Dynamic Integration Test Suite

•••

Nathan Cronk (Professor Wallingford)

Agenda

- Objectives
- Files
- Demo
- Challenges and Roadblocks
- Next Steps

Objectives

- Create an example production environment that replicates John Deere Machine Health's
- Create a dynamic integration test suite to ensure quality of data query functions and tables
- Enable my team with a baseline and ideas for implementation in industry

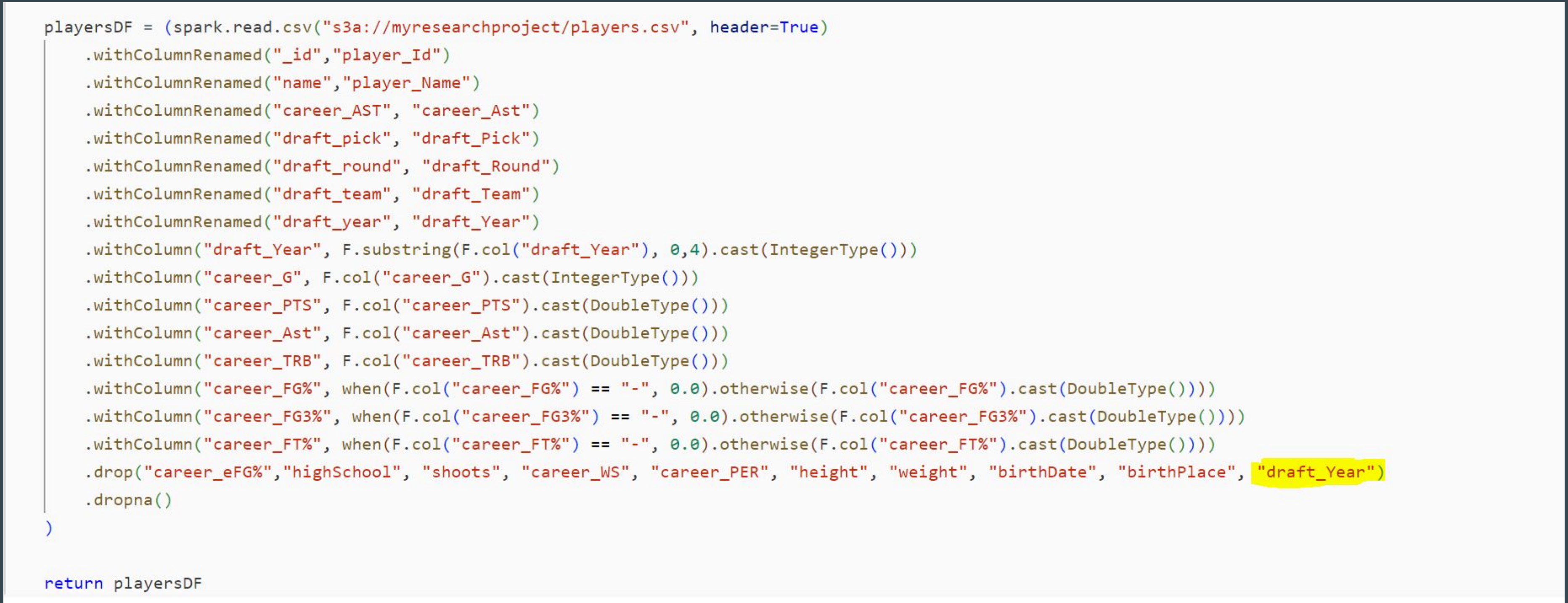
Files

- GitHub Repo: https://github.com/ncronk10/Undergrad_Project

Tools:

- Databricks
- PySpark
- Referencing Functions from work
- Amazon S3
- GitHub (actions)
- Data set from: https://data.world/datadavis/nba-salaries

Demo

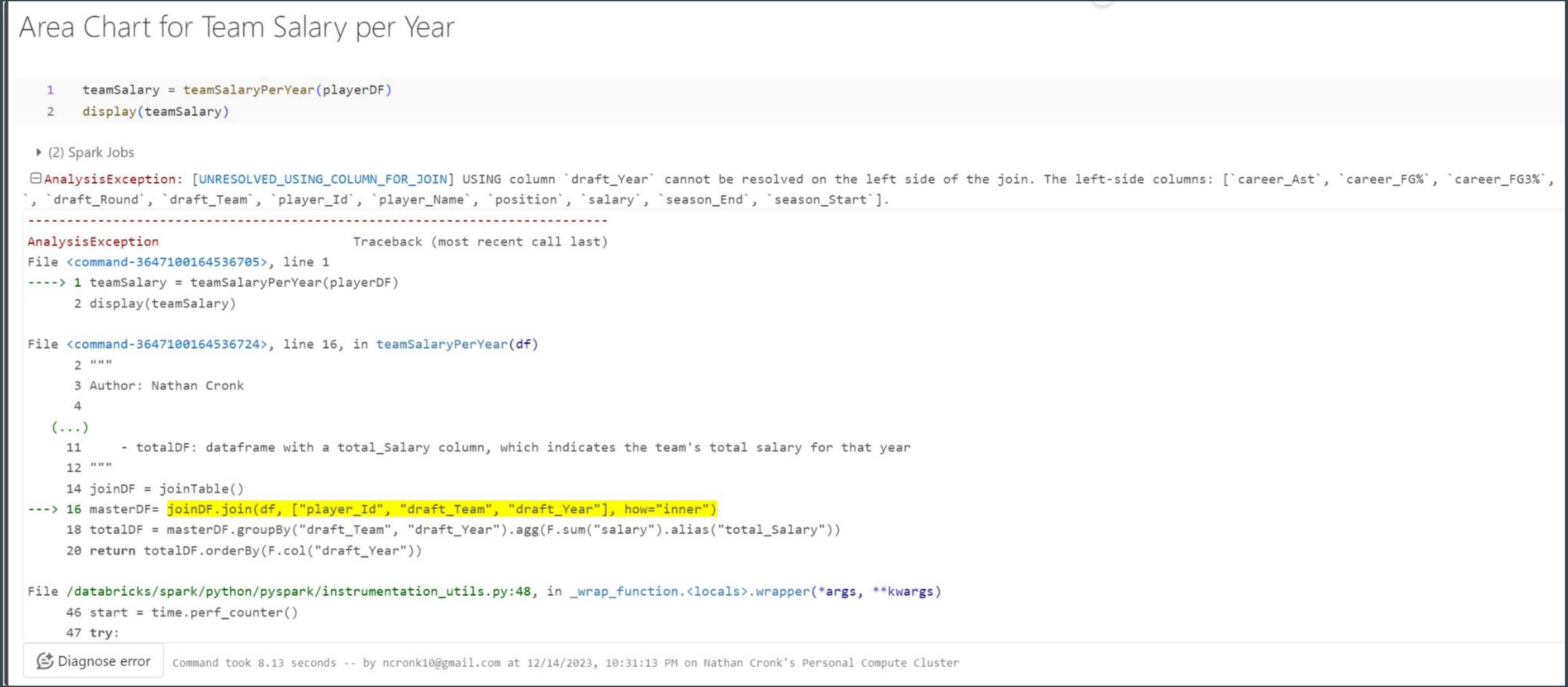


```
playersDF = (spark.read.csv("s3a://myresearchproject/players.csv", header=True)
    .withColumnRenamed("_id","player_Id")
    .withColumnRenamed("name","player_Name")
    .withColumnRenamed("career_AST", "career_Ast")
    .withColumnRenamed("draft_pick", "draft_Pick")
    .withColumnRenamed("draft_round", "draft_Round")
    .withColumnRenamed("draft_team", "draft_Team")
    .withColumnRenamed("draft_year", "draft_Year")
    .withColumn("draft_Year", F.substring(F.col("draft_Year"), 0,4).cast(IntegerType()))
    .withColumn("career_G", F.col("career_G").cast(IntegerType()))
    .withColumn("career_PTS", F.col("career_PTS").cast(DoubleType()))
    .withColumn("career_Ast", F.col("career_Ast").cast(DoubleType()))
    .withColumn("career_TRB", F.col("career_TRB").cast(DoubleType()))
    .withColumn("career_FG%", when(F.col("career_FG%") == "-", 0.0).otherwise(F.col("career_FG%").cast(DoubleType())))
    .withColumn("career_FG3%", when(F.col("career_FG3%") == "-", 0.0).otherwise(F.col("career_FG3%").cast(DoubleType())))
    .withColumn("career_FT%", when(F.col("career_FT%") == "-", 0.0).otherwise(F.col("career_FT%").cast(DoubleType())))
    .drop("career_eFG%","highSchool", "shoots", "career_WS", "career_PER", "height", "weight", "birthDate", "birthPlace", "draft_Year")
    .dropna()
)

return playersDF
```

Without Test Suite, dropped column “draft_Year”

Demo Cont.



Area Chart for Team Salary per Year

```python
teamSalary = teamSalaryPerYear(playerDF)
display(teamSalary)
```

```
⊟AnalysisException: [UNRESOLVED_USING_COLUMN_FOR_JOIN] USING column `draft_Year` cannot be resolved on the left side of the join. The left-side columns: [`career_Ast`, `career_FG%`, `career_FG3%`,
`, `draft_Round`, `draft_Team`, `player_Id`, `player_Name`, `position`, `salary`, `season_End`, `season_Start`].
---------------------------------------------------------------------------
AnalysisException                         Traceback (most recent call last)
File <command-3647100164536705>, line 1
----> 1 teamSalary = teamSalaryPerYear(playerDF)
      2 display(teamSalary)

File <command-3647100164536724>, line 16, in teamSalaryPerYear(df)
      2 """
      3 Author: Nathan Cronk
      4
   (...)
     11     - totalDF: dataframe with a total_Salary column, which indicates the team's total salary for that year
     12 """
     14 joinDF = joinTable()
---> 16 masterDF= joinDF.join(df, ["player_Id", "draft_Team", "draft_Year"], how="inner")
     18 totalDF = masterDF.groupBy("draft_Team", "draft_Year").agg(F.sum("salary").alias("total_Salary"))
     20 return totalDF.orderBy(F.col("draft_Year"))

File /databricks/spark/python/pyspark/instrumentation_utils.py:48, in _wrap_function.<locals>.wrapper(*args, **kwargs)
     46 start = time.perf_counter()
     47 try:
```

Command took 8.13 seconds -- by ncronk10@gmail.com at 12/14/2023, 10:31:13 PM on Nathan Cronk's Personal Compute Cluster

Challenges and Roadblocks

- Restructuring of Functions
- Data access issue with original storage location
- Mock (synthetic) data
- Databricks CLI (command line interface)

Next Steps

- Add the functionality to tests to use synthetic mock data

- Continue to learn and research possible quality enhancements to test suite

- Implement test suite on Machine Health platforms Data and Analytics environment

Questions?