EDA_Report_Leah_Seseri

November 6, 2025

Explore MIMIC-III Dataset to determine What Patient Groups Tend to Stay Longer in the ICU?

ICU stay time is important as it is a factor that detemines how severe one's injury is, affects the costs of a patient's hospital stay, and could be used as a factor to predict what kind of injury one enters the hospital with. Injury severity is typically correlated with ICU stay time as a more harmful injury will require a patient to stay longer in the ICU. Other factors that could determine ICU stay time that does not necessarily depend on injury severity are age and past disease history. ICU stay time also directly affects patient costs as long ICU times are typically more coslty. ICU times can also be used to predict injury severity under the assumption that patient's with longer ICU times are there due to their injury being severe.

MIMIC-III dataset has multiple tables to help determine what patient groups stay longer in the ICU: `D_ICD_DIAGNOSES`, `DIAGNOSES_ICD`, `ICUSTAYS`, and `PATIENTS`.

Import necessary packages:

[209]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import make_pipeline
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import Ridge
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

[4]:
```
diagnoses_icd = pd.read_csv("mimic-iii-clinical-database-1.4/DIAGNOSES_ICD.csv.
↪gz")
diagnoses_icd.head()
```

[4]:

	ROW_ID	SUBJECT_ID	HADM_ID	SEQ_NUM	ICD9_CODE
0	1297	109	172335	1.0	40301
1	1298	109	172335	2.0	486
2	1299	109	172335	3.0	58281
3	1300	109	172335	4.0	5855

```
4    1301         109    172335      5.0      4254
```

`diagnoses_icd` is used to determine ICD9 codes using the ICD-9 coding system.

[2]:
```
d_icd_diagnoses = pd.read_csv("mimic-iii-clinical-database-1.4/D_ICD_DIAGNOSES.
↪CSV")
d_icd_diagnoses.head()
```

[2]:
```
   ROW_ID ICD9_CODE              SHORT_TITLE  \
0     174     01166     TB pneumonia-oth test
1     175     01170    TB pneumothorax-unspec
2     176     01171   TB pneumothorax-no exam
3     177     01172  TB pneumothorx-exam unkn
4     178     01173  TB pneumothorax-micro dx

                                        LONG_TITLE
0  Tuberculous pneumonia [any form], tubercle bac…
1              Tuberculous pneumothorax, unspecified
2  Tuberculous pneumothorax, bacteriological or h…
3  Tuberculous pneumothorax, bacteriological or h…
4  Tuberculous pneumothorax, tubercle bacilli fou…
```

`d_icd_diagnoses` is used to map the ICD-9 codes to the ICU stay times. - can be done using `SUBJECT_ID`: unique to each patient - can be done using `HADM_ID`: each hospital admission of patient

[171]:
```
icu_stays = pd.read_csv("mimic-iii-clinical-database-1.4/ICUSTAYS.csv")
icu_stays.head()
```

[171]:
```
   ROW_ID  SUBJECT_ID  HADM_ID  ICUSTAY_ID DBSOURCE FIRST_CAREUNIT  \
0     365         268   110404      280836  carevue           MICU
1     366         269   106296      206613  carevue           MICU
2     367         270   188028      220345  carevue            CCU
3     368         271   173727      249196  carevue           MICU
4     369         272   164716      210407  carevue            CCU

  LAST_CAREUNIT  FIRST_WARDID  LAST_WARDID              INTIME  \
0          MICU            52           52 2198-02-14 23:27:38
1          MICU            52           52 2170-11-05 11:05:29
2           CCU            57           57 2128-06-24 15:05:20
3          SICU            52           23 2120-08-07 23:12:42
4           CCU            57           57 2186-12-25 21:08:04

               OUTTIME     LOS
0 2198-02-18 05:26:11  3.2490
1 2170-11-08 17:46:57  3.2788
2 2128-06-27 12:32:29  2.8939
3 2120-08-10 00:39:04  2.0600
```

```
4  2186-12-27 12:01:13  1.6202
```

`icu_stays` is used to determine the length of ICU stay times where `LOS` is the length of stay normalized to 1.0 for 24 hours.

[123]:
```
patients = pd.read_csv("mimic-iii-clinical-database-1.4/PATIENTS.csv")
patients.head()
```

[123]:
```
   ROW_ID  SUBJECT_ID GENDER                  DOB                  DOD  \
0     234         249      F  2075-03-13 00:00:00                  NaN
1     235         250      F  2164-12-27 00:00:00  2188-11-22 00:00:00
2     236         251      M  2090-03-15 00:00:00                  NaN
3     237         252      M  2078-03-06 00:00:00                  NaN
4     238         253      F  2089-11-26 00:00:00                  NaN

              DOD_HOSP DOD_SSN  EXPIRE_FLAG
0                  NaN     NaN            0
1  2188-11-22 00:00:00     NaN            1
2                  NaN     NaN            0
3                  NaN     NaN            0
4                  NaN     NaN            0
```

`patients` is used to determine how age affects ICU stay times by using `DOB` and `INTIME` from `icu_stays` to create the variable `age`.

Data Cleaning and Visualizations:

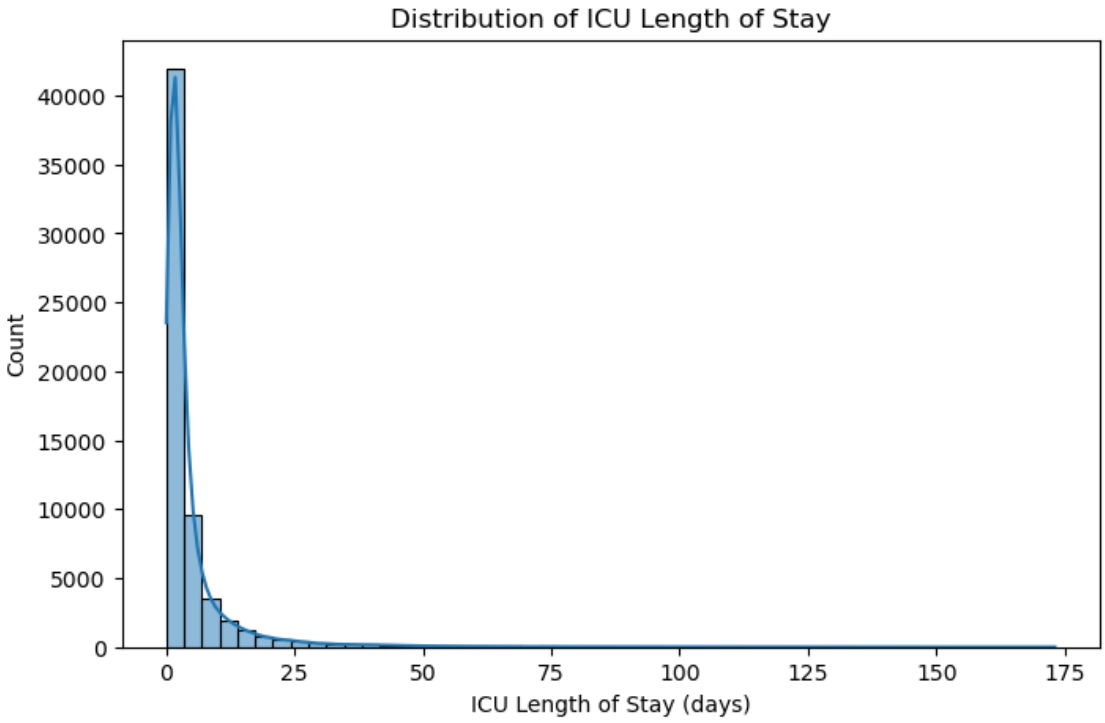
[21]:
```
icu = icu_stays.copy()

# convert LOS from "days" to hours
icu['LOS_hours'] = icu['LOS'] * 24
icu['LOS_hours'].describe()
```

[21]:
```
count    61522.000000
mean       118.031318
std        231.330822
min          0.002400
25%         26.592600
50%         50.214000
75%        107.596200
max       4153.740000
Name: LOS_hours, dtype: float64
```

[23]:
```
# merge diagnoses with description
diag = diagnoses_icd.merge(d_icd_diagnoses[['ICD9_CODE', 'LONG_TITLE']],␣
 ↪on='ICD9_CODE', how='left')

# mergy with ICU using HADM_ID
```

```
icu_diag = icu.merge(diag, on='HADM_ID', how='left')
```

[197]:
```
# Basic LOS distribution
plt.figure(figsize=(8,5))
sns.histplot(icu['LOS'], bins=50, kde=True)
plt.xlabel("ICU Length of Stay (days)")
plt.ylabel("Count")
plt.title("Distribution of ICU Length of Stay")
plt.show()
```



Determine what are the most common diagnoses in ICU patients:

[61]:
```
# most common
common_diag = (icu_diag['LONG_TITLE'].value_counts())
common_diag.head(20)
```

[61]:
```
LONG_TITLE
Unspecified essential hypertension
21530
Congestive heart failure, unspecified
14226
Atrial fibrillation
14048
Coronary atherosclerosis of native coronary artery
```

```
13107
Acute kidney failure, unspecified
10108
Diabetes mellitus without mention of complication, type II or unspecified type,
not stated as uncontrolled     9531
Other and unspecified hyperlipidemia
9095
Acute respiratory failure
8609
Urinary tract infection, site not specified
7375
Esophageal reflux
6552
Pure hypercholesterolemia
6124
Need for prophylactic vaccination and inoculation against viral hepatitis
5786
Anemia, unspecified
5756
Observation for suspected infectious condition
5589
Pneumonia, organism unspecified
5412
Unspecified acquired hypothyroidism
5211
Acute posthemorrhagic anemia
5026
Acidosis
5018
Chronic airway obstruction, not elsewhere classified
4699
Severe sepsis
4542
Name: count, dtype: int64
```

Compare length of ICU stay times between each ICU units:

[189]:
```
icu.groupby('FIRST_CAREUNIT')['LOS_hours'].median().sort_values(ascending=False)
```

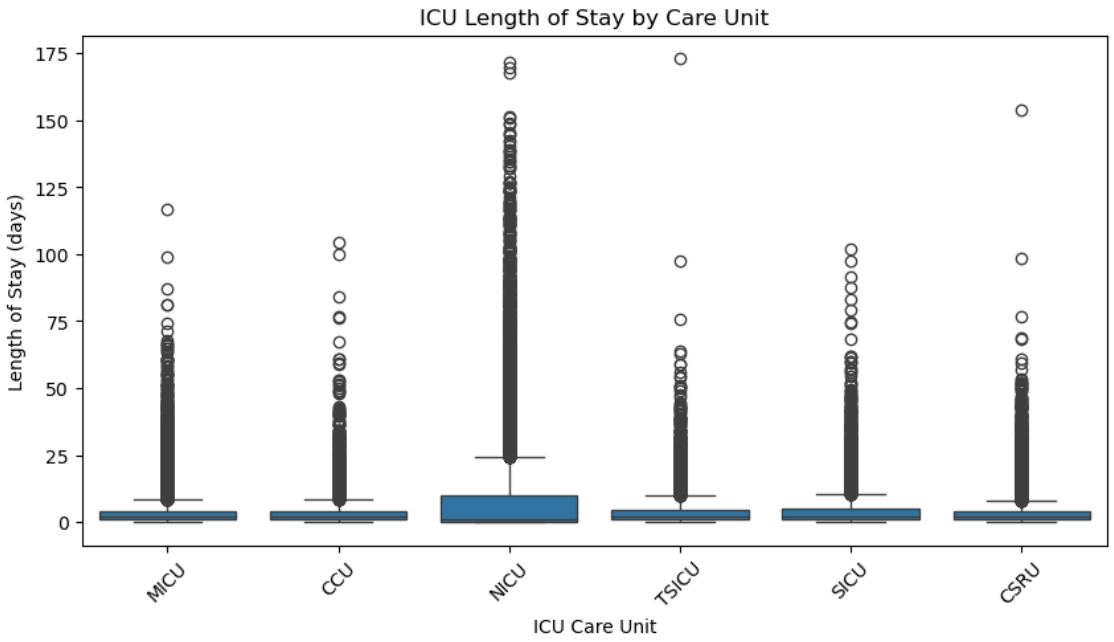
[189]:
```
FIRST_CAREUNIT
SICU     54.0528
CCU      52.7460
CSRU     51.6696
TSICU    50.6760
MICU     50.2920
NICU     19.2600
Name: LOS_hours, dtype: float64
```

The Surgical Intensive Care Unit (SICU) has the longest median ICU stay time.

[190]:
```
icu.groupby('FIRST_CAREUNIT')['LOS_hours'].mean().sort_values(ascending=False)
```

[190]:
```
FIRST_CAREUNIT
NICU     240.619333
SICU     112.961362
TSICU    106.652674
MICU      96.306450
CCU       93.637481
CSRU      93.600378
Name: LOS_hours, dtype: float64
```

[196]:
```
# LOS by ICU unit
plt.figure(figsize=(10,5))
sns.boxplot(data=icu, x="FIRST_CAREUNIT", y="LOS")
plt.xticks(rotation=45)
plt.ylabel("Length of Stay (days)")
plt.xlabel("ICU Care Unit")
plt.title("ICU Length of Stay by Care Unit")
plt.show()
```



The Neonatal Intensive Care Unit (NICU) has the longest average ICU stay time. This is a large contrast compared to it having the lowest median ICU stay time suggesting that it has extreme cases and outliers typically coming from premature newborns.

Determine how age factors into ICU stay time: Dates in the MIMIC-III database are randomly shifted to for deidentification, but consistently for a patient's records. Typically ages

over 89 are censored replaced with a fake date far in the future.

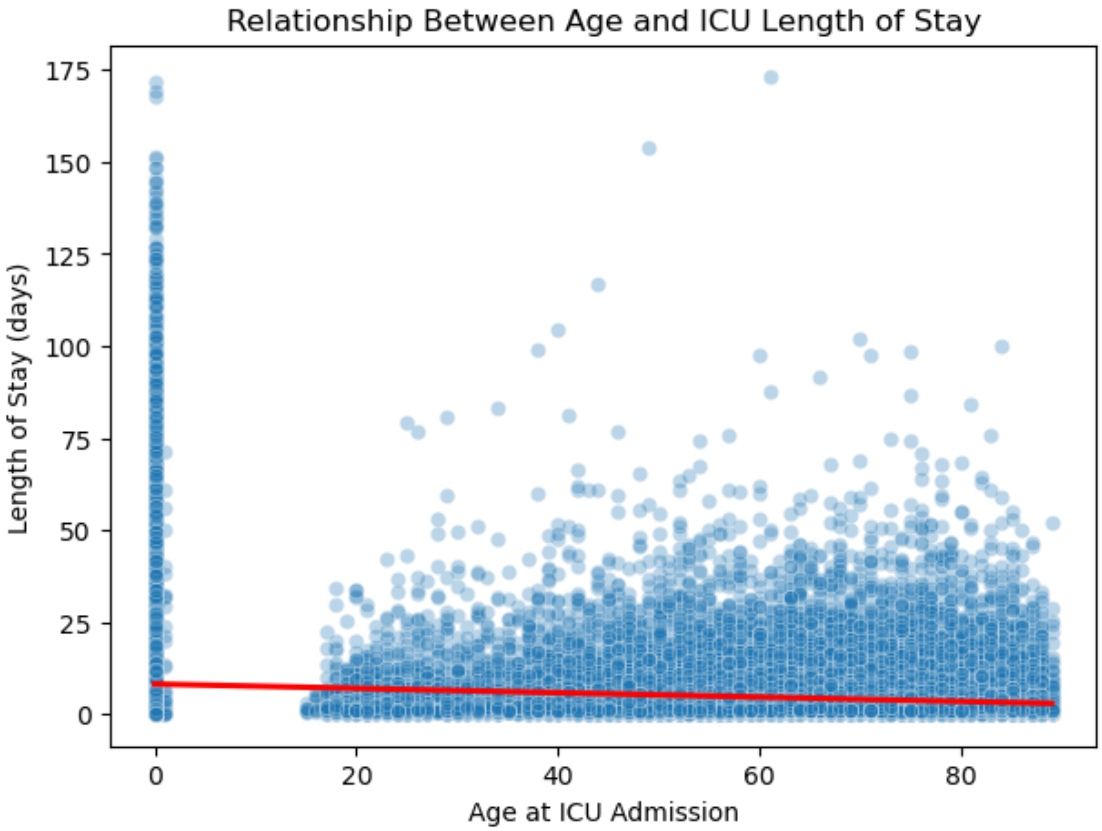
[167]:
```
icu['INTIME'] = pd.to_datetime(icu['INTIME'])
patients['DOB'] = pd.to_datetime(patients['DOB'])
icu_age = icu.merge(patients[['SUBJECT_ID', 'DOB']], on='SUBJECT_ID',␣
 ↪how='left')
icu_age['age'] = icu_age['INTIME'].dt.year - icu_age['DOB'].dt.year
# remove extreme ages
icu_age = icu_age[icu_age['age'] < 100]
# create age groups
icu_age['age_group'] = pd.cut(icu_age['age'], bins=[0,40,60,80,90],␣
 ↪labels=['<40','40-60','60-80','80+'])
```

[170]:
```
icu_age.groupby('age_group')['LOS_hours'].median()
```

```
/var/folders/q9/bn1wxfxn2vx3jr7pdxt_h1zw0000gn/T/ipykernel_33001/2315953813.py:4
: FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  icu_age.groupby('age_group')['LOS_hours'].median()
```

[170]:
```
age_group
<40      45.1848
40-60    49.6236
60-80    53.7636
80+      54.5112
Name: LOS_hours, dtype: float64
```

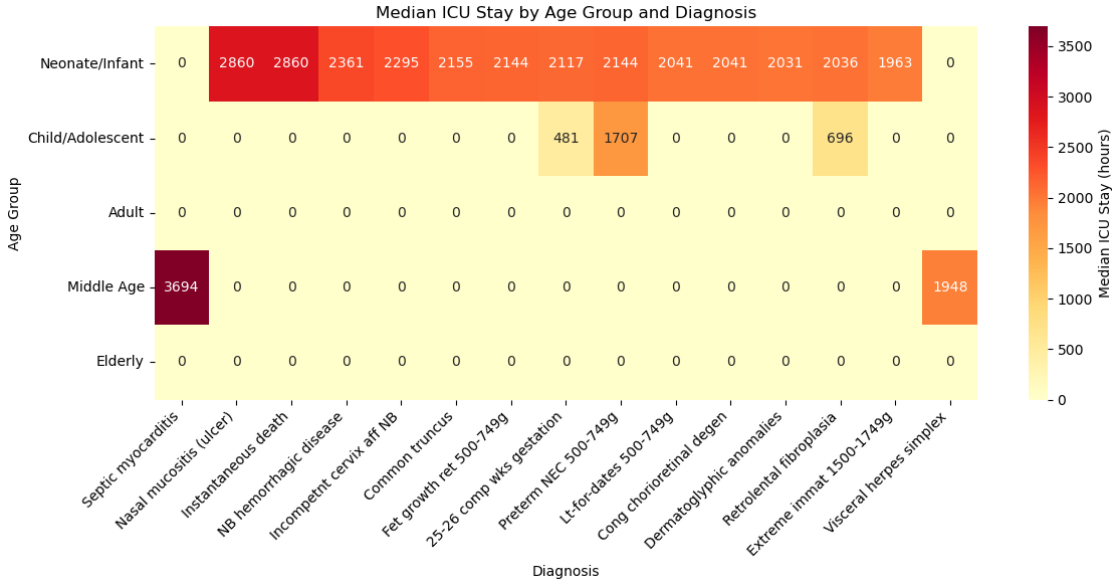
[194]:
```
plt.figure(figsize=(7,5))
sns.scatterplot(data=icu_age, x="age", y="LOS", alpha=0.3)
sns.regplot(data=icu_age, x="age", y="LOS", scatter=False, color='red')
plt.xlabel("Age at ICU Admission")
plt.ylabel("Length of Stay (days)")
plt.title("Relationship Between Age and ICU Length of Stay")
plt.show()
```



Typically the median of ICU stay times for people in the older age groups are larger than those in the younger age groups because they are more prone to illnesses and less healthy due to age. However, younger patients often have large outliers for premature infants that have very long ICU stays.

[207]:
```
# expand the bins to show younger age groups
bins = [0, 1, 18, 40, 65, 90]
labels = ['Neonate/Infant', 'Child/Adolescent', 'Adult', 'Middle Age',␣
 ↪'Elderly']
icu_age2 = icu_age.copy()
icu_age2['age_group'] = pd.cut(icu_age['age'], bins=bins, labels=labels,␣
 ↪right=False)

icu_diag_age = icu_age2.merge(icu_diag[['ICUSTAY_ID',␣
 ↪'SHORT_TITLE']],on='ICUSTAY_ID',how='left')
heatmap_data = icu_diag_age.groupby(['age_group', 'SHORT_TITLE'],␣
 ↪observed=True)['LOS_hours'].median().unstack(fill_value=0)
top_diag = icu_diag_age.groupby('SHORT_TITLE')['LOS_hours'].median().
 ↪sort_values(ascending=False).head(15).index
heatmap_data = heatmap_data[top_diag]
```

```
# plot the heatmap
plt.figure(figsize=(12,6))
sns.heatmap(heatmap_data, annot=True, fmt=".0f", cmap="YlOrRd",␣
 ↪cbar_kws={'label':'Median ICU Stay (hours)'})
plt.title("Median ICU Stay by Age Group and Diagnosis")
plt.ylabel("Age Group")
plt.xlabel("Diagnosis")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



The heatmap compares the diagnosis-level LOS per age group to highlight the extreme cases. It proves that babies typically due to being premature have more complications that are rare or extreme than the other age groups.

Compare ICU stay time between diagnoses

[175]:
```
# overall diagnoses with long ICU stays
icu_diag = icu_diag.merge(d_icd_diagnoses[['ICD9_CODE', 'SHORT_TITLE',␣
 ↪'LONG_TITLE']],on='ICD9_CODE',how='left')
diagnosis_los = (icu_diag.groupby('SHORT_TITLE')['LOS_hours'].median().
 ↪sort_values(ascending=False))
diagnosis_los.head(20)
```

[175]:
```
SHORT_TITLE
Septic myocarditis          3694.2720
Instantaneous death         2859.9408
```

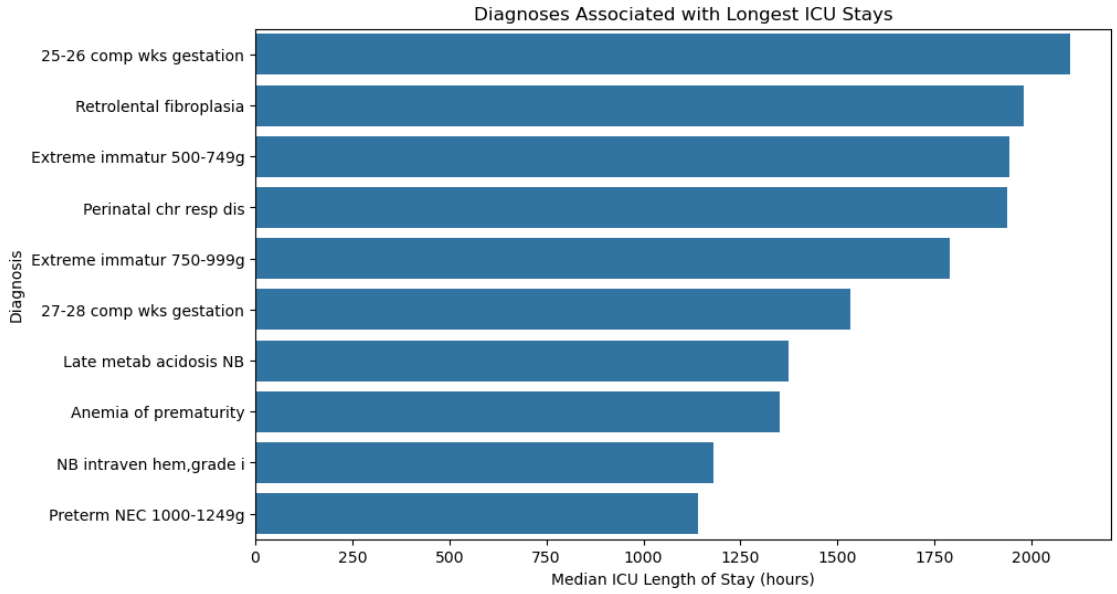
```
Nasal mucositis (ulcer)     2859.9408
NB hemorrhagic disease      2360.5176
Incompetnt cervix aff NB    2295.2160
Common truncus              2154.7344
Fet growth ret 500-749g     2143.9656
25-26 comp wks gestation    2099.8464
Preterm NEC 500-749g        2041.9500
Lt-for-dates 500-749g       2041.3968
Cong chorioretinal degen    2041.2024
Dermatoglyphic anomalies    2030.9496
Retrolental fibroplasia     1978.7472
Extreme immat 1500-1749g    1963.0344
Visceral herpes simplex     1947.9936
Extreme immatur 500-749g    1941.8208
Perinatal chr resp dis      1938.1224
Cystic fibrosis gene car    1923.2076
Oth neonatal coag dis       1915.9572
Vac-dis combinations NEC    1913.8440
Name: LOS_hours, dtype: float64
```

[177]:
```
# focus on common diagnoses and exclude rare outliers
diagnosis_counts = icu_diag['SHORT_TITLE'].value_counts()
common_diagnoses = diagnosis_counts[diagnosis_counts > 50].index
diagnosis_los_common = (icu_diag[icu_diag['SHORT_TITLE'].
 ↪isin(common_diagnoses)].groupby('SHORT_TITLE')['LOS_hours'].median().
 ↪sort_values(ascending=False))
diagnosis_los_common.head(20)
```

[177]:
```
SHORT_TITLE
25-26 comp wks gestation    2099.8464
Retrolental fibroplasia     1978.7472
Extreme immatur 500-749g    1941.8208
Perinatal chr resp dis      1938.1224
Extreme immatur 750-999g    1790.3304
27-28 comp wks gestation    1533.4776
Late metab acidosis NB      1372.7988
Anemia of prematurity       1351.1472
NB intraven hem,grade i     1181.7144
Preterm NEC 1000-1249g      1141.3464
NB septicemia [sepsis]      1090.5000
Patent ductus arteriosus     998.8176
29-30 comp wks gestation     971.5176
Preterm NEC 1250-1499g       783.1752
Neonatal conjunctivitis      716.0448
Neonatal candida infect      690.3516
Neonatal bradycardia         674.8776
Primary apnea of newborn     661.7952
```

```
Cong pulmon valve stenos     625.6152
Respiratory distress syn     568.7256
Name: LOS_hours, dtype: float64
```

[179]:
```
top10 = diagnosis_los_common.head(10).reset_index()

plt.figure(figsize=(10,6))
sns.barplot(data=top10, x='LOS_hours', y='SHORT_TITLE')
plt.xlabel('Median ICU Length of Stay (hours)')
plt.ylabel('Diagnosis')
plt.title('Diagnoses Associated with Longest ICU Stays')
plt.show()
```



Patients associated with the longest ICU stay tend to be extremely premature and low weight newborns as shown in the graph with less than 30 week gestations or less than 1000g patients. These infants require prolonged respiratory support, close monitoring, and treatment for complications that are common in premature babies, resulting in extended ICU stays.

Predict ICU length of stay in days using age, diagnosis, and ICU type.

Prepare Dataframes:

[220]:
```
# primary diagnosis table (SEQ_NUM == 1)
diag_primary = icu_diag[icu_diag['SEQ_NUM'] == 1].copy()
# take frequent diagnosis code if primary code is not available
if 'SEQ_NUM' not in icu_diag.columns or icu_diag['SEQ_NUM'].isna().all():
    diag_primary = (icu_diag.sort_values('ROW_ID_y').groupby('ICUSTAY_ID').
 ↪first().reset_index())
```

```
# Merge primary diagnosis into icu_age (left join, keep all stays)
df = icu_age.merge(diag_primary[['ICUSTAY_ID',␣
 ↪'SHORT_TITLE']],on='ICUSTAY_ID',how='left').copy()

# ensure LOS numeric and drop zeros or negative
df = df[df['LOS'].notna()].copy()
df = df[df['LOS'] >= 0]
```

Feature Engineering:

[221]:
```
# target
y = df['LOS'].values

df['age'] = df['age'].fillna(-1)
features_num = ['age']
features_cat = ['FIRST_CAREUNIT']

# for each diagnosis keep top 30
top_diag = df['SHORT_TITLE'].value_counts().head(topN).index.tolist()
df['diag_group'] = df['SHORT_TITLE'].where(df['SHORT_TITLE'].isin(top_diag),␣
 ↪other='Other')
features_cat.append('diag_group')

X = df[features_num + features_cat].copy()
```

Create Preprocessing Pipelines:

[222]:
```
# numerical features
numeric_transformer = make_pipeline(
    SimpleImputer(strategy='median')
)

# categorical features
categorical_transformer = make_pipeline(
    SimpleImputer(strategy='constant', fill_value='Unknown'),
    OneHotEncoder(handle_unknown='ignore')
)

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, features_num),
        ('cat', categorical_transformer, features_cat)
    ]
)
```

[223]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,␣
 ↪random_state=42)
```

```
# reduce skew for regression
y_train_log = np.log1p(y_train)
y_test_log = np.log1p(y_test)

X_train_proc = preprocessor.fit_transform(X_train)
X_test_proc = preprocessor.transform(X_test)

# feature importance
num_names = features_num
ohe = preprocessor.named_transformers_['cat'].named_steps['onehotencoder']
cat_ohe_names = ohe.get_feature_names_out(features_cat).tolist()
feature_names = num_names + cat_ohe_names
```

Ride Regression Model on log(LOS):

[224]:
```
# calculate metrics
def regression_metrics(y_true, y_pred, prefix=""):
    mae = mean_absolute_error(y_true, y_pred)
    rmse = mean_squared_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"{prefix} MAE: {mae:.3f}, RMSE: {rmse:.3f}, R2: {r2:.3f}")
```

[225]:
```
ridge = Ridge(alpha=1.0, random_state=42)
ridge.fit(X_train_proc, y_train_log)

y_pred_log_ridge = ridge.predict(X_test_proc)
y_pred_ridge = np.expm1(y_pred_log_ridge)

print("Ridge Regression:")
regression_metrics(y_test, y_pred_ridge, prefix="Ridge")
```

```
Ridge Regression:
Ridge MAE: 3.845, RMSE: 99.524, R2: 0.015
```

Random Forest Regression Model:

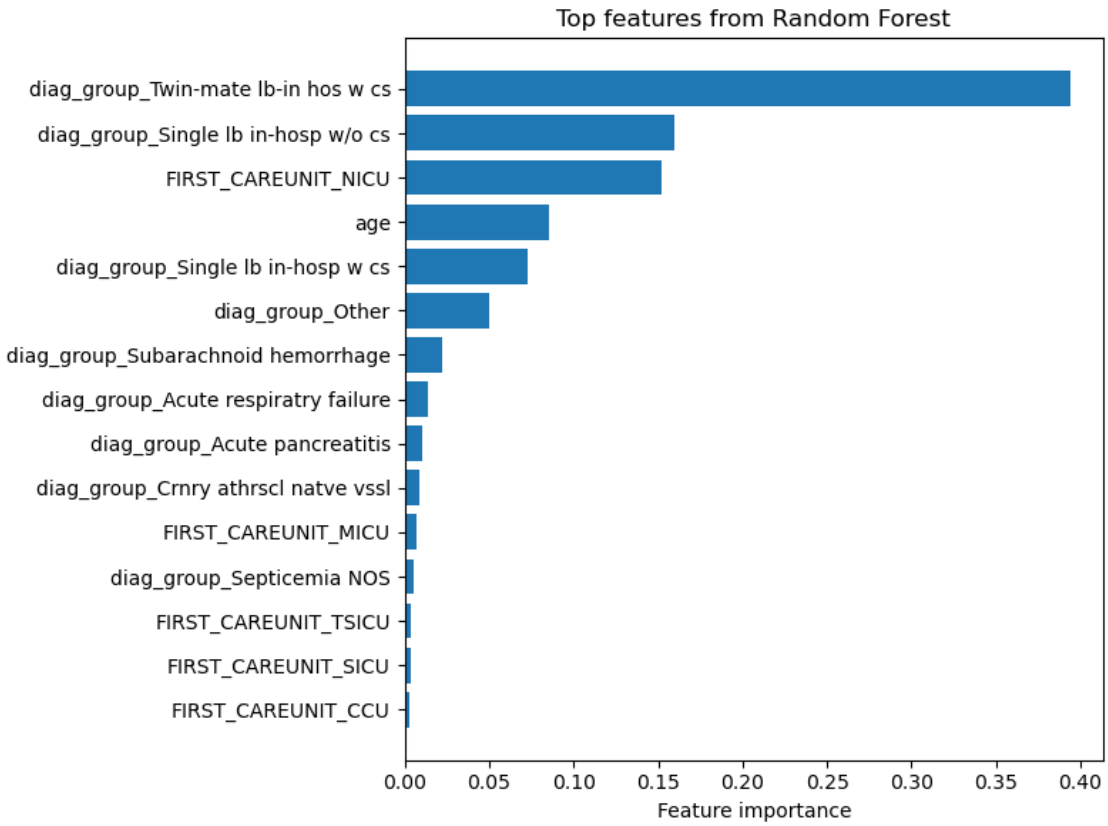
[226]:
```
rf = RandomForestRegressor(n_estimators=200, max_depth=10, random_state=42,␣
 ↪n_jobs=-1)
rf.fit(X_train_proc, y_train)

y_pred_rf = rf.predict(X_test_proc)
print("\nRandom Forest Regression:")
regression_metrics(y_test, y_pred_rf, prefix="RF")
```

```
Random Forest Regression:
RF MAE: 4.588, RMSE: 89.923, R2: 0.110
```

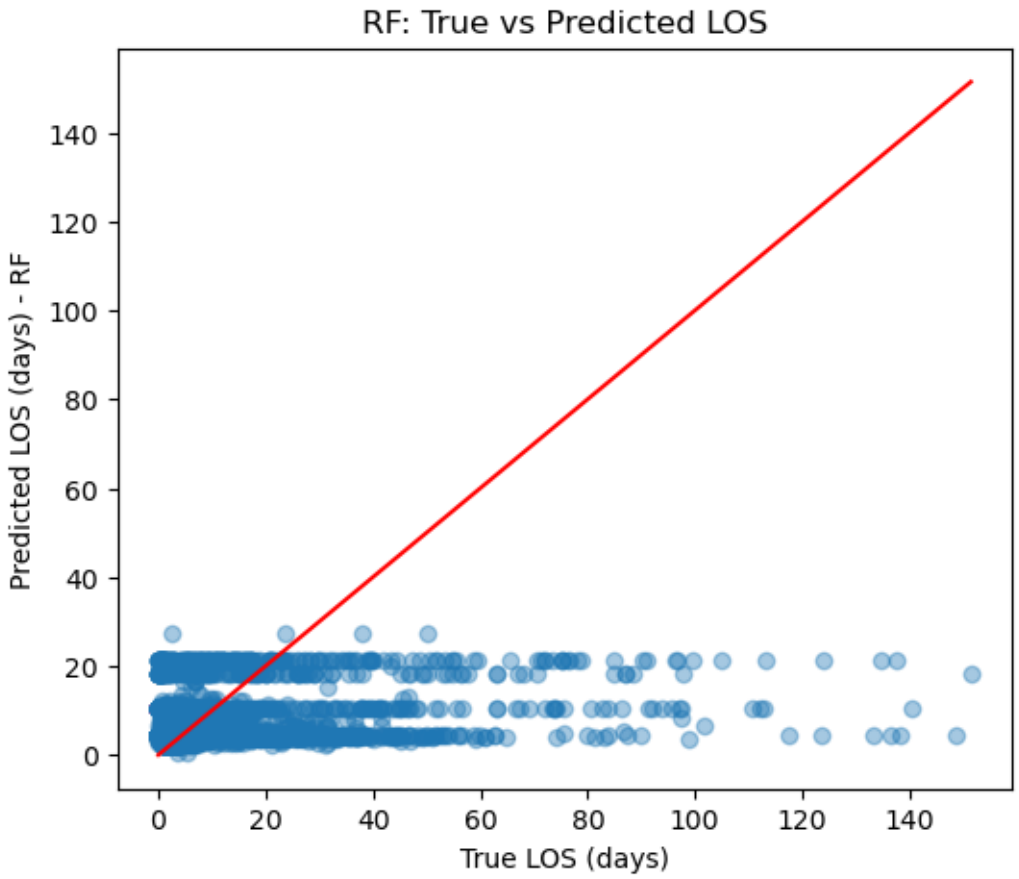
[227]:
```
# find feature importance
importances = rf.feature_importances_
imp_df = pd.DataFrame({'feature': feature_names, 'importance': importances})
imp_df = imp_df.sort_values('importance', ascending=False).
 ↪reset_index(drop=True)
print("\nTop 20 feature importances (RF):")
print(imp_df.head(20))
```

```
Top 20 feature importances (RF):
                                feature  importance
0   diag_group_Twin-mate lb-in hos w cs    0.393739
1   diag_group_Single lb in-hosp w/o cs    0.159779
2                  FIRST_CAREUNIT_NICU    0.152011
3                                  age    0.085726
4    diag_group_Single lb in-hosp w cs    0.072649
5                    diag_group_Other    0.049753
6    diag_group_Subarachnoid hemorrhage    0.021984
7   diag_group_Acute respiratry failure    0.014202
8         diag_group_Acute pancreatitis    0.010582
9   diag_group_Crnry athrscl natve vssl    0.008734
10                 FIRST_CAREUNIT_MICU    0.007317
11           diag_group_Septicemia NOS    0.005322
12                FIRST_CAREUNIT_TSICU    0.003898
13                 FIRST_CAREUNIT_SICU    0.003713
14                  FIRST_CAREUNIT_CCU    0.003103
15                 FIRST_CAREUNIT_CSRU    0.002588
16   diag_group_Other postop infection    0.002264
17   diag_group_DMI ketoacd uncontrold    0.000510
18   diag_group_Food/vomit pneumonitis    0.000381
19    diag_group_Mitral valve disorder    0.000339
```

[228]:
```
# plot top features
plt.figure(figsize=(8,6))
plt.barh(imp_df['feature'].head(15)[::-1], imp_df['importance'].head(15)[::-1])
plt.xlabel('Feature importance')
plt.title('Top features from Random Forest')
plt.tight_layout()
plt.show()
```



[229]:
```
# residual plot
plt.figure(figsize=(6,5))
plt.scatter(y_test, y_pred_rf, alpha=0.4)
plt.plot([0, max(y_test.max(), y_pred_rf.max())], [0, max(y_test.max(),␣
 ↪y_pred_rf.max())], color='red')
plt.xlabel('True LOS (days)')
plt.ylabel('Predicted LOS (days) - RF')
plt.title('RF: True vs Predicted LOS')
plt.show()
```



Model Performance: - For the Ridge Regression Model, the MAE was 3.845 so on average the predicted ICU LOS was off by about 3.8 days. The RMSE was 99.524 which is high suggesting that there are extreme outliers. The R^2 was 0.015 which is very low showing that a simple linear model does not capture the patterns well and that 1.5% of the variance in ICU LOS was explained by the model. - For the Random Forest Model, the MAE was 4.588 which is worse than the Ridge model. The RMSE was 89.923 which is better than the Ridge model, however, it is still high. The R^2 was 0.110 which is also better than the Ridge model showing that it can capture some non-linear patterns and that about 11% of the variance in ICU LOS was explained by the model.

Thus, overall ICU stay time is hard to predict using only age, primary diagnosis, and the care unit. There is a lot of variation that is not explained by the model. The rank feature importance shows that twin birth is strongly associated with ICU stay time as twins likely stay longer in the NICU. Single births and early life ICU stays are also common showing that the NICU unit is most predictive of stay time. Age is another highly important feature, however, it is not as significant as ICU type and diagnosis.

Conclusion

The analysis of the MIMIC-III dataset shows that patient age, the primary diagnosis, and ICU type are associated with ICU stay time, but ICU LOS varies highly among patients. Newborns

and infants in the NICU tend to have the longest stays. Diagnoses like prematurity or respiratory conditions are linked to longer ICU stays as well. Age plays a role in how long a patient stays in the ICU, but remains less predictive than diagnosis and ICU type. By applying predictive modeling using Ridge and Random Forest regression, I was able to predict ICU LOS based on age, diagnosis, and ICU type. The linear ridge model explained little variance and the random forest model captured some non-linear patterns. Feature importance confirmed the EDA findings that NICU status, twin births, and newborn births were the strongest predictors, followed by age. Overall, while the general trends are clear, ICU stay time is difficult to predict accurately with only age, diagnosis and, ICU type. Further analysis and exploration would have to be done to combat outliers showing the complexity of ICU care. The EDA, however, provides an understanding of which patient groups tend to stay longer in the ICU.