Analysis on diabete1

In [74]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

In [65]:
```
Type1 = pd.read_csv("C:/Users/Shu/Downloads/type1_diabetes.csv")
Type1.head()

age_order = ['<10', '10-20', '20-30', '30-40', '40+']

# Convert Age_Group column to categorical type with the correct order
Type1['Age_Group'] = pd.Categorical(Type1['Age_Group'], categories=age_order, or
```

Variable description

- HYPEV - Ever been told you have hypertension
- CHLEV - Ever been told you had high cholesterol
- CHDEV - Ever been told you had coronary heart disease
- ANGEV - Ever been told you had angina pectoris
- MIEV - Ever been told you had a heart attack
- HRTEV - Ever been told you had a heart condition/disease
- STREV - Ever been told you had a stroke
- EPHEV - Ever been told you had emphysema
- COPDEV - Ever been told you had COPD (Chronic Obstructive Pulmonary Disease)
- AASMEV - Ever been told you had asthma
- ULCEV - Ever been told you have an ulcer
- ULCCOLEV - Ever been told you had Crohn's disease or ulcerative colitis
- CANEV - Ever told by a doctor you had cancer
- SINYR - Told that you had sinusitis, past 12 months
- CBRCHYR - Told you had chronic bronchitis, past 12 months
- KIDWKYR - Told you had weak/failing kidneys, past 12 months
- LIVYR - Told you had a liver condition, past 12 months
- ARTH1 - Ever been told you had arthritis
- VIM_GLEV - Ever been told you had glaucoma
- FLA1AR - Any functional limitation, all conditions

Feature Correlation Analysis

Chi-sqaure test for features

Why we use Chi-square?

Determines whether the distribution of one categorical variable (e.g., the age group) differs depending on another categorical variable (e.g., the presence of a complication). Specifically, it answers the question: "Is the occurrence of a complication related to the age of diabetes diagnosis?"

In [66]:
```
from scipy.stats import chi2_contingency

for complication in ['HYPEV', 'CHLEV', 'CHDEV', 'ANGEV', 'MIEV', 'HRTEV', 'STREV
    contingency_table = pd.crosstab(Type1[complication], Type1['Age_Group'])
    chi2, p, dof, ex = chi2_contingency(contingency_table)
```

In [67]:
```
significant_complications = {
    'Complication': ['HYPEV', 'CHLEV', 'CHDEV', 'COPDEV', 'AASMEV', 'CANEV', 'AR
    'P-value': [3.857202605167316e-11, 0.029468324837065636, 0.04558035159381288
}

nonsignificant_complications = {
    'Complication': ['ANGEV', 'MIEV', 'HRTEV', 'STREV', 'EPHEV', 'ULCEV', 'ULCCO
    'P-value': [0.34678963052251033, 0.3117310205810393, 0.6430184288990237, 0.2
}

significant_df = pd.DataFrame(significant_complications)
nonsignificant_df = pd.DataFrame(nonsignificant_complications)

print("Features that are significantly correalted:\n", significant_df)
print("Features that are not significantly correalted:\n", nonsignificant_df)
```

```
Features that are significantly correalted:
   Complication       P-value
0        HYPEV  3.857203e-11
1        CHLEV  2.946832e-02
2        CHDEV  4.558035e-02
3       COPDEV  4.324934e-03
4       AASMEV  1.136035e-02
5        CANEV  3.995791e-02
6        ARTH1  3.044532e-04
7       FLA1AR  5.769036e-05
Features that are not significantly correalted:
    Complication   P-value
0         ANGEV  0.346790
1          MIEV  0.311731
2         HRTEV  0.643018
3         STREV  0.279781
4         EPHEV  0.554722
5         ULCEV  0.420774
6      ULCCOLEV  0.559138
7         SINYR  0.107738
8       CBRCHYR  0.817249
9       KIDWKYR  0.802256
10        LIVYR  0.549789
11     VIM_GLEV  0.369243
```

Correlation Analysis

Create plots to show complications and diagnoses

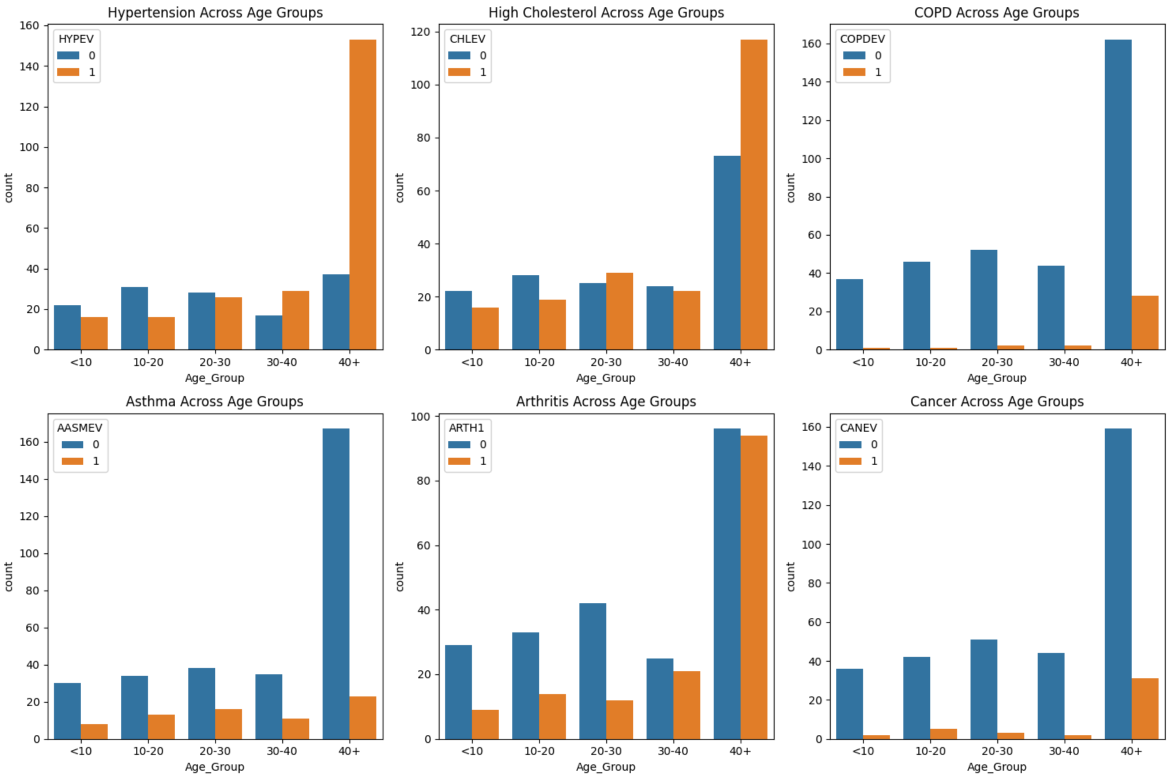
In [82]:
```
# 0 as No and 1 as Yes
complications = ['HYPEV', 'CHLEV', 'COPDEV', 'AASMEV', 'ARTH1', 'CANEV']
titles = ['Hypertension', 'High Cholesterol', 'COPD', 'Asthma', 'Arthritis', 'Ca

n_plots = len(complications)
n_cols = 3
n_rows = (n_plots + n_cols - 1) // n_cols
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))

axes = axes.flatten()
for i, complication in enumerate(complications):
    sns.countplot(x='Age_Group', hue=complication, data=Type1, ax=axes[i])
    axes[i].set_title(f'{titles[i]} Across Age Groups')

for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

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



Using logistic regression and calculate odds ratio for variables

In [75]:
```
X = pd.get_dummies(Type1['Age_Group'], drop_first=True)

y = Type1['HYPEV']
X = sm.add_constant(X)
model = sm.Logit(y, X)
result = model.fit()

print(result.summary())
print("Odds Ratios:")
print(np.exp(result.params))
```

```
Optimization terminated successfully.
         Current function value: 0.579663
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                  HYPEV   No. Observations:                  375
Model:                          Logit   Df Residuals:                      370
Method:                           MLE   Df Model:                            4
Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                  0.1129
Time:                        19:21:28   Log-Likelihood:                -217.37
converged:                       True   LL-Null:                       -245.03
Covariance Type:            nonrobust   LLR p-value:                 2.789e-11
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3185      0.329     -0.969      0.332      -0.962       0.326
10-20         -0.3429      0.450     -0.762      0.446      -1.225       0.540
20-30          0.2443      0.427      0.573      0.567      -0.592       1.081
30-40          0.8525      0.449      1.900      0.057      -0.027       1.732
40+            1.7380      0.376      4.620      0.000       1.001       2.475
==============================================================================
Odds Ratios:
const    0.727273
10-20    0.709677
20-30    1.276786
30-40    2.345588
40+      5.685811
dtype: float64
```

40+ Years Since Diagnosis:

- People who have had Type I diabetes for 40 or more years are 5.69 times more likely to have hypertension compared to those who have had diabetes for less than 10 years. This result is highly significant ($p < 0.05$).

30-40 Years Since Diagnosis:

- People who have had Type I diabetes for 30-40 years are 2.35 times more likely to have hypertension compared to those who have had it for less than 10 years. This result is marginally significant ($p \approx 0.057$).

Conclusion: People who have had Type I diabetes for a long time, particularly those who have had it for 40+ years, are much more likely to develop hypertension compared to those diagnosed more recently (less than 10 years).

Do the same analysis for the rest of significant features

In [76]:
```
complications = ['CHLEV', 'COPDEV', 'AASMEV', 'ARTH1', 'CANEV']
results = {}

for complication in complications:
    y = Type1[complication]
    X = pd.get_dummies(Type1['Age_Group'], drop_first=True)
    X = sm.add_constant(X)

    model = sm.Logit(y, X)
    result = model.fit()
```

```
    odds_ratios = np.exp(result.params)
    results[complication] = {
        'summary': result.summary(),
        'odds_ratios': odds_ratios
    }

results
```

```
Optimization terminated successfully.
         Current function value: 0.675345
         Iterations 4
Optimization terminated successfully.
         Current function value: 0.281832
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.468014
         Iterations 6
Optimization terminated successfully.
         Current function value: 0.643811
         Iterations 5
Optimization terminated successfully.
         Current function value: 0.341603
         Iterations 7
```

```
Out[76]: {'CHLEV': {'summary': <class 'statsmodels.iolib.summary.Summary'>
  """
                             Logit Regression Results
  ==============================================================================
=
  Dep. Variable:                  CHLEV   No. Observations:                   37
5
  Model:                          Logit   Df Residuals:                       37
0
  Method:                           MLE   Df Model:
4
  Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                  0.0208
5
  Time:                        20:55:42   Log-Likelihood:                 -253.2
5
  converged:                       True   LL-Null:                        -258.6
5
  Covariance Type:            nonrobust   LLR p-value:                    0.0290
8
  ==============================================================================
=
                   coef    std err          z      P>|z|      [0.025      0.97
5]
  ------------------------------------------------------------------------------
-
  const         -0.3185      0.329     -0.969      0.332      -0.962       0.32
6
  10-20         -0.0693      0.443     -0.156      0.876      -0.938       0.79
9
  20-30          0.4669      0.427      1.093      0.274      -0.370       1.30
4
  30-40          0.2314      0.442      0.524      0.600      -0.634       1.09
7
  40+            0.7902      0.361      2.190      0.029       0.083       1.49
7
  ==============================================================================
=
  """,
  'odds_ratios': const    0.727273
  10-20    0.933036
  20-30    1.595000
  30-40    1.260417
  40+      2.203767
  dtype: float64},
 'COPDEV': {'summary': <class 'statsmodels.iolib.summary.Summary'>
  """
                             Logit Regression Results
  ==============================================================================
=
  Dep. Variable:                 COPDEV   No. Observations:                   37
5
  Model:                          Logit   Df Residuals:                       37
0
  Method:                           MLE   Df Model:
4
  Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                  0.0731
6
  Time:                        20:55:42   Log-Likelihood:                 -105.6
9
  converged:                       True   LL-Null:                        -114.0
```

```
3
  Covariance Type:            nonrobust   LLR p-value:                  0.00222
6
  ==============================================================================
=
                   coef    std err          z      P>|z|      [0.025      0.97
5]
  ------------------------------------------------------------------------------
-
  const         -3.6109      1.013     -3.563      0.000      -5.597      -1.62
5
  10-20         -0.2177      1.431     -0.152      0.879      -3.023       2.58
8
  20-30          0.3528      1.243      0.284      0.777      -2.084       2.79
0
  30-40          0.5199      1.245      0.418      0.676      -1.920       2.96
0
  40+            1.8555      1.034      1.795      0.073      -0.171       3.88
2
  ==============================================================================
=
  """,
  'odds_ratios': const    0.027027
  10-20    0.804348
  20-30    1.423077
  30-40    1.681818
  40+      6.395062
  dtype: float64},
 'AASMEV': {'summary': <class 'statsmodels.iolib.summary.Summary'>
  """
                             Logit Regression Results
  ==============================================================================
=
  Dep. Variable:                 AASMEV   No. Observations:                  37
5
  Model:                          Logit   Df Residuals:                      37
0
  Method:                           MLE   Df Model:
4
  Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                 0.0355
3
  Time:                        20:55:42   Log-Likelihood:                -175.5
1
  converged:                       True   LL-Null:                       -181.9
7
  Covariance Type:            nonrobust   LLR p-value:                   0.0116
2
  ==============================================================================
=
                   coef    std err          z      P>|z|      [0.025      0.97
5]
  ------------------------------------------------------------------------------
-
  const         -1.3218      0.398     -3.322      0.001      -2.102      -0.54
2
  10-20          0.3603      0.514      0.700      0.484      -0.648       1.36
9
  20-30          0.4568      0.497      0.919      0.358      -0.518       1.43
1
  30-40          0.1643      0.527      0.312      0.755      -0.869       1.19
```

```
7
  40+           -0.6607      0.456     -1.449      0.147      -1.554       0.23
3
  ==============================================================================
=
  """,
  'odds_ratios': const    0.266667
  10-20    1.433824
  20-30    1.578947
  30-40    1.178571
  40+      0.516467
  dtype: float64},
 'ARTH1': {'summary': <class 'statsmodels.iolib.summary.Summary'>
  """
                             Logit Regression Results
  ==============================================================================
=
  Dep. Variable:                  ARTH1   No. Observations:                  37
5
  Model:                          Logit   Df Residuals:                      37
0
  Method:                           MLE   Df Model:
4
  Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                 0.0433
9
  Time:                        20:55:42   Log-Likelihood:                -241.4
3
  converged:                       True   LL-Null:                       -252.3
8
  Covariance Type:            nonrobust   LLR p-value:                 0.000209
8
  ==============================================================================
=
                   coef    std err          z      P>|z|      [0.025      0.97
5]
  ------------------------------------------------------------------------------
-
  const         -1.1701      0.382     -3.066      0.002      -1.918      -0.42
2
  10-20          0.3126      0.497      0.629      0.530      -0.662       1.28
7
  20-30         -0.0827      0.503     -0.164      0.869      -1.068       0.90
3
  30-40          0.9957      0.483      2.062      0.039       0.049       1.94
2
  40+            1.1490      0.408      2.815      0.005       0.349       1.94
9
  ==============================================================================
=
  """,
  'odds_ratios': const    0.310345
  10-20    1.367003
  20-30    0.920635
  30-40    2.706667
  40+      3.155093
  dtype: float64},
 'CANEV': {'summary': <class 'statsmodels.iolib.summary.Summary'>
  """
                             Logit Regression Results
  ==============================================================================
```

```
=
  Dep. Variable:                  CANEV   No. Observations:                  37
5
  Model:                          Logit   Df Residuals:                      37
0
  Method:                           MLE   Df Model:                            
4
  Date:                Mon, 21 Oct 2024   Pseudo R-squ.:                 0.0408
8
  Time:                        20:55:42   Log-Likelihood:                -128.1
0
  converged:                       True   LL-Null:                       -133.5
6
  Covariance Type:            nonrobust   LLR p-value:                   0.0274
9
  ==============================================================================
=
                   coef    std err          z      P>|z|      [0.025      0.97
5]
  ------------------------------------------------------------------------------
-
  const         -2.8904      0.726     -3.979      0.000      -4.314      -1.46
6
  10-20          0.7621      0.867      0.879      0.379      -0.937       2.46
1
  20-30          0.0572      0.938      0.061      0.951      -1.782       1.89
7
  30-40         -0.2007      1.025     -0.196      0.845      -2.210       1.80
8
  40+            1.2555      0.753      1.668      0.095      -0.220       2.73
0
  ==============================================================================
=
  """,
  'odds_ratios': const    0.055556
  10-20    2.142857
  20-30    1.058824
  30-40    0.818182
  40+      3.509434
  dtype: float64}}
```

Why some features in chi square are significant, but in logistic they are not any more?

- There might be some multicolinearity within the data
- Logistic regression can account for interaction effects between predictors. If a variable interacts with other predictors (e.g., how two age groups combined affect the likelihood of developing a complication), logistic regression can capture this, whereas a Chi-Square test looks only at the variables in isolation.

Conclusion for features:

- CHLEV (High Cholesterol): There is a significant association with having Type I diabetes for 40+ years, which increases the likelihood of developing high cholesterol.
- COPDEV (COPD): No significant relationship with any of the age groups.
- AASMEV (Asthma): No significant relationship with any of the age groups.

- ARTH1 (Arthritis): There is a significant association for both the 30-40 years and 40+ years groups, indicating a higher likelihood of developing arthritis after living with Type I diabetes for 30+ years.
- CANEV (Cancer): No significant relationship with any of the age groups.

The logistic regression and the graph align with each other and give us the conclusion that people who have Type 1 diabetes, particularly having diabetes over 30+ years are more likely to complications Hypertension, High Cholesterol, and Arthritis.

PCA analysis for variables

In [70]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

X_numeric = Type1.drop(columns=['Age_Group'])

# Also, store the target variable (Age_Group) for later use
y = Type1['Age_Group']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_numeric)
pca = PCA(n_components=10)  # Change number of components as needed
X_pca = pca.fit_transform(X_scaled)

# Print explained variance ratio for each principal component
print(pca.explained_variance_ratio_)

# Print cumulative explained variance
print(pca.explained_variance_ratio_.cumsum())
model = LogisticRegression()
model.fit(X_pca, y)
```

```
[0.17020286 0.08354231 0.07097845 0.06242344 0.06180468 0.05464603
 0.05183745 0.04893914 0.04768258 0.044672  ]
[0.17020286 0.25374517 0.32472362 0.38714706 0.44895175 0.50359778
 0.55543523 0.60437437 0.65205695 0.69672896]
```

Out[70]:
```
LogisticRegression()
```

In [10]:
```
le = LabelEncoder()
Type1['Age_Group_encoded'] = le.fit_transform(Type1['Age_Group'])
X = Type1.drop(columns=['Age_Group', 'Age_Group_encoded'])
y = Type1['Age_Group_encoded']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_
```

it is not realistic to talk about the all of the features since we only have 340 data points while there are 800 features which will make contingency table very small and decrease the predict power

Logistic regression

In [33]:
```
# Logistic Regression Model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
```

```
y_pred_log_reg = log_reg.predict(X_test)

# Evaluation for Logistic Regression
log_reg_acc = accuracy_score(y_test, y_pred_log_reg)
log_reg_report = classification_report(y_test, y_pred_log_reg, target_names=le.c
print("Logistic Regression Accuracy:", log_reg_acc)
print("Logistic Regression Report:\n", log_reg_report)
```

```
Logistic Regression Accuracy: 0.46017699115044247
Logistic Regression Report:
               precision    recall  f1-score   support

       10-20       0.20      0.06      0.09        18
       20-30       0.29      0.10      0.14        21
       30-40       0.00      0.00      0.00        13
         40+       0.49      0.98      0.66        50
         <10       0.00      0.00      0.00        11

    accuracy                           0.46       113
   macro avg       0.20      0.23      0.18       113
weighted avg       0.30      0.46      0.33       113
```

Tried: Interaction term: 44% Scalar: 44% PCA, regularization also does not work to improve performance

Decision tree

In [14]:
```
# Decision Tree Model
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, y_train)
y_pred_decision_tree = decision_tree.predict(X_test)

# Evaluation for Decision Tree
decision_tree_acc = accuracy_score(y_test, y_pred_decision_tree)
decision_tree_report = classification_report(y_test, y_pred_decision_tree, targe
print("Decision Tree Accuracy:", decision_tree_acc)
print("Decision Tree Report:\n", decision_tree_report)
```

```
Decision Tree Accuracy: 0.40707964601769914
Decision Tree Report:
               precision    recall  f1-score   support

       10-20       0.21      0.22      0.22        18
       20-30       0.14      0.05      0.07        21
       30-40       0.25      0.31      0.28        13
         40+       0.67      0.68      0.67        50
         <10       0.15      0.27      0.19        11

    accuracy                           0.41       113
   macro avg       0.28      0.31      0.29       113
weighted avg       0.40      0.41      0.40       113
```

Random Forest Classifier

In [15]:
```
# Random Forest Model
random_forest = RandomForestClassifier()
random_forest.fit(X_train, y_train)
y_pred_random_forest = random_forest.predict(X_test)

# Evaluation for Random Forest
random_forest_acc = accuracy_score(y_test, y_pred_random_forest)
random_forest_report = classification_report(y_test, y_pred_random_forest, targe
print("Random Forest Accuracy:", random_forest_acc)
print("Random Forest Report:\n", random_forest_report)
```

```
Random Forest Accuracy: 0.415929203539823
Random Forest Report:
               precision    recall  f1-score   support

       10-20       0.22      0.11      0.15        18
       20-30       0.10      0.05      0.06        21
       30-40       0.00      0.00      0.00        13
         40+       0.56      0.80      0.66        50
         <10       0.25      0.36      0.30        11

    accuracy                           0.42       113
   macro avg       0.23      0.26      0.23       113
weighted avg       0.33      0.42      0.36       113
```

Bias

In [ ]: