In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
# Load the LABEVENTS and D_LABITEMS tables
labevents = pd.read_csv('LABEVENTS.csv', compression='gzip')
d_labitems = pd.read_csv('D_LABITEMS.csv', compression='gzip')
```

In [148…
```
# Display basic info
print(f"LABEVENTS shape: {labevents.shape}")
print(f"D_LABITEMS shape: {d_labitems.shape}")
print(f'Null values LABEVENTS  {labevents['HADM_ID'].isnull().sum()}')
print(labevents.columns.tolist())
print(d_labitems.columns.tolist())
```

```
LABEVENTS shape: (27854055, 7)
D_LABITEMS shape: (753, 6)
Null values LABEVENTS  5609021
['HADM_ID', 'ITEMID', 'CHARTTIME', 'VALUE', 'VALUENUM', 'VALUEUOM', 'FLAG']
['ROW_ID', 'ITEMID', 'LABEL', 'FLUID', 'CATEGORY', 'LOINC_CODE']
```

In [149…
```
labevents = labevents[['HADM_ID','ITEMID','CHARTTIME','VALUE','VALUENUM','VALUEU

labevents.head(3)
```

Out[149…

	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUENUM	VALUEUOM	FLAG
0	NaN	50820	2101-10-12 16:07:00	7.39	7.39	units	NaN
1	NaN	50800	2101-10-12 18:17:00	ART	NaN	NaN	NaN
2	NaN	50802	2101-10-12 18:17:00	-1	-1.00	mEq/L	NaN

In [ ]:
```
```

In [10]:
```
# Create a search function for lab items
def find_lab_items(search_terms):
    """Find ITEMIDs for given search terms"""
    results = {}
    for term in search_terms:
        # Case-insensitive search in label
        mask = d_labitems['LABEL'].str.contains(term, case=False, na=False)
        matches = d_labitems[mask][['ITEMID', 'LABEL', 'FLUID']]
        results[term] = matches
        print(f" {term}")
        print(matches.head(10).to_string())
    return results
```

In [150…
```
# Search for your specific lab tests
search_terms = ['creatinine', 'lactate', 'bilirubin', 'wbc', 'platelet',
                'hemoglobin', 'sodium', 'potassium']

lab_mappings = find_lab_items(search_terms)
```

creatinine

	ITEMID	LABEL	FLUID
168	50841	Creatinine, Ascites	Ascites
239	50912	Creatinine	Blood
347	51021	Creatinine, Joint Fluid	Joint Fluid
358	51032	Creatinine, Body Fluid	Other Body Fluid
378	51052	Creatinine, Pleural	Pleural
393	51067	24 hr Creatinine	Urine
396	51070	Albumin/Creatinine, Urine	Urine
399	51073	Amylase/Creatinine Ratio, Urine	Urine
406	51080	Creatinine Clearance	Urine
407	51081	Creatinine, Serum	Urine

lactate

	ITEMID	LABEL	FLUID
140	50813	Lactate	Blood
170	50843	Lactate Dehydrogenase, Ascites	Ascites
281	50954	Lactate Dehydrogenase (LD)	Blood
341	51015	Lactate Dehydrogenase, CSF	Cerebrospinal Fluid (CSF)
380	51054	Lactate Dehydrogenase, Pleural	Pleural

bilirubin

	ITEMID	LABEL	FLUID
118	51464	Bilirubin	Urine
119	51465	Bilirubin Crystals	Urine
165	50838	Bilirubin, Total, Ascites	Ascites
210	50883	Bilirubin, Direct	Blood
211	50884	Bilirubin, Indirect	Blood
212	50885	Bilirubin, Total	Blood
338	51012	Bilirubin, Total, CSF	Cerebrospinal Fluid (CSF)
354	51028	Bilirubin, Total, Body Fluid	Other Body Fluid
375	51049	Bilirubin, Total, Pleural	Pleural

wbc

	ITEMID	LABEL	FLUID
17	51363	WBC, CSF	Cerebrospinal Fluid (CSF)
38	51384	WBC, Joint Fluid	Joint Fluid
93	51439	WBC, Other Fluid	Other Body Fluid
112	51458	WBC, Pleural	Pleural
454	51128	WBC, Ascites	Ascites
626	51300	WBC Count	Blood
715	51516	WBC	Urine
716	51517	WBC Casts	Urine
717	51518	WBC Clumps	Urine
732	51533	WBCP	BLOOD

platelet

	ITEMID	LABEL	FLUID
566	51240	Large Platelets	Blood
590	51264	Platelet Clumps	Blood
591	51265	Platelet Count	Blood
592	51266	Platelet Smear	Blood

hemoglobin

	ITEMID	LABEL	FLUID
132	50805	Carboxyhemoglobin	Blood
138	50811	Hemoglobin	Blood
141	50814	Methemoglobin	Blood
179	50852	% Hemoglobin A1c	Blood
182	50855	Absolute Hemoglobin	Blood
538	51212	Fetal Hemoglobin	Blood
548	51222	Hemoglobin	Blood
549	51223	Hemoglobin A2	Blood
550	51224	Hemoglobin C	Blood
551	51225	Hemoglobin F	Blood

```
sodium
     ITEMID                LABEL             FLUID
151   50824  Sodium, Whole Blood             Blood
161   50834   Sodium, Body Fluid  Other Body Fluid
175   50848      Sodium, Ascites           Ascites
310   50983               Sodium             Blood
368   51042   Sodium, Body Fluid  Other Body Fluid
384   51058      Sodium, Pleural           Pleural
391   51065        Sodium, Stool             Stool
426   51100        Sodium, Urine             Urine
potassium
     ITEMID                   LABEL             FLUID
149   50822  Potassium, Whole Blood             Blood
160   50833               Potassium  Other Body Fluid
174   50847      Potassium, Ascites           Ascites
298   50971               Potassium             Blood
367   51041   Potassium, Body Fluid  Other Body Fluid
383   51057      Potassium, Pleural           Pleural
390   51064        Potassium, Stool             Stool
423   51097        Potassium, Urine             Urine
```

In [151…

```python
# Common ITEMIDs dictionary (based on typical MIMIC-III values)
common_labs = {
    'creatinine': [50912, 791],
    'lactate': [50813],
    'bilirubin': [50885],
    'wbc': [51300, 51301],
    'platelets': [51265],
    'hemoglobin': [51222, 50811],
    'sodium': [50983, 50824],
    'potassium': [50971, 50822]
}
```

In [152…

```python
# Create a flattened list of all ITEMIDs of interest
all_itemids = []
for lab_list in common_labs.values():
    all_itemids.extend(lab_list)
all_itemids = list(set(all_itemids))  # Remove duplicates
```

In [153…

```python
# Filter lab events for common tests
filtered_labevent = labevents[labevents['ITEMID'].isin(all_itemids)]
filtered_labevent.head()
```

Out[153…

	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUENUM	VALUEUOM	FLAG
6	NaN	50813	2101-10-12 18:17:00	1.8	1.8	mmol/L	NaN
15	NaN	50912	2101-10-13 03:00:00	1.7	1.7	mg/dL	abnormal
19	NaN	50971	2101-10-13 03:00:00	4.3	4.3	mEq/L	NaN
20	NaN	50983	2101-10-13 03:00:00	141	141.0	mEq/L	NaN
28	NaN	50912	2101-10-13 15:47:00	1.5	1.5	mg/dL	abnormal

In [154…
```
filtered_labevent.nunique()
```

Out[154…
```
HADM_ID        58098
ITEMID            12
CHARTTIME    1284618
VALUE           4019
VALUENUM        3539
VALUEUOM           5
FLAG               2
dtype: int64
```

In [155…
```
filtered_labs = filtered_labevent.merge(
    d_labitems[['ITEMID', 'LABEL', 'FLUID','CATEGORY']],
    on='ITEMID',
    how='left'
)
```

In [156…
```
# Handle numeric conversion safely
def safe_numeric(x):
    try:
        return pd.to_numeric(x, errors='coerce')
    except:
        return np.nan

filtered_labs['VALUE_NUM'] = filtered_labs['VALUE'].apply(safe_numeric)

filtered_labs.shape
```

Out[156…
```
(5517472, 11)
```

In [157…
```
# Remove rows with invalid values
initial_count = len(filtered_labs)

# 1. Remove where VALUE_NUM is NaN (non-numeric values)
filtered_labs = filtered_labs[filtered_labs['VALUE_NUM'].notna()]
filtered_labs.shape
```

Out[157…
```
(5515888, 11)
```

In [158…
```
# 2. Remove extreme outliers (values outside 0.1-99.9 percentile for each test)
labevents_clean = pd.DataFrame()
for lab_name, itemids in common_labs.items():
    lab_data = filtered_labs[filtered_labs['ITEMID'].isin(itemids)].copy()

    # Calculate percentiles for each lab test
    lower = lab_data['VALUE_NUM'].quantile(0.001)
    upper = lab_data['VALUE_NUM'].quantile(0.999)
     # Filter out extreme outliers
    lab_data = lab_data[(lab_data['VALUE_NUM'] >= lower) & (lab_data['VALUE_NUM'

    # Add lab name column
    lab_data['LAB_TEST'] = lab_name
    labevents_clean = pd.concat([labevents_clean, lab_data])
print(f"Removed {initial_count - len(labevents_clean):,} invalid/extreme outlier
print(f"Remaining lab records: {len(labevents_clean):,}")
```

```
Removed 10,557 invalid/extreme outlier records
Remaining lab records: 5,506,915
```

In [159…
```
labevents_clean=labevents_clean.dropna(subset='HADM_ID')
```

In [160…
```
labevents_clean.head()
```

Out[160…

	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUENUM	VALUEUOM	FLAG	LA
30	145834.0	50912	2101-10-20 16:40:00	3.2	3.2	mg/dL	abnormal	Creati
34	145834.0	50912	2101-10-22 04:00:00	1.9	1.9	mg/dL	abnormal	Creati
45	145834.0	50912	2101-10-22 21:15:00	1.7	1.7	mg/dL	abnormal	Creati
49	145834.0	50912	2101-10-23 03:45:00	1.6	1.6	mg/dL	abnormal	Creati
59	145834.0	50912	2101-10-20 19:59:00	2.5	2.5	mg/dL	abnormal	Creati

In [161…
```
labevents_clean['CHARTTIME'] = pd.to_datetime(labevents_clean['CHARTTIME'], erro
labevents_clean['CHARTTIME'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 4371431 entries, 30 to 5517469
Series name: CHARTTIME
Non-Null Count    Dtype
--------------    -----
4371431 non-null  datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 66.7 MB
```

In [162…
```
   # Merge with admissions data
add_ICU=pd.read_csv('ADMISSIONS.csv',compression='gzip')

add_ICU=add_ICU[['HADM_ID','ADMITTIME','DISCHTIME','HOSPITAL_EXPIRE_FLAG']]
```

In [163…
```
add_ICU['ADMITTIME']=pd.to_datetime(add_ICU['ADMITTIME'], errors='coerce')
add_ICU['DISCHTIME']=pd.to_datetime(add_ICU['DISCHTIME'], errors='coerce')
```

In [164…
```
lab_admissions = pd.merge(labevents_clean, add_ICU,
                          on=['HADM_ID'],
                          how='left')
```

In [165…
```
print(f'FInal data from labevent and admission length ({len(lab_admissions)})')
```

```
FInal data from labevent and admission length (4371431)
```

In [167…
```
#valid test
lab_admissions = lab_admissions[
    (lab_admissions['CHARTTIME'] >= lab_admissions['ADMITTIME']) &
    (lab_admissions['CHARTTIME'] <= lab_admissions['DISCHTIME'])
]
```

In [170…
```
(lab_admissions).head(2)
```

Out[170…

	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUENUM	VALUEUOM	FLAG	LAE
1	145834.0	50912	2101-10-22 04:00:00	1.9	1.9	mg/dL	abnormal	Creatin
2	145834.0	50912	2101-10-22 21:15:00	1.7	1.7	mg/dL	abnormal	Creatin

In [171…
```
# For each admission and lab test, take the first recorded value
first_labs = lab_admissions.sort_values('CHARTTIME').groupby(['HADM_ID', 'LAB_TE
```

In [172…
```
# Check for unique HADM_IDs
print(f"Unique admissions with lab data: {first_labs['HADM_ID'].nunique()}")
```

```
Unique admissions with lab data: 57448
```

In [174…
```
# Pivot to wide format (one row per admission)
pivot_data = first_labs.pivot_table(
    index='HADM_ID',
    columns='LAB_TEST',
    values='VALUE_NUM',
    aggfunc='first'
).reset_index()

print(f"\nPivot data shape: {pivot_data.shape}")
print(f"Columns in pivot data: {list(pivot_data.columns)}\n\n")
print(pivot_data.head())
```

```
Pivot data shape: (57448, 9)
Columns in pivot data: ['HADM_ID', 'bilirubin', 'creatinine', 'hemoglobin', 'lact
ate', 'platelets', 'potassium', 'sodium', 'wbc']


LAB_TEST   HADM_ID  bilirubin  creatinine  hemoglobin  lactate  platelets  \
0         100001.0        NaN         2.3        11.0      NaN      376.0
1         100003.0        5.5         1.2         7.1      1.1      148.0
2         100006.0        NaN         0.6        10.3      4.5      204.0
3         100007.0        NaN         0.6        12.3      1.9      259.0
4         100009.0        0.3         0.8        14.5      1.1      167.0

LAB_TEST  potassium  sodium   wbc
0               4.2   143.0  11.2
1               5.0   133.0  13.4
2               3.7   131.0  13.6
3               4.1   140.0  12.3
4               4.2   137.0   7.8
```

In [182…
```
ADD_ICU_PET=pd.read_csv('ADD_ICU_PAT_ONE_HOT.csv')
ADD_ICU_PET=ADD_ICU_PET[['HADM_ID','ICU_MORTALITY']]
final_data_lab_add=lab_admissions.merge(ADD_ICU_PET,on='HADM_ID',how='left')
final_data_lab_add.head(2)
```

Out[182…

	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUENUM	VALUEUOM	FLAG	LAE
0	145834.0	50912	2101-10-22 04:00:00	1.9	1.9	mg/dL	abnormal	Creatin
1	145834.0	50912	2101-10-22 21:15:00	1.7	1.7	mg/dL	abnormal	Creatin

In [185…

```
final_data_lab_add=final_data_lab_add.dropna(subset='ICU_MORTALITY').copy()

print(f"Mortality rate in final dataset: {final_data_lab_add['ICU_MORTALITY'].me
```

```
Mortality rate in final dataset: 3.69%
```

In [187…

```
final_data_lab_add.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 898525 entries, 14 to 4125045
Data columns (total 16 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   HADM_ID               898525 non-null  float64
 1   ITEMID                898525 non-null  int64
 2   CHARTTIME             898525 non-null  datetime64[ns]
 3   VALUE                 898525 non-null  object
 4   VALUENUM              898525 non-null  float64
 5   VALUEUOM              898525 non-null  object
 6   FLAG                  341525 non-null  object
 7   LABEL                 898525 non-null  object
 8   FLUID                 898525 non-null  object
 9   CATEGORY              898525 non-null  object
 10  VALUE_NUM             898525 non-null  float64
 11  LAB_TEST              898525 non-null  object
 12  ADMITTIME             898525 non-null  datetime64[ns]
 13  DISCHTIME             898525 non-null  datetime64[ns]
 14  HOSPITAL_EXPIRE_FLAG  898525 non-null  int64
 15  ICU_MORTALITY         898525 non-null  float64
dtypes: datetime64[ns](3), float64(4), int64(2), object(7)
memory usage: 116.5+ MB
```

In [189…

```
final_data_lab_add['ICU_MORTALITY'].drop_duplicates()
len(final_data_lab_add)
```

Out[189… 898525

In [191…

```
# Group by HADM_ID to get average lab values per admission
lab_stats = final_data_lab_add.groupby(['HADM_ID', 'LABEL'])['VALUENUM'].mean().

# Get mortality for each admission
mortality_per_admission = final_data_lab_add.groupby('HADM_ID')['ICU_MORTALITY']
```

In [207…

```
combined_data = lab_stats.merge(mortality_per_admission, left_index=True, right_
```

In [208…

```
combined_data.corr()['ICU_MORTALITY']
```

Out[208…
```
Bilirubin, Total          0.084349
Creatinine                0.091009
Hemoglobin               -0.021245
Lactate                   0.456790
Platelet Count           -0.045967
Potassium                 0.134874
Potassium, Whole Blood    0.084255
Sodium                    0.054390
Sodium, Whole Blood       0.003294
WBC Count                 0.183381
White Blood Cells         0.177252
ICU_MORTALITY             1.000000
Name: ICU_MORTALITY, dtype: float64
```

In [211…
```
combined_data=combined_data[['Lactate','Potassium','White Blood Cells','WBC Coun
combined_data.head()
```

Out[211…

HADM_ID	Lactate	Potassium	White Blood Cells	WBC Count	ICU_MORTALITY
100003.0	1.100	4.600000	12.566667	NaN	0.0
100010.0	0.825	4.050000	10.125000	NaN	0.0
100020.0	1.050	4.285714	9.184615	NaN	0.0
100021.0	NaN	3.956000	5.648936	NaN	0.0
100024.0	2.400	4.300000	13.175000	NaN	0.0

In [212…
```
combined_data.to_csv('D_LAB_CORR_ICU_MORTALITY.csv',index=False)
```