

テニスの勝敗に重要な要素とは？

研究動機

部活動でテニスをしていて、より上達するために練習するだけでなくどこを重点的に意識してプレイすると良いのか根拠とともに知りたいと考えた。

研究内容

テニスの試合において1stサーブの成功率や得点率など多くの要素があるがどれが勝敗に大きく関わるのかについて分析することで試合で何を意識するのが重要か判断できると考えた。

1991年から2016年にかけてのATP(世界男子プロテニス協会)のシングルス全試合(91956試合分)のデータを使用した。
データ元:https://datahub.io/sports-data/atp-world-tour-tennis-data/r/match_stats_1991-2016_unindexed.csv(2023.9)

研究 1

研究方法

それぞれのスコアについて、勝者と敗者に分けてグラフを作成して比較した。
データ量が膨大なためグラフ作成にpythonを用いた。
以下がプログラムの一例。

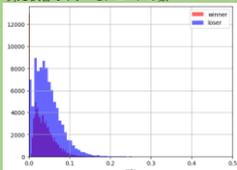
```
import pandas as pd
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt

data = pd.read_csv("/match_stats_1991-2016_unindexed.csv")
data["winner"] = data["winner_double_faults"] / data["winner_service_points_total"]
data["loser"] = data["loser_double_faults"] / data["loser_service_points_total"]
data = data[data["winner"] <= 1]
data = data[data["loser"] <= 1]
data["winner"].hist(bins=100, color="red", label="winner", alpha=0.6)
data["loser"].hist(bins=100, color="blue", label="loser", alpha=0.6)
plt.xlim(0, 0.5)
plt.xlabel("rate")
plt.legend()

print(data["winner"].mean())
print(data["loser"].mean())
print((data["loser"] == 0).sum())
print((data["winner"] == 0).sum())
```

研究結果

例: 1試合でのサービスエースの数



プログラムを実行して左のようなグラフが複数作成できた。
縦軸がプレイヤーの人数で横軸がそれぞれの要素の大きさを表している。勝者を赤で示し、敗者を青で示している。

考察

- ・1stサーブの成功率は勝敗によってほぼ差がみられなく大きな影響はないだろう。
- ・得点率はどちらも勝敗で10%ほどの差が見られるため1stサーブと2ndサーブでどちらが重要かというのではないだろう。しかし1stサーブでの得点率において勝者はほぼ全員が50%を超えていることから1stサーブで有利をとることができると勝つことはできないと考えられる。
- ・サービスエースとダブルフォルトの確率は勝敗でそこまでの差がみられないがグラフ左端の試合中一度もなかったという選手の数勝敗によって大きく違う。サービスエースやダブルフォルトというのは試合の流れを左右する原因になりやすく勝敗に大きく影響する要素だと考えられる。

研究 2

研究方法

より数学的に分析をするため二つの要素についてpythonのstatmodelsライブラリを用いて多変量の線形回帰分析を行いそれらの要素がどれだけ勝敗に影響しているかを調べた。
これは勝利:1と敗北:0と置きこれ(結果変数)に対して複数の要素(説明変数)がそれぞれどれだけの重み(影響力)を持っているか調べるものである。
その際説明変数Xを確率にして $0 \leq X \leq 1$ に揃えるようにした。
以下がデータの処理を含めたプログラムである。

```
import pandas as pd
import numpy as np
import scipy as sp
import statsmodels.formula.api as smf

df = pd.read_csv("/match_stats_1991-2016_unindexed.csv")
df["winner_fault"] = df["winner_double_faults"] / df["winner_service_points_total"]
df["loser_fault"] = df["loser_double_faults"] / df["loser_service_points_total"]
df = df[df["winner_fault"] <= 1]
df = df[df["loser_fault"] <= 1]
df["winner_ace"] = df["winner_aces"] / df["winner_service_points_total"]
df["loser_ace"] = df["loser_aces"] / df["loser_service_points_total"]
df = df[df["winner_ace"] <= 1]
df = df[df["loser_ace"] <= 1]
df_winner = df[["winner_fault", "winner_ace"]].rename(columns={"winner_fault": "fault", "winner_ace": "ace"})
df_loser = df[["loser_fault", "loser_ace"]].rename(columns={"loser_fault": "fault", "loser_ace": "ace"})
df_loser["result"] = 0
df_winner["result"] = 1
df_new = pd.concat([df_loser, df_winner], axis=0)
results = smf.ols("result ~ fault + ace", data=df_new).fit()

results.summary()
```

研究結果

OLS Regression Results						
Dep. Variable:	result	R-squared:	0.091			
Model:	OLS	Adj. R-squared:	0.091			
Method:	Least Squares	F-statistic:	9157.			
Date:	Thu, 09 Nov 2023	Prob (F-statistic):	0.00			
Time:	17:42:52	Log Likelihood:	-12333e+05			
No. Observations:	181940	AIC:	2.467e+05			
Df Residuals:	181937	BIC:	2.467e+05			
Df Model:	2					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.4712	0.002	206.854	0.000	0.467	0.476
fault	-2.9561	0.037	-80.644	0.000	-3.028	-2.884
ace	2.1526	0.019	109.702	0.000	2.085	2.215
Omnibus:	794735.858	Durbin-Watson:	0.161			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21542.061			
Skew:	0.018	Prob(JB):	0.00			
Kurtosis:	1.315	Cond. No.	22.9			

プログラムを実行して左の出力が得られた。Coefの数値からわかるようにダブルフォルトのほうが勝敗に影響が大きいことが分かる。

考察

- ・サーブにおいてはサービスエースを取った時の勝率の上昇よりもダブルフォルトをした時の勝率の減少のほうが大きいため、ダブルフォルトの数をサービスエースの数よりも少なくすべきである。

結論：勝利へのポイント

- ➡ 1stサーブ → サービスエースを積極的に狙い返ってきて必ず有利をとれるように！
- ➡ 2ndサーブ → ダブルフォルトを起こさないために必ず入れる！
- ➡ そしてダブルフォルトの数がサービスエースを超えない程度のサーブを意識する。

今後の展望

今回の研究では特に研究2において分析する要素の数を絞ってしまったため、より多くの要素について研究したい。また、プロの試合のデータからだけでなく自分自身でデータをとってみるなど他の方法で研究していきたい。