STAtistical REliability

Automotive Reliability Engineering with OpenTURNS

N. BACHELIER, V. FEUILLARD, N. FORISSIER : Renault Group
O. BRAYDI, A. DUMAS, G. GARCIA, J.SCHUELLER : Phimeca

OpenTURNS User's day 2024
June 14^{th} 2024

Executive Summary

Context

Reliability Stress-Strength computations and Validation Plans are done with various methods & tools within the group that are not always shared and not always up to date.

Key points

- Python module with easy-to-use notebook interface
- Full sphinx documentation
- Features
  - (Non-)parametric distribution fitting using factories or kernel smoothing
  - Taking into account extreme values for distribution tail fitting
  - Stress projection
  - Graphs for engineering judgment

Table of Contents

Reliability pipeline and StaRe

Reliability pipeline

Customer Usage Modeling → Stress Prediction → Strength Specification → Validation Plans Specification

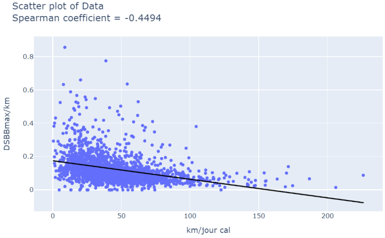
Customer Usage Modeling	Stress Prediction	Strength Specification	Validation Plans Specification
Inference on observation mileage per year Usage per km	Usage projection @ reliability horizons (km, years)	Strength computing according to specifications	# of parts @ # of cycles without failure

Customer data modelling

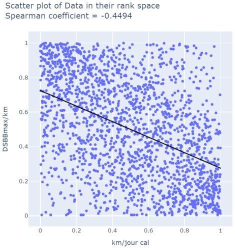
StaRe : Customer Usage observation

Customer Usage Modeling → Stress Prediction → Strength Specification → Validation Plans Specification

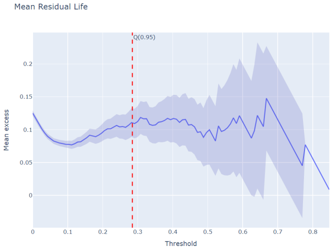
- Usage-Mileage scatter plot



- Usage-Mileage scatter plot in rank space



- Usage Mean Excess Plot for GDP threshold confirmation

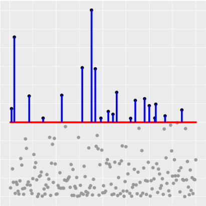


StaRe : Customer Usage observation

Details

- Raw data from csv imported as openturns.Sample/pandas.DataFrame
- Plotly visualization for enhanced interactivity
- As the module was developed with OTv1.21, Mean Excess Plot has been re-implemented

Exceedances

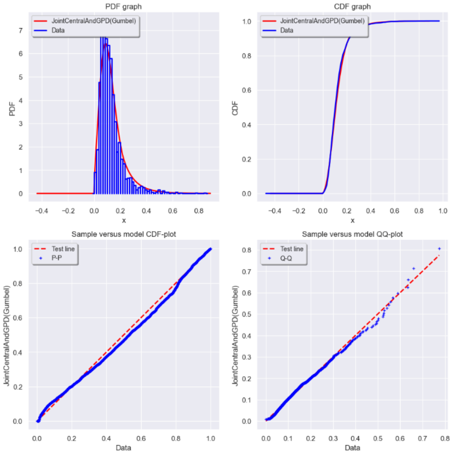


from Thomas Opitz, atelier statistique de la SFdS.

StaRe : Customer Usage modelling

- List of fitted distributions ordered by AIC or BIC or K score.
- Visual validation plots of the chosen distribution (here, Gumbel with GPD tail fit)

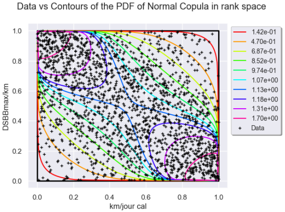
	Distribution - DSBBmax/km	AIC_1	BIC_1	Kolmogorov_1	Acceptance_1 (0.05)
0	JointCentralAndGPD(TruncatedDistribution(Gumbel(beta = 0.0577769, gamma = 0.0870315), bounds = [0, (2.27946) +inf[), upper)	-2.666281	-2.666281	1.668748e-03	False
1	JointCentralAndGPD(TruncatedDistribution(Laplace(mu = 0.103895, lambda = 17.3974), bounds = [0, (1.91725) +inf[), upper)	-2.628789	-2.628789	3.807281e-09	False



Renault Group
PHIMECA The responsible engineering

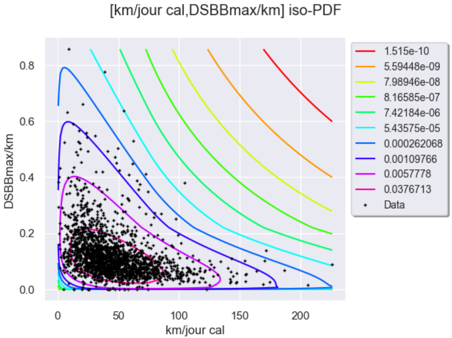
StaRe : Mileage-Usage dependency fitting

- Copula inference using all available copulae factories



	Copula Type	BIC	Analytical Spearman coef	Empirical Spearman coef	Delta Spearman	Copula Class
0	Frank	-2.248448e-01	-0.452627	-0.449399	0.003229	FrankCopula(theta = -3.03263)
1	Normal	-2.012900e-01	-0.451140	-0.449399	0.001741	NormalCopula(R = [[1 -0.468051]\n [-0.468051 1]])
2	Independent	0.000000e+00	0.000000	-0.449399	0.449399	IndependentCopula(dimension = 2)
3	Clayton	1.797693e+308	-0.438143	-0.449399	0.011256	ClaytonCopula(theta = -0.473384)

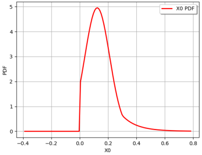
- JointDistribution built using marginal and copula inference results

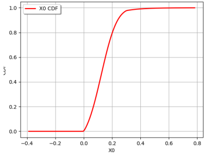


StaRe : Customer Usage modelling

Details

- Marginal inference handles truncated fit support
  Fit is performed using MaximumLikelihoodFactory, taking into account truncation parameters
- JointCentralAndGPD inherits from openturns.PythonDistribution
  computeCDF(), computePDF(), computeQuantile() and getRealization() have been re-implemented to ensure PDF continuity and to improve performance





$$CDF(X) = \begin{cases} CDF_{central}(X) & \text{, if } X \leq X_t \\ 1 - cCDF_{GPD}(X - X_t) \cdot cCDF_{central}(X_t) & \text{, if } X \geq X_t \end{cases}$$

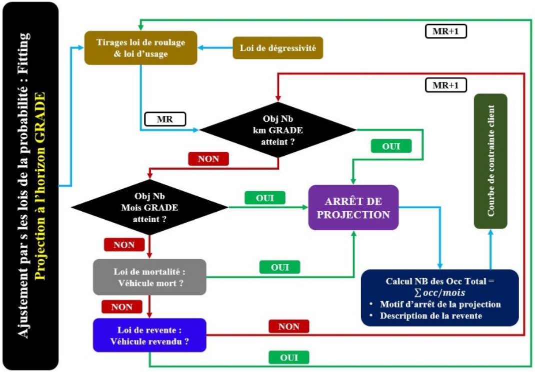
$$PDF(X) = \begin{cases} PDF_{central}(X) & \text{, if } X \leq X_t \\ PDF_{GPD}(X - X_t) \cdot \frac{PDF_{central}(X_t)}{PDF_{GPD}(0)} & \text{, if } X \geq X_t \end{cases}$$

Stress prediction

Stress prediction : 2 algorithms

Customer Usage Modeling → **Stress Prediction** → Strength Specification → Validation Plans Specification

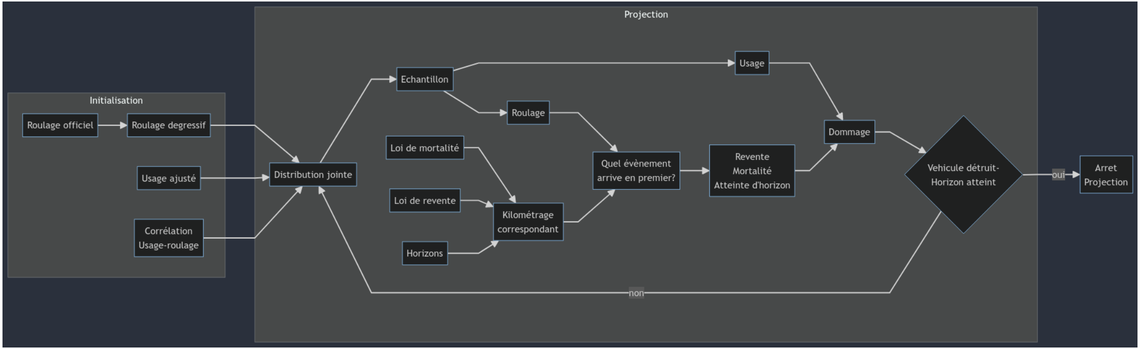
- Historical algorithm
- Simulated car by simulated car
- Month by month



Stress prediction : 2 algorithms

Customer Usage Modeling → **Stress Prediction** → Strength Specification → Validation Plans Specification

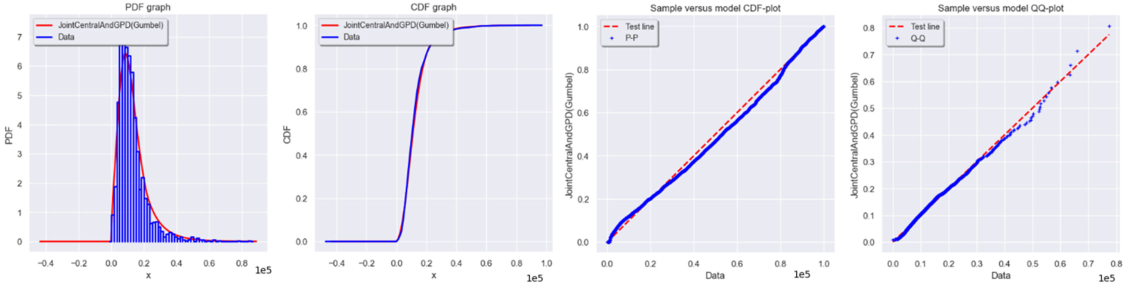
- New algorithm, vector of simulated cars, car life event by car life event
  → Allows 200 times faster prediction



Stress fitting

Customer Usage Modeling → **Stress Prediction** → Strength Specification → Validation Plans Specification

- Stress prediction distribution is fitted using the same methods applied previously to raw data

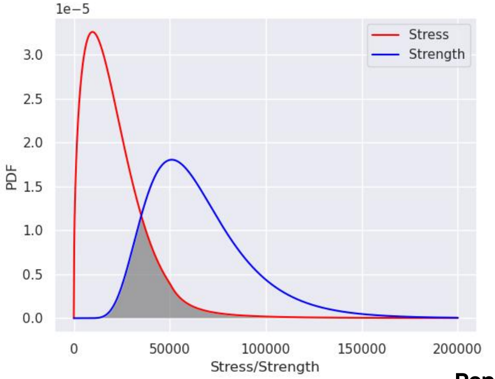


Strength specification

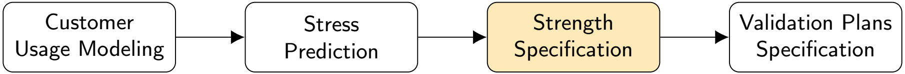
Strength specification

Customer Usage Modeling → Stress Prediction → **Strength Specification** → Validation Plans Specification

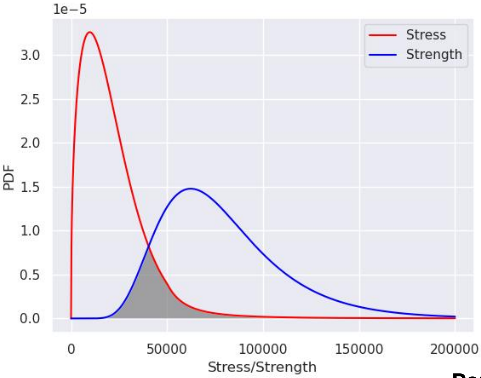
- With hypothesis on shape parameter, strength scale parameter is optimized to match
  proba (Stress > Strength) max
- Strength is modelised by LogNormal or WeibullMin distribution
- Stress 99^{th} percentile $\sim 8.4 \cdot 10^4$ in this case



Strength specification



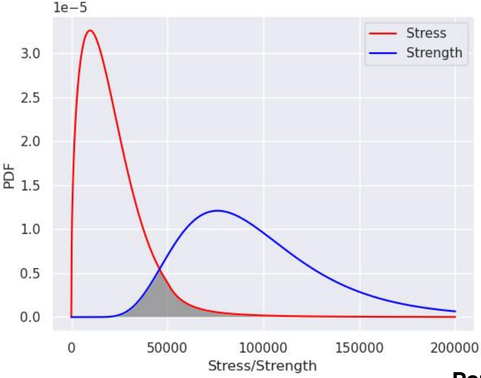
- With hypothesis on shape parameter, strength scale parameter is optimized to match
  proba (Stress > Strength) max
- Strength is modelised by LogNormal or WeibullMin distribution
- Stress 99^{th} percentile $\sim 8.4 \cdot 10^4$ in this case



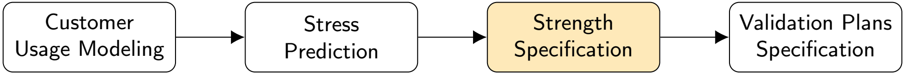
Strength specification

Customer Usage Modeling → Stress Prediction → Strength Specification → Validation Plans Specification

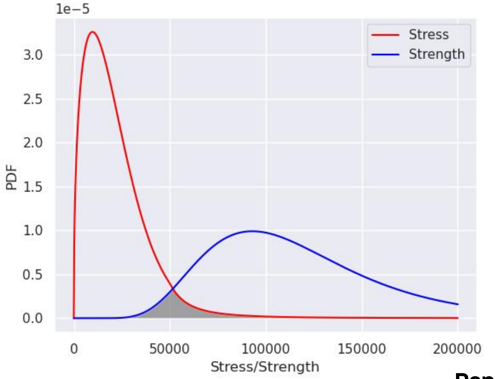
- With hypothesis on shape parameter, strength scale parameter is optimized to match proba (Stress > Strength) max
- Strength is modelised by LogNormal or WeibullMin distribution
- Stress 99^{th} percentile $\sim 8.4 \cdot 10^4$ in this case



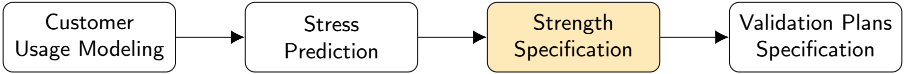
Strength specification



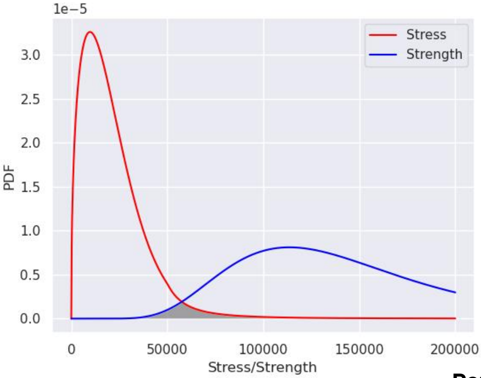
- With hypothesis on shape parameter, strength scale parameter is optimized to match
  proba (Stress > Strength) max
- Strength is modelised by LogNormal or WeibullMin distribution
- Stress 99^{th} percentile $\sim 8.4 \cdot 10^4$ in this case



Strength specification



- With hypothesis on shape parameter, strength scale parameter is optimized to match
  proba (Stress > Strength) max
- Strength is modelised by LogNormal or WeibullMin distribution
- Stress 99^{th} percentile $\sim 8.4 \cdot 10^4$ in this case

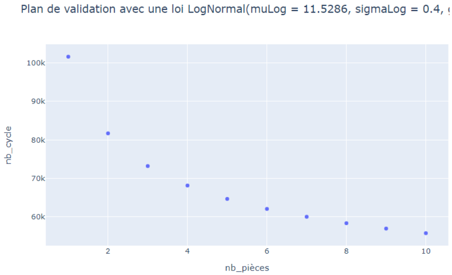


Validation plan specification

Validation plan specification

Customer Usage Modeling → Stress Prediction → Strength Specification → **Validation Plans Specification**

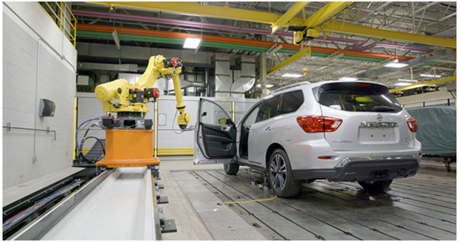
- Validation plan proposal :
  Number of parts @ number of cycles without failure
- Strength is tested by statistical hypothesis with confidence level:
  Hypothesis on proportion of failures is tested (unilateral) on small sample without failure
  → binomial distribution with number of success (ie failure) equal to 0



Validation plan specification

Customer Usage Modeling → Stress Prediction → Strength Specification → **Validation Plans Specification**

- Reciprocal procedure available :
  From a number of parts, each with number of cycles with (and/or without) failure
- Strength distribution scale parameter is determined leading to failure probability estimation



Conclusion & outlook

Conclusion & outlook

Deployment: sharing good practices and methodology

Reliability	Python User	Tool	Status	Trained	Interested
Specialist	Yes	Stare module Notebook	Delivered	7	5
Specialist	No	No code GUI	Decided	0	10
Intermediate	No	Simple GUI	To be decided & framed		

Conclusion

- StaRe module allows better distribution fitting leading to improved validation plans
- Dedicated notebook interface allows versatile reliability tools usage for (non-)coding experts
- Opening discussion for including in OT
  - fit automatisation, taking into account support
  - stare.JointCentralAndGPD implementation

Thank you!