File Handling Full Example

February 1, 2026

[1]:
```
# read from data.txt
# create a new column OfferPrice(based on the material of the cloth)
# wool --> 5% discount on price, cotton -->7%, cashmere --> 6%
# leather -->2%, polyster-->10%(offerprice= price*(1-d%/100))
# round the value to 2 decimal places

# create a new column trend based on popularity score
# if ps>9 --> Emerging, ps> 7--> Trending,ps > 5--> classic, else outdated
# print header & processed data
```

[ ]:
```
new_lines=[]
with open("data.txt","r") as f:
    header=f.readline().strip().split(",")
    # new_lines.append(header+",OfferPrice,Trend\n")
    header.pop(1)  # remove Category column from header
    print(",".join(header)+",OfferPrice,Trend")

    # now modify columns
    for line in f:
        cols=line.strip().split(",")

        # modifying a current column
        # like appending Ni to col 0
        cols[0]="Ni-"+cols[0]

        # creating new column OfferPrice
        material=cols[2]
        price=float(cols[3])
        discount={
            "Wool":0.95, # 5% discount --> pay 95%
            "Cotton":0.93,
            "Cashmere":0.94,
            "Leather":0.98,
            "Polyster":0.90
        }
        offerprice=str(round(price*discount.get(material,1),2)) # default no␣
 ↪discount
```

```
        cols.append(offerprice)
        # creating new column Trend
        ps=float(cols[4])
        if ps>9:
            trend="Emerging"
        elif ps>7:
            trend="Trending"
        elif ps>5:
            trend="Classic"
        else:
            trend="Outdated"

        cols.append(trend)
        # new_lines.append(",".join(cols)) # basic mistake if we are asked to write the content into new file,there add "\n"

        # now if we want to delete any column
        del cols[1] # delete Category column

        # delete a row based on some condition
        if material=="Polyster": # removes all polyster rows
            continue  # skip this line

        # delete
        new_lines.append(",".join(cols))

    # for line in new_lines:
    #     print(line.strip())

# to write updated data into new file
with open("data_updated.txt","w") as f:
    f.write(",".join(header)+",OfferPrice,Trend\n")
    for line in new_lines:
        f.write(line+"\n")
```

```
Brand,Material,Price,Popularity_Score,OfferPrice,Trend
```

[15]:
```
# fully updated code
def process_file():
    with open("products.txt","r") as f:
        lines=f.readlines()
    new_lines=[]

    # process header
    header=lines[0].strip().split(",") # list of columns

    # first delete an unwanted column(entire column)
```

```
del_idx = header.index("Category")
header.pop(del_idx)

# then calculate the updated column idexes
price_idx=header.index("Price")
material_idx=header.index("Material")
popularity_idx=header.index("Popularity_Score")


# add new column OfferPrice & Trend
header.append("OfferPrice")
header.append("Trend")

# append to new lines
new_lines.append(','.join(header))

# Row processing
for line in lines[1:]:
    cols=line.strip().split(",")

    # here also delete column value also
    cols.pop(del_idx)
    # delete entire row based on condition
    # delete row if price <100
    if float(cols[price_idx])<100:
        continue

    # update an existing column
    # increase popularity score by 1
    cols[popularity_idx]=str(float(cols[popularity_idx])+1)

    # delete particular value
    # remove material value if it is polyster
    if cols[material_idx]=="Polyster":
        cols[material_idx]=""


    # now create new column OfferPrice
    material=cols[material_idx]
    price=float(cols[price_idx])
    discount={
        "Wool":0.95, # 5% discount --> pay 95%
        "Cotton":0.93,
        "Cashmere":0.94,
        "Leather":0.98,
        "Polyster":0.90
    }
```

```
        offerprice=str(round(price*discount.get(material,1),2)) # default no discount

        cols.append(offerprice)

        # trend
        ps=float(cols[popularity_idx])
        if ps > 9:
            trend = "Emerging"
        elif ps > 7:
            trend = "Trending"
        elif ps > 5:
            trend = "Classic"
        else:
            trend = "Outdated"
        cols.append(trend)

        new_lines.append(','.join(cols))

    print(",".join(header))
    for line in new_lines[1:]:
        print(line)

    # now write to new file
    with open("products_updated.txt","w") as f:
        f.write(",".join(header)+"\n")
        for line in new_lines[1:]:
            f.write(line+"\n")

process_file()
```

```
Brand,Material,Price,Popularity_Score,OfferPrice,Trend
Nike,Wool,710.14,5.0,674.63,Outdated
Nike,Cashmere,414.74,10.4,389.86,Emerging
Nike,Wool,239.64,3.4,227.66,Outdated
Nike,Cotton,435.24,8.7,404.77,Trending
Nike,Leather,782.43,9.3,766.78,Emerging
Nike,,188.42,6.4,188.42,Classic
Nike,,313.17,10.3,313.17,Emerging
```

[ ]:
```
# # New Example
# # ProductID,Brand,Category,Material,CostPrice,SellingPrice,Stock,Rating
# P101, Adidas, Shoes, Leather, 3200, 4599, 12, 8.6
# P102, Nike, Shoes, Polyster, 2800, 3999, 5, 7.2
# P103, Puma, Jacket, Cotton, 2100, 3499, 0, 6.1
# P104, Adidas, Jacket, Wool, 4100, 5999, 8, 9.1
# P105, Nike, TShirt, Cotton, 700, 1299, 25, 8.9
```

```
# P106, Puma, Shoes, Leather, 3600, 4999, 3, 9.4
# P107, Reebok, TShirt, Polyster, 650, 1199, 0, 5.8
# P108, Nike, Jacket, Cashmere, 5200, 7999, 2, 9.7
# P109, Adidas, Shoes, Cotton, 2600, 3799, 15, 7.5
# P110, Puma, TShirt, Cotton, 800, 1499, 18, 8.1
```

[ ]:
```
def process_file():
    new_lines=[]
    with open("inventory.txt","r") as f:
        lines=f.readlines()

    header=lines[0].strip().split(",")

    # 1. Delete Material Column
    del_idx = header.index("Material")
    header.pop(del_idx)


    stock_idx=header.index("Stock")
    sellingprice_idx=header.index("SellingPrice")
    costprice_idx=header.index("CostPrice")
    category_idx=header.index("Category")
    rating_idx=header.index("Rating")

    # append new column Profit
    header.append("Profit")

    # append a new column Stock status
    header.append("StockStatus")

    new_lines.append(",".join(header))

    for line in lines[1:]:
        cols=line.strip().split(",")
        cols.pop(del_idx)

        # 2.delete all rows where stock=0
        if int(cols[stock_idx])==0:
            continue

        # 3. now update selling price if category is shoes & rating >8
        if cols[category_idx]=="Shoes" and float(cols[rating_idx])>8:
            updated_price=round(float(cols[sellingprice_idx])*1.10,2)
            cols[sellingprice_idx]=str(updated_price)

        # 4. add new column Profit
        cost_price=float(cols[costprice_idx])
```

```
        selling_price=float(cols[sellingprice_idx])
        profit=round(selling_price - cost_price,2)
        # 6. delete rows where profit<500 or rating<6
        if profit<500 or float(cols[rating_idx])<6:
            continue

        cols.append(str(profit))

        # 5. add a new column StockStatus
        if int(cols[stock_idx])<5:
            stock_status="Low Stock"
        else:
            stock_status="Available"

        cols.append(stock_status)


        new_lines.append(",".join(cols))


    with open("inventory_final.txt","w") as f:
        for row in new_lines:
            f.write(row+"\n")

process_file()
```

[21]:
```
# sort by profit descending
header=new_lines[0]
data_rows=new_lines[1:]

data_rows.sort(key=lambda x:float(x.split(",")[-2]),reverse=True)
new_lines=[header]+data_rows

# aggregation(total profit per brand)
brand_profit={}
for row in new_lines[1:]:
    cols=row.split(",")
    brand=cols[1]
    profit=float(cols[-2])
    if brand in brand_profit:
        brand_profit[brand]+=profit
    else:
        brand_profit[brand]=profit

# write to a file
with open("brand_profit.txt","w") as f:
    f.write("Brand,TotalProfit\n")
```

```
    for brand,total_profit in brand_profit.items():
        f.write(f"{brand},{round(total_profit,2)}\n")
```

0.0.1 References

https://chatgpt.com/share/697ef276-ee28-8002-bbd4-9e5aeda7054e