DATA ANLYSIS

Jiyoon Kim

DATA ANALYSIS

Method 1

Linear regression 2

Correlation 3

K-Means 4

1. Methode

Data collection

1. Pulling from a rest API.

2. Pulling from an SQLite database.

3. Reading from a .csv file

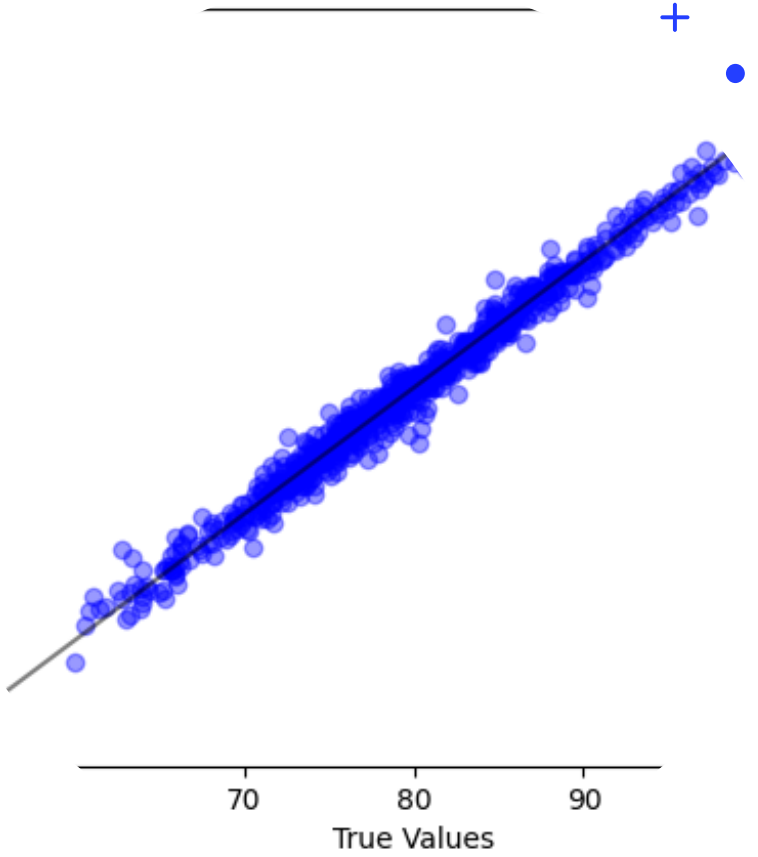
Data Cleaning Techniques

- Dropping NaNs
- Dropping Duplicates (there were no duplicates)
- Dropping Few negative values
- Converting strange characters converted tot NaNs and dropping
- Leaving Outliers for further analysis

EDA conclusions(dataset to use for the final version)

We decided that df4 would be the best dataset to use for the final regression model.

4	**Experiment Tracking Table**							
5	train, test = train_test_split(mydf, test_size=0.2, random_state=42)							
6								
7	**Experiment #**	**NaN drop**	**Dupli drop**	**Neg drop**	**object -> float**	**BMI feature**	**IQR-clip**	**IQR-drop**
8	DF1	x	x	x	x			
9	DF2	x	x	x	x	x		
10	DF3	x	x	x	x	x	x	
11	DF4	x	x	x	x	x		x



2.LinearRegressions

LINEAR REGRESSIONS AND SPLIT DATASETS USING SKLEARN
MRSE(MEAN ROOT SQUARED ERROR)

THE RESULT OF F4

```
train, test = train_test_split(df, test_size=0.2, random_state=42)

X = train[['genetic', 'length', 'mass', 'exercise', 'smoking', 'alcohol', 'sugar','bmi']]
y = train.lifespan

regr = LinearRegression()
regr.fit(X, y)

score = regr.score(test[['genetic', 'length', 'mass', 'exercise', 'smoking', 'alcohol', 'sugar','bmi']],test.lifespan)
print(f'coefficient of determination(R\N{SUPERSCRIPT TWO}) vanilla:', score)
a1=score
```

[13] ✓ 0.0s Python

```
coefficient of determination(R²) vanilla: 0.9820618333051058
```

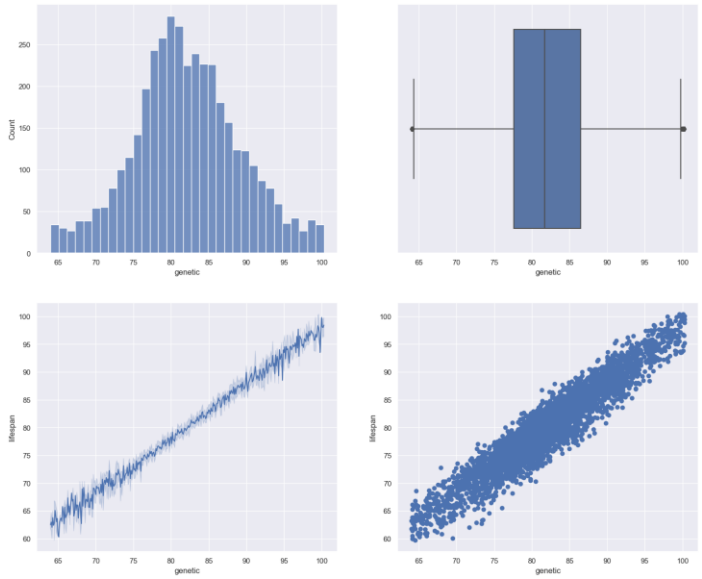
```
print('Mean Absolute Error:', mean_absolute_error( test['lifespan'], p_test))
print('Mean Squared Error:', mean_squared_error(test['lifespan'], p_test))
import math
print('Mean Root Squared Error:', math.sqrt(mean_squared_error(test['lifespan'], p_test)))
```

[16] ✓ 0.0s Python

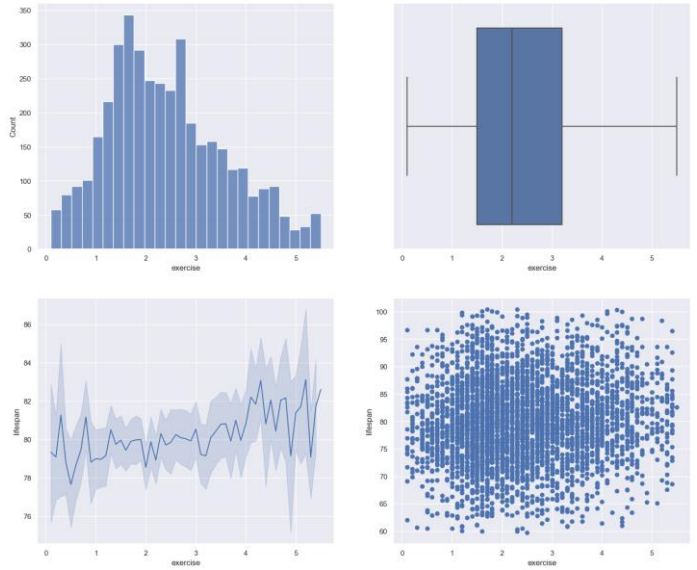
```
Mean Absolute Error: 0.7632040181701046
Mean Squared Error: 1.0510579968056664
Mean Root Squared Error: 1.0252111961960162
```

If Mean Root Squard Error score is lower, it is more reliable.

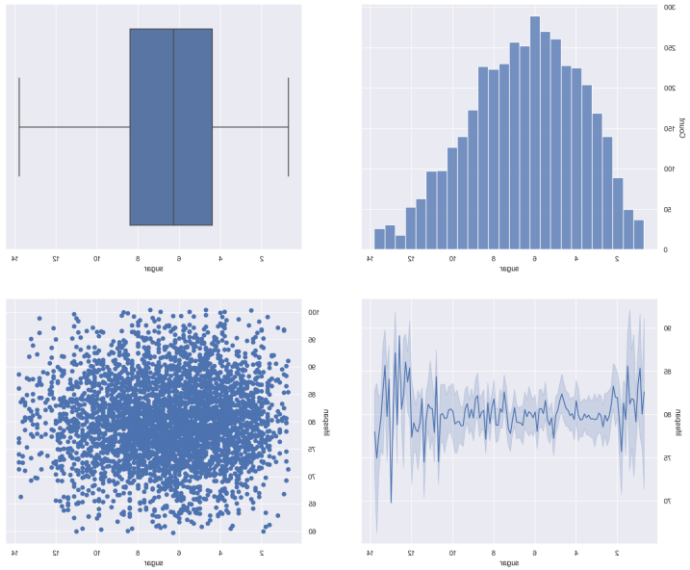
3. The correlation with lifespan



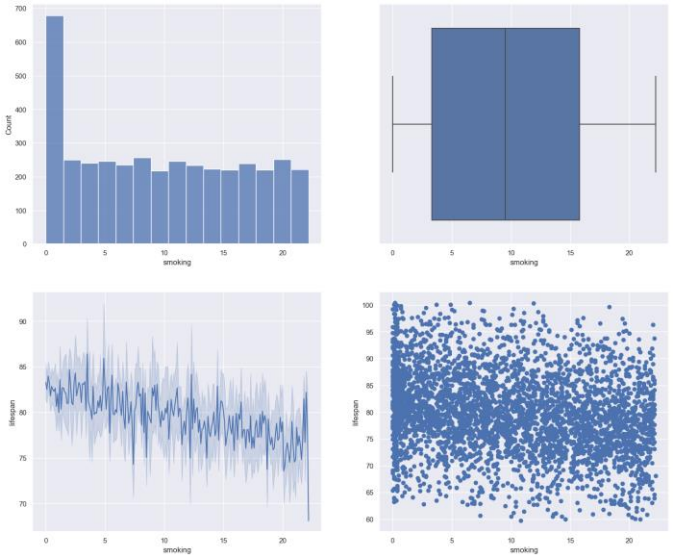
lifespan – genetic
Highly positive relation



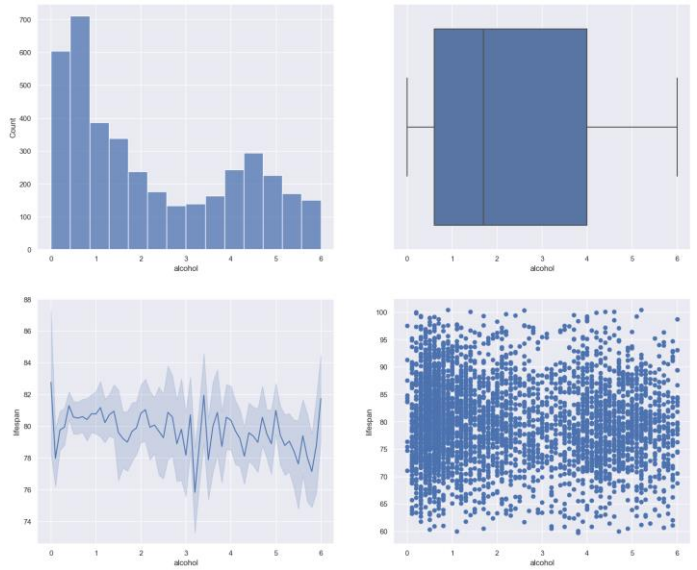
lifespan – exercise
Postive relation



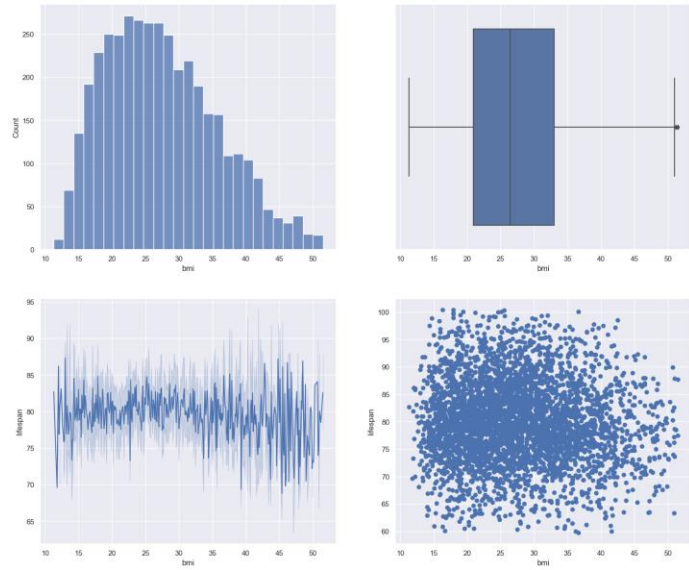
lifespan - sugar



lifespan – smoking
negative relation

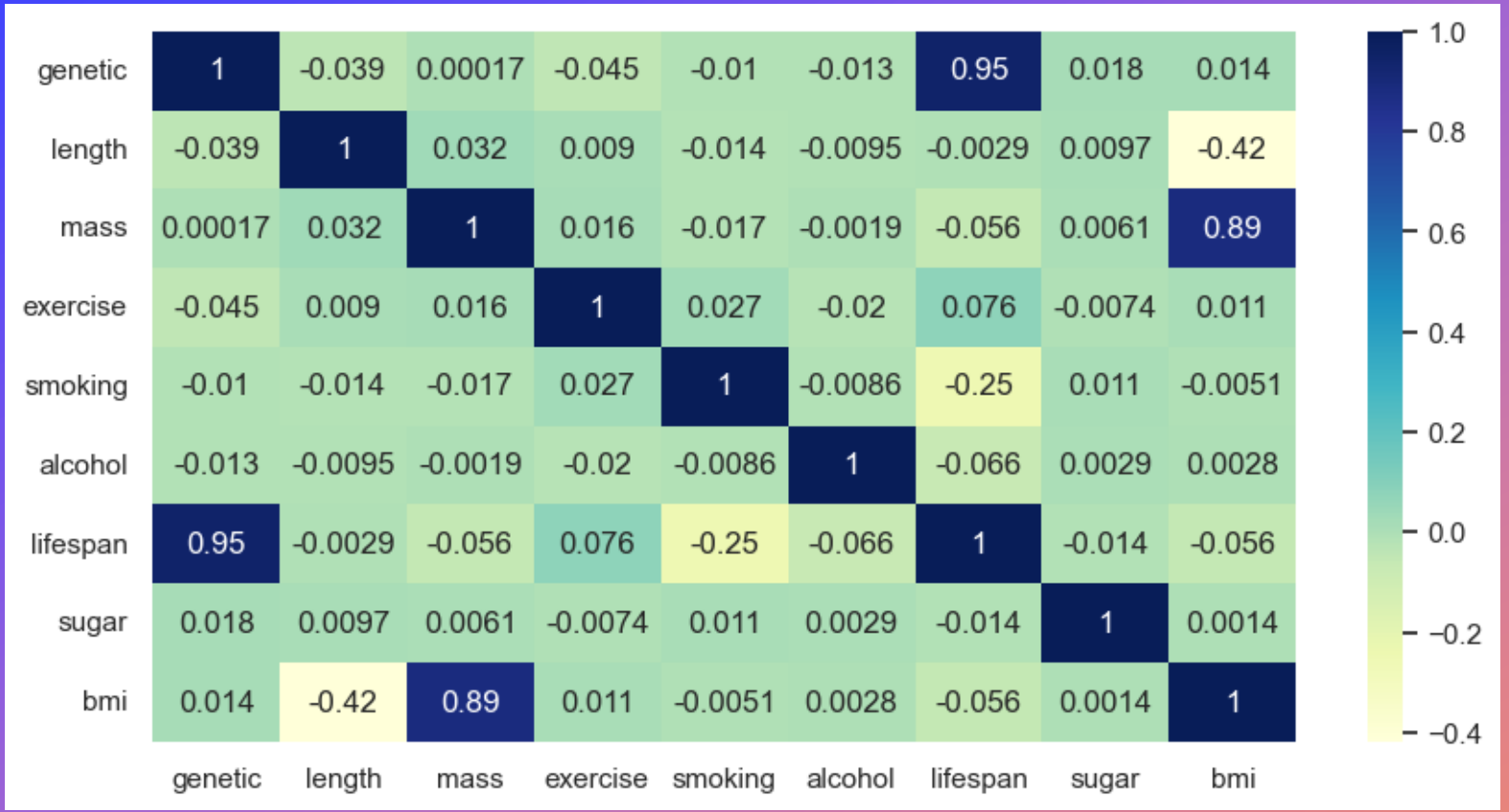


lifespan – alcohol
negative relation



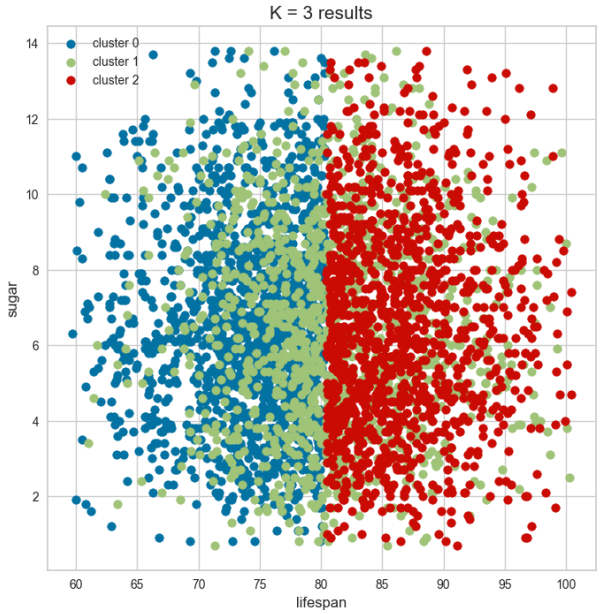
lifespan - bmi

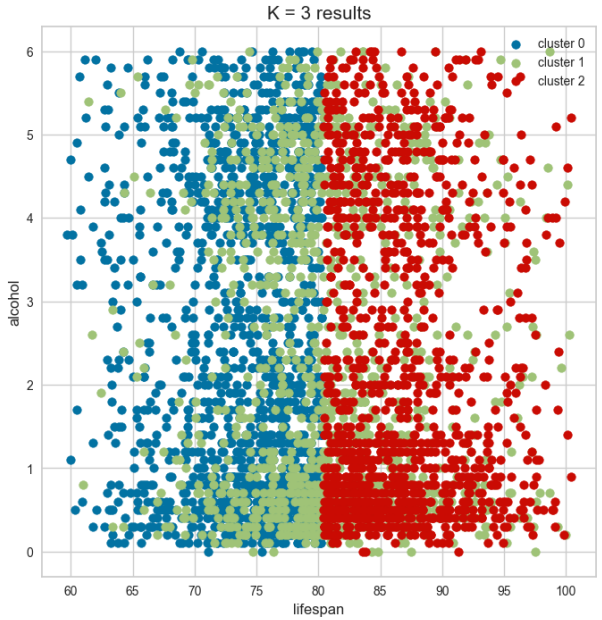
correlation matrix view

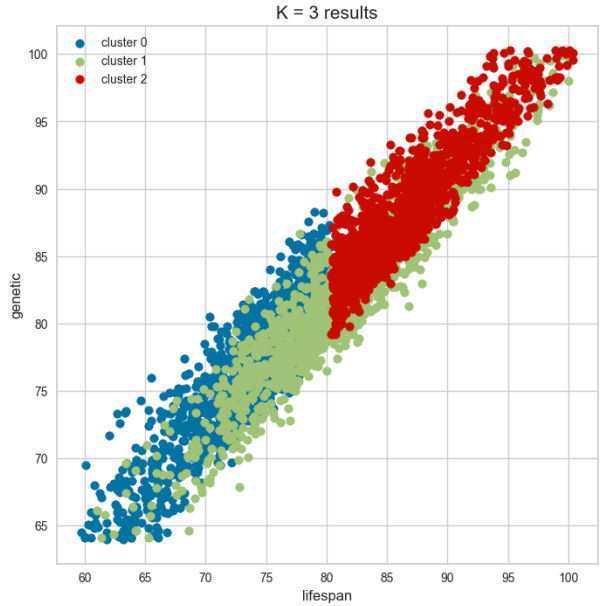


- Genetic shows strong correlation with lifespan
- Secondly exercise, has correlation with lifespan
- Smoking and alcohol has negative correlation with lifespan
- Other variables have also some negative correlation

Implements the "elbow" method of selecting the optimal number of clusters by fitting the K-Means model with a range of values for K.





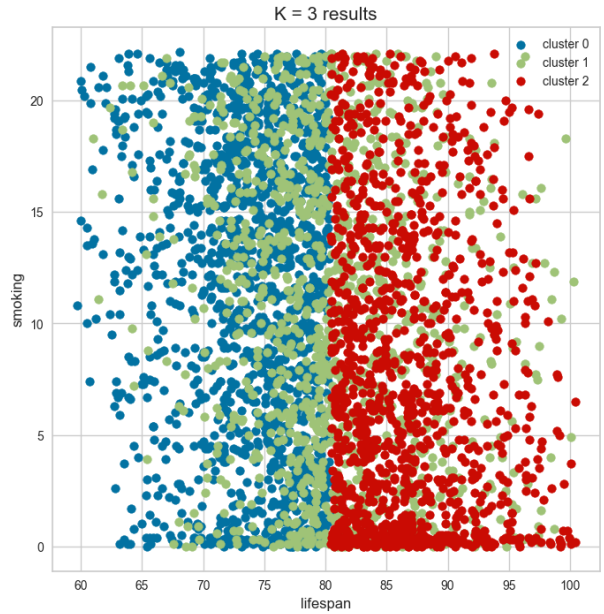


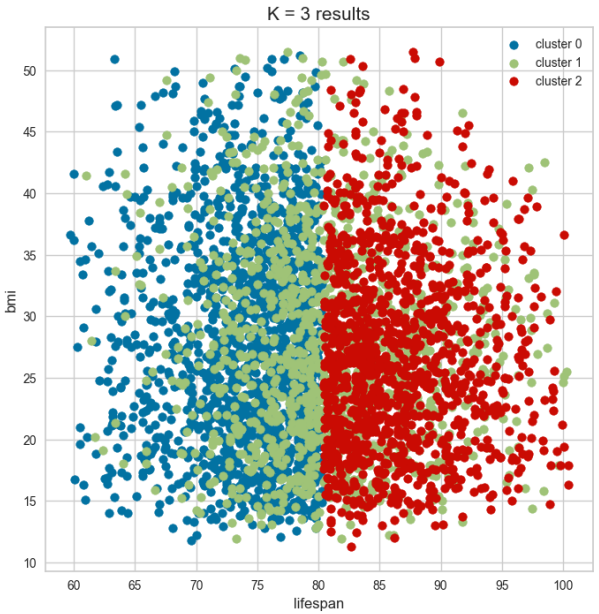
4. K-Means

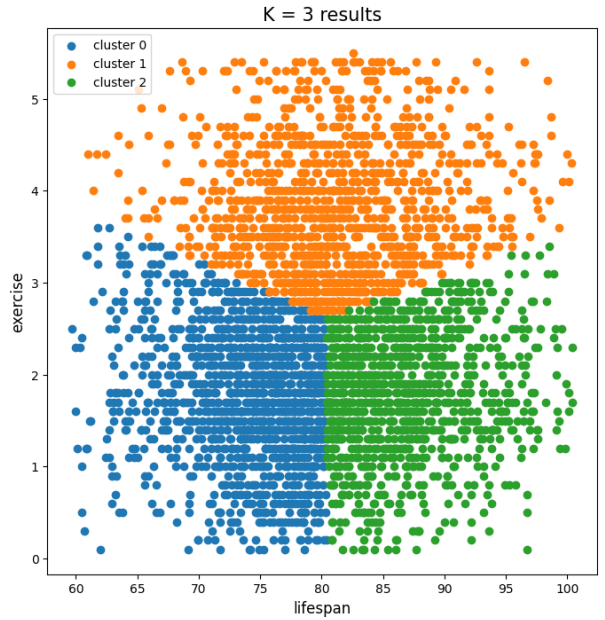
<Lifespan – Genetic>
Clustered in one line

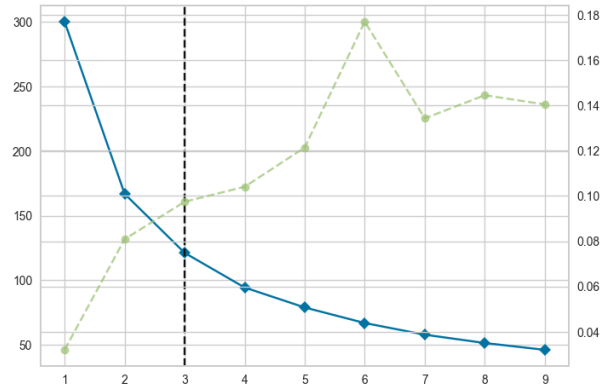
<Lifespan-Exercise>
Clusterd 3 shapes
This pattern looks different with other combinations

<Interesting use for K-means>
1.Delivery store optimization
2. Identifying Crime Localities
3. Cyber-profiling Criminals
4. Fantasy League Stat Analysis









KElbowVisualizer selecting the optimal number of clusters