

# Beautiful Operationalization

**Jim Wisnowski, Adsurgo LLC**

**Andrew Karl, Adsurgo LLC**

# Aim

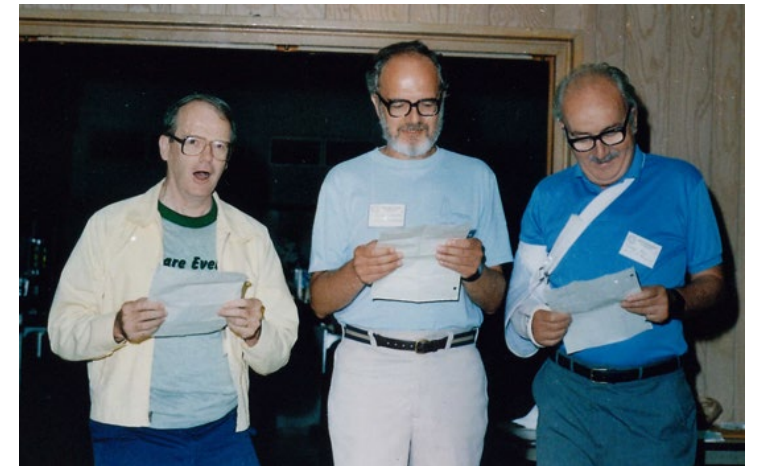
- Understand the definition and critical importance of operationalizing your solution
- Demonstrate quick wins in JMP for helping operationalization
- Provide an example from recent customer engagement that extended JMP Pro reporting capabilities using an AI platform to achieve beautiful results in both form and function

*The most that can be expected from any model is that it can supply a useful approximation to reality: All models are wrong; some models are useful.*

BHH 2<sup>nd</sup> Ed. P. 404

*All your models are useless to me if they are not integrated into our production environment or somehow supporting my decision.*

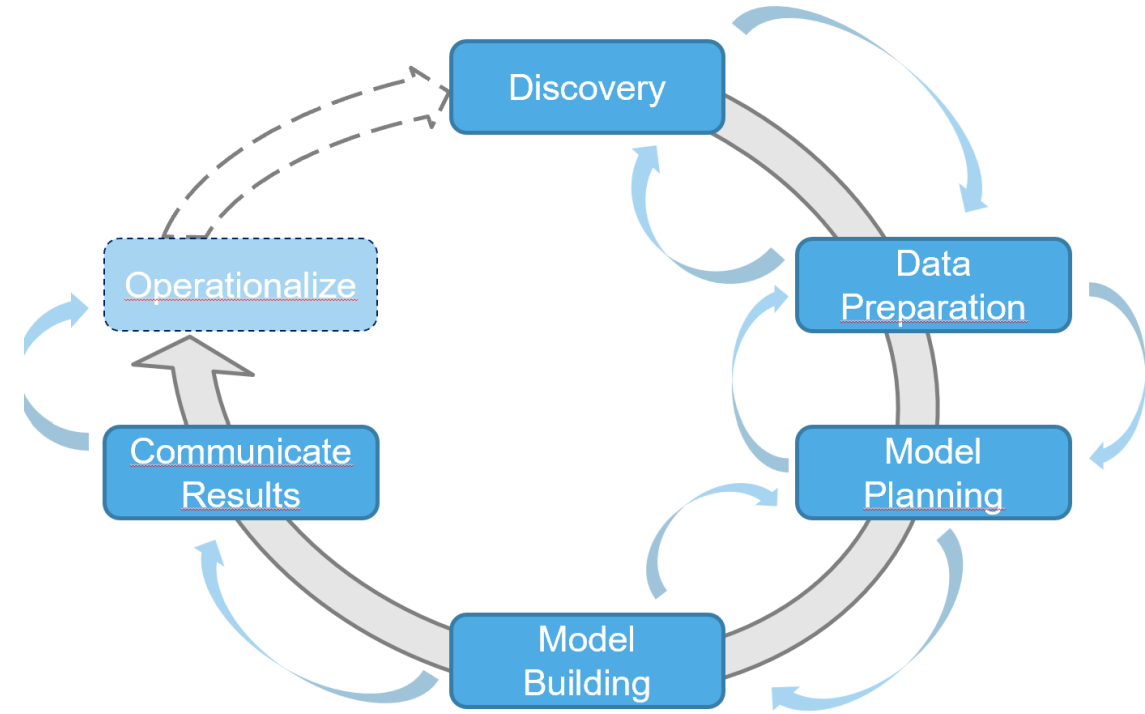
Your boss or boss's boss



There's no theorem like Bayes' Theorem

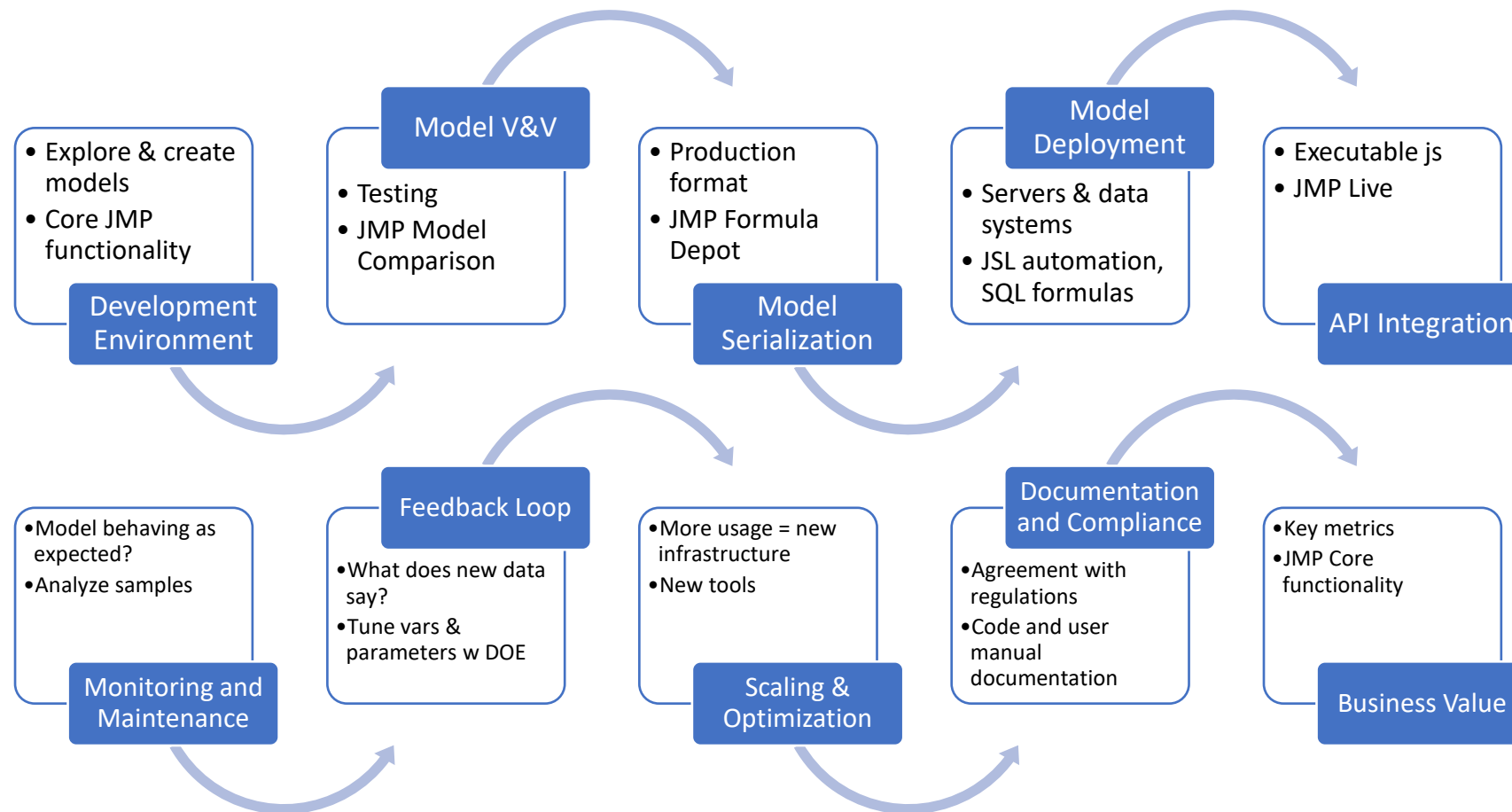
# What is Operationalization?

- What? So What? **Now What?**
- The last step of the iterative data science process whereby we want to deploy our solution that we have we have love so much.
- Going from the lab to production line
- *As organizations explore the potential of AI to solve complex business problems, they have to find a sustainable way to put those models into production in a governed, trustworthy and automated way.*



Introduction to Data Science <https://sherbold.github.io/intro-to-data-science>

# Operationalization for Business Value



# Operationalization

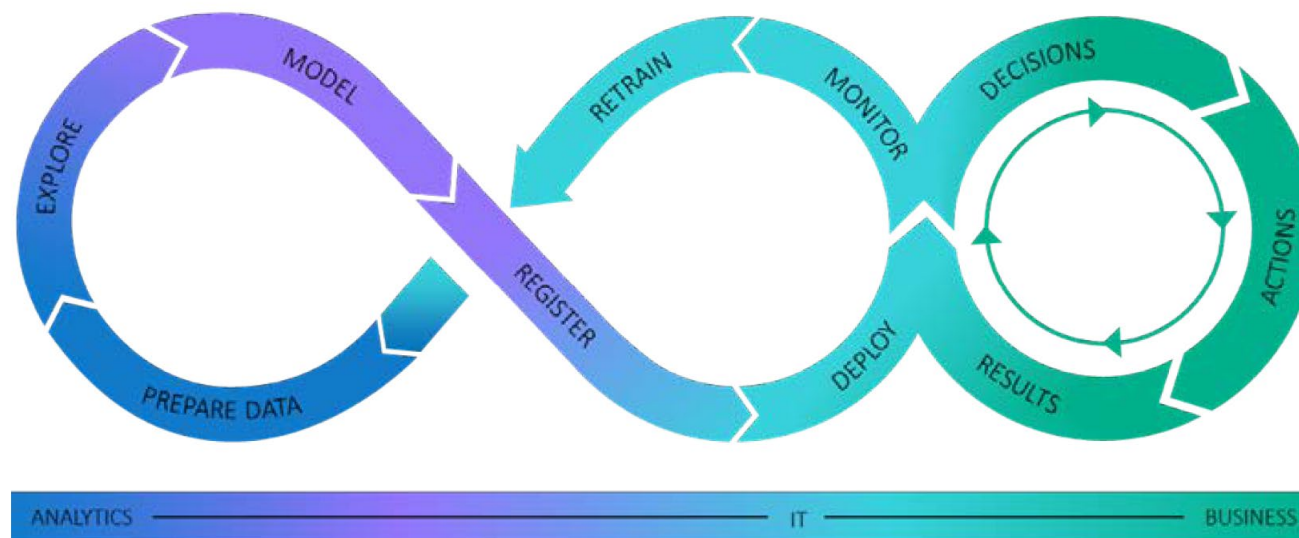
- Business engagement and strategic planning
  - Management and stakeholder buy-in to actually use your analytic efforts
- Building data pipelines for production environment
  - Transition to dynamic environment ensuring repeatable processes
- Model development, measurement, and improvement
  - Iteratively refine, integrate new business realities, monitor model drift & adaption
- Insights to action
- Adoption and measurement
  - Tie key performance indicators to financial results

# Operationalization

- Struggle with the last mile to deploy analytics in operational systems
- Challenging to transition from data science team to users
  - Less than 50% of best models deployed; 90% take > 3 months
  - Analysis team and Production/IT team
  - Not clearly communicate data curation/feature engineering process
  - Model configuration management challenges

# Operationalization with ModelOps

- DevOps is well known business buzz word describing efficient methodologies to optimize software development efforts
- ModelOps is the culture, process, and technology to quickly & effectively deploy useful analytics



# Transitioning Challenges in Operationalization

- Data collection in analysis phase required cleaning...in operational phase those methods have to be automated
  - Nice that we have workflow builder and recode formula
- Error management (dirty data, data access, and all the other obstacles) in development resolved with SME interaction one at a time; production environment does not account for anomalies not included in the transition logic; add logs, alerts, system health monitoring
- Communicating the story changes from the proof of concept to production phase—different customer, graphics, and metrics
- Security becomes much larger issue as grow from a few people having access to business insights to many across the enterprise
- Model management of data inputs, algorithms, and hyperparameters essential to maintain configuration control and avoid obsolescence



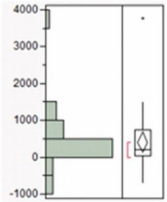

# Demonstration of Formula Depot+

- Outstanding presentations from previous Discovery conferences
  - **Scoring Outside the Box**, Nascif Abousalh-Neto and Dan Valente, 2016
  - **A Journey from Modeling to Scoring**, Nascif Abousalh-Neto, 2017
- [Better together: Extending JMP with open source software](#), Nascif Abousalh-Neto, Meijian Guan, Ruth Hummel 2020

Nascif Abousalh-Neto, JMP Principal Software Developer, SAS  
Daniel Valente, PhD, JMP Senior Product Manager, SAS


**Scoring Outside the Box**  
**MOTIVATION** DEPLOYMENT: IMMEDIATE VS LONG-TERM NEEDS

Give a man a fish, Teach a man to fish,


Presentation: Reports, Dashboards  Deployment (Production)  Scoring: Deployed Model

he'll eat for a day. he'll eat for life.




**Stat** The ISI's Journal for the Rapid Dissemination of Statistics Research

SPECIAL ISSUE PAPER | [Open Access](#) 

**Better together: Extending JMP<sup>®</sup> with open-source software**

Nascif Abousalh-Neto, Meijian Guan, Ruth Hummel 

First published: 03 December 2020 | <https://doi.org/10.1002/sta4.336>

SECTIONS  PDF  TOOLS  SHARE

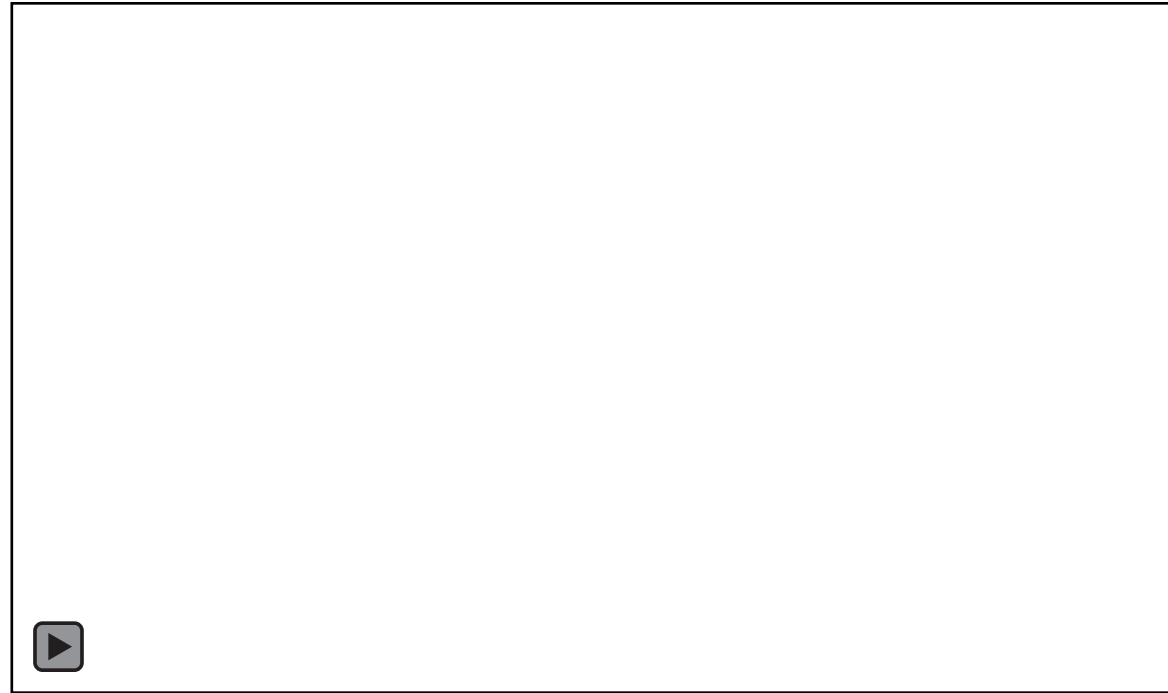
**Abstract**

JMP is commercial software designed for interactive data analysis and exploration. JMP's high-level, visual interface makes it an outstanding tool for teaching best practices, methods and model building techniques. JMP is also designed for extensibility, with features that allow the embedding of and deployment to open-source packages and environments. In this paper, we will explore use cases that illustrate how JMP users can take advantage of the broad support and rapid pace of change of open-source software while still benefiting from JMP's streamlined interface and robust capabilities.

# Demonstration of Formula Depot+

- Build a main effects model for Boston Housing data (Median Value=Number of Rooms + Distance + Lstat) and a Bootstrap Forest
- Demo pptx with dynamic profiler
- Save formulas to the formula depot in Python
- Create a Jupyter notebook to score new data
- Output predicted values to Excel file

# Placeholder for Easy Profiler Import



# How to Make Jupyter Notebook

JA

I have python code generated from a model in the JMP formula depot, create a jupyter notebook to score new data from an excel file



```
# Import necessary libraries
import pandas as pd
from jmp_scoring_script import score # Importing the score function from

# Load the Excel data into a pandas DataFrame
data = pd.read_excel('data_to_score.xlsx', engine='openpyxl')

# Check the first few rows of the data to ensure it's loaded correctly
print(data.head())

# Score the data using the function from JMP's scoring script
# Make sure to adapt this step based on the specifics of your JMP script
scored_data = score(data)

# Display the first few rows of the scored data for verification
print(scored_data.head())

# Save the scored data back to a new Excel file
scored_data.to_excel('scored_data.xlsx', engine='openpyxl', index=False)

print("Scoring completed and saved to scored_data.xlsx!")
```

JA

ok, the function score(data) does not seem to be working. Error is score() missing 1 required positional argument: 'outdata'

```
# Create an empty DataFrame for the scores. If 'outdata' requires specific columns
outdata = pd.DataFrame()

# Score the data using the function from JMP's scoring script
score(data, outdata)
```

# A Beautiful Jupyter Notebook

```
jupyter ChatGPT_BH_Notebook Last Checkpoint: 11 hours ago (autosaved) Python 3 (ipykernel)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
+ %< > Run C Code
In [ ]: # Jupyter notebook to create predicted values from a saved script in Python from JMP Formula Depot
# Place your score file.py from JMP, Excel file you want to score, this notebook, and
# jmp_score.py from your install such as "C:\Program Files\SAS\JMPPRO\17\Scoring\Python\jmp_score.py"
In [21]: import pandas as pd
# Import necessary Libraries
In [22]: from Fit_Least_Squares_BH2 import score # Importing the score function from the JMP script
In [23]: data = pd.read_excel('BH2.xlsx', engine='openpyxl')
# Load the Excel data into a pandas DataFrame
In [24]: outdata = pd.DataFrame()
# Set up output file for export
In [25]: print(data.head())
# Check the first few rows of the data to ensure it's loaded correctly
      crim  zn  indus  chas  nox  rooms  age  distance  radial  tax  \
0  0.00632  18.0   2.31    0  0.538  7.575  65.2   4.0900    1  296
1  0.02731   0.0   7.07    0  0.469  6.421  78.9   4.9671    2  242
2  0.02729   0.0   7.07    0  0.469  7.185  61.1   4.9671    2  242
3  0.03237   0.0   2.18    0  0.458  6.998  45.8   6.0622    3  222
4  0.06905   0.0   2.18    0  0.458  7.147  54.2   6.0622    3  222
      pt      b  lstat  mvalue
0  15.3  396.90   4.98   24.0
1  17.8  396.90   9.14   21.6
2  17.8  392.83   4.03   34.7
3  18.7  394.63   2.94   33.4
4  18.7  396.90   5.33   36.2
In [26]: scored_data = score(data, outdata)
# Score the data using the function from JMP's scoring script
In [27]: print(outdata.head())
# Display the first few rows of the scored data for verification
      Pred Formula  mvalue
0           34.235360
1           25.197827
2           32.617314
3           31.989272
4           30.986646
In [28]: outdata.to_excel('scored_data_for_demo.xlsx', engine='openpyxl', index=False)
# Save the scored data back to a new Excel file
In [29]: print("Scoring completed and saved to scored_data.xlsx!")
Scoring completed and saved to scored_data.xlsx!
```

# Automating JMP Workflow with JSL

- JSL and Workflow Builder can easily automate repetitive production tasks
- Not limited to simple data imports and analysis...complex data lake connections, queries, data prep, modelling, and output reports
- Word on the street is JMP is the best JSL coder as we save our scripts to data tables and use Workflow builder along with enhanced logs...is it better than AI?
- Demo (optional)

# Use Case: Generate Many Reports Following Specific Template Format (Very Common Ask for Us)

- National medical provider has large database detailing the transactions of patients across its clinics.
- Leadership would like to give every provider a summary of their activity by code (e.g. ICD 10) for 2022 along with recommendations for increased productivity opportunities.
- JMP can easily tabulate results and be used to calculate what expected normal ranges of services
- JMP also has nice native capability to export output to pptx, pdf, Word and so forth
  - Even better when coupled with JSL to extract portions of JMP reports and placing them in existing report formats
- Challenge was the specific templates in Word and Excel had formats exceeding even advanced JSL scripting
  - Solution could have been using VBA within Word and Excel...a veritable nightmare.
  - We've heard Python has some good capabilities.
- We don't know Python. Can an AI solution like ChatGPT 4.0 be helpful?

# Use Case: Generate Many Reports Following Specific Template Format (Very Common Ask for Us)

Summary Table – Dr No

Label	2022 Count	Percentile	Rank (184)	Mean	Max	Avg per D
Doctor Days	188	35.7%	119	191.1	252	
Exam Short	22	57.3%	79	103.4	1565	0
Exam Med	0	7.6%	171	527.0	2582	0
Exam Long	954	80.5%	36	348.4	1940	5
Consultation	39	67.6%	60	178.2	1762	0
Total Exam	1015	41.1%	109	1157.0	2628	5
Total Xrays	400	35.7%	119	747.9	3102	2
Total CT Scans	241	77.3%	42	154.2	1175	1
Procedure A Standard	0	40.5%	110	6.5	237	0
Procedure A Complex	140	53.0%	87	322.9	2178	0
Procedure A Advanced	438	12.4%	162	1081.6	5068	2
Procedure A Simple	204	90.8%	17	66.2	474	1
Total Procedure A	782	18.9%	150	1477.1	5248	4
Procedure B Standard	101	67.0%	61	91.5	579	0
Procedure B Complex	243	50.3%	92	300.2	2892	1
Procedure B Advanced	1313	84.9%	28	832.1	2939	6
Procedure B Simple	59	89.2%	20	24.3	383	0
Total Procedure B	1716	80.5%	36	1248.2	5748	9
Total A and B	2498	50.8%	91	2725.3	10299	13
Total C	22	55.7%	82	33.2	485	0
Procedure D Simple	0	56.8%	80	4.0	130	0
Procedure D Complex	93	87.6%	23	40.0	678	0
Procedure E Simple	110	24.9%	139	219.8	859	0
Procedure E Complex	0	9.2%	168	44.9	413	0
Procedure E Advanced	0	44.3%	103	12.3	210	0
Procedure F Simple	4	59.5%	75	12.1	270	0
Procedure F Complex	17	85.9%	26	8.7	66	0
Procedure F Advanced	0	25.9%	137	6.2	78	0
Procedure G	92	42.7%	106	152.5	755	0
Administer drug A	571	47.6%	97	691.7	3818	3
Administer drug B	1267	69.7%	56	1001.3	3411	6
Administer drug C	12	29.2%	131	77.7	624	0
Administer drug D	0	34.1%	122	50.3	1165	0
Administer drug E	0	32.4%	125	50.9	863	0
Administer drug F	63	67.6%	60	157.6	1077	0
Administer drug G	47	76.8%	43	95.2	876	0

Hello Dr J,

You should get an email link to our file server with a zip file containing all the Word and Excel files.

The summary table in Word will take a few seconds to format:

1. Select the entire table with the + sign that shows up in the upper left hand corner.
2. Right click Autofit>Autofit to Window.
3. This shrinks the Label column making some entries take 2 lines, go to right side of Label cell and resize so all fit on a single line.
4. Right click in the selected table again and choose the Borders icon next to the paint can and select all borders.

Let us know if you need anything else.

Jim

