PA3

- Due at 12:30 PM on Apr 8, 2025
- The .ipynb file containing the code should be submitted on Blackboard in the Assignments section
- If there is any ambiguity or contradiction or confusion that you may have while working on the assignment, feel free to ask questions and clarifications in class or on Teams or discuss individually with Danish.
- Please do not post your code or code-snippet on Teams
- This is an individual assignment and focuses on data frames, and data visualization.

PA3

Congratulations. You have been selected to help conduct some exploratory data analysis for an upstart online car review website. They need help reading review data on cars into a python data frame, transform the data and conduct some statistical and visualization tasks. To start, read inventory and the review data provided in cars_pa3.txt and reviews_pa3.txt in to data frames cars_df and reviews_df, respectively.

ADDING COLUMNS

Add the following columns in the data frame reviews_df:

- **year**
  Holds the values of the years extracted from the dates in the column *date*.
- **month**
  Holds the values of the months extracted from the dates in the column *date*.
- **word_count**
  Holds the values of the number of words from the textual comments in the column *comment*. The number of words for each comment is calculated after all the characters except alphabets and blank spaces have been removed from each of the comments. Also ensure that before you calculate the number of words, in each comment, there should be exactly one white space between every pair of neighboring words and there should not be any leading and trailing spaces. If a comment is not available/missing or has 0 characters left after the comments are transformed using the aforementioned technique then the word_count for that comment should be 0. You can utilize pandas **isnull** function to check whether a comment is missing/empty.

After adding these columns, display the data in the data frame on the console. The output should look like that shown on the next slide (don't' worry about the borders or left/right indentation of the columns). In several comments there are leading, training or more than one whitespace between the words as can be observed in the output in the next slide.

PA3: ADDING COLUMNS

serial_no	name	date	rating	comment	year	month	word_count
1	rockstar	1-18-2019	3	It's alright. Gives me some trouble some times.	2019	1	8
2	rockstar	5-11-2019	5	Awesome car. Love it.	2019	5	4
3	suave	6-15-2019	4	Nice car. Have driven it quite a bit with much trouble .	2019	6	11
4	trend	6-21-2019	3	Had issues with the brakes.	2019	6	5
5	maverick	9-15-2019	3	Average car .	2019	9	2
6	trend	12-18-2019	NaN		2019	12	0
7	suave	3-2-2020	2	I do not like it. Don't get it.	2020	3	8
8	starship	5-1-2020	4	Nice car. Would recommend it.	2020	5	5
9	rockstar	6-1-2020	NaN	NaN	2020	6	0
10	spiral	1-6-2021	5	One of the best cards I've ever had!	2021	1	8
11	spiral	1-25-2021	1	NaN	2021	1	0
12	trend	3-2-2021	2	Decent mileage. Looks good but needs regular maintenance	2021	3	8
13	rockstar	3-10-2021	4	Have had it for several years now.	2021	3	7
14	nebula	3-21-2021	5		2021	3	0
15	rockstar	9-20-2021	3	NaN	2021	9	0
16	nebula	9-25-2021	NaN	Decent car. Nothing outstanding though...	2021	9	5
17	trend	9-27-2021	4	Drives fine, decent mileage, looks great...	2021	9	6
18	trend	10-5-2021	5	NaN	2021	10	0
19	nebula	11-15-2021	2	Would not recommend it	2021	11	4
20	nebula	11-25-2021	5	Love this car!!!	2021	11	3
21	spiral	12-5-2021	3	Has had issues... needs lot of maintenance	2021	12	7

DESCRIPTIVE STATISTICS

Display the descriptive statistics of rating and word_count on the console as shown below (using the data from reviews_df).

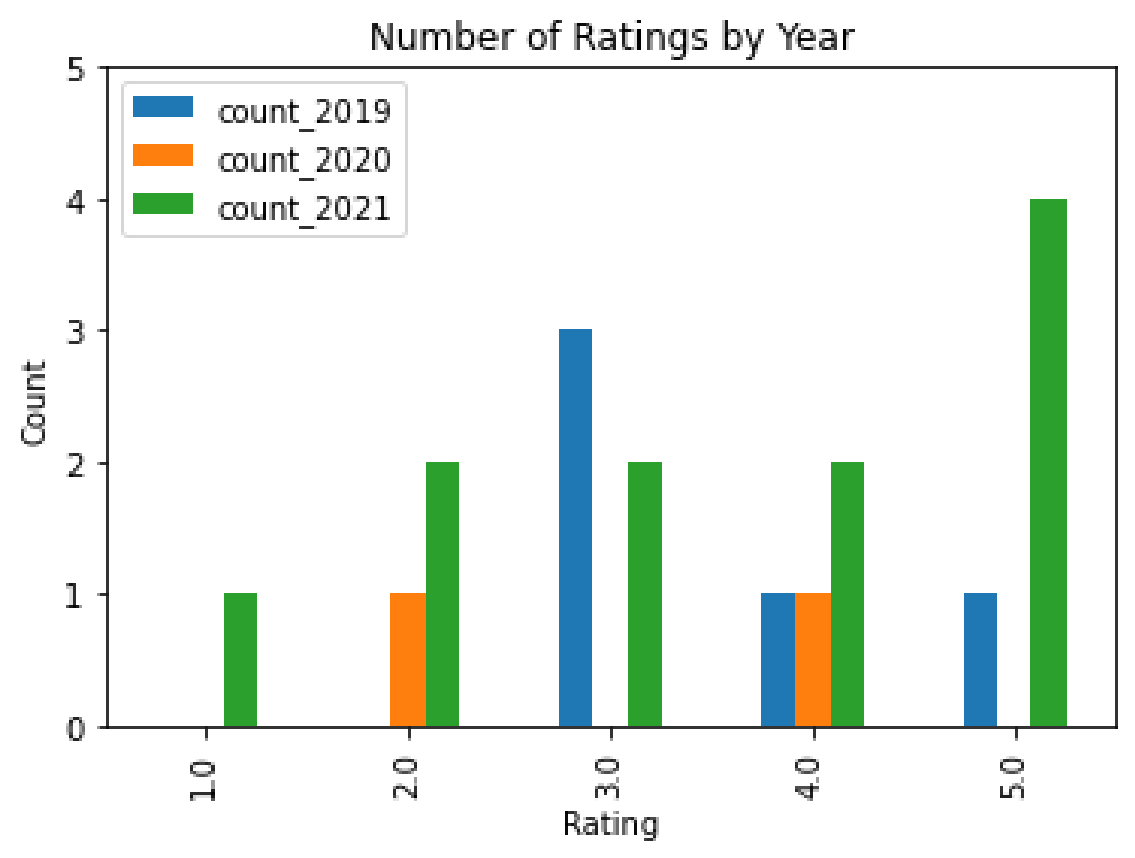
```
Average Rating:
count    18.000000
mean      3.500000
std       1.248529
min       1.000000
25%       3.000000
50%       3.500000
75%       4.750000
max       5.000000
Name: rating, dtype: float64

Average Word Count:
count    21.000000
mean      4.333333
std       3.439961
min       0.000000
25%       0.000000
50%       5.000000
75%       7.000000
max      11.000000
Name: word_count, dtype: float64
```

RATINGS COUNT DISTRIBUTION BY YEAR

Plot the distribution of the rating values' (1, 2, 3 ,4 and 5) count across the years. For this, first use reviews_df to create three data frames, each of which holds the count of the rating values in the years 2019, 2020 and 2021, respectively. Then, merge these three data frames (on outer join) on *rating* column. The resulting dataframe will have the rating values' count across the years as shown below. The index values in the resulting dataframe should be ratings. The data should be sorted in ascending order of the index values of this dataframe. Finally, display this dataframe on the console and create a plot using this dataframe as shown below.

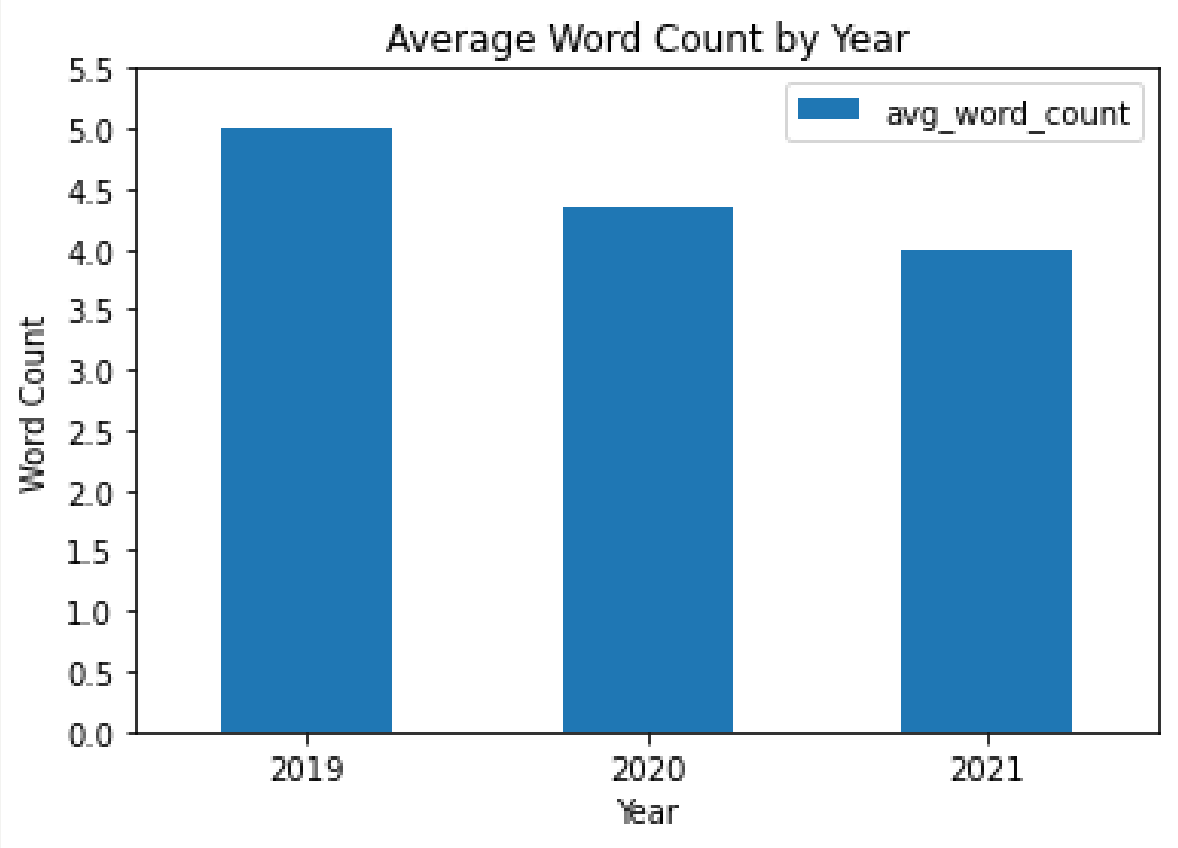
rating	count_2019	count_2020	count_2021
1.0	0	0	1
2.0	0	1	2
3.0	3	0	2
4.0	1	1	2
5.0	1	0	4



AVERAGE WORD COUNT BY YEAR

Plot the distribution of the average word counts of the comments across the years. For this, first use reviews_df to create a data frame which holds the average word counts by year as shown below. The index values in this dataframe are years. The data should be sorted in ascending order of the index values of this dataframe. Finally, display this dataframe on the console and create a plot using this dataframe as shown below.

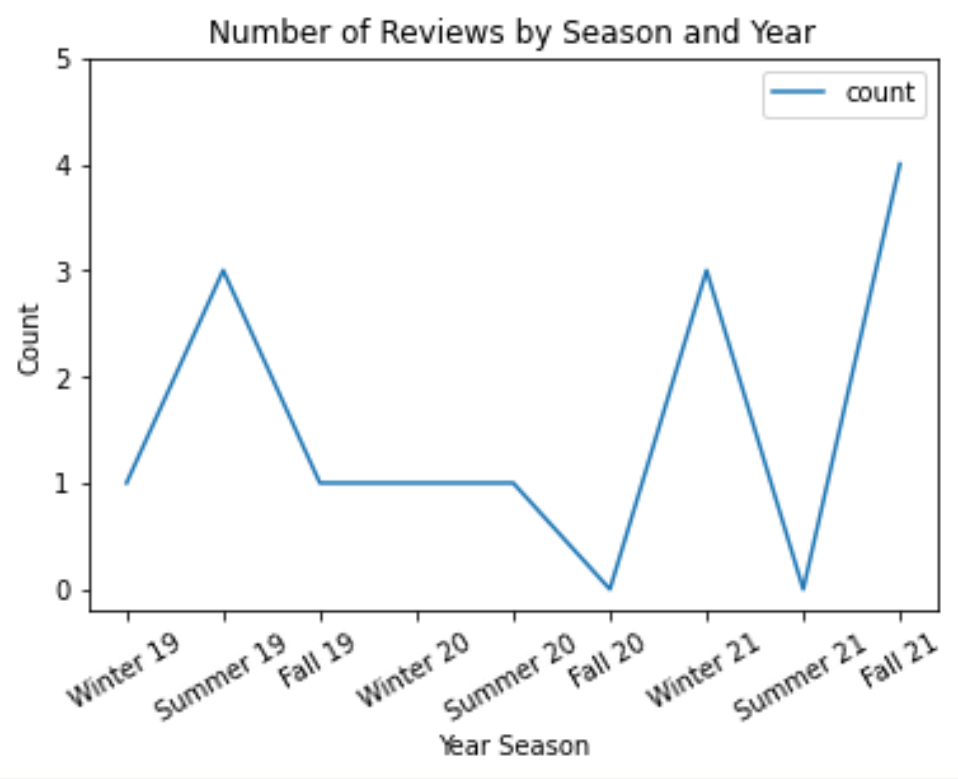
year	avg_word_count
2019	5.000000
2020	4.333333
2021	4.000000



RATINGS COUNT DISTRIBUTION BY SEASONS AND YEARS

Plot the distribution of count of reviews across the seasons and years. For this, first drop any record in reviews_df if it has a missing value in any of the two columns, rating or comment, or if the word_count is 0. Next, use this dataframe to create a dataframe which holds the count of reviews in each season of the years 2019, 2020 and 2021 as shown below. The index values in this dataframe should be seasonname_year values which are a combination of season name and year (only 2 digits). The winter, summer and fall seasons should comprise of the first 4, middle 4 and last 4 months of the year, respectively. The data should be sorted in ascending order of the year_season column values of this dataframe. Finally, display this dataframe on the console and create a plot (X axis values rotated 30 degrees) using this dataframe as shown below.

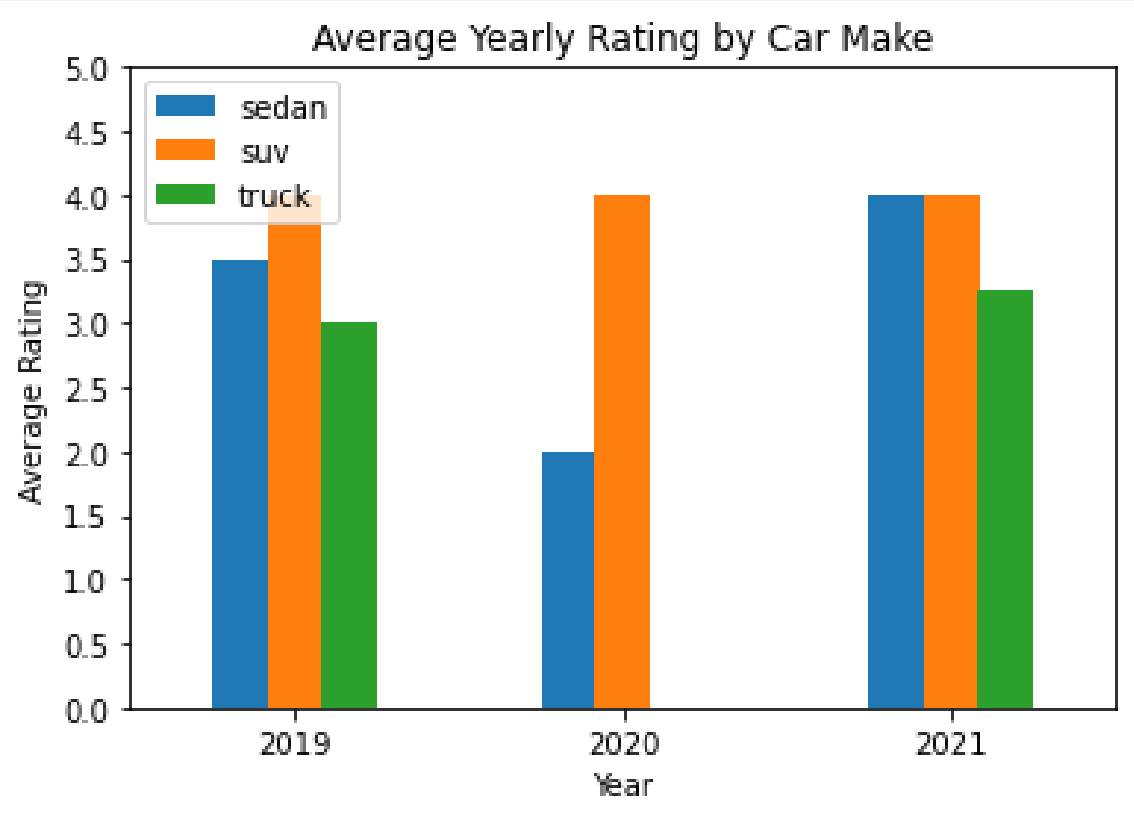
seasonname_year	year_season	count
Winter 19	20191	1
Summer 19	20192	3
Fall 19	20193	1
Winter 20	20201	1
Summer 20	20202	1
Fall 20	20203	0
Winter 21	20211	3
Summer 21	20212	0
Fall 21	20213	4



AVERAGE YEARLY RATINGS DISTRIBUTION BY CAR MAKE

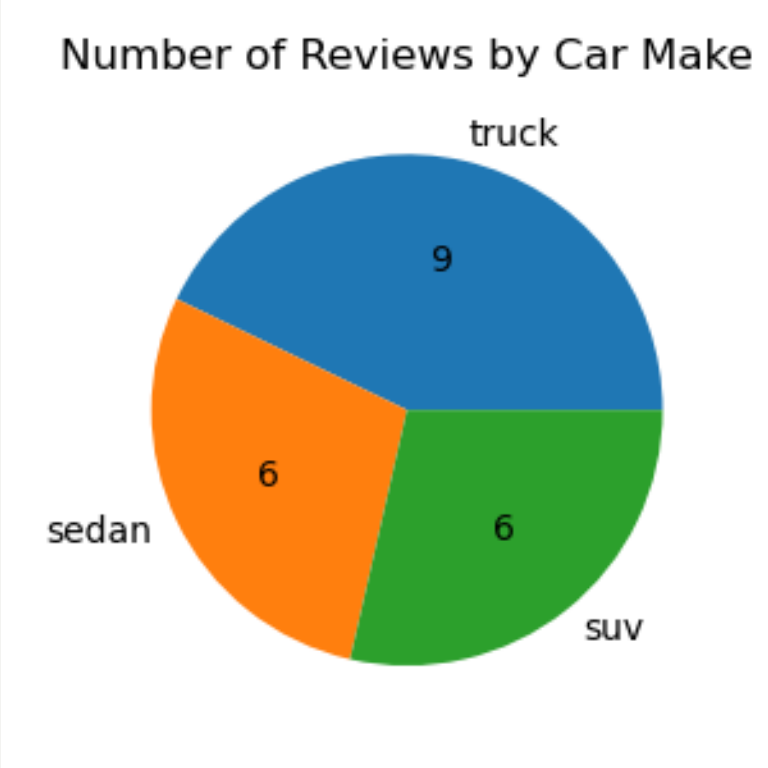
Plot the distribution of the average car-make (sedan, suv and truck) ratings (1 to 5) across the years. For this, first merge cars_df and reviews_df (on outer join). In this dataframe with merged data, drop any record if it has a missing value in any of the columns or if the comment has no characters. Next, create a data frame that holds the average ratings grouped by car make and year. Next, create a pivot dataframe that will have the average car-make rating across the years as shown below. The index values in this dataframe should be years. The data should be sorted in ascending order of the index values of this dataframe. Finally, display this dataframe on the console and create a plot using this dataframe as shown below.

make	sedan	suv	truck
year			
2019	3.5	4.0	3.00
2020	2.0	4.0	NaN
2021	4.0	4.0	3.25



NUMBER OF REVIEWS BY CAR MAKE

Plot the distribution of the number of reviews by car-make (sedan, suv and truck). For this, first merge cars_df and reviews_df (on outer join). Using this dataframe with merged data, create a dataframe/series number of reviews by car make. Finally, create a pie chart/plot using this dataframe/series as shown below.

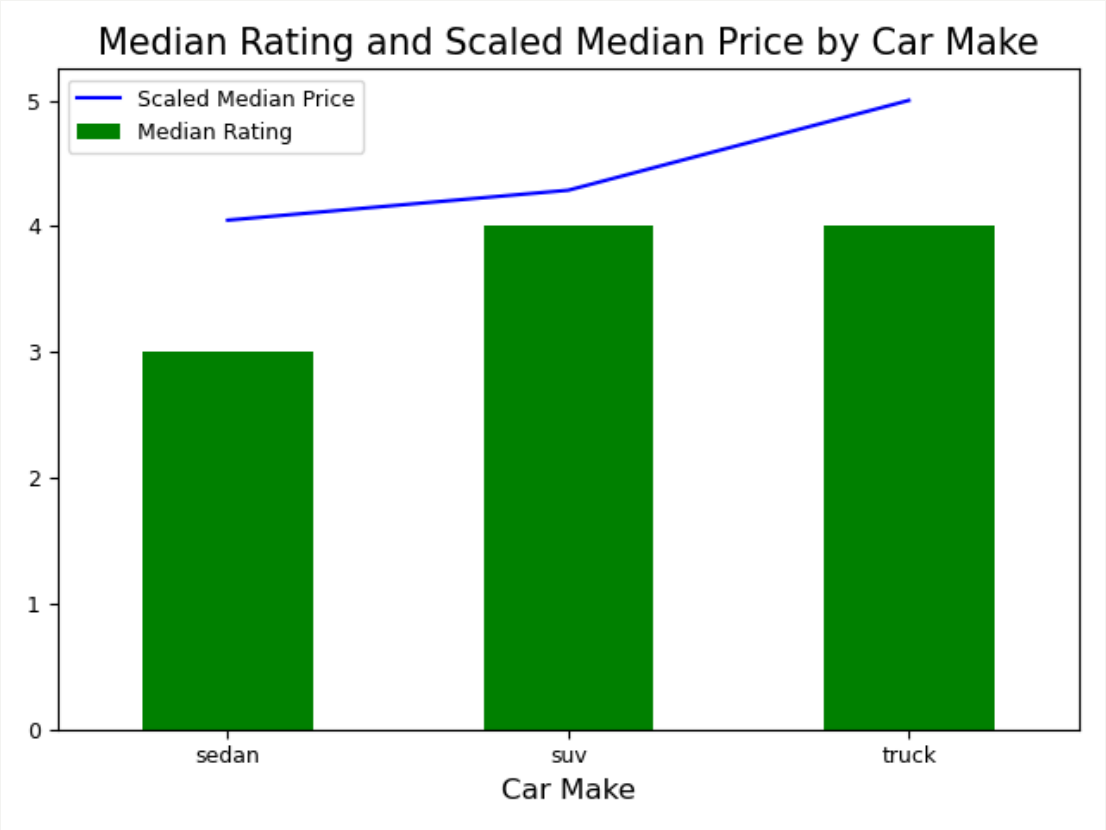


MEDIAN RATING AND SCALED MEDIAN PRICE BY CAR MAKE

Plot the distribution of the median ratings (1 to 5) and scaled median prices by car-makes (sedan, suv and truck). For this, first merge cars_df and reviews_df (on outer join). Using this dataframe with merged data, create a dataframe/series that holds the median ratings by car make and a dataframe/series that holds the median prices by car make. Next, scale the median prices by dividing the prices with the highest prices and multiplying this ratio with 5. Finally, display these dataframes/series on the console and create a plot with these dataframes/series as shown below.

```
make
sedan  3.0
suv    4.0
truck  4.0
Name: rating, dtype: float64

make
sedan  4.047649
suv    4.285714
truck  5.000000
Name: price, dtype: float64
```



IMPORTANT NOTE

- The format, values and content of your the console outputs and plots should match the format, values and content of the console outputs and content shown in the corresponding slides (don’t worry about the spacing in the console outputs).

- Values in any of the dataframes/series or plots should not be hard-coded. They also should not be modified, delated or added individually or collectively using hard-coded values in a list, dictionary or any other Python data structure.

- The code for the construction of dataframes/series and/or columns for each of the deliverables should follow the instructions given on the corresponding slides. For those parts of the deliverables for which the instructions are not given on the slides, you will need to utilize an appropriate logic to derive the expected console outputs and plots.

PA3 GRADING (300 POINTS TOTAL)

Deliverable	Points
Adding the year, month and word_count columns	20
Descriptive Statistics of Rating, Word Count	5
Ratings Count Distribution by Year	35
Average Word Count by Year	25
Ratings Count Distribution by Season and Year	75
Average Yearly Ratings Distribution by Car Make	45
Number of Reviews by Car Make	35
Median Rating and Scaled Median Price by Car Make	45
Efficient Program Structure e.g. Code breakdown in appropriate methods, comments and markdowns	15