

Analyze the provided dataset to identify its key characteristics for data-driven decision-making. Upon receiving a specific analytical question, address it strictly using the dataset. Your deliverables must include a clear, concise answer to the question and the Python code that extracts and processes the data used to derive your answer. The dataset should be cleaned by filtering admission_period to include only values between 2006.1 and 2019.2, removing entries where enrollment_type is labeled as "Exempted", and excluding those where situation is "In Progress". The resulting cleaned dataset should be stored in a new DataFrame named df_filtered.

Anexo oculto

I'll first analyze the dataset's key characteristics and then provide the cleaning code. I'll wait for your specific analytical question to proceed with targeted analysis.

Key Characteristics of the Dataset for Data-Driven Decision-Making

1. Student Demographics:

- Columns: gender, color_code, color_name, admission_age, dropout_age
- Use: Understand student diversity, age trends, and potential correlations with academic outcomes.

2. Admission Details:

- Columns: admission_code, admission_period, admission_form, general_average
- Use: Analyze entry pathways (e.g., Entrance Exam, SISU), admission timing, and initial academic performance.

3. Academic Progress:

- Columns: enrollment_period, enrollment_type, situation, final_average, credits, workload
- Use: Track student performance, completion rates, and course load impact.

4. Dropout Information:

- Columns: dropout_code, dropout_form, dropout_period, dropout_period_count
- Use: Identify dropout patterns, reasons, and timing for retention strategies.

5. Course and Subject Details:

- Columns: curriculum_code, subject_code, subject_name, sector_name, relative_period
- Use: Assess curriculum effectiveness, subject difficulty, and academic unit performance.

6. Quota and Affirmative Action:

- Columns: affirmative_action, quota_type, quota_form
- Use: Evaluate equity policies and their impact on enrollment and success.

7. High School Background:

- Columns: high_school_type, high_school
- Use: Correlate pre-university education with university outcomes.

Data Cleaning Code

Below is the Python code to clean the dataset as specified:

- Filter admission_period between 2006.1 and 2019.2.
- Remove entries where enrollment_type is "Exempted".
- Exclude entries where situation is "In Progress".
- Store the result in df_filtered.

```
python ... ⌂ Copiar
import pandas as pd

# Assuming the dataset is loaded as a DataFrame named 'df'
# If you provide a file, replace this with pd.read_csv('filename.csv')
data = """id,curriculum_code,admission_code,admission_period,admission_form,dropout_code
Q386391,1990,1,1979,2,Entrance Exam,4,Cancellation Due to Abandonment,1989,1,20,0,Dropou
A261799,1999,16,2011,1,Entrance Exam / ENEM,1,Graduate,2016,1,11,0,Graduate,No,1,Without
..."""\ # Truncated for brevity; replace with full data loading

# Load the dataset (replace with actual file path if provided)
df = pd.read_csv(pd.compat.StringIO(data)) # For demo; use pd.read_csv('file.csv') in production

# Clean the dataset
df_filtered = df[
```

```

        (df['admission_period'].between(2006.1, 2019.2)) & # Filter admission period
        (df['enrollment_type'] != 'Exempted') & # Remove Exempted enrollment type
        (df['situation'] != 'In Progress') # Exclude In Progress situations
    ].copy()

# Display the shape of the cleaned dataset
print(f"Original dataset shape: {df.shape}")
print(f"Filtered dataset shape: {df_filtered.shape}")

```

Notes

- The code assumes the dataset is loaded into a DataFrame `df`. If you provide a file, I'll adjust the loading method.
- The filtered DataFrame `df_filtered` is ready for specific analytical questions.
- Please provide your question, and I'll deliver a concise answer with tailored Python code using `df_filtered`.

Let me know your specific analytical question!

Analyze how a student's age of entry and dropout influences their overall average (mean of non-null `final_average` grades). Apply one statistical method (e.g., regression or correlation) to measure significance, considering confounding factors.

I'll analyze how a student's admission age and dropout age influence their overall average (mean of non-null `final_average` grades) using the cleaned dataset. I'll apply a **multiple linear regression** to assess the relationships, considering confounding factors like `gender`, `admission_form`, and `high_school_type`. This approach allows us to measure the significance of each predictor while controlling for others.

Steps

- Use the cleaned `df_filtered` DataFrame (admission_period 2006.1–2019.2, no "Exempted" enrollment_type, no "In Progress" situation).
- Calculate each student's overall average as the mean of their non-null `final_average` grades.
- Perform multiple linear regression with `admission_age` and `dropout_age` as primary predictors, including confounders.
- Report coefficients, p-values, and R-squared to evaluate significance and fit.

Python Code

```

python ... ⚡ Copiar

import pandas as pd
import statsmodels.api as sm
import numpy as np

# Assuming df_filtered is already created from the previous cleaning step
# If not, here's the cleaning recap (replace with full data loading if needed):
data = """id,curriculum_code,admission_code,admission_period,admission_form,dropout_code
Q386391,1990,1,1979.2,Entrance Exam,4,Cancellation Due to Abandonment,1989.1,20.0,Dropou
A261799,1999,16,2011.1,Entrance Exam / ENEM,1,Graduate,2016.1,11.0,Graduate,No,1,Without
..."""\n# Truncated; replace with full dataset
df = pd.read_csv(pd.compat.StringIO(data))
df_filtered = df[
    (df['admission_period'].between(2006.1, 2019.2)) &
    (df['enrollment_type'] != 'Exempted') &
    (df['situation'] != 'In Progress')
].copy()

# Calculate overall average per student (mean of non-null final_average)
student_avg = df_filtered.groupby('id')['final_average'].mean().reset_index()
student_avg.columns = ['id', 'overall_average']

# Merge with student-level data (taking first occurrence for simplicity)
student_data = df_filtered.drop_duplicates(subset='id')[[
    'id', 'admission_age', 'dropout_age', 'gender', 'admission_form', 'high_school_type
]]
df_analysis = pd.merge(student_avg, student_data, on='id', how='left')

# Handle missing values
df_analysis = df_analysis.dropna(subset=['overall_average', 'admission_age']) # Drop rows
df_analysis['dropout_age'] = df_analysis['dropout_age'].fillna(0) # Replace NaN dropout

# Prepare variables for regression
X = df_analysis[['admission_age', 'dropout_age']]
X = pd.get_dummies(X.join(df_analysis[['gender', 'admission_form', 'high_school_type']]))

y = df_analysis['overall_average']

# Add constant for intercept
X = sm.add_constant(X)

# Fit multiple linear regression model

```

```
model = sm.OLS(y, X).fit()

# Print summary
print(model.summary())

# Extract key results
coeff_admission_age = model.params['admission_age']
pval_admission_age = model.pvalues['admission_age']
coeff_dropout_age = model.params['dropout_age']
pval_dropout_age = model.pvalues['dropout_age']
r_squared = model.rsquared
```

Answer

Based on the dataset filtered for admission periods 2006.1–2019.2, excluding "Exempted" enrollment types and "In Progress" situations, here's the analysis:

- **Admission Age:** The regression coefficient for `admission_age` indicates its effect on the overall average. For example, if the coefficient is -0.05 (hypothetical from sample size), each additional year of admission age decreases the overall average by 0.05 points, holding other factors constant. The p-value (e.g., 0.03 if $p < 0.05$) suggests statistical significance, implying a meaningful relationship.
- **Dropout Age:** The coefficient for `dropout_age` shows its influence. If it's 0.02 with a p-value of 0.15 (hypothetical), it suggests a slight positive effect (0.02 points per year) but lacks statistical significance ($p > 0.05$), meaning the effect is not reliably different from zero.
- **Model Fit:** The R-squared value (e.g., 0.12) indicates that 12% of the variance in overall average is explained by the model, including confounders like gender, admission form, and high school type.
- **Confounders:** Variables like `gender_Male`, `admission_form_Unified Selection System (SISU)`, and `high_school_type_Only Private School` (dummy variables) control for their effects. Significant coefficients here ($p < 0.05$) would indicate their influence on grades.

Interpretation

- **Admission Age:** A significant negative coefficient suggests older entrants may struggle academically, possibly due to longer gaps since high school or differing life responsibilities.
- **Dropout Age:** A non-significant coefficient implies dropout age has little consistent impact on grades, perhaps because it reflects post-performance decisions rather than ability.
- **Limitations:** With only 88 rows in the sample (and fewer after cleaning), results may lack power. Actual coefficients and p-values depend on the full dataset.

Please run the code with the complete dataset for precise coefficients and p-values. Let me know if you'd like adjustments or a different statistical method!



Entre para continuar a conversa