Big Data Science in Finance (WI3436TU) Formula sheet for the written exam, 2026/27

1 Univariate data and distributions

Returns: for the price $S_t > 0$, raw return $M_t = \dfrac{S_t - S_{t-1}}{S_{t-1}}$, log-return $r_t = \log S_t - \log S_{t-1} = \log(1 + M_t) \approx M_t$ for small moves; $\log(S_t/S_{t-k}) = r_{t-k+1} + \cdots + r_t$.
Empirical moments: for $r_1, \ldots, r_n$, $\bar r = \frac{1}{n}\sum_t r_t$, $\widehat\sigma^2 = \frac{1}{n}\sum_t (r_t - \bar r)^2$.
Gaussian: $R \sim \mathcal{N}(\mu, \sigma^2)$ with

$$\phi_{\mu,\sigma^2}(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(r-\mu)^2}{2\sigma^2}\Big), \quad Z = \frac{R-\mu}{\sigma} \sim \mathcal{N}(0,1).$$

Φ, ϕ: standard normal cdf and density; $\Phi(-z) = 1 - \Phi(z)$, $\mathbb{P}(|Z| \le z) = 2\Phi(z) - 1$.
Student-t: $R = \mu + sT_\nu$, location μ, scale $s > 0$, ν degrees of freedom. $\mathbb{E}[T_\nu] = 0$ ($\nu > 1$), $\mathrm{Var}(T_\nu) = \frac{\nu}{\nu-2}$ ($\nu > 2$); $\mathbb{E}\big[|T_\nu|^k\big] < \infty$ only for $k < \nu$. Smaller ν: heavier tails.
MLE: $L(\theta) = \prod_{t=1}^n f_\theta(r_t)$, $\ell(\theta) = \sum_{t=1}^n \log f_\theta(r_t)$, $\widehat\theta_{\mathrm{MLE}} \in \arg\max_\theta \ell(\theta)$. Gaussian: $\widehat\mu = \bar r$, $\widehat\sigma^2_{\mathrm{MLE}} = \frac{1}{n}\sum_t (r_t - \bar r)^2$.
Density histogram: origin a, bin width b, bins $B_j = [a + (j-1)b,\ a + jb)$; height $\dfrac{\#\{r_t \in B_j\}}{nb}$ on B_j; bar area $\#\{r_t \in B_j\}/n$; total area 1.
KDE: kernel $K \ge 0$ with $\int K = 1$, bandwidth $h > 0$, $\widehat f_h(x) = \dfrac{1}{nh}\sum_{t=1}^n K\Big(\dfrac{x - r_t}{h}\Big)$. Gaussian kernel: $K = \phi$.
Q–Q plot: k-th smallest observation against the p_k-quantile $F^{-1}(p_k)$ of the reference distribution, $p_k = k/(n+1)$.
Loss, VaR, ES: for the return R of a position over one period, with cdf F_R, the loss is $L = -R$, with cdf F_L; level $\alpha \in (0,1)$:

$$\mathrm{VaR}_\alpha(L) = \inf\{\ell \in \mathbb{R} : F_L(\ell) \ge \alpha\} = -F_R^{-1}(1-\alpha)$$

(for continuous, strictly increasing F_R), so $\mathbb{P}(L > \mathrm{VaR}_\alpha) = 1 - \alpha$;

$$\mathrm{ES}_\alpha(L) = \mathbb{E}\big[L \mid L \ge \mathrm{VaR}_\alpha(L)\big] = \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_u(L)\,du \ \ge\ \mathrm{VaR}_\alpha(L).$$

Gaussian $R \sim \mathcal{N}(\mu, \sigma^2)$, $z_\alpha = \Phi^{-1}(\alpha)$:

$$\mathrm{VaR}_\alpha = -\mu + \sigma z_\alpha, \qquad \mathrm{ES}_\alpha = -\mu + \sigma\,\frac{\phi(z_\alpha)}{1-\alpha}.$$

Student-t $R = \mu + sT_\nu$: $\mathrm{VaR}_\alpha = -\mu + s\,t_{\nu,\alpha}$, with $t_{\nu,\alpha}$ the α-quantile of T_ν.

2 Dependence and bivariate data

Joint and marginal: $F_{X,Y}(x,y) = \mathbb{P}(X \le x, Y \le y)$, $f_{X,Y} = \dfrac{\partial^2 F_{X,Y}}{\partial x\,\partial y}$, $F_X(x) = F_{X,Y}(x, +\infty)$, $f_X(x) = \int_{\mathbb{R}} f_{X,Y}(x,y)\,dy$.
Independence: $F_{X,Y} = F_X F_Y$ (equivalently $f_{X,Y} = f_X f_Y$).
Covariance, Pearson correlation:
$\mathrm{Cov}(X,Y) = \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(Y - \mathbb{E}[Y]\big)\Big]$,
$\rho = \mathrm{Corr}(X,Y) = \dfrac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}} \in [-1,1]$. Empirical, for pairs (x_t, y_t):

$$\widehat\sigma_{XY} = \frac{1}{n}\sum_t (x_t - \bar x)(y_t - \bar y), \qquad \widehat\rho = \frac{\widehat\sigma_{XY}}{\widehat\sigma_X \widehat\sigma_Y}.$$

Kendall's tau: $(X_1, Y_1), (X_2, Y_2)$ independent copies are concordant if $(X_1 - X_2)(Y_1 - Y_2) > 0$, discordant if < 0; $\tau = \mathbb{P}(\text{conc.}) - \mathbb{P}(\text{disc.})$; $\widehat\tau = \dfrac{C - D}{\binom{n}{2}}$ (C, D: numbers of concordant and discordant pairs).
Bivariate Gaussian: $\mathcal{N}_2(\boldsymbol\mu, \boldsymbol\Sigma)$, $\boldsymbol\Sigma = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}$.
Simulation with $\varepsilon_1, \varepsilon_2 \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$: $X = \mu_X + \sigma_X\varepsilon_1$, $Y = \mu_Y + \sigma_Y\big(\rho\varepsilon_1 + \sqrt{1-\rho^2}\,\varepsilon_2\big)$.
Copula: $U = F_X(X)$, $V = F_Y(Y)$ uniform on $[0,1]$; $C(u,v) = \mathbb{P}(U \le u, V \le v)$, density $c = \dfrac{\partial^2 C}{\partial u\,\partial v}$. Sklar (continuous marginals): $F_{X,Y}(x,y) = C(F_X(x), F_Y(y))$. Pseudo-observations: $u_t = \dfrac{\mathrm{rank}(x_t)}{n+1}$, $v_t = \dfrac{\mathrm{rank}(y_t)}{n+1}$. z-scale: $Z_1 = \Phi^{-1}(U)$, $Z_2 = \Phi^{-1}(V)$, density $g(z_1, z_2) = c(\Phi(z_1), \Phi(z_2))\,\phi(z_1)\phi(z_2)$.
Rotations: for a copula C of (U,V), $(1-U, V) \sim C^{90}$, $C^{90}(u,v) = v - C(1-u, v)$;
$(1-U, 1-V) \sim C^{180}$, $C^{180}(u,v) = u + v - 1 + C(1-u, 1-v)$;
$(U, 1-V) \sim C^{270}$, $C^{270}(u,v) = u - C(u, 1-v)$.
Model selection: BIC $= -2\ell_n(\widehat{\boldsymbol\theta}) + k\log n$, $k = \dim\boldsymbol\theta$; smaller is better. **Simulation**: $(U^{\mathrm{sim}}, V^{\mathrm{sim}}) \sim C_{\widehat{\boldsymbol\theta}}$, then $X^{\mathrm{sim}} = \widehat F_X^{-1}(U^{\mathrm{sim}})$, $Y^{\mathrm{sim}} = \widehat F_Y^{-1}(V^{\mathrm{sim}})$.

3 Linear algebra and SVD

Data matrix: in Lectures 3–5, $\mathbf{X} \in \mathbb{R}^{n\times m}$, rows are n assets, columns are m dates; column $\mathbf{x}_t$ is the return vector of date t.
Basics: $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top\mathbf{x}}$; $\mathbf{x} \perp \mathbf{y}$ iff $\mathbf{x}^\top\mathbf{y} = 0$. Orthogonal $\mathbf{Q}$: $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$, $\|\mathbf{Q}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$. $\mathrm{rank}(\mathbf{X}) \le \min\{n, m\}$; $\mathrm{rank}(\mathbf{X}\mathbf{X}^\top) = \mathrm{rank}(\mathbf{X})$.
Eigenpair: $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, $\mathbf{v} \ne \mathbf{0}$; eigenvalues solve $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. For 2×2: $\lambda_1 + \lambda_2 = \mathrm{tr}\,\mathbf{A}$, $\lambda_1\lambda_2 = \det\mathbf{A}$. Symmetric $\mathbf{A}$: real eigenvalues, orthonormal eigenvectors, $\sum_j \lambda_j^2 = \mathrm{tr}(\mathbf{A}^2) = \sum_{i,j} a_{ij}^2$.
SVD: for $\mathbf{X}$ with rank r,

$$\mathbf{X} = \mathbf{U}_r\boldsymbol\Sigma_r\mathbf{V}_r^\top = \sum_{j=1}^r \sigma_j\mathbf{u}_j\mathbf{v}_j^\top, \qquad \sigma_1 \ge \cdots \ge \sigma_r > 0,$$

$\mathbf{U}_r \in \mathbb{R}^{n\times r}$, $\mathbf{V}_r \in \mathbb{R}^{m\times r}$ with orthonormal columns.
$\mathbf{X}\mathbf{X}^\top = \mathbf{U}_r\boldsymbol\Sigma_r^2\mathbf{U}_r^\top$, $\mathbf{X}^\top\mathbf{X} = \mathbf{V}_r\boldsymbol\Sigma_r^2\mathbf{V}_r^\top$; $\mathbf{v}_j = \mathbf{X}^\top\mathbf{u}_j/\sigma_j$, $\mathbf{u}_j = \mathbf{X}\mathbf{v}_j/\sigma_j$.
Truncation: for $1 \le k < r$, $\mathbf{X}_k = \sum_{j=1}^k \sigma_j\mathbf{u}_j\mathbf{v}_j^\top = \mathbf{U}_k\boldsymbol\Sigma_k\mathbf{V}_k^\top$, $\mathrm{rank}(\mathbf{X}_k) = k$. Norms: $\|\mathbf{A}\|_F^2 = \sum_{i,j} a_{ij}^2 = \sum_j \sigma_j^2$, $\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2=1}\|\mathbf{A}\mathbf{x}\|_2 = \sigma_1$. Eckart–Young–Mirsky:

$$\min_{\mathrm{rank}(\mathbf{B})\le k} \|\mathbf{X} - \mathbf{B}\|_F = \|\mathbf{X} - \mathbf{X}_k\|_F = \Big(\sum_{j=k+1}^r \sigma_j^2\Big)^{1/2}.$$

Retained squared energy $\eta_k = \dfrac{\sum_{j\le k}\sigma_j^2}{\sum_{j\le r}\sigma_j^2}$; $\dfrac{\|\mathbf{X} - \mathbf{X}_k\|_F}{\|\mathbf{X}\|_F} = \sqrt{1 - \eta_k}$.

4 Principal component analysis

Centering: $\mathbf{R} \in \mathbb{R}^{n\times m}$, $\bar{\mathbf{r}} = \frac{1}{m}\mathbf{R}\mathbf{1}_m$, $\mathbf{X} = \mathbf{R} - \bar{\mathbf{r}}\mathbf{1}_m^\top$ (each row has mean 0). **PCA via SVD**: for $\mathbf{X} = \mathbf{U}_r\boldsymbol\Sigma_r\mathbf{V}_r^\top$, principal directions (loading vectors) $\mathbf{u}_k$, loading of asset i: u_{ik}. Scores $y_{kt} = \mathbf{u}_k^\top\mathbf{x}_t$, $\mathbf{Y} = \mathbf{U}_r^\top\mathbf{X} = \boldsymbol\Sigma_r\mathbf{V}_r^\top$, $\mathbf{y}_k = \mathbf{X}^\top\mathbf{u}_k = \sigma_k\mathbf{v}_k$; $\mathbf{x}_t = \sum_k y_{kt}\mathbf{u}_k$. Projection on a unit $\mathbf{u}$: $\|\mathbf{x}_t\|_2^2 = y_t^2 + \|\mathbf{x}_t - y_t\mathbf{u}\|_2^2$, $y_t = \mathbf{u}^\top\mathbf{x}_t$.
Empirical covariance and component variances:

$$\widehat{\mathbf{C}} = \tfrac{1}{m}\mathbf{X}\mathbf{X}^\top = \mathbf{U}_r\tfrac{\boldsymbol\Sigma_r^2}{m}\mathbf{U}_r^\top, \qquad \lambda_k = \widehat{\mathrm{Var}}(\mathbf{y}_k) = \tfrac{1}{m}\textstyle\sum_t y_{kt}^2 = \dfrac{\sigma_k^2}{m}.$$

$\widehat{\mathbf{C}}\mathbf{u}_k = \lambda_k\mathbf{u}_k$. Total variance $= \sum_i \widehat{\mathrm{Var}}(X_i) = \mathrm{tr}(\widehat{\mathbf{C}}) = \sum_k \lambda_k$.
$\mathrm{EVR}_k = \dfrac{\lambda_k}{\lambda_1 + \cdots + \lambda_r}$, $\mathrm{CEV}_K = \dfrac{\lambda_1 + \cdots + \lambda_K}{\lambda_1 + \cdots + \lambda_r}$.
Covariance PCA: uses $\mathbf{X}$. **Correlation PCA**: uses $z_{it} = x_{it}/s_i$, $s_i^2 = \frac{1}{m}\sum_t x_{it}^2$; then every row has variance 1 and $\mathrm{tr}(\mathbf{Z}\mathbf{Z}^\top/m) = n$.

5 Signal versus noise

Sample covariance: $\mathbf{S} = \frac{1}{m}\mathbf{X}\mathbf{X}^\top$ ($= \widehat{\mathbf{C}}$); sample spectrum: its eigenvalues $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$. If $n > m$: $\mathrm{rank}(\mathbf{S}) \le m$, at least $n - m$ zero eigenvalues. Empirical eigenvalue distribution: $\mu_n = \frac{1}{n}\sum_{j=1}^n \delta_{\lambda_j}$ (δ_c: point mass at c).
Zero-diagonal Wigner matrix: $\mathbf{W}_n$ symmetric, zero diagonal, entries $W_{ij} = W_{ji} = \xi_{ij}$ ($i < j$), with ξ_{ij} iid, $\mathbb{E}[\xi_{ij}] = 0$, $\mathbb{E}[\xi_{ij}^2] = 1$, $\mathbb{E}[\xi_{ij}^4] < \infty$. Then

$$\mathrm{tr}(\mathbf{W}_n) = 0, \qquad \mathrm{tr}(\mathbf{W}_n^2) = 2\sum_{i<j}\xi_{ij}^2, \qquad \mathbb{E}\Big[\frac{1}{n}\sum_{j=1}^n \lambda_j(\mathbf{W}_n)^2\Big] = n - 1.$$

Normalized: $\mathbf{H}_n = \mathbf{W}_n/\sqrt{n}$, $\lambda_j(\mathbf{H}_n) = \lambda_j(\mathbf{W}_n)/\sqrt{n}$.
Semicircle law: as n grows, μ_n of $\mathbf{H}_n$ approaches the density $\rho_{\mathrm{sc}}(x) = \dfrac{\sqrt{4 - x^2}}{2\pi}$ for $|x| \le 2$ (0 otherwise), so for large n

$$\frac{\#\{j : \lambda_j(\mathbf{H}_n) \in [x_1, x_2]\}}{n} \approx \int_{x_1}^{x_2} \rho_{\mathrm{sc}}(x)\,dx.$$

$\int \sqrt{4 - x^2}\,dx = \frac{x}{2}\sqrt{4 - x^2} + 2\arcsin\frac{x}{2} + \text{const}$.
Marchenko–Pastur law: entries of $\mathbf{X} \in \mathbb{R}^{n\times m}$ iid with $\mathbb{E}[X_{it}] = 0$, $\mathbb{E}[X_{it}^2] = \sigma^2$, $\mathbb{E}[X_{it}^4] < \infty$; $n, m \to \infty$, $n/m \to \gamma \in (0, \infty)$. μ_n of $\mathbf{S}$ converges to

$$f_{\gamma,\sigma^2}(x) = \frac{\sqrt{(b-x)(x-a)}}{2\pi\gamma\sigma^2 x}, \quad a < x < b, \qquad \begin{aligned} a &= \sigma^2(1 - \sqrt{\gamma})^2, \\ b &= \sigma^2(1 + \sqrt{\gamma})^2, \end{aligned}$$

plus a point mass $1-1/\gamma$ at 0 if $\gamma>1$. Almost surely $\lambda_1\to b$; $\lambda_n\to a$ ($\gamma<1$), $\lambda_n\to 0$ ($\gamma=1$); for $\gamma>1$ the smallest positive eigenvalue $\to a$.

6 Supervised learning in finance

Layout: dates in rows. $\mathbf{X}\in\mathbb{R}^{m\times p}$ (p predictors on m dates), $\mathbf{y}\in\mathbb{R}^m$. Predictors standardized and response centered with training data: $z_{tj}=\dfrac{x_{tj}-\bar{x}_j^{\text{tr}}}{s_j^{\text{tr}}}$, $(s_j^{\text{tr}})^2=\frac{1}{m_{\text{tr}}}\sum_{t\in\text{tr}}(x_{tj}-\bar{x}_j^{\text{tr}})^2$. Training, validation and test dates in time order.

OLS: $\widehat{\boldsymbol{\beta}}_{\text{OLS}}=\arg\min_{\boldsymbol{\beta}}\frac{1}{2m}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2=\Big(\dfrac{\mathbf{X}^\top\mathbf{X}}{m}\Big)^{-1}\dfrac{\mathbf{X}^\top\mathbf{y}}{m}$.

$\dfrac{\mathbf{X}^\top\mathbf{X}}{m}=\mathbf{V}\,\text{diag}(d_1,\dots,d_p)\mathbf{V}^\top$, $d_1\ge\cdots\ge d_p\ge 0$, eigenvectors $\mathbf{v}_j$.

Ridge: for $\lambda\ge 0$, minimize $\frac{1}{2m}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2$:

$$\widehat{\boldsymbol{\beta}}_{\text{ridge}}=\Big(\frac{\mathbf{X}^\top\mathbf{X}}{m}+\lambda\mathbf{I}_p\Big)^{-1}\frac{\mathbf{X}^\top\mathbf{y}}{m},\qquad \mathbf{v}_j^\top\widehat{\boldsymbol{\beta}}_{\text{ridge}}=\frac{d_j}{d_j+\lambda}\,\mathbf{v}_j^\top\widehat{\boldsymbol{\beta}}_{\text{OLS}}.$$

LASSO: minimize $\frac{1}{2m}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1$, $\|\boldsymbol{\beta}\|_1=\sum_j|\beta_j|$.
$\widehat{\boldsymbol{\beta}}_{\text{LASSO}}=\mathbf{0}$ iff $\lambda\ge\lambda_{\max}=\max_j|\mathbf{x}_j^\top\mathbf{y}|/m$ ($\mathbf{x}_j$: column j).

Elastic net: penalty $\lambda\big[\rho\|\boldsymbol{\beta}\|_1+\frac{1-\rho}{2}\|\boldsymbol{\beta}\|_2^2\big]$, $0\le\rho\le 1$.

Rolling windows: at refit date t, use the last W rows s whose response is known at t, standardize with their means and standard deviations, forecast the next rows.

$$R^2_{\text{OOS}}=1-\frac{\sum_t(y_t-\widehat{y}_t)^2}{\sum_t(y_t-\bar{y}_t^{\text{win}})^2},$$

$\bar{y}_t^{\text{win}}$: mean response of the window used for date t.

7 Neural networks in finance

Neuron: $z=\mathbf{w}^\top\mathbf{x}+b$, $a=\phi(z)$. **One hidden layer**: p inputs, H units,

$$\mathbf{z}=\mathbf{W}^{(1)}\mathbf{x}+\mathbf{b}^{(1)},\quad \mathbf{h}=\phi(\mathbf{z}),\quad o=\mathbf{w}^{(2)\top}\mathbf{h}+b^{(2)},\quad \widehat{y}=g(o),$$

$\mathbf{W}^{(1)}\in\mathbb{R}^{H\times p}$, $\mathbf{b}^{(1)},\mathbf{w}^{(2)}\in\mathbb{R}^H$, $b^{(2)}\in\mathbb{R}$; ϕ acts on each coordinate, g is the output activation. Dense network with widths $n_0=p,n_1,\dots,n_{L+1}$, $\mathbf{W}^{(k)}\in\mathbb{R}^{n_k\times n_{k-1}}$: $P=\sum_{k=1}^{L+1}n_k(n_{k-1}+1)$ parameters.

Activation	Definition	Range	Derivative
sigmoid	$\sigma(z)=\frac{1}{1+e^{-z}}$	$(0,1)$	$\sigma(z)(1-\sigma(z))$
tanh	$\tanh(z)$	$(-1,1)$	$1-\tanh^2(z)$
ReLU	$\max\{z,0\}$	$[0,\infty)$	$\mathbf{1}\{z>0\}$, $z\neq 0$

Output and loss: $Y\in\mathbb{R}$: identity output, squared error $\frac{1}{2}(y-\widehat{y})^2$.
$Y\in\{0,1\}$: sigmoid output $\widehat{p}=\sigma(o)$, binary cross-entropy (BCE) $\ell(y,\widehat{p})=-y\log\widehat{p}-(1-y)\log(1-\widehat{p})$.

Training: $\boldsymbol{\theta}=$ all weights and biases; empirical risk $J(\boldsymbol{\theta})=\frac{1}{m}\sum_{t=1}^m\ell(y_t,f_{\boldsymbol{\theta}}(\mathbf{x}_t))$; gradient descent $\boldsymbol{\theta}_{r+1}=\boldsymbol{\theta}_r-\eta\nabla J(\boldsymbol{\theta}_r)$, learning rate $\eta>0$. Mini-batch: gradient averaged over a random batch; epoch: one pass through the training data.

Backpropagation: one date, sigmoid output, BCE, $\dfrac{\partial\ell}{\partial o}=\widehat{p}-y$;

$\dfrac{\partial\ell}{\partial\mathbf{w}^{(2)}}=(\widehat{p}-y)\mathbf{h}$, $\dfrac{\partial\ell}{\partial b^{(2)}}=\widehat{p}-y$;

$\delta_k=\dfrac{\partial\ell}{\partial z_k}=(\widehat{p}-y)\,w_k^{(2)}\,\phi'(z_k)$; $\dfrac{\partial\ell}{\partial W_{kj}^{(1)}}=\delta_k x_j$, $\dfrac{\partial\ell}{\partial b_k^{(1)}}=\delta_k$.

Weight decay: $J_\lambda=J+\frac{\lambda}{2}\sum_k\|\mathbf{W}^{(k)}\|_F^2$ (biases not penalized); step $\mathbf{W}^{(k)}\leftarrow(1-\eta\lambda)\mathbf{W}^{(k)}-\eta\nabla_{\mathbf{W}^{(k)}}J$. **Early stopping**: record training and validation loss per epoch; keep $\boldsymbol{\theta}_{r^\star}$ with the smallest validation loss.

Tables

Standard normal quantiles: $z_p=\Phi^{-1}(p)$

p	0.90	0.95	0.975	0.99	0.995	0.999
z_p	1.282	1.645	1.960	2.326	2.576	3.090
$\phi(z_p)$	0.1755	0.1031	0.0584	0.0267	0.0145	0.0034
$\frac{\phi(z_p)}{1-p}$	1.755	2.063	2.338	2.665	2.892	3.367

Standard normal cdf: $\Phi(z)$

z	0.5	1.0	1.5	2.0	2.5	3.0	3.5
$\Phi(z)$	.6915	.8413	.9332	.9772	.9938	.9987	.9998

Student-t quantiles: $t_{\nu,p}$, the p-quantile of T_ν; $t_{\nu,1-p}=-t_{\nu,p}$

$\nu\backslash p$	0.90	0.95	0.975	0.99	0.995	0.999
3	1.638	2.353	3.182	4.541	5.841	10.215
4	1.533	2.132	2.776	3.747	4.604	7.173
5	1.476	2.015	2.571	3.365	4.032	5.893
6	1.440	1.943	2.447	3.143	3.707	5.208
10	1.372	1.812	2.228	2.764	3.169	4.144
30	1.310	1.697	2.042	2.457	2.750	3.385

Python reference

NumPy (`import numpy as np`):

- `A @ B` matrix product; `A.T` transpose; `np.eye(p)` identity; `np.diag(s)` diagonal matrix from `s`; `np.outer(a, b)` $=\mathbf{ab}^\top$. `A[i]` row i, `A[:, j]` column j (counting from 0); `x[-1]` last entry; `x[x>c]` entries above c; `np.arange(a, b)`: $a,\dots,b-1$.
- `axis=0`: along rows, one result per column; `axis=1`: one result per row; `keepdims=True` keeps a column shape `(n, 1)`.
- `x.mean()`, `x.std()`, `np.var(x)`: divide by the number of observations (`ddof=0`). `np.cov(X)`: rows are variables, divisor: number of observations minus 1 (`ddof=0`: number of observations); `np.corrcoef(X)`: correlation matrix of the rows.
- `np.quantile(x, q)`: empirical q-quantile (linear interpolation); `np.sort(x)` increasing order, `np.argsort(x)` the positions in that order (`[::-1]` reverses); `np.argmax(x)` position of the largest entry; `np.cumsum(x)` running sums; `np.searchsorted(c, v)`: first i with $c_i\ge v$ (`c` increasing); `np.log1p(x)` $=\log(1+x)$; `np.maximum(x, 0)` entrywise maximum.
- `np.linalg.svd(X, full_matrices=False)` returns `U, s, Vt`: `s` in decreasing order, `Vt` $=\mathbf{V}^\top$.
- `np.linalg.eigh(A)`, `np.linalg.eigvalsh(A)` (symmetric `A`): eigenvalues in increasing order; eigenvectors are the columns of the second output.
- `np.linalg.solve(A, b)` solves $\mathbf{A}\boldsymbol{\beta}=\mathbf{b}$; `np.linalg.lstsq(X, y)[0]` OLS coefficients; `np.linalg.norm(A, "fro")`, `np.linalg.norm(A, 2)`: $\|\mathbf{A}\|_F$, $\|\mathbf{A}\|_2$; `np.triu(A, 1)`: zeros on and below the diagonal.
- `rng = np.random.default_rng(seed)`; `rng.normal(loc, scale, size)`, `scale` the standard deviation; `rng.uniform(low, high, size)`; `rng.choice(values, size)`; `rng.multivariate_normal(mean, cov, size)`, one draw per row.

pandas: `S` is a Series of prices S_t, `df` a DataFrame; one row per date, in time order.

- `S.pct_change()`: raw returns, row t holds $(S_t-S_{t-1})/S_{t-1}$; `np.log(S).diff()`: log-returns, row t holds $\log S_t-\log S_{t-1}$. The first row is `NaN`; `.dropna()` drops rows with `NaN`.
- `S.shift(k)`: row t holds the value of row $t-k$ ($k<0$: of row $t+|k|$), else `NaN`.
- `S.rolling(W).mean()`, `.sum()`, `.std()`: row t uses rows $t-W+1,\dots,t$ (`NaN` for the first $W-1$); `.std()` divides by $W-1$, `.std(ddof=0)` by W. `S.resample("ME").last()`: last value of each month (older pandas: `"M"`).
- `df.mean()`, `df.std()`, `df.quantile(q)`: one value per column; `df.std()` and `df.describe()` divide by the number of observations minus 1; `df.to_numpy()`: the values as a NumPy array.

SciPy (`from scipy import stats`):

- `stats.norm.cdf`, `.sf`, `.pdf`, `.ppf`: Φ, $1-\Phi$, ϕ, Φ^{-1} (options `loc`, `scale`); the same for `stats.t` with `df=nu`: `stats.t.ppf(p, df=nu)` $=t_{\nu,p}$.
- `stats.norm.fit(r)`, `stats.t.fit(r)`: maximum likelihood estimates, `t.fit` returns `(nu, loc, scale)`; `stats.gaussian_kde(r)`: Gaussian KDE.
- `stats.kendalltau(x, y).statistic` $=\widehat{\tau}$ (no ties); `stats.rankdata(x)`: ranks $1,\dots,n$.

pyvinecopulib (`pv`): `m = pv.Bicop(data=uv, controls=pv.FitControlsBicop(family_set=[pv.gumbel]))` fits this family to the $n\times 2$ pseudo-observations `uv`; `m.bic(uv)`, `m.simulate(n)`.

scikit-learn: `model.fit(X, y)` estimates, `model.coef_` coefficients, `model.predict(X)` forecasts.

- `Lasso(alpha=lam, fit_intercept=False)` minimizes $\frac{1}{2m}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1$; `Ridge(alpha=a, fit_intercept=False)` minimizes $\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+a\|\boldsymbol{\beta}\|_2^2$.
- `ElasticNet(alpha=lam, l1_ratio=rho, fit_intercept=False)` minimizes $\frac{1}{2m}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\big[\rho\|\boldsymbol{\beta}\|_1+\frac{1-\rho}{2}\|\boldsymbol{\beta}\|_2^2\big]$.
- `MLPClassifier(hidden_layer_sizes=(H,), alpha=..., random_state=s)`: ReLU hidden units, sigmoid output; `alpha` penalizes the weights (larger: stronger); `.partial_fit` trains one epoch; `.predict_proba(X)[:, 1]` $=\widehat{p}$; `coefs_[k]`: shape (inputs, units) of layer $k+1$; `intercepts_[k]`: its biases.
- `sklearn.metrics`: `log_loss(y, p)` average BCE; `accuracy_score(y, yhat)` share of correct forecasts; `mean_squared_error(y, yhat)`; `r2_score(y, yhat)`: R^2 against the mean of `y` itself.