

# Student Scores: Ready-to-Upload PDF (expects student\_scores.csv)

This PDF is a ready-to-upload report template. It assumes there is a CSV file named **student\_scores.csv** in the same folder where you run the example Python script below.

What this PDF contains: A sample Python script that reads **student\_scores.csv** and computes mean, min, and max for numeric columns. A small example table and sample output computed from the example CSV included here for preview. Instructions on how to run the script locally.

## Example **student\_scores.csv** (5 rows):

student_id	name	math	physics	chemistry
1	Alice	78	85	92
2	Bob	66	73	70
3	Charlie	90	88	84
4	Diana	55	60	58
5	Evan	82	79	80

## Python script to analyze 'student\_scores.csv':

```
# sample_script.py
# Place this file in the same folder as student_scores.csv and run: python sample_script.py
import pandas as pd

def analyze_csv(file_path="student_scores.csv"):
    data = pd.read_csv(file_path)
    # compute mean, min, max for numeric columns only
    stats = data.describe().loc[['mean','min','max']]
    return stats

if __name__ == "__main__":
    stats = analyze_csv("student_scores.csv")
    print("== Descriptive statistics ==")
    print(stats)
```

## Sample output (from the example CSV shown above):

stat	student_id	math	physics	chemistry
mean	3.0	74.2	77.0	76.8
min	1.0	55.0	60.0	58.0
max	5.0	90.0	88.0	92.0

**Notes:** Make sure the CSV contains numeric columns (e.g., math, physics, chemistry) for which mean/min/max make sense. If your CSV uses a different filename or different column names, either rename the file to **student\_scores.csv** or edit the script's parameter. Run the script with Python 3 and ensure pandas is installed: pip install pandas.