

```
1 #%% md
2 # Context:
3
4 A new football club named 'GL United FC'
5 has just been inaugurated. This club
6 does not have a team yet. The team is
7 looking to hire players for their roster.
8 Management wants to make such decisions
9 using a data based approach.
10 During a recent hiring drive, you were
11 selected for the Data Science team as a
12 data scientist. Your team has been tasked
13 with creating a report which
14 recommends players for the main team. To
15 start with, a total of 15 players are
16 required. Player data for all teams has
17 been acquired from FIFA. This data
18 contains information about the players,
19 the clubs they are currently playing for
20 and various performance measures. The team
21 needs 20 possible players to
22 choose from. You have been requested to do
23 the analysis and formulate a
24 report in order to help the management
25 make a decision regarding potential
26 players.
27 #%% md
28 # Dataset Description:
29 The data contains details for over 18,000
30 players playing in various football
31 clubs in Europe. It contains information
32 on age, skill rating, wages and player
33 value, etc. The files provided are as
```

```
20 follows:  
21  
22 # fifa.csv - data file.  
23 # fifa_variable_information.csv -  
    information on individual variables  
24 #%% md  
25 # Questions: (Total points: 50)  
26 #%% md  
27 # Load and explore data (4 points)  
28 1. Import the required libraries and read  
    the dataset. (1 point)  
29 2. Check the first few samples, shape,  
    info of the data and try to familiarize  
30 yourself with different features. (2  
    points)  
31 3. Drop the columns which you think  
    redundant for the analysis. (1 point)  
32 [Hint: columns like 'Photo', 'Flag', 'Club  
    Logo']  
33 #%% md  
34 # 1. Import the required libraries and  
    read the dataset. (1 point)  
35 #%%  
36 #Step 1: Load and Explore Data (4 points)  
37 #Import Required Libraries and Read  
    Dataset  
38  
39  
40 import pandas as pd  
41 import numpy as np  
42 import matplotlib.pyplot as plt  
43 import seaborn as sns  
44 #%%
```

```
45 import warnings
46
47 # Suppress all warnings
48 warnings.filterwarnings("ignore")
49
50 # Your code that generates warnings
51
52 #%%
53 # Suppress FutureWarnings
54 warnings.simplefilter(action='ignore',
category=FutureWarning)
55
56 # Your code here
57
58 #%%
59 # Load the data
60 fifa_data = "./data/fifa.csv"
61 fifa_data_df = pd.read_csv(fifa_data)
62 fifa_var_info_data = "./data/
fifa_variable_information.csv"
63 fifa_var_info_df = pd.read_csv(
fifa_var_info_data)
64 #%% md
65 # 2. Check the first few samples, shape,
info of the data and try to familiarize
yourself with different features. (2
points)
66 #%%
67
68
69 #2. Check First Few Samples, Shape, and
Info
70
```

```
71
72 # Explore the data
73 fifa_data_df.head()
74
75 #%%
76 fifa_var_info_df.head()
77
78 #%%
79 fifa_data_df.shape
80
81 #%%
82 fifa_var_info_df.shape
83
84 #%%
85 fifa_data_df.info()
86
87 #%%
88 fifa_var_info_df.info()
89
90 #%% md
91 # Drop redundant columns
92 # 3. Drop the columns which you think
# redundant for the analysis. (1 point)
93 #%%
94
95
96 fifa_data_df.drop(['Photo', 'Flag', 'Club
# Logo'], axis=1, inplace=True)
97
98 #%%
99 fifa_data_df.head()
100
101 #%% md
```

```
102 # Data Cleaning and Preprocessing (18
    points)
103 4. Convert the columns "Value", "Wage", "
    Release Clause" to float datatype
104 after getting rid of currency symbol and
    suffix. (6 points)
105 - Note: When the record/entry has "M"(indicates millions) as suffix you
106 need to multiply that value with 1000000
107 - When the record/entry has "K"(indicates thousands) as suffix you
108 need to multiply that value with 1000
109 5. Convert the column "Joined" into
    integer data type with keeping only the
110 year. (2 points)
111 6. Convert the column "Contract Valid
    Until" to pandas datetime type. (2 points
    )
112 7. The column 'Height' is in inches with
    a quotation mark, Convert to float
113 with decimal points. (2 points)
114 8. The column "Weight" has the suffix as
    lbs, remove the suffix and convert
115 to float. (2 points)
116 9. Check for the percentage of missing
    values and impute them with
117 appropriate imputation techniques. (4
    points)
118
119 #%% md
120 # Convert currency columns to numeric
121 # 4. Convert the columns "Value", "Wage",
    "Release Clause" to float datatype
```

```
122 after getting rid of currency symbol and
123     suffix. (6 points)
124 - Note: When the record/entry has "M"(indicates millions) as suffix you
125 need to multiply that value with 1000000
126 - When the record/entry has "K"(indicates
127     thousands) as suffix you
128 need to multiply that value with 1000
129 #%%
130 def convert_currency(value):
131     # Check if the value is already a
132     float
133     if isinstance(value, float):
134         return value
135     try:
136         value = value.replace('€', '').
137         strip() # Remove currency symbol
138         if 'M' in value:
139             return float(value.replace('M',
140             '', '').strip()) * 1_000_000
141         elif 'K' in value:
142             return float(value.replace('K',
143             '', '').strip()) * 1_000
144         else:
145             return float(value)
146     except ValueError:
147         print(f"Error converting value: {value}")
148     return None # Or handle the
149     error as needed
150
151 for col in ['Value', 'Wage', 'Release
```

```
145 Clause']:
146     fifa_data_df[col] = fifa_data_df[col]
147         .apply(convert_currency)
148
149 # 5. Convert the column "Joined" into
150 # integer data type with keeping only the
151 # year. (2 points)
152
153 #%% md
154 # 6. Convert the column "Contract Valid
155 # Until" to pandas datetime type. (2 points
156
157
158
159
160 try:
161     # Attempt to convert with automatic
162     # format inference
163     fifa_data_df['Contract Valid Until']
164         = pd.to_datetime(
165             fifa_data_df['Contract Valid
166             Until'],
167                 errors='coerce', # Convert
168                 invalid dates to NaT
169                 infer_datetime_format=True # Try
170                 to infer format
171             )
```

```
167 except Exception as e:  
168     print(f"An error occurred: {e}")  
169  
170 # Optional: Check for NaT values  
171 if fifa_data_df['Contract Valid Until'].  
    isnull().any():  
172     print("There are some invalid date  
        entries that were set to NaT.")  
173  
174 %%  
175 fifa_data_df['Contract Valid Until'] = pd  
    .to_datetime(  
176     fifa_data_df['Contract Valid Until'],  
177     format='%b %d, %Y', # Specify the  
        date format explicitly  
178     errors='coerce'  
179 )  
180  
181 %% md  
182 # 7. The column 'Height' is in inches  
    with a quotation mark, Convert to float  
    with decimal points. (2 points)  
183 %%  
184  
185  
186 def convert_height_to_inches(height_str):  
187     if pd.isna(height_str): # Handle NaN  
        values  
188         return None  
189     try:  
190         height_str = height_str.replace(  
            '\"', '').strip() # Remove quotes and  
            strip whitespace
```

```
191         feet, inches = height_str.split(
192             "'") # Split by the apostrophe
193         total_inches = int(feet) * 12 +
194             int(inches) # Convert to total inches
195         return total_inches
196     except Exception as e:
197         print(f"Error converting height '{height_str}': {e}")
198     return None # Return None for
199     any invalid format
200
201 # Apply the conversion function
202 fifa_data_df['Height'] = fifa_data_df['
203 Height'].apply(convert_height_to_inches)
204
205 # Optional: Check for NaN values after
206 # conversion
207 if fifa_data_df['Height'].isnull().any():
208     print("There are some invalid height
209     entries that were set to NaN.")
210
211 # 8. The column "Weight" has the suffix
212 # as lbs, remove the suffix and convert to
213 # float. (2 points)
214
215 #%% md
216 # 9. Check for the percentage of missing
217 # values and impute them with appropriate
```

```
211 imputation techniques. (4 points)
212 #%%
213 import pandas as pd
214
215 # Calculate and print the percentage of
missing values
216 missing_percentage = fifa_data_df.isnull
().mean() * 100
217 print("Missing Percentage:\n",
missing_percentage[missing_percentage > 0
])
218
219 # Impute missing values in the 'Wage'
column with the median
220 fifa_data_df['Wage'] = fifa_data_df['Wage
'].fillna(fifa_data_df['Wage'].median())
221
222 # Optional: Verify the imputation
223 print("\nWage column after imputation:")
224 print(fifa_data_df['Wage'])
225
226 #%%
227 fifa_data_df.head()
228
229 #%%
230 fifa_data_df.shape
231
232 #%%
233 fifa_data_df.info()
234
235 #%% md
236 # Exploratory Data Analysis (28 points)
237 10. Plot the distribution of Overall
```

- 237 rating for all the players and write your
238 findings. (2 points)
- 239 11. Retrieve the names of top20 players
based on the Overall rating. (2 points)
- 240 12. Generate a dataframe which should
include all the information of the
241 Top 20 players based on the Overall
rating. (4 points)
- 242 13. What is the average "Age" and "Wage"
of these top 20 players? (use
243 the data frame created in the question 11
) (2 points)
- 244 14. Among the top 20 players based on the
Overall rating, which player has
245 the highest wage? Display the name of the
player with his wage. (2 points)
- 246 15. Generate a dataframe which should
include the "Player name", "Club
247 Name", "Wage", and 'Overall rating'. (4
Points)
- 248 - i) find the average Overall rating for
each club.
- 249 - ii) Display the average overall rating
of Top10 Clubs using a plot
- 250 16. What is the relationship between age
and individual potential of the
251 player? Visualize the relationship with
appropriate plot and Comment on
252 the same. (2 points)
- 253 17. Which features directly contribute to
the wages of the players?
- 254 Support your answer with a plot and a
metric. (2 points) (hint: use

```
255 potential, Overall, value, international  
reputation, and Release Clause)  
256 18. Find the position in the pitch where  
the maximum number of players  
257 play and the position where the minimum  
number of players play? Display  
258 it using a plot. (2 points)  
259 19. How many players are from the club '  
Juventus' and the wage is greater  
260 than 200K? Display all the information of  
such players. (2 points)  
261 20. Generate a data frame containing top  
5 players by Overall rating for  
262 each unique position. (2 Points)  
263 21. What is the average wage one can  
expect to pay for the top 5 players  
264 in every position? (use the data frame  
created in Q19) (2 points)  
265 #%% md  
266 # 10. Plot the distribution of Overall  
rating for all the players and write your  
findings. (2 points)  
267 #%%  
268 plt.figure(figsize=(10, 6))  
269 sns.histplot(fifa_data_df['Overall'],  
bins=30, kde=True)  
270 plt.title('Distribution of Overall Rating  
'')  
271 plt.xlabel('Overall Rating')  
272 plt.ylabel('Frequency')  
273 plt.show()  
274  
275 #%%
```

```
276 fifa_data_df.info()
277 #%% md
278 # 11. Retrieve the names of top20 players
      based on the Overall rating. (2 points)
279 #%%
280 top_20_players = fifa_data_df.nlargest(20
      , 'Overall')
281 print(top_20_players[['Name', 'Overall']])
282
283 #%%
284 top_20_df = top_20_players.copy()
285 top_20_df.head()
286
287 #%% md
288 # 12. Generate a dataframe which should
      include all the information of the Top 20
      players based on the Overall rating. (4
      points)
289 #%%
290 import pandas as pd
291
292 # Step 1: Ensure that the 'Overall'
      column is in numeric format
293 fifa_data_df['Overall'] = pd.to_numeric(
      fifa_data_df['Overall'], errors='coerce')
294
295 # Step 2: Sort the DataFrame by 'Overall'
      ' in descending order
296 top_players = fifa_data_df.sort_values(by
      ='Overall', ascending=False).head(20)
297
298 # Step 3: Create a new DataFrame with the
```

```
298 top 20 players
299 top_20_players_df = top_players.
    reset_index(drop=True)
300
301 # Display the new DataFrame
302 print(top_20_players_df)
303
304 #%% md
305 # 13. What is the average "Age" and "Wage"
# of these top 20 players? (use the data
frame created in the question 11) (2
points)
306 #%%
307 avg_age = top_20_df['Age'].mean()
308 avg_wage = top_20_df['Wage'].mean()
309 print(f'Average Age: {avg_age}, Average
Wage: {avg_wage}')
310
311 #%% md
312 # 14. Among the top 20 players based on
the Overall rating, which player has the
highest wage? Display the name of the
player with his wage. (2 points)
313 #%%
314 highest_wage_player = top_20_df.loc[
    top_20_df['Wage'].idxmax(), ['Name',
    'Wage']]
315 print(highest_wage_player)
316
317 #%% md
318 # 15. Generate a dataframe which should
include the "Player name", "Club Name",
"Wage", and 'Overall rating'. (4 Points)
```

```
319 - i) find the average Overall rating for
      each club.
320 - ii) Display the average overall rating
      of Top10 Clubs using a plot
321 #%%
322 club_wage_df = fifa_data_df[['Name', 'Club', 'Wage', 'Overall']]
323 avg_overall_by_club = club_wage_df.
    groupby('Club')['Overall'].mean().
    reset_index()
324 avg_overall_by_club
325 #%%
326 top_clubs = avg_overall_by_club.nlargest(
    10, 'Overall')
327 sns.barplot(data=top_clubs, x='Overall',
    y='Club')
328 plt.title('Top 10 Clubs by Average
    Overall Rating')
329 plt.show()
330
331
332 #%% md
333 # 16. What is the relationship between
      age and individual potential of the
      player? Visualize the relationship with
      appropriate plot and Comment on the same
      . (2 points)
334 #%%
335 plt.figure(figsize=(10, 6))
336 sns.scatterplot(data=fifa_data_df, x='Age',
    y='Potential')
337 plt.title('Relationship Between Age and
    Potential')
```

```
338 plt.xlabel('Age')
339 plt.ylabel('Potential')
340 plt.show()
341
342 #%%
343 import pandas as pd
344 import matplotlib.pyplot as plt
345 import seaborn as sns
346
347 # Assuming fifa_data_df is your existing
348 # DataFrame
349 # Step 1: Ensure 'Age' and 'Potential'
350 # are numeric
351 # fifa_data_df['Age'] = pd.to_numeric(
352 #     fifa_data_df['Age'], errors='coerce')
353 # fifa_data_df['Potential'] = pd.to_numeric(
354 #     fifa_data_df['Potential'], errors='
355 # coerce')
356 # Step 2: Drop rows with NaN values in '
357 # Age' or 'Potential'
358 data = fifa_data_df.dropna(subset=['Age',
359 , 'Potential'])
360
361 # Step 3: Create a scatter plot
362 plt.figure(figsize=(10, 6))
363 sns.scatterplot(data=data, x='Age', y='
364 Potential', alpha=0.6)
365 plt.title('Relationship between Age and
366 Player Potential')
367 plt.xlabel('Age')
368 plt.ylabel('Potential')
```

```
362 plt.axhline(y=data['Potential'].mean(),  
                 color='r', linestyle='--', label='Mean  
                 Potential')  
363 plt.legend()  
364 plt.grid(True)  
365 plt.show()  
366  
367 #%% md  
368 #Interpretation:  
369 Trend Analysis: In the scatter plot, we  
can look for trends. Typically, we might  
expect to see that younger players have  
higher potential, while older players  
might show a decline in potential.  
370 Correlation: If there is a visible  
downward trend, it may indicate that as  
players age, their potential decreases.  
Conversely, if potential remains  
relatively high for younger players and  
declines more slowly as age increases, it  
suggests a different pattern.  
371  
372 #Commenting on the Relationship:  
373 After generating the plot, observe the  
distribution of points. If most young  
players have higher potential, we might  
conclude that younger players are  
generally seen as having more room for  
growth.  
374 If there are outliers (e.g., older  
players with high potential), we can  
discuss how experience or skill can  
influence potential despite age.
```

```
375 %% md
376 # 17. Which features directly contribute
   to the wages of the players? Support your
   answer with a plot and a metric. (2
   points) (hint: use potential, Overall,
   value, international reputation, and
   Release Clause)
377 %%
378 features = ['Potential', 'Overall', '
   Value', 'International Reputation', '
   Release Clause']
379 sns.heatmap(fifa_data_df[features].corr()
   (), annot=True, cmap='coolwarm')
380 plt.title('Correlation Heatmap')
381 plt.show()
382
383 %%
384 import pandas as pd
385 import matplotlib.pyplot as plt
386 import seaborn as sns
387
388 # Assuming fifa_data_df is your existing
   DataFrame
389
390 # Step 1: Convert relevant columns to
   numeric and drop NaNs
391 features = ['Wage', 'Potential', 'Overall'
   ', 'Value', 'International Reputation', '
   Release Clause']
392 for feature in features:
393     fifa_data_df[feature] = pd.to_numeric(
       (fifa_data_df[feature], errors='coerce'))
394
```

```
395 # Drop rows with NaN values in the  
      relevant columns  
396 data = fifa_data_df.dropna(subset=  
      features)  
397  
398 # Step 2: Calculate correlation  
      coefficients  
399 correlation_matrix = data[features].corr()  
400 print(correlation_matrix['Wage'])  
401  
402 # Step 3: Visualize relationships  
403 plt.figure(figsize=(15, 10))  
404 for i, feature in enumerate(features[1:  
      :], start=1):  
405     plt.subplot(2, 3, i)  
406     sns.scatterplot(data=data, x=feature  
      , y='Wage', alpha=0.6)  
407     plt.title(f'Wage vs {feature}')  
408     plt.xlabel(feature)  
409     plt.ylabel('Wage')  
410     plt.grid(True)  
411  
412 plt.tight_layout()  
413 plt.show()  
414  
415 #%% md  
416 #Explanation:  
417 Counting Players: The value_counts()  
      method counts the occurrences of each  
      position in the "Position" column.  
418 Identifying Max and Min: The idxmax() and  
      idxmin() functions help to find the
```

```
418 positions with the maximum and minimum  
counts, respectively.  
419 Visualization: A bar plot is created to  
show the number of players in each  
position. Horizontal lines indicate the  
counts of the positions with the maximum  
and minimum players for clarity.  
420  
421 #Interpretation:  
422 After running the code, we will see a bar  
plot displaying the distribution of  
players across different positions.  
423 The printed output will specify the  
position with the maximum number of  
players and the position with the minimum  
number of players.  
424 #%% md  
425 # 18. Find the position in the pitch  
where the maximum number of players play  
and the position where the minimum number  
of players play? Display it using a plot  
. (2 points)  
426 #%%  
427 position_counts = fifa_data_df['Position'  
].value_counts()  
428 position_counts.plot(kind='bar')  
429 plt.title('Player Positions Count')  
430 plt.xlabel('Position')  
431 plt.ylabel('Number of Players')  
432 plt.show()  
433  
434 #%%  
435 import pandas as pd
```

```
436 import matplotlib.pyplot as plt
437
438 # Step 1: Count the number of players in
439 # each position
439 position_counts = fifa_data_df['Position']
439     .value_counts()
440
441 # Step 2: Identify the position with
441 # maximum and minimum players
442 max_position = position_counts.idxmax()
443 min_position = position_counts.idxmin()
444 max_count = position_counts.max()
445 min_count = position_counts.min()
446
447 # Print the results
448 print(f"Position with maximum players: {max_position} ({max_count} players)")
449 print(f"Position with minimum players: {min_position} ({min_count} players)")
450
451 # Step 3: Plot the number of players by
451 # position
452 plt.figure(figsize=(12, 6))
453 position_counts.plot(kind='bar', color='skyblue')
454 plt.title('Number of Players by Position')
455 plt.xlabel('Position')
456 plt.ylabel('Number of Players')
457 plt.xticks(rotation=45)
458 plt.grid(axis='y')
459
460 # Highlight the max and min positions
```

```
461 plt.axhline(y=max_count, color='red',
   linestyle='--', label=f'Max: {max_position}')
462 plt.axhline(y=min_count, color='green',
   linestyle='--', label=f'Min: {min_position}')
463 plt.legend()
464
465 plt.tight_layout()
466 plt.show()
467
468 #%% md
469 # 19. How many players are from the club
   'Juventus' and the wage is greater than
   200K? Display all the information of such
   players. (2 points)
470 #%%
471 juventus_players = fifa_data_df[(
   fifa_data_df['Club'] == 'Juventus') & (
   fifa_data_df['Wage'] > 200000)]
472 print(juventus_players)
473
474 #%% md
475 # 20. Generate a data frame containing
   top 5 players by Overall rating for each
   unique position. (2 Points)
476 #%%
477 top_players_per_position0 = fifa_data_df.
   groupby('Position').apply(
478     lambda x: x.nlargest(5, 'Overall'),
479     include_groups=False
480 ).reset_index(drop=True)
481
```

```
482 # Display the result
483 print(top_players_per_position0)
484
485 #%%
486 top_players_per_position0.head()
487
488 #%%
489 import pandas as pd
490
491 # Group by 'Position' and get top 5
492 # players by 'Overall'
493 top_players_per_position1 = (
494     fifa_data_df.groupby('Position',
495     as_index=False)
496     .apply(lambda x: x.nlargest(5,
497     'Overall'))
498     .reset_index(drop=True)
499 )
500
501 #%%
502 top_players_per_position2 = (
503     fifa_data_df.groupby('Position',
504     as_index=False)
505     .apply(lambda x: x.nlargest(5,
506     'Overall'))[['Name', 'Overall']])
507     .reset_index(drop=True)
508 )
509 %%
```

```
510 print(top_players_per_position0.head())
511 print(top_players_per_position0.columns)
512
513 #%%
514 print(top_players_per_position0.columns.
      tolist())
515
516 #%%
517 top_players_per_position3 = (
518     fifa_data_df.groupby('Position',
519     as_index=False)
520     .apply(lambda x: x.nlargest(5,
521         'Overall'))
522 )
523
524 # Resetting index to clean up the
525 # DataFrame
526 top_players_per_position3 =
527     top_players_per_position3.reset_index(
528     drop=True)
529
530 # Display the result
531 print(top_players_per_position3)
532
533 #%%
534 # Top 5 Players with Overall Rating for
535 # each Unique Position
536 # Display the DataFrame and its columns
537 print(top_players_per_position0.head())
538 print(top_players_per_position0.columns.
      tolist())
539
540 # Strip any whitespace from column names
```

```
535 top_players_per_position0.columns =  
      top_players_per_position0.columns.str.  
      strip()  
536  
537 # Check for the updated column names  
538 print(top_players_per_position0.columns.  
      tolist())  
539  
540 # Assuming "Position" is the correct name  
    , calculate the average wage  
541 if 'Position' in  
    top_players_per_position0.columns and '  
    Wage' in top_players_per_position0.  
    columns:  
542     avg_wage_by_position0 =  
     top_players_per_position0.groupby('  
    Position')[['Wage']].mean().reset_index()  
543     print(avg_wage_by_position0)  
544 else:  
545     print("Columns 'Position' or 'Wage'  
    are missing.")  
546  
547 %% md  
548 # 21. What is the average wage one can  
    expect to pay for the top 5 players in  
    every position? (use the data frame  
    created in Q19) (2 points)  
549 %%  
550 # Making sure both columns exist  
551 if 'Position' in  
    top_players_per_position0.columns and '  
    Wage' in top_players_per_position0.  
    columns:
```

```
552     avg_wage_by_position = (
553         top_players_per_position0.groupby(
554             ['Position'])['Wage'].mean().reset_index()
555     )
556     # Rename columns for clarity (
557     # optional)
558     avg_wage_by_position.columns = [
559         'Position', 'Average Wage']
560     # Display the result
561     print(avg_wage_by_position)
562 else:
563     print("Columns 'Position' or 'Wage'
564 are missing.")
565 # Position and Average Wage using
566 # fifa_data_df
567 import pandas as pd
568
569 # fifa_data_df is your DataFrame with '
570 # Position' and 'Wage' columns
571 # Step 1: Ensure 'Wage' is in a numeric
572 # format if it's not already
573 fifa_data_df['Wage'] = pd.to_numeric(
574     fifa_data_df['Wage'], errors='coerce')
575
576 # Step 2: Group by 'Position' and select
577 # the top 5 players by 'Wage'
578 top_players_per_position = (
579     fifa_data_df
```

```
576     .loc[:, ['Position', 'Wage']] #  
      Select only relevant columns first  
577     .groupby('Position', group_keys=False  
      ) # group_keys=False to suppress the  
      warning  
578     .apply(lambda x: x.nlargest(5, 'Wage'  
      ))  
579 )  
580  
581 # Step 3: Calculate the average wage for  
      the top 5 players in each position  
582 avg_wage_by_position = (  
583     top_players_per_position.groupby(  
      'Position')['Wage'].mean().reset_index()  
584 )  
585  
586 # Rename columns for clarity (optional)  
587 avg_wage_by_position.columns = ['Position'  
      , 'Average Wage']  
588  
589 # Display the result  
590 print(avg_wage_by_position)  
591  
592 #%%  
593 # Alternate Solution  
594 import pandas as pd  
595  
596 # Step 1: Convert 'Wage' to string to  
      handle it safely  
597 fifa_data_df['Wage'] = fifa_data_df['Wage'  
      ].astype(str)  
598  
599 # Step 2: Clean and convert 'Wage' to
```

```
599 numeric
600 fifa_data_df['Wage'] = (
601     fifa_data_df['Wage']
602     .str.replace('€', '', regex=False)
# Remove Euro sign
603     .str.replace('K', '', regex=False)
# Remove K
604     .str.replace(',', '', regex=False)
# Remove commas
605     .astype(float) * 1 # Convert to float and multiply by 1
606 )
607
608 # Step 3: Group by Position and find the top 5 players by Wage
609 top_players_per_position = fifa_data_df.
groupby('Position').apply(lambda x: x.
nlargest(5, 'Wage')).reset_index(drop=True)
610
611 # Step 4: Calculate the average wage for the top 5 players in each position
612 avg_wage_by_position =
top_players_per_position.groupby(
'Position')['Wage'].mean().reset_index()
613
614 # Step 5: Display the average wages
615 print(avg_wage_by_position)
616
```