Dear students,

Below given are the Assignment #1 submission guidelines of each team:
Due Date: 04:00PM of 06.09.2024

*If necessary wherever required add more fields

Title of the Dataset:	**California Housing Prices**
Name of Student – 1	Vaishnavi Chelagola
Roll No of Student – 1	160122771073
Name of Student – 2	Pravalika Eppalapally
Roll No of Student - 2	160122771075
Name of Student – 3	Mukku Neha Prabha
Roll No of Student - 3	160122771082
Description of Dataset:	
Size of the dataset	Code: dim(data) Output: 20640 10 No. of rows: 20,640 No. of columns: 10
URL of the dataset	https://www.kaggle.com/datasets/camnugent/california-housing-prices
Summary / Description and Domain of each attribute	This dataset is commonly used to build predictive models for estimating housing prices based on various features such as location, population, and housing characteristics. It is also a good dataset for demonstrating data preprocessing techniques, such as handling missing values, scaling, and encoding categorical features. Attributes: 1. longitude: Continuous values (~ -124.35 to -114.31) indicating how far west. 2. latitude: Continuous values (~ 32.54 to 41.95) indicating how far north. 3. housingMedianAge: Integer values (1 to 52+) for house age within a block. 4. totalRooms: Integer values representing total rooms within a block. 5. totalBedrooms: Integer values for total bedrooms within a block. 6. population: Integer values representing total residents within a block. 7. households: Integer values for total households within a block. 8. medianIncome: Continuous values (~ 0.5 to 15) in tens of thousands of USD. 9. medianHouseValue: Continuous values

	(~ $14,999 to $500,001+). 10. oceanProximity: Categorical values like "<1H OCEAN," "INLAND," "NEAR OCEAN," "NEAR BAY," "ISLAND."

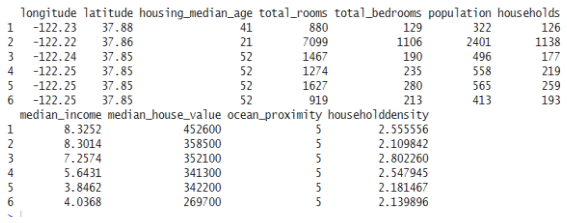
No. of missing values in each attribute	Code: colSums(is.na(data))
	Output:
	`> colSums(is.na(data4))`
	`longitude latitude housing_median_age total_rooms total_bedrooms`
	`0 0 0 0 207`
	`population households median_income median_house_value ocean_proximity`
	`0 0 0 0 0`

Please mention if you have created the dataset or added some more fields to the dataset. Give detailed description of your contribution with resources from which the data is crated	To find density of each household, we have divided population with no. of households and added it as a new column "householddensity" using "dplyr" library
	Code:
	install.packages("dplyr")
	library(dplyr)
	data<- data %>%
	mutate(householddensity=population/households)
	head(data)
	Output:
	`> head(data4)`
	`longitude latitude housing_median_age total_rooms total_bedrooms population households`
	`1 -122.23 37.88 41 880 129 322 126`
	`2 -122.22 37.86 21 7099 1106 2401 1138`
	`3 -122.24 37.85 52 1467 190 496 177`
	`4 -122.25 37.85 52 1274 235 558 219`
	`5 -122.25 37.85 52 1627 280 565 259`
	`6 -122.25 37.85 52 919 213 413 193`
	`median_income median_house_value ocean_proximity householddensity`
	`1 8.3252 452600 NEAR BAY 2.555556`
	`2 8.3014 358500 NEAR BAY 2.109842`
	`3 7.2574 352100 NEAR BAY 2.802260`
	`4 5.6431 341300 NEAR BAY 2.547945`
	`5 3.8462 342200 NEAR BAY 2.181467`
	`6 4.0368 269700 NEAR BAY 2.139896`

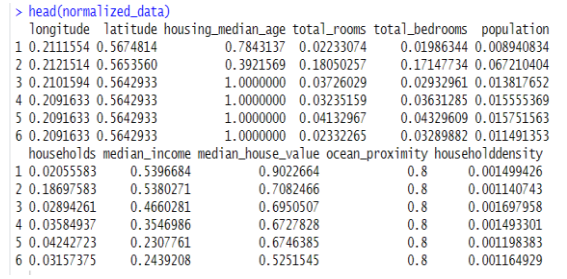
Description of Task#1	**Dealing with null values**
Required Pre-processing	Checking and removing null valuesthat are there in the dataset (#total_bedrooms is the only column with null values.We replaced them with median and hence the dataset all no null values after preprocessing.)
Attributes involved	total_bedrooms
R-Code with necessary comments	median_value = median(data$total_bedrooms, na.rm = TRUE) data$total_bedrooms[is.na(data$total_bedrooms)] = median_value colSums(is.na(data))

Output	`> median_value = median(data4$total_bedrooms, na.rm = TRUE)`
	`> data4$total_bedrooms[is.na(data4$total_bedrooms)] = median_value`
	`> colSums(is.na(data4))`
	`longitude latitude housing_median_age total_rooms total_bedrooms`
	`0 0 0 0 0`
	`population households median_income median_house_value ocean_proximity`
	`0 0 0 0 0`
	`householddensity`
	`0`

Description of Task#2	Encoding Categorical Data
Required Pre-processing	Attribute ocean_proximity has categorical data "<1H OCEAN," "INLAND," "NEAR OCEAN," "NEAR BAY," "ISLAND." We have encoded them to numericals.

Attributes involved	ocean_proximity
R-Code with necessary comments	data_encoded = data%>% mutate(ocean_proximity =as.numeric(factor(ocean_proximity))) head(data_encoded)
Output	 longitude latitude housing_median_age total_rooms total_bedrooms population households 1 -122.23 37.88 41 880 129 322 126 2 -122.22 37.86 21 7099 1106 2401 1138 3 -122.24 37.85 52 1467 190 496 177 4 -122.25 37.85 52 1274 235 558 219 5 -122.25 37.85 52 1627 280 565 259 6 -122.25 37.85 52 919 213 413 193 median_income median_house_value ocean_proximity householddensity 1 8.3252 452600 5 2.555556 2 8.3014 358500 5 2.109842 3 7.2574 352100 5 2.802260 4 5.6431 341300 5 2.547945 5 3.8462 342200 5 2.181467 6 4.0368 269700 5 2.139896
Description of Task#3	Standardization
Required Pre-processing	Normalising numerical variables
Attributes involved	All the attributes in dataset are numeric with categorical data being encoded. Hence all the attributes are involved in Standardization.

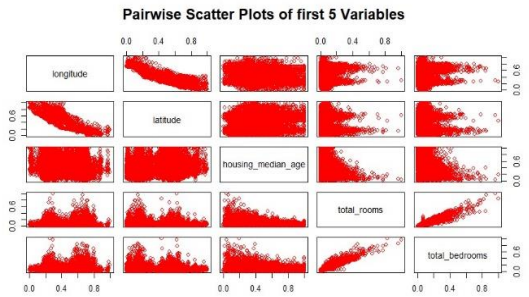
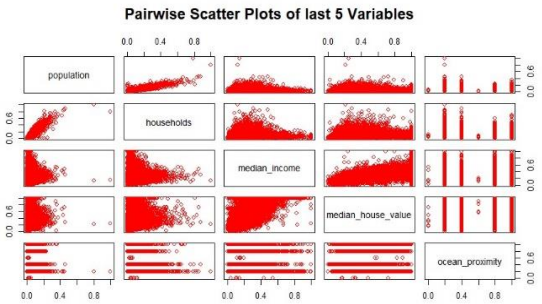
R-Code with necessary comments	**1. Min-Max Scaling**
	# Function to normalize a vector
	normalize <- function(x) {
	return ((x - min(x)) / (max(x) - min(x)))
	}
	
	# Apply the normalize function to the entire dataset
	normalized_data <-
	as.data.frame(lapply(data[sapply(data,
	is.numeric)], normalize))
	
	# View the normalized data
	head(normalized_data)

Output	 > head(normalized_data) longitude latitude housing_median_age total_rooms total_bedrooms population 1 0.2111554 0.5674814 0.7843137 0.02233074 0.01986344 0.008940834 2 0.2121514 0.5653560 0.3921569 0.18050257 0.17147734 0.067210404 3 0.2101594 0.5642933 1.0000000 0.03726029 0.02932961 0.013817652 4 0.2091633 0.5642933 1.0000000 0.03235159 0.03631285 0.015555369 5 0.2091633 0.5642933 1.0000000 0.04132967 0.04329609 0.015751563 6 0.2091633 0.5642933 1.0000000 0.02332265 0.03289882 0.011491353 households median_income median_house_value ocean_proximity householddensity 1 0.02055583 0.5396684 0.9022664 0.8 0.001499426 2 0.18697583 0.5380271 0.7082466 0.8 0.001140743 3 0.02894261 0.4660281 0.6950507 0.8 0.001697958 4 0.03584937 0.3546986 0.6727828 0.8 0.001493301 5 0.04242723 0.2307761 0.6746385 0.8 0.001198383 6 0.03157375 0.2439208 0.5251545 0.8 0.001164929

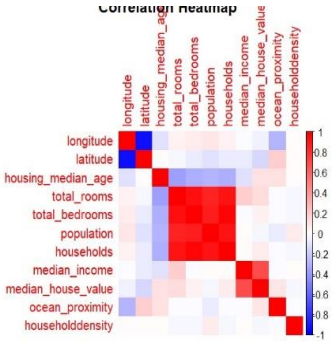
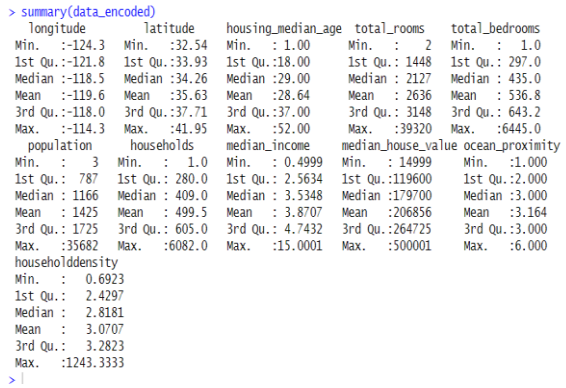
	2. Z-Score Scaling
	#standardizing using scale() method
	standardized_data <-
	as.data.frame(scale(data[sapply(data,
	is.numeric)]))
	
	# View the standardized data
	head(standardized_data)

Output	

```
> head(standardized_data)
  longitude latitude housing_median_age total_rooms total_bedrooms population households
1 -1.327803 1.052523          0.9821189  -0.8047996     -0.9724529 -0.9744050 -0.9770092
2 -1.322812 1.043159         -0.6070042   2.0458405      1.3571106  0.8614180  1.6699206
3 -1.332794 1.038478          1.8561366  -0.5357329     -0.8270042 -0.8207575 -0.8436165
4 -1.337785 1.038478          1.8561366  -0.6241995     -0.7197060 -0.7660095 -0.7337637
5 -1.337785 1.038478          1.8561366  -0.4623928     -0.6124078 -0.7598283 -0.6291419
6 -1.337785 1.038478          1.8561366  -0.7869229     -0.7721629 -0.8940491 -0.8017678
  median_income median_house_value ocean_proximity householddensity
1    2.34470896          2.1295799        1.291869      -0.04959533
2    2.33218146          1.3141243        1.291869      -0.09250999
3    1.78265622          1.2586629        1.291869      -0.02584190
4    0.93294491          1.1650718        1.291869      -0.05032808
5   -0.01288068          1.1728711        1.291869      -0.08561369
6    0.08744452          0.5445977        1.291869      -0.08961625
>
```

Description of Task#4	Plotting
Required Pre-processing	
Attributes involved	
R-Code with necessary comments	**1. Pair plot** # To see relationships between multiple variables, pairs() function creates scatter plots for each pair of variables #fort first 5 variables pairs(normalized_data[, 1:5], main = "Pairwise Scatter Plots of First 5 Variables", col = "red") #for remaining 5 variables pairs(normalized_data[, 6:10], main = "Pairwise Scatter Plots of First 5 Variables", col = "red")
Output	 

	2. Heat map
	#visualize the correlations between variables using a correlation matrix and a heatmap
	# Install corrplot if you don't have it
	install.packages("corrplot")
	# Load the library
	library(corrplot)
	# Calculate the correlation matrix
	cor_matrix <- cor(normalized_data)
	# Plot the correlation matrix using corrplot
	corrplot(cor_matrix, method = "color",
	col = colorRampPalette(c("blue", "white", "red"))(100),
	title = "Correlation Heatmap",
	cl.pos = "r", # Add colorbar on the right
	addgrid.col = NA) # Remove grid lines between tiles

Output	
Description of Task#5	Summarizing
Required Pre-processing	
Attributes involved	All attributes
R-Code with necessary comments	summary(data)
Output	 > summary(data_encoded) longitude latitude housing_median_age total_rooms total_bedrooms Min. :-124.3 Min. :32.54 Min. : 1.00 Min. : 2 Min. : 1.0 1st Qu.:-121.8 1st Qu.:33.93 1st Qu.:18.00 1st Qu.: 1448 1st Qu.: 297.0 Median :-118.5 Median :34.26 Median :29.00 Median : 2127 Median : 435.0 Mean :-119.6 Mean :35.63 Mean :28.64 Mean : 2636 Mean : 536.8 3rd Qu.:-118.0 3rd Qu.:37.71 3rd Qu.:37.00 3rd Qu.: 3148 3rd Qu.: 643.2 Max. :-114.3 Max. :41.95 Max. :52.00 Max. :39320 Max. :6445.0 population households median_income median_house_value ocean_proximity Min. : 3 Min. : 1.0 Min. : 0.4999 Min. : 14999 Min. :1.000 1st Qu.: 787 1st Qu.: 280.0 1st Qu.: 2.5634 1st Qu.:119600 1st Qu.:2.000 Median : 1166 Median : 409.0 Median : 3.5348 Median :179700 Median :3.000 Mean : 1425 Mean : 499.5 Mean : 3.8707 Mean :206856 Mean :3.164 3rd Qu.: 1725 3rd Qu.: 605.0 3rd Qu.: 4.7432 3rd Qu.:264725 3rd Qu.:3.000 Max. :35682 Max. :6082.0 Max. :15.0001 Max. :500001 Max. :6.000 householddensity Min. : 0.6923 1st Qu.: 2.4297 Median : 2.8181 Mean : 3.0707 3rd Qu.: 3.2823 Max. :1243.3333 >	
Description of Task#6	Converting the scaled data back to original dataframe
Required Pre-processing	
Attributes involved	All attributes
R-Code with necessary comments	max_val=max(normalized_data) min_val=min(normalized_data) # Reverse normalization

	original_data <- normalized_data * (max_val - min_val) + min_val
	
	# Convert to DataFrame
	df <- data.frame(original_data)
	head(df)

Output	> head(df) longitude latitude housing_median_age total_rooms total_bedrooms population 1 0.2111554 0.5674814 0.7843137 0.02233074 0.01986344 0.008940834 2 0.2121514 0.5653560 0.3921569 0.18050257 0.17147734 0.067210404 3 0.2101594 0.5642933 1.0000000 0.03726029 0.02932961 0.013817652 4 0.2091633 0.5642933 1.0000000 0.03235159 0.03631285 0.015555369 5 0.2091633 0.5642933 1.0000000 0.04132967 0.04329609 0.015751563 6 0.2091633 0.5642933 1.0000000 0.02332265 0.03289882 0.011491353 households median_income median_house_value ocean_proximity householddensity 1 0.02055583 0.5396684 0.9022664 0.8 0.001499426 2 0.18697583 0.5380271 0.7082466 0.8 0.001140743 3 0.02894261 0.4660281 0.6950507 0.8 0.001697958 4 0.03584937 0.3546986 0.6727828 0.8 0.001493301 5 0.04242723 0.2307761 0.6746385 0.8 0.001198383 6 0.03157375 0.2439208 0.5251545 0.8 0.001164929

NOTE: Each team consists of 3 members. Continuation to this will be done in the 2nd assignment.