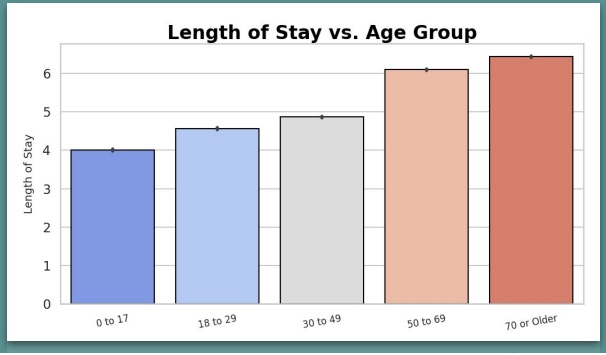
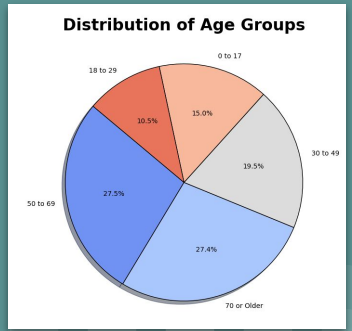
CVS Data Science

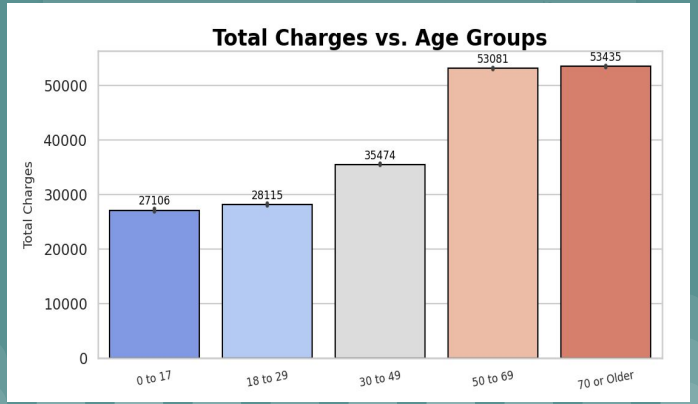
How are the total charges affected by the patient's length of stay, age, gender, race, and ethnicity?

Older age groups are more likely to be admitted into hospitals, stay there longer, and be charged more

- More than 50% of admissions were from people 50 or older.
- A patient's length of stay has a direct relationship to their age.
- Patients 50 or older were charged a significantly larger amount than younger patients.



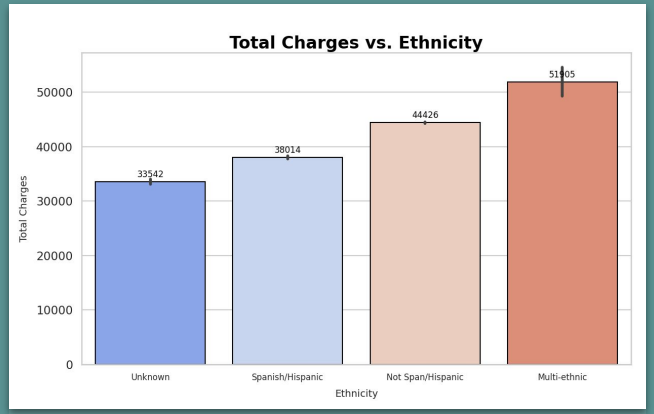


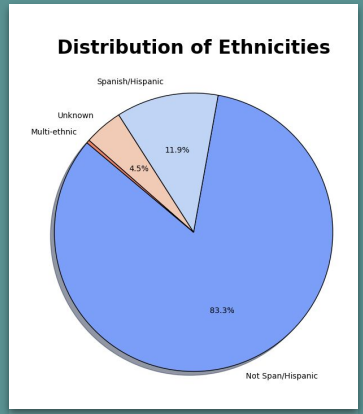


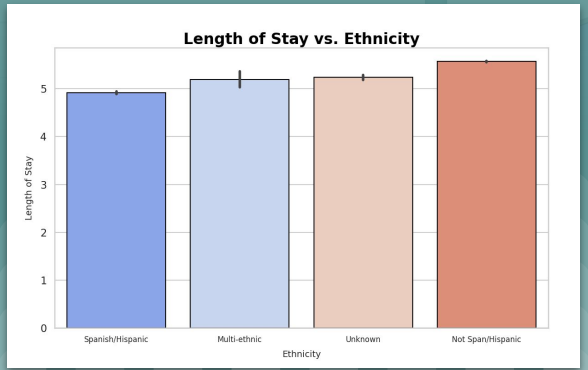
Ethnicity had little effect on length of stay, but a significant effect on total charges

- Despite making up an extremely small portion of the total admissions, Multi-ethnic people were charged the most.

- Most admissions were made up by Not Span/Hispanic people.

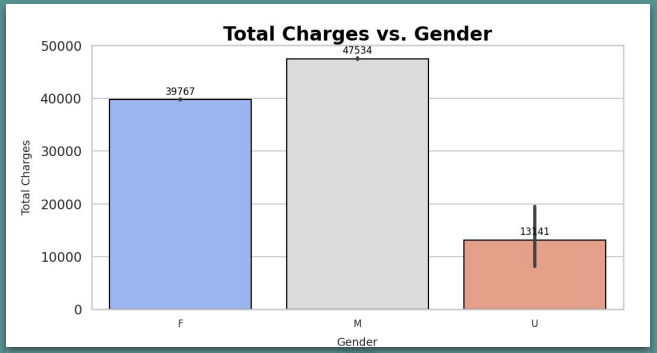


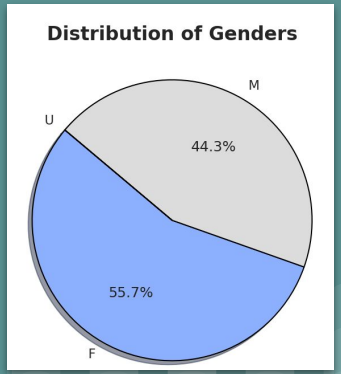


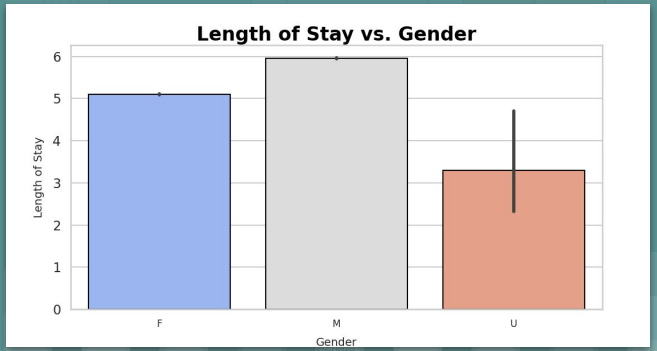


Men were affected more in total charges and length of stay than women

- Men seem to be charged around ~$7,500 more than women on average.
- Men also seem to have longer stays than women, by about a day on average.
- These results are interesting because more women were admitted in total than men.

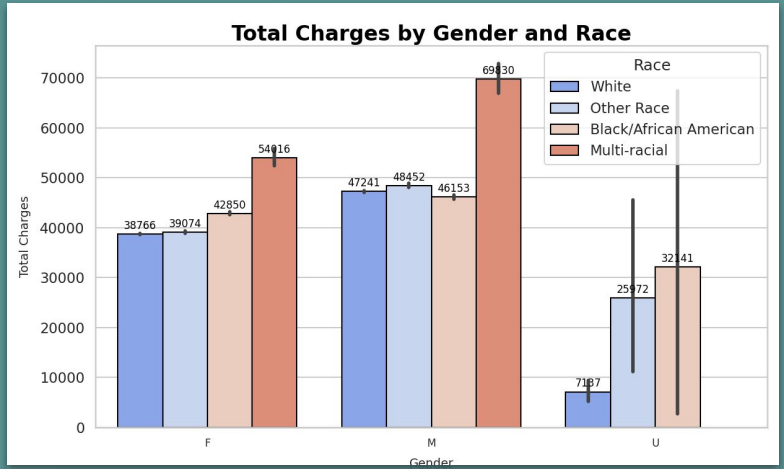


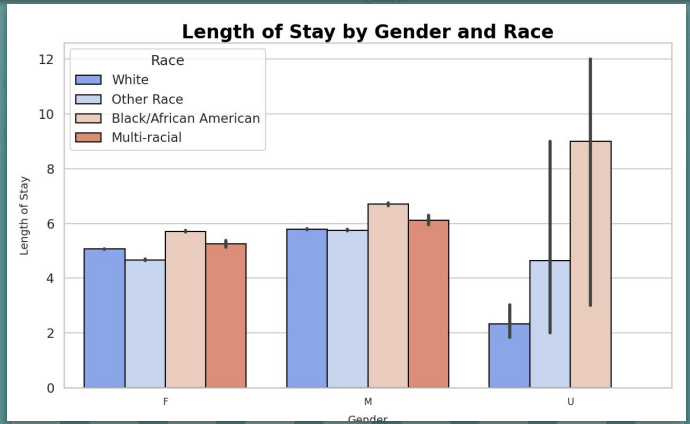




Multi-racial individuals were charged significantly more on average

- In both genders, multi-racial people were charged the most.
- All others races were charged about the same.
- Unspecified gender in Length of Stay have large error bars, indicating they likely have very differing values.





Data Cleaning involved removing nulls, casting columns, and dropping irrelevant columns

- Dropped every column that wasn't involved in our main question.
- Would hopefully make the predictive model more accurate, as it could only base it's predictions on the columns mentioned in the main question.
- Removing all nulls simplified the cleaning process significantly, and allowed for us to quickly have a dataset almost ready to create a model with.

```
df2 = df.drop(columns=['Health Service Area', 'Hospital County',
       'Operating Certificate Number', 'Facility Id', 'Facility Name',
       'Zip Code - 3 digits',
       'Type of Admission', 'Patient Disposition',
       'Discharge Year', 'CCS Diagnosis Code', 'CCS Diagnosis Description',
       'CCS Procedure Code', 'CCS Procedure Description', 'APR DRG Code',
       'APR DRG Description', 'APR MDC Code', 'APR MDC Description',
       'APR Severity of Illness Code', 'APR Severity of Illness Description',
       'APR Risk of Mortality', 'APR Medical Surgical Description',
       'Payment Typology 1', 'Payment Typology 2', 'Payment Typology 3',
       'Attending Provider License Number',
       'Operating Provider License Number', 'Other Provider License Number',
       'Birth Weiht', 'Abortion Edit Indicator',
       'Emergency Department Indicator'])
```

```
df2 = df2.dropna()
```

Feature engineering involved one hot encoding through the get_dummies method

- Final adjustments consisted of formatting the Total Charges & Total Costs columns as numbers by removing their dollar sign, then casting them.

- get_dummies easily converted the categorical columns into numerical columns usable by models.

```
df2["Total Charges"] = df2["Total Charges"].str.replace("$", "")
df2["Total Costs"] = df2["Total Costs"].str.replace("$", "")
df2["Total Charges"] = df2["Total Charges"].astype(float)
df2["Total Costs"] = df2["Total Costs"].astype(float)
```

```
df3 = pd.get_dummies(df2,drop_first=True)
df3 = df3.dropna()
```

Model overview

- Linear Regression was chosen after we determined that it had the lowest absolute error out of our tested models.

- test_size of 0.2 allowed for 80% training and 20% testing data. This made sure the model got more of an accurate scope of the dataset as a whole.

```
Y = df3["Total Charges"]
X = df3.drop("Total Charges", axis=1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)

regressor = LinearRegression()
regressor.fit(X_train, Y_train)

Y_pred_train = regressor.predict(X_train)
Y_pred_test = regressor.predict(X_test)
```

Predictions varied. Some came very close to the expected values, some were extremely different.

- The training and testing results were both extremely similar.
- Some of the model's predictions came extremely close to the real value, such as the first row in the training table. However, some others were not nearly as accurate, such as the ~$100,000 prediction with the ~$200,000 real value.

```
pd.DataFrame({"Model Prediction": Y_pred_train, "Expected/Real values": Y_train}).head(10)
```

	Model Prediction	Expected/Real values
677498	70999.154741	76372.09
102042	41438.962132	29572.76
216886	16506.115950	8919.28
47581	4441.897877	4229.01
538760	18494.929746	15050.60
911092	100613.544468	207153.05
565828	13998.880472	12117.00
382839	45530.974145	51956.81
1455920	35425.804872	35990.00
1024763	50366.500568	59027.88

```
pd.DataFrame({"Model Prediction": Y_pred_test, "Expected/Real values": Y_test}).head(10)
```

	Model Prediction	Expected/Real values
1031866	7840.525946	12039.42
460234	15905.163544	8316.32
312646	15024.354279	5651.17
1927074	25946.618079	16560.81
1964464	28114.700496	40560.15
1546131	143895.667998	153579.65
694119	72654.183234	97478.42
1972622	114500.471212	147606.84
404077	14874.054236	6065.40
1887854	24683.120945	18537.80

Absolute error of both training and testing data barely differed. Model meshes well to new unseen data.

- With only a ~15 difference in predictions in the training and testing data, our model reacts well to unseen data.
- Although the difference between data is small, the absolute error is still very large when considering it represents money.

```
mean_absolute_error(Y_train, Y_pred_train)
```

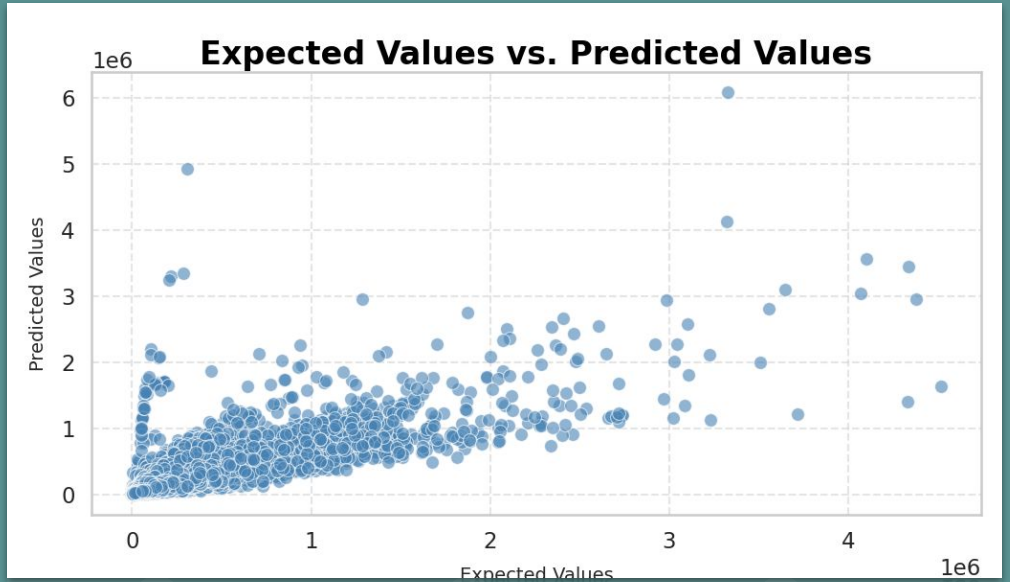
```
16744.103875314722
```

```
mean_absolute_error(Y_test, Y_pred_test)
```

```
16759.373371070757
```

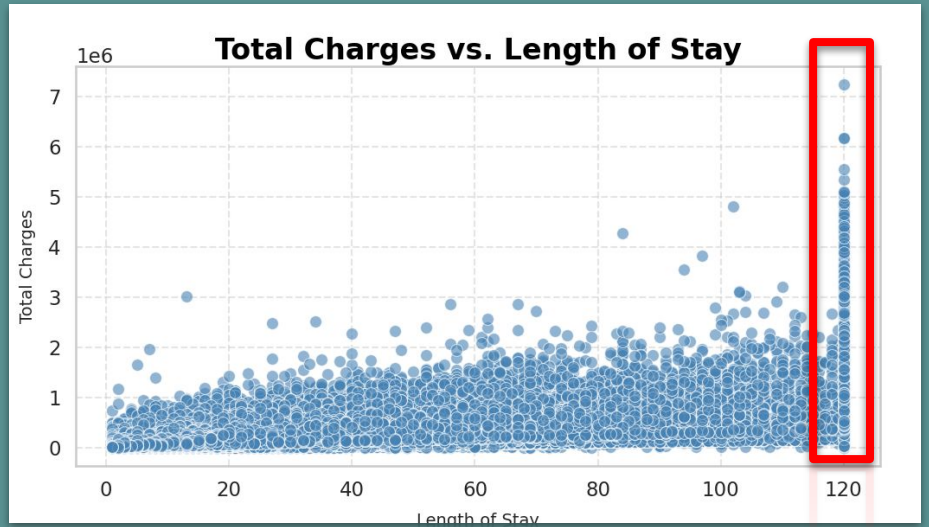
Expected Values vs. Predicted Values graph shows model generally predicting values smaller than expected.

- Preferred slope for line of best fit would be 1.
- The real line of best fit would likely show a slope of around ⅓.
- This graph tells us that our model is generally predicting values that are smaller than what the real values are.



Further and more in-depth Data Cleaning and Feature Engineering would likely produce better predictions

- Given more time and further exploration of the data, we could've optimized the dataset further for more accurate predictions.
- One change would be determining a better solution for length of stays longer than 120 days.
- We also likely could have included more of the initial columns in the dataset. For example, columns related to the hospital's location would have provided more information about the total charges.
- Could possibly even use another prediction model to determine Length of Stay based on factors such as the type of admission, discharge year, etc.




```
df.loc[df['Length of Stay'] == '120 +', 'Length of Stay'] = 120
df['Length of Stay']=df['Length of Stay'].astype(int)
```

Possible results for the use of this prediction model

- Anticipating the charges of a patient's visit could help the hospitals prepare for the impact of those costs.

- Being able to predict possible charges ahead of time could help hospitals better manage their finances, overall having a positive impact on their spending.