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

// © TheMacroStrategist

//@version=5

// # ========================================================================= #

// # | Indicator |

// # ========================================================================= #

SCRIPT\_NAME = "Chris V5"

indicator(

title = SCRIPT\_NAME,

shorttitle = SCRIPT\_NAME,

overlay = false,

max\_lines\_count = 500,

max\_labels\_count = 500,

max\_boxes\_count = 500

)

// # ========================================================================= #

// # | Indicator |

// # ========================================================================= #

// # ========================================================================= #

// # | Inputs |

// # ========================================================================= #

i\_table\_chris\_position = input.string(defval = "Top Right", title = "Big Table Position", options=["Top Right", "Middle Right", "Bottom Right", "Top Left", "Middle Left", "Bottom Left", "Middle Center", "Top Center", "Bottom Center"], group = "Table Positions")

i\_table\_chris2\_position = input.string(defval = "Top Left", title = "Big Table Position", options=["Top Right", "Middle Right", "Bottom Right", "Top Left", "Middle Left", "Bottom Left", "Middle Center", "Top Center", "Bottom Center"], group = "Table Positions")

table\_bg\_color = input.color(defval = color.white, title = "Table Background Color", group = "Table")

table\_border\_color = input.color(defval = color.black, title = "Table Borders Color", group = "Table")

table\_border\_width = input.int(defval = 1, title = "Table Borders Width", group = "Table")

table\_frame\_color = input.color(defval = color.black, title = "Table Frame Color", group = "Table")

table\_frame\_width = input.int(defval = 3, title = "Table Frame Width", group = "Table")

sec1 = input.symbol(defval = "SPY", title = "Ticker 1", group = "Tickers")

sec2 = input.symbol(defval = "QQQ", title = "Ticker 2", group = "Tickers")

sec3 = input.symbol(defval = "IWM", title = "Ticker 3", group = "Tickers")

sec4 = input.symbol(defval = "TLT", title = "Ticker 4", group = "Tickers")

sec5 = input.symbol(defval = "SLV", title = "Ticker 5", group = "Tickers")

sec6 = input.symbol(defval = "GLD", title = "Ticker 6", group = "Tickers")

sec7 = input.symbol(defval = "USO", title = "Ticker 7", group = "Tickers")

sec8 = input.symbol(defval = "VIX", title = "Ticker 8", group = "Tickers")

sec9 = input.symbol(defval = "EURUSD", title = "Ticker 9", group = "Tickers")

sec10 = input.symbol(defval = "USDJPY", title = "Ticker 10", group = "Tickers")

kc\_length = input.int(20, minval=1, group = "KC")

kc\_mult = input.float(2.0, "Multiplier", group = "KC")

kc\_src = input.source(close, title="Source", group = "KC")

kc\_exp = input.bool(true, "Use Exponential MA", group = "KC")

kc\_BandsStyle = input.string("Average True Range", options = ["Average True Range", "True Range", "Range"], title="Bands Style", group = "KC")

kc\_atrlength = input.int(10, "ATR Length", group = "KC")

zscore\_period = input.int(20, minval=1, title = "ZScore Period", group = "ZScore")

// # ========================================================================= #

// # | Inputs |

// # ========================================================================= #

// # ========================================================================= #

// # | General |

// # ========================================================================= #

f\_security(\_sym, \_res, \_src) =>

request.security(\_sym, \_res, \_src[1], barmerge.gaps\_off, barmerge.lookahead\_on)

// # ========================================================================= #

// # | Calculations |

// # ========================================================================= #

esma(source, length)=>

s = ta.sma(source, length)

e = ta.ema(source, length)

kc\_exp ? e : s

x = 0

[CLOSE\_1, HIGH\_1, LOW\_1, TR\_1, ATR\_1, RMA\_1, STDEV1, SMA1] = request.security(sec1 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_2, HIGH\_2, LOW\_2, TR\_2, ATR\_2, RMA\_2, STDEV2, SMA2] = request.security(sec2 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_3, HIGH\_3, LOW\_3, TR\_3, ATR\_3, RMA\_3, STDEV3, SMA3] = request.security(sec3 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_4, HIGH\_4, LOW\_4, TR\_4, ATR\_4, RMA\_4, STDEV4, SMA4] = request.security(sec4 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_5, HIGH\_5, LOW\_5, TR\_5, ATR\_5, RMA\_5, STDEV5, SMA5] = request.security(sec5 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_6, HIGH\_6, LOW\_6, TR\_6, ATR\_6, RMA\_6, STDEV6, SMA6] = request.security(sec6 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_7, HIGH\_7, LOW\_7, TR\_7, ATR\_7, RMA\_7, STDEV7, SMA7] = request.security(sec7 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_8, HIGH\_8, LOW\_8, TR\_8, ATR\_8, RMA\_8, STDEV8, SMA8] = request.security(sec8 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_9, HIGH\_9, LOW\_9, TR\_9, ATR\_9, RMA\_9, STDEV9, SMA9] = request.security(sec9 , timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

[CLOSE\_10, HIGH\_10, LOW\_10, TR\_10, ATR\_10, RMA\_10, STDEV10, SMA10] = request.security(sec10, timeframe.period, [kc\_src[x], high[x], low[x], ta.tr(true)[x], ta.atr(kc\_atrlength)[x], ta.rma(high - low, kc\_length)[x], ta.stdev(close, zscore\_period)[x], ta.sma(close, zscore\_period)[x]], barmerge.gaps\_off, barmerge.lookahead\_on)

pip() => syminfo.mintick \* (syminfo.type == "forex" ? 10 : 1)

f\_kc\_symbol(\_src, \_high, \_low, \_tr, \_atr, \_rma) =>

kc\_ma = esma(\_src, kc\_length)

kc\_rangema = kc\_BandsStyle == "True Range" ? \_tr : kc\_BandsStyle == "Average True Range" ? \_atr : \_rma

upper = kc\_ma + (kc\_rangema \* kc\_mult)// \* pip())

lower = kc\_ma - (kc\_rangema \* kc\_mult)// \* pip())

[upper, lower]

[UPPER\_1, LOWER\_1] = f\_kc\_symbol(CLOSE\_1, HIGH\_1, LOW\_1, TR\_1, ATR\_1, RMA\_1)

[UPPER\_2, LOWER\_2] = f\_kc\_symbol(CLOSE\_2, HIGH\_2, LOW\_2, TR\_2, ATR\_2, RMA\_2)

[UPPER\_3, LOWER\_3] = f\_kc\_symbol(CLOSE\_3, HIGH\_3, LOW\_3, TR\_3, ATR\_3, RMA\_3)

[UPPER\_4, LOWER\_4] = f\_kc\_symbol(CLOSE\_4, HIGH\_4, LOW\_4, TR\_4, ATR\_4, RMA\_4)

[UPPER\_5, LOWER\_5] = f\_kc\_symbol(CLOSE\_5, HIGH\_5, LOW\_5, TR\_5, ATR\_5, RMA\_5)

[UPPER\_6, LOWER\_6] = f\_kc\_symbol(CLOSE\_6, HIGH\_6, LOW\_6, TR\_6, ATR\_6, RMA\_6)

[UPPER\_7, LOWER\_7] = f\_kc\_symbol(CLOSE\_7, HIGH\_7, LOW\_7, TR\_7, ATR\_7, RMA\_7)

[UPPER\_8, LOWER\_8] = f\_kc\_symbol(CLOSE\_8, HIGH\_8, LOW\_8, TR\_8, ATR\_8, RMA\_8)

[UPPER\_9, LOWER\_9] = f\_kc\_symbol(CLOSE\_9, HIGH\_9, LOW\_9, TR\_9, ATR\_9, RMA\_9)

[UPPER\_10, LOWER\_10] = f\_kc\_symbol(CLOSE\_10, HIGH\_10, LOW\_10, TR\_10, ATR\_10, RMA\_10)

[kc\_upper, kc\_lower] = f\_kc\_symbol(close, high, low, ta.tr(true), ta.atr(kc\_atrlength), ta.rma(high - low, kc\_length))

f\_zscore\_symbol(\_close, \_stdev, \_sma) =>

nRes = (\_close - \_sma) / \_stdev

ZSCORE\_1 = f\_zscore\_symbol(CLOSE\_1, STDEV1, SMA1)

ZSCORE\_2 = f\_zscore\_symbol(CLOSE\_2, STDEV2, SMA2)

ZSCORE\_3 = f\_zscore\_symbol(CLOSE\_3, STDEV3, SMA3)

ZSCORE\_4 = f\_zscore\_symbol(CLOSE\_4, STDEV4, SMA4)

ZSCORE\_5 = f\_zscore\_symbol(CLOSE\_5, STDEV5, SMA5)

ZSCORE\_6 = f\_zscore\_symbol(CLOSE\_6, STDEV6, SMA6)

ZSCORE\_7 = f\_zscore\_symbol(CLOSE\_7, STDEV7, SMA7)

ZSCORE\_8 = f\_zscore\_symbol(CLOSE\_8, STDEV8, SMA8)

ZSCORE\_9 = f\_zscore\_symbol(CLOSE\_9, STDEV9, SMA9)

ZSCORE\_10 = f\_zscore\_symbol(CLOSE\_10, STDEV10, SMA10)

[D\_CLOSE\_TABLE\_2, D\_VOLUME\_TABLE\_2, PREV\_D\_CLOSE\_TABLE\_2, PREV\_D\_VOLUME\_TABLE\_2] = request.security(syminfo.tickerid, "D", [close, volume, close[1], volume[1]], barmerge.gaps\_off, barmerge.lookahead\_on)

[W\_CLOSE\_TABLE\_2, W\_VOLUME\_TABLE\_2, PREV\_W\_CLOSE\_TABLE\_2, PREV\_W\_VOLUME\_TABLE\_2] = request.security(syminfo.tickerid, "W", [close, volume, close[1], volume[1]], barmerge.gaps\_off, barmerge.lookahead\_on)

[M\_CLOSE\_TABLE\_2, M\_VOLUME\_TABLE\_2, PREV\_M\_CLOSE\_TABLE\_2, PREV\_M\_VOLUME\_TABLE\_2] = request.security(syminfo.tickerid, "M", [close, volume, close[1], volume[1]], barmerge.gaps\_off, barmerge.lookahead\_on)

[M3\_CLOSE\_TABLE\_2, M3\_VOLUME\_TABLE\_2] = request.security(syminfo.tickerid, "3M", [close, volume], barmerge.gaps\_off, barmerge.lookahead\_on)

[M6\_CLOSE\_TABLE\_2, M6\_VOLUME\_TABLE\_2] = request.security(syminfo.tickerid, "6M", [close, volume], barmerge.gaps\_off, barmerge.lookahead\_on)

f\_variation\_rate(\_start, \_end) =>

\_res = ((\_end - \_start) / \_start) \* 100

DoD\_CLOSE\_perc = f\_variation\_rate(PREV\_D\_CLOSE\_TABLE\_2, D\_CLOSE\_TABLE\_2)

WoW\_CLOSE\_perc = f\_variation\_rate(PREV\_W\_CLOSE\_TABLE\_2, W\_CLOSE\_TABLE\_2)

MoM\_CLOSE\_perc = f\_variation\_rate(PREV\_M\_CLOSE\_TABLE\_2, M\_CLOSE\_TABLE\_2)

DoD\_VOLUME\_perc = f\_variation\_rate(PREV\_D\_VOLUME\_TABLE\_2, D\_VOLUME\_TABLE\_2)

WoW\_VOLUME\_perc = f\_variation\_rate(PREV\_W\_VOLUME\_TABLE\_2, W\_VOLUME\_TABLE\_2)

MoM\_VOLUME\_perc = f\_variation\_rate(PREV\_M\_VOLUME\_TABLE\_2, M\_VOLUME\_TABLE\_2)

// # ========================================================================= #

// # | Calculations |

// # ========================================================================= #

// # ========================================================================= #

// # | Display |

// # ========================================================================= #

table\_chris\_position = switch i\_table\_chris\_position

"Top Right" => position.top\_right

"Middle Right" => position.middle\_right

"Bottom Right" => position.bottom\_right

"Top Left" => position.top\_left

"Middle Left" => position.middle\_left

"Bottom Left" => position.bottom\_left

"Middle Center" => position.middle\_center

"Top Center" => position.top\_center

"Bottom Center" => position.bottom\_center

table\_chris2\_position = switch i\_table\_chris2\_position

"Top Right" => position.top\_right

"Middle Right" => position.middle\_right

"Bottom Right" => position.bottom\_right

"Top Left" => position.top\_left

"Middle Left" => position.middle\_left

"Bottom Left" => position.bottom\_left

"Middle Center" => position.middle\_center

"Top Center" => position.top\_center

"Bottom Center" => position.bottom\_center

var chris\_table = table.new(

position = table\_chris\_position,

columns = 5,

rows = 11,

bgcolor = table\_bg\_color,

frame\_color = table\_frame\_color,

frame\_width = table\_frame\_width,

border\_color = table\_border\_color,

border\_width = table\_border\_width

)

if barstate.islast

// Headers

table.cell(table\_id = chris\_table, column = 0, row = 0, text = "TICKER")

table.cell(table\_id = chris\_table, column = 1, row = 0, text = "SPOT")

table.cell(table\_id = chris\_table, column = 2, row = 0, text = "TOP RANGE")

table.cell(table\_id = chris\_table, column = 3, row = 0, text = "BOTTOM RANGE")

table.cell(table\_id = chris\_table, column = 4, row = 0, text = "Z-SCORE")

// Tickers

table.cell(table\_id = chris\_table, column = 0, row = 1, text = sec1)

table.cell(table\_id = chris\_table, column = 0, row = 2, text = sec2)

table.cell(table\_id = chris\_table, column = 0, row = 3, text = sec3)

table.cell(table\_id = chris\_table, column = 0, row = 4, text = sec4)

table.cell(table\_id = chris\_table, column = 0, row = 5, text = sec5)

table.cell(table\_id = chris\_table, column = 0, row = 6, text = sec6)

table.cell(table\_id = chris\_table, column = 0, row = 7, text = sec7)

table.cell(table\_id = chris\_table, column = 0, row = 8, text = sec8)

table.cell(table\_id = chris\_table, column = 0, row = 9, text = sec9)

table.cell(table\_id = chris\_table, column = 0, row = 10, text = sec10)

// Spot

table.cell(table\_id = chris\_table, column = 1, row = 1, text = str.tostring(CLOSE\_1, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 2, text = str.tostring(CLOSE\_2, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 3, text = str.tostring(CLOSE\_3, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 4, text = str.tostring(CLOSE\_4, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 5, text = str.tostring(CLOSE\_5, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 6, text = str.tostring(CLOSE\_6, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 7, text = str.tostring(CLOSE\_7, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 8, text = str.tostring(CLOSE\_8, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 9, text = str.tostring(CLOSE\_9, "#.##"))

table.cell(table\_id = chris\_table, column = 1, row = 10, text = str.tostring(CLOSE\_10, "#.##"))

// Top Range

table.cell(table\_id = chris\_table, column = 2, row = 1, text = str.tostring(UPPER\_1, "#.#####"))

table.cell(table\_id = chris\_table, column = 2, row = 2, text = str.tostring(UPPER\_2, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 3, text = str.tostring(UPPER\_3, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 4, text = str.tostring(UPPER\_4, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 5, text = str.tostring(UPPER\_5, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 6, text = str.tostring(UPPER\_6, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 7, text = str.tostring(UPPER\_7, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 8, text = str.tostring(UPPER\_8, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 9, text = str.tostring(UPPER\_9, "#.##"))

table.cell(table\_id = chris\_table, column = 2, row = 10, text = str.tostring(UPPER\_10, "#.##"))

// Bottom Range

table.cell(table\_id = chris\_table, column = 3, row = 1, text = str.tostring(LOWER\_1, "#.#####"))

table.cell(table\_id = chris\_table, column = 3, row = 2, text = str.tostring(LOWER\_2, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 3, text = str.tostring(LOWER\_3, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 4, text = str.tostring(LOWER\_4, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 5, text = str.tostring(LOWER\_5, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 6, text = str.tostring(LOWER\_6, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 7, text = str.tostring(LOWER\_7, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 8, text = str.tostring(LOWER\_8, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 9, text = str.tostring(LOWER\_9, "#.##"))

table.cell(table\_id = chris\_table, column = 3, row = 10, text = str.tostring(LOWER\_10, "#.##"))

// Z-Score

table.cell(table\_id = chris\_table, column = 4, row = 1, text = str.tostring(ZSCORE\_1, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 2, text = str.tostring(ZSCORE\_2, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 3, text = str.tostring(ZSCORE\_3, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 4, text = str.tostring(ZSCORE\_4, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 5, text = str.tostring(ZSCORE\_5, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 6, text = str.tostring(ZSCORE\_6, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 7, text = str.tostring(ZSCORE\_7, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 8, text = str.tostring(ZSCORE\_8, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 9, text = str.tostring(ZSCORE\_9, "#.##"))

table.cell(table\_id = chris\_table, column = 4, row = 10, text = str.tostring(ZSCORE\_10, "#.##"))

var chris\_table2 = table.new(

position = table\_chris2\_position,

columns = 8,

rows = 3,

bgcolor = table\_bg\_color,

frame\_color = table\_frame\_color,

frame\_width = table\_frame\_width,

border\_color = table\_border\_color,

border\_width = table\_border\_width

)

if barstate.islast

// Headers

table.cell(table\_id = chris\_table2, column = 0, row = 0, text = "")

table.cell(table\_id = chris\_table2, column = 1, row = 0, text = "1D Avg.")

table.cell(table\_id = chris\_table2, column = 2, row = 0, text = "1M Avg.")

table.cell(table\_id = chris\_table2, column = 3, row = 0, text = "3M Avg.")

table.cell(table\_id = chris\_table2, column = 4, row = 0, text = "6M Avg.")

table.cell(table\_id = chris\_table2, column = 5, row = 0, text = "DoD %")

table.cell(table\_id = chris\_table2, column = 6, row = 0, text = "WoW %")

table.cell(table\_id = chris\_table2, column = 7, row = 0, text = "MoM %")

// Rows Header

table.cell(table\_id = chris\_table2, column = 0, row = 1, text = "PRICE")

table.cell(table\_id = chris\_table2, column = 0, row = 2, text = "VOLUME")

// 1D Avg

table.cell(table\_id = chris\_table2, column = 1, row = 1, text = str.tostring(D\_CLOSE\_TABLE\_2, "#.##"))

table.cell(table\_id = chris\_table2, column = 1, row = 2, text = str.tostring(D\_VOLUME\_TABLE\_2, "#"))

// 1M Avg

table.cell(table\_id = chris\_table2, column = 2, row = 1, text = str.tostring(M\_CLOSE\_TABLE\_2, "#.##"))

table.cell(table\_id = chris\_table2, column = 2, row = 2, text = str.tostring(M\_VOLUME\_TABLE\_2, "#"))

// 3M Avg

table.cell(table\_id = chris\_table2, column = 3, row = 1, text = str.tostring(M3\_CLOSE\_TABLE\_2, "#.##"))

table.cell(table\_id = chris\_table2, column = 3, row = 2, text = str.tostring(M3\_VOLUME\_TABLE\_2, "#"))

// 6M Avg

table.cell(table\_id = chris\_table2, column = 4, row = 1, text = str.tostring(M6\_CLOSE\_TABLE\_2, "#.##"))

table.cell(table\_id = chris\_table2, column = 4, row = 2, text = str.tostring(M6\_VOLUME\_TABLE\_2, "#"))

// DoD %

table.cell(table\_id = chris\_table2, column = 5, row = 1, text = str.tostring(DoD\_CLOSE\_perc, "#.##") + "%")

table.cell(table\_id = chris\_table2, column = 5, row = 2, text = str.tostring(DoD\_VOLUME\_perc, "#.##") + "%")

// WoW %

table.cell(table\_id = chris\_table2, column = 6, row = 1, text = str.tostring(WoW\_CLOSE\_perc, "#.##") + "%")

table.cell(table\_id = chris\_table2, column = 6, row = 2, text = str.tostring(WoW\_VOLUME\_perc, "#.##") + "%")

// MoM %

table.cell(table\_id = chris\_table2, column = 7, row = 1, text = str.tostring(MoM\_CLOSE\_perc, "#.##") + "%")

table.cell(table\_id = chris\_table2, column = 7, row = 2, text = str.tostring(MoM\_VOLUME\_perc, "#.##") + "%")

// # ========================================================================= #

// # | Display |

// # ========================================================================= #