TECHNOLOGY REVIEW: ‘statsmodels’ & ‘sqlite3’

John Ash
Hongyuan Liu
Wenbo Zhu
CSE 599 – 2/24/16

OUTLINE

- Project overview
- Project requirement being addressed by packages
- How the packages work
- Appeal of the packages
- Drawbacks of the packages

PROJECT OVERVIEW

- Transportation safety is a “hot” research area
- 2014 National Statistics/Motivation (NHTSA, 2016):
  - 29,989 fatal crashes
    - 32,675 fatalities
      - 1.07 per 100 million vehicle miles traveled (VMT)
      - 10.3 per 100,000 population
  - ~6.1 million police-reported crashes
  - ~2.3 million injuries
- Safety modeling is a common task performed by analysts
  - Crash frequency
  - Crash severity

REQUIREMENTS ADDRESSED BY statsmodels

- Conventional modeling techniques
- Crash frequency modeling
  - Regression of count data
    - Poisson regression
      - Limitations
    - Mixed-Poisson models
      - Negative binomial (basis for EB method)
- Crash severity modeling
  - Logistic regression
- Easily interpretable and extractable model results
  - Properties of model object

PACKAGE OVERVIEW: statsmodels

- "Python module that allows users to explore data, estimate statistical models, and perform statistical tests"
- Some features
  - Conventional regression (GLMs etc.)
  - Time series analysis
  - Non-parametric methods (kernel regression)
  - Plots (diagnostic [e.g., QQ plot], boxplot, ACF)
- Support
  - Active google group to ask/answer questions
  - Issues on github

HOW PACKAGE WORKS

- For regression…
  - Very similar to R in terms of formulas
  - Works with pandas DataFrames
  - Create models by specifying dependent variable as function of independent variables
  - Specify properties of model (e.g., family)
  - Uses common model fitting techniques (e.g., IRLS) and standard model forms
    - Gives options for use of alternate forms
      - Example: NB1 vs. NB2 (different formulations for variance)

```
import statsmodels.formula.api as smf
import statsmodels.api as sm
import pandas as pd
```

```
dataCrash = pd.read_csv("~/Desktop/CI_PI_Paper/try1.csv")
dataCrash.head()
```

	Crashes	F	LW	L	SW	CD	logf	off	Segment	AADT
0	0	15360	16.5	0.113	24	0.000000	9.639522	-0.570930	0.113	15360.000000
1	0	1660	16.5	0.116	20	8.620690	7.414573	-0.544727	0.116	1659.999999

COMPARISON TO R

Model development:

- modNB = smf.glm('Crashes ~ logf + LW +CD', data = dataCrash, family = sm.families.NegativeBinomial()).fit()
  - modNB = glm.nb(Crashes ~ logf + LW +CD, data = dataCrash)

Model summary:

- modNB.summary()
  - summary(modNB)

Model properties:

- modNB.aic
  - modNB$aic

EXAMPLE OF MODEL SUMMARY

In [4]:
```
model2.summary()
```

Out[4]:

Generalized Linear Model Regression Results

Dep. Variable:	Crashes	**No. Observations:**	1499
Model:	GLM	**Df Residuals:**	1495
Model Family:	NegativeBinomial	**Df Model:**	3
Link Function:	log	**Scale:**	2.12706923117
Method:	IRLS	**Log-Likelihood:**	-3061.4
Date:	Tue, 23 Feb 2016	**Deviance:**	1972.2
Time:	20:41:12	**Pearson chi2:**	3.18e+03
No. Iterations:	12		

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
Intercept	-5.7323	0.713	-8.037	0.000	-7.130 -4.334
logf	0.9965	0.075	13.233	0.000	0.849 1.144
LW	-0.1645	0.030	-5.499	0.000	-0.223 -0.106
CD	0.0339	0.019	1.770	0.077	-0.004 0.071

PACKAGE OVERVIEW: sqlite3

- SQL stands for Structured Query Language. SQL is used to communicate with a database.
- Oracle, Sybase, Microsoft SQL Server, Access, Ingres
- SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.
- Package: sqlite3 — DB-API 2.0 interface for SQLite databases
- Some features
  - DB interface & python interface
  - Faster and efficient in merging datasets

HOW PACKAGE WORKS

- Connect to Database

```
# create a db in sqlite3, establish a connection
conn = dbi.connect('crash_database')

# get a cursor for sql queries
cu = conn.cursor()
```

- Execute SQL statements through cursor

```
cu.execute('DROP TABLE IF EXISTS road')
cu.execute('DROP TABLE IF EXISTS elev')
```

WORKING WITH pandas

- Read CSV into data frame (pandas)

```
road = pd.read_csv(road_path)
elev = pd.read_csv(elev_path)
```

- Convert data frame into database table (pandas → sqlite3)

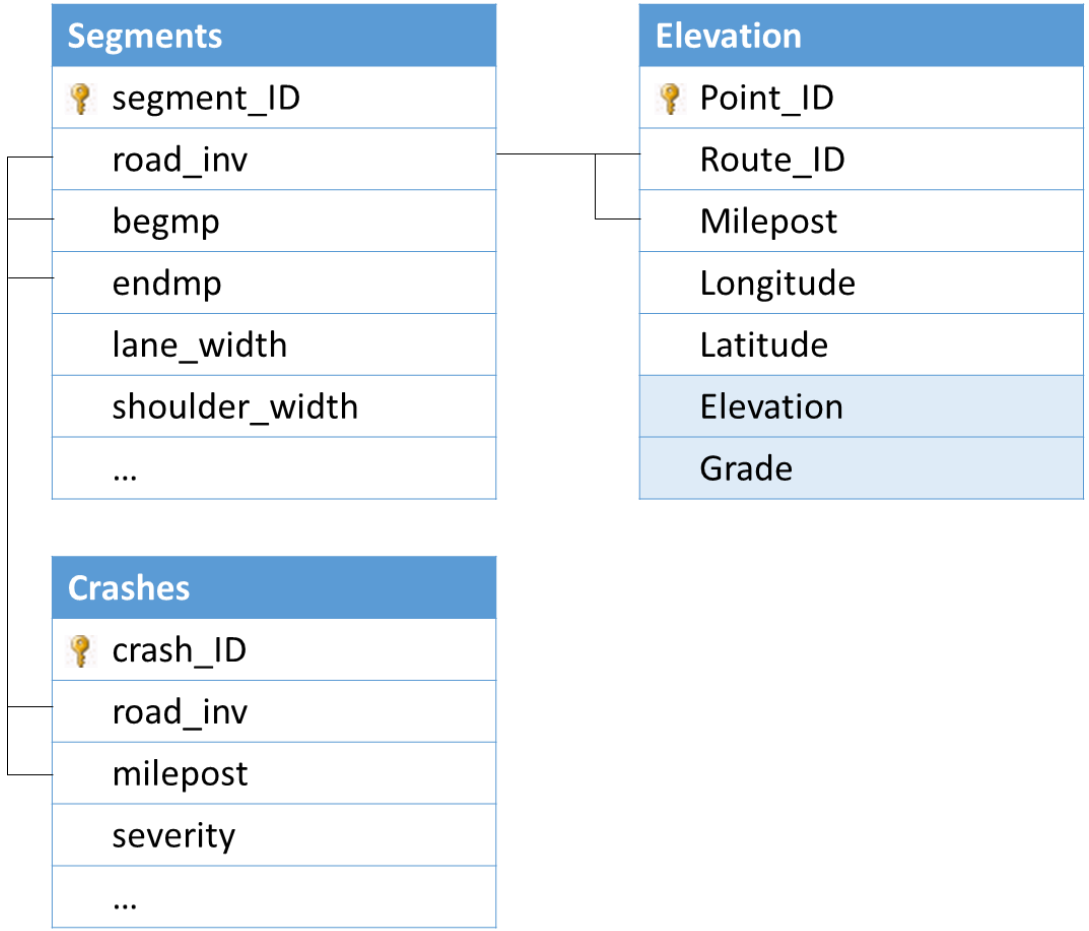
```
road.to_sql(name='road', con=conn)
elev.to_sql(name='elev', con=conn)
```

- Execute query statements and export to pandas data frame (sqlite3 → pandas)

```
# simple query example
# select all data from road table
select_all_query = 'SELECT * FROM road'
result_query = pd.read_sql(select_all_query, con=conn)
```

PACKAGE EXAMPLES

- Summarize crash counts & elevation profile for each road segment



APPEAL OF statsmodels

- Estimate statistical models
- Perform statistical tests
- Input-output tools for producing tables in common formats (Text, LaTex, HTML) and NumPy and Pandas compatible
- Extensive unit tests to ensure correctness of results
- Well-done documentation
- Active support team (Google group to get questions answered)
- Common dependencies: NumPy, SciPy and Pandas
- Open source

DRAWBACKS OF statsmodels

- Over 1,000 open bug reports
- Subject to updates
- There can be problems with data input and plotting for Python 3.x

APPEAL OF sqlite3

- Allows user to write sql queries to operate on data in their python application
  - Same syntax as in SQL Server, MS Access, SQLITE etc.
- No need to have database management software (e.g., SQL Server) installed on local machine
- Well-documented

DRAWBACKS OF sqlite3

- Cannot access data on a remote server
- Does not support Client/Server Applications

REFERENCES

National Highway Traffic Safety Administration (NHTSA). (2016). "Quick Facts 2014." *FARS Encyclopedia*, <http://www-nrd.nhtsa.dot.gov/Pubs/812234.pdf> (Feb. 23, 2016).

Seabold, Skipper, and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python." *Proceedings of the 9th Python in Science Conference*. 2010.

Evans, John. (2013). sqlite3, <https://pypi.python.org/pypi/db-sqlite3/0.0.1> (2-24-16).

Image (www5.vidowscomg.tk)