Partial Moments Equivalences

Below are some basic equivalences demonstrating partial moments' role as the elements of variance.

Why is this relevant?

The additional information generated from partial moments permits a level of analysis simply not possible with traditional summary statistics. There is further introductory material on partial moments and their extension into nonlinear analysis & behavioral finance applications available at:

https://www.linkedin.com/pulse/elements-variance-fred-viole

Installation

```
require(devtools); install_github('OVVO-Financial/NNS',ref = "NNS-Beta-Version")
```

Mean

A difference between the upside area and the downside area of f(x).

```
set.seed(123); x=rnorm(100); y=rnorm(100)

> mean(x)
[1] 0.09040591

> UPM(1,0,x)-LPM(1,0,x)
[1] 0.09040591
```

Variance

A sum of the squared upside area and the squared downside area.

```
> var(x)
[1] 0.8332328

# Sample Variance:
> UPM(2,mean(x),x)+LPM(2,mean(x),x)
[1] 0.8249005

# Population Variance:
> (UPM(2,mean(x),x)+LPM(2,mean(x),x))*(length(x)/(length(x)-1))
[1] 0.8332328

# Variance is also the co-variance of itself:
> (Co.LPM(1,1,x,x,mean(x),mean(x))+Co.UPM(1,1,x,x,mean(x),mean(x))-
D.LPM(1,1,x,x,mean(x),mean(x))-D.UPM(1,1,x,x,mean(x),mean(x)))*(length(x)/(
length(x)-1))
[1] 0.8332328
```

Standard Deviation

```
> sd(x)
[1] 0.9128159

> ((UPM(2,mean(x),x)+LPM(2,mean(x),x))*(length(x)/(length(x)-1)))^.5
[1] 0.9128159
```

Covariance

```
> cov(x,y)
[1] -0.04372107

> (Co.LPM(1,1,x,y,mean(x),mean(y))+Co.UPM(1,1,x,y,mean(x),mean(y))-
D.LPM(1,1,x,y,mean(x),mean(y))-D.UPM(1,1,x,y,mean(x),mean(y)))*(length(x)/(
length(x)-1))
[1] -0.04372107
```

Covariance Elements and Covariance Matrix

```
> cov(cbind(x,y))
            x           y
x  0.83323283 -0.04372107
y -0.04372107  0.93506310

> cov.mtx=PM.matrix(LPM.degree = 1,UPM.degree = 1,target = 'mean', variable =
cbind(x,y), pop.adj = TRUE)

> cov.mtx
$clpm
          x         y
x 0.4033078 0.1559295
y 0.1559295 0.3939005

$cupm
          x         y
x 0.4299250 0.1033601
y 0.1033601 0.5411626

$dlpm
          x         y
x 0.0000000 0.1469182
y 0.1560924 0.0000000

$dupm
          x         y
x 0.0000000 0.1560924
y 0.1469182 0.0000000

$matrix
            x           y
x  0.83323283 -0.04372107
y -0.04372107  0.93506310
```

Pearson Correlation

```
> cor(x,y)
[1] -0.04953215

> cov.xy=(Co.LPM(1,1,x,y,mean(x),mean(y))+Co.UPM(1,1,x,y,mean(x),mean(y))-
D.LPM(1,1,x,y,mean(x),mean(y))-D.UPM(1,1,x,y,mean(x),mean(y)))*(length(x)
/(length(x)-1))

> sd.x=((UPM(2,mean(x),x)+LPM(2,mean(x),x))*(length(x)/(length(x)-1)))^.5

> sd.y=((UPM(2,mean(y),y)+LPM(2,mean(y),y))*(length(y)/(length(y)-1)))^.5

> cov.xy/(sd.x*sd.y)
```

```
[1] -0.04953215
```

Skewness*

A normalized difference between upside area and downside area.

```
> skewness(x)
[1] 0.06049948

> ((UPM(3,mean(x),x)-LPM(3,mean(x),x))/(UPM(2,mean(x),x)+LPM(
2,mean(x),x))^(3/2))
[1] 0.06049948
```

UPM/LPM – a more intuitive measure of skewness. (Upside area / Downside area)

```
> UPM(1,0,x)/LPM(1,0,x)
[1] 1.282673
```

Kurtosis*

A normalized sum of upside area and downside area.

```
> kurtosis(x)
[1] -0.161053

> ((UPM(4,mean(x),x)+LPM(4,mean(x),x))/(UPM(2,mean(x),x)+LPM(2,mean(x),x))^2)-3
[1] -0.161053
```

CDFs

```
> P=ecdf(x)

> P(0);P(1)
[1] 0.48
[1] 0.83

> LPM(0,0,x);LPM(0,1,x)
[1] 0.48
[1] 0.83

# Vectorized targets:
> LPM(0,c(0,1),x)
[1] 0.48 0.83

# Joint CDF:
> Co.LPM(0,0,x,y,0,0)
[1] 0.28

# Vectorized targets:
> Co.LPM(0,0,x,y,c(0,1),c(0,1))
[1] 0.28 0.73
```

PDFs

```
> tgt=sort(x)

# Arbitrary d/dx approximation
> d.dx=(max(x)+abs(min(x)))/100

> PDF=(LPM.ratio(1,tgt+d.dx,x)-LPM.ratio(1,tgt-d.dx,x))
```

```
> plot(sort(x),PDF,col='blue',type='l',lwd=3,xlab="x")
```

Numerical Integration – [UPM(1,0,f(x))-LPM(1,0,f(x))]=[F(b)-F(a)]/[b-a]

```
# x is uniform sample over interval [a,b]; y = f(x)
> x=seq(0,1,.001);y=x^2

> UPM(1,0,y)-LPM(1,0,y)
[1] 0.3335
```