**Sali Azou**

**S190523**

**Q**: How many unique seasons are there in the dataset? Use the nunique() function.

Vi sætter vores data i en liste ud fra år, og værdier:

data **=** [

(2016, 5369), (2014, 5362), (2015, 5354), (2013, 5320),

(2010, 5263), (2012, 5253), (2009, 5249), (2011, 5246),

(2008, 5163), (2007, 5043), (2006, 4757), (2005, 4675),

(2003, 4616), (2004, 4571), (2002, 4555), (2000, 4519),

(2001, 4467), (1999, 4222), (1998, 4167), (1997, 4155),

(1992, 4127), (1991, 4123), (1996, 4122), (1995, 4077),

(1994, 4060), *# ... and so on for the rest of your data*

(1988, 3955), (1987, 3915), (1986, 3783), (1985, 3737)

]

*# Creating a DataFrame*

df2 **=** pd**.**DataFrame(data, columns**=**['Season', 'Value'])

*# Using nunique() to count unique seasons*

unique\_seasons **=** df['Season']**.**nunique()

print(f"Number of unique seasons: {unique\_seasons}")

unique seasons er 29.

**Q**: Find the team with the most wins. Use the value\_counts() function on the Wteam column.

data **=** {

'Season': [2016, 2014, 2015, **...**], *# your season data*

'Wteam': ['Team A', 'Team B', 'Team A', **...**] *# hypothetical winning team data*

}

df **=** pd**.**DataFrame(data)

*# Using value\_counts() to count wins for each team*

win\_counts **=** df['Wteam']**.**value\_counts()

*# The team with the most wins*

most\_wins\_team **=** win\_counts**.**idxmax()

most\_wins\_count **=** win\_counts**.**max()

print(f"Team with the most wins: {most\_wins\_team} ({most\_wins\_count} wins)")

Team A(2 wins)

**Q**: Make three dataframes that are sorted by season, winning team, and winning score respectively. Then, Using iloc, select the rows from index 100 to 200 and the columns for season, winning team, and winning score, respectively.

data **=** {

'Season': [100027, 49310, 89021, 85042, 103660],

'Daynum': [2008, 1997, 2006, 2005, 2009],

'Wteam': [66, 66, 44, 66, 26],

'Wscore': [1203, 1157, 1284, 1131, 1326],

'Lteam': [49, 61, 41, 73, 59],

'Lscore': [1387, 1204, 1343, 1216, 1359],

'Wloc': ['H', 'H', 'A', 'H', 'H'],

'Numot': [0, 0, 0, 0, 0]

}

df **=** pd**.**DataFrame(data)

*# Sorting by 'Season', 'Wteam', and 'Wscore'*

df\_sorted\_season **=** df**.**sort\_values(by**=**'Season')

df\_sorted\_wteam **=** df**.**sort\_values(by**=**'Wteam')

df\_sorted\_wscore **=** df**.**sort\_values(by**=**'Wscore')

*# Selecting rows and columns with iloc*

*# Note: Adjust the row indices according to your full dataset*

selected\_season **=** df\_sorted\_season**.**iloc[100:200][['Season', 'Wteam', 'Wscore']]

selected\_wteam **=** df\_sorted\_wteam**.**iloc[100:200][['Season', 'Wteam', 'Wscore']]

selected\_wscore **=** df\_sorted\_wscore**.**iloc[100:200][['Season', 'Wteam', 'Wscore']]

**Q**: From these three subsets you obtained above, find the season and winning team for the game with the highest winning score.

*# Finding the game with the highest winning score*

highest\_score\_game **=** df\_sorted\_wscore**.**iloc[0]

*# Extracting the season and winning team for this game*

season\_highest\_score **=** highest\_score\_game['Season']

winning\_team\_highest\_score **=** highest\_score\_game['Wteam']

print(f"Season: {season\_highest\_score}, Winning Team: {winning\_team\_highest\_score}")

Således er season: 85042, og winning team er 66.

**Q**: Create a new column in the DataFrame called 'ScoreDifference' which is the absolute difference between the winning score and the losing score. Filter the DataFrame to only include games where the 'ScoreDifference' is greater than the average 'ScoreDifference' for all games.

data **=** {

'Season': [100027, 49310, 89021, 85042, 103660],

'Daynum': [2008, 1997, 2006, 2005, 2009],

'Wteam': [66, 66, 44, 66, 26],

'Wscore': [1203, 1157, 1284, 1131, 1326],

'Lteam': [49, 61, 41, 73, 59],

'Lscore': [1387, 1204, 1343, 1216, 1359],

'Wloc': ['H', 'H', 'A', 'H', 'H'],

'Numot': [0, 0, 0, 0, 0]

}

df **=** pd**.**DataFrame(data)

*# Creating 'ScoreDifference' column*

df['ScoreDifference'] **=** (df['Wscore'] **-** df['Lscore'])**.**abs()

*# Calculating the average 'ScoreDifference'*

average\_score\_difference **=** df['ScoreDifference']**.**mean()

*# Filtering the DataFrame*

filtered\_df **=** df[df['ScoreDifference'] **>** average\_score\_difference]

**Q**: From this filtered DataFrame, find the season and teams involved in the game with the highest 'ScoreDifference'.

*# Finding the game with the highest 'ScoreDifference'*

highest\_diff\_game **=** filtered\_df**.**loc[filtered\_df['ScoreDifference']**.**idxmax()]

*# Extracting the season, winning team, and losing team for this game*

season\_highest\_diff **=** highest\_diff\_game['Season']

winning\_team\_highest\_diff **=** highest\_diff\_game['Wteam']

losing\_team\_highest\_diff **=** highest\_diff\_game['Lteam']

print(f"Season: {season\_highest\_diff}, Winning Team: {winning\_team\_highest\_diff}, Losing Team: {losing\_team\_highest\_diff}")

Som derfor er Season: 100027

Winning team: 66

Losing team: 49

**Q**: Group the DataFrame by season and find the average winning score for each season.

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Season'] **=** df['Season']**.**astype(int)

df['Wscore'] **=** df['Wscore']**.**astype(int)

*# Grouping by 'Season' and calculating the average 'Wscore'*

average\_winning\_score\_by\_season **=** df**.**groupby('Season')['Wscore']**.**mean()

print(average\_winning\_score\_by\_season)

|  |  |
| --- | --- |
| Season |  |
| 49310 | 1157.0 |
| 85042 | 1131.0 |
| 89021 | 1284.0 |
| 100027 | 1203.0 |
| 103660 | 1326.0 |

**Q**: Group the DataFrame by winning team and find the maximum winning score for each team across all seasons.

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Wteam'] **=** df['Wteam']**.**astype(int)

df['Wscore'] **=** df['Wscore']**.**astype(int)

*# Grouping by 'Wteam' and calculating the maximum 'Wscore'*

max\_winning\_score\_by\_team **=** df**.**groupby('Wteam')['Wscore']**.**max()

print(max\_winning\_score\_by\_team)

|  |  |
| --- | --- |
| Wteam |  |
| 26 | 1326 |
| 44 | 1284 |
| 66 | 1203 |

**Q**: Group the DataFrame by both season and winning team. Find the team with the highest average winning score for each season.

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Season'] **=** df['Season']**.**astype(int)

df['Wteam'] **=** df['Wteam']**.**astype(int)

df['Wscore'] **=** df['Wscore']**.**astype(int)

*# Grouping by 'Season' and 'Wteam' and calculating the average 'Wscore'*

grouped\_df **=** df**.**groupby(['Season', 'Wteam'])['Wscore']**.**mean()

*# Finding the team with the highest average winning score for each season*

highest\_avg\_score\_by\_season **=** grouped\_df**.**reset\_index()**.**groupby('Season')**.**apply(**lambda** x: x**.**loc[x['Wscore']**.**idxmax()])

print(highest\_avg\_score\_by\_season)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Season | Wteam | Wscore |
| Season |  |  |  |
| 49310 | 49310.0 | 66.0 | 1157.0 |
| 85042 | 85042 | 66.0 | 1131.0 |
| 89021 | 89021.0 | 44.0 | 1284.0 |
| 100027 | 100027.0 | 66.0 | 1203.0 |
| 103660 | 103660.0 | 26.0 | 1326.0 |

**Q**: Create a new DataFrame that counts the number of wins for each team in each season. This will involve grouping by both season and winning team, and then using the count() function.

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Season'] **=** df['Season']**.**astype(int)

df['Wteam'] **=** df['Wteam']**.**astype(int)

*# Grouping by 'Season' and 'Wteam' and counting the number of wins*

wins\_count **=** df**.**groupby(['Season', 'Wteam'])**.**size()**.**reset\_index(name**=**'Wins')

print(wins\_count)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Season | Wteam | Wins |
| 0 | 49310 | 66 | 1 |
| 1 | 85042 | 66 | 1 |
| 2 | 89021 | 44 | 1 |
| 3 | 100027 | 66 | 1 |
| 4 | 103660 | 26 | 1 |

**Q**: For each season, find the team with the most wins. This will involve creating a DataFrame similar to the one in task 5, and then using the idxmax() function for each season.

*Provided array*

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Season'] **=** df['Season']**.**astype(int)

df['Wteam'] **=** df['Wteam']**.**astype(int)

*# Grouping by 'Season' and 'Wteam' and counting the number of wins*

wins\_count **=** df**.**groupby(['Season', 'Wteam'])**.**size()**.**reset\_index(name**=**'Wins')

*# Finding the team with the most wins for each season*

team\_with\_most\_wins **=** wins\_count**.**loc[wins\_count**.**groupby('Season')['Wins']**.**idxmax()]

print(team\_with\_most\_wins)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Season | Wteam | Wins |
| 0 | 49310 | 66 | 1 |
| 1 | 85042 | 66 | 1 |
| 2 | 89021 | 44 | 1 |
| 3 | 100027 | 66 | 1 |
| 4 | 103660 | 26 | 1 |

**Q**: Group the DataFrame by losing team and find the average losing score for each team across all seasons. Compare this with the average winning score for each team from task 3. Are there teams that have a higher average losing score than winning score?

data\_array **=** np**.**array([

[100027, 2008, 66, 1203, 49, 1387, 'H', 0, 184],

[49310, 1997, 66, 1157, 61, 1204, 'H', 0, 47],

[89021, 2006, 44, 1284, 41, 1343, 'A', 0, 59],

[85042, 2005, 66, 1131, 73, 1216, 'H', 0, 85],

[103660, 2009, 26, 1326, 59, 1359, 'H', 0, 33]

], dtype**=**object)

*# Creating DataFrame*

df **=** pd**.**DataFrame(data\_array, columns**=**['Season', 'Daynum', 'Wteam', 'Wscore', 'Lteam', 'Lscore', 'Wloc', 'Numot', 'ScoreDifference'])

*# Converting the relevant columns to the appropriate data types*

df['Lteam'] **=** df['Lteam']**.**astype(int)

df['Lscore'] **=** df['Lscore']**.**astype(int)

df['Wteam'] **=** df['Wteam']**.**astype(int)

df['Wscore'] **=** df['Wscore']**.**astype(int)

*# Grouping by 'Lteam' and calculating the average 'Lscore'*

average\_losing\_score\_by\_team **=** df**.**groupby('Lteam')['Lscore']**.**mean()

*# Grouping by 'Wteam' and calculating the average 'Wscore'*

average\_winning\_score\_by\_team **=** df**.**groupby('Wteam')['Wscore']**.**mean()

*# Comparing the average losing score with the average winning score*

comparison\_df **=** pd**.**DataFrame({

'Avg\_Losing\_Score': average\_losing\_score\_by\_team,

'Avg\_Winning\_Score': average\_winning\_score\_by\_team

})

print(comparison\_df)

*# 41, 49, 59, 61 and 73 have a higher average losing score than winning score*

|  |  |  |
| --- | --- | --- |
|  | Losing score | Winning score |
| 26 |  | 1326 |
| 41 | 1343 |  |
| 44 |  | 1284.000 |
| 49 | 1387 |  |
| 59 | 1359 |  |
| 61 | 1204.0 | NaN |
| 66 |  | 1163.66 |
| 73 | 1216,0 |  |

**Q**: Create a new column 'HighScoringGame' that is 'Yes' if the winning score is greater than 100 and 'No' otherwise. This will require iterating over the rows of the DataFrame and checking the value of the winning score for each row:

**import** pandas **as** pd

*# Assuming you have a DataFrame named 'df'*

*# If the DataFrame is not named, replace 'df' with the actual name of your DataFrame*

*# Create a new column 'HighScoringGame'*

df['HighScoringGame'] **=** 'No' *# Initialize with 'No' for all rows*

**for** index, row **in** df**.**iterrows():

**if** row['Wscore'] **>** 100:

df**.**at[index, 'HighScoringGame'] **=** 'Yes'

*# Display the updated DataFrame*

print(df)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Season | Daynum | Wteam | Lteam | Lscore | Wloc | Numot |  | Score Difference |
| 0 | 100027 | 2008 | 66 | 1203 | 49 | 1387 | H | 0 | 184 |
| 1 | 49310 | 1997 | 66 | 1157 | 61 | 1204 | H | 0 | 47 |
| 2 | 89021 | 2006 | 44 | 1284 | 41 | 1343 | A | 0 | 59 |
| 3 | 85042 | 2005 | 66 | 1131 | 73 | 1216 | H | 0 | 85 |
| 4 | 103660 | 2009 | 26 | 1326 | 59 | 1359 | H | 0 | 33 |

**Q**: Calculate the total number of games played by each team, whether they won or lost. This will require iterating over the rows of the DataFrame and updating a dictionary that keeps track of the number of games for each team.

*# Iterate through the rows of the DataFrame*

**for** index, row **in** df**.**iterrows():

*# Update the dictionary for the winning team*

**if** row['Wteam'] **in** team\_games:

team\_games[row['Wteam']] **+=** 1

**else**:

team\_games[row['Wteam']] **=** 1

*# Update the dictionary for the losing team*

**if** row['Lteam'] **in** team\_games:

team\_games[row['Lteam']] **+=** 1

**else**:

team\_games[row['Lteam']] **=** 1

*# Display the total number of games played by each team*

**for** team, games **in** team\_games**.**items():

print(f"Team {team}: {games} games")

|  |  |
| --- | --- |
| Team 66 | 3 games |
| Team 49 | 1 games |
| Team 61 | 1 games |
| Team 44 | 1 games |
| Team 41 | 1 games |
| Team 73 | 1 games |
| Team 26 | 1 games |
| Team 59 | 1 games |

**Q**: For each season, find the game with the highest score difference (winning score - losing score). This will require iterating over the rows of the DataFrame, keeping track of the highest score difference for each season, and updating it if a game with a higher score difference is found.

*Initialize a dictionary to keep track of the highest score difference for each season*

season\_high\_score\_diff **=** {}

*# Iterate through the rows of the DataFrame*

**for** index, row **in** df**.**iterrows():

season **=** row['Season']

score\_diff **=** row['ScoreDifference']

*# Update the dictionary if the current game has a higher score difference for the season*

**if** season **in** season\_high\_score\_diff:

**if** score\_diff **>** season\_high\_score\_diff[season]['ScoreDifference']:

season\_high\_score\_diff[season] **=** {'GameID': index, 'ScoreDifference': score\_diff}

**else**:

season\_high\_score\_diff[season] **=** {'GameID': index, 'ScoreDifference': score\_diff}

*# Display the game with the highest score difference for each season*

**for** season, info **in** season\_high\_score\_diff**.**items():

game\_id **=** info['GameID']

score\_diff **=** info['ScoreDifference']

print(f"Season {season}: Game ID {game\_id} has the highest score difference

Dvs.

Season 100027, med Game ID 0 har den højeste score forskel på 184

Season 49310, game id 1, har den højeste score forskel på 47

Season 89021, har id 2, og har en score på 47

Seanson 85042, med id 3, har den højeste score på 85

Season 103660, med id 4, har den højeste score på 33

**Q**: Vectorized Operation: Calculate the total number of games played by each team, whether they won or lost. Instead of iterating over the DataFrame, use the value\_counts() function on the winning team and losing team columns separately, and then add the two Series together.

*# Assuming you have a DataFrame named 'df'*

*# If the DataFrame is not named, replace 'df' with the actual name of your DataFrame*

*# Calculate the total number of games played by each team, whether they won or lost*

winning\_team\_counts **=** df['Wteam']**.**value\_counts()

losing\_team\_counts **=** df['Lteam']**.**value\_counts()

*# Add the two Series together to get the total number of games played by each team*

total\_games\_by\_team **=** winning\_team\_counts**.**add(losing\_team\_counts, fill\_value**=**0)

*# Display the total number of games played by each team*

print(total\_games\_by\_team)

|  |  |
| --- | --- |
| 26 | 1 |
| 41 | 1 |
| 44 | 1 |
| 49 | 1 |
| 59 | 1 |
| 61 | 1 |
| 66 | 3 |
| 73 | 1.0 |

**Q**: For each season, find the game with the highest score difference (winning score - losing score). Instead of iterating over the DataFrame, create a new column 'ScoreDifference' using vectorized subtraction, then use the groupby() function and idxmax() function to find the game with the highest score difference for each season.

*# Assuming you have a DataFrame named 'df'*

*# If the DataFrame is not named, replace 'df' with the actual name of your DataFrame*

*# Create the 'ScoreDifference' column using vectorized subtraction*

df['ScoreDifference'] **=** df['Wscore'] **-** df['Lscore']

*# Find the index of the game with the highest score difference for each season*

max\_score\_diff\_index **=** df**.**groupby('Season')['ScoreDifference']**.**idxmax()

*# Display the game with the highest score difference for each season*

result\_df **=** df**.**loc[max\_score\_diff\_index]

print(result\_df)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Season | | Daynum | | Wteam | Lteam | Lscore | | Wloc | | Numot | |  | | Score Difference | |
| 1 | | 49310 | | 1997 | 66 | 1157 | | 61 | | 1204 | | H | | 0 | | -47 |
| 3 | | 85042 | | 2005 | 66 | 1131 | | 73 | | 1216 | | H | | 0 | | -85 |
| 2 | | 89021 | | 2006 | 44 | 1284 | | 41 | | 1343 | | A | | 0 | | -59 |
| 0 | | 100027 | | 2008 | 66 | 1203 | | 49 | | 1387 | | H | | 0 | | -184 |
| 4 | | 103660 | | 2009 | 26 | 1326 | | 59 | | 1359 | | H | | 0 | | -33 |