xxx-ROR-R2-S2-H-S-C-ddmmmyy

Peter B. Mandeville

10 August 2023

CHATGPT DIV VAL STOCK INDEX

Define work directory

The working directory is set to **E:/My Work/Finances**.

```
# getwd()
setwd("E:/My Work/Finances")
```

Read data.frame

Reads data from a **CSV** file **xxx-ROR-R2-H-S-ddmmmyy.csv** into **dat0** dataframe.

```
dat0 <- read.csv("POS-ROR-R2-S2-H-S-09ago23.csv", header = TRUE)
names(dat0)
```

```
##  [1] "Symbol"           "CompanyName"      "Exchange"         "LastPrice"
##  [5] "AvgVol3m"         "AvgVol10d"        "Wk52High"         "Wk52Low"
##  [9] "TrailingEPS"      "TrailingPE"       "ForwardPE"        "ForwardDivYield"
## [13] "DivShare"         "TrailingDivYield" "MarketCap"        "PriceBook"
## [17] "PriceSales"       "TargetEst1y"      "ForwardEPS"       "CurrentEPS"
## [21] "PEGRatio"         "Shares"           "CostShare"        "MarketValue"
## [25] "TotalChg"         "TotalChgPercent"  "GRO"              "FROR"
## [29] "Yahoo"            "Sector"           "Industry"         "Beta"
## [33] "Profit"           "Operating"        "ROA"              "ROE"
## [37] "DebtEquityRatio"  "CurrentRatio"     "Institutions"     "AvgDivYield5y"
## [41] "PayoutRatio"      "CurrentDivYield"  "Zacks"            "Frequency"
## [45] "DivChannel"
```

Reduce data base

Creates a new dataframe dat1 with selected columns from **dat0**.

```
dat1 <- with(dat0, data.frame(Symbol, TrailingPE, PriceBook, MarketCap, TrailingDivYield, CurrentDivYie
```

Smaller is better

Inverts selected columns in **dat1** to make **smaller is better** for those specific metrics.

```
dat1$TrailingPE <- 1 / dat1$TrailingPE
dat1$PriceBook <- 1 / dat1$PriceBook
dat1$DebtEquityRatio <- 1 / dat1$DebtEquityRatio
dat1$PriceSales <- 1 / dat1$PriceSales
dat1$PEGRatio <- 1 / dat1$PEGRatio
```

MarketCap

Converts **MarketCap** values from strings with suffix **B** or **M** to numerical values in millions or billions. Applies min-max normalization to the **MarketCap** values.

```
# Function to convert MarketCap with suffix to numerical values
convert_market_cap <- function(market_cap_with_suffix) {
  suffix <- substr(market_cap_with_suffix, nchar(market_cap_with_suffix), nchar(market_cap_with_suffix))
  value <- as.numeric(sub("[BM]", "", market_cap_with_suffix))
  if (suffix == "B") {
    return(value * 1000)
  } else if (suffix == "M") {
    return(value)
  } else {
    return(NA)
  }
}
# Convert Market Cap to numerical values in millions or billions
dat1$MarketCap <- sapply(dat1$MarketCap, convert_market_cap)
```

Normalize data (min-max normalization)

Applies min-max normalization to each column in **dat1** to scale values between **0** and **1**.

```
# Min-Max Normalization
MinMaxNor <- function(Column) {
  Min <- min(Column)
  Max <- max(Column)
  Column <- (Column - Min) / (Max - Min)
  return(Column)
}

# Larger is better
Larger <- function(Column) {
  Min <- min(Column, na.rm = TRUE)
  Column <- replace(Column, is.na(Column), Min)
  MinMaxNor(Column)
}
# Apply Min-Max Normalization to Selected Columns
selected_cols <- c("TrailingDivYield", "CurrentDivYield", "ForwardDivYield", "ROE", "ROA", "MarketCap",
dat1[, selected_cols] <- lapply(dat1[, selected_cols], Larger)

# Smaller is better
Smaller <- function(Column) {
  Max <- max(Column, na.rm = TRUE)
  Column <- replace(Column, is.na(Column), Max)
  MinMaxNor(Column)
}
# Apply Min-Max Normalization to Selected Columns
selected_cols <- c("TrailingPE", "PriceBook", "DebtEquityRatio", "PriceSales", "PEGRatio")
dat1[, selected_cols] <- lapply(dat1[, selected_cols], Smaller)
```

5 Year Average Dividend Yield

```
# Zero
Zero <- function(Column) {
  Column <- replace(Column, is.na(Column), 0)
  MinMaxNor(Column)
}
# Apply Min-Max Normalization
dat1$AvgDivYield5y <- Zero(dat1$AvgDivYield5y)
```

Weights

Applies pre-defined weights to each metric in **dat1**. The final **ChatGPT** score is calculated as a weighted sum of the individual metrics.

```
dat1$TrailingPE <- dat1$TrailingPE * 0.17
dat1$PriceBook <- dat1$PriceBook * 0.13
dat1$MarketCap <- dat1$MarketCap * 0.10
dat1$TrailingDivYield <- dat1$TrailingDivYield * 0.25
dat1$CurrentDivYield <- dat1$CurrentDivYield * 0.25
dat1$ForwardDivYield <- dat1$ForwardDivYield * 0.25
dat1$AvgDivYield5y <- dat1$AvgDivYield5y * 0.25
dat1$YieldMetrics <- (dat1$TrailingDivYield + dat1$CurrentDivYield + dat1$ForwardDivYield + dat1$AvgDiv
dat1$ROE <- dat1$ROE * 0.5
dat1$ROA <- dat1$ROA * 0.5
dat1$Management <- (dat1$ROE + dat1$ROA) * 0.09
dat1$Profit <- dat1$Profit * 0.09
dat1$DebtEquityRatio <- dat1$DebtEquityRatio * 0.07
dat1$CurrentRatio <- dat1$CurrentRatio * 0.06
dat1$PriceSales <- dat1$PriceSales * 0.06
dat1$PEGRatio <- dat1$PEGRatio * 0.05
dat1$ChatGPT <- dat1$TrailingPE + dat1$PriceBook + dat1$MarketCap + dat1$YieldMetrics + dat1$Management
dat1$ChatGPT <- round(dat1$ChatGPT, 4)
```

Sort and save

Sorts **dat1** based on **ChatGPT** score in descending order. Writes the sorted dataframe **dat2** to a new **CSV** file **POS-ChatGPT-DivValStocks-22jul23.csv**.

```
tmp <- cbind(dat1$Symbol, dat1$ChatGPT)
tmp <- data.frame(tmp)
names(tmp) <- c("Symbol", "ChatGPT")
dat2 <- tmp[order(tmp$ChatGPT, decreasing = TRUE), ]
write.csv(dat2, "POS-ROR-R2-S2-H-S-C-09ago23.csv", row.names = FALSE)
```

System Information

```
Sys.info()
```

```
##           sysname           release           version          nodename
##         "Windows"          "10 x64"     "build 19045" "DESKTOP-P8MLIQV"
##           machine             login              user    effective_user
##          "x86-64"        "mandevip"        "mandevip"        "mandevip"
```

References

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

RStudio Team. (2022). *RStudio: Integrated development environment for r*. RStudio, Inc. http://www.rstudio.com/