

## Experts em linguagem MQL4

### CorrelacaoPearson.mql

```
1. #property copyright "Copyright 2014, Cleiton Gomes"
2. #property link      "cleitoncsg@gmail.com"

3. #define TAKE_PROFIT 500
4. #define STOP_LOSS 500
5. #define ALAVANCAGEM 0.25
6. #define CORRELACAO_ACEITAVEL 0.89
7. #define SEXTA_FEIRA 5
8.
9. int ticket=0;
10. string nome = "CSG";
11. bool realizaOrdem;
12. double estado_mercado;
13.
14. int start(){
15.     bool venda, compra;
16.
17.     if(correlacao_pearson(55) > CORRELACAO_ACEITAVEL && correlacao_pearson(34) >
        CORRELACAO_ACEITAVEL &&
18.         correlacao_pearson(21) > CORRELACAO_ACEITAVEL && correlacao_pearson(13) >
        CORRELACAO_ACEITAVEL){
19.         realizaOrdem = true;
20.     }
21.
22.     if( DayOfWeek() != SEXTA_FEIRA ){
23.         if(realizaOrdem == true && estado_mercado > 0 ){
24.             compra = true;
25.         }
26.         if(realizaOrdem == true && estado_mercado < 0 ){
27.             venda = true;
28.         }
```

```

29.     }
30.
31.     if( ((compra == true && OrdersTotal() == 0)) ){
32.         RefreshRates();
33.         while (IsTradeContextBusy()) Sleep(5);
34.         ticket= OrderSend(Symbol(),OP_BUY,ALAVANCAGEM,Ask,0,Ask - TAKE_PROFIT*Point,
35.             Ask + TAKE_PROFIT*Point,nome,AccountNumber(),0,Yellow);
36.     }
37.     if( ((venda == true && OrdersTotal() == 0)) ){
38.         RefreshRates();
39.         while (IsTradeContextBusy()) Sleep(5);
40.         ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,Bid + STOP_LOSS*Point,
41.             Bid - TAKE_PROFIT*Point,nome,AccountNumber(),0,Green);
42.     }
43.     double ponto_positivo, ponto_negativo;
44.
45.     for(int j=0; j < OrdersHistoryTotal();j++){
46.         OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
47.         if(OrderSymbol()!=Symbol()) continue;
48.         if(OrderMagicNumber() != AccountNumber()) continue;
49.         if(OrderProfit() > 0){
50.             ponto_positivo++;
51.         }
52.         else{
53.             ponto_negativo++;
54.         }
55.     }
56.
57.     Comment(
58.         "Margem da Conta = ", AccountMargin() ,"\\n",
59.         "Ordens em lucro = ", ponto_positivo ,"\\n",
60.         "Ordens em prejuizo = ", ponto_negativo ,"\\n",
61.         "STOP LOSS = ", STOP_LOSS ,"\\n",
62.         "TAKE PROFIT = ", TAKE_PROFIT ,"\\n",

```

```

63.     "CORRELAÇÃO LINEAR 55 ", correlacao_pearson(55) , "\n",
64.     "CORRELAÇÃO LINEAR 34 ", correlacao_pearson(34) , "\n",
65.     "CORRELAÇÃO LINEAR 21 ", correlacao_pearson(21) , "\n",
66.     "CORRELAÇÃO LINEAR 13 ", correlacao_pearson(13) , "\n",
67.     ""
68. );
69. return(0);
70. }
71. double correlacao_pearson(int tempoCorrelacao){
72.     int c =0;
73.     double soma_ordenadas = 0;
74.     double soma_abcissas = 0;
75.     double soma_ordenadas_quadrado = 0;
76.     double soma_abcissas_quadrado = 0;
77.     double numero_abcissa;
78.     double numero_ordenada;
79.     double soma_X_vezes_Y = 0;
80.     double numerador, denominador_1,denominador, correlacao;
81.
82.     for(c=0; c<tempoCorrelacao; c++){
83.         numero_abcissa = NormalizeDouble(Open[c],5);
84.         numero_ordenada =NormalizeDouble(Close[c],5);
85.         soma_abcissas = soma_abcissas + numero_abcissa;
86.         soma_abcissas_quadrado = (soma_abcissas_quadrado) +
            (numero_abcissa)*(numero_abcissa);
87.         soma_ordenadas = soma_ordenadas + numero_ordenada;
88.         soma_ordenadas_quadrado = (soma_ordenadas_quadrado) +
            (numero_ordenada)*(numero_ordenada);
89.         soma_X_vezes_Y = soma_X_vezes_Y + (numero_ordenada*numero_abcissa);
90.     }
91.
92.     numerador
        =((tempoCorrelacao*soma_X_vezes_Y)-((soma_abcissas)*(soma_ordenadas)));

```

```

93.                                                                 denominador_1
    =((tempoCorrelacao*soma_abcissas_quadrado)-(soma_abcissas*soma_abcissas))*
94.
    ((tempoCorrelacao*soma_ordenadas_quadrado)-(soma_ordenadas*soma_ordenadas));
95.
96.         denominador = MathPow(denominador_1,1.0/2.0);
97.
98.         if(denominador != 0)
99.             correlacao = numerador/denominador;
100.        else
101.            correlacao = 0;
102.
103.        estado_mercado = soma_abcissas - soma_ordenadas;
104.
105.        return (correlacao);
106.    }

```

## Estocastico.mql

```

1.  #property copyright "Copyright 2014, Cleiton Gomes"
2.  #property link      "cleitoncsg@gmail.com"
3.
4.  #define TAKE_PROFIT 500
5.  #define STOP_LOSS 500
6.  #define ALAVANCAGEM 0.25
7.  #define TEMPO_OPERACAO 60
8.  #define SEXTA_FEIRA 5
9.  #define AJUSTE_TEMPORAL_MAIOR 5
10. #define AJUSTE_TEMPORAL_MENOR 3
11.
12. int ticket=0;
13. string nome = "CSG";
14. bool realizaOrdem;
15.
16. int start(){
17.     bool venda, compra;

```

```

18.
19.     HideTestIndicators(TRUE);
20.     RefreshRates();
21.     double estocastico
        =iStochastic(NULL,TEMPO_OPERACAO,AJUSTE_TEMPORAL_MAIOR,AJUSTE_TEMPORAL_MENOR,AJUSTE_TEMP
        ORAL_MENOR,MODE_SMA,0,MODE_MAIN,1);
22.     double sinal
        =iStochastic(NULL,TEMPO_OPERACAO,AJUSTE_TEMPORAL_MAIOR,AJUSTE_TEMPORAL_MENOR,AJUSTE_TEMP
        ORAL_MENOR,MODE_SMA,0,MODE_SIGNAL,1);
23.
24.     if( DayOfWeek() != SEXTA_FEIRA ){
25.
26.         if (estocastico > sinal){
27.             compra = true;
28.         }
29.
30.         if (estocastico < sinal){
31.             venda = true;
32.         }
33.     }
34.
35.     if( ((compra == true && OrdersTotal() == 0 )) ){
36.         RefreshRates();
37.         while (IsTradeContextBusy()) Sleep(5);
38.         ticket= OrderSend(Symbol(),OP_BUY,ALAVANCAGEM,Ask,0,Ask - TAKE_PROFIT*Point,
39.         Ask + TAKE_PROFIT*Point,nome,AccountNumber(),0,Yellow);
40.
41.     }
42.     if( ((venda == true && OrdersTotal() == 0 )) ){
43.         RefreshRates();
44.         while (IsTradeContextBusy()) Sleep(5);
45.         ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,Bid + STOP_LOSS*Point,
46.         Bid - TAKE_PROFIT*Point,nome,AccountNumber(),0,Green);
47.     }

```

```

48.
49.     double ponto_positivo, ponto_negativo;
50.
51.     for(int j=0; j < OrdersHistoryTotal();j++){
52.         OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
53.         if(OrderSymbol()!=Symbol()) continue;
54.         if(OrderMagicNumber() != AccountNumber()) continue;
55.         if(OrderProfit() > 0){
56.             ponto_positivo++;
57.         }
58.         else{
59.             ponto_negativo++;
60.         }
61.     }
62.     Comment(
63.         "Margem da Conta = ", AccountMargin() ,"\n",
64.         "Ordens em lucro = ", ponto_positivo ,"\n",
65.         "Ordens em prejuizo = ", ponto_negativo ,"\n",
66.         "STOP LOSS = ", STOP_LOSS ,"\n",
67.         "TAKE PROFIT = ", TAKE_PROFIT ,"\n",
68.         ""
69.     );
70.     return(0);
71. }

```

## Fibonacci.mql

```

1. #property link      "cleitoncsg@gmail.com"
2. #include "suporteResistencia.mq4"
3.
4. #define TAKE_PROFIT 500
5. #define STOP_LOSS 500
6. #define ALAVANCAGEM 0.25
7. #define FATOR_RETRACAO 0.38
8. #define MAX_CANDLES 34
9. #define SEXTA_FEIRA

```

```

10. #define ESTADO_VALIDO 0.01
11. double retracao_fibo;
12.
13. int ticket=0;
14. string nome = "CSG";
15.
16. int start(){
17.     bool venda;
18.
19.     if(NormalizeDouble(retracao_fibonacci()+ suporte(),4) == Bid &&
        estadoMercado(MAX_CANDLES) > ESTADO_VALIDO){
20.         venda = true;
21.     }
22.
23.     if( ((venda == true && OrdersTotal() == 0)) ){
24.         RefreshRates();
25.         while (IsTradeContextBusy()) Sleep(5);
26.
27.         ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,Bid + STOP_LOSS*Point,
28.             Bid - TAKE_PROFIT*Point,nome,AccountNumber(),0,Green);
29.     }
30.
31.     double ponto_positivo, ponto_negativo;
32.
33.     for(int j=0; j < OrdersHistoryTotal();j++){
34.         OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
35.
36.         if(OrderSymbol()!=Symbol()) continue;
37.         if(OrderMagicNumber() != AccountNumber()) continue;
38.         if(OrderProfit() > 0){
39.             ponto_positivo++;
40.         }
41.         else{
42.             ponto_negativo++;

```

```

43.         }
44.     }
45.     Comment(
46.         "Margem da Conta = ", AccountMargin() , "\n",
47.         "Ordens em lucro = ", ponto_positivo , "\n",
48.         "Ordens em prejuizo = ", ponto_negativo , "\n",
49.         "Suporte = ", suporte() , "\n",
50.         "Resistencia = ", resistencia() , "\n",
51.         "Retracao Fibo = ", retracao_fibonacci() , "\n",
52.         ""
53.     );
54.     return(0);
55. }
56. double estadoMercado(int tempoCorrelacao){
57.     double soma_ordenadas = 0, soma_abcissas = 0;
58.     double numero_abcissa, numero_ordenada;
59.
60.     for(int c=0; c<tempoCorrelacao; c++){
61.         numero_abcissa = NormalizeDouble(Open[c],5);
62.         numero_ordenada =NormalizeDouble(Close[c],5);
63.         soma_abcissas = soma_abcissas + numero_abcissa;
64.         soma_ordenadas = soma_ordenadas + numero_ordenada;
65.     }
66.     return ( soma_abcissas - soma_ordenadas);
67. }
68. double retracao_fibonacci(){
69.     double retracao = ( resistencia() - suporte())*FATOR_RETRACAO;
70.
71.     return (retracao);
72. }

```

## MediaMovel.mql

```

1. #property copyright "Copyright 2014, Cleiton Gomes"
2. #property link      "cleitoncsg@gmail.com"
3.

```



```

4. #define TAKE_PROFIT 500
5. #define STOP_LOSS 500
6. #define ALAVANCAGEM 0.25
7. #define TEMPO_OPERACAO 60
8. #define SEXTA_FEIRA 5
9.
10. extern int mediaMovelRapida = 12;
11. extern int mediaMovelLenta = 26;
12.
13. int ticket=0;
14. string nome = "CSG";
15. bool realizaOrdem;
16.
17. int start(){
18.     bool venda, compra;
19.
20.     double
        mediaMovelRapidaCorrente=iMA(NULL,TEMPO_OPERACAO,mediaMovelRapida,0,MODE_EMA,PRICE_CLOSE
        ,0);
21.     double mediaMovelLentaCorrente
        =iMA(NULL,TEMPO_OPERACAO,mediaMovelLenta,0,MODE_EMA,PRICE_CLOSE,0);
22.
23.     if( DayOfWeek() != SEXTA_FEIRA ){
24.
25.         if (mediaMovelLentaCorrente < mediaMovelRapidaCorrente){
26.             compra = true;
27.         }
28.
29.         if (mediaMovelLentaCorrente > mediaMovelRapidaCorrente){
30.             venda = true;
31.         }
32.     }
33.
34.     if( ((compra == true && OrdersTotal() == 0)) ){

```

```

35. RefreshRates();
36. while (IsTradeContextBusy()) Sleep(5);
37. ticket= OrderSend(Symbol(),OP_BUY,ALAVANCAGEM,Ask,0,Ask - TAKE_PROFIT*Point,
38. Ask + TAKE_PROFIT*Point,nome,AccountNumber(),0,Yellow);
39.
40. }
41. if( ((venda == true && OrdersTotal() == 0)) ){
42. RefreshRates();
43. while (IsTradeContextBusy()) Sleep(5);
44. ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,Bid + STOP_LOSS*Point,
45. Bid - TAKE_PROFIT*Point,nome,AccountNumber(),0,Green);
46. }
47.
48. double ponto_positivo, ponto_negativo;
49.
50. for(int j=0; j < OrdersHistoryTotal();j++){
51. OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
52. if(OrderSymbol()!=Symbol()) continue;
53. if(OrderMagicNumber() != AccountNumber()) continue;
54. if(OrderProfit() > 0){
55. ponto_positivo++;
56. }
57. else{
58. ponto_negativo++;
59. }
60. }
61.
62. Comment(
63. "Margem da Conta = ", AccountMargin() ,"\\n",
64. "Ordens em lucro = ", ponto_positivo ,"\\n",
65. "Ordens em prejuizo = ", ponto_negativo ,"\\n",
66. "STOP LOSS = ", STOP_LOSS ,"\\n",
67. "TAKE PROFIT = ", TAKE_PROFIT ,"\\n",
68. ""

```

```
69.     );  
70.  
71.     return(0);  
72. }
```

## MinimosQuadrados.mql

```
1.  #property copyright "Copyright 2014, Cleiton Gomes"  
2.  #property link      "cleitoncsg@gmail.com"  
3.  
4.  #define ALAVANCAGEM 0.25  
5.  #define QUANTIDADE_CANDLES 34  
6.  #define AJUSTE_SL 8  
7.  #define AJUSTE_CA 0.1  
8.  #define SEXTA_FEIRA 5  
9.  
10. extern double take_profit_fixo, stop_loss_fixo;  
11.  
12. int ticket=0;  
13. string nome = "CSG";  
14. double coeficienteAngular;  
15.  
16. int start(){  
17.     bool venda, compra;  
18.     double take_profit, stop_loss;  
19.     double produto_coeficienteAngular_cotacao;  
20.  
21.     produto_coeficienteAngular_cotacao = calculoCoeficienteLinear(QUANTIDADE_CANDLES);  
22.  
23.     if( DayOfWeek() != SEXTA_FEIRA ){  
24.         if(coeficienteAngular < 0){  
25.             coeficienteAngular = coeficienteAngular*(-1);  
26.         }  
27.         if(produto_coeficienteAngular_cotacao > 1){  
28.             compra = true;  
29.         }  
}
```

```

30.     if(produto_coeficienteAngular_cotacao < 0){
31.         venda = true;
32.     }
33.     take_profit = (Ask + (coeficienteAngular)*AJUSTE_CA);
34.     stop_loss = (Bid - AJUSTE_SL*(coeficienteAngular)*AJUSTE_CA);
35. }
36.
37. if( ((compra == true && OrdersTotal() == 0 && venda != true)) ){
38.     take_profit_fixo = take_profit;
39.     stop_loss_fixo = stop_loss;
40.     RefreshRates();
41.     while (IsTradeContextBusy()) Sleep(5);
42.     ticket= OrderSend(Symbol(),OP_BUY,ALAVANCAGEM,Ask,0,stop_loss_fixo,
43.         take_profit_fixo,nome,AccountNumber(),0,Yellow);
44. }
45. if( ((venda == true && OrdersTotal() == 0 && compra != true)) ){
46.     take_profit_fixo = take_profit;
47.     stop_loss_fixo = stop_loss;
48.
49.     RefreshRates();
50.     while (IsTradeContextBusy()) Sleep(5);
51.     ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,stop_loss_fixo,
52.         take_profit_fixo,nome,AccountNumber(),0,Green);
53. }
54.
55.     double ponto_positivo, ponto_negativo;
56.
57.     for(int j=0; j < OrdersHistoryTotal();j++){
58.         OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
59.         if(OrderSymbol()!=Symbol()) continue;
60.         if(OrderMagicNumber() != AccountNumber()) continue;
61.         if(OrderProfit() > 0){
62.             ponto_positivo++;
63.         }

```

```

64.         else{
65.             ponto_negativo++;
66.         }
67.     }
68.     Comment(
69.         "Margem da Conta = ", AccountMargin() ,"\n",
70.         "Ordens em lucro = ", ponto_positivo ,"\n",
71.         "Ordens em prejuizo = ", ponto_negativo ,"\n",
72.         "STOP LOSS ", stop_loss_fixo ,"\n",
73.         "TAKE PROFIT", take_profit_fixo ,"\n",
74.         "COEFICIENTE LINEAR ", calculoCoeficienteLinear(34) ,"\n",
75.         "COEFICIENTE ANGULAR ", coeficienteAngular ,"\n",
76.         ""
77.     );
78.     return(0);
79. }

80. double calculoCoeficienteLinear(int quantidadeVelas){
81.     double soma_x = 0, soma_y = 0;
82.     double numerador, denominador;
83.     double variacaoLinear;
84.     int i;
85.
86.     for(i = 1; i < quantidadeVelas; i++){
87.         soma_x = soma_x + Open[i];
88.         soma_y = soma_y + Close[i];
89.     }
90.     for(i = 1; i < quantidadeVelas; i++){
91.         numerador = Open[i]*(Close[i] - soma_x/quantidadeVelas);
92.         denominador = Close[i]*(Open[i] - soma_y/quantidadeVelas);
93.     }
94.     variacaoLinear = numerador/denominador;
95.     coeficienteAngular = soma_y/quantidadeVelas -
        (variacaoLinear*soma_x/quantidadeVelas);
96.

```

```
97.     return variacaoLinear;
98. }
```

### ResistenciaSuporte.mql

```
1.  #property copyright "Copyright 2014, Cleiton Gomes"
2.  #property link      "cleitoncsg@gmail.com"
3.
4.  #define QUANTIDADE_CANDLES 13
5.
6.  double resistencia(){
7.      double maior = -99;
8.
9.      for(int i = 0; i < QUANTIDADE_CANDLES;i++){
10.         if(Open[i] > maior)
11.             maior = Open[i];
12.     }
13.     return maior;
14. }
15. double suporte(){
16.     double menor = 99;
17.
18.     for(int i = 0; i < QUANTIDADE_CANDLES;i++){
19.         if(Close[i] < menor)
20.             menor = Close[i];
21.     }
22.     return menor;
23. }
```