

Skapa modeller i Vikingen

Generellt

En modell är egentligen ett datorprogram som utför saker åt dig. Det börjar med att du har en idé som du vill testa om den är lönsam eller inte. Du behöver skriva ned din idé så att datorn förstår den, dvs du behöver skapa ett datorprogram.

Alla program består av

par

var

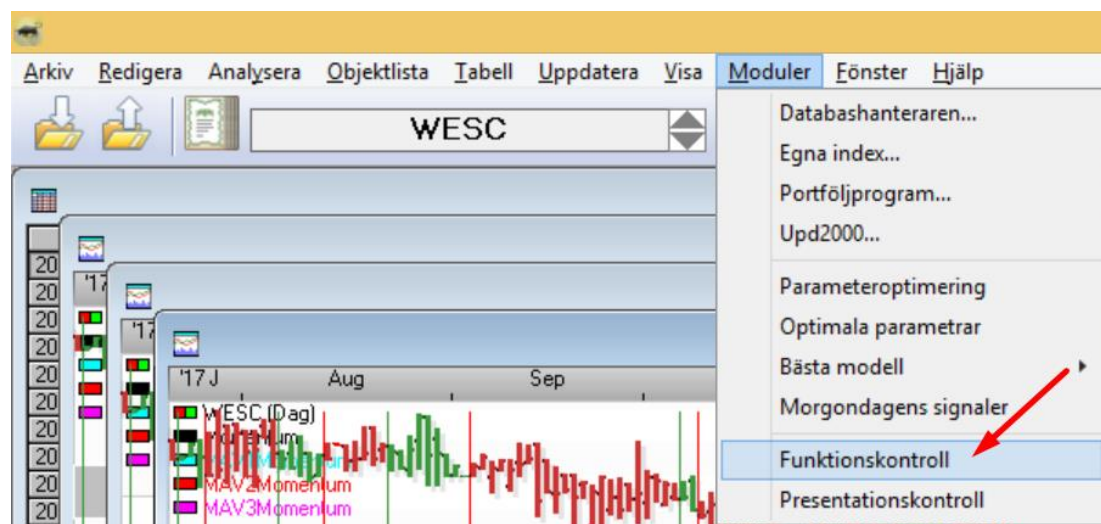
begin

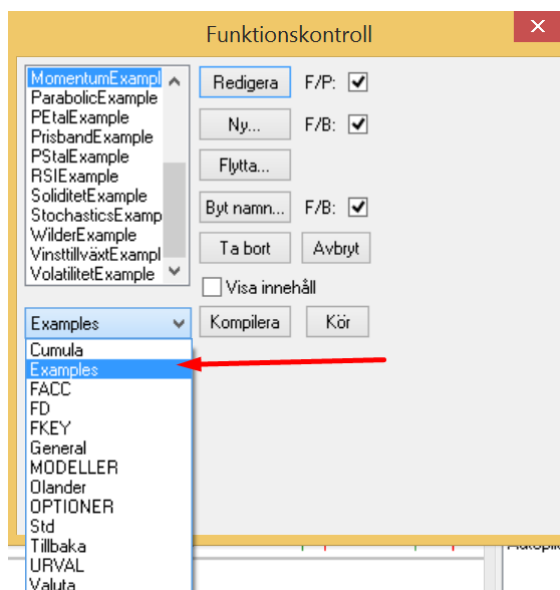
end

Datorn tror att allt som kommer före ett avslutstecken ; hänger ihop. Därför måste dina kommandon hållas isär med ;

Beskrivning av en enkel modell, momentum

Under menyn Moduler – Funktionskontroll – Example finns det några öppna exempelmodeller.





Klicka på Examples, sedan på MomentumExample och till slut Redigera.

Här nedan beskrivs **Momentumexample**

Förklaringar skrivs med kursiv stil

Vad gör modellen – en beskrivning

Momentum är skillnaden mellan ett medelvärde för slutkursen idag och samma medelvärde ett antal perioder tillbaka. Köpsignaler ges när ett medelvärde, MAV1MomentumLength långt, på momentum ändrar värde från negativt till positivt.

Säljsignaler ges när samma medelvärde på momentum ändrar värde från positivt till noll eller till negativt.

Du kan genom att välja längden på medelvärdena få momentum beräknat på enbart kursen.

Alla modeller börja med par(main:instrument; par(är inledningen på första delen av modellen som avslutas med); efter alla uppräknningar av de parametrar som du vill ha med i programmet. Här står de variabler som du kan ställa in under Modellinställningar.

var står för variabler, vilka siffror eller tillstånd vill du hitta på och använda i programmet. Du hittar själv på namnen på variablerna. En dator behöver veta vilken variant av variabel du vill använda. Är det tal med kommatecken, t.ex. 7,3, eller heltal, t.ex. 7. Tal med kommatecken kallas för real. Tal med heltal kallas för Integer. En följd, flera stycken siffror eller 0:or och ettor, kallas för en vector. Tänk dig en lång pil längs en x-axel med siffror.

```
par(main : instrument;
```

```
    var Momentum : realvector;
```

En variabel som rymmer flera värden med decimaltecken. Definitionen av variabeln kommer efter : OBS det står ett ; efter realvector.

```
    var MAV1Momentum, MAV2Momentum, MAV3Momentum : realvector;
```

Hmm, varför skriva var en gång till? Antagligen kom den som gjorde modellen på att det behövdes några fler variabler och la till dem. Alla 4 (inklusive Momentum) kunde ha stått efter var. Här är det viktigt att det är ett kommatecken, mellan variablerna. Alla variabler får typen efter : I det här fallet realvector.

Diff : integer;

Diff blir av typen heltal, inte en hel massa siffror, utan bara en siffra.

MAVPriceLength, Mav1MomentumLength, Mav2MomentumLength, Mav3MomentumLength : integer;

Ytterligare variabler av typen heltal. OBS! Deklarationen avslutas med ett ;

var BUY, SELL : booleanvector;

Vi behöver ett antal köp- och säljsignaler. En boolean är slarvigt uttryckt, en nolla eller en etta. Egentligen är det inte en siffra utan ett tillstånd, sant eller falskt. En booleanvector är flera ettor och nollor efter varandra.

var AktuellKöpålder, AktuellSäljålder: realvector;

Uppe i kolumnhuvudet vill vi se hur många perioder sedan som det var köp- eller säljsignal.

var KursdiffFrånSignal : realvector);

Uppe i kolumnhuvudet vill vi även se hur mycket kursen har gått sedan sista signalen.

OBS! här slutar parameterdelarna, det vill säga med ett) och sedan ;

var

Här börjar de interna variabel-deklarationerna med ordet var efter); ovan. De variabler som du hittar som behövs för mellanlagring inom programmet. Obs, här behövs inget (efter ordet var.

i : integer;

Man brukar ha i som en uppräknings-siffra. Det är ett heltal.

b3, b4 : booleanvector;

Du kan använda b0 till b9, tillståndsvariabler, etta eller nolla

FilledClose, MAVKurs : realvector;

FilledClose är alltid bra att ha med. Om det saknas ett värde en dag, fylls den dagen med senaste värdet. Avsluta med ;

Här börjar nästa del av programmet med ordet begin. Inget (efter ordet begin.

begin

// betyder att det som kommer efter inte har någon betydelse för datorn, dvs ignorera vad som står här. För oss människor är det mycket bra att kommentera vad du tänkte när du skrev modellen. Mycket lättare att förstå i efterhand och vad som kan återanvändas.

//Här kommer modelldefinitionen.

:= ett mycket viktigt tecken. Det betyder "tilldelas", dvs vad variabeln fylls med. FilledClose får senaste noteringen. Om det inte finns någon senast betalt för idag, tar den gårdagens värde osv. std betyder att den

letar efter funktionen FILL i standardbiblioteket. De som är riktigt duktiga kan flytta funktionen till en annan mapp och då ska den stå istället för std. FILL behöver veta vilket objekt och vilket värde som ska användas. Main innebär aktuellt objekt och CLOSE senaste noteringen. Istället för CLOSE kunde det ha stått OPEN,HIGH,LOW, VOL, men då hade du nog haft ett annat namn på variabeln som passar.

```
FilledClose := std.FILL(Main.Close);
```

```
MAVKurs := std.MAVN(FilledClose, MAVPriceLength);
```

”Funktionen MAVN ger ett glidande medelvärde FilledClose över perioden MAVPriceLength . Medelvärdet av vektorn FilledClose över perioden MAVPriceLength kommer att placeras i MAVKurs vektorn på det sista elementet som ingår i perioden MAVPriceLength”

Fler sådana här kryptiska beskrivningar som är vardagsspråk för en programmerare finns att läsa på Vikingens hemsida. www.vikingen.se . Februari 2019 låg Future-manualen (programspråket i Vikingen) under <https://www.vikingen.se/support/product-manuals/>

//Här kommer momentumdefinitionen.

```
Momentum := MAVKurs - SHIFT(MAVKurs, Diff);
```

Shift innebär att du flyttar värdet ett antal perioder tillbaka. I det här fallet ta skillnaden mellan medelvärdet nu och medelvärdet för antalet DIFF perioder tillbaka. Dvs det är definitionen på momentum. Varje defintion avslutas med ;

//Här kommer tre medelvärden på momentum.

```
MAV1Momentum := std.MAVN(Momentum, Mav1MomentumLength);
```

```
MAV2Momentum := std.MAVN(Momentum, Mav2MomentumLength);
```

```
MAV3Momentum := std.MAVN(Momentum, Mav3MomentumLength);
```

//Här läggs signalvillkoren på första medelvärdet på momentum.

b3 tilldelas tillståndet sant MAV1Momentum är större än noll. Avsluta med ;

```
b3 := MAV1Momentum > 0;
```

b4: här tilldelas b4 tvärtomtillståndet av b3. Som en förklaring har den som gjort modellen lagt in // och skrivit vad det betyder, vad är NOT b3

```
b4 := NOT b3; //b4 := MAV1Momentum <= 0;
```

Viktigt. Vi måste ha köp- och säljvillkor. Om b3 är sann blir det en köpsignal, om b4 är sann, blir det en säljsignal. De har defintionerna finns i varje modell. Fast det kan ju vara andra booleaner som b0, b1, b2, b5.... Avsluta varje rad med ;

```
BUY := std.FILTERBUY(b3, b4);
```

```
SELL := std.FILTERSELL(b3,b4);
```

//Här beräknas AktuellKöpålder och AktuellSäljålder.

Det här är en standardfunktion som alltid ska finnas om du vill använda samlingstabeller, och det vill du nog. Bara att kopiera och använda i dina modeller.

```
i := std.GetBuySellAge(BUY, SELL, AktuellKöpålder, AktuellSäljålder);
```

```
//Här beräknas kursutvecklingen från senaste signal.
```

Den här kan också vara bra att visa i dina tabeller. Hur mycket kursen har gått sedan senaste signal.

```
KursdiffFrånSignal := std.DiffFromSignal(BUY OR SELL, FilledClose);
```

OBS! Mycket viktigt. Här avslutas programmet med end; Ska alltid vara med

```
end;
```

Innan du börjar med modellen ska du trycka på ”Kompilera” om den är ny eller gjort en ändring. Du kanske har glömt något tecken. Programmet visar på vilken rad det fattas något.

Presentation av modellen

Vad vill du ha med i tabeller och diagram, vad ska visas? En funktionsmodell kan presenteras på flera olika sätt. Dvs en modell kan ha flera presentationer. Fast i praktiken brukar vi nöja oss med ett sätt.

```

var
  i : integer;
  b3, b4 : booleanvector;
  FilledClose, MAVKurs : realvector;
begin

//Här kommer modelldefinitionen.
  FilledClose := std.FILL(Main.Close);
  MAVKurs := std.MAVN(FilledClose, MAVPriceLength);

//Här kommer momentumdefinitionen.
  Momentum := MAVKurs - SHIFT(MAVKurs, Diff);

//Här kommer tre medelvärden på momentum.
  MAV1Momentum := std.MAVN(Momentum, Mav1MomentumLength);
  MAV2Momentum := std.MAVN(Momentum, Mav2MomentumLength);
  MAV3Momentum := std.MAVN(Momentum, Mav3MomentumLength);

//Här läggs signalvillkoren på första medelvärdet på momnetum.
  b3 := MAV1Momentum > 0;
  b4 := NOT b3; //b4 := MAV1Momentum <= 0;
  BUY := std.FILTERBUY(b3, b4);
  SELL := std.FILTERSELL(b3,b4);

//Här beräknas AktuellKöpålder och AktuellSäljålder.
  i := std.GetBuySellAge(BUY, SELL, AktuellKöpålder, AktuellSäljålder);

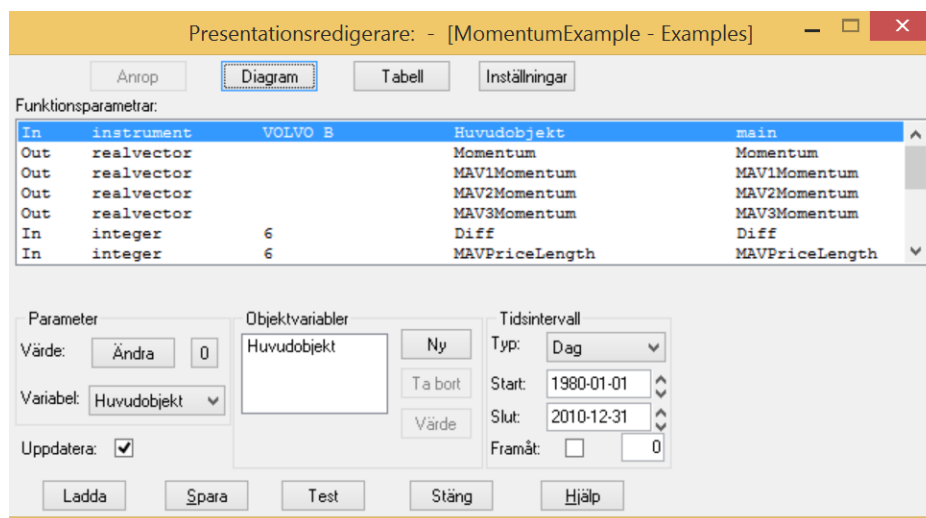
//Här beräknas kursutvecklingen från senaste signal.
  KursdiffFrånSignal := std.DiffFromSignal(BUY OR SELL, FilledClose);

end;

```



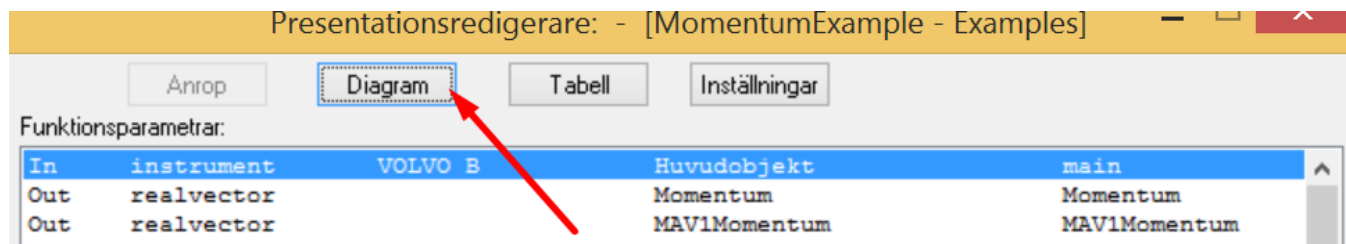
Vad är det för fel i den här bilden?



Slut har fel datum. Skriv in något stort. 2199-01-01, t.ex.

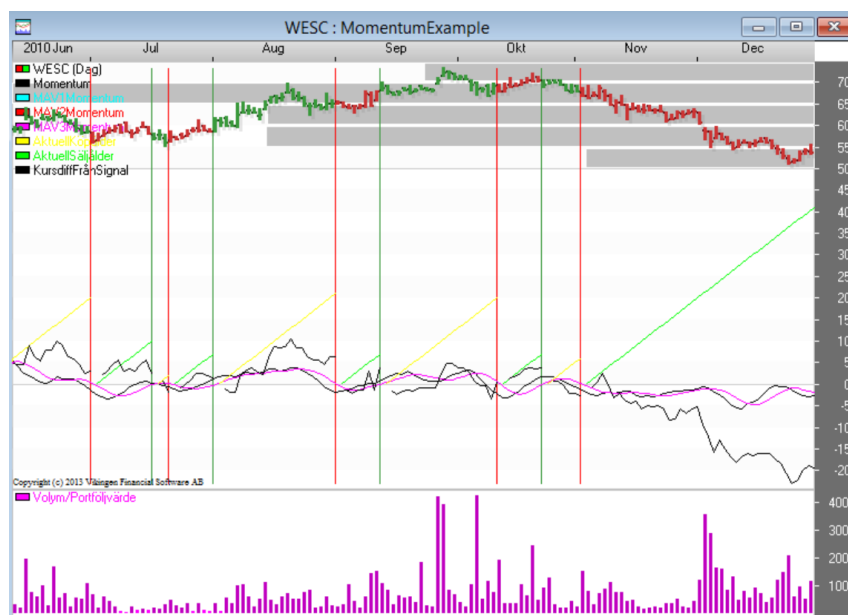
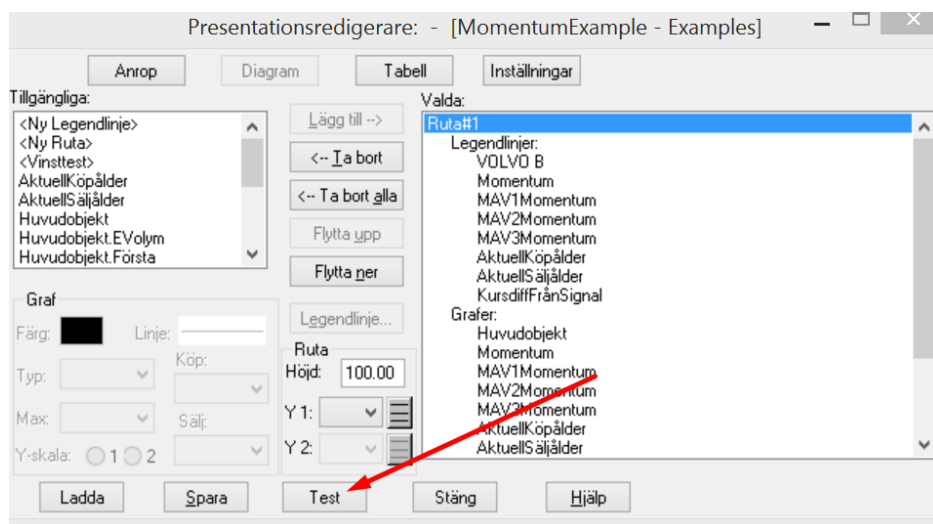
Diagram

Klicka på Diagram överst i fönstret



Hur ska diagrammet se ut, vad ska med?

Klicka på Test och kontrollera om diagrammet ser ut som det ska.



Här är det några konstiga sneda streck som vi vill ha bort. Legendlinjer beskrivs vilken variabel som har vilken färg. Under grafer står det vilka variabler som ska visas som graf. Huvudobjekt vill vi ha med, men Aktuell köpålder, Aktuell Säljålder och möjligen KursdiffFrånSignal vill vi inte ha med. De hör till tabellen.

Ta bort Aktuell köpålder, Aktuell Säljålder och möjligen KursdiffFrånSignal från Legendlinjer och graf från ruta 1.

Markera, tryck på ”Ta bort”. Tryck på Test och se om du är nöjd.

Vinsttest

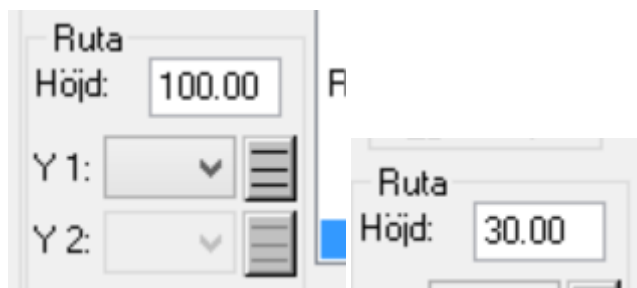
Men.. vinsttestet är inte med.. det var ju därför du gjorde modellen, för att testa dina ideer. <vinsttest> - funktionen hittar du till vänster. Dessutom behöver ett ytterligare fönster i fönstret. Lägg till ett fönster och lägg vinsttest i det.

1. Klicka på <Ny ruta> -> Lägg till -> Flytta ned.

2. Markera Grafer i din nya ruta

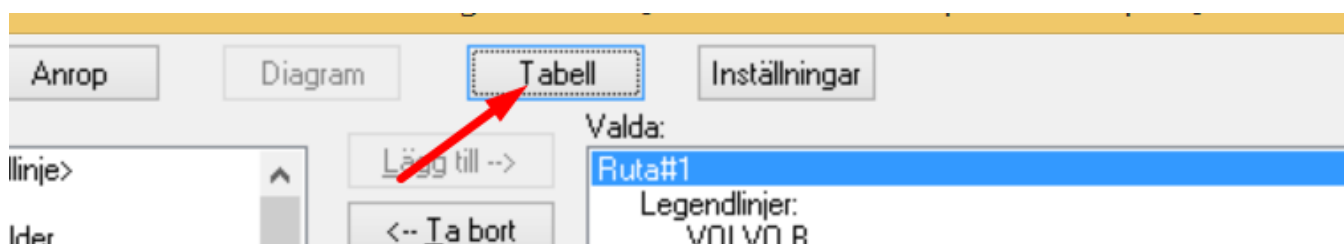
3. Klicka på <vinsttest> -> Lägg till

Klicka på test. Tog den för mycket utrymme? Ändra i så fall rutans höjd och testa igen.



Tabell

Vad ska visas samlingstabellen? Klicka på Tabell



Klicka på Test

Om du vill att Aktuell Köpålder ska visas längst till vänster, måste du flytta upp den.

Klicka på variabeln Aktuell Köpålder och sedan på Flytta Upp. Du kan också lägga till vinsttest.

Presentationsredigerare: - [MomentumExample - Examples]

Anrop Diagram Tabell Inställningar

Tillgängliga:

- BUY
- EVolym <Huvudobjekt.EVolym>
- SELL**

Valda:

- AktuellKöpålder
- AktuellSäljålder
- KursdiffFrånSignal
- Vinsttest**
- Momentum
- MAV1Momentum
- MAV2Momentum
- MAV3Momentum
- Högsta <Huvudobjekt.Högsta>
- Lägsta <Huvudobjekt.Lägsta>
- Första <Huvudobjekt.Första>
- Sista <Huvudobjekt.Sista>
- Volym <Huvudobjekt.Volym>

Kolumnrubrik:
Vinsttest

Aktivera på objektlista

Ta bort odefinierade rader.

Skapa objektlista automatiskt.

Ladda Spara Test Stäng Hjäl

Klicka på Test när du är klar. Nöjd?

En sista sak. Under Inställningar anger du vilka parametrar som du vill ha med under Modellinställningar.

Nu kan du spara resultatet. T.ex. Momentum-rensad under biblioteket Modeller.

Spara presentation

Namn: Modeller

Namn: Momentum-rensad

OK Avbryt

Tryck på OK och stäng. Nu finns din modell med i Analytikern under ”Modeller”

Cumula	AdvanceDecline	PetersSpecialare
FAGraph	ADVolum	Point And Figure
Modeller	BBandWidth	Prisvolum
Olander	Betavärden	PSVolla
Optioner	Bollingerband	PSVollamean
Urval	BollingerRSIMomentum	PSVollameannoline
-----	Börstermometern	RateOfChange
Objektlistor	Chaiken	RegressionsFörändring
Diagram	Coppock	Regressionslinje
Arbetsytor	CumulaOptima	RSI
Autopiloter	DCCI	RSIUtjämnad
	DSI	Spread
	DSIUtjämnad	SpreadSignal
	EMA	Stochastics
	Fibonacci	Tidsspread
	FibonacciNivåer	Trading
	FishNet	Trend
	FramaMV	Trendbrott
	HausseIndex	Trendlinjer
	Intradaydiagram	TrendTriggerFactor
	Kursband	Tvåobjekt
	Kursdiagram	Valuta
	Kursdiagram02	Wilder
	MACD	Volatilitet
	MACDXHST	Volatilitetssignal
	MedelvärdeFond	Volymindikatorer
	Medelvärden	Volymoscillator
	Momentum	
	Momentum02	
	Momentum-rensad	
	mRudolph3	