

## AO19 Final: events, deciders and winners

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#I read and prepare the events table.
event = pd.read_csv('/home/...../events.csv')
event = event[['rallyid', 'hitter', 'isservice', 'serve', 'type', 'stroke', 'time']]
event
```

```
# I check all the shots by type.
event.groupby('type').count()[['rallyid']]
```

```
# Strokes by hitter.
df = event[event['hitter'] == 'Nadal']
strokes = df.groupby('stroke').count()[['rallyid']]
strokes
```

```
# I merge the events and the rallies tables and I keep the columns needed.
rally = pd.read_csv('/home/...../rallies.csv')
event_rally = pd.merge(event, rally, on='rallyid', how='outer')
event_rally = event_rally[['rallyid', 'hitter', 'isservice', 'serve_x', 'type',
'stroke', 'winner', 'reason']]
event_rally
```

```
# I keep only the last rows of the events to get data on decider shots.
deciders = event_rally.groupby('rallyid').last()
deciders
```

```
# I get the types of winner shots hit by Djokovic, then
# the types of errors hit by Djokovic, then
# the types of winners hit by Nadal, then
# the types of errors hit by Nadal.
df = deciders[(deciders['hitter'] == 'Djokovic') & (deciders['winner'] ==
'Djokovic')]
types = df.groupby('type').count()[['hitter']]
types
```

```
# Forehand winners and errors then the backhand winners and errors by each player.
df = deciders[(deciders['hitter'] == 'Nadal') & (deciders['winner'] == 'Nadal') &
(deciders['stroke'] == 'backhand')]
strokes = df.groupby('stroke').count()[['hitter']]
strokes
```

```
# Number of volleys.
event[event['type'] == 'volley']
```

```
# I read the point table, I merge it with the event table.
point = pd.read_csv('/home/...../points.csv')
event_point = pd.merge(event, point, on='rallyid', how='outer')
event_point = event_point[['rallyid', 'hitter', 'isserve', 'serve_x', 'type',
'stroke', 'server', 'winner', 'reason', 'strokes', 'score']]
event_point
```

```
# I list all the winning shots as winners.
winners = event_point.groupby('rallyid').last()
winners = winners.dropna()
winners
```

```
# I list all the winners excluding serves.
winners = winners[winners['isserve'] == False]
winners
```

```
# Shot types when Djokovic lost the points.
djoko_loses = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Nadal')]
djoko_loses_num = djoko_loses.groupby('type').size()
djoko_loses_num
```

```
# Winner volleys
volley_points = winners[winners['type'] == 'volley']
volley_points
```

```
# Strikes when Nadal won the points.
nadal_wins = winners[(winners['winner'] == 'Nadal') & (winners['hitter'] ==
'Nadal')]
nadal_wins_num = nadal_wins.groupby('stroke').size()
nadal_wins_num
```

```
# Strikes when Djokovic lost the points.
nadal_wins = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Nadal')]
nadal_wins_num = nadal_wins.groupby('stroke').size()
nadal_wins_num
```

```
# Strikes when Nadal lost the points.
djoko_wins = winners[(winners['hitter'] == 'Nadal') & (winners['winner'] ==
'Djokovic')]
djoko_wins_num = djoko_wins.groupby('stroke').size()
djoko_wins_num
```

```
# Shot types when Djokovic won the points.
djoko_wins = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Djokovic')]
djoko_wins_num = djoko_wins.groupby('type').size()
djoko_wins_num
```

```
# Shot types when Nadal won the points.
nadal_wins = winners[(winners['hitter'] == 'Nadal') & (winners['winner'] ==
'Nadal')]
nadal_wins_num = nadal_wins.groupby('type').size()
nadal_wins_num
```

```
# Shot types when Nadal lost the points.
nadal_loses = winners[(winners['hitter'] == 'Nadal') & (winners['winner'] ==
'Djokovic')]
nadal_loses_num = nadal_loses.groupby('type').size()
nadal_loses_num
```

```
# Strikes when Djokovic won the points.
djoko_loses = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Nadal')]
djoko_loses_num = djoko_loses.groupby('stroke').size()
djoko_loses_num
```

```
# Strikes when players lost the point by hitting the ball into the net.
djoko_loses = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Nadal') & (winners['reason'] == 'net')]
djoko_loses_num = djoko_loses.groupby('stroke').size()
djoko_loses_num
```

```
# Strikes when players lost the point by hitting the ball out of the court.
djoko_loses = winners[(winners['hitter'] == 'Djokovic') & (winners['winner'] ==
'Nadal') & (winners['reason'] == 'out')]
djoko_loses_num = djoko_loses.groupby('stroke').size()
djoko_loses_num
```

```
# I list the most frequent reason of losing a point.
most_frequent_reason = winners.groupby('strokes')['reason'].agg(lambda x: x.mode()
[0])
```

```
# Print the result
print(most_frequent_reason)
```

```
# I list the winner of the most points by number of shots
most_frequent_reason = winners.groupby('strokes')['winner'].agg(lambda x: x.mode()
[0])
# Print the result
print(most_frequent_reason)
```

```
# I draw a bar chart of total points won by players
djokovic_total_points = winners[winners['winner'] == 'Djokovic']['score'].count()
nadal_total_points = winners[winners['winner'] == 'Nadal']['score'].count()

# Plot
plt.figure(figsize=(8, 6))
plt.bar(['Djokovic', 'Nadal'], [djokovic_total_points, nadal_total_points],
color=['blue', 'red'])
plt.xlabel('Player')
plt.ylabel('Total Points Won')
plt.title('Total Points Won by Each Player Across All Matches')
plt.show()
```