KYPHOSIS DISEASE ANALYSIS

Exploratory Data Analysis (EDA) Project

Duygu Jones | Data Scientist | May 2024
Follow me: duygujones.com | Linkedin | GitHub | Kaggle | Medium | Tableau

Table of Contents

Intoduction

What is the purpose of this exploratory data analysis (EDA) on Kyphosis dataset?

- In this project, we will perform basic Exploratory Data Analysis (EDA) on the Kyphosis disease dataset.
- The analysis will involve data cleaning, processing, visualization, and interpretation of key findings to derive actionable insights.

Objectives:

- To understand the distribution of kyphosis conditions across different vertebra groups.
- To visualize the range of kyphosis and recovery rates in the dataset.
- To identify potential patterns or insights that can inform further research or medical practices.

Data Overview

About the Kyphosis Dataset

- Kyphosis is an abnormally excessive convex curvature of the spine.
- This dataset provides information about children who have undergone corrective spinal surgery.
- Dataset contains 81 rows and 4 columns representing data on children who have had corrective spinal surgery.

Structural Analysis

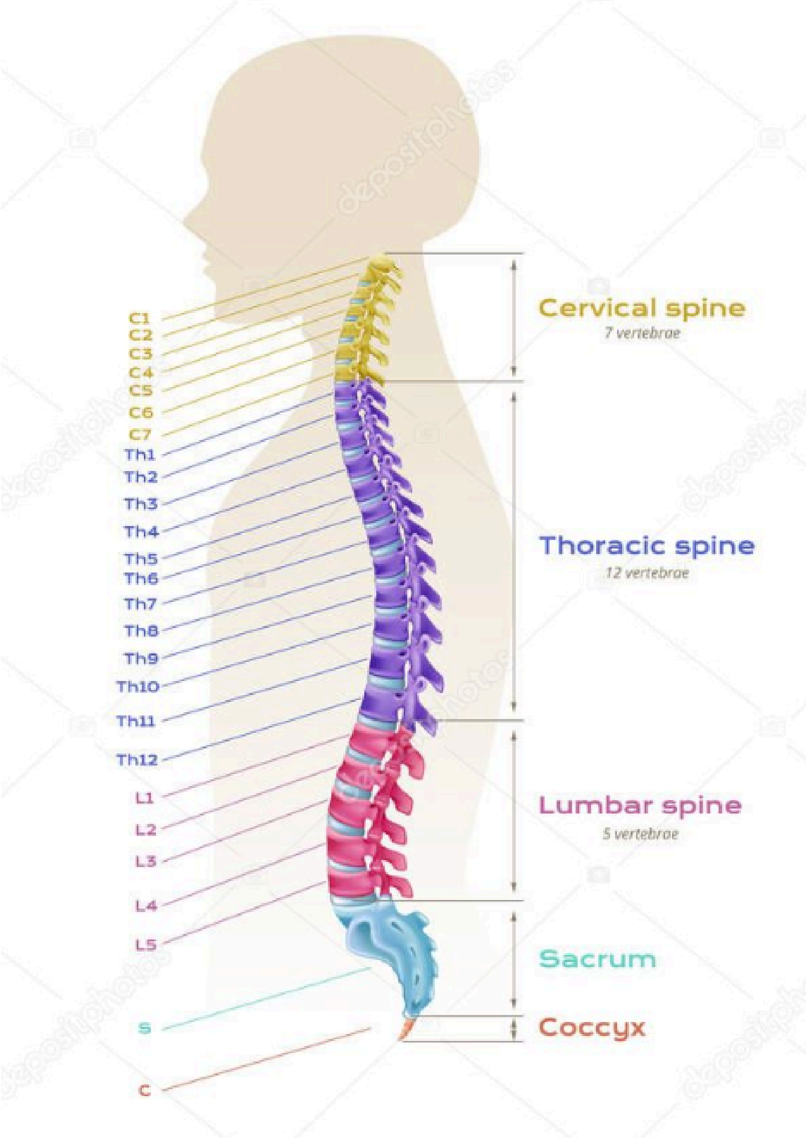
- **Dataset**: kyphosis.csv
- **Number of Rows**: 81
- **Number of Columns**: 4

- INPUTS: 1. **Age**: The age of the patient in months (children), 2. **Number**: The number of vertebrae involved in the surgery, 3. **Start**: The number of the first (top-most) vertebra operated on. 4. **Kyphosis**: Indicates whether kyphosis was **present** or **absent** after the operation.

- OUTPUTS: Represents a factor with variables **present** and **absent** indicating if **kyphosis** (a type of deformation) was present or not after the corrective spinal surgery.

What is Kyphosis?

Understanding the Anatomy of Vertebrae



Groups of Anatomic Vertebrea

1. **Cervical Spine (C1-C7)** (Neck)
   - Number of Vertebrae: 7
2. **Thoracic Spine (T1-T12)** (Back)
   - Number of Vertebrae: 12
3. **Lumbar Spine (L1-L5)** (Lower back)
   - Number of Vertebrea: 5
4. **Sacrum (S1-S5, fused)**
   - Number of Vertebrae: 5
5. **Coccyx (Co1-Co4, fused)**
   - Number of Vertebrae: 4

These groups represent the different sections of the vertebral column, with each section containing a specific number of vertebrae.

Classifying the 'Start' column in the Dataset to Anatomic Vertebrea Groups

In this Dataset; the **Start** column is the **first** (top-most) vertebra number operated on and containes the numbers between (1-18).

- We categorized the vertebra numbers in the Start column into Anatomical groups, and

- A new column 'Vertebrae Group' is created to classify the values in the Start column.

  Vertebrea Groups for the 'Start' column

  - 1-7: Cervical (C1-C7) (Neck)
  - 8-19: Thoracic (T1-T12) (Back)
  - 20-24: Lumbar (L1-L5) (Lower Back)
  - 25-29: Sacrum (S1-S5) (Sacral)

- We labelled each group of vertebras with a number and created a new column called 'Vertebrae Group Number'.

  Labels of the Vertebrea Groups

  - (1) Cervical (C1-C7)
  - (2) Thoracic (T1-T12)
  - (3) Lumbar (L1-L5)
  - (4) Sacrum (S1-S5)
  - (5) Unknown

Exploratory Data Analysis (EDA)

>

Import The Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")
warnings.warn("this will not show")

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

Read the Dataset

In [2]:
```
#Import the “kyphosis.csv” file using Pandas
df0 = pd.read_csv('/kaggle/input/kyphosis-dataset/kyphosis.csv')
df = df0.copy()
```

Overview of the Data

In [3]:
```
df.shape
```

Out[3]:
```
(81, 4)
```

In [4]:
```
df.dtypes
```

Out[4]:
```
Kyphosis    object
Age          int64
Number       int64
Start        int64
dtype: object
```

In [5]:
```
#Show the first couple of rows using .head()
df.head()
```

Out[5]:

	Kyphosis	Age	Number	Start
0	absent	71	3	5
1	absent	158	3	14
2	present	128	4	5
3	absent	2	5	1
4	absent	1	4	15

In [6]:
```
#Show the last couple of rows using .tail()
df.tail()
```

Out[6]:

	Kyphosis	Age	Number	Start
76	present	157	3	13
77	absent	26	7	13
78	absent	120	2	13
79	present	42	7	6
80	absent	36	4	13

In [7]:
```
# Obtain a Statistical Summary about the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Kyphosis  81 non-null     object
 1   Age       81 non-null     int64 
 2   Number    81 non-null     int64 
 3   Start     81 non-null     int64 
dtypes: int64(3), object(1)
memory usage: 2.7+ KB
```

Calculate the basic statistics

In [8]:
```
df.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
Age	81.0	83.654321	58.104251	1.0	26.0	87.0	130.0	206.0
Number	81.0	4.049383	1.619423	2.0	3.0	4.0	5.0	10.0
Start	81.0	11.493827	4.883962	1.0	9.0	13.0	16.0	18.0

In [9]:
```
df.describe(include="object").T
```

Out[9]:

	count	unique	top	freq
Kyphosis	81	2	absent	64

Check out the duplicated values

In [10]:
```
#Check the duplicated data if exists
df.duplicated().sum()
```

Out[10]: 0

Check out the missing values

In [11]:
```
#Check the duplicated data if exists
df.isnull().sum().sum()
```

Out[11]: 0

Feature Engineering

Classify the `Start` column into Anatomic `Vertebrea Groups`

In [12]:
```
# Classify vertebraes into Anatomic groups

def classify_vertebrae(start):
    if 1 <= start <= 7:
        return 'Cervical(Neck)'
    elif 8 <= start <= 19:
        return 'Thoracic(Back)'
    elif 20 <= start <= 24:
        return 'Lumbar(Lower Back)'
    elif 25 <= start <= 29:
        return 'Sacral'
    else:
        return 'Unknown'

# Apply the function to create a new column
df['vertebrae_group'] = df['Start'].apply(classify_vertebrae)

# Display the unique vertebra groups and their counts
vertebrae_groups = df['vertebrae_group'].value_counts()

print(vertebrae_groups)
```

```
vertebrae_group
Thoracic(Back)    64
Cervical(Neck)    17
Name: count, dtype: int64
```

Label the values of new column 'Vertebrea Groups'

In [13]:
```
# Function to Label the 'VertebreaGroups' values

def classify_vertebrae(start):
    if 1 <= start <= 7:
        return 1  # Cervical (Neck)
    elif 8 <= start <= 19:
        return 2  # Thoracic (Back)
    elif 20 <= start <= 24:
```

```
        return 3  # Lumbar (Lower Back)
    elif 25 <= start <= 29:
        return 4  # Sacral
    else:
        return 5  # Unknown

# Create a new 'Vertebrea Group Number' column and apply the classification function
df['vertebrae_group_number'] = df['Start'].apply(classify_vertebrae)

Num_of_Vertebrae_Group = df['vertebrae_group_number'].value_counts()

print(Num_of_Vertebrae_Group)
```

```
vertebrae_group_number
2    64
1    17
Name: count, dtype: int64
```

Convert the “Age” in months into “Age in Years”

In [14]:

```
# Function to convert months to years
def convert_months_to_years(months):
    return round(months / 12, 1)

# Apply the function to the 'Age' column
df['Age'] = df['Age'].apply(convert_months_to_years)

df.head()
```

Out[14]:

	Kyphosis	Age	Number	Start	vertebrae_group	vertebrae_group_number
0	absent	5.9	3	5	Cervical(Neck)	1
1	absent	13.2	3	14	Thoracic(Back)	2
2	present	10.7	4	5	Cervical(Neck)	1
3	absent	0.2	5	1	Cervical(Neck)	1
4	absent	0.1	4	15	Thoracic(Back)	2

Convert 'Kyphosis' column values to binary values

In [15]:

```
# Convert 'Kyphosis Status' to binary values
df['Kyphosis'] = df['Kyphosis'].map({'present': 1, 'absent': 0})
```

Rename the Columns

In [16]:

```
#features in dataset
df.columns
```

Out[16]:

```
Index(['Kyphosis', 'Age', 'Number', 'Start', 'vertebrae_group',
       'vertebrae_group_number'],
      dtype='object')
```

In [17]:

```
# Rename columns
df.rename(columns={
    'Kyphosis': 'kyphosis_status',
    'Number': 'number_of_vertebreae',
    'Start': 'start_vertebrea'
}, inplace=True)
```

In [18]:

```
df.head()
```

Out[18]:

	kyphosis_status	Age	number_of_vertebreae	start_vertebrea	vertebrae_group	vertebrae_group_numbe
0	0	5.9	3	5	Cervical(Neck)	
1	0	13.2	3	14	Thoracic(Back)	
2	1	10.7	4	5	Cervical(Neck)	
3	0	0.2	5	1	Cervical(Neck)	
4	0	0.1	4	15	Thoracic(Back)	

Calculate the average, minimum, and maximum ages to Analyze the ages of the participants.

In [19]:

```
age_statistics = df['Age'].agg(['mean', 'min', 'max'])

age_statistics
```

Out[19]:

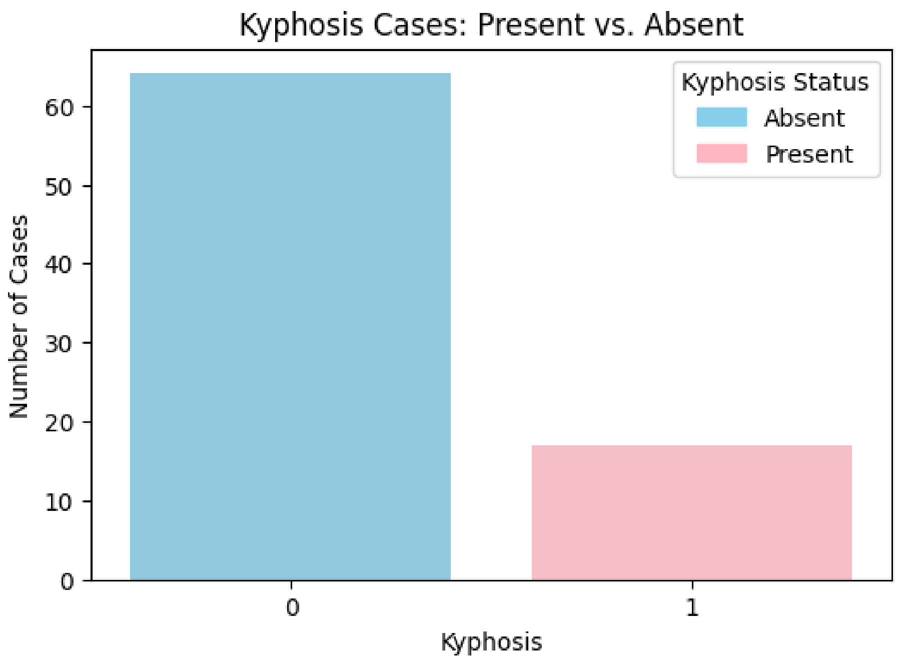
```
mean     6.97284
min      0.10000
max     17.20000
Name: Age, dtype: float64
```

Distributions

How many cases of kyphosis are present vs absent after spinal surgery?

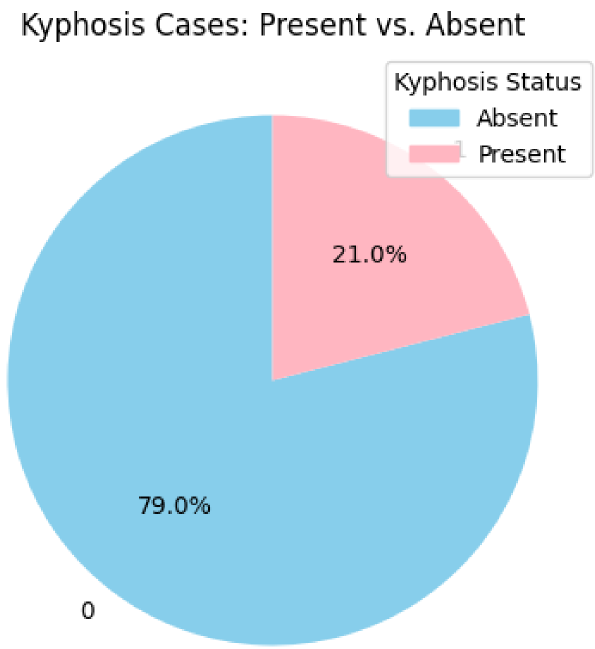
In [20]:

```
# Plot the counts using seaborn's countplot
import matplotlib.patches as mpatches

#plt.style.use('dark_background')

plt.figure(figsize=(6,4))
sns.countplot(data=df, x='kyphosis_status', palette=['skyblue', 'lightpink'])
plt.title('Kyphosis Cases: Present vs. Absent')
plt.xlabel('Kyphosis')
plt.ylabel('Number of Cases')
# Create custom legend
absent_patch = mpatches.Patch(color='skyblue', label='Absent')
present_patch = mpatches.Patch(color='lightpink', label='Present')
plt.legend(handles=[absent_patch, present_patch], title='Kyphosis Status', loc='upper right')


# Display the plot
plt.show()
```



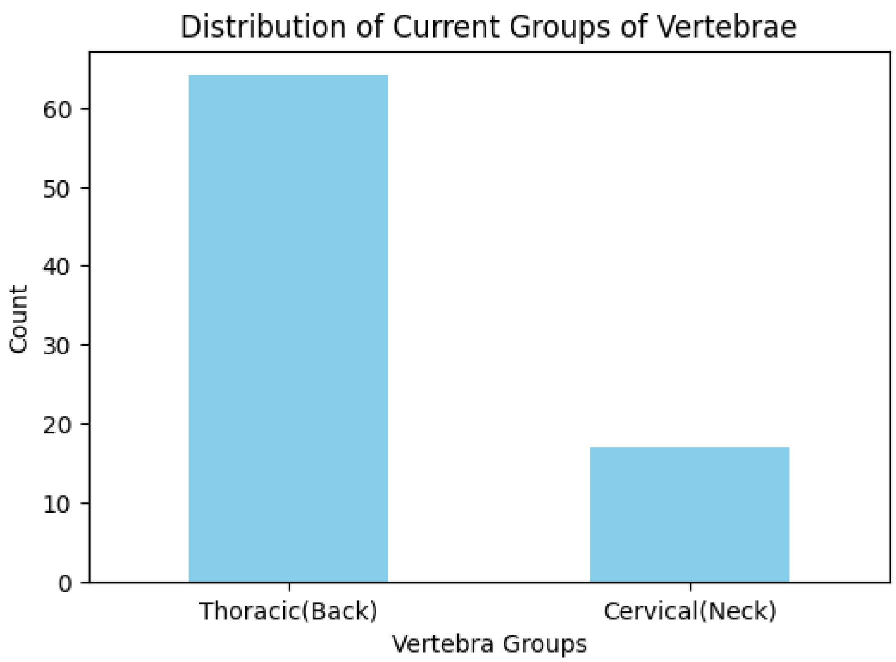
What is the percentage distribution of kyphosis cases?

In [21]:
```
# Plot the pie chart with percentages for Kyphosis cases

# Count the number of "Present" and "Absent" cases
kyphosis_counts = df['kyphosis_status'].value_counts()

# Plot the pie chart
plt.figure(figsize=(5,5))

plt.pie(kyphosis_counts, labels=kyphosis_counts.index, autopct='%1.1f%%', colors=['skyblue', 'li
plt.title('Kyphosis Cases: Present vs. Absent')

# Create custom legend
absent_patch = mpatches.Patch(color='skyblue', label='Absent')
present_patch = mpatches.Patch(color='lightpink', label='Present')
plt.legend(handles=[absent_patch, present_patch], title='Kyphosis Status', loc='upper right')

# Display the plot
plt.show()
```



How is the Distribution of Groups of Vertebrae?

In [22]:
```
plt.figure(figsize=(6,4))
vertebrae_groups.plot(kind='bar', color='skyblue')
plt.title('Distribution of Current Groups of Vertebrae')
plt.xlabel('Vertebra Groups')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```



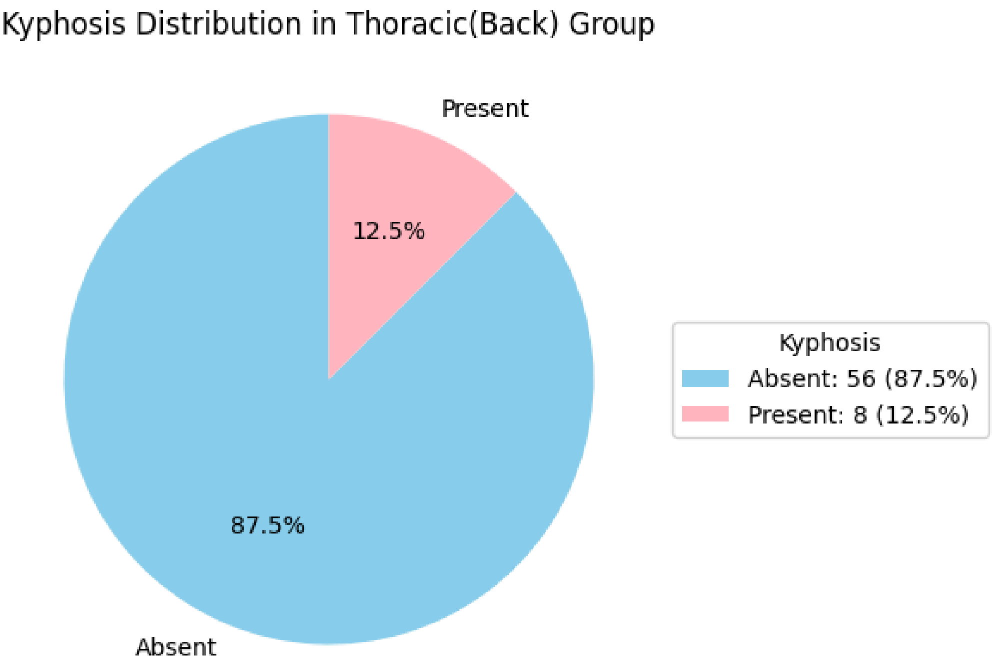
💡 Output: Distribution of Current Groups of Vertebrae

- The bar chart shows the distribution of vertebra groups among participants.
- It indicates that the majority of the cases (approximately 64) are related to the Thoracic (Back) vertebrae, while a smaller number (approximately 23) are related to the Cervical (Neck) vertebrae.

- This suggests that kyphosis or related spinal issues are more commonly associated with the thoracic region of the spine compared to the cervical region among the participants in this dataset.

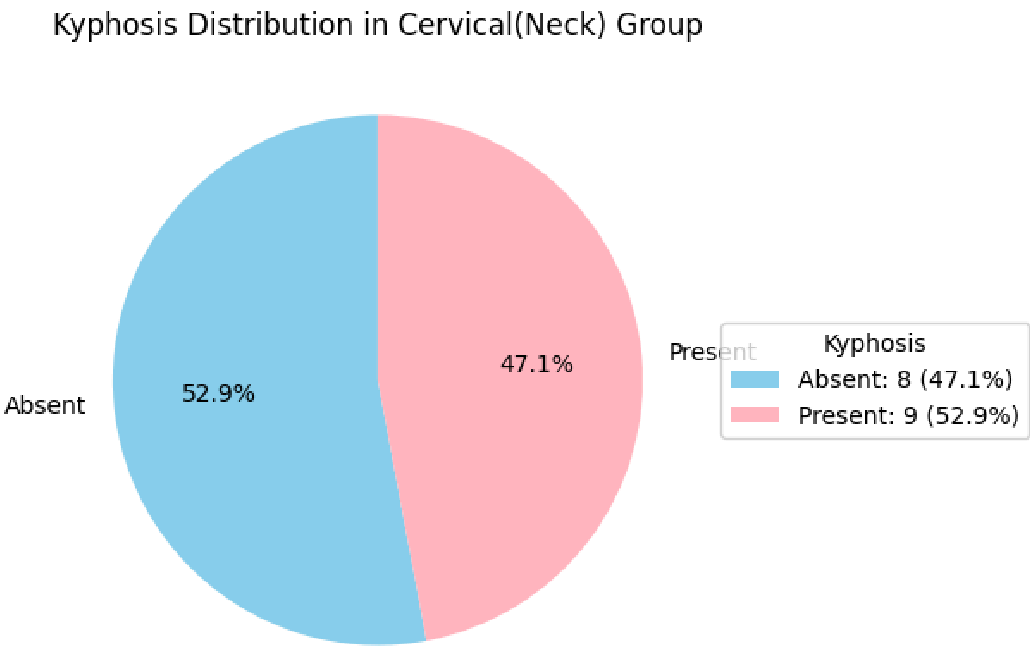
How is the Distribution of Kyphosis in Thoracic(Back) Group?

In [24]:
```
# Plot the pie chart with both numbers and percentages for the Thoracic group with custom legend
# Recreate the thoracic_data
thoracic_data = df[df['vertebrae_group'] == 'Thoracic(Back)']['kyphosis_status'].value_counts()

plt.figure(figsize=(5,5))
plt.pie(thoracic_data, labels=['Absent', 'Present'], autopct='%1.1f%%', colors=['skyblue', 'ligh
plt.title('Kyphosis Distribution in Thoracic(Back) Group')

# Add custom legend outside the pie chart
plt.legend([f'Absent: {thoracic_data[0]} ({(thoracic_data[0]/thoracic_data.sum()*100):.1f}%)',
            f'Present: {thoracic_data[1]} ({(thoracic_data[1]/thoracic_data.sum()*100):.1f}%)'],
           title='Kyphosis', loc='center left', bbox_to_anchor=(1, 0.5))

plt.show()
```



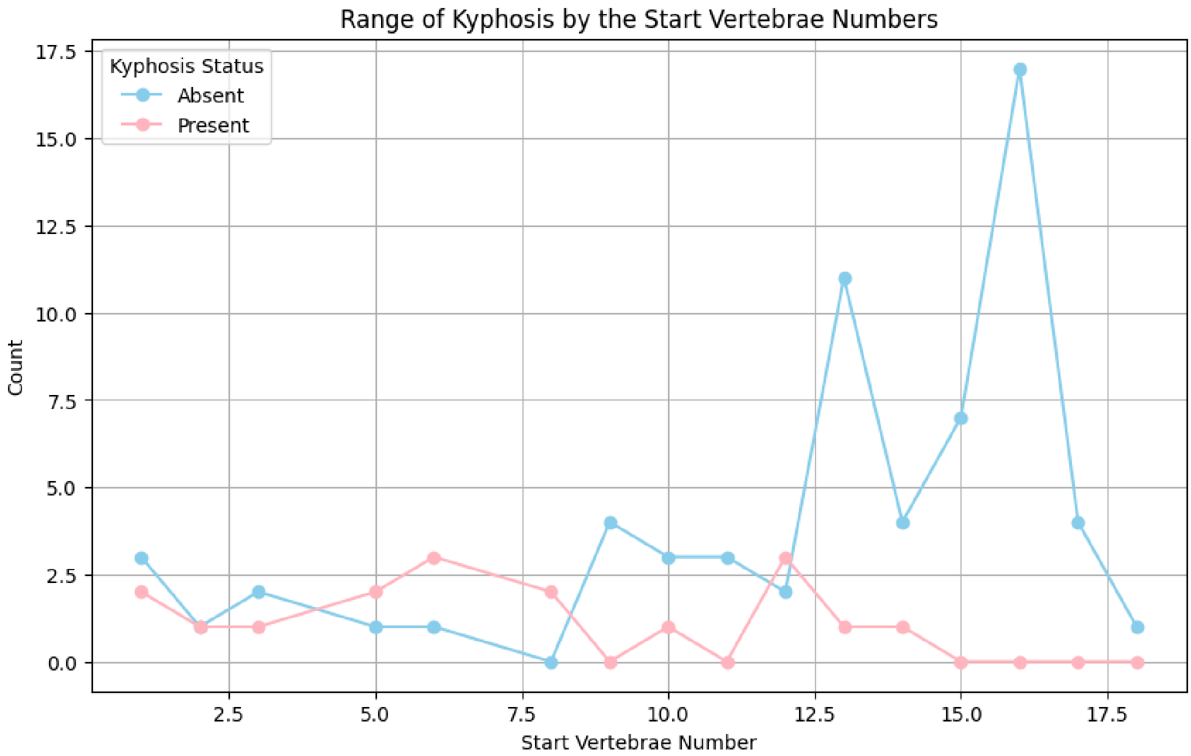
How is the Distribution of Kyphosis in Cervical(Neck) Group?

In [25]:
```
# Recreate the cervical_data
cervical_data = df[df['vertebrae_group'] == 'Cervical(Neck)']['kyphosis_status'].value_counts()

# Plot the pie chart with both numbers and percentages for the Cervical group with custom legend
plt.figure(figsize=(5,5))
plt.pie(cervical_data, labels=['Absent', 'Present'], autopct='%1.1f%%', colors=['skyblue', 'ligh
plt.title('Kyphosis Distribution in Cervical(Neck) Group')

# Add custom legend outside the pie chart
plt.legend([f'Absent: {cervical_data[0]} ({(cervical_data[0]/cervical_data.sum()*100):.1f}%)',
            f'Present: {cervical_data[1]} ({(cervical_data[1]/cervical_data.sum()*100):.1f}%)'],
           title='Kyphosis', loc='center left', bbox_to_anchor=(1, 0.5))

plt.show()
```



How is the Range of Kyphosis by the Start Vertebrae Numbers?

In [26]:
```
# Mevcut sütun isimlerini kullanarak kyphosis_by_start_vertebra veri çerçevesini yeniden oluştur
kyphosis_by_start_vertebra = df.pivot_table(
    index='start_vertebrea',
    columns='kyphosis_status',
    aggfunc='size',
    fill_value=0
).reset_index()

# Çizgi grafiğini oluşturma
plt.figure(figsize=(10,6))
plt.plot(kyphosis_by_start_vertebra['start_vertebrea'], kyphosis_by_start_vertebra[0], marker='c
plt.plot(kyphosis_by_start_vertebra['start_vertebrea'], kyphosis_by_start_vertebra[1], marker='c
plt.title('Range of Kyphosis by the Start Vertebrae Numbers')
plt.xlabel('Start Vertebrae Number')
plt.ylabel('Count')
plt.legend(title='Kyphosis Status')
plt.grid(True)
plt.show()
```



💡 Output: The distribution of kyphosis (present and absent) across vertebra numbers from 1 to 18.

- Each bar represents the proportion of kyphosis presence (orange) and absence (blue) for the corresponding vertebra number.
- Overall:
  - Kyphosis is more commonly present in vertebrae numbers 1, 2, 3, 5, 6, and 8,
  - while it is largely absent in vertebrae numbers 4, 7, 9-14, and 16-18.

Scale the raw Age column (in months) using both standardization and Normalization.

- Perform a sanity check.# Normalization is conducted on the 'Age' column to make feature values range from 0 to 1.

In [27]:
```
# Scale the raw Age column (in months) using both standardization and Normalization. Perform a s
# Normalization is conducted on the 'Age' column to make feature values range from 0 to 1.

from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Select the age column
age_data = df[['Age']]

# Perform standardization
scaler_standard = StandardScaler()
age_standardized = scaler_standard.fit_transform(age_data)

# Perform normalization
scaler_minmax = MinMaxScaler()
age_normalized = scaler_minmax.fit_transform(age_data)

# Create a new DataFrame to show the results
result_df = df.copy()
result_df['Age_standardized'] = age_standardized
result_df['Age_normalized'] = age_normalized

# Display the first few rows to check the results
print(result_df[['Age', 'Age_standardized', 'Age_normalized']].head())
```

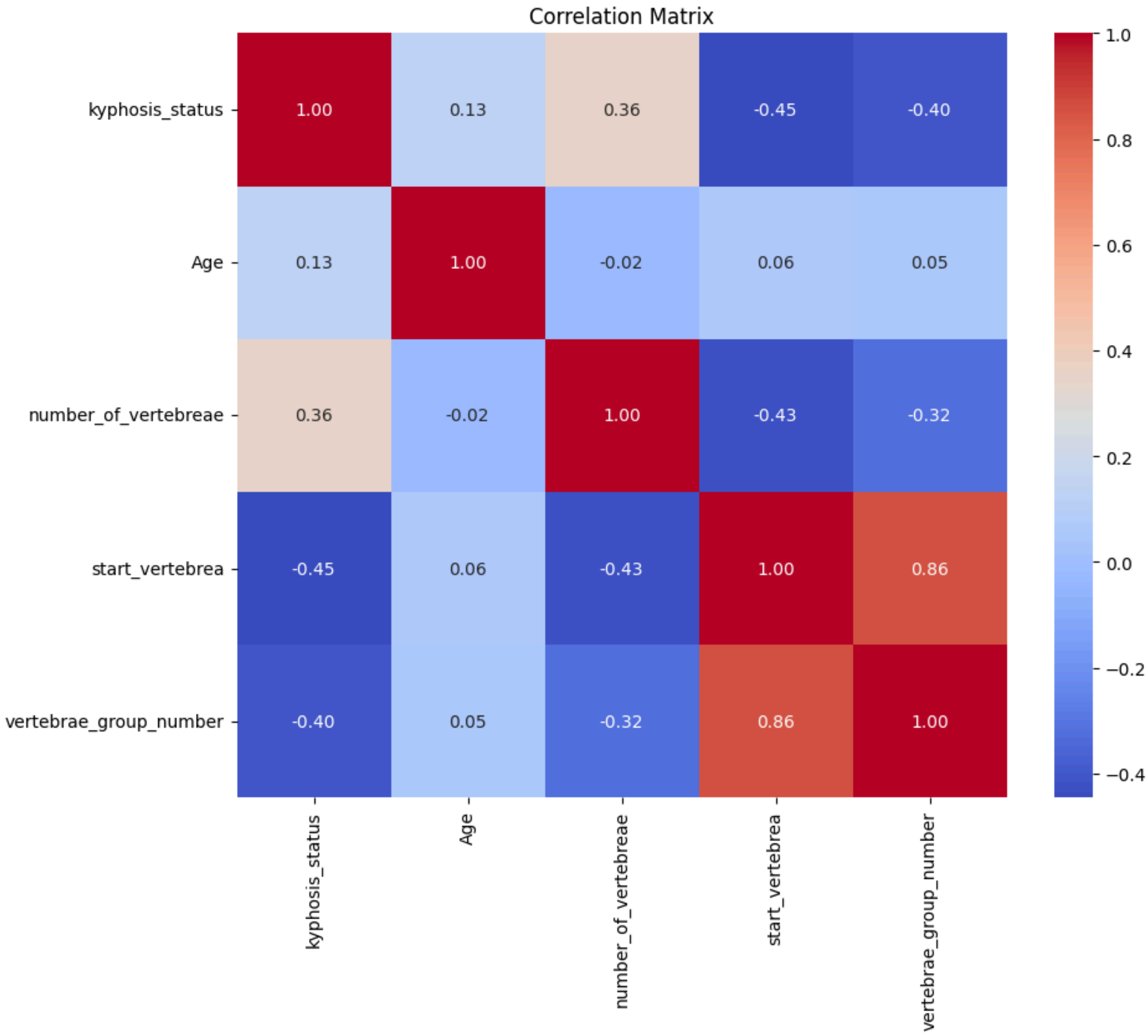
```
# Sanity check - The mean of the standardized age should be 0 and the standard deviation should
mean_standardized = age_standardized.mean()
std_standardized = age_standardized.std()

# Sanity check - The min of the normalized age should be 0 and the max should be 1
min_normalized = age_normalized.min()
max_normalized = age_normalized.max()

print(f"\nStandardized Age - Mean: {mean_standardized:.2f}, Std Dev: {std_standardized:.2f}")
print(f"Normalized Age - Min: {min_normalized:.2f}, Max: {max_normalized:.2f}")
```

```
    Age  Age_standardized  Age_normalized
0   5.9         -0.223052        0.339181
1  13.2          1.294678        0.766082
2  10.7          0.774907        0.619883
3   0.2         -1.408129        0.005848
4   0.1         -1.428920        0.000000

Standardized Age - Mean: 0.00, Std Dev: 1.00
Normalized Age - Min: 0.00, Max: 1.00
```

Correlations

How do the features correlate with each other?

In [30]:
```
# Sadece sayısal sütunları seçme
numeric_df = df.select_dtypes(include=[float, int])

# Korelasyon matrisini hesaplama
correlation_matrix = numeric_df.corr()

# Heatmap'i çizme
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



💡 Output:Correlation Matrix

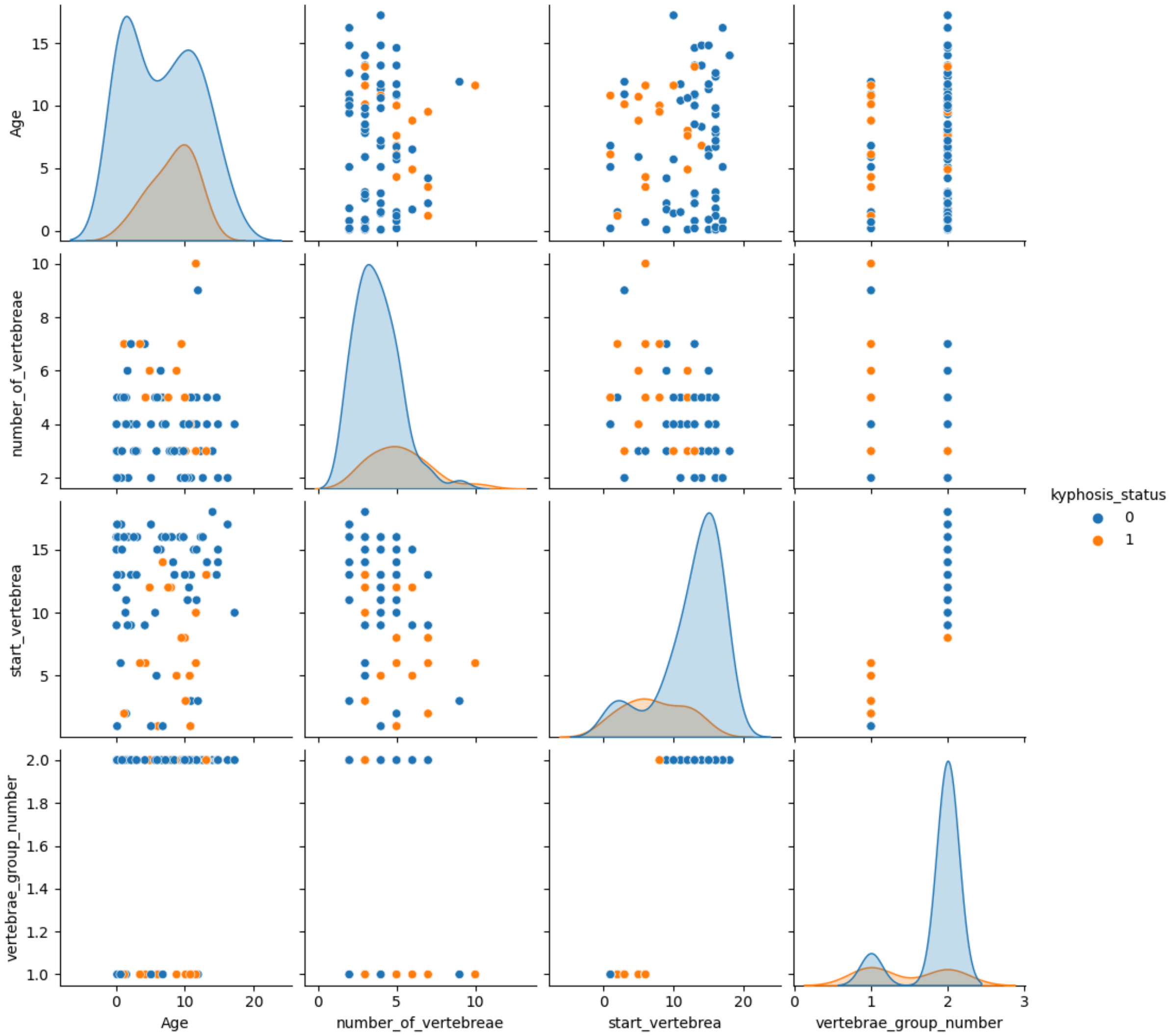
1. **Kyphosis Status:**
   - Negatively correlated with `Start Vertebra` (-0.45) and `Vertebra Group Number` (-0.40).
   - Positively correlated with `Number of Vertebrae` (0.36).
2. **Age:**
   - Weak positive correlation with `Kyphosis Status` (0.13) and weak negative correlations with other variables.
3. **Vertebrae Relationships:**
   - Strong positive correlation between `Start Vertebra` and `Vertebra Group Number` (0.86).
   - Moderate negative correlation between `Start Vertebra` and `Number of Vertebrae` (-0.43).

This suggests that kyphosis is more likely to be present with a higher number of vertebrae involved and less likely as the starting vertebra number increases.

What are the correlations between the features with Kyphosis present or absent ?

In [29]:
```
sns.pairplot(df,hue='kyphosis_status')
```

Out[29]: `<seaborn.axisgrid.PairGrid at 0x7984a308d4e0>`



💡 Output:"Paitplot for all features"

1. **Age Distribution:**
   - Participants without kyphosis (blue) are predominantly younger, especially around 0-2 years.
   - Participants with kyphosis (orange) are more spread out in age but still have a significant presence in the younger age range.
2. **Number of Vertebrae:**
   - Both groups have a similar distribution in the number of vertebrae affected, with no clear distinction between kyphosis presence or absence.
3. **Start Vertebra:**
   - The starting vertebra values are widely spread across both groups, showing no clear pattern distinguishing participants with and without kyphosis.
4. **Vertebrae Group Number:**
   - Most participants fall into the same vertebra group numbers regardless of kyphosis status, indicating no significant difference.

Overall, there are no strong patterns differentiating participants with and without kyphosis based on age, number of vertebrae, start vertebra, or vertebrae group number.

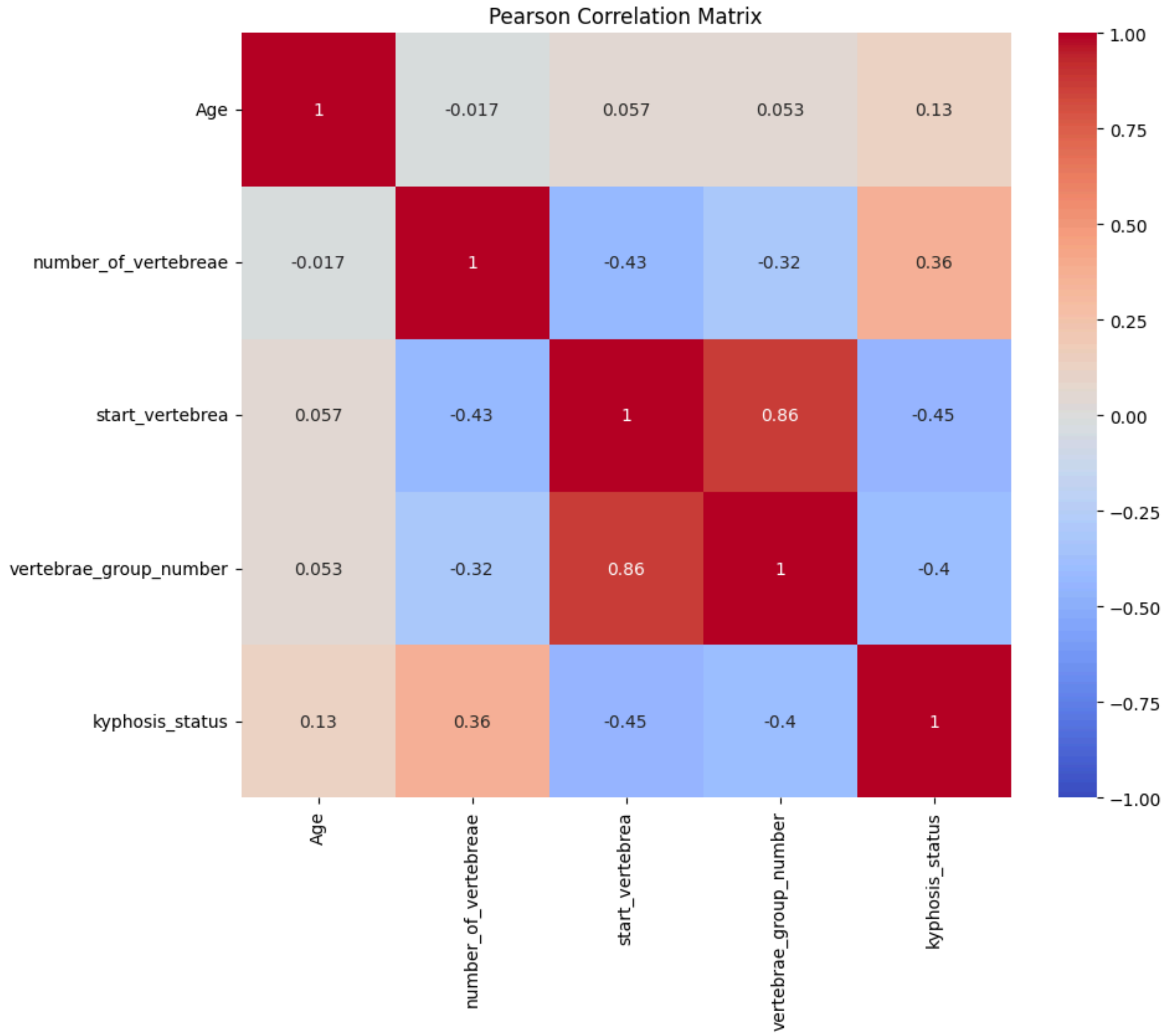
Pearson/Spearman Correlations

In [31]:
```
# Select relevant numerical variables for correlation analysis
numerical_vars = ['Age', 'number_of_vertebreae', 'start_vertebrea', 'vertebrae_group_number', 'k


# Pearson Correlation
pearson_corr = df[numerical_vars].corr(method='pearson')
print("Pearson Correlation Matrix:")
print(pearson_corr)

# Spearman Correlation
spearman_corr = df[numerical_vars].corr(method='spearman')
print("\nSpearman Correlation Matrix:")
print(spearman_corr)
```

```
Pearson Correlation Matrix:
                           Age  number_of_vertebreae  start_vertebrea  \
Age                   1.000000             -0.016892         0.057317
number_of_vertebreae -0.016892              1.000000        -0.425099
start_vertebrea       0.057317             -0.425099         1.000000
vertebrae_group_number 0.053188            -0.323260         0.864433
kyphosis_status       0.126452              0.360935        -0.445943

                      vertebrae_group_number  kyphosis_status
Age                                 0.053188         0.126452
number_of_vertebreae               -0.323260         0.360935
start_vertebrea                     0.864433        -0.445943
vertebrae_group_number              1.000000        -0.404412
kyphosis_status                    -0.404412         1.000000

Spearman Correlation Matrix:
                           Age  number_of_vertebreae  start_vertebrea  \
Age                   1.000000             -0.028449         0.019323
number_of_vertebreae -0.028449              1.000000        -0.482757
start_vertebrea       0.019323             -0.482757         1.000000
vertebrae_group_number 0.039557            -0.258950         0.710501
kyphosis_status       0.125805              0.338627        -0.459736

                      vertebrae_group_number  kyphosis_status
Age                                 0.039557         0.125805
number_of_vertebreae               -0.258950         0.338627
start_vertebrea                     0.710501        -0.459736
vertebrae_group_number              1.000000        -0.404412
kyphosis_status                    -0.404412         1.000000
```

Pearson Correlation Matrix

In [32]:
```
# Plot the Pearson Correlation Matrix
plt.figure(figsize=(10, 8))
sns.heatmap(pearson_corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Pearson Correlation Matrix')
plt.show()
```



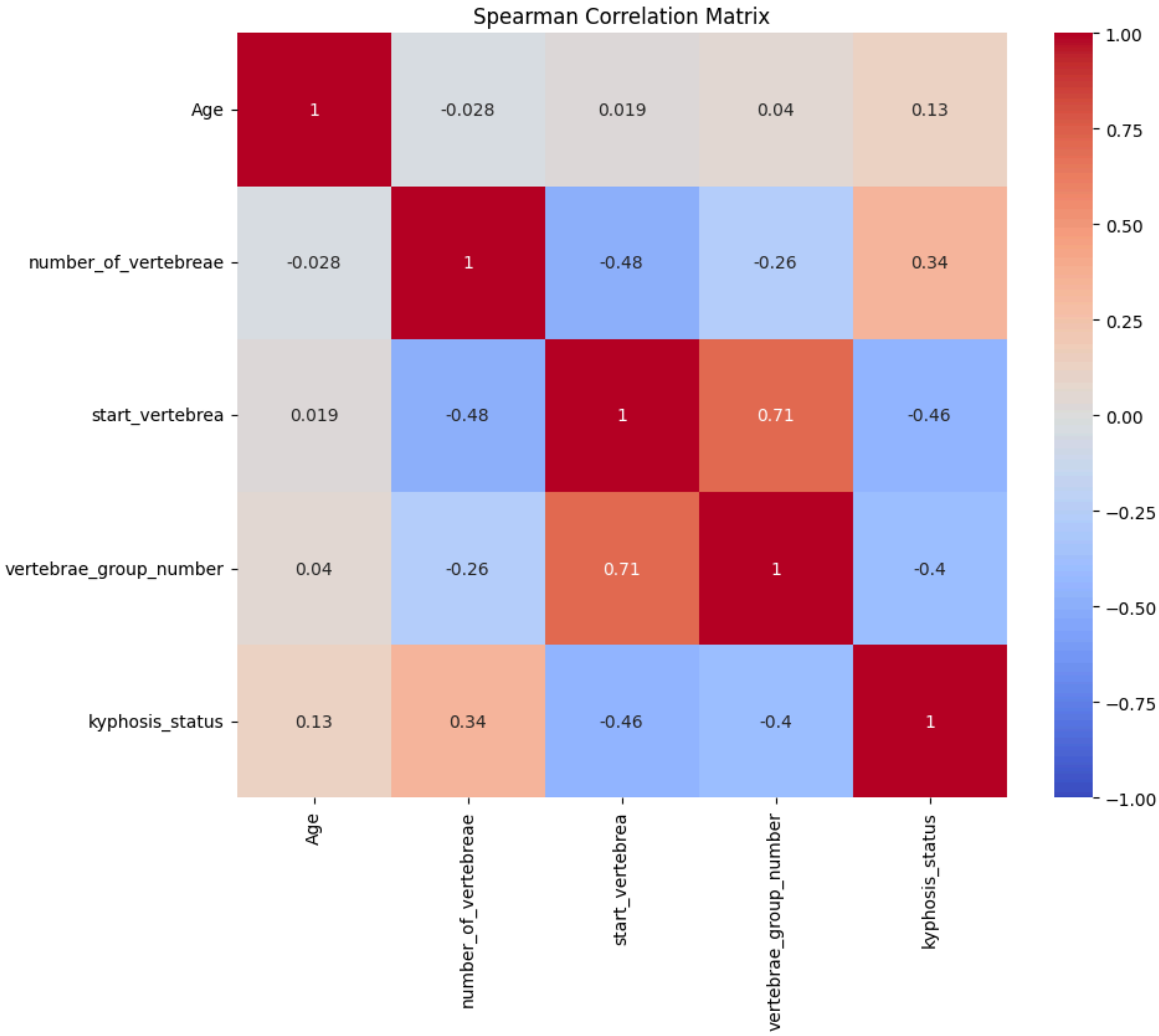
💡 Output:The Pearson correlation matrix reveals the following key points:

1. **Kyphosis Status:**
   - Positively correlated with `Number of Vertebrae` (0.36), indicating that a higher number of affected vertebrae is associated with kyphosis.
   - Negatively correlated with `Start Vertebra` (-0.45) and `Vertebra Group Number` (-0.40), suggesting that kyphosis is more likely to occur when the affected vertebrae are lower in the spine.
2. **Start Vertebra:**
   - Strong positive correlation with `Vertebra Group Number` (0.86), indicating that higher starting vertebra numbers are associated with higher vertebra group numbers.

Overall, the presence of kyphosis is associated with a higher number of affected vertebrae and tends to occur in lower vertebral positions.

Spearman Correlation Matrix

In [33]:
```
# Plot the Spearman Correlation Matrix
plt.figure(figsize=(10, 8))
sns.heatmap(spearman_corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Spearman Correlation Matrix')
plt.show()
```



💡 Output:Spearman correlation matrix:

1. **Kyphosis Status:**
   - Positively correlated with `Number of Vertebrae` (0.40).
   - Negatively correlated with `Start Vertebra` (-0.42) and `Vertebra Group Number` (-0.39).
2. **Start Vertebra:**
   - Strong positive correlation with `Vertebra Group Number` (0.86).

Kyphosis is associated with a higher number of affected vertebrae and tends to occur in lower vertebral positions.

💡 Overall Conclusion:

Overall Conclusion and Insights

Based on the EDA performed on the kyphosis dataset, here are the key findings:

1. **Dataset Overview:**
   - The dataset includes patient data related to the presence or absence of kyphosis, patient age, number of vertebrae operated on, the start vertebra number, and vertebrae group classifications (cervical, thoracic, etc.).
2. **Distribution of Kyphosis by Vertebra Groups:**

- **Cervical (Neck) Group:** Kyphosis presence and absence are balanced, with approximately 53% having kyphosis and 47% not having it.
- **Thoracic (Back) Group:** Kyphosis absence is predominant, with 87.5% not having kyphosis and only 12.5% showing presence.

3. **Kyphosis by Start Vertebra Numbers:**
   - Kyphosis is more common in certain vertebra numbers. Notably, vertebrae C1, C6, T12, and T13 show higher kyphosis presence, while vertebra T16 shows a higher absence.
4. **Correlation Analysis:**
   - There is a significant correlation between the number of vertebrae operated on and the start vertebra number.
   - Age and kyphosis status have a low correlation.

Key Takeaways:

1. **Treatment and Prevention Strategies:**
   - The balanced distribution of kyphosis in the cervical region suggests a need for careful planning of treatment and prevention strategies in this area.
   - The rarity of kyphosis in the thoracic region indicates a different approach may be required for patients in this area.
2. **Attention to Start Vertebra Numbers:**
   - Higher kyphosis presence in specific vertebrae suggests developing special treatment protocols targeting these vertebrae.
3. **Correlation Findings:**
   - The correlation between the number of vertebrae operated on and the start vertebra number can inform surgical planning. Operations starting from certain vertebrae tend to involve more vertebrae.

Further analysis with advanced statistical tests and machine learning models is recommended to better understand and predict kyphosis.

Thank you...

Duygu Jones | Data Scientist | May 2024
Follow me: duygujones.com | Linkedin | GitHub | Kaggle | Medium | Tableau