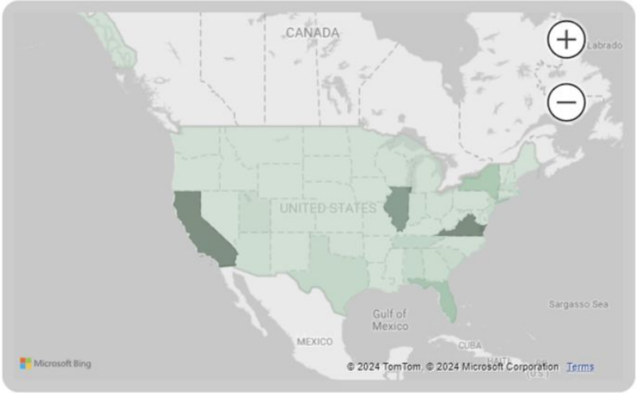
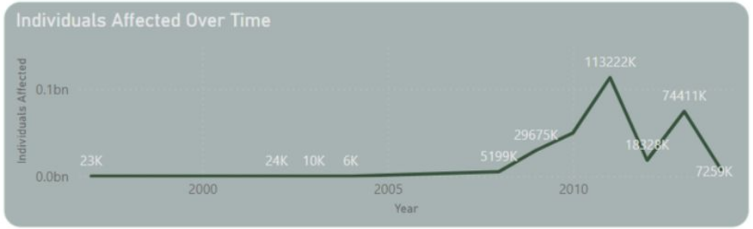
Security Breach Data Analysis

Mayah Bosworth

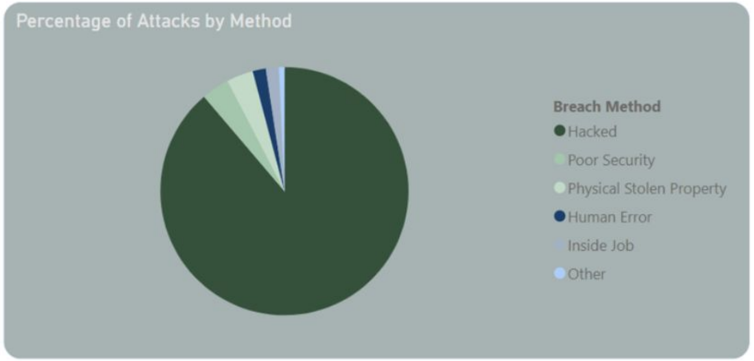
Dashboard



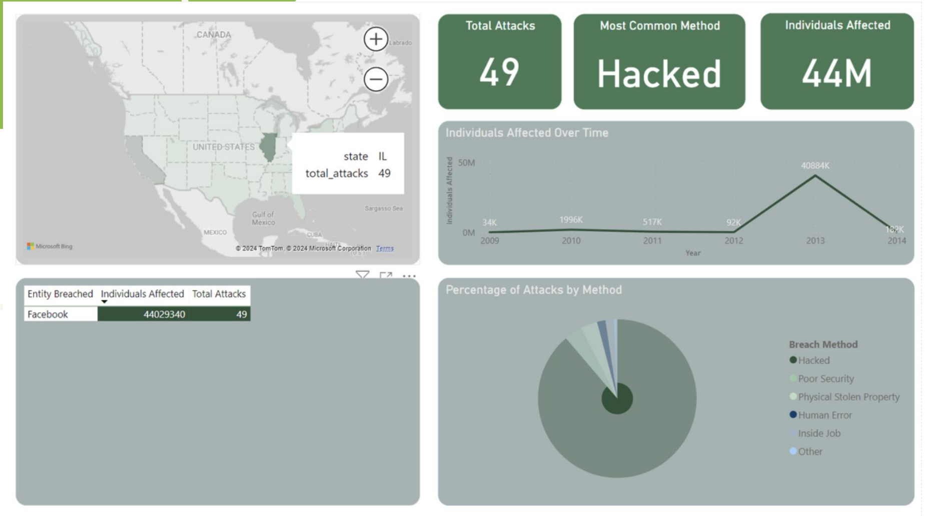
Total Attacks	Most Common Method	Individuals Affected
1407	Hacked	323M



Entity Breached	Individuals Affected	Total Attacks
Facebook	319314300	1060
AOL	30000	3
Citigroup	30000	3
AT&T	20000	2
Bell Canada	20000	2
Bethesda Game Studios	20000	2
Countrywide Financial Corp	20000	2
JP Morgan Chase	20000	2
Marriott International	20000	2
MongoDB	20000	2
Sony Pictures	20000	2
TD Ameritrade	20000	2
Twitch	20000	2



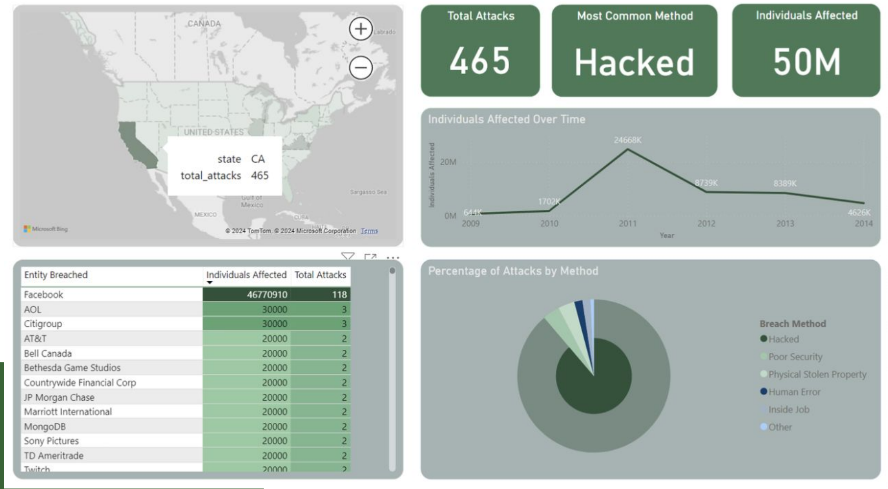
Interactive Elements



Entity Breached	Individuals Affected	Total Attacks
Facebook	44029340	49

Illinois

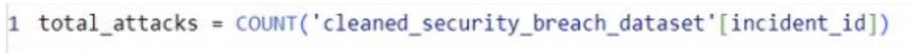
California



Entity Breached	Individuals Affected	Total Attacks
Facebook	46770910	118
AOL	30000	3
Citigroup	30000	3
AT&T	20000	2
Bell Canada	20000	2
Bethesda Game Studios	20000	2
Countrywide Financial Corp	20000	2
JP Morgan Chase	20000	2
Marriott International	20000	2
MongoDB	20000	2
Sony Pictures	20000	2
TD Ameritrade	20000	2
Twitch	20000	2

DAX Formulas

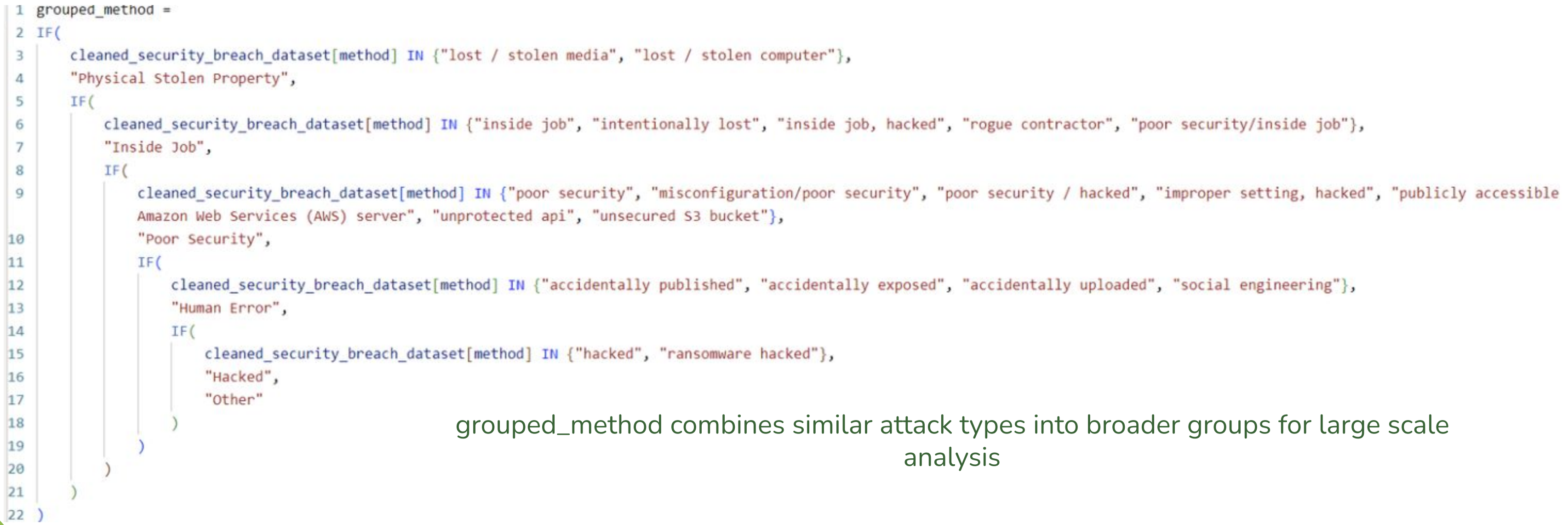
Total Attacks Measure:



```
total_attacks = COUNT('cleaned_security_breach_dataset'[incident_id])
```

total_attacks counts the total number of attacks in the dataset

Grouped Method Calculated Column:

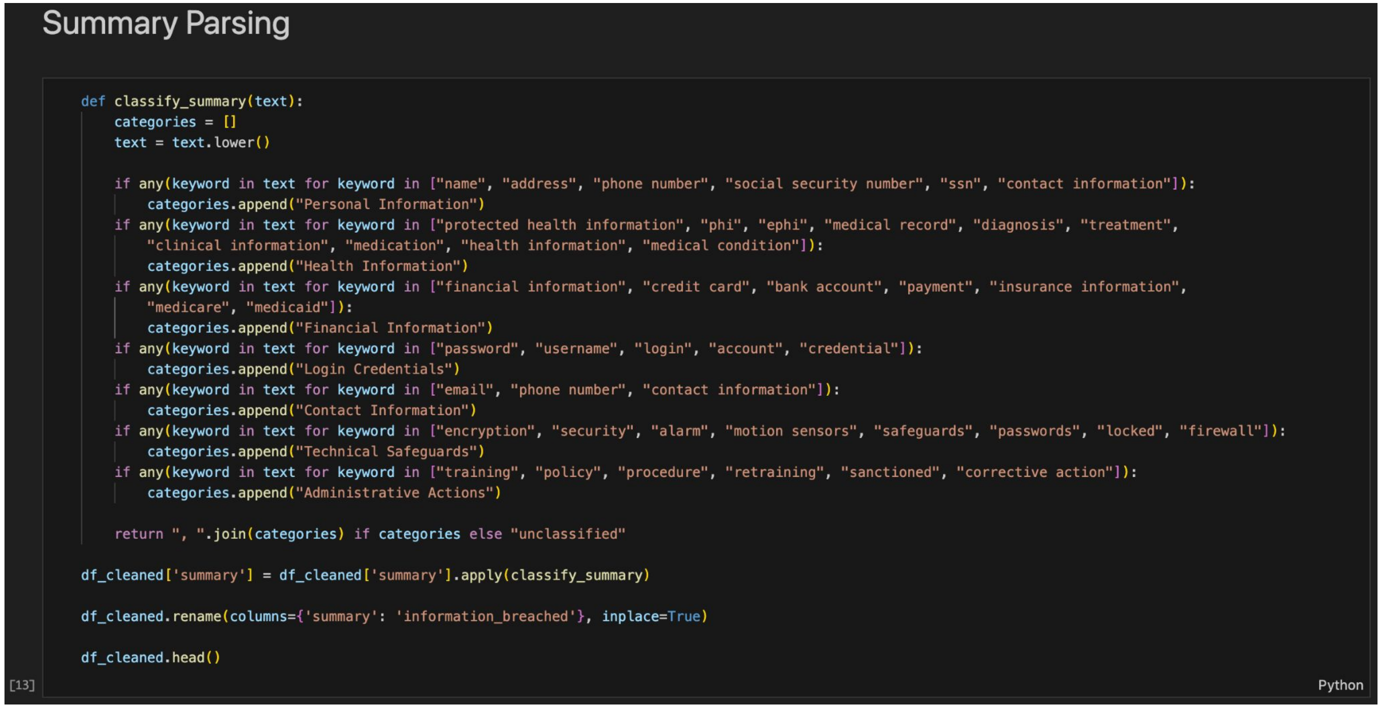


```
grouped_method =
IF(
    cleaned_security_breach_dataset[method] IN {"lost / stolen media", "lost / stolen computer"},
    "Physical Stolen Property",
    IF(
        cleaned_security_breach_dataset[method] IN {"inside job", "intentionally lost", "inside job, hacked", "rogue contractor", "poor security/inside job"},
        "Inside Job",
        IF(
            cleaned_security_breach_dataset[method] IN {"poor security", "misconfiguration/poor security", "poor security / hacked", "improper setting, hacked", "publicly accessible
            Amazon Web Services (AWS) server", "unprotected api", "unsecured S3 bucket"},
            "Poor Security",
            IF(
                cleaned_security_breach_dataset[method] IN {"accidentally published", "accidentally exposed", "accidentally uploaded", "social engineering"},
                "Human Error",
                IF(
                    cleaned_security_breach_dataset[method] IN {"hacked", "ransomware hacked"},
                    "Hacked",
                    "Other"
                )
            )
        )
    )
)
```

grouped_method combines similar attack types into broader groups for large scale analysis

Python Data Preparation

This code categorizes text based on specific keywords (like Personal or Health Information) and updates a DataFrame column with these categories. If no match is found, it labels the text as "unclassified" and renames the column. Because the summary column originally consisted of paragraphs explaining the incident, implementing this new column allows for more efficient analysis.



Summary Parsing

```python
def classify_summary(text):
    categories = []
    text = text.lower()

    if any(keyword in text for keyword in ["name", "address", "phone number", "social security number", "ssn", "contact information"]):
        categories.append("Personal Information")
    if any(keyword in text for keyword in ["protected health information", "phi", "ephi", "medical record", "diagnosis", "treatment",
        "clinical information", "medication", "health information", "medical condition"]):
        categories.append("Health Information")
    if any(keyword in text for keyword in ["financial information", "credit card", "bank account", "payment", "insurance information",
        "medicare", "medicaid"]):
        categories.append("Financial Information")
    if any(keyword in text for keyword in ["password", "username", "login", "account", "credential"]):
        categories.append("Login Credentials")
    if any(keyword in text for keyword in ["email", "phone number", "contact information"]):
        categories.append("Contact Information")
    if any(keyword in text for keyword in ["encryption", "security", "alarm", "motion sensors", "safeguards", "passwords", "locked", "firewall"]):
        categories.append("Technical Safeguards")
    if any(keyword in text for keyword in ["training", "policy", "procedure", "retraining", "sanctioned", "corrective action"]):
        categories.append("Administrative Actions")

    return ", ".join(categories) if categories else "unclassified"

df_cleaned['summary'] = df_cleaned['summary'].apply(classify_summary)

df_cleaned.rename(columns={'summary': 'information_breached'}, inplace=True)

df_cleaned.head()
```

[13] Python