safety_analysis.ipynb

Data Source

FINISHED

The safety dataset is from IRS SCHOOL SAFETY AND THE EDUCATIONAL CLIMATE (https://www.p12.nysed.gov/irs/school_safety/school_safety_data_reporting.htmll).

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:21 AM.

Data Analysis on Safety Data

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:21 AM.

FINISHED

```
val safetyIssueColumns = Seq(
  "Homocide",
  "Sexual_Offense",
  "Assault",
  "Weapons_Possession",
  "Dignity Act-Excluding_Cyberbullying",
  "Dignity Act-Cyberbullying",
  "Bomb_Threat",
  "False_Alarm",
  "Drugs",
  "Alcohol"
)
```

FINISHED

```
val nyccounties = "('QUEENS',  'BRONX', 'RICHMOND', 'NEW YORK', 'KINGS')"
```

```
nyccounties: String = ('QUEENS',  'BRONX', 'RICHMOND', 'NEW YORK', 'KINGS')
```

Took 0 sec. Last updated by yz6956_nyu_edu at December 13 2024, 1:16:30 PM.

Per District

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:22 AM.

In NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:22 AM.

SPARK JOB FINISHED

```
import org.apache.spark.sql.DataFrame

var totalNYCCountyDF: DataFrame = spark.emptyDataFrame

var count: Long = 0

for (year <- 2018 to 2023) {

    val filePath = s"project/cleaned_data/safety$year.csv"
    val safetydf = spark.read
        .option("multiLine", "true")
        .option("inferSchema", "true")
        .option("escape", "\"")
        .option("header", true)
        .csv(filePath)

    safetydf.createOrReplaceTempView(s"safety$year")

    count = safetydf.count()

    var countydf = spark.sql(s"""
        select `County`, ${safetyIssueColumns.map(col => s"avg(`$col`) as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
        from safety$year
        where `County` in ${nyccounties}
        group by `County`
    """).withColumn("Year", lit(year))

    val columns = Seq("Year") ++ countydf.columns.slice(0, 11)
    var countydfNew = countydf.select(columns.map(countydf.col): _*)

    totalNYCCountyDF = if (totalNYCCountyDF.isEmpty) countydfNew else totalNYCCountyDF.union(countydfNew)
}
```

```
z.show(totalNYCCountyDF)
```

settings

safety_analysis.ipynb

Available Fields

Year | County | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying | Avg_Bomb_Threat | Avg_False_Alarm | Avg_Drugs | Avg_Alcohol

keys

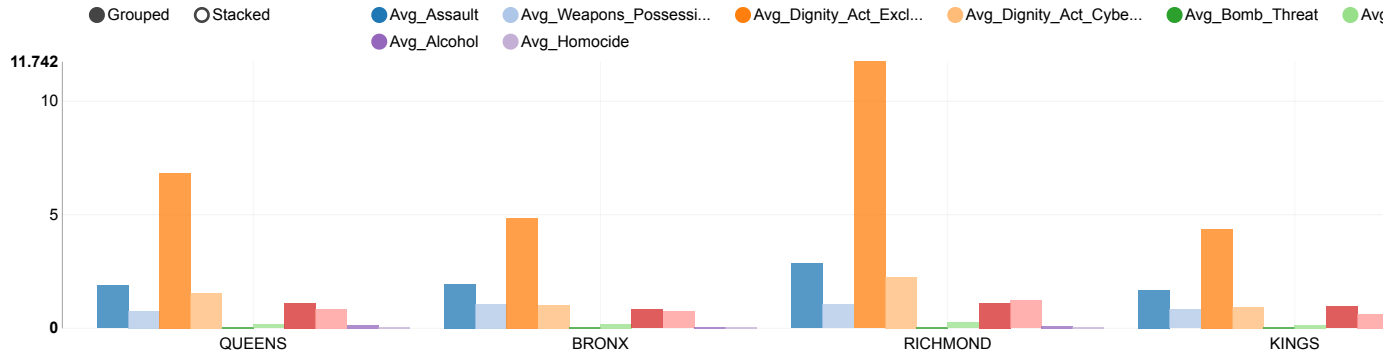
County ✖

groups

values

Avg_Assault **AVG** ✖
Avg_Weapons_Possession **AVG** ✖
Avg_Dignity_Act_Excluding_Cybe... **AVG** ✖
Avg_Dignity_Act_Cyberbullying **AVG** ✖
Avg_Bomb_Threat **AVG** ✖
Avg_False_Alarm **AVG** ✖

xAxis : Default | Rotate | Hide




```
import org.apache.spark.sql.DataFrame
totalNYCCountyDF: org.apache.spark.sql.DataFrame = [Year: int, County: string ... 10 more fields]
count: Long = 4730
```

Took 5 sec. Last updated by anonymous at December 10 2024, 11:39:16 AM.

```
val avgSafetyIssueColumns = totalNYCCountyDF.columns.filter(col => col != "Year" && col != "County")
```

FINISHED

```
avgSafetyIssueColumns: Array[String] = Array(Avg_Homocide, Avg_Sexual_Offense, Avg_Assault, Avg_Weapons_Possession, Avg_Dignity_Act_Excluding_Cyb
erbullying, Avg_Dignity_Act_Cyberbullying, Avg_Bomb_Threat, Avg_False_Alarm, Avg_Drugs, Avg_Alcohol)
```

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:28 AM.

```
totalNYCCountyDF.createOrReplaceTempView("totalNYCCounty")

val NYCcountyYoYDF = spark.sql(s"""
  select
```

SPARK JOB FINISHED

```
    cur.County,
    cur.Year,
    ${avgSafetyIssueColumns.map{col =>
        val name = col.replace(" ", "_").replace("-", "_")
        s"round((cur.`$col` - prev.`$col`) / prev.`$col` * 100, 2) AS `${name}_YoY (%)`"
    }.mkString(",\n    ")}
  from
    totalNYCCounty cur
  join
    totalNYCCounty prev
  on
    cur.Year = prev.Year + 1
    and cur.County = prev.County
""")

z.show(NYCcountyYoYDF)
```

safety_analysis.ipynb

settings

Table Options

Name	Value
useFilter	☐
showPagination	☐
showAggregationFooter	☐

County	Year	Avg_Homocide_YoY (%)	Avg_Sexual_Offense_YoY (%)	Avg_Assault_YoY (%)	Avg_Weapons
RICHMOND	2019	null	-42.82	-21.82	-13.62
BROOKLYN	2019	null	-28.61	-18.82	-9.0
QUEENS	2019	null	-40.94	-14.16	-6.49
BRONX	2019	null	-29.24	-39.49	-8.29
MANHATTAN	2019	null	-28.94	-39.27	-10.41
BROOKLYN	2020	null	-41.57	-28.91	-21.62
QUEENS	2020	null	-31.85	-41.01	-18.91
BRONX	2020	null	-47.38	-45.92	-19.43

```
NYCcountyYoYDF: org.apache.spark.sql.DataFrame = [County: string, Year: int ... 10 more fields]
```

Took 2 sec. Last updated by anonymous at December 10 2024, 11:18:30 AM. (outdated)

Outside of NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:30 AM.

SPARK JOB FINISHED

```
var totalOutCountyDF: DataFrame = spark.emptyDataFrame

for (year <- 2018 to 2023) {
    var countydf = spark.sql(s"""
        select `County`, ${safetyIssueColumns.map(col => s"avg(`$col`) as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
        from safety$year
        where `County` not in ${nyccounties}
        group by `County`
    """).withColumn("Year", lit(year))

    val columns = Seq("Year") ++ countydf.columns.slice(0, 11)
    var countydfNew = countydf.select(columns.map(countydf.col): _*)

    totalOutCountyDF = if (totalOutCountyDF.isEmpty) countydfNew else totalOutCountyDF.union(countydfNew)
}

z.show(totalOutCountyDF)
```

settings

Available Fields

Year | County | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying

Avg_Bomb_Threat Avg_False_Alarm Avg_Drugs Avg_Alcohol

safety_analysis.ipynb

keys

County ✖

groups

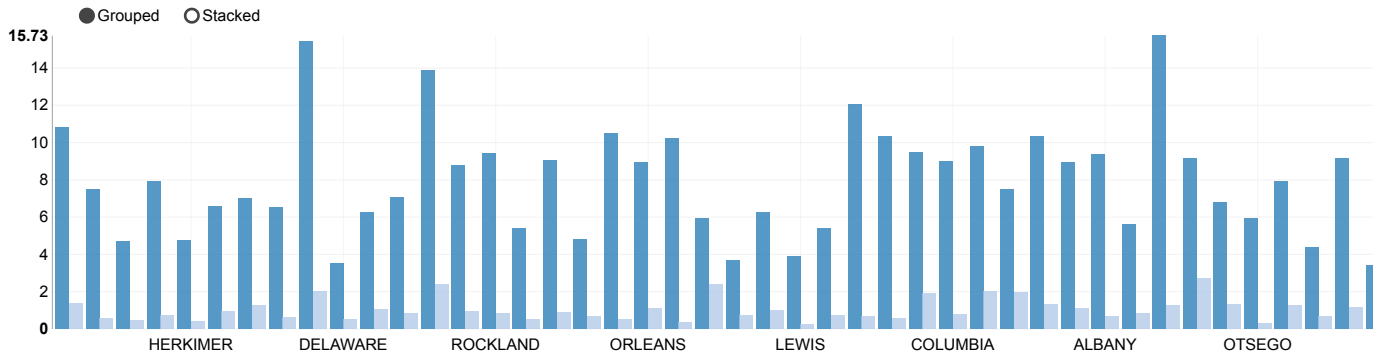
values

Avg_Drugs **SUM** ✖

Avg_Sexual_Offense **SUM** ✖

xAxis :

Default Rotate Hide




```
totalOutCountyDF: org.apache.spark.sql.DataFrame = [Year: int, County: string ... 10 more fields]
```

Took 3 sec. Last updated by anonymous at December 10 2024, 11:40:16 AM. (outdated)

```
val distinctCountiesDF = totalOutCountyDF.select("County").distinct()
distinctCountiesDF.show(Int.MaxValue, truncate = false)
```

SPARK JOB FINISHED

```
|CLINTON       |
|SUFFOLK       |
|ST. LAWRENCE  |
|STEUBEN       |
|TIOGA         |
|MONROE        |
|ONONDAGA      |
|BROOME        |
|SARATOGA      |
|GREENE        |
|MONTGOMERY    |
|SCHOHARIE     |
|ERIE          |
|FRANKLIN      |
|ONTARIO       |
|HAMILTON      |
|SAINT LAWRENCE|
+--------------+
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:35 AM.

safety_analysis.ipynb

SPARK JOB FINISHED

```
// totalOutCountyDF.createOrReplaceTempView("totalOutCounty")

val topOutCountyDF = totalOutCountyDF.withColumn(
  "Sum_Safety_Issues",
  avgSafetyIssuesColumns.map(colName => col(colName)).reduce(_ + _)
)

val avgSafetyIssuesDF = topOutCountyDF
  .groupBy("County")
  .agg(avg("Sum_Safety_Issues").as("Avg_Sum_Safety_Issues"))
  .orderBy(desc("Avg_Sum_Safety_Issues"))

z.show(avgSafetyIssuesDF)
```

settings

Available Fields

County Avg_Sum_Safety_Issues

keys

County ✖

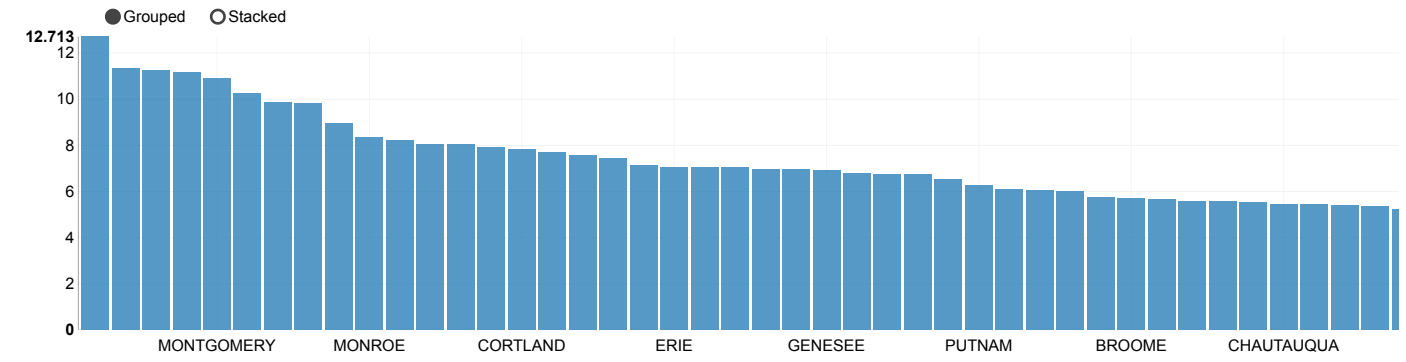
groups

values

Avg_Sum_Safety_Issues SUM ✖

xAxis :

Default Rotate Hide




```
topOutCountyDF: org.apache.spark.sql.DataFrame = [Year: int, County: string ... 11 more fields]
avgSafetyIssuesDF: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [County: string, Avg_Sum_Safety_Issues: double]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:43:16 AM. (outdated)

SPARK JOB FINISHED

```
avgSafetyIssuesDF.createOrReplaceTempView("avgSafetyIssues")

val temp2 = spark.sql("""
    select *
```

```
    from avgSafetyIssues
    where County = "NEW YORK"
""")

z.show(temp2)
```

safety_analysis.ipynb

settings

County	Avg_Sum_Safety_Issues

```
temp2: org.apache.spark.sql.DataFrame = [County: string, Avg_Sum_Safety_Issues: double]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:37 AM.

```
totalOutCountyDF.createOrReplaceTempView("totalOutCounty")

val outCountyYoYDF = spark.sql(s"""
  select
    cur.County,
    cur.Year,
    ${avgSafetyIssueColumns.map{col =>
        val name = col.replace(" ", "_").replace("-", "_")
        s"round((cur.`$col` - prev.`$col`) / prev.`$col` * 100, 2) AS `${name}_YoY (%)`"
    }.mkString(",\n    ")}
  from
    totalOutCounty cur
  join
    totalOutCounty prev
  on
    cur.Year = prev.Year + 1
    and cur.County = prev.County
""")

z.show(outCountyYoYDF)
```

SPARK JOB FINISHED

settings

County	Year	Avg_Homocide_YoY (%)	Avg_Sexual_Offense_YoY (%)	Avg_Assault_YoY (%)	Avg_Weapons
SENECA	2019	null	0.0	-57.89	150.0
WASHINGTON	2019	null	-42.86	-50.0	-80.0
WYOMING	2019	null	-77.08	3.12	22.22
RENSSELAER	2019	null	11.11	-76.5	42.86
HERKIMER	2019	null	0.0	85.71	-14.29
NASSAU	2019	null	35.86	-12.85	8.47
CAYUGA	2019	null	-20.0	-47.37	27.27
CHENANGO	2019	null	-85.71	-40.0	-27.27

```
outCountyYoYDF: org.apache.spark.sql.DataFrame = [County: string, Year: int ... 10 more fields]
```

Took 2 sec. Last updated by anonymous at December 10 2024, 11:18:39 AM.

Per School

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:39 AM.

```
// concat all the cleaned datasets views from 2018 to 2023 in
val allDF = spark.sql("""
  select * , 2018 as Year from safety2018
```

SPARK JOB (http://nyu-dataproc-sw-56rz.c.hpc-dataproc-19b8.internal:37287/jobs/job?id=382) FINISHED

```
  union all
  select * , 2019 as Year from safety2019
  union all
  select * , 2020 as Year from safety2020
  union all
  select * , 2021 as Year from safety2021
  union all
  select * , 2022 as Year from safety2022
  union all
  select * , 2023 as Year from safety2023
""")

allDF.createOrReplaceTempView("all")
```

safety_analysis.ipynb

settings

County	District	School_Name	BEDS_Code	School_Type	Enrollment	Sexual_Offense
ALBANY	ALBANY	THOMAS S O'BRIEN ACAD OF SCI & TECH	10100010028	Public	437	0
ALBANY	ALBANY	GIFFEN MEMORIAL ELEMENTARY SCHOOL	10100010029	Public	514	0
ALBANY	ALBANY	WILLIAM S HACKETT MIDDLE SCHOOL	10100010030	Public	676	0
ALBANY	ALBANY	ALBANY HIGH SCHOOL	10100010034	Public	2621	0

Output is truncated to 1000 rows. Learn more about **zeppelin.spark.maxResult**

```
allDF: org.apache.spark.sql.DataFrame = [County: string, District: string ... 15 more fields]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:40 AM.

In NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:40 AM.

```
val NYCschoolDF = spark.sql(s"""
    select BEDS_Code, County, ${safetyIssueColumns.map(col => s"avg(`$col`) as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
    from all
    where County in ${nyccounties}
    group by BEDS_Code, County
""").withColumn("Sum_Avg_Safety_Issues", avgSafetyIssueColumns.map(col).reduce(_ + _))

z.show(NYCschoolDF)
```

SPARK JOB FINISHED

settings

BEDS_Code	County	Avg_Homocide	Avg_Sexual_Offense	Avg_Assault	Avg_Weapons_Possession	Avg_D
660803020004	RICHMOND	0.0	0.0	0.0	0.0	2.0
342500011460	QUEENS	0.3333333333333333	0.6666666666666666	1.5	3.3333333333333335	3.1666
331700010353	BROOKLYN	0.0	0.0	1.5	1.25	0.5
332300011644	BROOKLYN	0.0	0.25	0.25	1.25	0.0
342700010210	QUEENS	0.3333333333333333	4.666666666666667	10.0	4.666666666666667	14.666
353100010032	RICHMOND	0.0	1.6666666666666667	4.0	0.5	4.1666
307500013140	BROOKLYN	0.0	2.6666666666666665	19.333333333333332	4.0	6.0
307500014277	QUEENS	0.0	0.2	2.6	0.0	3.8

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT**

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:41 AM.

```
NYCschoolDF.createOrReplaceTempView("NYCschool")
```

SPARK JOB FINISHED

```
val topNYCSchoolDF = spark.sql("""
    select *
    from NYCschool
    order by Sum_Avg_Safety_Issues desc
    limit 10
""")

z.show(topNYCSchoolDF)
```

safety_analysis.ipynb

settings

Available Fields

BEDS_Code | County | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying | Avg_Bomb_Threat | Avg_False_Alarm | Avg_Drugs | Avg_Alcohol | Sum_Avg_Safety_Issues

keys

BEDS_Code ✖

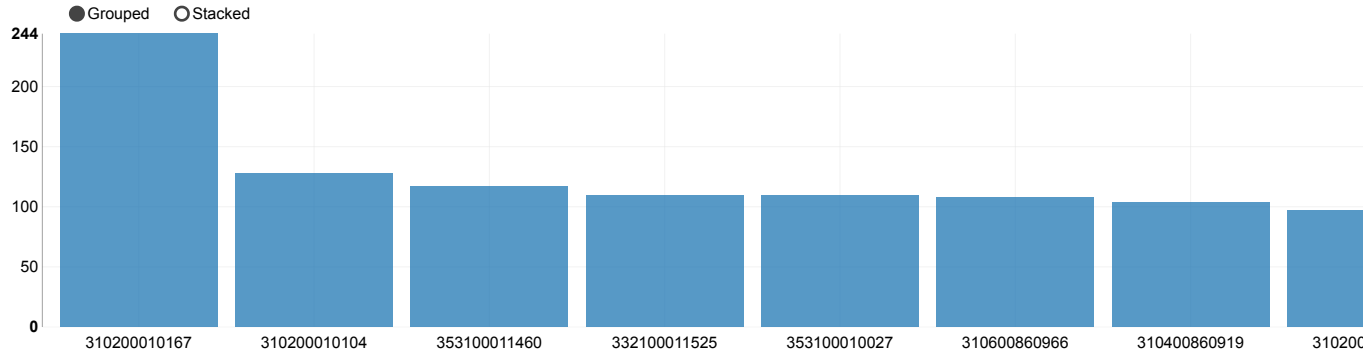
groups

values

Sum_Avg_Safety_Issues **SUM** ✖

xAxis :

Default | Rotate | Hide




```
topNYCSchoolDF: org.apache.spark.sql.DataFrame = [BEDS_Code: bigint, County: string ... 11 more fields]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:42 AM. (outdated)

Outside of NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:42 AM.

```
val outSchoolDF = spark.sql(s"""
```

SPARK JOB FINISHED

```
    select BEDS_Code, County, ${safetyIssueColumns.map(col => s"avg(`$col`) as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
    from all
    where County not in ${nyccounties}
    group by BEDS_Code, County
""").withColumn("Sum_Avg_Safety_Issues", avgSafetyIssueColumns.map(col).reduce(_ + _))
z.show(outSchoolDF)
```

safety_analysis.ipynb

settings

Table Options

Name	Value
useFilter	☐
showPagination	☐
showAggregationFooter	☐

BEDS_Code	County	Avg_Homocide	Avg_Sexual_Offense	Avg_Assault	Avg_Weapons_Possession	Avg_D
131701060004	DUTCHESS	0.0	0.5	0.0	0.0	3.0
660803020002	WESTCHESTER	0.0	0.0	0.16666666666666666	0.5	1.8333
421501060014	ONONDAGA	0.0	0.0	0.5	0.5	0.8333
471400010002	OTSEGO	0.6666666666666666	0.0	0.0	0.3333333333333333	4.0
541001040001	SCHOHARIE	0.0	0.0	0.0	1.1666666666666667	2.5
70901060010	CHEMUNG	0.6666666666666666	0.5	0.16666666666666666	0.16666666666666666	0.3333

Output is truncated to 102400 bytes. Learn more about **ZEPPELIN_INTERPRETER_OUTPUT_LIMIT**

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:43 AM. (outdated)

```
outSchoolDF.createOrReplaceTempView("outSchool")

val topOutSchoolDF = spark.sql("""
    select *
    from outSchool
    order by Sum_Avg_Safety_Issues desc
    limit 10
""")

z.show(topOutSchoolDF)
```

SPARK JOB FINISHED

settings

Available Fields

BEDS_Code | County | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying | Avg_Bomb_Threat | Avg_False_Alarm | Avg_Drugs | Avg_Alcohol | Sum_Avg_Safety_Issues

keys

BEDS_Code ✖

groups

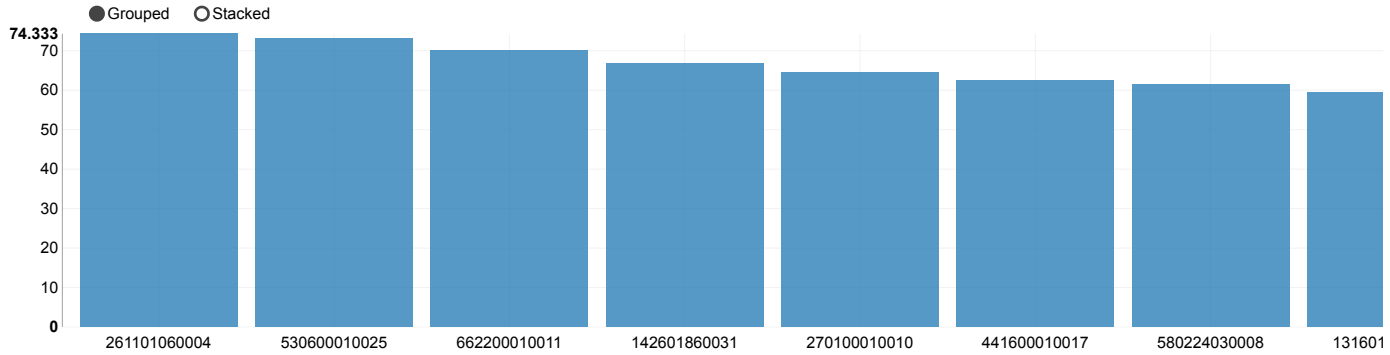
values

Sum_Avg_Safety_Issues_SUM ✖

safety_analysis.ipynb

xAxis :

Default | Rotate | Hide




```
topOutSchoolDF: org.apache.spark.sql.DataFrame = [BEDS_Code: bigint, County: string ... 11 more fields]
```

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:44 AM. (outdated)

Per Safety Issue

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:45 AM.

In NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:45 AM.

SPARK JOB FINISHED

```
var avgNYCSafetyIssueDF: DataFrame = spark.emptyDataFrame

for (year <- 2018 to 2023) {

    // sum the number of each violations / number of schools
    var avg = spark.sql(s"""
        select ${safetyIssueColumns.map(col => s"sum(`$col`)/$count as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
        from safety${year}
        where County in $nyccounties
    """).withColumn("Year", lit(year))

    val columns = Seq("Year") ++ avg.columns.slice(0, 10)
    var avgNew = avg.select(columns.map(avg.col): _*)

    avgNYCSafetyIssueDF = if (avgNYCSafetyIssueDF.isEmpty) avgNew else avgNYCSafetyIssueDF.union(avgNew)
}

z.show(avgNYCSafetyIssueDF)
```

settings

Available Fields

Year | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying | Avg_Bomb_Threat | Avg_False_Alarm | Avg_Drugs | Avg_Alcohol

keys

Year ✖

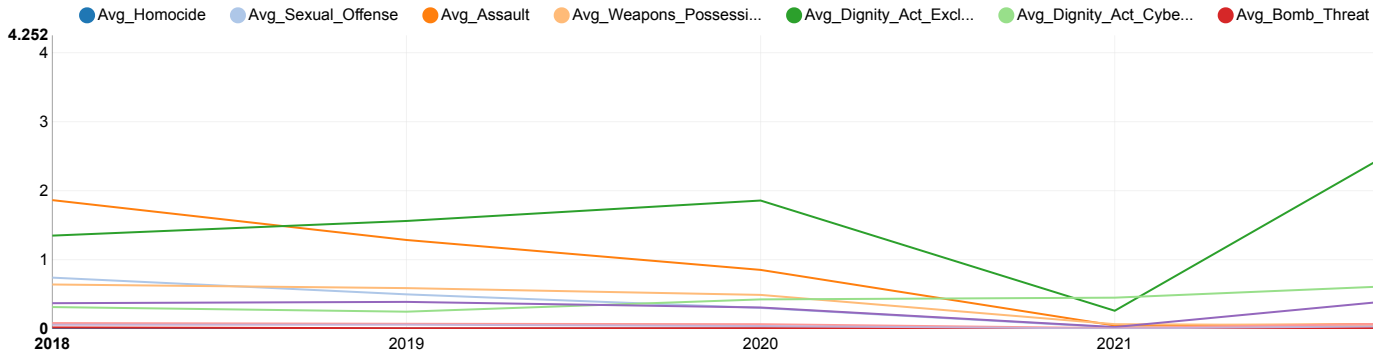
safety_analysis.ipynb

groups

values

Avg_Homocide **SUM** ✖
Avg_Sexual_Offense **SUM** ✖
Avg_Assault **SUM** ✖
Avg_Weapons_Possession **SUM** ✖
Avg_Dignity_Act_Excluding_Cybe... **SUM** ✖
Avg_Dignity_Act_Cyberbullying **SUM** ✖

☐ **force Y to 0**
☐ **zoom**
☐ **Date format**
xAxis :
Default | Rotate | Hide




```
avgNYCSafetyIssueDF: org.apache.spark.sql.DataFrame = [Year: int, Avg_Homocide: double ... 9 more fields]
```

Took 2 sec. Last updated by anonymous at December 10 2024, 11:18:47 AM.

SPARK JOB FINISHED

```
avgNYCSafetyIssueDF.createOrReplaceTempView("avg_nyc_safety_issues")

val maxNYCSafetyIssuesDF = spark.sql(s"""
    select *,
            case
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Homocide` then 'Avg_Homocide'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Sexual_Offense` then 'Avg_Sexual_Offense'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Assault` then 'Avg_Assault'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Weapons_Possession` then 'Avg_Weapons_Posse
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Dignity_Act_Excluding_Cyberbullying` then '
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Dignity_Act_Cyberbullying` then 'Avg_Dignit
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Bomb_Threat` then 'Avg_Bomb_Threat'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_False_Alarm` then 'Avg_False_Alarm'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Drugs` then 'Avg_Drugs'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Alcohol` then 'Avg_Alcohol'
                else 'Unknown'
            end as Max_Safety_Issue
    from avg_nyc_safety_issues
""")

z.show(maxNYCSafetyIssuesDF)
```

settings ▲

Table Options

safety_analysis.ipynb

Name	Value
useFilter	☐
showPagination	☐
showAggregationFooter	☐

_Dignity_Act_Excluding_Cyberbullying	Avg_Dignity_Act_Cyberbullying	Avg_Bomb_Threat...	Avg_False_Alarm ...	Avg_Drugs
86257928118393	0.313107822410148	0.0179704016913319 23	0.0811839323467230 4	0.369133192385
08879492600423	0.24608879492600422	0.0063424947145877 38	0.0708245243128964	0.388160676533 3
70824524312897	0.4241014799154334	0.0059196617336152 22	0.0649048625792811 9	0.306765327695 5
961945031712474	0.4492600422832981	0.0046511627906976 74	0.0073995771670190 27	0.023255813953 72
61945031712475	0.6630021141649048	0.0118393234672304 44	0.0898520084566596 2	0.508456659615

```
maxNYCSafetyIssuesDF: org.apache.spark.sql.DataFrame = [Year: int, Avg_Homocide: double ... 10 more fields]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:48 AM. (outdated)

Outside of NYC

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:48 AM.

SPARK JOB FINISHED

```
var avgOutSafetyIssueDF: DataFrame = spark.emptyDataFrame

for (year <- 2018 to 2023) {

    // sum the number of each violations / number of schools
    var avg = spark.sql(s"""
        select ${safetyIssueColumns.map(col => s"sum(`$col`)/$count as Avg_${col.replace(" ", "_").replace("-", "_")}").mkString(", ")}
        from safety${year}
        where County not in $nyccounties
    """).withColumn("Year", lit(year))

    val columns = Seq("Year") ++ avg.columns.slice(0, 10)
    var avgNew = avg.select(columns.map(avg.col): _*)

    avgOutSafetyIssueDF = if (avgOutSafetyIssueDF.isEmpty) avgNew else avgOutSafetyIssueDF.union(avgNew)
}

z.show(avgOutSafetyIssueDF)
```

settings

Available Fields

Year | Avg_Homocide | Avg_Sexual_Offense | Avg_Assault | Avg_Weapons_Possession | Avg_Dignity_Act_Excluding_Cybe... | Avg_Dignity_Act_Cyberbullying | Avg_Bomb_Threat | Avg_False_Alarm | Avg_Drugs | Avg_Alcohol

keys

Year ✖

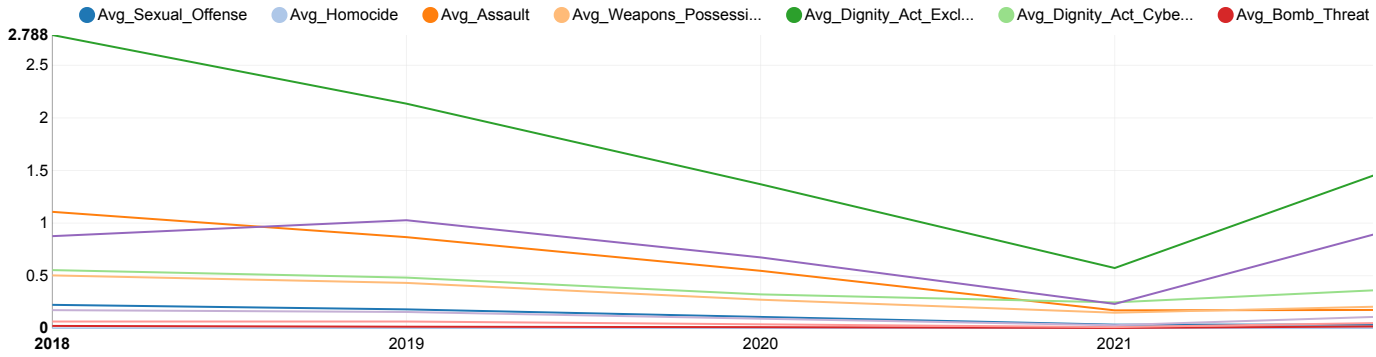
groups

values

Avg_Sexual_Offense SUM ✖
Avg_Homocide SUM ✖
Avg_Assault SUM ✖
Avg_Weapons_Possession SUM ✖
Avg_Dignity_Act_Excluding_Cybe... SUM ✖
Avg_Dignity_Act_Cyberbullying SUM ✖

safety_analysis.ipynb

☐ force Y to 0
☐ zoom
☐ Date format
xAxis :
Default | Rotate | Hide




```
avgOutSafetyIssueDF: org.apache.spark.sql.DataFrame = [Year: int, Avg_Homocide: double ... 9 more fields]
```

Took 3 sec. Last updated by anonymous at December 10 2024, 11:18:51 AM.

SPARK JOB FINISHED

```
avgOutSafetyIssueDF.createOrReplaceTempView("avg_out_safety_issues")

val maxOutSafetyIssuesDF = spark.sql(s"""
    select *,
            case
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Homocide` then 'Avg_Homocide'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Sexual_Offense` then 'Avg_Sexual_Offense'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Assault` then 'Avg_Assault'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Weapons_Possession` then 'Avg_Weapons_Posse
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Dignity_Act_Excluding_Cyberbullying` then '
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Dignity_Act_Cyberbullying` then 'Avg_Dignit
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Bomb_Threat` then 'Avg_Bomb_Threat'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_False_Alarm` then 'Avg_False_Alarm'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Drugs` then 'Avg_Drugs'
                when greatest(${avgSafetyIssueColumns.map(col => s"`${col}`").mkString(", ")}) = `Avg_Alcohol` then 'Avg_Alcohol'
                else 'Unknown'
            end as Max_Safety_Issue
    from avg_out_safety_issues
""")

z.show(maxOutSafetyIssuesDF)
```

settings

Year	Avg_Homocide	Avg_Sexual_Offense	Avg_Assault	Avg_Weapons_Possession	Avg_Dignity_Act_Excluding
2018	0.0	0.22325581395348837	1.1071881606765328	0.5031712473572939	2.7879492600422835
2019	0.0	0.18012684989429176	0.8665961945031713	0.4317124735729387	2.1353065539112053
2020	0.0	0.10824524312896405	0.546723044397463	0.27209302325581397	1.3693446088794925
2021	0.0	0.03488372093023256	0.1708245243128964	0.14841437632135307	0.5731501057082452
2022	4.2283298097251583E-4	0.03615221987315011	0.17610993657505286	0.22600422832980974	1.7767441860465116

```
maxOutSafetyIssuesDF: org.apache.spark.sql.DataFrame = [Year: int, Avg_Homocide: double ... 10 more fields]
```

Took 1 sec. Last updated by anonymous at December 10 2024, 11:18:52 AM.

safety_analysis.ipynb

Save data

FINISHED

Took 0 sec. Last updated by anonymous at December 10 2024, 11:18:52 AM.

SPARK JOB FINISHED

```
// Per District
totalNYCCountyDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/TotalNYCCounty.csv")
NYCcountyYoYDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/NYCcountyYoY.csv")
totalOutCountyDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/TotalOutCounty.csv")
outCountyYoYDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/OutCountyYoY.csv")

// Per School
topNYCSchoolDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/topNYCSchool.csv")
topOutSchoolDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/topOutSchool.csv")

// Per Safety Issue
maxNYCSafetyIssuesDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/MaxNYCSafetyIssues.csv")
maxOutSafetyIssuesDF.write.mode("overwrite").option("header", "true").csv("project/analyzed_data/MaxOutSafetyIssues.csv")
```

Took 8 sec. Last updated by anonymous at December 10 2024, 11:19:00 AM. (outdated)