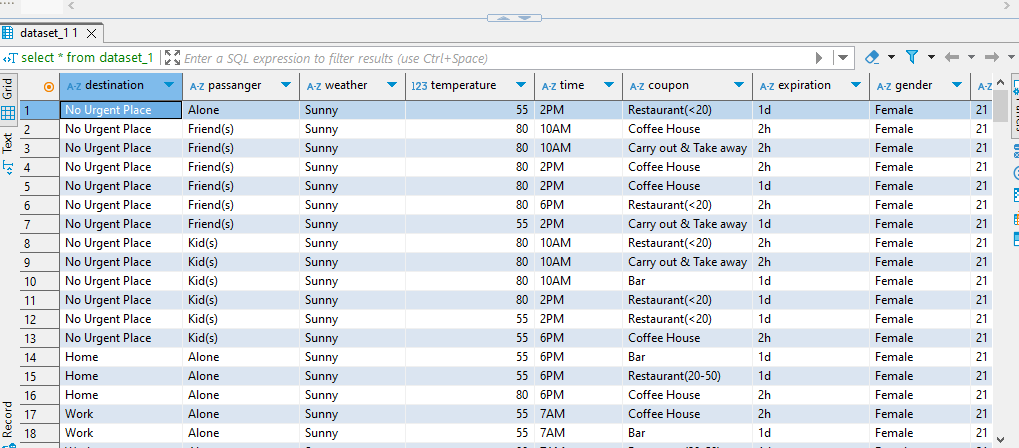
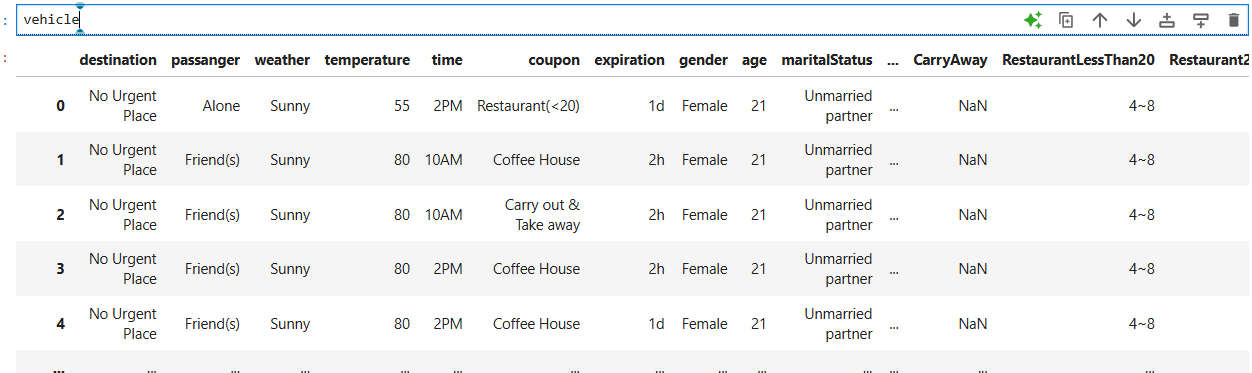
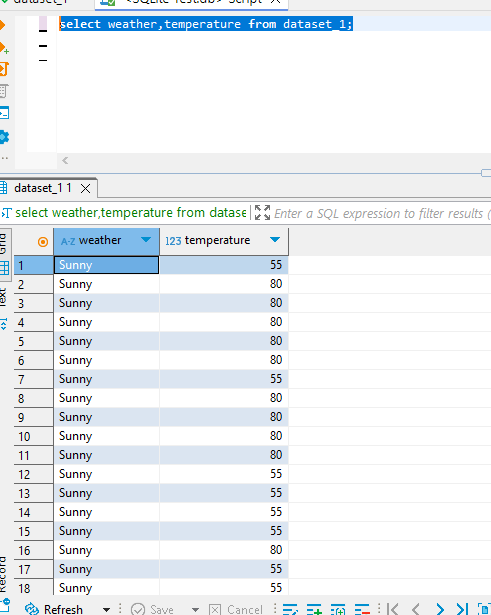
**SQL: select** \* **from** dataset\_1;



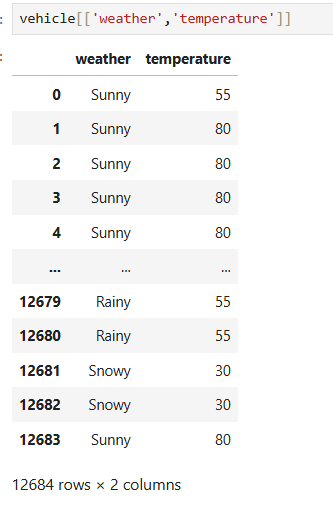
Python: df name is vehicle



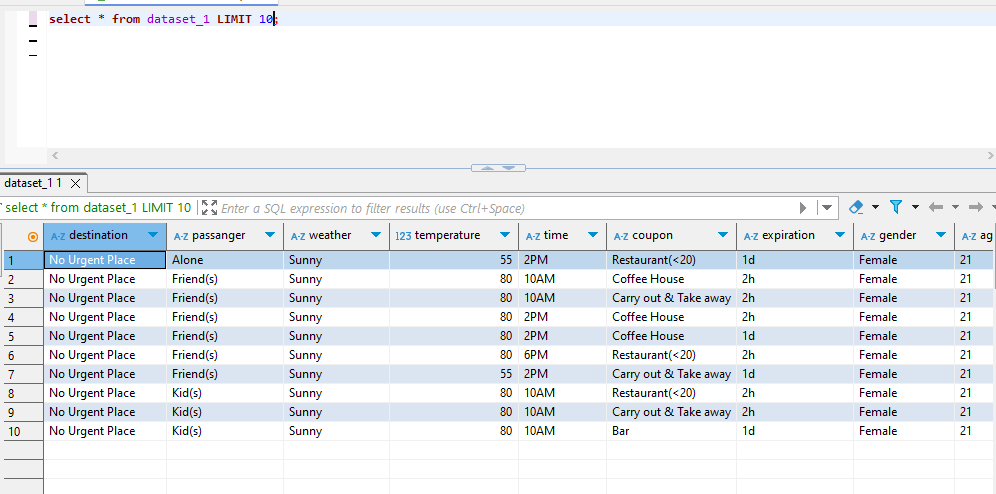
SQL: **select** weather,temperature **from** dataset\_1;



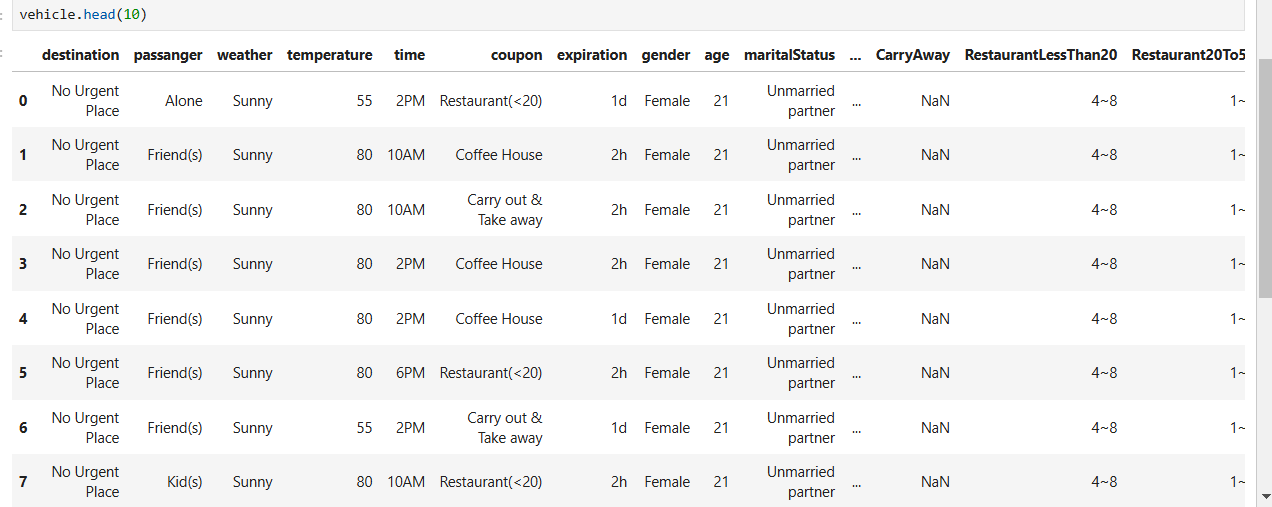
Python: vehicle[['weather','temperature']]



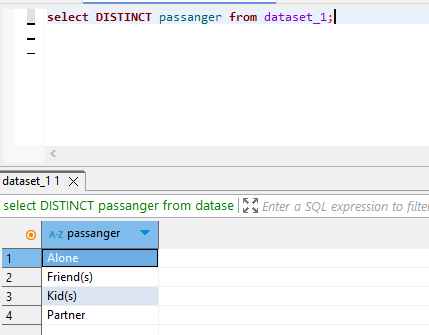
SQL: **select** \* **from** dataset\_1 **LIMIT** 10;



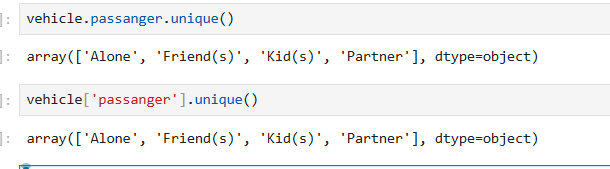
Python: vehicle.head(10)



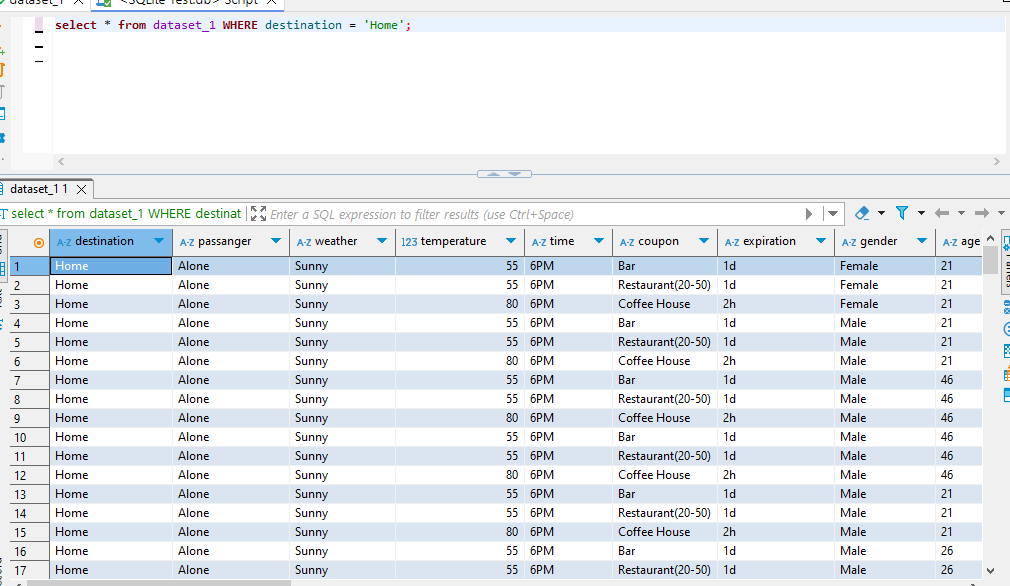
SQL: **select** **DISTINCT** passanger **from** dataset\_1;



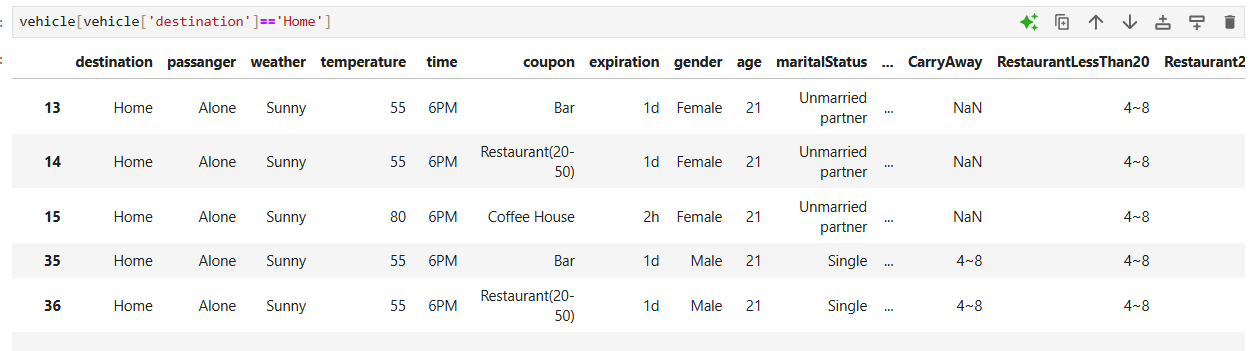
Python: vehicle.paasanger.unique() or vehicle[‘passanger’].unique()



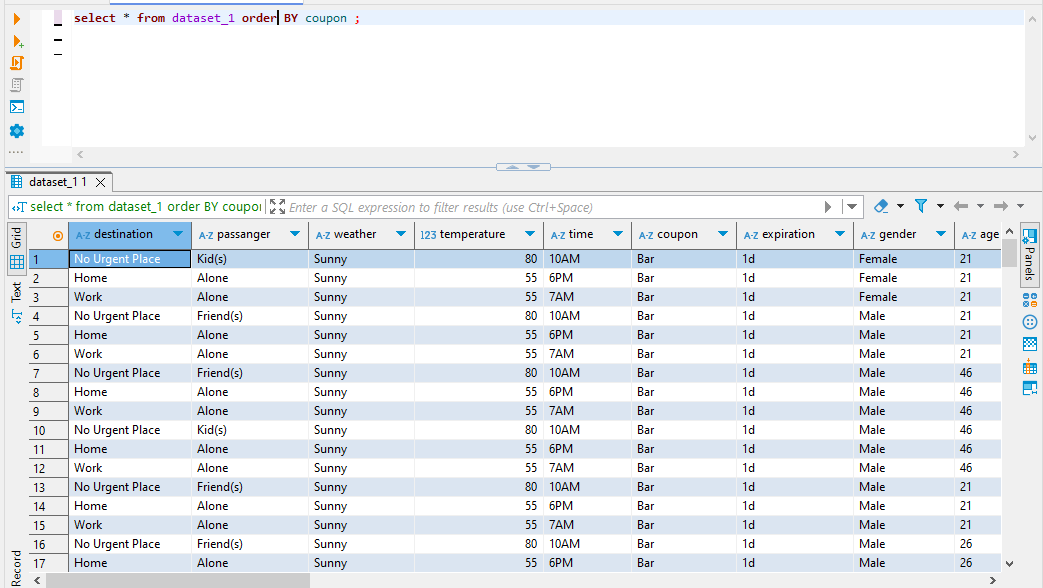
SQL: **select** \* **from** dataset\_1 **WHERE** destination = 'Home';



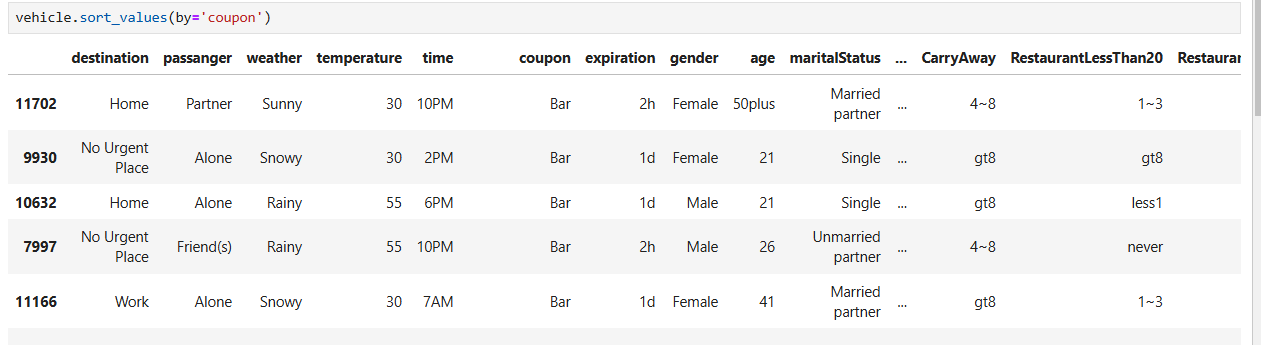
Pythin:vehicle[vehicle[‘destination’]==’Home’] or vehicle[vehicle.destination ==’Home’]



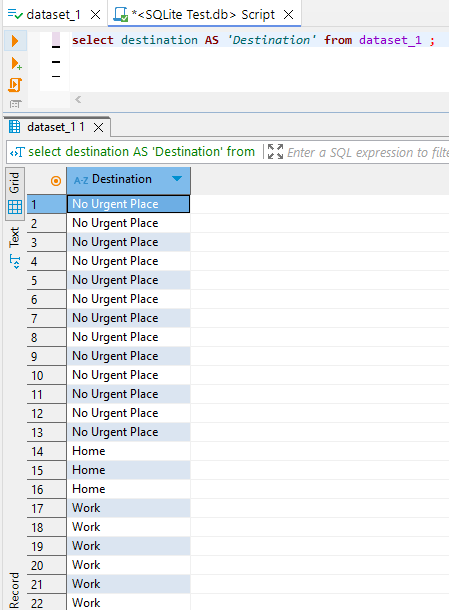
SQL: **select** \* **from** dataset\_1 **order** **BY** coupon ;



Python: vehicle.sort\_values(by=’coupon’)



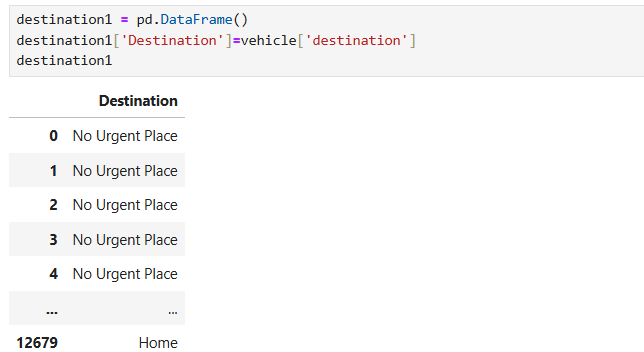
SQL: **select** destination **AS** *'Destination'* **from** dataset\_1 ;



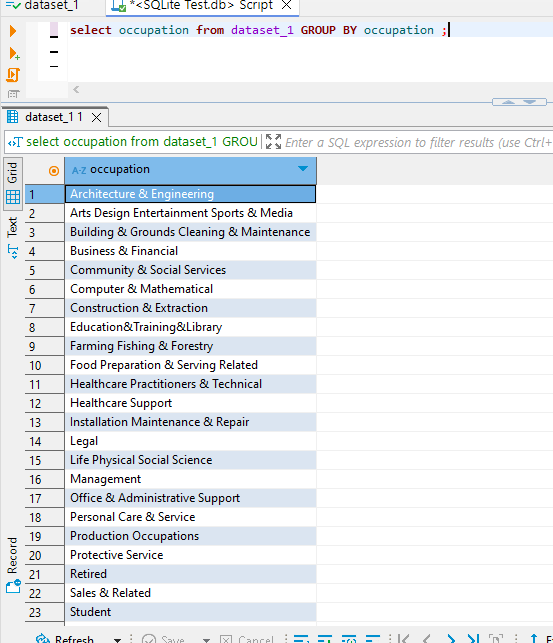
Python: dest1=pd.DataFrame()

Dest1=vehicle[‘destination’]

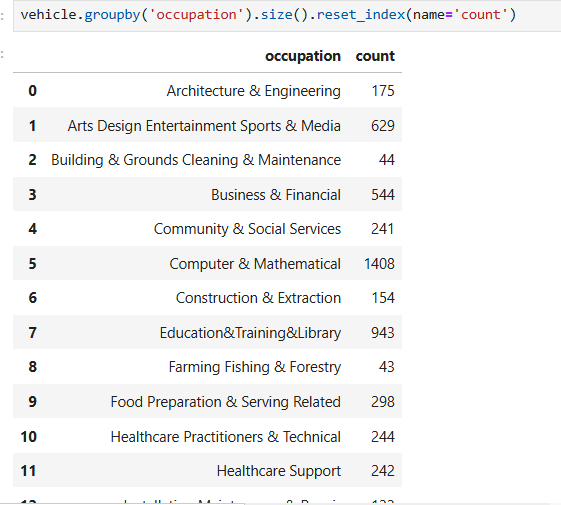
Dest1



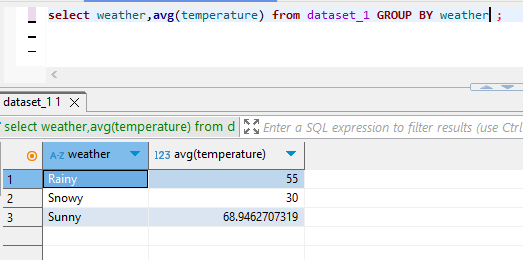
SQL: **select** occupation **from** dataset\_1 **GROUP** **BY** occupation ;



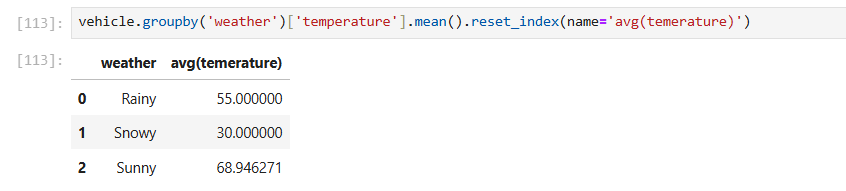
Python: vehicle.groupby(‘occupation’).size().reset\_index(name=’count’)



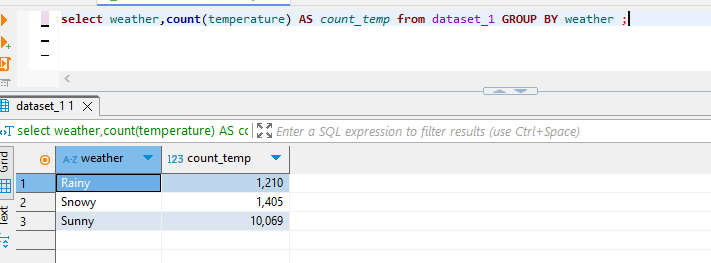
SQL: **select** weather,**avg**(temperature) **from** dataset\_1 **GROUP** **BY** weather ;



Python: vehicle.groupby(‘weather’)[‘temperature’].mean().reset\_index(name=’avg(temperature’))

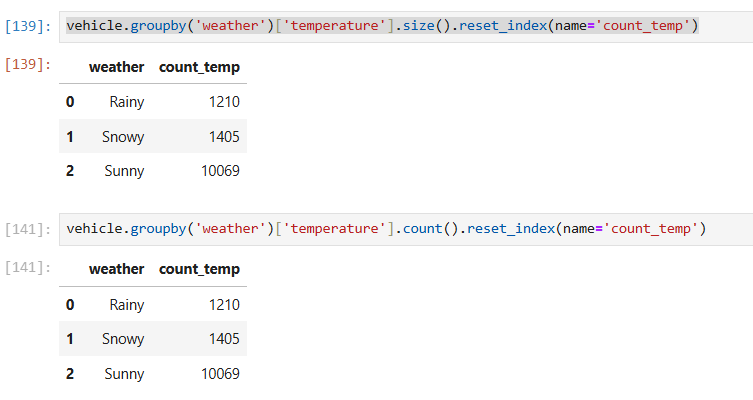


SQL: **select** weather,**count**(temperature) **AS** *count\_temp* **from** dataset\_1 **GROUP** **BY** weather ;

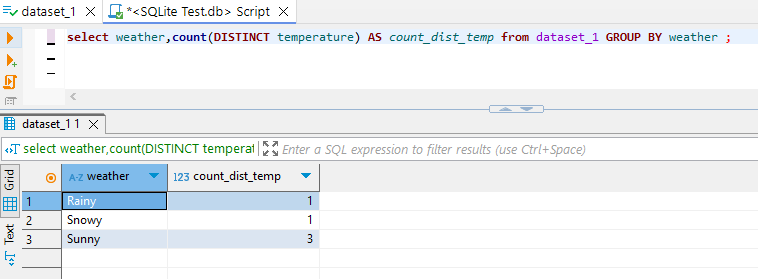


Python: vehicle.groupby(‘weather’)[‘temperature’].count().reset\_index(name=’count\_temp’)

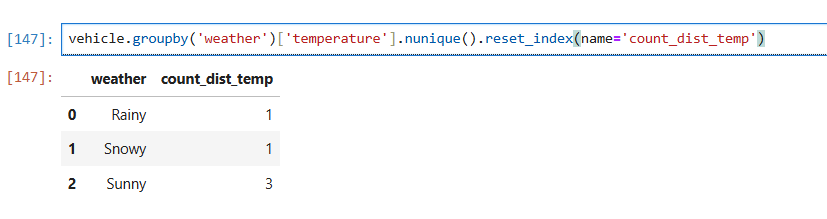
Vehicle.groupby(‘weather’)[‘temperature’].size().reset\_index(name=’count\_temp’)



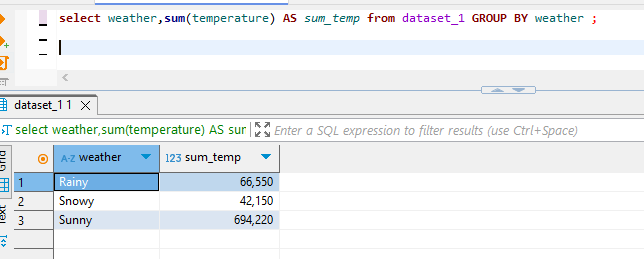
SQL: **select** weather,**count**(**DISTINCT** temperature) **AS** *count\_dist\_temp* **from** dataset\_1 **GROUP** **BY** weather ;



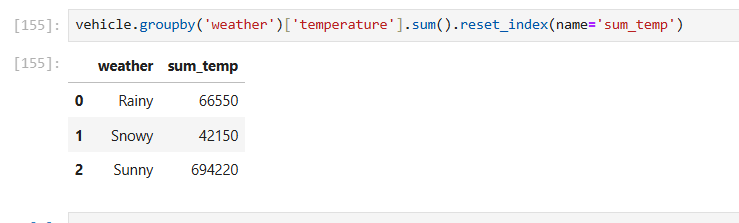
Python: vehicle.groupby(‘weather’)[‘temperature’].nunique().reset\_index(name=’count\_dist\_temp’)



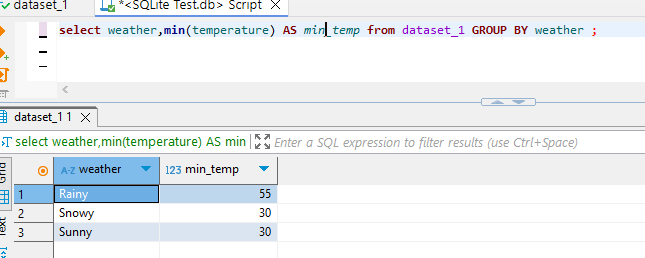
SQL: **select** weather,**sum**(temperature) **AS** *sum\_temp* **from** dataset\_1 **GROUP** **BY** weather ;



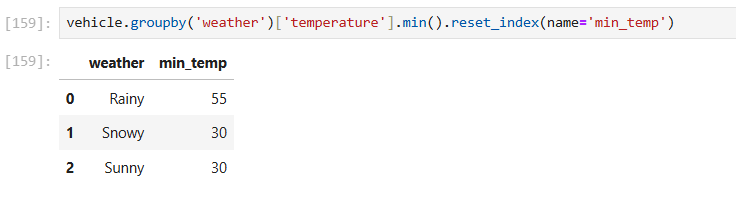
Python: vehicle.groupby(‘weather’)[‘temperature’].sum().reset\_index(name=’sum\_temp’)



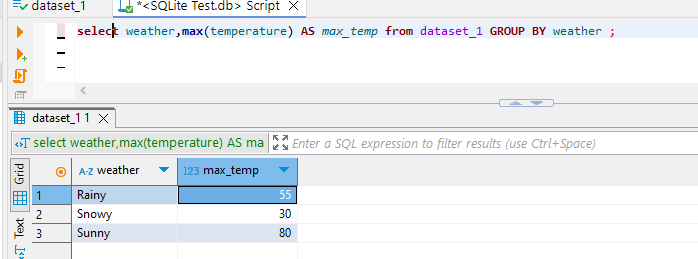
SQL: **select** weather,**min**(temperature) **AS** *min\_temp* **from** dataset\_1 **GROUP** **BY** weather ;



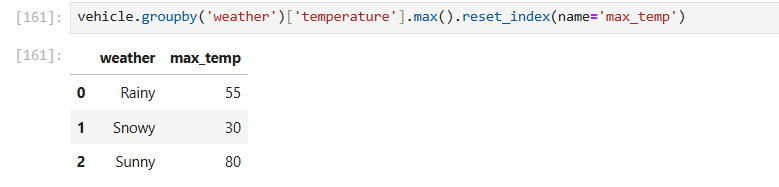
Python: vehicle.groupby(‘weather’)[‘temperature’].min().reset\_index(name=’min\_temp’)



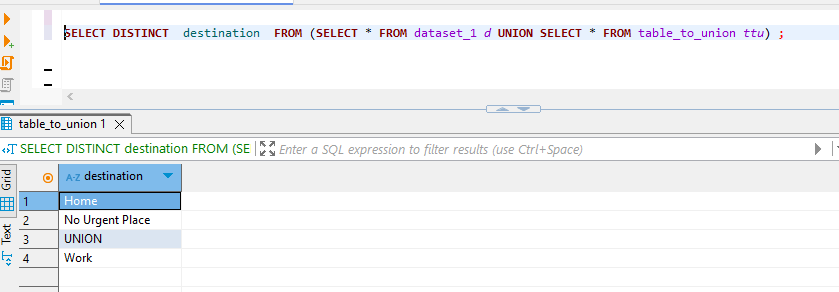
SQL: **select** weather,**max**(temperature) **AS** *max\_temp* **from** dataset\_1 **GROUP** **BY** weather ;



Python: vehicle.groupby(‘weather’)[‘temperature’].max().reset\_index(name=’max\_temp’)



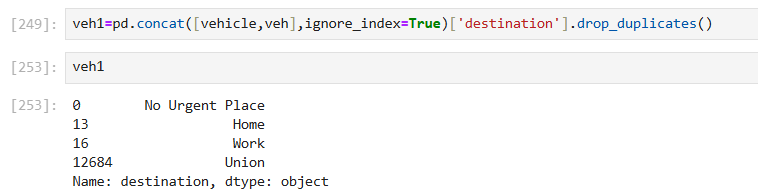
SQL: **SELECT** **DISTINCT** destination **FROM** (**SELECT** \* **FROM** dataset\_1 *d* **UNION** **SELECT** \* **FROM** table\_to\_union *ttu*) ;



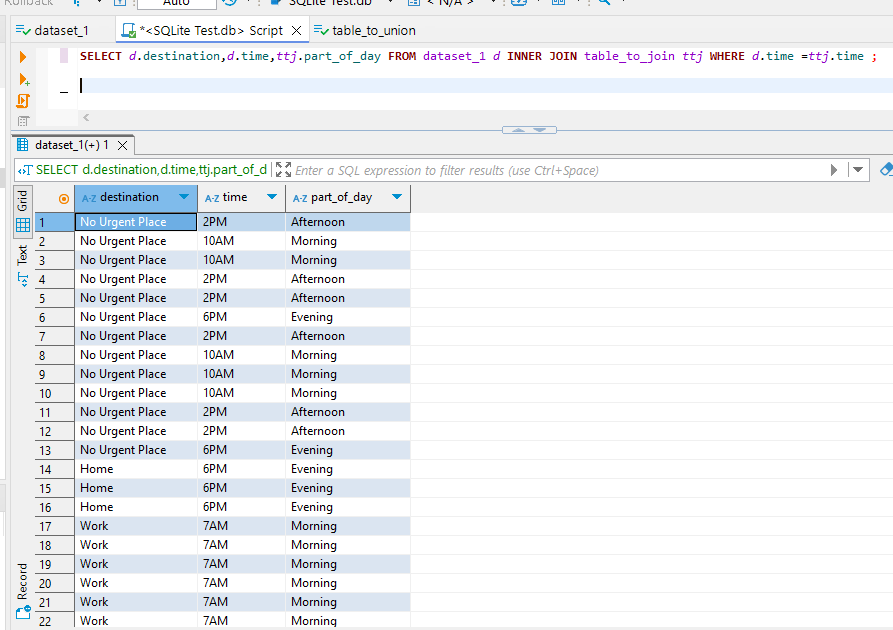
Python:veh1=pd.DataFrame(columns=vehicle.columns)

Veh1=’Union’

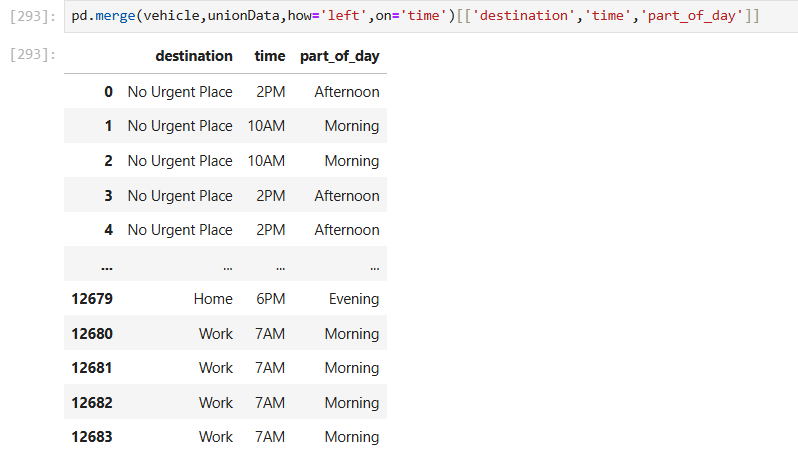
Pd.concat([vehicle,veh1])[‘destination’].drop\_duplicates()



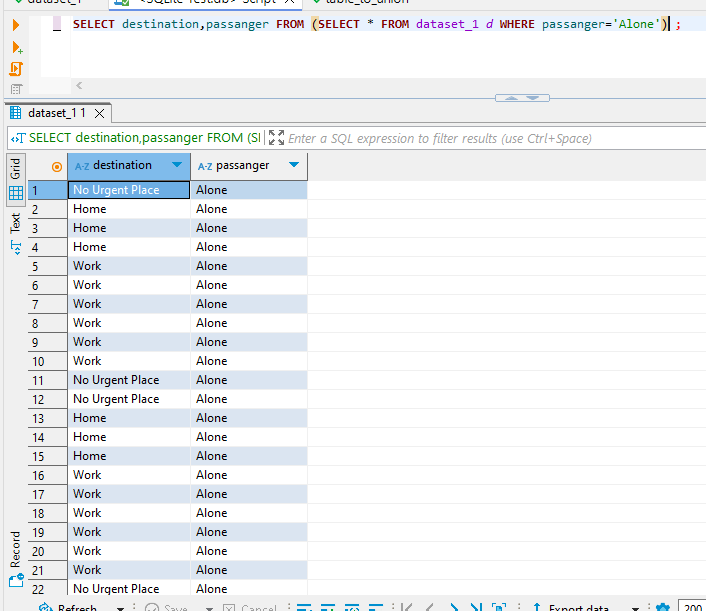
SQL: **SELECT** *d*.destination,*d*.time,*ttj*.part\_of\_day **FROM** dataset\_1 *d* **INNER** **JOIN** table\_to\_join *ttj* **WHERE** *d*.time =*ttj*.time ;



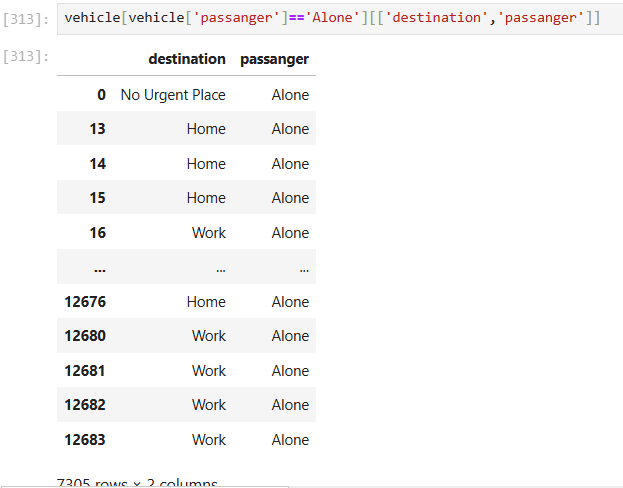
Python:pd.merge(vehicle,unionData,how=’inner’,on=’time’)[[‘destination’,’time’,’part\_of\_day’]]



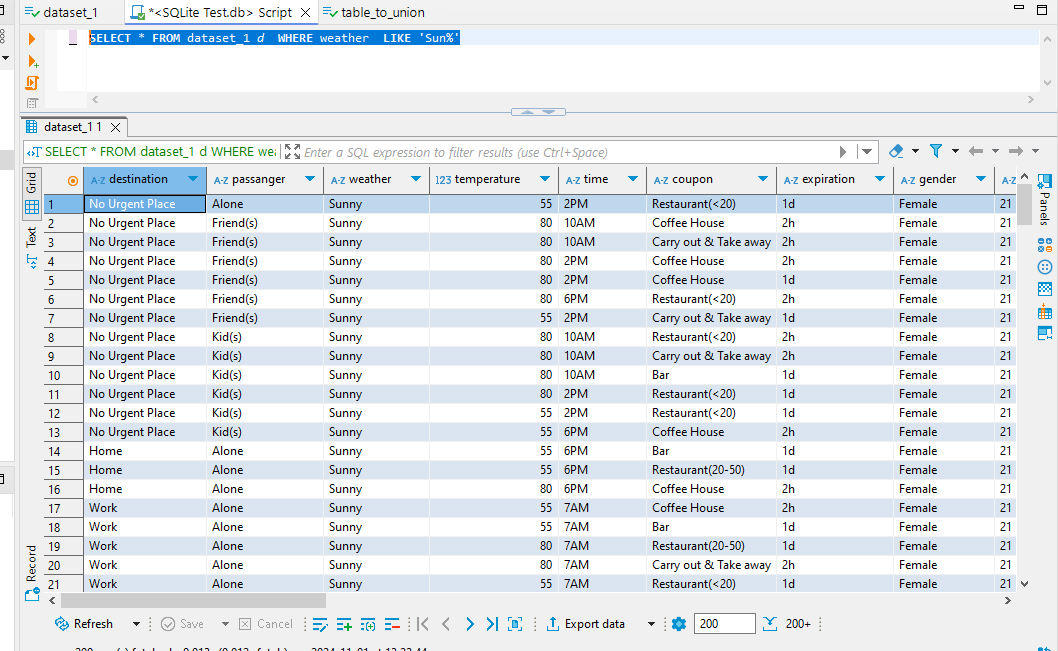
SQL: **SELECT** destination,passanger **FROM** (**SELECT** \* **FROM** dataset\_1 *d* **WHERE** passanger='Alone') ;



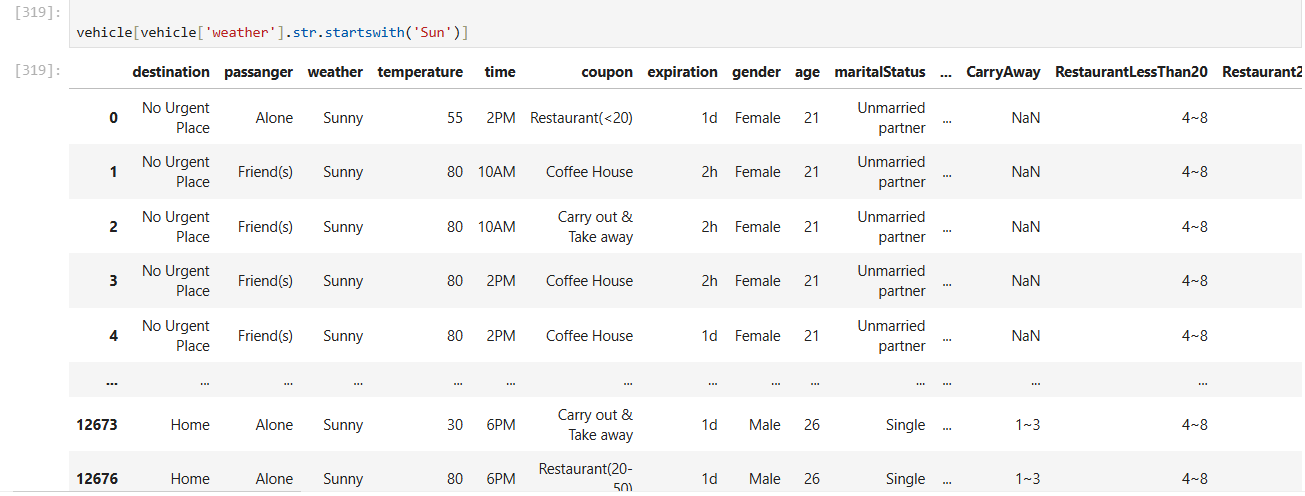
Python: vehicle[vehicle[‘passanger’]==’Alone’][[‘destination’,’passanger’]]



SQL: **SELECT** \* **FROM** dataset\_1 *d* **WHERE** weather **LIKE** 'Sun%'

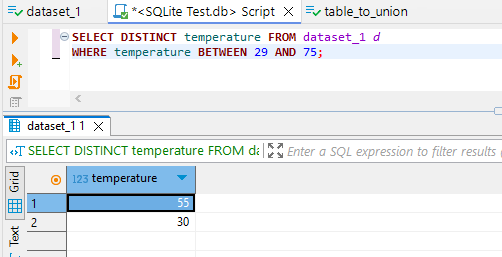


Python: vehicle[vehicle[‘weather’].str.startswith(‘Sun’)]



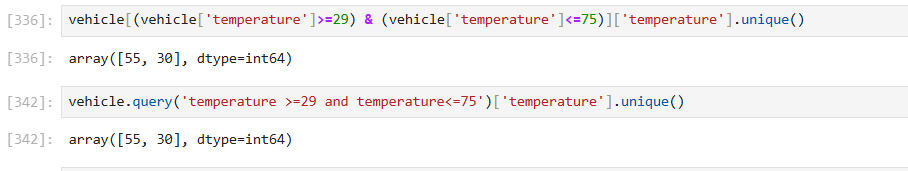
SQL: **SELECT** **DISTINCT** temperature **FROM** dataset\_1 *d*

**WHERE** temperature **BETWEEN** 29 **AND** 75;



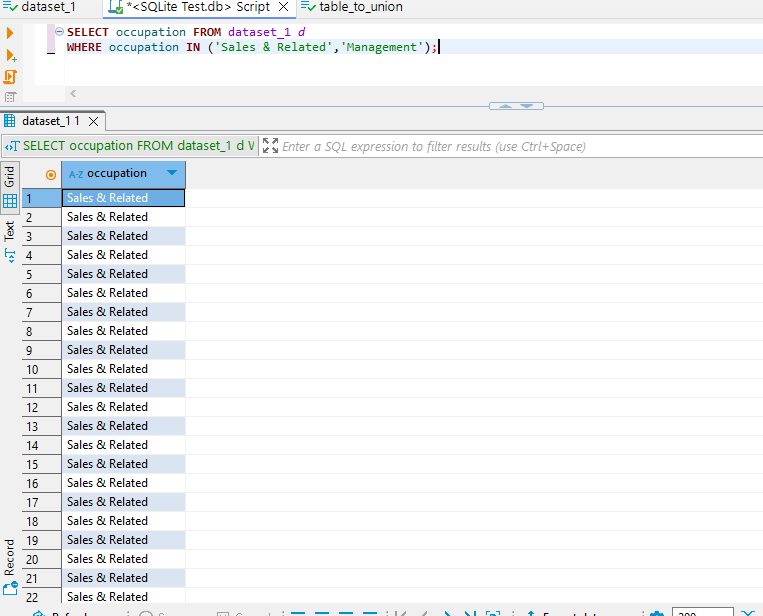
Python: vehicle[(vehicle[‘temperature’]>=29) & (vehicle[‘temperature’]<=75)][‘temperature’].unique()

Vehicle.query(‘temperature >=29 & temperature <=75’)[‘temperature’].unique()



SQL: **SELECT** occupation **FROM** dataset\_1 *d*

**WHERE** occupation **IN** ('Sales & Related','Management');



Python: vehicle.query(‘occupation==\’Sales & Related\’ | occupation==\’Management\’’)

Or

Vehicle[Vehicle[‘occupation’].isin(‘Sales & Related’,’Management’)][‘occupation’]

