

# OPEN DATA SCIENCE CONFERENCE

Burlingame | November 2nd 2017

Nov 02  
2:00 PM  
Room T2

Modeling big data with R, sparklyr, and Apache Spark

BIG DATARINTERMEDIATE

Dr. John Mount

Consulting Algorithmist/Researcher/Principal at Win-Vector LLC and  
Co-author of Practical Data Science with R



# Plan

- Talk about performance.
- Work through some extra topics
  - Using `spark_apply()` for R user defined functions
  - Using Spark SQL directly
- Remind you of topics and resources

# Performance

- Due to lazy evaluation semantics work is not done until results are *used by a non-lazy operator*.
  - Make timing difficult without explicit compute() steps.
- Queries, not data are collected in dplyr pipelines.
  - Looking at the same result twice may generate the same result twice

# Rules for using Sparklyr

- Can not over-emphasize how dependent you are on cluster deployment and configuration.
- Practice/debug on small data
  - But say in the sub-dialect of R/dplyr that works on Spark.
- Debugging lazy evaluation is difficult
  - Errors tend to be reported far from causes
  - If I had a dollar for every time somebody said "my calculation worked, but when I went to save it" ...
- Long sequences of calculation get expensive
  - Due to “view stacking” or “query nesting”
  - Sparklyr is supplying a “leaky abstraction” that pretends calculations are happening when you issue them (when they are not).
- Try to prefer creating more rows to creating more columns.

# A bit about Spark

- Representations are a bit more direct than you would expect.
  - Can "Bobby Tables" yourself just by partitioning on a column with non-trivial string values.
- Semantics are different than what R users expect
  - NULL / NA not identical concepts
  - No factors
  - Lifetime of values very critical
  - Optimization hints (tbl\_cache, partition) important.

# Mitigations

- Saving results
  - `dplyr::compute()`
  - checkpoint (saving data)
  - Need a temporary name management system
    - `replyr::makeTempNameGenerator()` is one such
- Put starting data in memory
  - `tbl_cache()`
- Partition
  - `sdf_repartition()`



# Work through markdown together

- Exercises/solutions/06-Spark-Extension.Rmd

# What we have achieved in this workshop

- We have worked through `dplyr` in detail.
- We applied `dplyr` data manipulation methods in a big-data environment (`Spark` / `SparklyR`).
- We ran supervised machine learning experiments in big-data environments (`SparkML`).
- We learned how to extend and use `Spark` more directly (`Spark SQL`, `SparklyR` extensions interface, and even a bit of `SparkR`).



# Some links

# This material

- <https://github.com/WinVector/ODSCWest2017>
- Detailed local install instructions:
  - README.Rmd
  - Exercises/solutions/RsparklingInstall.Rmd

# RStudio

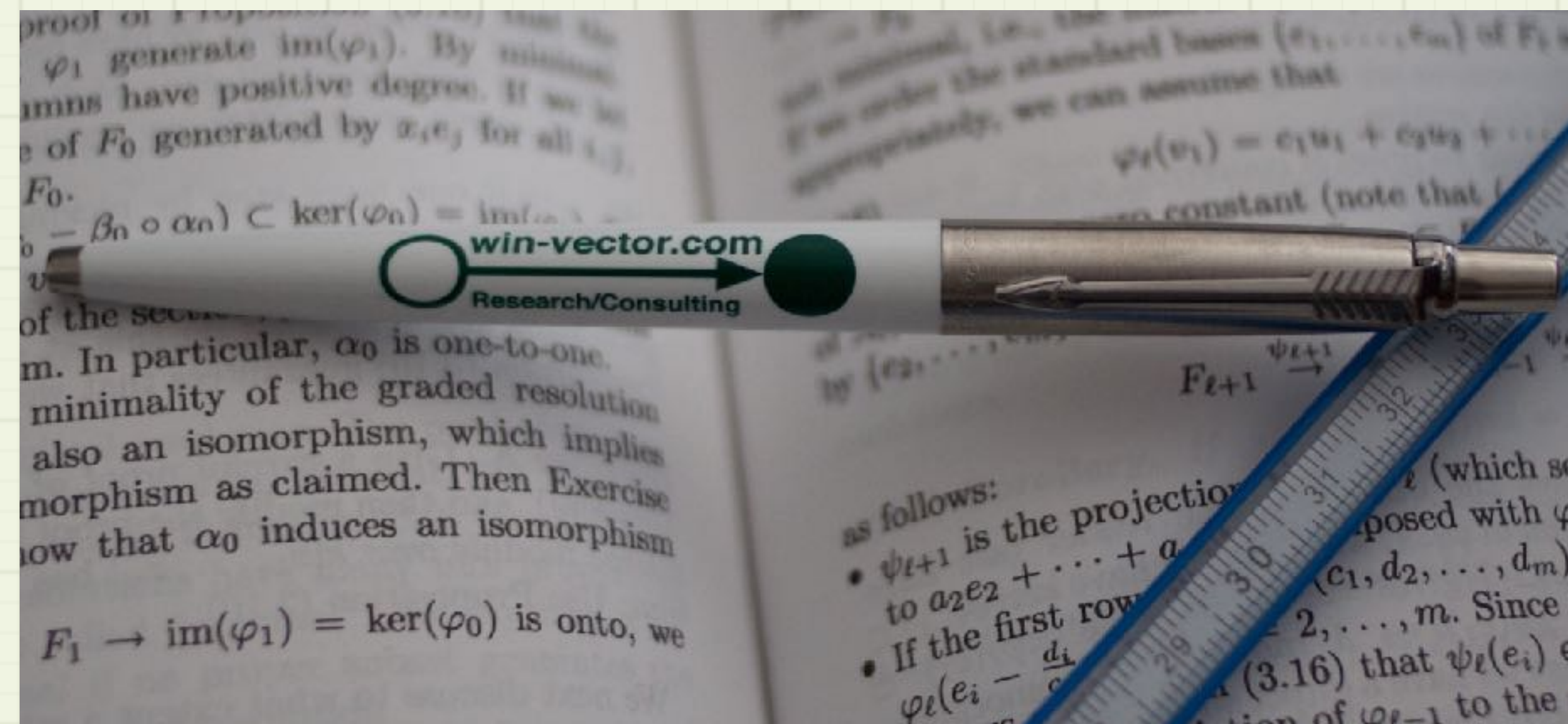
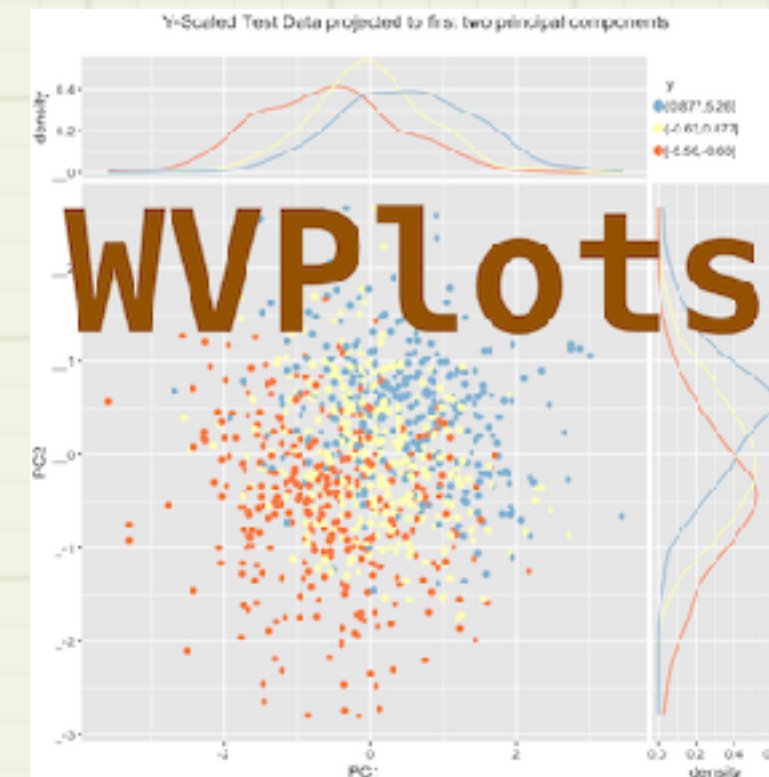
- <https://www.rstudio.com>
- <https://www.rstudio.com/products/rstudio-server-pro/>
- <https://www.rstudio.com/products/connect/>
- <https://www.rstudio.com/products/shiny-server-pro/>
- <https://www.rstudio.com/products/shinyapps/>
- <https://github.com/rstudio/rstudio>





# Win-Vector

- <http://www.win-vector.com>
- <http://www.win-vector.com/blog/>
- <https://github.com/WinVector>
- [@WinVectorLLC](#)
- [contact@win-vector.com](mailto:contact@win-vector.com)



# SparklyR

- <https://www.rstudio.com/resources/cheatsheets/>
- <http://spark.rstudio.com>
- <http://spark.rstudio.com/dplyr.html>
- <http://spark.rstudio.com/extensions.html>



# Demos

1. R markdown notebooks with dplyr (NYCFlights I 3 - Local mode)  
<https://beta.rstudioconnect.com/content/1706/>
2. Flexdashboard  
<http://colorado.rstudio.com:3838/nathan/flights-dash-spark/>  
<http://colorado.rstudio.com:3838/nathan/flights-dash-rdata/>  
<https://beta.rstudioconnect.com/content/1439/>
3. Comparison of ML classifiers (Titanic - Local mode)  
<https://beta.rstudioconnect.com/content/1518/>
4. Manipulate data at scale (NYC Taxi - Cluster mode)  
<https://beta.rstudioconnect.com/content/1704/>
5. End to end analysis (Flights - Cluster mode)  
<https://beta.rstudioconnect.com/content/1446/>



Thank you!!!!!!

# Questions? Comments?