Funding Safety Analysis

```
var df = spark.read.parquet("scr_data/funding_safety.parquet")
```

```
df: org.apache.spark.sql.DataFrame = [School_BEDS_Code: string, Year: int ... 26 more fields]
```

Took 36 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:02:36 AM.

```
z.show(df.limit(10))
df.printSchema
```

School_BEDS_Code	Year	County_Name	District_Name	School_Name	Homocide	Sexual_Offense	Assault	Weapons_Possession	Dignity_Act_Excluding_Cyberbullying	Dignity_Act_Cyberbullying	Bomb_Threat	False_Alarm	Drugs	A
10100010016	2022	ALBANY	ALBANY CITY SCHOOL DISTRICT	PINE HILLS ELEMENTARY SCHOOL	0	0	0	0	0	2	0	0	0	0
10100010018	2022	ALBANY	ALBANY CITY SCHOOL DISTRICT	DELAWARE COMMUNITY SCHOOL	0	0	0	0	0	0	0	0	0	0
10100010019	2022	ALBANY	ALBANY CITY SCHOOL DISTRICT	NEW SCOTLAND ELEMENTARY SCHOOL	0	0	0	0	0	0	0	0	0	0
10100010023	2022	ALBANY	ALBANY CITY	ALBANY SCHOOL	0	0	0	0	0	0	0	0	0	0

```
 |-- Bomb_Threat: integer (nullable = true)
 |-- False_Alarm: integer (nullable = true)
 |-- Drugs: integer (nullable = true)
 |-- Alcohol: integer (nullable = true)
 |-- District_BEDS_Code: integer (nullable = true)
 |-- School_Type: string (nullable = true)
 |-- Total_Enrollment: decimal(21,2) (nullable = true)
 |-- Total_Teachers: decimal(21,2) (nullable = true)
 |-- Teacher_per_Pupil: decimal(20,2) (nullable = true)
 |-- Total_Staff: decimal(20,2) (nullable = true)
 |-- Staff_per_Pupil: decimal(20,2) (nullable = true)
 |-- Total_Funding: decimal(20,2) (nullable = true)
 |-- Total_Funding_per_Pupil: decimal(20,2) (nullable = true)
 |-- Federal_Funding: decimal(20,2) (nullable = true)
 |-- Federal_Funding_per_Pupil: decimal(20,2) (nullable = true)
 |-- State_&_Local_Funding: decimal(20,2) (nullable = true)
 |-- State_&_Local_Funding_per_Pupil: decimal(20,2) (nullable = true)
```

Took 5 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:02:41 AM.

```
// add the total safety incidents column
df = df
.withColumn(
  "total_safety_incidents",
  coalesce(col("Homocide"), lit(0)) +
  coalesce(col("Sexual_Offense"), lit(0)) +
  coalesce(col("Assault"), lit(0)) +
  coalesce(col("Weapons_Possession"), lit(0)) +
  coalesce(col("Dignity_Act_Excluding_Cyberbullying"), lit(0)) +
  coalesce(col("Dignity_Act_Cyberbullying"), lit(0)) +
  coalesce(col("Bomb_Threat"), lit(0)) +
  coalesce(col("False_Alarm"), lit(0)) +
  coalesce(col("Drugs"), lit(0)) +
  coalesce(col("Alcohol"), lit(0))
)
.withColumn(
  "incidents_per_student",
  col("total_safety_incidents") / col("Total_Enrollment")
)

```

```
df: org.apache.spark.sql.DataFrame = [School_BEDS_Code: string, Year: int ... 28 more fields]
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:09:12 AM.

Correlation Inspect

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:02:42 AM.

Global Level

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:02:42 AM.

```
%spark
val cols = Array("total_safety_incidents", "Total_Enrollment", "Total_Teachers", "Total_Staff", "Total_Funding", "Staff_per_Pupil", "Total_Funding_per_Pupil", "incidents_per_student")
val global_correlations = cols.flatMap { col1 =>
  cols.filter(c => c != col1).map { col2 =>
    val _corr = df.select(corr(col1, col2).alias("correlation")).first().getAs[Double](0)
    (col1, col2, _corr)
  }
}
```

```
cols: Array[String] = Array(total_safety_incidents, Total_Enrollment, Total_Teachers, Total_Staff, Total_Funding, Staff_per_Pupil, Total_Funding_per_Pupil, incidents_per_student)
global_correlations: Array[(String, String, Double)] = Array((total_safety_incidents,Total_Enrollment,-0.07593009784585805), (total_safety_incidents,Total_Teachers,0.42561378987819526), (total_safety_incidents,Total_Staff,0.3464749047234564), (total_safety_incidents,Total_Funding,0.40135785066022456), (total_safety_incidents,Staff_per_Pupil,
0.03287951940982629), (total_safety_incidents,Total_Funding_per_Pupil,-0.008959781057661001), (total_safety_incidents,incidents_per_student,0.6354941668049005), (Total_Enrollment,total_safety_incidents,-0.07593009784585805)...
```

Took 17 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:02:59 AM.

```
%spark
val corrDF = global_correlations.map {
  case (col1, col2, _corr) => (col1, col2, if (_corr.isNaN) 0.0 else _corr)
}.toSeq.toDF("info1", "info2", "correlation")

z.show(corrDF)
```

info1	info2	correlation
Total_Staff	total_safety_incidents	0.3464749047234564
Total_Staff	Total_Enrollment	-0.03516675999722648
Total_Staff	Total_Teachers	0.9234566858577301
Total_Staff	Total_Funding	0.9362168086603451
Total_Staff	Staff_per_Pupil	0.32389932080797224
Total_Staff	Total_Funding_per_Pupil	0.27993871899294787
Total_Staff	incidents_per_student	-0.004875851962336963

```
corrDF: org.apache.spark.sql.DataFrame = [info1: string, info2: string ... 1 more field]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:00 AM.

Country Level

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:00 AM.

```
%spark
val countyMetrics = df.groupBy("County_Name")
.agg(
    avg("incidents_per_student").alias("avg_incidents_per_student"),
    avg("Total_Funding_per_Pupil").alias("avg_funding_per_student"),
    avg("Teacher_per_Pupil").alias("avg_teacher_ratio"),
    avg("Staff_per_Pupil").alias("avg_staff_ratio"),
    sum("Total_Enrollment").alias("total_county_enrollment"),
    sum("total_safety_incidents").alias("total_county_incidents"),
    count("School_BEDS_Code").alias("total_county_schools")
)
```

```
countyMetrics: org.apache.spark.sql.DataFrame = [County_Name: string, avg_incidents_per_student: decimal(38,26) ... 6 more fields]
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:00 AM.

```
%spark
val columnsToCorrelate = countyMetrics.drop("County_Name").columns
val correlations = columnsToCorrelate.flatMap { col1 =>
  columnsToCorrelate.filter(c => c != col1).map { col2 =>
    val _corr = countyMetrics.select(corr(col1, col2).alias("correlation")).first().getAs[Double](0)
    (col1, col2, _corr)
  }
}
```

```
columnsToCorrelate: Array[String] = Array(avg_incidents_per_student, avg_funding_per_student, avg_teacher_ratio, avg_staff_ratio, total_county_enrollment, total_county_incidents, total_county_schools)
correlations: Array[(String, String, Double)] = Array((avg_incidents_per_student,avg_funding_per_student,0.5168163117294754), (avg_incidents_per_student,avg_teacher_ratio,0.5987940585763333), (avg_incidents_per_student,avg_staff_ratio,0.6251941689624894), (avg_incidents_per_student,total_county_enrollment,0.17279123933427115), (avg_incidents_
per_student,total_county_incidents,0.20497316556930803), (avg_incidents_per_student,total_county_schools,0.09467006353134529), (avg_funding_per_student,avg_incidents_per_student,0.5168163117294756), (...
```

Took 15 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:15 AM.

```
%spark

// create correlation df from previous map values
val corrDF = correlations.map {
  case (col1, col2, _corr) => (col1, col2, if (_corr.isNaN) 0.0 else _corr)
}.toSeq.toDF("info1", "info2", "correlation")

z.show(corrDF)
```

info1	info2	correlation
avg_teacher_ratio	avg_funding_per_student	0.5292588268048728
avg_funding_per_student	avg_teacher_ratio	0.529258826804872
avg_funding_per_student	avg_incidents_per_student	0.5168163117294756
avg_incidents_per_student	avg_funding_per_student	0.5168163117294754
avg_incidents_per_student	total_county_incidents	0.20497316556930803
total_county_incidents	avg_incidents_per_student	0.20497316556930803
avg_incidents_per_student	total_county_enrollment	0.17279123933427115
total_county_enrollment	avg_incidents_per_student	0.17279123933427115

```
corrDF: org.apache.spark.sql.DataFrame = [info1: string, info2: string ... 1 more field]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:16 AM.

Compute Incident Index

Here we use (total incidetns count / enrolled students) as the index

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:16 AM.

```
val schoolTypeAnalysis = df
.groupBy("School_Type")
.agg(
    avg("total_safety_incidents").alias("Incidents"),
    avg("Total_Funding_per_Pupil").alias("Avg Funding"),
)
```

```
schoolTypeAnalysis: org.apache.spark.sql.DataFrame = [School_Type: string, Incidents: double ... 1 more field]
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:16 AM.

```
z.show(schoolTypeAnalysis.orderBy(desc("Incidents")))
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:17 AM.

```
z.show(schoolTypeAnalysis.orderBy(desc("Avg Funding")))
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:17 AM.

Aera Analysis

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:17 AM.

```
// areal analysis
val countyTotalFundingAnaluysis = df
.groupBy("County_Name")
.agg(sum("Total_Funding"), avg("Total_Funding_per_Pupil"))
```

```
countyTotalFundingAnaluysis: org.apache.spark.sql.DataFrame = [County_Name: string, sum(Total_Funding): decimal(30,2) ... 1 more field]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:18 AM.

```
z.show(countyTotalFundingAnaluysis.orderBy(desc("sum(Total_Funding)")))
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:18 AM.

```
z.show(countyTotalFundingAnaluysis.orderBy(desc("avg(Total_Funding_per_Pupil)")))
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:19 AM.

```
val countySafetyAnalysis = df
.groupBy("County_Name")
.agg(
  avg("incidents_per_student").alias("avg_incidents_per_student"),
  sum("total_safety_incidents").alias("total_incidents")
)
```

```
countySafetyAnalysis: org.apache.spark.sql.DataFrame = [County_Name: string, avg_incidents_per_student: decimal(38,26) ... 1 more field]
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:19 AM.

```
z.show(countySafetyAnalysis.orderBy(desc("avg_incidents_per_student")))
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:19 AM.

```
z.show(countySafetyAnalysis.orderBy(desc("total_incidents")))
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:03:20 AM.

Long Term Trend Analysis

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:11:35 AM.

```
%spark
import org.apache.spark.sql.expressions.Window
val windowBySchool = Window.partitionBy("School_BEDS_Code").orderBy("Year")
val schoolChanges = df
.withColumn("prev_incidents", lag("incidents_per_student", 1).over(windowBySchool).cast("int"))
.withColumn("prev_funding", lag("Total_Funding_per_Pupil", 1).over(windowBySchool).cast("double"))
.withColumn("incident_change", (col("incidents_per_student") - col("prev_incidents").cast("int")))
.withColumn("funding_change", (col("Total_Funding_per_Pupil") - col("prev_funding").cast("double")))
.withColumn("incident_change_pct", when(col("prev_incidents") === 0, 0).otherwise(col("incident_change") / col("prev_incidents") * 100))
.withColumn("funding_change_pct",col("funding_change") / col("prev_funding") * 100)
```

```
import org.apache.spark.sql.expressions.Window
windowBySchool: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.sql.expressions.WindowSpec@1c710d8c
schoolChanges: org.apache.spark.sql.DataFrame = [School_BEDS_Code: string, Year: int ... 34 more fields]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:09:42 AM.

```
val overallChanges = schoolChanges
  .groupBy("School_BEDS_Code", "School_Name", "District_Name", "County_Name")
  .agg(
    max("Year").alias("last_year"),
    min("Year").alias("first_year"),
    max("incidents_per_student").alias("max_incidents"),
    min("incidents_per_student").alias("min_incidents"),
    max("Total_Funding_per_Pupil").alias("max_funding"),
    min("Total_Funding_per_Pupil").alias("min_funding"),
    avg("incidents_per_student").alias("avg_incidents"),
    avg("Total_Funding_per_Pupil").alias("avg_funding"),
    sum(abs($"incident_change")).alias("total_incident_change"),
    sum(abs($"funding_change")).alias("total_funding_change")
  )
```

```
overallChanges: org.apache.spark.sql.DataFrame = [School_BEDS_Code: string, School_Name: string ... 12 more fields]
```

Took 0 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:10:38 AM.

```
val topIncidentChangeSchools = overallChanges.orderBy(desc("total_incident_change")).limit(10).select("School_Name", "District_Name", "County_Name", "total_incident_change", "total_funding_change", "avg_incidents", "avg_funding")
z.show(topIncidentChangeSchools)
```

Grouped
Stacked
BRONX
OSWEGO
WAYNE
NEW YORK
KINGS
QUEENS
0.501
0.4
0.2
0
ACADEMY OF PUBLIC RELATIONS
FULTON JUNIOR HIGH SCHOOL
MARION JUNIOR-SENIOR HIGH SCHOOL
JHS 167 ROBERT F WAGNER
HIGH SCHOOL FOR ACADEMIC EXCELLENCE
CITY KNOLL MIDDLE SCHOOL
BROWNSVILLE ACADEMY HIGH SCHOOL
JHS 189 DANIEL CARTER BEARD
MS 297
PS 175 CITY ISLAND

```
topIncidentChangeSchools: org.apache.spark.sql.DataFrame = [School_Name: string, District_Name: string ... 5 more fields]
```

Took 2 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:10:42 AM.

```
val topFundingChangeSchools = overallChanges.orderBy(desc("total_funding_change")).limit(10).select("School_Name", "District_Name", "County_Name", "total_incident_change", "total_funding_change", "avg_incidents", "avg_funding")
z.show(topFundingChangeSchools)
```

```
topFundingChangeSchools: org.apache.spark.sql.DataFrame = [School_Name: string, District_Name: string ... 5 more fields]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:10:45 AM.

```
val lowestIncidentSchools = overallChanges
.where("avg_incidents > 0")
.orderBy("avg_incidents")
.select("School_Name", "District_Name", "County_Name", "avg_incidents")

z.show(lowestIncidentSchools.limit(20))
```

Grouped
Stacked
SUFFOLK
WESTCHESTER
BRONX
BROOKLYN
NASSAU
KINGS
QUEENS
NEW YORK
ONTARIO
0
GARTEN-PRE K CENTER
CYCLE PATH KINDERGARTEN-PRE K CENTER
PARK SCHOOL
PS 583
PS 319
SMITH STREET ELEMENTARY SCHOOL
PS 272 CURTIS ESTABROOK
PS 377
YONKERS EARLY CHILDHOOD ACADEMY
PS 223 LYNDON B JOHNSON
THOMAS CORNELL ACADEMY
PS 36 MARGARET DOUGLAS
PS 49 WILLIS AVENUE
PS 4 DUKE ELLINGTON
PS 222-FF CHRISTOPHER A SANTORA
PS 48 PO MICHAEL J BUCZEK
OF MATH, SCIENCE & DESIGN TECHNOLOGY
WEST STREET ELEMENTARY SCHOOL
PS 161 ARTHUR ASHE SCHOOL
NORTH COLEMAN ROAD SCHOOL

```
lowestIncidentSchools: org.apache.spark.sql.DataFrame = [School_Name: string, District_Name: string ... 2 more fields]
```

Took 1 sec. Last updated by yx3494_nyu_edu at December 13 2024, 4:12:36 AM.