

```

1 # 1. Install Necessary Libraries
2 # Connecting to BigQuery from Google Colab and pulling data is a straightforward process, primarily involving authentication
3 # First, we'll need the google-cloud-bigquery library to interact with BigQuery and pandas to work with the data in a DataFrame
4 # !pip install google-cloud-bigquery pandas
5
6 # 2. Authenticate Your Colab Environment
7 # Google Colab provides a simple way to authenticate using your Google account. This is the most common method for interacting with Google Cloud services.
8
9 from google.colab import auth
10 auth.authenticate_user()
11 print('Authenticated')

```

```

1 # All Imports
2 from google.cloud import bigquery
3 from google.colab import auth
4 from google.colab import drive
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8
9 import spacy
10 import re
11 import string
12 from collections import Counter
13
14 import numpy as np
15
16 from spacy.pipeline import EntityRuler
17 from spacy.tokens import DocBin
18

```

```

1 # Connect Google Cloud Project ID
2 # We need to use the BigQuery client which Google Cloud Project your data resides in.
3 project_id = 'quantum-vigil-417213' # Replace with your actual Google Cloud Project ID

```

```

1 # Authenticate Your Colab Environment
2 # Google Colab provides a simple way to authenticate using your Google account. This is the most common method for interacting with Google Cloud services.
3 # following block is commented here because we're consolidating all imports in one place which will be shown later.
4 # from google.colab import auth
5
6 auth.authenticate_user()
7 print('Authenticated')

```

↗ Authenticated

```

1 # Initialize the BigQuery Client
2 # Create an instance of the BigQuery client. This object will manage your connection and allow you to send queries.
3 # following block is commented here because we're consolidating all imports in one place which will be shown later.
4 # from google.cloud import bigquery
5
6 client = bigquery.Client(project=project_id)
7 print('BigQuery client initialized.')

```

↗ BigQuery client initialized.

```

1 # Write Your SQL Query
2 # Craft your SQL query. Remember to specify the full path for your tables using the format resources.dataset_name.table_
3 # This Counts of patients who died within 7 days of hospital admission by marital status and age at death
4 query = """
5 SELECT
6     a.marital_status,
7     COUNT(*) AS death_count,
8     ROUND(AVG(DATE_DIFF(p.dod, p.dob, YEAR)), 1) AS avg_age_at_death
9 FROM
10     physionet-data.mimiciii_demo.patients AS p
11 JOIN
12     physionet-data.mimiciii_demo.admissions a
13     ON p.subject_id = a.subject_id
14 WHERE
15     p.dod IS NOT NULL
16     AND a.admittime IS NOT NULL
17     --AND p.dod <= DATE_ADD(a.admittime, INTERVAL 7 DAY)
18 GROUP BY
19     a.marital_status
20 ORDER BY
21     avg_age_at_death DESC;
22 """
23 print('SQL query defined.')

```

SQL query defined.

```

1 # Execute the SQL to retrieve data set
2 # import pandas as pd
3 def run_bq_query(query):
4     try:
5         df_bigquery_data = client.query(query).to_dataframe()
6         print('\nQuery executed successfully! Data loaded into pandas DataFrame.')
7         print('First 5 rows of the data:')
8         print(df_bigquery_data.head())
9         return df_bigquery_data
10    except Exception as e:
11        print(f"\nAn error occurred: {e}")
12        print("Please double-check:")
13        print(" - Your `project_id` is correct.")
14        print(" - The table names in your SQL query are fully qualified")
15        print(" - Your Google account has the necessary BigQuery permissions")
16        print(" - The BigQuery API is enabled for your Google Cloud Project.")

```

```

1 # Execute the cleaned SQL to retrieve data set
2 # import pandas as pd
3
4 df_bigquery_data = run_bq_query(query)

```

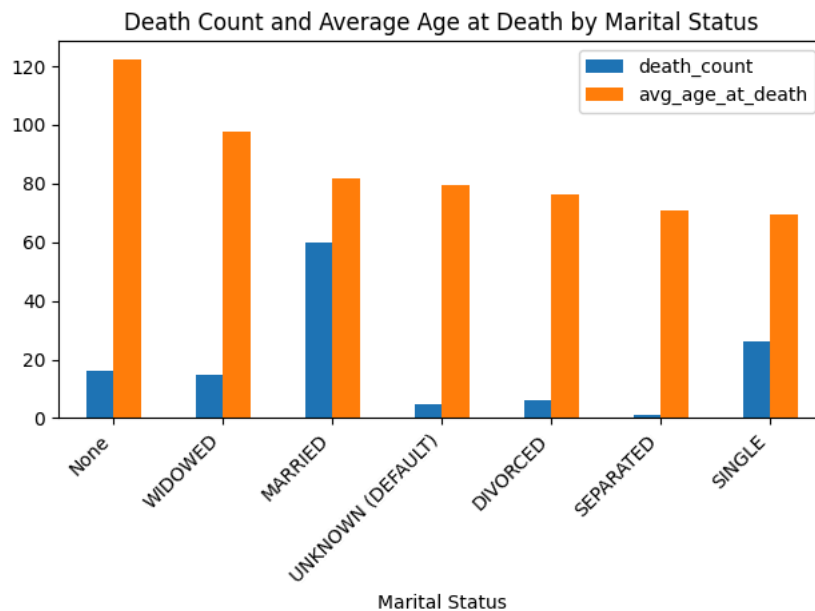
Query executed successfully! Data loaded into pandas DataFrame.
First 5 rows of the data:

	marital_status	death_count	avg_age_at_death
0	None	16	122.4
1	WIDOWED	15	97.7
2	MARRIED	60	81.9
3	UNKNOWN (DEFAULT)	5	79.6
4	DIVORCED	6	76.5

```

1 # Death Count by Marital Status (Bar Plot)
2 # import matplotlib.pyplot as plt
3
4 ax = df_bigquery_data.plot.bar(x='marital_status', y=['death_count', 'avg_age_at_death'])
5 plt.title('Death Count and Average Age at Death by Marital Status')
6 plt.xlabel('Marital Status')
7 plt.xticks(rotation=45, ha='right')
8 plt.tight_layout()
9 plt.show()
10

```



```

1 # SQL query for your analysis, incorporating data cleaning steps
2 query_cleaned = """
3 SELECT
4     CASE
5         WHEN a.marital_status IS NULL
6             OR a.marital_status IN ('UNKNOWN (DEFAULT)', 'None', '')
7             THEN 'Unknown'
8         ELSE a.marital_status
9     END AS marital_status_cleaned,
10    COUNT(*) AS death_count,
11    ROUND(AVG(DATE_DIFF(p.dod, p.dob, YEAR)), 1) AS avg_age_at_death
12 FROM
13     physionet-data.mimiciii_demo.patients AS p
14 JOIN
15     physionet-data.mimiciii_demo.admissions AS a
16 ON p.subject_id = a.subject_id
17 WHERE
18     p.dod IS NOT NULL
19     AND a.admittime IS NOT NULL
20     AND DATE_DIFF(p.dod, p.dob, YEAR) BETWEEN 0 AND 110
21     -- Uncomment the next line to restrict to deaths within 7 days of admission to check earli mortality after admission
22     -- AND p.dod <= DATE_ADD(a.admittime, INTERVAL 7 DAY)
23 GROUP BY
24     marital_status_cleaned
25 ORDER BY
26     avg_age_at_death DESC;
27 """
28 print('SQL query defined.')
29

```



SQL query defined.

```

1 # Execute the cleaned SQL to retrieve data set
2 # import pandas as pd
3
4 marital_counts_cleaned = run_bq_query(query_cleaned)

```



Query executed successfully! Data loaded into pandas DataFrame.

First 5 rows of the data:

	marital_status_cleaned	death_count	avg_age_at_death
0	WIDOWED	14	83.2
1	DIVORCED	6	76.5
2	SEPARATED	1	71.0
3	MARRIED	57	70.4
4	Unknown	17	68.0

```

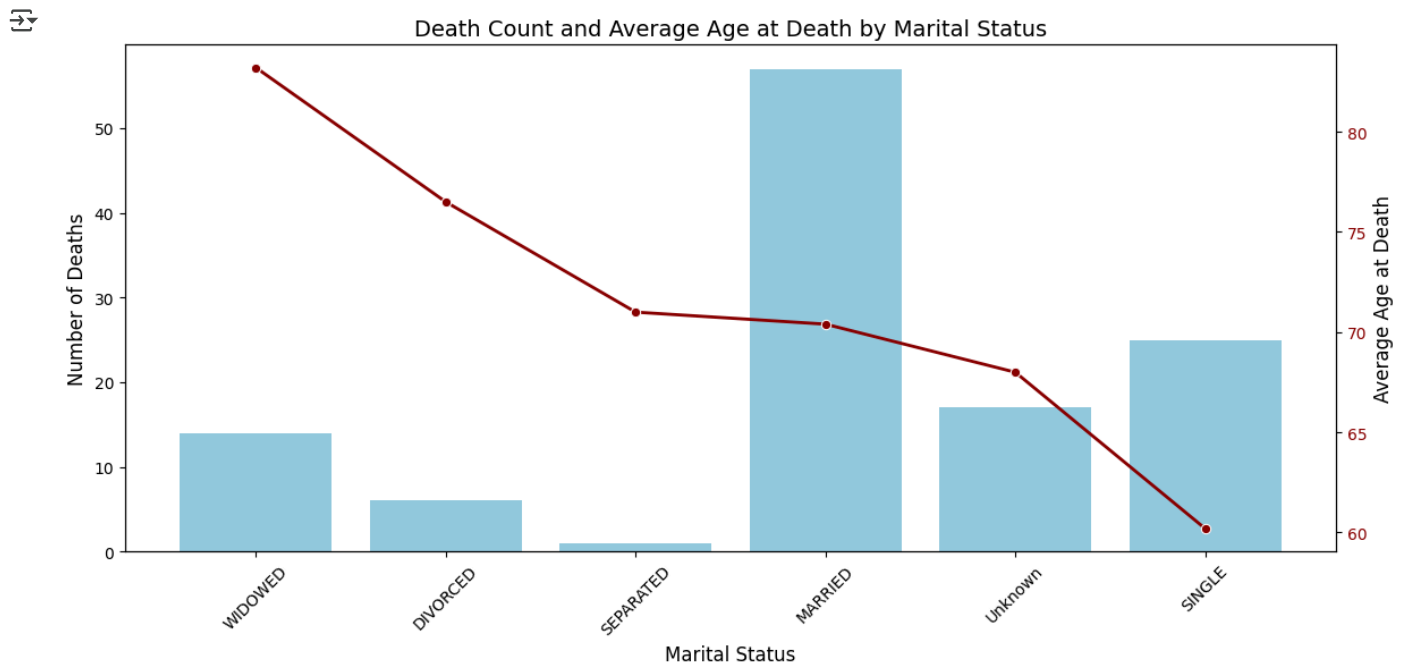
1 # Combined Bar + Line Plot: Death Count and Average Age at Death
2 fig, ax1 = plt.subplots(figsize=(12, 6))
3
4 # Bar plot for death count
5 sns.barplot(
6     x='marital_status_cleaned',
7     y='death_count',
8     data=marital_counts_cleaned,

```

```

9     ax=ax1,
10     color='skyblue'
11 )
12 ax1.set_ylabel('Number of Deaths', fontsize=12)
13 ax1.set_xlabel('Marital Status', fontsize=12)
14 ax1.set_title('Death Count and Average Age at Death by Marital Status', fontsize=14)
15 ax1.tick_params(axis='y')
16 plt.setp(ax1.get_xticklabels(), rotation=45)
17
18 # Line plot for average age at death on secondary y-axis
19 ax2 = ax1.twinx()
20 sns.lineplot(
21     x='marital_status_cleaned',
22     y='avg_age_at_death',
23     data=marital_counts_cleaned,
24     ax=ax2,
25     color='darkred',
26     marker='o',
27     linewidth=2
28 )
29 ax2.set_ylabel('Average Age at Death', fontsize=12)
30 ax2.tick_params(axis='y', labelcolor='darkred')
31
32 # Show plot
33 plt.tight_layout()
34 plt.show()
35

```



```

1 # Restrict to deaths within 7 days of admission to check early mortality after admission
2
3 query_early_mortality = """
4 SELECT
5     CASE
6         WHEN a.marital_status IS NULL
7             OR a.marital_status IN ('UNKNOWN (DEFAULT)', 'None', '')
8             THEN 'Unknown'
9         ELSE a.marital_status
10    END AS marital_status_cleaned,
11    COUNT(*) AS death_count,
12    ROUND(AVG(AGE(p.dod, p.dob, YEAR)), 1) AS avg_age_at_death
13 FROM
14     physionet-data.mimiciii_demo.patients AS p
15 JOIN
16     physionet-data.mimiciii_demo.admissions AS a
17 ON p.subject_id = a.subject_id
18 WHERE
19     p.dod IS NOT NULL
20     AND a.admittime IS NOT NULL
21     AND AGE(p.dod, p.dob, YEAR) BETWEEN 0 AND 110
22     AND p.dod <= DATE_ADD(a.admittime, INTERVAL 7 DAY)

```

```

23 GROUP BY
24     marital_status_cleaned
25 ORDER BY
26     avg_age_at_death DESC;
27 """
28 print('SQL query defined.')

```

SQL query defined.

```

1 # Execute the cleaned SQL to retrieve data set
2 # import pandas as pd
3
4 early_mortality_df = run_bq_query(query_early_mortality)

```

Query executed successfully! Data loaded into pandas DataFrame.

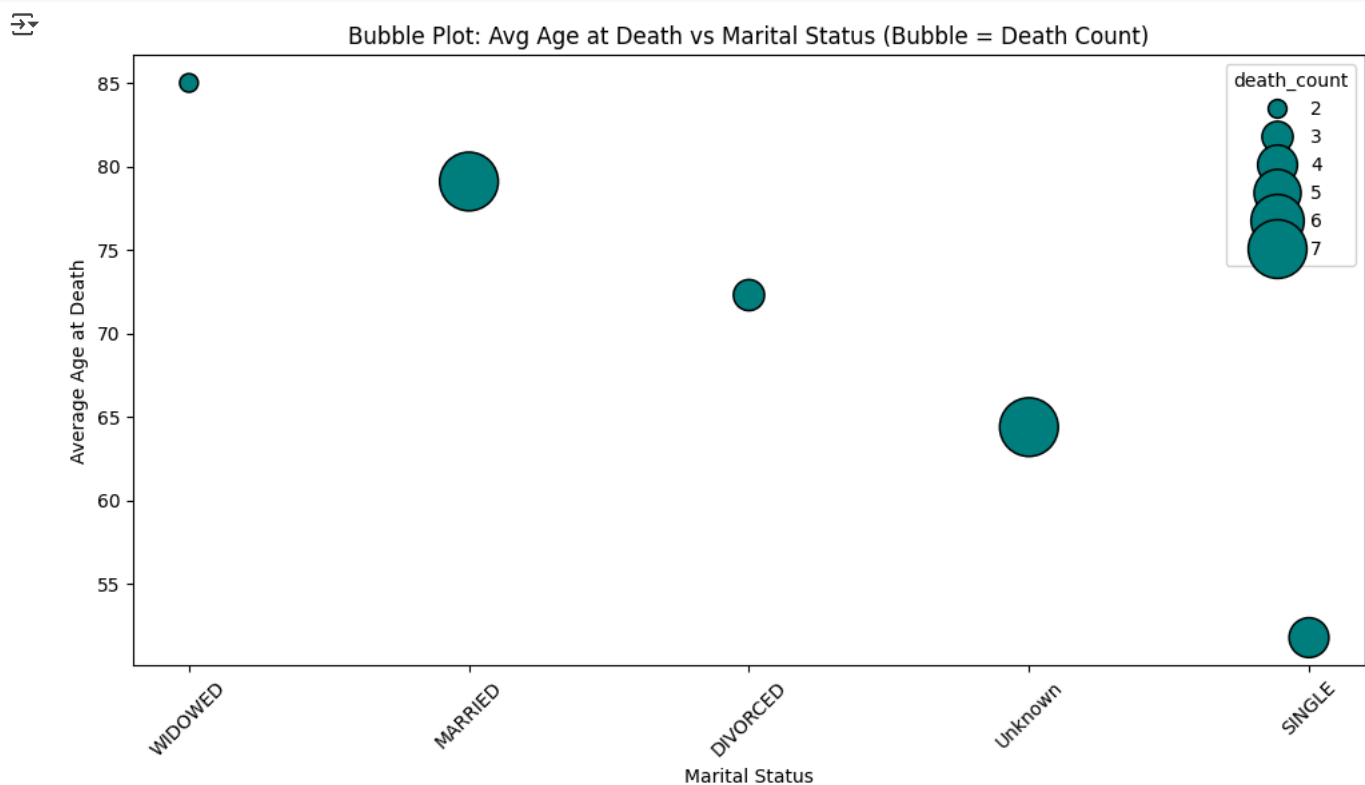
First 5 rows of the data:

	marital_status_cleaned	death_count	avg_age_at_death
0	WIDOWED	2	85.0
1	MARRIED	7	79.1
2	DIVORCED	3	72.3
3	Unknown	7	64.4
4	SINGLE	4	51.8

```

1 # Early Mortality with in 7 days of admission (Bubble Plot)
2 #import seaborn as sns
3
4 # Convert DataFrame to dictionary with lists as values
5 dict_stats = early_mortality_df.to_dict(orient='list')
6
7 # Bubble Plot: Death count (size), average age (y), marital status (x)
8 plt.figure(figsize=(10,6))
9 sns.scatterplot(
10     data=dict_stats,
11     x='marital_status_cleaned',
12     y='avg_age_at_death',
13     size='death_count',
14     legend='brief',
15     sizes=(100, 1000),
16     color='teal',
17     edgecolor='black'
18 )
19 plt.title('Bubble Plot: Avg Age at Death vs Marital Status (Bubble = Death Count)')
20 plt.xlabel('Marital Status')
21 plt.ylabel('Average Age at Death')
22 plt.xticks(rotation=45)
23 plt.tight_layout()
24 plt.show()

```



```

1 # Steps to Load CSV from Google Drive in Google Colab
2 # df_4019_radiology_notes.csv' is a pre conditioned csv files with ICD9 code 4019, which is "Hypertension"
3 # and we have filtered for "Radiology" events
4 from google.colab import auth, drive, files
5
6 drive.mount('/content/drive')
7 auth.authenticate_user()
8
9 df_copy = pd.read_csv('/content/drive/My Drive/df_4019_radiology_notes.csv')
10 print(df_copy.head(3))

```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

	ROW_ID	SUBJECT_ID	HADM_ID	CHARTDATE	CHARTTIME	STORETIME	
0	738405	93207	104110.0	2166-10-09	2166-10-09 20:24:00		NaN
1	738661	24552	NaN	2153-01-12	2153-01-12 22:03:00		NaN
2	738662	10390	NaN	2145-03-13	2145-03-13 02:58:00		NaN

	CATEGORY	DESCRIPTION	CGID	ISERROR	
0	Radiology	DISTINCT PROCEDURAL SERVICE	NaN	NaN	
1	Radiology	CHEST (PORTABLE AP)	NaN	NaN	
2	Radiology	CT PELVIS W/CONTRAST	NaN	NaN	

	TEXT
0	[[**Last Name (LF) **],[**First Name3 (LF) **]] ...
1	[[**2153-1-12**]] 10:03 PM\n CHEST (PORTABLE AP)...
2	[[**2145-3-13**]] 2:58 AM\n CT ABDOMEN W/CONTRAS...

1 Start coding or [generate](#) with AI.

```

1 # cleans and tokenize
2 # import spacy
3 # import re
4 # import string
5 # from collections import Counter
6
7 # import pandas as pd
8 # import numpy as np
9
10 # from spacy.pipeline import EntityRuler
11 # from spacy.tokens import DocBin
12
13 nlp = spacy.load("en_core_web_sm") # Use medSpaCy model here if installed
14 stop_words = nlp.Defaults.stop_words
15
16 def clean_and_tokenize(text,
17                         lowercase=True,
18                         remove_stopwords=True,
19                         remove_punctuation=True,
20                         min_token_len=2):
21
22     if lowercase:
23         text = text.lower()
24
25     doc = nlp(text)
26
27     tokens = []
28     for token in doc:
29         if remove_stopwords and token.text in stop_words:
30             continue
31         if remove_punctuation and token.text in string.punctuation:
32             continue
33         if len(token.text) < min_token_len:
34             continue
35         if token.is_space or token.is_digit:
36             continue
37         tokens.append(token.lemma_.strip())
38
39     return " ".join(tokens)

```

```
1 df_copy['cleaned_tokens'] = df_copy["TEXT"].apply(lambda text: clean_and_tokenize(text))
```

```

1 # from spacy import displacy
2
3 nlp = spacy.load("en_core_web_sm")
4 # Loop through first 3 processed notes and render entities
5 for i in range(10):
6     text = df_copy['cleaned_tokens'][i]
7     doc = nlp(text)
8     displacy.render(doc, style="ent", jupyter=True)
9     print("*****")

```



If name3 **ORG** If eu 8:24 pm **TIME** cta head w&w recon cta neck w&w oc recon clip clip number radiology ct brain perfusion -59 distinct procedural service reason eval ich cva **ORG** contrast optiray amt hospital medical condition year old woman new onset dysarthria know reason examination eval ich cva contraindication iv contrast pfi report evidence acute infarct intracranial hemorrhage brain evidence focal flow limit stenosis occlusion aneurysm great mm artery anterior posterior circulation head calcify plaque left distal common carotid artery cause approximately stenosis

10:03 pm **TIME** chest portable ap clip clip number radiology reason scv cvl **PERSON** check line ptx hospital medical condition year old woman gangrene rt foot **PERSON** rt fem tib reason examination scv cvl check line ptx final report clinical indication central venous catheter placement comparison previous study day early interval placement left subclavian **NORP** central venous catheter terminate junction left brachiocephalic vein **PERSON** superior vena cava pneumothorax identify cardiac silhouette remain enlarge bilateral pleural effusion adjacent area increase lung opacification low lobe note previous mediansternotomy coronary bypass surgery impression central venous catheter terminate confluence left brachiocephalic vein superior vena cava pneumothorax persistent bilateral pleural effusion adjacent lung opacity likely reflect atelectasis infectious process exclude proper clinical setting

2:58 **TIME** ct abdomen contrast ct pelvis contrast clip clip number radiology ct coronal sagittal obl reconstruction ct 150cc nonionic contrast **LOC** reason evaluate diverticulitis **PRODUCT** contrast optiray amt hospital medical condition year old woman diverticuli **ORG** recent admit diverticulitis **PRODUCT** low gi bleed -- ct show thickening stranding paracolic fluid left descending prox sigmoid colon worrisome diverticulitis vs ischemic bowel vs hemorrhage bowel wall present recurrent llq pain diarrhea evaluate diverticulitis perforation reason examination evaluate diverticulitis final report history diverticulitis technique helically acquire ct image lung basis pubic symphysis contrast ct abdomen contrast comparison prior ct scan date demonstrate lung basis mm **PERSON** calcify granuloma left base lung clear pleural effusion liver spleen pancrea adrenal kidney unremarkable see large cm gallbladder stone gallbladder appear unremarkable bowel upper abdomen appear unremarkable free fluid see ct pelvis contrast mark interval improvement thicken sigmoid colon see prior study numerous diverticula minimal stranding sigmoid colon likely clear residual disease free fluid see kidney demonstrate bilateral symmetrical contrast excretion ureter see course bladder filling defect bladder appear unremarkable free fluid present focal fluid collection identify bone window bone demonstrate suspicious lytic blastic lesion impression interval improvement patient sigmoid disease mild stranding sigmoid colon likely relate residua prior disease rule early diverticulitis

11:46 **CARDINAL** ct head w/o contrast clip clip number radiology reason cerebral edema central pontine myelinolysis mass hospital medical condition year old man liver transplant w/ persistent decrease mental status reason examination cerebral edema central pontine myelinolysis mass final report history year old **DATE** man status post liver transplant persistent decrease mental status rule cerebral edema central pontine myelinolysis technique contiguous axial image head obtain iv contrast skull base vertex ct head w/o contrast sulci ventricle appropriate patient age intra extraaxial fluid collection masse shift midline structure observe doctor -white matter differentiation unremarkable impression evidence cerebral edema

1:06 pm **TIME** chest portable ap clip clip number radiology reason chf hospital medical condition year old woman niddm cad aicd pace acute sob leg swelling reason examination chf final report indication year old **DATE** woman diabete cad present follow icd placement acute shortness breath leg swelling rule chf comparison upright ap chest radiograph interval placement left side dual chamber icd lead locate right atrium right ventricle persistent cardiomegaly slight decrease extent upper zone redistribution pulmonary vascularity increase size large right side pleural effusion associate right middle low lobe collapse consolidation linear opacity left cp angle likely represent subsegmental atelectasis interval decrease extent left pleural effusion soft tissue osseous structure unchanged impression new left side dual chamber icd lead appropriate position pneumothorax persistent cardiomegaly **GPE** interval decrease chf interval increase large right pleural effusion small left pleural effusion leave basilar subsegmental atelectasis right middle low lobe collapse consolidation

2:21 pm **TIME** chest portable ap clip clip number radiology reason acute sob hospital medical condition year old man pvd ischemic lle cad **PERSON** heparinwithacute sob low bp reason examination acute sob final report indication shortness breath low blood pressure comparison previous study early day **DATE** left picc line remain place terminate distal superior vena cava patient mediansternotomy coronary bypass surgery heart enlarge stable size previously note mild congestive heart failure pattern demonstrate significantly change allow difference technique confluent area consolidation see lung pleural effusion evident left cp sulcus exclude impression significant change recent study early day **DATE** persistent mild congestive heart failure pattern

4:00 pm **TIME** chest pa lat clip clip number radiology reason follow cxr pt pneumomia rml hospital medical condition year old man sob chest leave le dvt reason examination follow cxr pt pneumomia rml final report indication right middle lobe pneumonia chest view consolidation collapse right middle lobe anterior segment right low lobe significantly change minimal linear atelectasis left lung base small right pleural effusion present left costophrenic angle clear pulmonary vascularity

```

1 # Top most frequent diagnoses (by short title) grouped by marital status.
2
3 query_early_critical = """
4 SELECT
5     CASE
6         WHEN adm.marital_status IS NULL
7             OR adm.marital_status IN ('UNKNOWN (DEFAULT)', 'None', '')
8             THEN 'Unknown'
9         ELSE adm.marital_status
10    END AS marital_status_cleaned,
11    dd.short_title AS diagnosis_name,
12    COUNT(*) AS diagnosis_count
13 FROM
14     `physionet-data.mimiciii_demo.admissions` AS adm
15 JOIN
16     `physionet-data.mimiciii_demo.diagnoses_icd` AS d
17   ON adm.hadm_id = d.hadm_id
18 JOIN
19     `physionet-data.mimiciii_demo.d_icd_diagnoses` AS dd
20   ON d.icd9_code = dd.icd9_code
21 WHERE
22     dd.short_title IS NOT NULL
23 GROUP BY
24     marital_status_cleaned, diagnosis_name
25 ORDER BY
26     marital_status_cleaned, diagnosis_count DESC
27 LIMIT 10000;
28 """
29
30 query_early_critical_df = run_bq_query(query_early_critical)

```

```
1 query_early_critical_df = run_bq_query(query_early_critical)
```



Query executed successfully! Data loaded into pandas DataFrame.

First 5 rows of the data:

	marital_status_cleaned	diagnosis_name	diagnosis_count
0	DIVORCED	Hypertension NOS	4
1	DIVORCED	Atrial fibrillation	3
2	DIVORCED	CHF NOS	2
3	DIVORCED	Hypothyroidism NOS	2
4	DIVORCED	Ac posthemorrhag anemia	2

```

1 # Assign Critical Rating
2
3 # Assign critical rating based on frequency rank within each marital group
4 query_early_critical_df['critical_rating'] = query_early_critical_df.groupby('marital_status_cleaned')['diagnosis_count']
5                                           .transform(lambda x: pd.qcut(x.rank(method='first'), 5, labels=[1, 2, 3, 4,
6
7 # Preview
8 query_early_critical_df.head(1000)
9

```




1 to 50 of 896 entries

Filter



index	marital_status_cleaned	diagnosis_name	diagnosis_count	critical_rating
0	DIVORCED	Hypertension NOS	4	5
1	DIVORCED	Atrial fibrillation	3	5
2	DIVORCED	CHF NOS	2	5
3	DIVORCED	Hypothyroidism NOS	2	5
4	DIVORCED	Ac posthemorrhag anemia	2	5
5	DIVORCED	DMII wo cmp nt st uncnr	2	5
6	DIVORCED	Delirium d/t other cond	2	5
7	DIVORCED	Severe sepsis	2	5
8	DIVORCED	Acute respiratry failure	2	5
9	DIVORCED	Cholangitis	1	1
10	DIVORCED	Shock w/o trauma NEC	1	1
11	DIVORCED	Emphysema NEC	1	1
12	DIVORCED	Anoxic brain damage	1	1
13	DIVORCED	Urin tract infection NOS	1	1
14	DIVORCED	Traum arthropathy-pelvis	1	1
15	DIVORCED	Hyperlipidemia NEC/NOS	1	1
16	DIVORCED	Alcoh dep NEC/NOS-unspec	1	1
17	DIVORCED	Septic shock	1	1
18	DIVORCED	Ac kidney fail, tubr necr	1	1
19	DIVORCED	Fx malar/maxillary-close	1	1
20	DIVORCED	Ac alcoholic hepatitis	1	1
21	DIVORCED	Adv eff opiates	1	1
22	DIVORCED	Hypopotassemia	1	1
23	DIVORCED	Idiopathic scoliosis	1	2
24	DIVORCED	Hx-prostatic malignancy	1	2
25	DIVORCED	Mitral valve disorder	1	2
26	DIVORCED	Dvrtclo colon w/o hmrg	1	2
27	DIVORCED	Persistent vegtv state	1	2
28	DIVORCED	Paralysis agitans	1	2
29	DIVORCED	Acute pancreatitis	1	2
30	DIVORCED	Pneumonia, organism NOS	1	2
31	DIVORCED	Comp-ren dialys dev/grft	1	2
32	DIVORCED	Blood in stool	1	2
33	DIVORCED	Hx-circulatory dis NEC	1	2
34	DIVORCED	Septicemia NOS	1	2
35	DIVORCED	Hyposmolality	1	2
36	DIVORCED	Hyp kid NOS w cr kid V	1	3
37	DIVORCED	Ac systolic hrt failure	1	3
38	DIVORCED	Ventricular fibrillation	1	3
39	DIVORCED	Subendo infarct, initial	1	3
40	DIVORCED	End stage renal disease	1	3
41	DIVORCED	Hematoma complic proc	1	3
42	DIVORCED	Cardiogenic shock	1	3
43	DIVORCED	Unc behav neo bone	1	3
44	DIVORCED	Cerv spondyl w myelopath	1	3
45	DIVORCED	Personal history of fall	1	3
46	DIVORCED	Status autm crd dfbrltr	1	3
47	DIVORCED	latrogen pulm emb/infarc	1	3
48	DIVORCED	Other pulmonary insuff	1	3
49	DIVORCED	latrogenc hypotnsion NEC	1	3

Show 50 per page

1

2

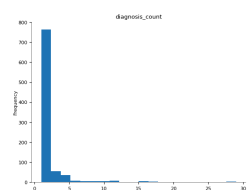
10

18

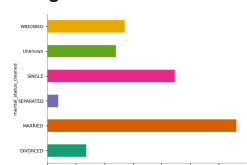
Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

WARNING:root:Quickchart encountered unexpected dtypes in columns: "(['critical_rating'],)"

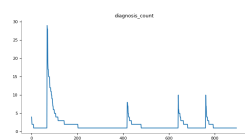
Distributions



Categorical distributions



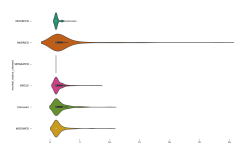
values



Faceted distributions

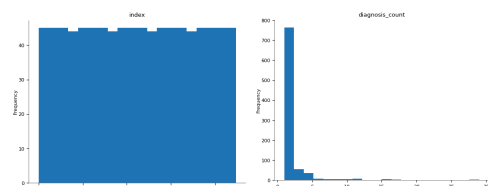
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`

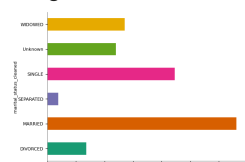


WARNING:root:Quickchart encountered unexpected dtypes in columns: "(['critical_rating'],)"

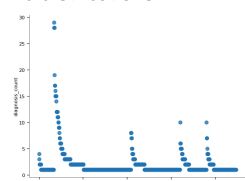
Distributions



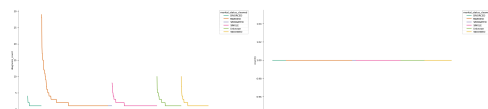
Categorical distributions



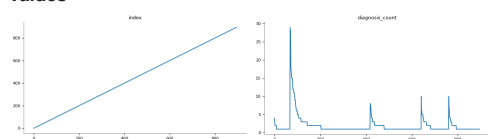
2-d distributions



Time series



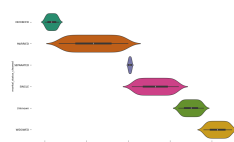
Values



Faceted distributions

<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`



<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`



Next steps:

[Generate code with query_early_critical_df](#)[View recommended plots](#)[New interactive sheet](#)

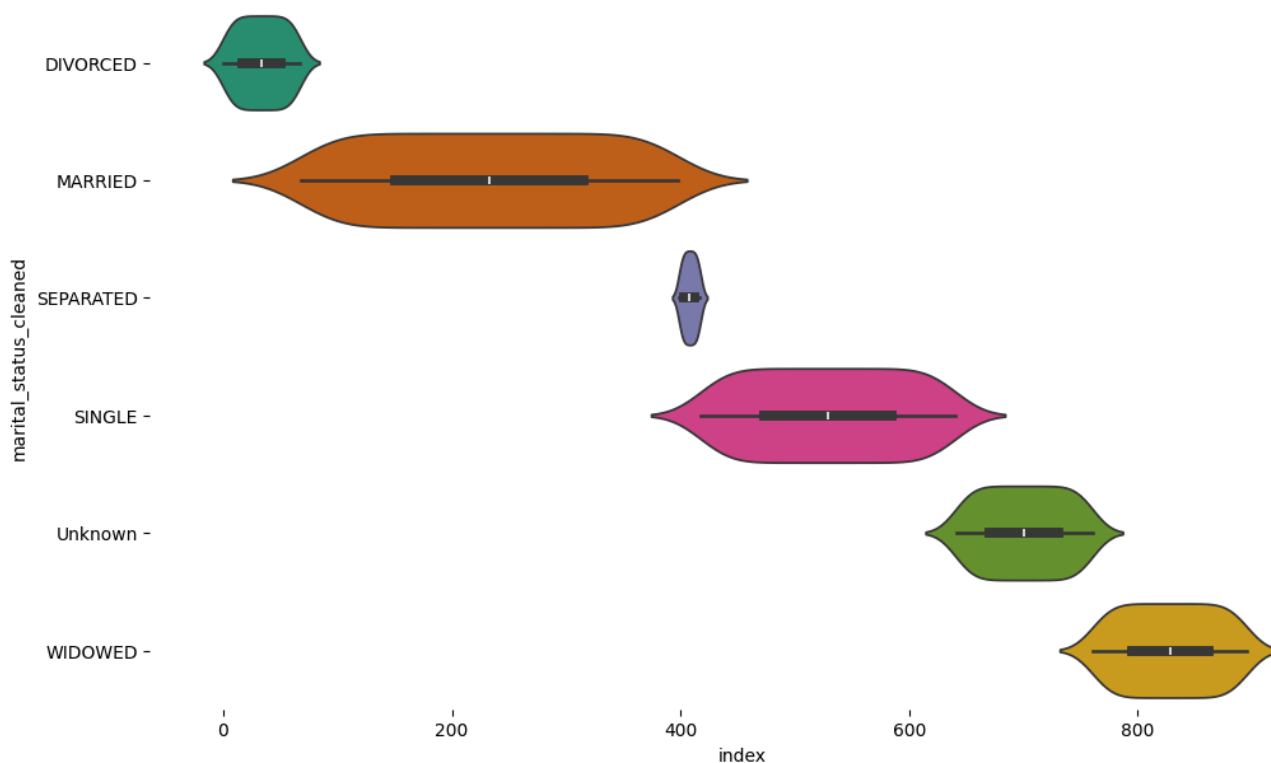
```

1 from matplotlib import pyplot as plt
2 import seaborn as sns
3 figsize = (12, 1.2 * len(_df_17['marital_status_cleaned'].unique()))
4 plt.figure(figsize=figsize)
5 sns.violinplot(_df_17, x='index', y='marital_status_cleaned', inner='box', palette='Dark2')
6 sns.despine(top=True, right=True, bottom=True, left=True)

```

⚠ /tmp/ipython-input-108-72477812.py:5: FutureWarning:

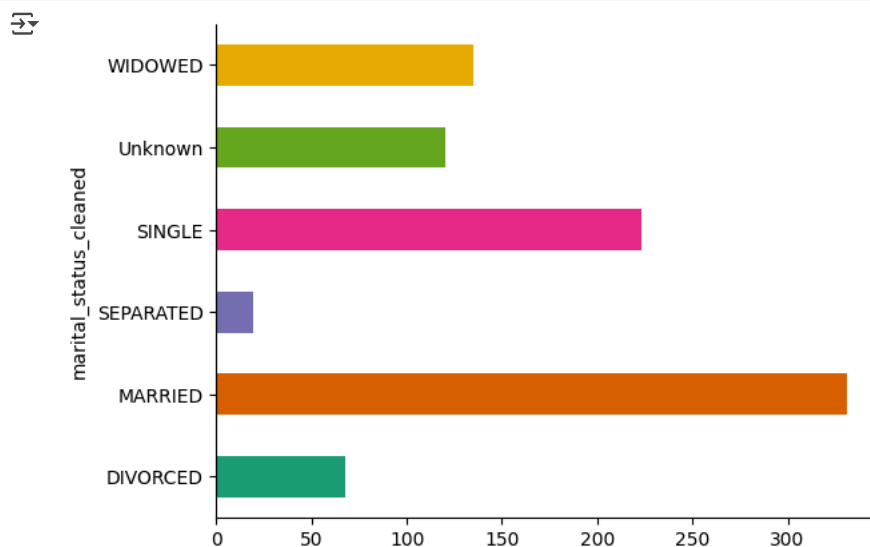
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue`



```

1 from matplotlib import pyplot as plt
2 import seaborn as sns
3 _df_11.groupby('marital_status_cleaned').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
4 plt.gca().spines[['top', 'right']].set_visible(False)

```



```

1 import plotly.express as px
2
3 fig = px.scatter(

```

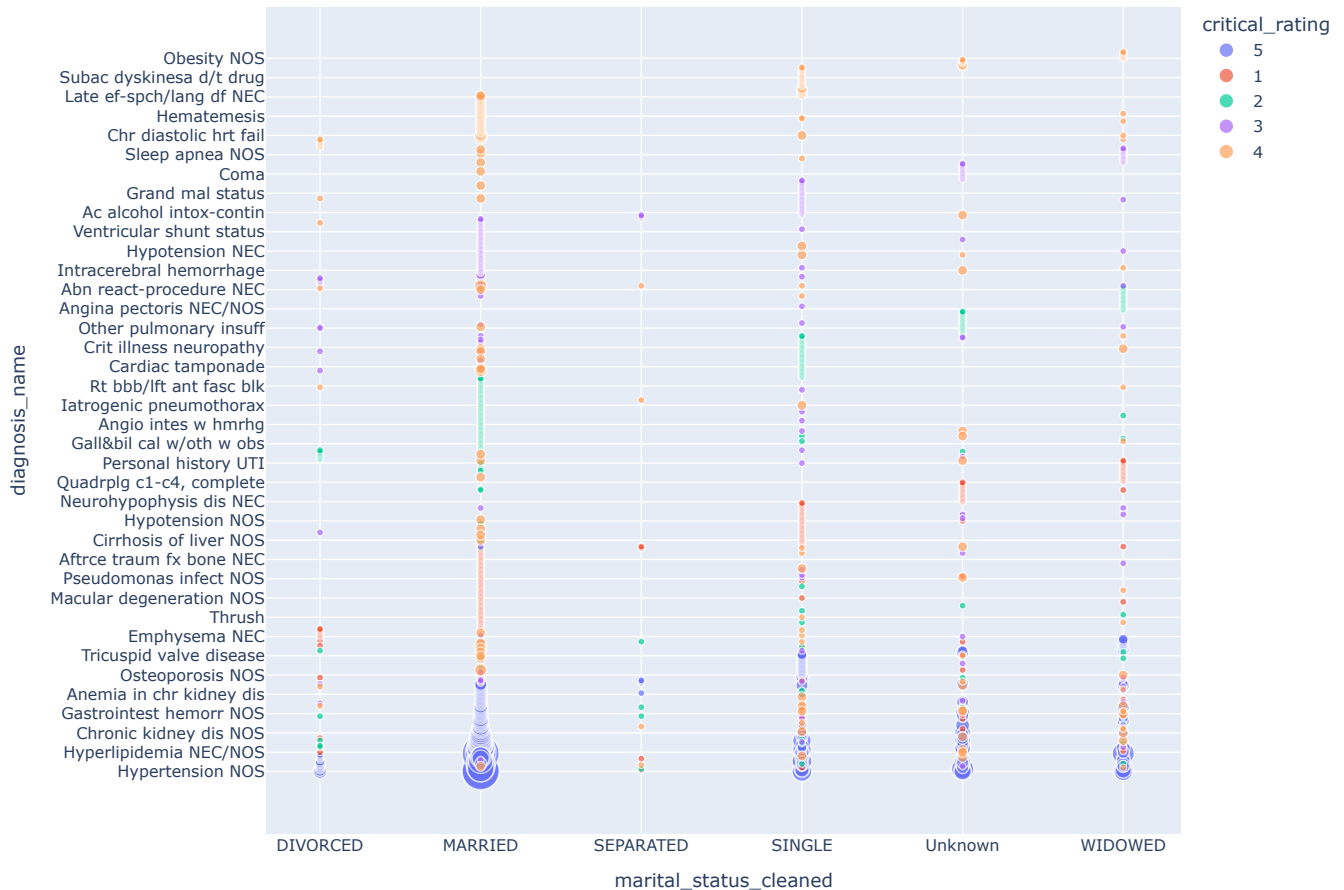
```

4 query_early_critical_df,
5 x='marital_status_cleaned',
6 y='diagnosis_name',
7 size='diagnosis_count',
8 color='critical_rating',
9 color_continuous_scale='Reds',
10 title='Critical Diagnoses per Marital Status',
11 hover_data=['diagnosis_name', 'diagnosis_count', 'critical_rating']
12 )
13 fig.update_layout(height=800)
14 fig.show()
15

```



Critical Diagnoses per Marital Status



```
1 query_early_critical_df['cleaned_tokens'] = query_early_critical_df["diagnosis_name"].apply(lambda text: clean_and_tokeni
```

Double-click (or enter) to edit

```

1
2 query_early_critical_df['cleaned_tokens'] = query_early_critical_df["diagnosis_name"].apply(lambda text: clean_and_tokeni
3
4 nlp = spacy.load("en_core_web_sm")
5 # Loop through first 3 processed notes and render entities
6 for i in range(100):
7     text = query_early_critical_df['cleaned_tokens'][i]
8     doc = nlp(text)
9     displacy.render(doc, style="ent", jupyter=True)
10    # print("*****")

```