```
"""
generate_report.py
Example script to read sample_data.csv, analyze it, and generate a PDF report.
"""
import pandas as pd
import matplotlib.pyplot as plt
from reportlab.lib.pagesizes import letter
from reportlab.lib.units import cm
from reportlab.pdfgen import canvas
from reportlab.platypus import Paragraph, SimpleDocTemplate, Spacer, Image, Table, TableStyle
from reportlab.lib import colors
from reportlab.lib.styles import getSampleStyleSheet
def main():
    data = pd.read_csv("sample_data.csv", parse_dates=["date"])
    summary = data["value"].describe().round(2).to_frame().T
    by_category =
data.groupby("category")["value"].agg(["count","mean","sum"]).round(2).reset_index()
    plt.figure(figsize=(6,3.5))
    plt.plot(data["date"], data["value"])
    plt.title("Values over time")
    plt.xlabel("Date")
    plt.ylabel("Value")
    plt.tight_layout()
    plt.savefig("chart.png")
    plt.close()
    doc = SimpleDocTemplate("report_output.pdf", pagesize=letter, rightMargin=2*cm, leftMargin=2*cm,
topMargin=2*cm, bottomMargin=2*cm)
    styles = getSampleStyleSheet()
    story = []
    story.append(Paragraph("Automated Report Generation - Sample Output", styles['Title']))
    story.append(Spacer(1,12))
    story.append(Paragraph("This report was generated by the example script.", styles['BodyText']))
    story.append(Spacer(1,12))
    # summary table
    summary_table_data = [["count","mean","std","min","25%","50%","75%","max"],
                          [str(int(summary['count'])), str(summary['mean'].values[0]),
str(summary['std'].values[0]), str(summary['min'].values[0]), str(summary['25%'].values[0]),
str(summary['50%'].values[0]), str(summary['75%'].values[0]), str(summary['max'].values[0])]]
    t = Table(summary_table_data, hAlign='LEFT')
    t.setStyle(TableStyle([('BACKGROUND',(0,0),(7,0),colors.HexColor("#F2F2F2")),
('GRID',(0,0),(-1,-1),0.5,colors.black)]))
    story.append(t)
    story.append(Spacer(1,12))
    # by-category table
    cat_table_data = [by_category.columns.tolist()] + by_category.values.tolist()
    cat_table = Table(cat_table_data, hAlign='LEFT')
    cat_table.setStyle(TableStyle([('BACKGROUND',(0,0),(-1,0),colors.HexColor("#F2F2F2")),
('GRID',(0,0),(-1,-1),0.5,colors.black)]))
    story.append(cat_table)
    story.append(Spacer(1,12))
    # chart
    story.append(Paragraph("Time-series chart:", styles['Heading2']))
    story.append(Spacer(1,6))
    img = Image("chart.png")
    img._restrictSize(16*cm, 9*cm)
    story.append(img)
    doc.build(story)
if __name__ == "__main__":
    main()
```