Tennis Project Report

Sepideh Niknezhad - Sanaz Hekmati - Marzieh Motamednia

2026-07

Q1

How many tennis players are included in the dataset?

چند بازیکن تنیس در این دیتاست وجود دارند؟

حل شده توسط: مرضیه معتمدنیا

برای محاسبه تعداد بازیکنان یکتای موجود در داده‌ها، در روش اول از جدول v_match_overview استفاده شد. کوئری زیر شناسه بازیکنان حاضر در نقش home و away را استخراج می‌کند:

```
query = """
    SELECT home_player_id, away_player_id
    FROM v_match_overview
"""
```

پس از اجرای کوئری و تبدیل خروجی به Pandas DataFrame، دو ستون player_id مربوط به home و player_id مربوط به away با یکدیگر ادغام شدند تا همه شناسه‌های بازیکنان در یک مجموعه واحد قرار گیرند. در ادامه، ابتدا مقادیر null حذف شد و سپس با استفاده از تابع nunique() تعداد شناسه‌های یکتای بازیکنان محاسبه شد.
نکته مهم در این مرحله آن است که مقادیر null مستقیماً در کوئری SQL حذف نشدند. دلیل این تصمیم آن بود که اگر یک سطر به علت null بودن یکی از ستون‌ها، مثلاً player_id، حذف می‌شد، مقدار ستون مقابل یعنی player_id مربوط به طرف دیگر نیز از دست می‌رفت؛ در حالی که ممکن بود آن مقدار مربوط به یک بازیکن یکتا و قابل استفاده باشد. به همین دلیل، حذف nullها پس از ادغام دو ستون در pandas انجام شد تا هیچ شناسه معتبری به اشتباه کنار گذاشته نشود.
روش دوم، که در سلول کامنت‌شده نوت‌بوک آمده است، بر پایه استخراج مستقیم player_id از دو جدول match_home_team و match_away_team انجام شد. در این روش، با استفاده از دو کوئری جداگانه، شناسه بازیکنان از هر دو جدول جمع‌آوری شد و سپس مشابه روش اول، داده‌ها با هم ادغام و تعداد شناسه‌های یکتا محاسبه شد. این روش در عمل نوعی راه جایگزین برای راستی‌آزمایی نتیجه روش اول به شمار می‌آید.

Q2

What is the average height of the 's?

میانگین قد بازیکنان چقدر است؟

حل شده توسط: مرضیه معتمدنیا

برای محاسبه میانگین قد بازیکنان، ابتدا اطلاعات موردنیاز از دو جدول match_home_team و match_away_team استخراج شد. به همین منظور، در هر جدول شناسه بازیکن (player_id) به همراه مقدار height انتخاب شد و تنها ردیف‌هایی نگه داشته شدند که مقدار قد آن‌ها null نبود:

```
query1 = """
    SELECT DISTINCT player_id, height
    FROM match_home_team
    WHERE height IS NOT NULL
"""
query2 = """
    SELECT DISTINCT player_id, height
    FROM match_away_team
    WHERE height IS NOT NULL
"""
```

پس از اجرای کوئری‌ها و تبدیل خروجی آن‌ها به DataFrame، داده‌های حاصل از دو جدول با یکدیگر ادغام شدند. از آنجا که ممکن بود یک بازیکن در هر دو جدول یا در چند سطر تکرار شده باشد، در ادامه با استفاده از drop_duplicates(subset=['player_id']) تنها یک رکورد برای هر بازیکن نگه داشته شد. این

کار باعث شد هر بازیکن فقط یک بار در محاسبه نهایی لحاظ شود.
در نهایت، با گرفتن میانگین از ستون height، میانگین قد بازیکنان موجود در داده‌هابرابر با ۱٫۸۲۱۶ به دست آمد.

Q3 Which player has the highest number of wins?

کدام بازیکن بیشترین تعداد برد را دارد؟

حل شده توسط: مرضیه معتمدنیا

برای تعیین بازیکنی که بیشترین تعداد برد را داشته است، از سه جدول match_home_team، match_away_team و match_event استفاده شد. هدف از این مرحله آن بود که برای هر مسابقه، علاوه بر شناسه و نام بازیکنان home و away، مقدار winner_code نیز در اختیار باشد تا بتوان برنده هر مسابقه را مشخص کرد.

در کوئری زیر، ابتدا جدول‌های مربوط به بازیکن میزبان و میهمان بر اساس match_id به یکدیگر متصل شدند و سپس اطلاعات مسابقه از جدول match_event به آن‌ها افزوده شد:

```
query = """
    SELECT e.match_id,
           h.player_id as home_id,
           h.name as home_name,
           a.player_id as away_id,
           a.name as away_name,
           winner_code
    FROM match_home_team as h
    INNER JOIN match_away_team as a
    ON h.match_id = a.match_id
    INNER JOIN match_event as e
    ON e.match_id = h.match_id
    WHERE winner_code IS NOT NULL
    QUALIFY ROW_NUMBER() OVER (
        PARTITION BY e.match_id
        ORDER BY e.match_id
    ) = 1
"""
```

در این کوئری، مسابقه‌هایی که مقدار winner_code آن‌ها null بود حذف شدند، زیرا در این حالت تعیین برنده ممکن نیست. همچنین با استفاده از ROW_NUMBER() و شرط QUALIFY، فقط یک رکورد برای هر match_id نگه داشته شد تا از تکرار مسابقه‌ها در محاسبه نهایی جلوگیری شود.
پس از تبدیل خروجی کوئری به DataFrame، نام بازیکن برنده برای هر مسابقه تعیین شد. به این صورت که اگر مقدار winner_code برابر ۱ بود، نام بازیکن home به عنوان برنده ثبت شد و در غیر این صورت نام بازیکن away در نظر گرفته شد. سپس با استفاده از value_counts()، تعداد بردهای هر بازیکن شمرده شد و بازیکنی که بیشترین فراوانی را داشت، استخراج شد.
بر اساس نتیجه این محاسبه، بازیکن Popko D. با ۲۸ برد، بیشترین تعداد پیروزی را در داده‌های مورد بررسی داشته است.

Q4 What is the longest match recorded in terms of duration?

طولانی‌ترین مسابقه ثبت‌شده از نظر مدت زمان کدام است؟

حل شده توسط: سپیده نیک‌نژاد

برای حل این سوال، پس از خواندن فایل‌های parquet، از جدول MatchTimeInfo استفاده می‌کنیم. با توجه به اطلاعات استخراج شده از این جدول، تمامی مقادیر در ستون‌های period_۱ و period_۲، NaN

هستند و برای تسریع محاسبات این دو ستون را drop می‌کنیم. از طرفی برخی داده‌های duplicate، صرفا در ستون date اختلاف دارند که این موضوع می‌تواند به دلیل ثبت یک داده در چندین روز باشد، در نتیجه ستون date را نیز drop می‌کنیم تا به راحتی داده‌های duplicate را حذف کنیم.
در ادامه، با یک بررسی می‌توان فهمید که ۲ مقدار منفی در ستون period_۲ وجود دارد که با در نظر گرفتن تعداد زیاد داده‌ها، می‌توان از دو سطر حاوی این مقادیر منفی صرف نظر کرد. از طرفی هیچ سطری با period_۱ یا period_۲ با مقدار NaN ندارد اما برخی سطرها در ستون period_۳ تعریف نشده‌اند که با منطق سازگار است و نشان دهنده اتمام بازی در ست دوم است.
در نهایت با ایجاد ستون match_duration که جمع مقادیر در ۳ ستون است، زمان کل بازی را محاسبه می‌کنیم. این کار را به ۴ طریق پیش خواهیم برد:

۱. طولانی‌ترین مسابقه بدون حذف داده‌های پرت: نتیجه این جستجو match_id = ۱۲۰۶۳۶۱۱ با مدت زمان ۳۳۶۷۹۰٫۰ ثانیه است. این بازی در ۲ ست بین Shapatava S و Branstine C انجام شده است پس این مقدار واقعی نیست و صحت ندارد.
۲. طولانی‌ترین مسابقه با حذف داده‌های پرت از ۳ ستون مدت زمان: نتیجه این جستجو match_id = ۱۲۰۲۴۲۴۵ با مدت زمان ۱۳۲۰۴٫۰ ثانیه یا است که با توجه به ۳ ست بودن بازی و کشیده شدن به tie_break در ۲ ست، می‌تواند منطقی باشد. این بازی بین Agafonov E و Iannaccone F انجام شده است.
۳. طولانی‌ترین مسابقه با حذف داده‌های پرت از ۳ ستون مدت زمان و ستون match_duration: نتیجه این جستجو match_id = ۱۲۱۲۵۰۵۵ با مدت زمان ۱۱۱۸۰٫۰ ثانیه است که واضحا از مورد قبلی کمتر است.
۴. طولانی‌ترین مسابقه با حذف داده‌های پرت از ستون match_duration: نتیجه این جستجو match_id = ۱۲۱۸۱۳۷۲ با مدت زمان ۱۱۸۷۶٫۰ ثانیه است که واضحا از مقدار به دست آمده در مورد دوم کمتر است. پس جواب نهایی به شکل زیر خواهد بود:

Match_id	Match_duration	Player۱	Player۲
۱۲۰۲۴۲۴۵	۱۳۲۰۴٫۰s	agafonov	iannaccone

داده‌های جدول بالا به کمک جدول MatchEventInfo به دست آمده اند.

Q5

How many sets are typically played in a tennis match?

معمولاً در یک مسابقه تنیس چند ست بازی می‌شود؟

حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال ۵، از جدول matchhome_team_score استفاده شد. این جدول امتیازهای هر ست مسابقه را برای بازیکن میزبان نگهداری می‌کند و شامل ستون‌های period_۱ تا period_۵ است که هر کدام نشان‌دهنده یک ست از مسابقه هستند.
ابتدا جدول امتیازات بازیکن میزبان از دیکشنری data فراخوانی شد و در متغیر home_score ذخیره گردید.
سپس نام ستون‌های مربوط به ست‌های مسابقه در یک لیست به نام period_cols قرار گرفت.
در مرحله بعد، با استفاده از تابع notna() بررسی شد که کدام یک از ستون‌های period_۱ تا period_۵ دارای مقدار هستند. وجود مقدار در هر ستون به این معناست که آن ست در مسابقه برگزار شده است.
سپس با استفاده از تابع sum(axis=۱) تعداد ستون‌های غیرخالی برای هر ردیف شمارش شد و نتیجه در ستون جدیدی با نام sets_played ذخیره گردید. این ستون تعداد ست‌های برگزارشده در هر مسابقه را نشان می‌دهد.
پس از محاسبه تعداد ست‌ها، با استفاده از تابع mean() میانگین تعداد ست‌های برگزارشده در تمام مسابقات محاسبه و نمایش داده شد.
در پایان نیز با استفاده از value_counts() تعداد مسابقاتی که در ۲، ۳، ۴ یا ۵ ست به پایان رسیده‌اند محاسبه شد و با sort_index() نتایج بر اساس تعداد ست‌ها به صورت مرتب نمایش داده شدند.

```
Average number of sets: 2.277800484170151

Distribution:
sets_played
2    22971
3     8836
Name: count, dtype: int64
```

در ابتدا تعداد ست‌های هر مسابقه بر اساس ستون‌های period_۱ تا period_۵ محاسبه شد. سپس رکوردهایی که تعداد ست آن‌ها صفر یا یک بود، به دلیل ناقص بودن یا ثبت نشدن کامل اطلاعات مسابقه، از تحلیل حذف شدند. پس از حذف این رکوردها، میانگین تعداد ست‌های هر مسابقه ۲٫۲۸ به دست آمد. همچنین، ۲۲,۹۷۱ مسابقه در دو ست و ۸,۸۳۶ مسابقه در سه ست به پایان رسیده‌اند. این نتایج نشان می‌دهد که بیشتر مسابقات بدون نیاز به ست سوم خاتمه یافته‌اند و تنها بخش کمتری از مسابقات برای تعیین برنده به ست سوم کشیده شده‌اند. بنابراین، در این مجموعه داده، رایج‌ترین حالت پایان مسابقه، پیروزی در دو ست است.

Q6

Which country has produced the most successful tennis players?

کدام کشور موفق‌ترین بازیکنان تنیس را پرورش داده است؟

حل شده توسط: سپیده نیک‌نژاد

برای حل این سوال از جدول MatchEventInfo و ستون‌های match_id و winner_code استفاده می‌کنیم. به کمک جدول‌های MatchHomeTeamInfo و MatchAwayTeamInfo، اطلاعات دو طرف بازی را استخراج کرده و ستون‌های home_country و away_country را ایجاد می‌کنیم. در ادامه با کمک کتابخانه numpy، ستون winner_country را با استفاده از winner_code پر کنیم.
اکنون با استفاده از ستون‌های ایجاد شده، جدول جدیدی ایجاد می‌کنیم که ستون‌های آن، اسم کشور، تعداد بردها و تعداد کل بازی‌هایش خواهد بود. برای پیدا کردن کشوری با بهترین عملکرد، موفقیت را به صورت success_rate = winnig_matches / total_matches تعریف می‌کنیم.
نکته: البته این معیار کافی نیست. برای مثال اگر یک کشور صرفا یک بازی داشته باشد و آن را ببرد، ضریب موفقیت ۱ خواهد داشت. این در حالی است که کشوری با ۱۰۰ بازی و ۸۰ برد، ضریب موفقیت ۰٫۸ دارد و واضحا کشور دوم، برخلاف داشتن ضریب کمتر، باید موفق‌تر اعلام شود. برای حل این مشکل معیار score را تعریف می‌کنیم که هم تعداد بازی‌ها و هم نرخ موفقیت در آن اثرگذار باشند. Score برابر است با (success rate*log(total_matches. دلیل استفاده از لگاریتم این است که دامنه اعداد با مقادیر success_rate در تعادل باشد و کمی نرمال شود.
نتیجه به دست آمده برای ۳ کشور برتر به صورت زیر است:

RANK	country	Number of winning_matches	Number of total_matches	Success_rate	Score
۱	France	۱۱۷۱	۲۱۳۹	۰.۵۴۷۴۵۲	۱.۸۲۳۱۳۱
۲	Italy	۱۱۳۵	۲۰۶۷	۰.۵۴۹۱۰۵	۱.۸۲۰۴۷۰
۳	USA	۱۰۰۵	۱۸۴۲	۰.۵۴۵۶۰۳	۱.۷۸۱۵۵۱

۳ کشور برتر بر اساس معیار score

پس بر اساس معیار score، برترین کشور فرانسه خواهد بود.

نکته: این سوال بدون حذف داده‌های پرت حل شده است، زیرا با توجه به تعداد بازی‌ها و سرچ گوگل، داده‌ها واقعی هستند و داده اشتباه نداریم. با این حال با حذف داده‌های پرتِ به دست آمده به کمک IQR از ستون total_matches، کشور برزیل با ۶۱۷ بازی و ۳۲۸ برد، برتر خواهد بود اما به دلیل تعداد بازی بسیار بیشتر فرانسه و البته success_rate بالاتر، همچنان پیشتاز خواهد بود.

Q7

What is the average number of aces per match?

میانگین تعداد ace در هر مسابقه چقدر است؟

حل شده توسط: سپیده نیک‌نژاد

برای حل این سوال، ابتدا جدول PeriodInfo را می‌خوانیم. این جدول شامل اطلاعاتی از هر ست می‌باشد. برای مثال در هر ست و در مجموع کل بازی، تعداد aceها، درصد first_serve و اطلاعات دیگری نمایش داده می‌شود. پس از خواندن جدول، با فیلتر روی ستون statistic_name، اطلاعات مربوط به aces را فیلتر می‌کنیم. از طرفی در ستون period، مدت زمان انتخابی را ALL که نشان دهنده کل بازی است، قرار می‌دهیم. در ادامه با حذف ستون date و duplicateها، جدول را تمیز می‌کنیم.
در ادامه ستون جدیدی به نام total_aces ایجاد می‌کنیم که حاصل جمع دو ستون تعداد aceها از دو طرف بازی است. با این ستون می‌توانیم به پاسخ مورد نظر برسیم. ستون‌های home_stat و away_stat که total_aces را تشکیل می‌دهند، مقدار NaN ندارند و از این لحاظ مشکلی ایجاد نمی‌کنند. در ادامه ۲ رویکرد برای پاسخ به سوال در نظر می‌گیریم:

۱. حذف نکردن داده‌های پرتِ به دست آمده به کمک IQR: با یک بررسی روی جدول نهایی و مرتب کردن بر اساس ستون total_aces، می‌توان فهمید که بیشترین تعداد ace مربوط به بازی با match_id=۱۲۰۴۹۶۴۸ است که بین Griekspoor T. و Hurkacz H. برگزار شده است. بر اساس داده‌های به دست آمده از گوگل، این بازی واقعا ۵۱ ضربه ace داشته و داده ثبت شده صحت دارد. با بررسی چند داده دیگر، می‌توان به درستی داده‌ها پی برد. پس داده‌ها برخلاف اشتباه به نظر رسیدن، واقعی و درست هستند. میانگین ace ها در این حالت برابر است با ۵٫۵ ضربه در هر بازی.

<u>میانگین ace/match بدون حذف داده‌های پرت = ۵٫۵۵</u>

۲. حذف داده‌های پرت به کمک مقدار IQR: در این روش، میانگین عدد ۴٫۸۸ به دست می‌آید.

<u>میانگین ace/match پس از حذف داده‌های پرت = ۴٫۸۸</u>

نتیجه نهایی : با توجه به اینکه در این جدول داده‌های پرت، معنادار هستند، میانگین نهایی ۵٫۵ خواهد بود اما اثر حذف داده‌های پرت می‌تواند میانگین را تا ۴٫۸۸ کاهش دهد.

Q8

Is there a difference in the number of double faults based on gender?

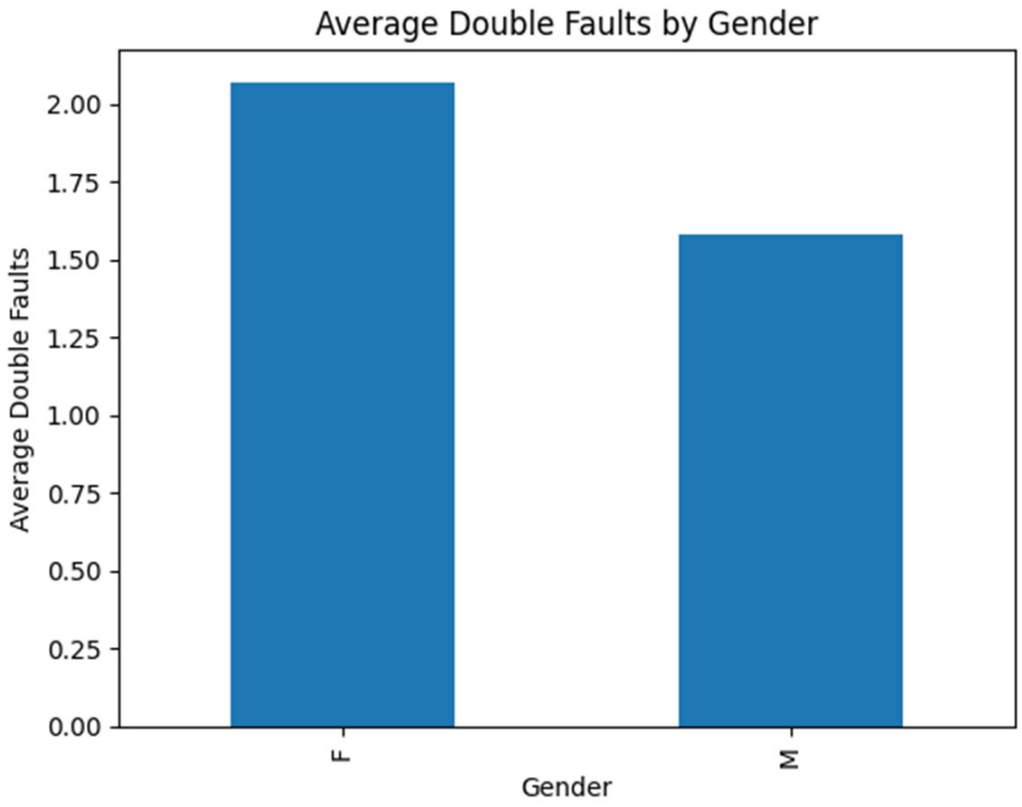
معمولاً در یک مسابقه تنیس چند ست بازی می‌شود؟

حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال ۸، ابتدا لازم بود ساختار جدول statisticsstatistics بررسی شود تا مشخص شود اطلاعات آماری مسابقات در چه ستون‌هایی ذخیره شده‌اند و نام دقیق آمارهای ثبت‌شده چیست.
ابتدا جدول statisticsstatistics از دیکشنری data فراخوانی و در متغیر statistics ذخیره شد.
سپس با استفاده از دستور statistics.columns نام تمام ستون‌های موجود در این جدول نمایش داده شد. این کار به منظور آشنایی با ساختار داده‌ها و شناسایی ستون‌های مورد نیاز برای تحلیل انجام گرفت. ستون‌هایی مانند home_value، statistic_name، match_id و away_value از مهم‌ترین ستون‌های این جدول هستند.
در ادامه، با استفاده از دستور statistics[«statistic_name»].unique() تمام مقادیر یکتای ستون statistic_name استخراج و نمایش داده شدند. این ستون شامل انواع مختلف آمار ثبت‌شده برای هر مسابقه است؛ مانند break_points_saved، first_serve، double_faults، aces و سایر شاخص‌های عملکرد بازیکنان.
هدف از اجرای این مرحله، شناسایی نام دقیق آماری بود که برای پاسخ به سؤال مورد نیاز است. با بررسی خروجی مشخص شد که مقدار صحیح برای تعداد دابل‌فالت‌ها double_faults است و از همین مقدار در مراحل بعدی تحلیل استفاده شد.
جدول statisticsstatistics که شامل آمار مسابقات است، از دیکشنری data فراخوانی شد. سپس تنها رکوردهایی که مقدار ستون statistic_name آن‌ها برابر با double_faults بود، انتخاب شدند. به این ترتیب فقط اطلاعات مربوط به تعداد دابل‌فالت‌های بازیکنان برای ادامه تحلیل مورد استفاده قرار گرفت.

در مرحله بعد، اطلاعات جنسیت بازیکنان از دو جدول matchhome_team و matchaway_team استخراج شد. از آنجا که در هر مسابقه یک بازیکن میزبان و یک بازیکن مهمان حضور دارد، لازم بود اطلاعات هر دو گروه به صورت جداگانه بررسی شوند.
سپس جدول دابل‌فالت‌ها با جدول بازیکنان میزبان بر اساس ستون match_id ادغام (Merge) شد تا جنسیت بازیکن میزبان به آمار مسابقه اضافه شود. تعداد دابل‌فالت بازیکن میزبان نیز از ستون home_value استخراج و در ستونی جدید با نام double_faults ذخیره شد.
همین فرآیند برای بازیکنان مهمان نیز تکرار شد. جدول دابل‌فالت‌ها با اطلاعات بازیکنان مهمان ادغام شد و تعداد دابل‌فالت بازیکن مهمان از ستون away_value استخراج گردید.
پس از آماده‌سازی اطلاعات هر دو گروه، داده‌های مربوط به بازیکنان میزبان و مهمان با استفاده از تابع pd.concat() در یک DataFrame واحد به نام result ترکیب شدند تا تمام بازیکنان صرف‌نظر از نقش آن‌ها در مسابقه، در یک مجموعه داده قرار گیرند.
در پایان، داده‌ها بر اساس ستون gender گروه‌بندی شدند. ابتدا با استفاده از تابع mean() میانگین تعداد دابل‌فالت برای هر جنسیت محاسبه شد. سپس با استفاده از تابع agg() تعداد نمونه‌ها (count)، میانگین (mean) و انحراف معیار (std) برای هر گروه محاسبه و نمایش داده شد تا امکان مقایسه آماری بین بازیکنان مرد و زن فراهم شود
پس از محاسبه میانگین تعداد دابل‌فالت برای بازیکنان مرد و زن، برای نمایش بهتر نتایج از یک نمودار ستونی (Bar Chart) استفاده شد.
ابتدا با استفاده از تابع groupby() داده‌ها بر اساس ستون gender گروه‌بندی شدند و سپس با تابع mean() میانگین تعداد دابل‌فالت هر گروه محاسبه و در متغیر mean_df ذخیره شد.
در ادامه، با استفاده از دستور plot(kind='bar') یک نمودار ستونی رسم شد که در آن هر ستون نشان‌دهنده میانگین تعداد دابل‌فالت یکی از دو گروه مردان و زنان است.

```
gender
F    2.070056
M    1.578071
Name: double_faults, dtype: float64
         count      mean       std
gender
F       122545  2.070056  2.207455
M       161935  1.578071  1.773147
```



بر اساس نتایج به‌دست‌آمده، میانگین تعداد خطاهای دوگانه (Double Faults) در مسابقات زنان ۲٫۰۷ و در مسابقات مردان ۱٫۵۸ است. این نتایج نشان می‌دهد که در این مجموعه داده، بازیکنان زن به طور متوسط خطاهای دوگانه بیشتری نسبت به بازیکنان مرد ثبت کرده‌اند. همچنین انحراف معیار در بخش زنان (۲٫۲۱) بیشتر از مردان (۱٫۷۷) است که بیانگر پراکندگی بیشتر تعداد خطاهای دوگانه در مسابقات زنان است. با توجه به حجم بالای داده‌ها (۱۲۲,۵۴۵ رکورد برای زنان و ۱۶۱,۹۳۵ رکورد برای مردان)، این نتایج از اعتبار آماری مناسبی برخوردار هستند. با این حال، این اختلاف به تنهایی نشان‌دهنده کیفیت پایین‌تر سرویس بازیکنان زن نیست، زیرا عوامل دیگری مانند سبک بازی، سرعت سرویس، سطح مسابقات و شرایط برگزاری نیز می‌توانند بر تعداد خطاهای دوگانه تأثیرگذار باشند.

Q9

Which player has won the most tournaments in a single month?

کدام بازیکن در یک ماه بیشترین تعداد تورنمنت را برده است؟

حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال۹، ابتدا جدول matchevent که اطلاعات مربوط به مسابقات را در اختیار قرار می‌دهد، فراخوانی شد. سپس جداول matchhome_team و matchaway_team نیز فراخوانی شدند و از هر کدام تنها ستون‌های match_id و full_name انتخاب گردید. برای جلوگیری از تداخل نام ستون‌ها، نام بازیکنان میزبان و مهمان به ترتیب به home_player و away_player تغییر داده شد. همچنین جدول matchtournament برای دسترسی به شناسه تورنمنت (tournament_id) به داده‌ها اضافه شد.
در مرحله بعد، این جداول بر اساس ستون match_id با یکدیگر ادغام شدند تا اطلاعات مسابقه، نام بازیکنان و شناسه تورنمنت در یک DataFrame قرار گیرد.
سپس ستون start_datetime که زمان برگزاری مسابقه را به صورت Unix Timestamp ذخیره کرده بود، با استفاده از تابع pd.to_datetime() و تعیین unit=s به تاریخ قابل خواندن تبدیل شد. پس از آن، ماه برگزاری هر مسابقه استخراج و در ستون جدیدی با نام month ذخیره شد.
برای تعیین برنده هر مسابقه، از ستون winner_code استفاده شد. اگر مقدار این ستون برابر ۱ بود، بازیکن میزبان و اگر برابر ۲ بود، بازیکن مهمان به عنوان برنده انتخاب شد و نتیجه در ستون جدیدی با نام winner ذخیره گردید.
از آنجا که هر تورنمنت شامل چند مسابقه است، ممکن بود یک تورنمنت چند بار برای یک بازیکن ثبت شود. بنابراین، برای جلوگیری از شمارش تکراری، ابتدا رکوردهای تکراری بر اساس month، winner و tournament_id حذف شدند تا هر تورنمنت فقط یک بار برای هر بازیکن در هر ماه در نظر گرفته شود.
در ادامه، داده‌ها بر اساس ماه برگزاری و نام بازیکن گروه‌بندی شدند و تعداد تورنمنت‌های منحصربه‌فرد هر بازیکن در هر ماه محاسبه شد. نتیجه در DataFrame جدیدی با نام wins و ستونی با عنوان tournaments_won ذخیره شد.
در نهایت، با استفاده از تابع idxmax() سطری که بیشترین مقدار tournaments_won را داشت انتخاب شد. نتایج نشان داد که Hazem Naw در فوریه ۲۰۲۴ با کسب ۷ عنوان قهرمانی، بیشترین تعداد تورنمنت‌های برده‌شده در یک ماه را در این مجموعه داده به دست آورده است.

```
All results:
        month                winner  tournaments_won
0     2024-01    Abbagnato, Anastasia                1
1     2024-01           Albie, Audrey                1
2     2024-01  Alexandrova, Ekaterina                1
3     2024-01         Appleton, Emily                1
4     2024-01        Arutiunian, Erik                1
...       ...                     ...              ...
3314  2024-04       Wagner, Stephanie                1
3315  2024-04         Waldner, Niklas                1
3316  2024-04      Wong, Hong Yi Cody                1
3317  2024-04            Zhu, Michael                1
3318  2024-04       Šrámková, Rebecca                1

[3319 rows x 3 columns]

Player with the most tournament wins in a single month:
month                 2024-02
winner             Naw, Hazem
tournaments_won             7
Name: 1001, dtype: object
```

برای پاسخ به این سؤال، ابتدا برنده هر مسابقه با استفاده از winner_code مشخص شد. سپس اطلاعات تورنمنت و تاریخ مسابقه به داده‌ها اضافه شد و ماه برگزاری از تاریخ استخراج گردید. برای جلوگیری از شمارش چندباره مسابقات یک تورنمنت، هر تورنمنت فقط یک بار برای هر بازیکن در هر ماه در نظر گرفته شد. نتایج نشان داد که Hazem Naw در فوریه ۲۰۲۴ با کسب ۷ عنوان قهرمانی، بیشترین تعداد قهرمانی در یک ماه را در این مجموعه داده به دست آورده است. بنابراین، او موفق‌ترین بازیکن از نظر تعداد تورنمنت‌های کسب‌شده در یک ماه در این بازه زمانی بوده است.

Q10

Is there a correlation between a player's height and their ranking?

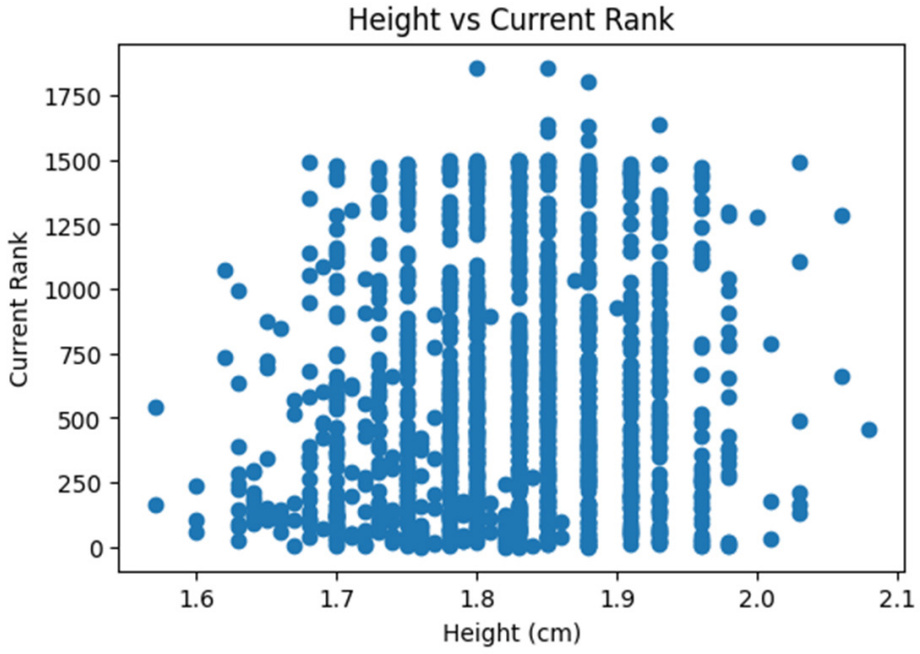
آیا بین قد بازیکن و رتبه او همبستگی وجود دارد؟

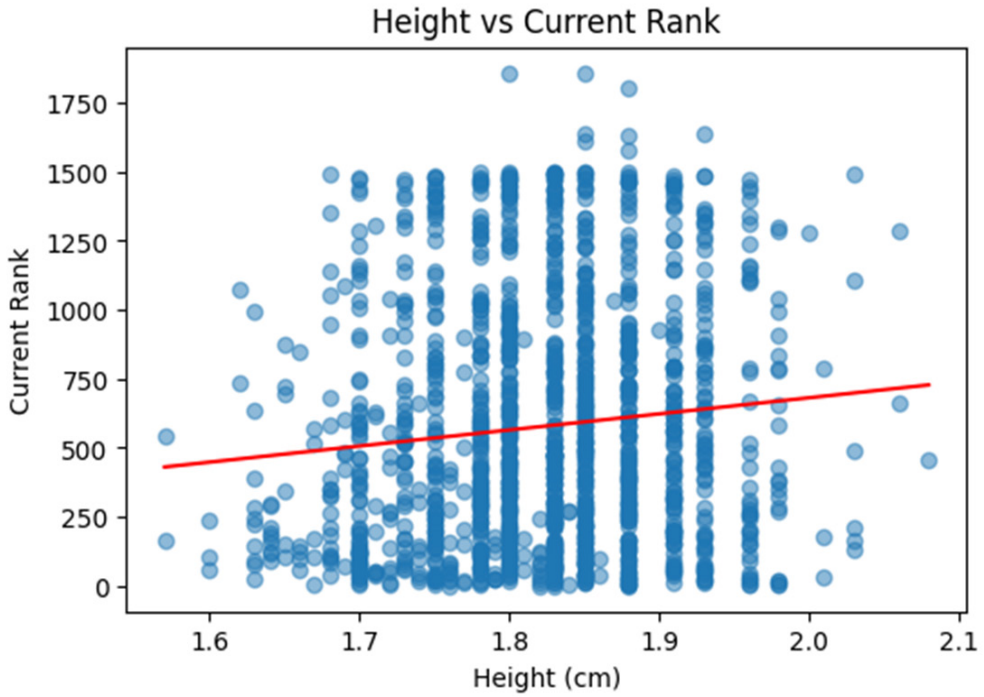
حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال۱۰، ابتدا لازم بود اطلاعات بازیکنان از جداول matchhome_team و matchaway_team استخراج شود تا بتوان رابطه بین قد بازیکنان و رتبه فعلی آن‌ها را بررسی کرد.
ابتدا جدول matchhome_team از دیکشنری data فراخوانی شد و تنها ستون‌های player_id، full_name، height و current_rank انتخاب شدند. این ستون‌ها به ترتیب شامل شناسه یکتای بازیکن، نام کامل، قد و رتبه فعلی بازیکن هستند. همین فرآیند برای جدول matchaway_team نیز انجام شد تا اطلاعات بازیکنان مهمان نیز در دسترس قرار گیرد.
در مرحله بعد، با استفاده از تابع pd.concat() اطلاعات بازیکنان میزبان و مهمان به صورت عمودی با یکدیگر ادغام شدند و در یک DataFrame جدید با نام players ذخیره گردید. برای اینکه شماره ردیف‌ها پس از ادغام به صورت پیوسته از ابتدا شماره‌گذاری شوند، از گزینه ignore_index=True استفاده شد.
از آنجا که هر بازیکن ممکن است در مسابقات مختلف چندین بار حضور داشته باشد، احتمال وجود اطلاعات تکراری وجود داشت. بنابراین با استفاده از تابع drop_duplicates() و بر اساس ستون player_id، رکوردهای تکراری حذف شدند تا برای هر بازیکن تنها یک رکورد در مجموعه داده باقی بماند.
در ادامه، برای انجام صحیح محاسبات آماری، ردیف‌هایی که مقدار height یا current_rank در آن‌ها ثبت نشده بود، با استفاده از تابع dropna() حذف شدند. حذف این داده‌های ناقص باعث می‌شود نتیجه محاسبه همبستگی تحت تأثیر مقادیر خالی قرار نگیرد.
پس از آماده‌سازی داده‌ها، رابطه بین متغیرهای height و current_rank با استفاده از تابع corr() محاسبه شد. این تابع ضریب همبستگی پیرسون را محاسبه می‌کند که میزان رابطه خطی بین دو متغیر را نشان می‌دهد. مقدار این ضریب عددی بین ۱- تا ۱+ است؛ مقادیر نزدیک به ۱+ نشان‌دهنده رابطه مستقیم، مقادیر نزدیک به ۱- نشان‌دهنده رابطه معکوس و مقادیر نزدیک به ۰ بیانگر نبود رابطه خطی قابل توجه بین دو متغیر هستند.
در نهایت، مقدار ضریب همبستگی در متغیری با نام correlation ذخیره شد و با استفاده از دستور print() نمایش داده شد. این مقدار نشان می‌دهد که آیا بین قد بازیکنان تنیس و رتبه فعلی آن‌ها ارتباط آماری معناداری وجود دارد یا خیر و شدت این ارتباط تا چه اندازه است.
پس از محاسبه ضریب همبستگی، برای بررسی بهتر رابطه بین قد بازیکنان و رتبه فعلی آن‌ها، یک نمودار پراکندگی (Scatter Plot) رسم شد.
ابتدا کتابخانه Matplotlib با نام مستعار plt فراخوانی شد تا امکان رسم نمودار فراهم شود.
سپس با استفاده از دستور plt.figure(figsize=(۶,۴)) یک شکل جدید با ابعاد ۶ × ۴ ایجاد شد. تعیین اندازه شکل باعث می‌شود نمودار با ابعاد مناسب نمایش داده شود و خوانایی آن افزایش یابد.
در ادامه، با استفاده از تابع plt.scatter() نمودار پراکندگی رسم شد. در این نمودار، مقادیر ستون height به عنوان محور افقی و مقادیر ستون current_rank به عنوان محور عمودی در نظر گرفته شدند. هر نقطه روی نمودار نمایانگر یک بازیکن است که موقعیت آن بر اساس قد و رتبه فعلی وی تعیین می‌شود.
سپس با استفاده از دستور plt.xlabel(«Height (cm)») عنوان Height (cm) برای محور افقی و با دستور plt.ylabel(«Current Rank») عنوان Current Rank برای محور عمودی تعیین شد تا مفهوم هر محور مشخص باشد.
در مرحله بعد، با استفاده از دستور plt.title(«Height vs Current Rank») عنوان Height vs Current Rank برای نمودار انتخاب شد تا موضوع نمودار به‌طور واضح مشخص شود.
در نهایت، با استفاده از دستور plt.show() نمودار روی صفحه نمایش داده شد.
هدف از رسم این نمودار، مشاهده بصری رابطه بین قد بازیکنان و رتبه فعلی آن‌ها است. در صورتی که نقاط نمودار الگوی مشخصی مانند روند صعودی یا نزولی داشته باشند، می‌توان وجود رابطه بین این دو متغیر را بررسی کرد؛ اما اگر نقاط به‌صورت پراکنده و بدون الگوی مشخص توزیع شده باشند، نشان

می‌دهد که رابطه خطی قابل توجهی بین قد بازیکنان و رتبه فعلی آن‌ها وجود ندارد.
قبل از انجام تحلیل آماری و رسم نمودار، لازم بود اطمینان حاصل شود که مقادیر قد بازیکنان و رتبه فعلی آن‌ها به صورت عددی ذخیره شده‌اند. به همین منظور، ستون‌های height و current_rank با استفاده از تابع pd.to_numeric() به نوع داده عددی تبدیل شدند.
در این مرحله، از پارامتر «errors=coerce» استفاده شد. این گزینه باعث می‌شود اگر مقداری قابل تبدیل به عدد نباشد (مانند متن یا کاراکترهای نامعتبر)، به جای ایجاد خطا، مقدار NaN (مقدار گمشده) در آن سطر قرار گیرد. این کار از بروز خطا در ادامه پردازش داده‌ها جلوگیری می‌کند.
پس از تبدیل نوع داده‌ها، با استفاده از تابع dropna() تمامی ردیف‌هایی که مقدار height یا current_rank در آن‌ها خالی (NaN) بود حذف شدند. حذف این داده‌های ناقص باعث می‌شود محاسبات آماری و رسم نمودار تنها بر اساس داده‌های معتبر انجام شود و نتایج از دقت بیشتری برخوردار باشند.
برای بررسی بهتر رابطه بین قد بازیکنان و رتبه فعلی آن‌ها، علاوه بر رسم نمودار پراکندگی، یک خط روند (Trend Line) نیز روی نمودار رسم شد تا جهت و نوع رابطه بین این دو متغیر به‌صورت بصری مشخص شود.
ابتدا کتابخانه NumPy با نام مستعار np برای انجام محاسبات عددی و کتابخانه Matplotlib با نام مستعار plt برای رسم نمودار فراخوانی شدند.
در ادامه، با استفاده از دستور plt.figure(figsize=(۶,۴)) یک شکل جدید با ابعاد ۶ × ۴ اینچ ایجاد شد تا نمودار با اندازه مناسب نمایش داده شود.
سپس با استفاده از تابع plt.scatter() نمودار پراکندگی رسم شد. در این نمودار، مقادیر ستون height روی محور افقی و مقادیر ستون current_rank روی محور عمودی قرار گرفتند. هر نقطه نمایانگر یک بازیکن است. همچنین با استفاده از پارامتر alpha=۰٫۵ میزان شفافیت نقاط برابر ۰٫۵ در نظر گرفته شد تا در نواحی که نقاط روی یکدیگر قرار می‌گیرند، تراکم داده‌ها بهتر قابل مشاهده باشد.
در مرحله بعد، با استفاده از تابع np.polyfit() یک رگرسیون خطی درجه یک روی داده‌ها برازش داده شد. در این تابع، ستون height به عنوان متغیر مستقل (X) و ستون current_rank به عنوان متغیر وابسته (Y) در نظر گرفته شد و مقدار ۱ نشان می‌دهد که یک خط مستقیم بر روی داده‌ها برازش می‌شود. ضرایب خط به دست آمده در متغیر z ذخیره شدند.
سپس با استفاده از تابع np.poly1d() یک تابع چندجمله‌ای از ضرایب محاسبه‌شده ساخته شد و در متغیر p قرار گرفت. این تابع امکان محاسبه مقدار خط روند را برای هر مقدار قد فراهم می‌کند.
برای اینکه خط روند به صورت منظم از چپ به راست رسم شود، ابتدا مقادیر ستون height با استفاده از تابع np.sort() به ترتیب صعودی مرتب شدند و در متغیر x ذخیره گردیدند.
در ادامه، با استفاده از دستور plt.plot() خط رگرسیون روی نمودار پراکندگی رسم شد. رنگ این خط با استفاده از پارامتر «color=red» به رنگ قرمز انتخاب شد تا از سایر نقاط نمودار قابل تشخیص باشد.
در پایان، با استفاده از دستورات plt.ylabel()، plt.xlabel() و plt.title() عنوان محور افقی، محور عمودی و عنوان نمودار تعیین شد و در نهایت با دستور plt.show() نمودار نمایش داده شد.

```
Correlation: 0.10512440005347558
```





برای بررسی رابطه بین قد بازیکنان و رتبه فعلی (Current Rank)، ضریب همبستگی پیرسون محاسبه شد که مقدار آن ۰٫۱۰۵ به دست آمد. این مقدار بسیار نزدیک به صفر است و نشان می‌دهد که بین قد بازیکنان و رتبه آن‌ها رابطه خطی بسیار ضعیفی وجود دارد. همچنین، نمودار پراکندگی نیز این نتیجه را تأیید می‌کند؛ زیرا نقاط به صورت پراکنده در نمودار توزیع شده‌اند و الگوی مشخصی مشاهده نمی‌شود. اگرچه خط روند شیب مثبت بسیار کمی دارد، اما این شیب آن‌قدر کم است که نمی‌توان آن را به عنوان یک رابطه معنادار در نظر گرفت. بنابراین، بر اساس این داده‌ها، قد بازیکن تأثیر قابل توجهی بر رتبه فعلی او ندارد و احتمالاً عوامل دیگری مانند مهارت، تجربه، آمادگی جسمانی و عملکرد در مسابقات نقش مهم‌تری در تعیین رتبه بازیکنان دارند.

Q11 What is the average duration of matches?

میانگین مدت زمان مسابقات چقدر است؟

حل شده توسط: مرضیه معتمدنیا

برای محاسبه میانگین زمان مسابقه‌ها، از جدول match_time استفاده شد. در این جدول، ستون‌های period_۱ تا period_۵ زمان سپری‌شده در هر ست را نشان می‌دهند. در کوئری زیر، ابتدا این پنج ستون به همراه شرط‌های لازم انتخاب شدند:

```
query = """
    SELECT  period_1,
            period_2,
            period_3,
            period_4,
            period_5
    FROM match_time
    WHERE   period_1 IS NOT NULL AND
            match_id IS NOT NULL
    QUALIFY ROW_NUMBER() OVER (
        PARTITION BY match_id
        ORDER BY match_id
    ) = 1
"""
```

در این مرحله، سطرهایی که مقدار period_۱ آن‌ها null بود حذف شدند تا فقط مسابقه‌هایی باقی بمانند که حداقل اطلاعات مربوط به ست اول آن‌ها ثبت شده است. همچنین با استفاده از ROW_NUMBER() و شرط QUALIFY، برای هر match_id تنها یک رکورد نگه داشته شد تا از تکرار مسابقه‌ها جلوگیری شود.
پس از تبدیل خروجی به DataFrame، مقادیر ستون‌های مربوط به ست‌ها از نظر معقول بودن بررسی شدند. در اینجا فرض شد که اگر زمان ثبت‌شده برای یک ست کمتر از ۳۰۰ ثانیه، یعنی کمتر از ۵ دقیقه باشد، آن مقدار به احتمال زیاد داده پرت یا ثبت نادرست است. به همین دلیل، چنین مقادیری کنار گذاشته شدند. سپس مقادیر null با صفر جایگزین شدند تا بتوان مجموع زمان ست‌های هر مسابقه را محاسبه کرد.
در نهایت، با جمع کردن زمان پنج ست برای هر مسابقه و سپس گرفتن میانگین این مجموع‌ها، میانگین زمان مسابقه‌ها به دست آمد. بر اساس این محاسبه، میانگین زمان هر مسابقه برابر با ۶۸۷۵ ثانیه است که تقریباً معادل ۱ ساعت و ۵۴ دقیقه می‌شود.

Q12

What is the average number of games per set in men's matches compared to women's matches?

میانگین تعداد گیم در هر ست در مسابقات مردان در مقایسه با مسابقات زنان چقدر است؟

حل شده توسط: مرضیه معتمدنیا

برای محاسبه میانگین تعداد گیم‌ها به تفکیک جنسیت، از سه جدول match_، match_home_team، away_team و power استفاده شد. جدول power اطلاعات مربوط به ست‌ها و گیم‌های هر مسابقه را در اختیار قرار می‌دهد و دو جدول دیگر نیز برای تعیین جنسیت بازیکنان home و away به کار گرفته شدند. به همین منظور، این سه جدول بر اساس match_id به یکدیگر متصل شدند تا برای هر مسابقه، تعداد گیم‌های هر ست و جنسیت بازیکنان در دسترس باشد.

در کوئری زیر، برای هر match_id و set_num، بیشترین مقدار game_num محاسبه شده است. این مقدار در عمل نشان‌دهنده تعداد گیم‌های انجام‌شده در آن ست است:

```
query = """
    SELECT p.match_id,
           set_num,
           MAX(game_num) AS game_num,
           h.gender AS h_gender,
           a.gender AS gender
    FROM match_home_team as h
    INNER JOIN match_away_team as a
    ON h.match_id = a.match_id
    INNER JOIN power as p
    ON p.match_id = h.match_id
    GROUP BY p.match_id, set_num, h.gender, a.gender
    QUALIFY ROW_NUMBER() OVER (
        PARTITION BY p.match_id
        ORDER BY p.match_id
    ) = 1

"""
```

پس از اجرای کوئری و تبدیل خروجی به DataFrame، ابتدا بررسی شد که آیا جنسیت بازیکن home و away در هر مسابقه یکسان است یا خیر. از آنجا که هدف سؤال، مقایسه تعداد گیم‌ها در مسابقات زنان و مردان بوده است، ردیف‌هایی که در آن‌ها جنسیت دو طرف متفاوت بود حذف شدند. تعداد این مسابقه‌ها بسیار کم و برابر با ۲۲ مورد بود، اما حذف آن‌ها برای جلوگیری از ورود داده‌های ناسازگار به تحلیل ضروری بود.

از آنجا که پس از این مرحله، دو ستون جنسیت عملاً اطلاعات یکسانی داشتند، یکی از آن‌ها حذف شد تا داده‌ها ساده‌تر و خواناتر شوند. سپس با استفاده از «groupby(«gender» و گرفتن میانگین از ستون game_num، میانگین تعداد گیم‌ها برای هر گروه محاسبه شد.

بر اساس این محاسبه، میانگین تعداد گیم‌ها در مسابقات زنان برابر با ۸٫۹ و در مسابقات مردان برابر با ۹٫۴ به دست آمد.

Q13

What is the distribution of left-handed versus right-handed players?

توزیع بازیکنان چپ‌دست در مقایسه با راست‌دست‌ها چگونه است؟

حل شده توسط: سپیده نیک‌نژاد

برای حل این سوال از جداول MatchHomeTeamInfo و MatchAwayTeamInfo استفاده می‌کنیم. این جداول شامل اطلاعات بازیکنان از جمله راست یا چپ دست بودن آن‌هاست. این دو جدول را با هم concatenate می‌کنیم تا اطلاعات تمامی بازیکنان را داشته باشیم. سپس بر اساس player_id، مقادیر duplicate را حذف می‌کنیم تا در جدول نهایی players، صرفا هر بازیکن ۱ بار ظاهر شود.
در نهایت با یک فیلتر گذاری ساده، تعداد راست و چپ دست‌ها را می‌یابیم. نتیجه به صورت زیر است:

Number of left_handed players	Number of right_handed players
۱۳۳	۱۰۱۳

آمار بازیکنان راست و چپ دست

نکته: همانطور که در سوال ۱ دیدیم، تعداد کل بازیکنان ۲۶۴۴ تا می‌باشد. از آنجایی که مقادیر NaN در ستون plays که نشان‌دهنده راست یا چپ دست بودن است، حذف شده‌اند، پس شمارش بالا صرفا روی نمونه ۱۱۴۶ تایی از بازیکنان انجام شده است.

۱. پس با توجه به جدول بالا و تعداد کل بازیکنان حاضر در مسابقات، ۵ درصد چپ‌دست، ۳۸ درصد راست‌دست و ۵۷ درصد نامعلوم هستند.
۲. اما بین ۱۱۴۶ بازیکن که وضعیت راست یا چپ دست بودنشان مشخص است، ۱۲ درصد چپ‌دست و ۸۸ درصد راست‌دست هستند.

Q14

What is the most common type of surface used in tournaments?

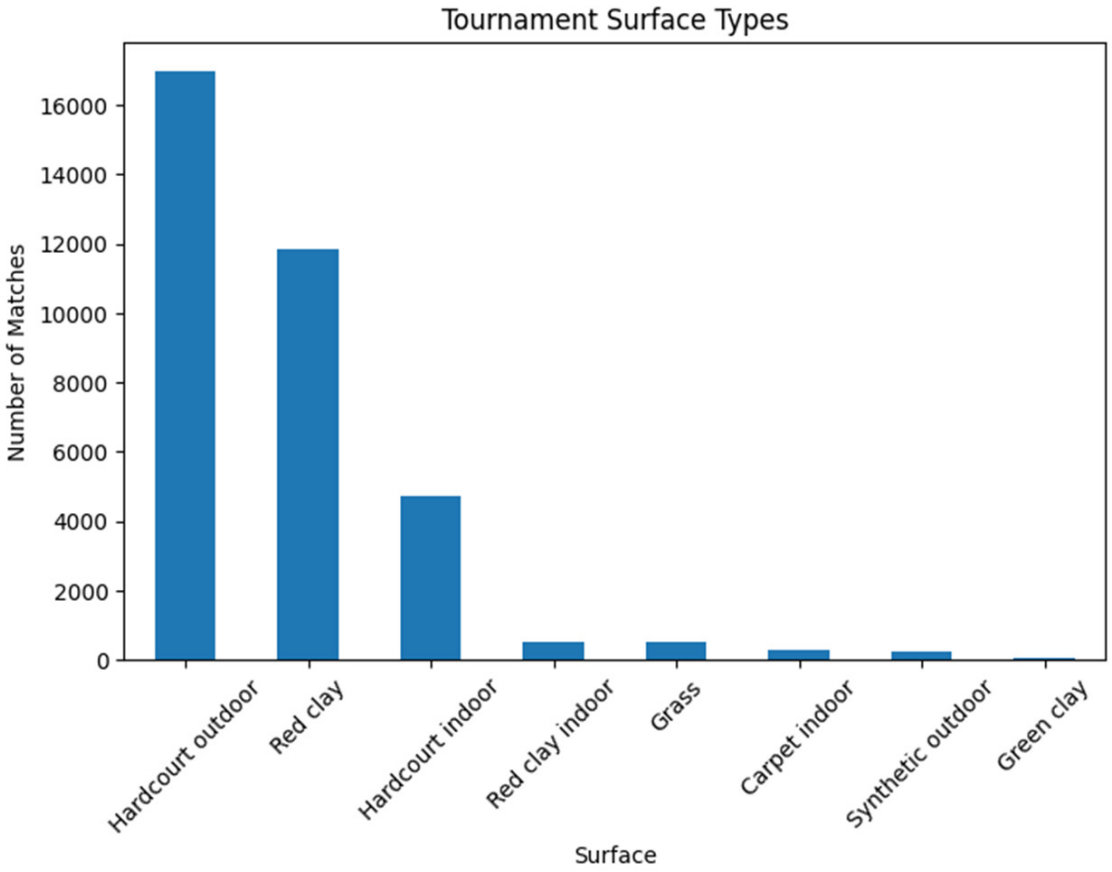
رایج‌ترین نوع زمین مورد استفاده در تورنمنت‌ها چیست؟

حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال۱۴، ابتدا جدول matchtournament از دیکشنری data فراخوانی و در متغیر tournament ذخیره شد. این جدول اطلاعات مربوط به تورنمنت‌های مختلف، از جمله نوع زمین برگزاری مسابقات را در اختیار قرار می‌دهد.
در ادامه، با استفاده از تابع dropna() تمامی ردیف‌هایی که در ستون ground_type مقدار خالی (NaN) داشتند حذف شدند و نتیجه در متغیر ground ذخیره شد. حذف این داده‌های ناقص باعث می‌شود تنها تورنمنت‌هایی که نوع زمین آن‌ها مشخص است در تحلیل مورد استفاده قرار گیرند.
سپس با استفاده از دستور ground[«ground_type»].value_counts() تعداد تکرار هر نوع زمین محاسبه شد. این تابع، مقادیر موجود در ستون ground_type را گروه‌بندی کرده و تعداد دفعات وقوع هر یک را به صورت نزولی محاسبه می‌کند. نتیجه این محاسبه در متغیر surface_count ذخیره شد.
در پایان، با استفاده از دستور print(surface_count) تعداد تورنمنت‌های برگزارشده روی هر نوع زمین نمایش داده شد.
هدف از این مرحله، بررسی توزیع انواع زمین‌های مسابقات تنیس و تعیین تعداد تورنمنت‌هایی است که روی هر یک از سطوح مختلف مانند زمین سخت، زمین خاکی و زمین چمن برگزار شده‌اند. این تحلیل می‌تواند برای مقایسه فراوانی برگزاری مسابقات روی سطوح مختلف مورد استفاده قرار گیرد.
پس از محاسبه تعداد تورنمنت‌های برگزارشده روی هر نوع زمین، برای نمایش بهتر نتایج و مقایسه فراوانی سطوح مختلف، یک نمودار میله‌ای (Bar Chart) رسم شد.
ابتدا کتابخانه Matplotlib با نام مستعار plt فراخوانی شد تا امکان رسم نمودار فراهم شود.

در ادامه، با استفاده از دستور surface_count.plot(kind='bar', figsize=(۸,۵)) یک نمودار میله‌ای از داده‌های موجود در متغیر surface_count رسم شد. در این نمودار، هر میله نشان‌دهنده یک نوع زمین مسابقه است و ارتفاع هر میله تعداد تورنمنت‌های برگزارشده روی آن سطح را نمایش می‌دهد. همچنین با استفاده از پارامتر figsize=(۸,۵) اندازه نمودار برابر ۸ × ۵ اینچ تعیین شد تا نمودار با ابعاد مناسب و خوانایی بیشتر نمایش داده شود.
سپس با استفاده از دستور plt.title('Tournament Surface Types') عنوان Tournament Surface Types برای نمودار تعیین شد تا موضوع آن مشخص باشد.
در مرحله بعد، با استفاده از دستورات plt.xlabel('Surface') و plt.ylabel('Number of Matches') عنوان محور افقی به Surface و عنوان محور عمودی به Number of Matches اختصاص داده شد. محور افقی انواع زمین‌های مسابقه را نمایش می‌دهد و محور عمودی تعداد تورنمنت‌های برگزارشده روی هر سطح را نشان می‌دهد.
برای جلوگیری از هم‌پوشانی برچسب‌های محور افقی و افزایش خوانایی نمودار، از دستور plt.xticks(rotation=۴۵) استفاده شد تا نام انواع زمین با زاویه ۴۵ درجه نمایش داده شوند.
در نهایت، با استفاده از دستور plt.show() نمودار روی صفحه نمایش داده شد.

```
ground_type
Hardcourt outdoor    16959
Red clay             11856
Hardcourt indoor      4741
Red clay indoor        512
Grass                  509
Carpet indoor          276
Synthetic outdoor      226
Green clay              30
Name: count, dtype: int64
```



با بررسی تعداد تورنمنت‌های برگزارشده روی انواع مختلف زمین مشاهده شد که Hardcourt outdoor زمین سخت فضای بازا ۱۶۹۵۹ مسابقه، بیشترین فراوانی را در بین تمامی سطوح دارد. پس از آن، Red clay زمین خاکی با ۱۱۸۵۶ مسابقه و Hardcourt indoor زمین سخت سرپوشیده با ۴۷۴۱ مسابقه در رتبه‌های بعدی قرار دارند.

در مقابل، زمین‌های Synthetic outdoor، Carpet indoor، Grass، Red clay indoor و Green clay تعداد بسیار کمتری از مسابقات را به خود اختصاص داده‌اند. کمترین فراوانی نیز مربوط به Green clay با تنها ۳۰ مسابقه است.
نمودار میله‌ای نیز این اختلاف را به‌خوبی نشان می‌دهد؛ به‌طوری‌که ارتفاع میله مربوط به Hardcourt outdoor به‌طور قابل توجهی بیشتر از سایر سطوح است و بیانگر برگزاری بخش عمده مسابقات روی این نوع زمین است.

Q15

How many distinct countries are represented in the dataset?

چند کشور متمایز در این دیتاست نماینده دارند؟

حل شده توسط: ساناز حکمتی

برای پاسخ به سؤال۱۵، ابتدا اطلاعات مربوط به کشور بازیکنان از جداول matchhome_team و matchaway_team استخراج شد. به همین منظور، از هر دو جدول تنها ستون country انتخاب و در دو متغیر جداگانه با نام‌های home و away ذخیره شد.
در ادامه، با استفاده از تابع pd.concat() اطلاعات بازیکنان میزبان و مهمان به صورت عمودی با یکدیگر ادغام شدند و نتیجه در یک DataFrame جدید با نام countries قرار گرفت. همچنین با استفاده از گزینه ignore_index=True شماره ردیف‌ها پس از ادغام به صورت پیوسته از ابتدا شماره‌گذاری شدند.
از آنجا که برخی از رکوردها فاقد اطلاعات کشور بودند، با استفاده از تابع dropna() تمامی ردیف‌هایی که مقدار ستون country در آن‌ها خالی (NaN) بود حذف شدند. این کار باعث شد تنها اطلاعات معتبر در تحلیل مورد استفاده قرار گیرد.
در مرحله بعد، با استفاده از تابع nunique() تعداد مقادیر یکتای موجود در ستون country محاسبه شد. این تابع تعداد کشورهای متفاوتی را که بازیکنان حاضر در مجموعه داده به آن‌ها تعلق دارند، بدون در نظر گرفتن مقادیر تکراری، محاسبه می‌کند.
در نهایت، نتیجه با استفاده از دستور print() نمایش داده شد.
پس از محاسبه تعداد کشورهای متمایز، برای مشاهده نام تمامی کشورهایی که بازیکنان به آن‌ها تعلق دارند، از دستور countries[«country»].unique() استفاده شد. این تابع تمامی مقادیر یکتا موجود در ستون country را استخراج می‌کند و مقادیر تکراری را حذف می‌کند.
در ادامه، برای اینکه نام کشورها به صورت منظم و قابل خواندن نمایش داده شوند، خروجی تابع unique() با استفاده از تابع sorted() به ترتیب حروف الفبا مرتب شد.
در نهایت، با استفاده از دستور print() فهرست مرتب‌شده کشورهای موجود در مجموعه داده نمایش داده شد.
هدف از این مرحله، نمایش فهرست کامل کشورهای شرکت‌کننده و بررسی تنوع جغرافیایی بازیکنان حاضر در مسابقات تنیس است. مرتب‌سازی نام کشورها باعث می‌شود بررسی و مقایسه آن‌ها آسان‌تر شده و یافتن یک کشور خاص نیز سریع‌تر انجام شود.
پس از شناسایی کشورهای موجود در مجموعه داده، برای بررسی تعداد بازیکنان هر کشور، داده‌های ستون country مورد تحلیل قرار گرفت. در این مرحله، تعداد دفعات تکرار نام هر کشور محاسبه شد تا مشخص شود هر کشور چند بازیکن در مجموعه داده دارد.
سپس نتیجه به صورت یک جدول سازمان‌دهی شد تا اطلاعات به شکل خواناتر و مناسب‌تری نمایش داده شوند. در این جدول، ستون اول نام کشورها و ستون دوم تعداد بازیکنان مربوط به هر کشور را نشان می‌دهد. همچنین داده‌ها به گونه‌ای مرتب شدند که کشورهایی با بیشترین تعداد بازیکن در ابتدای جدول و کشورهایی با تعداد کمتر در انتهای جدول قرار گیرند.
هدف از انجام این مرحله، بررسی توزیع بازیکنان بر اساس کشور و مقایسه میزان حضور کشورهای مختلف در مسابقات است. با استفاده از این جدول می‌توان به‌راحتی تشخیص داد که کدام کشورها بیشترین تعداد بازیکن را در این مجموعه داده دارند و میزان تنوع جغرافیایی بازیکنان حاضر در مسابقات را ارزیابی کرد.

```
         Country  Number of Players
0         France               4270
1          Italy               4057
2            USA               3644
3         Russia               2557
4        Germany               2297
..           ...                ...
96   El Salvador                  4
97      Barbados                  4
98         Qatar                  2
99       Bahamas                  2
100        Nepal                  1

[101 rows x 2 columns]
```

برای بررسی توزیع بازیکنان بر اساس کشور، تعداد بازیکنان هر کشور محاسبه شد. نتایج نشان می‌دهد که فرانسه با ۴,۲۷۰ بازیکن بیشترین تعداد بازیکن را در این مجموعه داده دارد. پس از آن، ایتالیا با ۴,۰۵۷ بازیکن و ایالات متحده آمریکا با ۳,۶۴۴ بازیکن در رتبه‌های دوم و سوم قرار دارند. همچنین روسیه با ۲,۵۵۷ بازیکن و آلمان با ۲,۲۹۷ بازیکن در رتبه‌های بعدی قرار گرفته‌اند.
در مقابل، برخی کشورها سهم بسیار کمی از بازیکنان را در این مجموعه داده دارند. برای مثال، السالوادور و باربادوس هر کدام تنها ۴ بازیکن، قطر ۲ بازیکن و نپال تنها ۱ بازیکن دارند.
این نتایج نشان می‌دهد که توزیع بازیکنان در کشورهای مختلف یکنواخت نیست و بخش عمده بازیکنان متعلق به چند کشور هستند. همچنین، کشورهای اروپایی مانند فرانسه، ایتالیا، آلمان و روسیه حضور پررنگی در این مجموعه داده دارند که می‌تواند نشان‌دهنده گستردگی مسابقات یا تعداد بیشتر بازیکنان حرفه‌ای در این کشورها باشد. در مقابل، کشورهای دارای تعداد بازیکن کمتر تأثیر کمتری بر تحلیل‌های آماری این مجموعه داده خواهند داشت.

Q16

Which player has the highest winning percentage against top 10 ranked opponents?

کدام بازیکن بیشترین درصد برد را مقابل حریفان دارای رتبه ۱۰ برتر داشته است؟

حل شده توسط: سپیده نیک‌نژاد

برای حل این سوال ابتدا جداول MatchHomeTeamInfo، MatchEventInfo و MatchAwayTeamInfo را می‌خوانیم. این جداول شامل اطلاعاتی از جمله کد برنده هر مسابقه، رتبه هر دو تیم حاضر در مسابقه و کد مسابقه است که تمام داده‌های مورد نیازمان را تامین می‌کند. این ۳ جدول را merge می‌کنیم تا اطلاعات هر مسابقه در یک ردیف قرار بگیرد.
نکته: برخورد با تکرار داده، حذف مقادیر NaN و duplicationها انجام شده است.
در ادامه دو جدول دیگر تشکیل می‌دهیم تا اطلاعات مربوط به افراد در جدول AwayTeam و HomeTeam را بر اساس نامشان تفکیک کنیم. با concat کردن این ۲ جدول، در هر سطر نام بازیکن، رنک حریف و برد یا باخت مشخص خواهد شد. با فیلتر رنک حریف، طبق صورت سوال، به جدول نهایی می‌رسیم.
در ادامه با تقسیم تعداد بردها در برابر حریف با رتبه کمتر مساوی ۱۰ بر تعداد کل بازی‌ها در برابر حریف با رتبه کمتر مساوی ۱۰، می‌توانیم موفق ترین بازیکن را بیابیم. نتایج به صورت زیر خواهد بود:

Number of players with opponent ranked <= ۱۰	Mean of number of matches vs opponent ranked <= ۱۰	Mean of number of wins vs opponent ranked <= ۱۰	Mean of win-percentage vs opponent ranked <= ۱۰
۱۳۲	۲.۰۳	۰.۶۱	۲۴.۳۹%

میانگین بازی‌ها، بردها و درصد پیروزی در برابر ۱۰ بازیکن برتر

نتیجه: تعداد زیادی از بازیکنان درصد پیروزی ۱۰۰ دارند، اما **Humbert** با ثبت بیشترین برد (۴ برد)، آمار بیشترین درصد پیروزی و بیشترین برد را دارد. از طرفی این تعداد برد، بیشترین تعداد برد در این جدول است. خوب است بدانیم که **Kalinskaya A** نیز با ۵ بازی و ۴ برد، آمار برد برابری با بهترین بازیکن در این جدول دارد.
این ۲ بازیکن، بیشترین تعداد برد را دارند، اما به دلیل آمار پیروزی ۱۰۰ درصد توسط بازیکن اول، او را به عنوان بهترین بازیکن در برابر ۱۰ نفر برتر، معرفی می‌کنیم.

Top player vs ۱۰ top players	Number of matches vs top ۱۰	Number of wins vs top ۱۰	Win percentage
Humbert U.	۴	۴	۱۰۰%

Q17

What is the average number of breaks of serve per match?

میانگین تعداد break of serve در هر مسابقه چقدر است؟

حل شده توسط: مرضیه معتمدنیا

در این سؤال، کل محاسبه با استفاده از یک کوئری SQL انجام شد و نیازی به پردازش تکمیلی در pandas نبود. هدف، محاسبه میانگین تعداد گیم‌هایی بود که در آن‌ها یک break of serve رخ داده است. برای این منظور، از جدول game_point_by_point استفاده شد.

در اولین مرحله، در یک زیرکوئری، game_id هایی استخراج شدند که در آن‌ها مقدار serving با مقدار scoring برابر نبود. با توجه به بررسی ساختار جدول، این حالت به عنوان نشانه‌ای برای گیم‌هایی در نظر گرفته شد که بازیکن سرویس‌زن در نهایت برنده آن نشده است. برای جلوگیری از شمارش تکراری، داده‌ها بر اساس match_id و game_id گروه‌بندی شدند تا هر گیم فقط یک بار در محاسبه لحاظ شود.
در مرحله بعد، تعداد این گیم‌ها برای هر مسابقه (match_id) شمرده شد و در نهایت با گرفتن میانگین این تعداد در بین همه مسابقه‌ها، مقدار نهایی به دست آمد. بر اساس این محاسبه، میانگین تعداد break of serve در هر مسابقه برابر با ۵٫۴ است.

سوالات تألیفی

Q18

Which match was the most predictable based on votes?

کدام مسابقه را میتوان بر اساس رای کاربران، قابل پیش بینی ترین مسابقه دانست؟

طرح شده توسط: سپیده نیک‌نژاد

برای حل این سوال از جداول MatchVotesInfo و MatchEventInfo استفاده می‌کنیم. از جدول MatchEventInfo، صرفا کد برنده را استخراج می‌کنیم و در ادامه این دو جدول را با هم merge می‌کنیم تا کد بازی، تعداد رای افراد به برنده شدن تیم اول، تعداد رای افراد به برنده شدن تیم دوم و کد تیم برنده را داشته باشیم.
پیش از ادامه کار، مقادیر تکراری و NaN را حذف می‌کنیم. سپس با ایجاد یک سری ستون جدید کار را پیش می‌بریم تا بتوانیم ستون‌های winner_fan_accuracy، loser_votes، winner_votes، total_votes و loser_fan_accuracy را ایجاد کنیم.
ستون winner_fan_accuracy مبنای قضاوت ما خواهد بود.
این جدول داده پرت ندارد اما برای منطقی بودن نتیجه، با توجه به توزیع ستون total_votes، ردیف‌هایی که مقدار loser_votes یا winner_votes کمتر از ۲۰ دارند را حذف می‌کنیم و نتیجه را گزارش می‌کنیم.
نتیجه: ۲ بازی با match_id= ۱۲۰۸۷۳۰۵ و match_id= ۱۲۰۸۷۹۴۲ بالاترین درصد رای درست به برنده بازی را دارند.

Match_id	Total votes	Winner_fan_accuracy	Winner ranking	Loser ranking
۱۲۰۸۷۳۰۵	۲۵۴۲	۹۸.۴۲۶۴۳۶	۲۲	۱۰۹
۱۲۰۸۷۹۴۲	۶۴۳۸	۹۸.۲۳۰۰۸۸	۱۹	۱۰۱

جدول اطلاعات بهترین پیشبینی

با توجه به اینکه دقت پیش‌بینی در این دو مورد بسیار بالا و نزدیک است و از طرفی مورد دوم، بیشترین تعداد رای بین دقت بالای ۹۰ درصد را دارد، پس بازی دوم که بین Baez S. و Varillas J. با شناسه مسابقه ۱۲۰۸۷۹۴۲ را به عنوان بازی با بیشتری دقت در رای‌دهی انتخاب می‌کنیم.

نکته: جالب است بدانیم در بازی با شناسه ۱۲۰۸۷۳۰۳، آمار رای‌دهی با نتیجه مسابقه همخوانی نداشته و ۹۵ درصد افراد به پیروزی بازنده مسابقات رای داده بودند. این مسابقه بین Collins D. با رتبه ۱۵ و Wang Xiy. با رتبه ۶۲ برگزار شده بود و Wang Xiy. با رتبه پایین‌تر، پیروز میدان شد.
به طور میانگین، رای دهی به بازیکنان با رتبه بهتر، بیشتر می‌باشد اما طبق نکته بالا، نتیجه همیشه به نفع بازیکن با ranking بهتر نمی‌شود.

Q19

What is the average BMI of the players?

میانگین شاخص BMI بازیکنان چقدر است؟

طرح شده توسط: مرضیه معتمدنیا

برای محاسبه میانگین شاخص BMI بازیکنان، ابتدا اطلاعات وزن و قد از دو جدول match_home_team و match_away_team استخراج شد. در هر دو کوئری، تنها ردیف‌هایی نگه داشته شدند که مقدار weight و height آن‌ها null نبود و همچنین height بزرگتر از صفر بود تا از خطای تقسیم بر صفر جلوگیری شود.

```
query1 = """
    SELECT
        player_id,
        weight,
        height,
        weight / POWER(height, 2) AS bmi
    FROM match_home_team
    WHERE weight IS NOT NULL
      AND height IS NOT NULL
      AND height > 0
"""
query2 = """
    SELECT
        player_id,
        weight,
        height,
        weight / POWER(height, 2) AS bmi
    FROM match_away_team
    WHERE weight IS NOT NULL
      AND height IS NOT NULL
      AND height > 0
"""
```

پس از اجرای کوئری‌ها و تبدیل خروجی آن‌ها به DataFrame، داده‌های دو جدول با یکدیگر ادغام شدند. از آنجا که ممکن بود یک بازیکن در هر دو جدول یا در چند ردیف تکرار شده باشد، در ادامه با استفاده از drop_duplicates(subset=['player_id']) فقط یک رکورد برای هر بازیکن نگه داشته شد. این کار باعث شد هر بازیکن در محاسبه نهایی فقط یک بار لحاظ شود.

در نهایت، با گرفتن میانگین از ستون bmi، مقدار میانگین شاخص BMI بازیکنان محاسبه شد. بر اساس این محاسبه، میانگین BMI بازیکنان برابر با ۲۲٫۴۷ به دست آمد.

Q20

What percentage of matches do players win when they are in their own country?

زمانی که بازیکنان در کشور خودشان هستند، چند درصد مسابقه را برنده می‌شوند؟

طرح شده توسط: مرضیه معتمدنیا

برای بررسی این سؤال، ابتدا مسابقه‌هایی انتخاب شدند که محل برگزاری آن‌ها با کشور بازیکن یکسان باشد. به این منظور، اطلاعات بازیکنان از جداول match_home_team و match_away_team در یک جدول پایه تجمیع شد و سپس کشور محل برگزاری مسابقه نیز از جدول match_venue استخراج شد. بعد از آن، این داده‌ها با جدول match_event ترکیب شدند تا مقدار winner_code برای هر مسابقه نیز در دسترس باشد.

```
ITH base_table AS (
    SELECT match_id, player_id, country, 1 as type
    FROM match_home_team
    Union
    SELECT match_id, player_id, country, 2 as type
    FROM match_away_team
),
ocation AS (
    SELECT match_id, country AS match_location
    FROM match_venue
    WHERE match_location IS NOT NULL

ELECT DISTINCT b.match_id, b.player_id, b.country, b.type, e.winner_code, l.match_location
ROM match_event AS e
NNER JOIN base_table AS b
N e.match_id = b.match_id
NNER JOIN location AS l
N e.match_id = l.match_id
HERE e.winner_code IS NOT NULL
ND b.country IS NOT NULL
```

در ادامه، مسابقه‌هایی که بیش از یکبار در خروجی ظاهر شده بودند شناسایی شدند و برای جلوگیری از اثر تکرار در محاسبه، این match_id ها حذف شدند. سپس تعداد کل مسابقه‌های معتبر و تعداد بردها محاسبه شد. در این مرحله، اگر مقدار type با winner_code برابر بود، آن مسابقه به عنوان برد بازیکن در نظر گرفته شد.
در نهایت، نسبت تعداد بردها به کل مسابقه‌ها محاسبه و در ۱۰۰ ضرب شد تا درصد نهایی به دست آید. بر اساس این محاسبه، بازیکنان زمانی که در کشور خودشان بازی می‌کنند، در ۵۴٫۸۴۷ درصد مسابقه‌ها برنده می‌شوند.

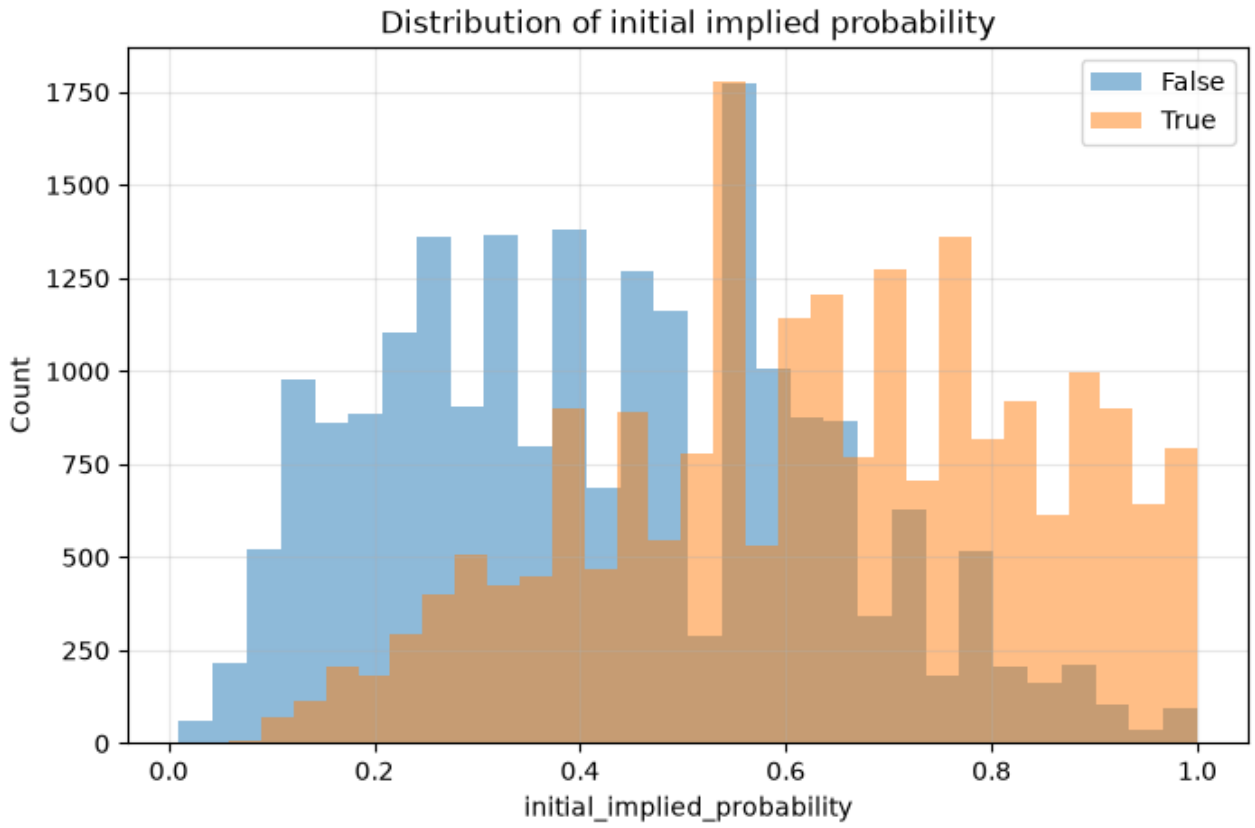
Q21

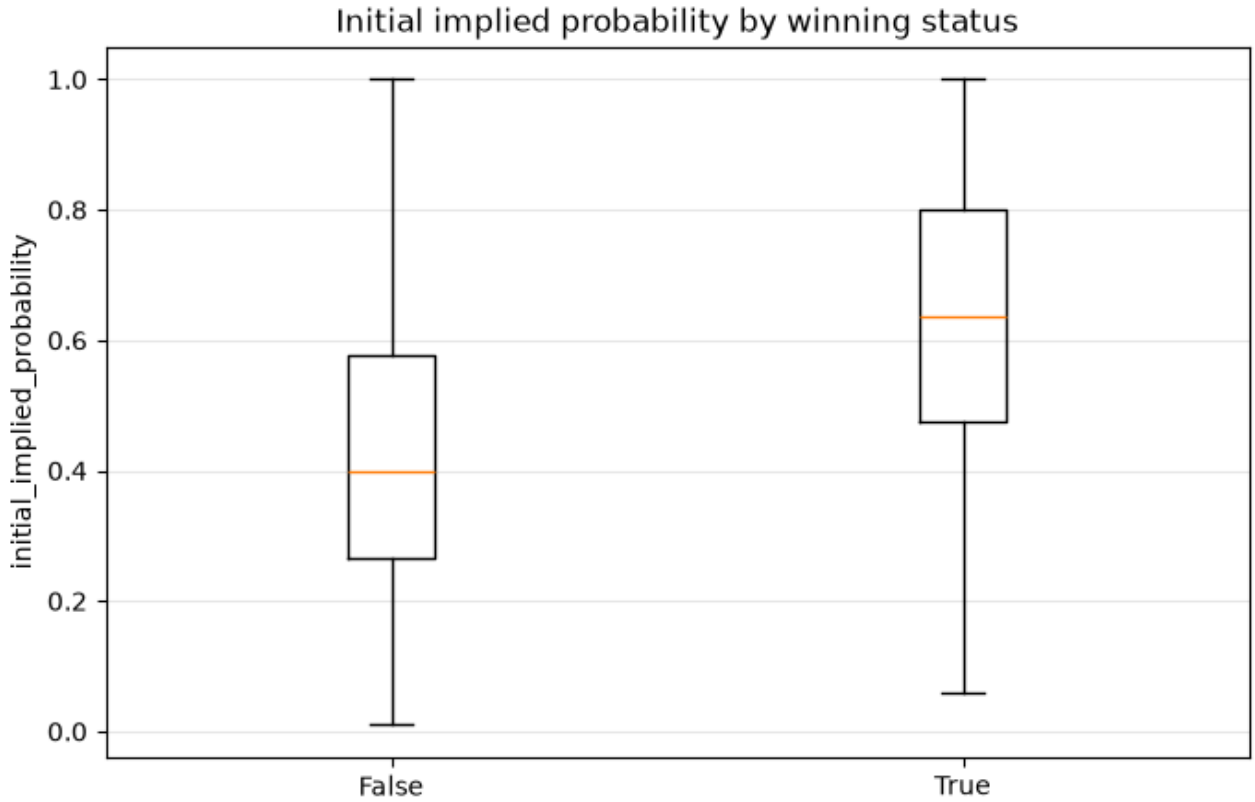
What are the descriptive differences between the initial odds of winners and losers in market_name=full_time?

ضریب اولیه برنده ها با بازنده‌ها در در market_name=full_time از لحاظ توصیفی چه تفاوت‌هایی با یکدیگر دارند؟

طرح شده توسط: مرضیه معتمدنیا

برای پاسخ به این سؤال، ابتدا از جدول odds فقط رکوردهایی انتخاب شدند که مربوط به بازار full_time بودند و برای آن‌ها هم مقدار initial_fractional_value و هم مقدار winnig ثبت شده بود.
از آنجا که initial_fractional_value به‌صورت ضریب کسری ذخیره شده بود، این مقدار ابتدا به ضریب اعشاری و همچنین احتمال ضمنی تبدیل شد. برای این کار دو تابع تعریف شد: یکی برای تبدیل ضریب کسری به ضریب اعشاری، و دیگری برای محاسبه احتمال ضمنی از همان ضریب کسری.
سپس این دو مقدار برای هر رکورد محاسبه شدند و رکوردهایی که تبدیل آن‌ها نامعتبر بود حذف شدند. در ادامه، داده‌ها بر اساس ستون winnig گروه‌بندی شدند و برای هر گروه، شاخص‌های آماری مانند تعداد، میانگین، میانه، انحراف معیار، کمینه و بیشینه برای هر دو متغیر initial_implied_probability و initial_decimal_odds محاسبه شد. (جدول کامل داده‌ها در نوت بوک مربوط به این سوال موجود است.)





Which tournament is the most competitive one ,based on the percentage of matches which finish in the final set?

رقابتی ترین تورنومنت بر اساس درصد بازی هایی که در ست آخر تعیین تکلیف می شوند، کدام است؟

طرح شده توسط: سپیده نیک‌نژاد

Q22

برای حل این سوال از دو جدول GameInfo و MatchTournamentInfo استفاده کرده‌ایم. به این صورت که از جدول مربوط به تورنومنت‌ها، ستون‌های match_id و tournament_name و از جدول مربوط به بازی، مجددا match_id و set_id استخراج شده است. در ستون set_id، شماره آخرین ست هر مسابقه وجود دارد.
در ادامه این دو جدول را merge می‌کنیم و برای هر تورنومنت، تعداد کل بازی‌ها در ستون total_matches و تعداد بازی‌های ۳ سته را در ستون three_set_matches، ایجاد می‌کنیم تا درصد بازی‌هایی که به ست سوم کشیده می‌شوند، پیدا شود.
نکته: با توجه به اینکه میانگین و میانه تعداد بازی‌های هر تورنومنت، عدد ۲۲ است، بیشترین درصد بازی‌هایی با ۳ ست را در ۲ محدوده پیدا می‌کنیم. یک بار در محدوده با تعداد کل بازی‌های کمتر از ۲۲ و بار دیگر، برعکس عمل می‌کنیم. نتایج به صورت زیر خواهند بود:

Tournament name	Total number of matches	Three-set matches	Percentage of three-set matches
Naples, Italy	۲۸	۱۷	۶۰.۷۱۴۲۸۶
Antalya, Turkiye, Qualifying	۱۲	۹	۷۵.۰۰۰۰۰۰

جدول مربوط به رقابتی‌ترین مسابقات

در نهایت، با توجه به اینکه در تورنومنت مربوط به کشور ایتالیا، مسابقات ۳ سته تقریبا ۲ برابرِ همین مقدار در تورنومنت کشور ترکیه است، پس آن‌را، برخلاف کمتر بودن درصد، به عنوان رقابتی‌ترین تورنومنت معرفی می‌کنیم.

Q23

Which tournament had the highest upset(wonder) rate?

در کدام تورنومنت، درصد شگفتی سازی بیشتر بوده است؟(شگفتی سازی یعنی تیمی با رنک بدتر، پیروز شده است.)

طرح شده توسط: سپیده نیک‌نژاد

برای حل این سوال ابتدا از ۳ جدول MatchHomeTeamInfo، MatchEventInfo و MatchAwayTeamInfo استفاده می‌کنیم تا یک جدول با ستون‌های homw_rank، away_rank، match_id و winning_code بسازیم. این جدول به ما کمک می‌کند تا ستون wonder را بسازیم. به این صورت که اگر بازیکن با رنک بدتر، پیروز شود، این ستون True و درغیر این صورت False خواهد شد.
در ادامه data frame حاصل شده را با جدول merge، MatchTournamentInfe می‌کنیم تا برای هر سطر، نام تورنومنت را نیز داشته باشیم. در نهایت برای هر تورنومنت، تعداد کل بازی‌ها و بازی‌های شگفتی‌ساز را محاسبه می‌کنیم تا درصد بازی‌های شگفتی‌ساز، به دست بیاید.
بر اساس توضیحات بالا، جدول را به دست آورده و بر اساس تعداد کل بازی‌ها، فیلتر می‌کنیم.(فقط تورنومنت‌ها با بیش از ۲۰ بازی)
در نهایت تورنومنت **Glasgow,GreatBritain** با **۴۰** بازی، **۲۷** بازی شگفتی‌ساز و درصد شگفتی **۶۷**، انتخاب می‌شود.

Q24

Does winning the first set increase the chances of winning the entire match?

آیا برنده شدن در ست اول، احتمال برنده شدن در کل مسابقه را افزایش می‌دهد؟

طرح شده توسط: ساناز حکمتی

برای بررسی ساختار جدول matchaway_team_score، نام تمام ستون‌های آن نمایش داده شد. این کار به منظور شناسایی ستون‌های موجود و انتخاب ستون‌های مورد نیاز برای تحلیل انجام شد. با استفاده از این مرحله مشخص شد که اطلاعات مربوط به امتیاز بازیکن مهمان، امتیازات هر ست و سایر اطلاعات مرتبط با نتیجه مسابقه در این جدول قرار دارد. بررسی نام ستون‌ها به درک ساختار داده‌ها و انتخاب صحیح ستون‌های مورد نیاز برای ادامه تحلیل کمک می‌کند.
ابتدا جداول matchhome_team_score و matchaway_team_score که شامل امتیازهای بازیکنان میزبان و مهمان در هر ست هستند، از دیکشنری data فراخوانی شدند. همچنین از جدول matchevent تنها ستون‌های match_id و winner_code استخراج شد. ستون winner_code مشخص می‌کند برنده نهایی هر مسابقه کدام بازیکن بوده است.

در مرحله بعد، اطلاعات امتیاز بازیکنان میزبان و مهمان با استفاده از ستون مشترک match_id با یکدیگر ادغام شدند. برای جلوگیری از یکسان بودن نام ستون‌های دو جدول، برای ستون‌های بازیکن میزبان پسوند _home و برای ستون‌های بازیکن مهمان پسوند _away در نظر گرفته شد.
سپس جدول حاصل با جدول matchevent ادغام شد تا اطلاعات مربوط به برنده نهایی هر مسابقه نیز به داده‌ها اضافه شود. این کار امکان مقایسه برنده ست اول با برنده نهایی مسابقه را فراهم کرد.
در ادامه، تمامی مسابقاتی که امتیاز ست اول آن‌ها برای یکی از بازیکنان ثبت نشده بود حذف شدند تا تنها مسابقاتی که اطلاعات کامل داشتند در تحلیل مورد استفاده قرار گیرند.
پس از آماده‌سازی داده‌ها، برنده ست اول هر مسابقه تعیین شد. در این مرحله، امتیاز بازیکن میزبان و مهمان در ست اول با یکدیگر مقایسه شد. اگر امتیاز بازیکن میزبان بیشتر بود، مقدار ۱ و در غیر این صورت مقدار ۲ به عنوان برنده ست اول ثبت شد و نتیجه در ستون جدیدی با نام first_set_winner ذخیره گردید.
در مرحله بعد، برنده ست اول با برنده نهایی مسابقه که در ستون winner_code قرار داشت مقایسه شد. اگر هر دو مقدار یکسان بودند، نتیجه True و در غیر این صورت False در ستون جدیدی با نام same_winner ثبت شد. این ستون مشخص می‌کند که آیا برنده ست اول، برنده نهایی مسابقه نیز بوده است یا خیر.
سپس با محاسبه میانگین مقادیر ستون same_winner و ضرب آن در ۱۰۰، درصد مسابقاتی که در آن‌ها برنده ست اول، برنده نهایی مسابقه نیز بوده است محاسبه شد. این مقدار در متغیری با نام success_rate ذخیره شد.
در پایان، درصد موفقیت برنده ست اول و همچنین تعداد مسابقات هر دو حالت (برنده ست اول همان برنده نهایی بوده است یا خیر) نمایش داده شد.

```
Winning percentage of first-set winner: 82.03%

Distribution:
same_winner
True     122944
False     26933
Name: count, dtype: int64
```

یکی از دلایل بالا بودن درصد (۸۲٫۰۳٪) این است که بخش زیادی از مسابقات موجود در این مجموعه داده به صورت Best of ۳ برگزار شده‌اند. در این نوع مسابقات، بازیکنی که ست اول را می‌برد تنها به یک ست دیگر برای پیروزی نیاز دارد، در حالی که حریف باید دو ست متوالی را برنده شود. بنابراین بردن ست اول شانس پیروزی نهایی را به میزان قابل توجهی افزایش می‌دهد. البته این موضوع به معنای قطعی بودن نتیجه نیست، زیرا حدود ۱۸ درصد از مسابقات نشان می‌دهند که برخی بازیکنان پس از واگذاری ست اول توانسته‌اند مسابقه را به سود خود به پایان برسانند.

Q25

Do right-handed players perform better on grass courts compared to left-handed players?

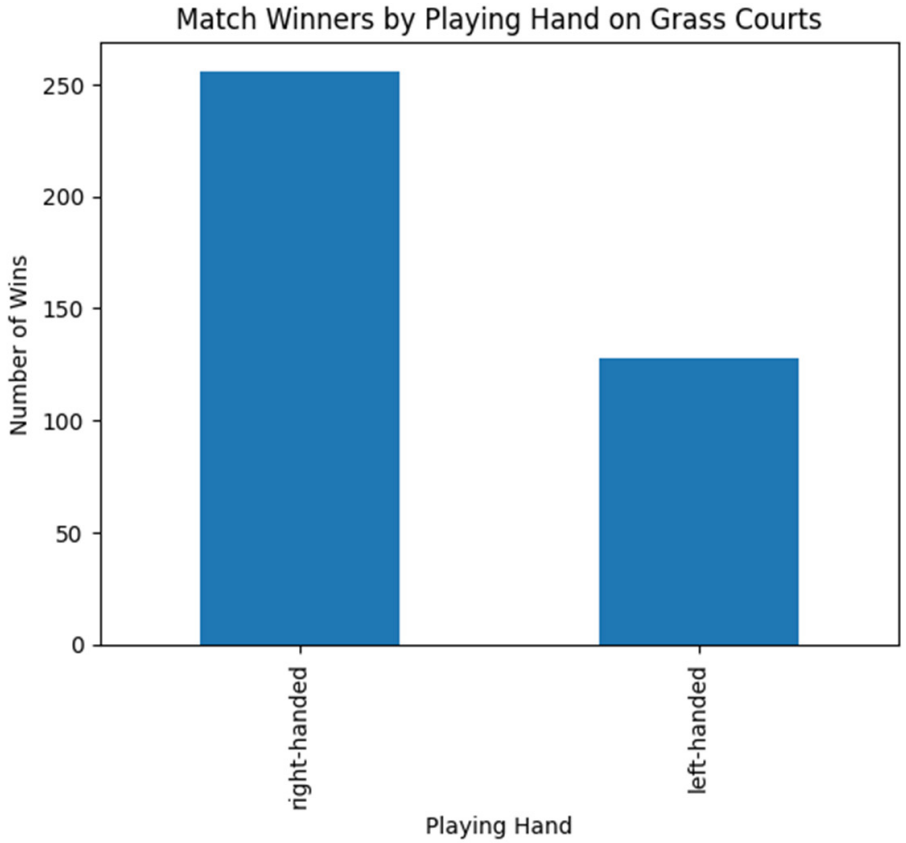
آیا بازیکنان راست‌دست در زمین‌های چمن نسبت به بازیکنان چپ‌دست عملکرد بهتری دارند؟

طرح شده توسط: ساناز حکمتی

برای بررسی ساختار جدول match_home_team، نام تمامی ستون‌های آن نمایش داده شد. این مرحله به منظور آشنایی با ساختار داده‌ها و شناسایی ستون‌های مورد نیاز برای تحلیل انجام شد. با بررسی ستون‌های این جدول، اطلاعاتی مانند شناسه بازیکن (player_id)، نام کامل (full_name)، کشور (country)، قد (height)، دست برتر (plays)، سن (age)، رتبه فعلی (current_rank) و سایر مشخصات بازیکنان قابل شناسایی بود. آگاهی از نام و نوع ستون‌ها، انتخاب صحیح داده‌های مورد نیاز را برای مراحل بعدی تحلیل امکان‌پذیر می‌کند.
برای بررسی ویژگی دست برتر بازیکنان، ستون plays از جدول matchhome_team مورد بررسی قرار گرفت. با استفاده از تابع unique() تمامی مقادیر یکتای موجود در این ستون استخراج و نمایش داده شدند.
ابتدا اطلاعات مربوط به نوع زمین مسابقه از جدول matchtournament و اطلاعات برنده نهایی مسابقه از جدول matchevent استخراج شدند. از جدول matchtournament ستون‌های match_id و ground_type و از جدول matchevent ستون‌های match_id و winner_code انتخاب شدند.
در ادامه، اطلاعات مربوط به دست برتر بازیکنان از جداول matchhome_team و matchaway_team استخراج شد. از هر دو جدول ستون‌های match_id و plays انتخاب شدند. سپس برای جلوگیری از تداخل نام ستون‌ها، نام ستون plays در جدول بازیکنان میزبان به home_hand و در جدول بازیکنان مهمان به away_hand تغییر داده شد.
پس از آماده‌سازی داده‌ها، جدول نوع زمین با جدول اطلاعات مسابقات ادغام شد و سپس اطلاعات بازیکنان میزبان و مهمان نیز بر اساس ستون مشترک match_id به آن اضافه گردید. در نتیجه، برای هر مسابقه اطلاعات مربوط به نوع زمین،

برنده نهایی و دست برتر هر دو بازیکن در یک جدول قرار گرفت.
در مرحله بعد، تنها مسابقاتی انتخاب شدند که در ستون ground_type مقدار Grass داشتند. به این ترتیب، تحلیل تنها روی مسابقات برگزارشده روی زمین چمن انجام شد.
سپس تمامی ردیف‌هایی که اطلاعات دست برتر یکی از بازیکنان در آن‌ها ثبت نشده بود حذف شدند تا تنها داده‌های کامل در تحلیل مورد استفاده قرار گیرند.
در ادامه، با استفاده از مقدار ستون winner_code، دست برتر بازیکن برنده هر مسابقه تعیین شد. اگر مقدار winner_code برابر ۱ بود، دست برتر بازیکن میزبان و در غیر این صورت دست برتر بازیکن مهمان به عنوان دست برتر برنده مسابقه انتخاب شد. نتیجه این مرحله در ستون جدیدی با نام winner_hand ذخیره گردید.
در پایان، با استفاده از تابع value_counts() تعداد پیروزی‌های هر گروه از بازیکنان بر اساس دست برتر محاسبه شد و نتیجه نمایش داده شد.
تعداد پیروزی‌های ثبت‌شده برای هر مقدار موجود در ستون winner_hand به درصد تبدیل شد. برای این منظور، تعداد پیروزی هر گروه بر مجموع کل پیروزی‌ها تقسیم و در ۱۰۰ ضرب شد تا سهم هر گروه از کل مسابقات برگزارشده روی زمین چمن (Grass) به صورت درصدی به دست آید.
پس از محاسبه تعداد پیروزی بازیکنان بر اساس دست برتر، برای نمایش بهتر نتایج و مقایسه بصری عملکرد بازیکنان راست‌دست و چپ‌دست در مسابقات برگزارشده روی زمین چمن، یک نمودار میله‌ای (Bar Chart) رسم شد.
در این نمودار، داده‌های مربوط به تعداد پیروزی‌های هر گروه که در متغیر wins ذخیره شده بودند، مورد استفاده قرار گرفتند. هر میله نشان‌دهنده یکی از گروه‌های راست‌دست (Right-handed) یا چپ‌دست (Left-handed) است و ارتفاع هر میله تعداد مسابقاتی را نشان می‌دهد که بازیکنان آن گروه در زمین چمن با پیروزی به پایان رسانده‌اند.
سپس عنوان Match Winners by Playing Hand on Grass Courts برای نمودار انتخاب شد تا موضوع آن به‌طور واضح مشخص باشد. همچنین عنوان محور افقی Playing Hand تعیین شد که نوع دست برتر بازیکنان را نمایش می‌دهد و عنوان محور عمودی Number of Wins مشخص می‌کند که ارتفاع هر میله بیانگر تعداد پیروزی‌های ثبت‌شده برای هر گروه است.
در نهایت، نمودار نمایش داده شد تا امکان مقایسه تعداد پیروزی بازیکنان راست‌دست و چپ‌دست در مسابقات برگزارشده روی زمین چمن به صورت بصری فراهم شود.

```
winner_hand
right-handed    66.666667
left-handed     33.333333
Name: proportion, dtype: float64
```



بررسی مسابقات برگزارشده روی زمین چمن (Grass) نشان می‌دهد که تعداد پیروزی‌های بازیکنان راست‌دست به‌طور قابل توجهی بیشتر از بازیکنان چپ‌دست است. بر اساس نمودار، بازیکنان راست‌دست حدود ۲۵۵ پیروزی و بازیکنان چپ‌دست حدود ۱۲۷ پیروزی کسب کرده‌اند؛ بنابراین تعداد پیروزی‌های بازیکنان راست‌دست تقریباً دو برابر بازیکنان چپ‌دست است.
با این حال، این نتیجه به‌تنهایی نشان نمی‌دهد که بازیکنان راست‌دست عملکرد بهتری روی زمین چمن دارند. دلیل این موضوع آن است که در دنیای تنیس، تعداد بازیکنان راست‌دست بسیار بیشتر از بازیکنان چپ‌دست است. در نتیجه، طبیعی است که تعداد پیروزی‌های آن‌ها نیز بیشتر باشد.
برای مقایسه منصفانه عملکرد دو گروه، لازم است نرخ پیروزی (Win Rate) هر گروه محاسبه شود؛ یعنی تعداد بردهای هر گروه نسبت به تعداد کل مسابقاتی که همان گروه در آن شرکت کرده است. تنها در این صورت می‌توان درباره برتری عملکرد یکی از دو گروه نتیجه‌گیری کرد.

Q26

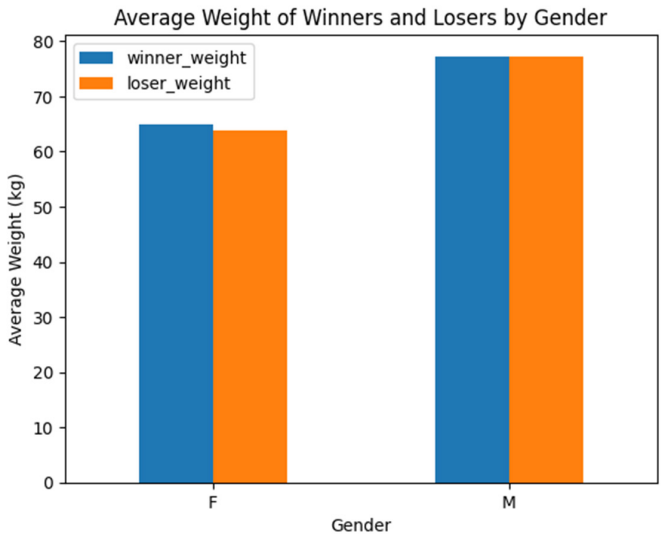
How does the average weight of winning players compare to that of losing players across different gender?

میانگین وزن بازیکنان برنده در مقایسه با بازیکنان بازنده در جنسیت‌های مختلف چگونه است؟

طرح شده توسط: ساناز حکمتی

در ابتدا کتابخانه‌های Pandas و NumPy برای پردازش و تحلیل داده‌ها فراخوانی شدند. سپس جدول matchevent که اطلاعات مربوط به مسابقات را شامل می‌شود، از دیکشنری data استخراج و در متغیر event ذخیره شد. در ادامه، از جدول matchhome_team ستون‌های weight، full_name، match_id و gender انتخاب شدند و برای جلوگیری از تداخل نام ستون‌ها پس از ادغام جداول، نام آن‌ها به ترتیب به home_weight، home_player و home_gender تغییر داده شد. همین عملیات برای جدول matchaway_team نیز انجام شد و ستون‌های مربوط به بازیکن مهمان با نام‌های away_weight، away_player و away_gender ذخیره شدند.
پس از آماده‌سازی داده‌ها، جدول مسابقات با اطلاعات بازیکنان میزبان و مهمان بر اساس ستون مشترک match_id با استفاده از تابع merge() ادغام شد تا تمامی اطلاعات موردنیاز هر مسابقه در یک DataFrame قرار گیرد. سپس با استفاده از تابع np.where() و مقدار ستون winner_code، وزن برنده و بازنده هر مسابقه تعیین شد. اگر مقدار winner_code برابر ۱ بود، وزن بازیکن میزبان به عنوان وزن برنده و وزن بازیکن مهمان به عنوان وزن بازنده ثبت شد و در غیر این صورت این مقادیر برعکس در نظر گرفته شدند. به همین روش، جنسیت بازیکن برنده نیز تعیین و در ستون جدیدی با نام gender ذخیره شد.
در مرحله بعد، رکوردهایی که اطلاعات وزن یا جنسیت آن‌ها ناقص بود، با استفاده از تابع dropna() حذف شدند تا تنها داده‌های کامل در تحلیل استفاده شوند. در نهایت، داده‌ها بر اساس جنسیت با استفاده از تابع groupby() گروه‌بندی شدند و میانگین وزن بازیکنان برنده و بازنده برای هر گروه جنسیتی با تابع mean() محاسبه گردید. همچنین با استفاده از تابع round(۲) نتایج تا دو رقم اعشار گرد شدند و در DataFrame جدیدی با نام result ذخیره شدند. در پایان نیز نتایج با دستور print(result) نمایش داده شد.

```
        winner_weight loser_weight
gender
F           64.978146    63.730972
M           77.217394     77.29284
```



برای بررسی تفاوت میانگین وزن بازیکنان برنده و بازنده به تفکیک جنسیت، میانگین وزن هر دو گروه محاسبه شد. نتایج نشان داد که در بخش زنان (F)، میانگین وزن بازیکنان برنده ۶۴٫۹۸ کیلوگرم و میانگین وزن بازیکنان بازنده ۶۳٫۷۳ کیلوگرم است. بنابراین، برندگان زن به طور متوسط حدود ۱٫۲۵ کیلوگرم سنگین‌تر از بازندگان هستند.
در بخش مردان (M)، میانگین وزن بازیکنان برنده ۷۷٫۲۲ کیلوگرم و میانگین وزن بازیکنان بازنده ۷۷٫۲۹ کیلوگرم به دست آمد. اختلاف بین این دو مقدار تنها حدود ۰٫۰۸ کیلوگرم است که بسیار ناچیز بوده و نشان می‌دهد از نظر وزن، تفاوت محسوسی بین بازیکنان برنده و بازنده مرد وجود ندارد.
در مجموع، نتایج این تحلیل نشان می‌دهد که وزن بازیکنان به تنهایی تأثیر قابل توجهی بر نتیجه مسابقه ندارد. اگرچه در بخش زنان برندگان به طور متوسط اندکی سنگین‌تر از بازندگان هستند، اما این اختلاف کم است و برای نتیجه‌گیری قطعی درباره تأثیر وزن بر پیروزی، انجام آزمون‌های آماری مانند آزمون t برای بررسی معناداری این اختلاف ضروری است.