Medical Insurance Cost Analysis

Business Scenario

Problem statement:

A significant public health concern is the rising cost of healthcare. Therefore, it's crucial to be able to predict future costs and gain a solid understanding of their causes. The insurance industry must also take this analysis seriously. This analysis may be used by healthcare insurance providers to make a variety of strategic and tactical decisions.

Objective:

The objective of this project is to predict patients' healthcare costs and to identify factors contributing to this prediction. It will also be useful to learn the interdependencies of different factors and comprehend the significance of various tools at various stages of the healthcare cost prediction process.

Dataset Snapshot

Hospitalization details.xlsx

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID
2	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013
3	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013
4	Id2333	1993	Jun	30	0	600	tier - 2	tier - 1	R1013
5	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013
6	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013
7	Id2330	2001	Nov	20	0	646.14	tier - 3	tier - 3	R1012
8	Id2329	1993	Jun	1	0	650	tier - 3	tier - 3	R1013
9	Id2328	1995	Jul	4	0	650	tier - 3	tier - 3	R1013
10	Id2327	2002	Nov	29	0	668	tier - 3	tier - 2	R1012
11	Id2326	1997	Nov	9	0	670	tier - 3	tier - 3	R1013
12	Id2325	2001	Sep	12	0	687.54	tier - 3	tier - 2	R1013
13	Id2324	1999	Dec	26	0	700	?	tier - 3	R1013
14	Id2323	1999	Dec	14	0	722.99	tier - 3	tier - 1	R1013
15	Id2322	2002	?	19	0	750	tier - 3	tier - 1	R1012
16	Id2321	1993	Aug	9	0	760	tier - 3	tier - 1	R1013
17	Id2320	1996	Oct	22	0	760	tier - 3	tier - 3	R1013
18	Id2319	1993	Jun	28	0	770	tier - 3	tier - 3	R1013
19	Id2318	1996	?	18	0	770.38	tier - 3	?	R1012
20	Id2317	1995	Dec	7	0	773.54	tier - 3	tier - 2	R1013

Dataset Description

Hospitalization details.xlsx

Variables	Description
Customer ID	Unique identification for beneficiary(primary)
year	Year of birth
month	Month of birth
date	Date of birth
children	No. of children as dependents
charges	Hospitalization cost
Hospital tier	Level of hospital, tier 1 being the best
City tier	Level of city per government document, tier 1 referring to the most developed
State ID	ID of the state

Dataset Snapshot

Medical Examinations.xlsx

	Customer ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeries	smoker
2	Id1	47.41	7.47	No	No	No	No major surgery	yes
3	Id2	30.36	5.77	No	No	No	No major surgery	yes
4	Id3	34.485	11.87	yes	No	No	2	yes
5	Id4	38.095	6.05	No	No	No	No major surgery	yes
6	Id5	35.53	5.45	No	No	No	No major surgery	yes
7	Id6	32.8	6.59	No	No	No	No major surgery	yes
8	Id7	36.4	6.07	No	No	No	No major surgery	yes
9	Id8	36.96	7.93	No	No	No	3	yes
10	Id9	41.14	9.58	yes	No	Yes	1	yes
11	Id10	38.06	10.79	No	No	No	No major surgery	yes
12	Id11	37.7	5.96	yes	No	No	2	yes
13	Id12	42.13	11.9	No	No	No	No major surgery	yes
14	Id13	40.92	8.41	No	No	No	No major surgery	yes
15	Id14	40.565	7.02	No	No	No	No major surgery	yes
16	Id15	36.385	7.59	yes	No	No	2	yes
17	Id16	39.9	11.32	No	No	No	No major surgery	yes
18	Id17	33.8	7.67	No	No	No	3	yes
19	Id18	36.765	7.29	yes	No	Yes	1	yes
20	Id19	36.955	4.72	yes	No	No	1	yes
21	Id20	42.9	11.41	No	No	No	No major surgery	yes
22	Id21	36.3	11.5	yes	No	No	2	yes

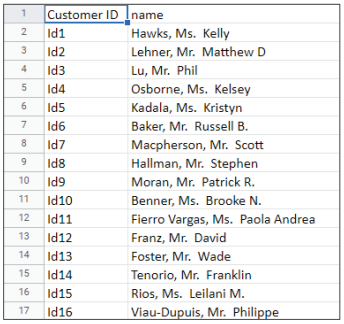
Dataset Description

Medical Examinations.xlsx

Variables	Description
Customer ID	Unique identification for beneficiary(primary)
BMI	Shows the body mass index of the individual (BMI measures body fat based on height and weight)
HBA1C	Shows the HBA1C report (HBA1C measures the amount of sugar in the blood (glucose), where HBA1C greater than 6.5 is considered diabetic
Heart Issues	Shows if a patient has heart-related issues
Any Transplants	Shows if a patient has any transplants in their body
Cancer history	Shows if a patient has any history of cancer in the family
NumberOfMajorSurgeries	Displays the number of major surgeries a patient has gone through
smoker	Indicates if a patient smokes cigarettes

Dataset Snapshot

Names.xlsx



	Customer ID	name
2	Id1	Hawks, Ms. Kelly
3	Id2	Lehner, Mr. Matthew D
4	Id3	Lu, Mr. Phil
5	Id4	Osborne, Ms. Kelsey
6	Id5	Kadala, Ms. Kristyn
7	Id6	Baker, Mr. Russell B.
8	Id7	Macpherson, Mr. Scott
9	Id8	Hallman, Mr. Stephen
10	Id9	Moran, Mr. Patrick R.
11	Id10	Benner, Ms. Brooke N.
12	Id11	Fierro Vargas, Ms. Paola Andrea
13	Id12	Franz, Mr. David
14	Id13	Foster, Mr. Wade
15	Id14	Tenorio, Mr. Franklin
16	Id15	Rios, Ms. Leilani M.
17	Id16	Viau-Dupuis, Mr. Philippe

Dataset Description

Names.xlsx

Variables	Description
Customer ID	Unique identification for beneficiary(primary)
name	Name of the beneficiary(primary)

Project Task: Week 1

Data science/data analysis

1. Collate the files so that all the information is in one place
2. Check for missing values in the dataset
3. Find the percentage of rows that have trivial value (for example, *?),* and delete such rows if they do not contain significant information
4. Use the necessary transformation methods to deal with the nominal and ordinal categorical variables in the dataset
5. The dataset has *State ID,* which has around 16 states. All states are not represented in equal proportions in the data. Creating dummy variables for all regions may also result in too many insignificant predictors. Nevertheless, only R1011, R1012, and R1013 are worth investigating further. Create a suitable strategy to create dummy variables with these restraints.
6. The variable *NumberOfMajorSurgeries* also appears to have string values. Apply a suitable method to clean up this variable.

Note: Use Excel as well as Python to complete the tasks

Project Task: Week 1

Data science/data analysis

7. *Age* appears to be a significant factor in this analysis. Calculate the patients' ages based on their dates of birth.
8. The gender of the patient may be an important factor in determining the cost of hospitalization. The salutations in a beneficiary's name can be used to determine their gender. Make a new field for the beneficiary's gender.
9. You should also visualize the distribution of costs using a histogram, box and whisker plot, and swarm plot.
10. State how the distribution is different across gender and tiers of hospitals
11. Create a radar chart to showcase the median hospitalization cost for each tier of hospitals
12. Create a frequency table and a stacked bar chart to visualize the count of people in the different tiers of cities and hospitals

Note: Use Excel as well as Python to complete the tasks

Project Task: Week 1

Data science/data analysis

13. Test the following null hypotheses:
    a. The average hospitalization costs for the three types of hospitals are not significantly different
    b. The average hospitalization costs for the three types of cities are not significantly different
    c. The average hospitalization cost for smokers is not significantly different from the average cost for nonsmokers
    d. Smoking and heart issues are independent

Note: Use Excel as well as Python to complete the tasks

Project Task: Week 2

Machine Learning

1. Examine the correlation between predictors to identify highly correlated predictors. Use a heatmap to visualize this.
2. Develop and evaluate the final model using regression with a stochastic gradient descent optimizer. Also, ensure that you apply all the following suggestions:

Note:

- Perform the stratified 5-fold cross-validation technique for model building and validation
- Use standardization and hyperparameter tuning effectively
- Use sklearn-pipelines
- Use appropriate regularization techniques to address the bias-variance trade-off

a. Create five folds in the data, and introduce a variable to identify the folds
b. For each fold, run a *for* loop and ensure that 80 percent of the data is used to train the model and the remaining 20 percent is used to validate it in each iteration
c. Develop five distinct models and five distinct validation scores (root mean squared error values)
d. Determine the variable importance scores, and identify the redundant variables

Project Task: Week 2

Machine Learning

3. Use random forest and extreme gradient boosting for cost prediction, share your cross-validation results, and calculate the variable importance scores
4. Case scenario:

   Estimate the cost of hospitalization for Christopher, Ms. Jayna (her date of birth is 12/28/1988, height is 170 cm, and weight is 85 kgs). She lives in a tier-1 city and her state's *State ID* is R1011. She lives with her partner and two children. She was found to be nondiabetic (HbA1c = 5.8). She smokes but is otherwise healthy. She has had no transplants or major surgeries. Her father died of lung cancer. Hospitalization costs will be estimated using tier-1 hospitals.
5. Find the predicted hospitalization cost using all five models. The predicted value should be the mean of the five models' predicted values.

Project Task: Week 2

SQL

1. To gain a comprehensive understanding of the factors influencing hospitalization costs, it is necessary to combine the tables provided. Merge the two tables by first identifying the columns in the data tables that will help you in merging.

   a. In both tables, add a *Primary Key* constraint for these columns

Hint: You can remove duplicates and null values from the column and then use *ALTER TABLE* to add a *Primary Key* constraint.

Project Task: Week 2

SQL

2. Retrieve information about people who are diabetic and have heart problems with their average age, the average number of dependent children, average BMI, and average hospitalization costs
3. Find the average hospitalization cost for each hospital tier and each city level
4. Determine the number of people who have had major surgery with a history of cancer
5. Determine the number of tier-1 hospitals in each state

Project Task: Week 2

Tableau

1. Create a dashboard in Tableau by selecting the appropriate chart types and business metrics

Note: Put more emphasis on data storytelling

Healthcare Insurance Analysis

In [162…
```
# Let's import the necessary library.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
%matplotlib inline
```

In [163…
```
# Let's remove the unnecessary warnings.
import warnings
warnings.filterwarnings("ignore")
```

In [164…
```
# Now importing the dataset for the further operation.
cust_details = pd.read_csv("Hospitalisation details.csv")
medical_details = pd.read_csv("Medical Examinations.csv")
cust_name = pd.read_excel("Names.xlsx")
```

In [165…
```
cust_details.head()
```

Out[165]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID
0	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013
1	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013
2	Id3	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012
3	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024
4	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012

In [166…
```
cust_details.shape
```

Out[166]:
```
(2335, 9)
```

In [167…
```
medical_details.head()
```

Out[167]:

	Customer ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeries	smoker
0	Id1	47.410	7.47	No	No	No	No major surgery	yes
1	Id2	30.360	5.77	No	No	No	No major surgery	yes
2	Id3	34.485	11.87	yes	No	No	2	yes
3	Id4	38.095	6.05	No	No	No	No major surgery	yes
4	Id5	35.530	5.45	No	No	No	No major surgery	yes

In [168…
```
medical_details.shape
```

Out[168]:
```
(2335, 8)
```

In [169…
```
cust_name.head()
```

Out[169]:

	Customer ID	name
0	Id1	Hawks, Ms. Kelly
1	Id2	Lehner, Mr. Matthew D
2	Id3	Lu, Mr. Phil
3	Id4	Osborne, Ms. Kelsey
4	Id5	Kadala, Ms. Kristyn

In [170…

```
cust_name.shape
```

Out[170]:

```
(2335, 2)
```

Project Task: Week 1

1. Collate the files so that all the information is in one place

In [171…

```
# Now combining the data so that all information could be examine in once go throug
cust_df1 = pd.merge(cust_name, cust_details, on = "Customer ID")
cust_df1.head()
```

Out[171]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	State ID
0	Id1	Hawks, Ms. Kelly	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013
1	Id2	Lehner, Mr. Matthew D	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013
2	Id3	Lu, Mr. Phil	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012
3	Id4	Osborne, Ms. Kelsey	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024
4	Id5	Kadala, Ms. Kristyn	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012

In [172…

```
# Now lets combine the last data set and Complete the all information.
final_df = pd.merge(cust_df1, medical_details, on = "Customer ID")
final_df.head()
```

Out[172]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HB
0	Id1	Hawks, Ms. Kelly	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	
1	Id2	Lehner, Mr. Matthew D	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	
2	Id3	Lu, Mr. Phil	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012	34.485	1
3	Id4	Osborne, Ms. Kelsey	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	
4	Id5	Kadala, Ms. Kristyn	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	

In [173…
```
final_df.shape
```

Out[173]:
```
(2335, 17)
```

2. Check for missing values in the dataset

In [174…
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2335 entries, 0 to 2334
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             2335 non-null   object 
 1   name                    2335 non-null   object 
 2   year                    2335 non-null   object 
 3   month                   2335 non-null   object 
 4   date                    2335 non-null   int64  
 5   children                2335 non-null   int64  
 6   charges                 2335 non-null   float64
 7   Hospital tier           2335 non-null   object 
 8   City tier               2335 non-null   object 
 9   State ID                2335 non-null   object 
 10  BMI                     2335 non-null   float64
 11  HBA1C                   2335 non-null   float64
 12  Heart Issues            2335 non-null   object 
 13  Any Transplants         2335 non-null   object 
 14  Cancer history          2335 non-null   object 
 15  NumberOfMajorSurgeries  2335 non-null   object 
 16  smoker                  2335 non-null   object 
dtypes: float64(3), int64(2), object(12)
memory usage: 328.4+ KB
```

In [175…
```
final_df.dtypes.value_counts()
```

Out[175]:
```
object     12
float64     3
int64       2
dtype: int64
```

In [176…
```
# Lets check the missing values in the data set.
final_df.isnull().sum()
```

Out[176]:
```
Customer ID               0
name                      0
year                      0
month                     0
date                      0
children                  0
charges                   0
Hospital tier             0
City tier                 0
State ID                  0
BMI                       0
HBA1C                     0
Heart Issues              0
Any Transplants           0
Cancer history            0
NumberOfMajorSurgeries    0
smoker                    0
dtype: int64
```

There is no missing value in the dataset it is clear fromt he above code, But there is some unusual value that we have to deal.

3. Find the percentage of rows that have trivial value (for example, ?), and delete such rows if they do not contain significant information

In [177…
```
trivial_value = final_df[final_df.eq("?").any(1)]
trivial_value
```

Out[177]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI
2	Id3	Lu, Mr. Phil	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012	34.485
169	Id170	Torphy, Mr. Bobby	2000	Sep	5	1	37165.16	tier - 1	tier - 3	?	37.620
559	Id560	Pearlman, Mr. Oz	1994	Jul	1	3	17663.14	tier - 1	tier - 3	R1013	23.980
634	Id635	Bruns, Mr. Zachary T	2004	Jul	17	0	15518.18	tier - 2	tier - 3	R1015	25.175
1285	Id1286	Ainsley, Ms. Katie M.	?	Dec	12	1	8547.69	tier - 2	tier - 1	R1013	29.370
1288	Id1289	Levine, Ms. Annie J.	?	Jul	24	0	8534.67	tier - 2	tier - 3	R1024	24.320
1792	Id1793	Capriolo, Mr. Michael	1995	Dec	1	3	4827.90	tier - 1	tier - 2	?	18.905
2317	Id2318	Gagnon, Ms. Candice M	1996	?	18	0	770.38	tier - 3	?	R1012	18.820
2321	Id2322	Street, Ms. Holly	2002	?	19	0	750.00	tier - 3	tier - 1	R1012	21.380
2323	Id2324	Duffy, Ms. Meghan K	1999	Dec	26	0	700.00	?	tier - 3	R1013	22.240

In [178…
```
trivial_value.shape
```

Out[178]:
```
(10, 17)
```

In [179…
```
# Percentage of row that have the trivial values
round(trivial_value.shape[0]/final_df.shape[0]*100, 2)
```

Out[179]:
```
0.43
```

There is total 0.43% of rows contain the trivial values.

In [180…
```
# Now lets drop the all row that contain the trivial values in the data set.
final_df.drop(final_df[final_df.eq("?").any(1)].index, axis=0, inplace=True)
```

In [181…
```
final_df.shape
```

Out[181]:
```
(2325, 17)
```

4. Use the necessary transformation methods to deal with the nominal and ordinal categorical variables in the dataset

In [182…
```
# First we will deal with the nominal categorical variable.
```

In [183…
```
final_df["Heart Issues"].value_counts()
```

Out[183]:
```
No     1405
yes     920
Name: Heart Issues, dtype: int64
```

In [184…
```
final_df["Any Transplants"].value_counts()
```

Out[184]:
```
No     2183
yes     142
Name: Any Transplants, dtype: int64
```

In [185…
```
final_df["Cancer history"].value_counts()
```

Out[185]:
```
No     1934
Yes     391
Name: Cancer history, dtype: int64
```

In [186…
```
final_df["smoker"].value_counts()
```

Out[186]:
```
No     1839
yes     486
Name: smoker, dtype: int64
```

In [187…
```
# We have some categorical values so first of all we have to transform then by usi
from sklearn.preprocessing import LabelEncoder
```

In [188…
```
le = LabelEncoder()
```

In [189…
```
final_df["Heart Issues"] = le.fit_transform(final_df["Heart Issues"])
final_df["Any Transplants"] = le.fit_transform(final_df["Any Transplants"])
final_df["Cancer history"] = le.fit_transform(final_df["Cancer history"])
final_df["smoker"] = le.fit_transform(final_df["smoker"])
```

In [190…
```
final_df["Heart Issues"].value_counts()
```

Out[190]:
```
0    1405
1     920
Name: Heart Issues, dtype: int64
```

In [191…
```
final_df.head()
```

Out[191]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HB
0	Id1	Hawks, Ms. Kelly	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	
1	Id2	Lehner, Mr. Matthew D	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	
3	Id4	Osborne, Ms. Kelsey	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	
4	Id5	Kadala, Ms. Kristyn	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	
5	Id6	Baker, Mr. Russell B.	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	

In [192…
```
# Now we will deal with the ordinal categorical variable.
```

In [193…
```
def clean_ordinal_variable(val):
    return int(val.replace("tier", "").replace(" ", "").replace("-", ""))
```

In [194…
```
final_df["Hospital tier"] = final_df["Hospital tier"].map(clean_ordinal_variable)
final_df["City tier"] = final_df["City tier"].map(clean_ordinal_variable)
```

In [195…
```
final_df["City tier"].value_counts()
```

Out[195]:
```
2    807
3    789
1    729
Name: City tier, dtype: int64
```

In [196…
```
final_df.head()
```

Out[196]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HB
0	Id1	Hawks, Ms. Kelly	1968	Oct	12	0	63770.43	1	3	R1013	47.410	
1	Id2	Lehner, Mr. Matthew D	1977	Jun	8	0	62592.87	2	3	R1013	30.360	
3	Id4	Osborne, Ms. Kelsey	1991	Jun	6	1	58571.07	1	3	R1024	38.095	
4	Id5	Kadala, Ms. Kristyn	1989	Jun	19	0	55135.40	1	2	R1012	35.530	
5	Id6	Baker, Mr. Russell B.	1962	Aug	4	0	52590.83	1	3	R1011	32.800	

5. The dataset has State ID, which has around 16 states. All states are not represented in equal proportions in the data. Creating dummy variables for all regions may also result in too many insignificant predictors. Nevertheless, only R1011, R1012, and R1013 are worth investigating further. Create a suitable strategy to create dummy variables with these restraints.

In [197…

```
final_df["State ID"].value_counts()
```

Out[197]:

```
R1013    609
R1011    574
R1012    572
R1024    159
R1026     84
R1021     70
R1016     64
R1025     40
R1023     38
R1017     36
R1019     26
R1022     14
R1014     13
R1015     11
R1018      9
R1020      6
Name: State ID, dtype: int64
```

It is clear from the above code some of the state is worth investigator like R1013, R1012, R1011 and R1024.

In [198…

```
Dummies = pd.get_dummies(final_df["State ID"], prefix= "State_ID")
```

In [199…

```
Dummies
```

Out[199]:

	State_ID_R1011	State_ID_R1012	State_ID_R1013	State_ID_R1014	State_ID_R1015	State_ID_F
0	0	0	1	0	0	
1	0	0	1	0	0	
3	0	0	0	0	0	
4	0	1	0	0	0	
5	1	0	0	0	0	
...	...	...	...	...	...	
2330	0	0	1	0	0	
2331	0	0	1	0	0	
2332	0	0	1	0	0	
2333	0	0	1	0	0	
2334	0	0	1	0	0	

2325 rows × 16 columns

In [200…

```
# lets take only those state id which play significant role in the data set.
Dummy = Dummies[['State_ID_R1011','State_ID_R1012', 'State_ID_R1013']]
Dummy
```

Out[200]:

	State_ID_R1011	State_ID_R1012	State_ID_R1013
0	0	0	1
1	0	0	1
3	0	0	0
4	0	1	0
5	1	0	0
...	...	...	...
2330	0	0	1
2331	0	0	1
2332	0	0	1
2333	0	0	1
2334	0	0	1

2325 rows × 3 columns

In [201…

```
final_df = pd.concat([final_df, Dummy], axis=1)
```

In [202…

```
final_df.drop(['State ID'], inplace=True, axis=1)
```

In [203…

```
final_df.head()
```

Out[203]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	BMI	HBA1C	H Is
0	Id1	Hawks, Ms. Kelly	1968	Oct	12	0	63770.43	1	3	47.410	7.47	
1	Id2	Lehner, Mr. Matthew D	1977	Jun	8	0	62592.87	2	3	30.360	5.77	
3	Id4	Osborne, Ms. Kelsey	1991	Jun	6	1	58571.07	1	3	38.095	6.05	
4	Id5	Kadala, Ms. Kristyn	1989	Jun	19	0	55135.40	1	2	35.530	5.45	
5	Id6	Baker, Mr. Russell B.	1962	Aug	4	0	52590.83	1	3	32.800	6.59	

6. The variable NumberOfMajorSurgeries also appears to have string values. Apply a suitable method to clean up this variable.

In [204…
```
final_df['NumberOfMajorSurgeries'].value_counts()
```

Out[204]:
```
No major surgery    1070
1                    961
2                    272
3                     22
Name: NumberOfMajorSurgeries, dtype: int64
```

The NumberOfMajorSurgeries variable contain string value no major Surgery that mean simpli is 0 surgery so we will replace this value into int value equal to zero.

In [205…
```
final_df['NumberOfMajorSurgeries'] = final_df['NumberOfMajorSurgeries'].replace('N
```

In [206…
```
final_df['NumberOfMajorSurgeries'] = final_df["NumberOfMajorSurgeries"].astype(int
```

7. Age appears to be a significant factor in this analysis. Calculate the patients' ages based on their dates of birth.

In [207…
```
final_df["year"] = pd.to_datetime(final_df["year"], format='%Y').dt.year
final_df["year"]
```

Out[207]:
```
0       1968
1       1977
3       1991
4       1989
5       1962
        ...
2330    1998
2331    1992
2332    1993
2333    1992
2334    1992
Name: year, Length: 2325, dtype: int64
```

In [208…
```
final_df["month"] = pd.to_datetime(final_df["month"], format='%b').dt.month
final_df["month"]
```

Out[208]:
```
0       10
1        6
3        6
4        6
5        8
        ..
2330     7
2331     9
2332     6
2333    11
2334     7
Name: month, Length: 2325, dtype: int64
```

In [209…
```
final_df['DateInt'] = final_df["year"].astype(str) + final_df["month"].astype(str)
```

In [210…
```
final_df['DOB'] = pd.to_datetime(final_df.DateInt, format = "%Y%m%d")
```

In [211…
```
final_df.drop(["DateInt"], inplace = True, axis=1)
```

In [212…
```
final_df.head()
```

Out[212]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	BMI	HBA1C	H Is
0	Id1	Hawks, Ms. Kelly	1968	10	12	0	63770.43	1	3	47.410	7.47	
1	Id2	Lehner, Mr. Matthew D	1977	6	8	0	62592.87	2	3	30.360	5.77	
3	Id4	Osborne, Ms. Kelsey	1991	6	6	1	58571.07	1	3	38.095	6.05	
4	Id5	Kadala, Ms. Kristyn	1989	6	19	0	55135.40	1	2	35.530	5.45	
5	Id6	Baker, Mr. Russell B.	1962	8	4	0	52590.83	1	3	32.800	6.59	

In [213…
```
import datetime as dt
current_date = dt.datetime.now()
```

In [214…
```
final_df['age'] = (((current_date - final_df.DOB).dt.days)/365).astype(int)
```

In [215…
```
final_df.head()
```

Out[215]:

	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	BMI	...	Heart Issues
0	Id1	Hawks, Ms. Kelly	1968	10	12	0	63770.43	1	3	47.410	...	0
1	Id2	Lehner, Mr. Matthew D	1977	6	8	0	62592.87	2	3	30.360	...	0
3	Id4	Osborne, Ms. Kelsey	1991	6	6	1	58571.07	1	3	38.095	...	0
4	Id5	Kadala, Ms. Kristyn	1989	6	19	0	55135.40	1	2	35.530	...	0
5	Id6	Baker, Mr. Russell B.	1962	8	4	0	52590.83	1	3	32.800	...	0

5 rows × 21 columns

8. The gender of the patient may be an important factor in determining the cost of hospitalization. The salutations in a beneficiary's name can be used to determine their gender. Make a new field for the beneficiary's gender.

In [216…
```
def gender(val):
    if "Ms." in val:
        return 0
    else:
        return 1
```

the salutation (Ms.) denote the female and (Mr.) denote the male.

> The gender will play the inportant role to predict the hospitalization cost so for model building we directly denote the gender by int.
>
> > Male = 1 & Female = 0

In [217…
```
final_df["gender"] = final_df["name"].map(gender)
```

In [218…
```
final_df.head()
```

Out[218]:

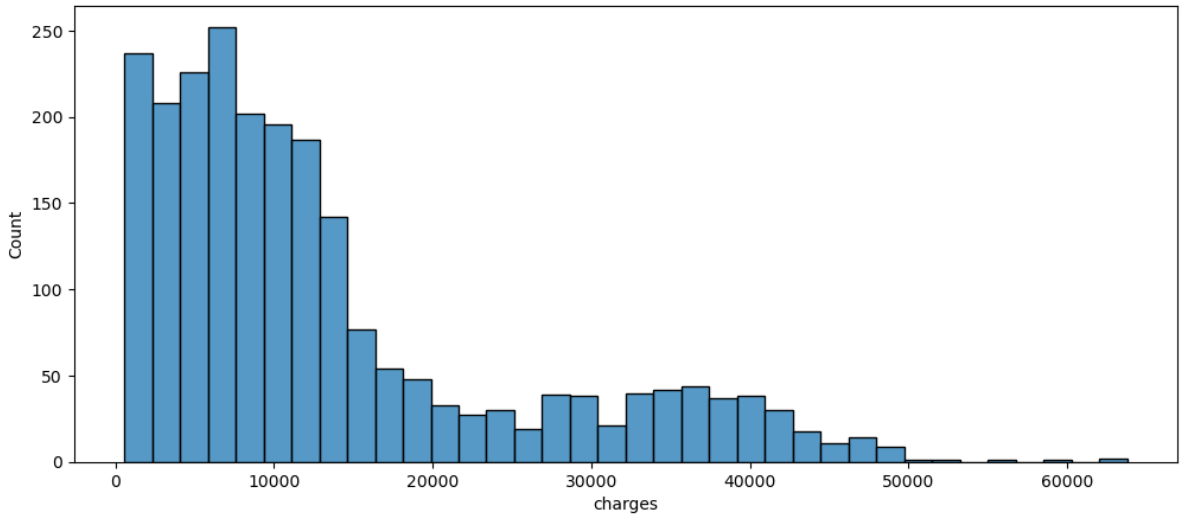
	Customer ID	name	year	month	date	children	charges	Hospital tier	City tier	BMI	...	Transpl
0	Id1	Hawks, Ms. Kelly	1968	10	12	0	63770.43	1	3	47.410	...	
1	Id2	Lehner, Mr. Matthew D	1977	6	8	0	62592.87	2	3	30.360	...	
3	Id4	Osborne, Ms. Kelsey	1991	6	6	1	58571.07	1	3	38.095	...	
4	Id5	Kadala, Ms. Kristyn	1989	6	19	0	55135.40	1	2	35.530	...	
5	Id6	Baker, Mr. Russell B.	1962	8	4	0	52590.83	1	3	32.800	...	

5 rows × 22 columns

9. You should also visualize the distribution of costs using a histogram, box and whisker plot, and swarm plot.

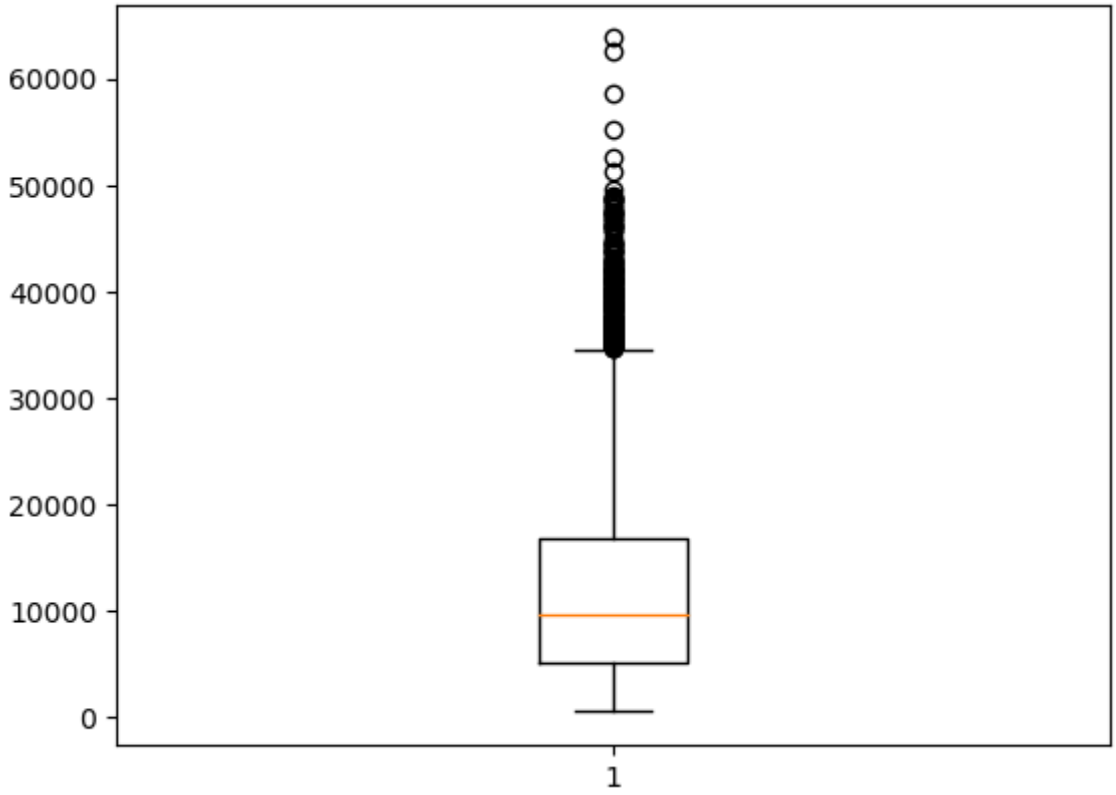
In [219…

```
# Lets make the histogram for the cost distribution.
plt.figure(figsize=(12,5))
sns.histplot(final_df['charges'])
plt.show()
```

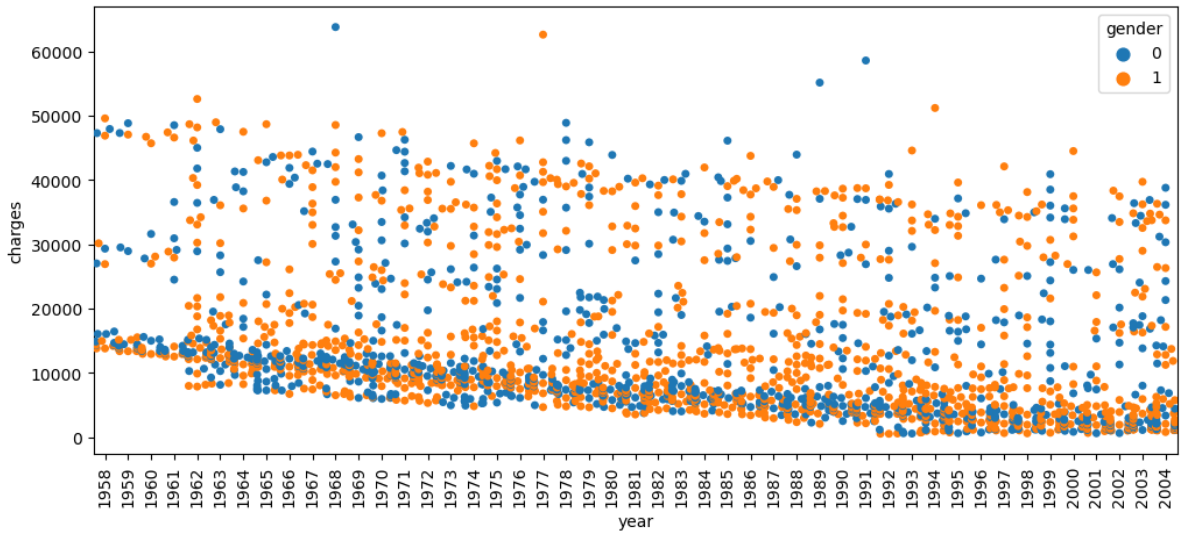


In [220…

```
# Now visualize the cost distribution of the hospitals by box or whisker plot.
plt.boxplot(final_df['charges'])
plt.show()
```

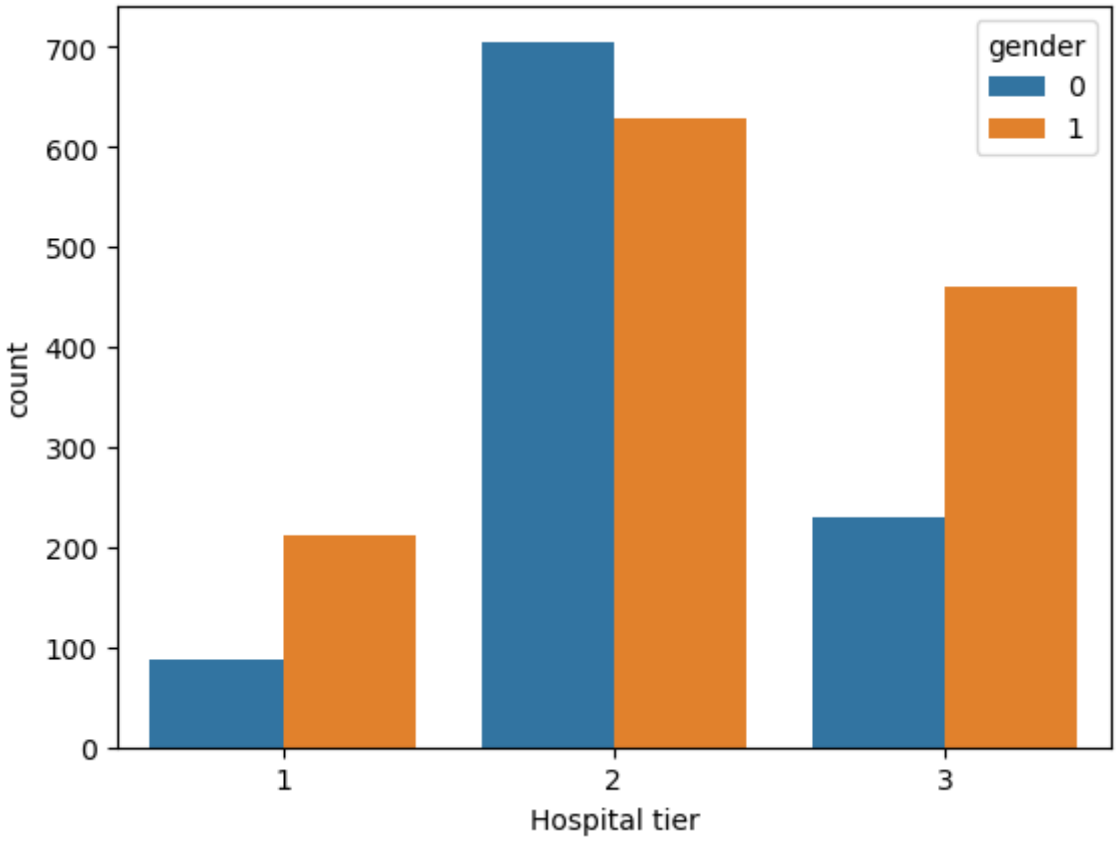


In [221…
```
# Now visualize the cost distribution of the hospitals by swarm plot.
plt.figure(figsize=(12,5))
sns.swarmplot(x='year', y='charges', hue="gender", data=final_df)
plt.xticks(rotation=90)
plt.show()
```



10. State how the distribution is different across gender and tiers of hospitals

In [222…
```
sns.countplot(data = final_df, x='Hospital tier', hue= 'gender')
plt.show()
```



From the above representation it is clear that the number of female in the tier 1 and 3 is half of the male just in tier 2 hospital female is little bit more as compare to male.

11. Create a radar chart to showcase the median hospitalization cost for each tier of hospitals

In [223…
```
print("median cost of tier 1 hospitals:", final_df[final_df["Hospital tier"]==1].cl
print("median cost of tier 2 hospitals:", final_df[final_df["Hospital tier"]==2].cl
print("median cost of tier 3 hospitals:", final_df[final_df["Hospital tier"]==3].cl
```

```
median cost of tier 1 hospitals: 32097.434999999998
median cost of tier 2 hospitals: 7168.76
median cost of tier 3 hospitals: 10676.83
```

In [224…
```
df = pd.DataFrame(dict(r=[32097.43, 7168.76, 10676.83],theta=['tier 1 hospital','t
```

In [225…
```
df
```

Out[225]:

	r	theta
0	32097.43	tier 1 hospital
1	7168.76	tier 2 hospital
2	10676.83	tier 3 hospital

In [226…
```
import plotly.express as px
fig = px.line_polar(df, r='r', theta='theta', line_close=True)
fig.update_traces(fill='toself')
fig.show()
```

12. Create a frequency table and a stacked bar chart to visualize the count of people in the different tiers of cities and hospitals

In [227…
```
# Frequency table for count of the people according to the tier of city and hospita
final_df["Hospital tier"].value_counts()
```

Out[227]:
```
2    1334
3     691
1     300
Name: Hospital tier, dtype: int64
```

In [228…
```
city_freq = final_df["City tier"].value_counts().rename_axis('City&hospital_tier')
```

In [229…
```
hospital_freq = final_df["Hospital tier"].value_counts().rename_axis('City&hospital
```

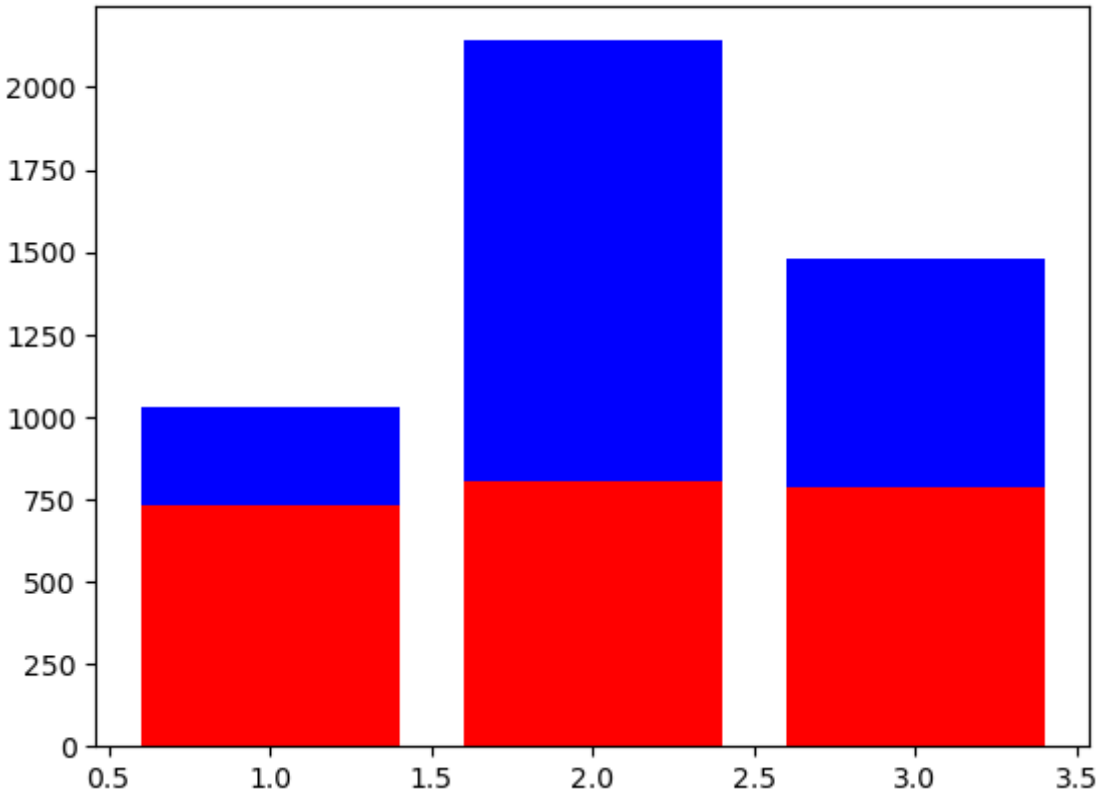
In [230…
```
df = pd.merge(city_freq, hospital_freq, on = 'City&hospital_tier')
```

In [231…
```
df
```

Out[231]:

	City&hospital_tier	city_counts	hospital_counts
0	2	807	1334
1	3	789	691
2	1	729	300

In [232…
```
plt.bar(df["City&hospital_tier"], df["city_counts"], color='r')
plt.bar(df["City&hospital_tier"], df["hospital_counts"], bottom=df["city_counts"],
plt.show()
```



13. Test the following null hypotheses:

a. The average hospitalization costs for the three types of hospitals are not significantly different

b. The average hospitalization costs for the three types of cities are not significantly different

c. The average hospitalization cost for smokers is not significantly different from the average cost for nonsmokers

d. Smoking and heart issues are independent

In [233…
```
from scipy.stats import ttest_1samp
```

In [234…
```
# a. The average hospitalization costs for the three types of hospitals are not sig
print("median cost of tier 1 hospitals:", final_df[final_df["Hospital tier"]==1].cl
print("median cost of tier 2 hospitals:", final_df[final_df["Hospital tier"]==2].cl
print("median cost of tier 3 hospitals:", final_df[final_df["Hospital tier"]==3].cl
```

```
median cost of tier 1 hospitals: 32097.434999999998
median cost of tier 2 hospitals: 7168.76
median cost of tier 3 hospitals: 10676.83
```

Interpretation

H0: the distributions of all samples are equal. || H1: the distributions of one or more samples are not equal

In [235…
```
from scipy.stats import friedmanchisquare
data1 = [32097.43]
data2 = [7168.76]
data3 = [10676.83]
stat, p = friedmanchisquare(data1, data2, data3)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=2.000, p=0.368
Probably the same distribution
```

In [236…
```
# b. The average hospitalization costs for the three types of cities are not signi
print("median cost of tier 1 city:", final_df[final_df["City tier"]==1].charges.me
print("median cost of tier 2 city:", final_df[final_df["City tier"]==2].charges.me
print("median cost of tier 3 city:", final_df[final_df["City tier"]==3].charges.me
```

```
median cost of tier 1 city: 10027.15
median cost of tier 2 city: 8968.33
median cost of tier 3 city: 9880.07
```

In [237…
```
data1 = [10027.15]
data2 = [8968.33]
data3 = [9880.07]
stat, p = friedmanchisquare(data1, data2, data3)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=2.000, p=0.368
Probably the same distribution
```

In [238…
```
# c. The average hospitalization cost for smokers is not significantly different fr
print("median cost of smoker:", final_df[final_df["smoker"]==1].charges.median())
print("median cost of non smoker:", final_df[final_df["smoker"]==0].charges.median
```

```
median cost of smoker: 34125.475
median cost of non smoker: 7537.16
```

In [239…
```
from scipy.stats import kruskal
data1 = [34125.475]
data2 = [7537.16]
stat, p = kruskal(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

```
stat=1.000, p=0.317
Probably the same distribution
```

Interpretation

H0: the two samples are independent. H1: there is a dependency between the samples.

In [240…
```
# d. Smoking and heart issues are independent
from scipy.stats import chi2_contingency
table = [[final_df["Heart Issues"].value_counts()],[final_df["smoker"].value_counts
```

```
stat, p, dof, expected = chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=191.145, p=0.000
Probably dependent
```

Project Task: Week 2

1. Examine the correlation between predictors to identify highly correlated predictors. Use a heatmap to visualize this.

In [241…
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2325 entries, 0 to 2334
Data columns (total 22 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             2325 non-null   object
 1   name                    2325 non-null   object
 2   year                    2325 non-null   int64
 3   month                   2325 non-null   int64
 4   date                    2325 non-null   int64
 5   children                2325 non-null   int64
 6   charges                 2325 non-null   float64
 7   Hospital tier           2325 non-null   int64
 8   City tier               2325 non-null   int64
 9   BMI                     2325 non-null   float64
 10  HBA1C                   2325 non-null   float64
 11  Heart Issues            2325 non-null   int32
 12  Any Transplants         2325 non-null   int32
 13  Cancer history          2325 non-null   int32
 14  NumberOfMajorSurgeries  2325 non-null   int32
 15  smoker                  2325 non-null   int32
 16  State_ID_R1011          2325 non-null   uint8
 17  State_ID_R1012          2325 non-null   uint8
 18  State_ID_R1013          2325 non-null   uint8
 19  DOB                     2325 non-null   datetime64[ns]
 20  age                     2325 non-null   int32
 21  gender                  2325 non-null   int64
dtypes: datetime64[ns](1), float64(3), int32(6), int64(7), object(2), uint8(3)
memory usage: 315.6+ KB
```

In [242…
```
# In the data frame same of the column are not usable to model building so lets fi
#then indentify the highly corelated predictor.
final_df.drop(["Customer ID",'name', 'year', 'month', 'date', 'DOB'], inplace=True
```

In [243…
```
final_df.shape
```

Out[243]:
```
(2325, 16)
```

In [244…
```
final_df.head()
```

Out[244]:

	children	charges	Hospital tier	City tier	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajo
0	0	63770.43	1	3	47.410	7.47	0	0	0	
1	0	62592.87	2	3	30.360	5.77	0	0	0	
3	1	58571.07	1	3	38.095	6.05	0	0	0	
4	0	55135.40	1	2	35.530	5.45	0	0	0	
5	0	52590.83	1	3	32.800	6.59	0	0	0	

In [245…

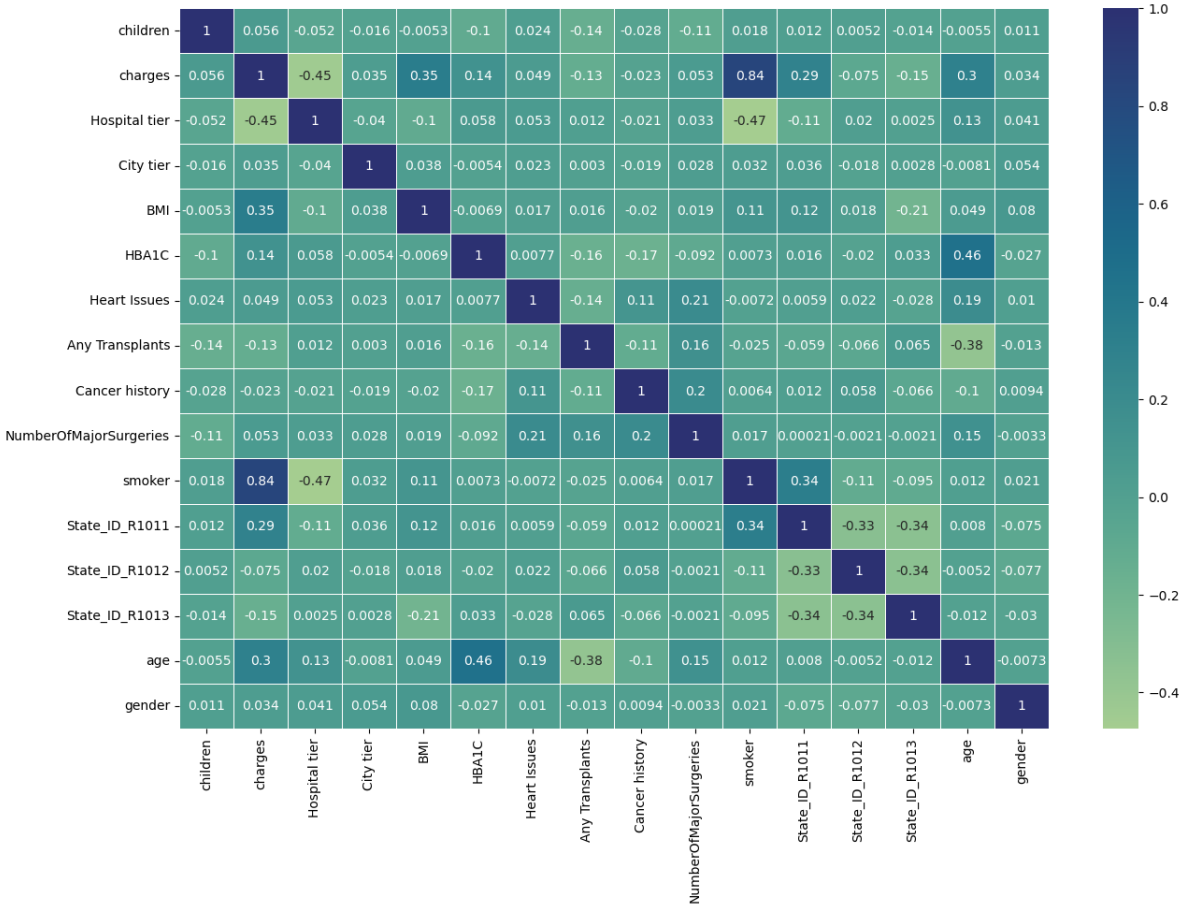
```
corr = final_df.corr()
corr
```

Out[245]:

	children	charges	Hospital tier	City tier	BMI	HBA1C	Hea Issu
children	1.000000	0.055901	-0.052438	-0.015760	-0.005339	-0.101379	0.02398
charges	0.055901	1.000000	-0.446687	0.035300	0.346730	0.139697	0.04929
Hospital tier	-0.052438	-0.446687	1.000000	-0.039755	-0.104771	0.057855	0.05337
City tier	-0.015760	0.035300	-0.039755	1.000000	0.038123	-0.005404	0.02315
BMI	-0.005339	0.346730	-0.104771	0.038123	1.000000	-0.006920	0.01712
HBA1C	-0.101379	0.139697	0.057855	-0.005404	-0.006920	1.000000	0.00769
Heart Issues	0.023984	0.049299	0.053376	0.023152	0.017129	0.007699	1.00000
Any Transplants	-0.142040	-0.127028	0.011729	0.002970	0.015893	-0.159855	-0.14026
Cancer history	-0.027880	-0.022522	-0.021429	-0.018639	-0.020235	-0.170921	0.11119
NumberOfMajorSurgeries	-0.113161	0.053308	0.033230	0.027937	0.018851	-0.091594	0.20614
smoker	0.017713	0.838462	-0.474077	0.032034	0.107126	0.007257	-0.00715
State_ID_R1011	0.011666	0.286956	-0.114685	0.036049	0.115671	0.015525	0.00585
State_ID_R1012	0.005247	-0.074636	0.020272	-0.018253	0.017939	-0.019513	0.02177
State_ID_R1013	-0.013834	-0.150634	0.002455	0.002766	-0.208744	0.033453	-0.02796
age	-0.005457	0.304395	0.133771	-0.008070	0.049260	0.460558	0.19227
gender	0.011205	0.034069	0.041261	0.054073	0.079930	-0.027339	0.01027

In [246…

```
plt.figure(figsize=(15,10))
sns.heatmap(corr, annot=True, linewidth=.5, cmap="crest")
plt.show()
```



From the above corelation its clear that somker variable is highly corealted to the output variable.

2. Develop and evaluate the final model using regression with a stochastic gradient descent optimizer. Also, ensure that you apply all the following suggestions:

Note:

• Perform the stratified 5-fold cross-validation technique for model building and validation • Use standardization and hyperparameter tuning effectively • Use sklearn-pipelines • Use appropriate regularization techniques to address the bias-variance trade-off

a. Create five folds in the data, and introduce a variable to identify the folds

b. For each fold, run a for loop and ensure that 80 percent of the data is used to train the model and the remaining 20 percent is used to validate it in each iteration

c. Develop five distinct models and five distinct validation scores (root mean squared error values)

d. Determine the variable importance scores, and identify the redundant variables

In [247…
```
# Lets first seperate the input and output data.
x = final_df.drop(["charges"], axis=1)
y = final_df[['charges']]
```

In [248…
```
# Lets split the data set into the training and testing data.
from sklearn.model_selection import train_test_split
```

In [249…
```
x_train, x_test, y_train, y_test  = train_test_split(x,y, test_size=.20, random_st
```

In [250…
```
# Now standardize the data.
from sklearn.preprocessing import StandardScaler
```

In [251…
```
sc = StandardScaler()
```

In [252…
```
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

In [253…
```
from sklearn.linear_model import SGDRegressor
```

In [254…
```
from sklearn.model_selection import GridSearchCV

params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1, 0.2,0.3,0.4,0.5,
                    0.6,0.7,0.8,0.9,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
                    9.0,10.0,20,50,100,500,1000],
          'penalty': ['l2', 'l1', 'elasticnet']}

sgd = SGDRegressor()

# Cross Validation
folds = 5
model_cv = GridSearchCV(estimator = sgd,
                        param_grid = params,
                        scoring = 'neg_mean_absolute_error',
                        cv = folds,
                        return_train_score = True,
                        verbose = 1)
model_cv.fit(x_train,y_train)
```

Out[254]:
```
Fitting 5 folds for each of 84 candidates, totalling 420 fits
GridSearchCV(cv=5, estimator=SGDRegressor(),
             param_grid={'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3,
                                   0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0, 3.0,
                                   4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 20, 50,
                                   100, 500, 1000],
                         'penalty': ['l2', 'l1', 'elasticnet']},
             return_train_score=True, scoring='neg_mean_absolute_error',
             verbose=1)
```

In [255…
```
model_cv.best_params_
```

Out[255]:
```
{'alpha': 50, 'penalty': 'l1'}
```

In [256…
```
sgd = SGDRegressor(alpha= 100, penalty= 'l1')
```

In [257…
```
sgd.fit(x_train, y_train)
```

Out[257]:
```
SGDRegressor(alpha=100, penalty='l1')
```

In [258…
```
sgd.score(x_test, y_test)
```

Out[258]:
```
0.8574058061920549
```

In [259…
```
y_pred = sgd.predict(x_test)
```

In [260…
```
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

In [261…
```
sgd_mae = mean_absolute_error(y_test, y_pred)
sgd_mse = mean_squared_error(y_test, y_pred)
sgd_rmse = np.sqrt(sgd_mse)
```

In [262…
```
print("MAE:", sgd_mae)
print("MSE:", sgd_mse)
print("RMSE:", sgd_rmse)
```

```
MAE: 3145.3196499430524
MSE: 23985187.279824603
RMSE: 4897.4674353000655
```

In [263…
```
# d. Determine the variable importance scores, and identify the redundant variables
importance = sgd.coef_
```

In [264…
```
pd.DataFrame(importance, index = x.columns, columns=['Feature_imp'])
```

Out[264]:

	Feature_imp
children	365.627972
Hospital tier	-1080.163416
City tier	0.000000
BMI	2669.591053
HBA1C	41.799507
Heart Issues	0.000000
Any Transplants	0.000000
Cancer history	47.258781
NumberOfMajorSurgeries	0.000000
smoker	8741.236701
State_ID_R1011	-150.534783
State_ID_R1012	0.000000
State_ID_R1013	-376.626750
age	3383.403770
gender	0.000000

3. Use random forest and extreme gradient boosting for cost prediction, share your crossvalidation results, and calculate the variable importance scores

random forest

In [265…
```
from sklearn.ensemble import RandomForestRegressor
```

In [266…
```
# Instantiate model with 1000 decision trees
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)

# Train the model on training data
rf.fit(x_train, y_train)
```

Out[266]:
```
RandomForestRegressor(n_estimators=1000, random_state=42)
```

In [267…
```
score = rf.score(x_test,y_test)
score
```

Out[267]:
```
0.9222696338245824
```

In [268…
```
y_pred = rf.predict(x_test)
```

In [269…
```
rf_mae = mean_absolute_error(y_test, y_pred)
```

In [270…
```
rf_mae
```

Out[270]:
```
1870.3529629462323
```

extreme gradient boosting

In [271…
```
from sklearn.ensemble import GradientBoostingRegressor
```

In [272…
```
# Instantiate model with 1000 decision trees
gbr = GradientBoostingRegressor(n_estimators = 1000, random_state = 42)

# Train the model on training data
gbr.fit(x_train, y_train)
```

Out[272]:
```
GradientBoostingRegressor(n_estimators=1000, random_state=42)
```

In [273…
```
score = gbr.score(x_test,y_test)
score
```

Out[273]:
```
0.9042734212625119
```

In [274…
```
y_pred = gbr.predict(x_test)
```

In [275…
```
gbr_mae = mean_absolute_error(y_test, y_pred)
gbr_mae
```

Out[275]:
```
2375.8700944163274
```

4. Case scenario:

Estimate the cost of hospitalization for Christopher, Ms. Jayna (her date of birth is 12/28/1988, height is 170 cm, and weight is 85 kgs). She lives in a tier-1 city and her state's State ID is R1011. She lives with her partner and two children. She was found to be nondiabetic (HbA1c = 5.8). She smokes but is otherwise healthy. She has had no transplants or major

surgeries. Her father died of lung cancer. Hospitalization costs will be estimated using tier-1 hospitals.

In [276…
```
# First we need to calculate the age of the person.
date = "19881228"
date1 = datetime.strptime(date,"%Y%m%d")
date1
```

Out[276]:
```
datetime.datetime(1988, 12, 28, 0, 0)
```

In [277…
```
current_date = datetime.now()
current_date
```

Out[277]:
```
datetime.datetime(2023, 3, 1, 22, 56, 36, 990464)
```

In [278…
```
age =str((current_date - date1)/365)
```

In [279…
```
print("Age=",age[:2])
```

```
Age= 34
```

In [280…
```
# now with the help of height and weight we will calculate the BMI.
height_m = 170/100
height_sq = height_m*height_m
BMI = 85/height_sq
np.round(BMI,2)
```

Out[280]:
```
29.41
```

In [281…
```
# Now lets gen
list = [[2,1,1,24.41,5.8,0,0,0,0,1,1,0,0,34,0]]
```

In [282…
```
df = pd.DataFrame(list, columns = ['children', 'Hospital tier', 'City tier', 'BMI',
                                   'Cancer history','NumberOfMajorSurgeries', 'smoker',
                                   'State_ID_R1013', 'age', 'gender'] )
df
```

Out[282]:

	children	Hospital tier	City tier	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeries
0	2	1	1	24.41	5.8	0	0	0	0

5. Find the predicted hospitalization cost using all models. The predicted value should be the mean of the five models' predicted values.

In [283…
```
Hospital_cost = []
```

In [284…
```
# Now lets predict the hospitalization cost through SGDRegressor
Cost1 = sgd.predict(df)
Hospital_cost.append(Cost1)
```

In [285…
```
# Now lets predict the hospitalization cost through Random Forest
Cost2 = rf.predict(df)
Hospital_cost.append(Cost2)
```

In [286…
```
# Now lets predict the hospitalization cost throug Extreme gradient Booster
Cost3 = gbr.predict(df)
Hospital_cost.append(Cost3)
```

In [287…
```
avg_cost = np.mean(Hospital_cost)
avg_cost
```

Out[287]:
```
103919.51224697009
```

So in the new case the avg predicted hospitalization cost is 103919.51

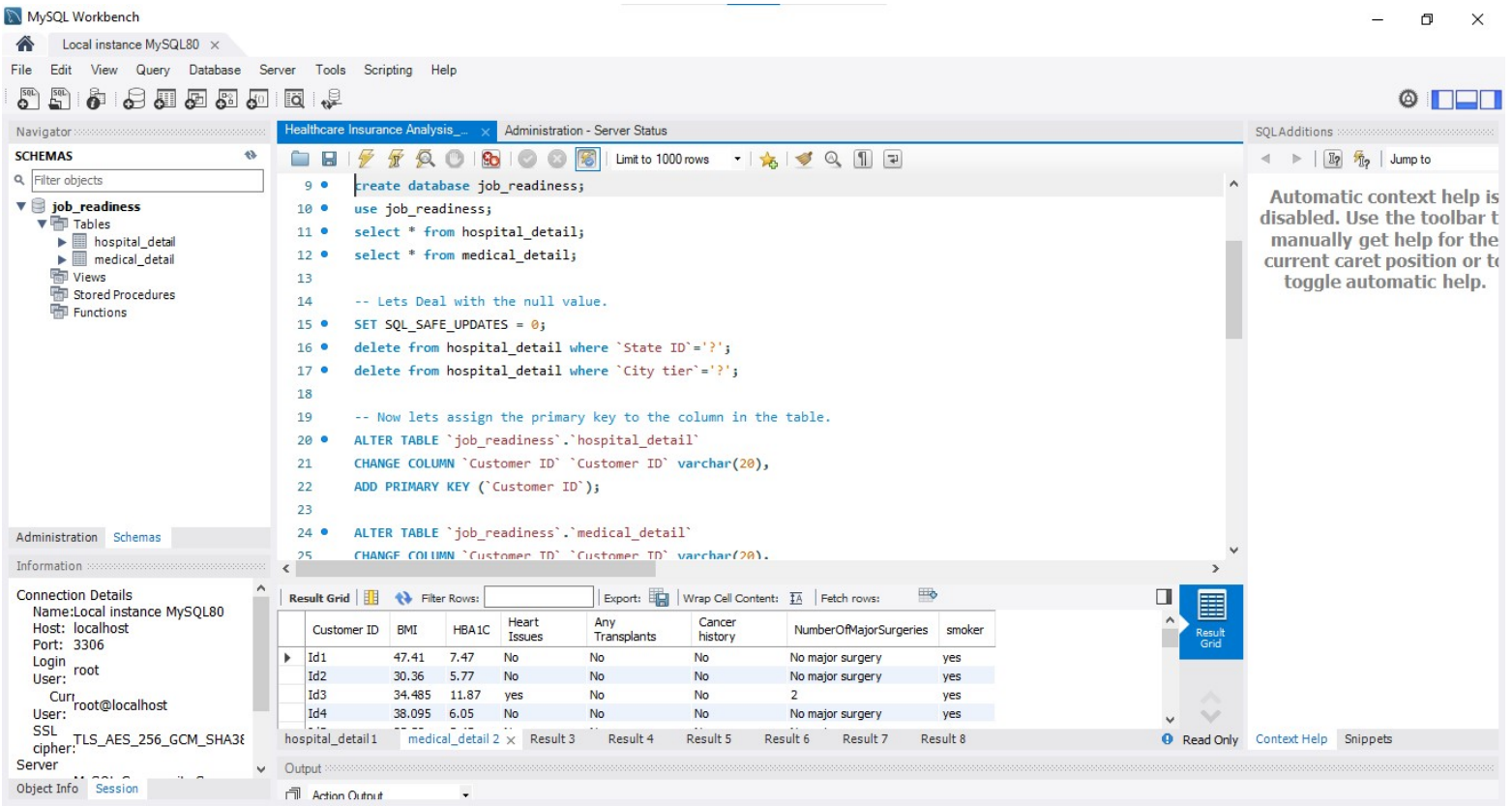
In [ ]:

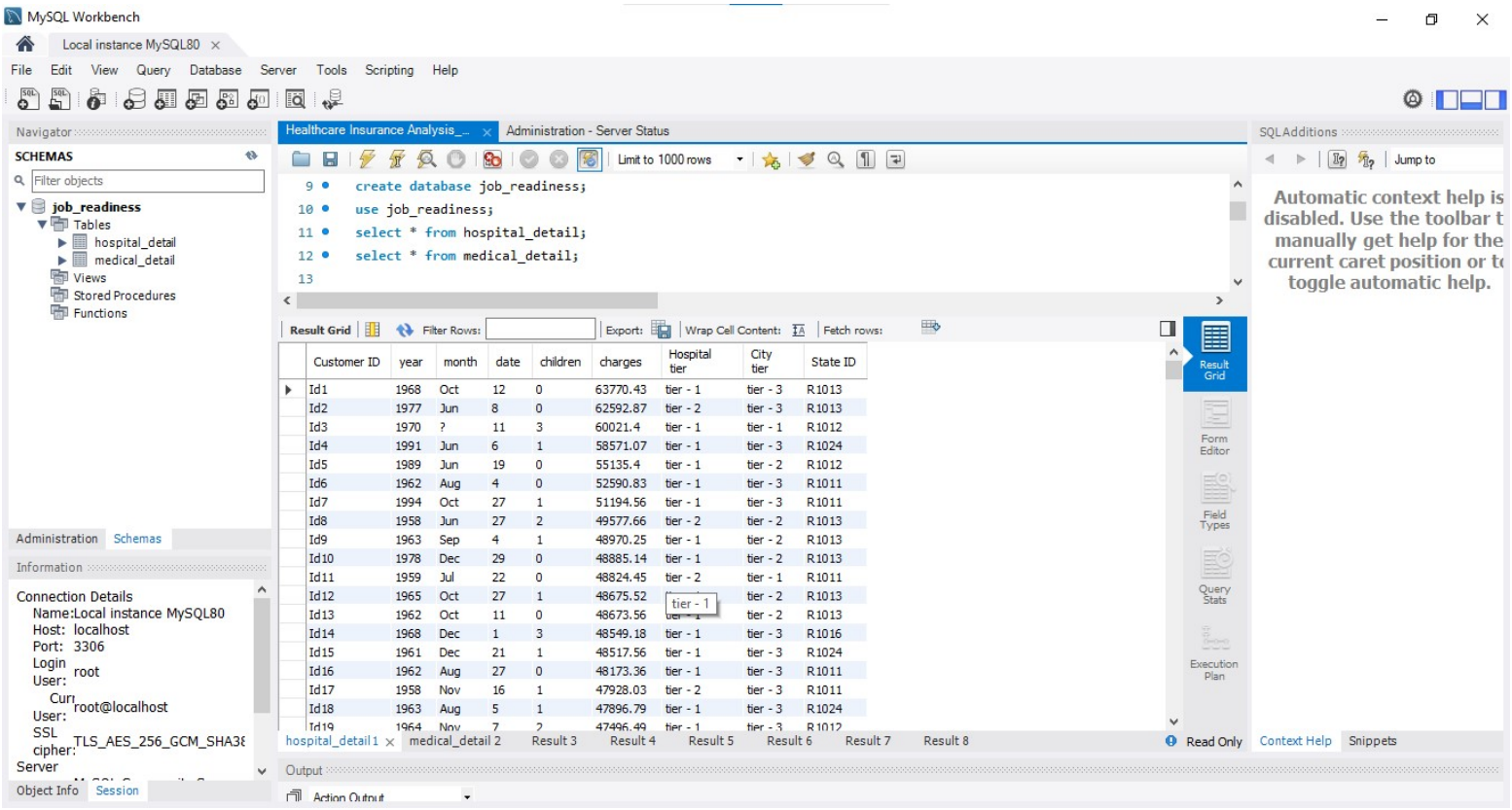
Project Task: Week 2 - SQL

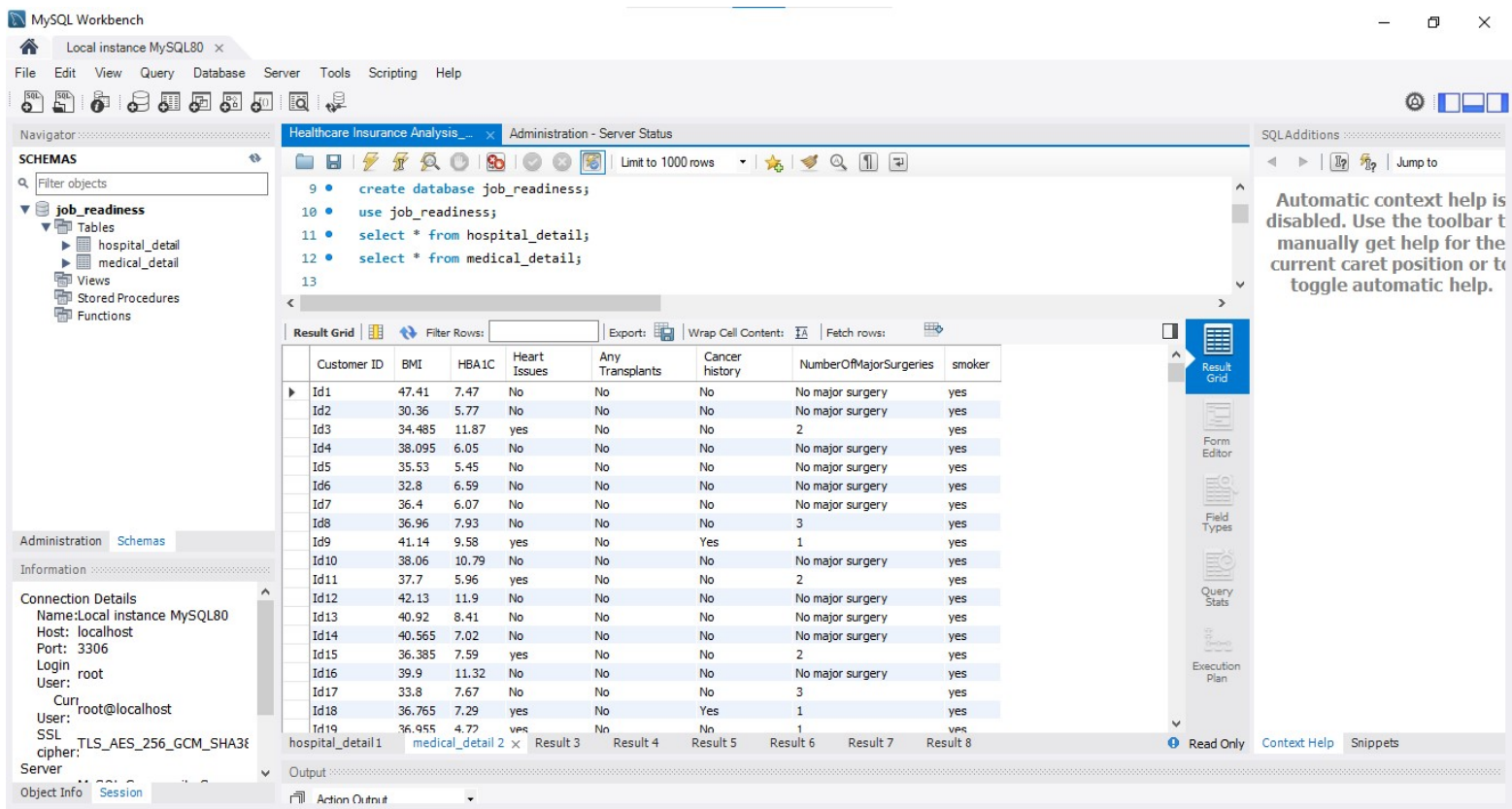
```sql
/* Question No:-1. To gain a comprehensive understanding of the factors influencing
hospitalization costs, it is necessary to combine the tables provided. Merge the two tables
by first identifying the columns in the data tables that will help you in merging.
a. In both tables, add a Primary Key constraint for these columns */

/* Hint: You can remove duplicates and null values from the column and then use ALTER TABLE
to add a Primary Key
constraint. */

create database job_readiness;
use job_readiness;
select * from hospital_detail;
select * from medical_detail;

-- Lets Deal with the null value.
SET SQL_SAFE_UPDATES = 0;
delete from hospital_detail where `State ID`='?';
delete from hospital_detail where `City tier`='?';

-- Now lets assign the primary key to the column in the table.
ALTER TABLE `job_readiness`.`hospital_detail`
CHANGE COLUMN `Customer ID` `Customer ID` varchar(20),
ADD PRIMARY KEY (`Customer ID`);

ALTER TABLE `job_readiness`.`medical_detail`
CHANGE COLUMN `Customer ID` `Customer ID` varchar(20),
ADD PRIMARY KEY (`Customer ID`);

-- Now lets merge the both table for better understanding of hospitalisation cost.
select * from hospital_detail as h inner join medical_detail as m
on h.`Customer ID` = m.`Customer ID`;

/* Question No:-2. Retrieve information about people who are diabetic and have heart problems
with their average age,
 the average number of dependent children, average BMI, and average hospitalization costs */

select m.HBA1C, m.`Heart Issues`, avg(h.children), avg(m.BMI), avg(h.charges)
from medical_detail as m
inner join hospital_detail as h
on h.`Customer ID` = m.`Customer ID`
where m.HBA1C>6.5 and m.`Heart Issues`= 'yes';

/* Question NO.3:- Find the average hospitalization cost for each hospital tier and each city
level.*/

select `Hospital tier`, avg(charges) as avg_cost from hospital_detail group by `Hospital
tier`;
select `City tier`, avg(charges) as avg_cost from hospital_detail group by `City tier`;
```

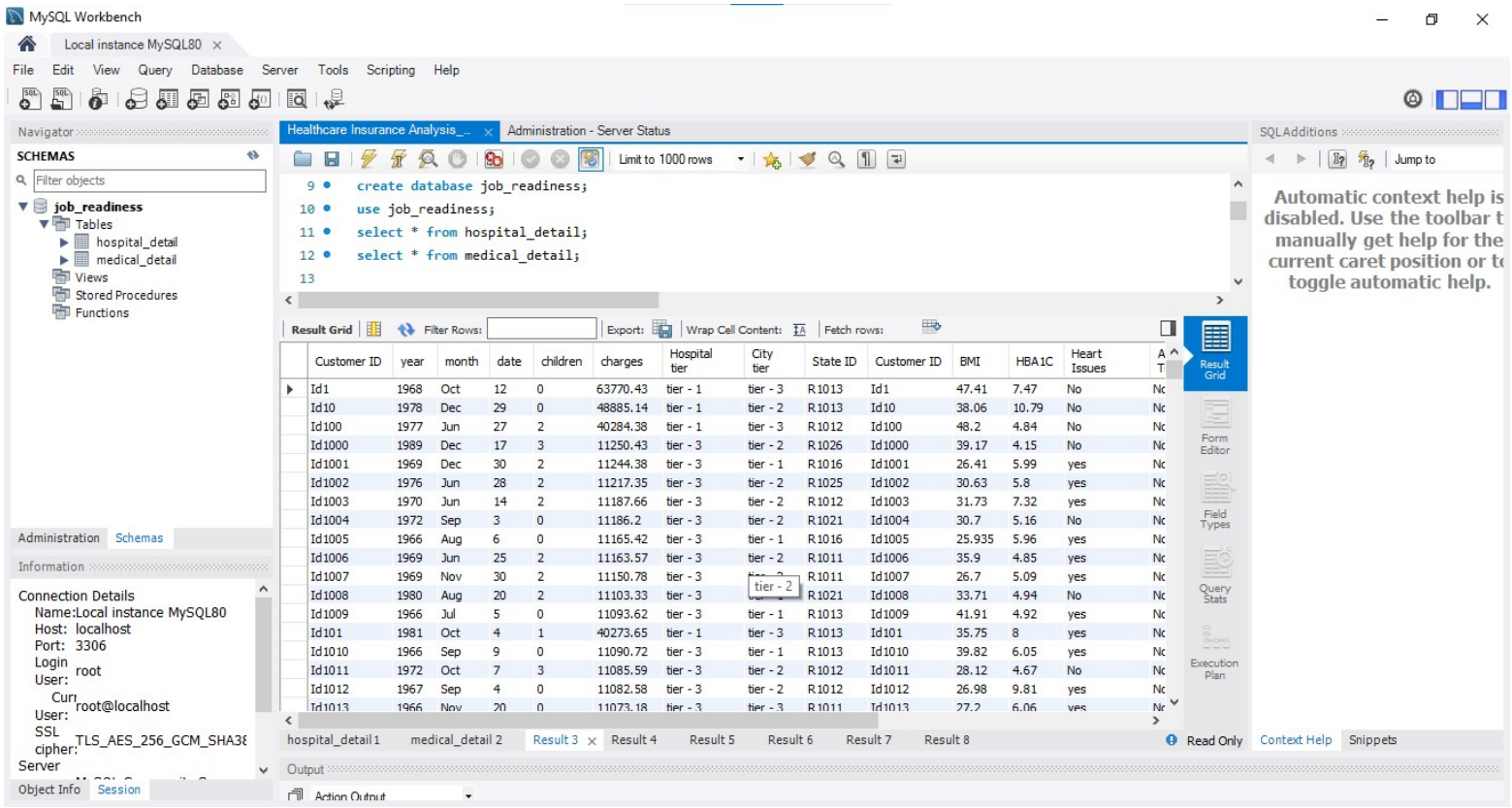
```sql
/* Question No4:- Determine the number of people who have had major surgery with a history of cancer. */

select count(`Customer ID`) from medical_detail where `Cancer history`='Yes' and NumberOfMajorSurgeries>0;

/* Question No5:- Determine the number of tier-1 hospitals in each state. */

select `State ID`, count(`Hospital tier`) from hospital_detail where `Hospital tier`='tier - 1' group by `State ID`;
```

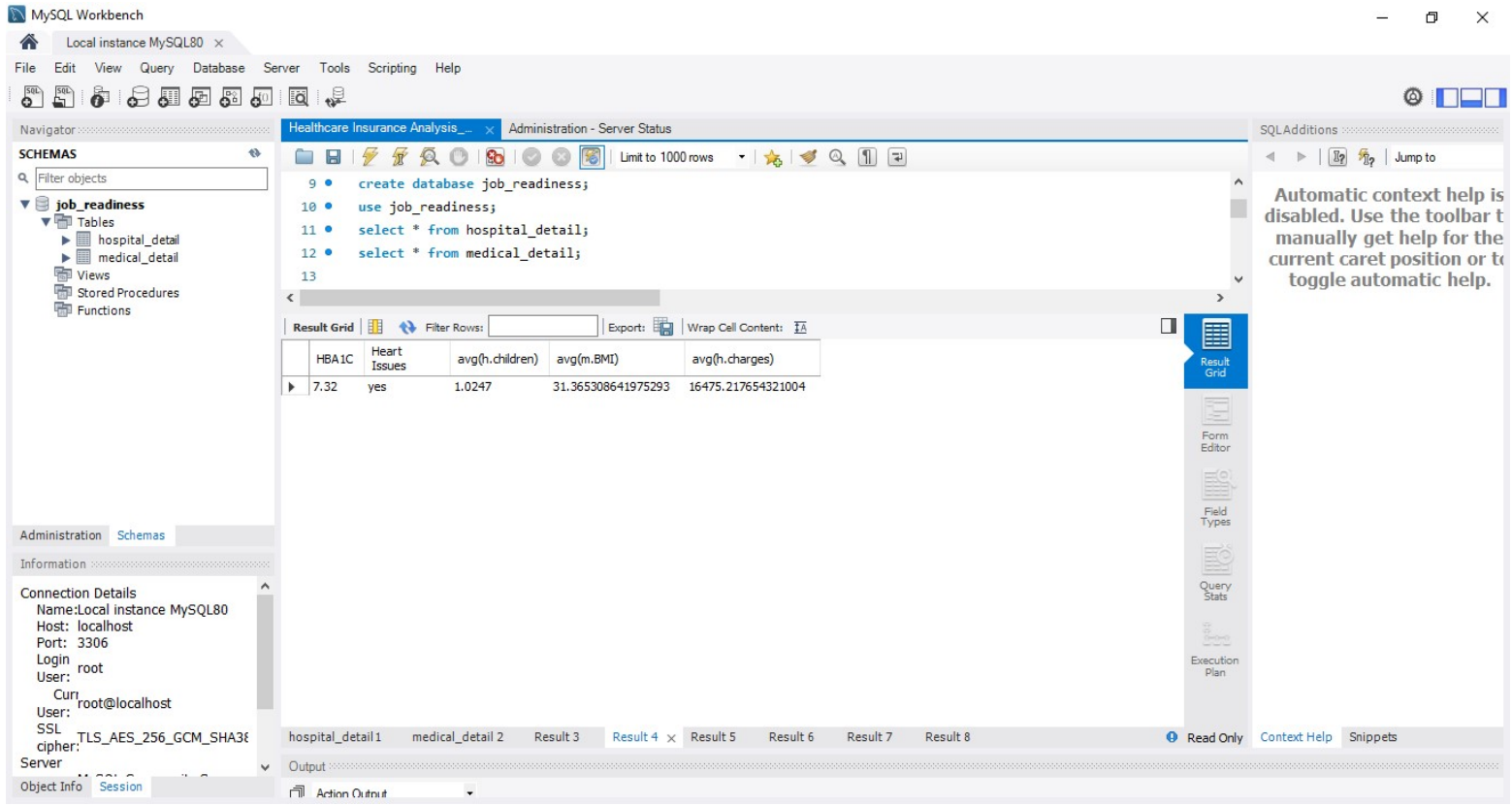
Screenshots of Output

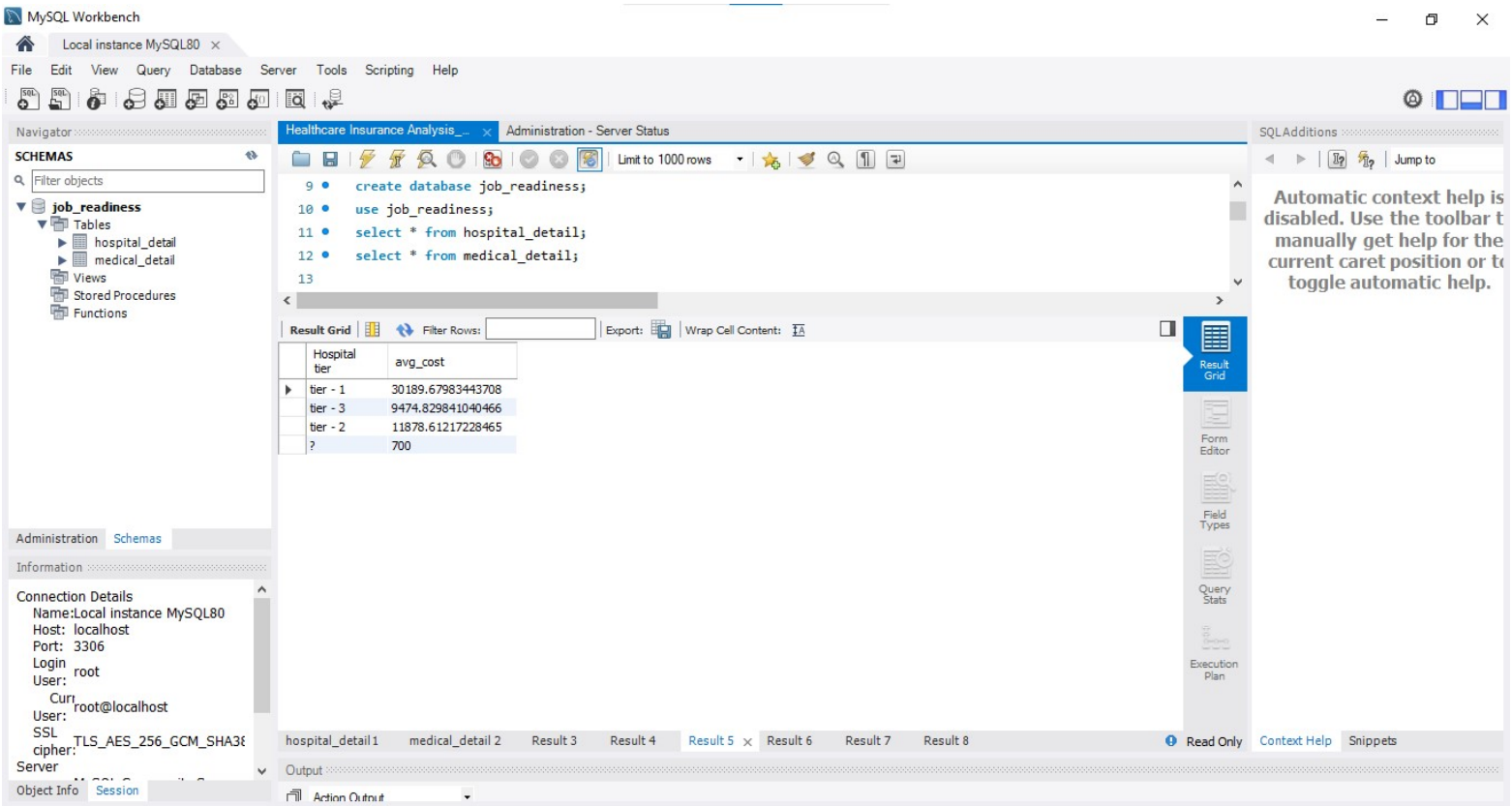


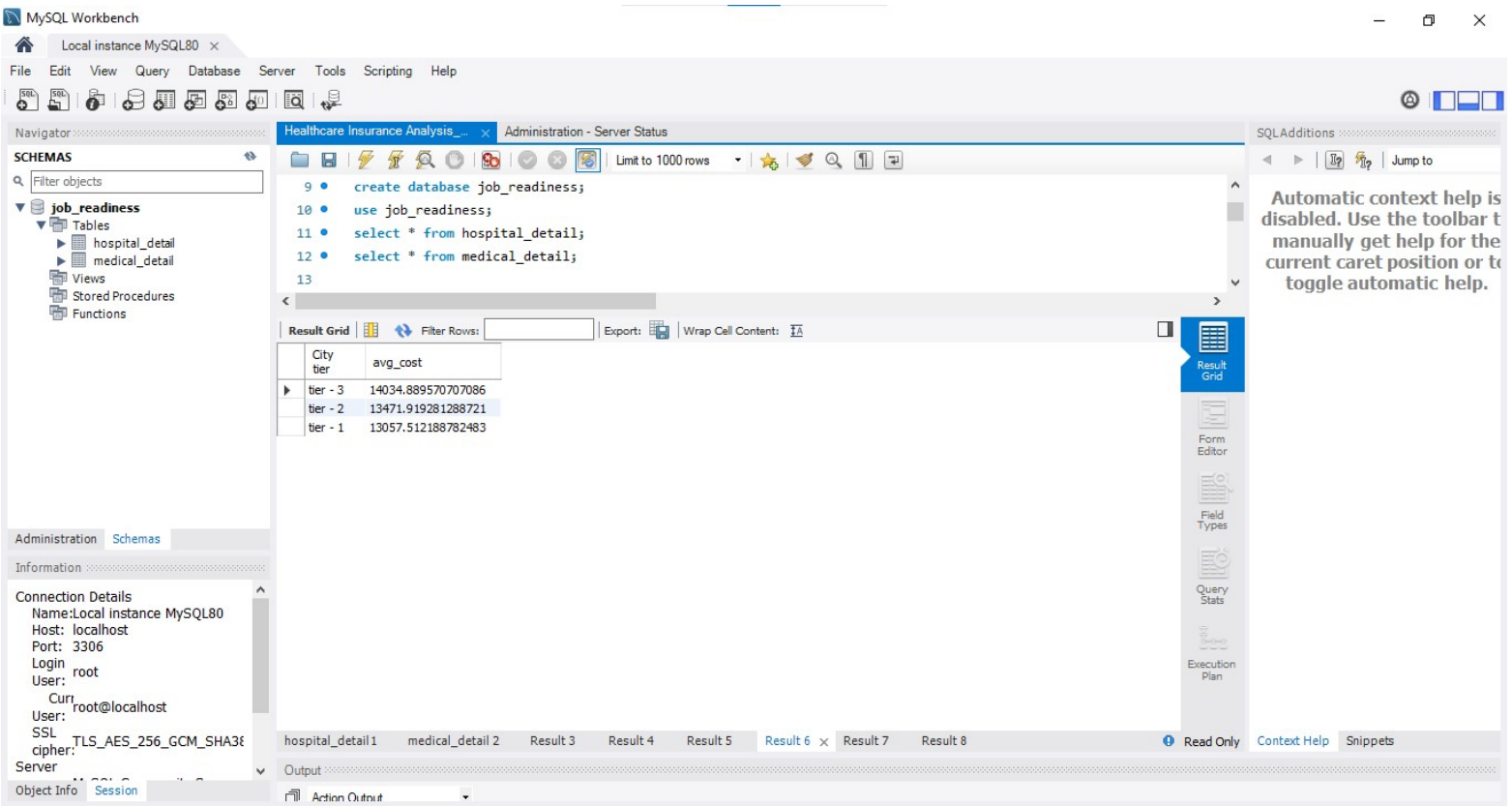

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
Customer ID year month date children charges Hospital tier City tier State ID
Id1 1968 Oct 12 0 63770.43 tier - 3 R1013
Id2 1977 Jun 8 0 62592.87 tier - 2 tier - 3 R1013
Id3 1970 ? 11 3 60021.4 tier - 1 tier - 1 R1012
Id4 1991 Jun 6 1 58571.07 tier - 1 tier - 3 R1024
Id5 1989 Jun 19 0 55135.4 tier - 1 tier - 2 R1012
Id6 1962 Aug 4 0 52590.83 tier - 1 tier - 3 R1011
Id7 1994 Oct 27 1 51194.56 tier - 1 tier - 3 R1011
Id8 1958 Jun 27 2 49577.66 tier - 2 tier - 2 R1013
Id9 1963 Sep 4 1 48970.25 tier - 1 tier - 2 R1013
Id10 1978 Dec 29 0 48885.14 tier - 1 tier - 2 R1013
Id11 1959 Jul 22 0 48824.45 tier - 2 tier - 1 R1011
Id12 1965 Oct 27 1 48675.52 tier - 2 R1013
Id13 1962 Oct 11 0 48673.56 tier - 2 R1013
Id14 1968 Dec 1 3 48549.18 tier - 1 tier - 3 R1016
Id15 1961 Dec 21 1 48517.56 tier - 1 tier - 3 R1024
Id16 1962 Aug 27 0 48173.36 tier - 1 tier - 3 R1011
Id17 1958 Nov 16 1 47928.03 tier - 2 tier - 3 R1011
Id18 1963 Aug 5 1 47896.79 tier - 1 tier - 3 R1024
tier - 1
hospital_detail 1
medical_detail 2
Result 3 Result 4 Result 5 Result 6 Result 7 Result 8
Read Only
Output
Action Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

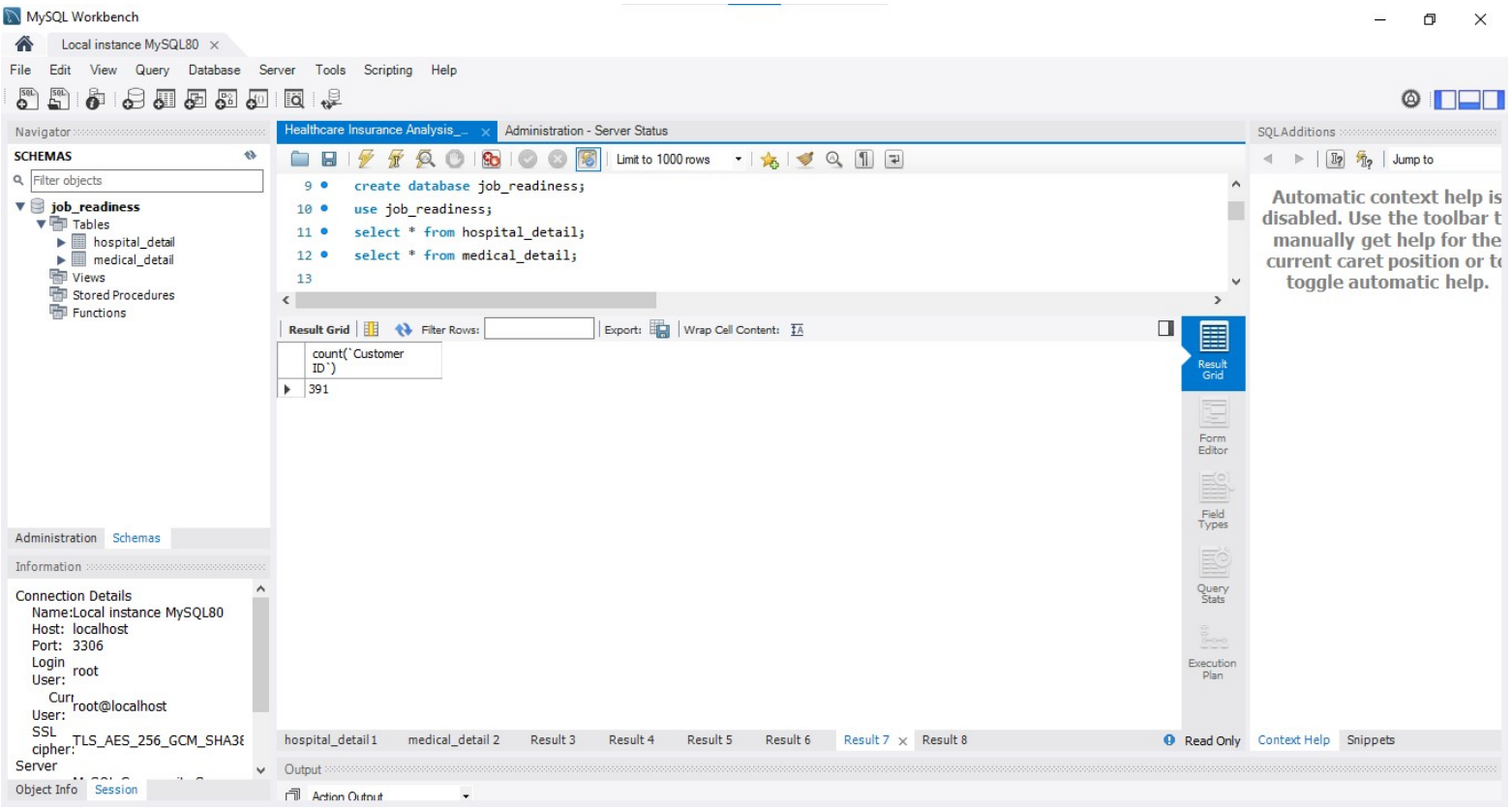

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
Customer ID BMI HBA1C Heart Issues Any Transplants Cancer history NumberOfMajorSurgeries smoker
Id1 47.41 7.47 No No No No major surgery yes
Id2 30.36 5.77 No No No No major surgery yes
Id3 34.485 11.87 yes No No 2 yes
Id4 38.095 6.05 No No No No major surgery yes
Id5 35.53 5.45 No No No No major surgery yes
Id6 32.8 6.59 No No No No major surgery yes
Id7 36.4 6.07 No No No No major surgery yes
Id8 36.96 7.93 No No No 3 yes
Id9 41.14 9.58 yes No Yes 1 yes
Id10 38.06 10.79 No No No No major surgery yes
Id11 37.7 5.96 yes No No 2 yes
Id12 42.13 11.9 No No No No major surgery yes
Id13 40.92 8.41 No No No No major surgery yes
Id14 40.565 7.02 No No No No major surgery yes
Id15 36.385 7.59 yes No No 2 yes
Id16 39.9 11.32 No No No No major surgery yes
Id17 33.8 7.67 No No No 3 yes
Id18 36.765 7.29 yes No Yes 1 yes
hospital_detail 1
medical_detail 2
Result 3 Result 4 Result 5 Result 6 Result 7 Result 8
Read Only
Output
Action Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

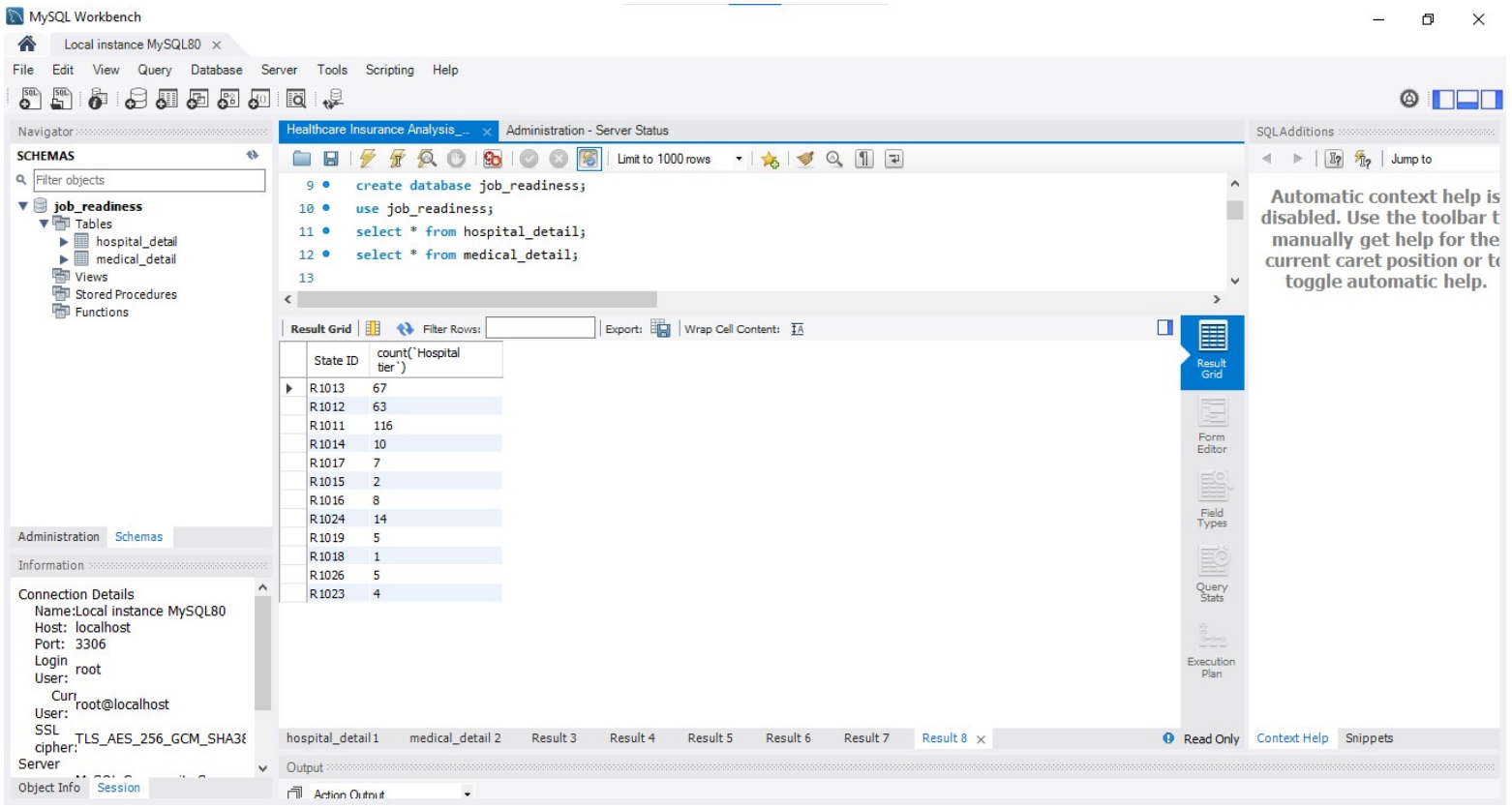

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Cur User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_... Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
Customer ID year month date children charges Hospital tier City tier State ID Customer ID BMI HBA1C Heart Issues
Id1 1968 Oct 12 0 63770.43 tier - 1 tier - 3 R1013 Id1 47.41 7.47 No
Id10 1978 Dec 29 0 48885.14 tier - 1 tier - 2 R1013 Id10 38.06 10.79 No
Id100 1977 Jun 27 2 40284.38 tier - 1 tier - 3 R1012 Id100 48.2 4.84 No
Id1000 1989 Dec 17 3 11250.43 tier - 3 tier - 2 R1026 Id1000 39.17 4.15 No
Id1001 1969 Dec 30 2 11244.38 tier - 3 tier - 1 R1016 Id1001 26.41 5.99 yes
Id1002 1976 Jun 28 2 11217.35 tier - 3 tier - 2 R1025 Id1002 30.63 5.8 yes
Id1003 1970 Jun 14 2 11187.66 tier - 3 tier - 2 R1012 Id1003 31.73 7.32 yes
Id1004 1972 Sep 3 0 11186.2 tier - 3 tier - 2 R1021 Id1004 30.7 5.16 No
Id1005 1966 Aug 6 0 11165.42 tier - 3 tier - 1 R1016 Id1005 25.935 5.96 yes
Id1006 1969 Jun 25 2 11163.57 tier - 3 tier - 2 R1011 Id1006 35.9 4.85 yes
Id1007 1969 Nov 30 2 11150.78 tier - 3 tier - 2 R1011 Id1007 26.7 5.09 yes
Id1008 1980 Aug 20 2 11103.33 tier - 3 tier - 2 R1021 Id1008 33.71 4.94 No
Id1009 1966 Jul 5 0 11093.62 tier - 3 tier - 1 R1013 Id1009 41.91 4.92 yes
Id101 1981 Oct 4 1 40273.65 tier - 1 tier - 3 R1013 Id101 35.75 8 yes
Id1010 1966 Sep 9 0 11090.72 tier - 3 tier - 1 R1013 Id1010 39.82 6.05 yes
Id1011 1972 Oct 7 3 11085.59 tier - 3 tier - 2 R1012 Id1011 28.12 4.67 No
Id1012 1967 Sep 4 0 11082.58 tier - 3 tier - 2 R1012 Id1012 26.98 9.81 yes
Id1013 1966 Nov 20 0 11073.18 tier - 3 tier - 3 R1011 Id1013 27.2 6.06 yes
tier - 2
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1 medical_detail 2 Result 3 Result 4 Result 5 Result 6 Result 7 Result 8
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Cur User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_... Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid Filter Rows: Export: Wrap Cell Content:
HBA1C Heart Issues avg(h.children) avg(m.BMI) avg(h.charges)
7.32 yes 1.0247 31.365308641975293 16475.217654321004
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1 medical_detail 2 Result 3 Result 4 Result 5 Result 6 Result 7 Result 8
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output

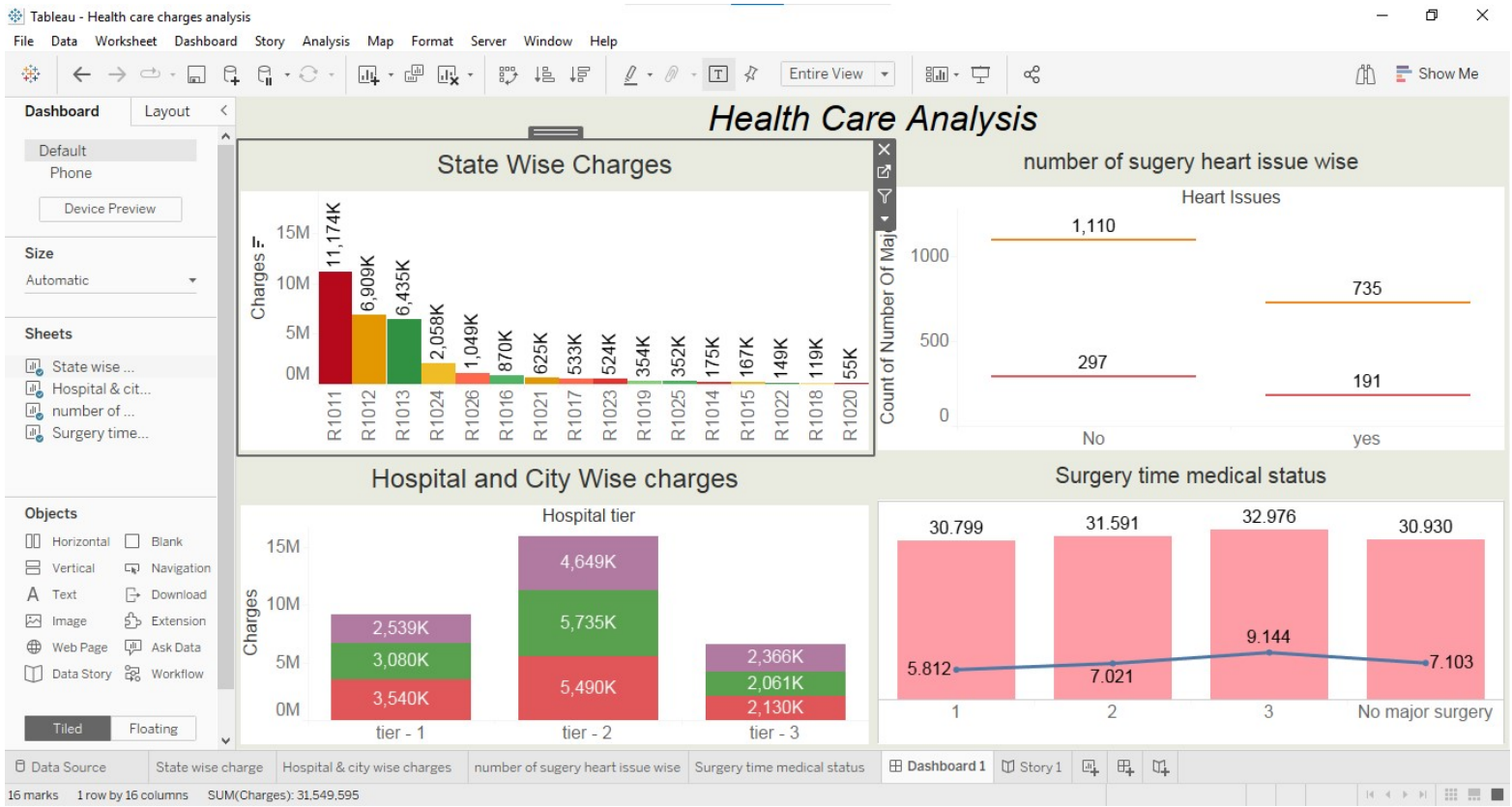

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Cur User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA3
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Hospital tier
avg_cost
tier - 1 30189.67983443708
tier - 3 9474.829841040466
tier - 2 11878.61217228465
? 700
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1
medical_detail 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Output
Action Output

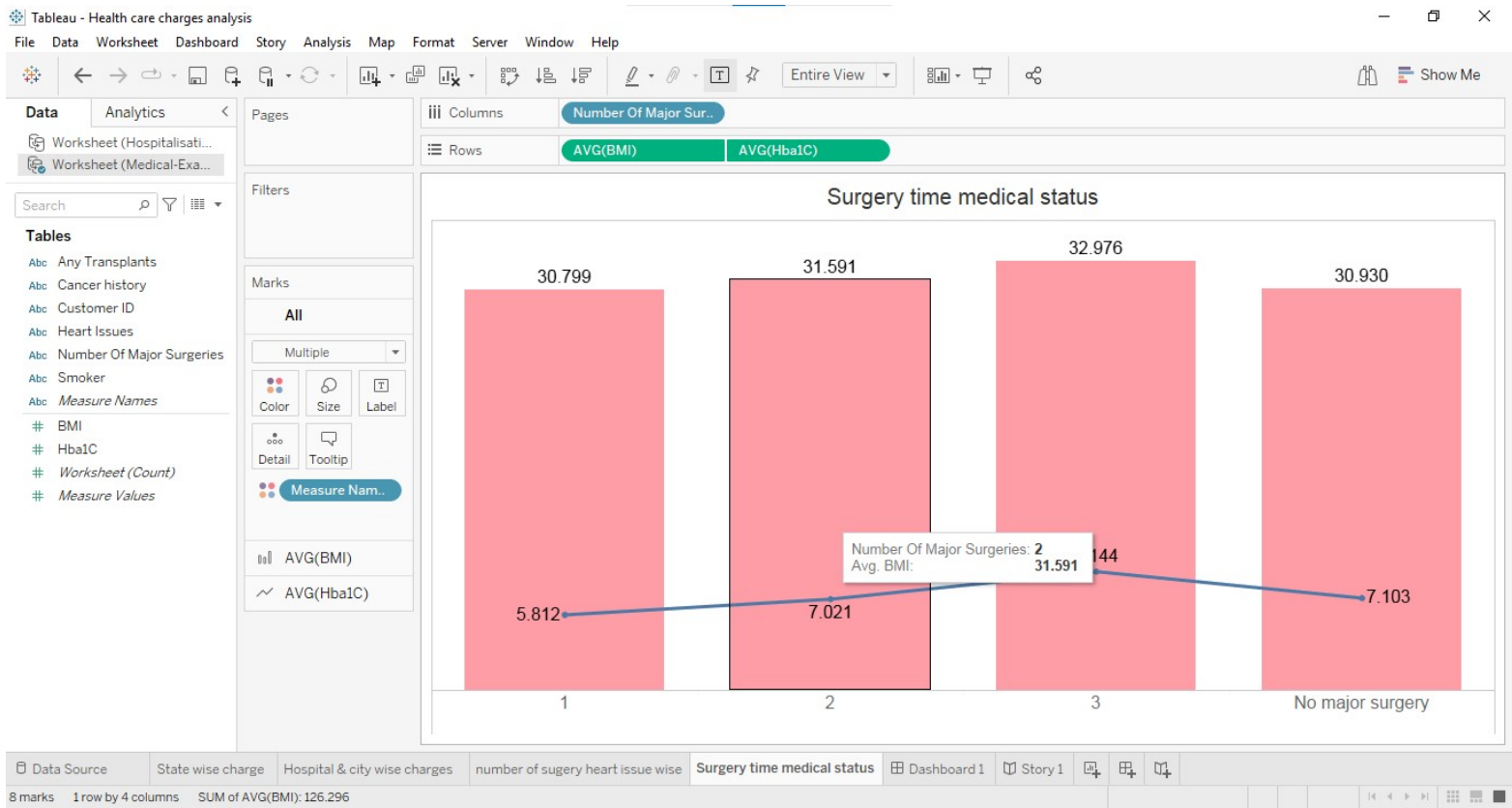

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Cur User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA3
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
City tier
avg_cost
tier - 3 14034.889570707086
tier - 2 13471.919281288721
tier - 1 13057.512188782483
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1
medical_detail 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Output
Action Output


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
count(`Customer ID`)
391
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1
medical_detail 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Read Only
Output
Action Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
job_readiness
Tables
hospital_detail
medical_detail
Views
Stored Procedures
Functions
Administration Schemas
Information
Connection Details
Name:Local instance MySQL80
Host: localhost
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Object Info Session
Healthcare Insurance Analysis_...
Administration - Server Status
Limit to 1000 rows
9 create database job_readiness;
10 use job_readiness;
11 select * from hospital_detail;
12 select * from medical_detail;
13
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
State ID
count(`Hospital tier`)
R1013 67
R1012 63
R1011 116
R1014 10
R1017 7
R1015 2
R1016 8
R1024 14
R1019 5
R1018 1
R1026 5
R1023 4
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
hospital_detail 1
medical_detail 2
Result 3
Result 4
Result 5
Result 6
Result 7
Result 8
Read Only
Output
Action Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets

Project Task: Week 2- Tableau





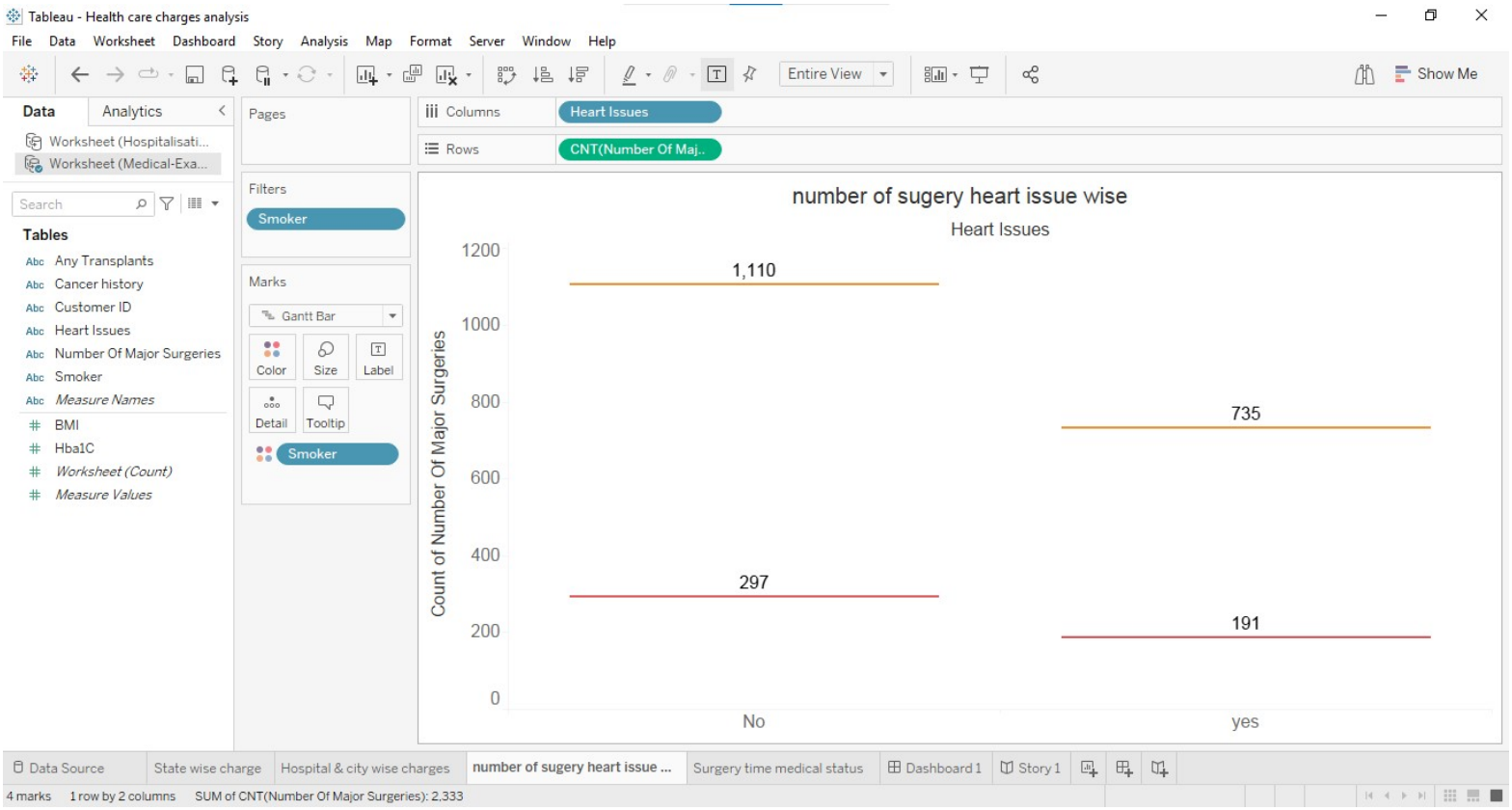


Tableau - Health care charges analysis
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Entire View
Show Me
Data
Analytics
Worksheet (Hospitalisati...
Worksheet (Medical-Exa...
Search
Tables
Abc Any Transplants
Abc Cancer history
Abc Customer ID
Abc Heart Issues
Abc Number Of Major Surgeries
Abc Smoker
Abc Measure Names
BMI
Hba1C
Worksheet (Count)
Measure Values
Pages
Filters
Smoker
Marks
Gantt Bar
Color
Size
Label
Detail
Tooltip
Smoker
Columns
Heart Issues
Rows
CNT(Number Of Maj..
number of sugery heart issue wise
Heart Issues
Count of Number Of Major Surgeries
1200
1000
800
600
400
200
0
1,110
297
735
191
No
yes
Data Source
State wise charge
Hospital & city wise charges
number of sugery heart issue ...
Surgery time medical status
Dashboard 1
Story 1
4 marks
1 row by 2 columns
SUM of CNT(Number Of Major Surgeries): 2,333

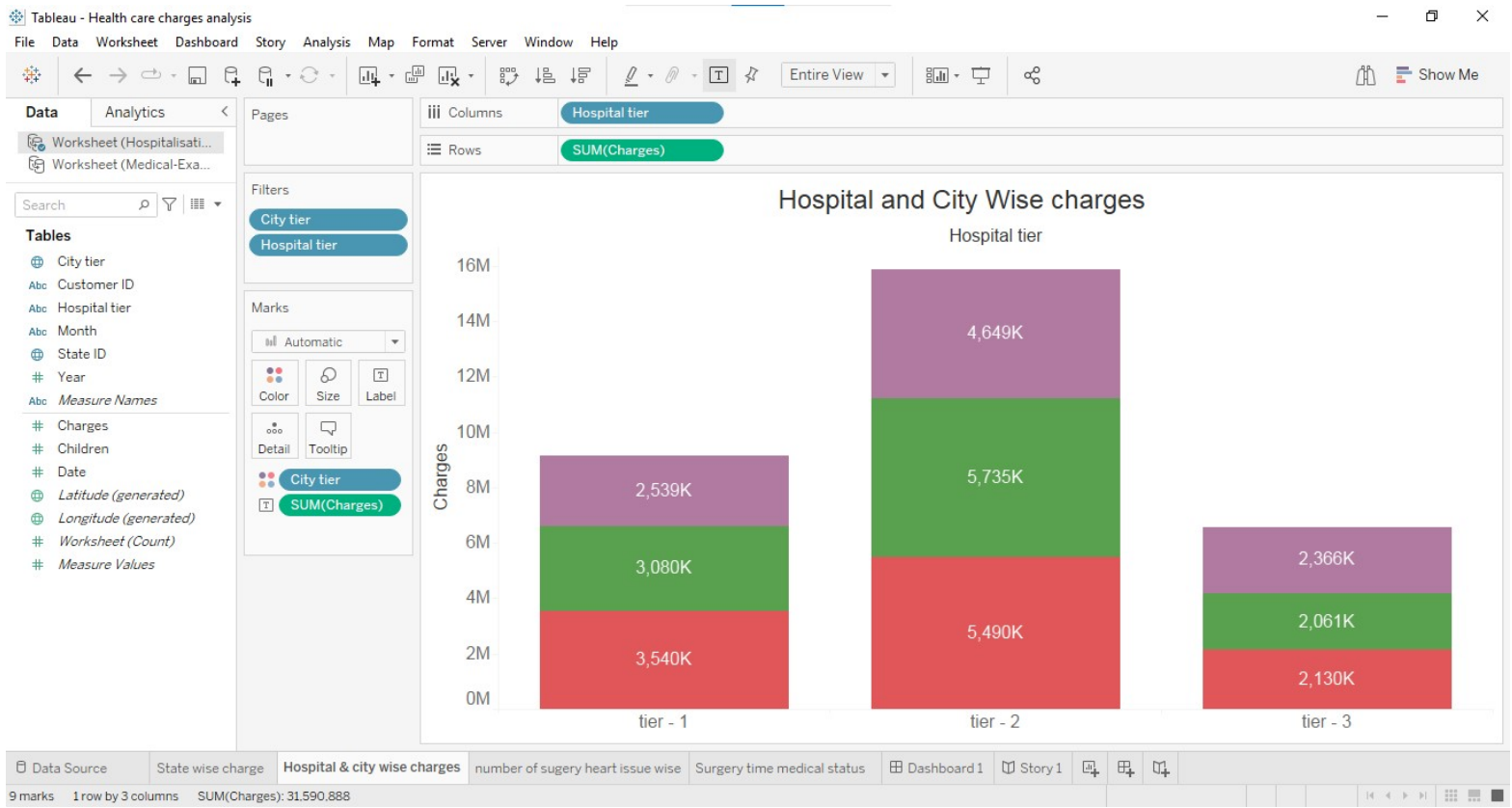


Tableau - Health care charges analysis
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Entire View
Show Me
Data
Analytics
Worksheet (Hospitalisati...
Worksheet (Medical-Exa...
Search
Tables
City tier
Abc Customer ID
Abc Hospital tier
Abc Month
State ID
Year
Abc Measure Names
Charges
Children
Date
Latitude (generated)
Longitude (generated)
Worksheet (Count)
Measure Values
Pages
Filters
City tier
Hospital tier
Marks
Automatic
Color
Size
Label
Detail
Tooltip
City tier
SUM(Charges)
Columns
Hospital tier
Rows
SUM(Charges)
Hospital and City Wise charges
Hospital tier
Charges
16M
14M
12M
10M
8M
6M
4M
2M
0M
2,539K
3,080K
3,540K
4,649K
5,735K
5,490K
2,366K
2,061K
2,130K
tier - 1
tier - 2
tier - 3
Data Source
State wise charge
Hospital & city wise charges
number of sugery heart issue wise
Surgery time medical status
Dashboard 1
Story 1
9 marks
1 row by 3 columns
SUM(Charges): 31,590,888

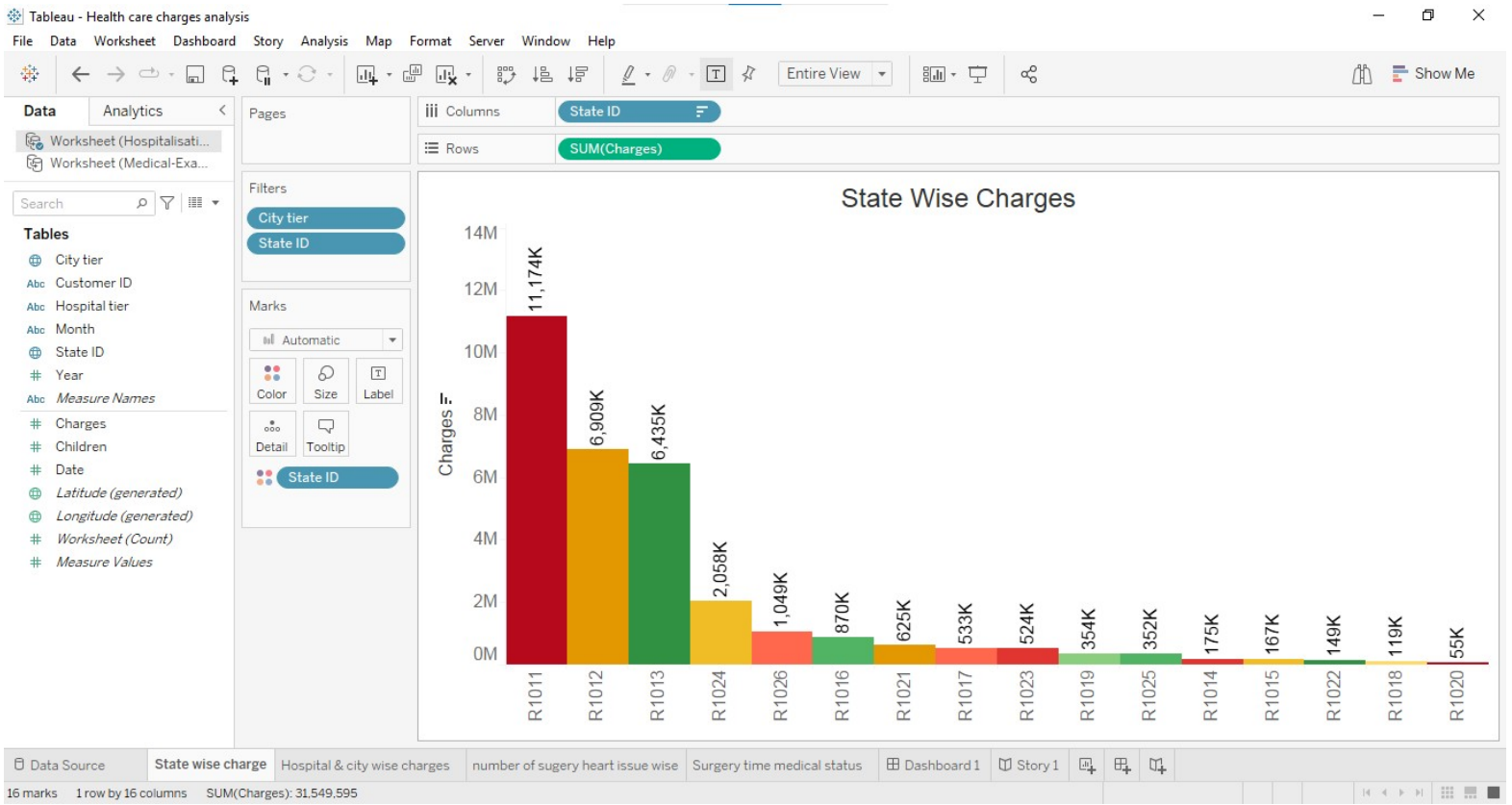


Tableau - Health care charges analysis
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Entire View
Show Me
Data
Analytics
Worksheet (Hospitalisati...
Worksheet (Medical-Exa...
Search
Tables
City tier
Customer ID
Hospital tier
Month
State ID
Year
Measure Names
Charges
Children
Date
Latitude (generated)
Longitude (generated)
Worksheet (Count)
Measure Values
Pages
Columns
State ID
Rows
SUM(Charges)
Filters
City tier
State ID
Marks
Automatic
Color
Size
Label
Detail
Tooltip
State ID
State Wise Charges
Charges
14M
12M
10M
8M
6M
4M
2M
0M
11,174K
6,909K
6,435K
2,058K
1,049K
870K
625K
533K
524K
354K
352K
175K
167K
149K
119K
55K
R1011
R1012
R1013
R1024
R1026
R1016
R1021
R1017
R1023
R1019
R1025
R1014
R1015
R1022
R1018
R1020
Data Source
State wise charge
Hospital & city wise charges
number of sugery heart issue wise
Surgery time medical status
Dashboard 1
Story 1
16 marks
1 row by 16 columns
SUM(Charges): 31,549,595

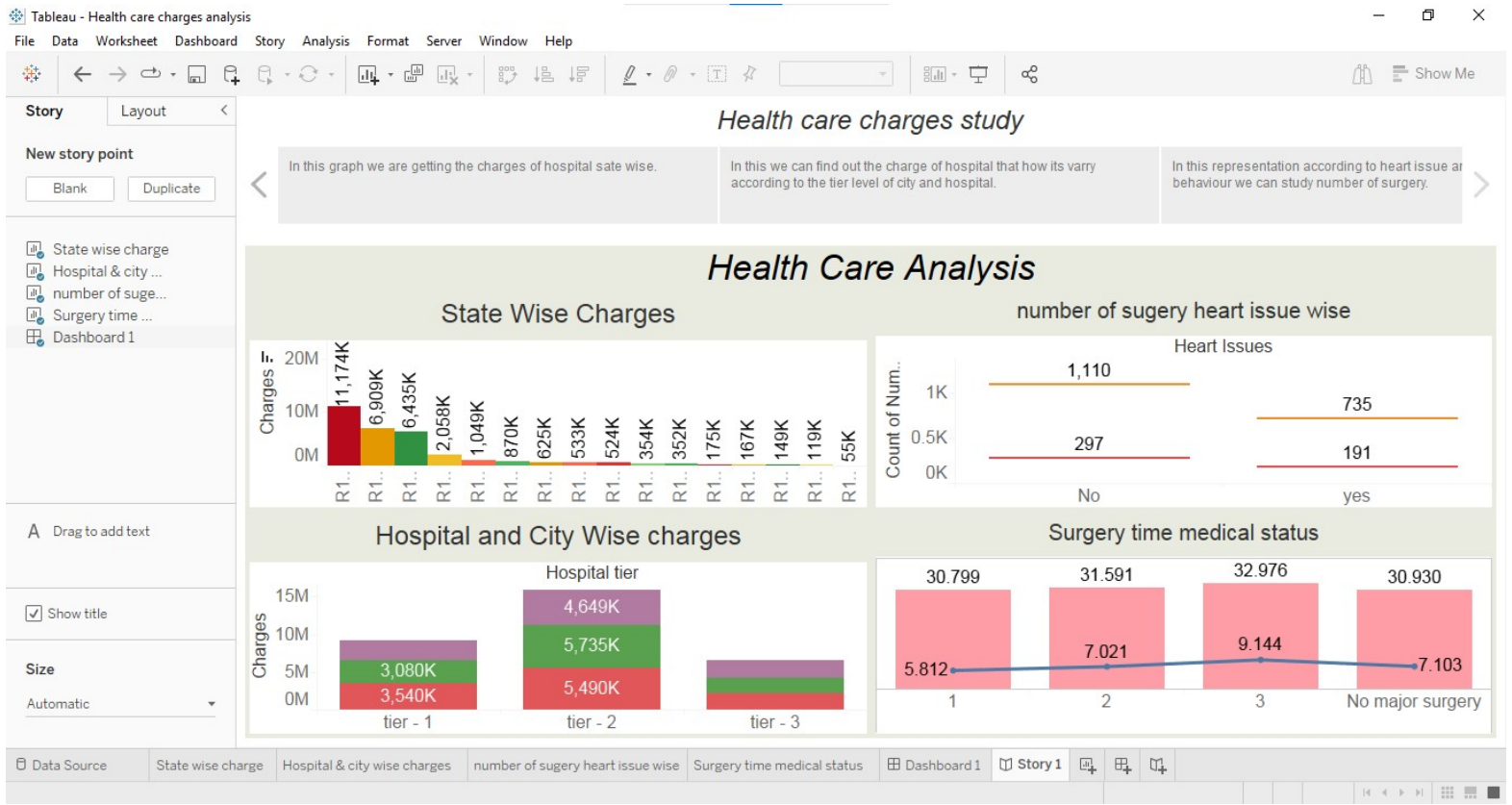


Tableau - Health care charges analysis
File Data Worksheet Dashboard Story Analysis Format Server Window Help
Show Me
Story
Layout
New story point
Blank
Duplicate
State wise charge
Hospital & city ...
number of suge...
Surgery time ...
Dashboard 1
Drag to add text
Show title
Size
Automatic
Health care charges study
In this graph we are getting the charges of hospital sate wise.
In this we can find out the charge of hospital that how its varry according to the tier level of city and hospital.
In this representation according to heart issue ar behaviour we can study number of surgery.
Health Care Analysis
State Wise Charges
Charges
20M
10M
0M
11,174K
6,909K
6,435K
2,058K
1,049K
870K
625K
533K
524K
354K
352K
175K
167K
149K
119K
55K
number of sugery heart issue wise
Heart Issues
Count of Num..
1K
0.5K
0K
1,110
297
735
191
No
yes
Hospital and City Wise charges
Hospital tier
Charges
15M
10M
5M
0M
3,080K
3,540K
4,649K
5,735K
5,490K
tier - 1
tier - 2
tier - 3
Surgery time medical status
30.799
31.591
32.976
30.930
5.812
7.021
9.144
7.103
1
2
3
No major surgery
Data Source
State wise charge
Hospital & city wise charges
number of sugery heart issue wise
Surgery time medical status
Dashboard 1
Story 1

Excel Part

hypothesis - Excel

	tier-1	tier-2	tier-3
	2302.3	5087.92	34975.68
	2721.32	5116.5	34976.42
	2867.12	5124.19	34979.86
	3490.55	5136.75	35000.73
	3645.09	5138.26	35050.62
	3866.86	5144.18	35160.13
	3877.3	5148.55	35315.96
	4441.21	5152.13	35345.73
	4449.46	5166.96	35517.19
	4504.66	5177.12	35547.47
	4646.76	5195.58	35573.26
	700	5198.69	35585.58
	5028.15	5207.97	35595.59
	5125.22	5213.22	35701.9
	5729.01	5216.48	35711.39
	6079.67	5227.99	35733.96
	6082.41	5240.77	35883.27
	6272.48	5245.23	36021.01
	6389.38	5246.05	36074.34
	6406.41	5257.51	36090.49
	6799.46	5261.47	36149.48
	6948.7	5266.37	36182.87

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
tier-1	309	9310917	30132.41906	151312085
tier-2	1339	15898789	11873.62875	127325764
tier-3	694	6558490	9450.273256	22205805.5

ANOVA

Source of Varia	SS	df	MS	F	P-value	F crit
Between (	1.0039E+11	2	50197375077	505.312362	4E-183	2.999572
Within Gr	2.3235E+11	2339	99339297.56			
Total	3.3275E+11	2341				

SUMMARY

Groups	Count	Sum	Average	Variance
tier-1	734	10621309	14470.45	2.43E+08
tier-2	812	10497978	12928.54	66791970
tier-3	796	10648839	13377.94	1.25E+08

ANOVA

ce of Varia	SS	df	MS	F	P-value	F crit
Between (	9.59E+08	2	4.79E+08	3.378633	0.034261	2.999572
Within Gr	3.32E+11	2339	1.42E+08			
Total	3.33E+11	2341				

Hospitalisation details

Ready

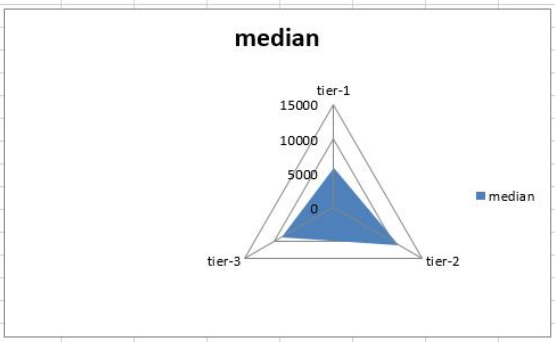
excel part - Excel

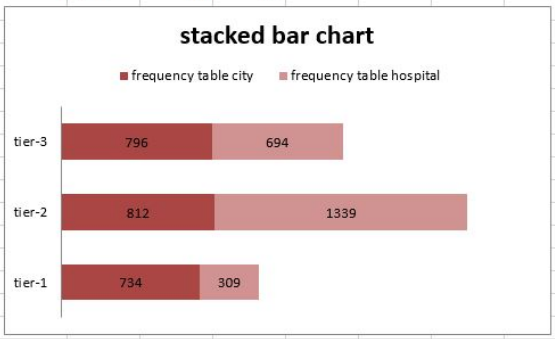
charges	Hospita	City tie	State ID
3866.86	tier - 1	tier - 1	R1012
4646.76	tier - 1	tier - 1	R1011
6079.67	tier - 1	tier - 1	R1014
7740.34	tier - 1	tier - 1	R1011
17479.53	tier - 1	tier - 1	R1012
17507.47	tier - 1	tier - 1	R1026
17560.38	tier - 1	tier - 1	R1012
17748.51	tier - 1	tier - 1	R1012
18033.97	tier - 1	tier - 1	R1012
18157.88	tier - 1	tier - 1	R1012
18218.16	tier - 1	tier - 1	R1013
18246.5	tier - 1	tier - 1	R1013
18648.42	tier - 1	tier - 1	R1019
18694.69	tier - 1	tier - 1	R1023
18767.74	tier - 1	tier - 1	R1013
18804.75	tier - 1	tier - 1	R1012
18815.53	tier - 1	tier - 1	R1023
18883.33	tier - 1	tier - 1	R1026
18954.56	tier - 1	tier - 1	R1026
18963.17	tier - 1	tier - 1	R1013
19144.58	tier - 1	tier - 1	R1024
19259.96	tier - 1	tier - 1	R1012

hospitalti	median
tier-1	5748.13
tier-2	11098.48
tier-3	8824.54

frequency table

cities/hos	city	hospital
tier-1	734	309
tier-2	812	1339
tier-3	796	694





Hospitalisation details

Enter