Network analysis and clustering in quantitative wealth and investment management QWIM

Cristian Homescu

December 2022

Abstract

This document provides details for this QWIM project, and it incorporates the following sections

- Motivation
- Relevant references
- Suggested project tasks and timelines
- Practical info
  - Recommended software tools
  - Recommended datasets
- Design and implementation for the project codes
- Potentially useful Python and R packages, codes and frameworks
- Appendices

Appendices include

- Overviews of investment processes and models in QWIM
- Comparison of investment portfolios using portfolios metrics and benchmark portfolios

Contents

1 Motivation for the project

1.1 Beware of correlations which are not consistent with intuition

In many cases, advanced quantitative approaches (including machine learning) may not necessarily deliver better results in $\mathbb{QWIM}$, especially when decision making relies on data-based spurious correlations rather than on real causality.

Vigen (*Spurious Correlations*, 2019): many examples of such data-based spurious correlations

- Divorce rate in Maine has 99% correlation with per capita consumption of margarin
- Per capita consumption of mozzarella cheese has 96% correlation with civil engineering doctorates awarded in US
- Per capita consumption of chicken has 90% correlation with total US crude oil imports

Example in $\mathbb{QWIM}$ Laurinaityte et al. ("Elephants and the Cross-Section of Expected Returns," 2019): population growth of captive Asian elephants explains cross-section of expected returns of usually sorted portfolios with $R^2 = 0.91$ and $tStat = 2.93$ for market price of risk.

Question: Does it mean that number of captive elephants is the new outstanding factor in empirical asset pricing?

Answer: Likely it is an artifact due to data mining rather than a proper factor for factor-based investing.

1.2 Portfolio optimization

While there are many challenges in a portfolio optimization process relying on the corelation (or covariance) matrix, some of the most important issues (including potential lack of robustness and diversification) are due to the fact that correlation matrix lacks the notion of hierarchy.

It was shown that many complex systems can be arranged in a natural hierarchy comprising nested substructures, and financial markets are no exception. While a correlation matrix makes no differentiation between assets, some assets seem closer substitutes of one another, while others seem complementary to one another. This can be better handled through network analysis and clustering.

Networks enable practical usage of high / low centrality concepts

- significant interconnectedness risk (tail events propagate more quickly) due to assets with high centrality scores
- "peripheral assets" carry relatively less interconnectedness risk

Network-based and clustering-based portfolio optimization is likely to deliver more robust and diversified portfolios, and achieve better risk-adjusted performances compared to portfolios obtained using commonly used portfolio optimization techniques. Since no single clustering algorithm can be said to perform best on all datasets, different strategies must be tested and compared.

1.3 Portfolio diversification

Diversification is one of the most important concepts in the financial world. It is often said that diversification is the only free lunch in finance. From a qualitative point of view, the concept of diversification is quite clear: a portfolio is well-diversified if shocks in the individual components do not heavily impact on the overall portfolio. Relatively simple to understand then but profoundly difficult to define. Indeed, there is no broadly accepted precise and quantitative definition of diversification.

One of the most vexing problems in investment management is that diversification seems to disappear when investors need it the most. A key challenge in the construction of diversified multi-asset portfolio strategies is that even a seemingly well-balanced allocation to many asset classes can eventually translate into a portfolio with a very concentrated set of underlying risk exposures.

Network analysis applied to structure of investment portfolios is very beneficial to analyze diversification properties. We can also consider a portfolio selection approach that combines diversification and optimization.

2 Relevant references

2.1 Main references

List of references:

Ackerman et al. ("Weighted clustering: Towards solving the user's dilemma," 2021)

Alfarra et al. ("Rethinking Clustering for Robustness," 2021)

Akansu et al. ("Quant investing in cluster portfolios," 2021)

Avellaneda and Serur ("Hierarchical PCA and Modeling Asset Correlations," 2020)

Baitinger ("Forecasting asset returns with network-based metrics: A statistical and economic analysis," 2021)

Baitinger and Flegel ("New Concepts in Financial Forecasting: Network-Based Information, Topological Data Analysis and their Combination," 2021)

Bnouachir and Mkhadri ("Efficient cluster-based portfolio optimization," 2021)

Chua et al. ("The Myth of Diversification," 2009)

Clemente et al. ("Smart network based portfolios," 2019)

Clemente et al. ("Asset allocation: new evidence through network approaches," 2021)

Coraggio and Coretto ("Selecting the number of clusters, clustering models, and algorithms. A unifying approach based on the quadratic discriminant score," 2021)

Dees et al. ("Portfolio Cuts: A Graph-Theoretic Framework to Diversification," 2020)

de Miranda Cardoso et al. ("Algorithms for Learning Graphs in Financial Markets," 2020)

Duarte and De Castro ("A Framework to Perform Asset Allocation Based on Partitional Clustering," 2020)

Dugué et al. ("Evaluating clustering quality using features salience: a promising approach," 2021)

Eidenvall ("Hierarchical Clustering To Improve Portfolio Tail Risk Characteristics," 2021)

Flint et al. ("Defining and measuring portfolio diversification," 2021)

Fu and Perry ("Estimating the Number of Clusters Using Cross-Validation," 2020)

Fusai et al. ("Equally Diversified or Equally Weighted?" 2020)

Giudici et al. ("Network models to improve robot advisory portfolios," 2022)

Guan and Loew ("A Distance-based Separability Measure for Internal Cluster Validation," 2021)

Guo et al. ("A Time-Varying Network for Cryptocurrencies," 2021)

Heckens and Guhr ("New collectivity measures for financial covariances and correlations," 2022)

Herteliu et al. ("Network analysis of pension funds investments," 2021)

Horvath et al. ("Clustering Market Regimes Using the Wasserstein Distance," 2021)

Huang et al. ("Financial risk propagation between Chinese and American stock markets based on multilayer networks," 2022)

Jaeger et al. ("Understanding machine learning for diversified portfolio construction by explainable AI," 2020)

Jaeger et al. ("Interpretable Machine Learning for Diversified Portfolio Construction," 2021)

Jaeger et al. ("Adaptive Seriational Risk Parity and other Extensions for Heuristic Portfolio Construction using Machine Learning and Graph Theory," 2021)

Katsouris ("Optimal Portfolio Choice and Stock Centrality for Tail Risk Events," 2021)

Kawamoto and Kabashima ("Cross-validation estimate of the number of clusters in a network," 2017)

Kaya ("Eccentricity in Asset Management," 2015)

Kaya ("Managing ambiguity in asset allocation," 2017)

Kinlaw et al. ("The Myth of Diversification Reconsidered," 2021)

Konstantinov et al. ("A network and machine learning approach to factor, asset, and blended allocation," 2020)

Koumou ("Diversification and portfolio theory: a review," 2020)

Kurtti ("How many stocks make a diversified portfolio in a continuous-time world?" 2020)

Laur ("Portfolio Optimization - Can Optimizing Portfolio Outperform Naive Diversification?" 2020)

Leon et al. ("Clustering algorithms for Risk-Adjusted Portfolio Construction," 2017)

Lim and Ong ("Portfolio Diversification Using Shape-Based Clustering," 2021)

Lohre et al. ("Hierarchical Risk Parity: Accounting for Tail Dependencies in Multi-asset Multi-factor Allocations," 2020)

Lopez de Prado ("Building Diversified Portfolios that Outperform Out of Sample," 2016)

Lopez de Prado ("Estimation of Theory-Implied Correlation Matrices," 2019)

Lopez de Prado ("Clustering," 2020)

Lopez de Prado (*Machine learning for asset managers*, 2020)

Lu et al. (“A Cluster Representative Selection Method for Stock Portfolio Based on Efficient Frontier,” 2021)

Marti et al. (“A review of two decades of correlations, hierarchies, networks and clustering in financial markets,” 2021)

Massahi et al. (“Development of an efficient cluster-based portfolio optimization model under realistic market conditions,” 2020)

Millington and Niranjan (“Stability and similarity in financial networks – How do they change in times of turbulence?” 2021)

Molyboga (“A Modified Hierarchical Risk Parity Framework for Portfolio Management,” 2020)

Olmo (“Optimal portfolio allocation and asset centrality revisited,” 2021)

Page and Panariello (“When Diversification Fails,” 2018)

Papenbrock et al. (“Matrix Evolutions: Synthetic Correlations and Explainable Machine Learning for Constructing Robust Investment Portfolios,” 2021)

Papenbrock et al. (“Can Adaptive Seriational Risk Parity Tame Crypto Portfolios?” 2021)

Parmentier (“Measures of Portfolio Diversification,” 2018)

Peralta and Zareei (“A network approach to portfolio selection,” 2016)

Raffinot (“Hierarchical Clustering-Based Asset Allocation,” 2017)

Raffinot (“The Hierarchical Equal Risk Contribution Portfolio,” 2018)

Roncalli (“Advanced Course in Asset Management,” 2021)

Sakurai et al. (“Correlation diversified passive portfolio strategy based on permutation of assets,” 2021)

Sass and Thos (“Risk reduction and portfolio optimization using clustering methods,” 2022)

Scherer (“Adding alternative assets: return enhancement, diversification or hedging?” 2021)

Schwendner et al. (“Adaptive Seriational Risk Parity and Other Extensions for Heuristic Portfolio Construction Using Machine Learning and Graph Theory,” 2021)

Serur and Avellaneda (“Hierarchical PCA and Modeling Asset Correlations,” 2021)

Shirota and Murakami (“Long-term Time Series Data Clustering of Stock Prices for Portfolio Selection,” 2021)

Snow (“Machine Learning in Asset Management - Part 2: Portfolio Construction - Weight Optimization,” 2020)

Swedroe (“The Importance of Diversification in Achieving Long-Term Goals,” 2020)

Tang et al. (“Asset Selection via Correlation Blockmodel Clustering,” 2021)

Thiagarajan et al. (“Financial Globalization and Its Implications for Diversification of Portfolio Risk,” 2021)

Thrun and Stier (“Fundamental clustering algorithms suite,” 2021)

Vankwikelberge et al. (“ExClus: Explainable Clustering on Low-dimensional Data Representations,” 2021)

Vyrost et al. (“Network-based asset allocation strategies,” 2019)

Wang and Aste (“Dynamic Portfolio Optimization with Inverse Covariance Clustering,” 2022)

Yang et al. (“Portfolio optimization based on empirical mode decomposition,” 2019)

Zaimovic et al. (“How Many Stocks Are Sufficient for Equity Portfolio Diversification? A Review of the Literature,” 2021)

Zhan et al. (“Graphical Models for Financial Time Series and Portfolio Selection,” 2021)

Zhao et al. (“Stock market as temporal network,” 2018)

Zhao et al. (“Robust portfolio rebalancing with cardinality and diversification constraints,” 2021)

2.2 Comprehensive list of references

2.2.1 Clustering within context of QWIM

List of references:

Akansu et al. (“Quant investing in cluster portfolios,” 2021)

Alokley and Albarrak (“Clustering of Extremes in Financial Returns: A Study of Developed and Emerging Markets,” 2020)

Avellaneda and Serur (“Hierarchical PCA and Modeling Asset Correlations,” 2020)

Begusic and Kostanjcar (“Cluster-Based Shrinkage of Correlation Matrices for Portfolio Optimization,” 2019)

Bennett et al. (“Lead-lag detection and network clustering for multivariate time series with an application to the US equity market,” 2022)

Bnouachir and Mkhadri (“Efficient cluster-based portfolio optimization,” 2021)

Cai et al. (“Clustering Approaches for Financial Data Analysis: a Survey,” 2016)

Cajas (“Robust Portfolio Selection with Near Optimal Centering,” 2019)

Custodio João et al. (“Clustering Dynamics and Persistence for Financial Multivariate Panel Data,” 2021)

Duarte and De Castro (“A Framework to Perform Asset Allocation Based on Partitional Clustering,” 2020)

Dugué et al. (“Evaluating clustering quality using features salience: a promising approach,” 2021)

Eidenvall (“Hierarchical Clustering To Improve Portfolio Tail Risk Characteristics,” 2021)

Emerson (“Machine learning for financial applications: self-organising maps, hierarchical clustering and dynamic time-warping for portfolio constructive,” 2019)

Garvey and Madhavan (“Reconstructing Emerging and Developed Markets Using Hierarchical Clustering,” 2019)

Guan and Loew (“A Distance-based Separability Measure for Internal Cluster Validation,” 2021)

Gubu et al. (“Robust mean-variance portfolio selection with time series clustering,” 2021)

Gupta and Chatterjee (“Financial Time Series Clustering,” 2018)

Han and Ge (“Effect of dimensionality reduction on stock selection with cluster analysis in different market situations,” 2020)

Horvath et al. (“Clustering Market Regimes Using the Wasserstein Distance,” 2021)

Heckens and Guhr (“New collectivity measures for financial covariances and correlations,” 2022)

Jaeger et al. (“Interpretable Machine Learning for Diversified Portfolio Construction,” 2021)

Jaeger et al. (“Adaptive Seriational Risk Parity and other Extensions for Heuristic Portfolio Construction using Machine Learning and Graph Theory,” 2021)

Jain and Jain (“Can Machine Learning-Based Portfolios Outperform Traditional Risk-Based Portfolios? The Need to Account for Covariance Misspecification,” 2019)

Kolrep et al. (“Economic Versus Statistical Clustering in Multi-Asset Multi-Factor Strategies,” 2020)

Lee and Seregina (“Optimal Portfolio Using Factor Graphical Lasso,” 2022)

Leon et al. (“Clustering algorithms for Risk-Adjusted Portfolio Construction,” 2017)

Lim and Ong (“Portfolio Diversification Using Shape-Based Clustering,” 2021)

Lohre et al. (“Hierarchical Risk Parity: Accounting for Tail Dependencies in Multi-asset Multi-factor Allocations,” 2020)

Lopez de Prado (“Clustering,” 2020)

Lopez de Prado (“Building Diversified Portfolios that Outperform Out of Sample,” 2016)

Lu et al. (“A Cluster Representative Selection Method for Stock Portfolio Based on Efficient Frontier,” 2021)

Mattera et al. (“Distribution-Based Entropy Weighting Clustering of Skewed and Heavy Tailed Time Series,” 2021)

Marti et al. (“A review of two decades of correlations, hierarchies, networks and clustering in financial markets,” 2021)

Millington and Niranjan (“Partial correlation financial networks,” 2020)

Molyboga (“A Modified Hierarchical Risk Parity Framework for Portfolio Management,” 2020)

Murialdo et al. (“Inferring multi-period optimal portfolios via detrending moving average cluster entropy,” 2021)

Nanakorn and Palmgren (“Hierarchical Clustering in Risk-Based Portfolio Construction,” 2021)

Naraoka et al. (“Detecting and explaining changes in various assets’ relationships in financial markets,” 2020)

Papenbrock et al. (“Matrix Evolutions: Synthetic Correlations and Explainable Machine Learning for Constructing Robust Investment Portfolios,” 2021)

Papenbrock et al. (“Can Adaptive Seriational Risk Parity Tame Crypto Portfolios?” 2021)

Park (“Clustering Approaches for Global Minimum Variance Portfolio,” 2020)

Pharasi et al. (“Dynamics of the market states in the space of correlation matrices with applications to financial markets,” 2021)

Poletaev and Spiridonova (“Hierarchical Clustering as a Dimension Reduction Technique for Markowitz Portfolio Optimization,” 2020)

Puerto et al. (“Clustering and portfolio selection problems: A unified framework,” 2020)

Putra et al. (“Portfolio Selection of KOMPAS-100 Stocks Index Using B-Spline Based Clustering,” 2021)

Raffinot (“Hierarchical Clustering-Based Asset Allocation,” 2017)

Raffinot (“The Hierarchical Equal Risk Contribution Portfolio,” 2018)

Sass and Thos (“Risk reduction and portfolio optimization using clustering methods,” 2022)

Schwendner et al. (“Adaptive Seriational Risk Parity and Other Extensions for Heuristic Portfolio Construction Using Machine Learning and Graph Theory,” 2021)

Serur and Avellaneda (“Hierarchical PCA and Modeling Asset Correlations,” 2021)

Sjostrand and Behnejad (“Exploration of Hierarchical Clustering in Long-only Risk-based Portfolio Optimization,” 2020)

Snow (“Machine Learning in Asset Management - Part 2: Portfolio Construction - Weight Optimization,” 2020)

Tang et al. (“Asset Selection via Correlation Blockmodel Clustering,” 2021)

Tola et al. (“Cluster analysis for portfolio optimization,” 2008)

Turner (“Graph Auto-Encoders for Financial Clustering,” 2021)

Vojtko and Cisár (“An Analysis of Volatility Clustering of Equity Factor Strategies,” 2021)

2.2.2 Network analysis within context of QWIM

List of references:

Ahelegbey and Giudici (“Market Risk, Connectedness and Turbulence: A Comparison of 21st Century Financial Crises,” 2020)

Baitinger (“Forecasting asset returns with network-based metrics: A statistical and economic analysis,” 2021)

Baitinger and Maier (“The (Mis)Behavior of Hedge Fund Strategies: A Network-Based Analysis,” 2019)

Baitinger and Papenbrock (“Interconnectedness Risk and Active Portfolio Management,” 2017)

Baitinger and Flegel (“New Concepts in Financial Forecasting: Network-Based Information, Topological Data Analysis and their Combination,” 2021)

Bardoscia et al. (“The Physics of Financial Networks,” 2021)

Barunik and Ellington (“Dynamic Networks in Large Financial and Economic Systems,” 2021)

Castilho et al. (“Forecasting Financial Market Structure from Network Features using Machine Learning,” 2021)

Chen et al. (“Constructing a multilayer network for stock market,” 2020)

Clemente et al. (“Smart network based portfolios,” 2019)

Clemente et al. (“Asset allocation: new evidence through network approaches,” 2021)

de Carvalho and Gupta (“A network approach to unravel asset price comovement using minimal dependence structure,” 2018)

de Miranda Cardoso et al. (“Algorithms for Learning Graphs in Financial Markets,” 2020)

Dey et al. (“Community detection in complex networks: From statistical foundations to data science applications,” 2021)

Di Cerbo and Taylor (“Graph theoretical representations of equity indices and their centrality measures,” 2021)

Duan (“Predicting with Structured Data: Graphs, Ranks, and Time Series,” 2021)

Escanciano and Hualde (“Measuring Asset Market Linkages: Nonlinear Dependence and Tail Risk,” 2021)

Giudici et al. (“Network Models to Enhance Automated Cryptocurrency Portfolio Management,” 2020)

Giudici et al. (“Network models to improve robot advisory portfolios,” 2022)

Jackson and Pernoud (“Systemic Risk in Financial Networks: A Survey,” 2020)

Jiang et al. (“Tail-event driven network of cryptocurrencies and conventional assets,” 2022)

Kalyagin et al. (“Reliability of MST identification in correlation-based market networks,” 2021)

Katsouris (“Optimal Portfolio Choice and Stock Centrality for Tail Risk Events,” 2021)

Kaya (“Eccentricity in Asset Management,” 2015)

Kaya (“Managing ambiguity in asset allocation,” 2017)

Loistl and Konstantinov (“Interactions and Interconnectedness Shape Financial Market Research,” 2020)

Konstantinov et al. (“A network and machine learning approach to factor, asset, and blended allocation,” 2020)

Konstantinov and Rusev (“The Bond-Equity-Fund Relation Using the Fama-French-Carhart Factors: A Practical Network Approach,” 2020)

Konstantinov and Simonian (“A Network Approach to Analyzing Hedge Fund Connectivity,” 2020)

Kukreti et al. (“A Perspective on Correlation-Based Financial Networks and Entropy Measures,” 2020)

Kumar et al. (“Ripples on financial networks,” 2021)

Magner et al. (“The Volatility Forecasting Power of Financial Network Analysis,” 2021)

Marti et al. (“A review of two decades of correlations, hierarchies, networks and clustering in financial markets,” 2021)

Millington and Niranjan (“Stability and similarity in financial networks – How do they change in times of turbulence?” 2021)

Olmo (“Optimal portfolio allocation and asset centrality revisited,” 2021)

Pang et al. (“An analysis of network filtering methods to sovereign bond yields during COVID-19,” 2021)

Peralta and Zareei (“A network approach to portfolio selection,” 2016)
Samal et al. (“Network-centric indicators for fragility in global financial indices,” 2021)
Son and Lee (“Graph-based multi-factor asset pricing model,” 2022)
Stavroglou (“Finding Hidden Structures in Financial Networks,” 2020)
Vyrost et al. (“Network-based asset allocation strategies,” 2019)
Yang et al. (“Portfolio optimization with idiosyncratic and systemic risks for financial networks,” 2021)
Yang et al. (“Portfolio optimization based on empirical mode decomposition,” 2019)
Zhan et al. (“Graphical Models for Financial Time Series and Portfolio Selection,” 2021)
Zhao et al. (“Stock market as temporal network,” 2018)
Zhao et al. (“Community detection and portfolio optimization,” 2021)

2.2.3 Network analysis and clustering

List of references:
Abboud et al. (“Subquadratic High-Dimensional Hierarchical Clustering,” 2019)
Ackerman et al. (“Weighted clustering: Towards solving the user’s dilemma,” 2021)
Adolfsson et al. (“To cluster, or not to cluster: An analysis of clusterability methods,” 2019)
Alfarra et al. (“Rethinking Clustering for Robustness,” 2021)
Bandara et al. (“Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach,” 2020)
Bouveyron et al. (*Model-Based Clustering and Classification for Data Science: With Applications in R*, 2019)
Brécheteau et al. (“Robust Bregman clustering,” 2021)
Campello et al. (“Density-based clustering,” 2020)
Casa et al. (“Better than the best? Answers via model ensemble in density-based clustering,” 2021)
Chavent et al. (“Combining clustering of variables and feature selection using random forests,” 2021)
Chehreghani (“Shift of Pairwise Similarities for Data Clustering,” 2021)
Chung et al. (“Statistical Connectomics,” 2022)
Coraggio and Coretto (“Selecting the number of clusters, clustering models, and algorithms. A unifying approach based on the quadratic discriminant score,” 2021)
Dalmia and Sia (“Clustering with UMAP: Why and How Connectivity Matters,” 2021)
De Luca and Zuccolotto (“Hierarchical time series clustering on tail dependence with linkage based on a multivariate copula approach,” 2021)
Den Teuling et al. (“Clustering of longitudinal data: A tutorial on a variety of approaches,” 2021)
Djouzi and Beghdad-Bey (“A Review of Clustering Algorithms for Big Data,” 2019)
Doreian et al. (*Advances in Network Clustering and Blockmodeling*, 2020)
Duan and Dunson (“Bayesian Distance Clustering,” 2021)
Elliott et al. (“Core–periphery structure in directed networks,” 2020)
Esmaeili et al. (“Probabilistic Fair Clustering,” 2021)
Exarchakis et al. (“A sampling-based approach for efficient clustering in large datasets,” 2022)
Ezhilmaran and Indira (“A survey on clustering techniques in pattern recognition,” 2020)
Faletto and Bien (“Cluster Stability Selection,” 2022)
Feng and Simon (“Ensembled sparse-input hierarchical networks for high-dimensional datasets,” 2020)
Franti and Sieranoja (“How much can k-means be improved by using better initialization and repeats?” 2019)
Fu and Perry (“Estimating the Number of Clusters Using Cross-Validation,” 2020)
Gagolewski (“genieclust: Fast and robust hierarchical clustering,” 2021)
Gao and Tsay (“Divide-and-Conquer: A Distributed Hierarchical Factor Approach to Modeling Large-Scale Time Series Data,” 2021)
Gherbaoui et al. (“Generation of Gaussian sets for clustering methods assessment,” 2021)
Ghosal et al. (“A Short Review on Different Clustering Techniques and Their Applications,” 2019)
Goel and Majumdar (“Transformed K-means Clustering,” 2021)
Grun (“Model-based Clustering,” 2018)
Guan and Loew (“A Distance-based Separability Measure for Internal Cluster Validation,” 2021)
Guijo-Rubio et al. (“Time-Series Clustering Based on the Characterization of Segment Typologies,” 2020)
Hua (“Clusterability, Model Selection and Evaluation,” 2019)

Huang et al. (“Learning From Networks: Algorithms, Theory, and Applications,” 2019)

Irani et al. (“Clustering Techniques and the Similarity Measures used in Clustering: A Survey,” 2016)

Javed et al. (“A Benchmark Study on Time Series Clustering,” 2020)

Jose-Garcia and Gomez-Flores (“A survey of cluster validity indices for automatic data clustering using differential evolution,” 2021)

Kawamoto and Kabashima (“Comparative analysis on the selection of number of clusters in community detection,” 2017)

Kawamoto and Kabashima (“Cross-validation estimate of the number of clusters in a network,” 2017)

Keranovic et al. (“Estimating the Number of Latent Factors in High-Dimensional Financial Time Series,” 2020)

Kumari and Sharma (“A review for the efficient clustering based on distance and the calculation of centroid,” 2020)

Landi et al. (“reval: a Python package to determine best clustering solutions with stability-based relative clustering validation,” 2020)

Lemenkova (“R Libraries (dendextend and magrittr) and Clustering Package scipy.cluster of Python For Modelling Diagrams of Dendrogram Trees,” 2020)

Leopold and Rose (“UNIC: A fast nonparametric clustering,” 2020)

Li et al. (“Network cross-validation by edge sampling,” 2020)

Li and Liu (“Multivariate time series clustering based on complex network,” 2021)

Lipor and Balzano (“Clustering quality metrics for subspace clustering,” 2020)

Ma (*Network Analysis Made Simple*, 2020)

Ma and Seth (*Network Analysis Made Simple*, 2020)

Maharaj et al. (*Time Series Clustering and Classification*, 2019)

Mahfuz et al. (“Review of single clustering methods,” 2019)

Mehta et al. (“Analytical review of clustering techniques and proximity measures,” 2020)

Millington and Niranjan (“Construction of Minimum Spanning Trees from Financial Returns using Rank Correlation,” 2020)

Peng et al. (“Multi-dimensional clustering through fusion of high-order similarities,” 2022)

Pimentel and de Carvalho (“A Meta-learning approach for recommending the number of clusters for clustering algorithms,” 2020)

Policastro et al. (“ROBustness In Network (robin): an R package for Comparison and Validation of communities,” 2021)

Rahgoshay and Salavatipour (“Hierarchical Clustering: New Bounds and Objective,” 2021)

Rehman and Belhaouari (“Divide well to merge better: A novel clustering algorithm,” 2022)

Romashchenko (“Clustering with Respect to the Information Distance,” 2021)

Sarda-Espinosa (“Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package,” 2019)

Sato-Ilic (“Cluster-scaled principal component analysis,” 2021)

Seabrook et al. (“Evaluating structural edge importance in temporal networks,” 2021)

Silva et al. (“Novel Features for Time Series Analysis: A Complex Networks Approach,” 2021)

Silva et al. (“Time series analysis via network science: Concepts and algorithms,” 2021)

Sobczyk et al. (“VARCLUST: clustering variables using dimensionality reduction,” 2020)

Stankovic et al. (“Data Analytics on Graphs Part I: Graphs, Graph Spectra, and Spectral Clustering,” 2020)

Stankovic et al. (“Data Analytics on Graphs Part II: Signals on Graphs,” 2020)

Stankovic et al. (“Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications,” 2020)

Tong et al. (“A density-peak-based clustering algorithm of automatically determining the number of clusters,” 2021)

Thrun (“The Exploitation of Distance Distributions for Clustering,” 2021)

Thrun and Stier (“Fundamental clustering algorithms suite,” 2021)

Underwood et al. (“Motif-based spectral clustering of weighted directed networks,” 2020)

Vankwikelberge et al. (“ExClus: Explainable Clustering on Low-dimensional Data Representations,” 2021)

Vázquez et al. (“An ensemble solution for multivariate time series clustering,” 2021)

Wadhwa and Scott (“Exploring complex networks with the ICON R package,” 2020)

Wang and Tsay (“Clustering Multiple Time Series with Structural Breaks,” 2019)

Wang et al. (“On the Efficiency of K-Means Clustering: Evaluation, Optimization, and Algorithm Selection,” 2020)

Wu et al. (“Analyses and applications of optimization methods for complex network reconstruction,” 2020)

Wu et al. (“Effective hierarchical clustering based on structural similarities in nearest neighbor graphs,” 2021)

Yelibi and Gebbie (“Agglomerative Likelihood Clustering,” 2021)

Zambelli (“Ensemble Method for Cluster Number Determination and Algorithm Selection in Unsupervised Learning,” 2021)

Zhang (“Weighted Clustering Ensemble: A Review,” 2021)

Zheng et al. (“Multi-view subspace clustering networks with local and global graph information,” 2021)

Zhong et al. (“Ensemble clustering based on evidence extracted from the co-association matrix,” 2019)

Zhong et al. (“Ensemble clustering based on evidence extracted from the co-association matrix,” 2019)

Zhong and Pun (“Subspace clustering by simultaneously feature selection and similarity learning,” 2020)

Zhou et al. (“Unsupervised feature selection for balanced clustering,” 2020)

2.2.4 Testing and comparison procedures for investment portfolios

References:

Adcock et al. (“Portfolio Performance Measurement: Monotonicity with Respect to the Sharpe Ratio and Multivariate Tests of Correlation,” 2017)

Arnott et al. (“A backtesting protocol in the era of machine learning,” 2019)

Bailey et al. (“Stock Portfolio Design and Backtest Overfitting,” 2017)

Bessler and Wolff (“Portfolio Optimization with Industry Return Prediction Models,” 2017)

Bessler et al. (“Multi-asset portfolio optimization and out-of-sample performance: an evaluation of Black Litterman, mean-variance, and naive diversification approaches,” 2017)

Bjerring et al. (“Feature selection for portfolio optimization,” 2017)

Bruni et al. (“On exact and approximate stochastic dominance strategies for portfolio selection,” 2017)

Bruni et al. (“Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models,” 2016)

Bryzgalova et al. (“Bayesian solutions for the factor zoo: we just ran two quadrillion models,” 2021)

Cesarone et al. (“On the stability of portfolio selection models,” 2019)

Cesarone et al. (“Why Small Portfolios Are Preferable and How to Choose Them,” 2018)

Chaudhuri and Lo (“Dynamic Alpha: A Spectral Decomposition of Investment Performance Across Time Horizons,” 2019)

Diris et al. (“Long-Term Strategic Asset Allocation: An Out-of-Sample Evaluation,” 2015)

Fabozzi and Lopez de Prado (“Being Honest in Backtest Reporting: A Template for Disclosing Multiple Tests,” 2018)

Greiner and Stoyanov (“Portfolio scoring by expected risk premium,” 2019)

Guidolin et al. (“Portfolio performance of linear SDF models: an out-of-sample assessment,” 2018)

Guo (“A Statistical Response to Challenges in Vast Portfolio Selection,” 2019)

Guo et al. (“When Does The 1/N Rule Work?” 2019)

Haley (“K-fold cross validation performance comparisons of six naive portfolio selection rules: how naive can you be and still have successful out-of-sample portfolio performance?” 2017)

Harvey et al. (“An Evaluation of Alternative Multiple Testing Methods for Finance Applications,” 2020)

Hens et al. (“Escaping the backtesting illusion,” 2020)

Hsu et al. (*Do Cross-Sectional Stock Return Predictors Pass the Test without Data-Snooping Bias?* 2017)

Hsu et al. (“Asset allocation strategies, data snooping, and the 1 / N rule,” 2018)

Huang and Yu (“A new procedure for resampled portfolio with shrinkaged covariance matrix,” 2020)

Hwang et al. (“Naive versus optimal diversification: Tail risk and performance,” 2018)

Ielpo et al. (*Engineering Investment Process: Making Value Creation Repeatable*, 2017)

Jaeger et al. (“Understanding machine learning for diversified portfolio construction by explainable AI,” 2020)

Kazak and Pohlmeier (“Testing out-of-sample portfolio performance,” 2019)

Kazak and Pohlmeier (*Portfolio Pretesting with Machine Learning*, 2020)

Kuntz (“Portfolio Strategies with Classical and Alternative Benchmarks,” 2018)

Lohre et al. (“Hierarchical Risk Parity: Accounting for Tail Dependencies in Multi-asset Multi-factor Allocations,” 2020)

Lopez de Prado (“A Data Science Solution to the Multiple-Testing Crisis in Financial Research,” 2019)

Lopez de Prado and Lewis (“Detection of false investment strategies using unsupervised learning methods,” 2019)

Malavasi et al. (“Second order of stochastic dominance efficiency vs mean variance efficiency,” 2021)

Mooney et al. (“Dynamic Regime Strategy for Stress Testing and Optimizing Institutional Investor Portfolios,” 2020)

Platanakis et al. (“Horses for Courses: Mean-Variance for Asset Allocation and 1/N for Stock Selection,” 2021)

Radovanov and Marcikic (“Testing The Performance Of The Investment Portfolio Using Block Bootstrap Method,” 2014)

Rebonato (“A financially justifiable and practically implementable approach to coherent stress testing,” 2019)

Schumann (“Backtesting,” 2019)

Seymour et al. (“Dynamic portfolio management strategies: A framework for historical analysis,” 2018)

Suhonen et al. (“Quantifying Backtest Overfitting in Alternative Beta Strategies,” 2017)

Taljaard and Maré (“Why has the equal weight portfolio underperformed and what can we do about it?” 2021)

Tayali (“A novel backtesting methodology for clustering in mean–variance portfolio optimization,” 2020)

Traccucci et al. (“A Triptych Approach for Reverse Stress Testing of Complex Portfolios,” 2019)

Valentine et al. (“Beyond p values: utilizing multiple methods to evaluate evidence,” 2019)

Vincent et al. (“Analyzing the Performance of Multifactor Investment Strategies under a Multiple Testing Framework,” 2018)

Vovk and Wang (“True and false discoveries with e-values,” 2020)

Vovk and Wang (“E-values: Calibration, combination, and applications,” 2021)

Wiecki et al. (“All That Glitters Is Not Gold: Comparing Backtest and Out-of-Sample Performance on a Large Cohort of Trading Algorithms,” 2016)

Yu (“Comparing Classical Portfolio Optimization and Robust Portfolio Optimization on Black Swan Events,” 2021)

Yuan and Zhou (“Why Naive 1/N Diversification Is Not So Naive, and How to Beat It?” 2022)

Zhang et al. (“DoubleEnsemble: A New Ensemble Method Based on Sample Reweighting and Feature Selection for Financial Data Analysis,” 2020)

Zhang et al. (“Information Coefficient as a Performance Measure of Stock Selection Models,” 2020)

Zhang et al. (“Deep Learning for Portfolio Optimization,” 2020)

2.2.5 Software implementations and frameworks

List of references:

Bonald et al. (“Scikit-network: Graph Analysis in Python,” 2020)

Charrad et al. (“NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set,” 2014)

de Miranda Cardoso et al. (“Algorithms for Learning Graphs in Financial Markets,” 2020)

Ferraro et al. (“fclust: An R Package for Fuzzy Clustering,” 2019)

Fischer et al. (“REPPlab: An R package for detecting clusters and outliers using exploratory projection pursuit,” 2021)

Fop and Murphy (“Variable Selection Methods for Model-based Clustering,” 2017)

Gagolewski (“genieclust: Fast and robust hierarchical clustering,” 2021)

Haddad and Bouguessa (“TopoDetect: Framework for Topological Features Detection in Graph Embeddings,” 2021)

Landi et al. (“reval: a Python package to determine best clustering solutions with stability-based relative clustering validation,” 2020)

Lemenkova (“R Libraries (dendextend and magrittr) and Clustering Package scipy.cluster of Python For Modelling Diagrams of Dendrogram Trees,” 2020)

Louiset et al. (“UCSL : A Machine Learning Expectation-Maximization framework for Unsupervised Clustering driven by Supervised Learning,” 2021)

McCabe et al. (“netrd: A library for network reconstruction and graph distances,” 2020)

Miranda et al. (“HiClass: a Python library for local hierarchical classification compatible with scikit-learn,” 2022)

Montero and Vilar (“TSclust: An R Package for Time Series Clustering,” 2015)

Mori et al. (“Distance Measures for Time Series in R: The TSdist Package,” 2016)

Rusch et al. (“Cluster Optimized Proximity Scaling,” 2021)

Ruta et al. (“SAX Navigator: Time Series Exploration through Hierarchical Clustering,” 2020)

Sarda-Espinosa (“Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package,” 2019)

Sekula et al. (“optCluster: An R Package for Determining the Optimal Clustering Algorithm,” 2017)

Sobczyk et al. (“VARCLUST: clustering variables using dimensionality reduction,” 2020)

Tellaroli et al. (“Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters,” 2016)

Valk and Cybis (“U-Statistical Inference for Hierarchical Clustering,” 2021)

Wang et al. (“Thresher: determining the number of clusters while removing outliers.,” 2018)

Weylandt et al. (“Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization,” 2019)

Yu et al. (“Bootstrapping estimates of stability for clusters, observations and model selection,” 2018)

3 Practical details for the project

The main purpose of the project described in this document is to provide exposure to students on important (and interesting) practical topics in quantitative wealth and investment management QWIM.

The level of complexity depends on the number of hours designated for the project. For example, 50-60 hours for a regular project, and 100-120 hours for a thesis/capstone project. Upon request, the scope (and the corresponding number if hours) of any given project can be extended.

The students would work on the project as part of a team (usually with 2-3 students).

All QWIM projects were selected such that the students' efforts have a good chance of producing results relevant to the industry, and at least as good as the results presented in the QWIM literature. Thus for each project we may consider (on an optional basis, based primarily on students' preference) to submit a corresponding article to journals widely followed by practitioners and academics in investment and wealth management, with participating students included as the leading coauthors of the submitted article.

The main challenge for each project is to identify the criteria for what would be considered **"good enough"**. Similar to projects in the industry, the meaning of "good enough" is based on a combination of comprehensive literature review, discussions within team and with me (and/or my colleagues) and analysis of results. Emphasis is placed on creating a narrative (with the aid of an interactive visualizer) for convincing the intended audience that what was done in the project delivers **"good enough"** outcome.

3.1 Interaction with students

For each project I would make myself available for meetings on a weekly basis (for discussions and guidance). Some of my colleagues have also expressed interest to participate in such meetings. Due to our work schedule and deliverables, most of the discussions will have to be scheduled outisde working hours (in weekends or evenings). The meetings will take place through video conferencing such as WebEx, Zoom, Google Meet, Microsoft Teams, etc., based on the team's preference. If the meetings are through WebEx, I would provide a link, while the student team will provide a link for any other video conferencing tool.

The students working on a given project can also send questions by email (my recommendation is to aggregate the questions from team members into an email sent once a day). We aim to provide answers within 1-2 days, either by email or through a phone discussion.

3.2 Data

Due to compliance reasons all projects would be based on publicly available, non-proprietary and non-confidential data (indices, ETFs, mutual funds, etc.). Since neither I nor my team are allowed to provide these datasets, I can only provide a list of suggested datasets. This list is included in a later section named Practical Info.

The datasets were selected to have the following features:

- be good proxies for most representative asset and subasset classes
- to be widely available
- to be as liquid as possible
- to have daily granularity
- to encompass periods with as many market regimes as possibles (most proposed daily datasets are from 1990 or 1991)
- time series have "nicer" statistical properties compared to time series of, say, individual stocks or bonds

3.3 Private GitHub repository for the QWIM project

The team will create a private GitHub repository, which will store relevant project materials, including codes. The team will use Git Desktop application as source control repository linked to the GitHub repository.

3.4 Deliverables

The project deliverables include literature survey, numerical results, analysis and visualization. For each project references will be provided for a comprehensive literature survey, and students are encouraged to identify additional relevant literature. Regarding the implementation, the project will primarily use existing codes:

- Python and R packages from official repositories (PyPi for Python and CRAN for R)
- machine learning platforms such as TensorFlow, PyTorch, CNTK, Chainer, mlr3, H20, PlaidML, mlpack, etc.
- implementations of articles through codes available in repositories such as GitHub, BitBucket, GitLab, etc.

Visualization of data and results visualization will be interactive and it will be based on Shiny R framework; to reduce programming effort, a template for such a Shiny visualizer will be provided in the team private GitHub repository.

The deliverables are:

- written report including literature survey and numerical results
- interactive visualizer (most likely Shiny-based visualizer using R and Python packages)
- (optional) presentation slides, and/or RMarkdown presentation, and/or Jupyter Notebook(s)

3.5 (Optional) Article submission to leading journals

On an optional basis (based primarily on students' preference), a version of the report can be prepared for submision to leading journals such as Journal of Financial Data Science, Journal of Portfolio Management, Journal of Asset Management, Journal of Investment Strategies, Quantitative Finance, Journal of Wealth Management, Journal of Investing, Journal of Machine Learning in Finance, etc.

4 Project tasks and timelines

For each project the main tasks are:

1) literature review

2) decide on the appropriate metrics and quantitative methods within context of "good enough" for the project

3) write-up summary of literature review: methods, metrics, testing procedures

4) identification of Python and/or R packages which are most appropriate for the selected methods and metrics

5) code design to decide on main code components

6) implementation of code components

7) interactive visualization of numerical results

8) project report containing description of methods, metrics, and tests, and analysis of results.

4.1 Suggested timelines for project tasks

The table below suggests a timeline for the project tasks and the corresponding percentages of project time:

Table 1: Suggested timeline for project tasks

Task ID	Task Name	Percentage of project time
1	Literature review	15%
2	Identification of "good enough" metrics and quantitative methods	5%
3	Write-up of summary of literature review	5%
4	Identification of appropriate packages in Python and/or R	10%
5	Code design for main components of project coding framework	5%
6	Implementation of coding framework and components	40%
7	Interactive visualizer using the provided Shiny template	10%
8	Project report and presentation	10%

4.2 Literature review

The first task is based on a comprehensive literature survey, included in the preliminary document of the project. Students are encouraged to identify additional relevant literature.

This task may be the most important of the project, since it provides an overview of what was done, what works well and less well, and what appear to be the most promising avenues to complete the project.

Emphasis is placed on information contained in the Main References; the other References would be considered only if time permits and the team is interested in exploring other avenues.

When reading the literature, there are 4 main directions to consider:

1) methods

2) metrics to assess the performance/robustness of the methods

3) testing procedures

4) numerical results

The primary focus would be on the the references included in "Main References" subsection of the document for your QWIM project. Then, to the extent there is time, to consider the other references included in the project document. In the same time, you are encouraged to identify other references that might be considered "Main references", and to share those references with me for discussion.

For the articles in Main References category, the suggested approach would be the following:

- For each article focus primarily on Abstract, Conclusion, and Numerical Results
- Do this for all articles considered to be Main References, such that you gain a high-level understanding of what is currently done in the literature
- Select the metrics that you may want to use in order to quantify the meaning of "good enough" for the project.
- Select the quantitative methods which appear to be most likely to be "good enough" for the project.
- Perform a "deeper dive" into the articles containing the approaches you consider the most promising,

For the articles which are not in "Main References" category, read Abstract, Conclusion, and Numerical Results, to see whether any of those articles might need to be considered for inclusion in your summary.

4.3 Write-up summary of literature review

The write-up summary summarizes the methods, metrics, testing procedures, and numerical results identified during the literature review. The write-up could also be incorporated within reports and/or presentations for the QWIM project.

4.4 Identification of appropriate Python and/or R packages

Based on the literature review and on diiscussions, we identify the most potentially useful methods, metrics and testing procedures. Then wee identify the most appropriate implementations of the selected methods and metrics.

The primary sources of implementatins are existing codes from:

- Python and R packages from official repositories (PyPi for Python and CRAN for R)
- machine learning platforms such as TensorFlow, PyTorch, CNTK, Chainer, mlr3, H20, PlaidML, mlpack, etc.
- Codes available in repositories such as GitHub, BitBucket, GitLab, etc.

4.5 Code design

An important task is to have a code design session to decide in advance on the main code components, which are meant to be modular and encapsulated, such that the entire team can work on the codes.

Examples of such main components include extracting data, calculate metrics for the considered procedures, portfolio metrics, performing tests, construct interactive visualizer, etc.

The code design procedure consists of:

1) visual display of major components of the coding framework
2) UML diagrams for each of the components.

The Appendix contains an illustrative example within context of a QWIM project on forecasting of financial time series. The first figure shows the major components, while the second figure shows UML diagrams of those components (the names of data members and methods are currently generic, and one would need to change them to appropriate names)

While these figures were obtained through Microsoft Visio using a code design file (.vsd file), there are other software tools (either online or installed locally) which can be used to create such code design diagrams. NOTE: if you have access to Microsoft Visio and you want to use it for code design diagrams, you can ask me for the .vsd file which was exported into the PDF from which I have extracted the snapshots.

List of software tools for code design diagrams, which are either free (open source) or have a free type of account

- Modelio (either desktop version or online version)
- LucidChart (online)
- draw.io (either desktop version or online version, now called app.diagrams.net)

- Visual Paradigm (online)
- UMLet (either desktop or online version)
- Curated list of UML tools – 2019 edition
- Top online UML modeling tools in 2019

4.6 Implementation of coding framework and components

The implementation is done using identified packages or codes, in Python and/or R. The project will primarily use existing codes:

- Python and R packages from official repositories (PyPi for Python and CRAN for R)
- machine learning platforms such as TensorFlow, PyTorch, CNTK, Chainer, mlr3, H20, PlaidML, mlpack, etc.
- implementations of articles through codes available in repositories such as GitHub, BitBucket, GitLab, etc.

4.7 Interactive visualizer

While visualization of data and numerical results can be done through various tools (including Jupyter notebooks or Dash in Python), my recommendation is to consider an interactive visualizer based on Shiny framework in R. A template for the Shiny visualizer will be provided in the private GitHub repository set up by the team for the project.

Some information about Shiny:

- Shiny from RStudio: tutorials and gallery
- Why R Shiny Trumps UI and JavaScript Based Visualization Tools
- Shiny's Holy Grail: Interactivity with reproducibility

4.8 Project report and presentation

The report containing description of methods, metrics, and tests, and analysis of results.

While the report can be written using various tools (including Microsoft Word), my recommendation is to use LyX to write both the project report and the project presentation. Two LyX templates for creating reports and, respectively, presentations will be provided in the private GitHub repository set up by the team for the project.

Some information about Shiny:

- LyX features
- LyX tutorial with PDF here
- LyX Tutorial video Part One and Part Two
- LyX tutorial video Part One and Part Two and Part Three and Part Four
- Introduction to LyX
- Insert figures in LyX
- Essentials of LyX

5 Design and implementation for the project codes

This section describes a possible approach for the design process and for the implementation (folder structure) of the project. This approach is presented only to exemplify how it could be done. Each student team has freedom to consider their own design process.

Design and implementation would be based on following principles:

- coding framework is Python based, with calls to functions available in existing Python and R packages
- leverage common components (such as data input/output, numerical methods, time series, testing, interactive visualization and reporting, etc.)
- reusability
- incorporate best practices in coding and numerical implementations
- use, augment and enhance (to largest extent possible) existing Python and R packages and codes

5.1 Visualize project workflow and coding framework

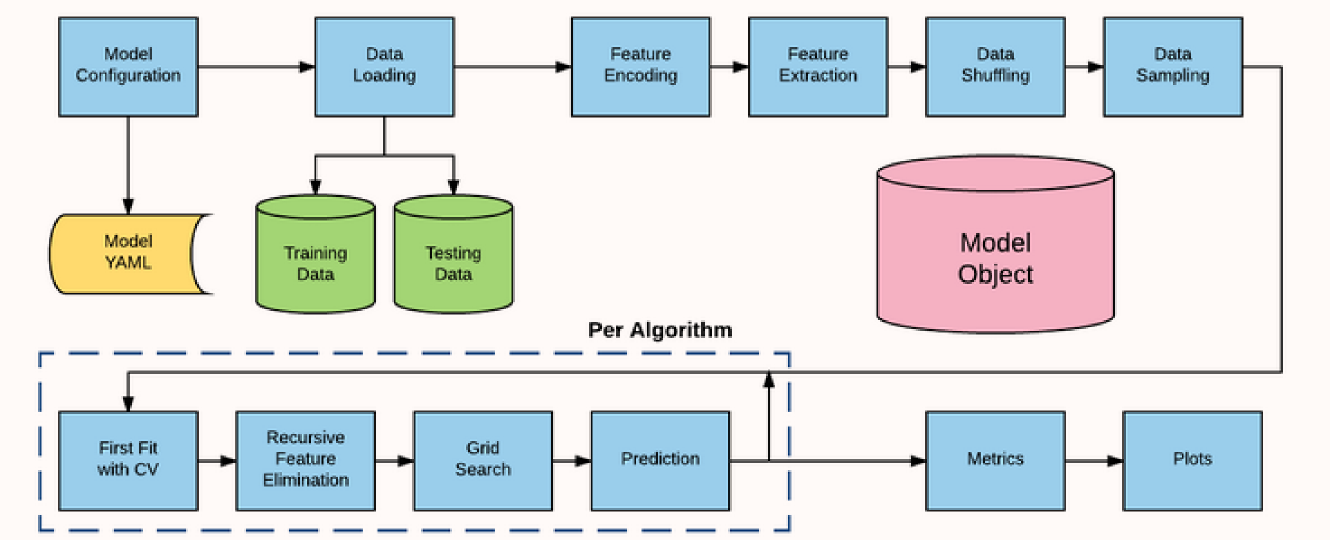
The starting point is to visualize the project workflow in terms of major components, and then to design the code framework.

The code design procedure consists of:

1) visual display of major components of the coding framework
2) UML diagrams for each of the components.

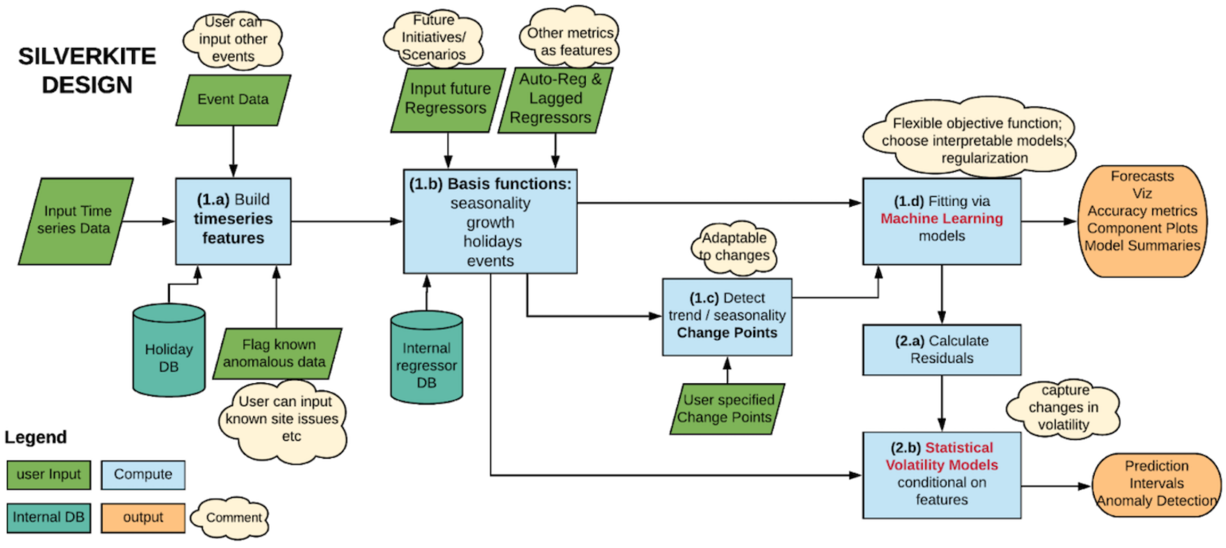
We present examples below for projects including time series forecasting and analysis, machine learning for portfolio construction, etc.

Figure 1: Examples of architecture of coding framework: AlphaPy



Source: AlphaPy

Figure 2: Examples of architecture of coding framework: Greykite


SILVERKITE DESIGN
User can input other events
Event Data
Future Initiatives/ Scenarios
Other metrics as features
Input future Regressors
Auto-Reg & Lagged Regressors
Flexible objective function; choose interpretable models; regularization
Input Time series Data
(1.a) Build timeseries features
(1.b) Basis functions: seasonality growth holidays events
(1.d) Fitting via Machine Learning models
Forecasts Viz Accuracy metrics Component Plots Model Summaries
Adaptable to changes
(1.c) Detect trend / seasonality Change Points
Holiday DB
Flag known anomalous data
Internal regressor DB
(2.a) Calculate Residuals
User can input known site issues etc
User specified Change Points
capture changes in volatility
(2.b) Statistical Volatility Models conditional on features
Prediction Intervals Anomaly Detection
Legend
user Input
Compute
Internal DB
output
Comment

Source: Geykite

Figure 3: QLib Framework

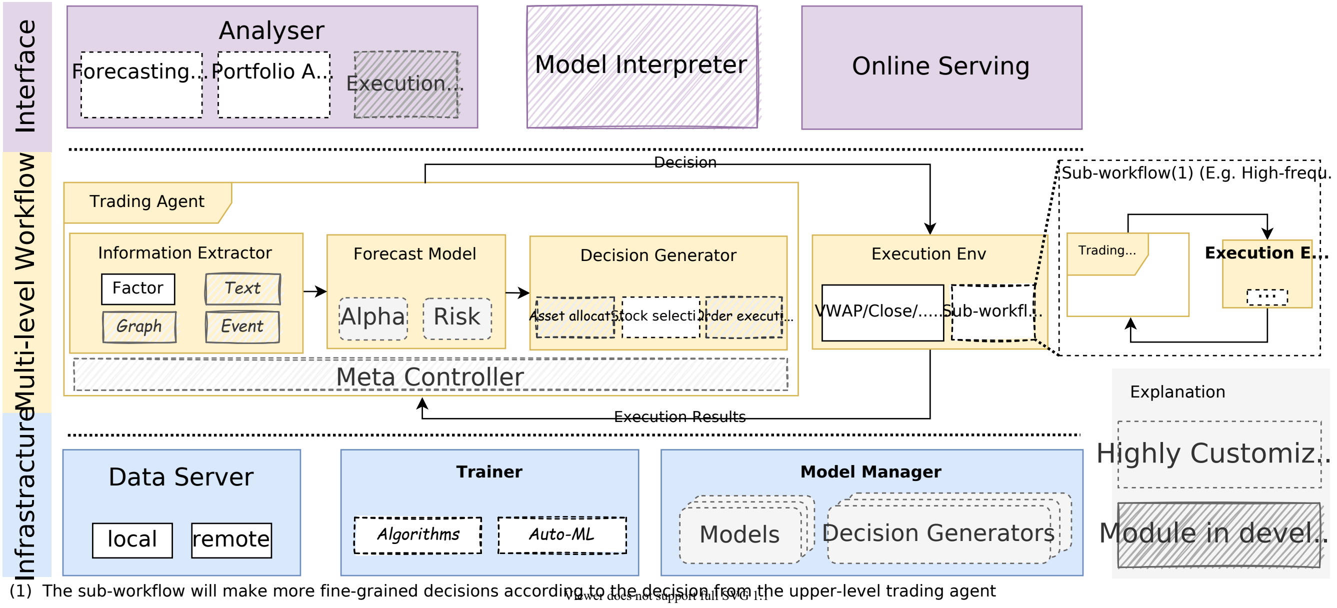


Figure 4: Examples of major components of coding framework (top) and UML diagrams (bottom)

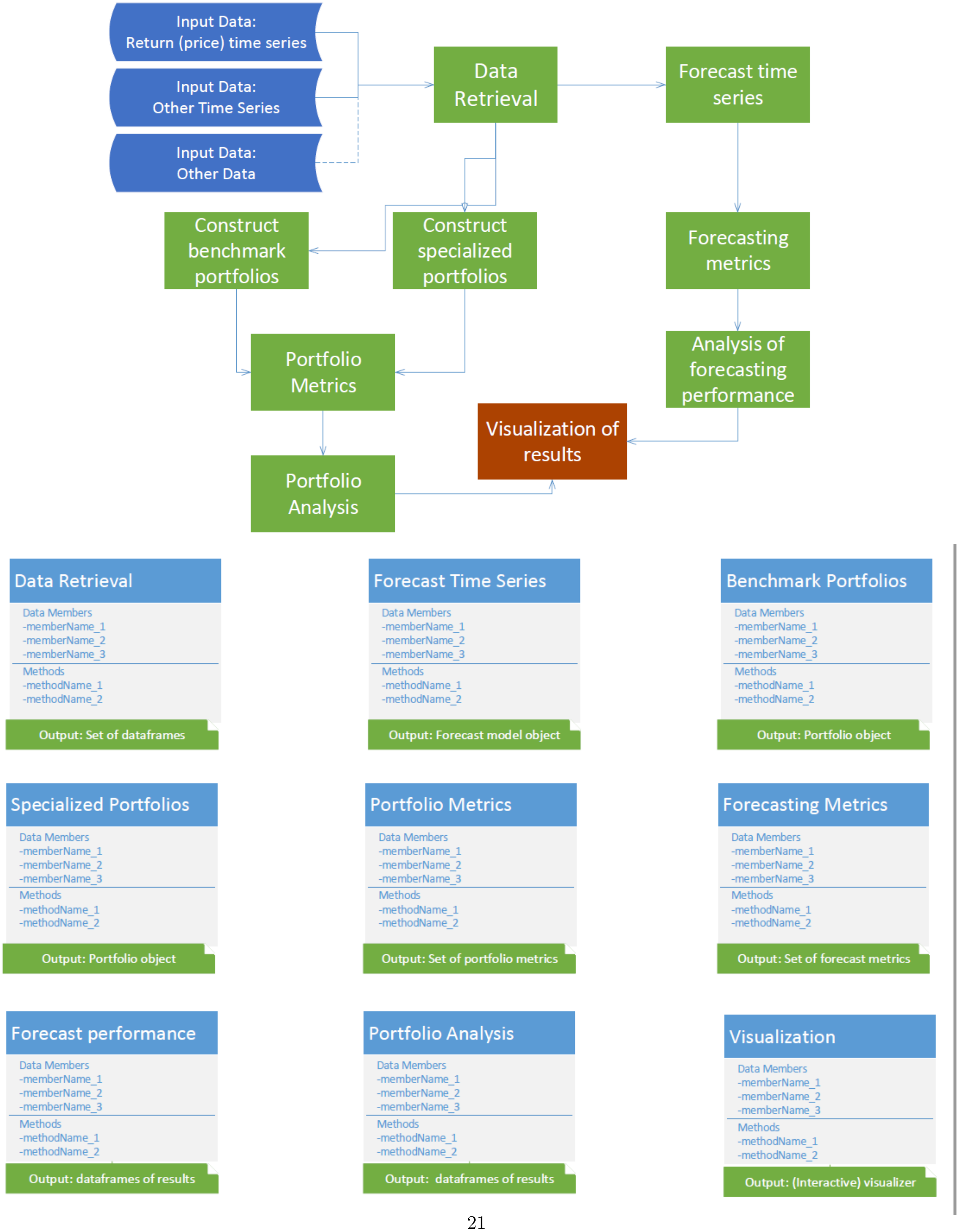
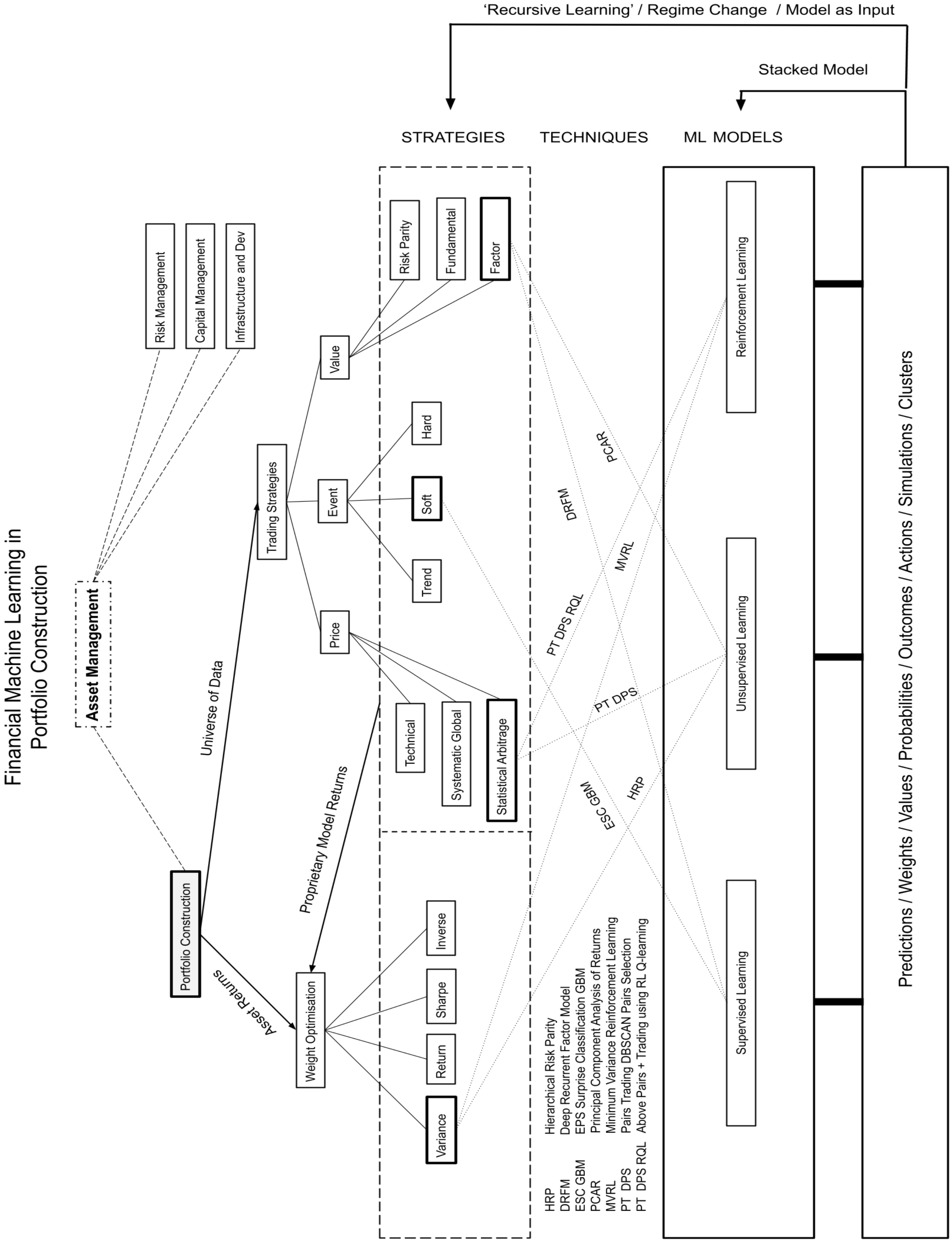


Figure 5: Financial Machine Learning in Portfolio Construction


Financial Machine Learning in Portfolio Construction
Asset Management
Risk Management
Capital Management
Infrastructure and Dev
Portfolio Construction
Universe of Data
Asset Returns
Trading Strategies
Value
Event
Price
Proprietary Model Returns
Weight Optimisation
STRATEGIES
TECHNIQUES
ML MODELS
'Recursive Learning' / Regime Change / Model as Input
Stacked Model
Risk Parity
Fundamental
Factor
Hard
Soft
Trend
Technical
Systematic Global
Statistical Arbitrage
Inverse
Sharpe
Return
Variance
PCAR
DRFM
MVRL
PT DPS RQL
PT DPS
ESC GBM
HRP
Reinforcement Learning
Unsupervised Learning
Supervised Learning
Predictions / Weights / Values / Probabilities / Outcomes / Actions / Simulations / Clusters
HRP Hierarchical Risk Parity
DRFM Deep Recurrent Factor Model
ESC GBM EPS Surprise Classification GBM
PCAR Principal Component Analysis of Returns
MVRL Minimum Variance Reinforcement Learning
PT DPS Pairs Trading DBSCAN Pairs Selection
PT DPS RQL Above Pairs + Trading using RL Q-learning

Source: Machine Learning in Asset Management

5.2 Representative examples of Python libraries with well designed folder structure

List of Python libraries

- QLib is a AI-oriented quantitative investment platform in Python developed by Microsoft researchers
- GluonTS is a Python library deveoped by Amazon researchers for probabilistic time series modeling
- sktime is a unified framework for machine learning with time series, developed by researchers at Alan Turing Institute for data science and artificial intelligence.
- darts is a Python library for easy manipulation and forecasting of time series, developed by researchers at Unit8 AI and data analytics company.
- Kats is a Python library developed by Facebook researchers to analyze time series data.
- Kats is a Python library developed by Tinkoff AI researchers to analyze time series data.
- MLFinLab (Machine Learning Financial Laboratory) is a Python library developed by researchers at Hudson & Thames.

6 Practical Info

6.1 Recommended software tools

The sections below describe the recommended software tools, including corresponding versions/subversions, tutorials and details

6.1.1 Python

The recommended versions are:

- Python version 3.8 (subversion Python 3.8.10 or 3.8.15)
- Python version 3.9 (subversion Python 3.9.10 or 3.9.15)
- Python version 3.10 (latest subversion, currently Python 3.10.8)

There are also relevant Python packages, identified while you are working the project. As a starting point you can consider the packages included in section on Potentially useful Python and R packages.

6.1.2 R

The recommended versions are:

- R version 4.2 (recommended is latest subversion, currently R 4.2.2)
- R version 3.6 (subversion R 3.6.3)

On Windows computers you also need to install Rtools to build R packages from source through compilation, since not all packages have associated Windows binaries.

There are also relevant R packages, identified while you are working the project. As a starting point you can consider the packages included in section on Potentially useful Python and R packages.

6.1.3 R IDE

The recommended R IDE is RStudio Desktop Open Source

- latest version, currently 2022.07.2+576

6.1.4 Python IDE

The recommended Python IDE is Visual Studio Code VSC

- latest version, currently VSC 1.73

Then add Python extension and other Visual Studio Code extensions from Visual Studio MarketPlace.

Note: Upon request I can provide a list of potentially useful VSC extensions, which can be installed on your computer (see for example link)

6.1.5 Bibliography Manager

The recomemnded bibliography manager is JabRef

- latest version: version 5.7, or
- latest development version from link

I can provide you with a bibliography file which contains all refeernces mentioned in the project description, This file (of extension bib) can be viewed and edited with JabRef, and used together with LyX to write your project related documents (report, presentation, etc.).

You can easily add/delete/edit this bib file using JabRef.

There are video tutorials on using JabRef: link 1, link 2, link 3.

In addition to these video tutorials, I can also have a video online session, to provide an overview and answer your questions on using LyX and JabRef. This online session (through Google Meet Google Meet) can be recorded and shared with you afterwards.

6.1.6 Document processor

The recommended document processor is LyX, which is a document processor that encourages an approach to writing based on the structure of your documents (WYSIWYM) and not simply their appearance (WYSIWYG).

LyX combines the power and flexibility of TeX/LaTeX with the ease of use of a graphical interface. It shoudl be emohasized that you do not need to know/learn LaTeX in order to tuse LyX.

To install LyX, you need to download and install first TeXLive (see link), which is a packaged distribution of LaTeX and associated packages

Then install LyX using installers, making sure that you are pointing to location of installed TeXLive when asked for a LaTeX distribution during the run of LyX installer.

Recommended versions:

- TexLive (recommended is latest version, currently TeXLive 2022)
- LyX (recommended is latest subversion, currently LyX 2.3.6.1)

There are video tutorials (link 1 and link 2).

In addition to these video tutorials, I can also have a video online session, to provide an overview and answer your questions on using LyX and JabRef. This online session (through Google Meet Google Meet) can be recorded and shared with you afterwards.

6.1.7 Source control manager

The recommended source control manager is GitHub desktop, which can be used in conjunction with thr private GitHub repository that each student team will create for their project

- latest subversion, currently GitHub Desktop 3.1.2

6.1.8 File editor

The recommended file editor is Notepad++

- latest subversion (currently Notepad++ 8.4.7) with various plugins (see list of available plugins at link 1 and link 2)

6.1.9 Runtime libraries

Many Python and R packages require runtime libraries such as Microsoft Visual C++ Redistributable

- latest version, currently Microsoft Visual C++ Redistributable 64-bit for Visual Studio 2015, 2017, 2019, and 2022

6.2 Recommended datasets

The datasets below were selected to have the following features:

- to be representative proxies for most relevant asset and subasset classes
- to be widely available

- to be as liquid as possible
- to have daily granularity
- to encompass time periods containing as many market regimes as possibles (under this consideration, the recommended daily datasets start from early 1990s)
- to have “nicer” statistical properties, which will make modeling easier (under this consideration, time series of recommended financial indices have “nicer” statistical properties compared to time series of individual stocks or bonds)

The following datasets are suggested

Table 2: Daily data sets

Name	Description	Name	Description
BCOMTR	Bloomberg Commodity Index Total Return	RU20VATR	iShares Russell 2000 Value ETF
HFRIFWI	HFRI Fund Weighted Composite Index	RUMCINTR	iShares Russell Mid-Cap ETF
LBUSTRUU	Bloomberg Barclays US Aggregate Bond Index	RUMRINTR	iShares Micro-Cap ETF
LG30TRUU	Bloomberg Barclays Global High Yield Total Return Index Value Unhedge	RUTPINTR	iShares Russell Top 200 ETF
LMBITR	Bloomberg Barclays Municipal Bond Index Total Return Index Value Unhedged USD	S5COND	S&P 500 Consumer Discretionary Index
NDDUE15X	Amundi MSCI Europe Ex UK Ucits ETF Dr	S5CONS	S&P 500 Consumer Staples Index
NDDUJN	MSCI Japan Index	S5ENRS	S&P 500 Energy Index
NDDUNA	iShares MSCI North America UCITS ETF	S5FINL	S&P 500 Financials Sector GICS Level 1 Index
NDDUPXJ	MSCI Pacific ex Japan UCITS ETF	S5HLTH	S&P 500 Health Care Index
NDDUUK	iShares MSCI UK ETF	S5INDU	S&P 500 Industrials Index
NDDUWXUS	MSCI World ex USA total net return	S5INFT	S&P 500 Information Technology Index
NDUEEGF	SPDR MSCI Emerging Markets UCITS ETF	S5MATR	S&P 500 Materials Index
RU10GRTR	iShares Russell 1000 Growth ETF	S5RLST	S&P 500 Real Estate Index
RU10VATR	iShares Russell 1000 Value ETF	S5TELS	S&P 500 Communication Services Index
RU20GRTR	iShares Russell 2000 Growth ETF	S5UTIL	S&P 500 Utilities Index
RU20INTR	Russell 2000 Total Return	SPXT	Proshares S&P 500 EX Technology ETF

Table 3: Monthly data sets

Name	Description	Name	Description
IBXXSHY1	iShares 0-5 Year High Yield Corporate Bond ETF	M2USEV	MSCI USA Enhanced Value Index
IDCT20RT	ICE U.S. Treasury 20+ Year Bond Total Return Index	M2USRWGT	MSCI USA Risk Weighted Index
LBUSTRUU	Bloomberg Barclays US Agg Total Return Value Unhedged USD	M2USSNQ	MSCI USA Sector Neutral Quality Index
LC07TRUU	Bloomberg Barclays U.S. Universal Total Return Index Value Unhedged	MID	S&P 400 Mid Cap Index index
LD01TRUU	Bloomberg Barclays 1-3 Yr Credit Total Return Index Value Unhedged US	MXEA	MSCI EAFE Index
LT01TRUU	Bloomberg Barclays US Treasury 1-3 Year Index	MXEF	MSCI Emerging Markets Index
LUICTRUU	Bloomberg Barclays U.S. Intermediate Credit Total Return Index	MXUSMVOL	MSCI USA Minimum Volatility Index
LULCTRUU	Bloomberg Barclays U.S. Long Credit Index	MXWD	MSCI All Countries World Index
M1CXBRU	iShares Core MSCI International Developed Markets ETF	MXWOUIM	MSCI All Countries World Index
M1USMVOL	MSCI USA Minimum Volatility (USD) Index	NDDUUS	MSCI Daily Total Return Net USA USD Index
M2US000$	iShares Edge MSCI USA Momentum Factor ETF	SPX	S&P 500 Index

7 Potentially useful Python and R software implementations: packages, codes and frameworks

7.1 Collections and repositories of resources

For Data Science, Numerical Methods/ Algorithms, Programming

List of links:

- Data Science CheatSheet
- professional-programming: collection of full-stack resources for programmers.

For Python

List of links:

- Awesome Python
- Awesome Python frameworks, libraries, software and resources
- Best of Python
- Curated list of Python frameworks, libraries, software and resources
- Pythonidae: Curated decibans of scientific programming resources in Python
- Ranked list of Python open-source Machine Learning libraries and tools
- Ranked list of Python open-source libraries and tools
- Ranked list of Python developer tools and libraries
- Time series: analytics, statistics, machine learning, frameworks and databases
- Time series Python packages

For R

List of links:

- Available CRAN Packages By Date of Publication
- CRAN Task Views

7.2 Connection between Python and R codes

List of links:

- arrow: R interface to 'Apache' 'Arrow', a cross-language for accelerated data interchange in-memory data
- pyarrow: Python library for Apache Arrow
- reticulate: R Interface to 'Python' modules, classes, and functions
- rpy2: Python interface to the R language
- rpy2-arrow: Share Apache Arrow datasets between Python and R
- R Extension for Visual Studio Code

7.3 Anomaly detection and data outliers

Collections of resources

List of links:

- Anomaly detection related books, papers, videos, and toolboxes

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- adtk: Python toolkit for rule-based/unsupervised anomaly detection in time series
- Anomaly Detection Learning Resources
- Awesome anomaly detection resources
- Curve: time series data anomaly detection by Baidu
- kats: kit to analyze time series data by Facebook
- luminaire: ML driven package by Zillow for monitoring time series data
- Merlion: A Machine Learning Framework for Time Series Intelligence by SalesForce
- PyGOD: Graph Outlier Detection (Anomaly Detection)
- PyOD: Python Toolbox for Scalable Outlier Detection (Anomaly Detection)
- PyODDS: An End-to-end Outlier Detection System
- ruptures: change point detection in Python
- seriesdistancematrix: implements the Series Distance Matrix framework, a flexible component-based framework that bundles various Matrix Profile related techniques
- Software tools and datasets for anomaly detection on time series data
- Tools and datasets for anomaly detection on time-series data.
- tsad: Time Series Forecasting and Anomaly Detection
- TODS: An Automated Time-series Outlier Detection System
- tsmoothie: time-series smoothing and outlier detection

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- amelie: Anomaly Detection with Normal Probability Functions
- ANN2: Artificial Neural Networks for Anomaly Detection
- anomaly: Detecting Anomalies in Data
- AnomalyDetection: package by Twitter to detect anomalies
- anomalize: Tidy Anomaly Detection
- composits: Compositional, Multivariate and Univariate Time Series Outlier Ensemble
- dobin: Dimension Reduction for Outlier Detection

- dsos: Dataset Shift with Outlier Scores
- HDoutliers: Leland Wilkinson's Algorithm for Detecting Multidimensional Outliers
- isotree: Isolation-Based Outlier Detection
- kssa: automatically identify and validate the best method for missing data imputation in a time series
- lookout: Leave One Out Kernel Density Estimates for Outlier Detection
- mvoutlier: Multivariate Outlier Detection Based on Robust Methods
- odetector: Outlier Detection Using Partitioning Clustering Algorithms
- otsad: Online Time Series Anomaly Detectors
- outForest: Multivariate Outlier Detection and Replacement
- outliers: Tests for Outliers
- outliertree: Explainable Outlier Detection Through Decision Tree Conditioning
- stray: Anomaly Detection in High Dimensional and Temporal Data
- TagAnomaly: Anomaly detection analysis and labeling tool by Microsoft
- trendsegmentR: Linear Trend Segmentation and Point Anomaly Detection
- tsoutliers: Detection of Outliers in Time Series
- univOutl: Detection of Univariate Outliers

7.4 Bayesian analysis and modeling

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- ArviZ: Exploratory analysis of Bayesian models with Python
- baal: enable Bayesian active learning in your research or labeling work
- bambi: BAyesian Model-Building Interface (Bambi)
- bilby: Bayesian inference library
- BayesianOptimization: implementation of global optimization with gaussian processes
- BayesTSA: ayesian methods for solving estimation and forecasting problems in time series analysis
- BoTorch: Bayesian optimization in PyTorch
- Bumps: data fitting and uncertainty estimation
- nutpie: A fast sampler for bayesian posteriors
- Orbit: Bayesian forecasting package by Uber
- PyApprox: high-dimensional approximation and uncertainty quantification
- pyMC: Bayesian Modeling and Probabilistic Machine Learning with Aesara
- PyStan: Python interface to Stan, a platform for statistical modeling
- zeus: Lightning Fast MCMC

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ashr: Methods for Adaptive Shrinkage, using Empirical Bayes
- bain: Bayes Factors for Informative Hypotheses (equality, inequality, and about equality constrained hypotheses)
- bamp: Bayesian Age-Period-Cohort Modeling and Prediction
- bsamGP: Bayesian Spectral Analysis Models using Gaussian Process Priors
- bayesdfa: Bayesian Dynamic Factor Analysis (DFA) with 'Stan'
- bayefdr: Bayesian Estimation and Optimisation of Expected False Discovery Rate
- BayesFM: Bayesian Inference for Factor Modeling
- bayesforecast: Bayesian Time Series Modeling with Stan
- BayesHMM: Full Bayesian Inference for Hidden Markov Models
- bayesian: Bindings for Bayesian TidyModels
- bayesmodels: The 'Tidymodels' Extension for Bayesian Models
- bayesplot: Plotting for Bayesian Models
- BayesPostEst: Generate Postestimation Quantities for Bayesian MCMC Estimation
- bayestestR: Understand and Describe Bayesian Models and Posterior Distributions
- BayesTools: Tools for Bayesian Analyses
- BayesVarSel: Bayes Factors, Model Choice and Variable Selection in Linear Models
- BEST: Bayesian Estimation Supersedes the t-Test
- beyondWhittle: Bayesian Spectral Inference for Stationary Time Series
- BFpack: Flexible Bayes Factor Testing of Scientific Expectations
- BMAmevt: Multivariate Extremes: Bayesian Estimation of the Spectral Measure
- bmixture: Bayesian Estimation for Finite Mixture of Distributions
- bnmonitor: An Implementation of Sensitivity Analysis in Bayesian Networks
- BNPmix: Bayesian Nonparametric Mixture Models
- bpcs: Bayesian Paired Comparison Analysis with Stan
- bpgmm: Bayesian Model Selection Approach for Parsimonious Gaussian Mixture Models
- brms: Bayesian Regression Models using 'Stan'
- BSL: Bayesian Synthetic Likelihood
- bspec: Bayesian Spectral Inference
- bsvars: Bayesian Estimation of Structural Vector Autoregressive Models
- dalmatian: Automating the Fitting of Double Linear Mixed Models in 'JAGS' and 'nimble'
- dbnR: Dynamic Bayesian Network Learning and Inference

- DEBBI: Differential Evolution-Based Bayesian Inference
- ensembleBMA: Probabilistic Forecasting using Ensembles and Bayesian Model Averaging
- fbst: The Full Bayesian Evidence Test, Full Bayesian Significance Test and the e-Value
- greta: scalable statistical modelling in R
- LaplacesDemon: Complete Environment for Bayesian Inference
- mBvs: Bayesian Variable Selection Methods for Multivariate Data
- mlr3mbo: Flexible Bayesian Optimization
- mombf: Bayesian Model Selection and Averaging for Non-Local and Local Priors
- networkABC: Network Reverse Engineering with Approximate Bayesian Computation
- nimble: MCMC, Particle Filtering, and Programmable Hierarchical Modeling
- Nmix: Bayesian Inference on Univariate Normal Mixtures
- posterior: Tools for Working with Posterior Distributions
- rBayesianOptimization: Bayesian Optimization of Hyperparameters
- Rbeast: Bayesian Change-Point Detection and Time Series Decomposition
- REBayes: Empirical Bayes Estimation and Inference
- Revticulate: Interaction with ”RevBayes” in R
- rstan: R Interface to Stan
- rstanarm: Bayesian Applied Regression Modeling via Stan
- SequenceSpikeSlab: Exact Bayesian Model Selection Methods for the Sparse Normal Sequence Model
- shrinkTVP: Efficient Bayesian Inference for Time-Varying Parameter Models with Shrinkage
- tidybayes: Tidy Data and ’Geoms’ for Bayesian Models

7.5 Causality, inference and dependencies

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- bilby: Bayesian inference library
- CausalDiscoveryToolbox: causal inference in graphs and in the pairwise settings
- causality: Tools for causal analysis
- causalml: package by Uber for Uplift modeling and causal inference with machine learning algorithms
- copulae: Multivariate data modelling with Copulas
- DoWhy: library by Microsoft for causal inference that supports explicit modeling and testing of causal assumptions
- HiDimStat: High-dimensional statistical inference tool
- tigramite: time series analysis python module for causal discovery

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- causal.decomp: Causal Decomposition Analysis
- CausalImpact: toolkit by Google to infer Causal Effects using Bayesian Structural Time-Series Models
- causaloptim: An Interface to Specify Causal Graphs and Compute Bounds on Causal Effects
- copula: Multivariate Dependence with Copulas
- dCovTS: Distance Covariance and Correlation for Time Series Analysis
- estimatr: Fast Estimators for Design-Based Inference
- flipr: Flexible Inference via Permutations in R
- generalCorr: Generalized Correlations, Causal Paths and Portfolio Selection
- HellCor: The Hellinger Correlation
- infer: Tidy Statistical Inference
- jackstraw: Statistical Inference for Unsupervised Learning
- konfound: Quantify the Robustness of Causal Inferences
- mashr: Multivariate Adaptive Shrinkage
- multivariance: Measuring Multivariate Dependence Using Distance Multivariance
- NlinTS: Models for Non Linear Causality Detection in Time Series
- NNS: Nonlinear nonparametric statistics using partial moments
- pcalg: Methods for Graphical Models and Causal Inference
- qmd: Quantification of Multivariate Dependence
- rmcfs: The MCFS-ID Algorithm for Feature Selection and Interdependency Discovery
- sherlock: package by Netflix for causal machine learning for segment discovery and analysis
- SIHR: Statistical Inference in High Dimensional Regression
- tlverse: One Stop to Targeted Learning in R
- tscopula: Time Series Copula Models
- VLTimeCausality: Variable-Lag Time Series Causality Inference Framework

7.6 Classification, Motifs, Neighbors, Wavelets, Transforms

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best-Subset Selection Library
- catboost: Gradient Boosting on Decision Trees by Yandex
- HiClass: hierarchical classification compatible with scikit-learn

- LightGBM: fast, distributed, high performance gradient boosting (GBT, GBDT, GBRT, GBM or MART) framework by Microsoft
- Local Cascade Ensemble (LCE) is a high-performing, scalable and user-friendly machine learning method for the general tasks of Classification and Regression
- matrixprofile: time series data mining tasks, utilizing matrix profile algorithms
- pyts: time series classification
- scikit-learn: machine learning in Python
- seriesdistancematrix: implements the Series Distance Matrix framework, a flexible component-based framework that bundles various Matrix Profile related techniques
- sktime: unified framework for machine learning with time series
- stumpy: modern time series analysis
- tslearn: machine learning toolkit dedicated to time-series data

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best-Subset Selection Library
- AUC: Threshold Independent Performance Measures for Probabilistic Classifiers
- bcTSNE: Projected t-SNE for Batch Correction
- biwavelet: Conduct Univariate and Bivariate Wavelet Analyses
- caret: Classification and Regression Training
- classmap: Visualizing Classification Results
- classifly: Explore Classification Models in High Dimensions
- ContaminatedMixt: Clustering and Classification with the Contaminated Normal
- CORElearn: Classification, Regression and Feature Evaluation
- cvms: Cross-Validation for Model Selection
- ddalpha: Depth-Based Classification and Calculation of Data Depth
- dtw: Dynamic Time Warping Algorithms
- greed: Clustering and Model Selection with the Integrated Classification Likelihood
- ipred: Improved Predictors
- klaR: Classification and Visualization
- matrixProfile: Matrix Profile
- matrixprofiler: Matrix Profile for R
- mclust: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation
- MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions
- MixMatrix: Classification with Matrix Variate Normal and t Distributions

- mixSPE: Mixtures of Power Exponential and Skew Power Exponential Distributions for Use in Model-Based Clustering and Classification
- mixture: Mixture Models for Clustering and Classification
- randomUniformForest: Random Uniform Forests for Classification, Regression and Unsupervised Learning
- rbooster: AdaBoost Framework for Any Classifier
- rebmix: Finite Mixture Modeling, Clustering & Classification
- regtools: Regression and Classification Tools
- Rmixmod: Classification with Mixture Modelling
- RSSL: Implementations of Semi-Supervised Learning Approaches for Classification
- Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation
- sbfc: Selective Bayesian Forest Classifier
- SKNN: A Super K-Nearest Neighbor (SKNN) Classification Algorithm
- stacks: Tidy Model Stacking
- TSMining: Mining Univariate and Multivariate Motifs in Time-Series Data
- tsmp: Time Series with Matrix Profile
- yardstick: Tidy Characterizations of Model Performance

7.7 Clustering

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- cclust: Convex Clustering Methods and Clustering Indexes
- ChronoClust: perform clustering on each of a time-series of discrete datasets, and explicitly track the evolution of clusters over time
- classix: Fast and explainable clustering based on sorting
- ClusterEnsembles: package for cluster ensembles
- clustergram: Visualization and diagnostics for cluster analysis in Python
- Clusteval: methods for unsupervised cluster validation
- deeptime: analysis of time series data including dimensionality reduction, clustering, and Markov model estimation
- dtaidistance: Time series distances: Dynamic Time Warping
- DTCR: Learning Representations for Time Series Clustering
- DTW_kmedoids: Multivariate time series clustering using Dynamic Time Warping (DTW) and k-mediods
- ETNA Time Series Library by Tinkoff AI
- faiss: efficient similarity search and clustering of dense vectors
- fastcluster: Fast hierarchical clustering routines

- genieclust: Fast and Robust Hierarchical Clustering with Noise Point Detection
- hcluster: Hierarchical Clustering Algorithms
- hdbscan: high performance implementation of HDBSCAN clustering
- scikit-learn: machine learning in Python
- TimeSeriesDeepClustering: End-to-end deep representation learning for time series clustering
- tslearn: machine learning toolkit dedicated to time-series data
- validclust: Validate clustering results

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- apcluster: Affinity Propagation Clustering
- bahc: bahc: Filter Covariance and Correlation Matrices with Bootstrapped-Averaged Hierarchical Ansatz
- bootcluster: Bootstrapping Estimates of Clustering Stability
- cclust: Convex Clustering Methods and Clustering Indexes
- clue: Cluster Ensembles
- clusrank: Wilcoxon Rank Tests for Clustered Data
- clustAnalytics: Cluster Evaluation on Graphs
- ClustAssess: Tools for Assessing Clustering
- ClustBlock: Hierarchical and partitioning algorithms of blocks of variables
- cluster: ”Finding Groups in Data”: Cluster Analysis Extended Rousseeuw et al.
- clusterability: Performs Tests for Cluster Tendency of a Data Set
- ClusterBootstrap: Analyze Clustered Data with Generalized Linear Models using the Cluster Bootstrap
- Clustering: Techniques for Evaluating Clustering
- clusterSEs: Calculate Cluster-Robust p-Values and Confidence Intervals
- ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering
- clusterSim: Searching for Optimal Clustering Procedure for a Data Set
- clustrd: Methods for Joint Dimension Reduction and Clustering
- clustree: Visualise Clusterings at Different Resolutions
- clValid: Validation of Clustering Results
- cmbClust: Conditional Mixture Modeling and Model-Based Clustering
- Ckmeans.1d.dp: Optimal, Fast, and Reproducible Univariate Clustering
- diceR: Diverse Cluster Ensemble in R
- dtwclust: Time Series Clustering Along with Optimizations for the Dynamic Time Warping Distance

- evclust: Evidential Clustering
- fastcluster: Fast Hierarchical Clustering Routines for R and 'Python'
- fastkmedoids: Faster K-Medoids Clustering Algorithms: FastPAM, FastCLARA, FastCLARANS
- FCPS: Fundamental Clustering Problems Suite
- flexclust: Flexible Cluster Algorithms
- fpc: Flexible Procedures for Clustering
- genie: Fast, Robust, and Outlier Resistant Hierarchical Clustering
- genieclust: The Genie++ Hierarchical Clustering Algorithm with Noise Points Detection
- heatmaply: Interactive Cluster Heat Maps Using 'plotly' and 'ggplot2'
- HierPortfolios: Hierarchical Clustering-Based Portfolio Allocation Strategies
- htestClust: Reweighted Marginal Hypothesis Tests for Clustered Data
- kselection: Selection of K in K-Means Clustering
- l1spectral: An L1-Version of the Spectral Clustering
- leaderCluster: Leader Clustering Algorithm
- LearnClust: Learning Hierarchical Clustering Algorithms
- MatTransMix: Clustering with Matrix Gaussian and Matrix Transformation Mixture Models
- mclust: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation
- mclustcomp: Measures for Comparing Clusters
- mdendro: Extended Agglomerative Hierarchical Clustering
- Mercator: Clustering and Visualizing Distance Matrices
- MixGHD: Model Based Clustering, Classification and Discriminant Analysis Using the Mixture of Generalized Hyperbolic Distributions
- MixSim: Simulating Data to Study Performance of Clustering Algorithms
- mixSPE: Mixtures of Power Exponential and Skew Power Exponential Distributions for Use in Model-Based Clustering and Classification
- mixture: Mixture Models for Clustering and Classification
- MKMeans: A Modern K-Means (MKMeans) Clustering Algorithm
- mlr3cluster: Cluster Extension for 'mlr3'
- motifcluster: Motif-Based Spectral Clustering of Weighted Directed Networks
- MSclust: Multiple-Scaled Clustering
- mstknnclust: MST-kNN Clustering Algorithm
- NNS: Nonlinear nonparametric statistics using partial moments
- ProjectionBasedClustering: Projection Based Clustering
- protoclust: Hierarchical Clustering with Prototypes

- pytorch_cluster: PyTorch Extension Library of Optimized Graph Cluster Algorithms
- QuClu: Quantile-Based Clustering Algorithms
- rebmix: Finite Mixture Modeling, Clustering & Classification
- RCTS: Clustering Time Series While Resisting Outliers
- RMBC: Robust Model Based Clustering
- sClust: R Toolbox for Unsupervised Spectral Clustering
- sigclust: Statistical Significance of Clustering
- SLBDD: Statistical Learning for Big Dependent Data
- Spectrum: Fast Adaptive Spectral Clustering for Single and Multi-View Data
- T4cluster: Tools for Cluster Analysis
- tclust: Robust Trimmed Clustering
- tglkmeans: Efficient Implementation of K-Means++ Algorithm
- TSclust: Time Series Clustering Utilities
- vimpclust: Variable Importance in Clustering

7.8 Coding utilities and frameworks

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Algviz is an algorithm visualization tool for your Python code
- asteval: minimalistic evaluator of python expression using ast module
- autoflake: Removes unused imports and unused variables as reported by pyflakes
- autopep8: automatically formats Python code to conform to the PEP 8 style guide
- autoray: Write numeric code that automatically works with any numpy-ish libraries
- bandit: find common security issues in Python code
- birdseye: Graphical debugger to view the values of all evaluated expressions
- black: uncompromising Python code formatter
- BLUE: The slightly less uncompromising Python code formatter
- Bowler: Safe code refactoring by Facebook for modern Python
- Comprehensive Python Cheatsheet
- conda-deps: Generate conda environment files from Python and R source code
- Crashtest is a Python library that makes exceptions handling and inspection easier.
- darker: Apply black reformatting to Python files only in regions changed since a given commit
- enum_tools: Tools to expand Python's enum module.
- erdantic: tool for drawing entity relationship diagrams (ERDs) for Python data model classes.

- flake8: glues together pycodestyle, pyflakes, mccabe, and third-party plugins to check the style and quality of code
- flake8-black: flake8 plugin to run black for checking Python coding style
- friendly: replaces standard tracebacks by something easier to understand
- Hatch is a modern, extensible Python project manager.
- icecream: Never use print() to debug again
- ipdb: exports functions to access the IPython debugger
- isort: utility / library to sort imports
- jedi: autocompletion, static analysis and refactoring library
- jsonschema: implementation of the JSON Schema specification for Python
- kedro: framework for creating reproducible, maintainable and modular data science code
- kedro-viz: Visualise your Kedro data and machine-learning pipelines and track your experiments.
- libfyaml: Fully feature complete YAML parser and emitter
- luddite: Checks for out-of-date package versions
- makepackage: easy packaging of Python code
- mamba: Fast Cross-Platform Package Manager (reimplementation of the conda package manager in C++)
- memray: memory profiler for Python
- metaflow: build and manage real-life data science projects
- mkdocs: Project documentation with Markdown
- mkdocs-material: Technical documentation that just works
- MonkeyType: toolkit by Instagram to generate static type annotations by collecting runtime types
- Monty: supplementary useful functions for Python that are not part of the standard library
- mypy: Optional static typing for Python
- nptyping: Type hints for Numpy
- numpydoc: Numpy's Sphinx extensions
- pdbpp: a drop-in replacement for pdb (the Python debugger)
- PlantUML: Generate UML diagram from textual description
- poetry: dependency management and packaging made easy
- Pretty_Errors: Prettify Python exception output to make it legible
- prospector: Inspects source files and provides information about type and location of classes, methods
- ptvsd: debugger package by Microsoft for use with Visual Studio and Visual Studio Code
- pudb: Full-screen console debugger for Python
- pyan: Static call graph generator
- pycodestyle: Simple Python style checker

- pydantic: Data parsing and validation using Python type hints
- pyDeprecate: tooling for marking deprecated functions or classes and re-routing to the new successors' instance.
- pyflakes: checks Python source files for errors
- pylint: static code analysis tool
- pyquickhelper: automation of many things
- pyre: framework for building scientific applications in Python
- pyre-check: Performant type-checking toolkit by Facebook
- pyright: Static type checker by Microsoft
- PyScaffold: Python project template generator with batteries included
- PySnooper: Never use print for debugging again
- py-spy: Sampling profiler for Python programs
- pytools: a big bag of things that are ”missing” from the Python standard library
- pytype: static type analyzer by Google
- radon: tool that computes various metrics from the source code
- rope: refactoring library
- scalene: high-performance, high-precision CPU, GPU, and memory profiler for Python
- sphinx: Sphinx documentation builder
- StrictYAML is a type-safe YAML parser that parses and validates a restricted subset of the YAML specification
- tryceratops: linter to prevent exception handling antipatterns in Python
- typeguard: Run-time type checker for Python
- TypePigeon: type converter focused on converting values between various Python data types.
- varname:Dark magics about variable names in python
- vulture: Find dead Python code
- xlwings: ibrary that makes it easy to call Python from Excel and vice versa
- yapf: formatter by Google for Python files
- yappi: Yet Another Python Profiler, but this time multithreading, asyncio and gevent aware.

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- adaptalint: Check Code Style Painlessly
- baguette: Efficient Model Functions for Bagging
- box: Write Reusable, Composable and Modular R Code
- butcher: Model Butcher: axe components of fitted model objects and help reduce the size of model objects saved to disk
- cartbonate: Create beautiful images of source code using 'carbon.js
- checkmate: Fast and Versatile Argument Checks
- checkpoint: Install Packages from Snapshots on the Checkpoint Server for Reproducibility
- cleanr: Helps You to Code Cleaner
- delayed: A Framework for Parallelizing Dependent Tasks
- goodpractice: Advice on R Package Building
- hardhat: Construct Modeling Packages
- IRdisplay: 'Jupyter' Display Machinery
- IRkernel: Native R Kernel for the 'Jupyter Notebook'
- jetpack: A Friendly Package Manager
- leprechaun: Create Simple 'Shiny' Applications as Packages
- lintr: A 'Linter' for R Code
- lvec: Out of Memory Vectors
- memuse: Memory Estimation Utilities
- metaflow: build and manage real-life data science projects
- miniCRAN: Create a Mini Version of CRAN Containing Only Selected Packages
- mongolite: Fast and Simple 'MongoDB' Client for R
- packager: Create, Build and Maintain Packages
- parsnip: A Common API to Modeling and Analysis Functions
- prettifyAddins: 'RStudio' Addins to Prettify 'JavaScript', 'C++', 'Python', and More
- R6: Encapsulated Classes with Reference Semantics
- R6P: Design Patterns in R
- recipes: Preprocessing and Feature Engineering Steps for Modeling
- renv: Project Environments
- rhino: A Framework for Enterprise Shiny Applications
- roxut: Document Unit Tests Roxygen-Style
- roxygen2: In-Line Documentation for R

- rstudio.prefs: Set 'RStudio' Preferences
- tidymodules: obust framework for developing 'Shiny' modules based on R6 classes which should facilitates inter-modules communication.
- waldo: Find Differences Between R Objects
- vetiver: Version, Share, Deploy, and Monitor Models
- workflows: Modeling Workflows
- workflowsets: Create a Collection of 'tidymodels' Workflows

7.9 Computational performance

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Aesara: definie, optimize, and efficiently evaluate mathematical expressions involving multi-dimensional arrays.
- arctic: High performance datastore by Man Group for time series and tick data
- bottleneck: Fast NumPy array functions written in C
- Dask: Parallel computing with task scheduling
- Dask-ML provides scalable machine learning in Python using Dask alongside popular machine learning libraries
- datatable: library for fast multi-threaded data manipulation and munging
- fairscale: PyTorch extensions for high performance and large scale training.
- fastcore: Python supercharged for the fastai library
- hypre: high performance preconditioners
- jax: automatically differentiate native Python and NumPy functions
- modin: ake your pandas code run faster by changing one line of code
- multiprocess: better multiprocessing and multithreading in python
- numexpr: Fast numerical expression evaluator for NumPy
- PandaPy: speed of NumPy and the usability of Pandas but much faster
- pandarallel: parallelize Pandas operations on all available CPUs
- pandasvault:Advanced Pandas Vault - Utilities, Functions and Snippets
- polars: Fast multi-threaded DataFrame library
- ppft: distributed and parallel python
- PyArma: Linear algebra library for Python
- PyArmadillo: an alternative approach to linear algebra in Python
- pyperf: Toolkit to run Python benchmarks
- pyperformance: Python Performance Benchmark Suite
- py-spy: Sampling profiler for Python programs

- scalene: high-performance, high-precision CPU, GPU, and memory profiler for Python
- swifter: efficiently applies any function to a pandas dataframe or series in the fastest available manner
- tempeh is a framework to TEst Machine learning PErformance exHaustively which includes tracking memory usage and run time.

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- collapse: Advanced and Fast Data Transformation
- dataPreparation: Automated Data Preparation
- delayed: A Framework for Parallelizing Dependent Tasks
- dplyr: A Grammar of Data Manipulation
- MatrixStats: Methods that Apply to Rows and Columns of Matrices (and to Vectors)
- mirai: Minimalist Async Evaluation Framework for R
- purrr: Functional Programming Tools
- tidyverse: set of packages that work in harmony because they share common data representations and 'API' design
- timetk: A Tool Kit for Working with Time Series in R
- tibble: Simple Data Frames
- tidytidbits: A Collection of Tools and Helpers Extending the Tidyverse
- tsibble: Tidy Temporal Data Frames and Tools

7.10 Containers, projects, pipelines and deployment

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Driblet - Google Cloud based ML pipeline by Google
- MLflow: A Machine Learning Lifecycle Platform
- metaflow: Python/R library by Netflix to build and manage real-life data science projects
- mlflow: Interface to 'MLflow'
- mlxtend: extension and helper modules for data analysis and machine learning libraries
- NNI: toolkit by Microsoft to help users automate Feature Engineering, Neural Architecture Search, Hyperparameter Tuning and Model Compression
- petastorm: toolkit by Uber for single machine or distributed training and evaluation of deep learning models (Tensorflow, Pytorch, and PySpark) from datasets in Apache Parquet format
- pipelines: Machine Learning Pipelines for Kubeflow
- Prefect: second-generation dataflow coordination and orchestration platform
- PyTorch Lightning: The lightweight PyTorch wrapper for high performance AI research
- Tango: toolkit by Allen Institute of Articial Intelligence for choreographing machine learning research

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- DriveML: Self-Drive Machine Learning Projects
- metaflow: Python/R library by Netflix to build and manage real-life data science projects
- mlflow: Interface to 'MLflow'

7.11 Covariances, correlations and volatilities

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- numpy: scientific computing
- precise: online covariance and precision forecasting, portfolios, and model ensembles
- PyPortfolioOpt: Financial portfolio optimization
- sklearn.covariance: covariance estimation in scikit-learn
- statsmodels: statistical modeling and econometrics

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- bahc: Filter Covariance and Correlation Matrices with Bootstrapped-Averaged Hierarchical Ansatz
- BBcor: Bayesian Bootstrapping Correlations
- BEKKs: Multivariate Conditional Volatility Modelling and Forecasting
- BSCOV: Detection of Multiple Structural Breaks in Large Covariance Matrices
- clubSandwich: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections
- cocor: Comparing Correlations
- corpcor: Efficient Estimation of Covariance and (Partial) Correlation
- correlation: Methods for Correlation Analysis
- corx: Create and Format Correlation Matrices
- CovTools: Statistical Tools for Covariance Analysis
- cvCovEst: Cross-Validated Covariance Matrix Estimation
- dcortools: Providing Fast and Flexible Functions for Distance Correlation Analysis
- dCovTS: Distance Covariance and Correlation for Time Series Analysis
- fitHeavyTail: Mean and Covariance Matrix Estimation under Heavy Tails
- FRCC: Fast Regularized Canonical Correlation Analysis
- gencor: Generate Customized Correlation Matrices
- generalCorr: Generalized Correlations, Causal Paths and Portfolio Selection
- mashr: Multivariate Adaptive Shrinkage

- MatrixCorrelation: Matrix Correlation Coefficients
- MTS: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models
- NonParRolCor: a Non-Parametric Statistical Significance Test for Rolling Window Correlation
- NNS: Nonlinear nonparametric statistics using partial moments
- rags2ridges: Ridge Estimation of Precision Matrices from High-Dimensional Data
- rmcorr: Repeated Measures Correlation
- robcor: Robust Correlations
- robustcov: Collection of Robust Covariance and (Sparse) Precision Matrix Estimators
- RSC: Robust and Sparse Correlation Matrix
- sandwich: Robust Covariance Matrix Estimators
- WGCNA: Weighted Correlation Network Analysis

7.12 Data analysis and exploration

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- AutoViz: Automatically Visualize any dataset, any size with a single line of code.
- daal4py: simplified API to Intel oneAPI Data Analytics Library
- DeepGraph: scalable, general-purpose data analysis with Pandas-based Networks
- D-tale:Visualizer by Man Group for pandas data structures
- dython: Data analysis tools
- empiricaldist: empirical distribution functions
- hyperspy: Multidimensional data analysis
- Lux: automate the visualization and data analysis process
- mlxtend: extension and helper modules for Python's data analysis and machine learning libraries.
- numericalunits: Units and dimensional analysis compatible with everything
- Orange: Interactive data analysis
- pandas-profiling: Create HTML profiling reports from pandas DataFrame objects
- PyApprox: high-dimensional approximation and uncertainty quantification
- sweetviz: Visualize and compare datasets, target values and associations

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- checkmate: Fast and Versatile Argument Checks
- collapse: Advanced and Fast Data Transformation
- datacleanr: Interactive and Reproducible Data Cleaning
- DataEditR: An Interactive Editor for Viewing, Entering, Filtering & Editing Data
- DataExplorer: Automate Data Exploration and Treatment
- datamods: Modules to Import and Manipulate Data in 'Shiny'
- dataprep: Efficient and Flexible Data Preprocessing Tools
- DataVisualizations: Visualizations of High-Dimensional Data
- datawizard: Easy Data Wrangling
- DescTools: Tools for Descriptive Statistics
- dimensio: Multivariate Data Analysis
- discoveR: Exploratory Data Analysis System
- dlookr: Tools for Data Diagnosis, Exploration, Transformation
- EasyDescribe: A Convenient Way of Descriptive Statistics
- esquisse: Explore and Visualize Your Data Interactively
- explor: Interactive Interfaces for Results Exploration
- exploratory: A Tool for Large-Scale Exploratory Analyses
- explore: Simplifies Exploratory Data Analysis
- factoextra: extract and visualize the output of multivariate data analyses, including 'PCA' (Principal Component Analysis), 'CA' (Correspondence Analysis), 'MCA' (Multiple Correspondence Analysis), 'FAMD' (Factor Analysis of Mixed Data), 'MFA' (Multiple Factor Analysis) and 'HMFA' (Hierarchical Multiple Factor Analysis)
- FactoInvestigate: Automatic Description of Factorial Analysis
- FactoMineR: Multivariate Exploratory Data Analysis and Data Mining
- ggESDA: Exploratory Symbolic Data Analysis with 'ggplot2'
- HDTSA: High Dimensional Time Series Analysis Tools
- infotheo: Information-Theoretic Measures
- kfa: K-Fold Cross Validation for Factor Analysis
- MazamaRollUtils: Efficient Rolling Functions
- mmpca: Integrative Analysis of Several Related Data Matrices
- praznik: Tools for Information-Based Feature Selection and Scoring
- predictoR: Predictive Data Analysis System

- rigr: Regression, Inference, and General Data Analysis Tools in R
- robCompositions: Compositional Data Analysis
- rrcov: Scalable Robust Estimators with High Breakdown Point
- SmartEDA: Summarize and Explore the Data
- statsExpressions: Tidy Dataframes and Expressions with Statistical Details
- Statsomat: Shiny Apps for Automated Data Analysis and Automated Interpretation
- thinkr: Tools for Cleaning Up Messy Files
- tswge: Time Series for Data Science.Accompanies the texts Time Series for Data Science and Applied Time Series Analysis with R,
- validata: Validate Data Frames
- validate: Data Validation Infrastructure
- validatetools: Checking and Simplifying Validation Rule Sets
- wrangle: A Systematic Data Wrangling Idiom

7.13 Data augmentation, scenario generation and synthetic time series

Collections of resources

List of links:

- Synthetic data generation by Van Der Schaar Lab
- Useful data augmentation resources

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- agots: Anomaly Generator on Time Series
- benchmark_VAE: Unifying Generative Autoencoder implementations in Python
- Copulas: model multivariate data using copulas
- CTGAN: Conditional GAN for Tabular Data
- COMET Flows: Towards Generative Modeling of Multivariate Extremes and Tail Dependence
- DataGeneration: Synthetic financial correlation matrix and time series generation
- DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks
- DeepEcho: Synthetic Data Generation for mixed-type, multivariate time series
- deltapy: Tabular Data Augmentation
- extremeIndex: Forecast Verification for Extreme Events
- ixmp: platform for integrated and cross-cutting scenario analysis
- MLlforHealthLab: Machine Learning and Artificial Intelligence for Medicine
- pydantic-factories: Pydantic based mock data generation

- pythae: Unifying Generative Autoencoder implementations in Python
- RDT: Reversible Data Transforms to reproduce realistic data
- scattering_covariance: analysis and generation of time series
- SDMetrics: Metrics for Synthetic Data Generation Projects
- SDGym: Benchmarking synthetic data generation methods
- SDV: Synthetic Data Generation for tabular, relational and time series data
- SignalFilters: Signal Filtering and Generation of Synthetic Time-Series
- snorkel: system for quickly generating training data with weak supervision
- synthia: Multidimensional synthetic data generation in Python
- TGAN: Generative adversarial training for generating synthetic tabular data
- TimeGAN: Time-series Generative Adversarial Networks
- time-series-generator: Time Series Generator
- TimeSynth: Synthetic Time Series Generation
- tsaug: time series augmentation
- tsBNgen: Generate Time Series Data Based on an Arbitrary Bayesian Network Structure
- tsGAN: Time-series Generative Adversarial Networks
- ydata-synthetic: Synthetic structured data generators

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- anySim: Simulation of Non-Gaussian Correlated Random Variables, Stochastic Processes and Random Fields
- bootComb: Combine Parameter Estimates via Parametric Bootstrap
- conjurer: A Parametric Method for Generating Synthetic Data
- covsim: VITA, IG and PLSIM Simulation for Given Covariance and Marginals
- fabricatr: Imagine Your Data Before You Collect It
- fwb: Fractional Weighted Bootstrap
- gencor: Generate Customized Correlation Matrices
- gratis: Generating Time Series with Diverse and Controllable Characteristics
- meboot: Maximum Entropy Bootstrap for Time Series
- metamer: Create Data with Identical Statistics
- missMethods: Methods for Missing Data
- MixSim: Simulating Data to Study Performance of Clustering Algorithms
- modeltime.resample: Resampling Tools for Time Series Forecasting
- MonteCarlo: Automatic Parallelized Monte Carlo Simulations

- MSCMT: Multivariate Synthetic Control Method Using Time Series
- mvlognCorrEst: Sampling from Multivariate Lognormal Distributions and Estimating Correlations from Uncomplete Correlation Matrix
- naive: Empirical Extrapolation of Time Feature Patterns
- RMT4DS: Computation of Random Matrix Models
- rsample: General Resampling Infrastructure
- segen: Sequence Generalization Through Similarity Network
- SimJoint: Simulate Joint Distribution
- simmer: Discrete-Event Simulation for R
- simts: Time Series Analysis Tools
- spooky: Time Feature Extrapolation Using Spectral Analysis and Jack-Knife Resampling
- Synth: Synthetic Control Group Method for Comparative Case Studies
- synthesis: Generate Synthetic Data from Statistical Models
- tetragon: Automatic Sequence Prediction by Expansion of the Distance Matrix
- TidyDensity: Functions for Tidy Analysis and Generation of Random Data
- tscopula: Time Series Copula Models

7.14 Data cleaning, preparation and validation

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- cerberus: Lightweight, extensible data validation library
- datatest: Tools for test driven data-wrangling and data validation
- doubtlab: Doubt your data, find bad labels
- framework: Data management framework for Python that provides functionality to describe, extract, validate, and transform tabular data
- formencode: validation and form generation
- pandera: perform data validation on dataframes
- pydantic: Data parsing and validation using Python type hints
- pyjanitor: Clean APIs for data cleaning. Python implementation of R package Janitor
- PyOptimus: framework for cleaning and pre-processing data in a distributed fashion
- scikit-learn: machine learning in Python
- schema: library for validating Python data structures
- serde: framework for defining, serializing, deserializing, and validating data structures
- typical: Fast, simple, & correct data-validation using Python 3 typing.
- validators: Python data validation for Humans

- Voluptuous: data validation library.
- validr: simple, fast, extensible python library for data validation
- wtforms: flexible forms validation and rendering library

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- cleanTS: Testbench for Univariate Time Series Cleaning
- dataPreparation: Automated Data Preparation
- data.validator: Automatic Data Validation and Reporting
- datawizard: Easy Data Wrangling
- errorlocate: Locate Errors with Validation Rules
- pointblank: Data Validation and Organization of Metadata for Local and Remote Tables
- testdat: Data Unit Testing for R
- tsrobprep: Robust Preprocessing of Time Series Data
- validate: Data Validation Infrastructure
- validatetools: Checking and Simplifying Validation Rule Sets
- wrangle: A Systematic Data Wrangling Idiom

7.15 Data Imputation

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- AutoImpute: Imputation Methods
- Clairvoyance: a Unified, End-to-End AutoML Pipeline for Medical Time Series
- fancyimpute: Multivariate imputation and matrix completion algorithms
- HyperImpute: framework for prototyping and benchmarking imputation methods
- imputena: automated and customized treatment of missing values in datasets
- miceforest: Fast, Memory Efficient Imputation with LightGBM
- MissForestExtra: nonparametric imputation on missing values
- scikit-learn: machine learning in Python
- statsmodels: statistical modeling and econometrics
- tsai: time series tasks like classification, regression, forecasting, imputation

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- Amelia: A Program for Missing Data
- CoImp: Copula Based Imputation Method
- deductive: Data Correction and Imputation Using Deductive Methods
- ggmice: Visualizations for 'mice' with 'ggplot2'
- howManyImputations: Calculate How many Imputations are Needed for Multiple Imputation
- imputeFin: Imputation of Financial Time Series with Missing Values and/or Outliers
- imputeGeneric: Ease the Implementation of Imputation Methods
- imputeTestbench: Test Bench for the Comparison of Imputation Methods
- imputeTS: Time Series Missing Value Imputation
- Iscores: Proper Scoring Rules for Missing Value Imputation
- mdgc: Missing Data Imputation Using Gaussian Copulas
- mice: Multivariate Imputation by Chained Equations
- miceadds: Some Additional Multiple Imputation Functions, Especially for 'mice'
- miceafter: Data and Statistical Analyses after Multiple Imputation
- miceFast: Fast Imputations Using 'Rcpp' and 'Armadillo'
- micemd: Multiple Imputation by Chained Equations with Multilevel Data
- misPRIME: Partial Replacement Imputation Estimation for Missing Covariates
- missMDA: Handling Missing Values with Multivariate Data Analysis
- missMethods: Methods for Missing Data
- missRanger: Fast Imputation of Missing Values
- mlim: Multiple Imputation with Automated Machine Learning
- NADIA: NA Data Imputation Algorithms
- naniar: Data Structures, Summaries, and Visualisations for Missing Data
- rego: Automatic Time Series Forecasting and Missing Value Imputation
- Rforestry: Random Forests, Linear Trees, and Gradient Boosting for Inference and Interpretability
- simputation: Simple Imputation
- SLBDD: Statistical Learning for Big Dependent Data
- smcfcs: Multiple Imputation of Covariates by Substantive Model Compatible Fully Conditional Specification
- univOutl: Detection of Univariate Outliers
- VIM: Visualization and Imputation of Missing Values
- yaImpute: Nearest Neighbor Observation Imputation and Evaluation Tools

7.16 Data regimes, states and changepoints: analysis and modeling

Collections of resources

List of links:

- Classifying market regimes
- TCPD: toolkit by UK national institute for data science and artificial intelligence for Turing Change Point Dataset - A collection of time series for the evaluation and development of change point detection algorithms
- TCPDBench: toolkit by UK national institute for data science and artificial intelligence for Turing Change Point Detection Benchmark: An Extensive Benchmark Evaluation of Change Point Detection Algorithms on real-world data

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- changeforest: Random Forests for Change Point Detection
- deeptime: analysis of time series data including dimensionality reduction, clustering, and Markov model estimation
- greykite: flexible, intuitive and fast forecasting library
- HMMLearn: Hidden Markov Models in Python with scikit-learn like API
- kalmanfilter: Kalman Filter
- kats: tookit by Facebook for time series analysis and forecasting
- kimfilter: Rcpp' implementation of the multivariate Kim filter, which combines the Kalman and Hamilton filters for state probability inference
- Merlion: A Machine Learning Framework for Time Series Intelligence by SalesForce
- msmtools: estimation and analysis of discrete state space Markov chains via Markov state models (MSM)
- PyEMMA: Emma's Markov Model Algorithms
- pyGPCCA: Generalized Perron Cluster Cluster Analysis to coarse-grain reversible and non-reversible Markov state models.
- pyhsmm: Bayesian inference in HSMMs and HMMs
- pymc3-hmm: Hidden Markov models in PyMC3
- ruptures: change point detection
- SST: fast implementation of Singular Spectrum Transformation
- Stone-Soup: framework for the development and testing of tracking algorithms
- statsmodels: Markov switching models in statsmodels
- transitionMatrix: Statistical analysis and visualization of state transition phenomena
- tsmoothie: time-series smoothing and outlier detection

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- BayesHMM: Full Bayesian Inference for Hidden Markov Models
- breakfast: Methods for Fast Multiple Change-Point Detection and Estimation
- BSCOV: Detection of Multiple Structural Breaks in Large Covariance Matrices
- ChangepointInference: Tools to test for a change in mean after changepoint detection
- changepoints: A Collection of Change-Point Detection Methods
- ChangePointTaylor: Identify Changes in Mean
- chngpt: Estimation and Hypothesis Testing for Threshold Regression
- cpss: Change-Point Detection by Sample-Splitting Methods
- crossvalidationCP: Cross-Validation for Change-Point Regression
- depmixS4: Dependent Mixture Models - Hidden Markov Models of GLMs and Other Distributions in S4
- dynr: Dynamic Models with Regime-Switching
- earlywarnings: Early Warning Signals Toolbox for Detecting Critical Transitions in Timeseries
- fabisearch: Change Point Detection in High-Dimensional Time Series Networks
- fHMM: Fitting Hidden Markov Models to Financial Data
- inflection: Finds the Inflection Point of a Curve
- InspectChangepoint: High-Dimensional Changepoint Estimation via Sparse Projection
- jcp: Joint Change Point Detection
- HMM: Hidden Markov Models
- hmm.discnp: Hidden Markov Models with Discrete Non-Parametric Observation Distributions
- hmmr: ”Mixture and Hidden Markov Models with R” Datasets and Example Code
- KFAS: Kalman Filter and Smoother for Exponential Family State Space Models
- ldhmm: Hidden Markov Model for Financial Time-Series Based on Lambda Distribution
- mHMMbayes: Multilevel Hidden Markov Models Using Bayesian Estimation
- MSGARCH: Markov-Switching GARCH Models
- MSTest: Hypothesis Testing for Markov Switching Models
- NHMSAR: Non-Homogeneous Markov Switching Autoregressive Models
- onlineBcp: Online Bayesian Methods for Change Point Analysis
- plotHMM: Plot Hidden Markov Models
- pomp: Statistical Inference for Partially Observed Markov Processes
- Rbeast: Bayesian Change-Point Detection and Time Series Decomposition
- RChest: Locating Distributional Changes in Highly Dependent Time Series

- robcp: Robust Change-Point Tests
- segmented: Regression Models with Break-Points / Change-Points Estimation
- seqHMM: Mixture Hidden Markov Models for Social Sequence Data and Other Multivariate, Multichannel Categorical Time Series
- trendchange: Innovative Trend Analysis and Time-Series Change Point Analysis
- tsDyn: Nonlinear Time Series Models with Regime Switching
- wbsts: Multiple Change-Point Detection for Nonstationary Time Series

7.17 Data structures, storage and serialization

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- addict: Python Dict
- anndata: package for handling annotated data matrices in memory and on disk
- arctic: High performance datastore by Man Group for time series and tick data
- cloudpickle: serialize Python constructs not supported by the default pickle module
- dataclassy is a reimplementation of data classes in Python
- datatable: fast multi-threaded data manipulation and munging
- dill: extends Python's pickle module for serializing and deserializing python objects to the majority of the built-in python types.
- extendedjson: Easily extend JSON to encode and decode arbitrary Python objects
- framework: Data management framework for Python that provides functionality to describe, extract, validate, and transform tabular data
- MarketStore: DataFrame Server for Financial Timeseries Data
- marshmallow: lightweight library for converting complex objects to and from simple Python datatypes
- modin.pandas DataFrame is a parallel and distributed drop-in replacement for panda
- Mongita is to MongoDB as SQLite is to SQL
- mongo-arrow: Tools for using Apache Arrow with MongoDB
- multidict: multidict implementation
- Odo provides a uniform API for moving data between different formats
- pandas: data structures for data analysis, time series, and statistics
- pandasvault: Advanced Pandas Vault - Utilities, Functions and Snippets
- pickle: Python object serialization
- polars: Fast multi-threaded DataFrame library
- pyarrow: Python API for Apache Arrow, a language independent columnar memory format for flat and hierarchical data

- PyMongo - the Python driver for MongoDB
- PyStore: Fast data store for Pandas time-series data
- PyTables: package for managing hierarchical datasets
- rpy2-arrow: Share Apache Arrow datasets between Python and R
- serde: framework for defining, serializing, deserializing, and validating data structures
- sklearn-pandas: bridge between Scikit-Learn's machine learning methods and pandas-style Data Frames
- sortedcontainers: Sorted Containers – Sorted List, Sorted Dict, Sorted Set
- sqlite: Persistent dict, backed by sqlite3 and pickle, multithread-safe.
- sparse: Sparse Multidimensional Arrays
- srsly: Modern high performance serialization utilities
- tablib: Module for Tabular Datasets in XLS, CSV, JSON, YAML,
- tabmat: Efficient matrix representations for working with tabular data
- TileDB: powerful engine for storing and accessing dense and sparse multi-dimensional arrays
- tidypandas: grammar of data manipulation for pandas inspired by tidyverse
- tinyarray: Tinyarrays are similar to NumPy arrays, but optimized to be much faster for small sizes
- TinyDB is a lightweight document oriented database optimized for your happiness
- tinyflux: iny time series database optimized for your happiness
- torcharrow: torch.Tensor-like DataFrame library by Facebook using Arrow as a common memory format
- ubermagtable: package for manipulating tabular data
- ultrajson: Ultra fast JSON decoder and encoder written in C with Python bindings
- Vector: arrays of 2D, 3D, and Lorentz vectors
- Woodwork is a Python library that provides robust methods for managing and communicating data typing information
- xarray: multidimensional labeled arrays and datasets
- xpandas: Universal 1d/2d data containers with Transformers functionality for data analysis

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- arrow: Integration to Apache Arrow
- dibble: Dimensional Data Frames
- fst: Lightning Fast Serialization of Data Frames
- gluedown: Wrap Vectors in Markdown Formatting
- listdown: Create R Markdown from Lists
- motifcluster: Motif-Based Spectral Clustering of Weighted Directed Networks

- qs: Quick Serialization of R Objects
- RcppSimdJson: 'Rcpp' Bindings for the 'simdjson' Header-Only Library for 'JSON' Parsing
- tibble: stricter checking and better formatting than the traditional data frame
- tibblify: Rectangle Nested Lists
- tidytable: Tidy Interface to 'data.table'
- tiledb: Universal Storage Engine for Sparse and Dense Multidimensional Arrays
- tsibble: Tidy Temporal Data Frames and Tools
- tsbox: Class-Agnostic Time Series
- vtreat: A Statistically Sound data.frame Processor/Conditioner

7.18 Dates and times

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- arrow: Better dates and times for Python
- dateparser: parser for human readable dates
- dateutil: Useful extensions to the standard Python datetime features
- orjson: Fast, correct Python JSON library supporting dataclasses, datetimes, and numpy
- parsedatetime: human-readable date/time strings
- pendulum: datatimes made easy
- Pyrsistent: Persistent/Functional/Immutable data structures
- python-dateutil: Useful extensions to the standard Python datetime features
- PyTime: operate datetime by string

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- clock: Date-Time Types and Tools
- lubridate: Make Dealing with Dates a Little Easier
- qlcal: R Bindings to the Calendaring Functionality of 'QuantLib'
- tidyquant: Tidy Quantitative Financial Analysis
- timechange: Efficient Manipulation of Date-Times
- timetk: A Tool Kit for Working with Time Series in R
- tsbox: Class-Agnostic Time Series
- TSrepr: Time Series Representations
- xts: eXtensible Time Series
- zoo: S3 Infrastructure for Regular and Irregular Time Series

7.19 Dimensionality reduction

Python

List of packages/codes/frameworks/links:

- abess: Fast Best-Subset Selection Library
- deeptime: analysis of time series data including dimensionality reduction, clustering, and Markov model estimation
- direpack: State-of-the-Art Statistical Dimension Reduction Methods
- EZyRB: Easy Reduced Basis method ; performs a data-driven model order reduction for parametrized problems exploiting the recent approaches.
- humap: Hierarchical Manifold Approximation and Projection (HUMAP) is a technique based on UMAP for hierarchical non-linear dimensionality reduction.
- pyFIt-SNE: FFT-accelerated Interpolation-based t-SNE (FIt-SNE)
- scikit-dimension: intrinsic dimension estimation
- scikit-learn: machine learning in Python
- (t-SNE: t-Distributed Stochastic Neighbor Embedding (t-SNE) for dimensionality reduction
- UMAP: Uniform Manifold Approximation and Projection

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best-Subset Selection Library
- abundant: High-Dimensional Principal Fitted Components and Abundant Regression
- bayesdfa: Bayesian Dynamic Factor Analysis (DFA) with 'Stan'
- clustrd: Methods for Joint Dimension Reduction and Clustering
- dimRed: A Framework for Dimensionality Reduction
- DLPCA: The Distributed Local PCA Algorithm
- dobin: Dimension Reduction for Outlier Detection
- dyndimred: Dimensionality Reduction Methods in a Common Format
- EMD: Empirical Mode Decomposition and Hilbert Spectral Analysis
- ForeCA: Forecastable Component Analysis
- freqdom: Frequency Domain Based Analysis: Dynamic PCA
- ica: Independent Component Analysis
- ICtest: Estimating and Testing the Number of Interesting Components in Linear Dimension Reduction
- prinvars: Principal Variables (methods for reducing the number of features within a data set)
- quantdr: Dimension Reduction Techniques for Conditional Quantiles
- rrpack: Reduced-Rank Regression

- Rdimtools: Dimension Reduction and Estimation Methods
- RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems
- shrinkTVP: Efficient Bayesian Inference for Time-Varying Parameter Models with Shrinkage
- spcr: Sparse Principal Component Regression
- SuperPCA: Supervised Principal Component Analysis
- svd: Interfaces to Various State-of-Art SVD and Eigensolvers
- tapkee: tapkee: Wrapper for 'tapkee' Dimension Reduction Library
- tidydr: Unify Dimensionality Reduction Results
- TSrepr: Time Series Representations (dimensionality reduction, preprocessing, feature extraction)
- umap: Uniform Manifold Approximation and Projection

7.20 Distances and Similarity

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- DataGene: Identify How Similar TS Datasets Are to One Another
- dcor: Distance correlation and related E-statistics
- dtaidistance: Distance measures for time series
- dtw-python: comprehensive implementation of dynamic time warping (DTW) algorithms
- faiss: efficient similarity search and clustering of dense vectors
- FLANN: Fast Library for Approximate Nearest Neighbors
- GraKeL: scikit-learn compatible library for graph kernels
- khiva-python: Python binding for Khiva library for time series analytics
- mass-ts: MASS (Mueen's Algorithm for Similarity Search)
- MatrixProfile: ime series data mining tasks utilizing matrix profile
- matrixprofile-ts: detect patterns and anomalies in massive datasets using Matrix Profile
- netrd: library for network {reconstruction, distances, dynamics}
- POT : Python Optimal Transport
- PyMD: imple but general framework for embedding, called Minimum-Distortion Embedding (MDE), for finite sets of items, such as images, biological cells, nodes in a network, or any other abstract object
- PySCAMP: SCAlable Matrix Profile
- seriesdistancematrix: implements the Series Distance Matrix framework, a flexible component-based framework that bundles various Matrix Profile related techniques
- sktime: unified framework for machine learning with time series by UK national institute for data science and artificial intelligence
- Stone-Soup: framework for the development and testing of tracking algorithms

- stumpy: variety of time series data mining tasks
- tidydr: Unify Dimensionality Reduction Results
- timesmash: Quantifier of universal similarity amongst arbitrary data streams without a priori knowledge, features, or training
- wildboar: Time series learning

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- dispRity: Measuring Disparity (multidimensional space occupancy)
- Distance: Distance Sampling Detection Function and Abundance Estimation
- distantia: Assessing Dissimilarity Between Multivariate Time Series
- dtw: Dynamic Time Warping Algorithms
- dtwclust: Time Series Clustering Along with Optimizations for the Dynamic Time Warping Distance
- fICA: Classical, Reloaded and Adaptive FastICA Algorithms
- gdm: Generalized Dissimilarity Modeling
- IncDTW: Incremental Calculation of Dynamic Time Warping
- KernelKnn: Extends the simple k-nearest neighbors algorithm by incorporating numerous kernel functions and a variety of distance metrics
- MatrixCorrelation: Matrix Correlation Coefficients
- mclustcomp: Measures for Comparing Clusters
- Mercator: Clustering and Visualizing Distance Matrices
- philentropy: Similarity and Distance Quantification Between Probability Functions
- proxy: Distance and Similarity Measures
- segen: Sequence Generalization Through Similarity Network
- tetragon: Automatic Sequence Prediction by Expansion of the Distance Matrix
- TSclust: set of measures of dissimilarity between time series to perform time series clustering
- TSdist: Distance Measures for Time Series Data
- tsmp: UCR Matrix Profile Algorithm
- VPdtw: Variable Penalty Dynamic Time Warping

7.21 ESG and Impact Investing

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- ESG AI: ESG scoring as an automatic, data-driven process
- ESG-BERT: Domain Specific BERT Model for Text Mining in Sustainable Investing

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- EnvStats: Package for Environmental Statistics, Including US EPA Guidance
- ESGBoost: ESG and ECHO-based model for smart investing
- gfer: Green Finance and Environmental Risk
- text2sdg: Detecting UN Sustainable Development Goals in Text

7.22 Explainability, Interpretability, Fairness, Data Privacy

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- AIF360: comprehensive set of fairness metrics for datasets and machine learning models, explanations for these metrics, and algorithms to mitigate bias in datasets and models
- captum: Model interpretability and understanding for PyTorch
- CrypTen: framework for Privacy Preserving Machine Learning
- Dice-ML: Generate Diverse Counterfactual Explanations for any machine learning model
- DoWhy: toolkit by Microsoft for causal inference that supports explicit modeling and testing of causal assumptions
- Interpret: Fit interpretable models by Microsoft. Explain blackbox machine learning
- Interpretability dashboard, for understanding model predictions
- Lime: Local Interpretable Model-Agnostic Explanations for machine learning classifiers
- Lucid: neural network interpretability
- PyExplainer: A Local Rule-Based Model-Agnostic Technique
- Skater: Model Interpretation/Explanations
- transformers-interpret: Model explainability that works seamlessly with transformers
- XAI: eXplainability toolbox for machine learning
- Xplique: toolkit dedicated to explainability, currently based on Tensorflow

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- DALEX: moDel Agnostic Language for Exploration and eXplanation
- distillML: Model Distillation and Interpretability Methods for Machine Learning Models
- fairml: Fair Models in Machine Learning
- iml: Interpretable Machine Learning
- interpret: Fit Interpretable Machine Learning Models
- modelDown: Make Static HTML Website for Predictive Models
- modelStudio: Interactive Studio for Explanatory Model Analysis

- pdp: Partial Dependence Plots
- pre: Prediction Rule Ensembles
- Rforestry: Random Forests, Linear Trees, and Gradient Boosting for Inference and Interpretability
- rSAFE: Surrogate-Assisted Feature Extraction
- sensitivity: Global Sensitivity Analysis of Model Outputs
- triplot: Explaining Correlated Features in Machine Learning Models
- yhat: Interpreting Regression Effects

7.23 Features for time series

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- cesium: Platform for Time Series Inference
- Clairvoyance: a Unified, End-to-End AutoML Pipeline for Medical Time Series
- FeatureTools: automated feature engineering
- Featurewiz: advanced feature engineering strategies
- khiva-python: Python binding for Khiva library for time series analytics
- mne-features: MNE-Features software for extracting features from multivariate time series
- scikit-learn: machine learning in Python
- seglearn: integrated pipeline for segmentation, feature extraction, feature processing, and final estimator
- tsfeatures: Calculates various features from time series data
- tsfel: extract features from time series
- tsflex: Flexible time series feature extraction & processing
- tsfresh: extract features from time series

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- autostsm: Automatic Structural Time Series Models
- bfast: Breaks for Additive Season and Trend
- entropy: Estimation of Entropy, Mutual Information and Related Quantities
- feasts: Feature Extraction and Statistics for Time Series
- fsMTS: Feature Selection for Multivariate Time Series
- naive: Empirical Extrapolation of Time Feature Patterns
- plsVarSel: Variable Selection in Partial Least Squares
- MDFS: MultiDimensional Feature Selection
- Rcatch22: Calculation of 22 CAnonical Time-Series CHaracteristics

- theft: Tools for Handling Extraction of Features from Time Series
- tsfeatures: Time Series Feature Extraction
- TSrepr: Time Series Representations (dimensionality reduction, preprocessing, feature extraction)

7.24 Filtering and spectral analysis for time series

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- FilterPy: Kalman filtering and optimal estimation library
- mkl_fft: NumPy-based Python interface to Intel (R) MKL FFT functionality
- pyfilter: Particle filtering and sequential parameter inference
- PyWavelets: Wavelet Transforms in Python
- simdkalman: Kalman filters vectorized as Single Instruction, Multiple Data
- Stone-Soup: framework for the development and testing of tracking algorithms

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ASSA: Applied Singular Spectrum Analysis (ASSA)
- beyondWhittle: Bayesian Spectral Inference for Stationary Time Series
- BMAmevt: Multivariate Extremes: Bayesian Estimation of the Spectral Measure
- bsamGP: Bayesian Spectral Analysis Models using Gaussian Process Priors
- bspec: Bayesian Spectral Inference
- cohortBuilder: Data Source Agnostic Filtering Tools
- EMD: Empirical Mode Decomposition and Hilbert Spectral Analysis
- FKF: Fast Kalman Filter
- FKF.SP: Fast Kalman Filtering Through Sequential Processing
- frequencyConnectedness: Spectral Decomposition of Connectedness Measures
- kalmanfilter: Kalman Filter
- KFAS: Kalman Filter and Smoother for Exponential Family State Space Models
- kimfilter: Rcpp' implementation of the multivariate Kim filter, which combines the Kalman and Hamilton filters for state probability inference
- LMfilteR: Filter Methods for Parameter Estimation in Linear and Non Linear Regression Models
- mlr3filters: Filter Based Feature Selection for 'mlr3'
- multitaper: Spectral Analysis Tools using the Multitaper Method
- neverhpfilter: An Alternative to the Hodrick-Prescott Filter
- praznik: Tools for Information-Based Feature Selection and Scoring

- psd: Adaptive, Sine-Multitaper Power Spectral Density and Cross Spectrum Estimation
- psdr: Use Time Series to Generate and Compare Power Spectral Density
- quantspec: Quantile-Based Spectral Analysis of Time Series
- Rfssa: Functional Singular Spectrum Analysis
- rhosa: Higher-Order Spectral Analysis
- robfilter: Robust Time Series Filters
- RobKF: Innovative and/or Additive Outlier Robust Kalman Filtering
- RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems
- Rssa: A Collection of Methods for Singular Spectrum Analysis
- Rwave: Time-Frequency Analysis of 1-D Signals
- SLBDD: Statistical Learning for Big Dependent Data
- spectral: Common Methods of Spectral Data Analysis
- Spectrum: Fast Adaptive Spectral Clustering for Single and Multi-View Data
- spooky: Time Feature Extrapolation Using Spectral Analysis and Jack-Knife Resampling
- wavethresh: Wavelets Statistics and Transforms

7.25 Forecasting time series

Collections of resources

List of links:

- Popular Python Time Series Packages
- State of the art research (with codes) on time series forecasting

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- anticipy: time series forecasting
- atspy: Automated Time Series Models in Python
- Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting
- AutoTS: Automated Time Series Forecasting
- Auto_TS: Automatically build multiple Time Series models using a Single Line of Code
- Clairvoyance: Unified, End-to-End AutoML Pipeline for Medical Time Series
- darts: toolkit by Unit8 for easy manipulation and forecasting of time series
- ETNA Time Series Library by Tinkoff AI
- fbprophet: forecasting toolkit by Facebook
- fireTS: multi-variate time series prediction library working with sklearn
- Flow Forecast: Deep learning PyTorch library for time series forecasting, classification, and anomaly detection

- glum: Generalized linear models
- GluonTS: toolkit by Amazon for Probabilistic time series modeling in Python
- greykite: flexible, intuitive and fast forecasting library by LinkedIn
- hcrystallball: unifies the API for most commonly used libraries and modeling techniques for time-series forecasting in the Python ecosystem
- HierarchicalForecast: Hierarchical forecasting with statistical and econometric methods
- kats: tookit by Facebook for time series analysis and forecasting
- lazypredict: build models without much code
- Local Cascade Ensemble (LCE) is a high-performing, scalable and user-friendly machine learning method for the general tasks of Classification and Regression
- MAPIE: scikit-learn-compatible module for estimating prediction intervals.
- Merlion: A Machine Learning Framework for Time Series Intelligence by SalesForce
- MLForecast: Scalable machine learning based time series forecasting
- NGBoost: Natural Gradient Boosting for Probabilistic Prediction
- N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting
- NeuralForecast: time series forecasting with deep learning models
- nixtla: Automated time series processing and forecasting
- Orbit: Bayesian forecasting package by Uber
- piecewise-regression: For fitting straight line models to data with one or more breakpoints where the gradient changes
- pmdarima: tatistical library designed to fill the void in Python's time series analysis capabilities
- predictionrevisited: implements the core statistical concepts from the book "Prediction Revisited: The Importance of Observation"
- Prophet: Automatic Forecasting Procedure by Facebook
- PyAF: Automatic Time Series Forecasting
- PyFlux: modern time series models, nference options (frequentist and Bayesian) that can be applied to these models
- pyFTS: Fuzzy Time Series for Python
- pysf: Supervised forecasting of sequential data by UK national institute for data science and artificial intelligence
- PyTorch Forecasting: Forecasting timeseries with PyTorch - dataloaders, normalizers, metrics and models
- pyts: time series classification
- pytsal: Time Series analysis, visualization, forecasting along with AutoTS
- scikit-hts: Hierarchical Time Series Forecasting
- scikit-learn: machine learning in Python

- sktime: unified framework for machine learning with time series by UK national institute for data science and artificial intelligence
- slearn: package linking symbolic representation with scikit-learn machine learning
- statsforecast: Lightning fast forecasting with statistical and econometric models
- Statsmodels: statistical modeling and econometrics in Python
- tbats: BATS and TBATS time series forecasting methods
- timemachines: Autonomous, univariate, k-step ahead time-series forecasting functions assigned Elo ratings
- TIMEX: time series forecasting as a service
- TSCV: Time Series CrossValidation
- ts-eval: Time Series analysis and evaluation tools
- tslearn: machine learning toolkit dedicated to time series data

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ACV: Optimal Out-of-Sample Forecast Evaluation and Testing under Stationarity
- AIafter: Forecast Combination Using the AI-AFTER Algorithm
- arfima: Fractional ARIMA (and Other Long Memory) Time Series Modeling
- ATAforecasting: Automatic Time Series Analysis and Forecasting Using the Ata Method
- autoTS: Automatic Model Selection and Prediction for Univariate Time Series
- baguette: Efficient Model Functions for Bagging
- bigtime: Sparse Estimation of Large Time Series Models
- BINtools: Bayesian BIN (Bias, Information, Noise) Model of Forecasting
- boot.pval: Bootstrap p-Values
- caretForecast: Time Series Forecasting Using Caret Infrastructure
- cvms: Cross-Validation for Model Selection
- dsos: Dataset Shift with Outlier Scores
- ensembleBMA: Probabilistic Forecasting using Ensembles and Bayesian Model Averaging
- fable: Forecasting Models for Tidy Time Series
- fable.ata: ATAforecasting Modelling Interface for fable Framework
- fable.prophet: Prophet Modelling Interface for 'fable'
- fabletools: Core Tools for Packages in the 'fable' Framework
- FinnTS: Microsoft Finance Time Series Forecasting Framework
- flexmix: Flexible Mixture Modeling
- ForeCA: Forecastable Component Analysis

- ForecastComb: Forecast Combination Methods
- forecastHybrid: Convenient Functions for Ensemble Time Series Forecasts
- forecastML: Time Series Forecasting with Machine Learning Methods
- forecastSNSTS: Forecasting for Stationary and Non-Stationary Time Series
- ForecastTB: Test Bench for the Comparison of Forecast Methods
- FoReco: Point Forecast Reconciliation
- forecTheta: Forecasting Time Series by Theta Models
- fpp3: Data for ”Forecasting: Principles and Practice” (3rd Edition)
- fracdiff: Fractionally Differenced ARIMA aka ARFIMA(P,d,q) Models
- fwildclusterboot: Fast Wild Cluster Bootstrap Inference for Linear Models
- greybox: Toolbox for Model Building and Forecasting
- Greymodels: Shiny App for Grey Forecasting Model
- hts: Hierarchical and Grouped Time Series
- ipred: Improved Predictors
- legion: Forecasting Using Multivariate Models
- MAPA: Multiple Aggregation Prediction Algorithm
- mFLICA: Leadership-Inference Framework for Multivariate Time Series
- modeltime: The Tidymodels Extension for Time Series Modeling
- modeltime.ensemble: Ensemble Algorithms for Time Series Forecasting with Modeltime
- modeltime.gluonts: ’GluonTS’ Deep Learning
- modeltime.resample: Resampling Tools for Time Series Forecasting
- ngboostForecast: Probabilistic Time Series Forecasting
- OOS: Out-of-Sample Time Series Forecasting
- origami: Generalized Framework for Cross-Validation
- pre: Prediction Rule Ensembles
- predtoolsTS: Time Series Prediction Tools
- profoc: Probabilistic Forecast Combination Using CRPS Learning
- prophet: Automatic Forecasting Procedure
- PSF: Forecasting of Univariate Time Series Using the Pattern Sequence-Based Forecasting (PSF) Algorithm
- PTSR: Positive Time Series Regression
- RFpredInterval: Prediction Intervals with Random Forests and Boosted Forests
- rigr: Regression, Inference, and General Data Analysis Tools in R
- Rlgt: Bayesian Exponential Smoothing Models with Trend Modifications

- robets: Forecasting Time Series with Robust Exponential Smoothing
- robustarima: Robust ARIMA Modeling
- scoringfunctions: A Collection of Scoring Functions for Assessing Point Forecasts
- scoringRules: Scoring Rules for Parametric and Simulated Distribution Forecasts
- scoringutils: Utilities for Scoring and Assessing Predictions
- s2dverification: Set of Common Tools for Forecast Verification
- see: Visualisation Toolbox for 'easystats' and Extra Geoms, Themes and Color Palettes for 'ggplot2'
- seer: Feature-Based Forecast Model Selection
- segmented: Regression Models with Break-Points / Change-Points Estimation
- sense: Automatic Stacked Ensemble for Regression Tasks
- shrink: Global, Parameterwise and Joint Shrinkage Factor Estimation
- SLBDD: Statistical Learning for Big Dependent Data
- smooth: Forecasting Using State Space Models
- spcr: Sparse Principal Component Regression
- SPlit: Split a Dataset for Training and Testing
- StabilizedRegression: Stabilizing Regression and Variable Selection
- stacks: Tidy Model Stacking
- subsemble: An Ensemble Method for Combining Subset-Specific Algorithm Fits
- tensorTS: Factor and Autoregressive Models for Tensor Time Series
- tfarima: Transfer Function and ARIMA Models
- thief: Temporal Hierarchical Forecasting
- tidymv: Tidy Model Visualisation for Generalised Additive Models
- traineR: Predictive Models Homologator
- TSdeeplearning: Deep Learning Model for Time Series Forecasting
- tsDyn: Nonlinear Time Series Models with Regime Switching
- tsensembler: Dynamic Ensembles for Time Series Forecasting
- TSPred: Functions for Benchmarking Time Series Prediction
- TSstudio: Functions for Time Series Analysis and Forecasting
- tsutils: Time Series Exploration, Modelling and Forecasting
- tswge: Time Series for Data Science.Accompanies the texts Time Series for Data Science and Applied Time Series Analysis with R,
- vars: VAR Modelling
- yardstick: Tidy Characterizations of Model Performance
- yhat: Interpreting Regression Effects

7.26 Graphs and graphical modeling

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- ogb: Benchmark datasets, data loaders, and evaluators for graph machine learning
- pathpy: analysis of time series data on networks using higher-order and multi-order graphical models
- PGM: Probabilistic Graphical Models
- pgmpy: Probabilistic Graphical Models
- PGM_PyLib: Inference and Learning of Probabilistic Graphical Models
- pyaGrUM: Bayesian networks and other Probabilistic Graphical Models
- scikit-network: nalysis of large graphs
- skggm: Scikit-learn compatible estimation of general graphical models
- vishwakarma: visualization library for Probabilistic Graphical Models, Discrete & Continuous Distributions, and a lot more

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- backbone: Extracts the Backbone from Graphs
- deepgp: Deep Gaussian Processes using MCMC
- gmgm: Gaussian Mixture Graphical Model Learning and Inference
- pcalg: Methods for Graphical Models and Causal Inference
- Revticulate: Interaction with ”RevBayes” in R
- tgp: Bayesian Treed Gaussian Process Models
- tidygraph: A Tidy API for Graph Manipulation

7.27 Linear algebra

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- arctic: High performance datastore by Man Group for time series and tick data
- PyArma: Linear algebra library for Python
- PyArmadillo: an alternative approach to linear algebra in Python
- PyPardiso: Python interface to the Intel MKL Pardiso library to solve large sparse linear systems of equations
- Scipy: mathematics, science, and engineering

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- EigenR: Complex Matrix Algebra with 'Eigen'
- fastmatrix: Fast Computation of some Matrices Useful in Statistics
- freqdom: Frequency Domain Based Analysis: Dynamic PCA
- ica: Independent Component Analysis
- Matrix: Sparse and Dense Matrix Classes and Methods
- MatrixExtra: Extra Methods for Sparse Matrices
- matsbyname: An Implementation of Matrix Mathematics
- proxyC: Computes Proximity in Large Sparse Matrices
- rARPACK: Solvers for Large Scale Eigenvalue and SVD Problems
- RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library
- RcppEigen: 'Rcpp' Integration for the 'Eigen' Templated Linear Algebra Library
- Rlinsolve: Iterative Solvers for (Sparse) Linear System of Equations
- RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems
- sanic: Solving Ax = b Nimbly in C++
- SparseChol: Sparse Cholesky LDL Decomposition of Symmetric Matrices
- SparseM: Sparse Linear Algebra
- svd: Interfaces to Various State-of-Art SVD and Eigensolvers

7.28 Machine Learning

Collections of resources

List of links:

- Curated list of open source libraries to deploy, monitor, version and scale machine learning
- Dive into Machine Learning
- Artificial Intelligence and Machine Learning For Quantum Technologies

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best Subset Selection
- AIF360: comprehensive set of fairness metrics for datasets and machine learning models, explanations for these metrics, and algorithms to mitigate bias in datasets and models
- benchmark_VAE: Unifying Generative Autoencoder implementations in Python
- bindsnet: Simulation of spiking neural networks (SNNs) using PyTorch
- biosphere: Simple, fast random forests

- Catalyst: PyTorch framework for Deep Learning Research and Development
- catboost: Gradient Boosting on Decision Trees by Yandex
- Chainer: flexible framework of neural networks for deep learning
- combo: A Python Toolbox for Machine Learning Model Combination
- compose: machine learning tool for automated prediction engineering
- coremltools: convert machine learning models from third-party libraries to the Core ML format (by Apple)
- CrypTen: framework for Privacy Preserving Machine Learning
- DeepChecks: Testing and Validating ML Models and Data
- DoubleML: Double Machine Learning in Python
- Driblet - Google Cloud based ML pipeline by Google
- geotorch: Constrained optimization toolkit for PyTorch
- GPyTorch: Gaussian processes for modern machine learning systems.
- Hub for Tensorflow: library for transfer learning by reusing parts of TensorFlow models
- Hummingbird: library by Microsoft for compiling trained traditional ML models into tensor computations
- InvarianceUnitTests: Linear unit-tests for invariance discovery
- JAX: toolkit by Google for composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more
- jraph: Graph Neural Network Library in Jax
- karateclub: Framework for Unsupervised Learning on Graphs
- keras: deep learning API written in Python, running on top of the machine learning platform TensorFlow
- Local Cascade Ensemble (LCE) is a high-performing, scalable and user-friendly machine learning method for the general tasks of Classification and Regression
- LightGBM: fast, distributed, high performance gradient boosting (GBT, GBDT, GBRT, GBM or MART) framework by Microsoft
- Merlion: A Machine Learning Framework for Time Series Intelligence by SalesForce
- mlflow: Interface to 'MLflow'
- MLForecast: Scalable machine learning based time series forecasting
- mlinsights: Extends scikit-learn with new models, transformers, metrics, plotting.
- MLJAR Automated Machine Learning for Humans
- mlxtend: extension and helper modules for Python's data analysis and machine learning libraries.
- MMdnn: toolkit by Microsoft to convert models between Caffe, Keras, MXNet, Tensorflow, CNTK, PyTorch Onnx and CoreML.
- Model Garden for TensorFlow
- mvlearn is an open-source Python software package for multiview learning tools.

- NannyM: estimate post-deployment model performance (without access to targets), detect data drift, and intelligently link data drift alerts back to changes in model performance
- NeuralForecast: time series forecasting with deep learning models
- NGBoost: Natural Gradient Boosting for Probabilistic Prediction
- nimbusml: toolkit by Microsoft that provides Python bindings for ML.NET
- nolearn: Combines the ease of use of scikit-learn with the power of Theano/Lasagne
- norse: Deep learning with spiking neural networks (SNNs) in PyTorch.
- OPACUS: Training PyTorch models with differential privacy
- ptgnn: PyTorch Graph Neural Network Library
- PyCaret : machine learning library
- PyTorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration
- PyTorch Lightning: lightweight PyTorch wrapper for ML researchers
- Ray: packaged with RLlib, a scalable reinforcement learning library, and Tune, a scalable hyperparameter tuning librar
- scikit-learn: machine learning in Python
- scikit-learn-intelex: Intel Extension for Scikit-learn
- sklearn-onnx converts scikit-learn models to ONNX
- skorch: scikit-learn compatible neural network library that wraps PyTorch
- SNNTORCH: Deep and online learning with spiking neural networks
- tensorflow: end-to-end open source platform for machine learning
- tf2onnx: Convert TensorFlow, Keras, Tensorflow.js and Tflite models to ONNX
- Transfer Learning Library for Domain Adaptation, Task Adaptation, and Domain Generalization
- transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX
- Trax: Deep Learning by Google with Clear Code and Speed
- tslearn: machine learning toolkit dedicated to time-series data
- xformers: Hackable and optimized Transformers building blocks, supporting a composable construction
- yellowbrick: Visual analysis and diagnostic tools to facilitate machine learning model selection

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best Subset Selection
- agua: 'tidymodels' Integration with 'h2o'
- APML: An Approach for Machine-Learning Modelling
- arenar: Arena for the Exploration and Comparison of any ML Models
- brulee: High-Level Modeling Functions with 'torch'

- distillML: Model Distillation and Interpretability Methods for Machine Learning Models
- elmNNRcpp: The Extreme Learning Machine Algorithm
- fairmodels: Flexible Tool for Bias Detection, Visualization, and Mitigation
- familiar: End-to-End Automated Machine Learning and Model Evaluation
- KernelKnn: Extends the simple k-nearest neighbors algorithm by incorporating numerous kernel functions and a variety of distance metrics
- lightgbm: Light Gradient Boosting Machine by Microsoft
- MachineShop: Machine Learning Models and Tools
- mcboost: Multi-Calibration Boosting
- MetricsWeighted: Weighted Metrics, Scoring Functions and Performance Measures for Machine Learning
- mikropml: User-Friendly R Package for Supervised Machine Learning Pipelines
- mlflow: Interface to 'MLflow'
- mlquantify: Algorithms for Class Distribution Estimation
- mlr3: Machine Learning in R - Next Generation
- mlr3cluster: Cluster Extension for 'mlr3'
- mlr3learners: Recommended Learners for 'mlr3'
- mlr3tuning: hyperparameter tuning with 'mlr3'
- mlr3verse: package family is a set of packages for machine-learning purposes built in a modular fashion
- mlr3viz: Visualizations for'mlr3
- mlrintermbo: Model-Based Optimization for 'mlr3' Through 'mlrMBO'
- mlrMBO: Bayesian Optimization and Model-Based Optimization of Expensive Black-Box Functions
- multiview: Cooperative Learning for Multi-View Analysis
- rTorch: R Bindings to 'PyTorch'
- SPlit: Split a Dataset for Training and Testing
- tensorflow: R Interface to 'TensorFlow'
- tfdatasets: Interface to 'TensorFlow' Datasets
- tfprobability: Interface to 'TensorFlow Probability'
- TSdeeplearning: Deep Learning Model for Time Series Forecasting
- xgboost: Extreme Gradient Boosting

7.29 Machine Learning frameworks (includes Automated ML and hyperparameters tuning)

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- AI2 Tango: library for choreographing machine learning research
- AutoGluon: toolkit by Amazon on AutoML for Text, Image, and Tabular Data
- AutoKeras: An AutoML system based on Keras
- AutoPyTorch: Automatic architecture search and hyperparameter optimization for PyTorch
- auto-sklearn: Automated Machine Learning with scikit-learn
- BayesianOptimization: global optimization with gaussian processes.
- cesium: Machine Learning Time-Series Platform
- Clairvoyance: Unified, End-to-End AutoML Pipeline for Medical Time Series
- Colossal-AI: A Unified Deep Learning System for Big Model Era
- EvalML is an AutoML library which builds, optimizes, and evaluates machine learning pipelines using domain-specific objective functions.
- FLAML: accurate machine learning models automatically, efficiently and economically (by Microsoft)
- flax: neural network library for JAX that is designed for flexibility
- H2O is an Open Source, Distributed, Fast & Scalable Machine Learning Platform
- Hypernets: General Automated Machine Learning framework
- HyperOpt: Distributed Asynchronous Hyperparameter Optimization
- hyperopt-sklearn: Hyper-parameter optimization for sklearn
- kedro: framework for creating reproducible, maintainable and modular data science code
- kedro-viz: Visualise your Kedro data and machine-learning pipelines and track your experiments.
- keras-tuner: hyperparameter optimization framework
- MLBox: Automated Machine Learning library
- mlpack: 'Rcpp' Integration for the 'mlpack' Library
- mlr3tuning: Tuning for 'mlr3'
- model_search: framework (by Google) that implements AutoML algorithms for model architecture search at scale
- NannyM: estimate post-deployment model performance (without access to targets), detect data drift, and intelligently link data drift alerts back to changes in model performance
- NNI: toolkit by Microsoft to help users automate Feature Engineering, Neural Architecture Search, Hyperparameter Tuning and Model Compression
- oneflow: OneFlow is a deep learning framework designed to be user-friendly, scalable and efficient.
- ONNX: Open Neural Network Exchange is an Open standard for machine learning interoperability

- Optuna: hyperparameter optimization framework
- PyCaret : machine learning library
- squirrel-core: library that enables ML teams to share, load, and transform data in a collaborative, flexible, and efficient way.
- Relevance AI - The ML Platform for Unstructured Data Analysis
- Talos: Hyperparameter Optimization for TensorFlow, Keras and PyTorch
- trax: end-to-end library (by Google Brain) for deep learning that focuses on clear code and speed.
- tune-sklearn: drop-in replacement for Scikit-Learn's GridSearchCV / RandomizedSearchCV – but with cutting edge hyperparameter tuning techniques
- vowpal_wabbit: machine learning system which pushes the frontier of machine learning with techniques such as online, hashing, allreduce, reductions, learning2search, active, and interactive learning

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- autokeras: R Interface to 'AutoKeras'
- automl: Deep Learning with Metaheuristic
- DriveML: Self-Drive Machine Learning Projects
- familiar: End-to-End Automated Machine Learning and Model Evaluation
- mlpack: 'Rcpp' Integration for the 'mlpack' Library
- mlr3tuningspaces: Search Spaces for Hyperparameter Tuning
- ParBayesianOptimization: Parallel Bayesian Optimization of Hyperparameters
- rBayesianOptimization: Bayesian Optimization of Hyperparameters
- RemixAutoML: automation of machine learning, forecasting, feature engineering, model evaluation, model interpretation, recommenders, and EDA.

7.30 Network and graph analysis

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- dantro: handle, transform, and visualize hierarchically structured data
- deeptime: nalysis of time series data including dimensionality reduction, clustering, and Markov model estimation
- ETNA Time Series Library by Tinkoff AI
- fastpath: find the path through a network of nodes
- GraKeL: scikit-learn compatible library for graph kernels
- grapharray: handle network link/node attributes as Numpy arrays
- GraphVite: A General and High-performance Graph Embedding System
- karateclub: Framework for Unsupervised Learning on Graphs

- netrd: etwork {reconstruction, distances, dynamics}
- networkit: toolkit for large-scale network analysis
- NetworkX: Network Analysis in Python
- pandana: Pandas Network Analysis: fast accessibility metrics and shortest paths, using contraction hierarchies
- pyvis: visualizing interactive network graphs
- rustworkx: high performance Python graph library implemented in Rust
- scikit-learn: machine learning in Python
- tslearn: machine learning toolkit dedicated to time-series data

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- backbone: identify the most ‘important’ or ‘significant’ edges in a network
- bnmonitor: An Implementation of Sensitivity Analysis in Bayesian Networks
- bootnet: Bootstrap Methods for Various Network Estimation Routines
- CINNA: Deciphering Central Informative Nodes in Network Analysis
- dbnR: Dynamic Bayesian Network Learning and Inference
- diceR: Diverse Cluster Ensemble in R
- dtwclust: Time Series Clustering Along with Optimizations for the Dynamic Time Warping Distance
- fabisearch: Change Point Detection in High-Dimensional Time Series Networks
- fastkmedoids: Faster K-Medoids Clustering Algorithms: FastPAM, FastCLARA, FastCLARANS
- gRain: Graphical Independence Networks
- heatmaply: Interactive Cluster Heat Maps Using ’plotly’ and ’ggplot2’
- influential: Identification and Classification of the Most Influential Nodes
- MatTransMix: Clustering with Matrix Gaussian and Matrix Transformation Mixture Models
- Mercator: Clustering and Visualizing Distance Matrices
- MixSim: Simulating Data to Study Performance of Clustering Algorithms
- mixture: Mixture Models for Clustering and Classification
- MKMeans: A Modern K-Means (MKMeans) Clustering Algorithm
- ndtv: Network Dynamic Temporal Visualizations
- network: Classes for Relational Data
- networkABC: Network Reverse Engineering with Approximate Bayesian Computation
- networkDynamic: Dynamic Extensions for Network Objects
- NetworKit: tool suite for high-performance network analysis
- networktools: Tools for Identifying Important Nodes in Networks

- statnet: Software Tools for the Statistical Analysis of Network Data
- visNetwork: Network Visualization using 'vis.js' Library
- wdnet: Weighted and Directed Networks
- WGCNA: Weighted Correlation Network Analysis

7.31 Numerical methods (includes numerical optimization)

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- ADE: Asynchronous Differential Evolution, with efficient multiprocessing
- autoray: Write numeric code that automatically works with any numpy-ish libraries
- BayesianOptimization: global optimization with gaussian processes
- CasADi is a symbolic framework for numeric optimization implementing automatic differentiation in forward and reverse modes on sparse matrix-valued computational graphs
- cmaes: Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
- coco: Numerical Black-Box Optimization Benchmarking Framework
- cp_solver: CP-SAT Solver by Google
- cvxopt: convex optimization
- cvxpy: convex optimization
- DEAP: Distributed Evolutionary Algorithms in Python
- derivative: Numerical differentiation of noisy time series data
- Differential Evolution expensiveopt
- eigenpy: Efficient Python bindings between Numpy/Eigen
- ELA drframework: Dimensionality Reduction Framework for Exploratory Landscape Analysis
- evol: grammar for evolutionary algorithms and heuristics
- fcmaes complements scipy optimize by providing additional optimization methods, faster C++/Eigen based implementations and a coordinated parallel retry mechanism.
- gemseo: Generic Engine for Multi-disciplinary Scenarios, Exploration and Optimization
- General Purpose Optimization Library GPOL
- HiGHS: Linear optimization
- hyperactive: optimization and data collection toolbox for convenient and fast prototyping of computationally expensive models
- ipopt: Cython interface for the interior point optimzer IPOPT
- ipyopt: interface for the interior point optimizer COIN-OR IPOpt
- mystic: highly-constrained non-convex optimization and uncertainty quantification
- nevergrad: Python toolbox for performing gradient-free optimization by Facebook

- nlopt: nonlinear optimization
- Open MDAO: optimization framework
- optima: library for numerical optimization calculations
- OR-Tools: optimization toolkit by Google
- osqp: Operator Splitting QP Solver
- pybobyqa: Derivative-Free Optimization with Bound Constraints
- pycma: Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
- pymoo: Multi-objective Optimization
- pyomo: supports a diverse set of optimization capabilities for formulating and analyzing optimization models.
- PyOptSparse: object-oriented framework for formulating and solving nonlinear constrained optimization problems
- PyPDE: solve partial differential equations using finite differences.
- qpsolvers: Quadratic programming solvers in Python with a unified API
- root_numpy: interface between ROOT and NumPy
- scikit-opt: Swarm Optimization methods
- scikit-optimize: Sequential model-based optimization with a 'scipy.optimize' interface
- Scipy: Fundamental algorithms for scientific computing
- SHADE: Success-History Based Parameter Adaptation for Differential Evolution
- stgaircase: data analysis package based on mathematical step functions
- theseus: differentiable nonlinear optimization
- torchquad: High-performance numerical integration on the GPU with PyTorch, JAX and Tensorflow
- torchsde: Differentiable SDE solvers with GPU support and efficient sensitivity analysis
- trust-region:trust-region subproblem solvers for nonlinear optimization

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ao: Alternating Optimization
- bbotk: Black-Box Optimization Toolkit
- CGNM: Cluster Gauss-Newton Method: Find multiple solutions of a nonlinear least squares problem
- CVXR: Disciplined Convex Optimization
- DEoptim: Global Optimization by Differential Evolution
- DEoptimR: Differential Evolution Optimization in Pure R
- ECOSolveR: Embedded Conic Solver in R
- ggblanket: Simplify 'ggplot2' Visualisation

- graDiEnt: derivative-free, optim-style Stochastic Quasi-Gradient Differential Evolution optimization
- itp: The Interpolate, Truncate, Project (ITP) Root-Finding Algorithm
- LowRankQP: Low Rank Quadratic Programming
- miesmuschel: Mixed Integer Evolution Strategies
- minqa: Derivative-Free Optimization Algorithms by Quadratic Approximation
- mlr3mbo: Flexible Bayesian Optimization
- NMOF: Numerical Methods and Optimization in Finance
- osqp: Quadratic Programming Solver using the 'OSQP' Library
- RcppEnsmallen: Header-Only C++ Mathematical Optimization Library for 'Armadillo'
- rvinecopulib: High Performance Algorithms for Vine Copula Modeling
- rgenoud: R Version of GENetic Optimization Using Derivatives
- rmoo: Multi-Objective Optimization in R
- scs: Splitting Conic Solver for linear programs ('LPs'), second-order cone programs ('SOCPs'), semidefinite programs ('SDPs'), exponential cone programs ('ECPs'), and power cone programs ('PCPs'), or problems with any combination of those cone
- SimEngine: A Modular Framework for Statistical Simulations in R
- trustOptim: Trust Region Optimization for Nonlinear Functions with Sparse Hessians

7.32 Probabilistic modeling (includes mixture models and Gaussian Processes)

Links to resources

- Professionally curated list of awesome Conformal Prediction videos, tutorials, books, papers, PhD and MSc theses, articles and open-source libraries

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- beanmachine: inference on probabilistic models
- celerite2: fast and scalable Gaussian Process (GP) Regression
- conformal-rnn: code for "Conformal time-series forecasting", NeurIPS 2021
- crepes: Conformal Regressors and Conformal Predictive Systems
- EnbPI: Ensemble batch prediction intervals
- EnCQR: ensemble conformalized quantile regression (EnCQR)
- GluonTS: toolkit by Amazon for Probabilistic time series modeling in Python
- gptools: Gaussian processes with arbitrary derivative constraints and predictions.
- GPy: Gaussian processes framework
- GPyTorch: Gaussian processes for modern machine learning systems.
- MAPIE: scikit-learn-compatible module for estimating prediction intervals

- NGBoost: Natural Gradient Boosting for Probabilistic Prediction
- orbit-ml: Bayesian forecasting package by Uber
- pgmpy: Probabilistic Graphical Models – learning (Structure and Parameter), inference (Probabilistic and Causal), and simulations in Bayesian Networks
- pplbench: Evaluation Framework for Probabilistic Programming Languages
- PyMC: Bayesian Modeling and Probabilistic Machine Learning with Aesara
- pyro: Deep universal probabilistic programming with Python and PyTorch
- PySloth: Probabilistic Prediction
- skpro: toolkit by UK national institute for data science and artificial intelligence for Supervised domain-agnostic prediction framework for probabilistic modelling
- tinyGP: The tiniest of Gaussian Process libraries
- zhusuan: probabilistic programming library for Bayesian deep learning, generative models, based on Tensorflow

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- AdequacyModel: Adequacy of Probabilistic Models and General Purpose Optimization
- AdMit: Adaptive Mixture of Student-t Distributions
- aldvmm: Adjusted Limited Dependent Variable Mixture Models
- bgmm: Gaussian Mixture Modeling Algorithms and the Belief-Based Mixture Modeling
- bmixture: Bayesian Estimation for Finite Mixture of Distributions
- BNPmix: Bayesian Nonparametric Mixture Models
- bpgmm: Bayesian Model Selection Approach for Parsimonious Gaussian Mixture Models
- ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering
- conformalInference.multi: Conformal Inference Tools for Regression with Multivariate Response
- DistributionOptimization: Distribution Optimization
- distributionsrd: Distribution Fitting and Evaluation
- EMCluster: EM Algorithm for Model-Based Clustering of Finite Mixture Gaussian Distribution
- evmix: Extreme Value Mixture Modelling, Threshold Estimation and Boundary Corrected Kernel Density Estimation
- flexmix: Flexible Mixture Modeling
- flexmixNL: Finite Mixture Modeling of Generalized Nonlinear Models
- GauPro: Gaussian Process Fitting
- gmgm: Gaussian Mixture Graphical Model Learning and Inference
- greta.gp: Gaussian Process Modelling in 'greta'

- hmmr: ”Mixture and Hidden Markov Models with R” Datasets and Example Code
- ltmix: Left-Truncated Mixtures of Gamma, Weibull, and Lognormal Distributions
- MatrixMixtures: Model-Based Clustering via Matrix-Variate Mixture Models
- MGMM: Missingness Aware Gaussian Mixture Models
- mistr: Mixture and Composite Distributions
- mixComp: Estimation of Order of Mixture Distributions
- MixMatrix: Classification with Matrix Variate Normal and t Distributions
- MixSim: Simulating Data to Study Performance of Clustering Algorithms
- mixsmsn: Fitting Finite Mixture of Scale Mixture of Skew-Normal Distributions
- mixreg: Functions to Fit Mixtures of Regressions
- mixSPE: Mixtures of Power Exponential and Skew Power Exponential Distributions for Use in Model-Based Clustering and Classification
- mixsqp: Sequential Quadratic Programming for Fast Maximum-Likelihood Estimation of Mixture Proportions
- mixtools: Tools for Analyzing Finite Mixture Models
- mixture: Mixture Models for Clustering and Classification
- mclust: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation
- mlr3proba: Probabilistic Supervised Learning for ’mlr3’
- MoMPCA: Inference and Clustering for Mixture of Multinomial Principal Component Analysis
- mvgb: Multivariate Probabilities of Scale Mixtures of Multivariate Normal Distributions via the Genz and Bretz (2002) QRSVN Method
- ngboostForecast: Probabilistic Time Series Forecasting
- nlsmsn: Fitting Nonlinear Models with Scale Mixture of Skew-Normal Distributions
- Nmix: Bayesian Inference on Univariate Normal Mixtures
- nvmix: Multivariate Normal Variance Mixtures
- opGMMassessment: Optimized Automated Gaussian Mixture Assessment
- pgmm: Parsimonious Gaussian Mixture Models
- pGPx: Pseudo-Realizations for Gaussian Process Excursions
- pks: Probabilistic Knowledge Structures
- plgp: Particle Learning of Gaussian Processes
- plotmm: Tidy Tools for Visualizing Mixture Models
- QuantileGH: Quantile Least Mahalanobis Distance Estimator for Tukey g-&-h Mixture
- rebmix: Finite Mixture Modeling, Clustering & Classification
- Revticulate: Interaction with ”RevBayes” in R
- RGMM: Robust Mixture Model

- RMixtComp: Mixture Models with Heterogeneous and (Partially) Missing Data
- robmixglm: Robust Generalized Linear Models (GLM) using Mixtures
- Rmixmod: Classification with Mixture Modelling
- RobMixReg: Robust Mixture Regression
- rrMixture: Reduced-Rank Mixture Models
- seqHMM: Mixture Hidden Markov Models for Social Sequence Data and Other Multivariate, Multichannel Categorical Time Series
- skewlmm: Scale Mixture of Skew-Normal Linear Mixed Models
- skewMLRM: Estimation for Scale-Shape Mixtures of Skew-Normal Distributions
- uGMAR: Estimate Univariate Gaussian and Student's t Mixture Autoregressive Models

7.33 Reinforcement learning

Collections of resources

List of links:

- Awesome Reinforcement Learning: Reinforcement learning resources curated
- Awesome Deep RL: curated list of awesome Deep Reinforcement Learning resources

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Acme: a research framework by DeepMind for reinforcement learning
- Baconian: Model-based Reinforcement Learning Framework
- Open AI Baselines: high-quality implementations by OpenAI of reinforcement learning algorithms
- Catalyst.RL: Distributed Framework for Reproducible RL Research
- ChainerRL: deep reinforcement learning library built on top of Chainer
- Coach: Reinforcement Learning by Intel AI Lab
- d3rlpy: offline deep reinforcement learning library
- Decision Transformer: Reinforcement Learning via Sequence Modeling
- DRL with PyTorch: PyTorch implementations of deep reinforcement learning algorithms and environments
- Deep Reinforcement Learning Hands-On
- deer: DEEp Reinforcement learning framework
- Dopamine: research framework by Google for fast prototyping of reinforcement learning algorithms
- ElegantRL: Lightweight and scalable deep reinforcement learning using PyTorch
- FinRL: Deep Reinforcement Learning for Quantitative Finance
- FinRL-Meta: Universe of Near-Real Market Environments for Data-Driven Financial Reinforcement Learning
- garage: toolkit for reproducible reinforcement learning research

- Gym: ttolkit by openAI for toolkit for developing and comparing reinforcement learning algorithms
- HRAC: Generating Adjacency-Constrained Subgoals in Hierarchical Reinforcement Learning
- keras-rl: Deep Reinforcement Learning for Keras
- Mava: library of multi-agent reinforcement learning components and systems
- Multi-Agent Resource Optimization (MARO) platform is an instance of Reinforcement Learning as a Service (RaaS) for real-world resource optimization problems.
- MBRL-Lib: toolbox by Facebook for facilitating development of Model-Based Reinforcement Learning algorithms
- Mushroom RL: modular toolkit able to use modularity allows to use libraries for tensor computation (e.g. PyTorch, Tensorflow) and RL benchmarks (e.g. OpenAI Gym, PyBullet, Deepmind Control Suite)
- PettingZoo: Gym for multi-agent reinforcement learning
- PFRL: PyTorch-based deep reinforcement learning library
- PGPortfolio: Policy Gradient Portfolio
- PyTorchRL: reinforcement learning library focused on modularity and simplicity
- Rainbow: Combining Improvements in Deep Reinforcement Learning
- ReAgent: platform by Facebook for Reasoning systems (Reinforcement Learning, Contextual Bandits, etc.)
- rl: modular, primitive-first, python-first PyTorch library for Reinforcement Learning.
- RLkit: Collection of reinforcement learning algorithms
- RLlib: Ray is packaged with RLlib, a scalable reinforcement learning library, and Tune, a scalable hyperparameter tuning librar
- RLMeta is a light-weight flexible framework for Distributed Reinforcement Learning Research
- rlpyt: Reinforcement Learning in PyTorch
- rlstructures: Facebook library to facilitate the implementation of new reinforcement learning algorithms
- skrl: Modular reinforcement learning
- Stable Baselines3: PyTorch version of Stable Baselines, reliable implementations of reinforcement learning algorithms
- Tensorforce: TensorFlow library for applied reinforcement learning
- TensorLayer: Deep Learning and Reinforcement Learning Library for Scientists and Engineers
- TF-Agents: TensorFlow library for Contextual Bandits and Reinforcement Learning
- Tianshou: PyTorch deep reinforcement learning library
- Tonic RL: Tonic RL library
- TorchBeast: A PyTorch Platform by Facebook for Distributed RL
- TRFL: TensorFlow Reinforcement Learning by DeepMind
- vowpal_wabbit: machine learning system which pushes the frontier of machine learning with techniques such as online, hashing, allreduce, reductions, learning2search, active, and interactive learning

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- Hands-On Reinforcement Learning
- QLearning: Reinforcement Learning using the Q Learning Algorithm
- reinforcelearn: reinforcement learning, including Q-Learning algorithm
- ReinforcementLearning: Model-Free Reinforcement Learning
- RLT: Reinforcement Learning Trees

7.34 Robust numerical methods

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- derivative: Numerical differentiation of noisy time series data
- hypothesize: hypothesis testing using robust statistics
- robusta: interface to many common statistical analyses, performed using through R and RPY2.
- Robustats is a Python library for high-performance computation of robust statistical estimators
- robustbase: Statistical Estimators (Sn, Qn, MAD, IQR)

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- clubSandwich: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections
- l1spectral: An L1-Version of the Spectral Clustering
- L2E: Robust Structured Regression via the L2 Criterion
- pcaPP: Robust PCA by Projection Pursuit
- RCTS: Clustering Time Series While Resisting Outliers
- RDnp: Robust Test for Complete Independence in High-Dimensions
- revss: Robust Estimation in Very Small Samples
- RGMM: Robust Mixture Model
- rigr: Regression, Inference, and General Data Analysis Tools in R
- robcp: Robust Change-Point Tests
- robcor: Robust Correlations
- robfilter: Robust Time Series Filters
- robmixglm: Robust Generalized Linear Models (GLM) using Mixtures
- RobMixReg: Robust Mixture Regression
- RobStatTM: Robust Statistics: Theory and Methods
- robust: Port of the S+ ”Robust Library”

- RobustANOVA: Robust One-Way ANOVA Tests under Heteroscedasticity and Nonnormality
- robustbase: Basic Robust Statistics
- RobustCalibration: Robust Calibration of Imperfect Mathematical Models
- robustcov: Collection of Robust Covariance and (Sparse) Precision Matrix Estimators
- robustHD: Robust Methods for High-Dimensional Data
- rrcov: Scalable Robust Estimators with High Breakdown Point
- RSC: Robust and Sparse Correlation Matrix
- sandwich: Robust Covariance Matrix Estimators
- StabilizedRegression: Stabilizing Regression and Variable Selection
- tsrobprep: Robust Preprocessing of Time Series Data
- walrus: Robust Statistical Methods

7.35 Selection of features, variables, models, data splits

Collections of resources

List of links:

- Data Science Feature Engineering and Selection Tutorials
- Feature Engineering and Selection: A Practical Approach for Predictive Models
- Guide for Feature Engineering and Feature Selection

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best-Subset Selection Library
- boruta_py: Boruta all-relevant feature selection method
- dython: Data analysis tools
- featureclass: Feature engineering library to keep track of feature dependencies, documentation and schema
- feature_engine: library with multiple transformers to engineer and select features for use in machine learning models
- FeatureTools: automated feature engineering
- Featurewiz: advanced feature engineering strategies
- ITMO_FS: Feature selection library
- KnockPy: Knockoffs for controlled variable selection
- kydavra: feature selection
- Py_FS: Feature Selection
- pyHSICLasso: Versatile Nonlinear Feature Selection Algorithm for High-dimensional Data
- python_stepwiseSelection: Automated Backward and Forward Selection

- scikit-learn: machine learning in Python
- scikit-rebate: scikit-learn-compatible Python implementation of ReBATE, a suite of Relief-based feature selection algorithms
- Sklearn-genetic-opt: Hyperparameters tuning and feature selection, using evolutionary algorithms
- sktime: unified framework for machine learning with time series by UK national institute for data science and artificial intelligence
- tsfeatures: Calculates various features from time series data. Python implementation of the R package tsfeatures
- UltraNest: Fit and compare complex models reliably and rapidly. Advanced nested sampling
- zoofs: feature selection using a variety of nature-inspired wrapper algorithms

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- abess: Fast Best-Subset Selection Library
- BAS: Bayesian Variable Selection and Model Averaging using Bayesian Adaptive Sampling
- basad: Bayesian Variable Selection with Shrinking and Diffusing Priors
- BayesVarSel: Bayes Factors, Model Choice and Variable Selection in Linear Models
- bpgmm: Bayesian Model Selection Approach for Parsimonious Gaussian Mixture Models
- bravo: Bayesian Screening and Variable Selection
- care: High-Dimensional Regression and CAR Score Variable Selection
- dials: Tools for Creating Tuning Parameter Values
- EMVS: The Expectation-Maximization Approach to Bayesian Variable Selection
- FeatureTerminatoR: Feature Selection Engine to Remove Features with Minimal Predictive Power
- FSinR: Feature Selection
- fsMTS: Feature Selection for Multivariate Time Series
- gausscov: The Gaussian Covariate Method for Variable Selection
- greybox: Toolbox for Model Building and Forecasting
- hrqglas: Group Variable Selection for Quantile and Robust Mean Regression
- knockoff: The Knockoff Filter for Controlled Variable Selection
- mBvs: Bayesian Variable Selection Methods for Multivariate Data
- MDFS: MultiDimensional Feature Selection
- mlr3fselect: Feature Selection for 'mlr3'
- mplot: Graphical Model Stability and Variable Selection Procedures
- MXM: Feature Selection (Including Multiple Solutions) and Bayesian Networks
- nestfs: Cross-Validated (Nested) Forward Selection

- NonpModelCheck: Model Checking and Variable Selection in Nonparametric Regression
- pcaPP: Robust PCA by Projection Pursuit
- picR: Predictive Information Criteria for Model Selection
- plsVarSel: Variable Selection in Partial Least Squares
- praznik: Tools for Information-Based Feature Selection and Scoring
- prinvars: Principal Variables (methods for reducing the number of features within a data set)
- projpred: Projection Predictive Feature Selection
- Rforestry: Random Forests, Linear Trees, and Gradient Boosting for Inference and Interpretability
- rmcfs: The MCFS-ID Algorithm for Feature Selection and Interdependency Discovery
- rSAFE: Surrogate-Assisted Feature Extraction
- rstanarm: Bayesian Applied Regression Modeling via Stan
- SelectBoost: A General Algorithm to Enhance the Performance of Variable Selection Methods in Correlated Datasets
- SignifReg: Consistent Significance Controlled Variable Selection in Generalized Linear Regression
- sivs: Stable Iterative Variable Selection
- smoothic: Variable Selection Using a Smooth Information Criterion
- SPlit: Split a Dataset for Training and Testing
- splitTools: Tools for Data Splitting
- stabiliser: Stabilising Variable Selection
- stabm: Stability Measures for Feature Selection
- stacks: Tidy Model Stacking
- stepgbm: Stepwise Variable Selection for Generalized Boosted Regression Modeling
- SWIM: Scenario Weights for Importance Measurement
- theft: Tools for Handling Extraction of Features from Time Series
- tornado: Plots for Model Sensitivity and Variable Importance
- valse: Variable Selection with Mixture of Models
- WLasso: Variable Selection for Highly Correlated Predictors

7.36 Sensitivity analysis and numerical derivatives

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- derivative: Numerical differentiation of noisy time series data
- higher: obtain higher order gradients
- jacobi: Numerical derivatives for Python
- JAX: toolkit by Google for composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more
- OMSens: OpenModelica sensitivity analysis and optimization module
- PyApprox: high-dimensional approximation and uncertainty quantification by Sandia Labs
- SALib: Sensitivity Analysis Library (Contains Sobol, Morris, FAST, and other methods)
- sensitivity: Sensitivity Analysis
- tangent: library (by Google) for automatic differentiation providing Source-to-Source Debuggable Derivatives in Pure Python
- torchsde: Differentiable SDE solvers with GPU support and efficient sensitivity analysis

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- bnmonitor: An Implementation of Sensitivity Analysis in Bayesian Networks
- GSA.UN: Global Sensitivity Analysis Tool
- reval: Argument Table Generation for Sensitivity Analysis
- samon: Sensitivity Analysis for Missing Data
- sensemakr: Sensitivity Analysis Tools for Regression Models
- sensitivity: Global Sensitivity Analysis of Model Outputs
- sensobol: Computation of Variance-Based Sensitivity Indices
- SWIM: Scenario Weights for Importance Measurement
- tornado: Plots for Model Sensitivity and Variable Importance

7.37 Statistics and Probability

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- distfit: probability density function fitting and hypothesis testing
- empiricaldist: empirical distribution functions
- momentum: Running mean, variance, skew, and kurtosis
- pingouin: Statistical package in Python based on Pandas
- probs: Probability library

- PyProbables: Probabilistic data structures in python
- PyStats: statistical analysis and distributions
- RunStats: Computing Statistics and Regression in One Pass
- statsmodels: statistical modeling and econometrics
- tensorflow-probability: Probabilistic reasoning and statistical analysis in TensorFlow
- wquantiles: Weighted quantiles

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- arsenal: An Arsenal of 'R' Functions for Large-Scale Statistical Summaries
- ashr: Methods for Adaptive Shrinkage, using Empirical Bayes
- confintr: Confidence Intervals
- DEM: The Distributed EM Algorithms in Multivariate Gaussian Mixture Models
- DescTools: Tools for Descriptive Statistics
- distr6: The Complete R6 Probability Distributions Interface
- distr: Object Oriented Implementation of Distributions
- distrEx: Extensions of Package 'distr'
- distributionsrd: Distribution Fitting and Evaluation
- DPQ: Density, Probability, Quantile ('DPQ') Computations
- EasyDescribe: A Convenient Way of Descriptive Statistics
- entropy: Estimation of Entropy, Mutual Information and Related Quantities
- estimatr: Fast Estimators for Design-Based Inference
- evd: Functions for Extreme Value Distributions
- expectreg: Expectile and Quantile Regression
- fitur: Fit Univariate Distributions
- fromo: Fast Robust Moments
- Gmedian: Geometric Median, k-Medians Clustering and Robust Median PCA
- HSAUR3: A Handbook of Statistical Analyses Using R (3rd Edition)
- lmom: L-Moments
- lmomco: L-Moments, Censored L-Moments, Trimmed L-Moments, L-Comoments, and Many Distributions
- matrixdist: Statistics for Matrix Distributions
- MatrixModels: Modelling with Sparse and Dense Matrices
- matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors)
- minsample2: The Minimum Sample Size

- mlquantify: Algorithms for Class Distribution Estimation
- mvtnorm: Multivariate Normal and t Distributions
- NNS: Nonlinear nonparametric statistics using partial moments
- overlapping: Estimation of Overlapping in Empirical Distributions
- PCDimension: Finding the Number of Significant Principal Components
- philentropy: Similarity and Distance Quantification Between Probability Functions
- pls: Partial Least Squares and Principal Component Regression
- psre: Presenting Statistical Results Effectively
- Qest: Quantile-Based Estimator
- qp: Quantile parametrization for probability distribution functions
- RcppRoll: Efficient Rolling / Windowed Operations
- revss: Robust Estimation in Very Small Samples
- RobStatTM: Robust Statistics: Theory and Methods
- robustbase: Basic Robust Statistics
- roll: Rolling and Expanding Statistics
- statsExpressions: Tidy Dataframes and Expressions with Statistical Details
- walrus: Robust Statistical Methods
- weights: Weighting and Weighted Statistics

7.38 Stress testing, rare events, extreme values and scenarios, survival analysis

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- pyextremes: Extreme Value Analysis
- pycox is a python package for survival analysis and time-to-event prediction with PyTorch
- scikit-extremes: univariate extreme value calculations

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- BMAmevt: Multivariate Extremes: Bayesian Estimation of the Spectral Measure
- climextRemes: Tools for Analyzing Climate Extremes
- extRemes: Extreme Value Analysis
- extremeStat: Extreme Value Statistics and Quantile Estimation
- evd: Functions for Extreme Value Distributions
- evmix: Extreme Value Mixture Modelling, Threshold Estimation and Boundary Corrected Kernel Density Estimation

- ExtremalDep: Extremal Dependence Models
- ExtremeRisks: Extreme Risk Measures
- lax: Loglikelihood Adjustment for Extreme Value Models
- lite: Likelihood-Based Inference for Time Series Extremes
- mev: Modelling of Extreme Values
- survivalmodels: Models for Survival Analysis

7.39 Symbolic regression & data-driven model discovery and machine learning

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- 2SEGP: Simple Simultaneous Ensemble Learning in Genetic Programming
- AIFeynman: Physics-Inspired Method for Symbolic Regression
- BindingGP: Symbolic Regression with Dimension Calculation
- Data Driven Symbolic Regression
- DEAP: Distributed Evolutionary Algorithms
- DeepSymReg: Neural Network-Based Symbolic Regression in Deep Learning for Scientific Discovery
- DeepSymRegTorch: PyTorch implementation of the EQL network, a neural network for symbolic regression
- Deep symbolic optimization
- diffeqpy: Solving differential equations in Python using DifferentialEquations.jl and the SciML Scientific Machine Learning organization
- ellyn: python-wrapped version of ellen, a linear genetic programming system for symbolic regression and classification
- EQLearner: A Seq2Seq approach to Symbolic Regression
- ffx: Fast Function Extraction for symbolic regressio
- geppy: framework for gene expression programming
- gplearn: Genetic Programming in Python, with a scikit-learn inspired API
- hal-cgp: Cartesian genetic programming
- Neural Symbolic Regression That Scales
- pyglyph: library based on deap providing abstraction layers for symbolic regression problems
- pymbolic: Easy Expression Trees and Term Rewriting
- PySR: High-Performance Symbolic Regression in Python
- PySINDy: sparse identification of nonlinear dynamical systems from data
- pySRURGS: Symbolic regression by uniform random global search
- salmon-lm: symbolic algebra of linear regression and modeling
- slearn: package linking symbolic representation with scikit-learn machine learning

- SR Bench: benchmark framework for symbolic regression
- SymEngine is a fast symbolic manipulation library
- symfit: Symbolic Fitting; fitting as it should be.
- symbolic experiments: Repository for symbolic regression/classification experiments
- Symbolic Regression Boosting
- Simpy: symbolic mathematics
- symreg: A Symbolic Regression engine

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- DiffEqR: Solving differential equations in R using DifferentialEquations.jl and the SciML Scientific Machine Learning ecosystem
- gramEvol: Grammatical Evolution for R
- symbolicDA: Analysis of Symbolic Data
- symengine: Interface to the 'SymEngine' Library

7.40 Testing (numerical, statistical, etc.), comparison and ranking

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- AutoTS: Automated Time Series Forecasting
- darts: toolkit by Unit8 for easy manipulation and forecasting of time series
- goftests: Generic goodness of fit tests for random plain old data
- hypothesize: hypothesis testing using robust statistics
- hypothetical: Hypothesis and statistical testing
- hyppo: multivariate hypothesis testing
- InvarianceUnitTests: Linear unit-tests for invariance discovery
- MAPIE: scikit-learn-compatible module for estimating prediction intervals.
- Merlion: A Machine Learning Framework for Time Series Intelligence by SalesForce
- permute: permutation tests and confidence sets
- PhiK: practical correlation constant that works consistently between categorical, ordinal and interval variables
- pingouin: Statistical package in Python based on Pandas
- responsible-ai-toolbox: Error Analysis dashboard, for identifying model errors and discovering cohorts of data for which the model underperforms.
- RunStats: Computing Statistics and Regression in One Pass
- scikit-learn: machine learning in Python
- statsmodels: statistical modeling and econometrics
- UltraNest: Fit and compare complex models reliably and rapidly. Advanced nested sampling.
- xskillscore: Metrics for verifying forecasts

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ACV: Optimal Out-of-Sample Forecast Evaluation and Testing under Stationarity
- amp: Statistical Test for the Multivariate Point Null Hypotheses
- ashr: Methods for Adaptive Shrinkage, using Empirical Bayes
- bayefdr: Bayesian Estimation and Optimisation of Expected False Discovery Rate
- BEST: Bayesian Estimation Supersedes the t-Test
- BFpack: Flexible Bayes Factor Testing of Scientific Expectations
- blocklength: Select an Optimal Block-Length to Bootstrap Dependent Data (Block Bootstrap)
- boot: Bootstrap Functions
- boot.pval: Bootstrap p-Values
- bootUR: Bootstrap Unit Root Tests
- CADFtest: A Package to Perform Covariate Augmented Dickey-Fuller Unit Root Tests
- ChangepointTesting: Change Point Estimation for Clustered Signals
- clusrank: Wilcoxon Rank Tests for Clustered Data
- cocor: Comparing Correlations
- corTESTsrd: Significance Testing of Rank Cross-Correlations under SRD
- CovTools: Statistical Tools for Covariance Analysis
- crossvalidationCP: Cross-Validation for Change-Point Regression
- crseEventStudy: A Robust and Powerful Test of Abnormal Stock Returns in Long-Horizon Event Studies
- cvCovEst: Cross-Validated Covariance Matrix Estimation
- cvms: Cross-Validation for Model Selection
- CVST: Fast Cross-Validation via Sequential Testing
- dgof: Discrete Goodness-of-Fit Tests
- digitTests: Tests for Detecting Irregular Digit Patterns
- DiscreteFDR: Multiple Testing Procedures with Adaptation for Discrete Tests
- dsos: Dataset Shift with Outlier Scores
- elo: Ranking Teams by Elo Rating and Comparable Methods
- energy: E-Statistics: Multivariate Inference via the Energy of Data
- exactRankTests: Exact Distributions for Rank and Permutation Tests
- FactorAssumptions: Set of Assumptions for Factor and Principal Component Analysis
- FAMT: Factor Analysis for Multiple Testing (FAMT) : Simultaneous Tests under Dependence in High-Dimensional Data
- fbst: The Full Bayesian Evidence Test, Full Bayesian Significance Test and the e-Value

- fdrci: Permutation-Based FDR Point and Confidence Interval Estimation
- FDRestimation: Estimate, Plot, and Summarize False Discovery Rates
- funtimes: Nonparametric estimators and tests for time series analysis
- fwb: Fractional Weighted Bootstrap
- fwildclusterboot: Fast Wild Cluster Bootstrap Inference for Linear Models
- gvlma: Global Validation of Linear Models Assumptions
- gt: Easily Create Presentation-Ready Display Tables
- gtExtras: Extending 'gt' for Beautiful HTML Tables
- heplots: Visualizing Hypothesis Tests in Multivariate Linear Models
- HSAUR3: A Handbook of Statistical Analyses Using R (3rd Edition)
- htestClust: Reweighted Marginal Hypothesis Tests for Clustered Data
- ICtest: Estimating and Testing the Number of Interesting Components in Linear Dimension Reduction
- inferr: Inferential Statistics (parametric and non-parametric statistical tests)
- L2DensityGoFtest: Density Goodness-of-Fit Test
- locits: Test of Stationarity and Localized Autocovariance
- mashr: Multivariate Adaptive Shrinkage
- mcStats: Visualize Results of Statistical Hypothesis Tests
- melt: Multiple Empirical Likelihood Tests
- metrica: evaluate prediction performance of point-forecast models
- MixedIndTests: Tests of Randomness and Tests of Independence
- modeltime.resample: Resampling Tools for Time Series Forecasting
- MSTest: Hypothesis Testing for Markov Switching Models
- multDM: Multivariate Version of the Diebold-Mariano Test
- MultiFit: Multiscale Fisher's Independence Test for Multivariate Dependence
- MultiHorizonSPA: Multi Horizon Superior Predictive Ability
- multiverse: 'Explorable Multiverse' Data Analysis and Reports to show the robustness of statistical inference
- MVTests: Multivariate Hypothesis Tests and the confidence intervals
- nestedcv: Nested Cross-Validation with 'glmnet' and 'caret'
- NonParRolCor: a Non-Parametric Statistical Significance Test for Rolling Window Correlation
- OOS: Out-of-Sample Time Series Forecasting
- origami: Generalized Framework for Cross-Validation
- OptSig: Optimal Level of Significance for Regression and Other Statistical Tests
- OPTtesting: Optimal Testing

- OutliersO3: Draws Overview of Outliers (O3) Plots
- pbo: Probability of Backtest Overfitting
- performance: Assessment of Regression Models Performance
- permutes: Permutation Tests for Time Series Data
- poolr: Methods for Pooling P-Values from (Dependent) Tests
- portes: Portmanteau Tests for Univariate and Multivariate Time Series Models
- randtoolbox: Toolbox for Pseudo and Quasi Random Number Generation and Random Generator Tests
- RDieHarder: R Interface to the 'DieHarder' RNG Test Suite
- RDnp: Robust Test for Complete Independence in High-Dimensions
- rigr: Regression, Inference, and General Data Analysis Tools in R
- Rita: Automated Transformations, Normality Testing, and Reporting
- rmcorr: Repeated Measures Correlation
- RobustANOVA: Robust One-Way ANOVA Tests under Heteroscedasticity and Nonnormality
- robusTest: Calibrated Correlation, Two-Sample Tests
- rsample: General Resampling Infrastructure
- rstatix: Pipe-Friendly Framework for Basic Statistical Tests
- s2dverification: Set of Common Tools for Forecast Verification
- scoringfunctions: A Collection of Scoring Functions for Assessing Point Forecasts
- scoringRules: Scoring Rules for Parametric and Simulated Distribution Forecasts
- scoringutils: Utilities for Scoring and Assessing Predictions
- sdafilter: distribution free multiple testing rules for false discovery rate (FDR) control under general dependence
- sgof: Multiple Hypothesis Testing
- SHT: Statistical Hypothesis Testing Toolbox
- slider: Sliding Window Functions
- SlidingWindows: Methods for Time Series Analysis
- SPlit: Split a Dataset for Training and Testing
- splitTools: Tools for Data Splitting
- statsExpressions: Tidy Dataframes and tests (parametric, nonparametric, robust, etc)
- tidyposterior: Bayesian Analysis to Compare Models using Resampling Statistics
- tidystats: Save Output of Statistical Tests
- UnitStat: Performs Unit Root Test Statistics
- urca: Unit Root and Cointegration Tests for Time Series Data
- USP: U-Statistic Permutation Tests of Independence for all Data Types
- walrus: Robust Statistical Methods
- yardstick: Tidy Characterizations of Model Performance

7.41 Testing software codes

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- benchmark: microbenchmark support library
- bugsnag error monitoring and error reporting
- case: Python unittest Utilities
- cxxtest: CxxTest Unit Testing Framework
- dirty-equals: make python code (generally unit tests) more declarative and therefore easier to read and write.
- expectest: implements expect tests (also known as ”golden” tests)
- formencode: validation and form generation
- freezegun: allows your Python tests to travel through time by mocking the datetime module
- green: clean, colorful, fast python test runner
- Hypothesis: family of testing libraries which let you write tests parametrized by a source of examples
- Mamba Test Runner: definitive testing tool for Python
- mutattest: Safely run mutation trials without source code modifications and see what will get past your test suite.
- nose2: unittest with plugins.
- nox: Flexible test automation for Python
- partialtesting: toolkit by Man Group to run only the tests relevant for code changes
- playwright-python: Python version of the Playwright testing and automation library
- Pynguin: PYthoN General UnIt Test geNerator
- pyperformance: intended to be an authoritative source of benchmarks for all Python implementations
- pytest: easy to write small tests, yet scales to support complex functional testing
- pytest-benchmark: py.test fixture for benchmarking code
- pytest-check: pytest plugin that allows multiple failures per test.
- pytest-html: Plugin for generating HTML reports for pytest results
- pytest-parallel: pytest plugin for parallel and concurrent testing
- pytest-regressions: Pytest plugin for regression testing
- stestr: parallel Python test runner built around subunit
- TestSlide: test framework by Facebook
- testtools: extensions to the Python standard library’s unit testing framework.
- tox: Command line driven CI frontend and development task automation tool
- ward: modern test framework for Python with a focus on productivity and readability.

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- exampletestr: Help for Writing Unit Tests Based on Function Examples
- melt: Multiple Empirical Likelihood Tests
- mockthat: Function Mocking for Unit Testing
- patrick: Parameterized Unit Testing by Google
- realtest: When Expectations Meet Reality: Realistic Unit Testing
- shinytest2: Testing for Shiny Applications
- testdat: Data Unit Testing for R
- testthat: Unit Testing for R
- testthis: Utils and 'RStudio' Addins to Make Testing Even More Fun
- ttdo: Extend 'tinytest' with 'diffobj'
- unitizer: Interactive R Unit Tests
- unittest: TAP-Compliant Unit Testing
- xpectr: Generates Expectations for 'testthat' Unit Testing

7.42 Time series analysis and modeling

Collections of resources

List of links:

- Curated list with python packages for time series analysis
- Popular Python Time Series Packages
- Resources for working with time series and sequence data

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Clairvoyance: Unified, End-to-End AutoML Pipeline for Medical Time Series
- darts: toolkit by Unit8 for easy manipulation and forecasting of time series
- DataGene: Identify How Similar TS Datasets Are to One Another
- deeptime: analysis of time series data including dimensionality reduction, clustering, and Markov model estimation
- EntropyHub: open-source toolkit for entropic time-series analysis.
- ETNA Time Series Library by Tinkoff AI
- fastreg: Fast sparse regressions with advanced formula syntax. OLS, GLM, Poisson, Maxlike, and more. High-dimensional fixed effects
- Featuretools: automated feature engineering
- glum: Generalized linear models

- hcrystalball: unifies the API for most commonly used libraries and modeling techniques for time-series forecasting in the Python ecosystem
- HyperTools: toolbox for gaining geometric insights into high-dimensional data
- HyperTS: Full-Pipeline Automated Time Series (AutoTS) Analysis Toolkit
- kats: tookit by Facebook for time series analysis and forecasting
- KFAS: Kalman Filter and Smoother for Exponential Family State Space Models
- khiva-python: Python binding for Khiva library for time series analytics
- Loud ML: inference engine for metrics and events
- luminaire: ML driven solutions for monitoring time series data
- matrixprofile-ts: detect patterns and anomalies in massive datasets using Matrix Profile
- MatrixStats: Methods that Apply to Rows and Columns of Matrices (and to Vectors)
- mkl_fft: NumPy-based Python interface to Intel (R) MKL FFT functionality
- nixtla: Automated time series processing and forecasting
- pandas: data structures for data analysis, time series, and statistics
- pyFFTW is a pythonic wrapper around FFTW 3, the speedy FFT library
- pyFIt-SNE: FFT-accelerated Interpolation-based t-SNE (FIt-SNE)
- pyts: time series classification
- pytsal: Time Series analysis, visualization, forecasting along with AutoTS
- seglearn: machine learning for time series
- sktime: unified framework for machine learning with time series by UK national institute for data science and artificial intelligence
- slearn: package linking symbolic representation with scikit-learn machine learning
- statsmodels: statistical modeling and econometrics
- stumpy: variety of time series data mining tasks
- theft: Tools for Handling Extraction of Features from Time Series
- timemachines: Evaluation and standardization of popular time series packages
- timetk: A Tool Kit for Working with Time Series in R
- Traces: library for unevenly-spaced time series analysis
- tsai: time series tasks like classification, regression, forecasting, imputation
- tsam: time series aggregation module (tsam)
- ts-eval: Time Series analysis and evaluation tools
- tsfresh: extracts relevant characteristics from time series
- tslearn: machine learning toolkit dedicated to time-series data
- tspreprocess: package to preprocess time series
- tsmoothie: time-series smoothing and outlier detection in a vectorized way
- vectorbt: library for backtesting and analyzing trading strategies at scale

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ASSA: Applied Singular Spectrum Analysis
- astsa: Applied Statistical Time Series Analysis
- autostsm: Automatic Structural Time Series Models
- bdots: Bootstrapped Differences of Time Series
- bfast: Breaks for Additive Season and Trend
- bimets: Time Series and Econometric Modeling
- bootUR: Bootstrap Unit Root Tests
- ctbi: A Procedure to Clean, Decompose and Aggregate Timeseries
- energy: E-Statistics: Multivariate Inference via the Energy of Data
- entropy: Estimation of Entropy, Mutual Information and Related Quantities
- freqdom: Frequency Domain Based Analysis: Dynamic PCA
- funtimes: Functions for Time Series Analysis
- garchx: Flexible and Robust GARCH-X Modelling
- LMD: A Self-Adaptive Approach for Demodulating Multi-Component Signal
- LSTS: Locally Stationary Time Series
- lubridate: Make Dealing with Dates a Little Easier
- mcvis: Multi-Collinearity Visualization
- MixedIndTests: Tests of Randomness and Tests of Independence
- MTS: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models
- NonlinearTSA: Nonlinear Time Series Analysis
- nonlinearTseries: Nonlinear Time Series Analysis
- nortsTest: Assessing Normality of Stationary Process
- NTS: Nonlinear Time Series Analysis
- Rfssa: Functional Singular Spectrum Analysis
- rhosa: Higher-Order Spectral Analysis
- rrcov: Scalable Robust Estimators with High Breakdown Point
- rrMixture: Reduced-Rank Mixture Models
- Rssa: A Collection of Methods for Singular Spectrum Analysis
- rtrend: Trend Estimating Tools
- Rwave: Time-Frequency Analysis of 1-D Signals
- seastests: Seasonality Tests

- shrink: Global, Parameterwise and Joint Shrinkage Factor Estimation
- simts: Time Series Analysis Tools
- SLBDD: Statistical Learning for Big Dependent Data
- svars: Data-Driven Identification of SVAR Models
- tempdisagg: Temporal Disaggregation and Interpolation of Time Series
- theft: Tools for Handling Extraction of Features from Time Series
- TidyDensity: Functions for Tidy Analysis and Generation of Random Data
- timetk: A Tool Kit for Working with Time Series in R
- TSA: Time Series Analysis
- tsbox: Class-Agnostic Time Series
- tscopula: Time Series Copula Models
- tseries: Time Series Analysis and Computational Finance
- TSrepr: Time Series Representations
- tsrobprep: Robust Preprocessing of Time Series Data
- TSstudio: Functions for Time Series Analysis and Forecasting
- tsutils: Time Series Exploration, Modelling and Forecasting
- tsviz: Easy and Interactive Time Series Visualization
- vars: VAR Modelling
- xts: eXtensible Time Series

7.43 Text, sentiment and topic analytics (including NLP)

Python software implementations

- AllenNLP: toolkit by Allen Institute of Articial Intelligence for NLP research
- EmTract: Extracting Emotions from Social Media Text Tailored for Financial Contexts
- EvoMSA: Sentiment Analysis System based on B4MSA and EvoDAG
- fairseq: Facebook AI Research Sequence-to-Sequence Toolkit
- FastFormers: toolkit by Microsoft to achieve inference of Transformer models for Natural Language Understanding
- gensim: topic modelling, document indexing and similarity retrieval with large corpora
- GPT-3: Language Models are Few-Shot Learners
- LIT: Language Interpretability Tool: Interactively analyze NLP models for model understanding
- LangTech Text Library (LTTL) is an open-source python package for text processing and analysis.
- Natural Language Processing Best Practices and Examples by Microsoft
- netts: toolkit by UK national institute for data science and artificial intelligence for creating networks capturing semantic content of speech transcripts

- nlpaug: Data augmentation for NLP
- nltk: Natural Language Toolkit
- pytext: A natural language modeling framework based on PyTorch
- PyTorch-NLP: Basic Utilities for PyTorch Natural Language Processing (NLP)
- Senta: Baidu's open-source Sentiment Analysis System.
- spaCy: Industrial-strength Natural Language Processing (NLP) in Python
- stocksight: Stock market analyzer and predictor using Elasticsearch, Twitter, News headlines, NLP and sentiment analysis
- sumy: automatic summarization of text documents and HTML pages
- textacy: NLP, before and after spaCy
- vaderSentiment: VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool
- wordfreq: Access a database of word frequencies, in various natural languages.

R software implementations

- cleanNLP: Tidy Data Model for Natural Language Processing
- doc2concrete: Measuring Concreteness in Natural Language
- fastTextR: An Interface to the 'fastText' Library
- globaltrends: Google Trends portal.
- lsa: Latent Semantic Analysis
- LSX: Model for Semisupervised Text Analysis Based on Word Embeddings
- meanr: Sentiment Analysis Scorer
- NLP: Natural Language Processing Infrastructure
- opitools: Analyzing the Opinions in a Big Text Document
- quanteda: Quantitative Analysis of Textual Data
- saotd: Sentiment Analysis of Twitter Data
- sentiment.ai: Simple Sentiment Analysis Using Deep Learning
- SentimentAnalysis: Dictionary-Based Sentiment Analysis
- sentimentr: Calculate Text Polarity Sentiment
- sentometrics: Integrated Framework for Textual Sentiment Time Series Aggregation and Prediction
- spacyr: Wrapper to the 'spaCy' 'NLP' Library
- sweater: Speedy Word Embedding Association Test and Extras Using R
- syuzhet: Extracts Sentiment and Sentiment-Derived Plot Arcs from Text
- tau: Text Analysis Utilities
- text2map: R Tools for Text Matrices, Embeddings, and Networks

- text2sdg: Detecting UN Sustainable Development Goals in Text
- text2vec: Modern Text Mining Framework for R
- texter: An Easy Text and Sentiment Analysis Library
- TextForecast: Regression Analysis and Forecasting Using Textual Data from a Time-Varying Dictionary
- textTinyR: Text Processing for Small or Big Data Files
- tidytext: Text Mining using 'dplyr', 'ggplot2', and Other Tidy Tools
- transforEmotion: Sentiment Analysis for Text and Qualitative Data
- tsentiment: Fetching Tweet Data for Sentiment Analysis
- Xplortext: Statistical Analysis of Textual Data

7.44 Uncertainty: analysis and modeling

Links to resources

- Professionally curated list of awesome Conformal Prediction videos, tutorials, books, papers, PhD and MSc theses, articles and open-source libraries

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Bumps: data fitting and uncertainty estimation
- conformal-rnn: code for ”Conformal time-series forecasting”, NeurIPS 2021
- crepes: Conformal Regressors and Conformal Predictive Systems
- EasyVVUQ: verification, validation and uncertainty quantification in high performance computing
- EnbPI: Ensemble batch prediction intervals
- EnCQR: ensemble conformalized quantile regression (EnCQR)
- MAPIE: scikit-learn-compatible module for estimating prediction intervals
- mystic: highly-constrained non-convex optimization and uncertainty quantification
- OpenTURNS (Open source initiative to Treat Uncertainties, Risks'N Statistics)
- PySloth: Probabilistic Prediction
- UncertaintyToolbox: predictive uncertainty quantification, calibration, metrics, and visualization
- UQpy: UQpy (Uncertainty Quantification with python) is a general purpose Python toolbox for modeling uncertainty in physical and mathematical systems

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- bootComb: Combine Parameter Estimates via Parametric Bootstrap

7.45 Visualization and reporting

Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- Algviz is an algorithm visualization tool for your Python code
- appmode: Jupyter extension that turns notebooks into web applications
- Best of Streamlit
- clustergram: Visualization and diagnostics for cluster analysis in Python
- dash: framework for building ML and data science web apps
- dash-extensions:extensions to the Plotly Dash framework
- D-tale:Visualizer by Man Group for pandas data structures
- FlameScope: visualization ny Netflix for exploring different time ranges as Flame Graphs.
- HyperTools: toolbox for gaining geometric insights into high-dimensional data
- ipyslides: Create Interactive Slides in Jupyter Notebook with all kind of rich content
- itables: Pandas DataFrames as Interactive DataTables
- Lux: automate the visualization and data analysis process
- Markdown: Python implementation of markdown
- matplotlib: omprehensive library for creating static, animated, and interactive visualizations
- mpl-animators: interative animation framework for matplotlib
- Orange: Interactive data analysis
- plotly: graphing library makes interactive, publication-quality graphs
- Plotly Resampler: Visualize large time-series data in plotly
- plotnine: A grammar of graphics for Python
- plottable: Beautifully customized tables with matplotlib
- psyplot: interactive data visualization
- PyGraphistry: quickly load, shape, embed, and explore big graphs with the GPU-accelerated Graphistry visual graph analyzer
- PyMetis: Python wrapper around Metis, a graph partitioning package
- PyShiny: Shiny for Python
- pyvis: visualizing interactive network graphs
- seaborn: statistical data visualization
- seaborn analyzer: data analysis and visualization tool using Seaborn library
- streamlit: fastest way to build and share data apps
- tensorboard: TensorFlow's Visualization Toolkit
- torchsde: Differentiable SDE solvers with GPU support and efficient sensitivity analysis

- tourr: Tour Methods for Multivariate Data Visualisation
- Vega-Altair is a declarative statistical visualization library for Pytho
- VisPy: interactive scientific visualization in Python
- visdom: lexible tool for creating, organizing, and sharing visualizations of live, rich data

R software implementations

List of packages and/or codes and/or frameworks and/or links:

- apexcharter: Create Interactive Chart with the JavaScript 'ApexCharts' Library
- autoplotly: Automatic Generation of Interactive Visualizations for Statistical Results
- classmap: Visualizing Classification Results
- cleanrmd: Clean Class-Less 'R Markdown' HTML Documents
- clustree: Visualise Clusterings at Different Resolutions
- ComplexUpset: Create Complex UpSet Plots Using 'ggplot2' Components
- condformat: Conditional Formatting in Data Frames
- conductor: Create Tours in 'Shiny' Apps Using 'Shepherd.js'
- d3po: Fast and Beautiful Interactive Visualization for 'Markdown' and 'Shiny'
- DataVisualizations: Visualizations of High-Dimensional Data
- descriptr: Generate Descriptive Statistics
- DT: A Wrapper of the JavaScript Library 'DataTables'
- echarty: Minimal R/Shiny Interface to JavaScript Library 'ECharts'
- esquisse: Explore and Visualize Your Data Interactively
- fmtr: Easily Apply Formats to Data
- ggalluvial: Alluvial Plots in 'ggplot2'
- GGally: Extension to 'ggplot2'
- gganimate: A Grammar of Animated Graphics
- ggbreak: Set Axis Break for 'ggplot2'
- ggcharts: Shorten the Distance from Data Visualization Idea to Actual Plot
- ggcorrplot: Visualization of a Correlation Matrix using 'ggplot2'
- ggcorset: The Corset Plot
- ggdag: Analyze and Create Elegant Directed Acyclic Graphs
- ggdist: Visualizations of Distributions and Uncertainty
- ggDoubleHeat: A Heatmap-Like Visualization Tool
- ggeffects: Create Tidy Data Frames of Marginal Effects for 'ggplot' from Model Outputs
- ggESDA: Exploratory Symbolic Data Analysis with 'ggplot2'

- ggfocus: Scales that Focus Specific Levels in your ggplot
- ggforce: Accelerating 'ggplot2'
- ggformula: Formula Interface to the Grammar of Graphics
- ggfortify: Data Visualization Tools for Statistical Analysis Results
- gghdr: Visualisation of Highest Density Regions in 'ggplot2'
- ggheatmap: Plot Heatmap
- gghighlight: Highlight Lines and Points in 'ggplot2'
- ggh4x: Hacks for 'ggplot2'
- ggiraph: Make 'ggplot2' Graphics Interactive
- ggmatplot: Plot Columns of Two Matrices Against Each Other Using 'ggplot2'
- ggmice: Visualizations for 'mice' with 'ggplot2'
- ggmosaic: Mosaic Plots in the 'ggplot2' Framework
- ggmulti: High Dimensional Data Visualization
- ggnetwork: Geometries to Plot Networks with 'ggplot2'
- ggpattern: 'ggplot2' Pattern Geoms
- ggpie: pie, donut and rose pie plots with ggplot2
- ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics
- ggplotify: Convert Plot to 'grob' or 'ggplot' Object
- ggpmisc: Miscellaneous Extensions to 'ggplot2'
- ggpubr: 'ggplot2' Based Publication Ready Plots
- ggpval: Annotate Statistical Tests for 'ggplot2'
- ggquickeda: Quickly Explore Your Data Using 'ggplot2' and 'table1' Summary Tables
- ggside extends 'ggplot2' by allowing users to add graphical information about one of the main panel's axis using a familiar 'ggplot2' style API with tidy data
- ggsignif: Significance Brackets for 'ggplot2'
- ggstance: Horizontal 'ggplot2' Components
- ggstar: Multiple Geometric Shape Point Layer for 'ggplot2'
- ggstatsplot: 'ggplot2' Based Plots with Statistical Details
- ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'
- ggtrace: Provides ggplot2 geoms that allow groups of data points to be outlined or highlighted for emphasis
- gluedown: Wrap Vectors in Markdown Formatting
- gridstackeR: easy way to create responsive layouts with just a few lines of code using gridstack.js
- gt: Easily Create Presentation-Ready Display Tables
- gtExtras: additional functions for creating tables with gt

- gtsummary: Presentation-Ready Data Summary and Analytic Result Tables
- heatmaply: Interactive Cluster Heat Maps Using 'plotly' and 'ggplot2'
- heplots: Visualizing Hypothesis Tests in Multivariate Linear Models
- htmlTable: Advanced Tables for Markdown/HTML
- huxtable: Easily Create and Style Tables for LaTeX, HTML and Other Formats
- jjAnno: An Annotation Package for 'ggplot2' Output
- kableExtra: Construct Complex Table with 'kable' and Pipe Syntax
- listdown: Create R Markdown from Lists
- loon: Interactive Statistical Data Visualization
- loon.ggplot: A Grammar of Interactive Graphics
- magick: Advanced Graphics and Image-Processing in R
- memoiR: R Markdown and Bookdown Templates to Publish Documents
- ndtv: Network Dynamic Temporal Visualizations
- numform: Tools to Format Numbers for Publication
- performance: Assessment of Regression Models Performance
- plot.matrix: Visualizes a Matrix as Heatmap
- presenter: Present Data with Style
- prompter: Add Tooltips in 'Shiny' Apps with 'Hint.css'
- psre: Presenting Statistical Results Effectively
- quarto: R Interface to 'Quarto' Markdown Publishing System
- r2resize: In-Text Resizing for Containers, Images and Data Tables in 'Shiny', 'Markdown' and 'Quarto' Documents
- r3js: allow WebGL-based 3D plotting using the three.js library
- reactR: Make it easy to use 'React' in R with 'htmlwidget' scaffolds
- reporter: Creates Statistical Reports
- rheroicons: A Zero Dependency 'SVG' Icon Library for 'Shiny'
- rhino: A Framework for Enterprise Shiny Applications
- rintrojs: Wrapper for the 'Intro.js' Library
- rmarkdown: Dynamic Documents for R
- rsvg: Render SVG Images into PDF, PNG, (Encapsulated) PostScript, or Bitmap Arrays
- semantic.dashboard: Dashboard with Fomantic UI Support for Shiny
- shapviz: visualize SHapley Additive exPlanations (SHAP) - waterfall, force, importance, dependence plots
- shiny: Web Application Framework for R
- shinyChakraUI: A Wrapper of the 'React' Library 'Chakra UI' for 'Shiny'

- shinydlplot: Add a Download Button to a 'shiny' Plot or 'plotly'
- shinyHugePlot: Efficient Plotting of Large-Sized Data
- shinyMobile: Mobile Ready 'shiny' Apps with Standalone Capabilities
- shinySelect: A Wrapper of the 'react-select' Library
- shiny.semantic: Semantic UI Support for Shiny
- shinytest: Test Shiny Apps
- shinyWidgets: Custom Inputs Widgets for Shiny
- starry: Explore Data with Plots and Tables
- statsExpressions: Tidy Dataframes and Expressions with Statistical Details
- sugrrants: Supporting Graphs for Analysing Time Series
- tidybayes: Tidy Data and 'Geoms' for Bayesian Models
- tidycharts: Generate Tidy Charts Inspired by 'IBCS'
- tidyHeatmap: A Tidy Implementation of Heatmap
- tourr: Tour Methods for Multivariate Data Visualisation
- tornado: Plots for Model Sensitivity and Variable Importance
- trelliscopejs: Create Interactive Trelliscope Displays
- tsviz: Easy and Interactive Time Series Visualization
- UpSetR: A More Scalable Alternative to Venn and Euler Diagrams for Visualizing Intersecting Sets
- visNetwork: Network Visualization using 'vis.js' Library
- visStatistics: Automated Visualization of Statistical Tests
- vtable: Variable Table for Variable Documentation
- xaringan: Presentation Ninja
- yardstick: Tidy Characterizations of Model Performance

8 Codes for QWIM (Quantitative Wealth and Investment Management)

8.1 Collections of resources

List of links:

- Curated list of practical financial machine learning tools and applications
- EliteQuant: online resources for quantitative modeling, trading, portfolio management

8.2 Research studies with code

Ardia et al. (“RiskPortfolios: Computation of Risk-Based Portfolios in R,” 2017)

Boileau et al. (“cvCovEst: Cross-validated covariance matrix estimator selection and evaluation in R,” 2021)

Brugiere (*Quantitative Portfolio Management with Applications in Python*, 2020)

Bryzgalova et al. (“Forest Through the Trees: Building Cross-Sections of Stock Returns,” 2021)

Cajas (“Entropic Portfolio Optimization: a Disciplined Convex Programming Framework,” 2021)

Cajas (“OWA Portfolio Optimization: a Disciplined Convex Programming Framework,” 2021)

Chen and Zimmermann (“Open Source Cross-Sectional Asset Pricing,” 2022)

Chib (*R package czfactor*, 2020)

Chib and Zhao (*R package czzg*, 2020)

Coqueret and Guida (*Machine Learning for Factor Investing: R Version*, 2020)

Coqueret (*Perspectives in sustainable equity investing (website version)*, 2022)

de Carvalho and Rua (“Real-time nowcasting the US output gap: Singular spectrum analysis at work,” 2017)

Ding et al. (“A Python package for multi-stage stochastic programming,” 2020)

Dixon et al. (*Machine Learning in Finance: from theory to practice*, 2020)

Dixon and Polson (“Deep Fundamental Factor Models,” 2020)

Dong et al. (“Anomalies and the expected market return,” 2022)

Guijarro-Ordonez et al. (“Deep Learning Statistical Arbitrage,” 2021)

Gurdogan and Kercheval (“Multi Anchor Point Shrinkage for the Sample Covariance Matrix (Extended Version),” 2021)

Ho et al. (“Moving beyond P values: data analysis with estimation graphics,” 2019)

Irlam (“Multi Scenario Financial Planning via Deep Reinforcement Learning AI,” 2020)

Irlam (*AI Planner*, 2020)

Irlam (“Machine learning for retirement planning,” 2020)

Jansen (*Machine Learning for Algorithmic Trading (Second Edition)*, 2020)

Kakushadze and Yu (“Statistical Risk Models,” 2016)

Kakushadze and Yu (“Open Source Fundamental Industry Classification,” 2017)

Kakushadze and Yu (“Betas, Benchmarks, and Beating the Market,” 2018)

Kakushadze and Yu (“Decoding stock market with quant alphas,” 2018)

Kakushadze and Yu (“Machine learning risk models,” 2019)

Kakushadze and Yu (“Machine learning treasury yields,” 2020)

Lai et al. (“TODS: An Automated Time Series Outlier Detection System,” 2021)

Lettau and Pelger (“Factors That Fit the Time Series and Cross-Section of Stock Returns,” 2020)

Li et al. (“FinRL-Podracer: High Performance and Scalable Deep Reinforcement Learning for Quantitative Finance,” 2021)

Liu et al. (“FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance,” 2021)

Liu et al. (“FinRL-Meta: A Universe of Near-Real Market Environments for Data-Driven Deep Reinforcement Learning in Quantitative Finance,” 2022)

Marinescu (“Risk-Based Optimal Portfolio Strategies: A Compendium,” 2022)

Martin (“PyPortfolioOpt: portfolio optimization in Python,” 2021)

Marwood and Minnen (“Safely Boosting Retirement Income by Harmonizing Drawdown Paths,” 2020)

McIndoe (“A Data Driven Approach to Market Regime Classification,” 2020)

Micheli and Neuman (“Evidence of Crowding on Russell 3000 Reconstitution Events,” 2022)

Milevsky (*Retirement Income Recipes in R: From Ruin Probabilities to Intelligent Drawdowns*, 2020)

Qian et al. (“Combining forecasts for universally optimal performance,” 2022)

Rao and Jelvis (*Foundations of Reinforcement Learning with Applications in Finance*, 2022)

Sarmas et al. (*Multicriteria Portfolio Construction with Python*, 2020)

Sharma et al. (“DoWhy: Addressing Challenges in Expressing and Validating Causal Assumptions,” 2021)

Shi et al. (“Deep Learning Algorithms for Hedging with Frictions,” 2022)

Siebert et al. (“A systematic review of Python packages for time series analysis,” 2021)

Simos et al. (“Time-varying Black–Litterman portfolio optimization using a bio-inspired approach and neuronets,” 2021)

Snow (“Machine learning in asset management,” 2019)

Snow (“Machine Learning in Asset Management Part 1: Portfolio Construction Trading Strategies,” 2020)

Snow (“Machine Learning in Asset Management - Part 2: Portfolio Construction - Weight Optimization,” 2020)

Tatsat et al. (*Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*, 2020)

Tuck et al. (“Portfolio Construction Using Stratified Models,” 2022)

Ungolo et al. (“affine_mortality: A Github repository for estimation, analysis, and projection of affine mortality models,” 2021)

Vamossy and Skog (“EmTract: Investor Emotions and Market Behavior,” 2021)

Vinod (“R Package GeneralCorr Functions for Portfolio Choice,” 2021)

Yang et al. (“FinBERT: A Pretrained Language Model for Financial Communications,” 2020)

Yu et al. (“An AI approach to measuring financial risk,” 2020)

8.3 Python software implementations

List of packages and/or codes and/or frameworks and/or links:

- alive-progress: new kind of Progress Bar, with real-time throughput, ETA, and very cool animations
- alphalens: Performance analysis of predictive (alpha) stock factors
- AlphaPy: Automated Machine Learning [AutoML] with Python, scikit-learn, Keras, XGBoost, LightGBM, and CatBoost
- auquantoolbox: Backtesting toolbox for trading strategies
- azapy: Financial Portfolio Optimization Algorithms
- bt: flexible backtesting framework
- btgym: Scalable, event-driven, deep-learning-friendly backtesting library
- Clairvoyant: identify and monitor social/historical cues for short term stock movement
- CVXPortfolio: Portfolio optimization and simulation
- cyanure: Toolbox for Empirical Risk Minimization
- Elegant-FinRL: algorithmic strategies using Reinforcement Learning
- Empyrial: AI and data-driven quantitative portfolio management for risk and performance analytics
- empyrical: Common financial risk and performance metrics
- EmTract: Extracting Emotions from Social Media Text Tailored for Financial Contexts
- ffn: Financial functions for Python
- FinDataPy: download market data via Bloomberg, Eikon, Quandl, Yahoo etc.
- FinMarketPy: backtesting trading strategies and analyzing financial markets
- finnhub-python: Finnhub Python API Client to provide financial data(real-time stock price, global fundamentals, global ETFs holdings and alternative data)
- FinRL: Deep Reinforcement Learning for Quantitative Finance
- FinRL-Meta: Universe of Near-Real Market Environments for Data-Driven Financial Reinforcement Learning
- fredapi is a Python interface to the Federal Reserve Economic Data (FRED) and ALFRED databases
- lifelib: Actuarial models in Python
- lifelines: Survival analysis in Python, including Kaplan Meier, Nelson Aalen and regression

- lrsm_portfolio: Portfolio Construction using Stratified Models
- Machine Learning and Data Science Blueprints for Finance (codes for the book)
- Machine Learning fior asset management
- Machine Learning for Algorithmic Trading (codes for the book)
- MLFinLab: Machine Learning Financial Laboratory
- momentum: Running mean, variance, skew, and kurtosis
- Multicriteria Portfolio Construction with Python
- okama: investment portfolio analyzing and optimization tools
- OpenBBTerminal: modern Python-based integrated environment for investment research
- OptimalPortfolio: portfolio optimization
- QuantAxis: Quantitative Financial FrameWork
- QuantEcon: quantitative economics
- Pandas TA: Technical Analysis Indicators
- portfolio-backtest: backtest portfolio asset allocation
- precise: online covariance and precision forecasting, portfolios, and model ensembles
- predictionrevisited: implements the core statistical concepts from the book ”Prediction Revisited: The Importance of Observation”
- pyfinance: general financial and security returns analysis
- pyfolio: Portfolio and risk analytics in Python
- pyhrp: hierarchical risk parity algorithms
- PyPortfolioOpt: Financial portfolio optimisation
- Qlib: Microsoft AI-oriented quantitative investment platform
- QuantEcon.py: quantitative economics
- QuantLib: Python bindings for the QuantLib library
- QuantResearch: Quantitative analysis, strategies and backtests
- Quantropy: Financial pipeline for the data-driven investor to research, develop and deploy robust strategie
- QuantStats: Portfolio analytics for quants
- Riskfolio-Lib: Portfolio Optimization and Quantitative Strategic Asset Allocation
- Robust Risk-aware reinforcement learning
- stocksight: Stock market analyzer and predictor using Elasticsearch, Twitter, News headlines, NLP and sentiment analysis
- ta: Technical Analysis Library using Pandas and Numpy
- TA-lib: Python wrapper for TA-Lib Technical Analysis Library
- Tax-Calculator: USA Federal Individual Income and Payroll Tax Microsimulation Model
- tf-quant-finance: High-performance TensorFlow library by Google for quantitative finance.
- vectorbt: Supercharged backtesting and technical analysis for quants
- zipline: Algorithmic Trading Library

8.4 R software implementations

List of packages and/or codes and/or frameworks and/or links:

- ASSA: Applied Singular Spectrum Analysis
- AssetAllocation: Backtesting Simple Asset Allocation Strategies
- BEKKs: Multivariate Conditional Volatility Modelling and Forecasting
- crseEventStudy: A Robust and Powerful Test of Abnormal Stock Returns in Long-Horizon Event Studies
- DOSPortfolio: Dynamic Optimal Shrinkage Portfolio
- ExtremeRisks: Extreme Risk Measures
- FFdownload: Download Data from Kenneth French's Website
- FinnTS: Microsoft Finance Time Series Forecasting Framework
- finreportr: Financial Data from U.S. Securities and Exchange Commission
- fHMM: Fitting Hidden Markov Models to Financial Data
- FinnTS: Microsoft Finance Time Series Forecasting Framework
- fitHeavyTail: Mean and Covariance Matrix Estimation under Heavy Tails
- fixedincome: Fixed Income Models, Calculations, Data Structures and Instruments
- generalCorr: Generalized Correlations, Causal Paths and Portfolio Selection
- greeks: Sensitivities of Prices of Financial Options
- HDShOP: High-Dimensional Shrinkage Optimal Portfolios
- HierPortfolios: Hierarchical Clustering-Based Portfolio Allocation Strategies
- highOrderPortfolios: Design of High-Order Portfolios via Mean, Variance, Skewness, and Kurtosis
- imputeFin: Imputation of Financial Time Series with Missing Values and/or Outliers
- MarkowitzR: Statistical Significance of the Markowitz Portfolio
- MortCast: Estimation and Projection of Age-Specific Mortality Rates
- parma: Portfolio Allocation and Risk Management Applications
- pbo: Probability of Backtest Overfitting
- pec: Prediction Error Curves for Risk Prediction Models in Survival Analysis
- pedquant: Public Economic Data and Quantitative Analysis
- PerformanceAnalytics: Econometric Tools for Performance and Risk Analysis
- PortfolioAnalytics: Portfolio Analysis, Including Numerical Methods for Optimization of Portfolios
- portfolioBacktest: Automated Backtesting of Portfolios over Multiple Datasets
- portvine: portfolio level risk estimates using ARMA-GARCH and vine copulas
- priceR: Economics and Pricing Tools
- qlcal: R Bindings to the Calendaring Functionality of 'QuantLib'

- qrmtools: Tools for Quantitative Risk Management
- quantmod: Quantitative Financial Modelling Framework
- RcppQuantuccia: R Bindings to the Calendaring Functionality of 'QuantLib'
- riskParityPortfolio: Design of Risk Parity Portfolios
- RiskPortfolios: Computation of Risk-Based Portfolios
- riskRegression: Risk Regression Models and Prediction Scores for Survival Analysis with Competing Risks
- RPESE: Estimates of Standard Errors for Risk and Performance Measures
- RQuantLib: R Interface to the 'QuantLib' Library
- SharpeR: Statistical Significance of the Sharpe Ratio
- sharpeRratio: Moment-Free Estimation of Sharpe Ratios
- sparseIndexTracking: Design of Portfolio of Stocks to Track an Index
- SWIM: Scenario Weights for Importance Measurement
- TextForecast: Regression Analysis and Forecasting Using Textual Data from a Time-Varying Dictionary
- tidyquant: Tidy Quantitative Financial Analysis
- Trading: CCR, Advanced Correlation & Beta Estimates, Betting Strategies
- tseries: Time Series Analysis and Computational Finance
- ufRisk: Risk Measure Calculation in Financial TS
- usincometaxes: wrapper to the NBER's TAXSIM 35 tax simulator for federal and state income taxes
- ycevo: Nonparametric Estimation of the Yield Curve Evolution
- yfR: Downloads and Organizes Financial Data from Yahoo Finance

References

Abboud, A., Cohen-Addad, V., and Houdrouge, H. (2019). "Subquadratic High-Dimensional Hierarchical Clustering." In: *Advances in Neural Information Processing Systems 32 (NIPS 2019)*.

Ackerman, M., Ben-David, S., Brânzei, S., and Loker, D. (2021). "Weighted clustering: Towards solving the user's dilemma." In: *Pattern Recognition* 120, p. 108152.

Adcock, C., Areal, N., Armada, M. R., Cortez, M. C., Oliveira, B., and Silva, F. (2017). "Portfolio Performance Measurement: Monotonicity with Respect to the Sharpe Ratio and Multivariate Tests of Correlation." In: *SSRN e-Print*.

Adolfsson, A., Ackerman, M., and Brownstein, N. C. (2019). "To cluster, or not to cluster: An analysis of clusterability methods." In: *Pattern Recognition* 88, pp. 13–26.

Ahelegbey, D. F. and Giudici, P. (2020). "Market Risk, Connectedness and Turbulence: A Comparison of 21st Century Financial Crises." In: *SSRN e-Print*.

Akansu, A., Avellaneda, M., and Xiong, A. (2021). "Quant investing in cluster portfolios." In: *The Journal of Investment Strategies* 9(4), pp. 61–78.

Alfarra, M., Perez, J. C., Bibi, A., Thabet, A., Arbelaez, P., and Ghanem, B. (2021). "Rethinking Clustering for Robustness." In: *arXiv e-Print*.

Alokley, S. A. and Albarrak, M. S. (2020). "Clustering of Extremes in Financial Returns: A Study of Developed and Emerging Markets." In: *Journal of Risk and Financial Management* 13(7), p. 141.

Ardia, D., Boudt, K., and Gagnon-Fleury, J.-P. (2017). "RiskPortfolios: Computation of Risk-Based Portfolios in R." In: *The Journal of Open Source Software* 2(10), pp. 171+.

Arnott, R. D., Harvey, C. R., and Markowitz, H. (2019). "A backtesting protocol in the era of machine learning." In: *The Journal of Financial Data Science* 1(1), pp. 64–74.

Avellaneda, M. and Serur, J. A. (2020). "Hierarchical PCA and Modeling Asset Correlations." In: *arXiv e-Print*.

Bacon, C. R. (2019). "Performance Attribution: History and Progress." In: *CFA Institute Research Foundation Publications*.

Bailey, D. H., Borwein, J. M., and Lopez de Prado, M. (2017). "Stock Portfolio Design and Backtest Overfitting." In: *Journal of Investment Management* 15(1), pp. 75–87.

Baitinger, E. (2021). "Forecasting asset returns with network-based metrics: A statistical and economic analysis." In: *Journal of Forecasting*.

Baitinger, E. and Flegel, S. (2021). "New Concepts in Financial Forecasting: Network-Based Information, Topological Data Analysis and their Combination." In: *SSRN e-Print*.

Baitinger, E. and Maier, T. (2019). "The (Mis)Behavior of Hedge Fund Strategies: A Network-Based Analysis." In: *The Journal of Alternative Investments* 22 (1), pp. 57–74.

Baitinger, E. and Papenbrock, J. (2017). "Interconnectedness Risk and Active Portfolio Management." In: *Journal of Investment Strategies* 6(2), pp. 63–90.

Bandara, K., Bergmeir, C., and Smyl, S. (2020). "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach." In: *Expert Systems with Applications* 140, pp. 112896+.

Bardoscia, M., Barucca, P., Battiston, S., Caccioli, F., Cimini, G., Garlaschelli, D., Saracco, F., Squartini, T., and Caldarelli, G. (2021). "The Physics of Financial Networks." In: *arXiv e-Print*.

Barunik, J. and Ellington, M. (2021). "Dynamic Networks in Large Financial and Economic Systems." In: *arXiv e-Print*.

Begusic, S. and Kostanjcar, Z. (2019). "Cluster-Based Shrinkage of Correlation Matrices for Portfolio Optimization." In: *11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, pp. 301–305.

Bennett, S., Cucuringu, M., and Reinert, G. (2022). "Lead-lag detection and network clustering for multivariate time series with an application to the US equity market." In: *arXiv e-Print*.

Bessler, W., Opfer, H., and Wolff, D. (2017). "Multi-asset portfolio optimization and out-of-sample performance: an evaluation of Black Litterman, mean-variance, and naive diversification approaches." In: *The European Journal of Finance* 23(1), pp. 1–30.

Bessler, W. and Wolff, D. (2017). "Portfolio Optimization with Industry Return Prediction Models." In: *SSRN e-Print*.

Bjerring, T., Ross, O., and Weissensteiner, A. (2017). "Feature selection for portfolio optimization." In: *Annals of Operations Research* 256, pp. 21–40.

Bnouachir, N. and Mkhadri, A. (2021). "Efficient cluster-based portfolio optimization." In: *Communications in Statistics - Simulation and Computation* 50, pp. 3241–3255.

Boileau, P., Hejazi, N., Collica, B., Laan, M. van der, and Dudoit, S. (2021). "cvCovEst: Cross-validated covariance matrix estimator selection and evaluation in R." In: *Journal of Open Source Software* 6(63), p. 3273.

Bonald, T., Lara, N. de, Lutz, Q., and Charpentier, B. (2020). "Scikit-network: Graph Analysis in Python." In: *Journal of Machine Learning Research* 21(185), pp. 1–6.

Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge University Press. 446 pp.

Brécheteau, C., Fischer, A., and Levrard, C. (2021). "Robust Bregman clustering." In: *Annals of Statistics* 49(3).

Brugiere, P. (2020). *Quantitative Portfolio Management with Applications in Python*. Springer International Publishing. 189 pp.

Bruni, R., Cesarone, F., Scozzari, A., and Tardella, F. (2016). "Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models." In: *Data in Brief* 8, pp. 858–862.

Bruni, R., Cesarone, F., Scozzari, A., and Tardella, F. (2017). "On exact and approximate stochastic dominance strategies for portfolio selection." In: *European Journal of Operational Research* 259(1), pp. 322–329.

Bryzgalova, S., Huang, J., and Julliard, C. (2021a). "Bayesian solutions for the factor zoo: we just ran two quadrillion models." In: *SSRN e-Print*.

Bryzgalova, S., Pelger, M., and Zhu, J. (2021b). "Forest Through the Trees: Building Cross-Sections of Stock Returns." In: *SSRN e-Print*.

Cai, F., Le-Khac, N.-A., and Kechadi, T. (2016). "Clustering Approaches for Financial Data Analysis: a Survey." In: *arXiv e-Print*.

Cajas, D. (2019). "Robust Portfolio Selection with Near Optimal Centering." In: *SSRN e-Print*.

Cajas, D. (2021a). "Entropic Portfolio Optimization: a Disciplined Convex Programming Framework." In: *SSRN e-Print*.

Cajas, D. (2021b). "OWA Portfolio Optimization: a Disciplined Convex Programming Framework." In: *SSRN e-Print*.

Campello, R. J. G. B., Kroger, P., Sander, J., and Zimek, A. (2020). "Density-based clustering." In: *WIREs Data Mining and Knowledge Discovery* 10(2).

Casa, A., Scrucca, L., and Menardi, G. (2021). "Better than the best? Answers via model ensemble in density-based clustering." In: *Advances in Data Analysis and Classification* 15(3), pp. 599–623.

Castilho, D., Souza, T. T. P., Kang, S. M., Gama, J., and Carvalho, A. C. P. L. F. de (2021). "Forecasting Financial Market Structure from Network Features using Machine Learning." In: *arXiv e-Print*.

Cesarone, F., Moretti, J., and Tardella, F. (2018). "Why Small Portfolios Are Preferable and How to Choose Them." In: *SSRN e-Print*.

Cesarone, F., Mottura, C., Ricci, J. M., and Tardella, F. (2019). "On the stability of portfolio selection models." In: *SSRN e-Print*.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set." In: *Journal of Statistical Software*.

Chaudhuri, S. E. and Lo, A. W. (2019). "Dynamic Alpha: A Spectral Decomposition of Investment Performance Across Time Horizons." In: *Management Science* 65(9), pp. 4440–4450.

Chavent, M., Genuer, R., and Saracco, J. (2021). "Combining clustering of variables and feature selection using random forests." In: *Communications in Statistics - Simulation and Computation* 50(2), pp. 426–445.

Chehreghani, M. H. (2021). "Shift of Pairwise Similarities for Data Clustering." In: *arXiv e-Print*.

Chen, A. Y. and Zimmermann, T. (2022). "Open Source Cross-Sectional Asset Pricing." In: *American Finance Association Annual Meeting*.

Chen, W., Jiang, M., and Jiang, C. (2020). "Constructing a multilayer network for stock market." In: *Soft computing* 24, pp. 6345–6361.

Chib, S. (2020). *R package czfactor*. Tech. rep. Washington University.

Chib, S. and Zhao, L. (2020). *R package czzg*. Tech. rep. Washington University.

Chua, D. B., Kritzman, M., and Page, S. (2009). "The Myth of Diversification." In: *The Journal of Portfolio Management* 36(1), pp. 26–35.

Chung, J., Bridgeford, E., Arroyo, J., Pedigo, B. D., Saad-Eldin, A., Gopalakrishnan, V., Xiang, L., Priebe, C. E., and Vogelstein, J. T. (2022). "Statistical Connectomics." In: *Annual Review of Statistics and Its Application* 8(1), pp. 463–492.

Clemente, G. P., Grassi, R., and Hitaj, A. (2019). "Smart network based portfolios." In: *arXiv e-Print*.

Clemente, G. P., Grassi, R., and Hitaj, A. (2021). "Asset allocation: new evidence through network approaches." In: *Annals of Operations Research* 299, pp. 61–80.

Coqueret, G. (2022). *Perspectives in sustainable equity investing (website version)*.

Coqueret, G. and Guida, T. (2020). *Machine Learning for Factor Investing: R Version*. Chapman and Hall/CRC. 341 pp.

Coraggio, L. and Coretto, P. (2021). "Selecting the number of clusters, clustering models, and algorithms. A unifying approach based on the quadratic discriminant score." In: *arXiv e-Print*.

Custodio João, I., Lucas, A., and Schaumburg, J. (2021). "Clustering Dynamics and Persistence for Financial Multivariate Panel Data." In: *SSRN e-Print*.

Dalmia, A. and Sia, S. (2021). "Clustering with UMAP: Why and How Connectivity Matters." In: *arXiv e-Print*.

de Carvalho, M. and Rua, A. (2017). "Real-time nowcasting the US output gap: Singular spectrum analysis at work." In: *International Journal of Forecasting* 33(1), pp. 185–198.

de Carvalho, P. J. C. and Gupta, A. (2018). "A network approach to unravel asset price comovement using minimal dependence structure." In: *Journal of Banking & Finance* 91, pp. 119–132.

De Luca, G. and Zuccolotto, P. (2021). "Hierarchical time series clustering on tail dependence with linkage based on a multivariate copula approach." In: *International Journal of Approximate Reasoning* 139, pp. 88–103.

de Miranda Cardoso, J. V., Ying, J., and Palomar, D. P. (2020). "Algorithms for Learning Graphs in Financial Markets." In: *arXiv e-Print*.

Dees, B. S., Stankovic, L., Constantinides, A. G., and Mandic, D. P. (2020). "Portfolio Cuts: A Graph-Theoretic Framework to Diversification." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Den Teuling, N., Pauws, S., and Heuvel, E. van den (2021). "Clustering of longitudinal data: A tutorial on a variety of approaches." In: *arXiv e-Print*.

Dey, A. K., Tian, Y., and Gel, Y. R. (2021). "Community detection in complex networks: From statistical foundations to data science applications." In: *WIREs Computational Statistics*.

Di Cerbo, L. F. and Taylor, S. (2021). "Graph theoretical representations of equity indices and their centrality measures." In: *Quantitative Finance* 21(4), pp. 523–537.

Ding, L., Ahmed, S., and Shapiro, A. (2020). "A Python package for multi-stage stochastic programming." In: *Optimization Online e-Print*.

Diris, B., Palm, F., and Schotman, P. (2015). "Long-Term Strategic Asset Allocation: An Out-of-Sample Evaluation." In: *Management Science* 61(9), pp. 2185–2202.

Dixon, M. and Polson, N. (2020). "Deep Fundamental Factor Models." In: *SIAM Journal on Financial Mathematics* 11(3), SC–26–SC–37.

Dixon, M. F., Halperin, I., and Bilokon, P. (2020). *Machine Learning in Finance: from theory to practice*. Springer International Publishing. 548 pp.

Djouzi, K. and Beghdad-Bey, K. (2019). "A Review of Clustering Algorithms for Big Data." In: *International Conference on Networking and Advanced Systems (ICNAS)*. IEEE.

Dong, X., Li, Y., Rapach, D., and Zhou, G. (2022). "Anomalies and the expected market return." In: *Journal of Finance* 27(1), pp. 639–681.

Doreian, P., Batagelj, V., and Ferligoj, A. (2020). *Advances in Network Clustering and Blockmodeling*. Wiley. 432 pp.

Duan, J. (2021). "Predicting with Structured Data: Graphs, Ranks, and Time Series." PhD thesis. Kyoto University.

Duan, L. L. and Dunson, D. B. (2021). "Bayesian Distance Clustering." In: *Journal of Machine Learning Research* 22(224), pp. 1–27.

Duarte, F. G. and De Castro, L. N. (2020). "A Framework to Perform Asset Allocation Based on Partitional Clustering." In: *IEEE Access* 8, pp. 110775–110788.

Dugué, N., Lamirel, J.-C., and Chen, Y. (2021). "Evaluating clustering quality using features salience: a promising approach." In: *Neural Computing and Applications* 33(19), pp. 12939–12956.

Eidenvall, A. (2021). "Hierarchical Clustering To Improve Portfolio Tail Risk Characteristics." MA thesis. Lund University.

Elliott, A., Chiu, A., Bazzi, M., Reinert, G., and Cucuringu, M. (2020). "Core–periphery structure in directed networks." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476(2241), p. 20190783.

Emerson, S. (2019). "Machine learning for financial applications: self-organising maps, hierarchical clustering and dynamic time-warping for portfolio constructive." PhD thesis. University College Cork.

Escanciano, J. C. and Hualde, J. (2021). "Measuring Asset Market Linkages: Nonlinear Dependence and Tail Risk." In: *Journal of Business & Economic Statistics* 39(2), pp. 453–465.

Esmaeili, S. A., Brubach, B., Tsepenekas, L., and Dickerson, J. P. (2021). "Probabilistic Fair Clustering." In: *arXiv e-Print*.

Exarchakis, G., Oubari, O., and Lenz, G. (2022). "A sampling-based approach for efficient clustering in large datasets." In: *arXiv e-Print*.

Ezhilmaran, D. and Indira, D. V. (2020). "A survey on clustering techniques in pattern recognition." In: *AIP Conference Proceedings*. AIP Publishing.

Fabozzi, F. J., Fabozzi, F. A., Lopez de Prado, M., and Stoyanov, S. (2021). *Asset Management: Tools and Issues*. World Scientific. 516 pp.

Fabozzi, F. J. and Lopez de Prado, M. (2018). "Being Honest in Backtest Reporting: A Template for Disclosing Multiple Tests." In: *The Journal of Portfolio Management* 45(1), pp. 141–147.

Faletto, G. and Bien, J. (2022). "Cluster Stability Selection." In: *arXiv e-Print*.

Feng, J. and Simon, N. (2020). "Ensembled sparse-input hierarchical networks for high-dimensional datasets." In: *arXiv e-Print*.

Ferraro, M. B., Giordani, P., and Serafini, A. (2019). "fclust: An R Package for Fuzzy Clustering." In: *The R Journal*.

Fischer, D., Berro, A., Nordhausen, K., and Ruiz-Gazen, A. (2021). "REPPlab: An R package for detecting clusters and outliers using exploratory projection pursuit." In: *Communications in Statistics - Simulation and Computation*.

Flint, E., Seymour, A., and Chikurunhe, F. (2021). "Defining and measuring portfolio diversification." In: *South African Actuarial Journal* 20(1), pp. 17–48.

Fop, M. and Murphy, T. B. (2017). "Variable Selection Methods for Model-based Clustering." In: *arXiv e-Print*.

Franti, P. and Sieranoja, S. (2019). "How much can k-means be improved by using better initialization and repeats?" In: *Pattern Recognition* 93, pp. 95–112.

Fu, W. and Perry, P. O. (2020). "Estimating the Number of Clusters Using Cross-Validation." In: *Journal of Computational and Graphical Statistics* 29(1), pp. 162–173.

Fusai, G., Mignacca, D., Nardon, A., and Human, B. (2020). "Equally Diversified or Equally Weighted?" In: *Risk (Cutting Edge)*.

Gagolewski, M. (2021). "genieclust: Fast and robust hierarchical clustering." In: *SoftwareX* 15, p. 100722.

Gao, Z. and Tsay, R. S. (2021). "Divide-and-Conquer: A Distributed Hierarchical Factor Approach to Modeling Large-Scale Time Series Data." In: *arXiv e-Print*.

Garvey, G. and Madhavan, A. (2019). "Reconstructing Emerging and Developed Markets Using Hierarchical Clustering." In: *The Journal of Financial Data Science* 1 (4), pp. 84–102.

Gherbaoui, R., Ouali, M., and Benamrane, N. (2021). "Generation of Gaussian sets for clustering methods assessment." In: *Data & Knowledge Engineering* 131-132, p. 101876.

Ghosal, A., Nandy, A., Das, A. K., Goswami, S., and Panday, M. (2019). "A Short Review on Different Clustering Techniques and Their Applications." In: *Advances in Intelligent Systems and Computing*. Springer Singapore, pp. 69–83.

Giudici, P., Pagnottoni, P., and Polinesi, G. (2020). "Network Models to Enhance Automated Cryptocurrency Portfolio Management." In: *Frontiers in Artificial Intelligence* 3.

Giudici, P., Polinesi, G., and Spelta, A. (2022). "Network models to improve robot advisory portfolios." In: *Annals of Operations Research* 313, pp. 965–989.

Goel, A. and Majumdar, A. (2021). "Transformed K-means Clustering." In: *arXiv e-Print*.

Grealish, A. and Kolm, P. N. (2021). "Robo-Advisory: From Investing Principles and Algorithms to Future Developments." In: *SSRN e-Print*.

Greiner, S. P. and Stoyanov, S. V. (2019). "Portfolio scoring by expected risk premium." In: *The Journal of Portfolio Management* 45(4), pp. 83–90.

Grun, B. (2018). "Model-based Clustering." In: *arXiv e-Print*.

Guan, S. and Loew, M. (2021). "A Distance-based Separability Measure for Internal Cluster Validation." In: *arXiv e-Print*.

Gubu, L., Rosadi, D., and Abdurakhman (2021). "Robust mean-variance portfolio selection with time series clustering." In: *AIP Conference Proceedings*. AIP Publishing.

Guidolin, M., Hansen, E., and Lozano-Banda, M. (2018). "Portfolio performance of linear SDF models: an out-of-sample assessment." In: *Quantitative Finance* 18(8), pp. 1425–1436.

Guijarro-Ordonez, J., Pelger, M., and Zanotti, G. (2021). "Deep Learning Statistical Arbitrage." In: *SSRN e-Print*.

Guijo-Rubio, D., Duran-Rosal, A. M., Gutierrez, P. A., Troncoso, A., and Hervas-Martinez, C. (2020). "Time-Series Clustering Based on the Characterization of Segment Typologies." In: *IEEE Transactions on Cybernetics*.

Guo, D. (2019). "A Statistical Response to Challenges in Vast Portfolio Selection." PhD thesis. University of Waterloo.

Guo, D., Boyle, P. P., Weng, C., and Wirjanto, T. S. (2019). "When Does The 1/N Rule Work?" In: *SSRN e-Print*.

Guo, L., Hardle, W. K., and Tao, Y. (2021). "A Time-Varying Network for Cryptocurrencies." In: *arXiv e-Print*.

Gupta, K. and Chatterjee, N. (2018). "Financial Time Series Clustering." In: *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2*. Ed. by S. C. Satapathy and A. Joshi. Vol. 84. Smart Innovation, Systems and Technologies. Springer International Publishing, pp. 146–156.

Gurdogan, H. and Kercheval, A. (2021). "Multi Anchor Point Shrinkage for the Sample Covariance Matrix (Extended Version)." In: *arXiv e-Print*.

Haddad, M. and Bouguessa, M. (2021). "TopoDetect: Framework for Topological Features Detection in Graph Embeddings." In: *arXiv e-Print*.

Haley, M. R. (2017). "K-fold cross validation performance comparisons of six naive portfolio selection rules: how naive can you be and still have successful out-of-sample portfolio performance?" In: *Annals of Finance* 13, pp. 341–353.

Han, J. and Ge, Z. (2020). "Effect of dimensionality reduction on stock selection with cluster analysis in different market situations." In: *Expert systems with applications* 147 (113226).

Harvey, C. R., Liu, Y., and Saretto, A. (2020). "An Evaluation of Alternative Multiple Testing Methods for Finance Applications." In: *The Review of Asset Pricing Studies* 10(2), pp. 199–248.

Heckens, A. J. and Guhr, T. (2022). "New collectivity measures for financial covariances and correlations." In: *Physica A: Statistical Mechanics and its Applications* 604, p. 127704.

Hens, T., Schenk-Hoppe, K. R., and Woesthoff, M.-H. (2020). "Escaping the backtesting illusion." In: *The Journal of Portfolio Management* 46(4), pp. 81–93.

Herteliu, C., Levantesi, S., and Rotundo, G. (2021). "Network analysis of pension funds investments." In: *Physica A: Statistical Mechanics and its Applications* 579, p. 126139.

Ho, J., Tumkaya, T., Aryal, S., Choi, H., and Claridge-Chang, A. (2019). "Moving beyond P values: data analysis with estimation graphics." In: *Nature Methods* 16(7), pp. 565–566.

Homescu, C. (2014). "Many risks, one (optimal) portfolio." In: *SSRN e-Print*.

Homescu, C. (2015). "Better Investing Through Factors, Regimes and Sensitivity Analysis." In: *SSRN e-Print*.

Horvath, B., Issa, Z., and Muguruza, A. (2021). "Clustering Market Regimes Using the Wasserstein Distance." In: *SSRN e-Print*.

Hsu, Y.-C., Lin, H.-W., and Vincent, K. (2017). *Do Cross-Sectional Stock Return Predictors Pass the Test without Data-Snooping Bias?* Tech. rep. Institute of Economics Academia Sinica.

Hsu, P.-H., Han, Q., Wu, W., and Cao, Z. (2018). "Asset allocation strategies, data snooping, and the 1 / N rule." In: *Journal of Banking & Finance* 97, pp. 257–269.

Hua, K. (2019). "Clusterability, Model Selection and Evaluation." PhD thesis. University of Massachusetts Boston.

Huang, Q.-A., Zhao, J.-C., and Wu, X.-Q. (2022). "Financial risk propagation between Chinese and American stock markets based on multilayer networks." In: *Physica A: Statistical Mechanics and its Applications* 586, p. 126445.

Huang, M. and Yu, S. (2020). "A new procedure for resampled portfolio with shrinkaged covariance matrix." In: *Journal of Applied Statistics* 47(44), pp. 642–652.

Huang, X., Cui, P., Dong, Y., Li, J., Liu, H., Pei, J., Song, L., Tang, J., Wang, F., Yang, H., and Zhu, W. (2019). "Learning From Networks: Algorithms, Theory, and Applications." In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

Hwang, I., Xu, S., and In, F. (2018). "Naive versus optimal diversification: Tail risk and performance." In: *European Journal of Operational Research* 265(1), pp. 372–388.

Ielpo, F., Merhy, C., and Simon, G. (2017). *Engineering Investment Process: Making Value Creation Repeatable*. Elsevier. 430 pp.

Irani, J., Pise, N., and Phatak, M. (2016). "Clustering Techniques and the Similarity Measures used in Clustering: A Survey." In: *International Journal of Computer Applications* 134(7), pp. 9–14.

Irlam, G. (2020a). *AI Planner*. URL: `https://www.aiplanner.com/`.

Irlam, G. (2020b). "Machine learning for retirement planning." In: *The Journal of Retirement* 8(1), pp. 32–29.

Irlam, G. (2020c). "Multi Scenario Financial Planning via Deep Reinforcement Learning AI." In: *SSRN e-Print*.

Jackson, M. O. and Pernoud, A. (2020). "Systemic Risk in Financial Networks: A Survey." In: *SSRN e-Print*.

Jaeger, M., Krugel, S., Marinelli, D., Papenbrock, J., and Schwendner, P. (2020). "Understanding machine learning for diversified portfolio construction by explainable AI." In: *SSRN e-Print*.

Jaeger, M., Krugel, S., Marinelli, D., Papenbrock, J., and Schwendner, P. (2021a). "Interpretable Machine Learning for Diversified Portfolio Construction." In: *The Journal of Financial Data Science* 3(3), pp. 31–51.

Jaeger, M., Krugel, S., Papenbrock, J., and Schwendner, P. (2021b). "Adaptive Seriational Risk Parity and other Extensions for Heuristic Portfolio Construction using Machine Learning and Graph Theory." In: *SSRN e-Print*.

Jain, P. and Jain, S. (2019). "Can Machine Learning-Based Portfolios Outperform Traditional Risk-Based Portfolios? The Need to Account for Covariance Misspecification." In: *Risks* 7(3), pp. 74+.

Jansen, S. (2020). *Machine Learning for Algorithmic Trading (Second Edition)*. Packt Publishing. 820 pp.

Javed, A., Lee, B. S., and Rizzo, D. M. (2020). "A Benchmark Study on Time Series Clustering." In: *arXiv e-Print*.

Jiang, W., Xu, Q., and Zhang, R. (2022). "Tail-event driven network of cryptocurrencies and conventional assets." In: *Finance Research Letters* 46 (Part B) (102424).

Jose-Garcia, A. and Gomez-Flores, W. (2021). "A survey of cluster validity indices for automatic data clustering using differential evolution." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM.

Jurczenko et al. (2020). *Machine Learning for Asset Management*. Ed. by E. Jurczenko. Wiley. 445 pp.

Kakushadze, Z. and Yu, W. (2016). "Statistical Risk Models." In: *SSRN e-Print*.

Kakushadze, Z. and Yu, W. (2017). "Open Source Fundamental Industry Classification." In: *MDPI Data* 22 (2).

Kakushadze, Z. and Yu, W. (2018a). "Betas, Benchmarks, and Beating the Market." In: *The Journal of Trading*.

Kakushadze, Z. and Yu, W. (2018b). "Decoding stock market with quant alphas." In: *Journal of Asset Management*, pp. 1–11.

Kakushadze, Z. and Yu, W. (2019). "Machine learning risk models." In: *SSRN e-Print*.

Kakushadze, Z. and Yu, W. (2020). "Machine learning treasury yields." In: *SSRN e-Print*.

Kalyagin, V. A., Koldanov, A. P., and Koldanov, P. A. (2021). "Reliability of MST identification in correlation-based market networks." In: *arXiv e-Print*.

Katsouris, C. (2021). "Optimal Portfolio Choice and Stock Centrality for Tail Risk Events." In: *arXiv e-Print*.

Kawamoto, T. and Kabashima, Y. (2017a). "Comparative analysis on the selection of number of clusters in community detection." In: *arXiv e-Print*.

Kawamoto, T. and Kabashima, Y. (2017b). "Cross-validation estimate of the number of clusters in a network." In: *Scientific Reports* 7(1).

Kaya, H. (2015). "Eccentricity in Asset Management." In: *Journal of Network Theory in Finance* 1(1), pp. 1–32.

Kaya, H. (2017). "Managing ambiguity in asset allocation." In: *Journal of Asset Management* 18(3), pp. 163–187.

Kazak, E. and Pohlmeier, W. (2019). "Testing out-of-sample portfolio performance." In: *International Journal of Forecasting* 35(2), pp. 540–554.

Kazak, E. and Pohlmeier, W. (2020). *Portfolio Pretesting with Machine Learning*. Tech. rep. University of Lancaster.

Keranovic, V., Begusic, S., and Kostanjcar, Z. (2020). "Estimating the Number of Latent Factors in High-Dimensional Financial Time Series." In: *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE.

Kinlaw, W. B., Kritzman, M., Page, S., and Turkington, D. (2021). "The Myth of Diversification Reconsidered." In: *The Journal Of Portfolio Management* 47(8).

Kolrep, M., Lohre, H., Radatz, E., and Rother, C. (2020). "Economic Versus Statistical Clustering in Multi-Asset Multi-Factor Strategies." In: *Risk & Reward*, pp. 26–32.

Konstantinov, G., Chorus, A., and Rebmann, J. (2020). "A network and machine learning approach to factor, asset, and blended allocation." In: *The Journal of Portfolio Management* 46 (6), pp. 54–71.

Konstantinov, G. and Rusev, M. (2020). "The Bond-Equity-Fund Relation Using the Fama-French-Carhart Factors: A Practical Network Approach." In: *The Journal of Financial Data Science* 2 (1), pp. 24–44.

Konstantinov, G. S. and Simonian, J. (2020). "A Network Approach to Analyzing Hedge Fund Connectivity." In: *The Journal of Financial Data Science* 22(3) (3), pp. 55–72.

Koumou, G. B. (2020). "Diversification and portfolio theory: a review." In: *Financial Markets and Portfolio Management* 34, pp. 267–312.

Kritzman, M., Kinlaw, W., and Turkington, D. (2017). *A Practitioner's Guide to Asset Allocation*. Wiley. 256 pp.

Kukreti, V., Pharasi, H. K., Gupta, P., and Kumar, S. (2020). "A Perspective on Correlation-Based Financial Networks and Entropy Measures." In: *Frontiers in Physics* 8.

Kumar, S., Bansal, A., and Chakrabarti, A. S. (2021). "Ripples on financial networks." In: *The European Journal of Finance*, Early View.

Kumari, I. and Sharma, V. (2020). "A review for the efficient clustering based on distance and the calculation of centroid." In: *International Journal of Advanced Technology and Engineering Exploration (IJATEE)* 7(63), pp. 48–52.

Kuntz, L.-C. (2018). "Portfolio Strategies with Classical and Alternative Benchmarks." PhD thesis. Georg August University of Gottingen.

Kurtti, M. (2020). "How many stocks make a diversified portfolio in a continuous-time world?" MA thesis. University of Oulu.

Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D., and Hu, X. (2021). "TODS: An Automated Time Series Outlier Detection System." In: *arXiv e-Print*.

Landi, I., Mandelli, V., and Lombardo, M. V. (2020). "reval: a Python package to determine best clustering solutions with stability-based relative clustering validation." In: *arXiv e-Print*.

Laur, B. (2020). "Portfolio Optimization - Can Optimizing Portfolio Outperform Naive Diversification?" In: *SSRN e-Print*.

Laurinaityte, N., Meinerding, C., Schlag, C., and Thimme, J. (2019). "Elephants and the Cross-Section of Expected Returns." In: *SSRN e-Print*.

Lee, T.-H. and Seregina, E. (2022). "Optimal Portfolio Using Factor Graphical Lasso." In: *arXiv e-Print*.

Lemenkova, P. (2020). "R Libraries (dendextend and magrittr) and Clustering Package scipy.cluster of Python For Modelling Diagrams of Dendrogram Trees." In: *Carpathian Journal of Electronic and Computer Engineering* 13(3), pp. 5–12.

Leon, D., Aragon, A., Sandoval, J., Hernandez, G., Arevalo, A., and Nino, J. (2017). "Clustering algorithms for Risk-Adjusted Portfolio Construction." In: *Procedia Computer Science* 108, pp. 1334–1343.

Leopold, N. and Rose, O. (2020). "UNIC: A fast nonparametric clustering." In: *Pattern Recognition* 100, p. 107117.

Lettau, M. and Pelger, M. (2020). "Factors That Fit the Time Series and Cross-Section of Stock Returns." In: *The Review of Financial Studies* 33(5), pp. 2274–2325.

Li, H. and Liu, Z. (2021). "Multivariate time series clustering based on complex network." In: *Pattern Recognition* 115, p. 107919.

Li, T., Levina, E., and Zhu, J. (2020). "Network cross-validation by edge sampling." In: *Biometrika* 107(2), pp. 257–276.

Li, Z., Liu, X.-Y., Zheng, J., Wang, Z., Walid, A., and Guo, J. (2021). "FinRL-Podracer: High Performance and Scalable Deep Reinforcement Learning for Quantitative Finance." In: *ACM International Conference on AI in Finance*.

Lim, T. and Ong, C. S. (2021). "Portfolio Diversification Using Shape-Based Clustering." In: *The Journal of Financial Data Science* 3(1), pp. 111–126.

Lipor, J. and Balzano, L. (2020). "Clustering quality metrics for subspace clustering." In: *Pattern Recognition* 104, p. 107328.

Liu, X.-Y., Rui, J., Gao, J., Yang, L., Yang, H., Wang, Z., Wang, C. D., and Guo, J. (2022). "FinRL-Meta: A Universe of Near-Real Market Environments for Data-Driven Deep Reinforcement Learning in Quantitative Finance." In: *arXiv e-Print*.

Liu, X.-Y., Yang, H., Gao, J., and Wang, C. (2021). "FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance." In: *SSRN e-Print*.

Lohre, H., Rother, C., and Schafer, K. A. (2020). "Hierarchical Risk Parity: Accounting for Tail Dependencies in Multi-asset Multi-factor Allocations." In: *Machine Learning for Asset Management: New Developments and Financial Applications*. Ed. by E. Jurczenko. Wiley, pp. 329–368.

Loistl, O. and Konstantinov, G. S. (2020). "Interactions and Interconnectedness Shape Financial Market Research." In: *The Journal of Financial Data Science* (2), 2, pp. 51–63.

Lopez de Prado, M. (2016). "Building Diversified Portfolios that Outperform Out of Sample." In: *The Journal of Portfolio Management* 42(4), pp. 59–69.

Lopez de Prado, M. (2019a). "A Data Science Solution to the Multiple-Testing Crisis in Financial Research." In: *The Journal of Financial Data Science* 1(1), pp. 99–110.

Lopez de Prado, M. (2019b). "Estimation of Theory-Implied Correlation Matrices." In: *SSRN e-Print*.

Lopez de Prado, M. (2020a). "Clustering." In: *SSRN e-Print*.

Lopez de Prado, M. (2020b). *Machine learning for asset managers*. Cambridge University Press. 190 pp.

Lopez de Prado, M. and Lewis, M. J. (2019). "Detection of false investment strategies using unsupervised learning methods." In: *Quantitative Finance* 19(9), pp. 1555–1565.

Louiset, R., Gori, P., Dufumier, B., Houenou, J., Grigis, A., and Duchesnay, E. (2021). "UCSL : A Machine Learning Expectation-Maximization framework for Unsupervised Clustering driven by Supervised Learning." In: *arXiv e-Print*.

Lu, Y., Li, M., Tang, X., and Wang, H. (2021). "A Cluster Representative Selection Method for Stock Portfolio Based on Efficient Frontier." In: *IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE.

Ma, E. (2020). *Network Analysis Made Simple*. URL: `https://ericmjl.github.io/Network-Analysis-Made-Simple/`.

Ma, E. and Seth, M. (2020). *Network Analysis Made Simple*. 278 pp.

Magner, N. S., Lavin, J. F., Valle, M. A., and Hardy, N. (2021). "The Volatility Forecasting Power of Financial Network Analysis." In: *Complexity* 2020 (7051402).

Maharaj, E. A., D'Urso, P., and Caiado, J. (2019). *Time Series Clustering and Classification*. CRC Press. 244 pp.

Mahfuz, N. M., Yusoff, M., and Ahmad, Z. (2019). "Review of single clustering methods." In: *IAES International Journal of Artificial Intelligence (IJ-AI)* 8(3), p. 221.

Malavasi, M., Lozza, S. O., and Truck, S. (2021). "Second order of stochastic dominance efficiency vs mean variance efficiency." In: *European Journal of Operational Research* 290(3), pp. 1192–1206.

Marinescu, M. (2022). "Risk-Based Optimal Portfolio Strategies: A Compendium." In: *SSRN e-Print*.

Marti, G., Nielsen, F., Bihkowski, M., and Donnat, P. (2021). "A review of two decades of correlations, hierarchies, networks and clustering in financial markets." In: *Progress in Information Geometry*, pp. 245–274.

Martin, R. (2021). "PyPortfolioOpt: portfolio optimization in Python." In: *Journal of Open Source Software* 6(61), p. 3066.

Marwood, D. and Minnen, D. (2020). "Safely Boosting Retirement Income by Harmonizing Drawdown Paths." In: *Journal of Financial Planning* 33(11), pp. 46–60.

Maschner, C., Moritz, B., and Schmitz, M. (2021). "Modern Asset Management." In: *SSRN e-Print*.

Massahi, M., Mahootchi, M., and Khamseh, A. A. (2020). "Development of an efficient cluster-based portfolio optimization model under realistic market conditions." In: *Empirical Economics*.

Mattera, R., Giacalone, M., and Gibert, K. (2021). "Distribution-Based Entropy Weighting Clustering of Skewed and Heavy Tailed Time Series." In: *Symmetry* 13(6), p. 959.

McCabe, S., Torres, L., LaRock, T., Haque, S. A., Yang, C.-H., Hartle, H., and Klein, B. (2020). "netrd: A library for network reconstruction and graph distances." In: *arXiv e-Print*.

McIndoe, C. (2020). "A Data Driven Approach to Market Regime Classification." MA thesis. Imperial College.

Mehta, V., Bawa, S., and Singh, J. (2020). "Analytical review of clustering techniques and proximity measures." In: *Artificial Intelligence Review* 53(8), pp. 5995–6023.

Micheli, A. and Neuman, E. (2022). "Evidence of Crowding on Russell 3000 Reconstitution Events." In: *arXiv e-Print*.

Milevsky, M. A. (2020). *Retirement Income Recipes in R: From Ruin Probabilities to Intelligent Drawdowns*. Springer International Publishing. 302 pp.

Millington, T. and Niranjan, M. (2020a). "Construction of Minimum Spanning Trees from Financial Returns using Rank Correlation." In: *arXiv e-Print*.

Millington, T. and Niranjan, M. (2020b). "Partial correlation financial networks." In: *Applied Network Science* 5(1) (11).

Millington, T. and Niranjan, M. (2021). "Stability and similarity in financial networks – How do they change in times of turbulence?" In: *Physica A: Statistical Mechanics and its Applications* 574, p. 126016.

Miranda, F. M., Koehnecke, N., and Renard, B. Y. (2022). "HiClass: a Python library for local hierarchical classification compatible with scikit-learn." In: *arXiv e-Print*.

Molyboga, M. (2020). "A Modified Hierarchical Risk Parity Framework for Portfolio Management." In: *The Journal of Financial Data Science* 2(3), pp. 128–139.

Montero, P. and Vilar, J. A. (2015). "TSclust: An R Package for Time Series Clustering." In: *Journal of Statistical Software* 62.

Mooney, T., Rapaka, R., and Vera, T. (2020). "Dynamic Regime Strategy for Stress Testing and Optimizing Institutional Investor Portfolios." In: *SSRN e-Print*.

Mori, U., Mendiburu, A., and Lozano, J. A. (2016). "Distance Measures for Time Series in R: The TSdist Package." In: *The R Journal*.

Murialdo, P., Ponta, L., and Carbone, A. (2021). "Inferring multi-period optimal portfolios via detrending moving average cluster entropy." In: *EPL (Europhysics Letters)* 133(6), p. 60004.

Nanakorn, N. and Palmgren, E. (2021). "Hierarchical Clustering in Risk-Based Portfolio Construction." MA thesis. KTH.

Naraoka, M., Hayashi, T., Yoshino, T., Sugie, T., Takano, K., and Ohsawa, Y. (2020). "Detecting and explaining changes in various assets' relationships in financial markets." In: *arXiv e-Print*.

Olmo, J. (2021). "Optimal portfolio allocation and asset centrality revisited." In: *Quantitative Finance* 21(9), pp. 1475–1490.

Page, S. and Panariello, R. A. (2018). "When Diversification Fails." In: *Financial Analysts Journal* 74(3), pp. 19–32.

Pang, R. K.-K., Granados, O. M., Chhajer, H., and Legara, E. F. T. (2021). "An analysis of network filtering methods to sovereign bond yields during COVID-19." In: *Physica A: Statistical Mechanics and its Applications* 574, p. 125995.

Papenbrock, J., Schwendner, P., Jaeger, M., and Krugel, S. (2021a). "Matrix Evolutions: Synthetic Correlations and Explainable Machine Learning for Constructing Robust Investment Portfolios." In: *The Journal of Financial Data Science* 3(2), pp. 51–69.

Papenbrock, J., Schwendner, P., and Sandner, P. (2021b). "Can Adaptive Seriational Risk Parity Tame Crypto Portfolios?" In: *SSRN e-Print*.

Park, J. (2020). "Clustering Approaches for Global Minimum Variance Portfolio." In: *arXiv e-Print*.

Parmentier, L. (2018). "Measures of Portfolio Diversification." MA thesis. Louvain School of Management.

Peng, H., Wang, H., Hu, Y., Zhou, W., and Cai, H. (2022). "Multi-dimensional clustering through fusion of high-order similarities." In: *Pattern Recognition* 121, p. 108108.

Peralta, G. and Zareei, A. (2016). "A network approach to portfolio selection." In: *Journal of Empirical Finance* 38, pp. 157–180.

Perrin, S. and Roncalli, T. (2020). "Machine Learning Optimization Algorithms & Portfolio Allocation." In: *Machine Learning for Asset Management: New Developments and Financial Applications*. Ed. by E. Jurczenko. Wiley, pp. 261–328.

Pharasi, H. K., Sadhukhan, S., Majari, P., Chakraborti, A., and Seligman, T. H. (2021). "Dynamics of the market states in the space of correlation matrices with applications to financial markets." In: *arXiv e-Print*.

Pimentel, B. A. and de Carvalho, A. C. (2020). "A Meta-learning approach for recommending the number of clusters for clustering algorithms." In: *Knowledge-Based Systems* 195, p. 105682.

Platanakis, E., Sutcliffe, C. M., and Ye, X. (2021). "Horses for Courses: Mean-Variance for Asset Allocation and 1/N for Stock Selection." In: *European Journal of Operational Research* 288(1), pp. 302–317.

Poletaev, A. Y. and Spiridonova, E. M. (2020). "Hierarchical Clustering as a Dimension Reduction Technique for Markowitz Portfolio Optimization." In: *Modeling and Analysis of Information Systems* 27(1), pp. 62–71.

Policastro, V., Righelli, D., Carissimo, A., Cutillo, L., and Feis, I. D. (2021). "ROBustness In Network (robin): an R package for Comparison and Validation of communities." In: *arXiv e-Print*.

Puerto, J., Rodriguez-Madrena, M., and Scozzari, A. (2020). "Clustering and portfolio selection problems: A unified framework." In: *Computers & Operations Research* 117, p. 104891.

Putra, Y. E., Saepudin, D., and Aditsania, A. (2021). "Portfolio Selection of KOMPAS-100 Stocks Index Using B-Spline Based Clustering." In: *Procedia Computer Science* 179, pp. 375–382.

Qian, W., Rolling, C. A., Cheng, G., and Yang, Y. (2022). "Combining forecasts for universally optimal performance." In: *International Journal of Forecasting*.

Radovanov, B. and Marcikic, A. (2014). "Testing The Performance Of The Investment Portfolio Using Block Bootstrap Method." In: *Economic Themes* 52(2).

Raffinot, T. (2017). "Hierarchical Clustering-Based Asset Allocation." In: *The Journal of Portfolio Management* 44(2), pp. 89–99.

Raffinot, T. (2018). "The Hierarchical Equal Risk Contribution Portfolio." In: *SSRN e-Print*.

Rahgoshay, M. and Salavatipour, M. R. (2021). "Hierarchical Clustering: New Bounds and Objective." In: *arXiv e-Print*.

Rao, A. and Jelvis, T. (2022). *Foundations of Reinforcement Learning with Applications in Finance*.

Rebonato, R. (2019). “A financially justifiable and practically implementable approach to coherent stress testing.” In: *Quantitative Finance* 19(5), pp. 827–842.

Rehman, A. U. and Belhaouari, S. B. (2022). “Divide well to merge better: A novel clustering algorithm.” In: *Pattern Recognition* 122, p. 108305.

Romashchenko, A. (2021). “Clustering with Respect to the Information Distance.” In: *arXiv e-Print*.

Roncalli, T. (2021). “Advanced Course in Asset Management.” In: *SSRN e-Print*.

Rusch, T., Mair, P., and Hornik, K. (2021). “Cluster Optimized Proximity Scaling.” In: *Journal of Computational and Graphical Statistics*, pp. 1–12.

Ruta, N., Sawada, N., McKeough, K., Behrisch, M., and Beyer, J. (2020). “SAX Navigator: Time Series Exploration through Hierarchical Clustering.” In: *arXiv e-Print*.

Sakurai, Y., Yuki, Y., Katsuki, R., Yazane, T., and Ishizaki, F. (2021). “Correlation diversified passive portfolio strategy based on permutation of assets.” In: *The Journal of Investment Strategies*.

Samal, A., Kumar, S., Yadav, Y., and Chakraborti, A. (2021). “Network-centric indicators for fragility in global financial indices.” In: *arXiv e-Print*.

Sarda-Espinosa, A. (2019a). “Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package.” In: *SSRN e-Print*.

Sarda-Espinosa, A. (2019b). “Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package.” In: *The R Journal*.

Sarmas, E., Xidonas, P., and Doukas, H. (2020). *Multicriteria Portfolio Construction with Python*. Springer International Publishing.

Sass, J. and Thos, A.-K. (2022). “Risk reduction and portfolio optimization using clustering methods.” In: *Econometrics and Statistics*.

Sato-Ilic, M. (2021). “Cluster-scaled principal component analysis.” In: *WIREs Computational Statistics*.

Scherer, B. (2021). “Adding alternative assets: return enhancement, diversification or hedging?” In: *Journal of Asset Management* 22, pp. 437–442.

Schumann, E. (2019). “Backtesting.” In: *SSRN e-Print*.

Schwendner, P., Papenbrock, J., Jaeger, M., and Krugel, S. (2021). “Adaptive Seriational Risk Parity and Other Extensions for Heuristic Portfolio Construction Using Machine Learning and Graph Theory.” In: *The Journal of Financial Data Science* 3(4), pp. 65–83.

Seabrook, I. E., Barucca, P., and Caccioli, F. (2021). “Evaluating structural edge importance in temporal networks.” In: *EPJ Data Science* 10(1).

Sekula, M., Datta, S., and Datta, S. (2017). “optCluster: An R Package for Determining the Optimal Clustering Algorithm.” In: *Bioinformation* 13(03), pp. 101–103.

Serur, J. A. and Avellaneda, M. (2021). “Hierarchical PCA and Modeling Asset Correlations.” In: *SSRN e-Print*.

Seymour, A., Flint, E. J., and Chikurunhe, F. (2018). “Dynamic portfolio management strategies: A framework for historical analysis.” In: *SSRN e-Print*.

Sharma, A., Syrgkanis, V., Zhang, C., and Kiciman, E. (2021). “DoWhy: Addressing Challenges in Expressing and Validating Causal Assumptions.” In: *arXiv e-Print*.

Shi, X., Xu, D., and Zhang, Z. (2022). “Deep Learning Algorithms for Hedging with Frictions.” In: *arXiv e-Print*.

Shirota, Y. and Murakami, A. (2021). “Long-term Time Series Data Clustering of Stock Prices for Portfolio Selection.” In: *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE.

Siebert, J., Gross, J., and Schroth, C. (2021). “A systematic review of Python packages for time series analysis.” In: *Engineering Proceedings* 5(1) (22).

Silva, V. F., Silva, M. E., Ribeiro, P., and Silva, F. (2021a). “Novel Features for Time Series Analysis: A Complex Networks Approach.” In: *arXiv e-Print*.

Silva, V. F., Silva, M. E., Ribeiro, P., and Silva, F. (2021b). “Time series analysis via network science: Concepts and algorithms.” In: *WIREs Data Mining and Knowledge Discovery* 11(3).

Simos, T. E., Mourtas, S. D., and Katsikis, V. N. (2021). “Time-varying Black–Litterman portfolio optimization using a bio-inspired approach and neuronets.” In: *Applied Soft Computing* 112, p. 107767.

Sjostrand, D. and Behnejad, N. (2020). “Exploration of Hierarchical Clustering in Long-only Risk-based Portfolio Optimization.” MA thesis. Copenhagen Business School.

Snow, D. (2019). “Machine learning in asset management.” In: *SSRN e-Print*.

Snow, D. (2020a). “Machine Learning in Asset Management - Part 2: Portfolio Construction - Weight Optimization.” In: *The Journal of Financial Data Science* 2 (2), pp. 17–24.

Snow, D. (2020b). "Machine Learning in Asset Management Part 1: Portfolio Construction Trading Strategies." In: *The Journal of Financial Data Science* 2(1) (1), pp. 10–23.

Sobczyk, P., Wilczynski, S., Bogdan, M., Graczyk, P., Josse, J., Panloup, F., Seegers, V., and Staniak, M. (2020). "VARCLUST: clustering variables using dimensionality reduction." In: *arXiv e-Print*.

Son, B. and Lee, J. (2022). "Graph-based multi-factor asset pricing model." In: *Finance Research Letters* 44 (102032).

Stankovic, L., Mandic, D., Dakovic, M., Brajovic, M., Scalzo, B., and Constantinides, T. (2020a). "Data Analytics on Graphs Part I: Graphs, Graph Spectra, and Spectral Clustering." In: *Foundations and Trends in Machine Learning* 13 (1).

Stankovic, L., Mandic, D., Dakovic, M., Brajovic, M., Scalzo, B., Li, S., and Constantinides, A. G. (2020b). "Data Analytics on Graphs Part II: Signals on Graphs." In: *Foundations and Trends in Machine Learning* 13 (2-3).

Stankovic, L., Mandic, D., Dakovic, M., Brajovic, M., Scalzo, B., Li, S., and Constantinides, A. G. (2020c). "Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications." In: *Foundations and Trends in Machine Learning*.

Stavroglou, S. (2020). "Finding Hidden Structures in Financial Networks." PhD thesis. University of Liverpool.

Suhonen, A., Lennkh, M., and Perez, F. (2017). "Quantifying Backtest Overfitting in Alternative Beta Strategies." In: *The Journal of Portfolio Management* 43 (2), pp. 90–104.

Swedroe, L. (2020). "The Importance of Diversification in Achieving Long-Term Goals." In: *Advisor Perspectives*.

Taljaard, B. H. and Maré, E. (2021). "Why has the equal weight portfolio underperformed and what can we do about it?" In: *Quantitative Finance* 21(11), pp. 1855–1868.

Tang, W., Xu, X., and Zhou, X. Y. (2021). "Asset Selection via Correlation Blockmodel Clustering." In: *arXiv e-Print*.

Tatsat, H., Puri, S., and Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly. 400 pp.

Tayali, S. T. (2020). "A novel backtesting methodology for clustering in mean–variance portfolio optimization." In: *Knowledge-Based Systems* 209, p. 106454.

Tellaroli, P., Bazzi, M., Donato, M., Brazzale, A. R., and Draghici, S. (2016). "Cross-Clustering: A Partial Clustering Algorithm with Automatic Estimation of the Number of Clusters." In: *PLOS ONE* 11(3), e0152333+.

Thiagarajan, R., Han, J., Hurd, A., Im, H., and Mallik, G. (2021). "Financial Globalization and Its Implications for Diversification of Portfolio Risk." In: *The Journal of Investing* 30(6), pp. 22–33.

Thrun, M. C. (2021). "The Exploitation of Distance Distributions for Clustering." In: *International Journal of Computational Intelligence and Applications* 20(03).

Thrun, M. C. and Stier, Q. (2021). "Fundamental clustering algorithms suite." In: *SoftwareX* 13, p. 100642.

Tola, V., Lillo, F., Gallegati, M., and Mantegna, R. N. (2008). "Cluster analysis for portfolio optimization." In: *Journal of Economic Dynamics and Control* 32(1), pp. 235–258.

Tong, W., Liu, S., and Gao, X.-Z. (2021). "A density-peak-based clustering algorithm of automatically determining the number of clusters." In: *Neurocomputing* 458, pp. 655–666.

Traccucci, P., Dumontier, L., Garchery, G., and Jacot, B. (2019). "A Triptych Approach for Reverse Stress Testing of Complex Portfolios." In: *Risk (Cutting Edge)*.

Tuck, J., Barratt, S., and Boyd, S. (2022). "Portfolio Construction Using Stratified Models." In: *Machine Learning in Financial Markets: A guide to contemporary practices*. Ed. by A. Capponi and C.-A. Lehalle. Cambridge University Press.

Turner, E. (2021). "Graph Auto-Encoders for Financial Clustering." In: *arXiv e-Print*.

Underwood, W. G., Elliott, A., and Cucuringu, M. (2020). "Motif-based spectral clustering of weighted directed networks." In: *Applied Network Science* 5(1) (62).

Ungolo, F., Sherris, M., and Zhou, Y. (2021). "affine_mortality: A Github repository for estimation, analysis, and projection of affine mortality models." In: *SSRN e-Print*.

Valentine, K. D., Buchanan, E. M., Scofield, J. E., and Beauchamp, M. T. (2019). "Beyond p values: utilizing multiple methods to evaluate evidence." In: *Behaviormetrika* 46(1), pp. 121–144.

Valk, M. and Cybis, G. B. (2021). "U-Statistical Inference for Hierarchical Clustering." In: *Journal of Computational and Graphical Statistics* 30(1), pp. 133–143.

Vamossy, D. and Skog, R. (2021). "EmTract: Investor Emotions and Market Behavior." In: *arXiv e-Print*.

Vanini, P. (2020). "Asset Management." In: *SSRN e-Print*.

Vankwikelberge, X., Kang, B., Heiter, E., and Lijffijt, J. (2021). "ExClus: Explainable Clustering on Low-dimensional Data Representations." In: *arXiv e-Print*.

Vázquez, I., Villar, J. R., Sedano, J., Simić, S., and Cal, E. de la (2021). "An ensemble solution for multivariate time series clustering." In: *Neurocomputing* 457, pp. 182–192.

Vigen, T. (2019). *Spurious Correlations*. URL: https://www.tylervigen.com/spurious-correlations.

Vincent, K., Hsu, Y.-C., and Lin, H.-W. (2018). "Analyzing the Performance of Multifactor Investment Strategies under a Multiple Testing Framework." In: *The Journal of Portfolio Management* 44(4), pp. 113–126.

Vinod, H. D. (2021). "R Package GeneralCorr Functions for Portfolio Choice." In: *SSRN e-Print*.

Vojtko, R. and Cisár, D. (2021). "An Analysis of Volatility Clustering of Equity Factor Strategies." In: *SSRN e-Print*.

Vovk, V. and Wang, R. (2020). "True and false discoveries with e-values." In: *arXiv e-Print*.

Vovk, V. and Wang, R. (2021). "E-values: Calibration, combination, and applications." In: *Annals of Statistics* 49(3), pp. 1736–1753.

Vyrost, T., Lyocsa, S., and Baumohl, E. (2019). "Network-based asset allocation strategies." In: *The North American Journal of Economics and Finance* 47, pp. 516–536.

Wadhwa, R. R. and Scott, J. G. (2020). "Exploring complex networks with the ICON R package." In: *arXiv e-Print*.

Wang, M., Abrams, Z. B., Kornblau, S. M., and Coombes, K. R. (2018). "Thresher: determining the number of clusters while removing outliers." In: *BMC Bioinformatics* 19(1), p. 9.

Wang, S., Sun, Y., and Bao, Z. (2020). "On the Efficiency of K-Means Clustering: Evaluation, Optimization, and Algorithm Selection." In: *arXiv e-Print*.

Wang, Y. and Tsay, R. S. (2019). "Clustering Multiple Time Series with Structural Breaks." In: *Journal of Time Series Analysis* 40(2), pp. 182–202.

Wang, Y. and Aste, T. (2022). "Dynamic Portfolio Optimization with Inverse Covariance Clustering." In: *arXiv e-Print*.

Weylandt, M., Nagorski, J., and Allen, G. I. (2019). "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization." In: *arXiv e-Print*.

Wiecki, T., Campbell, A., Lent, J., and Stauth, J. (2016). "All That Glitters Is Not Gold: Comparing Backtest and Out-of-Sample Performance on a Large Cohort of Trading Algorithms." In: *The Journal of Investing* 25(3), pp. 69–80.

Wu, C., Peng, Q., Lee, J., Leibnitz, K., and Xia, Y. (2021). "Effective hierarchical clustering based on structural similarities in nearest neighbor graphs." In: *Knowledge-Based Systems* 228, p. 107295.

Wu, X., Wu, J., Zou, J., and Zhang, Q. (2020). "Analyses and applications of optimization methods for complex network reconstruction." In: *Knowledge-Based Systems* 193, p. 105406.

Yang, L., Zhao, L., and Wang, C. (2019). "Portfolio optimization based on empirical mode decomposition." In: *Physica A: Statistical Mechanics and its Applications* 531, p. 121813.

Yang, Y., Zhao, L., Chen, L., Wang, C., and Han, J. (2021). "Portfolio optimization with idiosyncratic and systemic risks for financial networks." In: *arXiv e-Print*.

Yang, Y., UY, M. C. S., and Huang, A. (2020). "FinBERT: A Pretrained Language Model for Financial Communications." In: *arXiv e-Print*.

Yelibi, L. and Gebbie, T. (2021). "Agglomerative Likelihood Clustering." In: *arXiv e-Print*.

Yu, H., Chapman, B., Di Florio, A., Eischen, E., Gotz, D., Jacob, M., and Blair, R. H. (2018). "Bootstrapping estimates of stability for clusters, observations and model selection." In: *Computational statistics* 34(1), pp. 349–372.

Yu, L. (2021). "Comparing Classical Portfolio Optimization and Robust Portfolio Optimization on Black Swan Events." MA thesis. University of Waterloo.

Yu, L., Hardle, W. K., Borke, L., and Benschop, T. (2020). "An AI approach to measuring financial risk." In: *SSRN e-Print*.

Yuan, M. and Zhou, G. (2022). "Why Naive 1/N Diversification Is Not So Naive, and How to Beat It?" In: *SSRN e-Print*.

Zaimovic, A., Omanovic, A., and Arnaut-Berilo, A. (2021). "How Many Stocks Are Sufficient for Equity Portfolio Diversification? A Review of the Literature." In: *Journal of Risk and Financial Management* 14(11), p. 551.

Zambelli, A. (2021). "Ensemble Method for Cluster Number Determination and Algorithm Selection in Unsupervised Learning." In: *arXiv e-Print*.

Zhan, N., Sun, Y., Jakhar, A., and Liu, H. (2021). "Graphical Models for Financial Time Series and Portfolio Selection." In: *arXiv e-Print*.

Zhang, C., Li, Y., Chen, X., Jin, Y., Tang, P., and Li, J. (2020a). “DoubleEnsemble: A New Ensemble Method Based on Sample Reweighting and Feature Selection for Financial Data Analysis.” In: *IEEE International Conference on Data Mining (ICDM)*. IEEE.

Zhang, F., Guo, R., and Cao, H. (2020b). “Information Coefficient as a Performance Measure of Stock Selection Models.” In: *arXiv e-Print*.

Zhang, M. (2021). “Weighted Clustering Ensemble: A Review.” In: *arXiv e-Print*.

Zhang, Z., Zohren, S., and Roberts, S. (2020c). “Deep Learning for Portfolio Optimization.” In: *The Journal of Financial Data Science* 22(4), pp. 8–20.

Zhao, L., Wang, C., Wang, G.-J., Stanley, H. E., and Chen, L. (2021a). “Community detection and portfolio optimization.” In: *arXiv e-Print*.

Zhao, L., Wang, G.-J., Wang, M., Bao, W., Li, W., and Stanley, H. E. (2018). “Stock market as temporal network.” In: *Physica A: Statistical Mechanics and its Applications* 506, pp. 1104–1112.

Zhao, Z., Xu, F., Du, D., and Meihua, W. (2021b). “Robust portfolio rebalancing with cardinality and diversification constraints.” In: *Quantitative Finance* 21(10), pp. 1707–1721.

Zheng, Q., Zhu, J., Ma, Y., Li, Z., and Tian, Z. (2021). “Multi-view subspace clustering networks with local and global graph information.” In: *Neurocomputing* 449, pp. 15–23.

Zhong, C., Hu, L., Yue, X., Luo, T., Fu, Q., and Xu, H. (2019). “Ensemble clustering based on evidence extracted from the co-association matrix.” In: *Pattern Recognition* 92, pp. 93–106.

Zhong, G. and Pun, C.-M. (2020). “Subspace clustering by simultaneously feature selection and similarity learning.” In: *Knowledge-Based Systems* 193, p. 105512.

Zhou, P., Chen, J., Fan, M., Du, L., Shen, Y.-D., and Li, X. (2020). “Unsupervised feature selection for balanced clustering.” In: *Knowledge-Based Systems* 193, p. 105417.

Appendix A: Overviews of investment processes and models in QWIM

References

List of references:

Coqueret and Guida (*Machine Learning for Factor Investing: R Version*, 2020)
Dixon et al. (*Machine Learning in Finance: from theory to practice*, 2020)
Fabozzi et al. (*Asset Management: Tools and Issues*, 2021)
Grealish and Kolm ("Robo-Advisory: From Investing Principles and Algorithms to Future Developments," 2021)
Homescu ("Many risks, one (optimal) portfolio," 2014)
Homescu ("Better Investing Through Factors, Regimes and Sensitivity Analysis," 2015)
Jansen (*Machine Learning for Algorithmic Trading (Second Edition)*, 2020)
Jurczenko et al. (*Machine Learning for Asset Management*, 2020)
Kritzman et al. (*A Practitioner's Guide to Asset Allocation*, 2017)
Lopez de Prado (*Machine learning for asset managers*, 2020)
Maschner et al. ("Modern Asset Management," 2021)
Perrin and Roncalli ("Machine Learning Optimization Algorithms & Portfolio Allocation," 2020)
Roncalli ("Advanced Course in Asset Management," 2021)
Vanini ("Asset Management," 2020)

Online courses

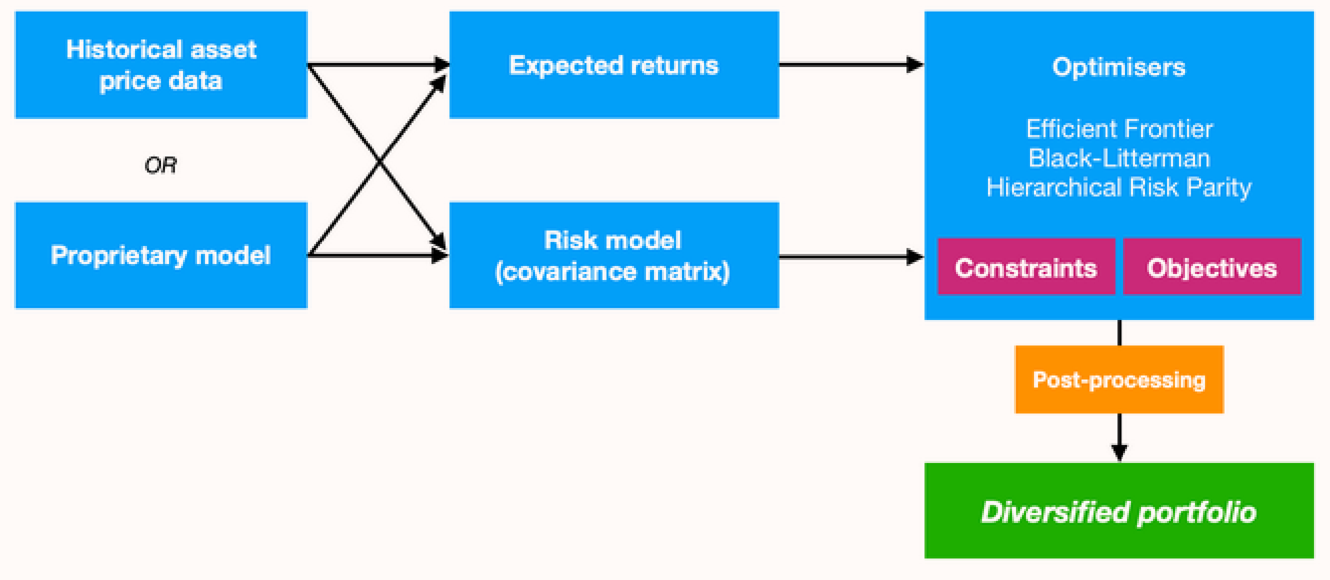
List of online courses:

- Investment Management with Python and Machine Learning Specialization
  - ◇ Introduction to Portfolio Construction and Analysis with Python
  - ◇ Advanced Portfolio Construction and Analysis with Python
  - ◇ Python and Machine Learning for Asset Management
  - ◇ Python and Machine Learning for Asset Management with Alternative Data Sets
- Machine Learning and Reinforcement Learning in Finance Specialization
  - ◇ Guided Tour of Machine Learning in Finance
  - ◇ Fundamentals of Machine Learning in Finance
  - ◇ Reinforcement Learning in Finance
  - ◇ Overview of Advanced Methods of Reinforcement Learning in Finance
- Investment Management Specialization
  - ◇ Understanding Financial Markets
  - ◇ Meeting Investors' Goals
  - ◇ Portfolio and Risk Management
  - ◇ Securing Investment Returns in the Long Run
  - ◇ Planning your Client's Wealth over a 5-year Horizon
- Investment and Portfolio Management Specialization
  - ◇ Global Financial Markets and Instruments
  - ◇ Portfolio Selection and Risk Management
  - ◇ Biases and Portfolio Selection
  - ◇ Investment Strategies and Portfolio Analysis
  - ◇ Build a Winning Investment Portfolio

Appendix C: Incorporating comparison of portfolio metrics using benchmark portfolios

For your QWIM project it is likely that you would compare investment portfolios constructed using your method(s) versus benchmark portfolios constructed using most common "optimal portfolio" types used in the industry and in academia. See below for an example of how this can be done.

Figure 6: Example of portfolio optimization process



Source: PyPortfolioOpt

Portfolio optimization methods

List of portfolio optimization methods may include (see Roncalli ("Advanced Course in Asset Management," 2021) and Perrin and Roncalli ("Machine Learning Optimization Algorithms & Portfolio Allocation," 2020) for a comprehensive overview of such methods):

- equal weighting
- mean variance optimization (Markowitz)
- minimum variance optimization
- maximum diversification
- risk budgeting/risk parity
- hierarchical risk parity
- Black-Litterman
- robust versions of some the above portfolio optimization methods

Some relevant links:

- Portfolio Optimization: A General Framework for Portfolio Choice
- Performance of risk-based asset allocation strategies
- Revisiting the Portfolio Optimization Machine Portfolio
- Construction Techniques Applied to Traditional Multi Asset Portfolios

Python and R packages/codes for portfolio optimization

- Codes mentioned in Snow ("Machine Learning in Asset Management - Part 2: Portfolio Construction - Weight Optimization," 2020)
- Empyrial
- MLFinLab
- Optimal Portfolio
- PortfolioAnalytics
- PortfolioLab
- PyPortfolioOpt
- Quantropy
- Riskfolio-Lib
- RiskPortfolios
- riskparityportfolio

Portfolio metrics

List of portfolio metrics may include some of the following (see Bacon ("Performance Attribution: History and Progress," 2019) for a comprehensive list):

- Sharpe ratio
- Sortino ratio
- Information ratio
- Maximum Drawdown
- expected shortfall
- maximum loss
- and more.

Some relevant links:

- Portfolio metrics
- Picking the Right Risk-Adjusted Performance Metric
- Risk-Adjusted Performance Measurement – State of the Art
- An Investor's Guide to the Risk Versus Return Conundrum
- How sharp is the Sharpe ratio? Risk-adjusted Performance Measures

Python and R packages/codes for portfolio metrics and performance evaluation

- bt
- empyrical
- ffn
- JFE
- MLFinLab
- PerformanceAnalytics
- portfolioBacktest
- Portfolio Optimization and Performance Evaluation
- Pyfolio
- QuantStats
- Riskfolio-Lib
- tidyquant

How to compare investment portfolios

Let us consider portfolio optimization methods (selected from the ones implemented in Python and/or R packages mentioned above, such as PyPortfolioOpt) which rely on based on expected returns and expected covariance matrix.

One would construct two portfolios (let's call them Traditional and Enhanced) using the same portfolio optimization method(s), where the only difference would be in terms of the inputs (expected returns and expected covariance matrix) to the optimization method:

As of the date of portfolio construction, expected returns and expected covariance matrix can be either calculated using only historical data or, respectively, output from your model. Then one would compare side-by-side various portfolio metrics for these two portfolios. Comparison would be done across the entire Out-Of-Sample period, and also across each market regine period.

NOTE: If you have N forecasting methods used in your coding framework, then for each optimizaton method you would end up with (1+N) optimal portfolios

To exemplifty, let's say that you want to construct portfolios at date of June 20, 2019, and you have data as below

- Range of entire dataset: January 1st, 1990 - August 1, 2020
- Range of Training dataset: January 1st, 1990- February 20, 2017
- Range of Test dataset: February 20, 2017 - August 1, 2020

For Traditional portfolio:

- vector of expected means is calculated based on historical data available at June 20, 2019 (namely from 1990 to June 19, 2019)
- expected covariance matrix is calculated based on historical data available at June 20, 2019 (namely from 1990 to June 19, 2019)

For Enhanced portfolio:

- vector of expected means is calculated based on forecasted values available at June 20, 2019 and obtained using the forecasted model trained on given training dataset (which is from 1990 to 2017)

- expected covariance matrix is calculated based on forecasted values available at June 20, 2019 and obtained using the forecasted model trained on given training dataset (which is from 1990 to 2017)

Then one would compare various portfolio metrics among the two portfolios. These metrics can be calculated on following time periods:

- from date of portfolio construction (June 20, 2019) to last date for which you have data (August 1, 2020)
- from starting date of dataset (January 1st, 1990) to last date for which you have data (August 1, 2020)
- from starting date of dataset (January 1st, 1990) to date of portfolio construction (June 20, 2019)

So you would have side-by-side comparisons of portfolio metrics for each of the above 3 time periods.

Portfolio metrics can be calculated using various Python and/or R packages mentioned above.