```
{------------------------------------------------------------------------
----------
 Copyright (c) Jonathan Kinlay 2017
------------------------------------------------------------------------
---------}

{ Strategy inputs }
Inputs: NL1 (4),             { Day of week (0 to 6 for Sunday to
Saturday), long trades }
        NS1 (3),             { Day of week (0 to 6 for Sunday to
Saturday), short trades }
        EntryPctL (3.3273),  { Value of percentage entry (stop/limit),
long trades }
        TargPctL (5.294),    { Value of percentage exit target, long
trades }
        MMStopSzL (0.05000), { Value of fixed size money management
stop per share/contract, long trades }
        NBarEnS1 (8),        { Indicator look-back length (bars),
short trades }
        NATRS (97),          { Indicator look-back length (bars),
short trades }
        EntFrS (1.1182),     { Multiple of price difference (e.g.,
ATR); entry, short trades }
        TargFrS (4.3631),    { Multiple of price difference (e.g.,
ATR); exit, short trades }
        NBarExS1 (67),       { Number of bars from entry for market
exit if profitable, short trades }
        PSParam (100000.00), { Position sizing parameter value }
        RoundPS (true),      { Round-to-nearest (true/false) }
        RoundTo (1),         { Round-to position size value }
        MinSize (1),         { Minimum allowable position size }
        SizeLimit (100000);  { Maximum allowable position size }

{ Variables for average true range for entry and exit orders }
Var:    ATRS (0);

{ Variables for entry and exit prices }
Var:    EntPrL   (0),
        EntPrS   (0),
        TargPrL  (0),
        TargPrS  (0),
        LStop    (0);

{ Variables for entry and exit conditions }
Var:    VarL1     (0),
        VarS1     (0),
        VarS2     (0),
        VarS3     (0),
        EntCondL  (false),
        EntCondS  (false),
```

```
        ExCondS   (false);

{ Variables for position sizing }
Var:    NSharesL (0),
        NSharesS (0);

{ Average true range }
ATRS = AvgTrueRange(NATRS);

{ Entry prices }
EntPrL = (1 + EntryPctL/100.0) * C;
EntPrS = TriAverage(L, NBarEnS1) - EntFrS * ATRS;

{ Entry and exit conditions }
VarL1 = DayOfWeek(date);
VarS1 = DayOfWeek(date);
VarS2 = LowD(0);
VarS3 = HighD(0);
EntCondL = false; {VarL1 <= NL1;}
EntCondS = VarS1 > NS1;
ExCondS = VarS2 >= VarS3;

{ Position sizing calculations }
NSharesL = PSParam/AbsValue(EntPrL * BigPointValue);
NSharesS = PSParam/AbsValue(EntPrS * BigPointValue);

If RoundPS and RoundTo > 0 then begin
    NSharesL = IntPortion(NSharesL/RoundTo) * RoundTo;
    NSharesS = IntPortion(NSharesS/RoundTo) * RoundTo;
end;

NSharesL = MaxList(NSharesL, MinSize);
NSharesL = MinList(NSharesL, SizeLimit);
NSharesS = MaxList(NSharesS, MinSize);
NSharesS = MinList(NSharesS, SizeLimit);

{ Entry orders }
If MarketPosition = 0 and EntCondL then begin
    Buy("EnStop-L") NSharesL shares next bar at EntPrL stop;
end;

If MarketPosition = 0 and EntCondS then begin
    Sell short("EnStop-S") NSharesS shares next bar at EntPrS stop;
end;

{ Exit orders, long trades }
If MarketPosition = 1 then begin
    If BarsSinceEntry = 0 then begin
        LStop = EntryPrice - MMStopSzL/BigPointValue;
    end;
```

```
    Sell("ExStop-L") next bar at LStop stop;

    TargPrL = (1 + TargPctL/100.0) * EntryPrice;
    Sell("ExTarg-L") next bar at TargPrL limit;
end;

{ Exit orders, short trades }
If MarketPosition = -1 then begin
    TargPrS = EntryPrice - TargFrS * TrueRange;
    Buy to cover("ExTarg-S") next bar at TargPrS limit;

    If ExCondS or (BarsSinceEntry >= NBarExS1 and C < EntryPrice) then
        Buy to cover("ExMark-S") next bar at market;
end;
```