

Surgical Site Infections Among Colorectal Cancer Patients

Erica Washington, MPH, CPH, CIC, CPHQ, FAPIC

November 24, 2025

<https://github.com/ericajwashington/datavizreport>

Outline

1. Introduction
2. Related Work
3. Methods
4. Results
5. Discussion
6. Future Work

Introduction

Specific Aim: To understand the relationship between colorectal surgery and time-to-infection and time-to-death among CRC patients in Louisiana.

Table 1. Causal approach demonstrating exposure* and outcome amongst study participants.

| Colorectal Cancer Patients | | | |
|-------------------------------------|-----------------------------------|-----------------------------------|---|
| | Infection (+) $A=1$ $y = 1$ | Infection (-) $A=1$ $y = 0$ | |
| *Surgical Type (+) $A=0$ $y = 1$ | a | b | |
| Surgical Type (-) $A=0$ $y = 0$ | c | d | |
| | Death (+) $A=1$ $y = 1$ | Death (-) $A=1$ $y = 0$ | N |

*The primary exposure is the type of surgery performed. The secondary exposure is if a multi drug-resistant (MDR) pathogen was identified, lending to virulence of the infection. Additional stratification will examine sub-categories of MDR pathogens by extreme drug resistance.

*Surgical types include partial and total colectomies, resections, and excisions.

About Colorectal Cancer

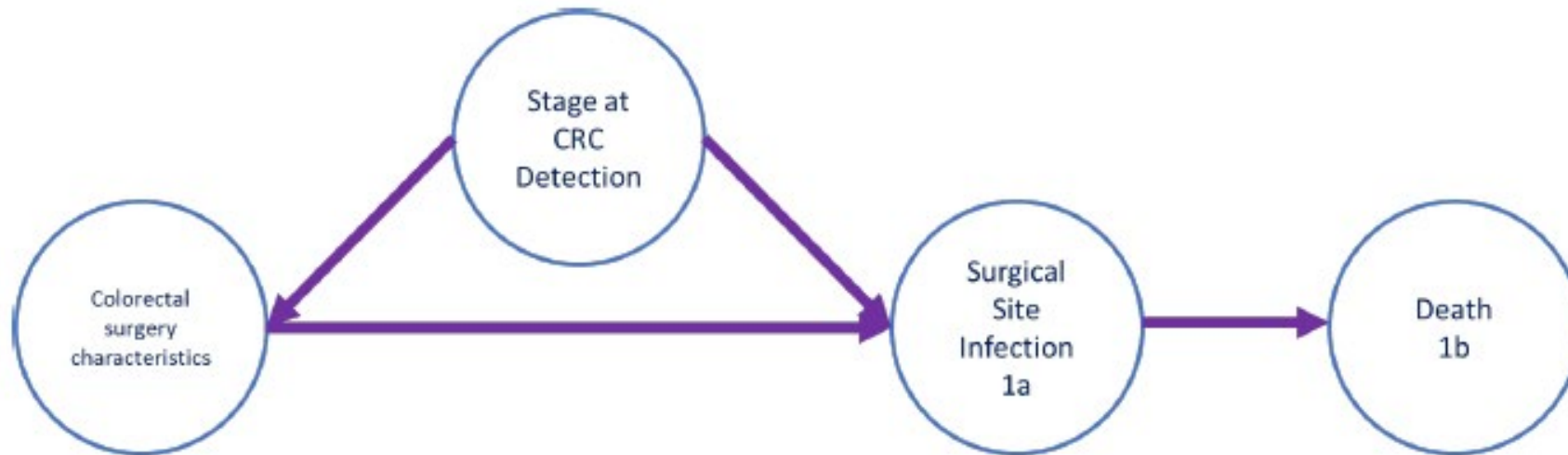
1. Colorectal cancer (CRC) begins as polyps in the colorectum.
2. A number of treatment options are available, including chemotherapy and immunotherapy.
3. Five-year survival for persons diagnosed with CRC is ~92% for stage 1.
 - 73% for stages 2-3 at diagnosis.
 - 13% for stage 4 at diagnosis.

Colorectal Cancer and Louisiana Populations

- Lynch Syndrome.
- Higher incidence for white and black men and black women in comparison to U.S. rates in Louisiana.
- CRC mortality accounted for 9.2% of all Louisiana cancer deaths from 2016-2020.
- Environmental factors and CRC.

Research Question

Does acquiring a surgical site infection affect morbidity and survival among CRC patients in Louisiana from 2015-2022?



Related Work

- Survival data through a retrospective cohort
- Cancers and the five-year survival
- Louisiana Tumor Registry

Li N, Wei S, Qi Y, Wei W. The effects of enhanced recovery after surgery on wound infection, complications, and postoperative hospital stay in patients undergoing colorectal surgery: A systematic review and meta-analysis. *Int Wound J*. 2023 Dec;20(10):3990-3998. doi:

Methods

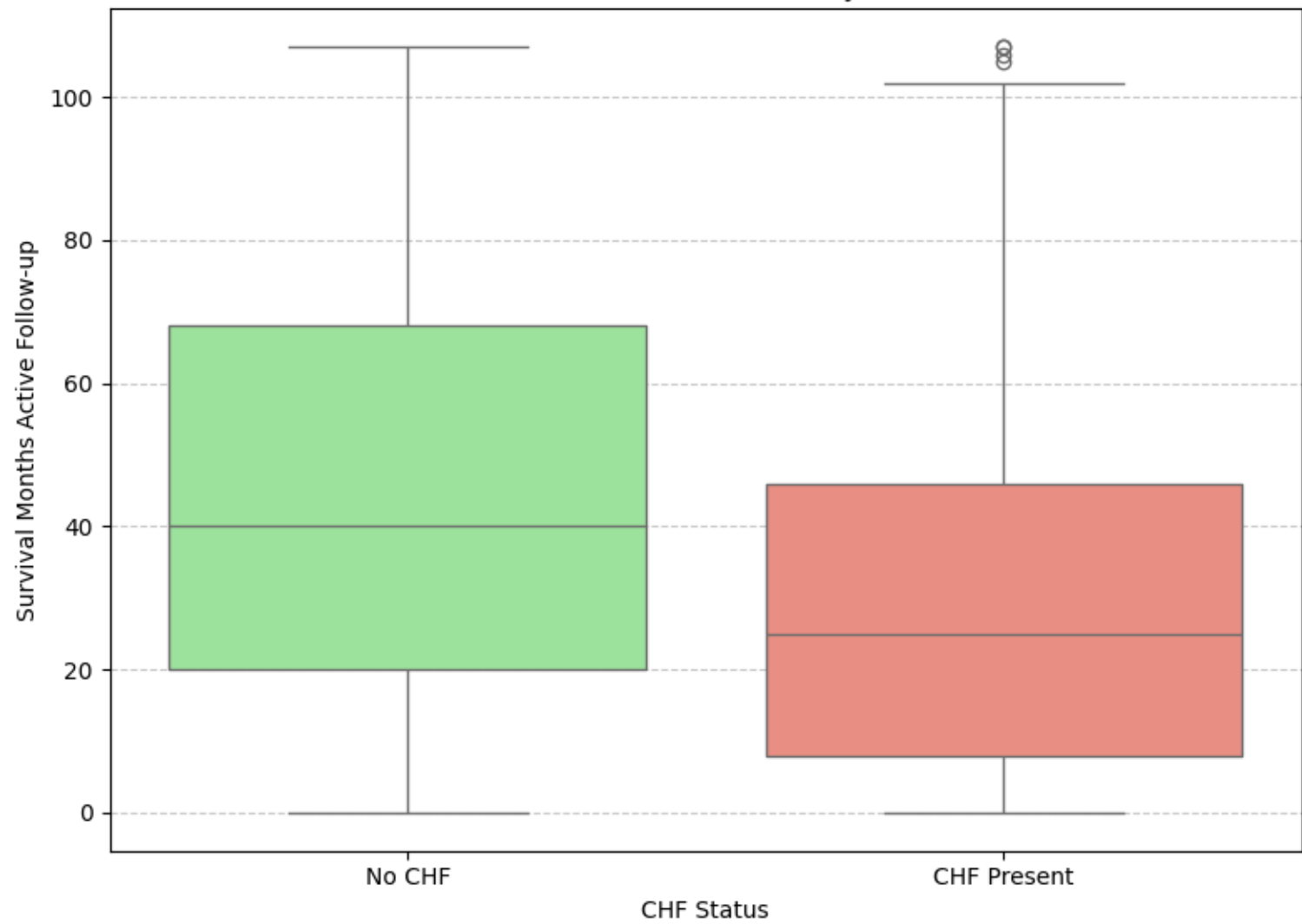
1. Python
2. (previously SAS as well)

Methods

- Exclusions
 - Anyone younger than 20 years of age at the time of diagnosis.
 - Those who had 0 days for length of stay.
 - Patients with unclassified urban-rural status.

Results

Distribution of Survival Months by CHF Status



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

# Map numerical 'chf' column to descriptive labels
chf_mapping = {0: 'No CHF', 1: 'CHF Present'}

df['chf_status'] =
df['chf'].map(chf_mapping).astype('category')

# Ensure the order of categories for consistent
plotting

df['chf_status'] = pd.Categorical(df['chf_status'],
categories=['No CHF', 'CHF Present'], ordered=True)

# Create the box plot

plt.figure(figsize=(8, 6))

sns.boxplot(
```

```
data=df,

x='chf_status',

y='surv_mos_active_fup',

hue='chf_status', # Assign x to hue to resolve
FutureWarning

palette={'No CHF': 'lightgreen', 'CHF Present':
'salmon'},

legend=False # Set legend to False as hue is x
variable

)

plt.title('Distribution of Survival Months by CHF
Status')

plt.xlabel('CHF Status')

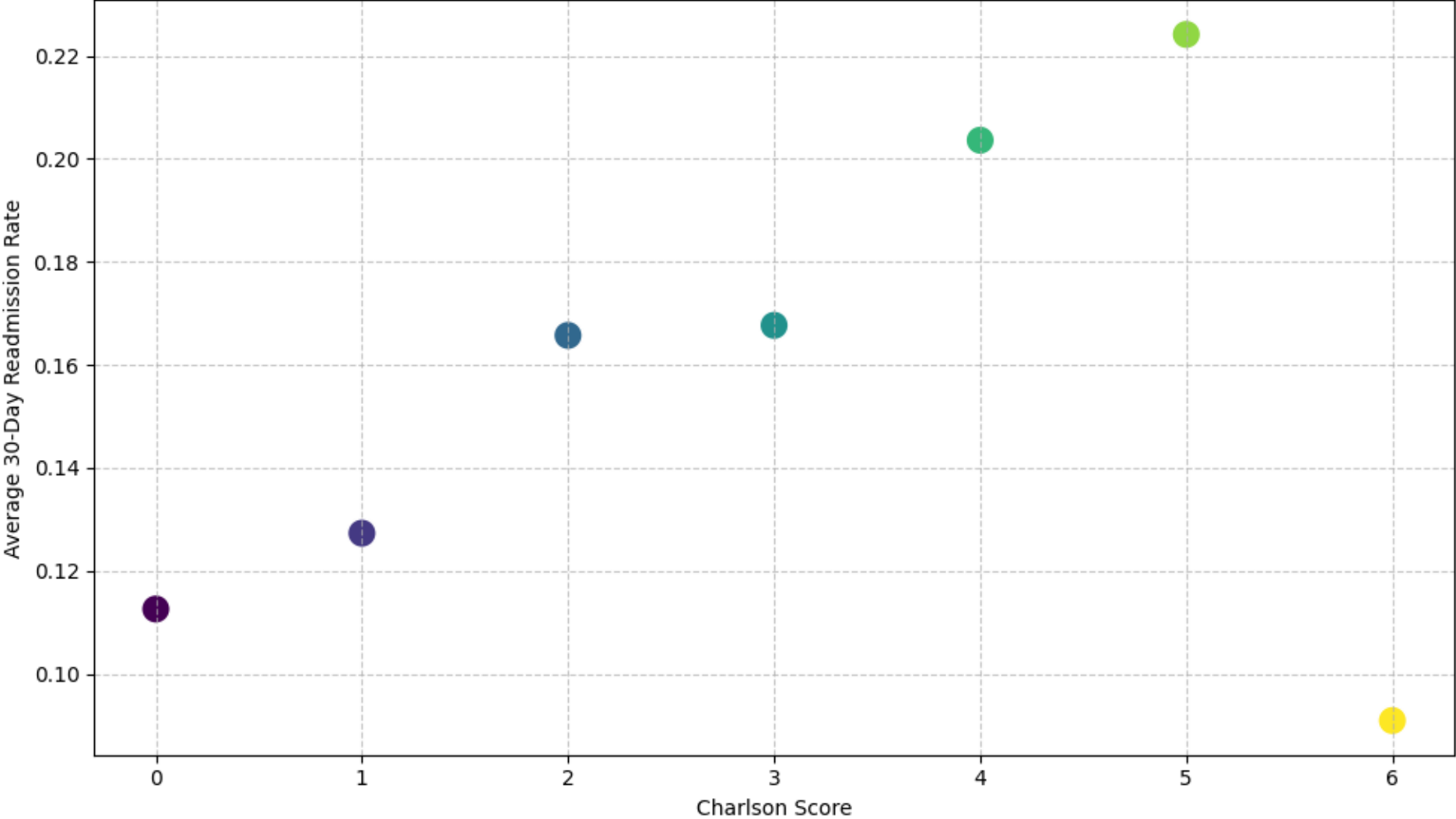
plt.ylabel('Survival Months Active Follow-up')

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()

plt.show()
```

Average 30-Day Readmission Rate by Charlson Score (Scatter Plot)



Final Project Washington Revised.ipynb

- Colab

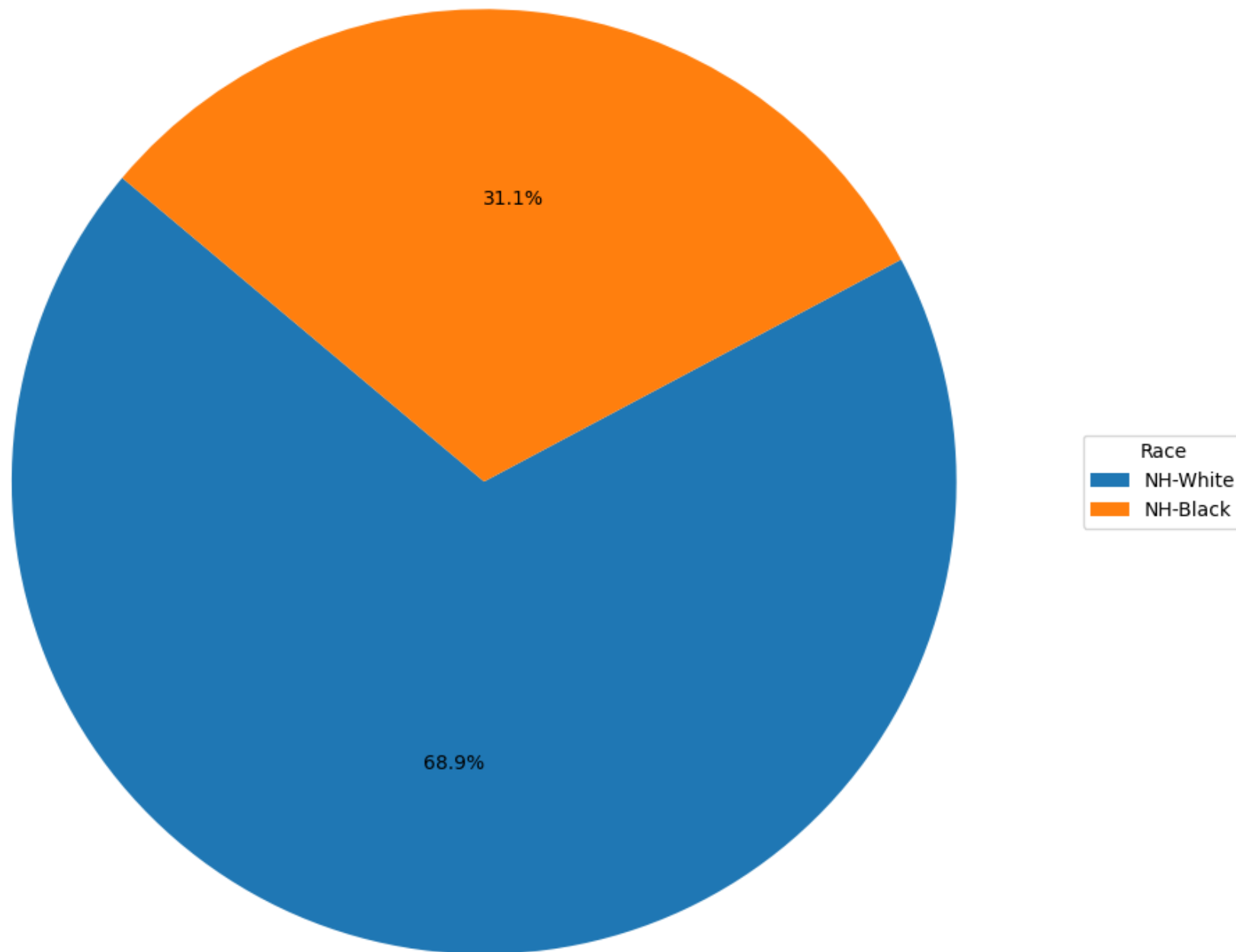
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=readmission_rate_by_charlson_overall,
    x='Charlson_score',
    y='readmission_30d',
    s=200, # Increased point size for better visibility
    hue='Charlson_score', # Color points by Charlson score
```

```
    palette='viridis', # Use a color palette
    legend=False # No need for a separate legend if hue is x
    variable
)

# Add title and labels
plt.title('Average 30-Day Readmission Rate by Charlson Score
(Scatter Plot)')
plt.xlabel('Charlson Score')
plt.ylabel('Average 30-Day Readmission Rate')
plt.xticks(rotation=0)
plt.grid(axis='both', linestyle='--', alpha=0.7) # Add grid for both x
and y axis
plt.tight_layout()
plt.show()
```

Distribution of Cases by Race



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt
```

```
# Count the occurrences of each race
```

```
race_counts = df['race'].value_counts()
```

```
# Plot the pie chart
```

```
plt.figure(figsize=(10, 8)) # Increased figure size to accommodate legend
```

```
plt.pie(race_counts, labels=None, autopct='%1.1f%%', startangle=140) # labels=None to prevent duplicate labels
```

```
plt.title('Distribution of Cases by Race')
```

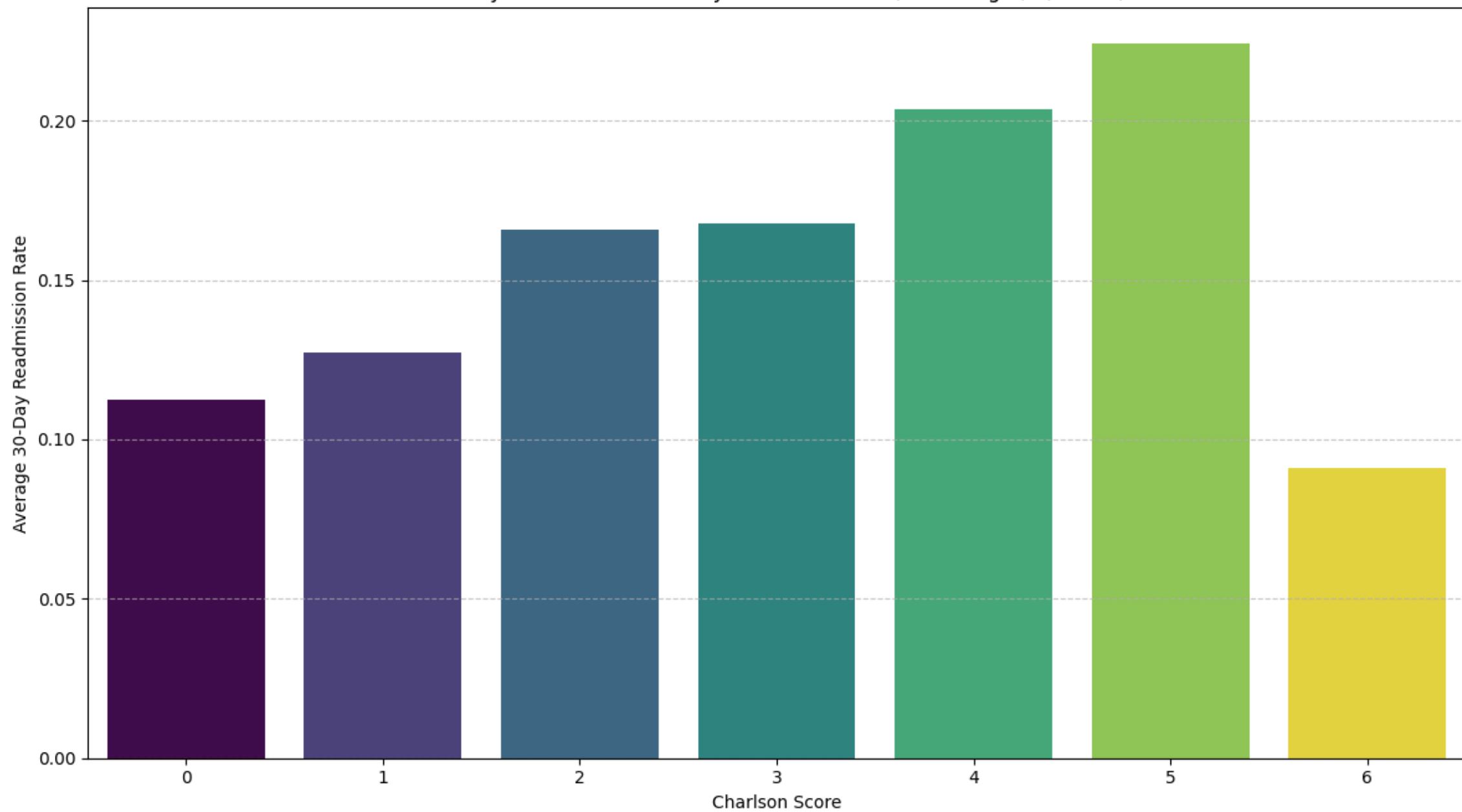
```
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
```

```
plt.legend(race_counts.index, title="Race", bbox_to_anchor=(1, 0.5), loc="center left") # Move legend to the right
```

```
plt.tight_layout()
```

```
plt.show()
```

30-Day Readmission Rate by Charlson Score (Excluding 7, 8, and 9)



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the mean readmission rate for each Charlson
score

readmission_rate_by_charlson =
df.groupby('Charlson_score')['readmission_30d'].mean().
reset_index()

# Remove categories 7, 8, and 9

readmission_rate_by_charlson =
readmission_rate_by_charlson[~readmission_rate_by_ch
arlson['Charlson_score'].isin([7, 8, 9])]

# Plot the bar chart

plt.figure(figsize=(12, 7))
```

```
sns.barplot(data=readmission_rate_by_charlson,
x='Charlson_score', y='readmission_30d',
hue='Charlson_score', palette='viridis', legend=False)

plt.title('30-Day Readmission Rate by Charlson Score
(Excluding 7, 8, and 9)')

plt.xlabel('Charlson Score')

plt.ylabel('Average 30-Day Readmission Rate')

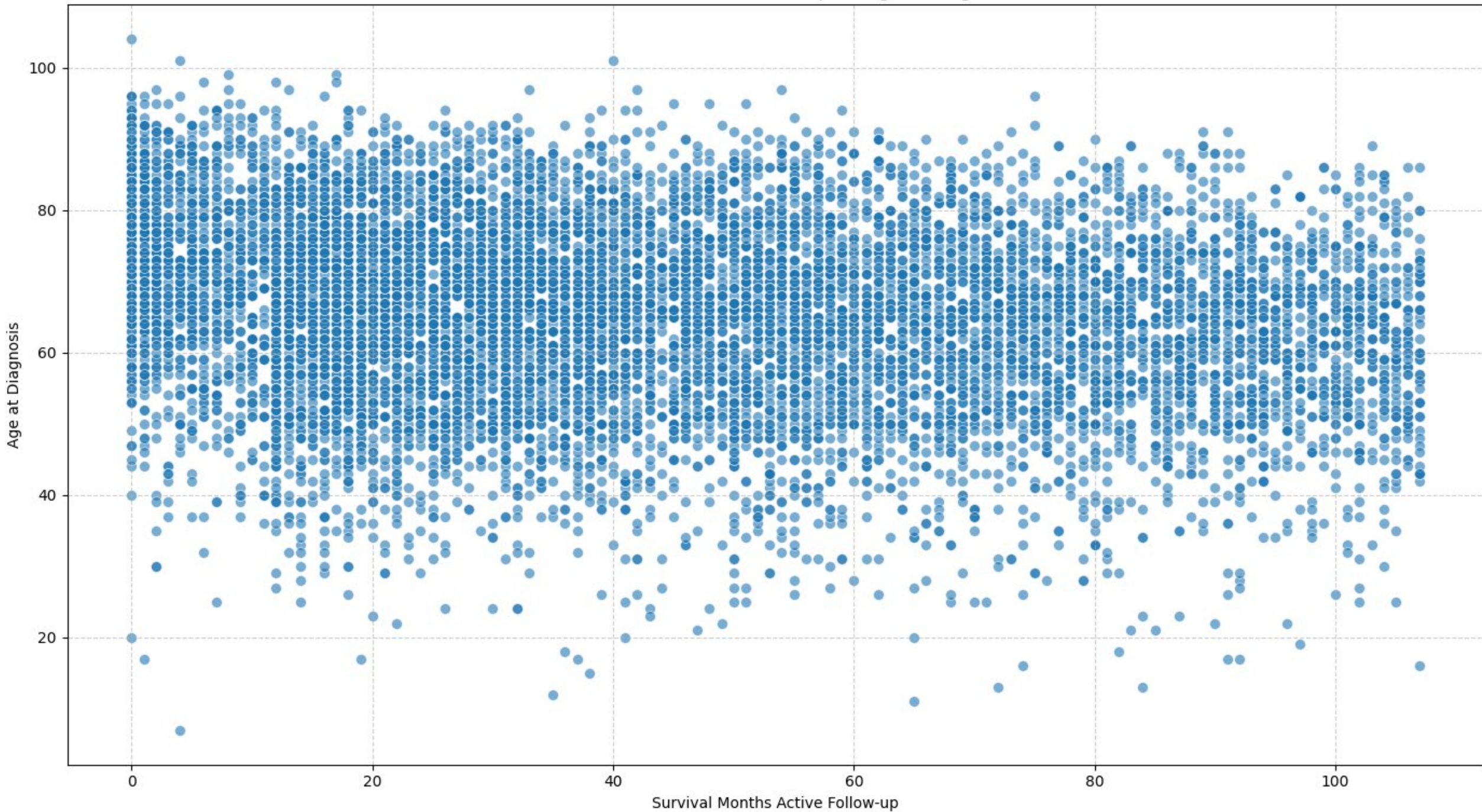
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.xticks(rotation=0)

plt.tight_layout()

plt.show()
```

Survival Months Active Follow-up vs. Age at Diagnosis



Final Project Washington Revised.ipynb

- Colab

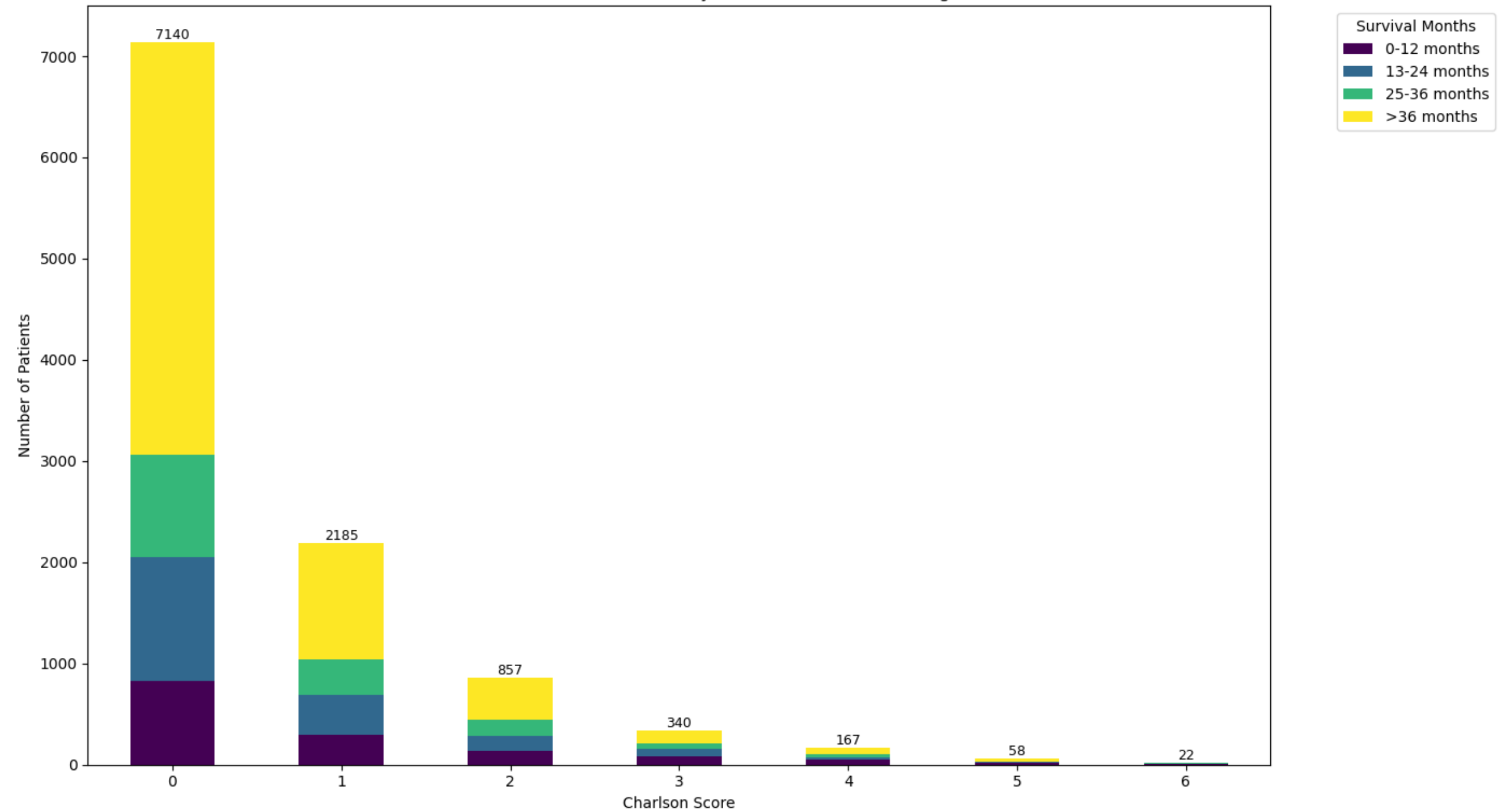
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create the scatter plot
plt.figure(figsize=(14, 8))
sns.scatterplot(
    data=df,
    x='surv_mos_active_fup', # Swapped x-axis
    y='age_at_diagnosis',   # Swapped y-axis
    s=50, # Set a fixed size for the points
```

```
    alpha=0.6 # Set transparency for better
visibility of overlapping points
)

plt.title('Survival Months Active Follow-up vs.
Age at Diagnosis')
plt.xlabel('Survival Months Active Follow-up')
# Updated x-axis label
plt.ylabel('Age at Diagnosis')          #
Updated y-axis label
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

Distribution of Binned Survival Months by Charlson Score (Excluding 7, 8, 9)



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt
import seaborn as sns

# Filter out Charlson scores 7, 8, and 9
filtered_stacked_bar_data = stacked_bar_data.drop(index=[7, 8, 9], errors='ignore')

# Plotting the stacked bar chart
ax = filtered_stacked_bar_data.plot(kind='bar', stacked=True, figsize=(14, 8),
cmap='viridis')

# Add total numbers above each bar
totals = filtered_stacked_bar_data.sum(axis=1) # Sum of all categories for each
Charlson score

# Iterate over the list of Charlson scores (x-axis labels)
for i, charlson_score_label in enumerate(filtered_stacked_bar_data.index):

    # Get the total height for this Charlson score
    total_count = totals.loc[charlson_score_label]
```

```
    # Find the position to place the text. We can use the x-coordinate of the bottom
    patch
```

```
    # of the first container, as all patches for a given x-label share the same x-
    coordinate.
```

```
    x_pos = ax.containers[0].patches[i].get_x() +
    ax.containers[0].patches[i].get_width() / 2
```

```
    y_pos = total_count # Place the text at the total height
```

```
    ax.text(x_pos, y_pos, str(int(total_count)), ha='center', va='bottom', fontsize=9)
```

```
plt.title('Distribution of Binned Survival Months by Charlson Score (Excluding 7, 8, 9)')
```

```
plt.xlabel('Charlson Score')
```

```
plt.ylabel('Number of Patients')
```

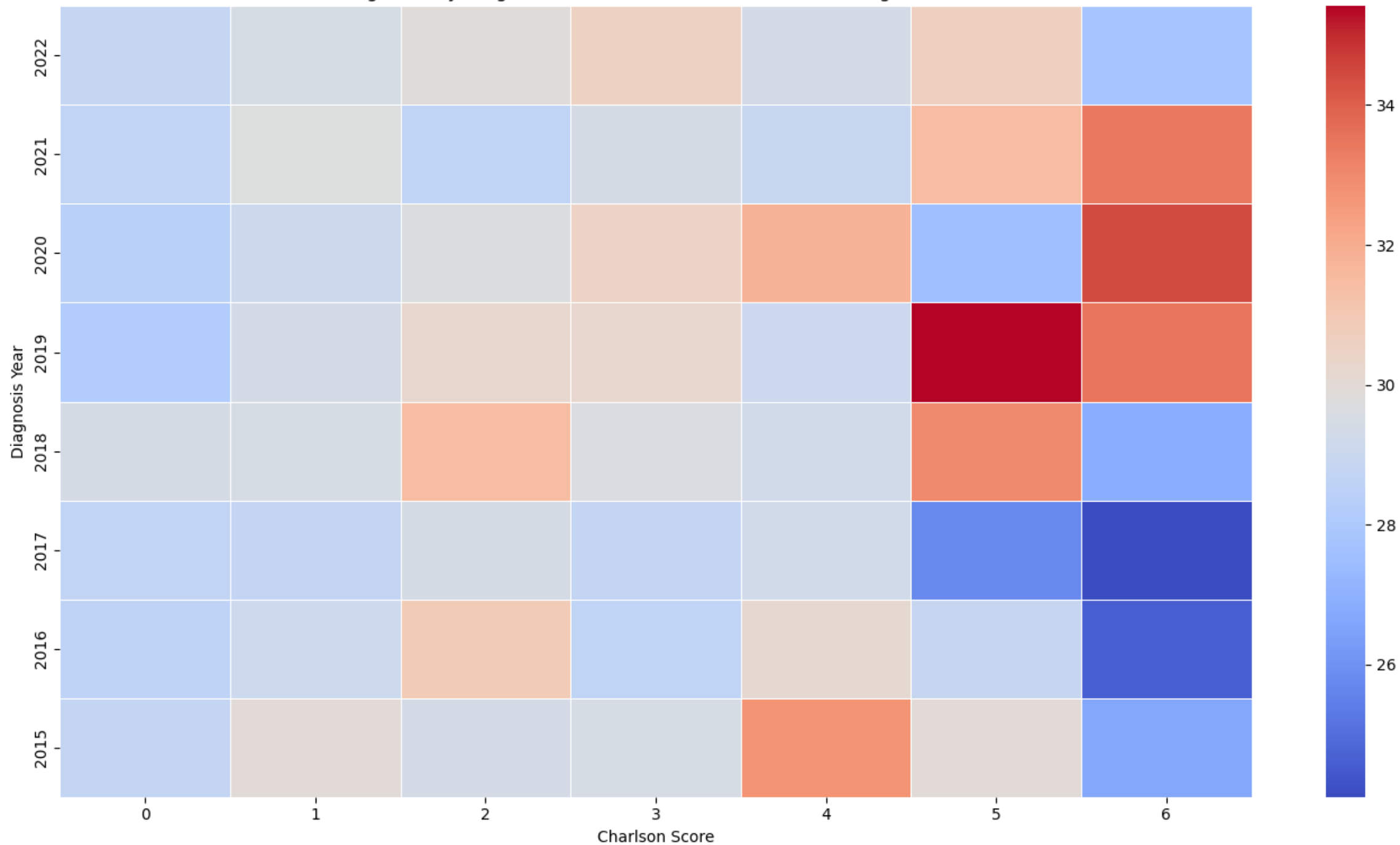
```
plt.xticks(rotation=0) # Keep x-axis labels horizontal for readability
```

```
plt.legend(title='Survival Months', bbox_to_anchor=(1.05, 1), loc='upper left')
```

```
plt.tight_layout()
```

```
plt.show()
```

Average BMI by Diagnosis Year and Charlson Score (Excluding 7, 8 and 9)



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# Create a pivot table: rows are years, columns are  
Charlson Scores, values are mean of BMI
```

```
bmi_heatmap_data = df.pivot_table(values='BMI',  
index='dx_yr', columns='Charlson_score',  
aggfunc='mean')
```

```
# Remove columns for Charlson scores 7, 8 and 9
```

```
bmi_heatmap_data =  
bmi_heatmap_data.drop(columns=[7, 8, 9],  
errors='ignore')
```

```
# Reverse the order of years in the index for plotting
```

```
bmi_heatmap_data = bmi_heatmap_data.iloc[::-1]
```

```
# Plot the heatmap
```

```
plt.figure(figsize=(14, 8))
```

```
sns.heatmap(bmi_heatmap_data, annot=False,  
cmap='coolwarm', fmt=".1f", linewidths=.5)
```

```
plt.title('Average BMI by Diagnosis Year and Charlson  
Score (Excluding 7, 8 and 9)')
```

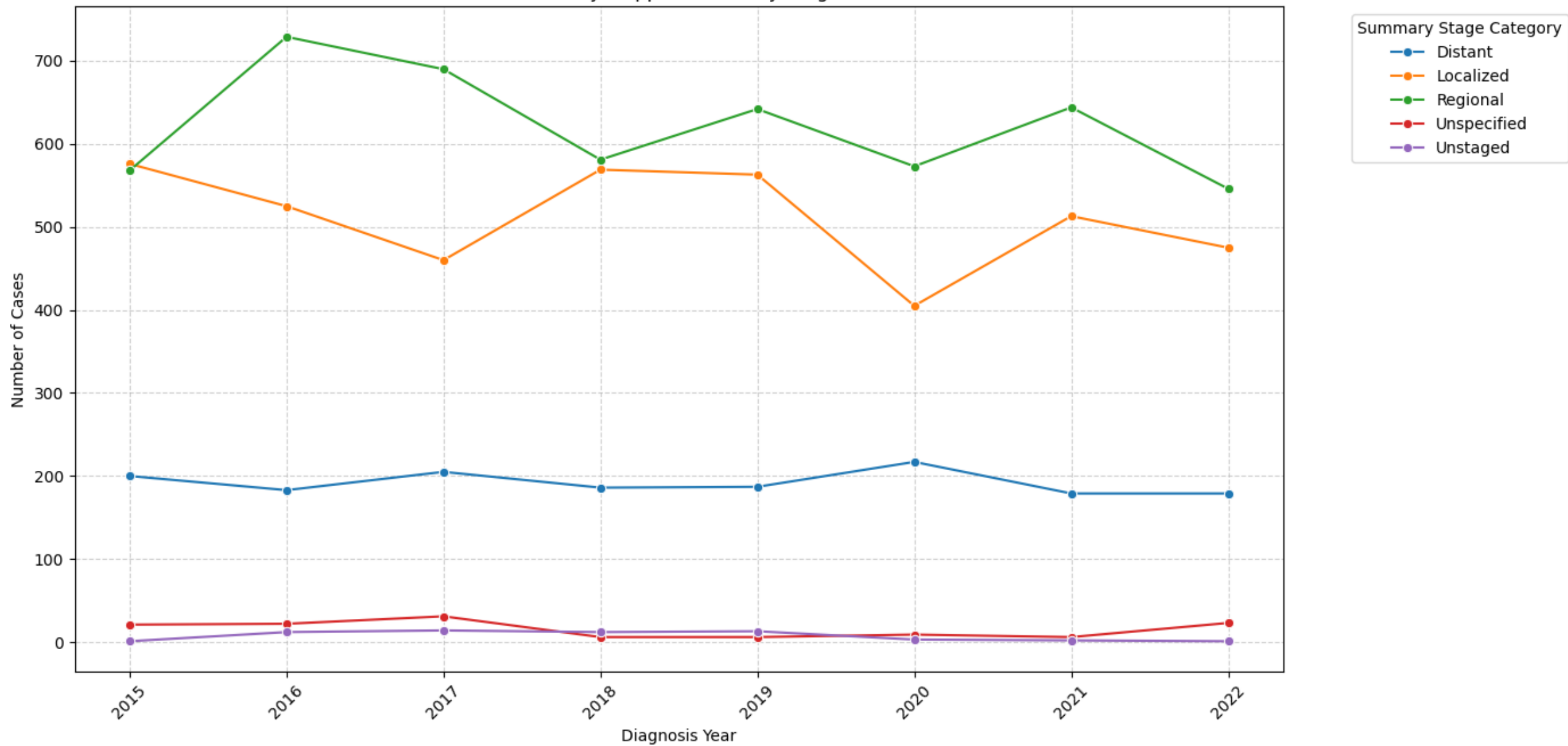
```
plt.xlabel('Charlson Score')
```

```
plt.ylabel('Diagnosis Year')
```

```
plt.tight_layout()
```

```
plt.show()
```

Number of Cases by Mapped Summary Stage Over Years



Final Project Washington Revised.ipynb

- Colab

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# Create a new column with mapped summary stages
```

```
def map_summary_stage(stage):
```

```
    if stage == 1:
```

```
        return 'Localized'
```

```
    elif stage in [2, 3, 4, 5]:
```

```
        return 'Regional'
```

```
    elif stage == 7:
```

```
        return 'Distant'
```

```
    elif stage == 9:
```

```
        return 'Unstaged'
```

```
    elif stage == 0: # Assuming stage 0 is an unspecified category
```

```
        return 'Unspecified'
```

```
    else:
```

```
        return 'Other' # Catch any other unexpected stages
```

```
df['mapped_summary_stage'] = df['summary_stage'].apply(map_summary_stage)
```

```
df['mapped_summary_stage'] = df['mapped_summary_stage'].astype('category')
```

```
# Group by diagnosis year and mapped summary stage, then count the number of cases
```

```
cases_by_year_stage = df.groupby(['dx_yr', 'mapped_summary_stage'], observed=False).size().reset_index(name='count')
```

```
# Plot the line graph
```

```
plt.figure(figsize=(14, 7))
```

```
sns.lineplot(data=cases_by_year_stage, x='dx_yr', y='count', hue='mapped_summary_stage', marker='o')
```

```
plt.title('Number of Cases by Mapped Summary Stage Over Years')
```

```
plt.xlabel('Diagnosis Year')
```

```
plt.ylabel('Number of Cases')
```

```
plt.xticks(rotation=45)
```

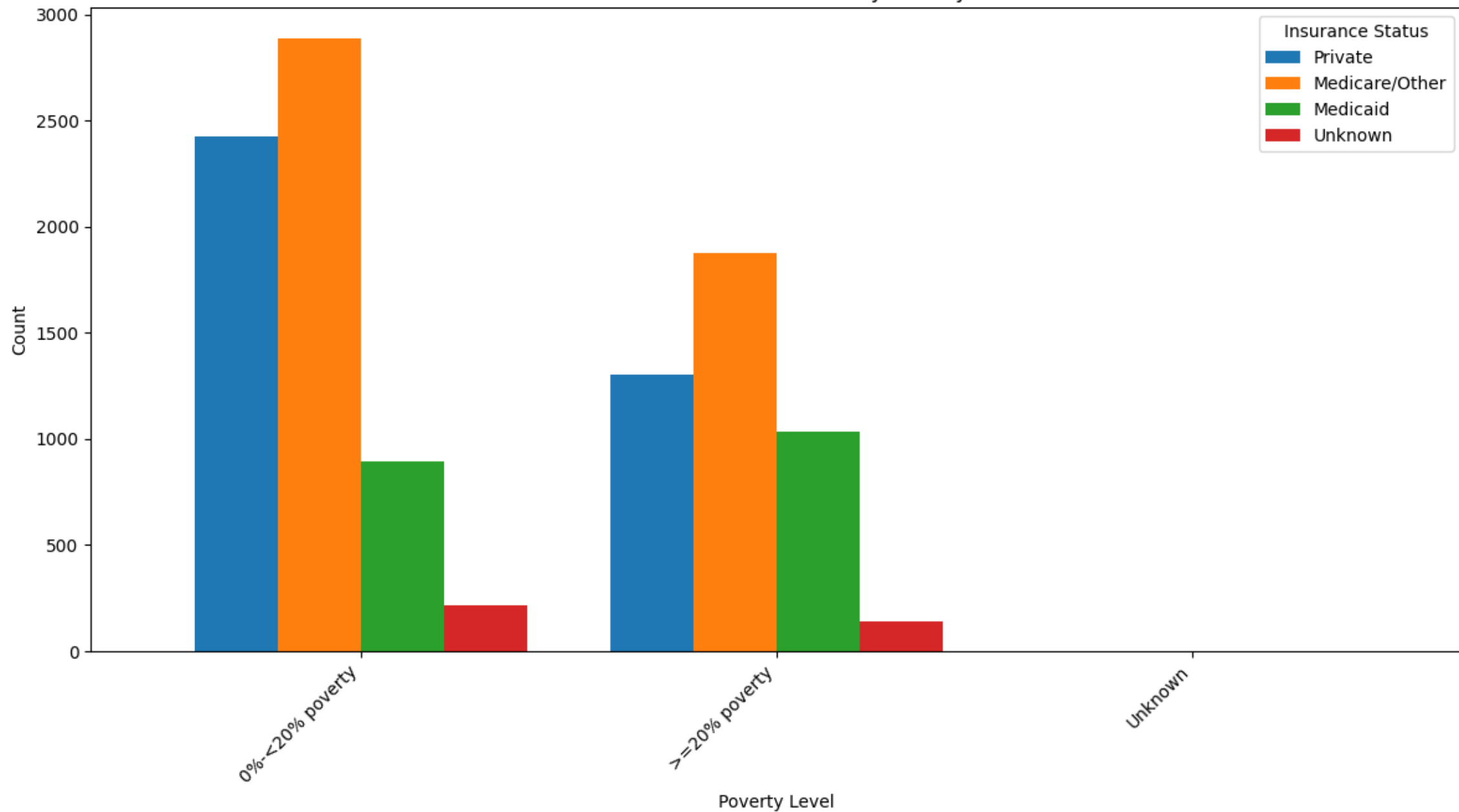
```
plt.legend(title='Summary Stage Category', bbox_to_anchor=(1.05, 1), loc='upper left')
```

```
plt.grid(True, linestyle='--', alpha=0.6)
```

```
plt.tight_layout()
```

```
plt.show()
```

Distribution of Insurance Status by Poverty Level



Final Project Washington Revised.ipynb

- Colab

#Is there an association between poverty and insurance status?

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# Ensure 'poverty' is treated as categorical
```

```
df['poverty'] = df['poverty'].astype('category')
```

```
# Filter out rows where insurance is '9' (Unknown) before mapping
```

```
df_filtered = df[df['insurance'] != 9].copy()
```

```
# Filter out rows where poverty is 'Unknown'
```

```
df_filtered = df_filtered[df_filtered['poverty'] != 'Unknown'].copy()
```

```
# Map numerical insurance categories to descriptive names, excluding 'Unknown'
```

```
insurance_mapping = {1: 'Private', 2: 'Medicare/Other', 3: 'Medicaid'}
```

```
df_filtered['insurance'] =  
df_filtered['insurance'].astype('category').cat.rename_categories(insurance_mapping)
```

```
# Create a cross-tabulation of 'poverty' and 'insurance'
```

```
pivot_df = df_filtered.groupby(['poverty', 'insurance'],  
observed=False).size().unstack(fill_value=0)
```

```
# Plotting the grouped bar chart
```

```
ax = pivot_df.plot(kind='bar', figsize=(12, 7), width=0.8)
```

```
plt.title('Distribution of Insurance Status by Poverty Level (Excluding Unknown)')
```

```
plt.xlabel('Poverty Level')
```

```
plt.ylabel('Count')
```

```
plt.xticks(rotation=45, ha='right')
```

```
plt.legend(title='Insurance Status')
```

```
plt.tight_layout()
```

```
plt.show()
```

Discussion

1. Covariate analysis and initial data steps
2. Preliminary questions can help me review incidence density among specific patient types as I make my way through the Cox regression analysis for the paper.

Future Work

1. Work in progress.
2. Reviewing journals for publication with my dissertation committee.
3. Cox Regression, incidence density, sensitivity analysis for missing data and counterfactual outcomes.

Thank you!