THE RUSSIAN GOVERNMENT
FEDERAL STATE AUTONOMUS EDUCATIONAL INSTITUTION
FOR HIGHER PROFESSIONAL EDUCATION
NATIONAL RESEARCH UNIVERSITY
"HIGHER SCHOOL OF ECONOMICS"

The Analysis of On-screen Movie Kills

Homework Project 2018/2019

The team:
Boris Tseitlin
Konstantin Romashchenko
Yulia Gurova

MSc Program Data Science
1^{st} year
Faculty of Computer Science

Moscow 2018

Table Of Contents

1 THE CHOICE OF THE DATASET **3**

2 K-means clustering **4**

2.1 Preprocessing 4

2.2 K-means implementation 5

2.3 Interpretation 7

2.4 Bootstrap 10

3 Contingency Table Analysis **14**

4 PCA: Hidden Factor & Data visualization **14**

5 2D regression **14**

6 Applications **14**

1 THE CHOICE OF THE DATASET

The dataset that is used for the project contains information about on-screen deaths in movies. There are 545 movies (more then 100 objects) and 8 characteristics including names, so the dataset meets the requirements.

The sourse for the data is the thematic web-site moviebodycounts.com. This dataset was processed and published on figshare.com. It was gathered in accordance with the rules, which are published on the web-site. We took several characteristics for the consideration: the year of the film, MPAA rating (Motion Picture Association of America film rating system), genre or genres, the name of the director, the lenth of the film in minutes, IMDB rating based on user ratings. The main feature that we consider is the number of on-screen deaths in the movie.

The analysis of the data may reveal how the ratings depend on the number of deaths, how this number is changing with the year of the release and so on. Moreover, genre and length of the film combined with the violence on the screen may give the idea of the age ratings. This is a good set for the classification and clustering problems, as the films are grouped on genres, MPAA ratings which are similar inside the groups, but dissimilar between them.

This study analysis may be the first step to the automation of age rating systems. Moreover it may be helpful in the development of the recommendation systems. And, as watching movies is the common interest of our team, the work with the dataset will inspire us to further conquest in Data Analysis.

2 K-means clustering

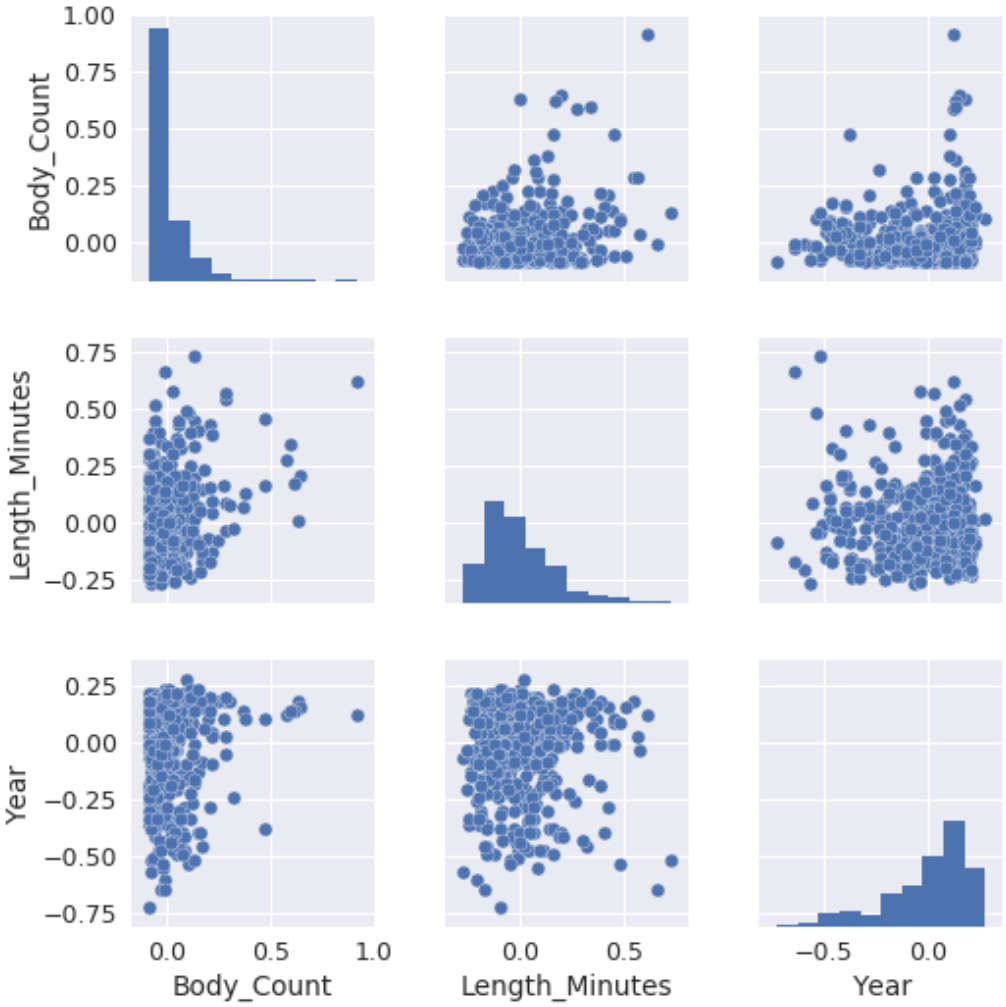
For this part of the task we choose three quantitative features: *Body_ Count*, the number of on-screen deaths in the movie; *Length_ Minutes*, the length in minutes; *Year*, the year of the release. This choice is restrained as these are the only quantitative features of our dataset.

2.1 Preprocessing

First, we normalize (standardize) the features:

```
In [11]:  def normalize(vec):
    return (vec - vec.mean())/(vec.max() - vec.min())
```

Before implementing the clustering methods, we visualize data on all possible pairplots. There are no obvious clusters in two-dimensional visualization.



The clusters might follow the categorical feature when it's used together with quantitative features. A categorical attribute with the least amount of variants we have is MPAA rating, that bounds the age restrictions. We color the graphs with accordance to the ratings to see how they divide our data.

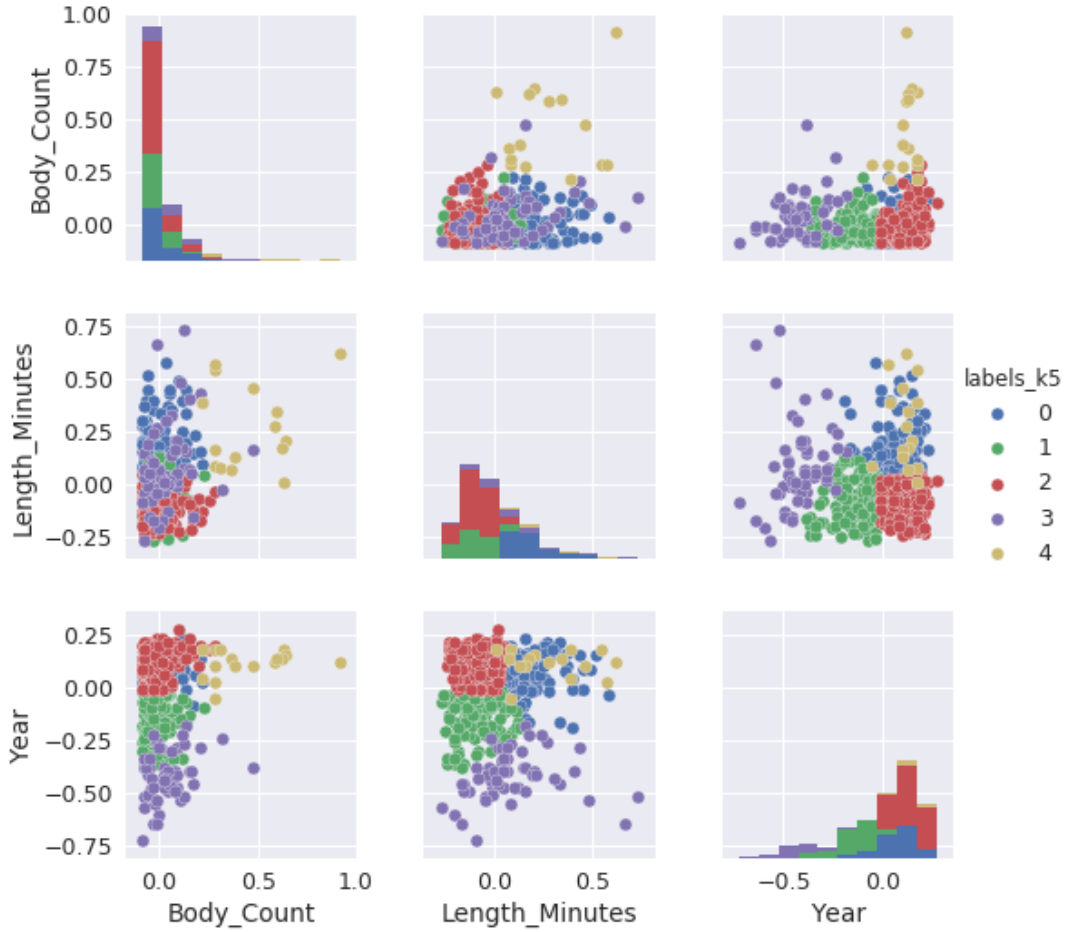
2.2 K-means implementation

The first method we apply to find the clusters is k-means at k=5. So we take random initializations of 5 cluster centers and choose the best by k-means criteria from 10 initializations. The sum of squared distances from points to cluster centers: 13.6126.

```
In [16]:
kmeans_k5 = KMeans(n_clusters=5, init='random', n_init=10, random_state=RANDOM_SEED)
kmeans_k5.fit(task_df)
task_df['labels_k5'] = pd.Series(kmeans_k5.predict(task_df))
print('Sum of squared distances from points to cluster centers, k=5:', kmeans_k5.inertia_)
sns.pairplot(task_df, hue='labels_k5', vars=quant_features)
```

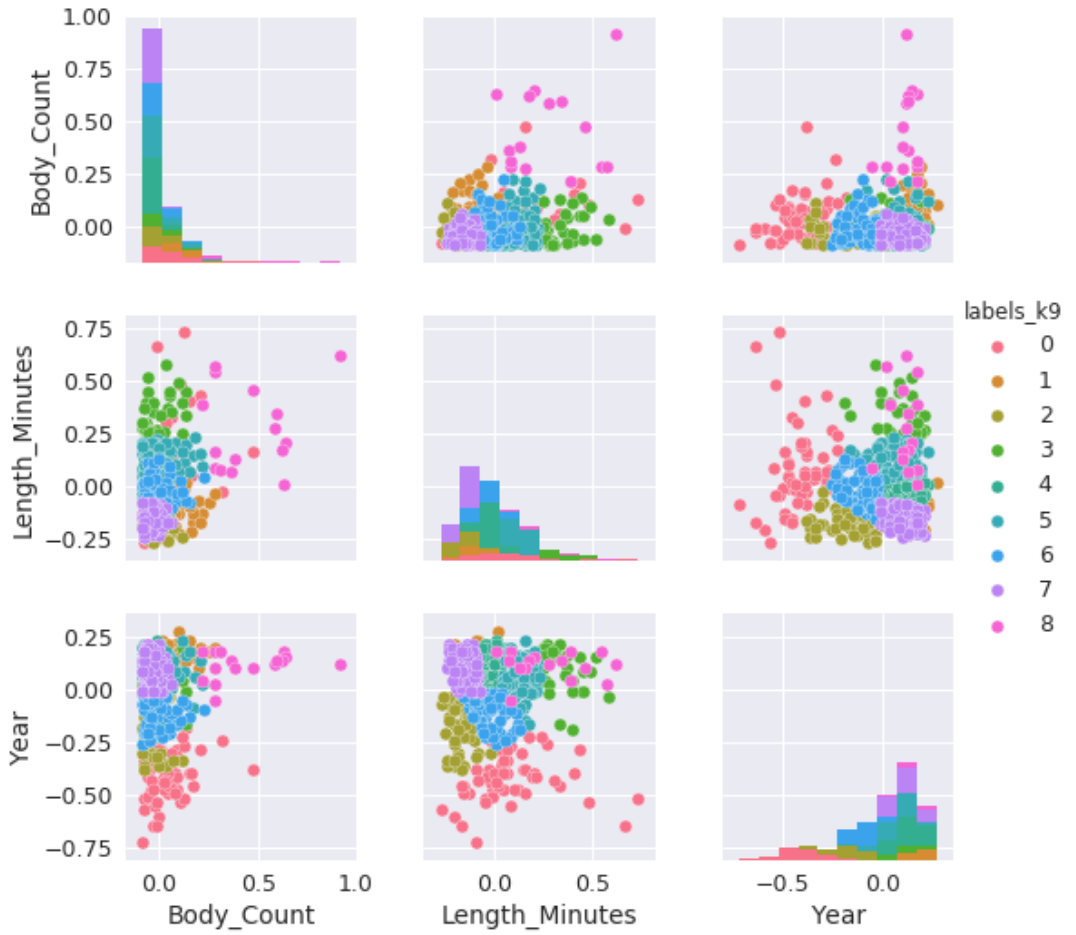
The obtained clusters are following (we visualize them by pairplots). They don't match to MPAA_Rating directly, but divide the data into reasonable

descriptive categories:



We would specially distinguish cluster 1 (orange colored), the old films, which are mostly not long and with small amount of deaths. Cluster 3 (red colored) is not large and depicts the films with the highest number of deaths, they differ in length, but are all relatively the new ones.

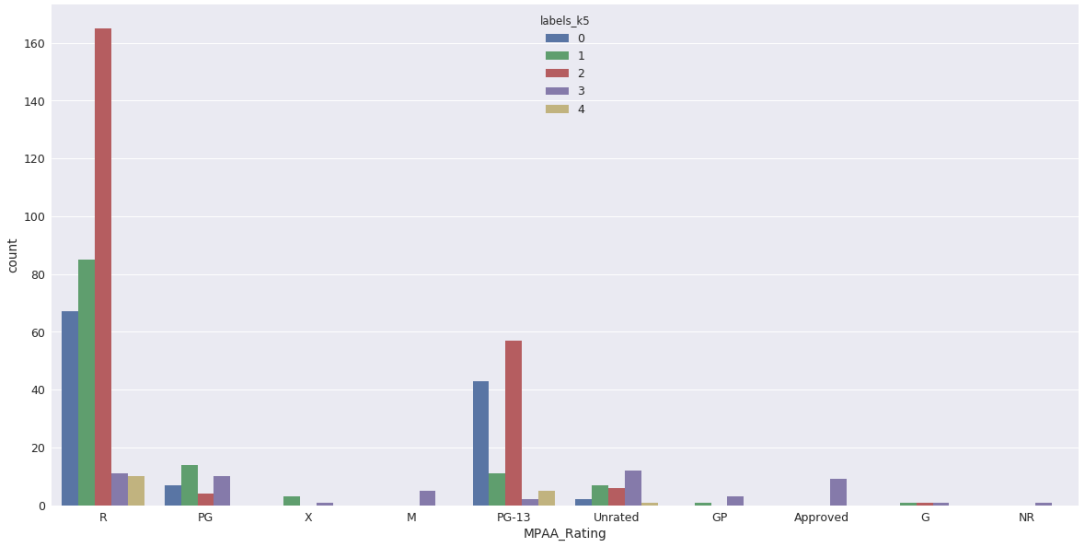
Now we take clustering at k=9 and get a lot of smaller clusters:



Some clusters are obviously inherited from the pevious ones: new cluster 7 (the gray one) follows cluster 3 (with the higest deaths rate), the new cluster 3 (red) follows the discussed cluster 1 (with old films) from the previous graphs. Besides these two there are many small clusters at the main body of the films, overlapping on the most graphs without having patent interpretation. We conclude that 9 clusters are too much for this dataset, and 5 clusters better describe the data: the cluster boundaries were more clear and the reasonably interpretable clusters were the same.

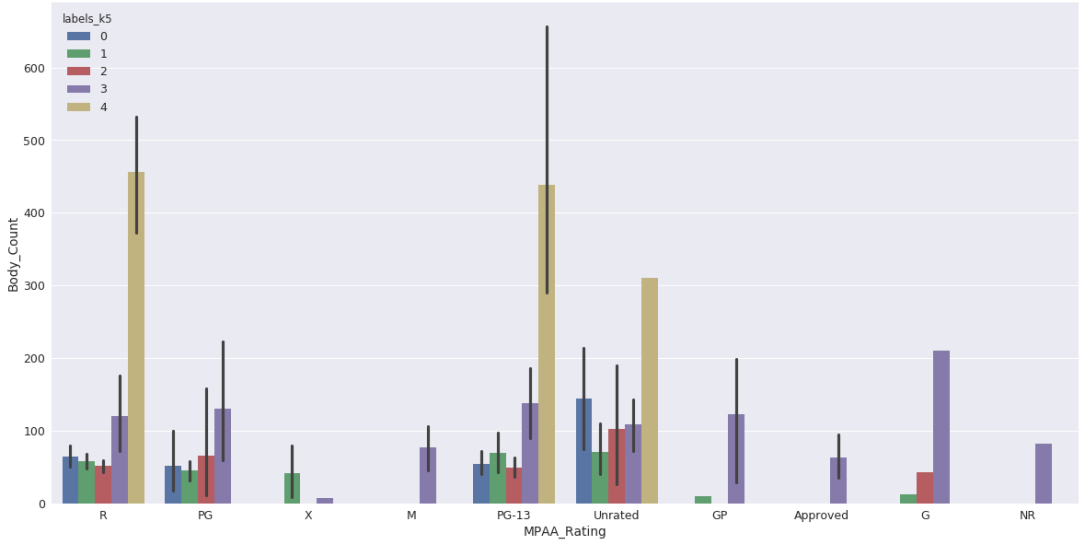
2.3 Interpretation

In the rest part of this task we consider the partition with k=5 as it is more reasonable for this dataset. First, we explore, how movies are destributed to clusters, minding their MPAA ratings (the correspondence of the ratings is in application 1).



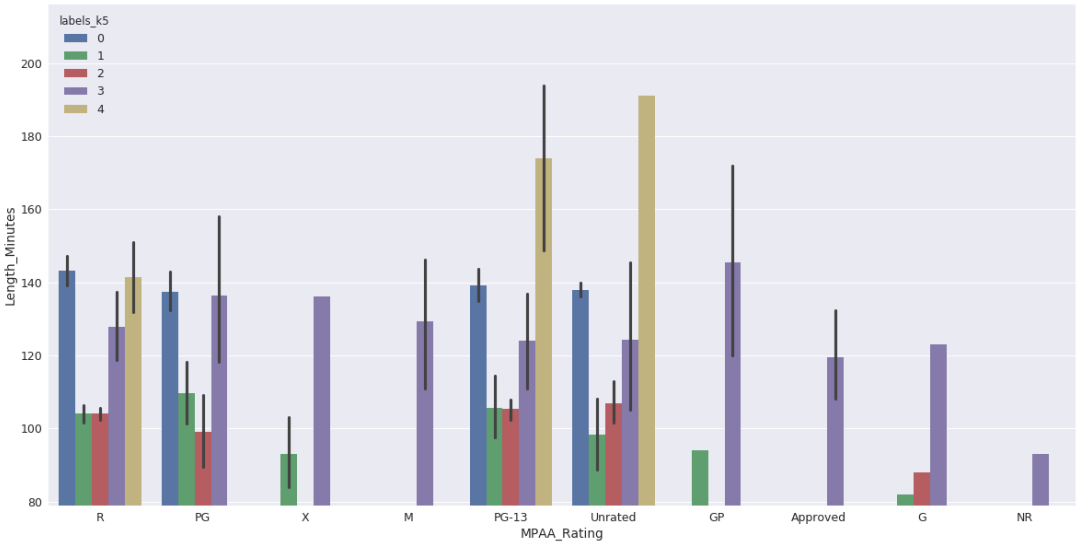
The plot shows the amount of movies of each rating in each cluster. For example we see that 165 R movies are in cluster 4. Clusters 0, 2, 4 aggregate the most part of the movies, mostly with R and PG-13 ratings. That may show that we have mostly adult films in the dataset. That is not a surprise due to considered subject. The number of films at the clusters are: 4 - 233, 0 - 122, 2 - 119, 1 - 55, 3 - 16.

Next we compare the mean values of the features between the clusters. The following barplot shows the correspondence of number of deaths in clusters, separated by ratings. The height of each bar shows the average body count of movies of a certain rating in a cluster. The vertical black bar represents the spread - the highest point shows the maximum, the lowest point shows the minimum.



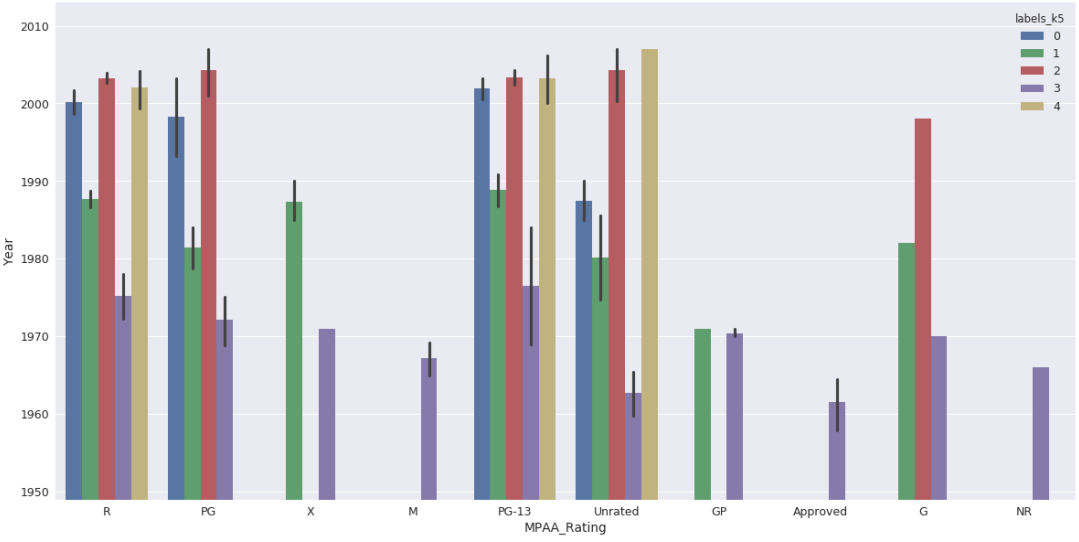
It's clear that cluster 3 contains the movies with the highest body count, as was supposed in the cluster interpretation. It contains the PG-13 movie with the highest body count in the dataset (836): *"Lord of the Rings: Return of the King"*. The highest point of the black bars shows this movie.

Next we consider the average length of the movies by clusters:



This plot is not very informative. We can see that clusters 0 and 4 are on avearage the shortest ones. And the movies from the cluster 3 (with highest mount of death) are long on average.

Next we consider the year of the movies by cluster and ratings. This plot has the same structure.



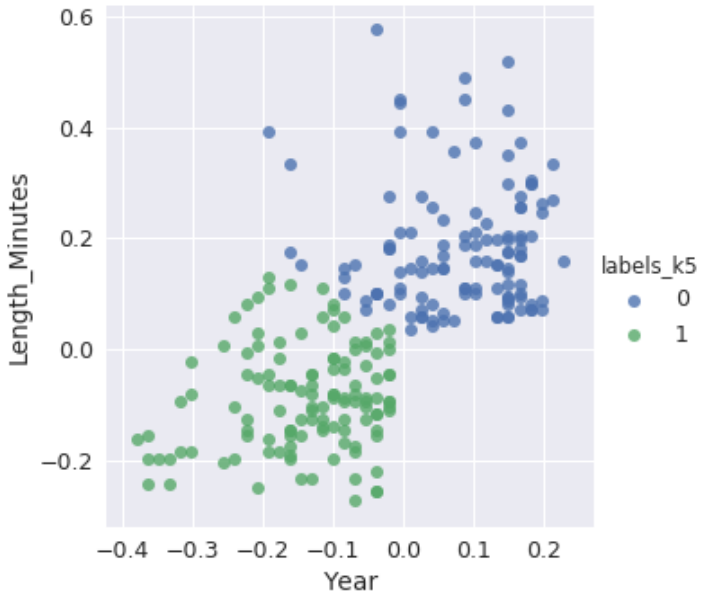
It can be seen that the cluster 1 contains mostly old movies, as was discovered before. Now we can see that these are mostly pre-1980 movies.

Cluster 0 contains movies from about the 90-s. And it is interesting that the highest body count movies, captured by cluster 3, are mostly recent - post-2010.

2.4 Bootstrap

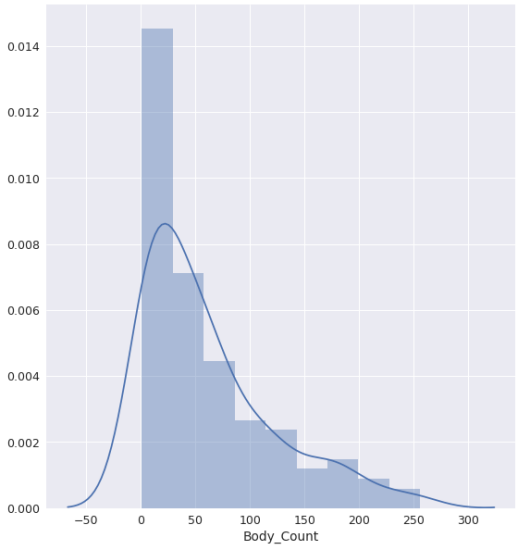
In this part of the paper we inspect closely two clusters: 0 and 1. As was mentioned, cluster 1 contains mostly old, pre-1980 movies and cluster 0 has movies from about the 90-s.

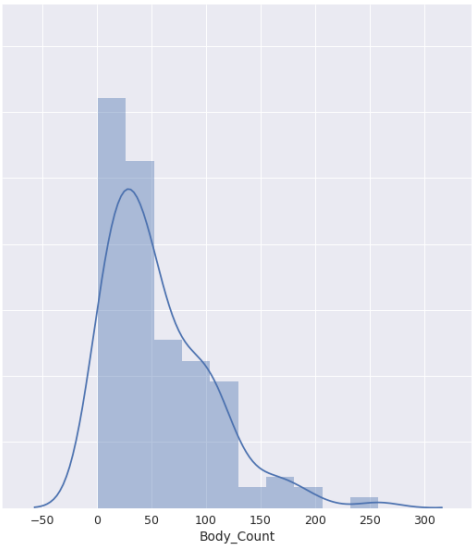
First, we compare the length and the year, and then we focus on our main feature, the body count. From the scatter plot the difference between the clusters can be clearly seen.



First, we consider the distribution of our prime feature, the body count, in the two clusters.

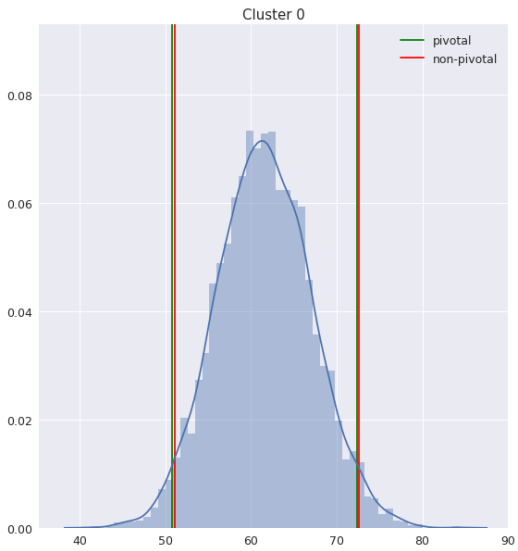
```
In [48]:
print(cluster_0[target_feature].describe())
print(cluster_1[target_feature].describe())
fig = plt.figure(figsize=(20,10))
axes = fig.subplots(1, 2, sharex=True, sharey=True)

sns.distplot(cluster_0.Body_Count, ax=axes[0])
sns.distplot(cluster_1.Body_Count, ax=axes[1])
```

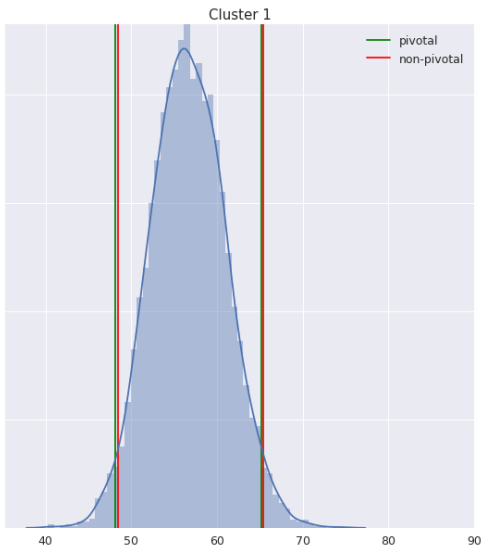




The parameters of the distributions are listed in the table. The distributions seem to be different but we cannot be sure as the samples are rather small.

Parameter	**Cluster 0**	**Cluster 1**
count	122.000000	55.000000
mean	56.795082	105.654545
std	47.931414	87.318323
min	1.000000	4.000000
25%	20.000000	44.500000
50%	42.500000	91.000000
75%	87.250000	147.000000
max	258.000000	471.000000





We apply bootstrap with 5000 samples to compare the distribution between the clusters. We obtain that the distributions of Body_Count really differ between these clusters. They not only have different means, as evident by confidence intervals, but also have different bell shapes. We conclude that the distribution of the feature is approximately Gaussian.

The code for bootstrap implication and the confidence intervals is below. We use pivotal and non-pivotal bootstrap for the confidence intervals. It can be seen from the graphs, that they are almost coincide, so the destributian of the feature is close to Gaussian.

```
In [91]: def bootstrap_sample(vec, size):
return np.random.choice(vec, size=(vec.shape[0], size), replace=True)

def bootstrap_means(srs,     sample_amount=5000):
    samples_ix = bootstrap_sample(srs.index, size=sample_amount).T
    means = np.array([srs.loc[sample].mean() for sample in samples_ix])
    return means

b_means_cluster_0 = bootstrap_means(cluster_0.Body_Count)
b_means_cluster_1 = bootstrap_means(cluster_1.Body_Count)

def confidence_interval_pivotal(vec):
    mean = vec.mean()
    std = vec.std()
    return [mean-1.96*std, mean+1.96*std]

def confidence_interval_non_pivotal(vec, alpha):
    left = np.percentile(vec, (100-alpha)/2)
    right = np.percentile(vec, alpha+(100-alpha)/2)
    return [left, right]

def distplot_with_conf_intervals(vec, ax=None):
    if not ax:
        ax = plt.gca()
    for x in confidence_interval_pivotal(vec):
        line = ax.axvline(x=x, color='g')
    line.set_label('pivotal')
    for x in confidence_interval_non_pivotal(vec, 95):
        line = ax.axvline(x=x, color='r')
    line.set_label('non-pivotal')
    ax.legend(loc='best')
    return sns.distplot(vec, ax=ax, norm_hist=False)
```

Next we build the 95% confidence intervals for the grand mean of the feature by using bootstrap. Also we use bootstrap to compare cluster 0 to the grand mean. We obtain:

For cluster 0

pivotal mean: 56.83192786885246, c.i.: [48.382188750197514, 65.2816669875074]

non pivotal mean: 56.83192786885246, c.i.: [48.90163934426229, 65.55758196721311]

For cluster 1

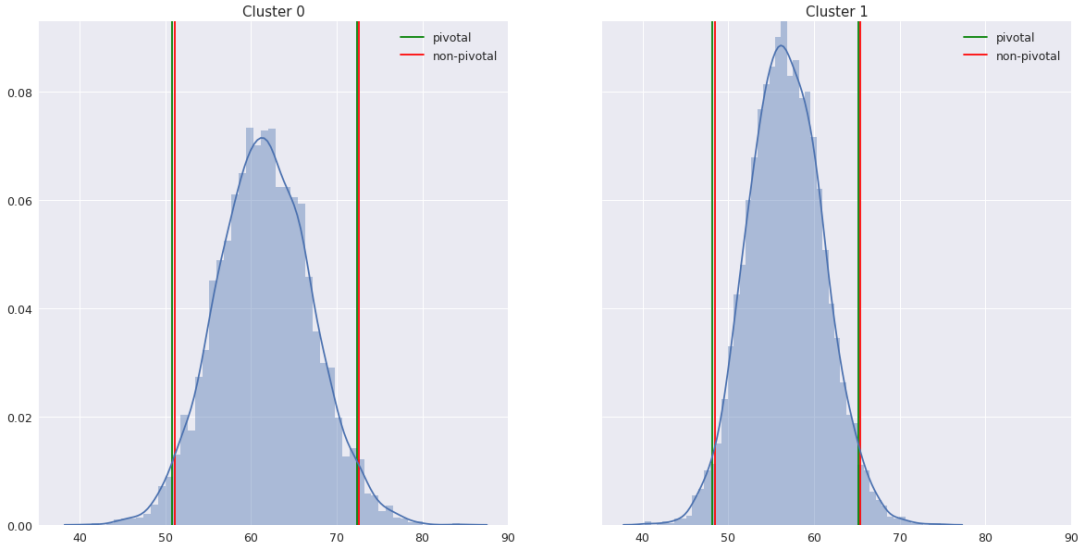
pivotal mean: 105.62885818181817, c.i.: [82.64639614766133, 128.611320215975]

non pivotal mean: 105.62885818181817, c.i.: [84.01772727272727, 130.05999999999992]

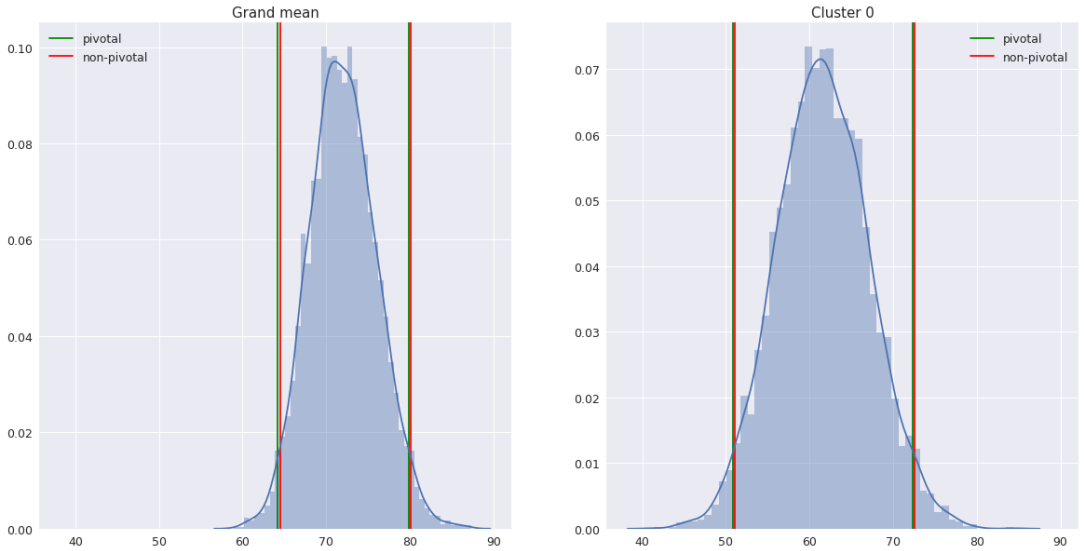
For the grand mean

pivotal 72.15726385321102 +- [64.37419120011303, 79.94033650630901] non pivotal 72.15726385321102 +- [64.65105504587156, 80.23775229357797]

Comparison between the grand mean and cluster 0:



We pay attention here to the fact that the bell shape of cluster 0 body count resembles the grand mean bell shape closely.



3 Contingency Table Analysis

4 PCA: Hidden Factor & Data visualization

5 2D regression

6 Applications

1. **MPAA Ratings interpretation.** Sourse: MPAA

   - **G General audiences**. All ages admitted. Nothing that would offend parents for viewing by children.
   - **PG Parental Guidance Suggested**. Some material may not be suitable for children. Parents urged to give "parental guidance". May contain some material parents might not like for their young children.
   - **M:** Suggested for Mature Audiences parental discretion advised (before 1984). The same as modern PG.
   - **GP** All Ages Admitted Parental Guidance Suggested (before 1972). Renamed to PG.
   - **PG-13 Parents Strongly Cautioned**. Some material may be inappropriate for children under 13. Parents are urged to be cautious. Some material may be inappropriate for pre-teenagers.
   - **R Restricted**. Under 17 requires accompanying parent or adult guardian. Contains some adult material. Parents are urged to learn more about the film before taking their young children with them.
   - **NC-17 Adults Only.** No One 17 and Under Admitted. Clearly adult. Children are not admitted.
   - **X**. No one under 17 admitted (before 1984). Was renamed to NC-17