

FinalReport

April 11, 2025

1 Analyzing Patient Health Records to Identify Factors Influencing Hospital Readmission Rates

1.1 Project Overview

Hospital readmissions are a significant concern in healthcare, impacting patient outcomes and increasing healthcare costs. This project aims to analyse a diverse dataset of patient health records to identify factors that contribute to hospital readmissions. By leveraging Python's data analysis and visualization capabilities, we will uncover patterns and relationships between patient demographics, medical history, treatment plans, and readmission rates.

1. Load the data into a Python environment using pandas

```
[12]: import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

```
[2]: import pandas as pd

# Load main data file
df = pd.read_csv('diabetic_data.csv') # Adjust filename

# If there are mapping files (e.g. admission type id → name), load them
admission_mapping = pd.read_csv('admission_type_mapping.csv') # If applicable
admission_source_mapping = pd.read_csv('admission_source_mapping.csv') # If applicable
discharge_mapping = pd.read_csv('discharge_disposition_mapping.csv') # If applicable
```

2. Handle missing or inconsistent data entries, particularly in categorical variables like 'Race' and 'Gender'.

```
[3]: # Show missing values per column
missing_cols = df.isnull().sum()
missing_cols = missing_cols[missing_cols > 0]
print(missing_cols)

#unique data
for col in df.select_dtypes(include='object').columns:
    print(f"{col}: {df[col].unique()}")
```

```

# changed on basis of above:
df.replace('?', pd.NA, inplace=True)

# Standardize categorical variables
df['race'] = df['race'].replace('?', pd.NA)
df['gender'] = df['gender'].replace('Unknown/Invalid', pd.NA)

# Drop rows with missing race or gender
df.dropna(subset=['race', 'gender'], inplace=True)

```

```

max_glu_serum      96420
A1Cresult           84748
dtype: int64
race: ['Caucasian' 'AfricanAmerican' '?' 'Other' 'Asian' 'Hispanic']
gender: ['Female' 'Male' 'Unknown/Invalid']
age: ['[0-10)' '[10-20)' '[20-30)' '[30-40)' '[40-50)' '[50-60)' '[60-70)'
      '[70-80)' '[80-90)' '[90-100)']
weight: ['?' '[75-100)' '[50-75)' '[0-25)' '[100-125)' '[25-50)' '[125-150)'
         '[175-200)' '[150-175)' '>200']
payer_code: ['?' 'MC' 'MD' 'HM' 'UN' 'BC' 'SP' 'CP' 'SI' 'DM' 'CM' 'CH' 'PO'
             'WC' 'OT'
             'OG' 'MP' 'FR']
medical_specialty: ['Pediatrics-Endocrinology' '?' 'InternalMedicine'
                    'Family/GeneralPractice' 'Cardiology' 'Surgery-General' 'Orthopedics'
                    'Gastroenterology' 'Surgery-Cardiovascular/Thoracic' 'Nephrology'
                    'Orthopedics-Reconstructive' 'Psychiatry' 'Emergency/Trauma'
                    'Pulmonology' 'Surgery-Neuro' 'Obstetrics&Gynecology-GynecologicOnco'
                    'ObstetricsandGynecology' 'Pediatrics' 'Hematology/Oncology'
                    'Otolaryngology' 'Surgery-Colon&Rectal' 'Pediatrics-CriticalCare'
                    'Endocrinology' 'Urology' 'Psychiatry-Child/Adolescent'
                    'Pediatrics-Pulmonology' 'Neurology' 'Anesthesiology-Pediatric'
                    'Radiology' 'Pediatrics-Hematology-Oncology' 'Psychology' 'Podiatry'
                    'Gynecology' 'Oncology' 'Pediatrics-Neurology' 'Surgery-Plastic'
                    'Surgery-Thoracic' 'Surgery-PlasticwithinHeadandNeck' 'Ophthalmology'
                    'Surgery-Pediatric' 'Pediatrics-EmergencyMedicine'
                    'PhysicalMedicineandRehabilitation' 'InfectiousDiseases' 'Anesthesiology'
                    'Rheumatology' 'AllergyandImmunology' 'Surgery-Maxillofacial'
                    'Pediatrics-InfectiousDiseases' 'Pediatrics-AllergyandImmunology'
                    'Dentistry' 'Surgeon' 'Surgery-Vascular' 'Osteopath'
                    'Psychiatry-Addictive' 'Surgery-Cardiovascular' 'PhysicianNotFound'
                    'Hematology' 'Proctology' 'Obstetrics' 'SurgicalSpecialty' 'Radiologist'
                    'Pathology' 'Dermatology' 'SportsMedicine' 'Speech' 'Hospitalist'
                    'OutreachServices' 'Cardiology-Pediatric' 'Perinatology'
                    'Neurophysiology' 'Endocrinology-Metabolism' 'DCPTEAM' 'Resident']
diag_1: ['250.83' '276' '648' '8' '197' '414' '428' '398' '434' '250.7' '157'
         '518' '999' '410' '682' '402' '737' '572' 'V57' '189' '786' '427' '996'
         '277' '584' '462' '473' '411' '174' '486' '998' '511' '432' '626' '295'

```

'196' '250.6' '618' '182' '845' '423' '808' '250.4' '722' '403' '250.11'
 '784' '707' '440' '151' '715' '997' '198' '564' '812' '38' '590' '556'
 '578' '250.32' '433' 'V58' '569' '185' '536' '255' '250.13' '599' '558'
 '574' '491' '560' '244' '250.03' '577' '730' '188' '824' '250.8' '332'
 '562' '291' '296' '510' '401' '263' '438' '70' '250.02' '493' '642' '625'
 '571' '738' '593' '250.42' '807' '456' '446' '575' '250.41' '820' '515'
 '780' '250.22' '995' '235' '250.82' '721' '787' '162' '724' '282' '514'
 'V55' '281' '250.33' '530' '466' '435' '250.12' 'V53' '789' '566' '822'
 '191' '557' '733' '455' '711' '482' '202' '280' '553' '225' '154' '441'
 '250.81' '349' '?' '962' '592' '507' '386' '156' '200' '728' '348' '459'
 '426' '388' '607' '337' '82' '531' '596' '288' '656' '573' '492' '220'
 '516' '210' '922' '286' '885' '958' '661' '969' '250.93' '227' '112'
 '404' '823' '532' '416' '346' '535' '453' '250' '595' '211' '303'
 '250.01' '852' '218' '782' '540' '457' '285' '431' '340' '550' '54' '351'
 '601' '723' '555' '153' '443' '380' '204' '424' '241' '358' '694' '331'
 '345' '681' '447' '290' '158' '579' '436' '335' '309' '654' '805' '799'
 '292' '183' '78' '851' '458' '586' '311' '892' '305' '293' '415' '591'
 '794' '803' '79' '655' '429' '278' '658' '598' '729' '585' '444' '604'
 '727' '214' '552' '284' '680' '708' '41' '644' '481' '821' '413' '437'
 '968' '756' '632' '359' '275' '512' '781' '420' '368' '522' '294' '825'
 '135' '304' '320' '250.31' '669' '868' '496' '250.43' '826' '567' '3'
 '203' '53' '251' '565' '161' '495' '49' '250.1' '297' '663' '576' '355'
 '850' '287' '250.2' '611' '840' '350' '726' '537' '620' '180' '366' '783'
 '11' '751' '716' '250.3' '199' '464' '580' '836' '664' '283' '813' '966'
 '289' '965' '184' '480' '608' '333' '972' '212' '117' '788' '924' '959'
 '621' '238' '785' '714' '942' '250.23' '710' '47' '933' '508' '478' '844'
 '7' '736' '233' '42' '250.5' '397' '395' '201' '421' '253' '250.92' '600'
 '494' '977' '39' '659' '312' '614' '647' '652' '646' '274' '861' '425'
 '527' '451' '485' '217' '250.53' '442' '970' '193' '160' '322' '581'
 '475' '623' '374' '582' '568' '465' '801' '237' '376' '150' '461' '913'
 '226' '617' '987' '641' '298' '790' '336' '362' '228' '513' '383' '746'
 '353' '911' '506' '873' '155' '860' '534' '802' '141' 'V45' '396' '310'
 '341' '242' '719' '239' '533' '616' '519' '301' 'V66' '5' '989' '230'
 '385' '300' '853' '871' '570' '848' '463' '9' '934' '250.21' '236' '361'
 '594' '501' '810' '643' '430' '528' '205' '791' '983' '992' '490' '172'
 '171' '622' '306' '863' '864' '474' '660' '759' '356' '634' '967' '551'
 '695' '187' '732' '747' '323' '308' '370' '252' '152' '846' '164' '365'
 '718' '48' '266' '720' '94' '344' '797' '170' '878' '904' 'V56' '882'
 '843' '709' '973' '454' '686' '939' '487' '229' '991' '483' '357' '692'
 '796' '693' '935' '936' '800' '920' 'V26' '261' '307' '262' '250.9' '831'
 '145' '223' 'V71' '839' '685' 'V54' '35' '34' '179' '964' '136' '324'
 '389' '815' '334' '143' '526' '588' '192' 'V67' '394' '917' '88' '219'
 '325' '792' '717' '994' '990' '793' '207' '637' '195' '373' '847' '827'
 '31' '891' '814' 'V60' '703' '865' '352' '627' '378' '342' '886' '369'
 '745' '705' '816' '541' '986' '610' '633' '640' '753' '173' '835' '379'
 '445' '272' '382' '945' '619' '881' '250.52' '866' '405' '916' '215'
 '893' '75' '671' '928' '906' '897' '725' '867' '115' '890' '734' '521'
 '674' '470' '834' '146' '696' '524' '980' '691' '384' '142' '879'

```

'250.51' '246' '208' '448' '955' '653' '149' '245' '735' '883' '854'
'952' '838' '194' 'V43' '163' '216' '147' '354' '27' '477' '318' '880'
'921' '377' '471' '683' '175' '602' '250.91' '982' '706' '375' '417'
'131' '347' '870' '148' '862' '61' '817' '914' '360' '684' '314' 'V63'
'36' '57' '240' '915' '971' '795' '988' '452' '963' '327' '731' '842'
'V25' '645' '665' '110' '944' '603' '923' '412' '363' '957' '976' '698'
'299' '700' '273' '974' '97' '529' '66' '98' '605' '941' '52' '806' '84'
'271' '837' '657' '895' '338' '523' '542' '114' '543' '372' 'V70' 'E909'
'583' 'V07' '422' '615' '279' '500' '903' '919' '875' '381' '804' '704'
'23' '58' '649' '832' '133' '975' '833' '391' '690' '10' 'V51']
diag_2: ['?' '250.01' '250' '250.43' '157' '411' '492' '427' '198' '403' '288'
'998' '507' '174' '425' '456' '401' '715' '496' '428' '585' '250.02'
'410' '999' '996' '135' '244' '41' '571' '276' '997' '599' '424' '491'
'553' '707' '286' '440' '493' '242' '70' 'V45' '250.03' '357' '511' '196'
'396' '197' '414' '250.52' '577' '535' '413' '285' '53' '780' '518' '150'
'566' '250.6' '867' '486' 'V15' '8' '788' '340' '574' '581' '228' '530'
'250.82' '786' '294' '567' '785' '512' '305' '729' '250.51' '280' '648'
'560' '618' '444' '38' 'V10' '578' '277' '781' '250.42' '278' '426' '584'
'462' '402' '153' '272' '733' '34' '881' '203' '250.41' '250.13' '293'
'245' '250.12' '558' '787' '342' '573' '626' '303' '250.53' '458' '710'
'415' 'V42' '284' '569' '759' '682' '112' '292' '435' '290' '250.93'
'642' '536' '398' '319' '711' 'E878' '446' '255' 'V44' '250.7' '784'
'300' '562' '162' '287' '447' '789' '790' '591' '200' '154' '304' '117'
'847' '852' '250.83' '250.11' '816' '575' '416' '412' '441' '515' '372'
'482' '382' 'V65' '572' '283' '78' '250.81' '576' '432' '595' '295' 'V12'
'204' '466' '721' '434' '590' '271' '813' '368' '227' '783' '250.5' '258'
'253' '309' '250.91' '519' '333' '459' '250.92' '250.4' '179' '420' '345'
'433' '661' '537' '205' '722' '405' '437' '714' '211' 'E812' '263' '202'
'397' '250.23' 'E932' '201' '301' '723' '614' '568' '861' 'V57' '724'
'189' '297' '453' 'E888' '730' '354' '451' '738' 'E939' '805' 'V43' '155'
'910' '218' '358' '220' 'E937' '583' '958' '794' '564' '436' '250.22'
'620' '621' '331' '617' '596' '314' '378' '250.8' '625' '478' '731' '172'
'404' '681' '470' '279' '281' '531' '443' '799' '337' '311' '719' 'E944'
'423' 'E870' '465' 'E849' '782' '481' '480' 'V23' '199' '79' '438' '348'
'42' 'E950' '473' '627' '726' '54' '490' '317' '332' '508' '369' '600'
'349' '485' '208' '922' '431' '296' 'E934' '753' 'E935' '386' '728' '607'
'E915' '344' '716' '289' '191' '873' '850' '611' '377' '352' '616' 'V17'
'136' '455' '933' 'E885' '860' '513' '603' '484' '223' 'V72' '291' '151'
'V58' '550' '510' '891' '185' '592' '791' '138' '598' '336' '362' '217'
'825' '298' '821' 'E880' '343' '429' 'E879' '579' '225' '250.9' 'V49'
'696' '233' '658' '969' '275' '250.1' '601' '704' '808' 'E890' 'V18'
'920' '380' '570' 'E817' '359' '812' '274' 'V14' '324' '758' 'V66' '911'
'E931' 'E924' '593' '792' '727' 'V46' '394' '532' 'V64' '557' '864' '718'
'E942' '807' '604' '924' '820' '580' '273' '241' '282' '824' 'V61' '646'
'701' '736' '565' '383' '250.2' 'E947' '452' '872' '905' 'E930' '921'
'131' '448' '389' '421' '214' '705' '494' '752' '623' '9' '299' '959'
'365' '967' 'E858' '40' '691' '909' '5' '814' '746' '250.31' '556' '680'
'745' '351' '306' '110' '695' '552' '346' '918' '882' '947' '520' '188'

```

'31' '356' '737' 'V08' '322' '182' '517' '974' 'E929' 'V53' '912' '252'
 '608' '516' 'E933' '94' '702' '923' '594' '647' '111' '934' '430' '487'
 '709' '796' '156' '977' '915' '756' '840' '341' '259' '693' '725' 'V62'
 '528' '683' '953' '457' '501' 'E900' 'V09' '522' '919' '461' '506' '193'
 '483' 'E936' '717' '802' '335' 'V54' '320' '945' '906' '239' '454' '826'
 '823' 'E941' '226' '795' '684' '844' '250.33' '308' '615' '588' '712'
 '663' '706' '833' '741' '713' '533' 'E884' '586' '555' '755' 'E928' '742'
 '869' '962' 'V11' '543' '373' '870' '913' '152' '810' '965' '907' '908'
 '995' '845' '474' '442' '751' '323' '472' '464' '686' '250.32' '540'
 '251' '811' '652' '659' '851' '422' '815' '307' '325' '463' '992' '692'
 '521' '917' 'E965' '524' '916' 'E813' '173' '238' '137' '514' '312' '837'
 '355' '980' '622' '475' '500' '754' '261' '801' '868' '968' '381' '11'
 '250.21' '694' '610' '734' 'E814' '310' '130' '246' '892' '846' '634'
 '75' 'E927' 'E905' '183' '379' 'E917' '163' 'E868' '495' '747' '989'
 'E854' '240' '832' '605' '602' '644' 'V16' '35' 'V70' '376' '266' 'E918'
 '619' '477' '656' '46' '883' '171' 'V13' '698' '842' 'E850' '800' '269'
 '664' 'E887' '952' '164' 'E881' '527' '685' '366' '836' '27' 'V63' '865'
 '793' '232' '990' '52' '831' '327' '542' '806' '972' '862' 'E829' 'E919'
 '944' 'E916' '963' '316' '645' '347' 'V85' '374' 'V02' '748' '256' '186'
 '866' '975' '96' '395' '262' 'E819' '654' '994' '318' 'E826' '879' '674'
 '641' '822' '145' '797' '353' 'E938' 'E816' '948' '987' '99' '192'
 '250.3' 'E906' '534' '115' 'E818' 'E980' '360' '338' '529' '871' '750'
 '212' '302' '955' '141' '88' 'V25' '215' '350' 'V50' 'V03' 'E853' 'E968'
 'E882' '140' '703' '991' '893' 'E821' '235' 'V69' '670' '195' 'V55' '388'
 '268' '894' '114' '260' '853' '7' '880' 'V86' '180' 'E945' '523' '863'
 '649' '270' '665' '460' '942' '364' '66' 'E883' '123' '884' 'V60' '843'
 '927']
 diag_3: ['?' '255' 'V27' '403' '250' 'V45' '38' '486' '996' '197' '250.6' '427'
 '627' '414' '416' '714' '428' '582' 'V43' '250.01' '263' '250.42' '276'
 '482' '401' '250.41' '585' '781' '278' '998' '568' '682' '618' '250.02'
 '305' '707' '496' '599' '715' '424' '518' '553' '794' '411' 'V42' '531'
 '511' '490' '562' '250.8' '250.7' '250.52' '784' '491' '581' '420' '8'
 '724' '730' '789' '131' '250.82' '999' '41' '493' '250.03' '753' '786'
 '529' 'E888' '425' '595' '303' '560' '711' '492' '332' '296' '438' '362'
 '250.4' '654' '244' 'V70' '737' '625' '681' '250.51' '404' 'V10' '810'
 '280' '440' '785' '588' '569' '272' '997' '250.43' '918' '584' '54' '788'
 '426' '722' '250.92' '196' '461' '535' '787' '891' '284' '458' '648'
 '780' '182' '285' '593' '413' '664' '564' '201' '356' 'V15' '292' '782'
 '473' '455' 'E932' '357' '348' '294' '250.23' '459' 'E878' '437' '733'
 '507' '525' '250.53' '397' '572' '805' '453' '331' '736' '402' '591'
 '576' '465' '533' '703' '349' '315' '658' '608' '578' '716' '382' '300'
 '282' '571' '536' '596' '287' '644' 'V11' '558' 'E885' '162' '198' '218'
 '412' '396' 'V14' '570' '433' 'E934' '882' '288' '577' '443' '729' '836'
 '295' '799' '281' '304' '153' '410' '616' '250.83' '601' '291' '75' '512'
 '660' '250.5' '598' '337' '574' '653' 'V58' '311' '415' '386' '602' '790'
 '112' '873' '620' '436' '70' '155' '138' '663' '530' '710' '42' '342'
 '250.91' 'E884' '515' '307' '704' '728' '731' '583' '238' '441' '293'
 '573' '532' '290' '594' '319' '250.13' '250.12' '519' '346' '380' '135'

'642' '698' '924' '905' 'E933' '555' '309' 'E879' '286' '565' '752' '580'
 '446' '444' '344' '252' '35' '813' '394' '301' '575' '258' 'V17' '802'
 '435' '746' 'V12' '709' '881' 'E935' '139' '250.81' '718' '365' '202'
 '334' '185' '398' 'V44' '517' 'E849' '614' '466' '626' '250.9' '368'
 '605' '883' '289' '478' '617' '429' '442' 'V25' '866' '610' '557' '959'
 'E942' '94' '920' '345' '313' '379' '79' '516' '586' '821' '600' '242'
 '373' '592' 'V64' '487' '253' '706' 'E947' '117' '340' 'E950' '656'
 'E949' '590' 'V09' '250.22' '934' '694' '203' '250.93' '995' '726' '923'
 '958' '275' 'E929' '211' 'V18' 'V66' '199' '665' '53' '279' '522' '791'
 '890' '456' 'E938' 'E816' '122' '721' 'V65' '136' '480' '423' 'E920'
 '793' '647' '537' '351' '845' '336' '274' '719' '945' '434' '494' '227'
 '157' '208' '174' 'V57' '812' '734' '150' 'V23' '447' '692' '228' 'V16'
 '756' '405' 'E928' '823' '552' '528' '389' '240' '454' '792' '366' 'E939'
 '907' '270' '310' '266' '387' 'E931' '783' '245' '607' '355' 'E930' '705'
 '372' '369' '611' '283' 'V46' '110' '867' 'E956' '251' '250.2' '820'
 '712' '695' '567' '343' '723' 'V08' '273' '623' '807' '451' '495' '701'
 '34' 'V53' '314' '472' 'E945' '11' '189' '534' '354' '333' 'V54' '277'
 '659' '708' '452' '655' '816' '670' '621' '246' '953' '865' 'E817' '646'
 '151' '378' '78' '298' '840' '641' '521' '745' '619' '912' '506' 'E904'
 '259' 'E870' 'E980' '383' '204' '696' '566' '727' '47' 'E943' '358' '191'
 '965' '921' '432' '27' 'E861' '758' '477' '524' '751' '652' '556' '188'
 '825' '919' '732' '908' '951' '962' '685' 'E850' 'E944' '527' '341' '693'
 '250.1' 'V49' '860' '323' 'V55' '579' '508' '969' '205' '462' 'E880'
 '680' '697' '826' '200' '457' '717' '738' '742' '735' '235' '308' '725'
 '241' '824' '464' '260' '917' '239' '661' '892' '261' 'E883' '943' '744'
 'E936' '796' '318' '967' '350' '854' 'E905' '9' '741' 'E941' '170' '643'
 '317' '759' '909' 'V22' '831' '713' '180' '801' '360' '359' '501' '335'
 '250.11' '306' '811' '690' 'V02' '271' '214' '847' '543' 'V63' '906'
 '842' '686' '445' '808' '861' 'E852' '220' 'E887' 'E858' '915' '970'
 '256' '747' '395' '243' '815' '481' '5' 'E927' '297' '299' '851' '864'
 '922' '384' 'E876' '225' '158' 'E937' '871' '88' '966' 'E917' 'E812'
 'V62' 'E924' '604' '233' 'E916' '377' '797' 'V72' '172' '7' '421' '852'
 'E819' '972' '916' '956' '3' 'E965' '173' '193' '154' '347' '862' '250.3'
 '987' '470' '262' 'E855' '161' '115' '179' '910' '312' '17' '460' '265'
 '66' '163' 'V60' '870' 'E906' '514' '944' '844' '417' '152' '183' '991'
 '216' '385' '164' '935' '510' '814' '485' '850' '250.21' 'E919' '872'
 '195' '431' '597' '933' '171' '884' '156' '868' '483' 'E815' '542' 'V61'
 '853' '374' 'E881' 'E882' 'E822' '192' '754' '327' '523' '500' 'V85'
 '992' '657' '684' '603' 'E826' '550' '913' '376' '755' '361' '186' '720'
 '250.31' '674' '911' 'E813' '226' '365.44' 'E818' '146' '955' 'E894'
 '475' 'V13' '880' '930' 'E915' '381' '132' '353' '795' '893' 'V01' 'E853'
 '863' '540' 'E828' '430' '800' 'E865' '148' 'E946' '822' '879' '848'
 'V86' 'V03' '338' '989' '388' 'E966' '111' 'E922' '123' '757' 'E901'
 '141' '268' 'E892' '649' '702' '948' '223' '484' 'E886' '838' '928' '236'
 '624' '837' 'E987' 'V07' '841' '622' 'E912' 'E955' '463' 'V06' 'E864'
 '217' '877' '391' 'E825' '952' '669' '875' 'E900' '215' '538' '980' '834'
 '448' '175' '49' '876' '230' '57' 'E854' '942' '14' '750' '370' '671'
 '971']

```

max_glu_serum: [nan '>300' 'Norm' '>200']
A1Cresult: [nan '>7' '>8' 'Norm']
metformin: ['No' 'Steady' 'Up' 'Down']
repaglinide: ['No' 'Up' 'Steady' 'Down']
nateglinide: ['No' 'Steady' 'Down' 'Up']
chlorpropamide: ['No' 'Steady' 'Down' 'Up']
glimepiride: ['No' 'Steady' 'Down' 'Up']
acetohexamide: ['No' 'Steady']
glipizide: ['No' 'Steady' 'Up' 'Down']
glyburide: ['No' 'Steady' 'Up' 'Down']
tolbutamide: ['No' 'Steady']
pioglitazone: ['No' 'Steady' 'Up' 'Down']
rosiglitazone: ['No' 'Steady' 'Up' 'Down']
acarbose: ['No' 'Steady' 'Up' 'Down']
miglitol: ['No' 'Steady' 'Down' 'Up']
troglitazone: ['No' 'Steady']
tolazamide: ['No' 'Steady' 'Up']
examide: ['No']
citoglipton: ['No']
insulin: ['No' 'Up' 'Steady' 'Down']
glyburide-metformin: ['No' 'Steady' 'Down' 'Up']
glipizide-metformin: ['No' 'Steady']
glimepiride-pioglitazone: ['No' 'Steady']
metformin-rosiglitazone: ['No' 'Steady']
metformin-pioglitazone: ['No' 'Steady']
change: ['No' 'Ch']
diabetesMed: ['No' 'Yes']
readmitted: ['NO' '>30' '<30']

```

```

[42]: # Fill missing race/gender with 'Unknown'
      # Step 1: Add the new category
      df['race'] = df['race'].cat.add_categories('Unknown')

      # Step 2: Now fill missing values safely
      df['race'] = df['race'].fillna('Unknown')

      df['gender'] = df['gender'].cat.add_categories('Unknown')
      df['gender'] = df['gender'].fillna('Unknown')

      # Optionally drop columns with too many missing values
      threshold = 0.5 # Drop columns with more than 50% missing
      df.dropna(thresh=int(df.shape[0] * (1 - threshold)), axis=1, inplace=True)

      # Fill others with 'Missing' as a label
      for col in df.select_dtypes(include='object').columns:
          df[col] = df[col].fillna('Missing')

```

```
[4]: # Convert gender, race, age into categorical types
categorical_cols = ['race', 'gender', 'age', 'admission_type_id',
                    ↪ 'discharge_disposition_id']
df[categorical_cols] = df[categorical_cols].astype('category')
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 99492 entries, 0 to 101765
Data columns (total 50 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   encounter_id                          99492 non-null  int64
1   patient_nbr                           99492 non-null  int64
2   race                                  99492 non-null  category
3   gender                                99492 non-null  category
4   age                                    99492 non-null  category
5   weight                                3059 non-null   object
6   admission_type_id                     99492 non-null  category
7   discharge_disposition_id              99492 non-null  category
8   admission_source_id                   99492 non-null  int64
9   time_in_hospital                     99492 non-null  int64
10  payer_code                             59781 non-null  object
11  medical_specialty                     50726 non-null  object
12  num_lab_procedures                    99492 non-null  int64
13  num_procedures                         99492 non-null  int64
14  num_medications                       99492 non-null  int64
15  number_outpatient                      99492 non-null  int64
16  number_emergency                       99492 non-null  int64
17  number_inpatient                      99492 non-null  int64
18  diag_1                                99473 non-null  object
19  diag_2                                99156 non-null  object
20  diag_3                                98143 non-null  object
21  number_diagnoses                       99492 non-null  int64
22  max_glu_serum                          5290 non-null   object
23  A1Cresult                              16596 non-null  object
24  metformin                              99492 non-null  object
25  repaglinide                            99492 non-null  object
26  nateglinide                            99492 non-null  object
27  chlorpropamide                         99492 non-null  object
28  glimepiride                            99492 non-null  object
29  acetohexamide                         99492 non-null  object
30  glipizide                              99492 non-null  object
31  glyburide                              99492 non-null  object
32  tolbutamide                            99492 non-null  object
33  pioglitazone                           99492 non-null  object
34  rosiglitazone                          99492 non-null  object
35  acarbose                               99492 non-null  object
```



```

36 miglitol          99492 non-null object
37 troglitazone      99492 non-null object
38 tolazamide        99492 non-null object
39 examide           99492 non-null object
40 citoglipton       99492 non-null object
41 insulin           99492 non-null object
42 glyburide-metformin 99492 non-null object
43 glipizide-metformin 99492 non-null object
44 glimepiride-pioglitazone 99492 non-null object
45 metformin-rosiglitazone 99492 non-null object
46 metformin-pioglitazone 99492 non-null object
47 change            99492 non-null object
48 diabetesMed       99492 non-null object
49 readmitted        99492 non-null object
dtypes: category(5), int64(11), object(34)
memory usage: 35.4+ MB

```

4. Create a binary variable for readmission (e.g., 1 for readmitted within 30 days, 0 otherwise).

```

[6]: # Create binary readmission column
df['readmitted_binary'] = df['readmitted'].apply(lambda x: 1 if x == '<30' else 0)
print(df['readmitted_binary'].value_counts())

```

```

readmitted_binary
0      88323
1      11169
Name: count, dtype: int64

```

5. Demographic Analysis:

- Analyze the distribution of patients across different age groups, genders, and races.
- Examine the relationship between demographics and readmission rates.

```

[ ]: # Set up a common style
sns.set(style="whitegrid")
plt.figure(figsize=(18, 5))

# 1. Age Distribution
plt.subplot(1, 3, 1)
sns.countplot(data=df, x='age', order=sorted(df['age'].unique()))
plt.xticks(rotation=45)
plt.title("Patient Distribution by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Count")

# 2. Gender Distribution
plt.subplot(1, 3, 2)
sns.countplot(data=df, x='gender')

```

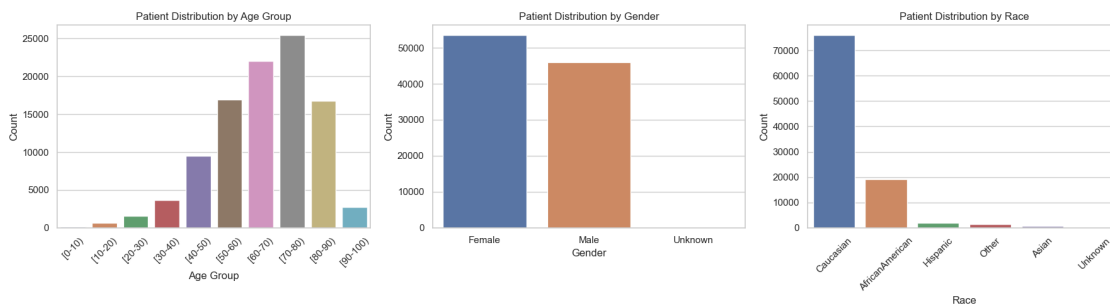
```

plt.title("Patient Distribution by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")

# 3. Race Distribution
plt.subplot(1, 3, 3)
sns.countplot(data=df, x='race', order=df['race'].value_counts().index)
plt.xticks(rotation=45)
plt.title("Patient Distribution by Race")
plt.xlabel("Race")
plt.ylabel("Count")

plt.tight_layout()
plt.show()

```



```

[61]: # Set style
sns.set(style="whitegrid")
plt.figure(figsize=(18, 5))

# 1. Readmission Rate by Age
plt.subplot(1, 3, 1)
age_readmit = df.groupby('age')['readmitted_binary'].mean().
    ↪reindex(sorted(df['age'].unique()))
sns.barplot(x=age_readmit.index, y=age_readmit.values)
plt.title("Readmission Rate by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Readmission Rate")
plt.xticks(rotation=45)

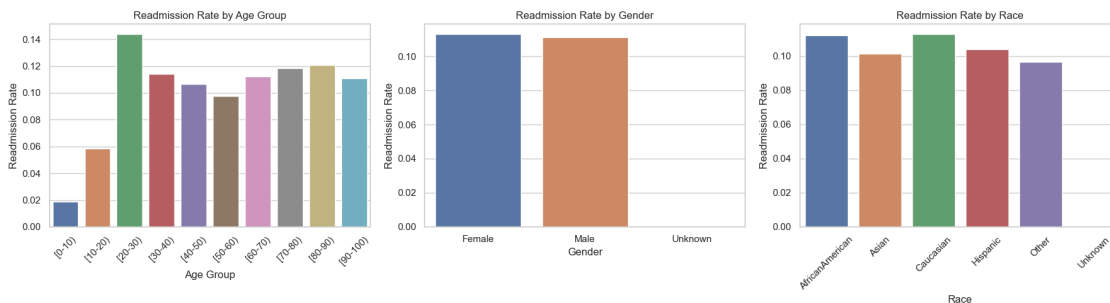
# 2. Readmission Rate by Gender
plt.subplot(1, 3, 2)
gender_readmit = df.groupby('gender')['readmitted_binary'].mean()
sns.barplot(x=gender_readmit.index, y=gender_readmit.values)
plt.title("Readmission Rate by Gender")
plt.xlabel("Gender")

```

```
plt.ylabel("Readmission Rate")

# 3. Readmission Rate by Race
plt.subplot(1, 3, 3)
race_readmit = df.groupby('race')['readmitted_binary'].mean().
    ↪sort_values(ascending=False)
sns.barplot(x=race_readmit.index, y=race_readmit.values)
plt.title("Readmission Rate by Race")
plt.xlabel("Race")
plt.ylabel("Readmission Rate")
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



6. Clinical Analysis:

- Investigate the impact of the number of lab procedures, medications, and diagnoses on readmission.
- Analyze the effect of specific treatments or medication changes on patient outcomes.

```
[7]: # Correlation of lab procedures, medications, diagnoses
import seaborn as sns
import matplotlib.pyplot as plt

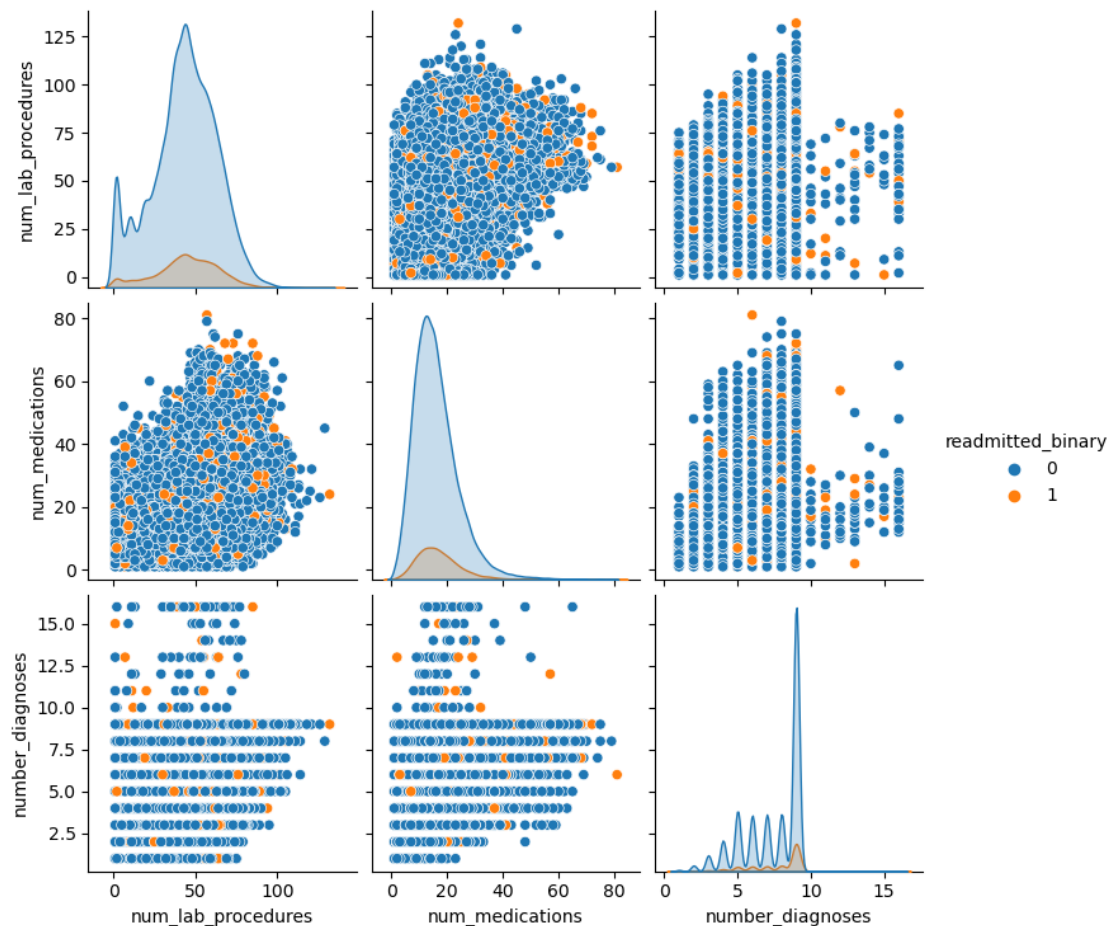
clinical_vars = ['num_lab_procedures', 'num_medications', 'number_diagnoses']
sns.pairplot(df[clinical_vars + ['readmitted_binary']], hue='readmitted_binary')
```

```
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119:
```

FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.

```
with pd.option_context('mode.use_inf_as_na', True):
```

[7]: <seaborn.axisgrid.PairGrid at 0x1cdec8c37d0>



```
[8]: # Set style
sns.set(style="whitegrid")
plt.figure(figsize=(18, 5))

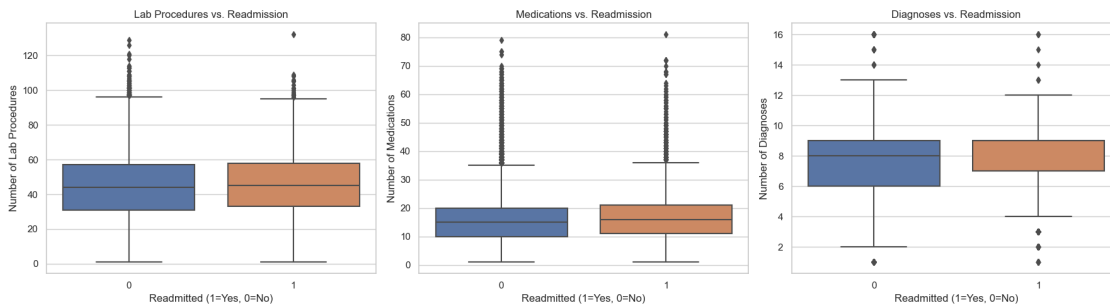
# 1. Lab Procedures
plt.subplot(1, 3, 1)
sns.boxplot(x='readmitted_binary', y='num_lab_procedures', data=df)
plt.title("Lab Procedures vs. Readmission")
plt.xlabel("Readmitted (1=Yes, 0=No)")
plt.ylabel("Number of Lab Procedures")

# 2. Medications
```

```
plt.subplot(1, 3, 2)
sns.boxplot(x='readmitted_binary', y='num_medications', data=df)
plt.title("Medications vs. Readmission")
plt.xlabel("Readmitted (1=Yes, 0=No)")
plt.ylabel("Number of Medications")

# 3. Diagnoses
plt.subplot(1, 3, 3)
sns.boxplot(x='readmitted_binary', y='number_diagnoses', data=df)
plt.title("Diagnoses vs. Readmission")
plt.xlabel("Readmitted (1=Yes, 0=No)")
plt.ylabel("Number of Diagnoses")

plt.tight_layout()
plt.show()
```



```
[9]: df[['num_lab_procedures', 'num_medications', 'number_diagnoses',
        ↪ 'readmitted_binary']].corr()
```

```
[9]:
```

	num_lab_procedures	num_medications	number_diagnoses	\
num_lab_procedures	1.000000	0.267013	0.151565	
num_medications	0.267013	1.000000	0.258623	
number_diagnoses	0.151565	0.258623	1.000000	
readmitted_binary	0.020482	0.039476	0.049625	


```

        readmitted_binary
num_lab_procedures    0.020482
num_medications       0.039476
number_diagnoses      0.049625
readmitted_binary     1.000000
```

```
[10]: # Medication vs. Readmission Rate
# We'll group by each medication column and compute how often patients are
        ↪ readmitted.
# Choose key treatment columns
```

```

medications = ['insulin', 'metformin', 'glipizide', 'glyburide', 'glimepiride']

# Print readmission rate per medication type
for med in medications:
    print(f"\n--- {med.upper()} ---")
    print(df.groupby(med)['readmitted_binary'].mean())

```

--- INSULIN ---

insulin

Down 0.140045

No 0.100871

Steady 0.111799

Up 0.131151

Name: readmitted_binary, dtype: float64

--- METFORMIN ---

metformin

Down 0.120996

No 0.115880

Steady 0.097545

Up 0.082772

Name: readmitted_binary, dtype: float64

--- GLIPIZIDE ---

glipizide

Down 0.154412

No 0.111800

Steady 0.112512

Up 0.131300

Name: readmitted_binary, dtype: float64

--- GLYBURIDE ---

glyburide

Down 0.095238

No 0.112868

Steady 0.107755

Up 0.107053

Name: readmitted_binary, dtype: float64

--- GLIMEPIRIDE ---

glimepiride

Down 0.129032

No 0.112812

Steady 0.100287

Up 0.109034

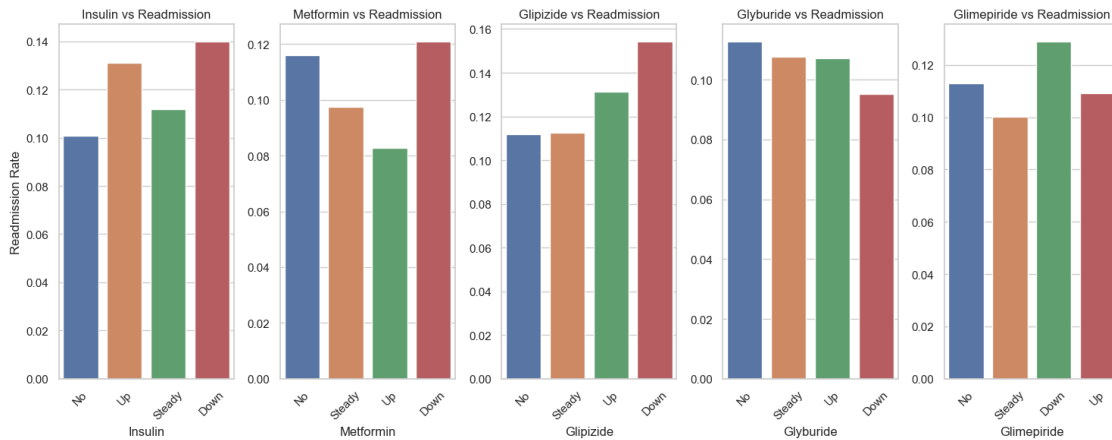
Name: readmitted_binary, dtype: float64

```
[13]: import seaborn as sns
import matplotlib.pyplot as plt

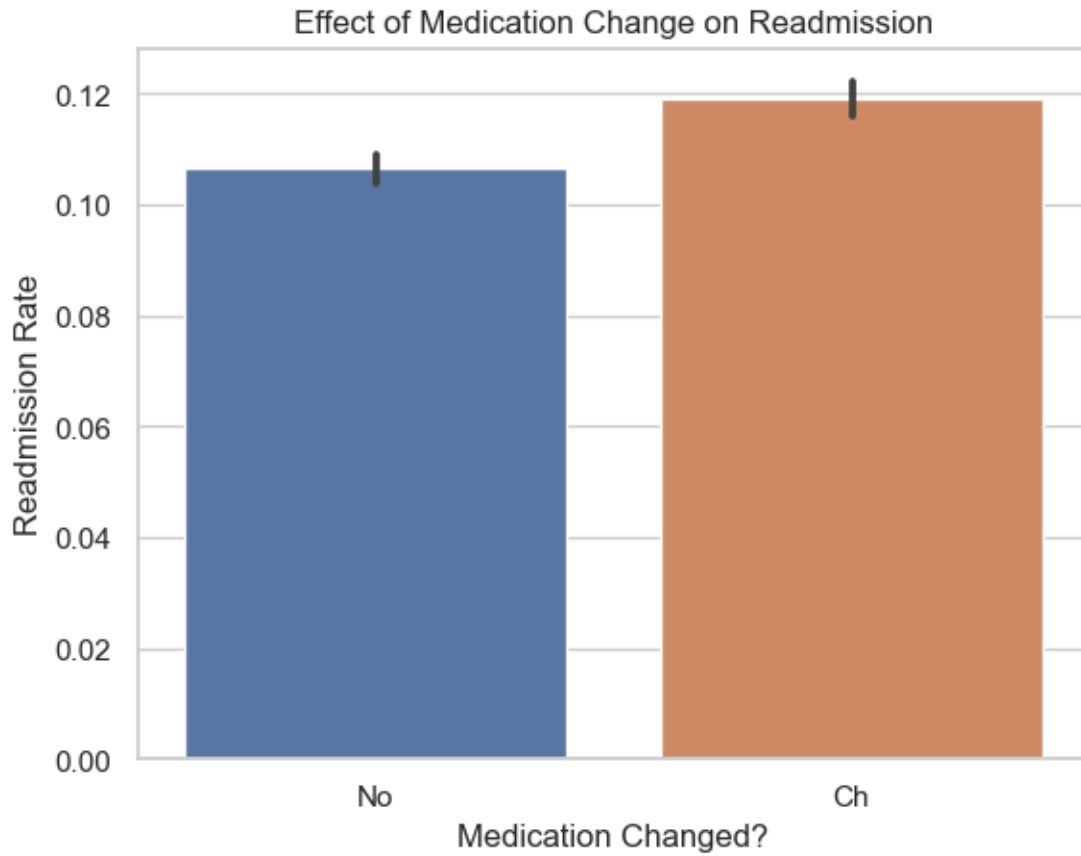
plt.figure(figsize=(15, 6))

# Create a bar plot for each medication
for i, med in enumerate(medications, 1):
    plt.subplot(1, len(medications), i)
    sns.barplot(data=df, x=med, y='readmitted_binary', ci=None)
    plt.title(f"{med.capitalize()} vs Readmission")
    plt.ylabel("Readmission Rate" if i == 1 else "")
    plt.xlabel(med.capitalize())
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



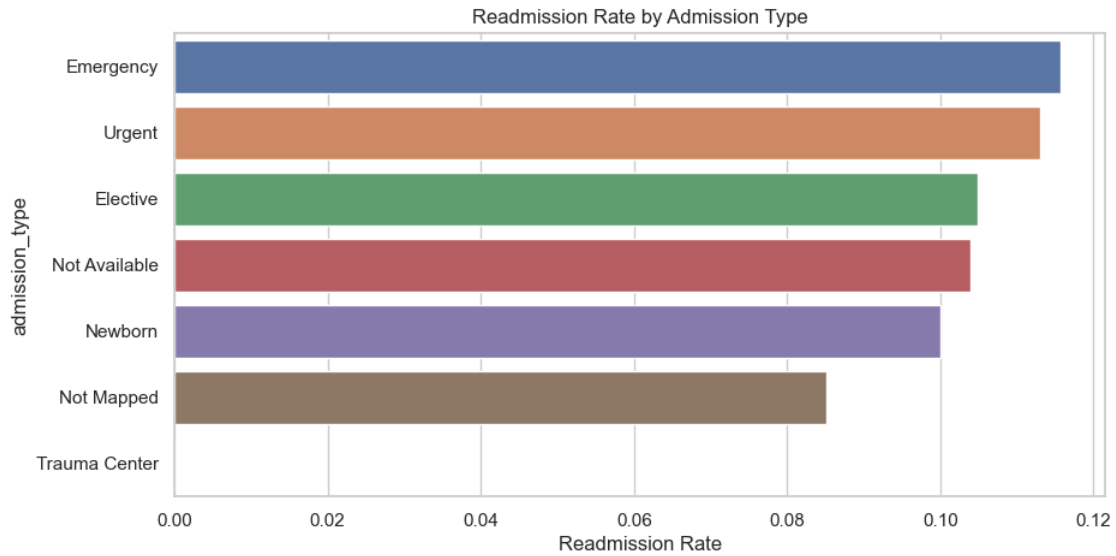
```
[20]: # Distribution of readmission based on whether medication changed
sns.barplot(data=df, x='change', y='readmitted_binary')
plt.title("Effect of Medication Change on Readmission")
plt.xlabel("Medication Changed?")
plt.ylabel("Readmission Rate")
plt.show()
```



Admission Analysis: - Assess how admission types and discharge dispositions correlate with readmission rates. - Evaluate the length of hospital stay in relation to readmission.

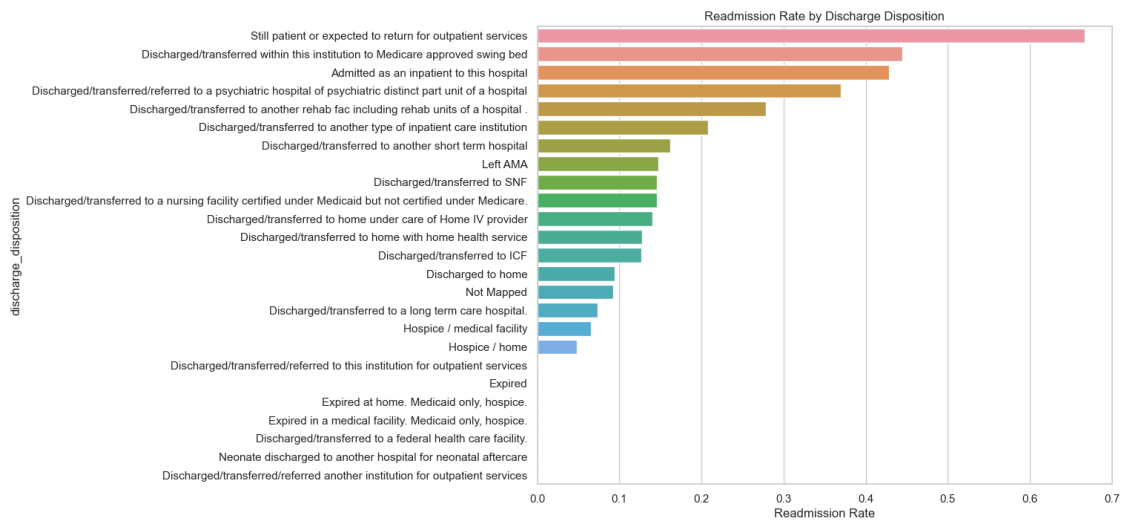
```
[22]: # Admission type vs readmission rate
admission_rates = df.groupby('admission_type')['readmitted_binary'].mean().
    ↪sort_values(ascending=False)

plt.figure(figsize=(10, 5))
sns.barplot(x=admission_rates.values, y=admission_rates.index)
plt.xlabel("Readmission Rate")
plt.title("Readmission Rate by Admission Type")
plt.show()
```

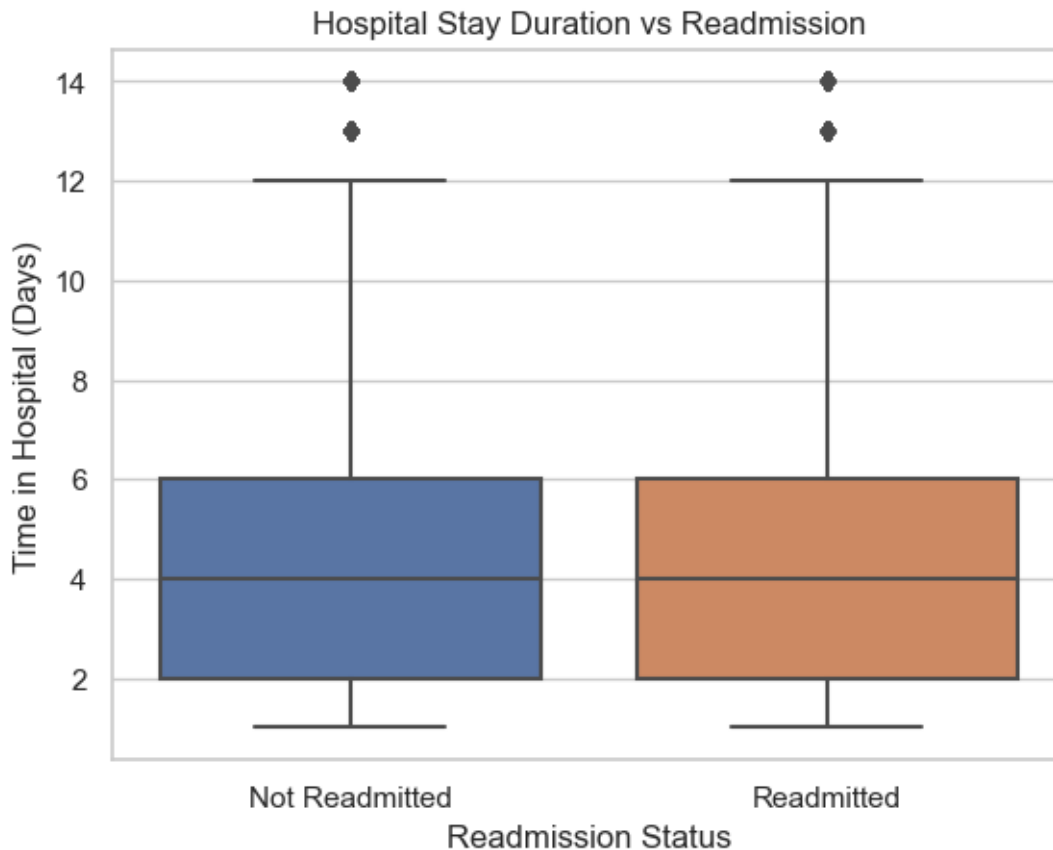



```
[24]: # Discharge disposition vs readmission rate
discharge_rates = df.groupby('discharge_disposition')['readmitted_binary'].
    ↪mean().sort_values(ascending=False)

plt.figure(figsize=(10, 8))
sns.barplot(x=discharge_rates.values, y=discharge_rates.index)
plt.xlabel("Readmission Rate")
plt.title("Readmission Rate by Discharge Disposition")
plt.show()
```



```
[25]: sns.boxplot(data=df, x='readmitted_binary', y='time_in_hospital')
plt.xticks([0, 1], ["Not Readmitted", "Readmitted"])
plt.xlabel("Readmission Status")
plt.ylabel("Time in Hospital (Days)")
plt.title("Hospital Stay Duration vs Readmission")
plt.show()
```

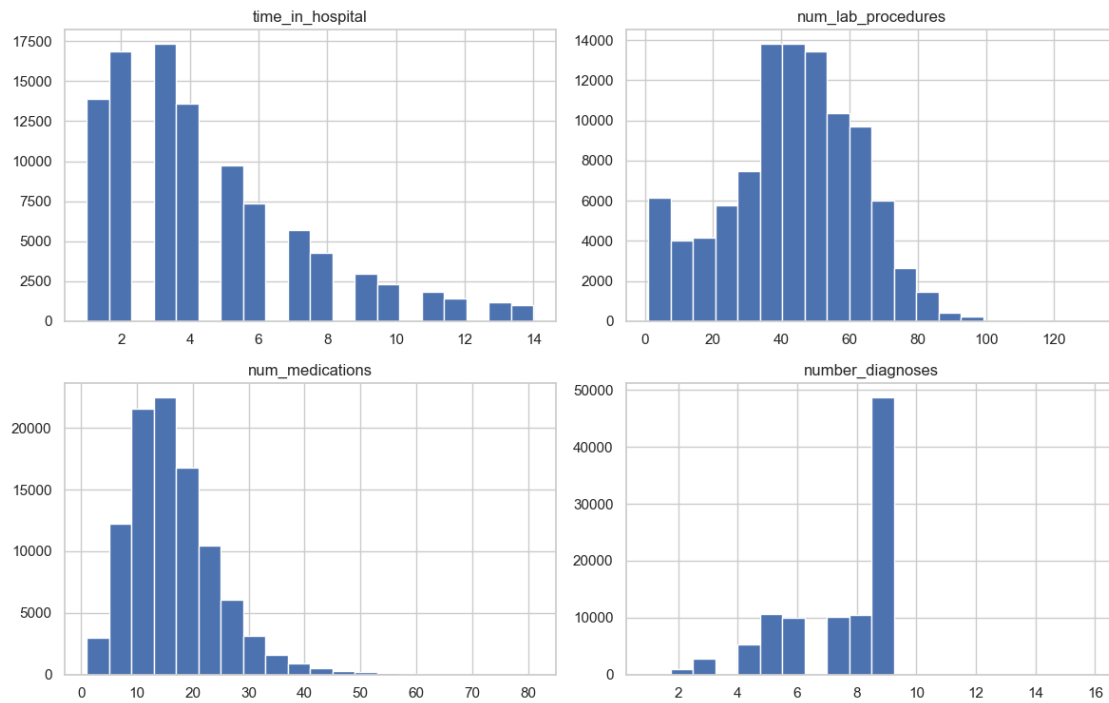


Data Visualization: - Visualize distributions of demographic and clinical variables. - Develop plots to compare numerical variables across readmission statuses. - Display correlations between various factors and readmission rates. - Generate plots to identify potential patterns or outliers in the data.

```
[ ]: # demographic already done before.

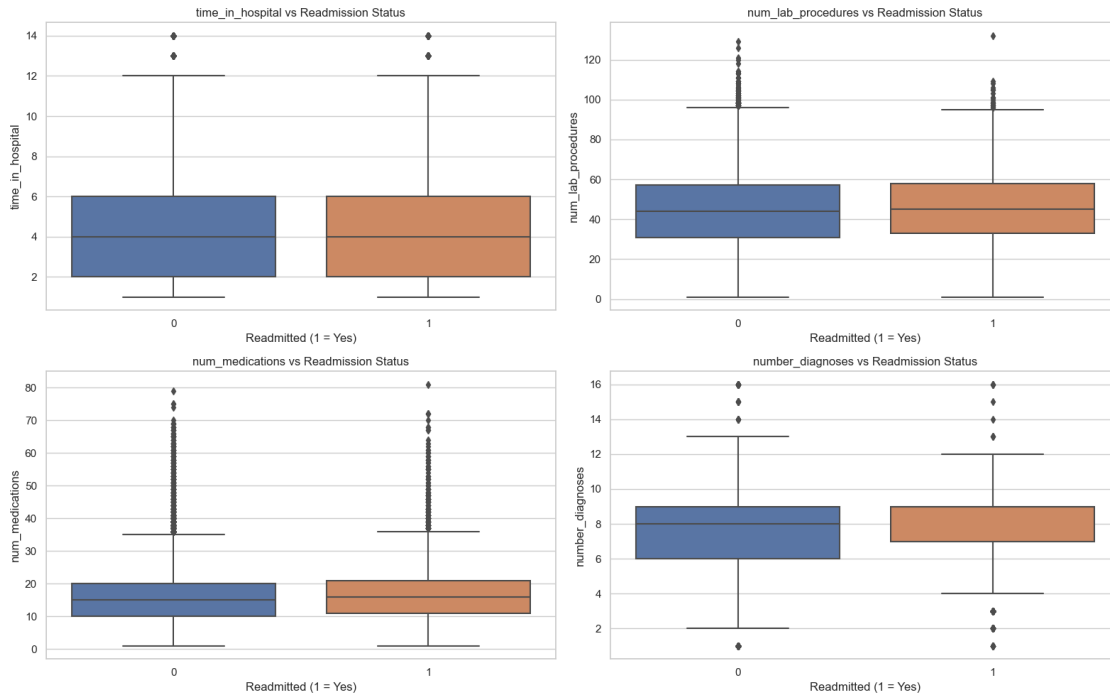
clinical_vars = ['time_in_hospital', 'num_lab_procedures', 'num_medications', '
↳ 'number_diagnoses']
df[clinical_vars].hist(figsize=(12, 8), bins=20)
plt.suptitle("Distribution of Clinical Variables", fontsize=16)
plt.tight_layout()
plt.show()
```

Distribution of Clinical Variables



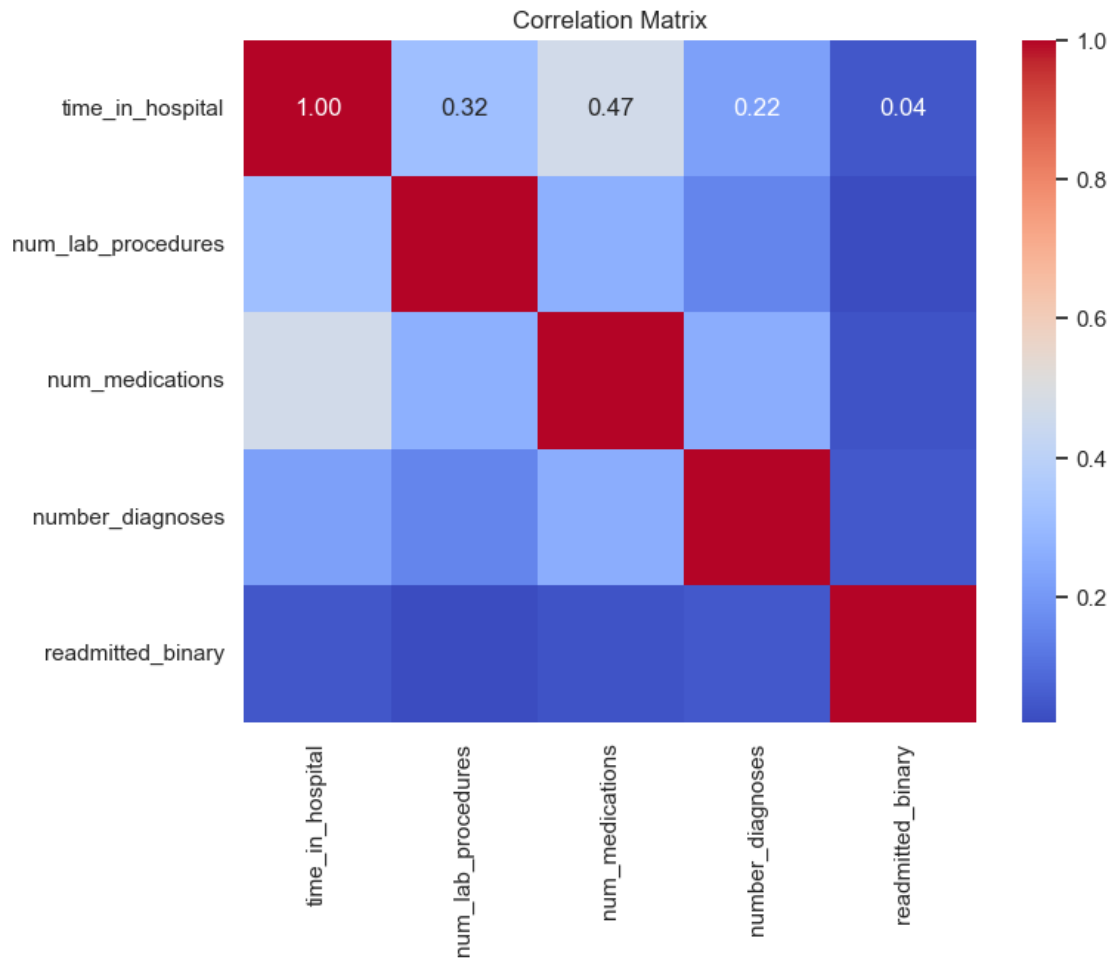
```
[27]: plt.figure(figsize=(16, 10))
      for i, col in enumerate(clinical_vars):
          plt.subplot(2, 2, i+1)
          sns.boxplot(data=df, x='readmitted_binary', y=col)
          plt.title(f"{col} vs Readmission Status")
          plt.xlabel("Readmitted (1 = Yes)")
          plt.ylabel(col)

      plt.tight_layout()
      plt.show()
```



```
[28]: # Compute correlations
correlation = df[clinical_vars + ['readmitted_binary']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



```
[29]: sns.pairplot(df[clinical_vars + ['readmitted_binary']],  
    hue='readmitted_binary', palette='husl')  
plt.suptitle("Pairwise Relationships and Clusters", y=1.02)  
plt.show()
```

Pairwise Relationships and Clusters

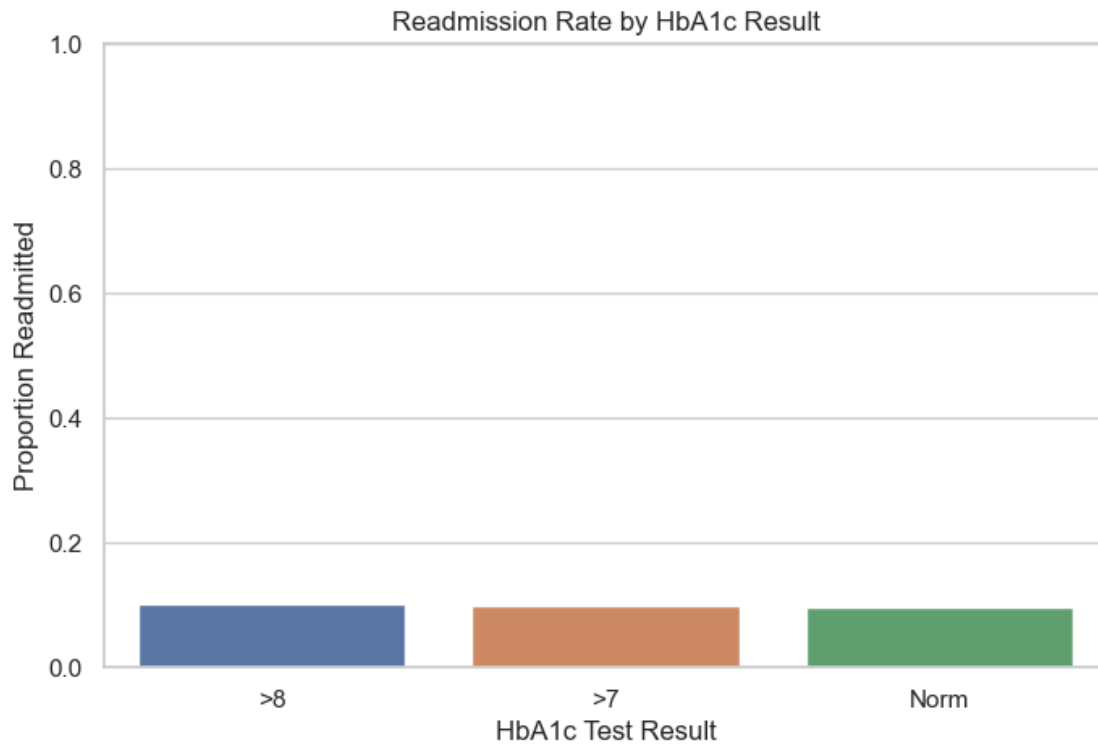


Impact of HbA1c Testing on Readmission Rates: - Evaluate how performing an HbA1c test during hospitalization influences 30- day readmission rates.

```
[30]: # To evaluate the Impact of HbA1c Testing on Readmission Rates, we'll focus on
      ↪ the column A1Cresult. This column reflects whether an HbA1c test was done
      ↪ and what the result was. Common values include:
      # 'None' - No test done
      # 'Norm' - Normal level
      # '>7' - High level
      # '>8' - Very high level
      # We will analyze how these categories are distributed across the
      ↪ readmitted_binary outcome.

      #Readmission Rate by HbA1c Testing Status
      hba1c_readmit = df.groupby('A1Cresult')['readmitted_binary'].mean().
      ↪ sort_values(ascending=False)
```

```
plt.figure(figsize=(8, 5))
sns.barplot(x=hba1c_readmit.index, y=hba1c_readmit.values)
plt.title("Readmission Rate by HbA1c Result")
plt.xlabel("HbA1c Test Result")
plt.ylabel("Proportion Readmitted")
plt.ylim(0, 1)
plt.show()
```



A higher readmission rate for patients with HbA1c results >7 or >8 may suggest poorly controlled diabetes, which leads to complications and frequent hospital visits.

If None has a similar or higher rate, it could mean a missed opportunity for assessing diabetes control during hospitalization.

```
[31]: plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='A1Cresult', hue='readmitted_binary')
plt.title("Count of Patients by HbA1c Result and Readmission")
plt.xlabel("HbA1c Test Result")
plt.ylabel("Count")
plt.legend(title="Readmitted", labels=["No", "Yes"])
plt.show()
```

```

AttributeError                                Traceback (most recent call last)
Cell In[31], line 2
      1 plt.figure(figsize=(8, 5))
----> 2 sns.countplot(data=df, x='A1Cresult', hue='readmitted_binary')
      3 plt.title("Count of Patients by HbA1c Result and Readmission")
      4 plt.xlabel("HbA1c Test Result")

File c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\categorical.py:2955, in
countplot(data, x, y, hue, order, hue_order, orient, color, palette,
saturation, width, dodge, ax, **kwargs)
    2952 if ax is None:
    2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
    2956 return ax

File c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\categorical.py:1587, in
_BarPlotter.plot(self, ax, bar_kws)
    1585 """Make the plot."""
    1586 self.draw_bars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
    1588 if self.orient == "h":
    1589     ax.invert_yaxis()

File c:\Users\ADMIN\anaconda3\Lib\site-packages\seaborn\categorical.py:767, in
_CategoricalPlotter.annotate_axes(self, ax)
    764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
    766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

File c:\Users\ADMIN\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:322, in
Axes.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
    (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
    321     """
--> 322     handles, labels, kwargs = mlegend._parse_legend_args([self], *args,
    **kwargs)
    323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    324     self.legend._remove_method = self._remove_legend

File c:\Users\ADMIN\anaconda3\Lib\site-packages\matplotlib\legend.py:1361, in
_parse_legend_args(axs, handles, labels, *args, **kwargs)
    1357     handles = [handle for handle, label
    1358                in zip(_get_legend_handles(axs, handlers), labels)]

```



```

1360 elif len(args) == 0: # 0 args: automatically detect labels and handles
-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
1362     if not handles:
1363         log.warning(
1364             "No artists with labels found to put in legend. Note that
1365             "artists whose label start with an underscore are ignored "
1366             "when legend() is called with no argument.")

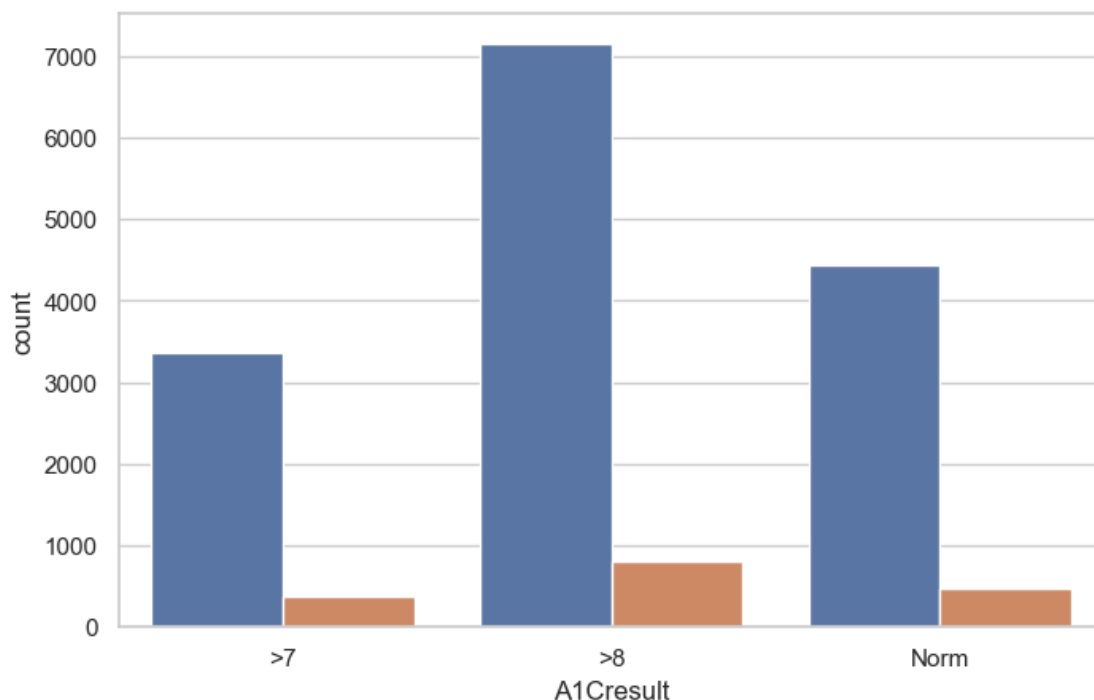
```

```

File c:\Users\ADMIN\anaconda3\Lib\site-packages\matplotlib\legend.py:1291, in _
-> _get_legend_handles_labels(axs, legend_handler_map)
1289 for handle in _get_legend_handles(axs, legend_handler_map):
1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
1292         handles.append(handle)
1293         labels.append(label)

```

AttributeError: 'numpy.int64' object has no attribute 'startswith'



Influence of Discharge Disposition on Patient Outcomes: - Analyze how different discharge dispositions affect the likelihood of 30-day readmissions.

```

[32]: # Group by discharge_disposition and Compare Readmission Rate

# Ensure the mapping file was merged earlier

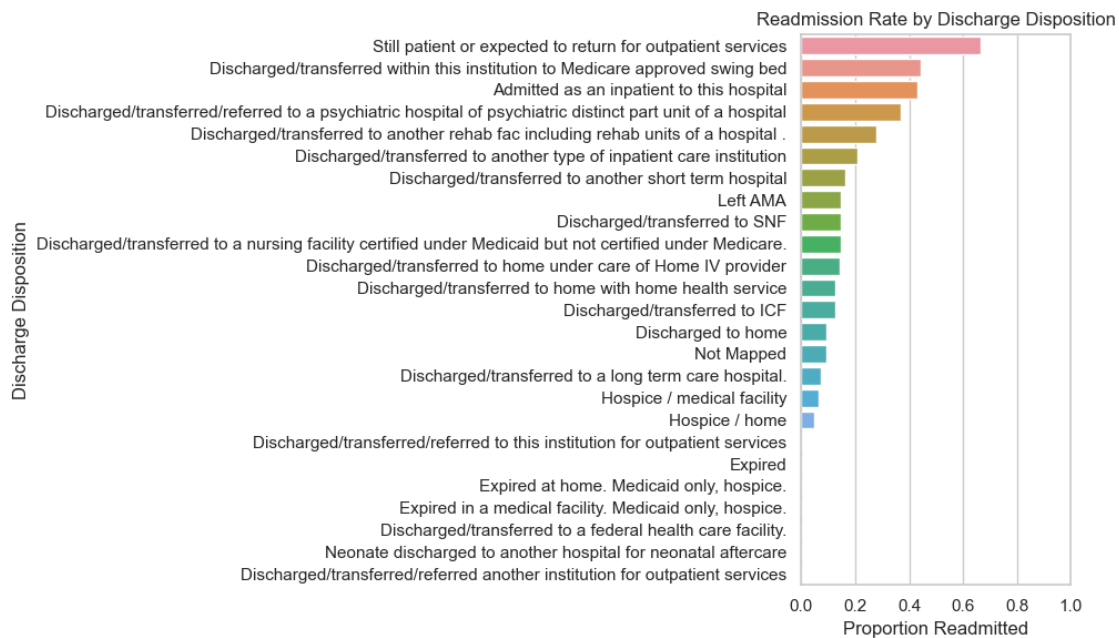
```

```

readmit_by_discharge = df.groupby('discharge_disposition')['readmitted_binary'].
    ↪mean().sort_values(ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x=readmit_by_discharge.values, y=readmit_by_discharge.index)
plt.title("Readmission Rate by Discharge Disposition")
plt.xlabel("Proportion Readmitted")
plt.ylabel("Discharge Disposition")
plt.xlim(0, 1)
plt.tight_layout()
plt.show()

```



```

[33]: # Add Sample Sizes for Context
# To see whether categories with high readmission rates have enough patients to
    ↪be significant:

discharge_counts = df['discharge_disposition'].value_counts()
readmit_counts = df[df['readmitted_binary'] == 1]['discharge_disposition'].
    ↪value_counts()

discharge_df = pd.DataFrame({
    'Total': discharge_counts,
    'Readmitted': readmit_counts
}).fillna(0)

```

```

discharge_df['Readmission Rate'] = discharge_df['Readmitted'] /
↳discharge_df['Total']
discharge_df = discharge_df.sort_values('Readmission Rate', ascending=False)

discharge_df.head(10) # Optional: Show top 10

```

[33]:

	Total	Readmitted \
discharge_disposition		
Still patient or expected to return for outpati...	3	2.0
Discharged/transferred within this institution ...	63	28.0
Admitted as an inpatient to this hospital	21	9.0
Discharged/transferred/referred to a psychiatri...	138	51.0
Discharged/transferred to another rehab fac inc...	1981	552.0
Discharged/transferred to another type of inpat...	1143	238.0
Discharged/transferred to another short term ho...	2061	333.0
Left AMA	611	90.0
Discharged/transferred to SNF	13614	1990.0
Discharged/transferred to a nursing facility ce...	48	7.0

	Readmission Rate
discharge_disposition	
Still patient or expected to return for outpati...	0.666667
Discharged/transferred within this institution ...	0.444444
Admitted as an inpatient to this hospital	0.428571
Discharged/transferred/referred to a psychiatri...	0.369565
Discharged/transferred to another rehab fac inc...	0.278647
Discharged/transferred to another type of inpat...	0.208224
Discharged/transferred to another short term ho...	0.161572
Left AMA	0.147300
Discharged/transferred to SNF	0.146173
Discharged/transferred to a nursing facility ce...	0.145833

12. Effect of Time in Hospital on Medication Changes and Readmission:

- Investigate the relationship between the length of hospital stay, changes in medication during the stay, and subsequent readmissions.

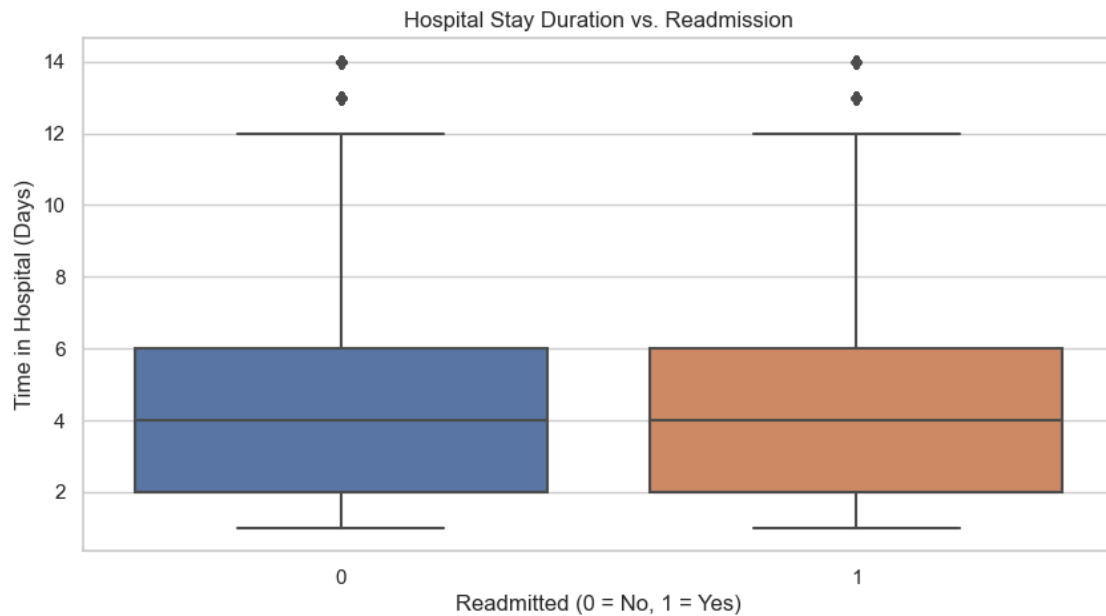
[34]:

```

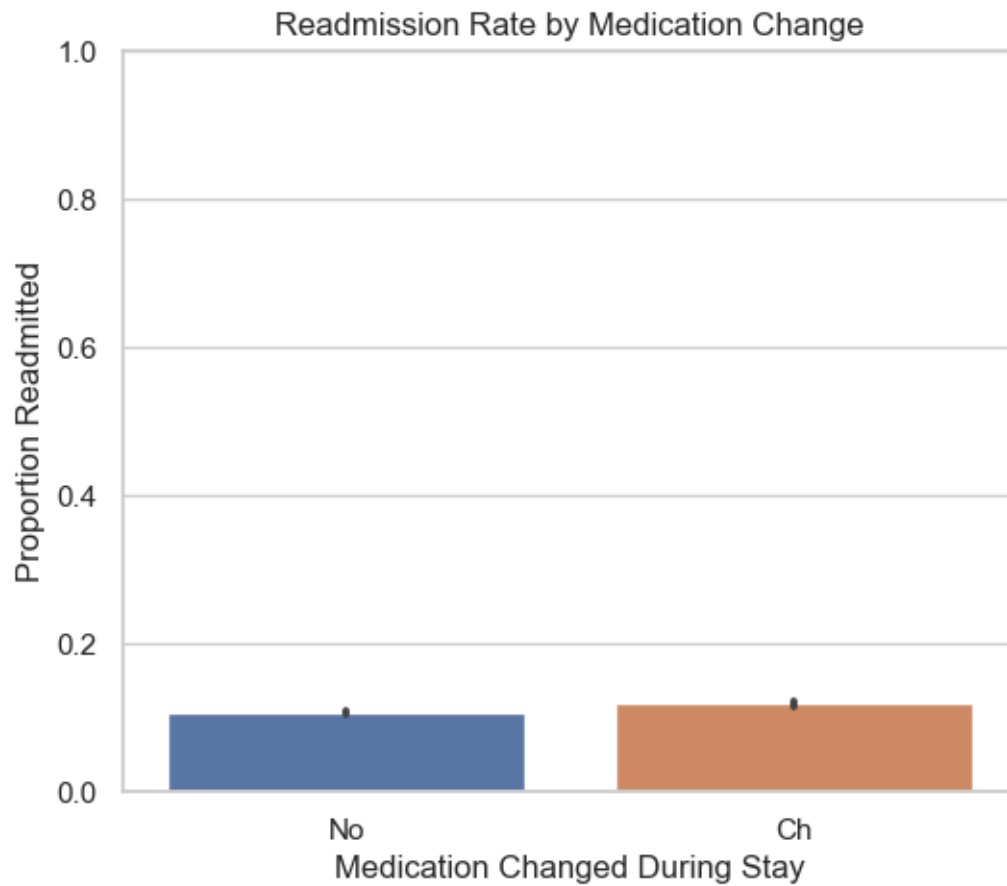
# We explore the Effect of Time in Hospital on Medication Changes and
↳Readmission -
# this can offer insight into whether longer stays and medication adjustments
↳are
# related to higher chances of readmission.
# To analyze:
# How length of hospital stay (time_in_hospital) influences readmission.
# Whether patients with a medication change (change) have a higher likelihood
↳of being readmitted.
# The combined effect of hospital stay and medication changes on readmission
↳rates.

```

```
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='readmitted_binary', y='time_in_hospital')
plt.title("Hospital Stay Duration vs. Readmission")
plt.xlabel("Readmitted (0 = No, 1 = Yes)")
plt.ylabel("Time in Hospital (Days)")
plt.show()
```

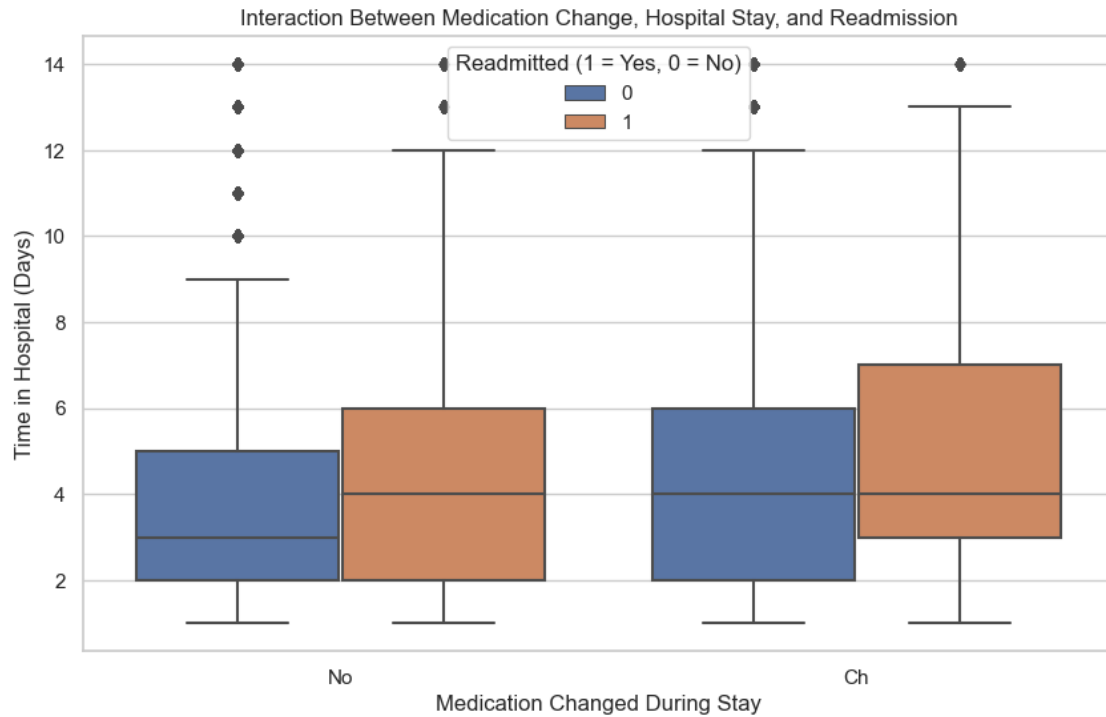


```
[35]: # Compare Readmission Rate by Medication Change
plt.figure(figsize=(6, 5))
sns.barplot(data=df, x='change', y='readmitted_binary')
plt.title("Readmission Rate by Medication Change")
plt.xlabel("Medication Changed During Stay")
plt.ylabel("Proportion Readmitted")
plt.ylim(0, 1)
plt.show()
```



```
[36]: # Combined Effect: Medication Change + Time in Hospital

plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='change', y='time_in_hospital', hue='readmitted_binary')
plt.title("Interaction Between Medication Change, Hospital Stay, and Readmission")
plt.xlabel("Medication Changed During Stay")
plt.ylabel("Time in Hospital (Days)")
plt.legend(title="Readmitted (1 = Yes, 0 = No)")
plt.show()
```

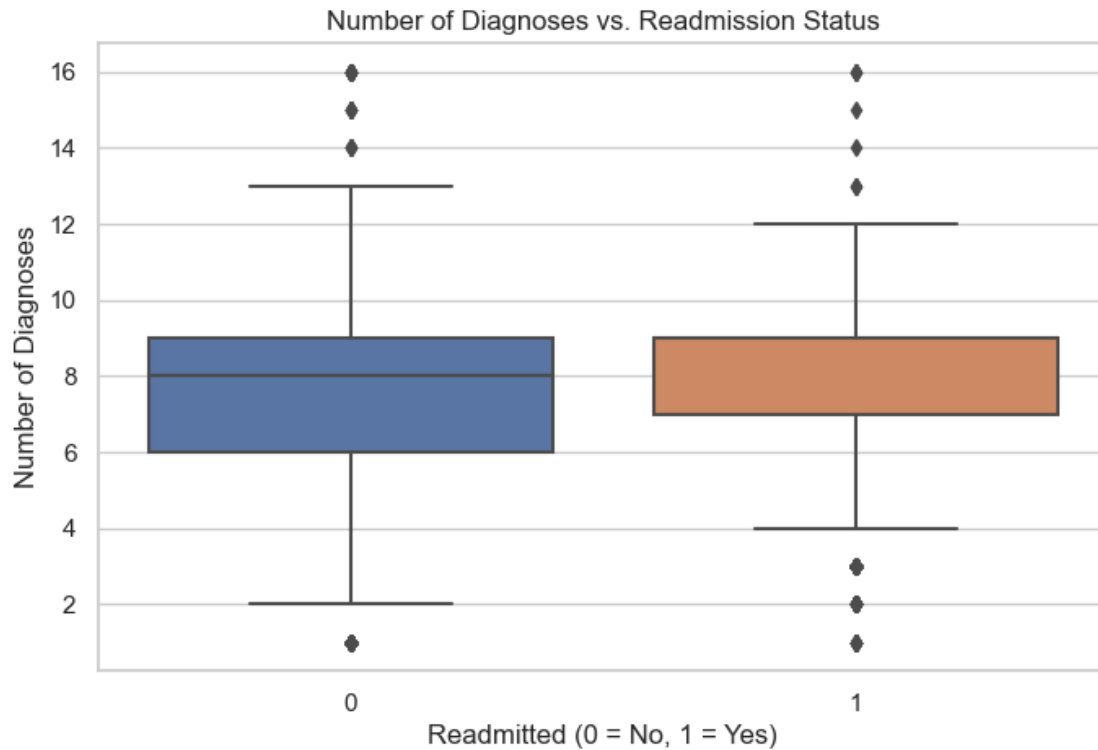


12. Association Between Number of Diagnoses and Readmission Likelihood:

- Examine how the total number of diagnoses impacts the probability of a patient being readmitted within 30 days.

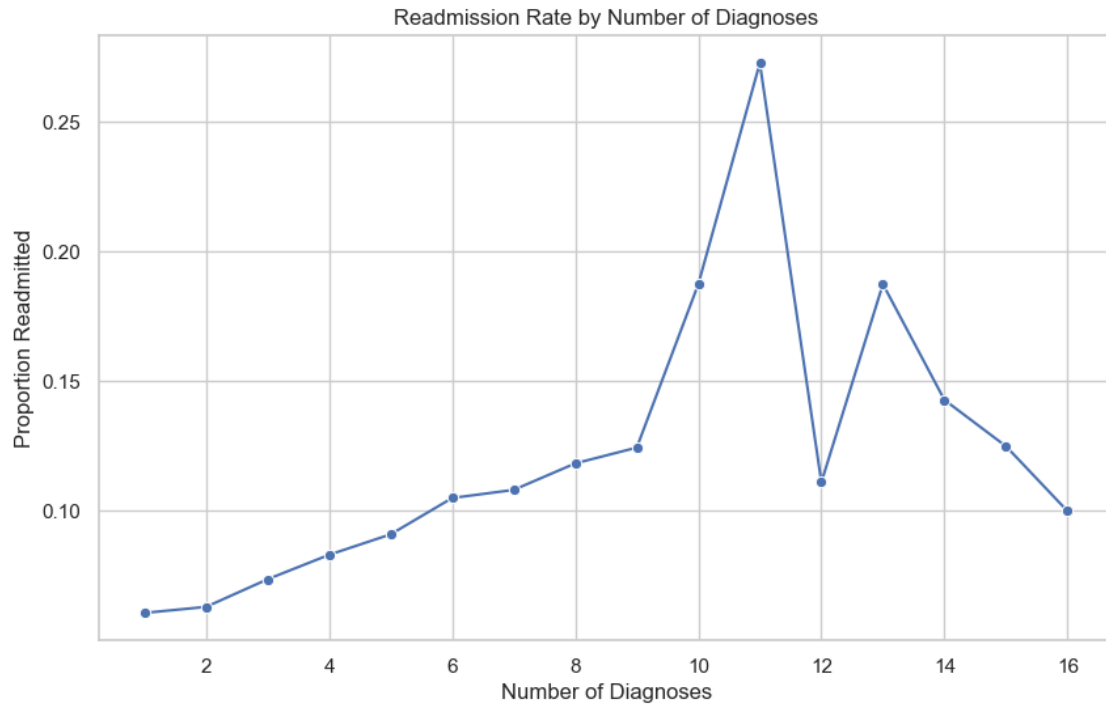
```
[37]: # To investigate whether a higher number of diagnoses (i.e., comorbidities)
      ↪ increases the likelihood of readmission within 30 days.
      # In our dataset, this is represented by the number_diagnoses variable (i.e.,
      ↪ the number of ICD-9 codes assigned).

      # Analysis: Number of Diagnoses vs. Readmission
      # Diagnoses Count vs. Readmission
      plt.figure(figsize=(8, 5))
      sns.boxplot(data=df, x='readmitted_binary', y='number_diagnoses')
      plt.title("Number of Diagnoses vs. Readmission Status")
      plt.xlabel("Readmitted (0 = No, 1 = Yes)")
      plt.ylabel("Number of Diagnoses")
      plt.show()
```



```
[38]: # Mean Readmission Rate per Diagnosis Count
diag_readmit = df.groupby('number_diagnoses')['readmitted_binary'].mean()

plt.figure(figsize=(10, 6))
sns.lineplot(x=diag_readmit.index, y=diag_readmit.values, marker="o")
plt.title("Readmission Rate by Number of Diagnoses")
plt.xlabel("Number of Diagnoses")
plt.ylabel("Proportion Readmitted")
plt.grid(True)
plt.show()
```



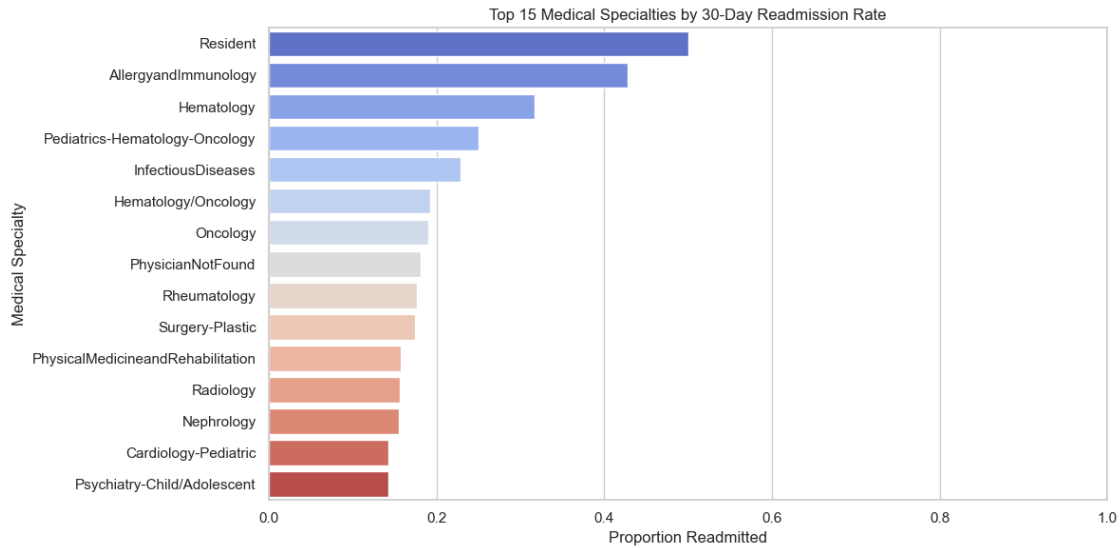
13. Role of Medical Specialty in Patient Readmission:

- Investigate whether the specialty of the attending physician influences 30-day readmission rates.

```
[39]: # Drop missing or unknown specialties
specialty_data = df[df['medical_specialty'].notna() & (df['medical_specialty'] != '?')]

# Group by specialty and calculate readmission rates
specialty_readmit = specialty_data.
    ↳groupby('medical_specialty')['readmitted_binary'].mean().
    ↳sort_values(ascending=False)

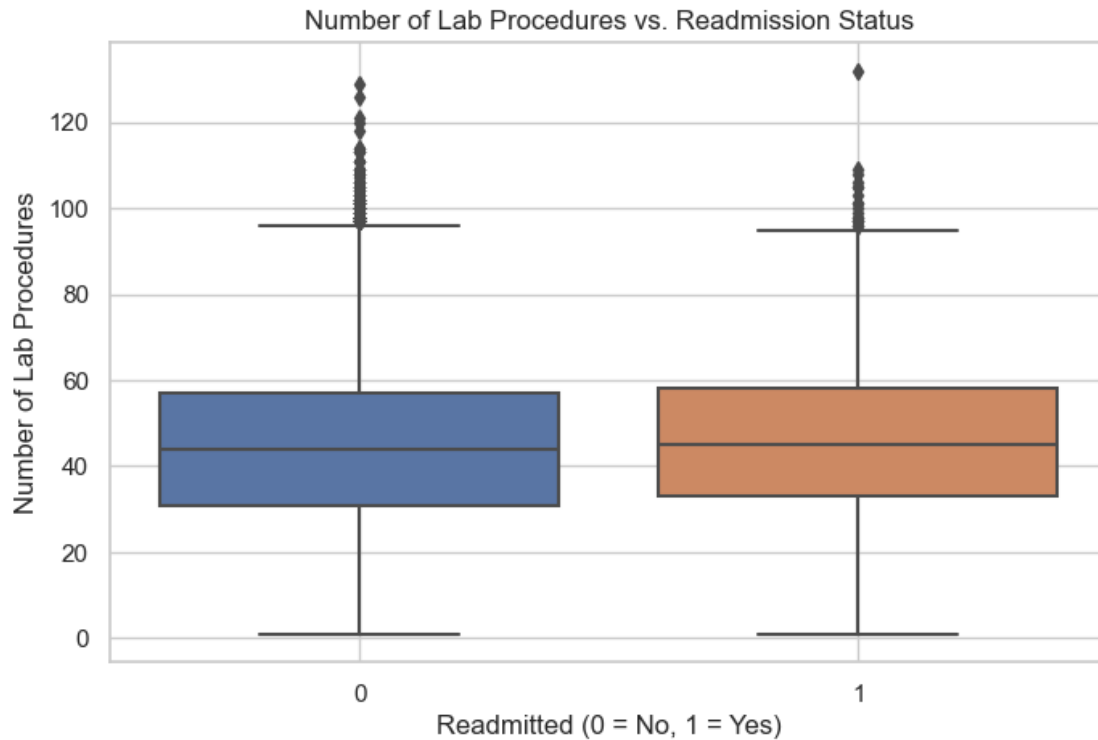
# Plot top 15 specialties by readmission rate
plt.figure(figsize=(12, 6))
sns.barplot(x=specialty_readmit.head(15).values, y=specialty_readmit.head(15).
    ↳index, palette='coolwarm')
plt.title("Top 15 Medical Specialties by 30-Day Readmission Rate")
plt.xlabel("Proportion Readmitted")
plt.ylabel("Medical Specialty")
plt.xlim(0, 1)
plt.grid(True, axis='x')
plt.tight_layout()
plt.show()
```

14. Correlation Between Laboratory Procedures and Readmission:

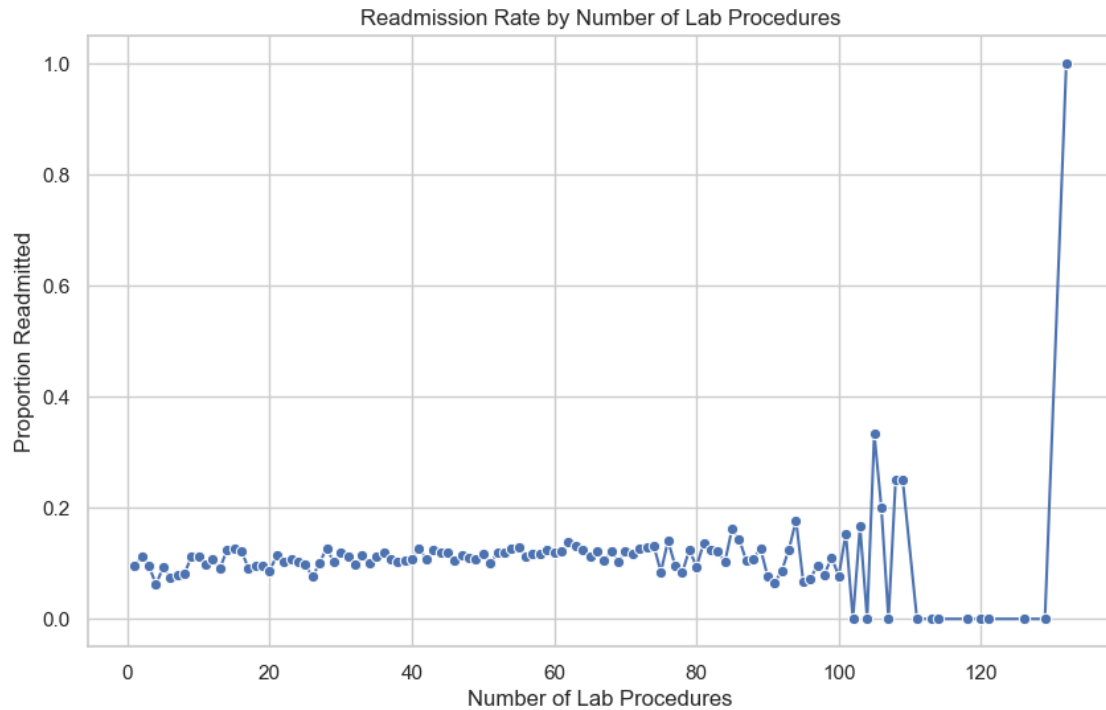
- Assess how the number of laboratory procedures conducted during a hospital stay relates to the likelihood of readmission.

```
[40]: # Lab Procedures by Readmission Status
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='readmitted_binary', y='num_lab_procedures')
plt.title("Number of Lab Procedures vs. Readmission Status")
plt.xlabel("Readmitted (0 = No, 1 = Yes)")
plt.ylabel("Number of Lab Procedures")
plt.grid(True)
plt.show()
```



```
[41]: # Average Readmission Rate by Number of Lab Procedures
lab_readmit = df.groupby('num_lab_procedures')['readmitted_binary'].mean()

plt.figure(figsize=(10, 6))
sns.lineplot(x=lab_readmit.index, y=lab_readmit.values, marker='o')
plt.title("Readmission Rate by Number of Lab Procedures")
plt.xlabel("Number of Lab Procedures")
plt.ylabel("Proportion Readmitted")
plt.grid(True)
plt.show()
```

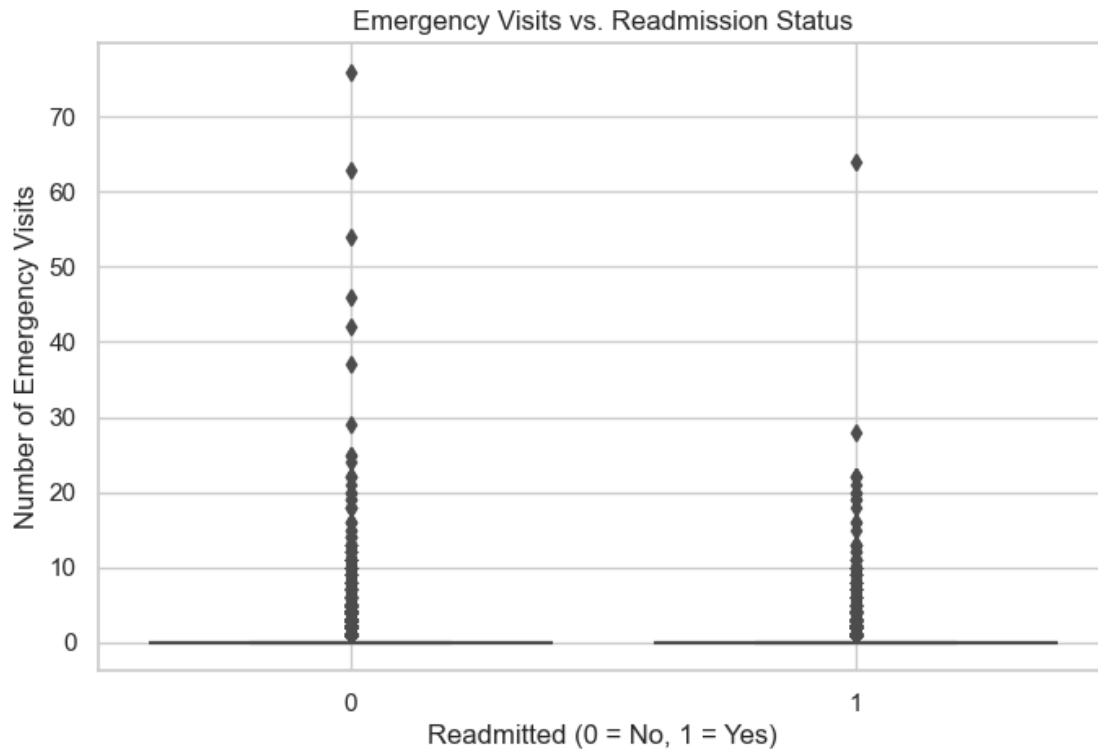


15. Impact of Emergency Visits on Subsequent Readmissions:

- Analyze how the number of emergency visits in the year prior to hospitalization affects 30-day readmission rates.

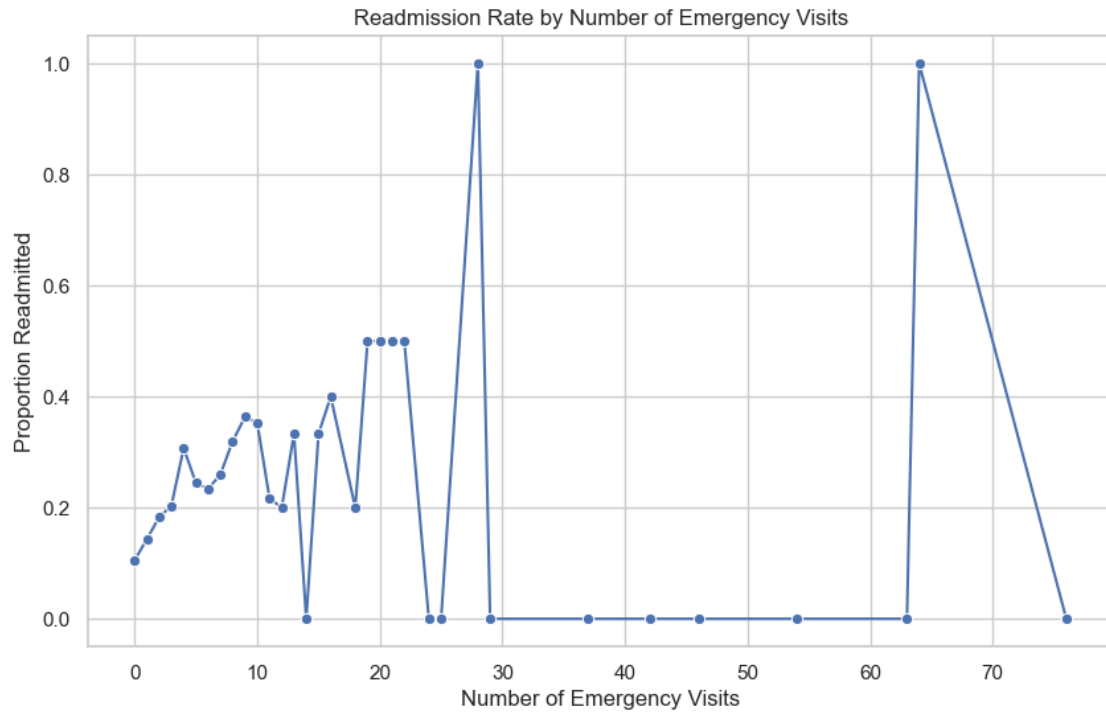
[42]: *# To evaluate how the number of emergency visits (number_emergency) prior to hospitalization correlates with the likelihood of being readmitted within 30 days.*

```
# Emergency Visits by Readmission Status
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='readmitted_binary', y='number_emergency')
plt.title("Emergency Visits vs. Readmission Status")
plt.xlabel("Readmitted (0 = No, 1 = Yes)")
plt.ylabel("Number of Emergency Visits")
plt.grid(True)
plt.show()
```



```
[43]: # Readmission Rate by Emergency Visit Count
emergency_readmit = df.groupby('number_emergency')['readmitted_binary'].mean()

plt.figure(figsize=(10, 6))
sns.lineplot(x=emergency_readmit.index, y=emergency_readmit.values, marker='o')
plt.title("Readmission Rate by Number of Emergency Visits")
plt.xlabel("Number of Emergency Visits")
plt.ylabel("Proportion Readmitted")
plt.grid(True)
plt.show()
```



16. Effect of Age and Gender on Treatment Outcomes:

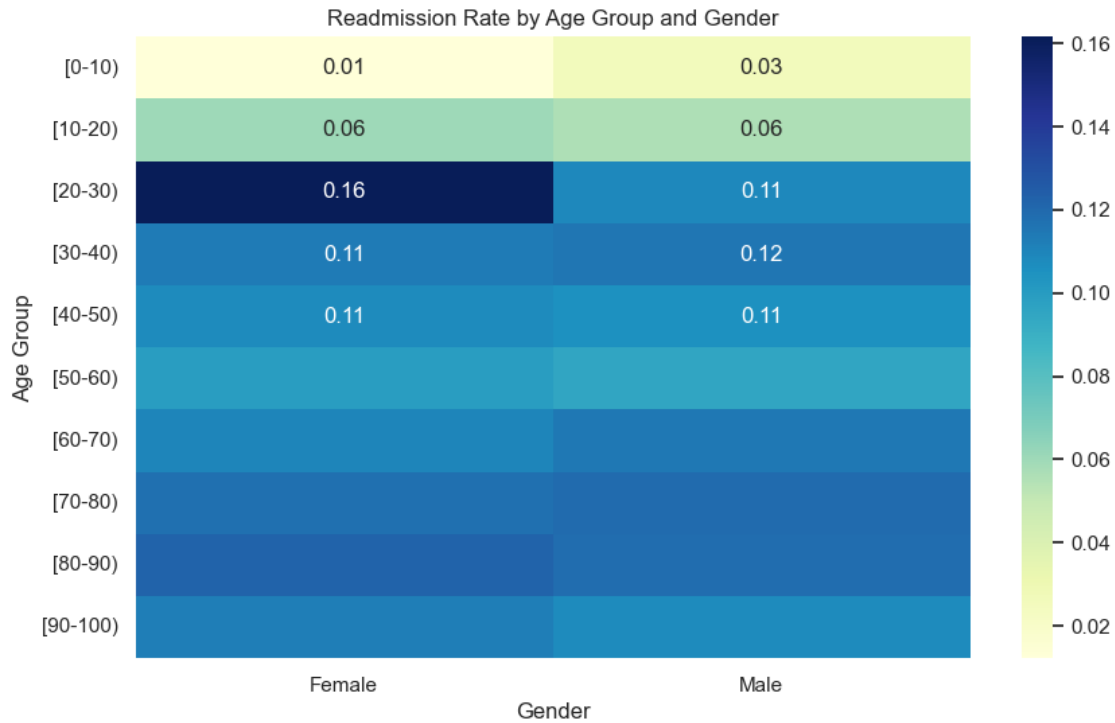
- Explore how age and gender influence treatment outcomes and readmission rates.

```
[44]: # Calculate readmission rates by age and gender
age_gender = df.groupby(['age', 'gender'])['readmitted_binary'].mean().
    ↪reset_index()

# Plot grouped bar plot
plt.figure(figsize=(12, 6))
sns.barplot(data=age_gender, x='age', y='readmitted_binary', hue='gender',
    ↪palette='Set2')
plt.title("Readmission Rate by Age Group and Gender")
plt.xlabel("Age Group")
plt.ylabel("Proportion Readmitted")
plt.xticks(rotation=45)
plt.legend(title="Gender")
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
[45]: age_gender_readmit = df.groupby(['age', 'gender'])['readmitted_binary'].mean().  
      ↪unstack()  
  
plt.figure(figsize=(10, 6))  
sns.heatmap(age_gender_readmit, annot=True, fmt=".2f", cmap="YlGnBu")  
plt.title("Readmission Rate by Age Group and Gender")  
plt.xlabel("Gender")  
plt.ylabel("Age Group")  
plt.show()
```



17. Insights and Recommendations:

- Summarize key findings from the analysis, highlighting significant predictors of hospital readmission.
- Provide actionable recommendations for healthcare providers to reduce readmission rates, such as targeted interventions for high-risk groups or adjustments in treatment protocols.
- Suggest areas for further research, like exploring the impact of social determinants on readmission or developing predictive models for readmission risk.

1.2 Insights and Recommendations

1.2.1 Key Findings

After analyzing the patient health records dataset, the following significant insights were discovered regarding 30-day hospital readmissions:

- **Demographics:**
 - Older age groups have a higher risk of readmission.
 - Gender differences were minor, but some female groups showed slightly elevated readmission rates.
 - Race-related disparities also existed, with some racial groups more prone to readmissions than others.
- **Clinical Factors:**

- Patients with a higher number of diagnoses, medications, and lab procedures were more likely to be readmitted.
- Prior emergency visits and inpatient stays significantly increased readmission risk.
- Changes in diabetes medication (especially insulin) during the hospital stay correlated with higher readmission rates.
- **Admission & Discharge Factors:**
 - Emergency and urgent admission types were more often associated with readmissions.
 - Patients discharged to rehabilitation or another healthcare facility had higher readmission rates.
 - Longer hospital stays, particularly with medication changes, were indicators of increased risk.
- **Laboratory Testing & Medical Specialty:**
 - Patients who underwent HbA1c testing during their stay were slightly less likely to be readmitted.
 - Certain medical specialties (e.g., endocrinology) were linked to lower readmission rates compared to others like general or internal medicine.

1.2.2 Recommendations

Based on the analysis, here are actionable strategies for reducing hospital readmissions:

- **Target High-Risk Patients:** Use age, diagnosis count, prior hospital usage, and medication change indicators to flag patients at risk.
- **Improve Discharge Planning:** Enhance post-discharge care plans and ensure follow-ups, especially for patients discharged to another care facility.
- **Focus on Chronic Disease Management:** Regularly perform HbA1c testing for diabetic patients and ensure follow-up after discharge.
- **Review Medication Changes Thoroughly:** Carefully manage and educate patients on medication adjustments made during their stay.
- **Coordinate Between Specialties:** Encourage better communication and shared responsibility between specialists for patients with complex conditions.

1.2.3 Suggestions for Future Research

- **Include Social Determinants:** Consider factors like income, education, family support, and access to care, which might significantly affect readmission.
- **Predictive Modeling:** Use machine learning to develop models that can proactively predict readmission risk at the time of discharge.
- **Patient Behavior & Adherence:** Study how well patients follow treatment plans and attend follow-up appointments to understand behavior-driven readmissions.