# Use of pinescript (version 4)

# Part A Create custom indicators

### Example one : EMA60 difference

//@version=4

study("EMA60 diffence")

EMA60=ema(close,60)

EMA\_diff=(close-EMA60)

plot(EMA\_diff,color=color.black)

### Example two: Another version of EMA60 difference

//@version=4

study("EMA60 diffence")

EMA50=ema(close,60)

EMA\_diff=(close-EMA50)

length = input(title="BB length", type=input.integer, defval=20)

mult=input(title="BB stadard deviation", type=input.float, defval=2)

basis = sma(EMA\_diff, length)

dev = mult \* stdev(EMA\_diff, length)

upper = basis + dev

lower = basis - dev

plot(EMA\_diff,color=color.black)

plot(basis, "Basis", color=color.red)

plot(upper, "Upper", color=color.blue)

plot(lower, "Lower", color=color.blue)

## Example three-KD indicator

// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

// © vickslee

//@version=4

study("KD\_crossover\_test1")

length **=** input(14**,** minval**=**1)

OverBought **=** input(80)

OverSold **=** input(20)

smoothK **=** 3

smoothD **=** 3

k **=** sma(stoch(close**,** high**,** low**,** length)**,** smoothK)

d **=** sma(k**,** smoothD)

co **=** crossover(k**,**d)

cu **=** crossunder(k**,**d)

alertcondition(co**,**"KD crossoverup"**,**"KD crossover happened")

alertcondition(cu**,**"KD crossoverdown"**,**"KD crossdown happened")

plot(k**,**color**=**color.black)

plot(d**,**color**=**color.red)

Another example

//**@version=4**

study("KD\_crossover\_test1")

length = input(14, minval=1)

OverBought = input(80)

OverSold = input(20)

smoothK = 3

smoothD = 3

k = sma(stoch(close, high, low, length), smoothK)

d = sma(k, smoothD)

co = crossover(k,d) and k<=OverSold

cu = crossunder(k,d) and k>=OverBought

alertcondition(co,"KD crossoverup","KD crossover happened")

alertcondition(cu,"KD crossoverdown","KD crossdown happened")

plot(k,color=color.black)

plot(d,color=color.red)

### Class exercise

// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

// © vickslee

//@version=4

// study() is used for custom indicator and define alert conditions

study(title**=**"EMA crossover indicator"**,** shorttitle**=**"EMA\_crossover"**,** overlay**=**true)

// With input() user could change the value. We will also specify the default value and type here

EMA\_short\_len **=** input(20**,** title**=**"EMA with shorter length"**,**type**=**input.integer)

EMA\_long\_len **=** input(60**,** title**=**"EMA with longer length"**,**type**=**input.integer)

ema\_short**=**ema(close**,**EMA\_short\_len)

ema\_long**=**ema(close**,**EMA\_long\_len)

plot(ema\_short**,**color**=**color.blue)

plot(ema\_long**,**color**=**color.red)

//Here we define the crossover and crossunder condition. They are boolean series data

ema\_crossover**=**crossover(ema\_short**,**ema\_long)

ema\_crossunder**=**crossunder(ema\_short**,**ema\_long)

// similar to if ...else

ema\_crossover1**=**ema\_crossover**?**low**:**na

ema\_crossunder1**=**ema\_crossunder**?**high**:**na

plotshape(ema\_crossover1**,**text**=**'L'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.black**,** textcolor**=**color.black**,** size**=**size.small)

plotshape(ema\_crossunder1**,**text**=**'S'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.purple**,** textcolor**=**color.red**,** size**=**size.small)

// plotshape(ema\_crossover,text='L', style=shape.circle, location=location.belowbar, color=color.black, textcolor=color.black, size=size.small)

// plotshape(ema\_crossunder,text='S', style=shape.circle, location=location.abovebar, color=color.purple, textcolor=color.red, size=size.small)

alertcondition(ema\_crossover1**,**"EMA crossover"**,**"EMA crossover happened")

alertcondition(ema\_crossunder1**,**"EMA crossunder"**,**"EMA crossover happened")

Another example

// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

// © vickslee

//@version=4

study(title**=**"EMA breakout indicator"**,** shorttitle**=**"EMA\_breakout"**,** overlay**=**true)

// With input() user could change the value. We will also specify the default value and type here

EMA\_short\_len **=** input(20**,** title**=**"EMA with shorter length"**,**type**=**input.integer)

EMA\_long\_len **=** input(60**,** title**=**"EMA with longer length"**,**type**=**input.integer)

ema\_short**=**ema(close**,**EMA\_short\_len)

ema\_long**=**ema(close**,**EMA\_long\_len)

plot(ema\_short**,**color**=**color.blue)

plot(ema\_long**,**color**=**color.red)

//Here we define the crossover and crossunder condition. They are boolean series data

//ema\_crossover=crossover(ema\_short,ema\_long)

//ema\_crossunder=crossunder(ema\_short,ema\_long)

ema\_breakout\_up**=**(ema\_short**>**ema\_long) **and** (open**<**ema\_short) **and** (close**>**ema\_short)

ema\_breakout\_dn**=**(ema\_short**<**ema\_long) **and** (open**>**ema\_short) **and** (close**<**ema\_short)

// similar to if ...else

ema\_breakout\_up1**=**ema\_breakout\_up**?**low**:**na

ema\_breakout\_dn1**=**ema\_breakout\_dn**?**high**:**na

plotshape(ema\_breakout\_up1**,**text**=**'L'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.black**,** textcolor**=**color.black**,** size**=**size.small)

plotshape(ema\_breakout\_dn1**,**text**=**'S'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.purple**,** textcolor**=**color.red**,** size**=**size.small)

// plotshape(ema\_crossover,text='L', style=shape.circle, location=location.belowbar, color=color.black, textcolor=color.black, size=size.small)

// plotshape(ema\_crossunder,text='S', style=shape.circle, location=location.abovebar, color=color.purple, textcolor=color.red, size=size.small)

alertcondition(ema\_breakout\_up1**,**"EMA breakout up"**,**"EMA upward breakout happened")

alertcondition(ema\_breakout\_dn1**,**"EMA breakout dn"**,**"EMA downward breakout happened")

# Part B: Backtesting of trading strategy

// This source code is subject to the terms of the Mozilla Public License 2.0 at https://mozilla.org/MPL/2.0/

// © vickslee

//@version=4

strategy("EMA breakout strategy"**,** shorttitle**=**"EMA\_breakout"**,**overlay**=**true**,**pyramiding**=**1)

//+------------------------------------------------------------------+

//| Input parameters (Time duration)                                 |

//+------------------------------------------------------------------+

start\_year**=**input(2020**,**title**=**"start year"**,**type**=**input.integer)

end\_year**=**input(2022**,**title**=**"end year"**,**type**=**input.integer)

start\_month**=**input(1**,**title**=**"start month"**,** type**=**input.integer)

start\_day**=**input(1**,**title**=**"start day"**,**type**=**input.integer)

end\_month**=**input(12**,**title**=**"end month"**,**type**=**input.integer)

end\_day**=**input(31**,**title**=**"end day"**,**type**=**input.integer)

timeframe1**=**input("0930-2300"**,**title**=**"time frame"**,**type**=**input.string)

//timeframe2=input("1530-1630",type=input.string)

//+------------------------------------------------------------------+

//| Input parameters (Stop loss and target profit)                   |

//+------------------------------------------------------------------+

stoploss\_enable**=**input(false**,**title**=**"stop loss enable"**,**type**=**input.bool)

target\_profit\_enable**=**input(false**,**title**=**"target profit enable"**,**type**=**input.bool)

stop\_loss **=** input(100**,** title**=**"Stop loss amount")

profit **=** input(300**,** title**=**"Profit amount")

//+------------------------------------------------------------------+

//| Input parameters (MA cloud)                                      |

//+------------------------------------------------------------------+

EMA\_short\_len **=** input(20**,** title**=**"EMA with shorter length"**,**type**=**input.integer)

EMA\_long\_len **=** input(60**,** title**=**"EMA with longer length"**,**type**=**input.integer)

//+------------------------------------------------------------------+

//| Data processing                                                  |

//+------------------------------------------------------------------+

ema\_short**=**ema(close**,**EMA\_short\_len)

ema\_long**=**ema(close**,**EMA\_long\_len)

plot(ema\_short**,**color**=**color.blue)

plot(ema\_long**,**color**=**color.red)

//Here we define the crossover and crossunder condition. They are boolean series data

//ema\_crossover=crossover(ema\_short,ema\_long)

//ema\_crossunder=crossunder(ema\_short,ema\_long)

ema\_breakout\_up**=**(ema\_short**>**ema\_long) **and** (open**<**ema\_short) **and** (close**>**ema\_short)

ema\_breakout\_dn**=**(ema\_short**<**ema\_long) **and** (open**>**ema\_short) **and** (close**<**ema\_short)

// similar to if ...else

ema\_breakout\_up1**=**ema\_breakout\_up**?**low**:**na

ema\_breakout\_dn1**=**ema\_breakout\_dn**?**high**:**na

plotshape(ema\_breakout\_up1**,**text**=**'L'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.black**,** textcolor**=**color.black**,** size**=**size.small)

plotshape(ema\_breakout\_dn1**,**text**=**'S'**,** style**=**shape.circle**,** location**=**location.absolute**,** color**=**color.purple**,** textcolor**=**color.red**,** size**=**size.small)

tradeWindow **=** (time **>=** timestamp(start\_year**,** start\_month**,** start\_day**,** 0**,** 0) **and** time <=timestamp(end\_year**,**end\_month**,**end\_day**,**0**,**0) )

timesession **=** time(timeframe.period**,** timeframe1)

//+------------------------------------------------------------------+

//| Trading logic                                                    |

//+------------------------------------------------------------------+

short\_condition**=**ema\_breakout\_dn **and** tradeWindow **and** timesession

long\_condition**=**ema\_breakout\_up **and** tradeWindow **and** timesession

long\_exit\_condition**=**close**<**ema\_long

short\_exit\_condition**=**close**>**ema\_short

stop\_price\_long**=**valuewhen(long\_condition**,**low[0]**-**stop\_loss**,**0)

profit\_price\_long**=**valuewhen(long\_condition**,**high[0]**+**profit**,**0)

stop\_price\_short**=**valuewhen(short\_condition**,**high[0]**+**stop\_loss**,**0)

profit\_price\_short**=**valuewhen(short\_condition**,**low[0]**-**profit**,**0)

//trailpoint1=trail\_enable?trailpoint:na

//trailoffset1=trail\_enable?trailoffset:na

stop\_price\_long1**=**stoploss\_enable**?**stop\_price\_long**:**na

profit\_price\_long1**=**target\_profit\_enable**?**profit\_price\_long**:**na

stop\_price\_short1**=**stoploss\_enable**?**stop\_price\_short**:**na

profit\_price\_short1**=**target\_profit\_enable**?**profit\_price\_short**:**na

//+------------------------------------------------------------------+

//| Define entry and exit/close conditions                           |

//+------------------------------------------------------------------+

**if** (long\_condition)

    // Define stoploss and profit target conditions

    strategy.entry("long"**,** strategy.long)

**if** (short\_condition)

    strategy.entry("short"**,** strategy.short)

**if** (strategy.position\_size**>**0)

    strategy.close("long"**,**when**=**long\_exit\_condition)

**if** (strategy.position\_size**>**0)

    strategy.exit("exit\_long"**,**from\_entry**=**"long"**,**limit**=**profit\_price\_long1**,**stop**=**stop\_price\_long1)

**if** (strategy.position\_size**<**0)

    strategy.exit("exit\_short"**,**from\_entry**=**"short"**,**limit**=**profit\_price\_short1**,**stop**=**stop\_price\_short1)

**if** (strategy.position\_size**<**0)

    strategy.close("short"**,**when**=**short\_exit\_condition)

plot(stop\_price\_short1**,**color**=**color.red)

plot(profit\_price\_short1**,**color**=**color.blue)

plot(stop\_price\_long1**,**color**=**color.red**,**style**=**plot.style\_cross)

plot(profit\_price\_long1**,**color**=**color.blue**,**style**=**plot.style\_cross)