Learning Goals

1. How to perform a head-to-head comparison between two candidates.
2. Definition of a Condorcet winner and a Weak Condorcet winner.
3. How to identify the Condorcet (Weak Condorcet) winner(s) if they exist.
4. Comparing the results of different systems of deciding on a winner/ranking (social choice) of condidates.
5. Condorcet completion processes, Nanson's method and Copeland's method.

Topic 7: Head-To-Head Comparisons, Condorcet Winner.

If we know the preference rankings of the voters, we can compare the votes of any two candidates to see which one would win in a plurality election in the absence of the other candidates. Such a comparison is called a **Head-To-Head** comparison of the two candidates.

Example There are 3 candidates for the position of President for the Notre Dame Squash Club. The preferences of the five members of the selection committee are shown in the following table:

Presidential preference rankings

#Voters	1	2	1	1
Colley	1	2	2	3
Henry	2	1	3	2
Taylor	3	3	1	1

(a) Which candidate would win in a head-to-head comparison between Colley and Henry?

(b) Which candidate would win in a head-to-head comparison between Colley and Taylor?

(c) Which candidate would win in a head-to-head comparison between Henry and Taylor?

(d) We can summarize the results of our head to head comparisons in a table (similar to that used for the results of a round robin). In the table below;

- For $i \neq j$ put a 1 in the (i, j) position if the candidate in row i wins in a head to head comparison with the candidate in column j.

- For $i \neq j$ put a -1 in the (i, j) position if the candidate in row i loses in a head to head comparison with the candidate in column j.
- For $i \neq j$ put a 0 in the (i, j) position if the candidate in row i draws in a head to head comparison with the candidate in column j.
- Put zeros in the (i, i) positions on the main diagonal.

	Colley	Henry	Taylor
Colley	0		
Henry		0	
Taylor			0

In the above example, Henry beat all of the other candidates in a head-to-head comparison. Such a candidate is an example of a Condorcet winner, named after the Marquis de Condorcet (1743-1794).

Condorcet Winner A candidate who is the winner of a head-to-head comparison with every other candidate is called a **Condorcet winner**. If a candidate beats or ties with every other candidate in a head-to-head comparison, that candidate is called a **weak Condorcet winner**. For any election, there may or may not be a Condorcet winner or a weak Condorcet winner.

Example In the example above, Henry is a Condorcet winner.

We can **shorten our calculations** to find a Condorcet winner. We can begin with two strong candidates and compare them. If one wins, then we can eliminate the loser of that comparison from future calculations. We compare the winner to a third candidate, eliminate the loser from our calculations and compare the winner to a fourth candidate, etc, etc, etc..... This reduces the number of calculations significantly. We can also write a program to have the computer check whether there is a Condorcet winner or not for us (see end of lecture).

Example Check that there is no Condorcet winner in the election for which the preferences of the voters are shown below.

#Voters	**1**	**1**	**1**
Alpha	1	3	2
Beta	2	1	3
Gamma	3	2	1

In the following example, we see that a Condorcet winner may not even make it to the runoff in a plurality election with a runoff.

Example In a poll to determine player of the year for the North Bend Tennis Club, the ranking preferences of the 100 voters are shown below.

#Voters	**9**	**28**	**11**	**23**	**23**	**6**	**R1**	**R2**
Federer	1	1	2	3	2	3		
Williams	2	3	1	1	3	2		
Nadal	3	2	3	2	1	1		

(a) Calculate the winner of an Instant Runoff election.

(b) Is there a Condorcet Winner?

In the following example, we see that a Condorcet winner may not be the winner in a Borda count (average rankings).

Example (Purely Fictitious) The following teams participated in 11 events in the Celtic Games 2009: Ireland, Wales, Scotland, Brittany and Cornwall. The rankings of the teams in the events and the number of events in which each given ranking occurred are shown in the table below. It remains for the tournament organizers to determine which team should be declared the winner of the tournament (Who should be Number 1?)

#Events	**2**	**1**	**2**	**1**	**1**	**4**
Brittany	1	3	5	5	5	5
Cornwall	2	1	1	3	3	4
Ireland	3	2	2	1	2	3
Scotland	4	4	3	2	1	1
Wales	5	5	4	4	4	2

(a) Who wins with the Borda method?

(b) Who wins a Plurality of the vote?

(c) Who wins using the plurality method with a runoff between first and second place winners?

(d) Which option, if any, is the Condorcet winner?

Notes

1. We have seen above that the Condorcet winner may not win under the Borda Count method and may not even make the runoff if the plurality with runoff method is used. Many argue that if a Condorcet winner exists, then that candidate should be declared the winner. (Think of our round robin tournament).

2. For three candidates, with voters voting randomly, the probability that there is a Condorcet winner is always above 90%. As the number of candidates (and the number of voters) increases , the probability of a Condorcet winner decreases; for six candidates it is always above 68%. In reality, there are often strong and weak candidates and voting is not random, this increases the likelihood of a Condorcet winner.

Condorcet Completion Process

In a contest where the Condorcet winner, if there is one, emerges victorious, a back-up procedure is necessary for a situation where no Condorcet winner emerges. A number of methods are available.

Black 1963 If no Condorcet winner exists, use Borda's method to decide the election.

Nanson
Round 1. : Use Borda count to eliminate all candidates with below average Borda count (= above average average rank).
Round 2. : Modify the original ranking of the voters to create new rankings for the remaining candidates. Calculate new Borda Counts (averages) and eliminate those with below average Borda count(above average average rank).
Round 3: Continue in this way until there is one candidate left.
Note if there is a Condorcet winner, that candidate will also be the winner using Nanson's method.

Copeland's method This is a simple method often used to decide a Round Robin Tournament, where every candidate plays every other candidate exactly once. With Copeland's method, candidates are ordered by the number of pairwise victories minus the number of pairwise defeats. This method is easily understood and easy to calculate, however it often leads to ties and puts more emphasis on the number of victories and defeats rather than their magnitude.

Method of Pairwise Comparisons (Equivalent to Copelands method) As with Copeland's method, candidates are compared in pairs. The one who wins gets one point for a win from such a comparison, 1/2 point for a tie and 0 points for a loss. The points from each pairwise comparison are added and the winner is the candidate with the most points.

Example Use Copeland's method and Nanson's method to decide the winner of the Nation's Cup in the Celtic games 2010, the result of which are shown below (there were only ten events in the 2010 games),

#Events	**3**	**3**	**2**	**1**	**1**	Ave.
Brittany	5	1	3	5	3	3.2
Cornwall	3	4	1	4	1	2.8
Ireland	1	2	4	2	5	2.4
Scotland	2	3	5	3	4	3.2
Wales	4	5	2	1	2	3.4
					Ave.	3

Copeland's Method In the following table, we will record a 1 if the row player wins in a head-to-head comparison between the row and column player, we will record a -1 if the column player wins and a 0 if the caparison is a draw. The sum of each row will give us W-L for the row player.

#**Teams**	Brittany	**Cornwall**	**Ireland**	**Scotland**	**Wales**	**W – L**
Brittany	–					
Cornwall		–				
Ireland			–			
Scotland				–		
Wales					–	

Condorcet winner, GOAT

Annette Pilkington

2/10/2019

Import data from GOAT file

```
GOAT<- read.csv(file="GOAT.csv", header=FALSE, sep=",")
head(GOAT)
```

```
##                V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
## 1 Michael Phelps  7  1  4  4  9  1  3  5   6   3   7   7   1   3   3
## 2     Usain Bolt  3  2  3  3  5  2  9  1   5   1   8   6   4   1   6
## 3   LeBron James  2  6  5  7  3  3  6  4   1   9   4   5   2   8   2
## 4      Tom Brady  4  5 12  1  2  5  1  6   3   8   1   1   3  11   8
## 5 Michael Jordan  5  7  6  2  1  4  2  3   4   2   2   2   5   5   1
## 6   Muhammad Ali  6  8  2  8  8 11  7  2   2   7   5   3   9   2   4
```

We want to create a matrix where the names of the rows and columns are the names of the candidates. We store the names in a vector. We have a small technicality to overcome (which happens a lot when cleaning data). Unfortunately, R has classified the data in column 1 of the GOAT file as a factor variable.
We need to convert to a character vector, hence the "as.vector" command below (We'll learn more about this later, when we study variables in more detail).

```
candidates<-as.vector(GOAT[,1])
head(candidates)
```

```
## [1] "Michael Phelps" "Usain Bolt"     "LeBron James"   "Tom Brady"
## [5] "Michael Jordan" "Muhammad Ali"
```

We want to create an N by N matrix where N is the number of candidates. We will will study these in greater detail in the next section. We start with a matrix that has 0's everywhere.

```
N<-length(candidates)
N
```

```
## [1] 12
```

```
WL<-mat.or.vec(N,N)
WL
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [3,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     0
```

Now we add the names to the rows and columns

```r
dimnames(WL)<-list(candidates,candidates)
WL
```

```
##                   Michael Phelps Usain Bolt LeBron James Tom Brady
## Michael Phelps                 0          0            0         0
## Usain Bolt                     0          0            0         0
## LeBron James                   0          0            0         0
## Tom Brady                      0          0            0         0
## Michael Jordan                 0          0            0         0
## Muhammad Ali                   0          0            0         0
## Ian Book                       0          0            0         0
## Serena Williams                0          0            0         0
## Lindsey Vonn                   0          0            0         0
## Tiger Woods                    0          0            0         0
## Joe Montana                    0          0            0         0
## Melisaa Gonzalez               0          0            0         0
##                   Michael Jordan Muhammad Ali Ian Book Serena Williams
## Michael Phelps                 0            0        0               0
## Usain Bolt                     0            0        0               0
## LeBron James                   0            0        0               0
## Tom Brady                      0            0        0               0
## Michael Jordan                 0            0        0               0
## Muhammad Ali                   0            0        0               0
## Ian Book                       0            0        0               0
## Serena Williams                0            0        0               0
## Lindsey Vonn                   0            0        0               0
## Tiger Woods                    0            0        0               0
## Joe Montana                    0            0        0               0
## Melisaa Gonzalez               0            0        0               0
##                   Lindsey Vonn Tiger Woods Joe Montana Melisaa Gonzalez
## Michael Phelps               0           0           0                0
## Usain Bolt                   0           0           0                0
## LeBron James                 0           0           0                0
## Tom Brady                    0           0           0                0
## Michael Jordan               0           0           0                0
## Muhammad Ali                 0           0           0                0
## Ian Book                     0           0           0                0
## Serena Williams              0           0           0                0
## Lindsey Vonn                 0           0           0                0
## Tiger Woods                  0           0           0                0
## Joe Montana                  0           0           0                0
## Melisaa Gonzalez             0           0           0                0
```

Now we want to calculate the winner in a head to head comparison between every possible pair of candidates and fill in a 1 in the (i,j) position if cand. i beats cand. j, a -1 if cand.j beats cand. i and 0 if it is a draw. For that we need to create a function of our own that comapares two rows

We'll get to the details of creating functions later, but this one is called "compare" and it carries out a head to head comparison between two rows of the matrix. The function called `compare` makes a head to head comparison between two candidates. Its arguments `x` and `y` will be the data in ther rows of the data frame `GOAT` corresponding to the candidates (the input is thus two vectors `x` and `y`). For the two rows `x` and `y`, we create an intermediate vector whose ith entry in the number of voters in column i if those voters prefer the candidate corresponding to row `x` to the other candidate, otherwise the ith entry of `wx` is 0. Clearly the sum of the entries in the vector `wx` is the number of voters who favour the candidate corresponding to row `x` over the other candidate.

The result of applying the compare funtion to the two rows `x` and `y` is 1 if `x` won and 0 otherwise.

```
compare<-function(x,y){
  wx<-mat.or.vec(1,length(x))
  for(i in 1:length(x)){
    if(x[i]<y[i]){
      wx[i]<-1}
    else{
      wx[i]<-0}
  }
    s<-sum(wx)
    if(s>length(x)-s){
      nw<-1}
  else{nw<-0}
  nw
  }
```

lets test the function on two rows. Lets run a head to head comparison between Michael Phelps (Row 1) and Usain Bolt (Row 2).

```
compare(GOAT[1,2:16],GOAT[2,2:16])
```

```
## [1] 0
```

```
compare(GOAT[2,2:16],GOAT[1,2:16])
```

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

We can now use the function `compare` to fill in the matrix `WL` with a 1 in the (i,j) position if athlete *i* beat athlete *j* and a 0 otherwise. This requires 66 comparisons, best done on computer (in the blink of an eye).

```
for(i in 1:length(candidates)){
  for(j in 1:length(candidates)){
    WL[i,j]<-compare(GOAT[i,2:16],GOAT[j,2:16])
  }
}
```

lets have a look at the new matrix `WL`

```
WL
```

```
##                  Michael Phelps Usain Bolt LeBron James Tom Brady
## Michael Phelps                0          0            1         1
## Usain Bolt                    1          0            0         1
## LeBron James                  0          1            0         1
## Tom Brady                     0          0            0         0
## Michael Jordan                1          0            1         0
## Muhammad Ali                  1          0            0         0
## Ian Book                      0          0            0         0
## Serena Williams               0          0            0         0
## Lindsey Vonn                  0          0            0         0
## Tiger Woods                   0          0            0         0
## Joe Montana                   0          0            0         0
## Melisaa Gonzalez              0          0            0         0
##                  Michael Jordan Muhammad Ali Ian Book Serena Williams
## Michael Phelps                0            0        1               1
## Usain Bolt                    1            1        1               1
## LeBron James                  0            1        1               1
## Tom Brady                     1            1        1               1
## Michael Jordan                0            1        1               1
## Muhammad Ali                  0            0        1               1
## Ian Book                      0            0        0               0
## Serena Williams               0            0        1               0
## Lindsey Vonn                  0            0        1               0
## Tiger Woods                   0            0        1               0
## Joe Montana                   0            0        1               0
## Melisaa Gonzalez              0            0        1               0
##                  Lindsey Vonn Tiger Woods Joe Montana Melisaa Gonzalez
## Michael Phelps              1           1           1                1
## Usain Bolt                  1           1           1                1
## LeBron James                1           1           1                1
## Tom Brady                   1           1           1                1
## Michael Jordan              1           1           1                1
## Muhammad Ali                1           1           1                1
## Ian Book                    0           0           0                0
## Serena Williams             1           1           1                1
## Lindsey Vonn                0           0           0                1
## Tiger Woods                 1           0           0                1
## Joe Montana                 1           1           0                1
## Melisaa Gonzalez            0           0           0                0
```

If a row sums to 15, then that candidate has 15 wins and is a Condorcet winner. Otherwise a Condorcet winner does not exist.

```r
GOAT$wins<-apply(WL,1,sum)
condorcet<-function(n){
  if(n==(length(candidates)-1)){
    "Condorcet Winner"}
  else{"False"}
}

GOAT$condorcet_winner<-sapply(GOAT$wins,condorcet)
result<-data.frame(candidates,GOAT$wins,GOAT$condorcet_winner)
result
```

```
##          candidates GOAT.wins GOAT.condorcet_winner
## 1    Michael Phelps         8                 False
## 2        Usain Bolt        10                 False
## 3      LeBron James         9                 False
## 4         Tom Brady         8                 False
## 5    Michael Jordan         9                 False
## 6      Muhammad Ali         7                 False
## 7          Ian Book         0                 False
## 8   Serena Williams         5                 False
## 9      Lindsey Vonn         2                 False
## 10      Tiger Woods         3                 False
## 11      Joe Montana         4                 False
## 12 Melisaa Gonzalez         1                 False
```

We have no Condorcet winner!