# SOS Calculations

*Mathieu Bray*

*April 25, 2017*

```
library(plyr)
library(dplyr)
library(tidyr)
library(rvest)
library(lubridate)
```

```
# Read in USCHO schedule
url.schedule <- "http://www.uscho.com/scoreboard/division-i-men/2016-2017/composite-schedule/"

raw.schedule <- url.schedule %>% # Scrape USCHO schedule
  read_html %>%
  html_nodes('.comp') %>%
  html_table(header=F, fill=F) %>%
  data.frame(stringsAsFactors=F)
```

```
# Add column names
colnames(raw.schedule) <- c("Day","Blank","Date","Time","Away","AwayScore","at",
                            "Home","HomeScore","OT","Notes","GameType","Box","Recap","TV")

head(raw.schedule) # Display preview of raw schedule
```

```
##    Day Blank      Date     Time                 Away AwayScore at
## 1 Fri.    NA 9/30/2016  7:07 AT         Simon Fraser         1  @
## 2 Sat.    NA 10/1/2016  5:07 ET   Alabama-Huntsville         2  @
## 3 Sat.    NA 10/1/2016  7:05 ET      Army West Point         2  @
## 4 Sat.    NA 10/1/2016  7:07 CT   (17) Michigan Tech         0  @
## 5 Sat.    NA 10/1/2016 12:00 ET              Windsor         2  @
## 6 Sat.    NA 10/1/2016  7:05 ET Prince Edward Island         2  @
##                    Home HomeScore OT Notes GameType Box Recap TV
## 1      Alaska-Anchorage         6                EX Box
## 2     (20) Ferris State         1                WC Box
## 3               Colgate         2 OT             NC Box
## 4  (6) Minnesota-Duluth         6                NC Box
## 5    (14) Bowling Green         3                EX Box
## 6 (4) Boston University        10                EX Box
```

```
# Function to strip out team rankings from team name in schedule
clean.team.names <- function(team.name){

  new.strings <- unlist(strsplit(team.name,split=') ')) # Split strings with rankings in parantheses
                                                        # eg. (1) Michigan
  new.strings <- new.strings[substring(new.strings,1,1)!="("] # Remove rankings in parantheses

  return(new.strings)
}

# Team names
teams <- c("Air Force", "Alabama-Huntsville", "Alaska", "Alaska-Anchorage",
```

```
           "American International", "Arizona State", "Army West Point", "Bemidji State",
           "Bentley", "Boston College", "Boston University", "Bowling Green",
           "Brown", "Canisius", "Clarkson", "Colgate", "Colorado College",
           "Connecticut", "Cornell", "Dartmouth", "Denver", "Ferris State",
           "Harvard", "Holy Cross", "Lake Superior", "Maine", "Massachusetts",
           "Massachusetts-Lowell", "Mercyhurst", "Merrimack", "Miami", "Michigan",
           "Michigan State", "Michigan Tech", "Minnesota", "Minnesota-Duluth",
           "Minnesota State", "Nebraska-Omaha", "New Hampshire", "Niagara",
           "North Dakota", "Northeastern", "Northern Michigan", "Notre Dame",
           "Ohio State", "Penn State", "Princeton", "Providence", "Quinnipiac",
           "Rensselaer", "RIT", "Robert Morris", "Sacred Heart", "St. Cloud State",
           "St. Lawrence", "Union", "Vermont", "Western Michigan", "Wisconsin", "Yale")

# Function to determine whether a team has won, lost, or tied
win <- function(home,homescore,awayscore){

  if (is.na(homescore) | is.na(awayscore)){
    return(as.character(NA))
  }

  result  <-  "Win"

  if(home){
    if (homescore > awayscore){
      result  <-  "Win"
    } else if (homescore < awayscore){
      result  <-  "Loss"
    } else {
      result  <-  "Tie"
    }
  } else {
    if (awayscore > homescore){
      result  <-  "Win"
    } else if (awayscore < homescore){
      result  <-  "Loss"
    } else {
      result  <-  "Tie"
    }
  }

  return(result)
}


# Clean schedule

clean.schedule <- raw.schedule %>%
  select(Away,AwayScore,at,Home,HomeScore,GameType) %>% # Keep relevant columns
  rowwise() %>%
  mutate(Home = clean.team.names(Home),                    # Strip rankings from home team name
         Away = clean.team.names(Away)) %>%                # Strip rankings from away team name
  filter(GameType != "EX", Home %in% teams, Away %in% teams) %>% # Ignore exhibition games
  mutate(HomeWin = win(T,HomeScore,AwayScore)) %>% # Determine whether home team won, lost or tied
```

```
  mutate(AwayWin = win(F,HomeScore,AwayScore)) %>% # Determine whether away team won, lost or tied
  select(-GameType) # Drop irrelevant columns

# Remove last three games of NCAA tournament (USCHO calculations have not factored these games in)
clean.schedule <- clean.schedule[1:(nrow(clean.schedule)-3),]

head(clean.schedule)
```

```
## # A tibble: 6 × 7
##                 Away AwayScore    at             Home HomeScore HomeWin
##                <chr>     <int> <chr>            <chr>     <int>   <chr>
## 1 Alabama-Huntsville         2     @     Ferris State         1    Loss
## 2    Army West Point         2     @          Colgate         2     Tie
## 3      Michigan Tech         0     @ Minnesota-Duluth         6     Win
## 4 Alabama-Huntsville         4     @     Ferris State         3    Loss
## 5      Michigan Tech         3     @ Minnesota-Duluth         4     Win
## 6   Western Michigan         5     @     Ferris State         3    Loss
## # ... with 1 more variables: AwayWin <chr>
```

```
# Split each game into two lines for each team, listing their outcome and whether they were playing
# at home, away, or at a neutral site

get.team.results <- function(schedule){

  # The 'at' column has either an '@' symbol for a regular game or a 'vs.' for a neutral site game
  if (schedule$at=="@"){
    status1 <- "Home"
    status2 <- "Away"
  } else {
    status1 <- "Neutral"
    status2 <- "Neutral"
  }

  home.result <- data.frame(Team=schedule$Home,Win=schedule$HomeWin,
                            Status=status1,Opponent=schedule$Away,
                            stringsAsFactors=F)
  away.result <- data.frame(Team=schedule$Away,Win=schedule$AwayWin,
                            Status=status2,Opponent=schedule$Home,
                            stringsAsFactors=F)

  results <- rbind(home.result,away.result)

  return(results)

}

results <- clean.schedule %>%
  rowwise() %>%
  do(get.team.results(.)) %>%
  ungroup()

head(results)
```

```
## # A tibble: 6 × 4
##                 Team   Win Status           Opponent
```

```
##                <chr> <chr>  <chr>              <chr>
## 1       Ferris State  Loss   Home Alabama-Huntsville
## 2 Alabama-Huntsville   Win   Away       Ferris State
## 3            Colgate   Tie   Home    Army West Point
## 4    Army West Point   Tie   Away            Colgate
## 5   Minnesota-Duluth   Win   Home      Michigan Tech
## 6      Michigan Tech  Loss   Away   Minnesota-Duluth
```

```
# Calculate each team's record, including their adjusted winning percentages...

record <- results %>%
  group_by(Team) %>%                            # For each team...
  summarize(Wins=sum(Win=="Win"),               # ... count the number of wins, losses, ties, etc.
            Losses=sum(Win=="Loss"),
            Ties=sum(Win=="Tie"),
            HomeWins=sum(Win=="Win" & Status=="Home"),
            HomeLosses=sum(Win=="Loss" & Status=="Home"),
            HomeTies=sum(Win=="Tie" & Status=="Home"),
            AwayWins=sum(Win=="Win" & Status=="Away"),
            AwayLosses=sum(Win=="Loss" & Status=="Away"),
            AwayTies=sum(Win=="Tie" & Status=="Away"),
            NeutralWins=sum(Win=="Win" & Status=="Neutral"),
            NeutralLosses=sum(Win=="Loss" & Status=="Neutral"),
            NeutralTies=sum(Win=="Tie" & Status=="Neutral")) %>%
  # Simple Win Pct. Adjusted Win Pct is below, using weights for each type of game
  mutate(Pct=(Wins+0.5*Ties)/(Wins+Losses+Ties),
         AdjPct=(1.2*AwayWins + 1*NeutralWins + 0.8*HomeWins +
                   0.6*AwayTies + 0.5*NeutralTies +0.4*HomeTies) /
                  ((1.2*AwayWins + 0.8*HomeWins + NeutralWins +
                    0.8*AwayLosses + NeutralLosses + 1.2*HomeLosses + Ties))) %>%
  arrange(Team) %>%
  mutate(AdjPctRound=round(AdjPct,4)) %>%
  select(Team,Wins,Losses,Ties,Pct,AdjPct,AdjPctRound)
```

```
# Compare to the adjusted win percentage from USCHO
url <- 'http://www.uscho.com/rankings/rpi/d-i-men'

uscho <- url %>% # Scrape USCHO data
  read_html%>%
  html_nodes('table') %>%
  html_table(header=T, fill=F) %>%
  data.frame(stringsAsFactors = F) %>%
  rename(AdjPct = Win.., QWBAdjustedRPI = QWB.Adj.RPI, AdjustedRPI = Adj.RPI) %>%
  rowwise() %>%
  mutate(Team = ifelse(Team == "Army","Army West Point",
                       ifelse(Team=="Omaha","Nebraska-Omaha",Team))) %>%
  ungroup() %>%
  select(Team,AdjPct,SOS,RPI) %>%
  arrange(Team)
```

```
head(uscho)
```

```
##                     Team AdjPct    SOS    RPI
## 1              Air Force 0.7071 0.4930 0.5466
## 2     Alabama-Huntsville 0.3257 0.4595 0.4260
```

```
## 3                 Alaska 0.3989 0.4683 0.4509
## 4       Alaska-Anchorage 0.2917 0.4696 0.4251
## 5 American International 0.3444 0.4645 0.4345
## 6          Arizona State 0.3401 0.5073 0.4655
```

```r
sum(uscho$AdjPct == record$AdjPctRound)  # = 60 if all 60 teams come out with the same value
```

```
## [1] 60
```

```r
# So far so good!


# Function that determines the weighted value of a game

weight <- function(win,status){

  if(win=="Win"){
    if (status=="Home"){
      val <- 0.8
    } else if (status=="Away"){
      val <- 1.2
    } else {
      val <- 1
    }
  } else if(win=="Loss") {
    if (status=="Home"){
      val <- 1.2
    } else if (status=="Away"){
      val <- 0.8
    } else {
      val <- 1
    }
  } else {
    val <- 1
  }
}


# Function to calculate the Opponent's Winning Percentage (OWP) for a team

OWP <- function(team,results){

  # Gather all opponents of our team in question, 'team'
  team_opponents <- unique((results %>% filter(Team==team))$Opponent)

  # Get opponent's record and adjusted winning percentage in games not involving 'team'
  opponents_record <- results %>%
    # Consider all games played by 'team's opponents, except those involving 'team'
    filter(Team %in% team_opponents & Opponent != team) %>%
    group_by(Team) %>%
    summarize(Wins=sum(Win=="Win"),  # Again, count number of wins, losses, ties, etc. by each team
              Losses=sum(Win=="Loss"),
              Ties=sum(Win=="Tie"),
              HomeWins=sum(Win=="Win" & Status=="Home"),
              HomeLosses=sum(Win=="Loss" & Status=="Home"),
```

```
              HomeTies=sum(Win=="Tie" & Status=="Home"),
              AwayWins=sum(Win=="Win" & Status=="Away"),
              AwayLosses=sum(Win=="Loss" & Status=="Away"),
              AwayTies=sum(Win=="Tie" & Status=="Away"),
              NeutralWins=sum(Win=="Win" & Status=="Neutral"),
              NeutralLosses=sum(Win=="Loss" & Status=="Neutral"),
              NeutralTies=sum(Win=="Tie" & Status=="Neutral")) %>%
    mutate(AdjPct=(AwayWins+NeutralWins+HomeWins+
                     0.5*AwayTies+0.5*NeutralTies+0.5*HomeTies) /
                    ((AwayWins + HomeWins + NeutralWins +
                        AwayLosses + NeutralLosses + HomeLosses + Ties))) %>%
    select(Team,AdjPct) # Calculate regular adjusted winning percentage

  # Now, assign the correct weight to each game, and take the weighted average of the
  # opponents adjusted winning percentages over each game
  opponent_winpct <- results %>%
    filter(Team == team) %>% # Only consider games involving 'team'
    inner_join(opponents_record,by=c("Opponent"="Team")) %>% # Merge in the opponent winning percentage
    rowwise() %>%
    mutate(Weight = weight(Win,Status)) # Assign weight to each game

  # Take the weighted average as the OWP
  opponent_winpct <- weighted.mean(opponent_winpct$AdjPct,opponent_winpct$Weight)

  return(opponent_winpct)
}

# Function to calculate the Opponent's Opponents Winning Percentage (OOWP) for a team
OOWP <- function(team,results,oppwinpcts){

  # Gather all opponents of our team in question, 'team'
  team_opponents <- unique((results %>% filter(Team==team))$Opponent)

  # For each game, assigne the correct weight and take the weighted average of the OWP as the OOWP
  opp_opp_winpct <- results %>%
    filter(Team == team) %>% # Only consider games involving 'team'
    inner_join(oppwinpcts,by=c("Opponent"="Team")) %>% # Merge in the OWP values for each opponent
    rowwise() %>%
    mutate(Weight = weight(Win,Status)) # Assign weight to each game

  # Take the weighted average as the OOWP
  opp_opp_winpct <- weighted.mean(opp_opp_winpct$OppWinPct,opp_opp_winpct$Weight)

  return(opp_opp_winpct)
}


# Get the OWP for each team, and paste to our record table
opp.win.pct <- sapply(record$Team,OWP,results=results)
record$OppWinPct <- opp.win.pct

head(record %>% select(Team,OppWinPct))
```

```
## # A tibble: 6 × 2
```

```
##                     Team OppWinPct
##                    <chr>     <dbl>
## 1             Air Force 0.5035642
## 2     Alabama-Huntsville 0.4343648
## 3                 Alaska 0.4663376
## 4       Alaska-Anchorage 0.4612387
## 5 American International 0.4408302
## 6          Arizona State 0.5081378
```

```
# Isolate the OWP values for each team
owps <- record %>% select(Team, OppWinPct)

# Get the OOWP for each team, and paste to our record table
opp.opp.win.pct <- sapply(record$Team,OOWP,results=results,oppwinpcts=owps)
record$OppOppWinPct <- opp.opp.win.pct

head(record %>% select(Team,OppOppWinPct))
```

```
## # A tibble: 6 × 2
##                     Team OppOppWinPct
##                    <chr>        <dbl>
## 1             Air Force    0.4863056
## 2     Alabama-Huntsville    0.4672630
## 3                 Alaska    0.4668487
## 4       Alaska-Anchorage    0.4713731
## 5 American International    0.4721252
## 6          Arizona State    0.5076280
```

```
# Calculate Strength of Schedule (SOS) and RPI
rpi.table  <-  record %>%
  mutate(SOS = (21*OppWinPct+54*OppOppWinPct)/75, # Strength of Schedule
         RPI = 0.25*AdjPct+0.21*OppWinPct+0.54*OppOppWinPct) %>% # RPI
  arrange(Team)

head(rpi.table %>% select(Team,SOS,RPI))
```

```
## # A tibble: 6 × 3
##                     Team       SOS       RPI
##                    <chr>     <dbl>     <dbl>
## 1             Air Force 0.4911380 0.5451392
## 2     Alabama-Huntsville 0.4580515 0.4249672
## 3                 Alaska 0.4667056 0.4497560
## 4       Alaska-Anchorage 0.4685355 0.4243183
## 5 American International 0.4633626 0.4336330
## 6          Arizona State 0.5077708 0.4658621
```

```
# Function to calculate correlation between values on USCHO website and our
# calculated Values
r.function <- function(x,y){
  return(paste0("R^2: ",round(summary(lm(y ~ x))$r.squared,4)))
}

r.function(rpi.table$AdjPct,uscho$AdjPct) # = 1, Perfect Agreement for Adjusted Win Pct
```

```
## [1] "R^2: 1"
```

```r
r.function(rpi.table$SOS,uscho$SOS) # = 0.9986, Near-Perfect Agreement for SOS, but not equal
```

```
## [1] "R^2: 0.9986"
```

```r
r.function(rpi.table$RPI,uscho$RPI) # = 0.9998, Near-Perfect Agreement for RPI, but not equal
```

```
## [1] "R^2: 0.9998"
```