Rapid Analysis : Clustering of Toxicity for Vaccine Lots

By Craig Paardekooper

Data Source:

2021 data VAERS USA : https://howbad.info/tox-data.csv

Data fields were

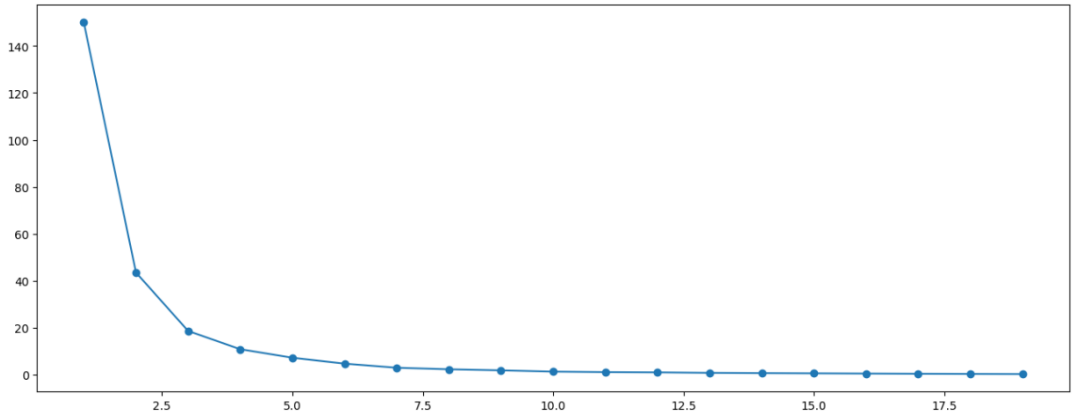
- **lot size shipped** for 150 different lots - https://howbad.info/lotsize.xlsx
- **number of adverse reaction** reports for each lot

Toxicity was defined as **number of adverse reactions per 100,000 doses shipped for each lot**.

Method :

K-means clustering was used to see if there was any grouping of toxicities.

The "Elbow method" showed 3 clusters of toxicity –



k-means clustering was then applied based on 3 clusters

Results :

Cluster 1

low toxicity - averaging 24.5 adverse reactions per 100,000 doses shipped

88 lots = 59% of the total number of lots analysed (88 out of 150)

Pfizer F series (FA, FC, FD, FE, FF, FG, FH, FJ)

Cluster 2

highest toxicity - averaging 231.7 adverse reactions per 100,000 doses shipped

15 lots = 10% of the total number of lots analysed (15 out of 150)

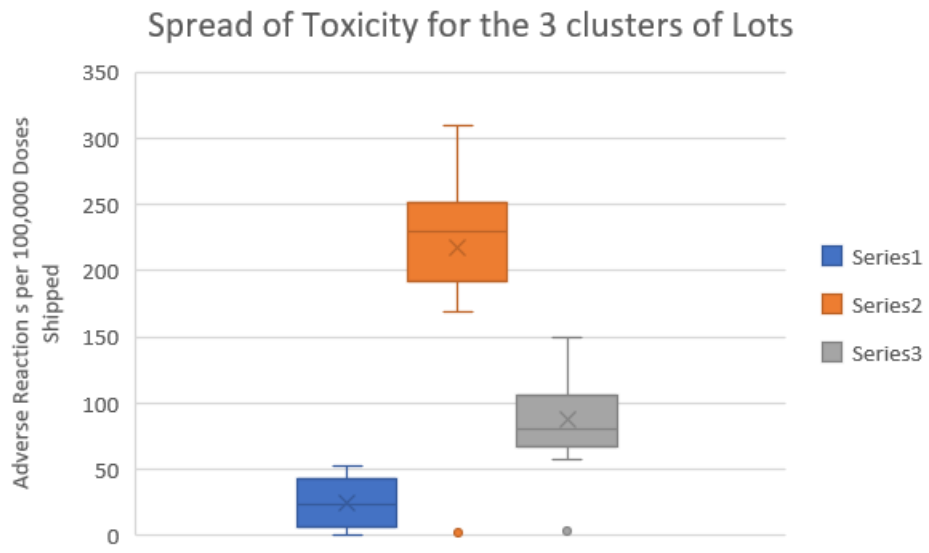
Pfizer E series (EH, EJ, EK, EL)

Cluster 3

medium toxicity - averaging 89 adverse reactions per 100,000 doses shipped

47 lots = 31% of the total number of lots analysed (47 out of 150)

Pfizer E series (EL, EM, EN, ER, EW)



Discussion :

For US data we see that almost two thirds of the lots were of low toxicity, almost one third of medium toxicity and 10% of high toxicity.

USA data shows –

High toxicity	10%	Pfizer E series (EH, EJ, EK, EL)
Medium toxicity	31%	Pfizer E series (EL, EM, EN, ER, EW)
Low toxicity	59%	Pfizer F series (FA, FC, FD, FE, FF, FG, FH, FJ)

We can compare this to the Denmark study

Denmark data showed –

High toxicity	4%	Pfizer EJ EK EL EM
Medium toxicity	64%	Pfizer EP ER ET EW EX EY FA FC FD FE FF and FG

Low toxicity 32% Pfizer FG FH FJ FK FL FM

Ref : Danish Study

Placebo Batch Numbers

The Pfizer lot number series that correspond to high, medium and low toxicity are similar in both the USA data and in the Danish study.

In both cases, as the alphabet ascends the toxicity appears to decrease.

The main difference between the USA and Danish data is that in the USA there appear to be a higher proportion of high toxicity and low toxicity batches. In other words there is more of a polarisation – with less medium toxicity.

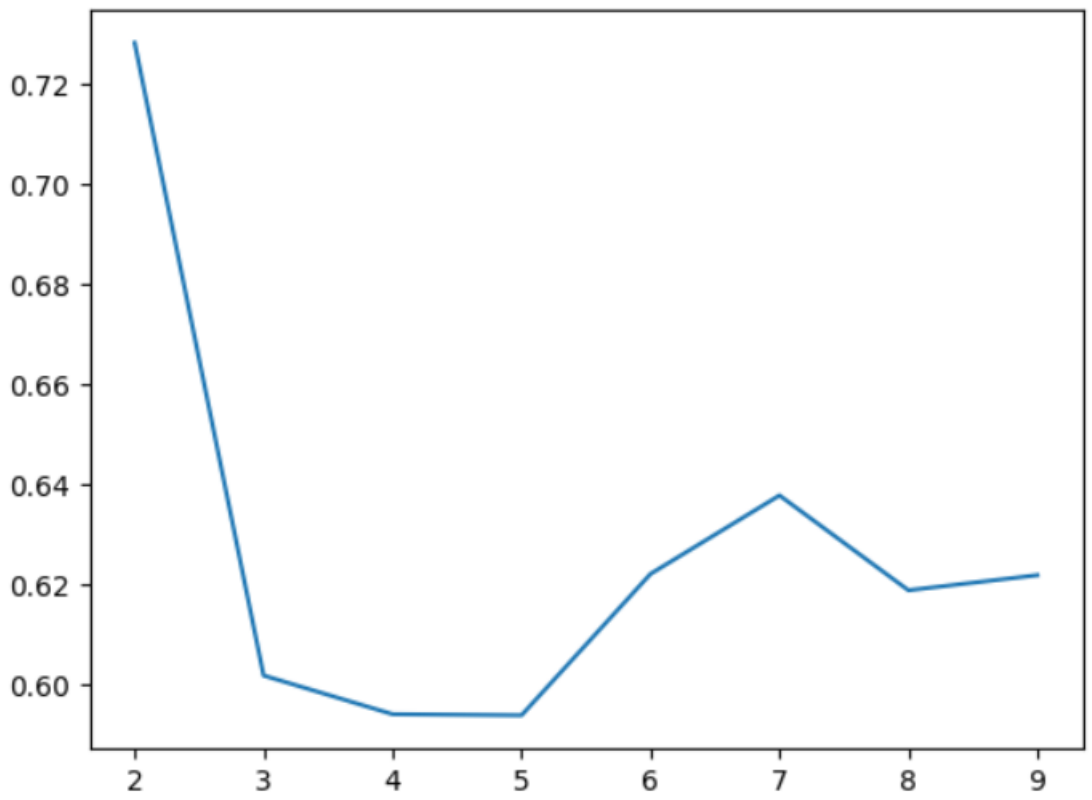
So in the USA 10% of the batches were highly toxic, and 60% were low toxicity. The larger number of low toxicity batches would generate more support for the vaccines, which would help maintain the vaccine rollout despite a larger % of high toxicity batches.

These findings are compatible with the V-Safe findings where 7.7% of vaccinated sought medical treatment after vaccination for COVID-19. See V-Safe Data

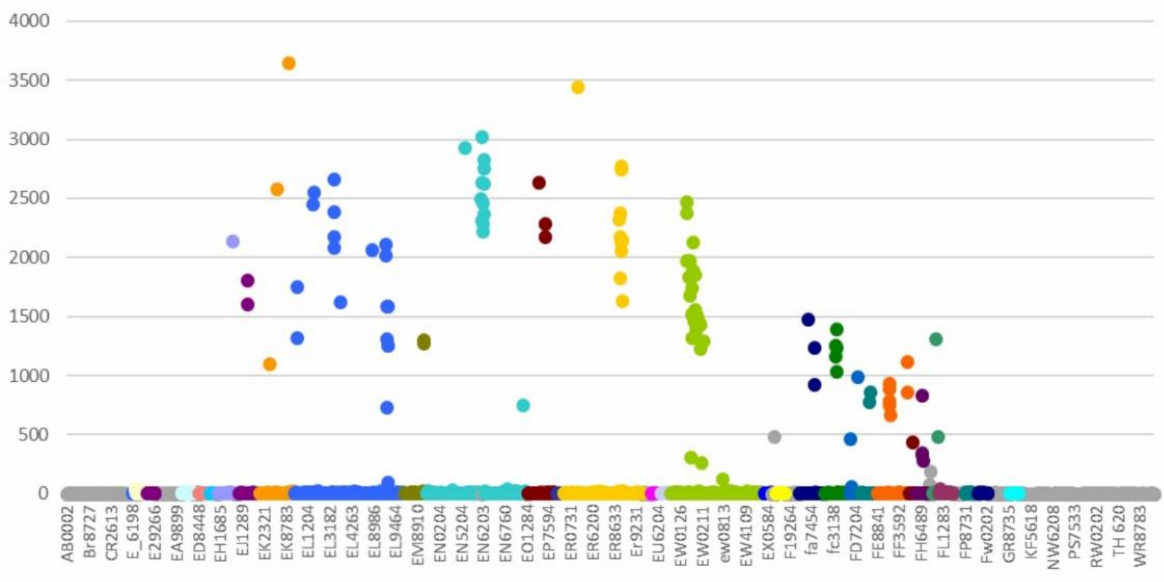
The average adverse reactions per 100,000 doses shipped provides a relative idea of toxicity variation between batches. However this does not take into account the under-reporting factor.

Repeating the Study

I also used the Silhouette method for determining number of clusters. Surprisingly it showed a peak at 7 clusters.



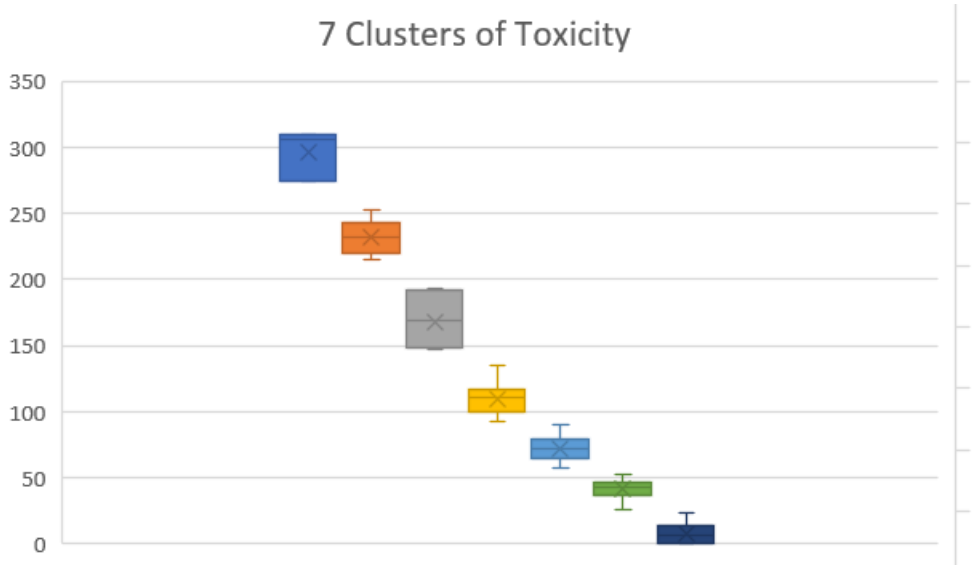
I reasoned that this might be a finer gradation of toxicity categories at the alphabet letter level, as had previously been noticed here –



[Clusters (Howbad.info)]

Results of applying K-means clustering with parameter set to 7 clusters

Here are the toxicities of each of the 7 clusters



And here are the lot numbers corresponding to each of these clusters –

- Cluster 1 Tox = 309-274 Lots = EK
- Cluster 2 Tox = 215-252 Lots = EH, EJ, EL
- Cluster 3 Tox = 146-192 Lots = EL
- Cluster 4 Tox = 92-134 Lots = EL, EM, EN
- Cluster 5 Tox = 57-86 Lots = EN, EP, ER, EW
- Cluster 6 Tox = 25-52 Lots = ER, EW, FA, FC, FD, FE, FF, FH
- Cluster 7 Tox = 0-22 Lots = FD, FE, FF, FG, FH, FJ, FL

When these clusters are arranged in order of their toxicity, they display an ascending alpha-numeric series of lot numbers.

The clusters of higher toxicity do partition at alphabetic boundaries – for example the highest toxicity cluster only has lot numbers starting with EK, and the third most toxic only has lot numbers beginning with EL.

As toxicity descends, there is more alphabetic overlap between clusters, however an ascending alphabetic range is still discernible.

Conclusion

Here I have used k-means clustering to define 7 categories of toxicity. These categories appear spontaneously as groupings within the data. The categories partition at alphabetic boundaries. This will help those who were coerced so they can get a better idea of the short and medium term risks associated with their Pfizer batch code.

The vaccine lots show huge variability in toxicity. This variability appears to be systematic since toxicity varies depending on the lot codes, and this variation is linear as the lot codes ascend alphanumerically. A case of "*Death by alphabet*".

Appendix Code Used in Analysis

```
import os, re, glob
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import folium
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from IPython.display import IFrame
from IPython.display import Image
from tqdm import tqdm
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
toxic_df = pd.read_csv(r"C:\Users\User\Documents\lot-toxicity-2021-data.csv")
```

```
toxicity = pd.DataFrame(columns=['Tox'])
toxicity['Tox'] = toxic_df['Tox']
```

```
scaler = StandardScaler()
X_scaled = scaler.fit(toxicity).transform(toxicity.astype(np.float))
```

```
cluster_range = range(1, 20)  # this is the number of clusters
cluster_errors = []
for num_clusters in cluster_range:
    clusters = KMeans(num_clusters) #
    clusters.fit(X_scaled)
    cluster_errors.append(clusters.inertia_)

clusters_df = pd.DataFrame({"num_clusters":cluster_range,"cluster_errors":cluster_errors})
plt.figure(figsize=(16,6))
plt.plot(clusters_df.num_clusters,clusters_df.cluster_errors,marker= "o");
```

```
import pandas as pd
import os, re, glob

import numpy as np
import matplotlib.pyplot as plt
import folium
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from IPython.display import IFrame
from IPython.display import Image
from tqdm import tqdm
import warnings
import silhouetteplot
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
from sklearn.metrics import silhouette_score
cluster_range = range(2,10)
silhouette_avg = []
for x in cluster_range :

    clusterer = KMeans(n_clusters=x)
    clusterer.fit(X_scaled)
    labels = clusterer.labels_
    score = silhouette_score(X_scaled, labels)
    silhouette_avg.append(score)


plt.plot(cluster_range, silhouette_avg)
plt.show()
```

```
#Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 7, init = 'k-means++', random_state = 10)
y_kmeans = kmeans.fit_predict(X_scaled)

# y_kmeans is an array with a single dimension with a number representi



#beginning of the cluster numbering with 1 instead of 0
y_kmeans1=y_kmeans+1
# New list called cluster
cluster = list(y_kmeans1) # we convert the y_kmeans array to a list, th
# Adding cluster to our data set
toxicity['cluster'] = cluster
toxicity.to_csv("toxicity7cluster.csv", index=False)
```

```
toxicity["cluster"].value_counts()
```

```
kmeans_mean_cluster = pd.DataFrame(round(toxicity.groupby('cluster').mean(),1))
kmeans_mean_cluster
rate = []
for i ,r in kmeans_mean_cluster.iterrows():
    rate1 = r['Tox']

    rate.append(rate1)

kmeans_mean_cluster.head(9)
```