built into the simulator.

Make the repository inspectable

A reviewer should find the question and main result on the first screen. Include your language version, dependencies, seed policy, command to run tests and expected output. Remove credentials, personal data, employer code and files you cannot distribute. Test the instructions from a new directory. A notebook that only runs because of hidden cell state is not reproducible.

If you used AI assistance, comply with the employer's assessment rules and your institution's policies. For a personal project, distinguish assistance from your own decisions and be able to explain every line you submit. Do not use external assistance during an assessment that prohibits it.

The five-minute project defense

01 State the question and why the baseline is a reasonable starting point.

02 Explain one difficult decision you personally made.

03 Show the evaluation design before the result.

04 Identify the strongest reason the result might not generalize.

05 Propose the next experiment and say what finding would make you abandon the idea.

Practice a two-minute version and a deeper version. A technically sophisticated listener may interrupt at the second sentence. Your structure should survive that interruption.