Highway Safety Project Prioritization Automation Tool

Background and User Guide

Transportation planning agencies need to prioritize highway safety improvement projects among a pool of proposed ones to better allocate the limited funds. In recent times, several states in the US use Excess Expected Crashes (EEC), a parameter dependent on Safety Performance Functions (SPFs) for ranking safety projects. However, this approach comes with several methodological limitations (e.g., the severity of the observed crashes and the magnitude of the projected crashes by the Empirical Bayes (EB) method are not considered). In 2022, Kentucky Transportation Cabinet (KYTC) developed and adopted a modified safety project scoring method for its Strategic Highway Investment Formula for Tomorrow (SHIFT) project prioritization process. This method considers crash severity and combines EB estimate and a "goal-driven" EEC metric in a multifactor score. Nonetheless, calculating a safety score for each project is a time and labor-intensive process. This user guide presents an open-source automation tool that can streamline this process with better efficiency and produces insights that would help to make robust decisions. This automation tool has been tested on the KYTC's list of potential SHIFT projects for the 2022 cycle. The source code is hosted in a public repository on GitHub which is easily accessible and customizable for a variety of potential uses.

BACKGROUND

Safety Performance Functions

SPFs are regression models that are used to predict the crash frequency for a specific roadway type (e.g., rural, urban) and geographic space (e.g., roadway segment, intersection, ramp, or any other special facility). Typically, these general linear models use Negative Binomial regression and are developed from a database of roadway segments (or intersections) containing information of segment length, historical crashes, and traffic volumes for each site (*1*). For SHIFT, five years of crash data are used, and the crashes are classified using the KABCO classification where K= fatal, A = incapacitating injury, B = non-incapacitating injury, C = possible injury, and O = no injury/property damage only (PDO). For incorporating crash severity into SHIFT's safety assessment, the SPFs are developed for two crash severity combinations:

KAB: More severe crashes

CO: Less severe crashes

The functional form of an SPF for roadway segments or ramps is shown in **Equation 1** and **Equation 2** shows the SPF of the intersections:

$$N_{SPF}(segment\ or\ ramp) = e^{\alpha} * L * AADT^{\beta} \quad (1)$$

$$N_{SPF}(intersection) = e^{\alpha} * AADT_{Major}^{\beta 1} * AADT_{Minor}^{\beta 2} \quad (2)$$

Where,

N_{SPF} = The predicted number of crashes by SPF;
L= Length of a segment;
AADT = Average Annual Daily Traffic;
$AADT_{Major}$ = Annual Average Daily Traffic of the major road
$AADT_{Minor}$ = Annual Average Daily Traffic of the minor road
α = Regression parameter for intercept
β, β_1, β_2 = Regression parameter for AADT.

The regression coefficients change based on the roadway type (e.g., rural/urban, divided/undivided,

two-lane/ multi-lane, etc.) and intersection configuration (e.g., number of legs, control types, etc.). Kentucky Transportation Center (KTC) has developed an open-source automation to automate the development of SPFs. The open-source automation tool is available on GitHub at http://github.com/irkgreen/SPF-R.

Empirical Bayes Method

The Empirical Bayes (EB) method is used to estimate the expected average crash count by combining the observed crash frequency for a site and the SPF predicted number of crashes. A weight parameter is used to combine the two crash measures and it is a function of how well the SPF model represents the dataset. If the SPF has a good correlation, the weight parameter places more emphasis on the predicted crashes, and vice versa. The EB method uses the following formulas (*2*):

$$N_{EB} = w * N_{SPF} + (1 - w) * N_{observed} \tag{3}$$

$$w = \frac{1}{1 + \frac{N_{SPF}/L}{\theta}} \tag{4}$$

Where,

N_{EB} = Expected average crash frequency by EB method;
w = weight factor, 0≤w≤1;
$N_{observed}$ = Historical crash frequency;
θ = Inverse overdispersion parameter (theta);
L = roadway segment length (L = 1 for intersections)

Excess Expected Crashes (EEC)

The difference between EB expected crashes and SPF predicted crashes is defined as EEC (See **Equation 5**). EEC measures the number of crashes occurring at a site more or less than expected for sites with similar characteristics (*3*).

$$EEC = N_{EB} - N_{SPF} \tag{5}$$

Goal-driven Excess Expected Crashes (EEC_{alt})

EEC represents the potential for reducing crashes up to the average of similar sites. However, a project has even more potential for crash reduction. The SHSP sets safety goals for every state to ensure improvement in all safety aspects and SHIFT safety uses a goal-driven, alternate version of EEC which is referred to as "Alternate EEC" or "EEC_{alt}". The equation for EEC_{alt} is shown in **Equation 6** with a graphical presentation in **Figure 1** (*4*).

$$EEC_{alt} = N_{EB} - \left(\frac{SHSP\ fatalities\ goal}{Current\ fatal\ crashes}\right) * N_{SPF} \tag{6}$$

According to the Kentucky 2020-2024 SHSP, there are 750 fatal crashes in Kentucky per year and the target is to reduce enough crashes that the annual fatalities fall at or below 500 by 2024 (*5*). This leads to the ratio of 500:750 or 2:3 for the SHSP goal to current fatalities.For Kentucky, the equation for goal-driven EEC_{alt} is mentioned below:

$$EEC_{alt}\ (KY) = N_{EB} - \left(\frac{2}{3}\right) * N_{SPF} \tag{7}$$

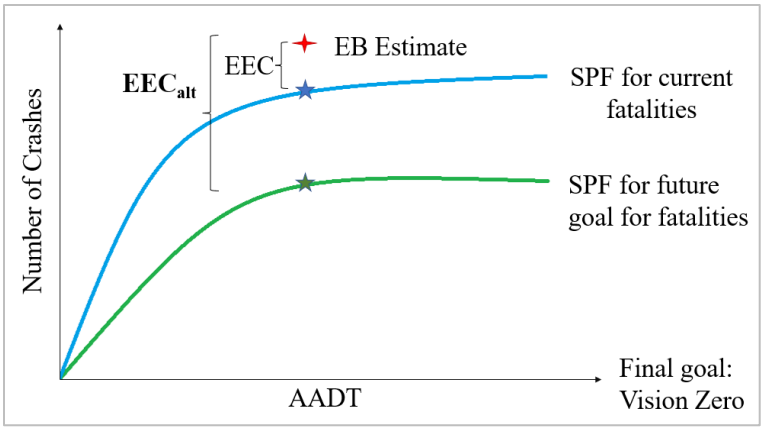

Figure 1 Graphical representation of EEC$_{alt}$

Final Safety Scoring Metric

For SHIFT 2020 cycle, the final safety metric for each project was calculated by taking the summation of all the segment EECs, intersection EECs, and ramp EECs that fall inside that project's beginning and ending mile points. However, SHIFT 2022 safety uses a multifactor scoring metric that incorporates three improvements. In this metric, KYTC included EB estimate as one of the components. EEC only compares the crash performance at a site with the average locations for that roadway type and AADT but fails to account for the magnitude of the overall number of crashes occurring at that site. EB estimate can address this problem as it adds information about the total number of crashes occurring at the site and provides the projected future crashes. The following improvements have been incorporated into the final ranking metric (*4*).

- It incorporates crash severity
- It integrates EB estimate
- It replaces EEC with EEC$_{alt}$

The final scoring metric S is expressed in **Equation 8** where the EB estimate and EEC$_{alt}$ from KAB and CO crashes are required for all the segments, intersections, and ramps under a project.

$$S = \frac{a}{2} * \sum N_{EB\,(KAB)} + \frac{a}{2} * \sum EEC_{alt\,(KAB)} + \frac{b}{2} * \sum N_{EB\,(CO)} + \frac{b}{2} * \sum EEC_{CO} \qquad (8)$$

Where,

a = proportion for KAB crash-related metrics,

b = proportion for CO crash-related metrics, and

a+b = 1

The proportions can be obtained from the crash frequency of each severity level and cost per crash from the common geographic location of the projects e.g., state, country, etc. The weighted average crash cost can be calculated using **Equation 9**. In Kentucky, the severity-weighted average cost per KAB crash is 89% and per CO crash is 11% of the total cost (see **TABLE 1**).

$$Weighted\ average\ cost = \frac{\sum Number\ of\ crashes\ by\ severity * cost\ per\ crash}{Total\ crashes} \qquad (9)$$

TABLE 1 Weighted average crash cost by crash severity groups [Source: KYTC, year 2019]

Severity	Number of crashes	Comprehensive cost	Weighted average cost	%
K	732	$9,281,571	$652,612 (KAB)	**89%**
A	2736	$537,913		
B	12257	$162,885		
C	359020	$102,957	$81,187 (CO)	**11%**
O	109313	$9,689		
Total			$733,799	100%

Finally, the equation for calculating the safety scoring metric for Kentucky is shown below:

$$S = 0.445 * \sum N_{EB\,(KAB)} + 0.445 * \sum EEC_{alt\,(KAB)} + 0.055 * \sum N_{EB\,(CO)} + 0.055 * \sum EEC_{CO} \quad (10)$$

AUTOMATING THE SAFETY PROJECT RANKING PROCESS

Calculating the multifactor scores manually for a large number of projects is a time and labor-intensive process. This section of the paper presents an open-source automation tool that can streamline this process with better efficiency.

Once the SPFs are developed, a database of the EB estimate and EEC_{alt} for all roadway segments can be created. Separate calculations are required for the estimation of EB and EEC_{alt} of the intersections and ramps. Finally, three datasets should be in hand when project-level scores are calculated. The summation of each factor needed for **Equation 8** should include the values of all the roadway segments, intersections, and ramps that fall inside that project (See **Equation 11**).

$$\sum X = \sum X_{Segments} + \sum X_{Intersections} + \sum X_{ramps} \quad (11)$$

Where,

$X = N_{EB(KAB)}$ or $N_{EB(CO)}$ or $EEC_{alt\,(KAB)}$ or EEC_{CO}

During the segmentation process, every segment gets a fixed beginning and ending mile points. Calculating the summation of each metric at the project level is quite straightforward if the beginning and ending mile points of a project coincide with those of the first and last roadway segments within the project. It becomes more complex when either the starting point, ending point or both mile points do not match with the segment mile points. In these cases, it is necessary to calculate the weighted value of each metric over the length of the roadway. The possible five scenarios (with equations) are described in **TABLE 2** with a schematic diagram presented in **Figure 2**.

TABLE 2 Descriptions and equations of project's metric calculation

Cases	Description	Equations
Case 1	The beginning and ending mile points of the project coincide with those of the segments.	$\sum X = X_1 + \cdots + X_n$
Case 2	Only the beginning mile point falls inside a segment.	$\sum X = \frac{L'}{L_1} * X_1 + \cdots + X_n$
Case 3	Only the ending mile point falls inside a segment.	$\sum X = X_1 + \cdots + \frac{L''}{L_2} * X_n$
Case 4	Both the beginning and ending mile points fall inside two different segments	$\sum X = \frac{L'}{L_1} * X_1 + \cdots + \frac{L''}{L_2} * X_n$
Case 5	Both the beginning and ending mile points fall inside the same segment	$\sum X = \frac{L'''}{L} * X_1$

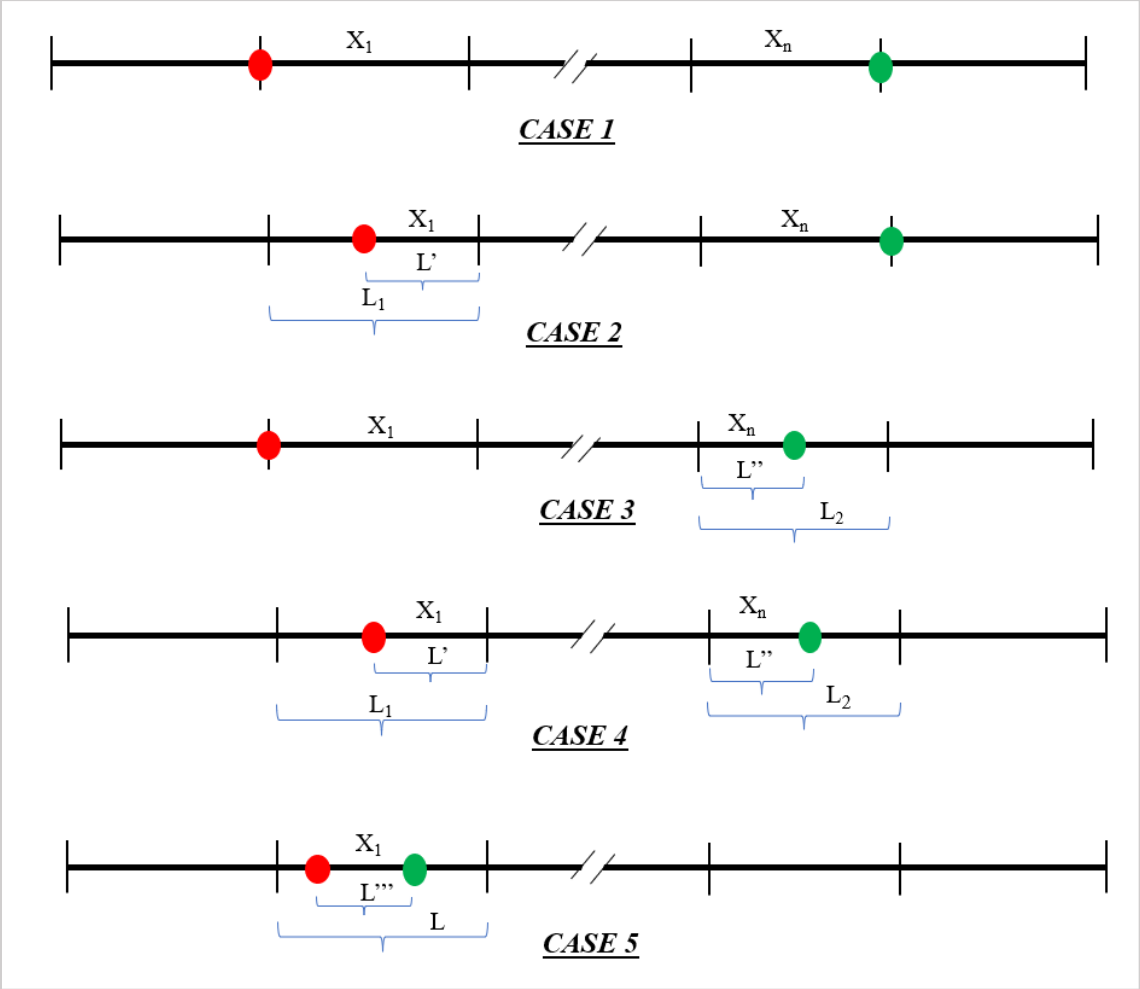

Figure 2 Visualization of five cases (Red and green dots refer to the beginning and the ending mile points of a project respectively)

PREPARATION OF INPUT DATA

The data preparation process requires four files in shapefile (.shp) format. It is necessary to maintain the format of the files because Spatial Joins are going to be performed at the beginning of the process. For example, a spatial join between the project list and the roadway segment database will match all the segments that fall under each project based on their relative spatial locations. **TABLE 3** summarizes the list and description of the four datasets required as inputs in the automation tool.

TABLE 3 Description of the required shapefiles as input in the automation tool

Dataset	Description	Required Fields
Shapefile 1	A file with all the project information	• Project_ID - A unique project identifier for each project • RT_Unique - A unique route identifier. KYTC populates its unique route identifiers Each state-maintained road in Kentucky. • BMP_prj and END_prj - The beginning and ending mile points of the routes
Shapefile 2	A file containing the EB estimate and EEC_{alt} of KAB and CO crashes of all the segments of the state-maintained roadways.	• RT_Unique – Every segment should have a unique route identifier. • BMP_seg and END_seg - The beginning and ending mile points of each segment • EB_KAB and EB_CO – EB estimate from KAB and CO crashes for each segment • EECalt_KAB and EECalt_CO – EECalt from KAB and CO crashes for each segment
Shapefile 3	A file containing the EB estimate and EEC_{alt} of KAB and CO crashes of all the intersections in the state.	• RT_Unique – Unique route identifiers of the two intersecting roads • MP_node – Mile point on the major road • EB_KAB and EB_CO – EB estimate from KAB and CO crashes for each intersection • EECalt_KAB and EECalt_CO – EECalt from KAB and CO crashes for each intersection
Shapefile 4	A file containing the EB estimate and EEC_{alt} of KAB and CO crashes of all the ramps in the state.	• RT_Unique – Every ramp should have a unique route identifier. • BMP_ramp and END_ramp - The beginning and ending mile points of each ramp • EB_KAB and EB_CO – EB estimate from KAB and CO crashes for each ramp • EECalt_KAB and EECalt_CO – EECalt from KAB and CO crashes for each ramp
CSV 1	A file containing the information shown in TABLE 1	• Severity – K, A, B, C, O • Number of crashes – Number of crashes of each severity level • Comprehensive cost – Cost per crash of each severity level.

SOURCE CODE

The script is written in Python and it is available on GitHub: https://github.com/rianatanzen068/Highway_Safety_Project_Prioritization_Tool . This is an online open-source code, which anyone can download and use for ranking highway safety improvement projects. The code can be modified for improvement and when the changes are committed to the GitHub repository, others can access and use the modification. Following is the cell-by-cell explanation of the interactive notebook:

- Cell 1: Imports the necessary packages: pandas, numpy and geopandas
- Cell 2-5: Imports the four necessary shapefiles mentioned in TABLE 3 as data frames
- Cell 6: Spatial join between projects and all roadway segments
- Cell 7: Spatial join between projects and all intersections
- Cell 8: Spatial join between projects and all ramps
- Cell 9: Defines a function "start_end" that labels the segments where the mile points of the project fall in between any segment (Cases 2-5 from TABLE 2).
    - If the beginning mile point is in the middle of a segment, it will get "S".
    - If the ending mile point is in the middle of a segment, it will get "E".
    - If both the beginning and ending mile points fall in between a segment, it will get "M".
    - Otherwise, it will get "O".
- Cell 10: Creates a new column using the function "start_end".
- Cell 11: Provides the counts of "S", "E", "M", and "O" tagged segments
- Cell 12: Defines a function "new_BMP" for replacing the BMPs of "S" and "M" segments with the project's BMP
- Cell 13: Defines a function "new_EMP" for replacing the EMPs of "E" and "M" segments with the project's EMP
- Cell 14: Adds three new columns
    - New_BMP (using the function from Cell 12)
    - New_EMP (using the function from Cell 13)
    - New_Length (difference between New_EMP and New_BMP)
- Cell 15: Calculates the length-weighted EB and EECalt for the segments.
- Cell 16: Keeps the necessary columns for the roadway segments
- Cell 17: Sums up each metric based on their "Project_ID"
- Cell 18: Renames columns with "_road" suffix
- Cell 19: Keeps the necessary columns from the matched intersections (from Cell 7)
- Cell 20: Sums up each metric based on their "Project_ID"
- Cell 21: Renames columns with "_int" suffix
- Cell 22: Keeps the necessary columns from the matched ramps (from Cell 8)
- Cell 23: Sums up each metric based on their "Project_ID"
- Cell 24: Renames columns with "_ramp" suffix
- Cell 25: From the main project shapefile creates a dataframe with only the project IDs
- Cell 26: Merges project IDs from Cell 25 and roadway segments from Cell 18 (Merge1)
- Cell 27: Merges "Merge1" and intersections from Cell 21 (Merge2)
- Cell 28: Merges "Merge2" and ramps from Cell 24 (Merge3)
- Cell 29: Replaces all the NaN with 0 (If a project only has segments but no intersections and ramps, it should get NaN from joins for those columns and should be replaced with 0)
- Cell 30: Sums up the roadway, intersection and ramp metrics for each project creating new columns: "EB_KAB", "EECalt_KAB", "EB_CO", and "EEC_CO"
- Cell 31: Importing csv file for weight calculation for KAB and CO
- Cell 32: Calculation of a and b from **Equation 8**.

- Cell 33: Creates a new column named "Final_Score" and calculates the final score using **Equation 8**
- Cell 34: Sorts the final scores in ascending order
- Cell 35: Adds a new column "RANK" with the ranks of each project (Projects with the highest scores get the highest ranks)
- Cell 36: Saves the resulting data frame to Comma Separated Values (.csv) format

CONCLUSION

The highway safety projects focus on the improvement of highway safety and aim to reduce the frequency and severity of crashes. However, the resources are limited and therefore, the funds should be allocated in a systemic and structured manner. The described automation tool minimizes the time and cost and maximizes the efficiency of calculating safety evaluation scores for project prioritization. Another major advantage of the tool is its open-source nature. Meaningful changes to the source code can be made and this can make the tool more user-friendly to safety professionals. This tool along with the "SPF-R" tool developed by Kentucky Transportation Center (KTC) is a complete package that can take care of the task of safety project prioritization from start to end with robust analysis.

REFERENCES

1. Srinivasan, R., and K. Bauer. *Safety Performance Function Development Guide: Developing Juridiction-Specific SPFs*. Publication FHWA-SA-14-005. Federal Highway Administration Office of Safety, 2013, p. 47.
2. AASHTHO. *Highway Safety Manual, First Edition*. American Association of State Highway and Transportation Officials, Washington, D.C., 2010.
3. Blackden, C., E. Green, R. Souleyrette, and W. Staats. *Automating Safety Performance Function Development to Improve Regression Models*. 2018.
4. Tanzen, R., Souleyrette, R., Wang, T. and Staats, W., 2022. Incorporating crash severity to improve highway safety project prioritization. Advances in transportation studies, (SI 3), pp.155-168.
5. *Kentucky Strategic Highway Safety Plan 2020-2024*. 2020.