Understanding the Importance of First Serve in Tennis with Data Science

Andrea Cazzaro

Abstract

Having a good serve is an essential aspect for every professional tennis player, but can we correlate a good first serve to an excellent performance? This study analyzes the effects of first serve on the overall performance of ATP players. The analysis was conducted by considering the results of every match played in the ATP circuit in 2019. The dataset was made available by Tennis Abstract, an outstanding website that provides statistics on several tennis matches.

Through correlation analyses, the study found that hitting a good percentage of serves within the service box is not enough for ATP players. Indeed, the best performers in the circuit maximize their chance of winning a point when serving by balancing the probability of hitting a successful serve and the probability of winning the point when hitting a successful serve. In addition, even though the probability of winning a point with a successful second serve is not correlated to performance, the second serve remains crucial to maximize the chance to win the point when serving.

Motivation

Tennis is a dynamic and complex sport. There are several shots involved in a single point, but only one of them is played without the opponent's influence: the serve. Indeed, the serve gives players the chance to start the point with a concrete advantage. Having a good serve is an essential aspect for every professional player, but can we correlate a good first serve to an excellent performance? Tennis players may have heard the say "You are only as good as your first serve". Is this true?

The insights of this study can help tennis players to fully understand the importance of their first serve with probabilistic logic. Having a good serve may not be enough to excel in the ATP circuit. Indeed, a good serve must be effective and efficient in order to maximize the chance of winning serving games.

Dataset(s)

Data used for the analysis has been gathered through Tennis Abstract, an outstanding website (founded by Jeff Sackmann) that offers thousands of datasets on tennis matches. For this analysis, the dataset on ATP matches played in 2019 was used. The dataset includes data on every match played in 126 tournaments, including Davis Cup matches, for a total of 2.782 matches.

The important columns in this dataset are the ones that show for each player and each match: the number of serve points played, the number of first serves in, the number of points won with the first serve, the number of points won with the second serve and the number of double faults.

Data Preparation and Cleaning

The dataset was outstanding and only a few rows were missing data. Rows with NAs have been deleted from the dataset. In addition, players with less than 10 wins in the circuit have been left out from the analysis, leaving a total of 89 heads in the dataset.

All rows have been grouped by the name of each tennis player in order to create summative statistics for the whole year. Since all figures represented the discrete number of points won and/or serves that landed within the service box, a new table with percentages of success had to be created (ex. First serves that land within the service box divided by all serves played gives the percentage of successful first serves). In addition, the performance of each player was calculated in a separate column by summing the number of wins.

Research Question(s)

Can we correlate a good first serve to an excellent performance in the ATP circuit?

Are the best performers in the ATP circuit able to maximize the probability of winning the point on their first serve?

Is a good second serve as important as a good first serve to the performance of an ATP player?

Are the best performers in the ATP circuit able to maximize the probability of winning the point when serving (both on first and second serve)?

Methods

In order to identify a correlation between first/second serve and performance, correlation analyses were conducted. Scatter plots were used to visualize the data, while Person's r calculations were used to identify potential correlations between the variables. In addition, a visual comparison of violin plots between the distribution of first and second serves was utilized to determine how best players make a difference with their first serve.

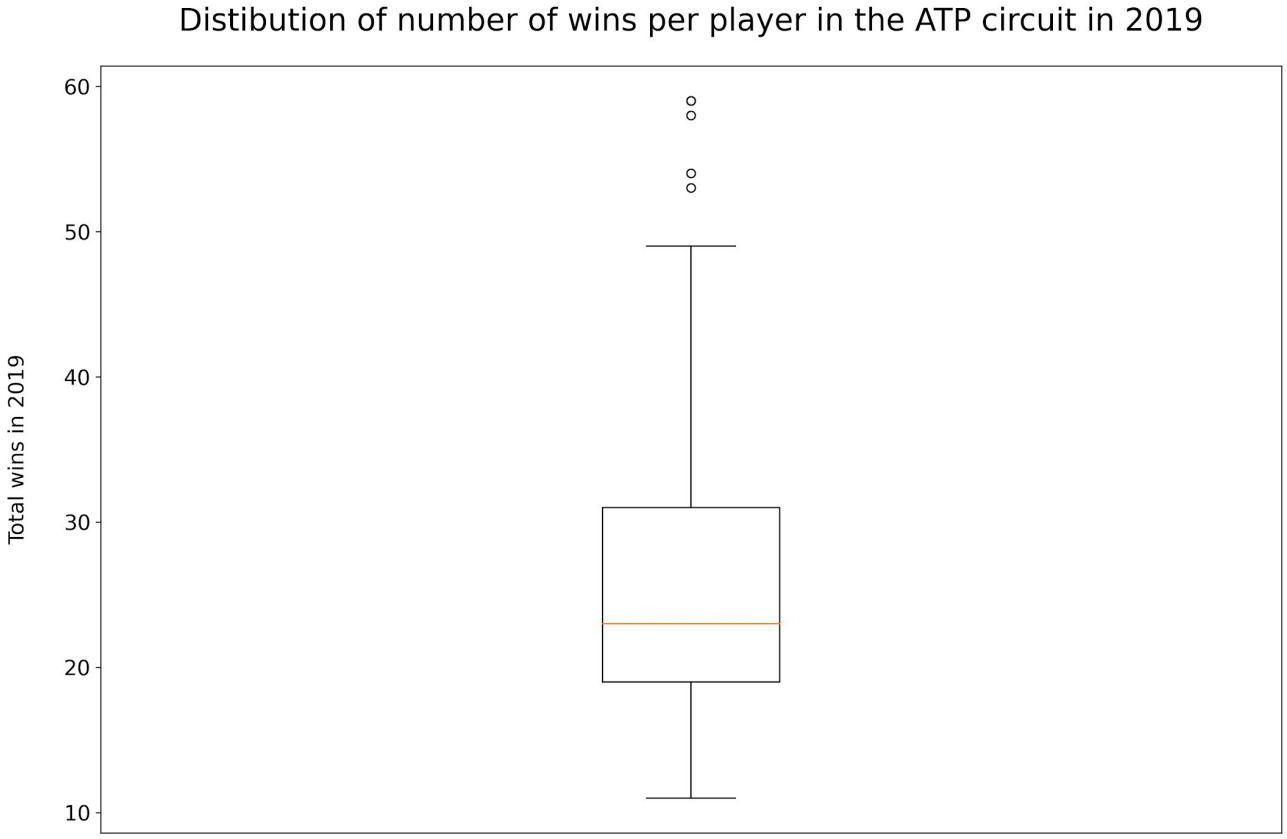
Findings

Performance

By counting the number of wins for each player to determine their performance, it was found that the dominance in the circuit is dictated by a handful of athletes. The top 10 performers won at least 40 matches, while the top 5 performers won at least 53 matches. By looking at the overall performance of the best 5 players, it was concluded that these athletes win 30% more matches than the bottom 5 players of the top 10 ranking.

In figure 1, the distribution of wins per player confirms that the best 5 performers are outliers, which means that their performance is way above the performance of other players.

Figure 1


Distibution of number of wins per player in the ATP circuit in 2019
Total wins in 2019
60
50
40
30
20
10

Findings

Successful first serves and percentage of points won

The percentage of successful first serves (serves that land within the service box) is considered an important aspect by tennis coaches. However, the correlation between successful first serves and performance is non-existent (Figure 2).

In addition, the ability of a player to win points on successful first serves is not correlated to performance (Figure 3). Hence, hitting serves within the service box or winning points on successful first serves is not enough for ATP players to excel in the circuit.

Figure 2

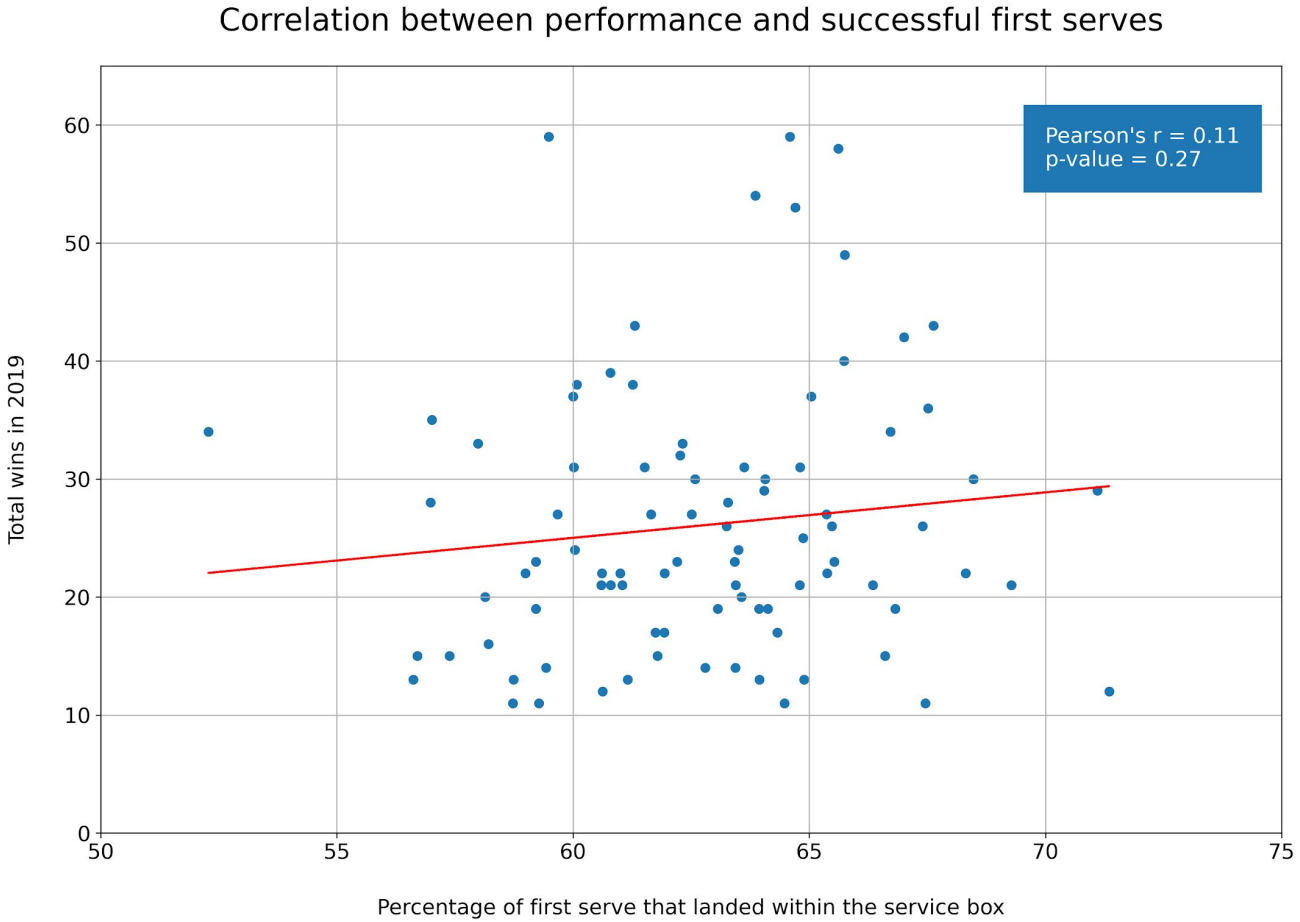
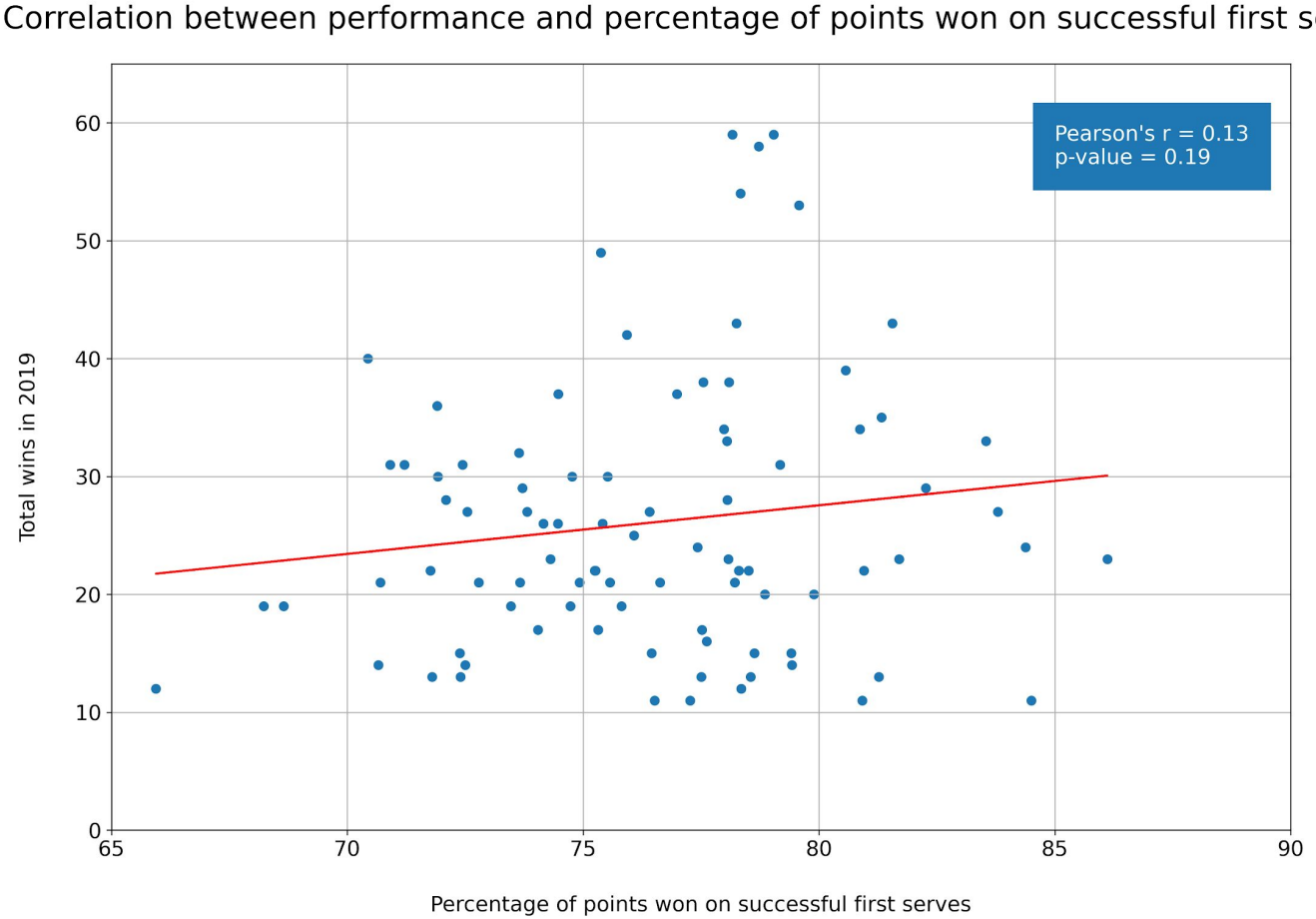

Correlation between performance and successful first serves
Pearson's r = 0.11
p-value = 0.27
Total wins in 2019
Percentage of first serve that landed within the service box
0
10
20
30
40
50
60
50
55
60
65
70
75

Figure 3


Correlation between performance and percentage of points won on successful first serves
Pearson's r = 0.13
p-value = 0.19
Total wins in 2019
Percentage of points won on successful first serves
0
10
20
30
40
50
60
65
70
75
80
85
90

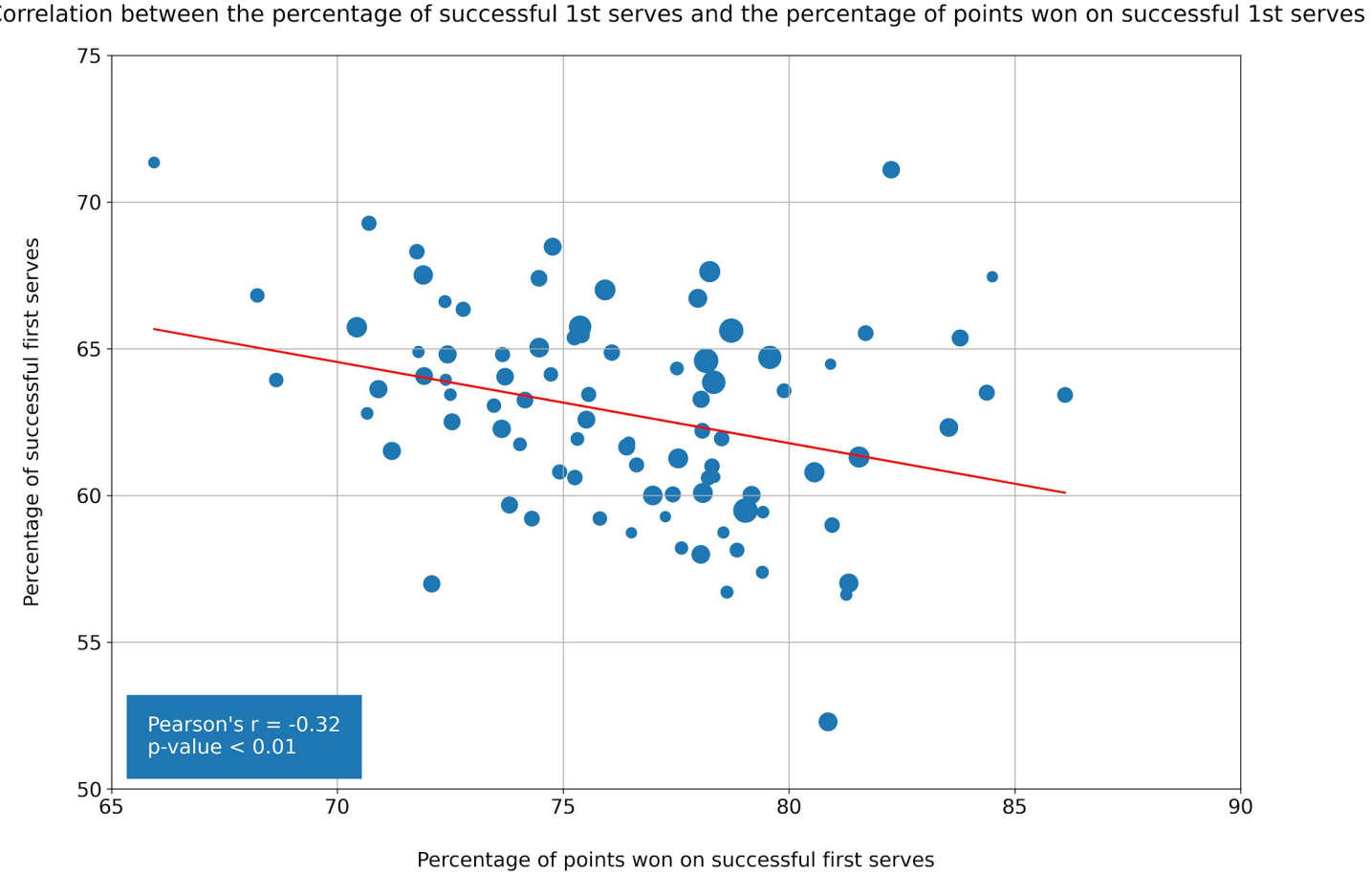
Findings

Analyzing the first serve from a different perspective

Successful results were found by analyzing the correlation between the percentage of successful first serves and the percentage of points won with successful first serves. The two variables are negatively correlated, which means that the higher the percentage of successful first serves, the lower the percentage of points won on successful first serves. Why? It is hard to say, but we can hypothesize that players with a high percentage of successful first serves are not taking enough risk to win the point and, therefore, are not winning many points with their first serve.

The size of the data points in Figure 4 is based on the player's number of wins. By looking at the biggest data points, it can be concluded that almost all of them fall within the area between $75 < X < 80$ and $60 < Y < 65$. This means that the best performers found a good equilibrium between risk and the chance to maximize the win.

Figure 4


Correlation between the percentage of successful 1st serves and the percentage of points won on successful 1st serves
Percentage of successful first serves
Percentage of points won on successful first serves
Pearson's r = -0.32
p-value < 0.01
75
70
65
60
55
50
65
70
75
80
85
90

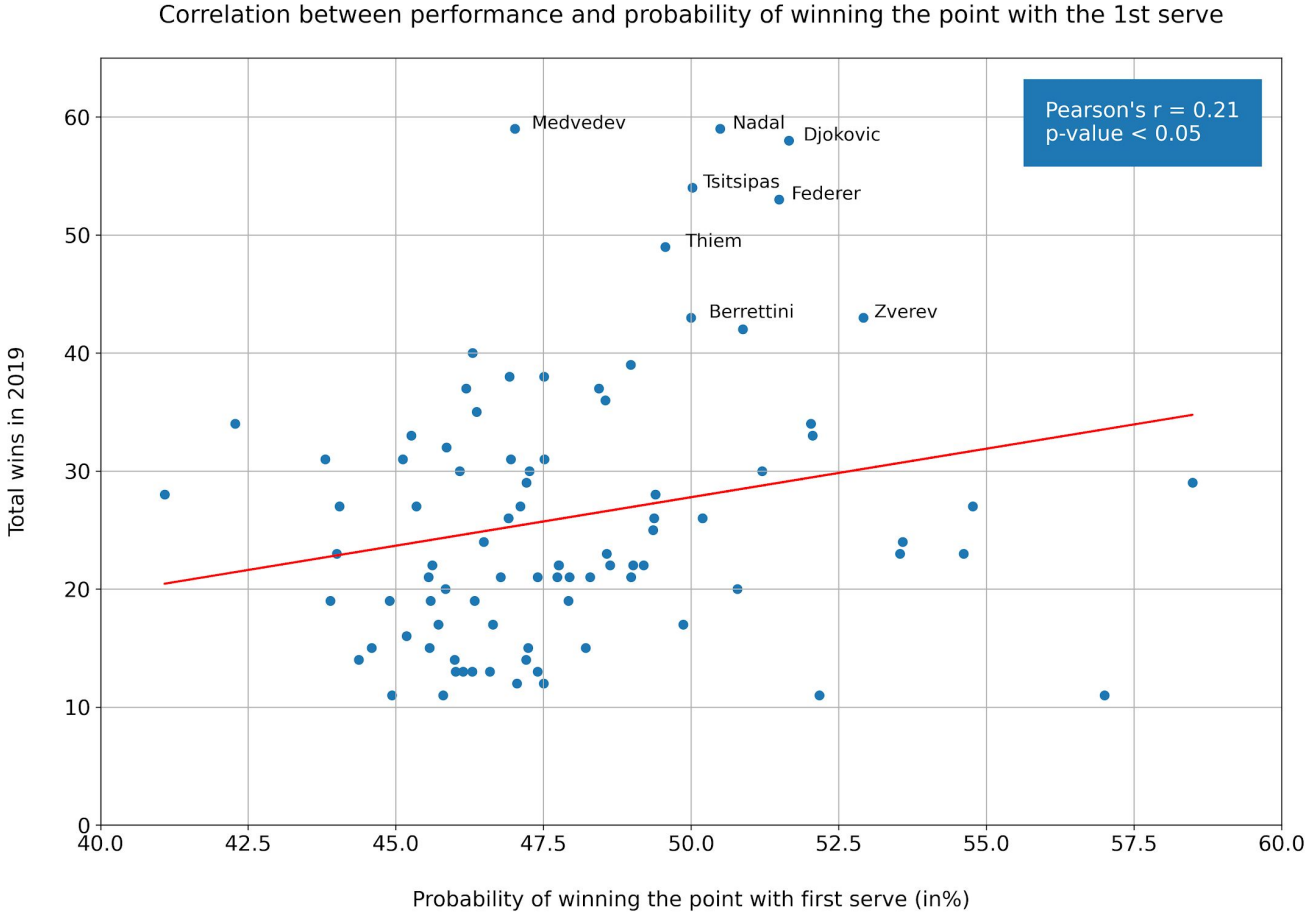
Findings

The perfect equilibrium

The probability of winning the point on the first serve is given by p1*q1, where p1 is the probability that the first serve will be inside the service box and q1 is the conditional probability that the point is won given that the first serve is successful.

The best performers (with a few exceptions) are the ones who maximize the probability of winning the point on their first serve. By look at the X-axis of Figure 5, it can be stated that the top five performers win more than 50% of their serving points with their first serve (except for Medvedev). Indeed, very few players are able to do that and the ones who do are usually big servers like John Isner, who perform worse in rallies. Maximizing the probability of winning the point with the first serve is done through finding the perfect equilibrium between the percentage of successful first serves and the percentage of points won with successful first serves.

Figure 5


Correlation between performance and probability of winning the point with the 1st serve
Pearson's r = 0.21
p-value < 0.05
Medvedev
Nadal
Djokovic
Tsitsipas
Federer
Thiem
Berrettini
Zverev
Total wins in 2019
Probability of winning the point with first serve (in%)
0
10
20
30
40
50
60
40.0
42.5
45.0
47.5
50.0
52.5
55.0
57.5
60.0

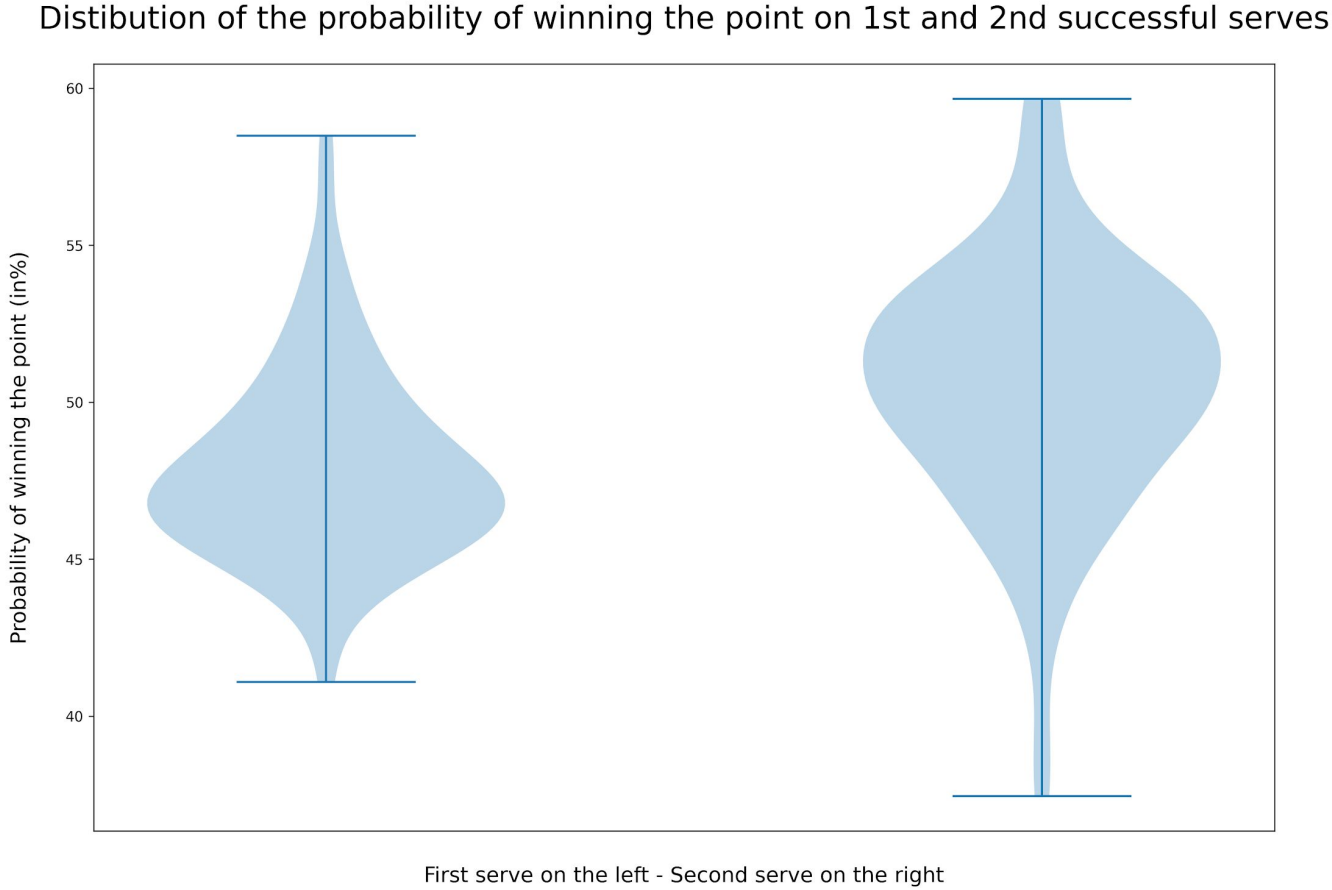
Findings

Second serve compared to first serve

There is no correlation between performance and probability of winning the point on the second serve. However, a few players like Federer and Nadal have incredible stats on their second serve, with a probability of winning the point above 60%.

As we can see in Figure 6, the distribution of the probability of winning the point on successful first serves tends to 0. What does that mean? It means that very few players have a high probability of winning the point on their successful first serves. Is it the same for second serves? Not exactly. By taking a closer look at the distribution, it can be concluded that the second violin is inverted when compared to the first. Indeed, many players have the same probability of winning the point on their successful second serves.

Figure 6


Distibution of the probability of winning the point on 1st and 2nd successful serves
Probability of winning the point (in%)
60
55
50
45
40
First serve on the left - Second serve on the right

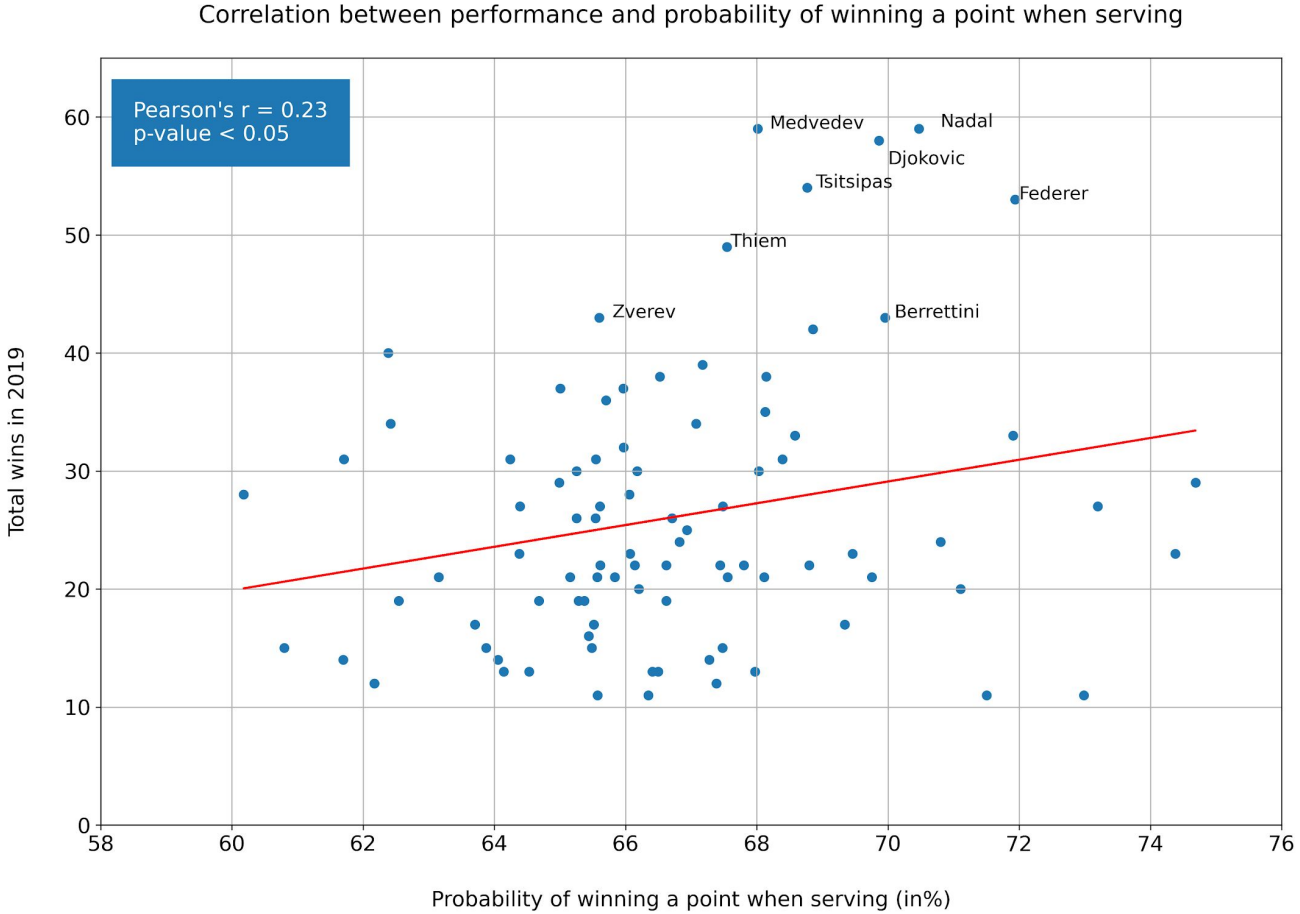
Findings

Overall probability of winning a point when serving

In order to determine the strength of a player's serve we can calculate the overall probability of winning a point when serving. According to O'Donoghue, the probability of winning a point when serving can be calculated with the formula p1*q1 + (1-p1) * p2*q2, where p is the probability that the serve is successful and q is the conditional probability that the point is won given that the serve is successful. One and two represent the first and second serve respectively.

Again, we can see from Figure 7 that the best performers have a high probability of winning a point when serving. Let's look at Federer and Zverev to understand better the importance of having a good first and second serve. If we go back to the last scatter plot, we can see that Zverev has a higher probability than Federer to win the point on a successful first serve (52.6 to 51.8). However, Federer's overall probability of winning a point when serving is 72% while Zverev's is 65.8%. This is quite a big difference since many matches are decided by two or three important points.

Figure 7


Correlation between performance and probability of winning a point when serving
Pearson's r = 0.23
p-value < 0.05
Medvedev
Nadal
Djokovic
Tsitsipas
Federer
Thiem
Zverev
Berrettini
Total wins in 2019
Probability of winning a point when serving (in%)
0
10
20
30
40
50
60
58
60
62
64
66
68
70
72
74
76

Limitations

Tennis is a dynamic complex sport. It is hard to conclude that some players perform better than others because they have a higher probability of winning the point on their successful first serves. Serves are the least hit shots along with volleys. In order to have a better analysis, we should take into consideration several other variables, especially the return of each player.

Due to length restrictions for this project, some analyses could not be explained in detail and may be hard to understand for readers who are not tennis players. The full text is available at https://bit.ly/2ZNqutE.

Conclusions

Hitting a good percentage of serves within the service box is not enough for ATP players. Indeed, players need to maximize their chance of winning a point when serving by balancing the probability of hitting a successful serve and the probability of winning the point when hitting a successful serve. A high probability of hitting a successful serve may indicate that the player is not taking enough risk to put his opponent under pressure.

The best performers in the ATP circuit are able to find a perfect equilibrium to maximize their chance to win the point with a successful first serve. In addition, even though the probability of winning a point with a successful second serve is not correlated to performance, the second serve remains crucial to maximize the chance to win the point when serving. Therefore, tennis players should try to find their perfect equilibrium by balancing power and risk in order to optimize their serving games.

References

- P. O'Donoghue, A. Ballantyne, The impact of speed of service in Grand Slam singles tennis (2004), Science and racket sports III: the proceedings of the eighth international table tennis federation sports science congress and the third world congress of science and racket sports.

- E. Gillet, D. Leroy, R. Thouvarecq, J. F. Stein, A Notational Analysis of Elite Tennis Serve and Serve-Return Strategies on Slow Surface (2009), Journal of Strength and Conditioning Research: Volume 23 — Issue 2 — p 532–539.

- I. Palacios Huerta, Professionals Play Minimax (2003), Review of Economic Studies: 70, 395-415.

Understanding the Importance of First Serve in Tennis with Data Science

In [1]:
```
import pandas as pd
pd.set_option('display.max_columns', 500)
```

In [2]:
```
# Importing the file
data = pd.read_csv('./Desktop/tennis_atp-master/atp_matches_2019.csv')
data.head()
```

Out[2]:

	tourney_id	tourney_name	surface	draw_size	tourney_level	tourney_date	match_num	winner_id	winner_seed	winner_entry	winner_name	wi
0	2019-M020	Brisbane	Hard	32	A	20181231	300	105453	2	NaN	Kei Nishikori	
1	2019-M020	Brisbane	Hard	32	A	20181231	299	106421	4	NaN	Daniil Medvedev	
2	2019-M020	Brisbane	Hard	32	A	20181231	298	105453	2	NaN	Kei Nishikori	
3	2019-M020	Brisbane	Hard	32	A	20181231	297	104542	NaN	PR	Jo-Wilfried Tsonga	
4	2019-M020	Brisbane	Hard	32	A	20181231	296	106421	4	NaN	Daniil Medvedev	

In [3]:
```
# Counting number of matches won by every player
group_winners = data['winner_name'].value_counts()
group_winners = pd.DataFrame(group_winners)
group_winners.columns = ['wins']
group_winners.head(20)
```

Out[3]:

	wins
Daniil Medvedev	59
Rafael Nadal	59
Novak Djokovic	58
Stefanos Tsitsipas	54
Roger Federer	53
Dominic Thiem	49
Matteo Berrettini	43
Alexander Zverev	43
Roberto Bautista Agut	42
Diego Schwartzman	40
Denis Shapovalov	39
Alex De Minaur	38
Andrey Rublev	38
Gael Monfils	37
David Goffin	37
Guido Pella	36
Jan Lennard Struff	35
Benoit Paire	34
Felix Auger Aliassime	34
Stan Wawrinka	33

In [4]:
```
# Counting number of matches lost by every player
group_losers = data['loser_name'].value_counts()
group_losers = pd.DataFrame(group_losers)
group_losers.head()
```

Out[4]:

	loser_name
Joao Sousa	31
Karen Khachanov	29
Jan Lennard Struff	29
Benoit Paire	29
Taylor Fritz	29

In [5]:
```
# Counting number of serve point played by every winning player
servepoints_winner = data[['winner_name', 'w_svpt']].groupby('winner_name').sum()
servepoints_winner = pd.DataFrame(servepoints_winner)
servepoints_winner.sort_values('w_svpt', ascending = False)
```

Out[5]:

winner_name	w_svpt
Stefanos Tsitsipas	4250.0
Daniil Medvedev	4090.0
Novak Djokovic	4011.0
Rafael Nadal	3985.0
Roger Federer	3655.0
...	...
Emil Ruusuvuori	0.0
Sandro Ehrat	0.0
Sanjar Fayziev	0.0
Dragos Dima	0.0
Jurabek Karimov	0.0

241 rows × 1 columns

In [6]:
```
# Counting number of first serves in by every winning player
firstin_winner = data[['winner_name', 'w_1stIn']].groupby('winner_name').sum()
firstin_winner = pd.DataFrame(firstin_winner)
firstin_winner.head()
```

Out[6]:

winner_name	w_1stIn
Adrian Mannarino	1138.0
Adrian Menendez Maceiras	53.0
Ajeet Rai	0.0
Albert Ramos	1300.0
Alejandro Davidovich Fokina	161.0

In [7]:
```
# Counting number of first serves won by every winning player
firstwon_winner = data[['winner_name', 'w_1stWon']].groupby('winner_name').sum()
firstwon_winner = pd.DataFrame(firstwon_winner)
firstwon_winner.head()
```

Out[7]:

winner_name	w_1stWon
Adrian Mannarino	840.0
Adrian Menendez Maceiras	42.0
Ajeet Rai	0.0
Albert Ramos	935.0
Alejandro Davidovich Fokina	109.0

In [8]:
```
# Counting number of first serves won by every winning player
secondwon_winner = data[['winner_name', 'w_2ndWon']].groupby('winner_name').sum()
secondwon_winner = pd.DataFrame(secondwon_winner)
secondwon_winner.head()
```

Out[8]:

winner_name	w_2ndWon
Adrian Mannarino	438.0
Adrian Menendez Maceiras	20.0
Ajeet Rai	0.0
Albert Ramos	414.0
Alejandro Davidovich Fokina	27.0

In [9]:
```
# Counting number of double faults by every player
df_winner = data[['winner_name', 'w_df']].groupby('winner_name').sum()
df_winner = pd.DataFrame(df_winner)
df_winner.head()
```

Out[9]:

winner_name	w_df
Adrian Mannarino	47.0
Adrian Menendez Maceiras	5.0
Ajeet Rai	0.0
Albert Ramos	44.0
Alejandro Davidovich Fokina	8.0

In [10]:
```
# Counting number of second serves played by every player
secondin_winner = servepoints_winner['w_svpt'] - firstin_winner['w_1stIn'] - df_winner['w_df']
secondin_winner = pd.DataFrame(secondin_winner)
secondin_winner.columns = ['w_2ndIn']
secondin_winner.head()
```

Out[10]:

winner_name	w_2ndIn
Adrian Mannarino	722.0
Adrian Menendez Maceiras	39.0
Ajeet Rai	0.0
Albert Ramos	685.0
Alejandro Davidovich Fokina	47.0

In [11]:
```
# Creating DataFrame
df = pd.concat([group_winners, servepoints_winner, firstin_winner, firstwon_winner, secondin_winner, df_winner, secondwon_winner],axis=1,sort=False)
df
```

Out[11]:

	wins	w_svpt	w_1stIn	w_1stWon	w_2ndIn	w_df	w_2ndWon
Daniil Medvedev	59	4090.0	2433.0	1923.0	1484.0	173.0	959.0
Rafael Nadal	59	3985.0	2574.0	2012.0	1308.0	103.0	859.0
Novak Djokovic	58	4011.0	2632.0	2072.0	1243.0	136.0	810.0
Stefanos Tsitsipas	54	4250.0	2714.0	2126.0	1428.0	108.0	857.0
Roger Federer	53	3655.0	2365.0	1882.0	1208.0	82.0	798.0
...	...	...	...	...	...	...	...
Markus Eriksson	1	0.0	0.0	0.0	0.0	0.0	0.0
Duck Hee Lee	1	66.0	40.0	28.0	24.0	2.0	17.0
Elliot Benchetrit	1	72.0	46.0	38.0	25.0	1.0	14.0
Adrian Menendez Maceiras	1	97.0	53.0	42.0	39.0	5.0	20.0
Tallon Griekspoor	1	79.0	48.0	38.0	28.0	3.0	15.0

241 rows × 7 columns

In [12]:
```
# Deleting players with less than 10 wins
winners = df.loc[df['wins'] > 10]
final_winners = winners.loc[df.iloc[:,4] > 100]
final_winners
```

Out[12]:

	wins	w_svpt	w_1stIn	w_1stWon	w_2ndIn	w_df	w_2ndWon
Daniil Medvedev	59	4090.0	2433.0	1923.0	1484.0	173.0	959.0
Rafael Nadal	59	3985.0	2574.0	2012.0	1308.0	103.0	859.0
Novak Djokovic	58	4011.0	2632.0	2072.0	1243.0	136.0	810.0
Stefanos Tsitsipas	54	4250.0	2714.0	2126.0	1428.0	108.0	857.0
Roger Federer	53	3655.0	2365.0	1882.0	1208.0	82.0	798.0
...	...	...	...	...	...	...	...
Hugo Dellien	12	712.0	508.0	335.0	191.0	13.0	115.0
Ivo Karlovic	11	1014.0	684.0	578.0	286.0	44.0	187.0
Andy Murray	11	928.0	545.0	417.0	351.0	32.0	209.0
Kevin Anderson	11	943.0	608.0	492.0	310.0	25.0	197.0
Denis Kudla	11	1024.0	607.0	469.0	375.0	42.0	234.0

89 rows × 7 columns

In [13]:
```
# Performing calculations for the final DataFrame
percentage_success = final_winners['w_1stWon'] / final_winners['w_1stIn'] * 100
percentage_in = final_winners['w_1stIn'] / final_winners['w_svpt'] * 100
percentage_in_snd = final_winners['w_2ndIn'] / (final_winners['w_2ndIn'] + final_winners['w_df']) * 100
percentage_success_snd = final_winners['w_2ndWon'] / (final_winners['w_2ndIn']+ final_winners['w_df']) * 100
wins = final_winners['wins']
percentage_success = pd.DataFrame(percentage_success)
percentage_in = pd.DataFrame(percentage_in)
percentage_in_snd = pd.DataFrame(percentage_in_snd)
percentage_success_snd = pd.DataFrame(percentage_success_snd)
wins = pd.DataFrame(wins)
percentage_success.columns = ['w_1stWonifIn']
percentage_in.columns = ['w_1stInTot']
percentage_success_snd.columns = ['w_2ndWonifIn']
```

In [14]:
```
# Creating final DataFrame
analysis = pd.concat([percentage_success, percentage_in, percentage_in_snd, percentage_success_snd, wins],axis=1,sort=False)
analysis.columns.values[2] = 'w_2ndInTot'
t = analysis.sort_values('wins', ascending = False)
t.head(10)
```

Out[14]:

	w_1stWonIfIn	w_1stInTot	w_2ndInTot	w_2ndWonIfIn	wins
Daniil Medvedev	79.038224	59.486553	89.559445	57.875679	59
Rafael Nadal	78.166278	64.592221	92.700213	60.878809	59
Novak Djokovic	78.723404	65.619546	90.137781	58.738216	58
Stefanos Tsitsipas	78.334562	63.858824	92.968750	55.794271	54
Roger Federer	79.577167	64.705882	93.643411	61.860465	53
Dominic Thiem	75.376254	65.750412	91.894061	57.142857	49
Matteo Berrettini	81.552468	61.310223	92.558846	55.732726	43
Alexander Zverev	78.247468	67.629541	80.036799	48.942042	43
Roberto Bautista Agut	75.929457	67.007346	93.514037	58.276864	42
Diego Schwartzman	70.434783	65.735154	89.156627	52.641335	40

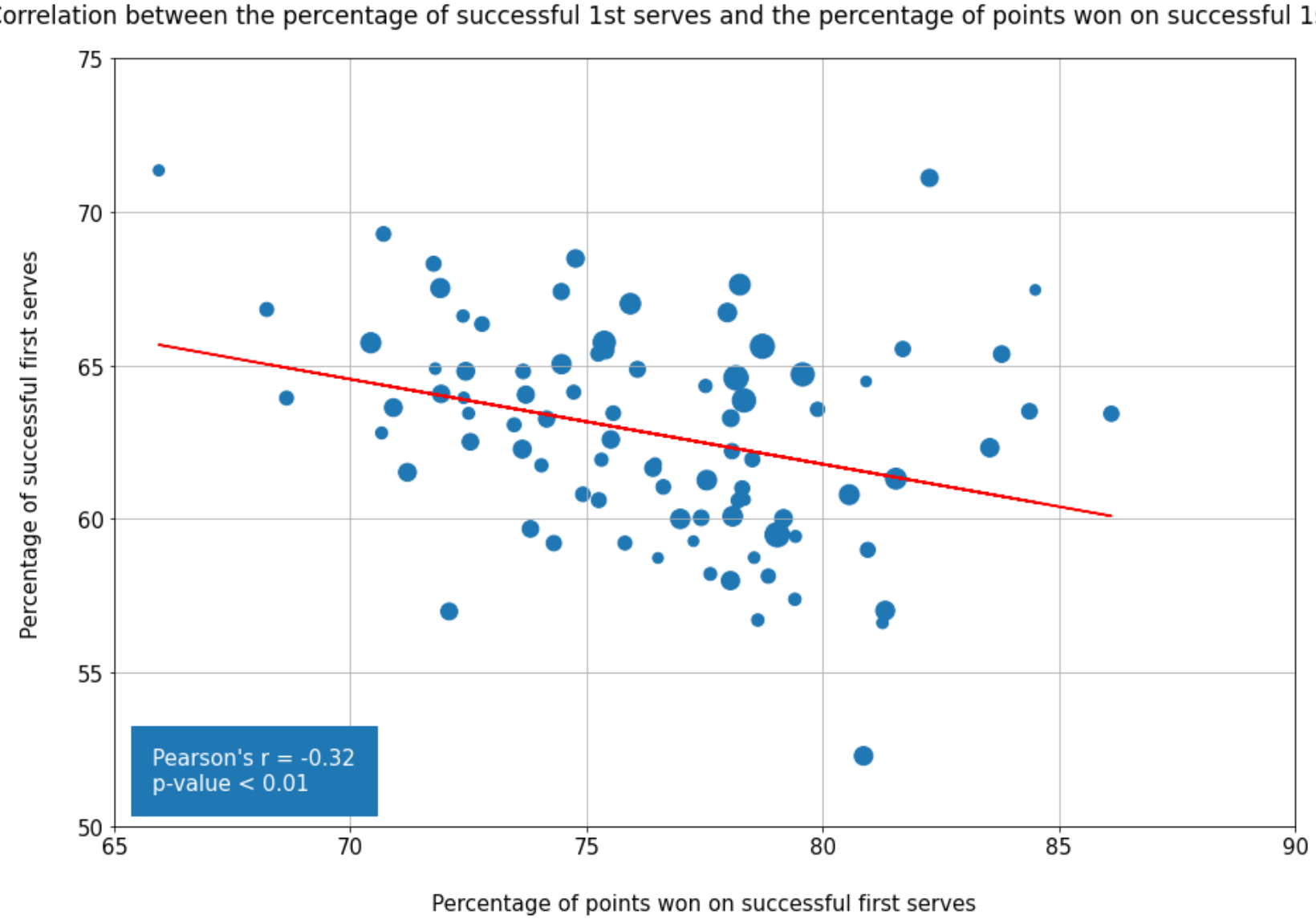
In [16]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between performance and successful first serves',fontsize=22, pad=25.0)
axis.set_xlabel('Percentage of first serve that landed within the service box',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)

textstr = '\n'.join(("Pearson's r = 0.11", "p-value = 0.27"))
plt.text(70, 56.5, textstr, color='white', fontsize=15,
        bbox=dict(facecolor='#1f77b4', edgecolor='#1f77b4', pad=15.0))

X = analysis['w_1stInTot'].values.reshape(-1, 1)
Y = analysis['wins'].values.reshape(-1, 1)

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X, Y)  # perform linear regression
Y_pred = linear_regressor_one.predict(X)  # make predictions

plt.plot(X, Y_pred, color='red')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylim(0, 65)
plt.xlim(50, 75)

axis.scatter(X, Y)
plt.savefig("./Desktop/wins_1stin_corr.png", dpi=300)
```


Correlation between performance and successful first serves
Pearson's r = 0.11
p-value = 0.27
Total wins in 2019
Percentage of first serve that landed within the service box
0
10
20
30
40
50
60
50
55
60
65
70
75

In [251]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr(analysis['w_1stInTot'], analysis['wins'])
```

Out[251]:
```
(0.11581269819998523, 0.27979893888062285)
```

In [282]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between the percentage of successful 1st serves and the percentage of points won
 on successful 1st serves',fontsize=17, pad=25.0)
axis.set_xlabel('Percentage of points won on successful first serves',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Percentage of successful first serves',fontsize=15, labelpad=25.0)

textstr = '\n'.join(("Pearson's r = -0.32", "p-value < 0.01"))
plt.text(65.8, 51.2, textstr, color='white', fontsize=15,
        bbox=dict(facecolor='#1f77b4', edgecolor='#1f77b4', pad=15.0))

X_one = analysis['w_1stWonifIn'].values.reshape(-1, 1)
Y_one = analysis['w_1stInTot'].values.reshape(-1, 1)

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X_one, Y_one)  # perform linear regression
Y_pred = linear_regressor_one.predict(X_one)  # make predictions

plt.plot(X_one, Y_pred, color='red')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylim(50, 75)
plt.xlim(65, 90)

axis.scatter(X_one, Y_one, s=analysis['wins']*5)
plt.savefig("./Desktop/1stin_1stwon_corr.png", dpi=300)
```



In [187]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr(analysis['w_1stWonifIn'], analysis['w_1stInTot'])
```

Out[187]:
```
(-0.31106418493082283, 0.0030065140024029746)
```

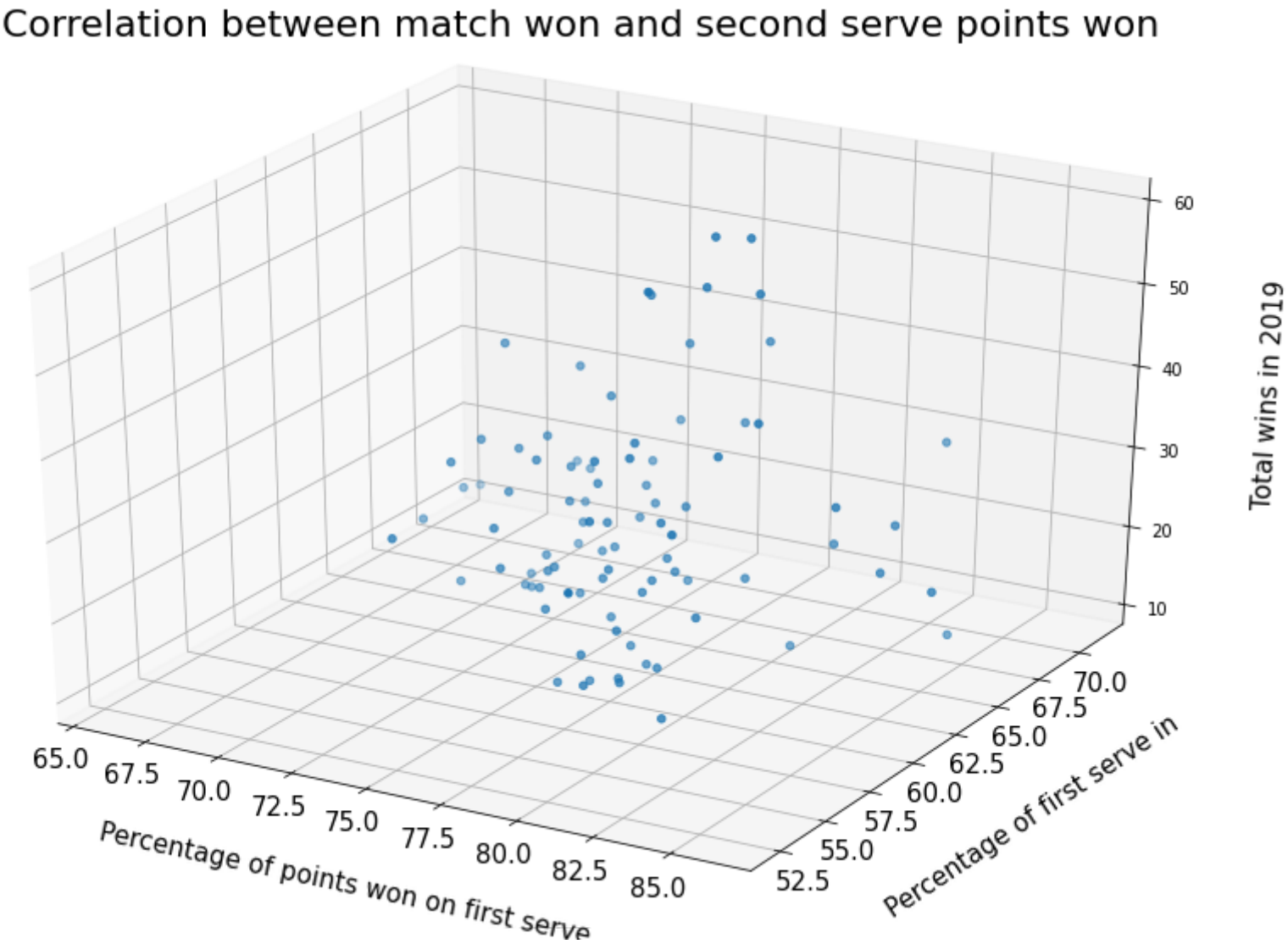
In [226]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between match won and second serve points won',fontsize=22, pad=25.0)
axis.set_xlabel('Percentage of points won on second serve',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)

X_two = analysis['w_2ndWonifIn']*(analysis['w_2ndInTot']/100)

X_two = X_two.values.reshape(-1, 1)
Y_two = analysis['wins'].values.reshape(-1, 1)

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X_two, Y_two)  # perform linear regression
Y_pred_two = linear_regressor_one.predict(X_two)  # make predictions

plt.plot(X_two, Y_pred_two, color='red')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

axis.scatter(X_two, Y_two)
plt.savefig("./Desktop/wins_sndsvwon.png", dpi=300)
```


Correlation between match won and second serve points won
Total wins in 2019
Percentage of points won on second serve
60
50
40
30
20
10
40
45
50
55
60

In [228]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr((analysis['w_2ndWonifIn']*(analysis['w_2ndInTot']/100)), analysis['wins'])
```

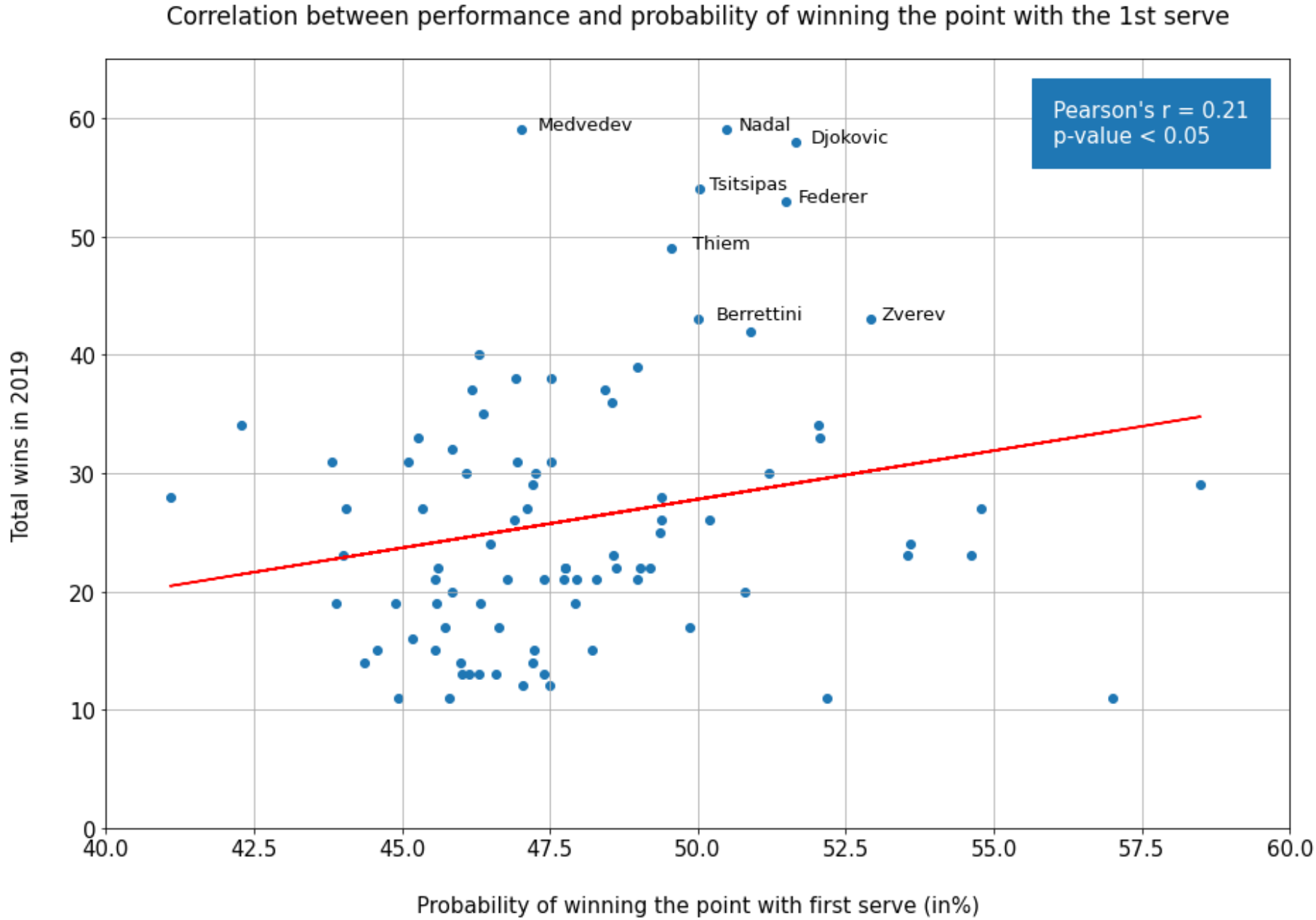
Out[228]:
```
(0.12600738371296955, 0.23933795616415066)
```

In [190]:
```
# Plot (TEST)
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(15,10))
axis = fig.add_subplot(111, projection='3d')
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between match won and second serve points won',fontsize=22, pad=25.0)
axis.set_xlabel('Percentage of points won on first serve',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Percentage of first serve in',fontsize=15, labelpad=25.0)
axis.set_zlabel('Total wins in 2019',fontsize=15, labelpad=25.0)

X_three = analysis['w_1stWonifIn']
Y_three = analysis['w_1stInTot']
Z_three = analysis['wins']

plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

axis.scatter(X_three, Y_three, Z_three)
plt.savefig("./Desktop/3d_analysis.png", dpi=300)
```


Correlation between match won and second serve points won
Total wins in 2019
60
50
40
30
20
10
65.0
67.5
70.0
72.5
75.0
77.5
80.0
82.5
85.0
Percentage of points won on first serve
70.0
67.5
65.0
62.5
60.0
57.5
55.0
52.5
Percentage of first serve in

In [288]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between performance and probability of winning the point with the 1st serve',font
size=17, pad=25.0)
axis.set_xlabel('Probability of winning the point with first serve (in%)',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)

points_won = analysis['w_1stWonifIn']*(analysis['w_1stInTot']/100)

plt.text(47.3, 59, "Medvedev", fontsize=13)
plt.text(50.7, 59, "Nadal", fontsize=13)
plt.text(51.9, 58, "Djokovic", fontsize=13)
plt.text(50.2, 54, "Tsitsipas", fontsize=13)
plt.text(51.7, 53, "Federer", fontsize=13)
plt.text(53.1, 43, "Zverev", fontsize=13)
plt.text(50.3, 43, "Berrettini", fontsize=13)
plt.text(49.9, 49, "Thiem", fontsize=13)

X_four = points_won.values.reshape(-1, 1)
Y_four = analysis['wins'].values.reshape(-1, 1)

textstr = '\n'.join(("Pearson's r = 0.21", "p-value < 0.05"))
plt.text(56, 58, textstr, color='white', fontsize=15,
        bbox=dict(facecolor='#1f77b4', edgecolor='#1f77b4', pad=15.0))

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X_four, Y_four)  # perform linear regression
Y_pred = linear_regressor_one.predict(X_four)  # make predictions

plt.plot(X_four, Y_pred, color='red')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlim(40, 60)
```

```
plt.ylim(0, 65)

axis.scatter(X_four, Y_four)
plt.savefig("./Desktop/wins_prob1st_corr.png", dpi=300)
```


Correlation between performance and probability of winning the point with the 1st serve
Pearson's r = 0.21
p-value < 0.05
Medvedev
Nadal
Djokovic
Tsitsipas
Federer
Thiem
Berrettini
Zverev
Total wins in 2019
Probability of winning the point with first serve (in%)
0
10
20
30
40
50
60
40.0
42.5
45.0
47.5
50.0
52.5
55.0
57.5
60.0

In [192]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr(points_won, analysis['wins'])
```

Out[192]:
```
(0.21784310321661512, 0.0402889208193865)
```

In [193]:
```
# Quick look at Roger stats
roger = data['winner_name'].str.contains('Roger Federer')
roger = pd.DataFrame(data[roger])
roger.describe()
```

Out[193]:

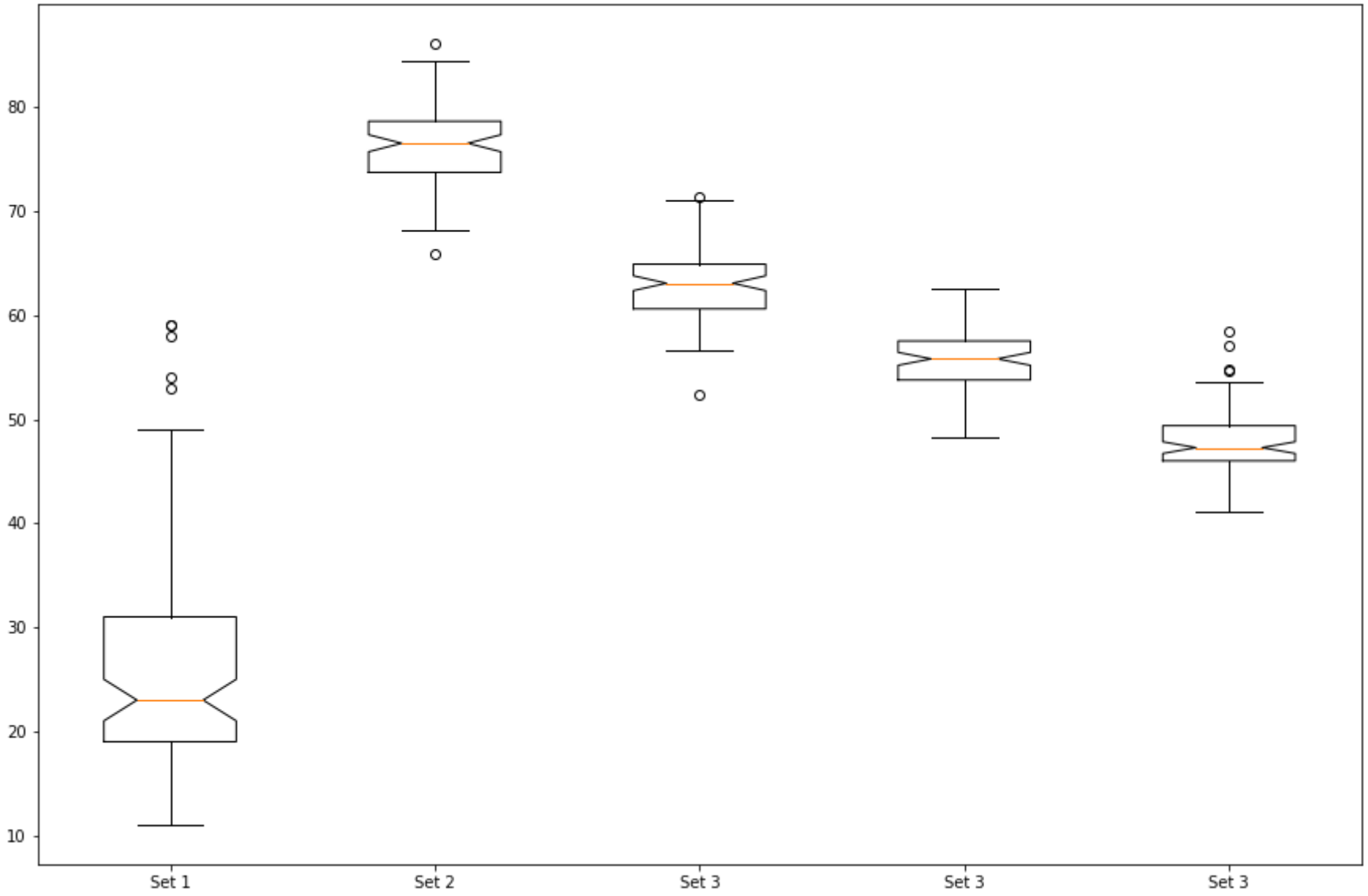
	draw_size	tourney_date	match_num	winner_id	winner_ht	winner_age	loser_id	loser_ht	loser_age	best_of	minutes	
count	53.000000	5.300000e+01	53.000000	53.0	53.0	53.000000	53.000000	31.000000	53.000000	53.000000	51.000000	51
mean	87.849057	2.019057e+07	360.735849	103819.0	185.0	37.815763	112423.094340	183.225806	27.908153	3.679245	97.215686	7
std	46.049888	2.907710e+02	317.818777	0.0	0.0	0.242177	16156.161692	9.545364	4.754138	0.956226	35.510175	3
min	8.000000	2.019011e+07	116.000000	103819.0	185.0	37.434634	104259.000000	163.000000	19.923340	3.000000	52.000000	2
25%	32.000000	2.019032e+07	246.000000	103819.0	185.0	37.607118	104919.000000	180.500000	23.216975	3.000000	72.500000	5
50%	128.000000	2.019053e+07	287.000000	103819.0	185.0	37.798768	105676.000000	185.000000	28.717317	3.000000	85.000000	7
75%	128.000000	2.019081e+07	298.000000	103819.0	185.0	38.009582	106432.000000	188.000000	32.353183	5.000000	116.500000	10
max	128.000000	2.019111e+07	1503.000000	103819.0	185.0	38.258727	200282.000000	206.000000	35.362081	5.000000	215.000000	16

In [194]:
```
# Quick look at Paire stats
paire = data['winner_name'].str.contains('Benoit Paire')
paire = pd.DataFrame(data[paire])
paire.describe()
```

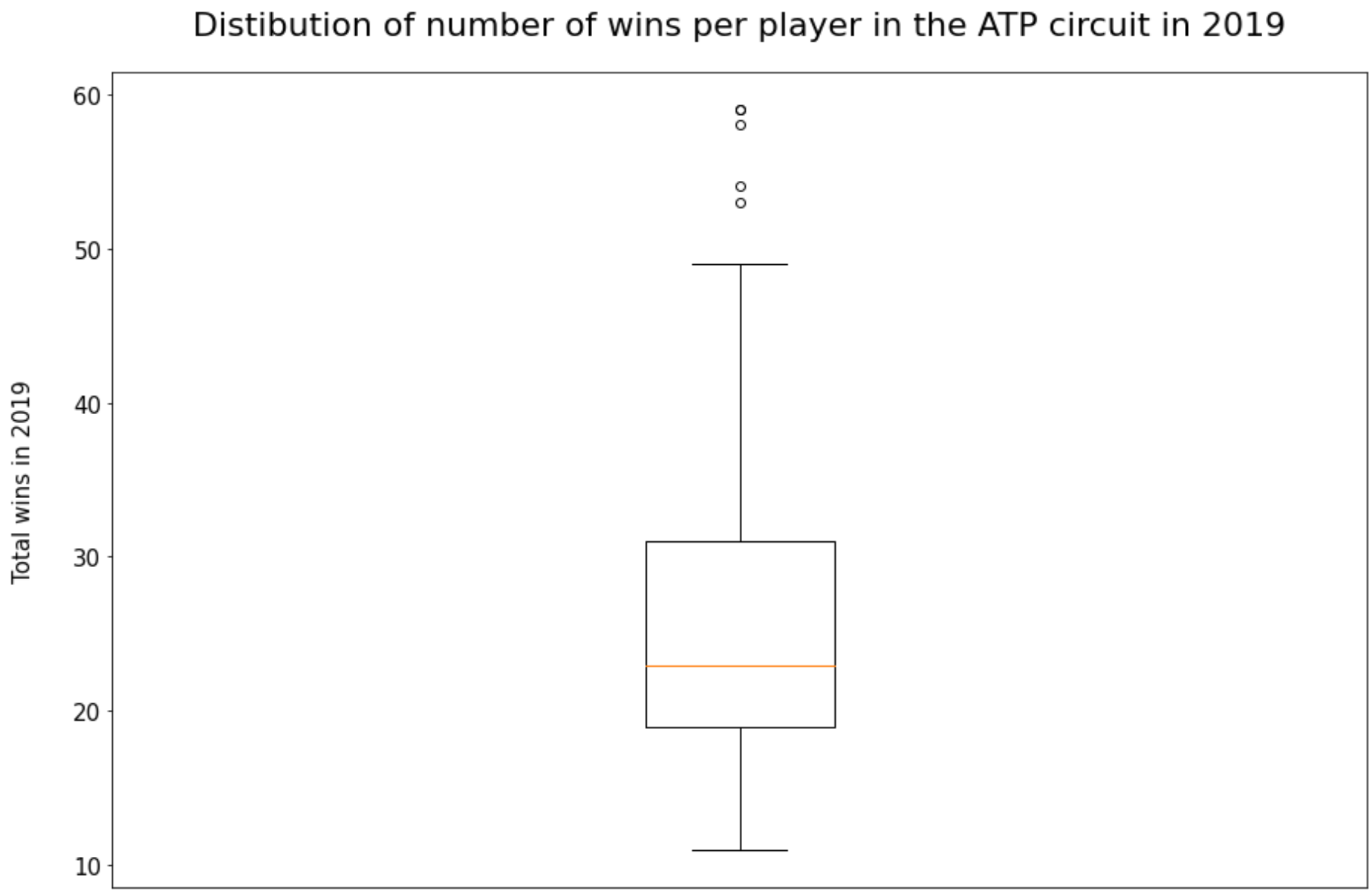
Out[194]:

	draw_size	tourney_date	match_num	winner_id	winner_ht	winner_age	loser_id	loser_ht	loser_age	best_of	minutes	
count	34.000000	3.400000e+01	34.000000	34.0	34.0	34.000000	34.000000	18.000000	34.000000	34.000000	34.000000	34
mean	62.117647	2.019007e+07	353.823529	105332.0	196.0	30.082216	117149.882353	186.222222	27.208922	3.411765	103.205882	10
std	36.900985	2.252338e+03	281.298016	0.0	0.0	0.211046	27624.834052	6.830344	4.531537	0.820851	43.905394	5
min	32.000000	2.018123e+07	128.000000	105332.0	196.0	29.648186	104269.000000	172.000000	18.778919	3.000000	48.000000	4
25%	32.000000	2.019041e+07	273.750000	105332.0	196.0	29.926078	105436.250000	183.000000	24.255305	3.000000	71.500000	7
50%	64.000000	2.019053e+07	289.500000	105332.0	196.0	30.050650	105903.500000	188.000000	27.608487	3.000000	86.500000	9
75%	64.000000	2.019082e+07	297.000000	105332.0	196.0	30.275838	110174.250000	188.000000	29.704312	3.000000	124.500000	14
max	128.000000	2.019103e+07	1314.000000	105332.0	196.0	30.472279	200175.000000	206.000000	35.739904	5.000000	273.000000	26

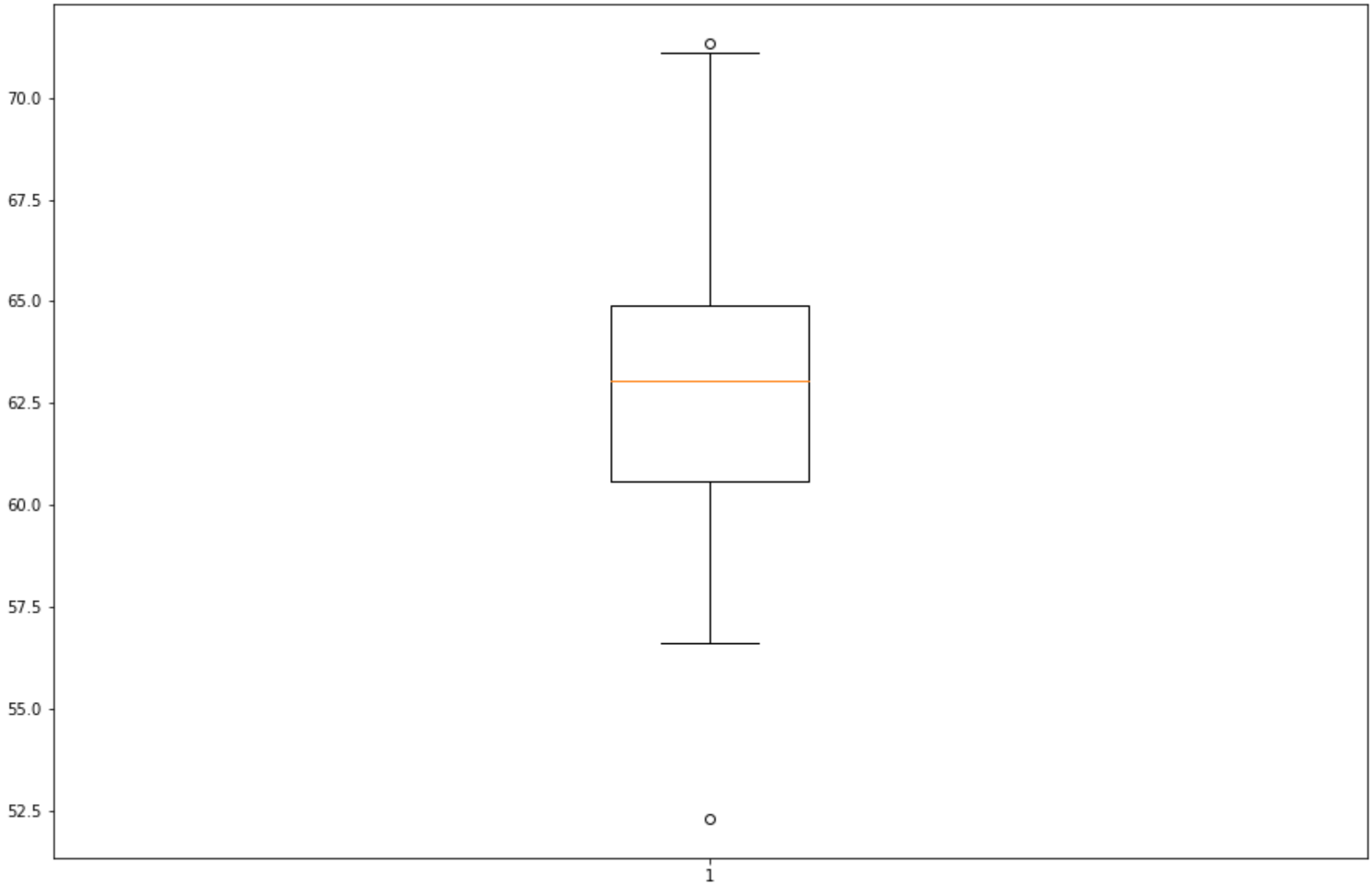
In [195]:
```
# Plot (TEST)
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

points_won = analysis['w_1stWonifIn']*(analysis['w_1stInTot']/100)

X_five = analysis['wins']
Y_five = analysis['w_1stWonifIn']
Z_five = analysis['w_1stInTot']
T_five = analysis['w_2ndWonifIn']
M_five = points_won

plt.boxplot((X_five, Y_five, Z_five, T_five, M_five), notch=True, sym="o", labels=["Set 1", "Set 2", "Set 3",
"Set 3", "Set 3"])
plt.savefig("./Desktop/points_won.png", dpi=300)
```


80
70
60
50
40
30
20
10
Set 1
Set 2
Set 3
Set 3
Set 3

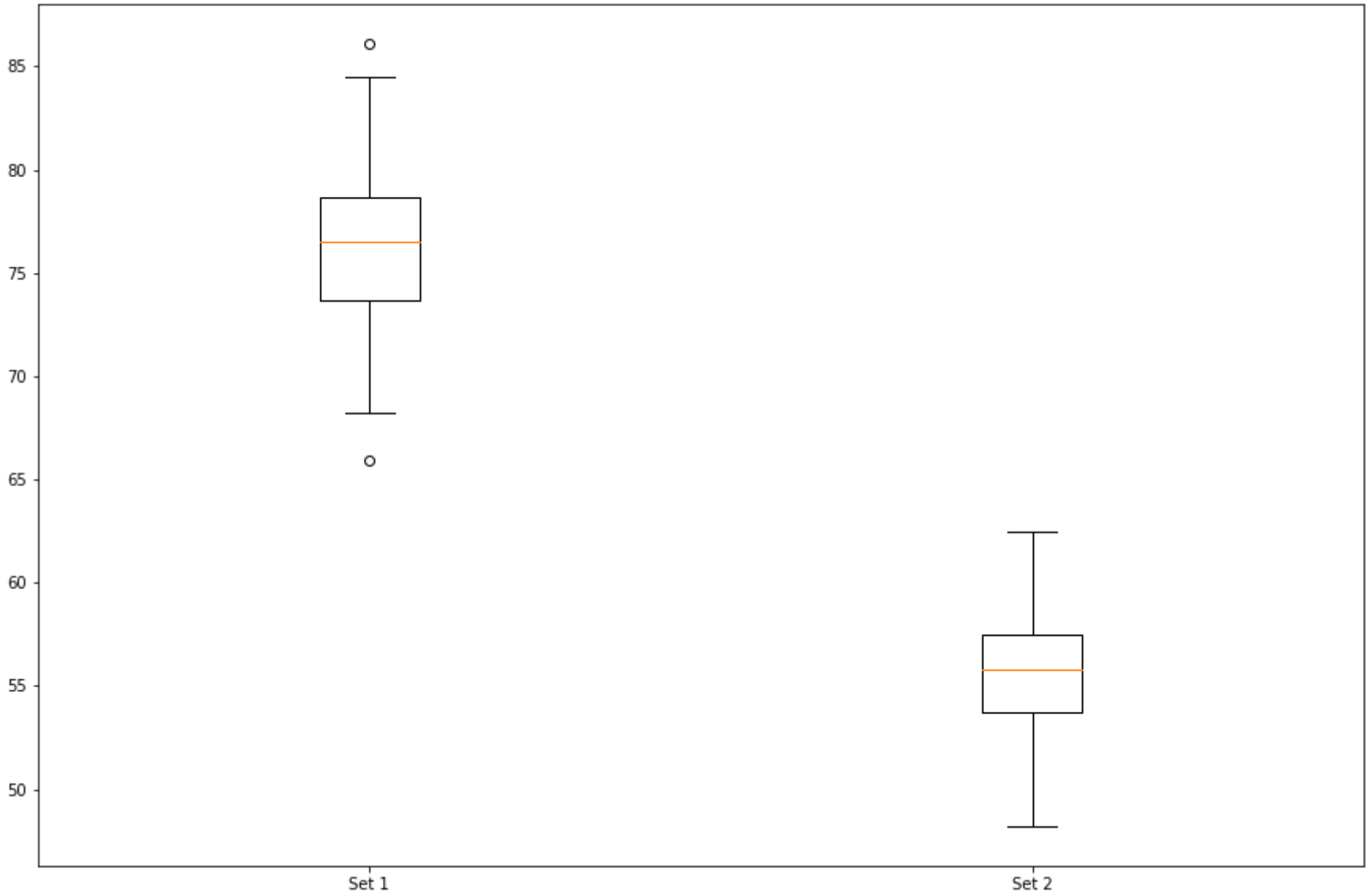
In [247]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.set_title('Distibution of number of wins per player in the ATP circuit in 2019',fontsize=22, pad=25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)
plt.yticks(fontsize=15)
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=False) # labels along the bottom edge are off

X_five = analysis['wins']

plt.boxplot((X_five), sym="o", labels=None)
plt.savefig("./Desktop/wins_distribution.png", dpi=300)
```


Distibution of number of wins per player in the ATP circuit in 2019
Total wins in 2019
60
50
40
30
20
10

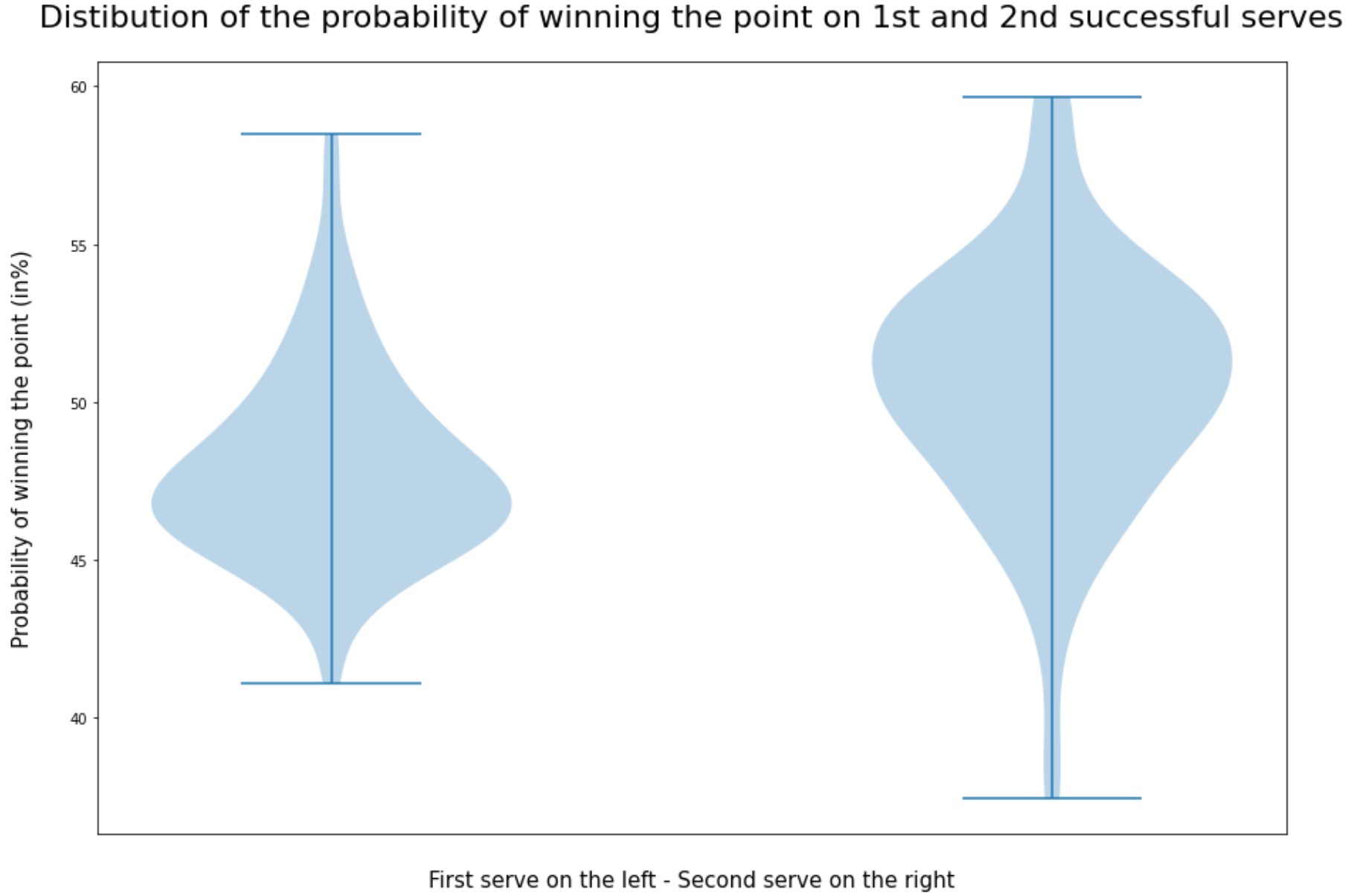
In [249]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

Z_five = analysis['w_1stInTot']

plt.boxplot(Z_five)
plt.savefig("./Desktop/1stserve_in.png", dpi=300)
```


70.0
67.5
65.0
62.5
60.0
57.5
55.0
52.5
1

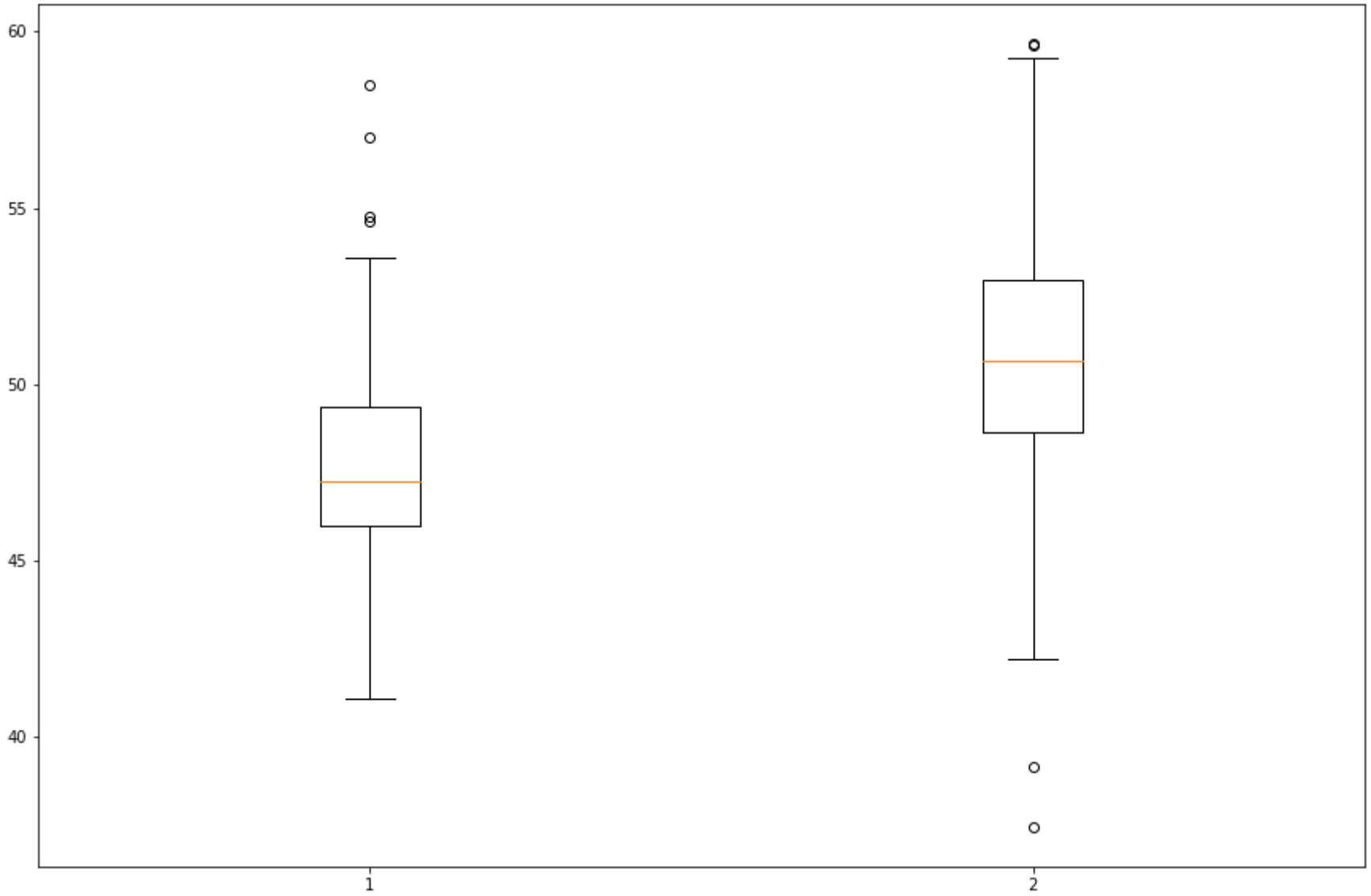
In [198]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

Y_five = analysis['w_1stWonifIn']
T_five = analysis['w_2ndWonifIn']

plt.boxplot((Y_five, T_five), sym="o", labels=["Set 1", "Set 2"])
plt.savefig("./Desktop/points_won.png", dpi=300)
```


85
80
75
70
65
60
55
50
Set 1
Set 2

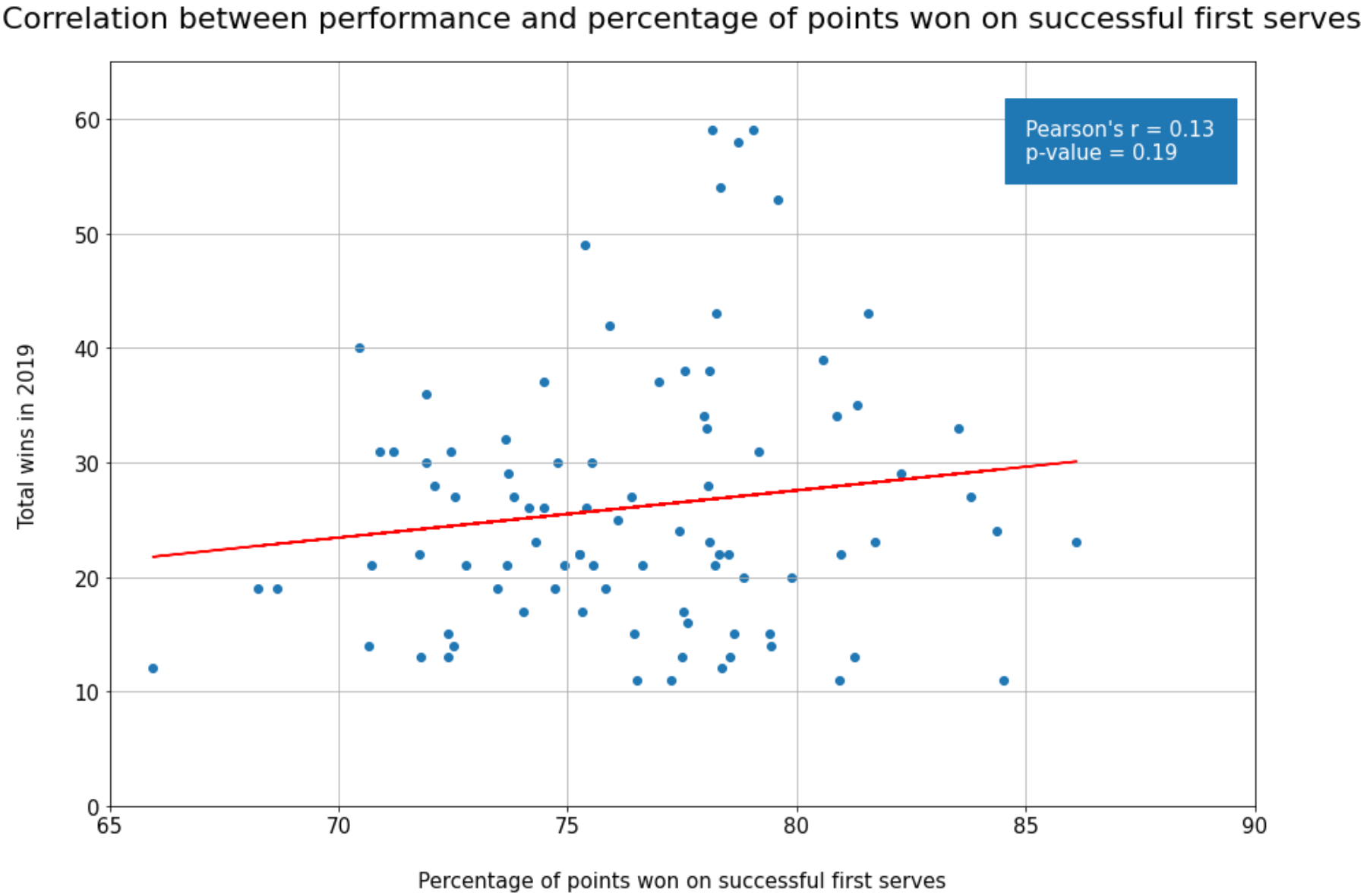
In [323]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.set_title('Distibution of the probability of winning the point on 1st and 2nd successful serves',fontsiz
e=22, pad=25.0)
axis.set_ylabel('Probability of winning the point (in%)',fontsize=15, labelpad=25.0)
axis.set_xlabel('First serve on the left - Second serve on the right',fontsize=15, labelpad=25.0)

prob_1st = analysis['w_1stWonifIn']*(analysis['w_1stInTot']/100)
prob_2nd = analysis['w_2ndWonifIn']*(analysis['w_2ndInTot']/100)

plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=False) # labels along the bottom edge are off

M_five = prob_1st
F_five = prob_2nd

vp = plt.violinplot((M_five, F_five))

plt.savefig("./Desktop/distr_allprob.png", dpi=300)
```


Distibution of the probability of winning the point on 1st and 2nd successful serves
Probability of winning the point (in%)
60
55
50
45
40
First serve on the left - Second serve on the right

In [225]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

prob_1st = analysis['w_1stWonifIn']*(analysis['w_1stInTot']/100)
prob_2nd = analysis['w_2ndWonifIn']*(analysis['w_2ndInTot']/100)

M_five = prob_1st
F_five = prob_2nd

plt.boxplot((M_five, F_five))
plt.savefig("./Desktop/points_won.png", dpi=300)
```


60
55
50
45
40
1
2

In [274]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between performance and percentage of points won on successful first serves',font
size=22, pad=25.0)
axis.set_xlabel('Percentage of points won on successful first serves',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)

textstr = '\n'.join(("Pearson's r = 0.13", "p-value = 0.19"))
plt.text(85, 56.5, textstr, color='white', fontsize=15,
        bbox=dict(facecolor='#1f77b4', edgecolor='#1f77b4', pad=15.0))

X_six = analysis['w_1stWonifIn'].values.reshape(-1, 1)
Y_six = analysis['wins'].values.reshape(-1, 1)

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X_six, Y_six)  # perform linear regression
Y_pred = linear_regressor_one.predict(X_six)  # make predictions

plt.plot(X_six, Y_pred, color='red')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylim(0, 65)
plt.xlim(65, 90)

axis.scatter(X_six, Y_six)
plt.savefig("./Desktop/wins_1stwon_corr.png", dpi=300)
```

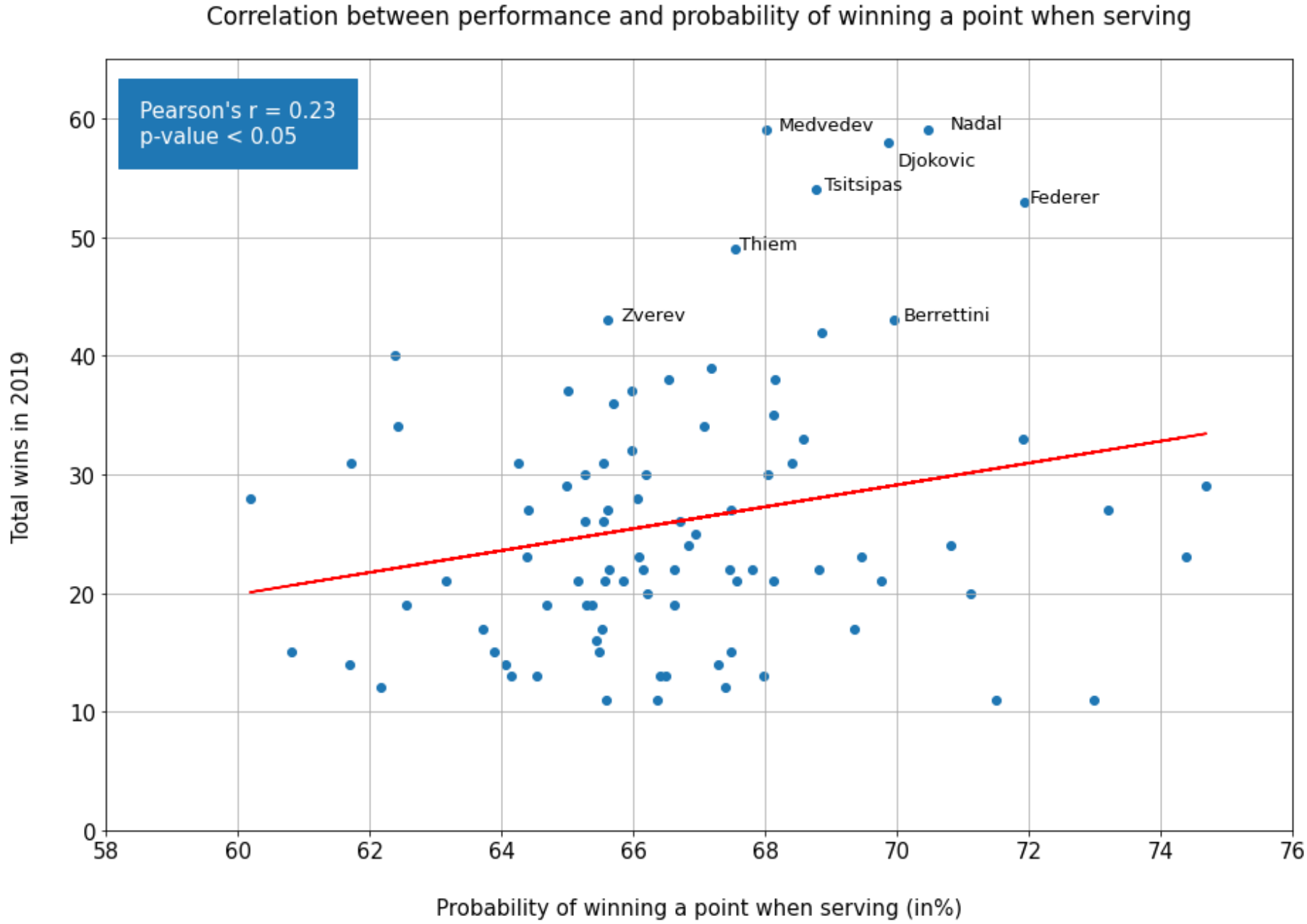

Correlation between performance and percentage of points won on successful first serves
Pearson's r = 0.13
p-value = 0.19
Total wins in 2019
Percentage of points won on successful first serves
0
10
20
30
40
50
60
65
70
75
80
85
90

In [272]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr(analysis['w_1stWonifIn'], analysis['wins'])
```

Out[272]:
```
(0.13955182233061825, 0.1921276749178409)
```

In [341]:
```
# Plot
%matplotlib inline
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

fig, axis = plt.subplots(figsize=(15,10))
# Grid lines, Xticks, Xlabel, Ylabel

axis.yaxis.grid(True)
axis.xaxis.grid(True)
axis.set_title('Correlation between performance and probability of winning a point when serving',fontsize=17,
pad=25.0)
axis.set_xlabel('Probability of winning a point when serving (in%)',fontsize=15, labelpad= 25.0)
axis.set_ylabel('Total wins in 2019',fontsize=15, labelpad=25.0)

points_won_1st = analysis['w_1stWonifIn']*(analysis['w_1stInTot']/100)/100
points_won_2nd = analysis['w_2ndWonifIn']*(analysis['w_2ndInTot']/100)/100
prob = 1-(analysis['w_1stInTot']/100)
final_prob = (points_won_1st + prob * points_won_2nd) * 100

plt.text(68.2, 59, "Medvedev", fontsize=13)
plt.text(70.8, 59.15, "Nadal", fontsize=13)
plt.text(70, 56, "Djokovic", fontsize=13)
plt.text(68.9, 54, "Tsitsipas", fontsize=13)
plt.text(72, 53, "Federer", fontsize=13)
plt.text(65.8, 43, "Zverev", fontsize=13)
plt.text(70.1, 43, "Berrettini", fontsize=13)
plt.text(67.6, 49, "Thiem", fontsize=13)

X_sev = final_prob.values.reshape(-1, 1)
Y_sev = analysis['wins'].values.reshape(-1, 1)

textstr = '\n'.join(("Pearson's r = 0.23", "p-value < 0.05"))
plt.text(58.5, 58, textstr, color='white', fontsize=15,
        bbox=dict(facecolor='#1f77b4', edgecolor='#1f77b4', pad=15.0))

linear_regressor_one = LinearRegression()  # create object for the class
linear_regressor_one.fit(X_sev, Y_sev)  # perform linear regression
Y_pred = linear_regressor_one.predict(X_sev)  # make predictions

plt.plot(X_sev, Y_pred, color='red')
```

```
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlim(58, 76)
plt.ylim(0, 65)

axis.scatter(X_sev, Y_sev)
plt.savefig("./Desktop/wins_prob1stand2nd_corr.png", dpi=300)
```


Correlation between performance and probability of winning a point when serving
Pearson's r = 0.23
p-value < 0.05
Medvedev
Nadal
Djokovic
Tsitsipas
Federer
Thiem
Zverev
Berrettini
Total wins in 2019
Probability of winning a point when serving (in%)
0
10
20
30
40
50
60
58
60
62
64
66
68
70
72
74
76

In [330]:
```
# Correlation analysis
import numpy as np
import scipy.stats
scipy.stats.pearsonr(final_prob, analysis['wins'])
```

Out[330]:
```
(0.22926377316153074, 0.030682308706806907)
```

In [335]:
```
final_prob.head(10)
```

Out[335]:
```
Daniil Medvedev          68.016506
Rafael Nadal             70.471639
Novak Djokovic           69.860783
Stefanos Tsitsipas       68.770404
Roger Federer            71.936373
Dominic Thiem            67.544962
Alexander Zverev         65.598445
Matteo Berrettini        69.958341
Roberto Bautista Agut    68.858336
Diego Schwartzman        62.382015
dtype: float64
```

In [ ]: