

Data Management in Python

Assignment

Background:

The data sets for analysis is insurance sector data in which premiums and claims information is provided for each policy holder for all the regions and zones.

Questions

1. Import Premiums data and name it as Premium.
2. Check number of rows, columns in the data.
3. Display first 10 rows and last 5 rows.
4. Describe (summarize) all variables.
5. Display top 5 and bottom 5 policies in terms of premium amount.
6. Calculate the sum for variable 'Sum_Assured' by 'Region' variable.
7. Create a subset of policies of Asia Standard Plan with Sum_Assured \leq 50,000. Keep variables Policy_No, Zone_name, Plan and Sum_Assured in the subset data.
8. Export the subsetted data into an xlsx file.
9. Import Premiums and Claims data and name it as Premium and Claim respectively.
10. Merge 'Premium' data set with 'Claims' data set.
11. Create a subset of Policy No., Region and Sub plan of the policy holders whose claim status is Yes.
12. Create a subset of data for all the regions in South zone except Mumbai with claim status "No". Keep only Policy No, Zone, Region, Premium and Claim status in the subset and sort the subsetted data by Premium descending.
13. Calculate average Premium amount by sub plan.
14. Derive a new column named 'Premium type' with 3 categories- Low (Premium < 5000), Medium (Premium between 5000 and 10000) and High (Premium > 10000).

Solutions

```
#Q1. Import Premiums data and name it as Premium
##A.
```

```
import pandas as pd
Premium = pd.read_csv("Premiums.csv")
```

```
#Q2. Check number of rows, columns in the data
##A.
```

```
Premium.shape[0]
Premium.shape[1]
```

```
#Q3 Display first 10 rows and last 5 rows
##A.
```

```
Premium.head(10)
Premium.tail(5)
```

```
#Q4 Describe (summarize) all variables
##A.
```

```
Premium.describe()
```

```
#Q5 Display top 5 and bottom 5 policies in terms of premium amount
##A.
```

```
newpremium = Premium.sort_values(by=['Premium'], ascending = [0])
newpremium.head()
newpremium.tail()
```

```
#Q6 Calculate the sum for variable 'Sum_Assured' by 'Region' variable
##A.
```

```
agg = Premium.groupby('REGION')['Sum_Assured'].sum()
agg.head()
```

```
#Q7 Create a subset of policies of Asia Standard Plan with Sum_Assured <= 50,000. Keep variables Policy_No, Zone_name, Plan and Sum_Assured in the subset data.
##A.
```

```
ASP = Premium.loc[(Premium.Plan=="Asia Standard Plan") &
(Premium.Sum_Assured<=50000),['POLICY_NO','ZONE_NAME','Plan','Sum_Assured']]
ASP.head()
```

```
#Q8      Export the data into an excel file
##A.
```

```
ASP.to_excel("file.xlsx",index=False)
```

```
#Q9 Import Premiums and Claims data and name it as Premium and Claim respectively
##A.
```

```
premium = pd.read_csv("Premiums.csv")
claim = pd.read_csv("Claims.csv")

premium.head()
claim.head()
```

```
#Q10      Merge 'Premium' data set with 'Claims' data set.
##A.
```

```
master = pd.merge(premium,claim,how='left')
master.head()
```

```
#Q11 Create a subset of Policy No., Region and Sub plan of the policy holders whose claim status is Yes
##A.
```

```
claimyes = master.loc[(master.Claim_Status=="Yes"),['POLICY_NO','REGION','Sub_Plan']]
claimyes.head()
```

```
#Q12 Create a subset of data for all the regions in South zone except Mumbai with claim status "No".
##A.
```

```
exceptmumbai = master.loc[(master.ZONE_NAME=="South") & (~master['REGION'].isin(["Mumbai I","Mumbai II"])) &
(master.Claim_Status=="No")]
exceptmumbai = exceptmumbai[['POLICY_NO','ZONE_NAME','REGION','Premium','Claim_Status']]
exceptmumbai = exceptmumbai.sort_values('Premium',ascending = [0]).reset_index()
exceptmumbai.head()
```

#Q13 Calculate average Premium amount by sub plan
##A.

```
avgpremium = master.groupby('Sub_Plan')['Premium'].mean()  
avgpremium
```

#Q14 Derive a new column named 'Premium type' with 3 categories- Low (Premium <5000), Medium (Premium between 5000 and 10000) and High (Premium >10000)
##A.

```
master['Premium_type'] = pd.cut(master['Premium'], bins=[0, 5000, 10000, float('Inf')], labels=["Low", "Medium", "High"])  
master.Premium_type
```