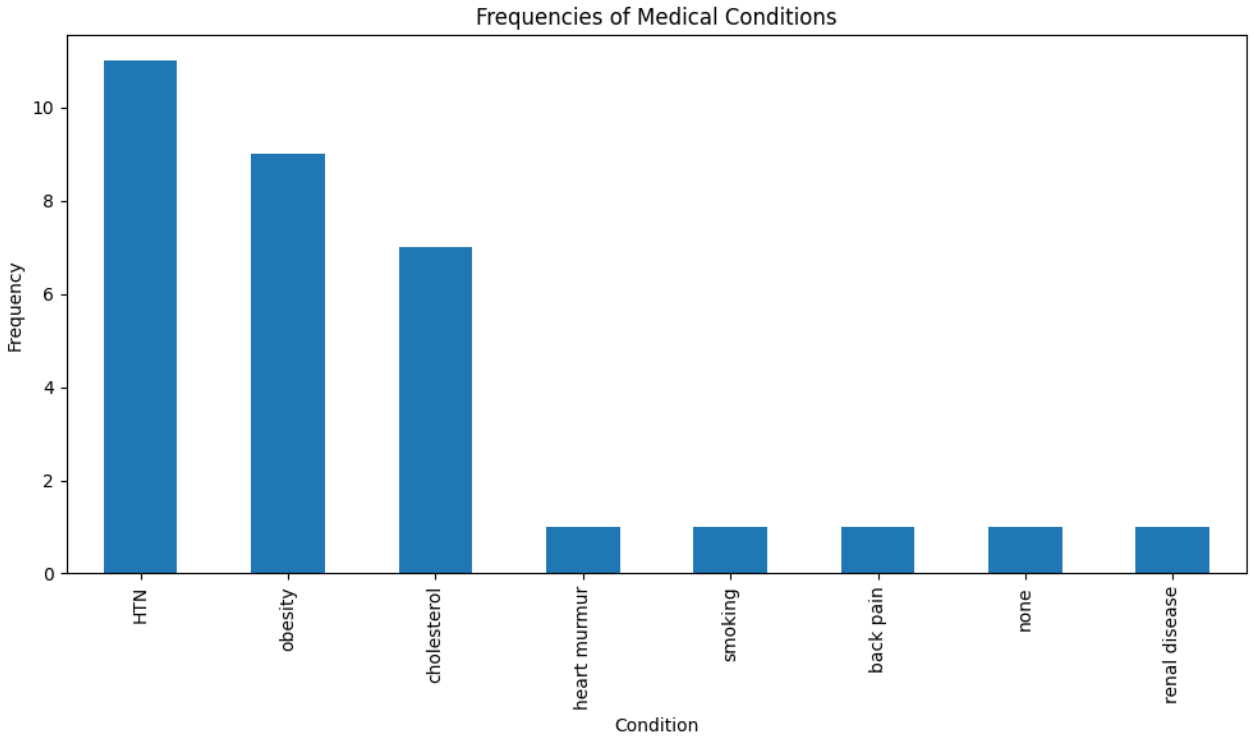
Same day Discharge

```
# Rettenmaier, et al 2012 Comorbidities
import pandas as pd
import matplotlib.pyplot as plt

# Step 2: Define the conditions
conditions = [
    'cholesterol',
    'HTN', 'cholesterol',
    'heart murmur',
    'HTN', 'cholesterol',
    'HTN', 'cholesterol',
    'HTN', 'cholesterol',
    'smoking',
    'obesity',
    'back pain',
    'HTN', 'obesity',
    'HTN', 'obesity',
    'cholesterol', 'obesity',
    'HTN', 'obesity',
    'HTN',
    'HTN', 'obesity',
    'cholesterol',
    'obesity',
    'none',
    'HTN', 'renal disease',
    'HTN', 'obesity',
    'obesity'
]

# Step 3: Convert the list into a pandas Series to easily calculate f
conditions_series = pd.Series(conditions)

# Step 4: Calculate the frequency of each unique condition
condition_frequencies = conditions_series.value_counts()

# Step 5: Display the frequencies
print("Frequencies of Conditions:")
print(condition_frequencies, condition_frequencies*100/21)

# Step 6: Plot the frequencies
plt.figure(figsize=(10, 6))
condition_frequencies.plot(kind='bar')
plt.title('Frequencies of Medical Conditions')
```

```
plt.xlabel('Condition')
plt.ylabel('Frequency')
plt.xticks(rotation=90)  # Rotate the x labels for better visibility
plt.tight_layout()
plt.show()
```

```
Frequencies of Conditions:
HTN               11
obesity            9
cholesterol        7
heart murmur       1
smoking            1
back pain          1
none               1
renal disease      1
Name: count, dtype: int64 HTN               52.380952
obesity           42.857143
cholesterol       33.333333
heart murmur       4.761905
smoking            4.761905
back pain          4.761905
none               4.761905
renal disease      4.761905
Name: count, dtype: float64
```



```
#Lee, et al. 2016
print ((729-36-435-256)*100/729, 'ASA>3 or unknown')
print ((36)*100/729, 'ASA 1')
print ((435)*100/729, 'ASA 2')
print ((256)*100/729, 'ASA 2')

print (100/729, 'MI postop')
```

```
0.27434842249657065 ASA>3 or unknown
4.938271604938271 ASA 1
59.67078189300412 ASA 2
35.116598079561044 ASA 2
0.13717421124828533 MI postop
```

```
# Praiss

print (242+577+696+254+59)

print ((13)*100/1828, '• Anterior repair  ')
print ((2 )*100/1828, ' • Posterior repair:  ')
print ((4)*100/1828, '• Incontinence repair:')
print ((23 )*100/1828, 'Oophorectomy  ')
print ((2 )*100/1828, 'Colpopexy')
print ((764 )*100/1828, 'Lymphadenectomy')

print ((242)*100/1828, '<50 years')
print ((577)*100/1828, '50-59 years')
print ((696)*100/1828, '60-69 years')
print ((254)*100/1828, '70-79 years')
print ((59)*100/1828, '≥80 years ')

print ((374)*100/1828, 'Normal')
print ((413 )*100/1828, 'Overweight')
print ((1032)*100/1828, 'Obese')
print ((9 )*100/1828, 'Unknown')


print ((1828-1522), (1828-1522)*100/1828, 'Diabetes')
print (13*100/1828, 'o  COPD')
print (2*100/1828, 'o   o   Ascites')
```

```
print (1*100/1828, 'o   o   Congestive heart failure')
print (823*100/1828, 'o o   HTN on medication')
print (1*100/1828, 'o   o   Acute renal failure')
print (1*100/1828, 'o   o   Dialysis')
print (1*100/1828, 'o   o   Open wound')
print (28*100/1828, 'o  o   Steroid use')
print (6*100/1828, 'o   o   Weight loss')
print (11*100/1828, 'o  o   Bleeding disorder')
print (3*100/1828, 'o   o   Transfusion')
print (141*100/1828, 'Smoking')


print (23*100/1828, '•  Albumin 1')
print (222*100/1828, '• Albumin 2')
print (502*100/1828, '• Albumin 3')
print (1081*100/1828, '•    Albumin 4')

print (63*100/1828, 'asa 1')
print (1007*100/1828, 'asa 2')
print (745*100/1828, 'asa 3')
print (14*100/1828, 'asa 4')

print (702*100/1828, 'o Low')
print (497 *100/1828, 'o    o   Medium Low')
print (374*100/1828, 'o Medium high')
print (255 *100/1828, 'high')

print (42 *100/1828, 're-admission')
```

```
1828
0.7111597374179431 • Anterior repair
0.10940919037199125  • Posterior repair:
0.2188183807439825 • Incontinence repair:
1.2582056892778994 Oophorectomy
0.10940919037199125 Colpopexy
41.794310722100654 Lymphadenectomy
13.238512035010942 <50 years
31.564551422319475 50–59 years
38.074398249452955 60–69 years
13.894967177242888 70–79 years
3.227571115973742 ≥80 years
20.45951859956236 Normal
22.592997811816193 Overweight
56.455142231947484 Obese
0.4923413566739606 Unknown
306 16.739606126914662 Diabetes
0.7111597374179431 o    COPD
0.10940919037199125 o   o       Ascites
0.05470459518599562 o   o       Congestive heart failure
```

```
45.0218818380744 o      o       HTN on medication
0.05470459518599562 o   o       Acute renal failure
0.05470459518599562 o   o       Dialysis
0.05470459518599562 o   o       Open wound
1.5317286652078774 o    o       Steroid use
0.3282275711159737 o    o       Weight loss
0.6017505470459519 o    o       Bleeding disorder
0.16411378555798686 o   o       Transfusion
7.713347921225383 Smoking
1.2582056892778994 •    Albumin 1
12.144420131291028 •    Albumin 2
27.461706783369802 •    Albumin 3
59.13566739606127 •     Albumin 4
3.446389496717724 asa 1
55.08752735229759 asa 2
40.754923413566736 asa 3
0.7658643326039387 asa 4
38.40262582056893 o     Low
27.188183807439824 o    o       Medium Low
20.45951859956236 o     Medium high
13.949671772428884 high
2.297592997811816 re-admission
```

```
# Cappuccio  et al
print (35+297+1860+7553)
print (35*100/9745, 'Abdo')
print (297*100/9745, 'Laparoscopic')
print (1860*100/9745, 'Robotic')
print (7553*100/9745, 'Vaginal')
print (9745*2/100, "readmission")
```

```
9745
0.35915854284248333 Abdo
3.047716777834787 Laparoscopic
19.08671113391483 Robotic
77.5064135454079 Vaginal
194.9 readmission
```

Start coding or generate with AI.

```
#Tait et al
number_SLN = 93 * 49.5 / 100
print(f"The number corresponding to 49.5% of 93 is: {number_SLN}")

number_TLN = 63 * 15.9 / 100
print(f"The number corresponding to 15.9% of 63 is: {number_TLN}")

print (number_SLN+number_TLN, 'total')

print (46*100/56, 'number_SLN')
print (10*100/56, 'number_TLN')
```

```
The number corresponding to 49.5% of 93 is: 46.035
The number corresponding to 15.9% of 63 is: 10.017000000000001
56.052 total
82.14285714285714 number_SLN
17.857142857142858 number_TLN
```

```
#Dioun et al
print (6164+14266+11024)

print (6164*100/31454, 'SLN')
print (14266*100/31454, 'LND')
print (11024*100/31454, 'no LND')
```

```
31454
19.59687162205125 SLN
45.35512176511732 LND
35.048006612831436 no LND
```

```
#Lees et al
print (32+50, 'total')
print (32*100/82, 'Pre-COVID')
print (50*100/82, 'COVID')
```

```
82 total
39.02439024390244 Pre-COVID
60.97560975609756 COVID
```

```
# Cappuccio et al

print ((774)*100/9745, 'age 18-44 years')
print ((1615)*100/9745, 'age 45-54 years')
print ((3793)*100/9745, 'age 55-64 years')
print ((3563)*100/9745, 'age ≥65 years')

print ((6597)*100/9745, 'Not Overweight or obese:')
print ((37)*100/9745, 'Overweight:')
print ((1231)*100/9745, 'Obese:  : ')
print ((1880)*100/9745, ' Morbidly obese: : ')


print ((6108)*100/9745, ' CCI<3:')
print ((2341)*100/9745, ' CCI=3 :')
print ((1296)*100/9745, '   CCI>3:')

print ((366)*100/9745, 'Uterus bleeding')
```

```
7.942534633145202 age 18-44 years
16.572601334017445 age 45-54 years
38.92252437147255 age 55-64 years
36.5623396613648 age ≥65 years
67.69625448948179 Not Overweight or obese:
0.3796818881477681 Overweight:
12.63211903540277 Obese:  :
19.291944586967677  Morbidly obese: :
62.67829656233966  CCI<3:
24.022575679835814  CCI=3 :
13.299127757824525      CCI>3:
3.7557721908671113 Uterus bleeding
```

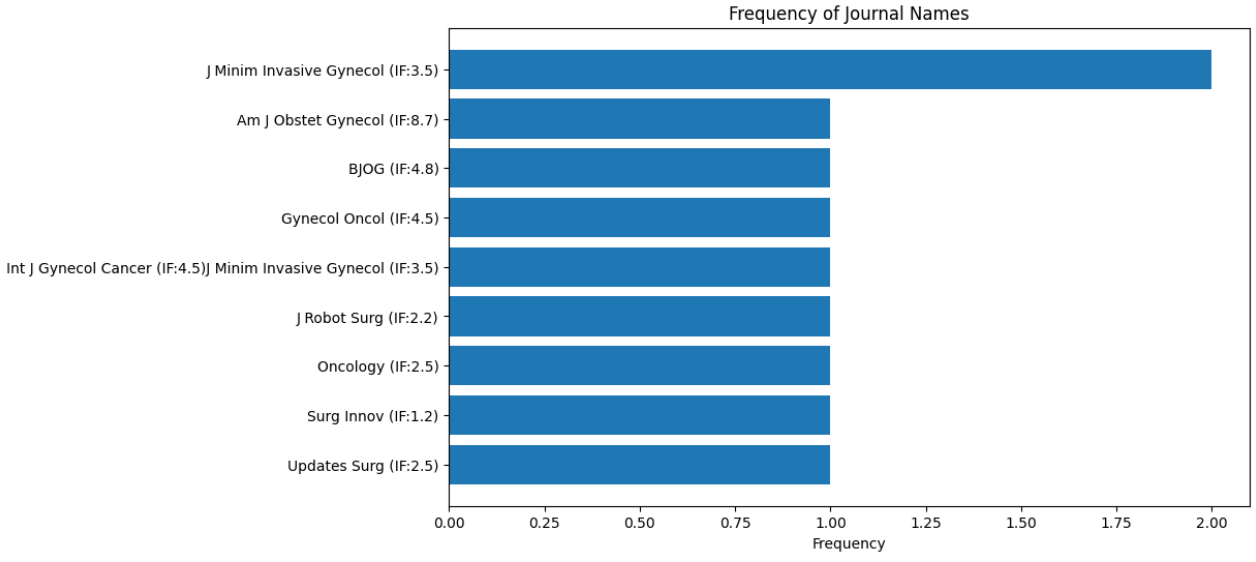
```
#Giannini et al
100/117
```

```
0.8547008547008547
```

```
# Figure on the journals

# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# List of journal names
journals = [
    "Am J Obstet Gynecol (IF:8.7)", "BJOG (IF:4.8)", "Gynecol Oncol (I
    "Int J Gynecol Cancer (IF:4.5)"
    "J Minim Invasive Gynecol (IF:3.5)", "J Minim Invasive Gynecol (II
```

```
    "J Minim Invasive Gynecol (IF:3.5)", "J Robot Surg (IF:2.2)",
    "Oncology (IF:2.5)", "Surg Innov (IF:1.2)", "Updates Surg (IF:2.5

# Create a DataFrame from the journal list
journal_df = pd.DataFrame(journals, columns=["Journal"])

# Count the frequency of each journal
journal_counts = journal_df["Journal"].value_counts().reset_index()
journal_counts.columns = ["Journal", "Frequency"]

# Display the table
print(journal_counts)

# Plot the frequency as a bar chart
plt.figure(figsize=(10, 6))
plt.barh(journal_counts["Journal"], journal_counts["Frequency"])
plt.xlabel("Frequency")
plt.title("Frequency of Journal Names")
plt.gca().invert_yaxis()  # Invert y-axis to show the highest frequen
plt.show()
```

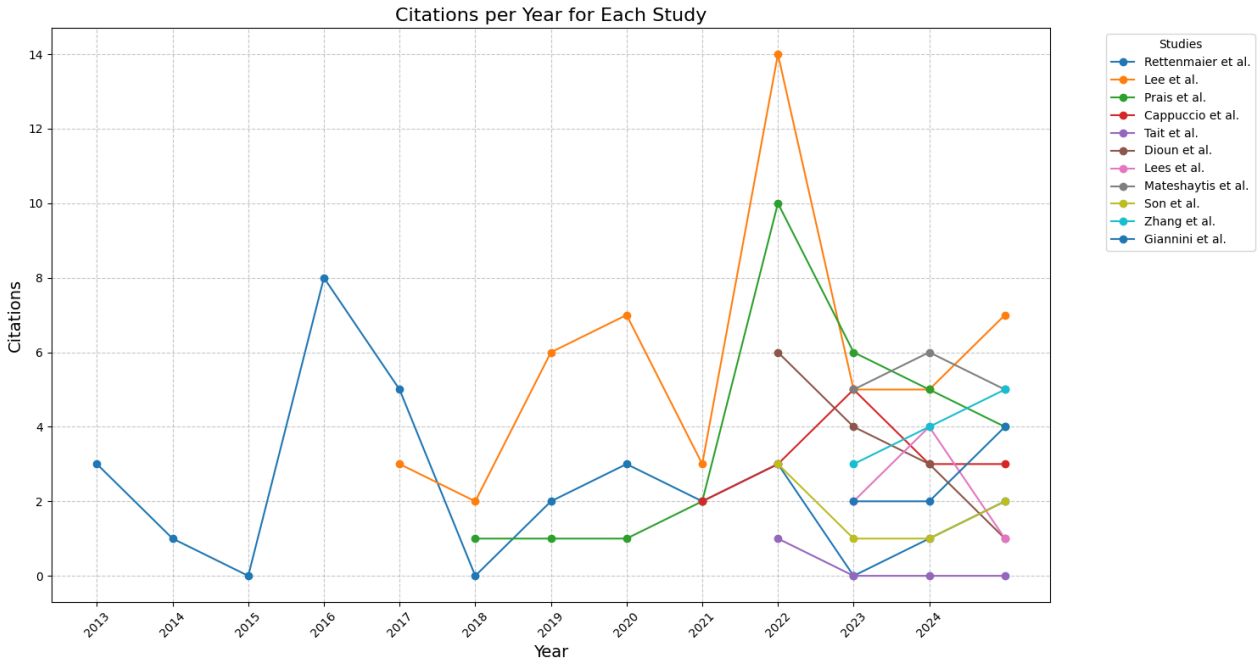
```
                                             Journal  Frequency
0                    J Minim Invasive Gynecol (IF:3.5)          2
1                         Am J Obstet Gynecol (IF:8.7)          1
2                                       BJOG (IF:4.8)          1
3                                Gynecol Oncol (IF:4.5)          1
4  Int J Gynecol Cancer (IF:4.5)J Minim Invasive ...            1
5                                J Robot Surg (IF:2.2)          1
6                                    Oncology (IF:2.5)          1
7                                  Surg Innov (IF:1.2)          1
8                                  Updates Surg (IF:2.5)          1
```



```
# Citations analysis
import matplotlib.pyplot as plt
import numpy as np

# Data input
studies = [
    {"name": "Rettenmaier et al.", "years": list(range(2013, 2026)), "
    {"name": "Lee et al.", "years": list(range(2017, 2026)), "citatio
    {"name": "Prais et al.", "years": list(range(2018, 2026)), "citat
    {"name": "Cappuccio et al.", "years": list(range(2021, 2026)), "c
    {"name": "Tait et al.", "years": list(range(2022, 2026)), "citati
    {"name": "Dioun et al.", "years": list(range(2022, 2026)), "citat
    {"name": "Lees et al.", "years": list(range(2023, 2026)), "citati
    {"name": "Mateshaytis et al.", "years": list(range(2023, 2026)), "
    {"name": "Son et al.", "years": list(range(2022, 2026)), "citatio
    {"name": "Zhang et al.", "years": list(range(2023, 2026)), "citat
    {"name": "Giannini et al.", "years": list(range(2023, 2026)), "ci
]

# Plotting
plt.figure(figsize=(15, 8))

for study in studies:
    years = study["years"]
    citations = study["citations"]
    plt.plot(years, citations, marker='o', label=study["name"])

# Customizing plot
plt.title("Citations per Year for Each Study", fontsize=16)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Citations", fontsize=14)
plt.xticks(np.arange(2013, 2025), rotation=45)
plt.legend(loc="upper left", bbox_to_anchor=(1.05, 1), title="Studies"
plt.grid(True, linestyle="--", alpha=0.7)
plt.tight_layout()

# Mount first (safe to do anytime before save)
from google.colab import drive
drive.mount('/content/drive')
```

```
# SAVE BEFORE SHOW!
plt.savefig("/content/drive/MyDrive/citations_per_year_600.png", dpi=(

# Now show it
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount
```




```
print("total citations", (30+ 52+ 30+ 16+ 1+ 14+ 7+ 16+ 7+ 12+8))
```

```
total citations 193
```

Close

```
Rettenmaier et al.
 Total citation: 30
2013: 3
```

Rettenmaier et al. Total citation: 30 2013: 3 2014: 1 2015: 0 2016: 8 2017: 5 2018: 0

2014: 1
2015: 0
2016: 8
2017: 5
2018: 0
2019: 2
2020: 3
2021: 2
2022: 3
2023: 0
2024: 1
2025: 2

Lee et al.
Total citation: 52
2017: 3
2018: 2
2019: 6
2020: 7
2021: 3
2022: 14
2023: 5
2024: 5
2025: 7

Prais et al.
Total citation: 30
2018: 1
2019: 1
2020: 1
2021: 2
2022: 10
2023: 6
2024: 5
2025: 4

Cappuccio et al:
Total citation: 16
2021: 2
2022: 3
2023: 5
2024: 3
2025: 3

Tait et al.
Total citation: 1
2022: 1

2019: 2 2020: 3 2021: 2 2022: 3 2023: 0 2024: 1 2025: 2

Lee et al. Total citation: 52 2017: 3 2018: 2 2019: 6 2020: 7 2021: 3 2022: 14 2023: 5 2024: 5 2025: 7

Prais et al. Total citation: 30 2018: 1 2019: 1 2020: 1 2021: 2 2022: 10 2023: 6 2024: 5 2025: 4

Cappuccio et al: Total citation: 16 2021: 2 2022: 3 2023: 5 2024: 3 2025: 3

Tait et al. Total citation: 1 2022: 1

Dioun et al. Total citation: 14 2022: 6 2023: 4 2024: 3 2025: 1

Lees et al. Total citation: 7 2023: 2 2024: 4 2025: 1

Mateshaytis et al. Total citation: 16 2023: 5 2024: 6 2025:5

Son et al. Total citation: 7 2022: 3 2023: 1 2024: 1 2025: 2

Zhang et al. Total citation: 12 2023: 3 2024: 4 2025: 5

Giannini et al. Total citation: 8 2023: 2 2024: 2 2025: 4

Dioun et al.
Total citation: 14
2022: 6
2023: 4
2024: 3
2025: 1

Lees et al.
Total citation: 7
2023: 2
2024: 4
2025: 1

Mateshaytis et al.
Total citation: 16
2023: 5
2024: 6
2025:5

Son et al.
Total citation: 7
2022: 3
2023: 1
2024: 1
2025: 2

Zhang et al.
Total citation: 12
2023: 3
2024: 4
2025: 5

Giannini et al.
Total citation: 8
2023: 2
2024: 2
2025: 4

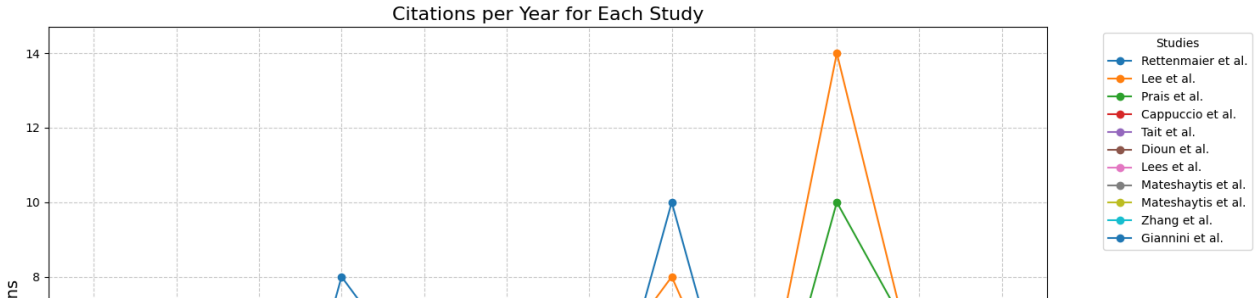
```
print("total patients", 21+729+1828+9745+56+31454+82+66+132+117,
      "max",max([21, 729, 1828, 9745, 56, 31454, 82, 66, 132, 117]),
      "min", min([21, 729, 1828, 9745, 56, 31454, 82, 66, 132, 117]),
      "range", max([21, 729, 1828, 9745, 56, 31454, 82, 66, 132, 117]


print("single center", 600/11)
print("multi center", 500/11)
print("retrospective", 1000/11)
print("ERAS", 400/11 )

print(1000/11, "from USA")
print(100/11, "from Canada")

print(900/11, "MIS")
print(200/11, "Laparoscopy")
print(200/11, "robotic")
print(35/44230, "abdominal")
print(297/44230, "vaginal")

print((46+6164+57+93+58)*100/44230, "% SLN", (46+6164+57+93+58), "Tota
```

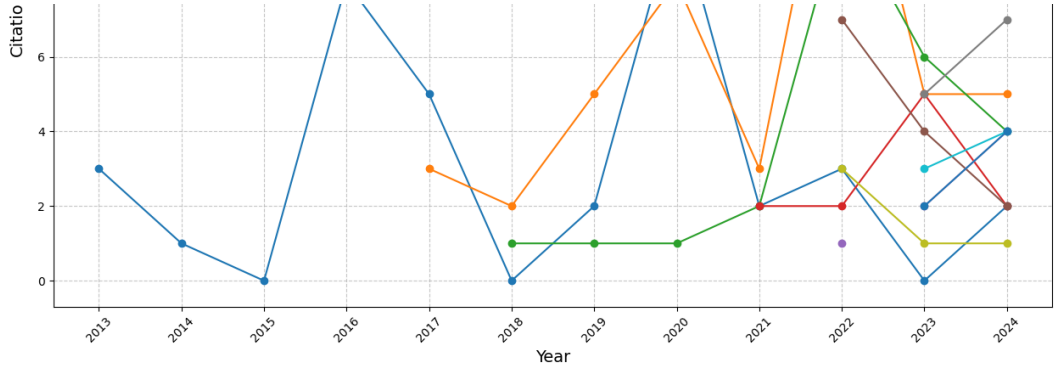
```
total patients 44230 max 31454 min 21 range 31433
single center 54.54545454545455
multi center 45.45454545454545
retrospective 90.9090909090909
ERAS 36.36363636363637
90.9090909090909 from USA
9.090909090909092 from Canada
81.81818181818181 MIS
18.181818181818183 Laparoscopy
18.181818181818183 robotic
0.0079713181098801719 abdominal
0.006714899389554601 vaginal
14.510513226316979 % SLN 6418 Total sln
```

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# (Optional) Define a folder in your Google Drive to save outputs
save_path = '/content/drive/MyDrive/ColabOutputs'
import os
os.makedirs(save_path, exist_ok=True)

# --- Your plotting code ---
```

```
import matplotlib.pyplot as plt
import numpy as np

# Data input
studies = [
    {"name": "Rettenmaier et al.", "years": list(range(2013, 2025)), '
    {"name": "Lee et al.", "years": list(range(2017, 2025)), "citatio
    {"name": "Prais et al.", "years": list(range(2018, 2025)), "citat
    {"name": "Cappuccio et al.", "years": list(range(2021, 2025)), "c
    {"name": "Tait et al.", "years": [2022], "citations": [1]},
    {"name": "Dioun et al.", "years": list(range(2022, 2025)), "citat
    {"name": "Lees et al.", "years": list(range(2023, 2025)), "citati
    {"name": "Mateshaytis et al.", "years": list(range(2023, 2025)), '
    {"name": "Mateshaytis et al.", "years": list(range(2022, 2025)), '
    {"name": "Zhang et al.", "years": list(range(2023, 2025)), "citat
    {"name": "Giannini et al.", "years": list(range(2023, 2025)), "ci
]

# Plotting
plt.figure(figsize=(15, 8))
for study in studies:
    years = study["years"]
    citations = study["citations"]
    plt.plot(years, citations, marker='o', label=study["name"])

plt.title("Citations per Year for Each Study", fontsize=16)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Citations", fontsize=14)
plt.xticks(np.arange(2013, 2025), rotation=45)
plt.legend(loc="upper left", bbox_to_anchor=(1.05, 1), title="Studies'
plt.grid(True, linestyle="--", alpha=0.7)
plt.tight_layout()

# Save the figure directly to Google Drive
save_file = f"{save_path}/citations_per_year.png"
plt.savefig(save_file, dpi=600, bbox_inches="tight")
plt.show()

print(f"✅ Figure saved to: {save_file}")
```

```
Mounted at /content/drive
```





✅ Figure saved to: /content/drive/MyDrive/ColabOutputs/citations_per_

```
# Mount your Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# List of journal names
journals = [
    "Am J Obstet Gynecol (IF:8.7)", "BJOG (IF:4.8)", "Gynecol Oncol (I
    "Int J Gynecol Cancer (IF:4.5)"
    "J Minim Invasive Gynecol (IF:3.5)", "J Minim Invasive Gynecol (IF
    "J Minim Invasive Gynecol (IF:3.5)", "J Robot Surg (IF:2.2)",
    "Oncology (IF:2.5)", "Surg Innov (IF:1.2)", "Updates Surg (IF:2.5

# Create a DataFrame from the journal list
journal_df = pd.DataFrame(journals, columns=["Journal"])

# Count the frequency of each journal
journal_counts = journal_df["Journal"].value_counts().reset_index()
journal_counts.columns = ["Journal", "Frequency"]
```

```
# Display the table
print(journal_counts)

# Plot the frequency as a bar chart
plt.figure(figsize=(10, 6))
plt.barh(journal_counts["Journal"], journal_counts["Frequency"])
plt.xlabel("Frequency", fontsize=12)
plt.title("Frequency of Journal Names")
plt.gca().invert_yaxis()  # Invert y-axis to show the highest frequen


# Save figure to Google Drive at 600 dpi
save_path = '/content/drive/MyDrive/journal_frequency_600dpi_v2.png'
plt.savefig(save_path, dpi=600, bbox_inches='tight', transparent=True

print(f"Figure successfully saved to: {save_path}")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount
                                              Journal  Frequency
0                    J Minim Invasive Gynecol (IF:3.5)          2
1                         Am J Obstet Gynecol (IF:8.7)          1
2                                      BJOG (IF:4.8)          1
3                                Gynecol Oncol (IF:4.5)          1
4  Int J Gynecol Cancer (IF:4.5)J Minim Invasive ...          1
5                                J Robot Surg (IF:2.2)          1
6                                    Oncology (IF:2.5)          1
7                                  Surg Innov (IF:1.2)          1
8                                Updates Surg (IF:2.5)          1
Figure successfully saved to: /content/drive/MyDrive/journal_frequency_
```