

```
▽ --> /* 4 ポートフォリオ理論: CAPM */

▽ --> /* [数値処理] → [自動的に数値で出力] */
    if numer#false then numer:false else numer:true;
    /*浮動小数点の桁数=7に指定 */
    fpprec : 7;
    /* 浮動小数点表示桁数を7に指定*/
    fpprintprec:7;

▽ --> /* 4.1 平均 - 分散ポートフォリオ */

▽ --> t : makelist(-0.5+0.05*i,i,0,40) /* -0.5 から 1.5 までの点列を 0.05 刻みで作る */;

▽ --> mu_1:0.1 /* 資産 1 の平均収益率 */;
    mu_2:0.05 /* 資産 2 の平均収益率 */;
    sigma_1:0.2 /* 資産 1 のボラティリティ */;
    sigma_2:0.1 /* 資産 2 のボラティリティ */;
    cov1:-0.015 /* 資産 1 と資産 2 の共分散 */;
    cov2:0;
    cov3:0.015;

▽ --> mu_p : t*mu_1+ (1-t) *mu_2 /* 収益期待値の列を作る */;

▽ --> sigma1_p : sqrt (t^2*sigma_1^2+ (1-t) ^2*sigma_2^2+2*t* (1-t) *cov1)
    /* ボラティリティの列を作る */;
    sigma2_p : sqrt (t^2*sigma_1^2+ (1-t) ^2*sigma_2^2+2*t* (1-t) *cov2);
    sigma3_p : sqrt (t^2*sigma_1^2+ (1-t) ^2*sigma_2^2+2*t* (1-t) *cov3);

▽ --> sigma_mu1: makelist([sigma1_p[i], mu_p[i]], i, 1, length(t))
    /* draw2dで描画のための[ボラティリティ, 期待収益率]リスト */;
    sigma_mu2: makelist([sigma2_p[i], mu_p[i]], i, 1, length(t));
    sigma_mu3: makelist([sigma3_p[i], mu_p[i]], i, 1, length(t));
```

```

▽ --> load(draw);
      draw2d(
        key_pos=top_left,
        points_joined=true,
        key="cov<0", point_type=circle,
        points(sigma_mu1),
        key="cov=0", point_type=asterisk,
        points(sigma_mu2),
        key="cov>0", point_type=plus,
        points(sigma_mu3),
        xlabel="sigma_p", ylabel="mu_p",
        xrange=[0,0.3]
      );

▽ --> /* 3資産のケース */
      mu_1 : 0.1 /* 資産 1 の平均収益率*/ ;
      mu_2 : 0.05 /* 資産 2 の平均収益率*/;
      mu_3 : 0.15 /* 資産 3 の平均収益率*/ ;
      sigma_1 : 0.2 /* 資産 1 のボラティリティ */;
      sigma_2 : 0.1 /* 資産 2 のボラティリティ */;
      sigma_3 : 0.3 /* 資産 3 のボラティリティ */;
      cov_12 : -0.015 /* 資産 1 と資産 2 の共分散 */;
      cov_13 : 0.01 /* 資産 1 と資産 3 の共分散 */;
      cov_23 : -0.02 /* 資産 2 と資産 23の共分散 */;

▽ --> /* 資産 1 , 資産 2 , 資産 3 保有比率 =t, s, 1-t-s
      ただし, s, tは, 0.02 刻みの -0.5 から 1.5 までの 101 個からなる点列とする */
      s(i):=-0.5+0.02*i; t(j):=-0.5+0.02*j;

▽ --> /* ポートフォリオ平均リターン */
      mu_p : makelist(
        makelist(t(j)*mu_1 + s(i)*mu_2 + (1-t(j)-s(i)) *mu_3, i, 0, 100), j, 0,100);

▽ --> /* ポートフォリオボラティリティ */
      sigma_p : makelist(
        makelist(
          sqrt(
            t(j)^2*sigma_1^2 + s(i)^2*sigma_2^2 + (1-t(j)-s(i)) ^2*sigma_3^2
            + 2*t(j)*s(i)*cov_12 + 2*t(j)* (1-t(j)-s(i)) *cov_13
            + 2*s(i)* (1-t(j)-s(i)) *cov_23
          ),
          i, 0, 100),
        j, 0,100);

```

```

--> /* mu_p, sigma_pは101x101の2重リスト, これらを1重のリストに書き換える */
mu:mu_p[1]; for i:2 thru length(mu_p) do mu:append(mu,mu_p[i]);
sigma:sigma_p[1]; for i:2 thru length(sigma_p) do sigma:append(sigma,sigma_p[i]);

--> /* scatterplot描画用データ[sigmaの列ベクトル, muの列ベクトル]を作成 */
sigma_mu : addcol(transpose(sigma), transpose(mu));

--> scatterplot(sigma_mu, xrange=[0,0.3], yrange=[0.05,0.15] ,
                xlabel="sigma", ylabel="mu");

--> /* 空売りが許されない場合 */
mu_1 : 0.1 ;
mu_2 : 0.05 ;
mu_3 : 0.15 ;
sigma_1 : 0.2 ;
sigma_2 : 0.1 ;
sigma_3 : 0.3 ;
cov_12 : -0.015 ;
cov_13 : 0.01 ;
cov_23 : -0.02 ;

--> s(i):=0.02*i; t(j):=0.02*j;

--> mu_p : makelist(
            makelist(t(j)*mu_1 + s(i)*mu_2 + (1-t(j)-s(i)) *mu_3, i, 0, 50-j), j, 0,50);

--> sigma_p : makelist(
            makelist(
                sqrt(
                    t(j)^2*sigma_1^2 + s(i)^2*sigma_2^2 + (1-t(j)-s(i)) ^2*sigma_3^2
                    + 2*t(j)*s(i)*cov_12 + 2*t(j)* (1-t(j)-s(i)) *cov_13
                    + 2*s(i)* (1-t(j)-s(i)) *cov_23
                ),
                i, 0, 50-j),
            j, 0,50);

--> mu:mu_p[1]; for i:2 thru length(mu_p) do mu:append(mu,mu_p[i]);

--> sigma:sigma_p[1]; for i:2 thru length(sigma_p) do sigma:append(sigma,sigma_p[i]);

--> /* sigma, muからdraw2d描画用[x座標, y座標]リストを作成 */
sigma_mu : makelist([sigma[i], mu[i]], i, 1, length(mu));

```

```
▽ --> load(draw);
      draw2d(
        points(sigma_mu), xrange=[0,0.3], yrange=[0.05,0.15] ,
          xlabel="sigma", ylabel="mu");

▽ --> /* 4.2 市場ポートフォリオ */
      /* 最大値を求めるmaxiを使うため、記事値統計パッケージdescriptiveを読み込む */
      load(descriptive);
      R_f : 0.07 /* 安全資産の収益率 */;
      a_max : maxi ((mu-R_f) /sigma) /* 直線の正の傾き */ ;

▽ --> draw2d(
      points(sigma_mu),
      explicit(R_f+a_max *x, x, 0, 0.3),
      xrange=[0,0.3], yrange=[0.05,0.15] ,
      xlabel="sigma", ylabel="mu");
```