




Safety Assessment of Roslagsbanan Rail System, 2021-2022

This manuscript ([permalink](#)) was automatically generated from [uiceds/cee-492-term-project-fall-2022-time@b06f483](#) on November 20, 2022.

Authors

- **Cecilia Karina Volpe Baridon**
·  [cvolpebaridon](#)
- **Elie Roudiere**
·  [elieroudiere](#)
Railway Group, KTH Royal Institute of Technology - Stockholm, Sweden
- **Berkhan Usta**
·  [berkanusta](#)

Dataset Description

The Roslagsbanan is a railway system operating in Stockholm. As part of an improvement in the transportation system, Stockholm Municipality invested in a new fleet of trains switching from X10p trains to the new X15p. In order to analyze the safety of the new trains, brake tests were conducted. During these tests, the Latronix Track Measurement (LTM) system was mounted on the train. The LTM uses laser beams, and an onboard computer to perform measurements over a railroad track while the train drives on the track at normal speed.

The data that will be used in this study was collected using the LTM system affixed to a Roslagsbanan vehicle. While the train was running over the sections, the level of each rail, alignment, curvature, cant, gauge, and twist were measured every 256mm. The data was compiled in csv files and was collected between October 2021 and May 2022 with an approximate interval of one month. In other words, a given section of track typically has around six measurement files. The total amount of data is 60 CSV files of 30,000 to 80,000 rows each, for a total of 1.5GB of data.

These data and measurements are property of Latronix AB of Sweden, and are used with their permission for research and educational purposes only.

In the CSV files the structure of columns is as follows:

- **Marker and Marker_Offset:** The kilometre post of the data point, where Marker is in km and Marker_offset is in m. In other words, if Marker=20 and Marker_offset=250.5, the kilometre post is 20,250.5
- **Coordinate_northing, Coordinate_easting, Latitude, Longitude:** As their name suggests, they are the coordinates of the data point in northings and eastings, as well as latitude and longitude.
- **Speed:** records the speed in km/h at which the measurement was taken. This is especially relevant as different operating speeds give different tolerances in terms of the measured properties.

- Level_left and Level_right D0, D1, D2, D3: The deviation of the longitudinal level (height) of the left and right rail in mm, compared to its reference position. D0, D1, D2, D3 correspond to different measurement wavelengths of the same data.
- Alignment_left and Alignment_right D0, D1, D2, D3: The deviation in longitudinal alignment ("straightness") of the left and right rail in mm, compared to its reference position. D0, D1, D2, D3 correspond to different measurement wavelengths of the same data.
- Track_gauge: The deviation in the spacing between the rails in mm, from a reference of 891mm.
- Curvature: The curvature of the track at that location, in m^{-1} .
- Cant: The cant (inclination) of the track at the location, in mm from the horizontal.
- Twist_3m, Twist_6m: The rate of change in cant over a 3m or 6m distance.
- Level_versine_left_5m, Level_versine_right_5m: Derived from Level_left and Level_right, this level deviation in mm corresponds to longitudinal versine measurement of the top of rail over a 5m distance. This historical metric is more commonly used in railway engineering and is thus provided.
- Alignment_versine_left_10m, Alignment_versine_right_10m: Similarly to the versine level, this measure of alignment in mm uses versines over a 10m distance.

Finally, Track_name, Track_class, Station_flag and Link_name, Pos_unfiltered Pos_filtered, NTP_sync, Cant_D1, Std_level, Std_alignment, Std_cant, Quality_1 are mainly for internal use and not part of the analysis.

Project Proposal

The main goal of this project is to identify patterns in the data, especially the track degradation of certain sections, and thus to analyze the change in railway safety during this time. The first approach will be to define degradation trends and build a model to process the data that finds them automatically. Then, the study will attempt to find external factors which caused the observed trends. The analysis will include graphics of the parameters over time as well as comparing them over the sections to provide a better understanding of the data visually.

This research will help assess system performance, identify possible maintenance schedules, and provide useful data for further research in this area. The study also falls within the scope of developing predictive maintenance in civil engineering and railways. Creating and improving analysis and predictive maintenance models such as this one can contribute to increase safety of rail networks and a higher quality of service for passengers. They will benefit from fewer unplanned service interruptions due to track maintenance and even fewer unsafe tracks causing line closures.

Exploratory Analysis

General Dataset Description:

This document will concentrate on Line 27 of the Roslagsbanan, which links Stockholms Ostra and Karsta. The reason that line 27 was chosen is because it has the most data out of the three line. The primary dataset for line 27 is comprised of sixty ".csv" files, each of which includes a collection of

measurements that were gathered from all, or a particular segment of, the line. Track measuring equipment, which was mounted on the trains and operated while the trial run was being conducted, was used to take the measurements. The dates of the measurements range from October 2021 to May 2022. In addition, the number of measurement runs that are performed on each sector of the track is differentiated from one another according to the relative importance of the sectors. For instance, the section of the track that extends between 27 and 38 kilometers has been measured in more than twenty different time points, whereas other sections do not have such a high number of runs. The map of the lines can be seen below in Figure 1.

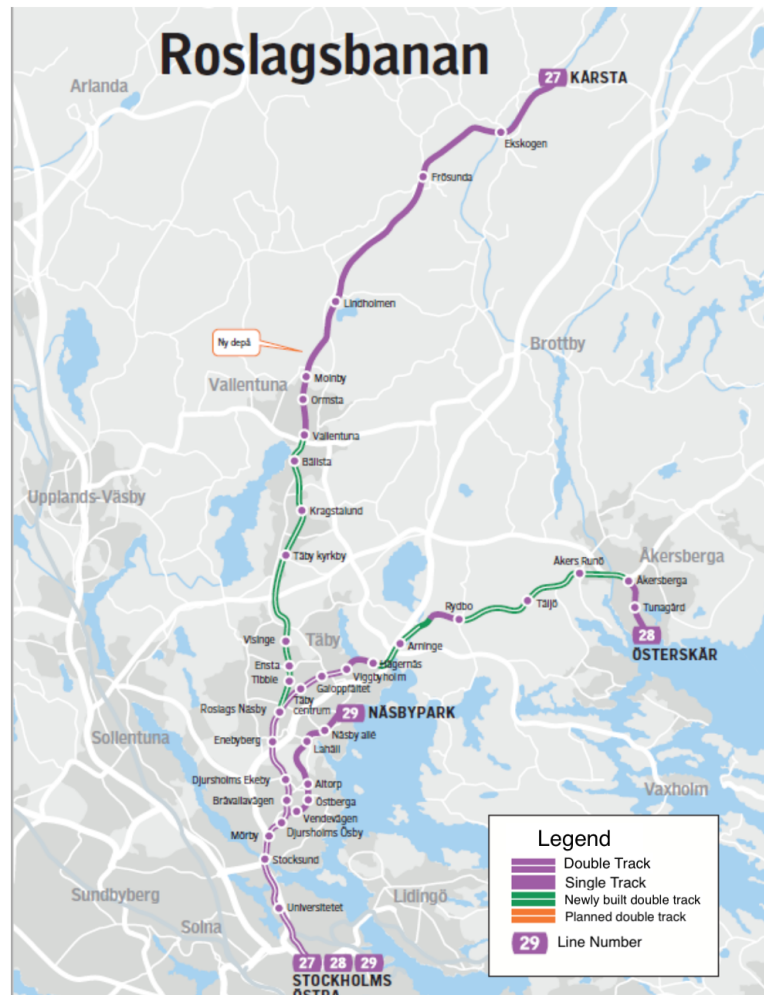


Figure 1: Line map overview of the Roslagsbanan. This study will focus on line 27 to Kårsta.

It is important to keep in mind that the data may require cleaning in some instances. For example, it has numerous NA cells. It also has some questionable data points, particularly during the time when the train was slowing down to a stop or starting to accelerate (i.e. close to 0 speed), which caused some outliers.

The date and time of the measurements are not hardcoded in the .csv document but only mentioned in the file name, as such the code for analyzing will have to take into account the name of the file when making comparisons. As shown in Figure 2, the measurement runs are also not completely synchronized, they must be matched or correlated in some way to build a reliable model.

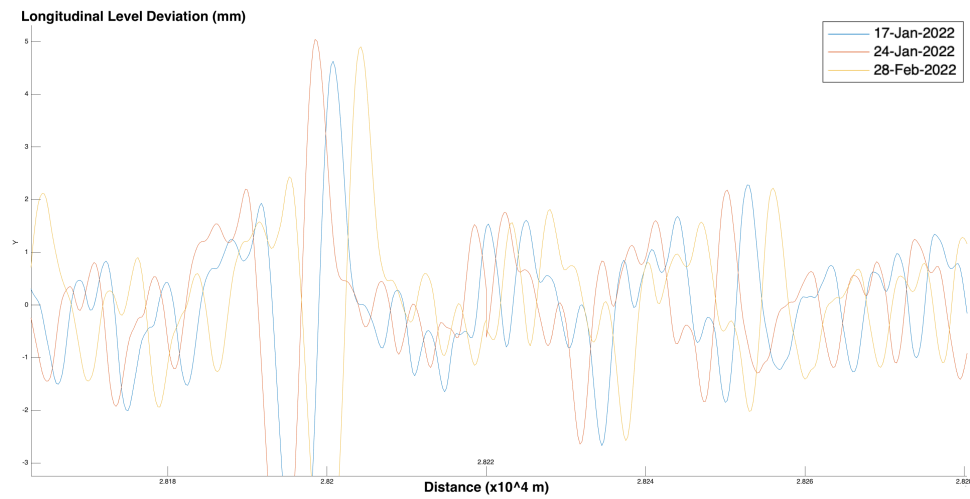


Figure 2: Plot showing the offset between measurement runs on three dates.

Summary Statistics:

In this section, we will discuss the science of data mining, which is the process of analyzing, and presenting large amounts of data in order to find hidden patterns and trends. Table 1 shows the general properties of the measurement files.

Table 1: Summary of general properties.

Properties	Value
Earliest measurement	21st October 2021
Latest measurement	10th May 2022
Line Length	41.700 km (25.91 miles)
Data points per file	20,000 to over 100,000

Table 2 shows the statistical analysis of certain measured parameters in order to show the typical range, mean, median and standard deviation of the values that will be used.

Table 2: Statistical analysis of the parameters for the Ostra-Karsta line, measurement date 02/08/2022.

Variable	Mean	Min	Median	Max	Standard Deviation
Curvature (1/m)	4.995e-5	-0.004	2.310e-6	0.004	0.001
Cant (mm)	1.629	-96.630	-0.490	91.393	32.158
Twist 3m	-0.003	-7.542	-0.001	6.703	1.172
Track Gauge Deviation (mm)	2.010	-7.989	1.779	19.149	1.721
Alignment Versine (Right) (mm)	-0.043	-51.756	-0.005	38.235	2.575
Level Versine (Right) (mm)	-0.003	-11.951	0.039	11.808	1.348

It can be observed that the mean and median values are close to 0, which is in accordance with the concept of a railway track having localised imperfections but being smooth on a more global scale. The standard deviation gives some insight as to how the data tends to vary throughout, compared to the generally low mean values.

Data Correlation:

An important fact of the parameters considered is that some of them are directly related in a mathematical sense. For example, cant or superelevation is proportional to the curvature, which means that when the track is being built, the superelevation is calculated based on the curvature at a given location, as follows:

$$E_e = \frac{G * V^2 * R}{g}$$

Where E_e is balancing superelevation or cant, G (track gauge), g (gravity), and V (line speed) are constants and R is curvature.[1]

Similarly, twist describes the rate of change of cant over a certain distance, and thus a close link between them is expected.

These parameters are summarised in Figure 3 below, courtesy of Kawasaki Track Technology[2]:



Figure 3: Summary diagram of the parameters studied in this report. Source: Kawasaki Track Technology LATGMS brochure.

Figure 4 shows these relations among the parameters for the track studied. From the curvature and cant plots, it can be observed that there is a strong relationship between them, as both exhibit similar patterns of variation. However, in the case of twist, similar patterns are not as clear. In spite of the fact that it is known to be mathematically correlated, additional investigation may be required in order to discern the pattern.

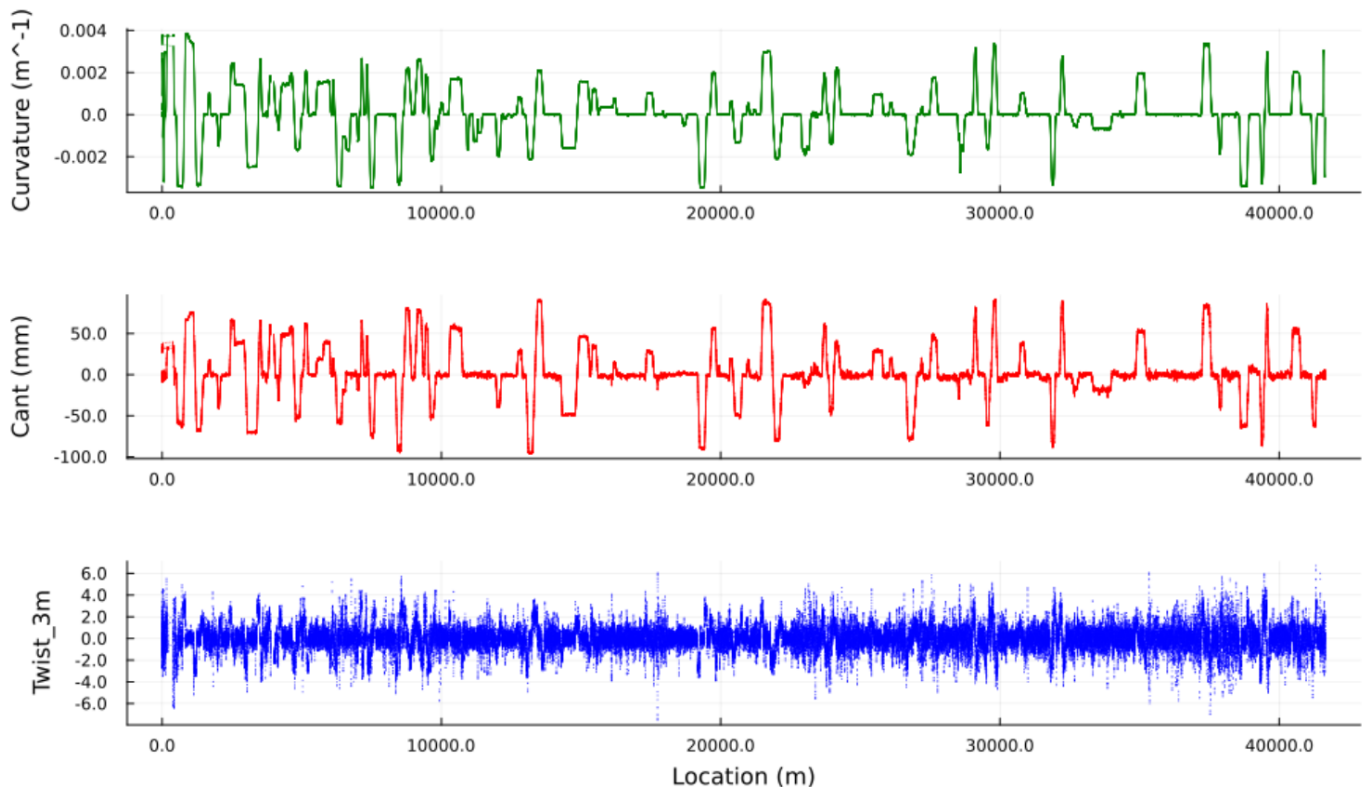


Figure 4: Ostra-Karsta line - Curvature, Cant and Twist vs Location.

As for the link between gauge and curvature, while there is no strict relationship between them when plotting the two parameters, a somewhat common pattern of variation can be observed. The plots indicate that at higher curvatures, deviation from the original gauge also increases. Figure 5 and Figure 6 evidence this fact.

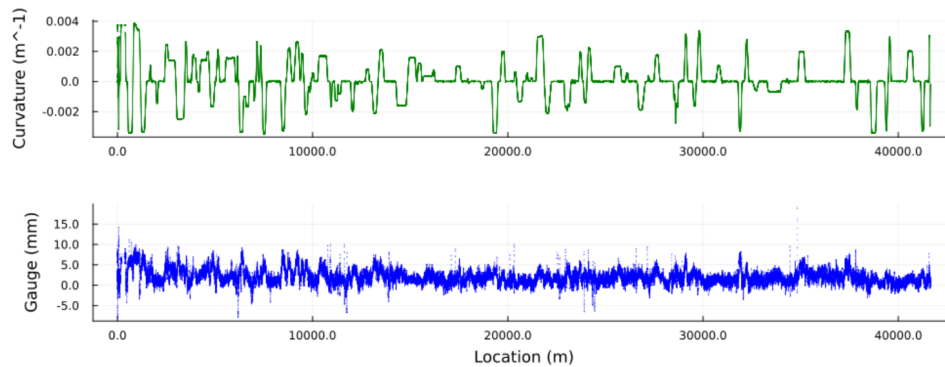


Figure 5: Ostra-Karsta line - Curvature and Gauge vs Location.

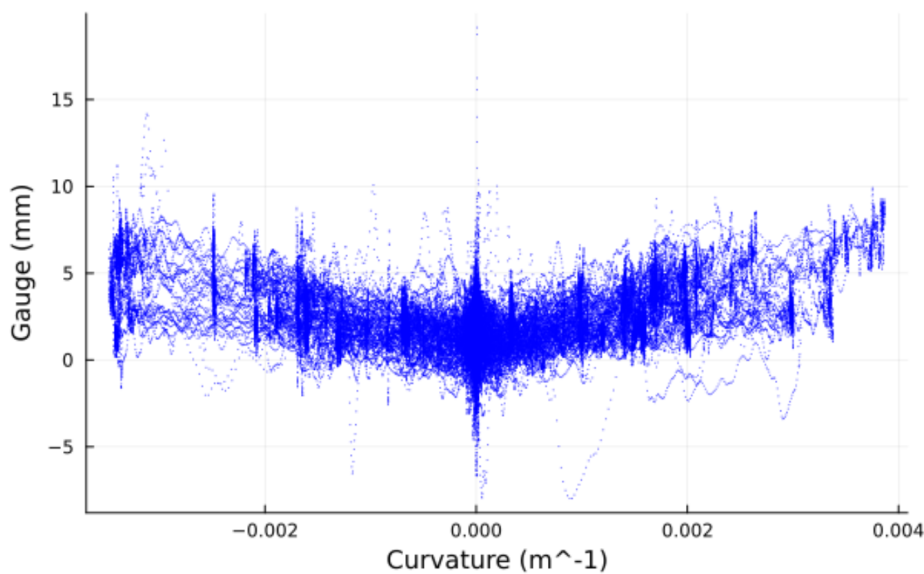


Figure 6: Ostra-Karsta line - Curvature vs Gauge.

In particular, on Figure 6, a general trend resembling an absolute value function can be discerned. Point density is very high around the point (0,0) of the graph, corresponding to straight track and intact gauge. As curvature increases (in the negative or positive), gauge also increases (only in the positive), seemingly showing that the rails tend to be stretched apart from one another in curved sections. This observation could play an important part in the predictive model.

Finally, Figure 7 shows the variation of the Alignment along the whole line.

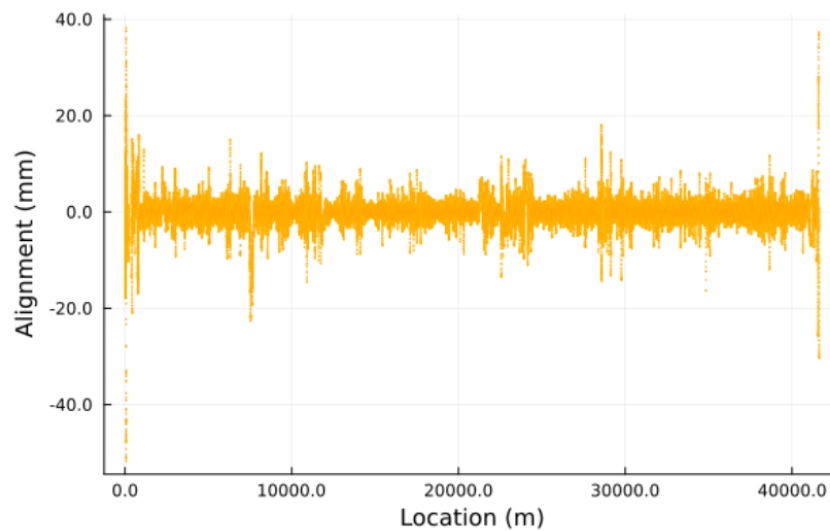


Figure 7: Ostra-Karsta line - Curvature vs Gauge.

Here it is interesting to note the outliers at the start and end points. At these locations the train was starting and stopping respectively, thus with a speed close to 0. This leads to unrealistic values which will need to be removed when exploiting the data in a model.

Predictive Model Considerations

Due to the somewhat limited number of independent variables already present inside the measurement set, there may be challenges when making a predictive model. Further evaluation of links between variables in the data will likely be necessary when making the model. Notwithstanding, the first part of the predictive model will try and find locations where the track degradation seems to form a clear trend (typically a linear increase in a given direction) over time. Here, time is the independent variable. From an initial analysis of the dataset and general knowledge of railway track engineering, it is likely this predictive model will yield exploitable results.

The second part of the predictive model could take into account external factors, in particular, temperature/weather changes and soil properties if obtainable. This could give us locations prone to developing trends in the short to medium term, which could be investigated by a maintenance team. However we have envisioned some possible issues, namely that obtaining and translating soil data into something usable by the model may be a challenge, and no link is guaranteed to appear (what if all soils along the line are similar?). A possible approach could be to grade soils based on properties, for example, a very stable soil could get a score of 1, and a “worse” soil from the point of view of building a railway line could get a high score, with appropriate intermediate values. However, obtaining soil data reliably for these locations is proving to be a challenge and may not be an option.

Finally, another research path for a predictive model would be to investigate more in depth the link between measured variables that are not mathematically correlated but seem to exhibit some kind of relationship, such as gauge and curvature earlier (although both are meant to be completely independent of a civil engineering point of view). The finality of this is being able to predict the state of certain properties of the track which are complex to measure, using knowledge of established properties that do not require special measuring gear. In this example, the curvature of the track is well-known and extremely stable, but small gauge variations necessitate state-of-the-art apparatus to measure hence the advantage of having a predictive model.

Predictive Modeling

Gauge vs Curvature Predictive Modeling

As shown in Figure 6, the Gauge and Curvature variables seem to be related. Since gauge measurements are not as easy to obtain as curvature measurements, finding the relationship between gauge and curvature could give technicians a way to measure gauge indirectly.

To start analyzing a model that fits the data, first, it can be observed that the data exhibit some symmetry around the curvature = 0 axis. Figure 8 shows the absolute values of curvature, and reflects how the positive and negative values are close in shape.

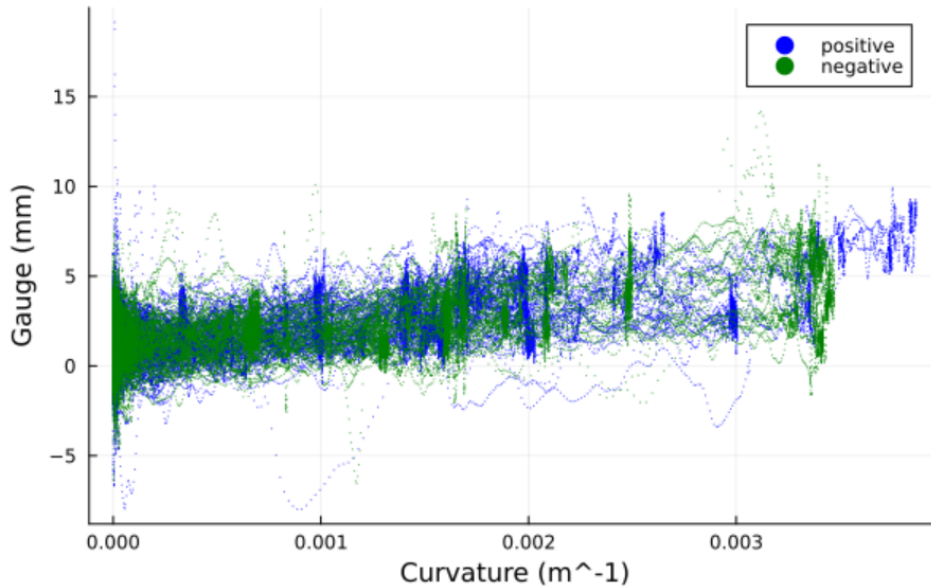


Figure 8: Ostra-Karsta line - Gauge vs Curvature Absolute Values.

Then, the method used to predict a model for this relationship was linear regression. Since it was unclear which model structure would best fit the data, two structures were assumed and compared to determine the best one. The two model structures used were,

$$\hat{y}_{linear} = \beta_1 * X + \beta_2$$

$$\hat{y}_{exp} = \beta_1 * X_{exp} + \beta_2 \text{ where } X_{exp} = e^x$$

\hat{y}_{linear} and \hat{y}_{exp} is the model prediction, the gauge prediction. X is the independent variable, the curvature. β_1 and β_2 are the parameters that are learnt using gradient descent.

In order to allow the gradient descent to work properly, the curvature data were normalized by subtracting the mean and dividing by the standard deviation.

The measure of error between the model predictions and the gauge raw data used is the mean squared error (MSE):

$$MSE = \sum_{i=0}^n \frac{(\hat{y}_i - y_i)^2}{n}$$

In order to avoid overfitting, the data is tested by the cross-validation method. Curvature values are divided into 6 groups, one part is used to predict the model, and the other part is used to evaluate it.

Finally, it calculated the root mean square error (RMSE) to compare which model gives more accurate results,

$$RMSE = \sqrt{\sum_{i=0}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

Figure 9 and Figure 10 show the graphs of the 6 data split groups for each model, detailing data training, data evaluation, and model prediction. The β_1 and β_2 values of the two models are shown in Table XX.

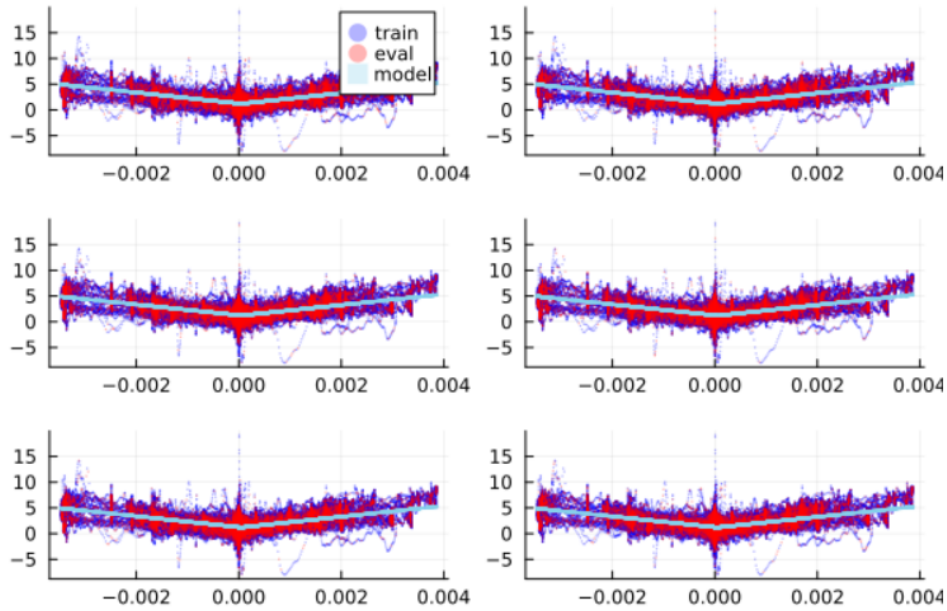


Figure 9: Ostra-Karsta line - Cross Validation in Linear Model for Gauge vs Curvature.

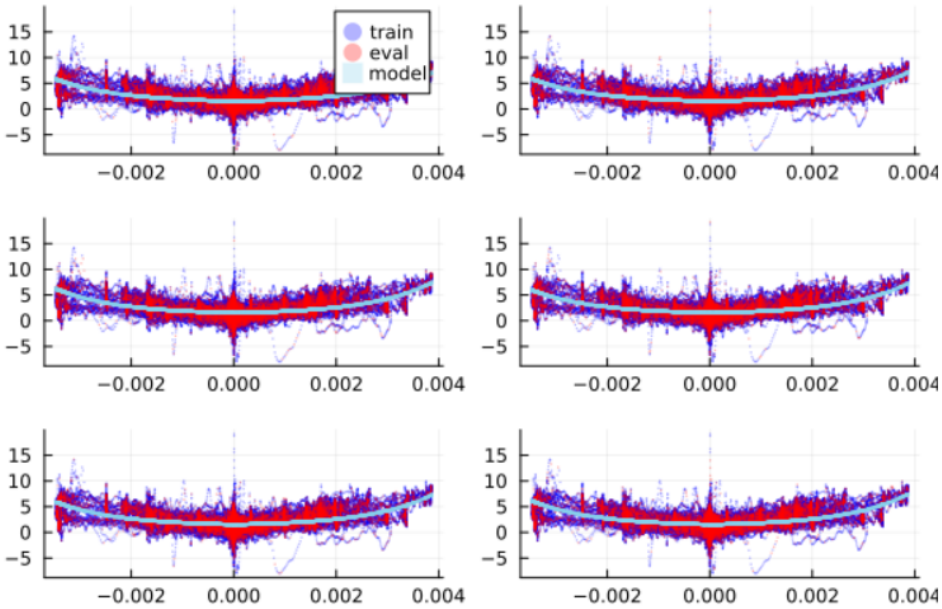


Figure 10: Ostra-Karsta line - Cross Validation in Exponential Model for Gauge vs Curvature.

Table 3: β_1 and β_2 values of the two models after cross-validation application .

β_1^{linear}	β_2^{linear}	β_1^{exp}	β_2^{exp}
1.298792	1.209706	0.271697	1.249403

1.311299	1.197514	0.273704	1.245887
1.307062	1.205466	0.271832	1.250013
1.309875	1.200968	0.272914	1.250626
1.306893	1.201039	0.272542	1.248218
1.317865	1.192273	0.274033	1.240401

Considering the above methodology and the average of β_1 and β_2 , the models found for the gauge-curvature relation were:

Linear Model:

$$Gauge_{linear} = \begin{cases} 1.309 * Curvature + 1.201 & \text{when } Curvature \geq 0 \\ -1.309 * Curvature + 1.201 & \text{when } Curvature < 0 \end{cases}$$

Linear Exponential:

$$Gauge_{exp} = \begin{cases} 0.273 * e^{Curvature} + 1.247 & \text{when } Curvature \geq 0 \\ 0.273 * e^{-Curvature} + 1.24 & \text{when } Curvature < 0 \end{cases}$$

The RMSE found for each model were:

$$RMSE_{linear} = 1.379$$

$$RMSE_{exp} = 1.434$$

The RMSEs are very close, but linear model is bit more accurate than the exponential one.

Figure 11 shows the data for negative and positive curvature values and the linear predicting model. Although the model is reliable and fits the data well as a whole, it seems to be rather difficult to capture the upper values.

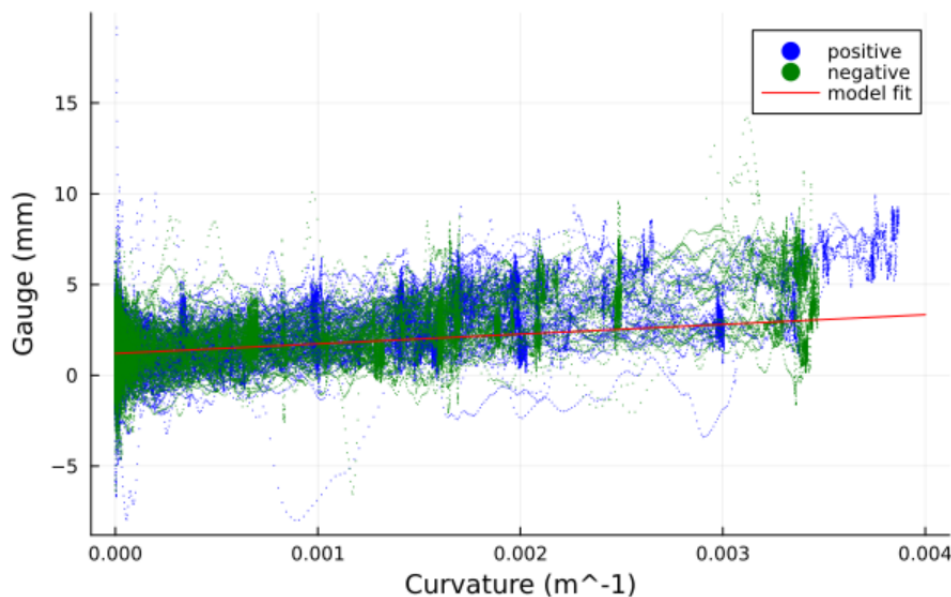


Figure 11: Ostra-Karsta line - Gauge vs Curvature Absolute Values and the Linear Predictive Model.

Time vs Longitudinal Level Predictive Modeling

Because of greater familiarity with the software, MATLAB was used for this part of the predictive modelling. The main tasks to be addressed were: 1) Find a way to import and manage the large quantity of .csv files; 2) Clean the files of NA data as well as unrelated measurements, and associate them with a measurement date; 3) Match or synchronize the different measurement runs as explained in the Exploratory Analysis; 4) Find a reasonable method for detecting locations showing steady and significant degradation of longitudinal level; 5) Aggregate those so that the person performing the analysis can judge the relevance of these locations

To address the first point, a starting idea was to use a loop of the “readtable” feature, importing one file at a time, and performing the desired operations on it. However, as a result of further research and speed concerns, the software uses the recent “datastore” feature of MATLAB, which can efficiently import, clean, and sort a very large amount of data, even beyond what can fit in memory. While Roslagsbanan data were still reasonable in size (60 files with a total size of approximately 1GB), this significantly increased the speed of the model compared to a more traditional “readtable” based loop and makes the method applicable to any size data set in the future. With this function, the desired columns (i.e., variables) can be selected and imported from the get-go, as opposed to bloating the memory with all the data and dropping large parts of it after the fact.

This brings on the second point, where the `dropna()` function was used to delete rows with N/A values easily. Then, a short custom function using “fileparts()” is capable of extracting the date of measurements from the file name and convert it to a date value usable by matlab – which will have its importance. Then, for each data file, the software finds peaks in the rail longitudinal level so long as they exceed alert level A (4mm), and they are spaced by a set distance, by means of the `findpeaks()` function. The distance was tweaked during the modelling to reach a satisfactory result, in the manner of a hyperparameter. A smaller distance shows more points but potentially more unwanted peaks while a larger distance is more restrictive but also could lead to missed locations. This process aims to reduce the number of analyzed points to only keep significant spikes in the positive or negative.

All these peaks and their respective position and time of measurement are assembled in a table and compared through a correlation analysis. The correlation function of MATLAB is used to compare the maximum value of peaks at the same location over a certain timespan, to show any possible trends. Only the locations where a certain threshold of correlation (typically 0.75, but this parameter was also adjusted during the analysis) is exceeded across at least four points of data are kept by the model for user review. Finally, this allows the model to output a shortlist of locations which display a spike or a sag with a degradation trend. The model also outputs a R^2 value to reflect the goodness of linear fit, and a crude estimate of when Alert C would be exceeded at this rate. While not absolute, perfect metrics these give a general idea of the rate of degradation and the criticality of the problem, and on a more macro level the relevance of the model output. After various adjustments to the parameters with a correlation threshold of 0.8 and peak spacing of 4m, the model output 100 points (including positive and negative trends). As could be expected from a physical perspective, twice as many negative (i.e., sinking) trends than positive trends appeared, possibly due to settlement and rail traffic. This gives some credibility to the algorithm.

In general, the model fulfilled its role and did find locations of clear, steady degradation over time which would have taken a human weeks of going over graphs and data manually – even for a domain professional. However not all results were satisfactory. In particular the model struggled with two aspects: - Measurement on double track sections with each track having a different level of degradation, as shown on figure [double track bad graph], tricked the model into seeing a trend which does not exist (as measurements were taken in both directions). - Locations which do show a trend but of extremely minute, though steady, variation. For example on figure [slowvar], following this

trend the level of the track would reach a concerning value after a decade or more. This is not so relevant for the engineer.

References

1. https://railroads.dot.gov/sites/fra.dot.gov/files/2019-12/Superelevation-Guidance_pdfa.pdf
2. <https://www.kawasaki-track.com/pdf/Kawasaki%20LATGMS%20Brochure.pdf>
3. **Sci-Hub provides access to nearly all scholarly literature**
Daniel S Himmelstein, Ariel Rodriguez Romero, Jacob G Levernier, Thomas Anthony Munro, Stephen Reid McLaughlin, Bastian Greshake Tzovaras, Casey S Greene
eLife (2018-03-01) <https://doi.org/ckcj>
DOI: [10.7554/elife.32822](https://doi.org/10.7554/elife.32822) · PMID: [29424689](https://pubmed.ncbi.nlm.nih.gov/29424689/) · PMCID: [PMC5832410](https://pubmed.ncbi.nlm.nih.gov/PMC5832410/)
4. **Reproducibility of computational workflows is automated using continuous analysis**
Brett K Beaulieu-Jones, Casey S Greene
Nature biotechnology (2017-04) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6103790/>
DOI: [10.1038/nbt.3780](https://doi.org/10.1038/nbt.3780) · PMID: [28288103](https://pubmed.ncbi.nlm.nih.gov/28288103/) · PMCID: [PMC6103790](https://pubmed.ncbi.nlm.nih.gov/PMC6103790/)
5. **Bitcoin for the biological literature.**
Douglas Heaven
Nature (2019-02) <https://www.ncbi.nlm.nih.gov/pubmed/30718888>
DOI: [10.1038/d41586-019-00447-9](https://doi.org/10.1038/d41586-019-00447-9) · PMID: [30718888](https://pubmed.ncbi.nlm.nih.gov/30718888/)
6. **Plan S: Accelerating the transition to full and immediate Open Access to scientific publications**
cOAlition S
(2018-09-04) <https://www.wikidata.org/wiki/Q56458321>
7. **Open access**
Peter Suber
MIT Press (2012)
ISBN: 9780262517638
8. **Open collaborative writing with Manubot**
Daniel S Himmelstein, Vincent Rubinetti, David R Slochower, Dongbo Hu, Venkat S Malladi, Casey S Greene, Anthony Gitter
Manubot (2020-05-25) <https://greenelab.github.io/meta-review/>
9. **Opportunities and obstacles for deep learning in biology and medicine**
Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, ... Casey S Greene
Journal of The Royal Society Interface (2018-04) <https://doi.org/gddkhn>
DOI: [10.1098/rsif.2017.0387](https://doi.org/10.1098/rsif.2017.0387) · PMID: [29618526](https://pubmed.ncbi.nlm.nih.gov/29618526/) · PMCID: [PMC5938574](https://pubmed.ncbi.nlm.nih.gov/PMC5938574/)
10. **Open collaborative writing with Manubot**
Daniel S Himmelstein, Vincent Rubinetti, David R Slochower, Dongbo Hu, Venkat S Malladi, Casey S Greene, Anthony Gitter
PLOS Computational Biology (2019-06-24) <https://doi.org/c7np>
DOI: [10.1371/journal.pcbi.1007128](https://doi.org/10.1371/journal.pcbi.1007128) · PMID: [31233491](https://pubmed.ncbi.nlm.nih.gov/31233491/) · PMCID: [PMC6611653](https://pubmed.ncbi.nlm.nih.gov/PMC6611653/)

This manuscript is a template (aka “rootstock”) for [Manubot](#), a tool for writing scholarly manuscripts. Use this template as a starting point for your manuscript.

The rest of this document is a full list of formatting elements/features supported by Manubot. Compare the input (`.md` files in the `/content` directory) to the output you see below.

Basic formatting

Bold text

Semi-bold text

Centered text

Right-aligned text

Italic text

Combined *italics* and **bold**

~~Strikethrough~~

1. Ordered list item
2. Ordered list item
 - a. Sub-item
 - b. Sub-item
 - i. Sub-sub-item
3. Ordered list item
 - a. Sub-item

- List item
- List item
- List item

subscript: H₂O is a liquid

superscript: 2¹⁰ is 1024.

[unicode superscripts](#)⁰¹²³⁴⁵⁶⁷⁸⁹

[unicode subscripts](#)₀₁₂₃₄₅₆₇₈₉

A long paragraph of text. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Putting each sentence on its own line has numerous benefits with regard to [editing](#) and [version control](#).

Line break without starting a new paragraph by putting two spaces at end of line.

Document organization

Document section headings:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

A heading centered on its own printed page

Horizontal rule:

Heading 1's are recommended to be reserved for the title of the manuscript.

Heading 2's are recommended for broad sections such as *Abstract*, *Methods*, *Conclusion*, etc.

Heading 3's and Heading 4's are recommended for sub-sections.

Links

Bare URL link: <https://manubot.org>

[Long link with lots of words and stuff and junk and bleep and blah and stuff and other stuff and more stuff yeah](#)

[Link with text](#)

[Link with hover text](#)

[Link by reference](#)

Citations

Citation by DOI [3].

Citation by PubMed Central ID [4].

Citation by PubMed ID [5].

Citation by Wikidata ID [6].

Citation by ISBN [7].

Citation by URL [8].

Citation by alias [9].

Multiple citations can be put inside the same set of brackets [3,7,9]. Manubot plugins provide easier, more convenient visualization of and navigation between citations [4,5,9,10].

Citation tags (i.e. aliases) can be defined in their own paragraphs using Markdown's reference link syntax:

Referencing figures, tables, equations

Figure [12](#)

Figure [13](#)

Figure [14](#)

Figure [15](#)

Table [1](#)

Equation [1](#)

Equation [2](#)

Quotes and code

Quoted text

Quoted block of text

Two roads diverged in a wood, and I—
I took the one less traveled by,
And that has made all the difference.

Code `in the middle` of normal text, aka `inline code`.

Code block with Python syntax highlighting:

```
from manubot.cite.doi import expand_short_doi

def test_expand_short_doi():
    doi = expand_short_doi("10/c3bp")
    # a string too long to fit within page:
    assert doi == "10.25313/2524-2695-2018-3-vliyanie-enhansera-copia-i-
        insulyatora-gypsy-na-sintez-ernk-modifikatsii-hromatina-i-
        svyazyvanie-insulyatornyh-belkov-vtransfetsirovannyh-geneticheskikh-
        konstruktsiyah"
```

Code block with no syntax highlighting:

```
Exporting HTML manuscript
Exporting DOCX manuscript
Exporting PDF manuscript
```

Figures



Figure 12: A square image at actual size and with a bottom caption. Loaded from the latest version of image on GitHub.

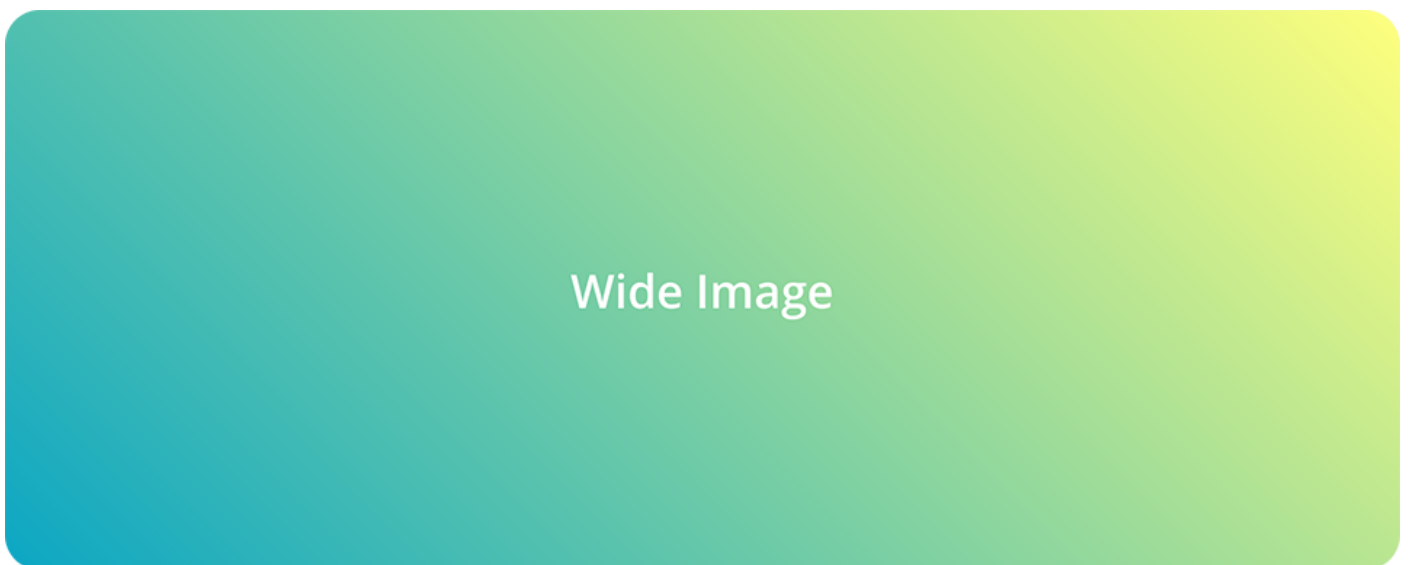


Figure 13: An image too wide to fit within page at full size. Loaded from a specific (hashed) version of the image on GitHub.

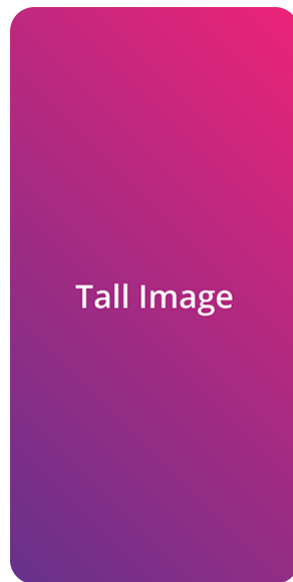


Figure 14: A tall image with a specified height. Loaded from a specific (hashed) version of the image on GitHub.



Figure 15: A vector `.svg` image loaded from GitHub. The parameter `sanitize=true` is necessary to properly load SVGs hosted via GitHub URLs. White background specified to serve as a backdrop for transparent sections of the image.

Tables

Table 1: A table with a top caption and specified relative column widths.

<i>Bowling Scores</i>	Jane	John	Alice	Bob
Game 1	150	187	210	105
Game 2	98	202	197	102
Game 3	123	180	238	134

Table 2: A table too wide to fit within page.

	Digits 1-33	Digits 34-66	Digits 67-99	Ref.
pi	3.14159265358979323846264338327950	288419716939937510582097494459230	781640628620899862803482534211706	<code>piday.org</code>
e	2.71828182845904523536028747135266	249775724709369995957496696762772	407663035354759457138217852516642	<code>nasa.gov</code>

Table 3: A table with merged cells using the `attributes` plugin.

	Colors	
Size	Text Color	Background Color
big	blue	orange
small	black	white

Equations

A LaTeX equation:

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

(1)

An equation too long to fit within page:

$$x = a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$$

(2)

Special

⚠ WARNING The following features are only supported and intended for `.html` and `.pdf` exports. Journals are not likely to support them, and they may not display correctly when converted to other formats such as `.docx`.

LINK STYLED AS A BUTTON

Adding arbitrary HTML attributes to an element using Pandoc’s attribute syntax:

Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Adding arbitrary HTML attributes to an element with the Manubot `attributes` plugin (more flexible than Pandoc’s method in terms of which elements you can add attributes to):

Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Available background colors for text, images, code, banners, etc:

white lightgrey grey darkgrey black lightred lightyellow lightgreen
lightblue lightpurple red orange yellow green blue purple

Using the [Font Awesome](#) icon set:

✓ ? ★ 🔔 ⛔ …



Light Grey Banner

useful for *general information* - manubot.org



Blue Banner

useful for *important information* - manubot.org



Light Red Banner

useful for *warnings* - manubot.org