Generalized Linear Models (GLMs)

By Adrian Chavez-Loya

Background

In 2014, the analytics website, FiveThirtyEight, published an analysis of the gender disparity in appearances in comic books at the two large studios, Marvel and DC. You're working for a small upstart comic book company and you want to work to understand what characteristics are associated with a reduction in appearances and thereby learn where gaps in representation can be filled by your firm. Using the data collected by FiveThirtyEight from the Marvel and DC wiki pages, you will fit an appropriate GLM to predict the number of appearances given this data.

Relevant Datasets

- `marvel-wikia-data.csv`
- `dc-wikia-data.csv`

Article: https://fivethirtyeight.com/features/women-in-comic-books/

Aggregated Data: https://github.com/fivethirtyeight/data/tree/master/comic-characters

Data Sources:

- http://dc.wikia.com/wiki/Main_Page
- http://marvel.wikia.com/Main_Page

Task 1: Concatenate the Marvel and DC datasets into a single dataset.

The function `pd.concat` will be useful for this. You should also create a new column in each dataset before concatenating that represents the studio for each. This will allow us to distinguish between the impact of one studio vs the other.

In [ ]:
```
# Imported data and data types to perform data cleaining
import pandas as pd
import numpy as np
marvel_data = pd.read_csv('marvel-wikia-data.csv')
dc_data = pd.read_csv('dc-wikia-data.csv')
print("Marvel Data Types and Head:")
print(marvel_data.dtypes)
print(marvel_data.head())
print("\nDC Data Types and Head:")
print(dc_data.dtypes)
print(dc_data.head())
```

```
Marvel Data Types and Head:
page_id              int64
name                object
urlslug             object
ID                  object
ALIGN               object
EYE                 object
HAIR                object
SEX                 object
GSM                 object
ALIVE               object
APPEARANCES        float64
FIRST APPEARANCE    object
Year               float64
dtype: object
   page_id                                 name  \
0     1678           Spider-Man (Peter Parker)
1     7139     Captain America (Steven Rogers)
2    64786  Wolverine (James \"Logan\" Howlett)
3     1868    Iron Man (Anthony \"Tony\" Stark)
4     2460                 Thor (Thor Odinson)

                                 urlslug                ID  \
0              \/Spider-Man_(Peter_Parker)   Secret Identity
1        \/Captain_America_(Steven_Rogers)   Public Identity
2  \/Wolverine_(James_%22Logan%22_Howlett)   Public Identity
3    \/Iron_Man_(Anthony_%22Tony%22_Stark)   Public Identity
4                \/Thor_(Thor_Odinson)  No Dual Identity

                ALIGN         EYE        HAIR              SEX  G
SM  \
0     Good Characters  Hazel Eyes  Brown Hair  Male Characters  N
aN
1     Good Characters   Blue Eyes  White Hair  Male Characters  N
aN
2  Neutral Characters   Blue Eyes  Black Hair  Male Characters  N
aN
3     Good Characters   Blue Eyes  Black Hair  Male Characters  N
aN
4     Good Characters   Blue Eyes  Blond Hair  Male Characters  N
aN

               ALIVE  APPEARANCES FIRST APPEARANCE    Year
0  Living Characters       4043.0           Aug-62  1962.0
1  Living Characters       3360.0           Mar-41  1941.0
2  Living Characters       3061.0           Oct-74  1974.0
3  Living Characters       2961.0           Mar-63  1963.0
4  Living Characters       2258.0           Nov-50  1950.0

DC Data Types and Head:
```

```
page_id                 int64
name                   object
urlslug                object
ID                     object
ALIGN                  object
EYE                    object
HAIR                   object
SEX                    object
GSM                    object
ALIVE                  object
APPEARANCES           float64
FIRST APPEARANCE       object
YEAR                  float64
dtype: object
   page_id                         name
urlslug  \
0     1422         Batman (Bruce Wayne)         \/wiki\/Batman_(B
ruce_Wayne)
1    23387        Superman (Clark Kent)        \/wiki\/Superman_
(Clark_Kent)
2     1458   Green Lantern (Hal Jordan)   \/wiki\/Green_Lantern_
(Hal_Jordan)
3     1659     James Gordon (New Earth)     \/wiki\/James_Gordon_
(New_Earth)
4     1576  Richard Grayson (New Earth)  \/wiki\/Richard_Grayson_
(New_Earth)

                ID            ALIGN         EYE        HAIR
SEX  \
0  Secret Identity  Good Characters   Blue Eyes  Black Hair  Male
Characters
1  Secret Identity  Good Characters   Blue Eyes  Black Hair  Male
Characters
2  Secret Identity  Good Characters  Brown Eyes  Brown Hair  Male
Characters
3  Public Identity  Good Characters  Brown Eyes  White Hair  Male
Characters
4  Secret Identity  Good Characters   Blue Eyes  Black Hair  Male
Characters

   GSM              ALIVE  APPEARANCES FIRST APPEARANCE    YEAR
0  NaN  Living Characters       3093.0        1939, May  1939.0
1  NaN  Living Characters       2496.0    1986, October  1986.0
2  NaN  Living Characters       1565.0    1959, October  1959.0
3  NaN  Living Characters       1316.0   1987, February  1987.0
4  NaN  Living Characters       1237.0      1940, April  1940.0
```

In [ ]:

```
marvel_data['Studio'] = 'Marvel' # Labeled by studio
dc_data['Studio'] = 'DC'
```

```
# Combined 2 data sets
combined_data = pd.concat([marvel_data, dc_data], ignore_index=Tr
print("Combined Data Types and Head:")
print(combined_data.dtypes)
print(combined_data.head())
```

```
Combined Data Types and Head:
page_id               int64
name                 object
urlslug              object
ID                   object
ALIGN                object
EYE                  object
HAIR                 object
SEX                  object
GSM                  object
ALIVE                object
APPEARANCES         float64
FIRST APPEARANCE     object
Year                float64
Studio               object
YEAR                float64
dtype: object
   page_id                                 name  \
0     1678           Spider-Man (Peter Parker)
1     7139     Captain America (Steven Rogers)
2    64786  Wolverine (James \"Logan\" Howlett)
3     1868    Iron Man (Anthony \"Tony\" Stark)
4     2460                 Thor (Thor Odinson)

                                   urlslug               ID  \
0              \/Spider-Man_(Peter_Parker)   Secret Identity
1        \/Captain_America_(Steven_Rogers)   Public Identity
2  \/Wolverine_(James_%22Logan%22_Howlett)   Public Identity
3    \/Iron_Man_(Anthony_%22Tony%22_Stark)   Public Identity
4                    \/Thor_(Thor_Odinson)  No Dual Identity

                ALIGN         EYE        HAIR              SEX  G
SM  \
0     Good Characters  Hazel Eyes  Brown Hair  Male Characters  N
aN
1     Good Characters   Blue Eyes  White Hair  Male Characters  N
aN
2  Neutral Characters   Blue Eyes  Black Hair  Male Characters  N
aN
3     Good Characters   Blue Eyes  Black Hair  Male Characters  N
aN
4     Good Characters   Blue Eyes  Blond Hair  Male Characters  N
aN

               ALIVE  APPEARANCES FIRST APPEARANCE    Year  Studi
o  YEAR
0  Living Characters       4043.0           Aug-62  1962.0  Marve
l   NaN
1  Living Characters       3360.0           Mar-41  1941.0  Marve
l   NaN
```

```
2  Living Characters        3061.0            Oct-74  1974.0  Marve
l   NaN
3  Living Characters        2961.0            Mar-63  1963.0  Marve
l   NaN
4  Living Characters        2258.0            Nov-50  1950.0  Marve
l   NaN
```

In [ ]:

```
# Subsetting
relevant_columns = ['ALIGN', 'SEX', 'ALIVE', 'APPEARANCES', 'ID',
subset_data = combined_data[relevant_columns].copy()
subset_data.loc[:, 'SEX'] = subset_data['SEX'].astype(str) # Sex
subset_data.loc[:, 'SEX'].replace('nan', 'Unknown', inplace=True)
subset_data.loc[:, 'Is_Male'] = subset_data['SEX'].apply(lambda x
filtered_data = subset_data[subset_data['APPEARANCES'] > 1] # Rem
print("Filtered Data Types and Head:")
print(filtered_data.dtypes)
print(filtered_data.head())
```

```
Filtered Data Types and Head:
ALIGN              object
SEX                object
ALIVE              object
APPEARANCES       float64
ID                 object
Studio             object
Is_Male             int64
dtype: object
                ALIGN              SEX              ALIVE  APPEAR
ANCES  \
0     Good Characters  Male Characters  Living Characters       4
043.0
1     Good Characters  Male Characters  Living Characters       3
360.0
2  Neutral Characters  Male Characters  Living Characters       3
061.0
3     Good Characters  Male Characters  Living Characters       2
961.0
4     Good Characters  Male Characters  Living Characters       2
258.0

                 ID  Studio  Is_Male
0   Secret Identity  Marvel        1
1   Public Identity  Marvel        1
2   Public Identity  Marvel        1
3   Public Identity  Marvel        1
4  No Dual Identity  Marvel        1
```

Task 2: Subset the data to relevant variables and observations.

If you want to limit the number of levels, a good list of variables would be: `ALIGN`, `SEX`, `ALIVE`, `APPEARANCES`, `ID`, `Studio`. Given that this dataset includes a few `SEX` categories with very few observations, create a new binary variable for a character's `SEX` being Male or not Male. Also, remove any characters that only appear once.

In [ ]:

```
# Final data cleaning
filtered_data = filtered_data.dropna(subset=['ALIGN', 'SEX', 'ALI
filtered_data['ALIGN'] = filtered_data['ALIGN'].astype('category'
filtered_data['SEX'] = filtered_data['SEX'].astype('category')
filtered_data['ALIVE'] = filtered_data['ALIVE'].astype('category'
filtered_data['Studio'] = filtered_data['Studio'].astype('categor
print("Cleaned Data Types and Head:")
print(filtered_data.dtypes)
print(filtered_data.head())
```

```
Filtered Data Types and Head:
ALIGN           object
SEX             object
ALIVE           object
APPEARANCES    float64
ID              object
Studio          object
Is_Male          int64
dtype: object
                ALIGN              SEX               ALIVE  APPEAR
ANCES  \
0     Good Characters  Male Characters  Living Characters       4
043.0
1     Good Characters  Male Characters  Living Characters       3
360.0
2  Neutral Characters  Male Characters  Living Characters       3
061.0
3     Good Characters  Male Characters  Living Characters       2
961.0
4     Good Characters  Male Characters  Living Characters       2
258.0

                 ID  Studio  Is_Male
0   Secret Identity  Marvel        1
1   Public Identity  Marvel        1
2   Public Identity  Marvel        1
3   Public Identity  Marvel        1
4  No Dual Identity  Marvel        1
Cleaned Data Types and Head:
ALIGN          category
SEX            category
ALIVE          category
APPEARANCES     float64
ID               object
Studio         category
Is_Male           int64
dtype: object
                ALIGN              SEX               ALIVE  APPEAR
ANCES  \
0     Good Characters  Male Characters  Living Characters       4
043.0
1     Good Characters  Male Characters  Living Characters       3
360.0
2  Neutral Characters  Male Characters  Living Characters       3
061.0
3     Good Characters  Male Characters  Living Characters       2
961.0
4     Good Characters  Male Characters  Living Characters       2
258.0
```

```
                 ID  Studio  Is_Male
0   Secret Identity  Marvel        1
1   Public Identity  Marvel        1
2   Public Identity  Marvel        1
3   Public Identity  Marvel        1
4  No Dual Identity  Marvel        1
```

Task 3: Split your data into train/test and fit an appropriate GLM to the training data.

It will be up to you to determine the appropriate choice of distribution or family of the GLM. Look at residual plots and see if there are any red flags with this model.

In [ ]:

```
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
relevant_data = combined_data[['ALIGN', 'SEX', 'ALIVE', 'APPEARAN


relevant_data['SEX'] = relevant_data['SEX'].fillna('Unknown')  #
relevant_data['Is_Male'] = relevant_data['SEX'].apply(lambda x: 1
filtered_data = relevant_data[relevant_data['APPEARANCES'] > 1]
filtered_data = filtered_data.drop(columns=['SEX']) # Dropped sex
X = pd.get_dummies(filtered_data[['ALIGN', 'ALIVE', 'Is_Male', 'S
y = filtered_data['APPEARANCES']

# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_si
X_train_sm = sm.add_constant(X_train)
```

```
/var/folders/vq/l_8lvyx12cxb7kcq563rstwh0000gn/T/ipykernel_4731/4
283741262.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFram
e.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/p
andas-docs/stable/user_guide/indexing.html#returning-a-view-versu
s-a-copy
  relevant_data['SEX'] = relevant_data['SEX'].fillna('Unknown')
# Replace NaN with 'Unknown'
/var/folders/vq/l_8lvyx12cxb7kcq563rstwh0000gn/T/ipykernel_4731/4
283741262.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFram
e.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/p
andas-docs/stable/user_guide/indexing.html#returning-a-view-versu
s-a-copy
  relevant_data['Is_Male'] = relevant_data['SEX'].apply(lambda x:
1 if 'Male' in x else 0)
```

In [ ]:

```
# Convert boolean columns to integers
X_train = X_train.astype(int)
print("\nX_train Numpy Array Type after Conversion:")
print(np.asarray(X_train).dtype)
```

```
X_train Numpy Array Type after Conversion:
int64
```

In [ ]:

```
# Multicollinearity check with VIF
from statsmodels.stats.outliers_influence import variance_inflati
import pandas as pd
X_train_with_const = sm.add_constant(X_train)
vif = pd.DataFrame()
vif['Feature'] = X_train_with_const.columns
vif['VIF'] = [variance_inflation_factor(X_train_with_const.values
print(vif)
```

```
                     Feature       VIF
0                      const  9.822023
1                    Is_Male  1.015725
2       ALIGN_Good Characters  1.107689
3    ALIGN_Neutral Characters  1.091771
4   ALIGN_Reformed Criminals  1.000572
5     ALIVE_Living Characters  1.008271
6               Studio_Marvel  1.016270
```

In [ ]:

```
# Fit Poisson GLM
import statsmodels.api as sm
```

```
poisson_model = sm.GLM(y_train, X_train, family=sm.families.Poiss
print("Poisson GLM Summary:")
print(poisson_model.summary())
```

```
Poisson GLM Summary:
                 Generalized Linear Model Regression Results
==================================================================
=============
Dep. Variable:            APPEARANCES   No. Observations:
12807
Model:                            GLM   Df Residuals:
12801
Model Family:                 Poisson   Df Model:
5
Link Function:                    Log   Scale:
1.0000
Method:                          IRLS   Log-Likelihood:
-5.2508e+05
Date:                Fri, 26 Jul 2024   Deviance:
9.9967e+05
Time:                        10:52:08   Pearson chi2:
4.39e+06
No. Iterations:                     7   Pseudo R-squ. (CS):
0.1889
Covariance Type:            nonrobust
==================================================================
===========================
                               coef    std err          z      P>
|z|      [0.025      0.975]
------------------------------------------------------------------
---------------------------
Is_Male                      0.8131      0.004    191.990      0.
000       0.805       0.821
ALIGN_Good Characters        1.8575      0.005    406.103      0.
000       1.849       1.866
ALIGN_Neutral Characters     1.3113      0.006    202.895      0.
000       1.299       1.324
ALIGN_Reformed Criminals    -0.0435      0.302     -0.144      0.
885      -0.634       0.548
ALIVE_Living Characters      1.2612      0.005    243.174      0.
000       1.251       1.271
Studio_Marvel                0.3450      0.004     88.183      0.
000       0.337       0.353
==================================================================
===========================
```

Task 4: Interpret coefficients in the context of our original research question of "what

characteristics are associated with a reduction in appearances?"

Note: You can ignore any broken assumptions at this point and simply treat it as an effective model.

Interpretation of Coefficients in the Context of the Research Question

The original research question is: "What characteristics are associated with a reduction in appearances?"

Given the Poisson GLM summary, we interpret the coefficients to understand how each characteristic influences the number of appearances.

Summary of Coefficients

1. **Is_Male (Coefficient: 0.8131)**
   - Being male is associated with an increase in the number of appearances. Male characters have a higher expected number of appearances compared to non-male characters.
2. **ALIGN_Good Characters (Coefficient: 1.8575)**
   - Being classified as a good character is strongly associated with an increase in appearances. Good characters are expected to appear significantly more often than non-good characters.
3. **ALIGN_Neutral Characters (Coefficient: 1.3113)**
   - Neutral characters see an increase in the number of appearances. Neutral characters are expected to appear more frequently compared to non-neutral characters.
4. **ALIGN_Reformed Criminals (Coefficient: -0.0435)**
   - Being a reformed criminal is associated with a slight reduction in appearances. Although this coefficient is not statistically significant ($p = 0.885$), it suggests a potential tendency for reformed criminals to have fewer appearances.
5. **ALIVE_Living Characters (Coefficient: 1.2612)**

- Living characters are associated with an increase in appearances. Characters who are alive tend to appear more frequently than those who are not.

6. **Studio_Marvel (Coefficient: 0.3450)**

   - Characters associated with Marvel Studios are more likely to have a higher number of appearances. Being part of the Marvel universe contributes to more frequent appearances.

Conclusion

- **Characteristics Associated with a Reduction in Appearances:**

  - **ALIGN_Reformed Criminals**: Despite being the only negative coefficient, it is not statistically significant. It suggests a potential, but weak, reduction in appearances.
- **Overall Finding:**

  - There are no strong characteristics identified that significantly reduce the number of appearances. Most characteristics analyzed, including being male, being a good or neutral character, being alive, and being part of Marvel Studios, are associated with an increase in appearances.