Combing Open-Source Programming Languages with GIS for Spatial Data Science

Maja Kalinic

Master's Thesis

Outline

- Introduction and Motivation
- Research Objectives
- Related Work
- Methodology and Case Study
- Outputs and Discussion
- Conclusion
- Limitations and Future Research

Introduction and Motivation

There is GIS, specifically designed software for spatial data handling, there are programming languages, a set of instructions forwarded to a machine for getting desired outputs, and it is a challenge to know how, but important to link them.

Information visualization and communication is, on the other hand, challenge for Cartographers and GIS professionals since their aim is to visually deliver useful information to aid humankind and their sustainability.

Research Objectives

Which open-source programming language(s) is (are) the most prominent one(s) to be combined with GIS for the analysis and interpretation of spatial data?

Which dataset should be used and which methodology is the best to develop a real world example scenario?

How can these language(s) be combined with GIS?

What is the most efficient way of dividing the work between open-source programming language and GIS?

Related Work

GIS software ranking by:		Programming languages ranking by:	
Popularity	ArcGIS Desktop QGIS	**PYPL**	Java Python
Employer	ArcGIS Desktop Map Info	**GIS scripting**	Python R
Research	ArcGIS Desktop QGIS	**Data processing and analyses**	Python R
Community	ArcGIS Desktop QGIS	**Geospatial libraries**	Python R

Sources: (1), (2), (3), (4), (5), (6), (7), (8)

Data

- Open-source, spatial data
- GIS in crime prevention
- Programming languages in crime prediction
- San Francisco criminal data
- 878049 rows, 9 columns
- Dates/ Descript/ Category/ DayOfWeek/ PdDistrict/ Resolution/ Address/ X / Y
- Size 0.119GB

Methodology

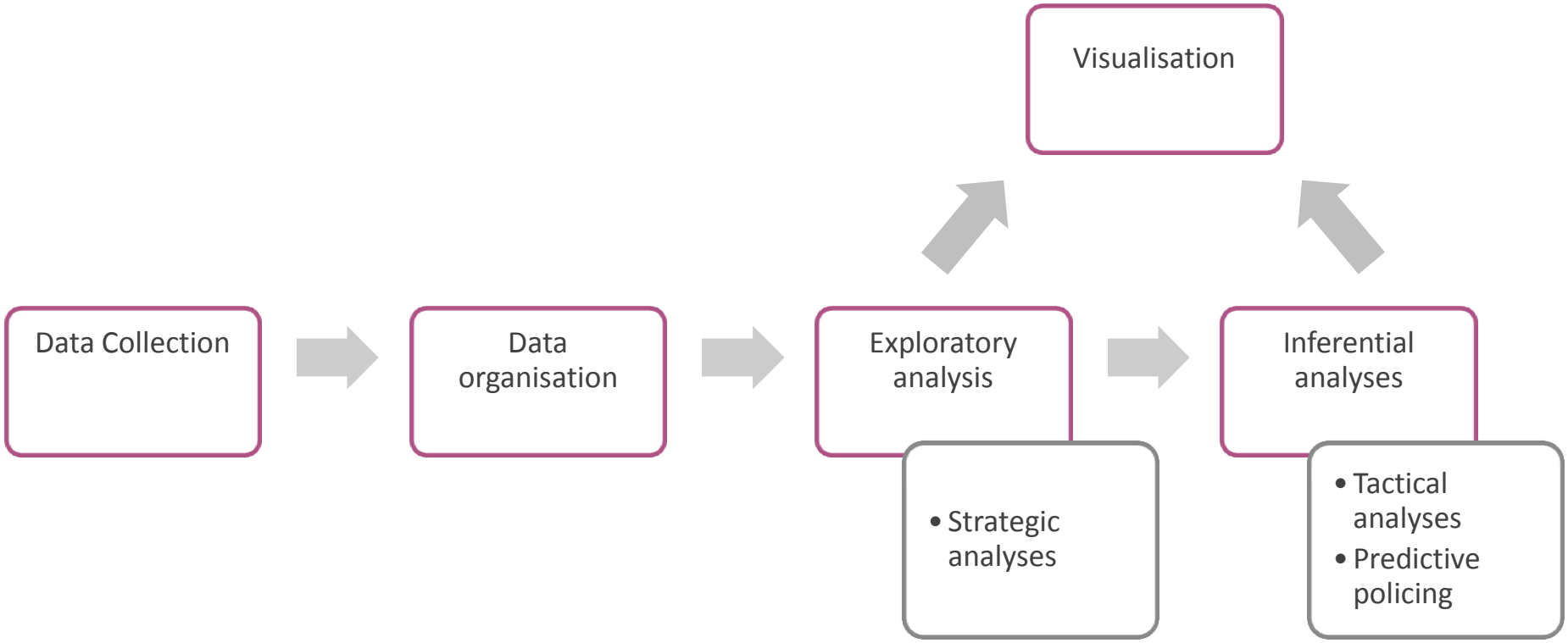

Figure 1: Methodology

Case Study Goals

- Analyse and visualize the spatial and temporal relationship of crimes
- Determine when and where arrests frequently occur and analyse how variables interact with each other
- Identify where crimes are emerging
- Identify areas with unusually high number of crimes
- Investigate factors that might influence unlawful behavior
- Predict where higher numbers of crimes are likely to occur

Steps and Tools

	Data organization	Strategic analyses	Tactical analyses	Predictive policing
	- Rename columns - Find and delete uncomplete entries - Adjust the Timestamp - Add new columns	- Spatial and temporal distribution of crimes	- Crime resolutions - Dataset variable dependency - Trends over space and time - Crime and population relationship - Influencers of unlawful behavior	- When - Where - Which crimes are most likely to occur
Tools	Python	Python	R, R-ArcGIS Bridge	Python
Package	pandas, numpy	matpolotlib	data.table, ggplot2, corrplot	scikit -learn

Results

Task	Tools	Outputs
Strategic analyses (criminal activity over time)	Python packages: Pandas, numpy, matplotlib	Graphs showing where and when (year, month, day of the week, hour) crimes occur
Tactical analyses (trends and patterns)	R libraries: ggplot2	Heat maps
	Emerging Hot Spot Analyses	Hot and Cold spots map
	Optimized Hot Spot Analyses	Map with unusually high number of crimes
	R libraries: corrplot	Correlation matrix
Predictive policing (where crimes are likely to occur)	Python model: scikit-learn	Classifiers accuracy: Logistic, AdaBoost Logistic Random Forest

Python
matplotlib package

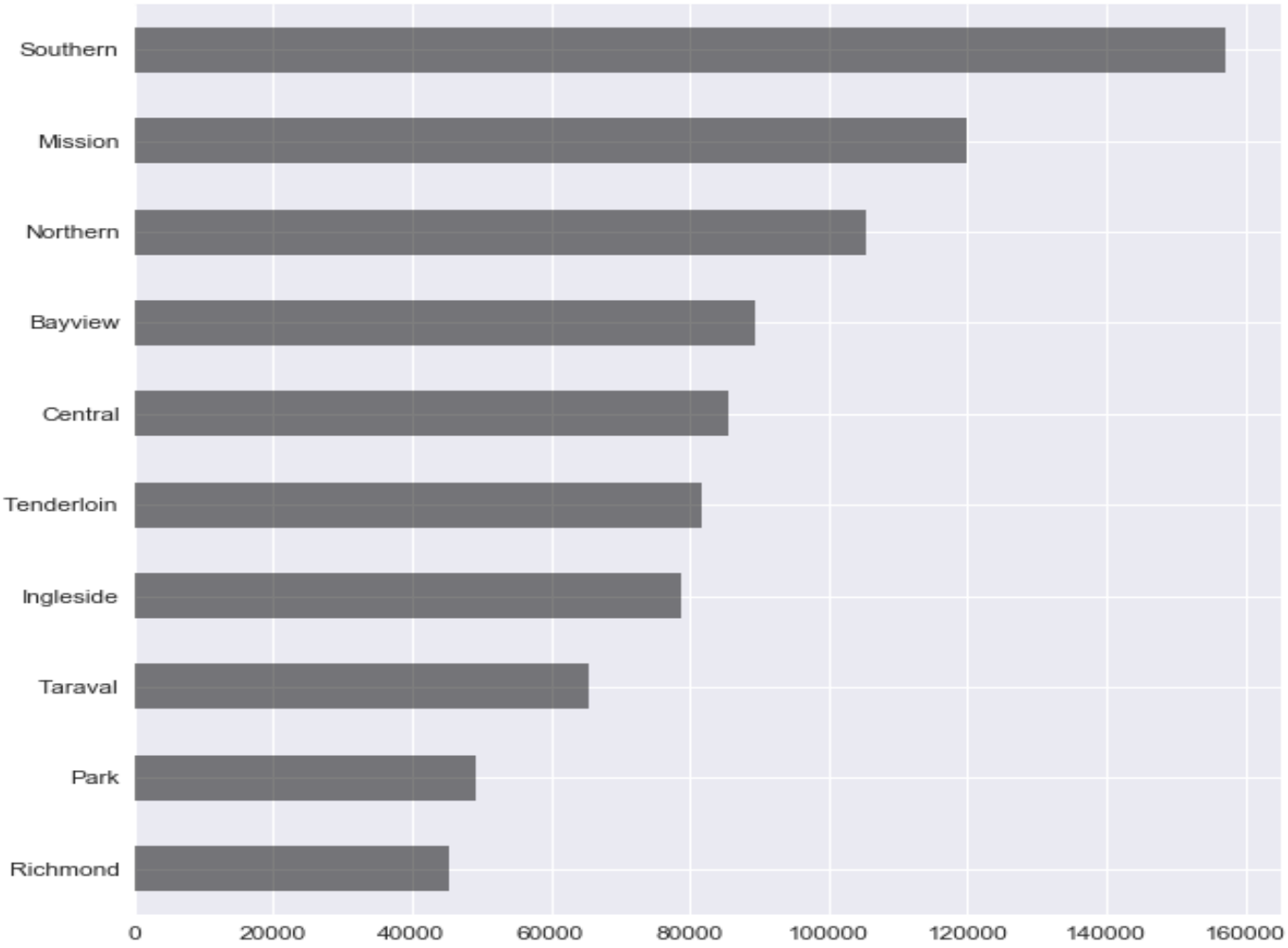

Figure 2: Bar chart visualization
(Number of crimes per district)

Python matplotlib package

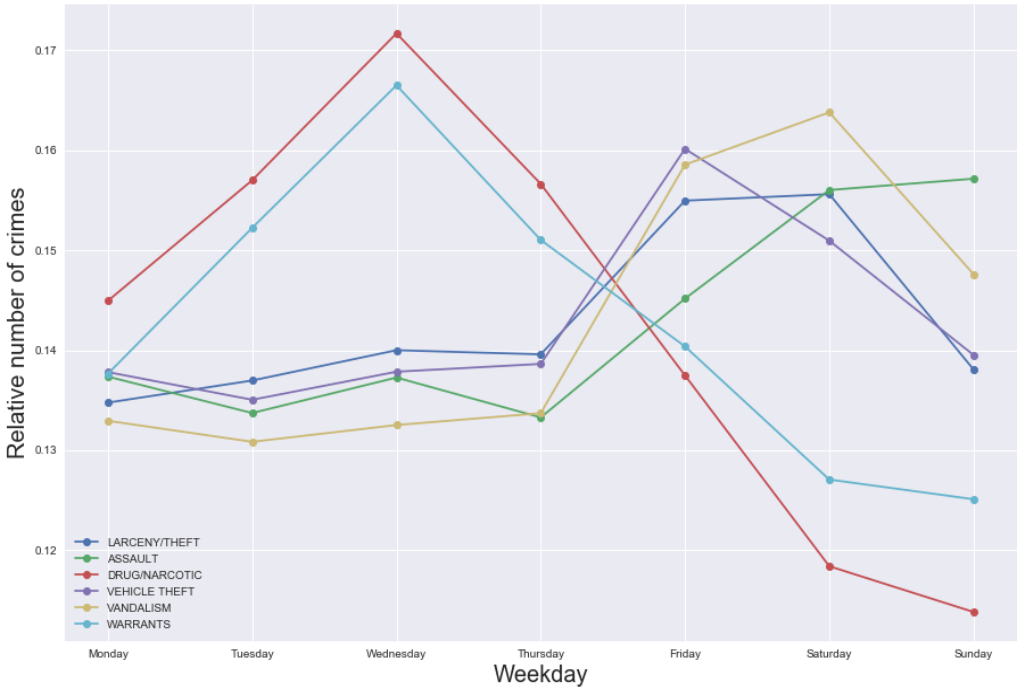

Figure 3: Line diagram (Relative number of top crimes per weekday)

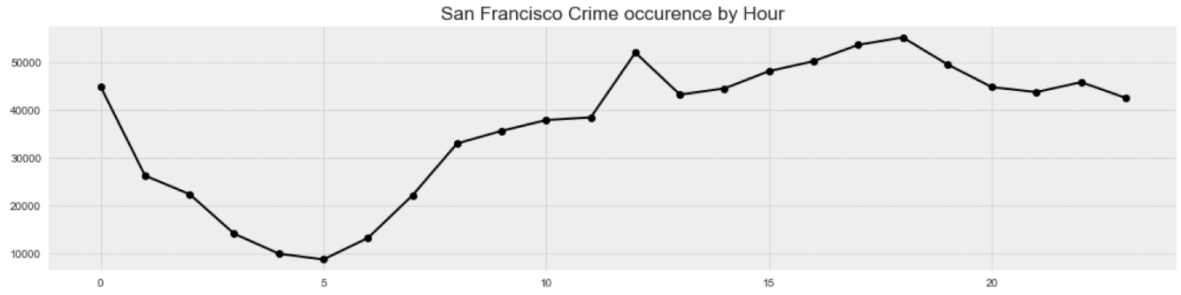

Figure 4: Line diagram (Number of crimes per Hour)

Python
matplotlib package

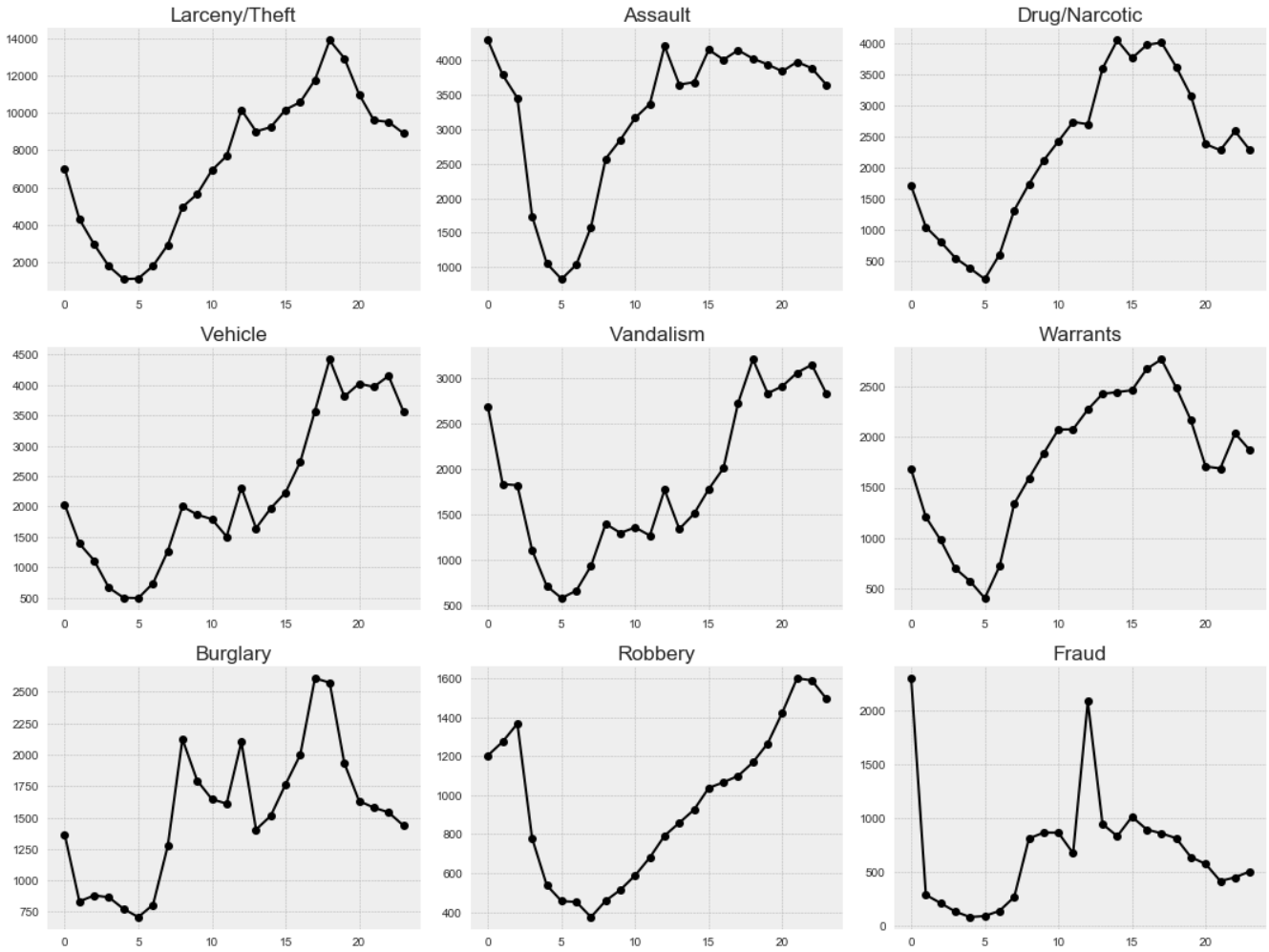

Figure 5: Line diagrams (Top Crimes occurrence by Hour)

R
ggplot2 library

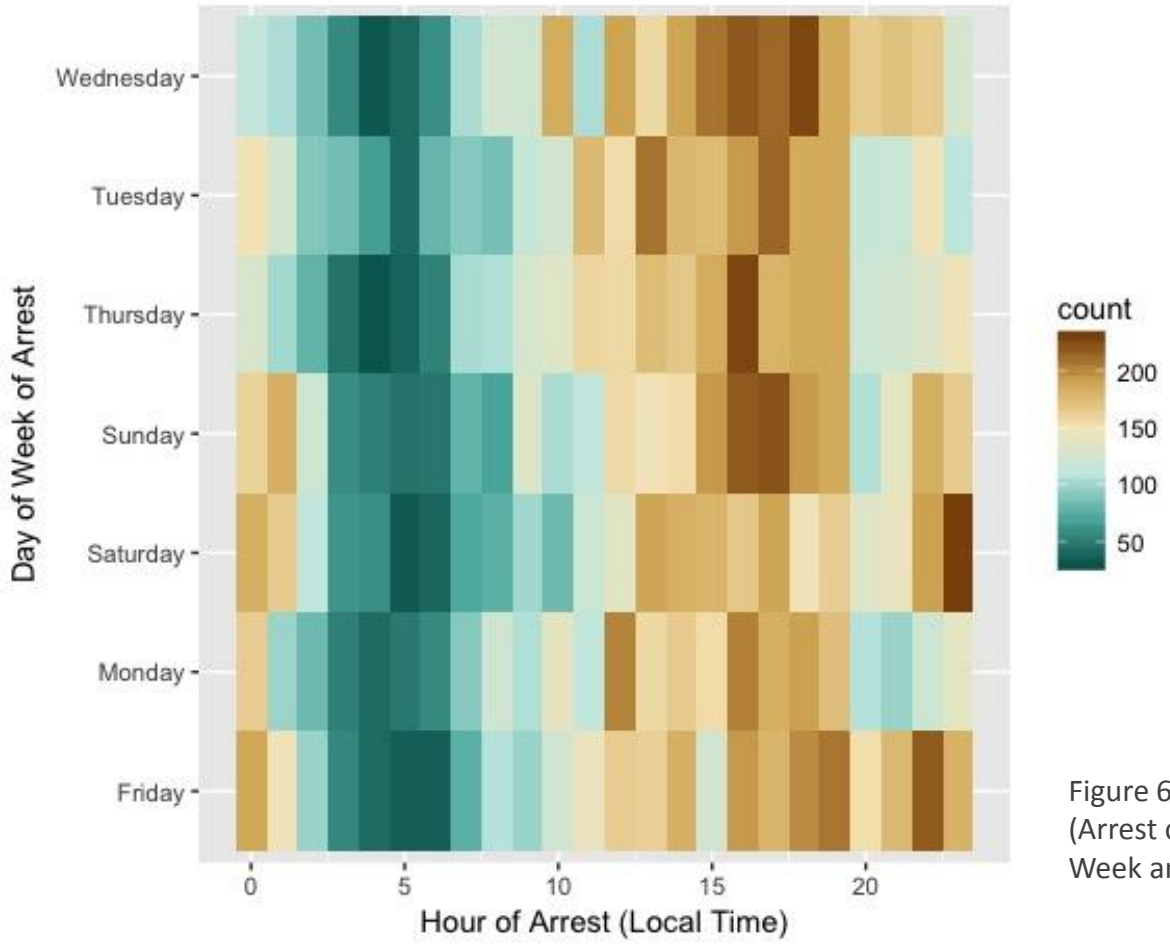

Figure 6: Heat map
(Arrest counts per Day of Week and Time)

R
ggplot2 library

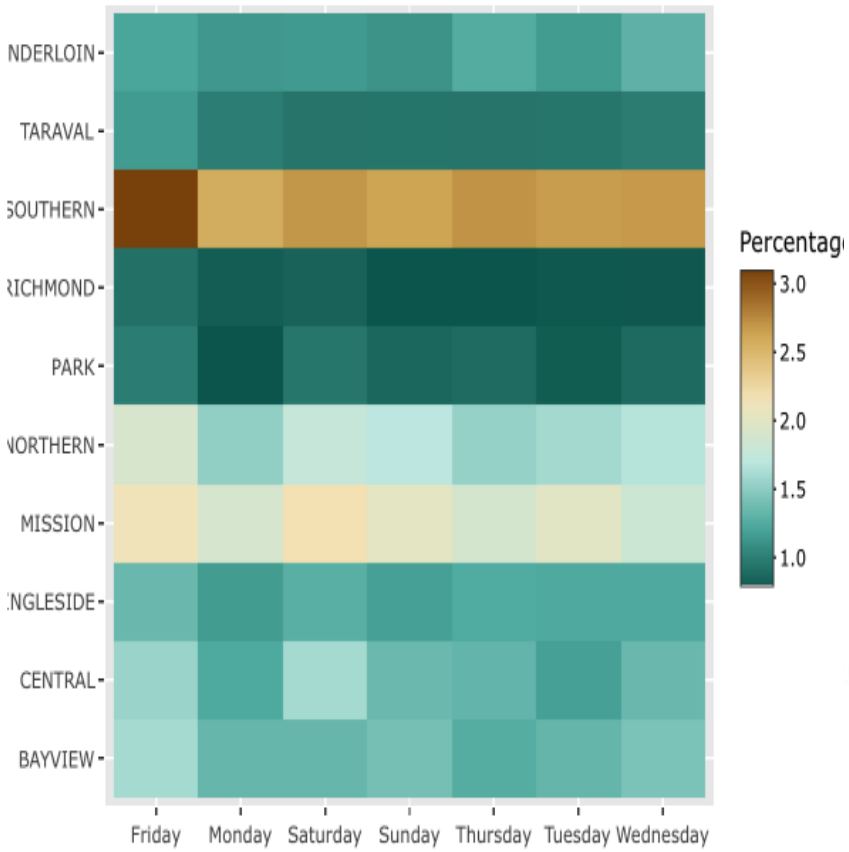

Figure 7: Heat map
(Day of the Week influence
on crime type and volume)

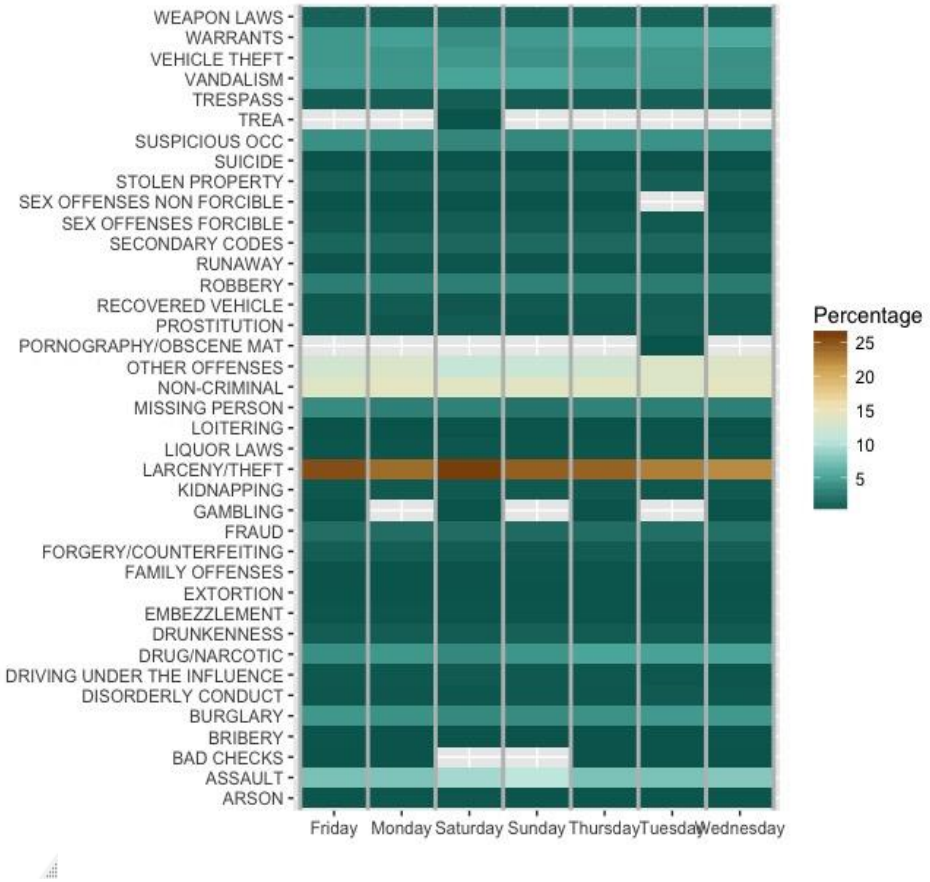

Figure 8: Heat map
(Relation between crime
types and day of the week)

ArcGIS Pro
Emerging Hot Spot Analyses

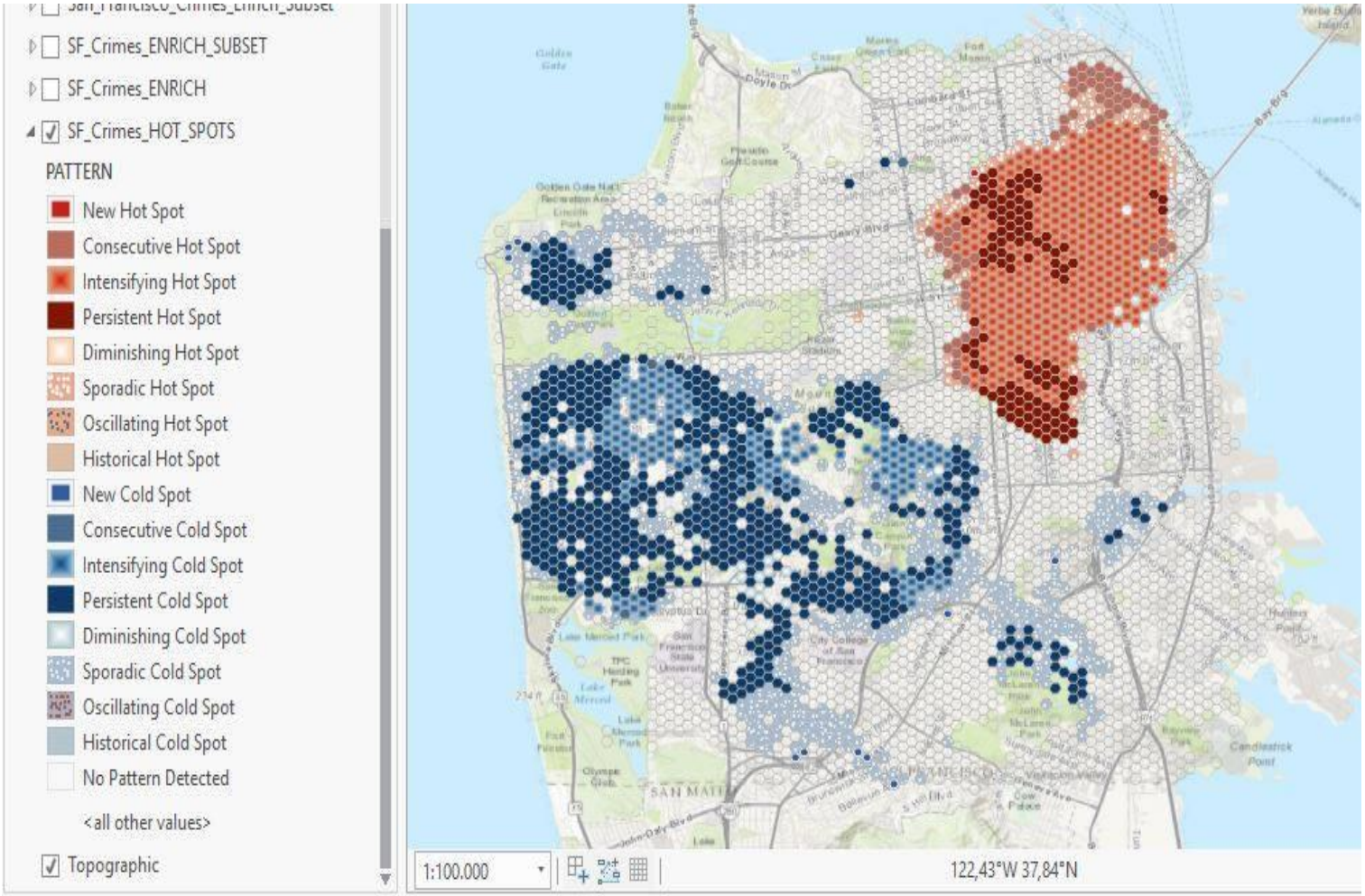

Figure 9: San Francisco Crimes Hot and Cold Spots visualization

ArcGIS Pro
Optimized Hot Spot Analyses

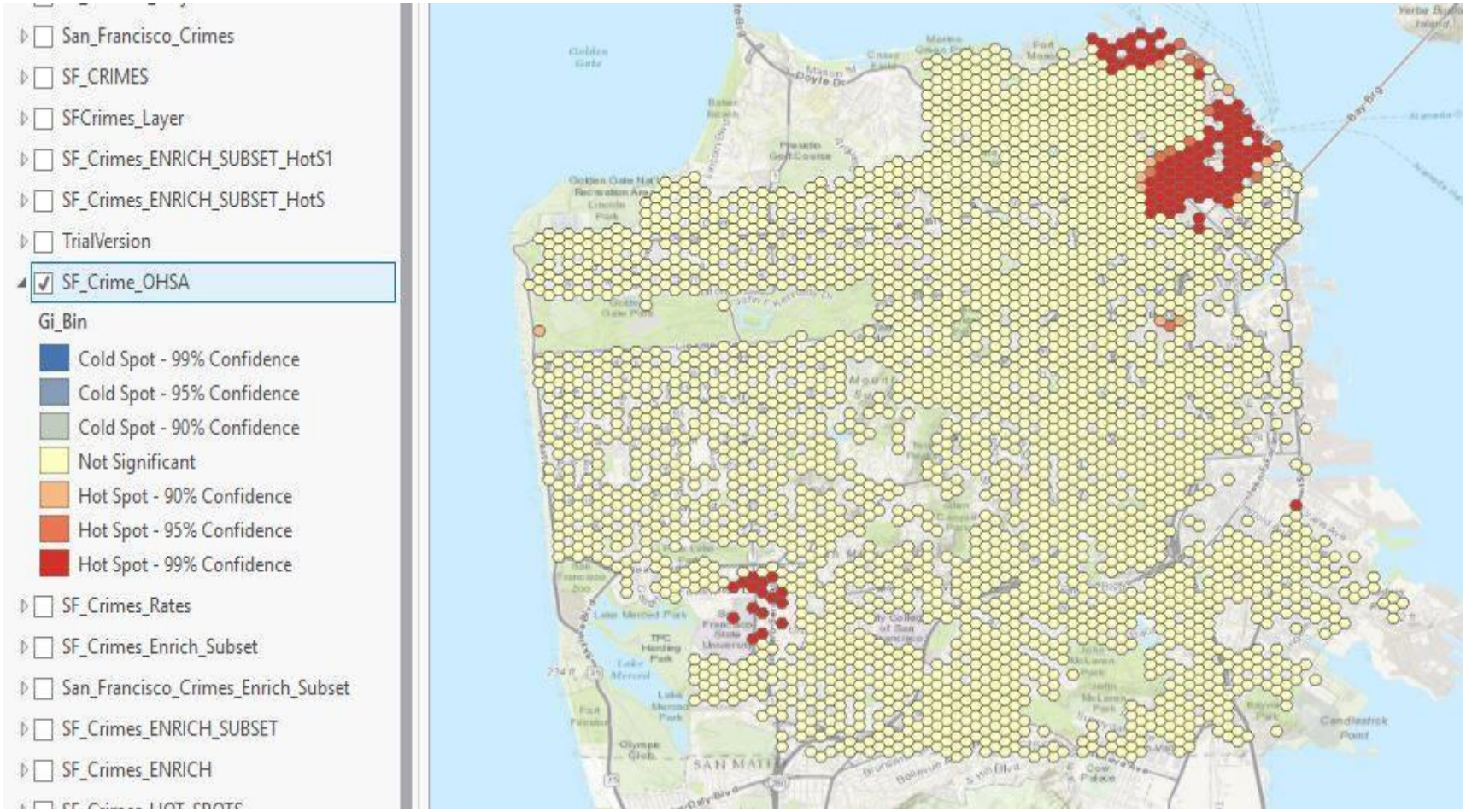

Figure 10: Areas with unusually high number of crimes

R
corrplot and hmisc package

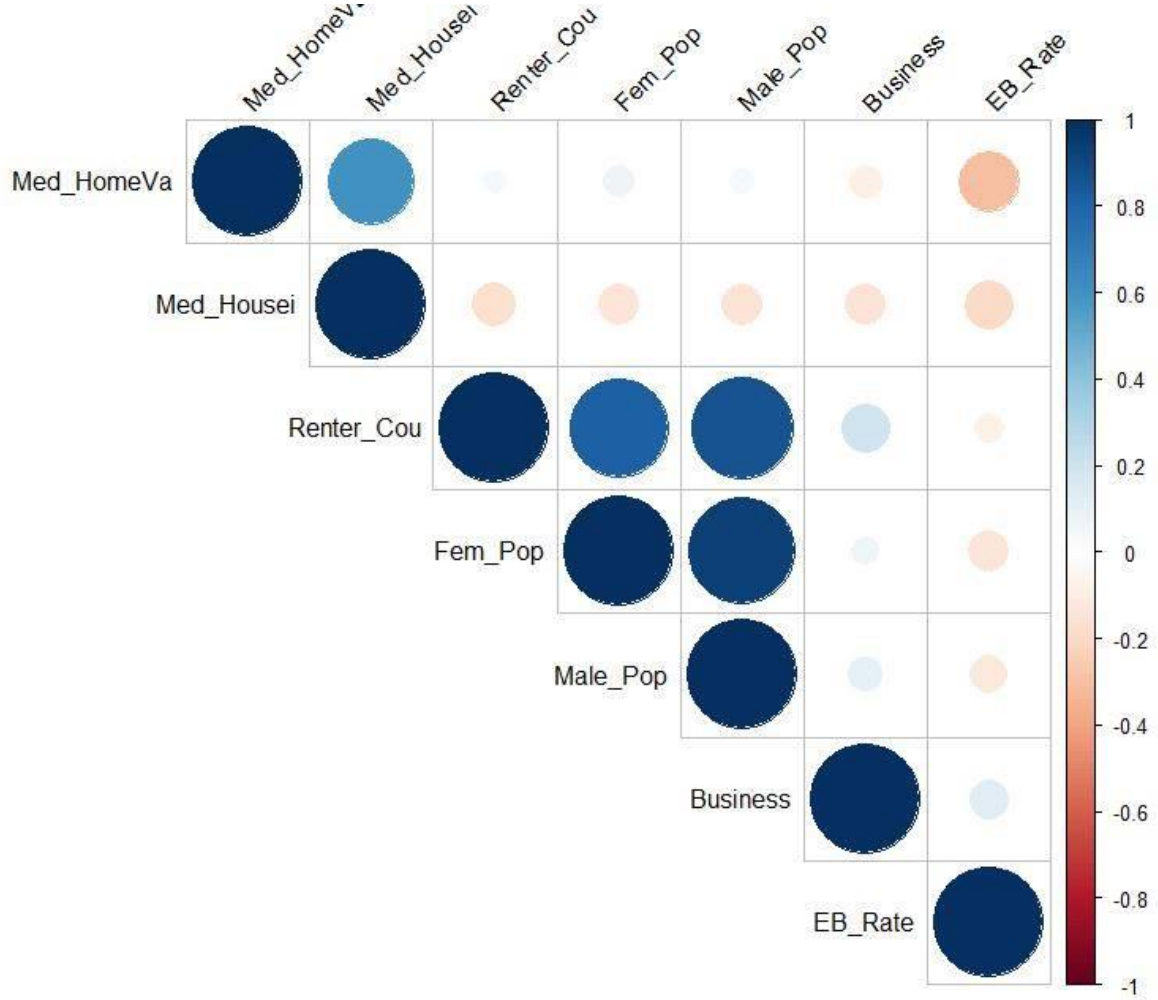

Figure 11: Correlation Matrix

Discussion

Python

- Used packages: pandas, numpy and matplotlib
- Stand alone script
- Easy and quick data pre-processing and handling
- Easy to understand and manipulate package
- Customizing and scalable graphics

Discussion

R

- Used packages: ggplot2, corrplot
- Stand alone script and R-ArcGIS integration
- Easy and quick data visualization (heat maps)
- Efficient statistical performance (correlation matrix)
- Easy to understand and manipulate package
- Customizing and scalable graphics
- Corrplot interpretation

Discussion

ArcGIS Pro

- Fast and clear performance (run the tool)
- Default legend
- Scalable
- Basemap
- Customising style and symbology

Conclusions

- Python and R libraries (matplotlib, ggplot2, corrplot) showed efficient performance in identifying and visualizing crime spatiotemporal relationships, crime trends and patterns which gave insights and more information about the whole dataset.

- ArcGIS Pro has very useful tools for crime analysis in its basic functionality, such as Emerging Hot Spot Analyses and Optimised Hot Spot Analyses.

- Combined, these tools leverage information extraction and visualisation out of criminal data.

Limitations and Future Work

- Data Scientist feedback
- Crime Analyst involvement

- Programming language choice
- GIS software choice
- Focus on prediction models
- Different data
- October = Danger; December = Safety. Why?
- Investigate why Larceny/Theft kept being highest criminal activity over the years in San Francisco?

Thank you for your attention.

Sources

(1) GIScomunity.com, Accessed: 20.05.2017.

(2) Popularity of Programming Languages, Accessed: 02.09.2017.

(3) Ciolobac, F. (2016, 11 16). What are the Top Programming Languages in the GIS World. *Geomatic Canada Magazine*.

(4) Robinson, A., Hardisty, F., Dutton, J., & Chaplin, G. (n.d.). Overview of Programming Languages for GIS. Retrieved 05/19/2017, Penn State University, https://www.e-education.psu.edu/geog583/node/67

(5) Piatetsky, G. (2017, 05). KDnuggets. Retrieved 06 06, 2017, from New Leader, Trends, and Surprises in Analytics, Data Science, Machine Learning Software Poll: http://www.kdnuggets.com/2017/05/poll-analytics-data-science-machine-learning-software-leaders.html

(6) Szczepankowska, K., Pawliczuk, K., & Źróbek, S. (2013). The Python programming language and its capabilities in the GIS environment. *International Geographic Information Systems Conference and Exhibition GIS Odyssey* (pp. 213-222). Zagreb, Croatia: Croatian Information Technology Society – GIS Forum.

(7) De Sarkar, A., Biyahut, N., Kritika, S., & Singh, N. (2012). An environment monitoring interface using GRASS GIS and Python. *Third International Conference on Emerging Applications of Information Technology* (pp. 235-238). Kolkota, India: IEEE.

(8) Anselin, L. (2012). From SpaceStat to CyberGIS. *International Regional Science Review, 35*(2), 131-157.

Questions?