《Python程序设计（实训）》

课程实训报告

设计名称： 课程实训1：数字档案

专业班级： 23高本智能 班

学生姓名： 莫锦培

组长姓名： 陈业信

组员姓名： 莫锦培

指导老师： 陆紫光

提交日期： 2025年6月18日星期三

# 一、作品成品截图

# 文化宫文化部

# 二、源代码（每一行代码加注释）

import streamlit as st

import pandas as pd

import plotly.express as px

def get\_dataframe\_from\_excel(file\_path):

"""从Excel文件读取数据并返回DataFrame"""

try:

df = pd.read\_excel(

file\_path,

sheet\_name='销售数据',

skiprows=1,

index\_col='订单号'

)

df['小时数'] = pd.to\_datetime(df["时间"], format="%H:%M:%S").dt.hour

return df

except FileNotFoundError:

st.error(f"文件未找到: {file\_path}")

return pd.DataFrame() # 返回空DataFrame

except Exception as e:

st.error(f"读取文件时出错: {str(e)}")

return pd.DataFrame()

def add\_sidebar\_func(df):

"""添加侧边栏筛选功能并返回筛选后的数据"""

with st.sidebar:

st.header("请筛选数据：")

city\_unique = df["城市"].unique()

city = st.multiselect(

"请选择城市：",

options=city\_unique,

default=city\_unique,

)

customer\_type\_unique = df["顾客类型"].unique()

customer\_type = st.multiselect(

"请选择顾客类型：",

options=customer\_type\_unique,

default=customer\_type\_unique,

)

gender\_unique = df["性别"].unique()

gender = st.multiselect(

"请选择性别：",

options=gender\_unique,

default=gender\_unique,

)

df\_selection = df.query(

"城市 == @city & 顾客类型 == @customer\_type & 性别 == @gender"

)

return df\_selection

def product\_line\_chart(df):

"""生成按产品类型划分的销售额图表"""

sales\_by\_product\_line = df.groupby(by=["产品类型"])[["总价"]].sum().sort\_values(by="总价")

fig\_product\_sales = px.bar(

sales\_by\_product\_line,

x="总价",

y=sales\_by\_product\_line.index,

orientation="h",

title="<b>按产品类型划分的销售额</b>",

)

return fig\_product\_sales

def hour\_chart(df):

"""生成按小时划分的销售额图表"""

sales\_by\_hour = df.groupby(by=["小时数"])[["总价"]].sum()

fig\_hour\_sales = px.bar(

sales\_by\_hour,

x=sales\_by\_hour.index,

y="总价",

title="<b>按小时数划分的销售额</b>",

)

return fig\_hour\_sales

def main\_page\_demo(df):

"""主界面展示函数"""

st.title(':bar\_chart:销售仪表板')

left\_key\_col, middle\_key\_col, right\_key\_col = st.columns(3)

total\_sales = int(df["总价"].sum())

average\_rating = round(df["评分"].mean(), 1)

star\_rating\_string = ":star:" \* int(round(average\_rating, 0))

average\_sale\_by\_transaction = round(df["总价"].mean(), 2)

with left\_key\_col:

st.subheader("总销售额：")

st.subheader(f"RMB ￥ {total\_sales:,}")

with middle\_key\_col:

st.subheader("顾客评分的平均值：")

st.subheader(f"{average\_rating} {star\_rating\_string}")

with right\_key\_col:

st.subheader("每单的平均销售额：")

st.subheader(f"RMB ￥ {average\_sale\_by\_transaction}")

st.divider()

left\_chart\_col, right\_chart\_col = st.columns(2)

with left\_chart\_col:

hour\_fig = hour\_chart(df)

st.plotly\_chart(hour\_fig, use\_container\_width=True)

with right\_chart\_col:

product\_fig = product\_line\_chart(df)

st.plotly\_chart(product\_fig, use\_container\_width=True)

def run\_app():

"""启动应用"""

st.set\_page\_config(

page\_title="销售仪表板",

page\_icon=":bar\_chart:",

layout="wide"

)

# 指定文件路径

file\_path = r'C:\Users\Lenovo\Desktop\supermarket\_sales.xlsm'

# 读取数据

sale\_df = get\_dataframe\_from\_excel(file\_path)

if not sale\_df.empty:

# 添加侧边栏筛选

df\_selection = add\_sidebar\_func(sale\_df)

# 显示主界面

main\_page\_demo(df\_selection)

# 运行应用

if \_\_name\_\_ == "\_\_main\_\_":

run\_app()

# 三、网络访问地址

# http://172.21.100.23:8501

# 四、心得与注意事项

****数据处理与可视化的结合：****使用 Streamlit 构建交互式仪表板非常高效，可以实时展示数据变化。Plotly Express 的图表美观且交互性强，适合业务数据展示。

****路径管理的重要性：****硬编码文件路径会导致代码可移植性差，建议使用配置文件或命令行参数管理路径。Windows 路径中的反斜杠需特别处理（使用原始字符串 r'path'），避免转义问题。

****依赖管理的必要性：****数据分析项目需明确列出所有依赖库（如 pandas, plotly, openpyxl），建议使用 requirements.txt 统一管理。不同格式的 Excel 文件（.xlsx vs .xlsm）可能需要额外的依赖（如 openpyxl）。要注意的是

****环境配置：****建议使用虚拟环境（venv 或 conda）隔离项目依赖，避免版本冲突。安装依赖时确保网络连接稳定，部分库（如 plotly）体积较大。

****Excel 文件处理：****xlsm 格式需安装 openpyxl 库，.xls 格式可能需要 xlrd 库。检查 Excel 文件的工作表名称、列名是否与代码中一致，避免拼写错误。