[STATISTICS] (Data Science & Analysis CheatSheet)

Descriptive Statistics

- **Mean**: mean(x)
- **Median**: median(x)
- **Mode**: mode(x)
- **Range**: max(x) - min(x)
- **Variance**: variance(x)
- **Standard Deviation**: std_deviation(x)
- **Coefficient of Variation**: std_deviation(x) / mean(x)
- **Percentiles**: percentile(x, p)
- **Interquartile Range (IQR)**: Q3(x) - Q1(x)
- **Skewness**: skewness(x)
- **Kurtosis**: kurtosis(x)
- **Mean Absolute Deviation (MAD)**: mean(abs(x - mean(x)))
- **Five Number Summary**: min, Q1, median, Q3, max

Probability Distributions

- **Binomial Distribution**: binom_dist(n, p, x)
- **Poisson Distribution**: poisson_dist(λ, x)
- **Normal Distribution**: norm_dist(μ, σ, x)
- **t-Distribution**: t_dist(v, x)
- **Chi-Squared Distribution**: chi2_dist(v, x)
- **F-Distribution**: f_dist(d1, d2, x)
- **Exponential Distribution**: exp_dist(λ, x)
- **Uniform Distribution**: uniform_dist(a, b, x)

Correlation and Covariance

- **Covariance**: covariance(X, Y)
- **Pearson Correlation Coefficient**: pearson_r(X, Y)
- **Spearman's Rank Correlation**: spearman_rho(X, Y)
- **Kendall's Tau**: kendall_tau(X, Y)

By: Waleed Mousa

Regression Analysis

- **Simple Linear Regression**: `y = b0 + b1*x`
- **Multiple Linear Regression**: `y = b0 + b1*x1 + b2*x2 + ... + bn*xn`
- **Polynomial Regression**: `y = b0 + b1*x + b2*x^2 + ... + bn*x^n`
- **Logistic Regression**: `logit(p) = ln(p/(1-p)) = b0 + b1*x`
- **Coefficient of Determination (R^2)**: `R² = 1 - (SS_res / SS_tot)`

Hypothesis Testing

- **Z-test**: `z = (x̄ - μ) / (σ/sqrt(n))`
- **t-test for Independent Samples**: `t = (x̄1 - x̄2) / sqrt(s_p²(1/n1 + 1/n2))`
- **t-test for Paired Samples**: `t = (d̄ - μd) / (Sd / sqrt(n))`
- **Chi-Squared Test**: `χ² = Σ((O - E)²/E)`
- **ANOVA (Analysis of Variance)**: `F = MST / MSE`
- **Mann-Whitney U Test**: `U = n1*n2 + (n1(n1+1)/2) - R1`
- **Wilcoxon Signed-Rank Test**: `W = Σ|ranked_differences|`
- **Kruskal-Wallis Test**: `H = (N-1)*Σ(ni(Ri²) / Σni)`

Sampling and Estimations

- **Simple Random Sampling**: `random_sample(n, population)`
- **Systematic Sampling**: `systematic_sample(k, population)`
- **Stratified Sampling**: `stratified_sample(strata, population)`
- **Cluster Sampling**: `cluster_sample(clusters, population)`
- **Point Estimation**: `point_estimate = x̄ or p̂`
- **Confidence Interval for Mean**: `(x̄ - Z*(σ/√n), x̄ + Z*(σ/√n))`
- **Confidence Interval for Proportion**: `(p̂ - Z*(√(p̂(1-p̂)/n)), p̂ + Z*(√(p̂(1-p̂)/n)))`

Significance and Power Analysis

- **Alpha Level (Type I Error)**: `α`
- **Beta Level (Type II Error)**: `β`
- **Power of a Test**: `1 - β`
- **Effect Size (Cohen's d)**: `d = (μ1 - μ2) / σ pooled`

By: Waleed Mousa

- **Sample Size Calculation for Mean Difference**: $n = ((Z\alpha/2 + Z\beta)^2 * 2\sigma^2) / d^2$
- **Sample Size Calculation for Proportions**: $n = (p1(1-p1) + p2(1-p2)) * (Z\alpha/2 + Z\beta)^2 / (p1-p2)^2$

Time Series Analysis

- **Moving Average**: moving_average(t, k)
- **Exponential Smoothing**: smoothed_value = α*current_value + (1-α)*previous_smoothed_value
- **Seasonal Decomposition**: decompose(time_series)
- **Autocorrelation Function (ACF)**: acf(time_series)
- **Partial Autocorrelation Function (PACF)**: pacf(time_series)
- **Seasonal ARIMA (SARIMA)**: SARIMA(time_series, p, d, q, P, D, Q, s)
- **Vector Autoregression (VAR)**: VAR(time_series)
- **Cointegration Test**: cointegration_test(series1, series2)
- **Exponential Triple Smoothing (ETS)**: ETS(time_series)

Non-parametric Methods

- **Mann-Kendall Trend Test**: mann_kendall(time_series)
- **Kolmogorov-Smirnov Test**: ks_test(sample1, sample2)
- **Kruskal-Wallis H Test**: kruskal_wallis(groups)
- **Spearman's Rank Correlation**: spearman_rank(X, Y)
- **Wilcoxon Signed-Rank Test**: wilcoxon_signed_rank(sample)
- **Mood's Median Test**: moods_median_test(sample1, sample2)
- **Friedman Test**: friedman_test(data)
- **Cochran's Q Test**: cochrans_Q(tests)
- **Run Test for Randomness**: runs_test(data)

Bayesian Statistics

- **Bayes' Theorem**: $P(A|B) = (P(B|A) * P(A)) / P(B)$
- **Posterior Distribution**: $\text{posterior} \propto \text{likelihood} * \text{prior}$
- **Beta Distribution as Prior**: beta(α, β)
- **Markov Chain Monte Carlo (MCMC)**: mcmc(samples, parameters)

Dimensionality Reduction

- **Principal Component Analysis (PCA)**: `PCA(data)`
- **Factor Analysis**: `factor_analysis(data)`
- **t-SNE (t-Distributed Stochastic Neighbor Embedding)**: `tSNE(data)`

Cluster Analysis

- **K-Means Clustering**: `kmeans(data, k)`
- **Hierarchical Clustering**: `hierarchical_clustering(data)`
- **DBSCAN**: `dbscan(data, ε, minPts)`
- **Silhouette Score**: `silhouette_score(data, labels)`

Association Analysis

- **Support in Association Rule**: `support(X) = P(X)`
- **Confidence in Association Rule**: `confidence(X=>Y) = P(X U Y) / P(X)`
- **Lift in Association Rule**: `lift(X=>Y) = confidence(X=>Y) / P(Y)`

Survival Analysis

- **Kaplan-Meier Estimate**: `kaplan_meier(survival_times)`
- **Log-Rank Test**: `log_rank(test, control)`
- **Cox Proportional Hazards Model**: `cox_proportional_hazards(data)`
- **Weibull Reliability Function**: `weibull_reliability(β, η, t)`
- **Log-Rank Test for Survival Data**: `log_rank(survival_times1, survival_times2)`
- **Proportional Hazards Assumption Check**: `proportional_hazards_test(data)`

Quality Control

- **Control Charts**: `control_chart(data)`
- **Pareto Chart**: `pareto_chart(data)`
- **Process Capability Index (Cpk)**: `Cpk(lower_spec, upper_spec, data)`
- **Deming Regression (for Method Comparison)**: `deming_regression(method1, method2)`
- **Six Sigma Process Capability**: `six_sigma_capability(process_data)`

- **Statistical Process Control (SPC)**: `SPC(control_data)`

Experimental Design

- **Analysis of Variance (ANOVA)**: `ANOVA(data)`
- **Covariance Analysis (ANCOVA)**: `ANCOVA(data)`
- **Factorial Design Analysis**: `factorial_design(data)`

Advanced Topics

- **Generalized Linear Models (GLM)**: `GLM(data, family)`
- **Mixed Effects Models**: `mixed_effects_model(data)`
- **Time Series Forecasting (ARIMA, etc.)**: `forecast_ARIMA(time_series)`
- **Machine Learning Algorithms**: `machine_learning_algorithm(data)`

Meta-Analysis

- **Fixed-Effect Model**: `fixed_effect(meta_data)`
- **Random-Effects Model**: `random_effects(meta_data)`

Decision Analysis

- **Expected Value Calculation**: `expected_value(decision_outcomes, probabilities)`
- **Decision Tree Analysis**: `decision_tree(decision, outcomes)`
- **Utility Function Modeling**: `utility(value, risk_aversion)`
- **Sensitivity Analysis**: `sensitivity_analysis(model, parameter)`
- **Monte Carlo Decision Making**: `monte_carlo_decision(decision_model, iterations)`

Advanced Probability

- **Conditional Probability**: `P(A|B) = P(A and B) / P(B)`
- **Joint Probability**: `P(A and B)`
- **Marginal Probability**: `P(A)`

Special Distributions and Functions

- **Gamma Distribution**: gamma_dist(shape, scale, x)
- **Beta Distribution**: beta_dist(α, β, x)
- **Weibull Distribution**: weibull_dist(λ, k, x)
- **Dirichlet Distribution**: dirichlet_dist(alpha)

Quality and Performance Metrics

- **Sensitivity/Recall/True Positive Rate**: TP / (TP + FN)
- **Specificity/True Negative Rate**: TN / (TN + FP)
- **Precision/Positive Predictive Value**: TP / (TP + FP)
- **F1 Score**: 2 * (Precision * Recall) / (Precision + Recall)

Multivariate Analysis

- **Canonical Correlation Analysis**: CCA(X, Y)
- **Multivariate Analysis of Variance (MANOVA)**: MANOVA(data)
- **Principal Component Regression (PCR)**: PCR(X, Y)
- **Partial Least Squares Regression (PLSR)**: PLSR(X, Y)

Advanced Modeling Techniques

- **Ridge Regression**: ridge_regression(X, Y, λ)
- **Lasso Regression**: lasso_regression(X, Y, λ)
- **Elastic Net**: elastic_net(X, Y, α, λ)
- **Support Vector Machines**: SVM(X, Y)

Model Evaluation and Validation

- **Cross-Validation**: cross_validation(model, data, k)
- **Bootstrapping for Error Estimation**: bootstrap_error(model, data)
- **AIC (Akaike Information Criterion)**: AIC(model)
- **BIC (Bayesian Information Criterion)**: BIC(model)

Advanced Probability and Distributions

- **Multinomial Distribution**: multinomial_dist(n, probabilities)
- **Negative Binomial Distribution**: negative_binomial(r, p)
- **Hypergeometric Distribution**: hypergeometric(N, K, n)

By: Waleed Mousa

- **Bivariate Normal Distribution**: `bivariate_normal(μ1, μ2, σ1, σ2, ρ)`

Spatial and Geostatistical Analysis

- **Kriging for Spatial Interpolation**: `kriging(spatial_data)`
- **Moran's I for Spatial Autocorrelation**: `morans_I(spatial_data)`
- **Geographically Weighted Regression (GWR)**: `GWR(spatial_data)`

Risk Analysis and Financial Statistics

- **Value at Risk (VaR)**: `VaR(portfolio, α)`
- **Expected Shortfall (CVaR)**: `CVaR(portfolio, α)`
- **Sharpe Ratio**: `sharpe_ratio(returns, risk_free_rate)`
- **Beta Coefficient in Finance**: `beta(stock_returns, market_returns)`

Advanced Cluster and Classification Methods

- **Gaussian Mixture Models (GMM)**: `GMM(data, components)`
- **Agglomerative Hierarchical Clustering**: `agglomerative_clustering(data)`
- **Dendrogram for Hierarchical Clustering**: `dendrogram(hierarchical_model)`
- **Naive Bayes Classifier**: `naive_bayes(features, labels)`

Psychometrics and Educational Statistics

- **Item Response Theory (IRT)**: `IRT(item_responses)`
- **Cronbach's Alpha for Reliability**: `cronbachs_alpha(data)`
- **ANOVA for Repeated Measures**: `repeated_measures_ANOVA(data)`

Scale Development and Validation

- **Exploratory Factor Analysis (EFA)**: `EFA(items)`
- **Confirmatory Factor Analysis (CFA)**: `CFA(items, model)`
- **Item Discrimination Analysis**: `item_discrimination(test_items)`

Network Analysis

- **Degree Centrality**: `degree_centrality(network)`
- **Betweenness Centrality**: `betweenness_centrality(network)`
- **Community Detection**: `community_detection(network)`

Advanced Techniques in Data Reduction

- **Multidimensional Scaling (MDS)**: `MDS(distance_matrix)`
- **Isomap for Nonlinear Dimensionality Reduction**: `isomap(data)`
- **Local Linear Embedding (LLE)**: `LLE(data)`

Miscellaneous Advanced Operations

- **Copula for Joint Distribution Modeling**: `copula(types, parameters)`
- **Gini Coefficient for Inequality**: `gini(income_distribution)`
- **Entropy for Information Theory**: `entropy(probabilities)`
- **Simpson's Diversity Index**: `simpsons_diversity(species_counts)`
- **Monte Carlo Simulations**: `monte_carlo(model, parameters)`
- **Bootstrap Resampling**: `bootstrap(sample)`
- **Jackknife Resampling**: `jackknife(sample)`