**Numerical Libraries & Data Structures**

* [numpy](https://www.numpy.org/) - NumPy is the fundamental package for scientific computing with Python.
* [scipy](https://www.scipy.org/) - SciPy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering.
* [pandas](https://pandas.pydata.org/) - pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
* [quantdsl](https://github.com/johnbywater/quantdsl) - Domain specific language for quantitative analytics in finance and trading.
* [statistics](https://docs.python.org/3/library/statistics.html) - Builtin Python library for all basic statistical calculations.
* [sympy](https://www.sympy.org/) - SymPy is a Python library for symbolic mathematics.
* [pymc3](https://docs.pymc.io/) - Probabilistic Programming in Python: Bayesian Modeling and Probabilistic Machine Learning with Theano.

**Financial Instruments and Pricing**

* [PyQL](https://github.com/enthought/pyql) - QuantLib's Python port.
* [pyfin](https://github.com/opendoor-labs/pyfin) - Basic options pricing in Python. [ARCHIVED]
* [vollib](https://github.com/vollib/vollib) - vollib is a python library for calculating option prices, implied volatility and greeks.
* [QuantPy](https://github.com/jsmidt/QuantPy) - A framework for quantitative finance In python.
* [Finance-Python](https://github.com/alpha-miner/Finance-Python) - Python tools for Finance.
* [ffn](https://github.com/pmorissette/ffn) - A financial function library for Python.
* [pynance](https://pynance.net/) - PyNance is open-source software for retrieving, analysing and visualizing data from stock and derivatives markets.
* [tia](https://github.com/bpsmith/tia) - Toolkit for integration and analysis.
* [hasura/base-python-dash](https://platform.hasura.io/hub/projects/hasura/base-python-dash) - Hasura quickstart to deploy Dash framework. Written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python.
* [hasura/base-python-bokeh](https://platform.hasura.io/hub/projects/hasura/base-python-bokeh) - Hasura quickstart to visualize data with bokeh library.
* [pysabr](https://github.com/ynouri/pysabr) - SABR model Python implementation.

**Indicators**

* [pandas\_talib](https://github.com/femtotrader/pandas_talib) - A Python Pandas implementation of technical analysis indicators.
* [Tulipy](https://github.com/cirla/tulipy) - Financial Technical Analysis Indicator Library (Python bindings for [tulipindicators](https://github.com/TulipCharts/tulipindicators))

**Trading & Backtesting**

* [TA-Lib](https://ta-lib.org/) - perform technical analysis of financial market data.
* [trade](https://github.com/rochars/trade) - trade is a Python framework for the development of financial applications.
* [zipline](https://www.zipline.io/) - Pythonic algorithmic trading library.
* [QuantSoftware Toolkit](https://github.com/QuantSoftware/QuantSoftwareToolkit) - Python-based open source software framework designed to support portfolio construction and management.
* [quantitative](https://github.com/jeffrey-liang/quantitative) - Quantitative finance, and backtesting library.
* [analyzer](https://github.com/llazzaro/analyzer) - Python framework for real-time financial and backtesting trading strategies.
* [bt](https://github.com/pmorissette/bt) - Flexible Backtesting for Python.
* [backtrader](https://github.com/backtrader/backtrader) - Python Backtesting library for trading strategies.
* [pythalesians](https://github.com/thalesians/pythalesians) - Python library to backtest trading strategies, plot charts, seamlessly download market data, analyse market patterns etc.
* [pybacktest](https://github.com/ematvey/pybacktest) - Vectorized backtesting framework in Python / pandas, designed to make your backtesting easier.
* [pyalgotrade](https://github.com/gbeced/pyalgotrade) - Python Algorithmic Trading Library.
* [tradingWithPython](https://pypi.org/project/tradingWithPython/) - A collection of functions and classes for Quantitative trading.
* [pandas-ta](https://github.com/twopirllc/pandas-ta) - An easy to use Python 3 Pandas Extension with 80+Technical Analysis Indicators
* [ta](https://github.com/bukosabino/ta) - Technical Analysis Library using Pandas (Python)
* [algobroker](https://github.com/joequant/algobroker) - This is an execution engine for algo trading.
* [pysentosa](https://pypi.org/project/pysentosa/) - Python API for sentosa trading system.
* [finmarketpy](https://github.com/cuemacro/finmarketpy) - Python library for backtesting trading strategies and analyzing financial markets.
* [binary-martingale](https://github.com/metaperl/binary-martingale) - Computer program to automatically trade binary options martingale style.
* [fooltrader](https://github.com/foolcage/fooltrader) - the project using big-data technology to provide an uniform way to analyze the whole market.
* [zvt](https://github.com/zvtvz/zvt) - the project using sql,pandas to provide an uniform and extendable way to record data,computing factors,select securites, backtesting,realtime trading and it could show all of them in clearly charts in realtime.
* [pylivetrader](https://github.com/alpacahq/pylivetrader) - zipline-compatible live trading library.
* [pipeline-live](https://github.com/alpacahq/pipeline-live) - zipline's pipeline capability with IEX for live trading.
* [zipline-extensions](https://github.com/quantrocket-llc/zipline-extensions) - Zipline extensions and adapters for QuantRocket.
* [moonshot](https://github.com/quantrocket-llc/moonshot) - Vectorized backtester and trading engine for QuantRocket based on Pandas.
* [PyPortfolioOpt](https://github.com/robertmartin8/PyPortfolioOpt) - Financial portfolio optimisation in python, including classical efficient frontier and advanced methods.
* [riskparity.py](https://github.com/dppalomar/riskparity.py) - fast and scalable design of risk parity portfolios with TensorFlow 2.0
* [mlfinlab](https://github.com/hudson-and-thames/mlfinlab) - Implementations regarding "Advances in Financial Machine Learning" by Marcos Lopez de Prado. (Feature Engineering, Financial Data Structures, Meta-Labeling)
* [pyqstrat](https://github.com/abbass2/pyqstrat) - A fast, extensible, transparent python library for backtesting quantitative strategies.
* [NowTrade](https://github.com/edouardpoitras/NowTrade) - Python library for backtesting technical/mechanical strategies in the stock and currency markets.
* [pinkfish](https://github.com/fja05680/pinkfish) - A backtester and spreadsheet library for security analysis.
* [aat](https://github.com/timkpaine/aat) - Async Algorithmic Trading Engine
* [Backtesting.py](https://kernc.github.io/backtesting.py/) - Backtest trading strategies in Python
* [catalyst](https://github.com/enigmampc/catalyst) - An Algorithmic Trading Library for Crypto-Assets in Python
* [quantstats](https://github.com/ranaroussi/quantstats) - Portfolio analytics for quants, written in Python
* [qtpylib](https://github.com/ranaroussi/qtpylib) - QTPyLib, Pythonic Algorithmic Trading [http://qtpylib.io](http://qtpylib.io/)
* [Quantdom](https://github.com/constverum/Quantdom) - Python-based framework for backtesting trading strategies & analyzing financial markets [GUI ]
* [freqtrade](https://github.com/freqtrade/freqtrade) - Free, open source crypto trading bot

**Risk Analysis**

* [pyfolio](https://github.com/quantopian/pyfolio) - Portfolio and risk analytics in Python.
* [empyrical](https://github.com/quantopian/empyrical) - Common financial risk and performance metrics.
* [fecon235](https://github.com/rsvp/fecon235) - Computational tools for financial economics include: Gaussian Mixture model of leptokurtotic risk, adaptive Boltzmann portfolios.
* [finance](https://pypi.org/project/finance/) - Financial Risk Calculations. Optimized for ease of use through class construction and operator overload.
* [qfrm](https://pypi.org/project/qfrm/) - Quantitative Financial Risk Management: awesome OOP tools for measuring, managing and visualizing risk of financial instruments and portfolios.
* [visualize-wealth](https://github.com/benjaminmgross/visualize-wealth) - Portfolio construction and quantitative analysis.
* [VisualPortfolio](https://github.com/wegamekinglc/VisualPortfolio) - This tool is used to visualize the perfomance of a portfolio.

**Factor Analysis**

* [alphalens](https://github.com/quantopian/alphalens) - Performance analysis of predictive alpha factors.

**Time Series**

* [ARCH](https://github.com/bashtage/arch) - ARCH models in Python.
* [statsmodels](http://statsmodels.sourceforge.net/) - Python module that allows users to explore data, estimate statistical models, and perform statistical tests.
* [dynts](https://github.com/quantmind/dynts) - Python package for timeseries analysis and manipulation.
* [PyFlux](https://github.com/RJT1990/pyflux) - Python library for timeseries modelling and inference (frequentist and Bayesian) on models.
* [tsfresh](https://github.com/blue-yonder/tsfresh) - Automatic extraction of relevant features from time series.
* [hasura/quandl-metabase](https://platform.hasura.io/hub/projects/anirudhm/quandl-metabase-time-series) - Hasura quickstart to visualize Quandl's timeseries datasets with Metabase.

**Calendars**

* [trading\_calendars](https://github.com/quantopian/trading_calendars) - Stock Exchange Trading Calendars.
* [bizdays](https://github.com/wilsonfreitas/python-bizdays) - Business days calculations and utilities.
* [pandas\_market\_calendars](https://github.com/rsheftel/pandas_market_calendars) - Exchange calendars to use with pandas for trading applications.

**Data Sources**

* [nsepy](https://github.com/swapniljariwala/nsepy) - Python Library to get publicly available data on NSE website ie. stock quotes, historical data, live indices.
* [nsetools](https://github.com/vsjha18/nsetools) - Python library for extracting realtime data from National Stock Exchange (India)
* [findatapy](https://github.com/cuemacro/findatapy) - Python library to download market data via Bloomberg, Quandl, Yahoo etc.
* [googlefinance](https://github.com/hongtaocai/googlefinance) - Python module to get real-time stock data from Google Finance API.
* [yahoo-finance](https://github.com/lukaszbanasiak/yahoo-finance) - Python module to get stock data from Yahoo! Finance.
* [pandas-datareader](https://github.com/pydata/pandas-datareader) - Python module to get data from various sources (Google Finance, Yahoo Finance, FRED, OECD, Fama/French, World Bank, Eurostat...) into Pandas datastructures such as DataFrame, Panel with a caching mechanism.
* [pandas-finance](https://github.com/davidastephens/pandas-finance) - High level API for access to and analysis of financial data.
* [pyhoofinance](https://github.com/innes213/pyhoofinance) - Rapidly queries Yahoo Finance for multiple tickers and returns typed data for analysis.
* [yfinanceapi](https://github.com/Karthik005/yfinanceapi) - Finance API for Python.
* [yql-finance](https://github.com/slawek87/yql-finance) - yql-finance is simple and fast. API returns stock closing prices for current period of time and current stock ticker (i.e. APPL, GOOGL).
* [ystockquote](https://github.com/cgoldberg/ystockquote) - Retrieve stock quote data from Yahoo Finance.
* [wallstreet](https://github.com/mcdallas/wallstreet) - Real time stock and option data.
* [stock\_extractor](https://github.com/ZachLiuGIS/stock_extractor) - General Purpose Stock Extractors from Online Resources.
* [Stockex](https://github.com/cttn/Stockex) - Python wrapper for Yahoo! Finance API.
* [finsymbols](https://github.com/skillachie/finsymbols) - Obtains stock symbols and relating information for SP500, AMEX, NYSE, and NASDAQ.
* [FRB](https://github.com/avelkoski/FRB) - Python Client for FRED® API.
* [inquisitor](https://github.com/econdb/inquisitor) - Python Interface to Econdb.com API.
* [yfi](https://github.com/nickelkr/yfi) - Yahoo! YQL library.
* [chinesestockapi](https://pypi.org/project/chinesestockapi/) - Python API to get Chinese stock price.
* [exchange](https://github.com/akarat/exchange) - Get current exchange rate.
* [ticks](https://github.com/jamescnowell/ticks) - Simple command line tool to get stock ticker data.
* [pybbg](https://github.com/bpsmith/pybbg) - Python interface to Bloomberg COM APIs.
* [ccy](https://github.com/lsbardel/ccy) - Python module for currencies.
* [tushare](https://pypi.org/project/tushare/) - A utility for crawling historical and Real-time Quotes data of China stocks.
* [jsm](https://pypi.org/project/jsm/) - Get the japanese stock market data.
* [cn\_stock\_src](https://github.com/jealous/cn_stock_src) - Utility for retrieving basic China stock data from different sources.
* [coinmarketcap](https://github.com/barnumbirr/coinmarketcap) - Python API for coinmarketcap.
* [after-hours](https://github.com/datawrestler/after-hours) - Obtain pre market and after hours stock prices for a given symbol.
* [bronto-python](https://pypi.org/project/bronto-python/) - Bronto API Integration for Python.
* [pytdx](https://github.com/rainx/pytdx) - Python Interface for retrieving chinese stock realtime quote data from TongDaXin Nodes.
* [pdblp](https://github.com/matthewgilbert/pdblp) - A simple interface to integrate pandas and the Bloomberg Open API.
* [tiingo](https://github.com/hydrosquall/tiingo-python) - Python interface for daily composite prices/OHLC/Volume + Real-time News Feeds, powered by the Tiingo Data Platform.
* [IEX](https://github.com/addisonlynch/iexfinance) - Python Interface for retrieving real-time and historical prices and equities data from The Investor's Exchange.
* [alpaca-trade-api](https://github.com/alpacahq/alpaca-trade-api-python) - Python interface for retrieving real-time and historical prices from Alpaca API as well as trade execution.
* [metatrader5](https://pypi.org/project/MetaTrader5/) - API Connector to MetaTrader 5 Terminal
* [akshare](https://github.com/jindaxiang/akshare) - AkShare is an elegant and simple financial data interface library for Python, built for human beings! [https://akshare.readthedocs.io](https://akshare.readthedocs.io/)
* [yahooquery](https://github.com/dpguthrie/yahooquery) - Python interface for retrieving data through unofficial Yahoo Finance API.

**Data Visualization**

* [matplotlib](https://matplotlib.org/) - Matplotlib is the first Python data visualization library, many other libraries are built on top of it or designed to work in tandem with it during analysis.
* [D-tale](https://github.com/man-group/dtale) - D-Tale is the combination of a Flask back-end and a React front-end to bring you an easy way to view & analyze Pandas data structures.
* [Seaborn](https://github.com/mwaskom/seaborn) - Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.
* [plotly](https://github.com/plotly/plotly.py) - interactive, open-source, and browser-based graphing library for Python -[bokeh](https://github.com/bokeh/bokeh) - interactive visualization library for modern web browsers

**Excel Integration**

* [xlwings](https://www.xlwings.org/) - Make Excel fly with Python.
* [openpyxl](https://openpyxl.readthedocs.io/en/latest/) - Read/Write Excel 2007 xlsx/xlsm files.
* [xlrd](https://github.com/python-excel/xlrd) - Library for developers to extract data from Microsoft Excel spreadsheet files.
* [xlsxwriter](https://xlsxwriter.readthedocs.io/) - Write files in the Excel 2007+ XLSX file format.
* [xlwt](https://github.com/python-excel/xlwt) - Library to create spreadsheet files compatible with MS Excel 97/2000/XP/2003 XLS files, on any platform.
* [DataNitro](https://datanitro.com/) - DataNitro also offers full-featured Python-Excel integration, including UDFs. Trial downloads are available, but users must purchase a license.
* [xlloop](http://xlloop.sourceforge.net/) - XLLoop is an open source framework for implementing Excel user-defined functions (UDFs) on a centralised server (a function server).
* [expy](http://www.bnikolic.co.uk/expy/expy.html) - The ExPy add-in allows easy use of Python directly from within an Microsoft Excel spreadsheet, both to execute arbitrary code and to define new Excel functions.
* [pyxll](https://www.pyxll.com/) - PyXLL is an Excel add-in that enables you to extend Excel using nothing but Python code.

**R**

**Numerical Libraries & Data Structures**

* [xts](https://cran.r-project.org/web/packages/xts/index.html) - eXtensible Time Series: Provide for uniform handling of R's different time-based data classes by extending zoo, maximizing native format information preservation and allowing for user level customization and extension, while simplifying cross-class interoperability.
* [data.table](https://cran.r-project.org/web/packages/data.table/index.html) - Extension of data.frame: Fast aggregation of large data (e.g. 100GB in RAM), fast ordered joins, fast add/modify/delete of columns by group using no copies at all, list columns and a fast file reader (fread). Offers a natural and flexible syntax, for faster development.
* [sparseEigen](https://github.com/dppalomar/sparseEigen) - Sparse pricipal component analysis.
* [TSdbi](http://tsdbi.r-forge.r-project.org/) - Provides a common interface to time series databases.
* [tseries](https://cran.r-project.org/web/packages/tseries/index.html) - Time Series Analysis and Computational Finance.
* [zoo](https://cran.r-project.org/web/packages/zoo/index.html) - S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations).
* [tis](https://cran.r-project.org/web/packages/tis/index.html) - Functions and S3 classes for time indexes and time indexed series, which are compatible with FAME frequencies.
* [tfplot](https://cran.r-project.org/web/packages/tfplot/index.html) - Utilities for simple manipulation and quick plotting of time series data.
* [tframe](https://cran.r-project.org/web/packages/tframe/index.html) - A kernel of functions for programming time series methods in a way that is relatively independently of the representation of time.

**Data Sources**

* [IBrokers](https://cran.r-project.org/web/packages/IBrokers/index.html) - Provides native R access to Interactive Brokers Trader Workstation API.
* [Rblpapi](https://cran.r-project.org/web/packages/Rblpapi/index.html) - An R Interface to 'Bloomberg' is provided via the 'Blp API'.
* [Quandl](https://www.quandl.com/tools/r) - Get Financial Data Directly Into R.
* [Rbitcoin](https://cran.r-project.org/web/packages/Rbitcoin/index.html) - Unified markets API interface (bitstamp, kraken, btce, bitmarket).
* [GetTDData](https://cran.r-project.org/web/packages/GetTDData/index.html) - Downloads and aggregates data for Brazilian government issued bonds directly from the website of Tesouro Direto.
* [GetHFData](https://cran.r-project.org/web/packages/GetHFData/index.html) - Downloads and aggregates high frequency trading data for Brazilian instruments directly from Bovespa ftp site.

**Financial Instruments and Pricing**

* [RQuantLib](http://dirk.eddelbuettel.com/code/rquantlib.html) - RQuantLib connects GNU R with QuantLib.
* [quantmod](https://cran.r-project.org/web/packages/quantmod/index.html) - Quantitative Financial Modelling Framework.
* [Rmetrics](https://www.rmetrics.org/) - The premier open source software solution for teaching and training quantitative finance.
  + [fAsianOptions](https://cran.r-project.org/web/packages/fAsianOptions/index.html) - EBM and Asian Option Valuation.
  + [fAssets](https://cran.r-project.org/web/packages/fAssets/index.html) - Analysing and Modelling Financial Assets.
  + [fBasics](https://cran.r-project.org/web/packages/fBasics/index.html) - Markets and Basic Statistics.
  + [fBonds](https://cran.r-project.org/web/packages/fBonds/index.html) - Bonds and Interest Rate Models.
  + [fExoticOptions](https://cran.r-project.org/web/packages/fExoticOptions/index.html) - Exotic Option Valuation.
  + [fOptions](https://cran.r-project.org/web/packages/fOptions/index.html) - Pricing and Evaluating Basic Options.
  + [fPortfolio](https://cran.r-project.org/web/packages/fPortfolio/index.html) - Portfolio Selection and Optimization.
* [portfolio](https://cran.r-project.org/web/packages/portfolio/index.html) - Analysing equity portfolios.
* [portfolioSim](https://cran.r-project.org/web/packages/portfolioSim/index.html) - Framework for simulating equity portfolio strategies.
* [sparseIndexTracking](https://github.com/dppalomar/sparseIndexTracking) - Portfolio design to track an index.
* [covFactorModel](https://github.com/dppalomar/covFactorModel) - Covariance matrix estimation via factor models.
* [riskParityPortfolio](https://github.com/dppalomar/riskParityPortfolio) - Blazingly fast design of risk parity portfolios.
* [sde](https://cran.r-project.org/web/packages/sde/index.html) - Simulation and Inference for Stochastic Differential Equations.
* [YieldCurve](https://cran.r-project.org/web/packages/YieldCurve/index.html) - Modelling and estimation of the yield curve.
* [SmithWilsonYieldCurve](https://cran.r-project.org/web/packages/SmithWilsonYieldCurve/index.html) - Constructs a yield curve by the Smith-Wilson method from a table of LIBOR and SWAP rates.
* [ycinterextra](https://cran.r-project.org/web/packages/ycinterextra/index.html) - Yield curve or zero-coupon prices interpolation and extrapolation.
* [AmericanCallOpt](https://cran.r-project.org/web/packages/AmericanCallOpt/index.html) - This package includes pricing function for selected American call options with underlying assets that generate payouts.
* [VarSwapPrice](https://cran.r-project.org/web/packages/VarSwapPrice/index.html) - Pricing a variance swap on an equity index.
* [RND](https://cran.r-project.org/web/packages/RND/index.html) - Risk Neutral Density Extraction Package.
* [LSMonteCarlo](https://cran.r-project.org/web/packages/LSMonteCarlo/index.html) - American options pricing with Least Squares Monte Carlo method.
* [OptHedging](https://cran.r-project.org/web/packages/OptHedging/index.html) - Estimation of value and hedging strategy of call and put options.
* [tvm](https://cran.r-project.org/web/packages/tvm/index.html) - Time Value of Money Functions.
* [OptionPricing](https://cran.r-project.org/web/packages/OptionPricing/index.html) - Option Pricing with Efficient Simulation Algorithms.
* [credule](https://cran.r-project.org/web/packages/credule/index.html) - Credit Default Swap Functions.
* [derivmkts](https://cran.r-project.org/web/packages/derivmkts/index.html) - Functions and R Code to Accompany Derivatives Markets.
* [FinCal](https://github.com/felixfan/FinCal) - Package for time value of money calculation, time series analysis and computational finance.
* [r-quant](https://github.com/artyyouth/r-quant) - R code for quantitative analysis in finance.
* [options.studies](https://github.com/taylorizing/options.studies) - options trading studies functions for use with options.data package and shiny.

**Trading**

* [TA-Lib](https://ta-lib.org/) - perform technical analysis of financial market data.
* [backtest](https://cran.r-project.org/web/packages/backtest/index.html) - Exploring Portfolio-Based Conjectures About Financial Instruments.
* [pa](https://cran.r-project.org/web/packages/pa/index.html) - Performance Attribution for Equity Portfolios.
* [TTR](https://cran.r-project.org/web/packages/TTR/index.html) - Technical Trading Rules.
* [QuantTools](https://quanttools.bitbucket.io/_site/index.html) - Enhanced Quantitative Trading Modelling.

**Risk Analysis**

* [PerformanceAnalytics](https://cran.r-project.org/web/packages/PerformanceAnalytics/index.html) - Econometric tools for performance and risk analysis.

**Time Series**

* [tseries](https://cran.r-project.org/web/packages/tseries/index.html) - Time Series Analysis and Computational Finance.
* [zoo](https://cran.r-project.org/web/packages/zoo/index.html) - S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations).
* [xts](https://cran.r-project.org/web/packages/xts/index.html) - eXtensible Time Series.
* [fGarch](https://cran.r-project.org/web/packages/fGarch/index.html) - Rmetrics - Autoregressive Conditional Heteroskedastic Modelling.
* [timeSeries](https://cran.r-project.org/web/packages/timeSeries/index.html) - Rmetrics - Financial Time Series Objects.
* [rugarch](https://cran.r-project.org/web/packages/rugarch/index.html) - Univariate GARCH Models.
* [rmgarch](https://cran.r-project.org/web/packages/rmgarch/index.html) - Multivariate GARCH Models.
* [tidypredict](https://github.com/edgararuiz/tidypredict) - Run predictions inside the database <https://tidypredict.netlify.com/>.
* [tidyquant](https://github.com/business-science/tidyquant) - Bringing financial analysis to the tidyverse.
* [timetk](https://github.com/business-science/timetk) - A toolkit for working with time series in R.
* [tibbletime](https://github.com/business-science/tibbletime) - Built on top of the tidyverse, tibbletime is an extension that allows for the creation of time aware tibbles through the setting of a time index.

**Calendars**

* [timeDate](https://cran.r-project.org/web/packages/timeDate/index.html) - Chronological and Calendar Objects
* [bizdays](https://cran.r-project.org/web/packages/bizdays/index.html) - Business days calculations and utilities

**Matlab**

**FrameWorks**

* [QUANTAXIS](https://github.com/yutiansut/quantaxis) - Integrated Quantitative Toolbox with Matlab.

**Julia**

* [QuantLib.jl](https://github.com/pazzo83/QuantLib.jl) - Quantlib implementation in pure Julia.
* [FinancialMarkets.jl](https://github.com/imanuelcostigan/FinancialMarkets.jl) - Describe and model financial markets objects using Julia.
* [Ito.jl](https://github.com/aviks/Ito.jl) - A Julia package for quantitative finance.
* [TALib.jl](https://github.com/femtotrader/TALib.jl) - A Julia wrapper for TA-Lib.
* [Miletus.jl](https://juliacomputing.com/docs/miletus/index.html) - A financial contract definition, modeling language, and valuation framework.
* [Temporal.jl](https://github.com/dysonance/Temporal.jl) - Flexible and efficient time series class & methods.
* [Indicators.jl](https://github.com/dysonance/Indicators.jl) - Financial market technical analysis & indicators on top of Temporal.
* [Strategems.jl](https://github.com/dysonance/Strategems.jl) - Quantitative systematic trading strategy development and backtesting.
* [TimeSeries.jl](https://github.com/JuliaStats/TimeSeries.jl) - Time series toolkit for Julia.
* [MarketTechnicals.jl](https://github.com/JuliaQuant/MarketTechnicals.jl) - Technical analysis of financial time series on top of TimeSeries.
* [MarketData.jl](https://github.com/JuliaQuant/MarketData.jl) - Time series market data.
* [TimeFrames.jl](https://github.com/femtotrader/TimeFrames.jl) - A Julia library that defines TimeFrame (essentially for resampling TimeSeries).

**Java**

* [Strata](http://strata.opengamma.io/) - Modern open-source analytics and market risk library designed and written in Java.
* [JQuantLib](http://www.jquantlib.org/) - JQuantLib is a free, open-source, comprehensive framework for quantitative finance, written in 100% Java.
* [finmath.net](http://finmath.net/) - Java library with algorithms and methodologies related to mathematical finance.
* [quantcomponents](https://github.com/lsgro/quantcomponents) - Free Java components for Quantitative Finance and Algorithmic Trading.
* [DRIP](https://lakshmidrip.github.io/DRIP) - Fixed Income, Asset Allocation, Transaction Cost Analysis, XVA Metrics Libraries.

**JavaScript**

**Data Visualization**

* [QUANTAXIS\_Webkit](https://github.com/yutiansut/QUANTAXIS_Webkit) an awesome visualization center based on quantaxis.

**Haskell**

* [quantfin](https://github.com/boundedvariation/quantfin) - quant finance in pure haskell.
* [hqfl](https://github.com/co-category/hqfl) - Haskell Quantitative Finance Library.

**Scala**

* [QuantScale](https://github.com/choucrifahed/quantscale) - Scala Quantitative Finance Library.
* [Scala Quant](https://github.com/frankcash/Scala-Quant) Scala library for working with stock data from IFTTT recipes or Google Finance.

**Ruby**

* [Jiji](https://github.com/unageanu/jiji2) - Open Source Forex algorithmic trading framework using OANDA REST API.

**Elixir/Erlang**

* [Tai](https://github.com/fremantle-capital/tai) - Open Source composable, real time, market data and trade execution toolkit.
* [Workbench](https://github.com/fremantle-industries/workbench) - From Idea to Execution - Manage your trading operation across a globally distributed cluster

**Frameworks**

* [QuantLib](https://www.quantlib.org/) - The QuantLib project is aimed at providing a comprehensive software framework for quantitative finance.
  + [JQuantLib](http://www.jquantlib.org/) - Java port.
  + [RQuantLib](http://dirk.eddelbuettel.com/code/rquantlib.html) - R port.
  + [QuantLibAddin](https://www.quantlib.org/quantlibaddin/) - Excel support.
  + [QuantLibXL](https://www.quantlib.org/quantlibxl/) - Excel support.
  + [QLNet](https://github.com/amaggiulli/qlnet) - .Net port.
  + [PyQL](https://github.com/enthought/pyql) - Python port.
  + [QuantLib.jl](https://github.com/pazzo83/QuantLib.jl) - Julia port.
* [TA-Lib](https://ta-lib.org/) - perform technical analysis of financial market data.

**CSharp**

* [QuantConnect](https://github.com/QuantConnect/Lean) - Lean Engine is an open-source fully managed C# algorithmic trading engine built for desktop and cloud usage.