# TIME TO READMISSSION IN DIABETIC PATIENTS USING SURVIVAL ANALYSIS

## Created by: AJIROLA AMUDAT

In [1]:
```
# Import Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from lifelines import KaplanMeierFitter, CoxPHFitter
from lifelines.statistics import logrank_test
from lifelines.statistics import multivariate_logrank_test
```

```
C:\Users\HELLO\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version
'1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
# To Load Dataset
data = pd.read_csv("diabetic_data.csv")
```

In [3]:
```
# Show the first 10 rows of the dataset
data.head(10)
```

Out [3]:

| | encounter_id | patient_nbr | race | gender | age | weight | admission_type_id | discharge_disposition_id | admission_source_id | time_in_hospital | ... | citoglipton |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2278392 | 8222157 | Caucasian | Female | [0-10) | ? | 6 | 25 | 1 | 1 | ... | No |
| **1** | 149190 | 55629189 | Caucasian | Female | [10-20) | ? | 1 | 1 | 7 | 3 | ... | No |
| **2** | 64410 | 86047875 | AfricanAmerican | Female | [20-30) | ? | 1 | 1 | 7 | 2 | ... | No |
| **3** | 500364 | 82442376 | Caucasian | Male | [30-40) | ? | 1 | 1 | 7 | 2 | ... | No |
| **4** | 16680 | 42519267 | Caucasian | Male | [40-50) | ? | 1 | 1 | 7 | 1 | ... | No |
| **5** | 35754 | 82637451 | Caucasian | Male | [50-60) | ? | 2 | 1 | 2 | 3 | ... | No |
| **6** | 55842 | 84259809 | Caucasian | Male | [60-70) | ? | 3 | 1 | 2 | 4 | ... | No |
| **7** | 63768 | 114882984 | Caucasian | Male | [70-80) | ? | 1 | 1 | 7 | 5 | ... | No |
| **8** | 12522 | 48330783 | Caucasian | Female | [80-90) | ? | 2 | 1 | 4 | 13 | ... | No |
| **9** | 15738 | 63555939 | Caucasian | Female | [90-100) | ? | 3 | 3 | 4 | 12 | ... | No |

10 rows × 50 columns

### DATA CLEANING

In [4]:
```
# To know the dataset columns
data.columns
```

Out [4]:
```
Index(['encounter_id', 'patient_nbr', 'race', 'gender', 'age', 'weight',
       'admission_type_id', 'discharge_disposition_id', 'admission_source_id',
       'time_in_hospital', 'payer_code', 'medical_specialty',
       'num_lab_procedures', 'num_procedures', 'num_medications',
       'number_outpatient', 'number_emergency', 'number_inpatient', 'diag_1',
       'diag_2', 'diag_3', 'number_diagnoses', 'max_glu_serum', 'A1Cresult',
       'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide',
       'glimepiride', 'acetohexamide', 'glipizide', 'glyburide', 'tolbutamide',
       'pioglitazone', 'rosiglitazone', 'acarbose', 'miglitol', 'troglitazone',
       'tolazamide', 'examide', 'citoglipton', 'insulin',
       'glyburide-metformin', 'glipizide-metformin',
       'glimepiride-pioglitazone', 'metformin-rosiglitazone',
       'metformin-pioglitazone', 'change', 'diabetesMed', 'readmitted'],
      dtype='object')
```

In [5]:
```
# To know the shape or size of the dataset
data.shape
```

Out [5]:
```
(101766, 50)
```

In [6]:
```
# To know the data types and get familiarise with the data to know how to work with them.
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   encounter_id              101766 non-null  int64 
 1   patient_nbr               101766 non-null  int64 
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    101766 non-null  object
 6   admission_type_id         101766 non-null  int64 
 7   discharge_disposition_id  101766 non-null  int64 
 8   admission_source_id       101766 non-null  int64 
 9   time_in_hospital          101766 non-null  int64 
 10  payer_code                101766 non-null  object
 11  medical_specialty         101766 non-null  object
 12  num_lab_procedures        101766 non-null  int64 
 13  num_procedures            101766 non-null  int64 
 14  num_medications           101766 non-null  int64 
 15  number_outpatient         101766 non-null  int64 
 16  number_emergency          101766 non-null  int64 
 17  number_inpatient          101766 non-null  int64 
 18  diag_1                    101766 non-null  object
 19  diag_2                    101766 non-null  object
 20  diag_3                    101766 non-null  object
 21  number_diagnoses          101766 non-null  int64 
 22  max_glu_serum             5346 non-null    object
 23  A1Cresult                 17018 non-null   object
 24  metformin                 101766 non-null  object
 25  repaglinide               101766 non-null  object
 26  nateglinide               101766 non-null  object
 27  chlorpropamide            101766 non-null  object
 28  glimepiride               101766 non-null  object
 29  acetohexamide             101766 non-null  object
 30  glipizide                 101766 non-null  object
 31  glyburide                 101766 non-null  object
 32  tolbutamide               101766 non-null  object
 33  pioglitazone              101766 non-null  object
 34  rosiglitazone             101766 non-null  object
 35  acarbose                  101766 non-null  object
 36  miglitol                  101766 non-null  object
 37  troglitazone              101766 non-null  object
 38  tolazamide                101766 non-null  object
 39  examide                   101766 non-null  object
 40  citoglipton               101766 non-null  object
 41  insulin                   101766 non-null  object
 42  glyburide-metformin       101766 non-null  object
 43  glipizide-metformin       101766 non-null  object
 44  glimepiride-pioglitazone  101766 non-null  object
 45  metformin-rosiglitazone   101766 non-null  object
 46  metformin-pioglitazone    101766 non-null  object
 47  change                    101766 non-null  object
 48  diabetesMed               101766 non-null  object
 49  readmitted                101766 non-null  object
dtypes: int64(13), object(37)
memory usage: 38.8+ MB
```

In [7]:
```
# To check for misssing value
data.isnull().sum()
```

Out [7]:
```
encounter_id                    0
patient_nbr                     0
race                            0
gender                          0
age                             0
weight                          0
admission_type_id               0
discharge_disposition_id        0
admission_source_id             0
time_in_hospital                0
payer_code                      0
medical_specialty               0
num_lab_procedures              0
num_procedures                  0
num_medications                 0
number_outpatient               0
number_emergency                0
number_inpatient                0
diag_1                          0
diag_2                          0
diag_3                          0
number_diagnoses                0
max_glu_serum               96420
A1Cresult                   84748
metformin                       0
repaglinide                     0
nateglinide                     0
chlorpropamide                  0
glimepiride                     0
acetohexamide                   0
glipizide                       0
glyburide                       0
tolbutamide                     0
pioglitazone                    0
rosiglitazone                   0
acarbose                        0
miglitol                        0
troglitazone                    0
tolazamide                      0
examide                         0
citoglipton                     0
insulin                         0
glyburide-metformin             0
glipizide-metformin             0
glimepiride-pioglitazone        0
metformin-rosiglitazone         0
metformin-pioglitazone          0
change                          0
diabetesMed                     0
readmitted                      0
dtype: int64
```

In [8]:
```
# To Keep necessary variables
data = data[['age', 'time_in_hospital', 'num_lab_procedures',
         'num_medications', 'number_diagnoses', 'insulin', 'readmitted']]
```

In [9]:
```
# To know the sum of duplicates in the dataset
duplicate_count = data.duplicated().sum()
duplicate_count
```

Out [9]:
```
3877
```

In [10]:
```
# To remove duplicate
data.drop_duplicates()
```

Out [10]:

| | age | time_in_hospital | num_lab_procedures | num_medications | number_diagnoses | insulin | readmitted |
|---|---|---|---|---|---|---|---|
| **0** | [0-10) | 1 | 41 | 1 | 1 | No | NO |
| **1** | [10-20) | 3 | 59 | 18 | 9 | Up | >30 |
| **2** | [20-30) | 2 | 11 | 13 | 6 | No | NO |
| **3** | [30-40) | 2 | 44 | 16 | 7 | Up | NO |
| **4** | [40-50) | 1 | 51 | 8 | 5 | Steady | NO |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **101761** | [70-80) | 3 | 51 | 16 | 9 | Down | >30 |
| **101762** | [80-90) | 5 | 33 | 18 | 9 | Steady | NO |
| **101763** | [70-80) | 1 | 53 | 9 | 13 | Down | NO |
| **101764** | [80-90) | 10 | 45 | 21 | 9 | Up | NO |
| **101765** | [70-80) | 6 | 13 | 3 | 9 | No | NO |

97889 rows × 7 columns

In [11]:
```
# To Remove rows with missing age
data = data[data['age'] != '?']
```

In [12]:
```
 # To Keep valid insulin types
data = data[data['insulin'].isin(['Up', 'Down', 'Steady'])]
```

In [13]:
```
# To Define event (readmitted <30 days = 1) and duration (time_in_hospital)
data['event'] = data['readmitted'].apply(lambda x: 1 if x == '<30' else 0)
data['duration'] = data['time_in_hospital']
```

In [14]:
```
# To create ordered age categories
age_order = ['[0-10)', '[10-20)', '[20-30)', '[30-40)', '[40-50)', '[50-60)',
             '[60-70)', '[70-80)', '[80-90)', '[90-100)']
data['age'] = pd.Categorical(data['age'], categories=age_order, ordered=True)
```

In [15]:
```
# To create medication groupings
data['med_group'] = pd.cut(data['num_medications'],
                           bins=[0, 10, 20, 40, data['num_medications'].max()],
                           labels=['Low', 'Moderate', 'High', 'Very High'],
                           include_lowest=True)
```

In [16]:
```
# To show the first 10 rows of the dataset
data.head(10)
```

Out [16]:

| | age | time_in_hospital | num_lab_procedures | num_medications | number_diagnoses | insulin | readmitted | event | duration | med_group |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | [10-20) | 3 | 59 | 18 | 9 | Up | >30 | 0 | 3 | Moderate |
| **3** | [30-40) | 2 | 44 | 16 | 7 | Up | NO | 0 | 2 | Moderate |
| **4** | [40-50) | 1 | 51 | 8 | 5 | Steady | NO | 0 | 1 | Low |
| **5** | [50-60) | 3 | 31 | 16 | 9 | Steady | >30 | 0 | 3 | Moderate |
| **6** | [60-70) | 4 | 70 | 21 | 7 | Steady | NO | 0 | 4 | High |
| **8** | [80-90) | 13 | 68 | 28 | 8 | Steady | NO | 0 | 13 | High |
| **9** | [90-100) | 12 | 33 | 18 | 8 | Steady | NO | 0 | 12 | Moderate |
| **10** | [40-50) | 9 | 47 | 17 | 9 | Steady | >30 | 0 | 9 | Moderate |
| **11** | [60-70) | 7 | 62 | 11 | 7 | Steady | <30 | 1 | 7 | Moderate |
| **12** | [40-50) | 7 | 60 | 15 | 8 | Down | <30 | 1 | 7 | Moderate |

In [17]:
```
# To know the shape or size of the dataset
data.shape
```

Out [17]:
```
(54383, 10)
```

## EXPLORATORY DATA ANALYSIS (EDA)

In [18]:
```
# To get the summary statistics of the data
data.describe()
```

Out [18]:

| | time_in_hospital | num_lab_procedures | num_medications | number_diagnoses | event | duration |
|---|---|---|---|---|---|---|
| **count** | 54383.000000 | 54383.000000 | 54383.000000 | 54383.000000 | 54383.000000 | 54383.000000 |
| **mean** | 4.671956 | 44.811191 | 17.601144 | 7.566188 | 0.121380 | 4.671956 |
| **std** | 3.052438 | 19.841404 | 8.606966 | 1.917152 | 0.326571 | 3.052438 |
| **min** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 |
| **25%** | 2.000000 | 33.000000 | 12.000000 | 6.000000 | 0.000000 | 2.000000 |
| **50%** | 4.000000 | 46.000000 | 16.000000 | 9.000000 | 0.000000 | 4.000000 |
| **75%** | 6.000000 | 59.000000 | 22.000000 | 9.000000 | 0.000000 | 6.000000 |
| **max** | 14.000000 | 132.000000 | 81.000000 | 16.000000 | 1.000000 | 14.000000 |

## KAPLAN-MEIR SURVIVAL CURVE

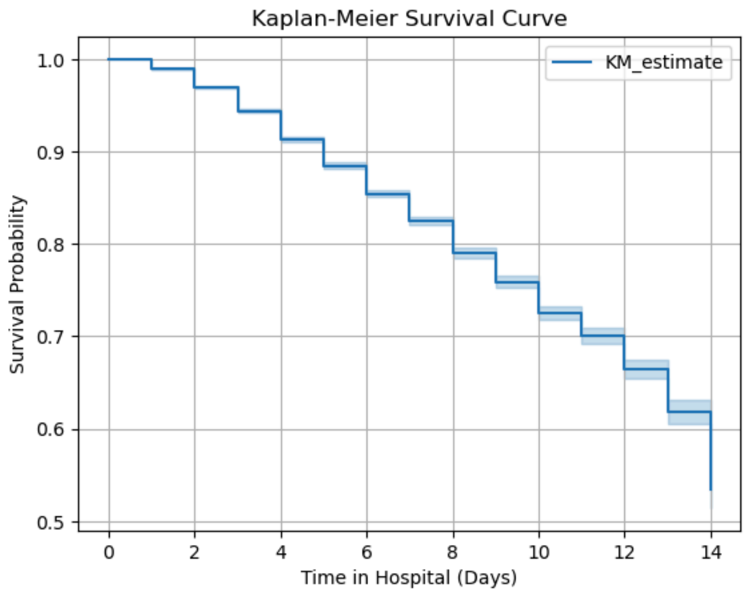
### Overall Kaplan-Meier Estimator

In [19]:
```
# Overall Kaplan-Meier Estimator
kmf = KaplanMeierFitter()
```

In [20]:
```
# To fit KM model
kmf.fit(data['duration'], event_observed=data['event'])
```

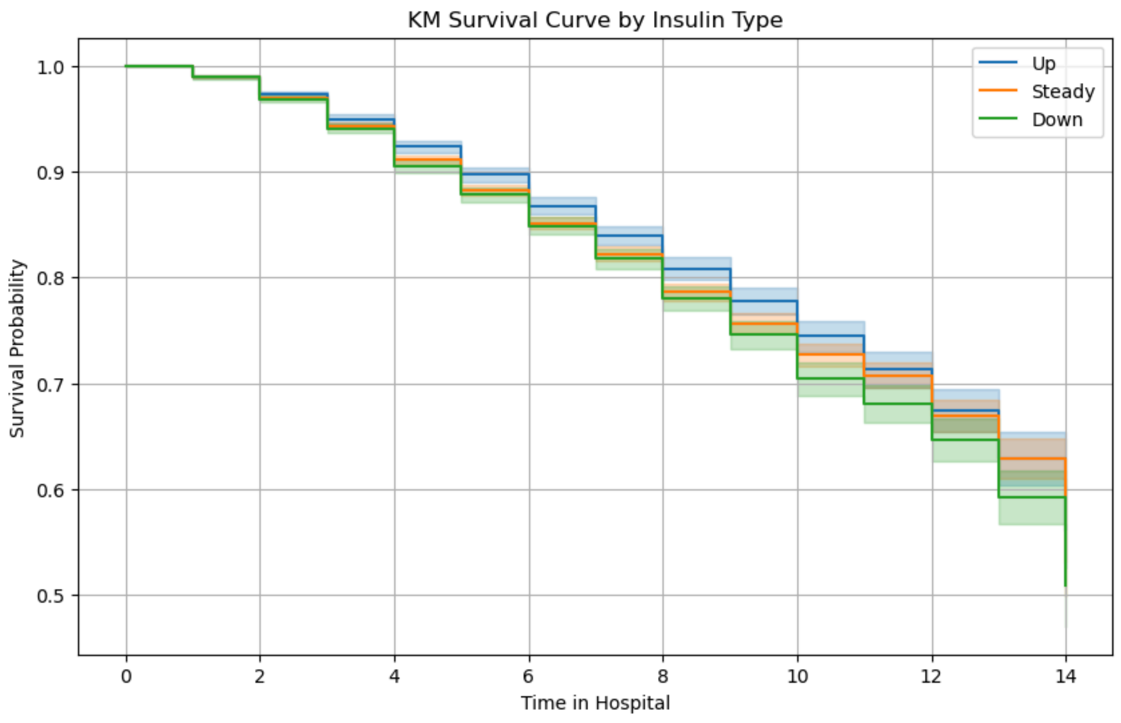
Out [20]:
```
<lifelines.KaplanMeierFitter:"KM_estimate", fitted with 54383 total observations, 47782 right-censored observations>
```

In [21]:
```
# To Plot KM survival function
kmf.plot_survival_function()
plt.title("Kaplan-Meier Survival Curve")
plt.xlabel("Time in Hospital (Days)")
plt.ylabel("Survival Probability")
plt.grid(True)
plt.show()
```



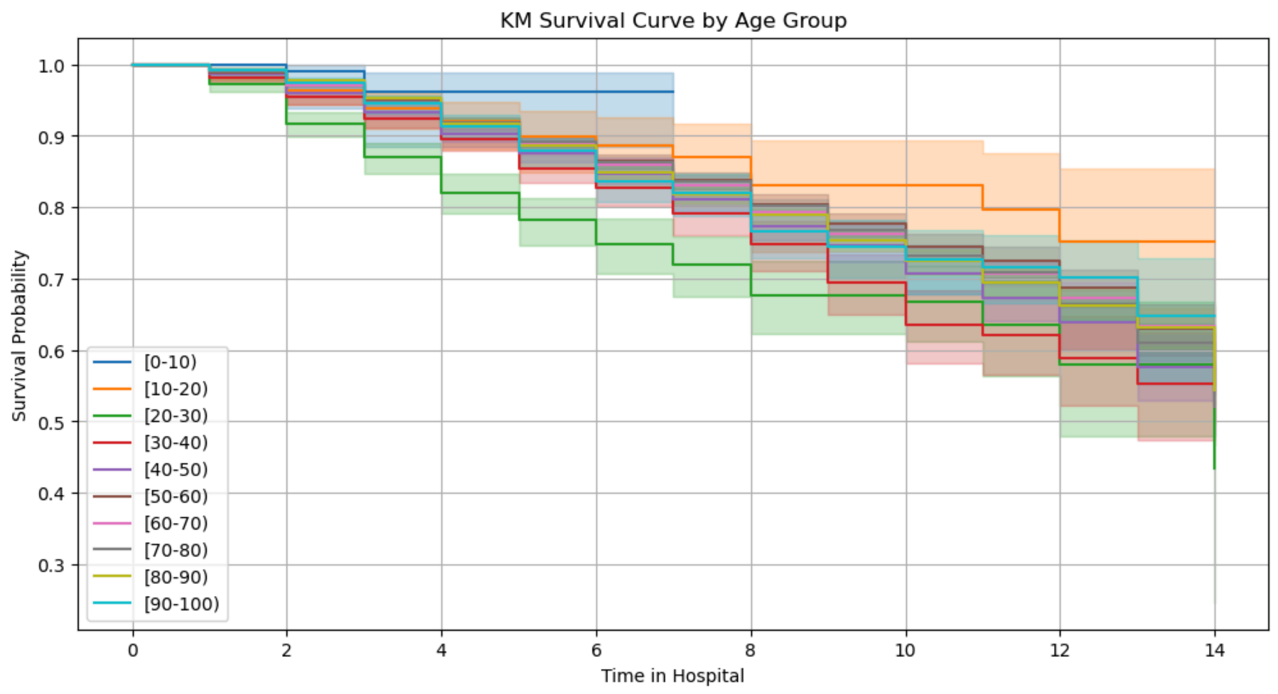
### Kaplan-Meier Curve by Insulin Type

In [22]:
```
# To plot Kaplan-Meier curve by insulin type
kmf = KaplanMeierFitter()
plt.figure(figsize=(10, 6))
for grp in data['insulin'].dropna().unique():
    grp_data = data[data['insulin'] == grp]
    kmf.fit(grp_data['duration'], grp_data['event'], label=grp)
    kmf.plot_survival_function()
plt.title("KM Survival Curve by Insulin Type")
plt.xlabel("Time in Hospital")
plt.ylabel("Survival Probability")
plt.legend()
plt.grid(True)
plt.show()
```



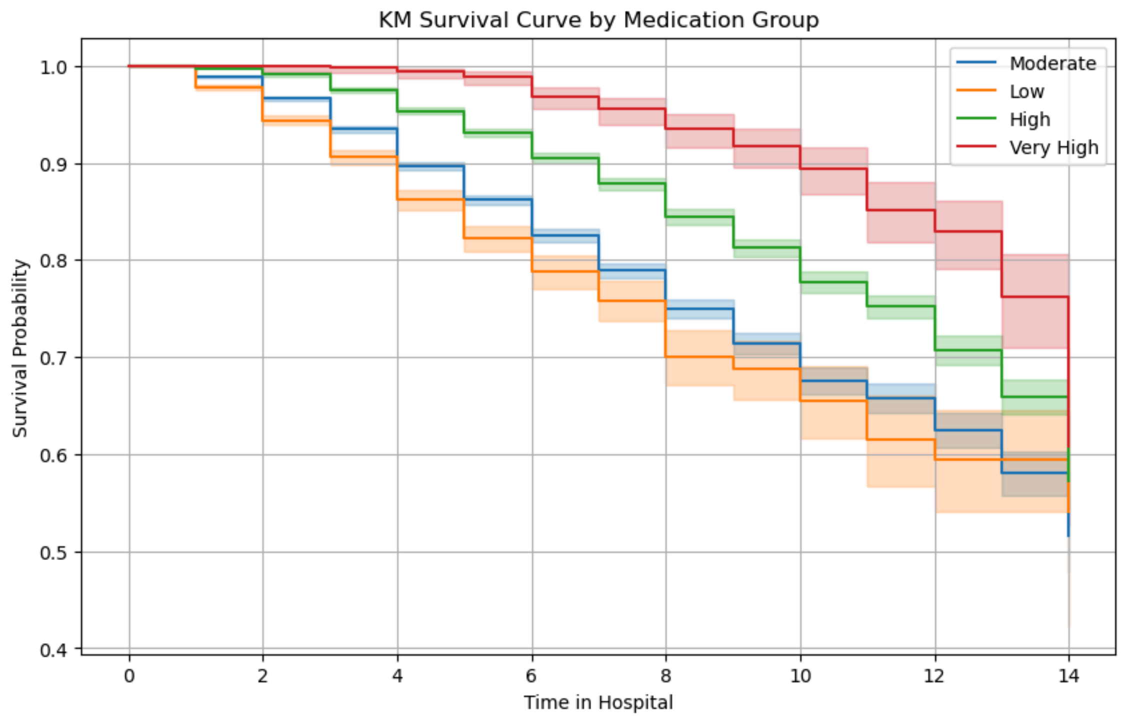
### Kaplan-Meier curve by Age group

In [23]:
```
# To plot Kaplan-Meier curve by Age group
plt.figure(figsize=(12, 6))
for grp in data['age'].cat.categories:
    grp_data = data[data['age'] == grp]
    if not grp_data.empty:
        kmf.fit(grp_data['duration'], grp_data['event'], label=str(grp))
        kmf.plot_survival_function()
plt.title("KM Survival Curve by Age Group")
plt.xlabel("Time in Hospital")
plt.ylabel("Survival Probability")
plt.legend()
plt.grid(True)
plt.show()
```



**Kaplan-Meier curve by Medication group**

In [24]:
```
# To plot Kaplan-Meier curve by medication group
plt.figure(figsize=(10, 6))
for grp in data['med_group'].dropna().unique():
    grp_data = data[data['med_group'] == grp]
    kmf.fit(grp_data['duration'], grp_data['event'], label=str(grp))
    kmf.plot_survival_function()
plt.title("KM Survival Curve by Medication Group")
plt.xlabel("Time in Hospital")
plt.ylabel("Survival Probability")
plt.legend()
plt.grid(True)
plt.show()
```



**LOG RANK TEST**

**Log-rank tests to compare survival curves across groups**

In [25]:
```
# To compare survival curves across groups
print(" Log-Rank Test: Insulin")
results_insulin = print(multivariate_logrank_test(data['duration'], data['insulin'], data['event']).summary)
print(results_insulin)
print(" Log-Rank Test: Age")
results_age = print(multivariate_logrank_test(data['duration'], data['age'], data['event']).summary)
print(results_age)
print(" Log-Rank Test: Medication Group")
results_medication = print(multivariate_logrank_test(data['duration'], data['med_group'], data['event']).summary)
print(results_medication)
```

```
 Log-Rank Test: Insulin
   test_statistic         p   -log2(p)
0        19.30635  0.000064  13.926588
None
 Log-Rank Test: Age
   test_statistic             p   -log2(p)
0      124.428671  1.653295e-22  72.357074
None
 Log-Rank Test: Medication Group
   test_statistic              p   -log2(p)
```

```
0      729.335946  9.135024e-158  521.67323
None
```

### Cox Proportional Hazards Model

In [26]:
```python
# To convert categorical variables to dummy variables
data_encoded = pd.get_dummies(data[['age', 'insulin', 'med_group',  'duration', 'event']],
                              columns=['age', 'insulin', 'med_group'], drop_first=True)
```

In [27]:
```python
# To fit Cox Model
cph = CoxPHFitter()
cph.fit(data_encoded, duration_col='duration', event_col='event')
cph.print_summary()
```

| | |
|---|---|
| **model** | lifelines.CoxPHFitter |
| **duration col** | 'duration' |
| **event col** | 'event' |
| **baseline estimation** | breslow |
| **number of observations** | 54383 |
| **number of events observed** | 6601 |
| **partial log-likelihood** | -65285.39 |
| **time fit was run** | 2025-05-26 11:08:57 UTC |

| | coef | exp(coef) | se(coef) | coef lower 95% | coef upper 95% | exp(coef) lower 95% | exp(coef) upper 95% | cmp to | z | p | -log2(p) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **age_[10-20)** | 0.71 | 2.04 | 0.60 | -0.46 | 1.89 | 0.63 | 6.63 | 0.00 | 1.19 | 0.24 | 2.09 |
| **age_[20-30)** | 1.65 | 5.21 | 0.58 | 0.51 | 2.79 | 1.66 | 16.31 | 0.00 | 2.84 | <0.005 | 7.77 |
| **age_[30-40)** | 1.35 | 3.85 | 0.58 | 0.21 | 2.49 | 1.23 | 12.01 | 0.00 | 2.32 | 0.02 | 5.62 |
| **age_[40-50)** | 1.27 | 3.54 | 0.58 | 0.13 | 2.40 | 1.14 | 11.03 | 0.00 | 2.18 | 0.03 | 5.11 |
| **age_[50-60)** | 1.14 | 3.13 | 0.58 | 0.01 | 2.28 | 1.01 | 9.75 | 0.00 | 1.97 | 0.05 | 4.37 |
| **age_[60-70)** | 1.19 | 3.29 | 0.58 | 0.06 | 2.32 | 1.06 | 10.23 | 0.00 | 2.06 | 0.04 | 4.66 |
| **age_[70-80)** | 1.15 | 3.15 | 0.58 | 0.01 | 2.28 | 1.01 | 9.78 | 0.00 | 1.98 | 0.05 | 4.39 |
| **age_[80-90)** | 1.12 | 3.08 | 0.58 | -0.01 | 2.26 | 0.99 | 9.56 | 0.00 | 1.94 | 0.05 | 4.26 |
| **age_[90-100)** | 1.08 | 2.96 | 0.58 | -0.06 | 2.23 | 0.94 | 9.27 | 0.00 | 1.86 | 0.06 | 3.99 |
| **insulin_Steady** | -0.14 | 0.87 | 0.03 | -0.20 | -0.08 | 0.82 | 0.92 | 0.00 | -4.72 | <0.005 | 18.72 |
| **insulin_Up** | -0.14 | 0.87 | 0.04 | -0.21 | -0.07 | 0.81 | 0.93 | 0.00 | -3.97 | <0.005 | 13.76 |
| **med_group_Moderate** | -0.34 | 0.71 | 0.04 | -0.42 | -0.27 | 0.66 | 0.76 | 0.00 | -9.29 | <0.005 | 65.81 |
| **med_group_High** | -0.90 | 0.41 | 0.04 | -0.98 | -0.82 | 0.38 | 0.44 | 0.00 | -21.70 | <0.005 | 344.38 |
| **med_group_Very High** | -1.47 | 0.23 | 0.10 | -1.66 | -1.28 | 0.19 | 0.28 | 0.00 | -15.36 | <0.005 | 174.36 |

| | |
|---|---|
| **Concordance** | 0.63 |
| **Partial AIC** | 130598.77 |
| **log-likelihood ratio test** | 836.15 on 14 df |
| **-log2(p) of ll-ratio test** | 560.38 |

## The cox model assumptions

In [28]:
```python
# To test the cox model assumptions
cph.fit(data_encoded, duration_col='duration', event_col='event')
cph.check_assumptions(data_encoded, p_value_threshold=0.05)
```

```
The ``p_value_threshold`` is set at 0.05. Even under the null hypothesis of no violations, some
covariates will be below the threshold by chance. This is compounded when there are many covariates.
Similarly, when there are lots of observations, even minor deviances from the proportional hazard
assumption will be flagged.

With that in mind, it's best to use a combination of statistical tests and visual tests to determine
the most serious violations. Produce visual plots using ``check_assumptions(..., show_plots=True)``
and looking for non-constant lines. See link [A] below for a full example.
```

| | |
|---|---|
| **null_distribution** | chi squared |
| **degrees_of_freedom** | 1 |
| **model** | <lifelines.CoxPHFitter: fitted with 54383 tota... |
| **test_name** | proportional_hazard_test |

| | | test_statistic | p | -log2(p) |
|---|---|---|---|---|
| **age_[10-20)** | **km** | 0.42 | 0.52 | 0.95 |
| | **rank** | 0.27 | 0.60 | 0.74 |
| **age_[20-30)** | **km** | 0.39 | 0.53 | 0.90 |
| | **rank** | 0.29 | 0.59 | 0.76 |
| **age_[30-40)** | **km** | 0.06 | 0.80 | 0.32 |
| | **rank** | 0.02 | 0.89 | 0.16 |
| **age_[40-50)** | **km** | 0.09 | 0.77 | 0.38 |
| | **rank** | 0.03 | 0.86 | 0.22 |
| **age_[50-60)** | **km** | 0.04 | 0.84 | 0.26 |
| | **rank** | 0.00 | 0.96 | 0.07 |
| **age_[60-70)** | **km** | 0.07 | 0.80 | 0.33 |
| | **rank** | 0.01 | 0.94 | 0.09 |
| **age_[70-80)** | **km** | 0.03 | 0.87 | 0.20 |

| | | test_statistic | p | -log2(p) |
|---|---|---|---|---|
| | rank | 0.00 | 1.00 | 0.01 |
| age_[80-90) | km | 0.02 | 0.89 | 0.17 |
| | rank | 0.02 | 0.89 | 0.18 |
| age_[90-100) | km | 0.08 | 0.77 | 0.37 |
| | rank | 0.01 | 0.94 | 0.08 |
| insulin_Steady | km | 0.13 | 0.72 | 0.48 |
| | rank | 0.28 | 0.59 | 0.75 |
| insulin_Up | km | 0.13 | 0.72 | 0.48 |
| | rank | 0.57 | 0.45 | 1.15 |
| med_group_High | km | 86.85 | <0.005 | 66.21 |
| | rank | 128.33 | <0.005 | 96.41 |
| med_group_Moderate | km | 5.66 | 0.02 | 5.84 |
| | rank | 17.48 | <0.005 | 15.07 |
| med_group_Very High | km | 125.86 | <0.005 | 94.62 |
| | rank | 80.55 | <0.005 | 61.61 |

```
1. Variable 'med_group_Moderate' failed the non-proportional test: p-value is <5e-05.

   Advice: with so few unique values (only 2), you can include `strata=['med_group_Moderate', ...]`
in the call in `.fit`. See documentation in link [E] below.

2. Variable 'med_group_High' failed the non-proportional test: p-value is <5e-05.

   Advice: with so few unique values (only 2), you can include `strata=['med_group_High', ...]` in
the call in `.fit`. See documentation in link [E] below.

3. Variable 'med_group_Very High' failed the non-proportional test: p-value is <5e-05.

   Advice: with so few unique values (only 2), you can include `strata=['med_group_Very High', ...]`
in the call in `.fit`. See documentation in link [E] below.

---
[A]  https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html
[B]  https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html#Bin-variable-and-stratify-on-it
[C]  https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html#Introduce-time-varying-covariates
[D]  https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html#Modify-the-functional-form
[E]  https://lifelines.readthedocs.io/en/latest/jupyter_notebooks/Proportional%20hazard%20assumption.html#Stratification
```

Out [28]:
```
[]
```

### Cox Proportional Hazards Model with Stratification

In [29]:
```python
# To group 'med_group' by num_medications
def group_meds(n):
    if n <= 5:
        return 'Low'
    elif n <= 10:
        return 'Moderate'
    elif n <= 15:
        return 'High'
    else:
        return 'Very High'
data['med_group'] = data['num_medications'].apply(group_meds)
```

In [30]:
```python
# To encode age and insulin (but NOT med_group)
data_model =  pd.get_dummies(data[['age', 'insulin',  'duration', 'event']],
                            columns=['age', 'insulin'], drop_first=True)
```

In [31]:
```python
# To add unencoded med_group from original data
data_model['med_group'] = data['med_group']
```

In [32]:
```python
# To know the unique variables in med_group
print(data_model['med_group'].unique())
```

```
['Very High' 'Moderate' 'High' 'Low']
```

In [33]:
```python
# To fit stratified Cox model
cph = CoxPHFitter()
cph.fit(data_model, duration_col='duration', event_col='event', strata=['med_group'])
cph.print_summary()
```

| | |
|---|---|
| **model** | lifelines.CoxPHFitter |
| **duration col** | 'duration' |
| **event col** | 'event' |
| **strata** | med_group |
| **baseline estimation** | breslow |
| **number of observations** | 54383 |
| **number of events observed** | 6601 |
| **partial log-likelihood** | -59280.96 |
| **time fit was run** | 2025-05-26 11:10:25 UTC |

| | coef | exp(coef) | se(coef) | coef lower 95% | coef upper 95% | exp(coef) lower 95% | exp(coef) upper 95% | cmp to | z | p | -log2(p) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| age_[10-20) | 0.84 | 2.31 | 0.60 | -0.34 | 2.02 | 0.71 | 7.51 | 0.00 | 1.39 | 0.16 | 2.61 |
| age_[20-30) | 1.79 | 6.02 | 0.58 | 0.65 | 2.94 | 1.92 | 18.86 | 0.00 | 3.08 | <0.005 | 8.91 |
| age_[30-40) | 1.49 | 4.46 | 0.58 | 0.35 | 2.63 | 1.43 | 13.94 | 0.00 | 2.57 | 0.01 | 6.62 |
| age_[40-50) | 1.41 | 4.10 | 0.58 | 0.27 | 2.55 | 1.32 | 12.79 | 0.00 | 2.43 | 0.01 | 6.06 |
| age_[50-60) | 1.27 | 3.56 | 0.58 | 0.13 | 2.41 | 1.14 | 11.08 | 0.00 | 2.19 | 0.03 | 5.12 |
| age_[60-70) | 1.33 | 3.77 | 0.58 | 0.19 | 2.46 | 1.21 | 11.75 | 0.00 | 2.29 | 0.02 | 5.50 |
| age_[70-80) | 1.30 | 3.67 | 0.58 | 0.16 | 2.44 | 1.18 | 11.42 | 0.00 | 2.24 | 0.02 | 5.32 |
| age_[80-90) | 1.30 | 3.65 | 0.58 | 0.16 | 2.43 | 1.17 | 11.38 | 0.00 | 2.23 | 0.03 | 5.30 |
| age_[90-100) | 1.29 | 3.64 | 0.58 | 0.15 | 2.44 | 1.16 | 11.43 | 0.00 | 2.21 | 0.03 | 5.22 |

|  | coef | exp(coef) | se(coef) | coef lower 95% | coef upper 95% | exp(coef) lower 95% | exp(coef) upper 95% | cmp to | z | p | -log2(p) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **insulin_Steady** | -0.12 | 0.89 | 0.03 | -0.17 | -0.06 | 0.84 | 0.94 | 0.00 | -3.86 | <0.005 | 13.10 |
| **insulin_Up** | -0.16 | 0.85 | 0.04 | -0.23 | -0.09 | 0.80 | 0.92 | 0.00 | -4.43 | <0.005 | 16.68 |

|  |  |
|---|---|
| **Concordance** | 0.53 |
| **Partial AIC** | 118583.91 |
| **log-likelihood ratio test** | 97.94 on 11 df |
| **-log2(p) of ll-ratio test** | 50.96 |

In [34]:
```
# To check PH assumptions
cph.check_assumptions(data_model, p_value_threshold=0.05)
```

```
Proportional hazard assumption looks okay.
```

Out [34]:
```
[]
```

In [35]:
```
# To fit stratified Cox Model
cph.print_summary()
```

|  |  |
|---|---|
| **model** | lifelines.CoxPHFitter |
| **duration col** | 'duration' |
| **event col** | 'event' |
| **strata** | med_group |
| **baseline estimation** | breslow |
| **number of observations** | 54383 |
| **number of events observed** | 6601 |
| **partial log-likelihood** | -59280.96 |
| **time fit was run** | 2025-05-26 11:10:25 UTC |

|  | coef | exp(coef) | se(coef) | coef lower 95% | coef upper 95% | exp(coef) lower 95% | exp(coef) upper 95% | cmp to | z | p | -log2(p) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **age_[10-20)** | 0.84 | 2.31 | 0.60 | -0.34 | 2.02 | 0.71 | 7.51 | 0.00 | 1.39 | 0.16 | 2.61 |
| **age_[20-30)** | 1.79 | 6.02 | 0.58 | 0.65 | 2.94 | 1.92 | 18.86 | 0.00 | 3.08 | <0.005 | 8.91 |
| **age_[30-40)** | 1.49 | 4.46 | 0.58 | 0.35 | 2.63 | 1.43 | 13.94 | 0.00 | 2.57 | 0.01 | 6.62 |
| **age_[40-50)** | 1.41 | 4.10 | 0.58 | 0.27 | 2.55 | 1.32 | 12.79 | 0.00 | 2.43 | 0.01 | 6.06 |
| **age_[50-60)** | 1.27 | 3.56 | 0.58 | 0.13 | 2.41 | 1.14 | 11.08 | 0.00 | 2.19 | 0.03 | 5.12 |
| **age_[60-70)** | 1.33 | 3.77 | 0.58 | 0.19 | 2.46 | 1.21 | 11.75 | 0.00 | 2.29 | 0.02 | 5.50 |
| **age_[70-80)** | 1.30 | 3.67 | 0.58 | 0.16 | 2.44 | 1.18 | 11.42 | 0.00 | 2.24 | 0.02 | 5.32 |
| **age_[80-90)** | 1.30 | 3.65 | 0.58 | 0.16 | 2.43 | 1.17 | 11.38 | 0.00 | 2.23 | 0.03 | 5.30 |
| **age_[90-100)** | 1.29 | 3.64 | 0.58 | 0.15 | 2.44 | 1.16 | 11.43 | 0.00 | 2.21 | 0.03 | 5.22 |
| **insulin_Steady** | -0.12 | 0.89 | 0.03 | -0.17 | -0.06 | 0.84 | 0.94 | 0.00 | -3.86 | <0.005 | 13.10 |
| **insulin_Up** | -0.16 | 0.85 | 0.04 | -0.23 | -0.09 | 0.80 | 0.92 | 0.00 | -4.43 | <0.005 | 16.68 |

|  |  |
|---|---|
| **Concordance** | 0.53 |
| **Partial AIC** | 118583.91 |
| **log-likelihood ratio test** | 97.94 on 11 df |
| **-log2(p) of ll-ratio test** | 50.96 |

In [ ]:
```
```