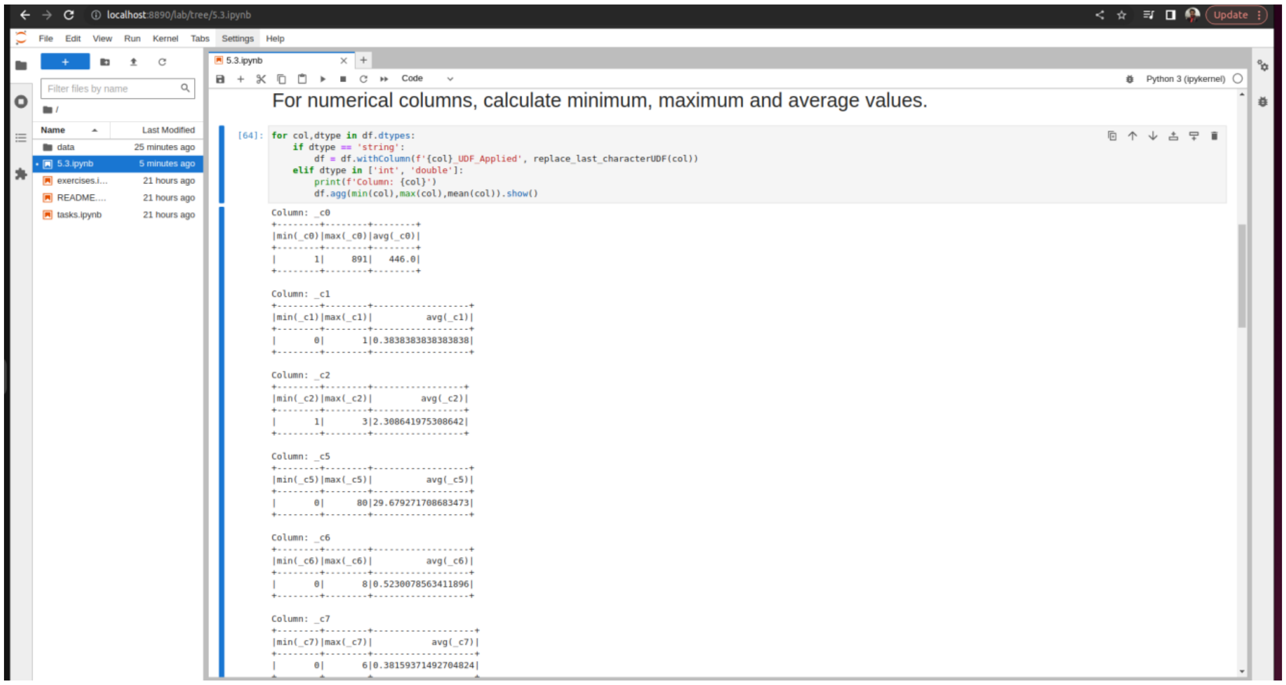
Talha Khan (2303.009.KHI.DEG)

Muhammad Moiz Khan (2303.022.KHI.DEG)

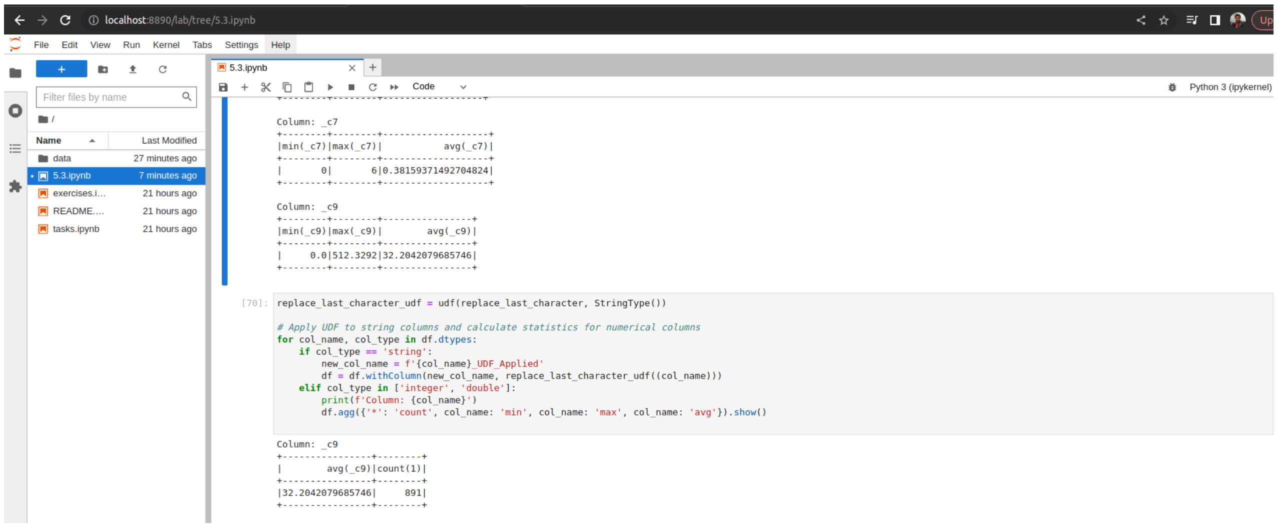
ASSIGNMNET 5.3



For numerical columns, calculate minimum, maximum and average values.

```
[64]: for col,dtype in df.dtypes:
          if dtype == 'string':
              df = df.withColumn(f'{col}_UDF_Applied', replace_last_characterUDF(col))
          elif dtype in ['int', 'double']:
              print(f'Column: {col}')
              df.agg(min(col),max(col),mean(col)).show()
```

```
Column: _c0
+--------+--------+--------+
|min(_c0)|max(_c0)|avg(_c0)|
+--------+--------+--------+
|       1|     891|   446.0|
+--------+--------+--------+

Column: _c1
+--------+--------+------------------+
|min(_c1)|max(_c1)|          avg(_c1)|
+--------+--------+------------------+
|       0|       1|0.3838383838383838|
+--------+--------+------------------+

Column: _c2
+--------+--------+-----------------+
|min(_c2)|max(_c2)|         avg(_c2)|
+--------+--------+-----------------+
|       1|       3|2.308641975308642|
+--------+--------+-----------------+

Column: _c5
+--------+--------+------------------+
|min(_c5)|max(_c5)|          avg(_c5)|
+--------+--------+------------------+
|       0|      80|29.679271708683473|
+--------+--------+------------------+

Column: _c6
+--------+--------+------------------+
|min(_c6)|max(_c6)|          avg(_c6)|
+--------+--------+------------------+
|       0|       8|0.5230078563411896|
+--------+--------+------------------+

Column: _c7
+--------+--------+-------------------+
|min(_c7)|max(_c7)|           avg(_c7)|
+--------+--------+-------------------+
|       0|       6|0.38159371492704824|
```



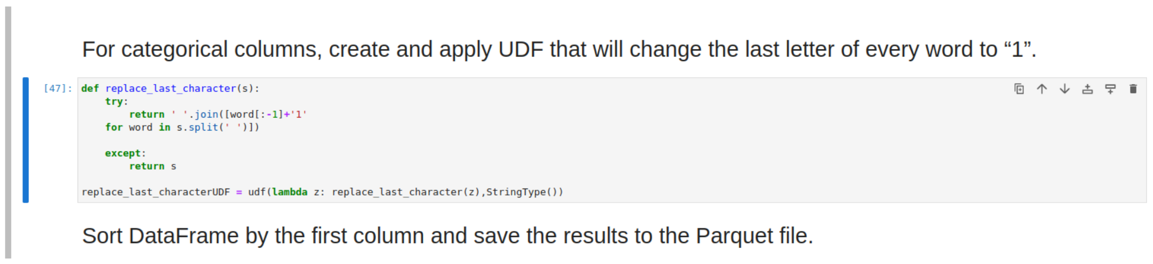
```
Column: _c7
+--------+--------+-------------------+
|min(_c7)|max(_c7)|           avg(_c7)|
+--------+--------+-------------------+
|       0|       6|0.38159371492704824|
+--------+--------+-------------------+

Column: _c9
+--------+--------+-----------------+
|min(_c9)|max(_c9)|         avg(_c9)|
+--------+--------+-----------------+
|     0.0|512.3292|32.2042079685746|
+--------+--------+-----------------+
```

```
[70]: replace_last_character_udf = udf(replace_last_character, StringType())

      # Apply UDF to string columns and calculate statistics for numerical columns
      for col_name, col_type in df.dtypes:
          if col_type == 'string':
              new_col_name = f'{col_name}_UDF_Applied'
              df = df.withColumn(new_col_name, replace_last_character_udf((col_name)))
          elif col_type in ['integer', 'double']:
              print(f'Column: {col_name}')
              df.agg({'*': 'count', col_name: 'min', col_name: 'max', col_name: 'avg'}).show()
```

```
Column: _c9
+----------------+--------+
|        avg(_c9)|count(1)|
+----------------+--------+
|32.2042079685746|     891|
+----------------+--------+
```

- We define a UDF called replace_last_character to replace the last character of a string column. We also create the UDF using udf and specify the return type as StringType().

- Next, we iterate over the columns of the DataFrame using df.dtypes. If the column type is 'string', we apply the UDF to that column using withColumn and create a new column with the suffix "_UDF_Applied".
- For numerical columns, we calculate statistics by using the agg function. The example code demonstrates how to print the minimum, maximum, and average values for each numerical column.

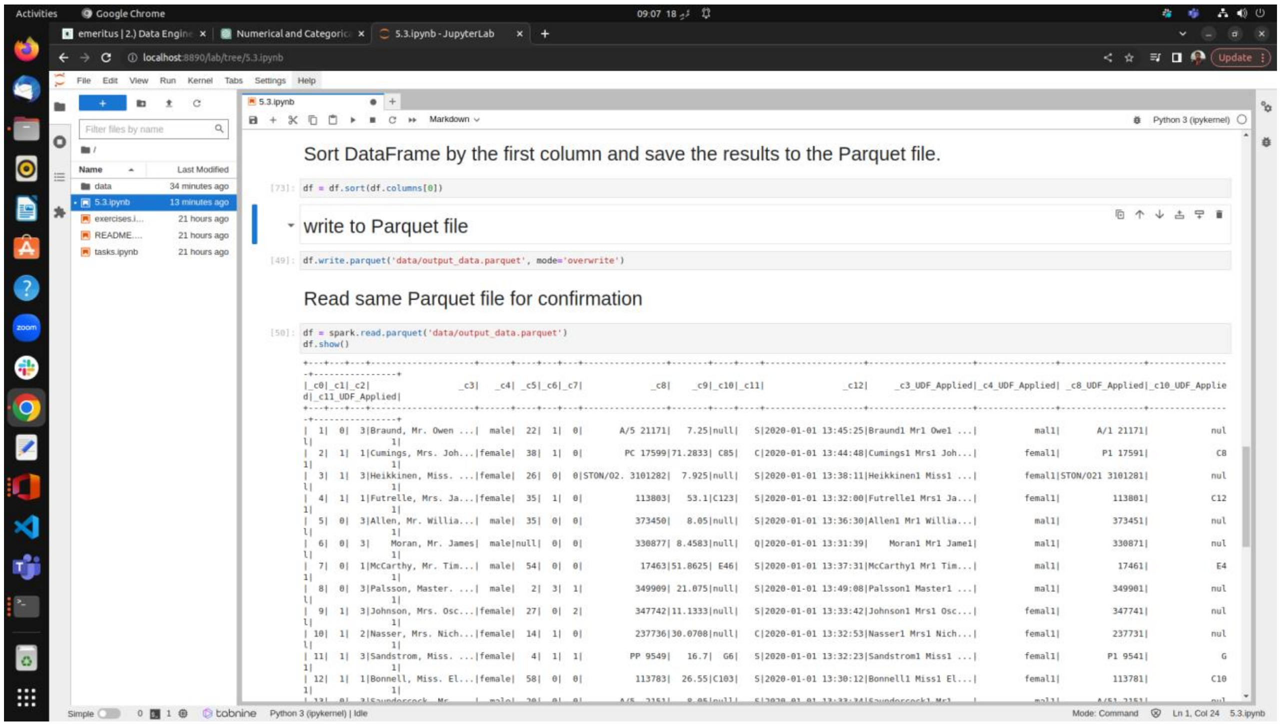


For categorical columns, create and apply UDF that will change the last letter of every word to “1”.

```
[47]: def replace_last_character(s):
          try:
              return ' '.join([word[:-1]+'1'
          for word in s.split(' ')])

          except:
              return s

      replace_last_characterUDF = udf(lambda z: replace_last_character(z),StringType())
```

Sort DataFrame by the first column and save the results to the Parquet file.

- This code defines the replace_last_character function that takes a string s as input. It splits the string into words, removes the last character from each word, and appends '1'. It then joins the modified words back into a string.
- The replace_last_character_udf is created using a lambda function that calls replace_last_character and wraps it with the udf function. It specifies the return type as StringType().



Sort DataFrame by the first column and save the results to the Parquet file.

```
[73]: df = df.sort(df.columns[0])
```

write to Parquet file

```
[49]: df.write.parquet('data/output_data.parquet', mode='overwrite')
```

Read same Parquet file for confirmation

```
[50]: df = spark.read.parquet('data/output_data.parquet')
      df.show()
```

```
+---+---+---+--------------------+------+----+---+---+------------------+-------+----+---+-------------------+--------------------+---------------+------------------+------------
-+----------------+
|_c0|_c1|_c2|                 _c3|   _c4| _c5|_c6|_c7|               _c8|    _c9|_c10|_c11|               _c12|    _c3_UDF_Applied|_c4_UDF_Applied| _c8_UDF_Applied|_c10_UDF_Applie
d|_c11_UDF_Applied|
+---+---+---+--------------------+------+----+---+---+------------------+-------+----+---+-------------------+--------------------+---------------+------------------+------------
-+----------------+
|  1|  0|  3|Braund, Mr. Owen ...|  male|  22|  1|  0|         A/5 21171|   7.25|null|   S|2020-01-01 13:45:25|Braund1 Mr1 Owe1 ...|           mal1|         A/1 21171|         nul
l|               1|
|  2|  1|  1|Cumings, Mrs. Joh...|female|  38|  1|  0|          PC 17599|71.2833| C85|   C|2020-01-01 13:44:48|Cumings1 Mrs1 Joh...|          femal1|          P1 17591|          C8
1|               1|
|  3|  1|  3|Heikkinen, Miss. ...|female|  26|  0|  0|STON/O2. 3101282|  7.925|null|   S|2020-01-01 13:38:11|Heikkinen1 Miss1 ...|          femal1|STON/O21 3101281|         nul
l|               1|
|  4|  1|  1|Futrelle, Mrs. Ja...|female|  35|  1|  0|            113803|   53.1|C123|   S|2020-01-01 13:32:00|Futrelle1 Mrs1 Ja...|          femal1|            113801|         C12
1|               1|
|  5|  0|  3|Allen, Mr. Willia...|  male|  35|  0|  0|            373450|   8.05|null|   S|2020-01-01 13:36:30|Allen1 Mr1 Willia...|           mal1|            373451|         nul
l|               1|
|  6|  0|  3|    Moran, Mr. James|  male|null|  0|  0|            330877| 8.4583|null|   Q|2020-01-01 13:31:39|    Moran1 Mr1 Jame1|           mal1|            330871|         nul
l|               1|
|  7|  0|  1|McCarthy, Mr. Tim...|  male|  54|  0|  0|             17463|51.8625| E46|   S|2020-01-01 13:37:31|McCarthy1 Mr1 Tim...|           mal1|             17461|          E4
1|               1|
|  8|  0|  3|Palsson, Master. ...|  male|   2|  3|  1|            349909| 21.075|null|   S|2020-01-01 13:49:08|Palsson1 Master1 ...|           mal1|            349901|         nul
l|               1|
|  9|  1|  3|Johnson, Mrs. Osc...|female|  27|  0|  2|            347742|11.1333|null|   S|2020-01-01 13:33:42|Johnson1 Mrs1 Osc...|          femal1|            347741|         nul
l|               1|
| 10|  1|  2|Nasser, Mrs. Nich...|female|  14|  1|  0|            237736|30.0708|null|   C|2020-01-01 13:32:53|Nasser1 Mrs1 Nich...|          femal1|            237731|         nul
l|               1|
| 11|  1|  3|Sandstrom, Miss. ...|female|   4|  1|  1|           PP 9549|   16.7|  G6|   S|2020-01-01 13:32:23|Sandstrom1 Miss1 ...|          femal1|           P1 9541|           G
1|               1|
| 12|  1|  1|Bonnell, Miss. El...|female|  58|  0|  0|            113783|  26.55|C103|   S|2020-01-01 13:30:12|Bonnell1 Miss1 El...|          femal1|            113781|         C10
1|               1|
| 13|  0|  3|Saundercock, Mr. ...|  male|  20|  0|  0|         A/5. 2151|   8.05|null|   S|2020-01-01 13:33:34|Saundercock1 Mr1 ...|           mal1|         A/51 2151|         nul
```

- Finally, we sort the DataFrame by the first column and save the sorted DataFrame to a Parquet file, overwriting it if it already exists, using the write method and mode('overwrite').
- Then, we use the spark.read.parquet() method to read the Parquet file located at 'data/output_data.parquet' into a DataFrame named df. Finally, we use df.show() to display the contents of the DataFrame.