Module 5.1: Option Boundaries and Parities
R Commentary

See module *Ch 5.1 Valuation Option Boundaries*. The main test program is *Valuation Option Boundaries Test.R* with supporting files *Call Lower Bound.R*, *Put Lower Bound.R* and *Put Call Parity.R*. The raw data is contained in SPYOptions20171115.xlsx.

Valuation Option Boundaries Test.R (Selected Excerpts and Output)

We take one option maturity on one instrument (SPY) and analyze the empirical evidence related to actual option prices and option lower bounds and put-call parities. The SPY options are American-style. This program repeatedly uses identical code so we place these portions in a separate file. Thus the test program is relatively short. The first part of this program provide the ability to fix the axis rather than take the full range for call boundaries, put boundaries, and put-call parity.

```
# Fix call axis for lower bounds and time values
FixXRangeCLB <- TRUE
FixYRangeCLB <- TRUE
MinXRangeCLB <- 75
MaxXRangeCLB <- 125
MinYRangeCLB <- 0
MaxYRangeCLB <- 30
FixXRangeCTV <- TRUE
FixYRangeCTV <- TRUE
MinXRangeCTV <- 75
MaxXRangeCTV <- 125
MinYRangeCTV <- 0
MaxYRangeCTV <- 15
# Fix put axis for lower bounds and time values
FixXRangePLB <- TRUE
FixYRangePLB <- TRUE
MinXRangePLB <- 75
MaxXRangePLB <- 125
MinYRangePLB <- 0
MaxYRangePLB <- 30
FixXRangePTV <- TRUE
FixYRangePTV <- TRUE
MinXRangePTV <- 75
MaxXRangePTV <- 125
MinYRangePTV <- 0
MaxYRangePTV <- 15
# Put-call parity range
FixXRangePCP <- TRUE
FixYRangePCP <- TRUE
MinXRangePCP <- 75
MaxXRangePCP <- 125
MinYRangePCP <- 0
MaxYRangePCP <- 30
```

The call lower bound manipulates the input data to be able to plot the lower bound and time values reported below. Snippets of code are reported below.

```
# Call Lower Bound.R
OptionType = 1   # 1 for call, 0 for put, 2 for both
TempData <- read.xlsx(xlsxFile = FileName, sheet = 1,
  skipEmptyRows=FALSE)
LengthTempData <- length(TempData$date)
for(i in 1:LengthTempData){
  if(TempData$cp_flag[i] == "C"){
    TempData$IntrinsicValue[i] <- max(0, TempData$MidPrice[i] -
      TempData$StrikePrice[i])
    TempData$LowerBound[i] <- max(0, TempData$MidPrice[i] -
      TempData$PVDiv[i] - TempData$StrikePrice[i] *
        exp(-(TempData$Rate[i]/100)*TempData$TTM[i]))
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
```

```
      TempData$IntrinsicValue[i])
  }
  if(TempData$cp_flag[i] == "P"){
    TempData$IntrinsicValue[i] <- max(0, TempData$StrikePrice[i] -
      TempData$MidPrice[i])
    TempData$LowerBound[i] <- max(0, TempData$StrikePrice[i] *
      exp(-(TempData$Rate[i]/100)*TempData$TTM[i]) + TempData$PVDiv[i]
      - TempData$MidPrice[i])
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$IntrinsicValue[i])
  }
}
```

The R code generate the results covered previously. Also, the remainder of the R code explores options with a bit more than a quarter of a year to maturity as well as a bit more than a month to maturity. You should examine carefully these plots seeking to understand how time to expiration influences the reported results above. Data for several other dates is also provided.

Call Lower Bound.R (Selected Excerpts and Output)

Based on provided input data for a particular day, a plot related to the lower boundary as well as a plot related to the time value are produced. The file, *Put Lower Bound.R*, is very similar.

```
# Call Lower Bound.R
OptionType = 1   # 1 for call, 0 for put, 2 for both
TempData <- read.xlsx(xlsxFile = FileName, sheet = 1,
  skipEmptyRows=FALSE)
LengthTempData <- length(TempData$date)
for(i in 1:LengthTempData){
  if(TempData$cp_flag[i] == "C"){
    TempData$IntrinsicValue[i] <- max(0, TempData$MidPrice[i] -
      TempData$StrikePrice[i])
    TempData$LowerBound[i] <- max(0, TempData$MidPrice[i] -
      TempData$PVDiv[i] - TempData$StrikePrice[i] *
        exp(-(TempData$Rate[i]/100)*TempData$TTM[i]))
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$IntrinsicValue[i])
  }
  if(TempData$cp_flag[i] == "P"){
    TempData$IntrinsicValue[i] <- max(0, TempData$StrikePrice[i] -
      TempData$MidPrice[i])
    TempData$LowerBound[i] <- max(0, TempData$StrikePrice[i] *
      exp(-(TempData$Rate[i]/100)*TempData$TTM[i]) + TempData$PVDiv[i]
      - TempData$MidPrice[i])
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$IntrinsicValue[i])
  }
}
# Plot footers
TS = paste0('S=', round(TempData$MidPrice[1],2))
TT = paste0(',T=', TempData$DTM[1],' Days')
TR = paste0(',R=', round(TempData$Rate[1],2))
Td = paste0(',d=', round((TempData$PVDiv[1]/TempData$MidPrice[1]) *
  100*(365.25/TempData$DTM[1]),2))
# TD = paste0(',', format(as.Date(TempData$JDate[1], origin = "1970-01-01"),
#   "%m/%d"), "/20XX")
TD = paste0(',', format(as.Date(TempData$JDate[1], origin = "1970-01-01"),
  "%m/%d/%y"))
sTitle = paste0(TS, TT, TR, Td, TD)
# Option value plots wrt strike price
if(OptionType == 1){
  TempDataC <- TempData[TempData$cp_flag == 'C',]
  TempDataC <- TempDataC[order(TempDataC$StrikePrice),]
  xC1 <- 100*TempDataC$StrikePrice/TempDataC$MidPrice
```

```
  yCIV <- 100*TempDataC$IntrinsicValue/TempDataC$MidPrice
  yCLB <- 100*TempDataC$LowerBound/TempDataC$MidPrice
  yCOP <- 100*TempDataC$OptionPrice/TempDataC$MidPrice
  yCTV <- 100*TempDataC$TimeValue/TempDataC$MidPrice
  MaxValuey = max(yCIV, yCLB, yCOP)
  MinValuey = min(yCIV, yCLB, yCOP)
  MaxValuex = max(xC1); MinValuex = min(xC1)
  MaxValuey2 = max(yCTV); MinValuey2 = min(yCTV)
}
ylim1 = c(1:2); ylim1[1] = MinValuey; ylim1[2] = MaxValuey
xlim1 = c(1:2); xlim1[1] = MinValuex; xlim1[2] = MaxValuex
if(FixXRangeCLB){
  xlim1[1] <- MinXRangeCLB
  xlim1[2] <- MaxXRangeCLB
}
# Plot call lower bound
if(CallLB){
  if(FixYRangeCLB){
    ylim1[1] <- MinYRangeCLB
    ylim1[2] <- MaxYRangeCLB
  }
  yTitle = "Value/Stock Price"
  xTitle = "Strike Price/StockPrice"
  lTitle = "Parameter"
  mTitle = "Call Options As Percentage of Stock Price"
  legtxt = c("Call Mid Price", "Call Intrinsic Value", "ES Call Lower Bound")
  plot(xC1, yCOP, type = "b", main = mTitle,
    sub = sTitle, xlab = xTitle, ylab = yTitle, col = "black", xlim = xlim1,
    ylim = ylim1, pch = 1, cex = 0.5)
  lines(xC1, yCIV, type = "b", col ="black", xlim = xlim1,
    ylim = ylim1, pch = 2, cex = 0.5)
  lines(xC1, yCLB, type = "b", col ="black", xlim = xlim1,
    ylim = ylim1, pch = 3, cex = 0.5)
  legend("topright", legtxt, cex = 0.75, lwd = c(1, 1, 1), lty = c(1, 1, 1),
    col = c("black","black", "black"), pch = c(1, 2, 3), bty = "n",
    title = lTitle)
}
# Plot call Time Value
if(CallTV){
  ylim1 = c(1:2); ylim1[1] = MinValuey2; ylim1[2] = MaxValuey2
  if(FixXRangeCTV){
    xlim1[1] <- MinXRangeCTV
    xlim1[2] <- MaxXRangeCTV
  }
  if(FixYRangeCTV){
    ylim1[1] <- MinYRangeCTV
    ylim1[2] <- MaxYRangeCTV
  }
  yTitle = "Value/Stock Price"
  xTitle = "Strike Price/StockPrice"
  mTitle = "Call Time Value As Percentage of Stock Price"
  plot(xC1, yCTV, type = "b", main = mTitle,
    sub = sTitle, xlab = xTitle, ylab = yTitle, col = "black", xlim = xlim1,
    ylim = ylim1, pch = 1, cex = 0.5)
}
```

Put-Call Parity.R

Based on provided input data for a particular day, a plot related to the put-call parities boundaries is produced (these are American-style options).

```
# Put Call Parity.R
OptionType = 2   # 1 for call, 0 for put, 2 for both
TempData <- read.xlsx(xlsxFile = FileName, sheet = 1,
  skipEmptyRows=FALSE)
```

```
LengthTempData <- length(TempData$date)
for(i in 1:LengthTempData){
  if(TempData$cp_flag[i] == "C"){
    TempData$IntrinsicValue[i] <- max(0, TempData$MidPrice[i] -
      TempData$StrikePrice[i])
    TempData$LowerBound[i] <- max(0, TempData$MidPrice[i] -
      TempData$PVDiv[i] -
      TempData$StrikePrice[i]*exp(-(TempData$Rate[i]/100)*TempData$TTM[i]))
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$IntrinsicValue[i])
    TempData$TimeValueLB[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$LowerBound[i])
  }
  if(TempData$cp_flag[i] == "P"){
    TempData$IntrinsicValue[i] <- max(0, TempData$StrikePrice[i] -
      TempData$MidPrice[i])
    TempData$LowerBound[i] <- max(0, TempData$StrikePrice[i] *
      exp(-(TempData$Rate[i]/100)*TempData$TTM[i]) + TempData$PVDiv[i] -
      TempData$MidPrice[i])
    TempData$TimeValue[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$IntrinsicValue[i])
    TempData$TimeValueLB[i] <- max(0, TempData$MidOptionPrice[i] -
      TempData$LowerBound[i])
  }
}
CallData <- TempData[TempData$cp_flag == 'C',]
PutData <- TempData[TempData$cp_flag == 'P',]
CallData$MidCallPrice <- CallData$MidOptionPrice
CallData$BidCallPrice <- CallData$best_bid
CallData$OfferCallPrice <- CallData$best_offer
PutData$MidPutPrice <- PutData$MidOptionPrice
PutData$BidPutPrice <- PutData$best_bid
PutData$OfferPutPrice <- PutData$best_offer
CallData$CallTTM <- CallData$TTM
PutData$PutTTM <- PutData$TTM
CallData$CallRate <- CallData$Rate
PutData$PutRate <- PutData$Rate
CallData$CallPVDiv <- CallData$PVDiv
PutData$PutPVDiv <- PutData$PVDiv
CallData$CallExerciseValue <- CallData$IntrinsicValue
PutData$PutExerciseValue <- PutData$IntrinsicValue
CallData$CallLowerBound <- CallData$LowerBound
PutData$PutLowerBound <- PutData$LowerBound
CallData$CallTimeValue <- CallData$TimeValue
PutData$PutTimeValue <- PutData$TimeValue
keep <- c("StrikePrice", "MidPrice", "BidCallPrice", "MidCallPrice",
  "OfferCallPrice", "CallTTM", "CallRate", "CallPVDiv", "CallExerciseValue",
  "CallLowerBound", "CallTimeValue")
CallData <- CallData[, keep]
keep <- c("StrikePrice",  "BidPutPrice", "MidPutPrice", "OfferPutPrice",
  "PutTTM", "PutRate", "PutPVDiv", "PutExerciseValue", "PutLowerBound",
  "PutTimeValue")
PutData <- PutData[, keep]
OptionData <- merge.data.frame(CallData, PutData, by.x = "StrikePrice",
  by.y = "StrikePrice")
OptionData$DF <- exp(-(OptionData$CallRate/100.0)*OptionData$CallTTM)
TC = 0.0
OptionData$PCPUB <- 1.0 + TC - (OptionData$StrikePrice*OptionData$DF -
  OptionData$MidPutPrice)/OptionData$MidPrice
OptionData$NormalizedCallPrice <- OptionData$MidCallPrice /
  OptionData$MidPrice
OptionData$PCPLB <- 1.0 - TC - ((OptionData$CallPVDiv +
  OptionData$StrikePrice - OptionData$MidPutPrice)/OptionData$MidPrice)
OptionData$NormalizedStrikePrice <- OptionData$StrikePrice /
```

```
  OptionData$MidPrice
# Plot footers
TS = paste0('S=', round(TempData$MidPrice[1],2))
TT = paste0(',T=', TempData$DTM[1],' Days')
TR = paste0(',R=', round(TempData$Rate[1],2))
Td = paste0(',d=', round((TempData$PVDiv[1]/TempData$MidPrice[1]) *
  100*(365.25/TempData$DTM[1]),2))
# TD = paste0(',', format(as.Date(TempData$JDate[1], origin = "1970-01-01"),
#   "%m/%d"), "/20XX")
TD = paste0(',', format(as.Date(TempData$JDate[1], origin = "1970-01-01"),
  "%m/%d/%y"))
sTitle = paste0(TS, TT, TR, Td, TD)
x1 <- OptionData$NormalizedStrikePrice*100
y1 <- OptionData$PCPLB*100
y2 <- OptionData$NormalizedCallPrice*100
y3 <- OptionData$PCPUB*100
MaxValuey = max(y1, y2, y3); MinValuey = min(y1, y2, y3)
MaxValuex = max(x1); MinValuex = min(x1)
ylim1 = c(1:2); ylim1[1] = MinValuey; ylim1[2] = MaxValuey
xlim1 = c(1:2); xlim1[1] = MinValuex; xlim1[2] = MaxValuex
if(FixXRangePCP){
  xlim1[1] <- MinXRangePCP
  xlim1[2] <- MaxXRangePCP
}
if(FixYRangePCP){
  ylim1[1] <- MinYRangePCP
  ylim1[2] <- MaxYRangePCP
}
yTitle = "Normalized PCP Bounds"
xTitle = "Strike Price/StockPrice"
lTitle = "Parameter"
mTitle = "Put Call Parity Bounds"
legtxt = c("PCP Lower Bound", "Normalized Call", "PCP Upper Bound")
plot(x1, y1, type = "b", main = mTitle, sub = sTitle, xlab = xTitle,
  ylab = yTitle, col = "black", xlim = xlim1, ylim = ylim1, pch = 1,
  cex = 0.5)
lines(x1, y2, type = "b", col ="black", xlim = xlim1, ylim = ylim1, pch = 2,
  cex = 0.5)
lines(x1, y3, type = "b", col ="black", xlim = xlim1, ylim = ylim1, pch = 3,
  cex = 0.5)
legend("topright", legtxt, cex = 0.75, lwd = c(1, 1, 1), lty = c(1, 1, 1),
  col = c("black","black", "black"), pch = c(1, 2, 3), bty = "n",
  title = lTitle)
```