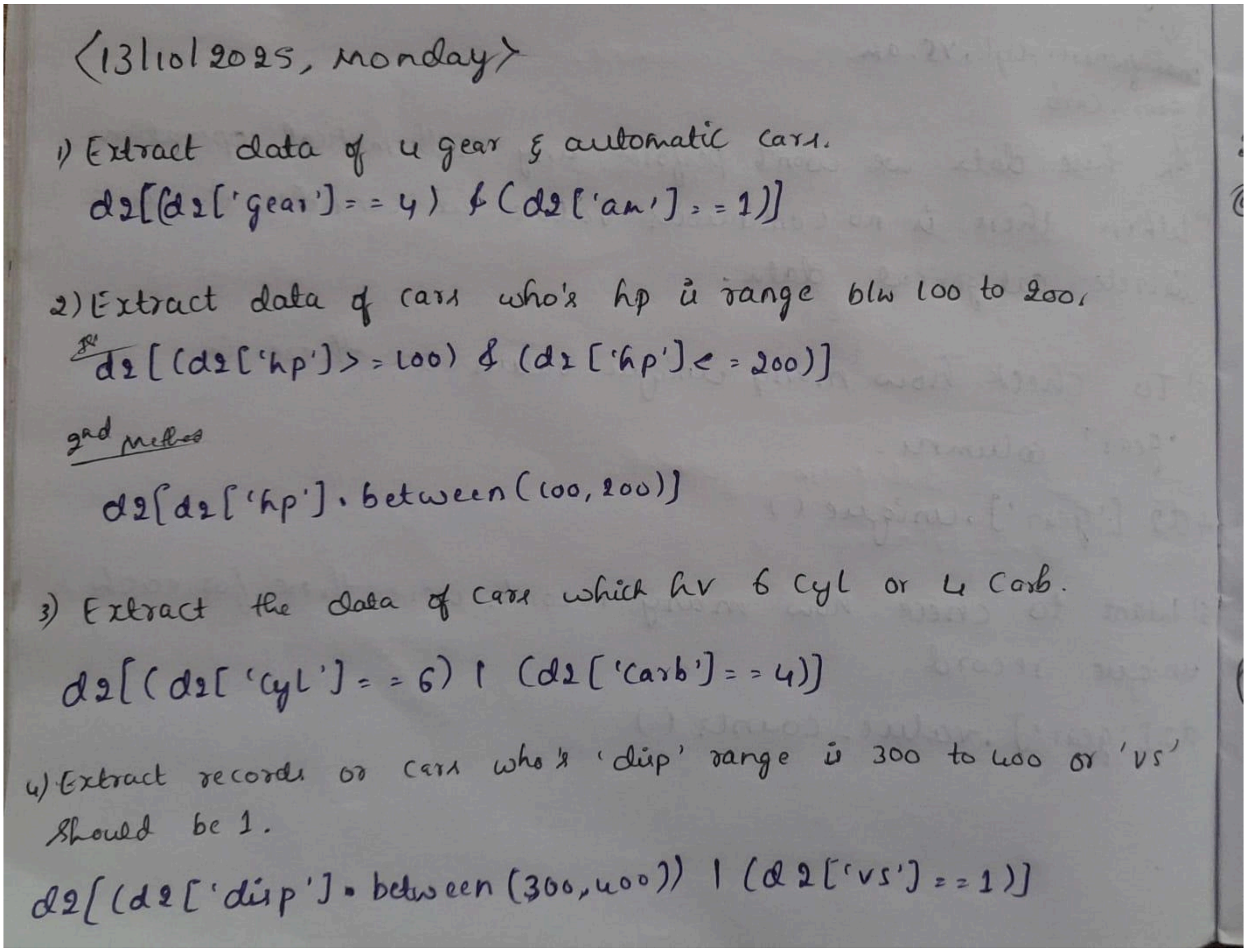

<13|10|2025, monday>

1) Extract data of 4 gear & automatic cars.

```
d2[(d2['gear']==4) & (d2['am']==1)]
```

2) Extract data of cars who's hp is range b/w 100 to 200.

1st
```
d2[(d2['hp']>=100) & (d2['hp']<=200)]
```

2nd method

```
d2[d2['hp'].between(100,200)]
```

3) Extract the data of cars which hv 6 cyl or 4 carb.

```
d2[(d2['cyl']==6) | (d2['carb']==4)]
```

4) Extract records or cars who's 'disp' range is 300 to 400 or 'vs' should be 1.

```
d2[(d2['disp'].between(300,400)) | (d2['vs']==1)]
```

Merge 3 files.

my files are like Sales 1, Sales2, Sales 3 I hv to merge these files in a single file.

final_sale = pd.concat(Sales1, Sales2, Sales 3)

↓

It will throwing through an error like concat () takes 1 positional argument but 3 were given

- To pass these 3 arguments in a single positional argument means we convert it to list. Imp

LIKE → final_sale = pd.concat([sales1, sales2, sales3])

Task

1) Convert objects data types to numerical.

final_sale ['Rate'] = pd.to_numeric (final_sale ['Rate'], errors = 'coerce')

errors= 'coerce' → It will used, or convert non-numeric values into NaN instead of throwing an error.

2) On Insurance data

(a) Each region wise total expenses.

reg_exp = (d1.groupby('Region'))['expenses'].sum()

Here I'm writing region colm in () bracket bcz which colm I have to group that should be written in () bracket.

- Groups your data by each unique region.
- ['expenses'] → focus only on the expenses colm inside those groups.
- sum → Adds up the expenses for each region.

(b) Gender wise avg bmi & expenses.

→ Double bracket for passing multiple columns.

gender_avg = d1.groupby('sex')[['bmi', 'expenses']].mean()

gender_avg