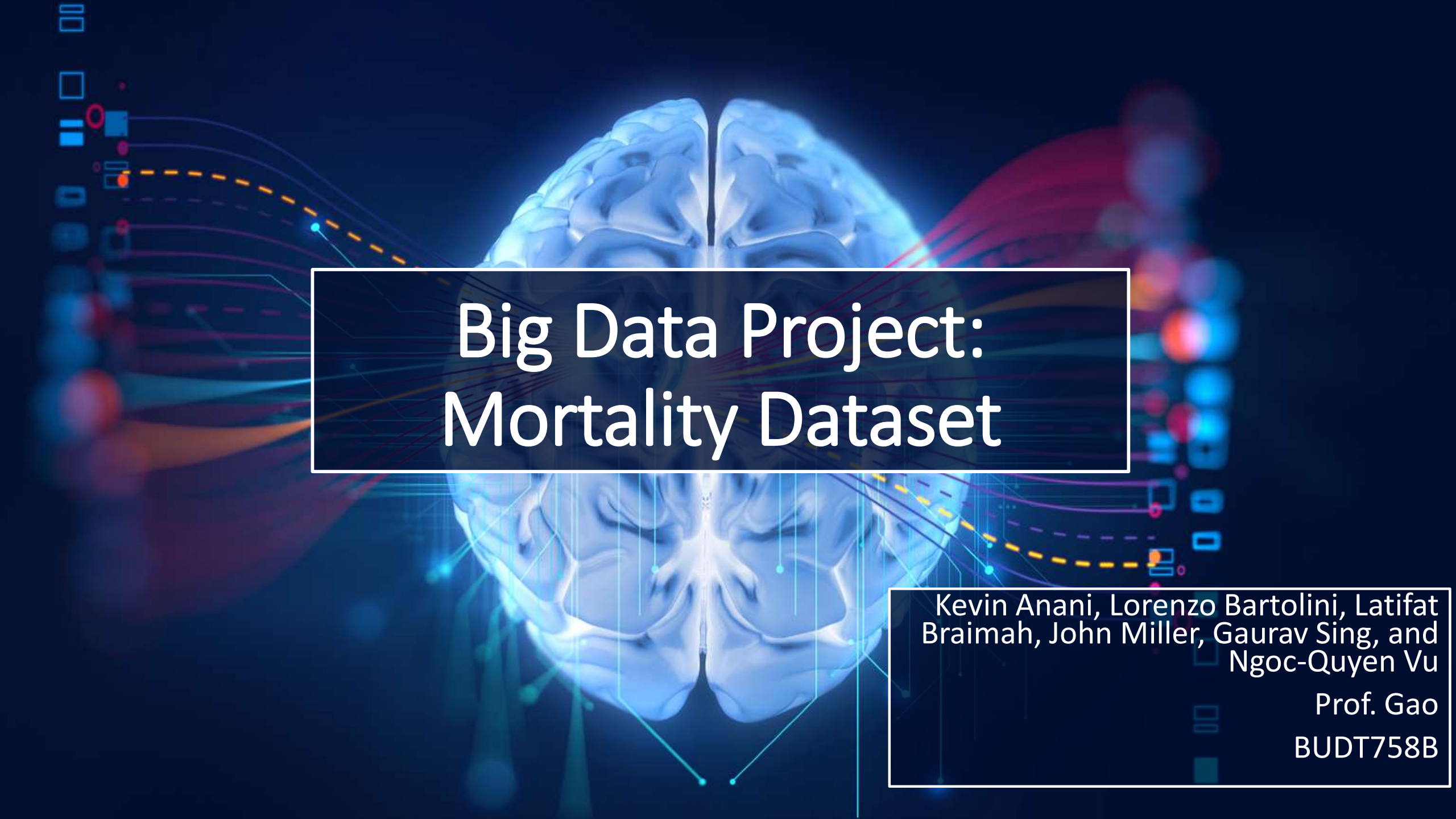


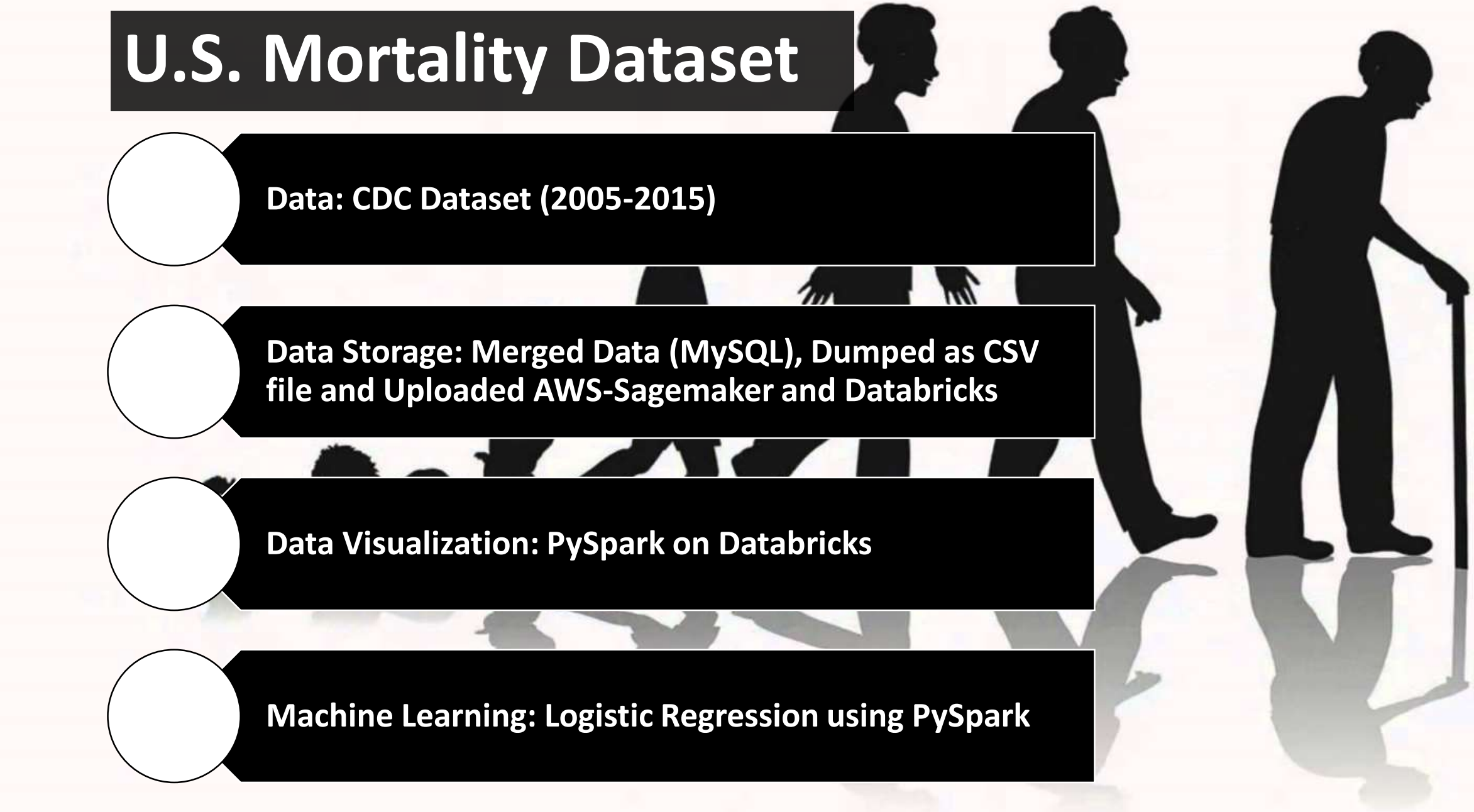
Big Data Project: Mortality Dataset

Kevin Anani, Lorenzo Bartolini, Latifat Braimah, John Miller, Gaurav Sing, and Ngoc-Quyen Vu

Prof. Gao

BUDT758B

U.S. Mortality Dataset



- Data: CDC Dataset (2005-2015)
- Data Storage: Merged Data (MySQL), Dumped as CSV file and Uploaded AWS-Sagemaker and Databricks
- Data Visualization: PySpark on Databricks
- Machine Learning: Logistic Regression using PySpark

Top 20 Causes of Death (Diseases)

```
%sql select  m.icd_code_10th_revision,i.description, count(m.icd_code_10th_revision)
from mergedTable m,icd10 i where i.code=m.icd_code_10th_revision group by i.description,m.icd_code_10th_revision
order by count(m.icd_code_10th_revision) DESC LIMIT 20
```

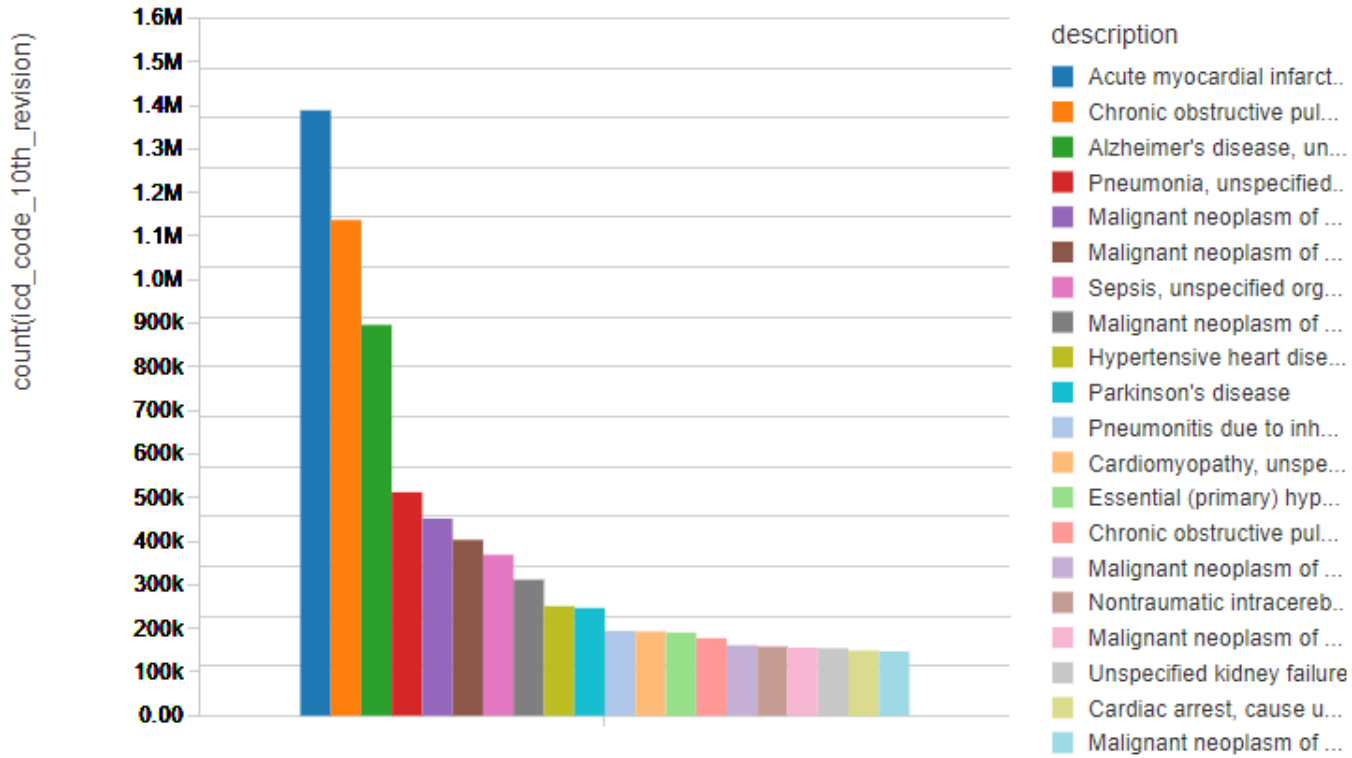
▸ (1) Spark Jobs

icd_code_10th_revision	description	count(icd_code_10th_revision)
I219	Acute myocardial infarction, unspecified	1388327
J449	Chronic obstructive pulmonary disease, unspecified	1136896
G309	Alzheimer's disease, unspecified	896200
J189	Pneumonia, unspecified organism Y	512409
C189	Malignant neoplasm of colon, unspecified	451882
C259	Malignant neoplasm of pancreas, unspecified	403164
A419	Sepsis, unspecified organism	368875
C61	Malignant neoplasm of prostate	311877
I119	Hypertensive heart disease without heart failure	251086
G20	Parkinson's disease	246814
J690	Pneumonitis due to inhalation of food and vomit	194079
I429	Cardiomyopathy, unspecified	192955
I10	Essential (primary) hypertension Y	190549
J440	Chronic obstructive pulmonary disease with acute lower respiratory infection	177701
C679	Malignant neoplasm of bladder, unspecified	161336
I619	Nontraumatic intracerebral hemorrhage, unspecified	158492
C159	Malignant neoplasm of esophagus, unspecified	155951
N19	Unspecified kidney failure	154515
I469	Cardiac arrest, cause unspecified	149181
C719	Malignant neoplasm of brain, unspecified	147226

Top 20 Causes of Death (Diseases)

```
%sql select  m.icd_code_10th_revision,i.description, count(m.icd_code_10th_revision)
from mergedTable m,icd10 i where i.code=m.icd_code_10th_revision group by i.description,m.icd_code_10th_revision
order by count(m.icd_code_10th_revision) DESC LIMIT 20
```

▸ (1) Spark Jobs



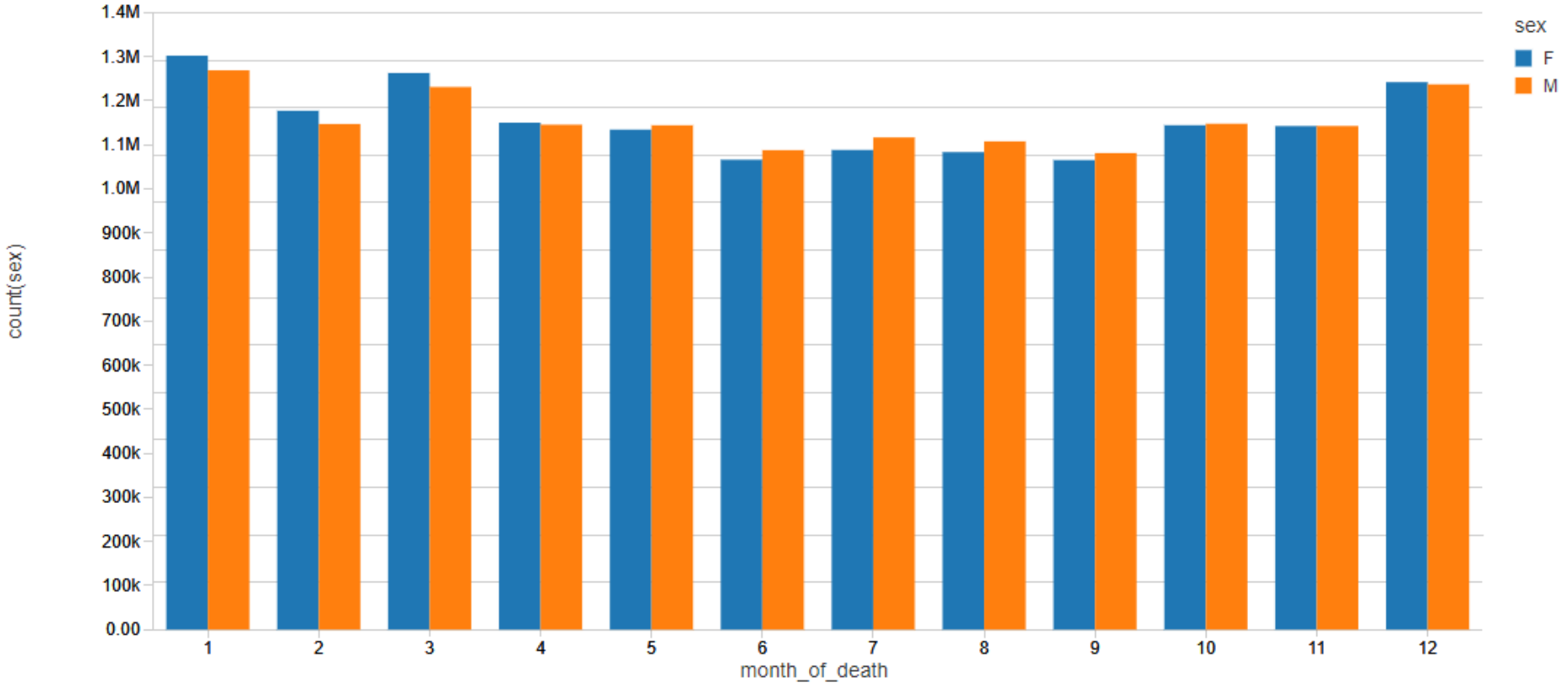
Plot Options...

Command took 5.45 seconds -- by gaurav.singh@rhsmith.umd.edu at 12/10/2018, 1:05:03 PM on big data

Male VS. Female Death rates per month

```
1 %sql select month_of_death,sex, count(sex) from mergedTable group by month_of_death,sex order by month_of_death,sex
```

▸ (1) Spark Jobs



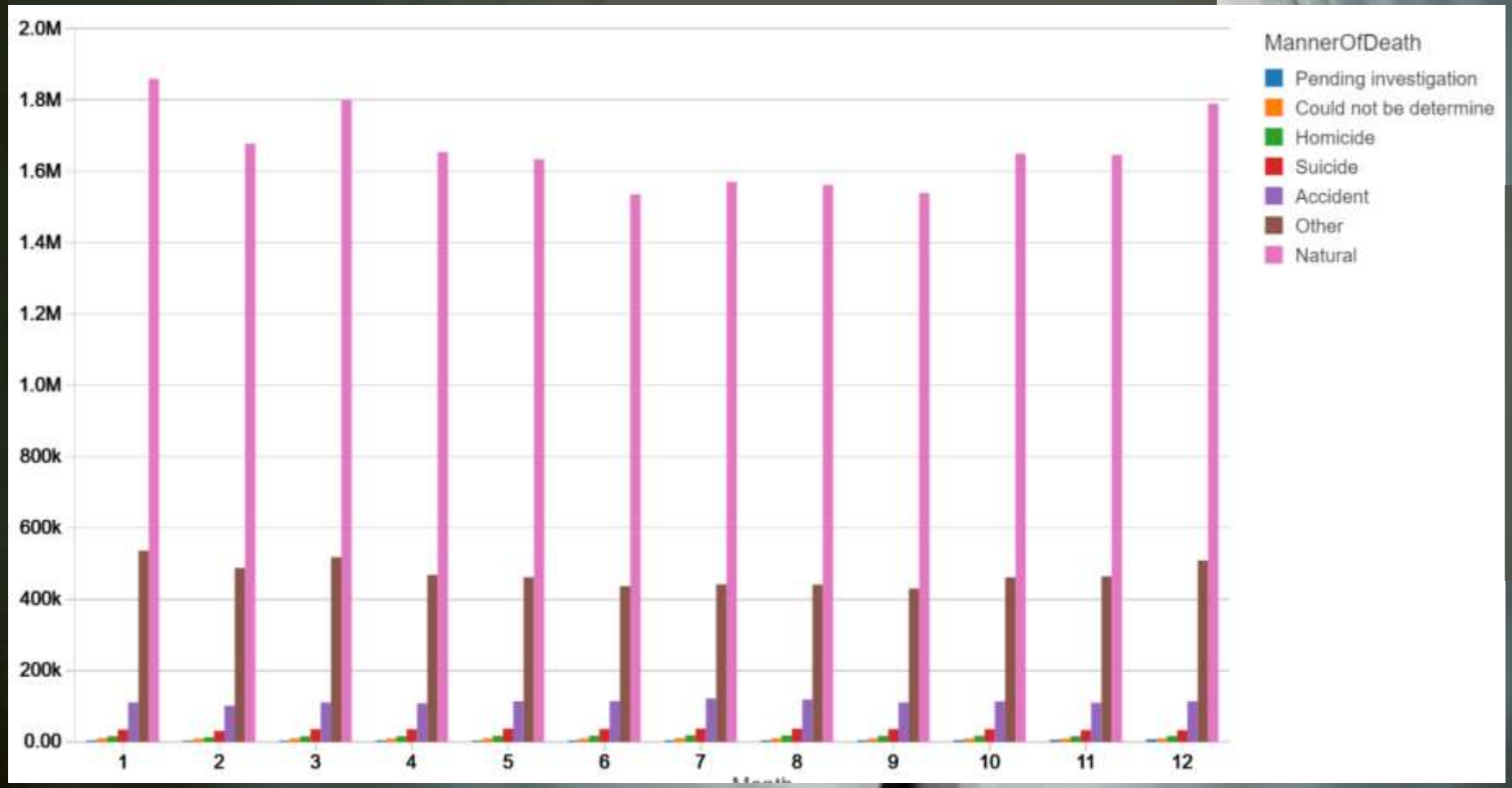
Plot Options...

Command took 4.08 seconds -- by gaurav.singh@rhsmith.umd.edu at 12/10/2018, 1:05:18 AM on big data

Male VS. Female Death rates per month

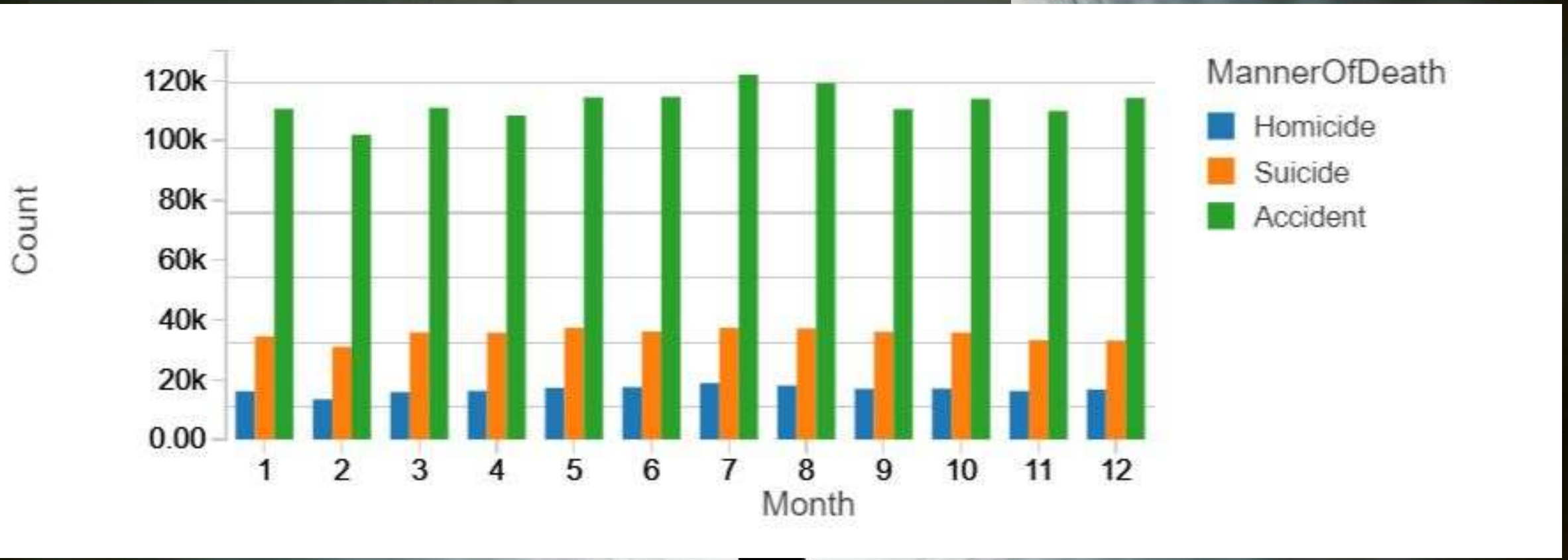
- Winter Months: Death Count for Females is Higher
- Summer Months: Death Count for Males is Higher

Manner of Death per Month



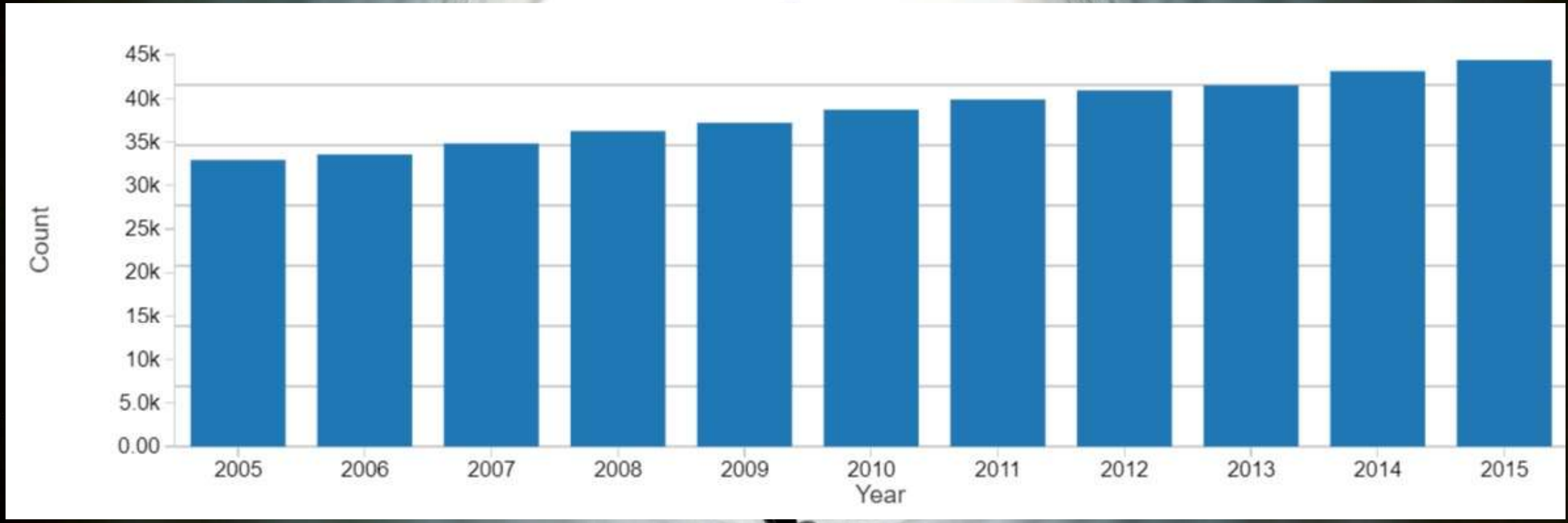
- **Maximum Deaths each Month due to Natural Causes**
- **Winter Months are the deadliest**

Manner of Death per Month



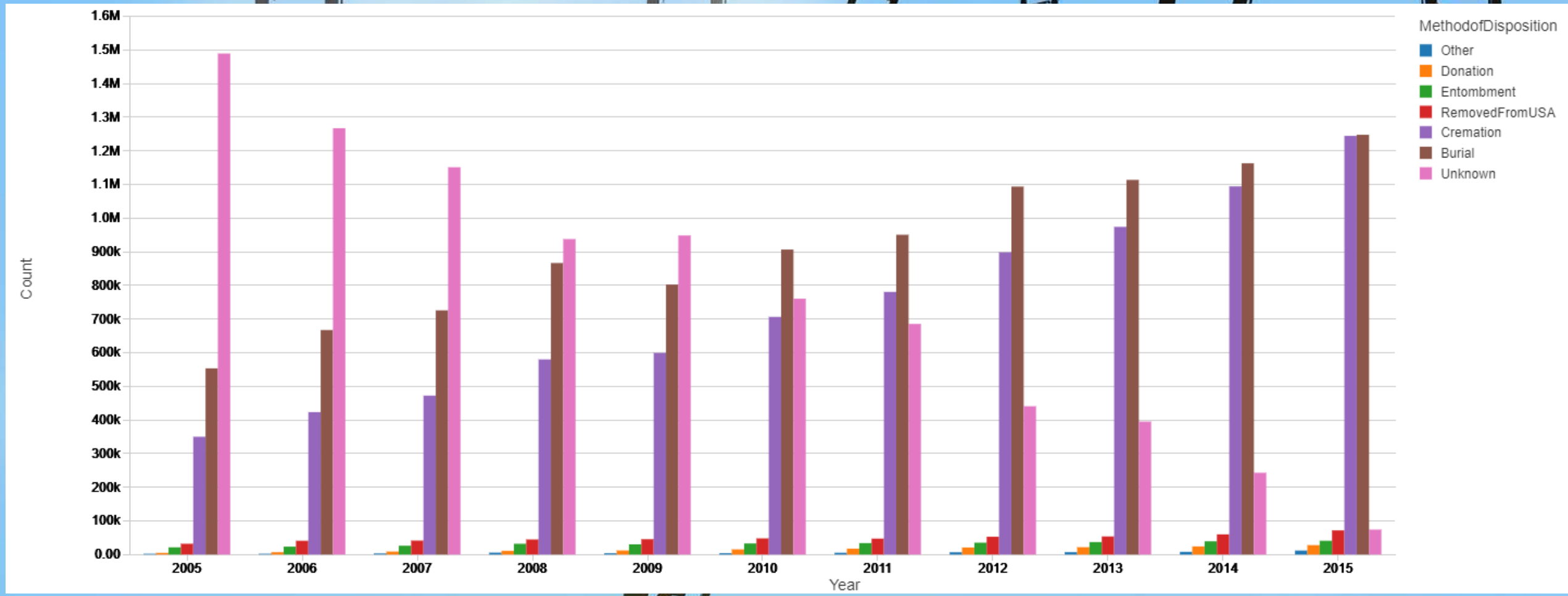
Highest Number of Homicides Committed in the Month of July

Total Suicides Committed 2005-2015



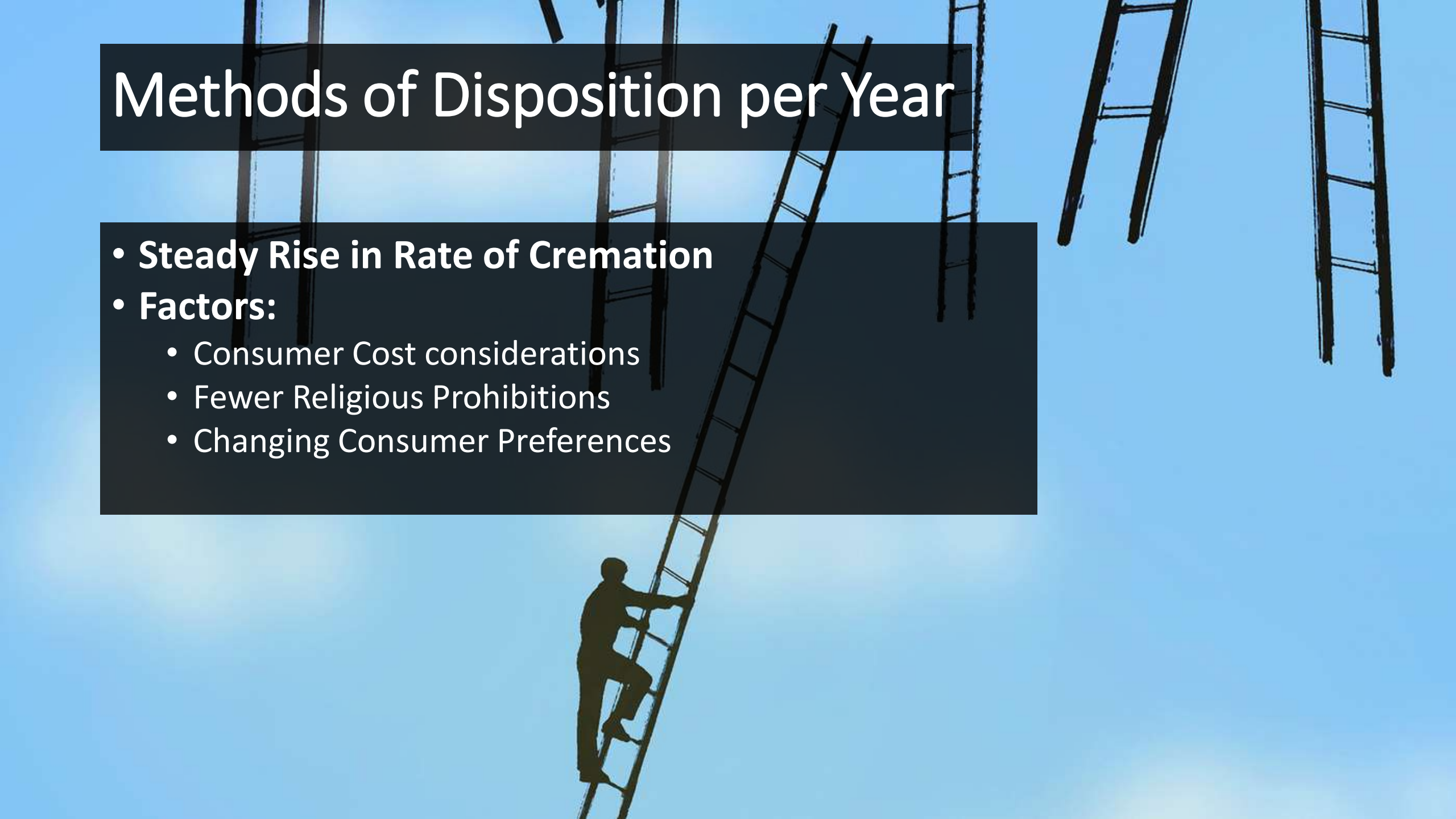
Uniform rise in total number of suicides committed per year

Methods of Disposition per Year



Methods of Disposition per Year

- **Steady Rise in Rate of Cremation**
- **Factors:**
  - Consumer Cost considerations
  - Fewer Religious Prohibitions
  - Changing Consumer Preferences



Predicting Burial vs Cremation

- **Aim**: Predict if a person is to be cremated or buried based on important characteristics (such as sex, age, race, marital status, manner of death)
- **Model used**: Logistic Burial
- **Package**: pyspark.ml
- Used Databricks
- **Pipeline**:
  - String Indexing
  - One Hot encoding
  - Vector Assembling

Model Accuracy

```
#Area under Curve
from pyspark.ml.evaluation import BinaryClassificationEvaluator

# Evaluate model
evaluator = BinaryClassificationEvaluator(rawPredictionCol="rawPrediction")
evaluator.evaluate(predictions)
```

▸ (4) Spark Jobs

```
Out[69]: 0.6653637177018963
```

```
#Accuracy
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Evaluate model
evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", metricName = 'accuracy')
evaluator.evaluate(predictions)
```

▸ (2) Spark Jobs

```
Out[71]: 0.6299281389338351
```


Thank You!