



Weibull 분포의 모수 추정을 위한 Python 라이브러리 설명

▼ 아래 링크에서도 확인 가능합니다.

 [Weibull 분포의 모수 추정을 위한 Python 라이브러리 설명](#)

사용한 라이브러리

- pandas → 데이터 전처리용
- surpyval → 분석 라이브러리
- reliability → 분석 라이브러리

Pandas 라이브러리

- Pandas에서는 기본적으로 정의되는 자료구조인 Series와 **Data Frame**을 사용합니다.
- 데이터 처리와 분석을 위한 라이브러리입니다. 테이블 형태의 데이터를 다루기 위한 데이터프레임 (DataFrame) 자료형을 제공합니다.





SurPyval 설명

<https://surpyval.readthedocs.io/en/latest/index.html>

<https://surpyval.readthedocs.io/en/latest/>

- 파이썬 수명 분석 라이브러리
- 모델을 다른 프로젝트에 재사용 가능
- 모수 추정이 가능한 라이브러리

SurPyval Modeling Methods

 Method	 Para/Non-Para	<input checked="" type="checkbox"/> Observed	 Censored	 Truncated
--	---	--	--	---

Aa Method	▼ Para/Non-Para	☑ Observed	☰ Censored	☰ Truncated
<u>Maximum Likelihood (MLE)</u>	Parametric	☑	Yes	Yes
<u>Probability Plotting (MPP)</u>	Parametric	☑	Yes	Limited
<u>Mean Square Error (MSE)</u>	Parametric	☑	Yes	Limited
<u>Method of Moments (MOM)</u>	Parametric	☑	No	No
<u>Maximum Product Spacing (MPS)</u>	Parametric	☑	Yes	No (planned)
<u>Kaplan-Meier</u>	Non-Parametric	☑	Right only	Left only
<u>Nelson-Aalen</u>	Non-Parametric	☑	Right only	Left only
<u>Fleming-Harrington</u>	Non-Parametric	☑	Right only	Left only
<u>Turnbull</u>	Non-Parametric	☑	Yes	Yes

- MLE가 디폴트 method이나 다른 modeling 기법도 사용 가능함
- 임의적인 관측치, 관측 중단 및 잘림 조합을 사용해 모든 분포에 대한 모수 추정이 가능하게 하는 라이브러리
- Weibull 분포 뿐만 아닌 Loglogistic 등 분포 사용 가능함

SurPyval 라이브러리 사용시 주의해야 할 점

- 파이썬 surpyval 라이브러리의 경우 관측 중단 값이 1, 실패(고장)값이 0으로 디폴트로 세팅되어 있음
→ 따라서 데이터 변환 및 생성 시에 주의 해야 함
- 미니탭의 경우 관측 중단 값 임의로 설정 가능
- 파이썬 Reliability 라이브러리의 경우 관측중단값, 고장값 모두 지정 가능함
 - 그러나, 데이터프레임 → 리스트로의 변환이 필요함



사용 방법

- jupyter notebook, colab, vscode 등에서 실행 가능

```
# 라이브러리 설치
pip install surpyval
```

```
# surpyval 라이브러리의 Weibull 분포 사용해 모수 추정

import surpyval as surv # weibull 분포 분석용 라이브러리

result = surv.Weibull.fit([데이터내 수명 정보(가동 시간)], [데이터내 중단정보(0/1)], how = '어떤 모델링기법? ex)MLE')
# ex) 수명정보 -> [722, 0, 30 ...]

# plot 출력
result.plot()

# MTTF 값
result.mean()

# 자세한 정보는 python 코드 참조
```

Reliability 라이브러리 설명

<https://reliability.readthedocs.io/en/latest/>

- 신뢰성 엔지니어링 및 생존 분석을 위한 파이썬 라이브러리
- scipy.stats의 기능 확장 버전
- plot이 Surpyval보다 직관적임

Supported Distributions

<u>Aa</u> Parametric Models	<u>≡</u> Fitting a specific distribution to data
<u>Weibull Distribution</u>	Fit_Weibull_2P
<u>Exponential Distribution</u>	Fit_Exponential_1P
<u>Normal Distribution</u>	Fit_Normal_2P
<u>Lognormal Distribution</u>	Fit_Lognormal_2P
<u>Gamma Distribution</u>	Fit_Gamma_2P
<u>Beta Distribution</u>	Fit_Beta_2P
<u>Loglogistic Distribution</u>	Fit_Loglogistic_2P
<u>Gumbel Distribution</u>	Fit_Gumbel_2P
<u>Location shifting the distributions</u>	

Reliability 라이브러리 사용시 주의해야 할 점

- 고장 값과 우측중단 값 모두 지정해 계산이 가능하나, 데이터의 fail, censored 각각의 수명정보를 따로 분리해 리스트로 변환이 필요함.
- 엑셀 파일의 경우 라이브러리 활용해 변환가능
→ https://reliability.readthedocs.io/en/latest/Converting_data_between_different_formats.html



사용 방법

라이브러리 설치

```
pip install reliability
```

파라미터 계산 및 plot 출력

```
from reliability.Fitters import Fit_Weibull_2P
from reliability.Probability_plotting import Weibull_probability_plot
from reliability.Distributions import Weibull_Distribution
import matplotlib.pyplot as plt
```

failure, right_censored 가동시간 정보 리스트로의 변환 필요

```
result = Fit_Weibull_2P(failure = [고장 가동시간], right_censored = [관측중단 가동시간])
```

plot 출력

```
Weibull_probability_plot(failures = list_fail, right_censored = list_censored)
plt.show()
```

MTTF

```
MTTF = Weibull_Distribution(alpha = result.alpha, beta = result.beta).mean
```

Minitab vs Python 출력 값

	Python-SurPyval	Python-Reliability	Minitab
shape parameter	0.47707	0.47707	0.47707
scale parameter	788024.148981	788021	788024
MTTF	1725283.34555	1725275.94047	1725283

- 미니탭과 SurPyval 간 결과값 가장 유사함

각 프로그램의 분석 결과값 비교

- 아래에 있는 결과값의 데이터들은 무작위로 생성한 것입니다.

Python - SurPyval

<u>Aa</u> Data	# Shape Parameter	# Scale Parameter	# MTTF	# Probability of Failure	# Reliability	# Number of Suspension	≡ Number of Fail
<u>Sample Data(1).</u>	0.48137	1197531.18433	2575438.40219	0.02794	0.97206	1129	44
<u>Sample Data(2).</u>	0.47368	1396342.62614	3101719.32397	0.02751	0.97249	1130	43
<u>Sample Data(3).</u>	0.47877	1302376.87262	2831225.08271	0.02737	0.97263	1147	43

Python - Reliability

<u>Aa</u> Period	# Shape Parameter	# Scale Parameter	# MTTF	# Probability of Failure	# Reliability	# Number of Suspension	≡ Number of Fail
<u>Sample Data(1).</u>	0.48139	1197182.60554	2574489.25144	0.02794	0.97206	1129	44
<u>Sample Data(2).</u>	0.47369	1396107.83235	3101057.05628	0.02751	0.97249	1130	43
<u>Sample Data(3).</u>	0.47877	1302438.1842	2831393.55533	0.02737	0.97263	1147	43

Minitab

<u>Aa</u> Period	# Shape Parameter	# Scale Parameter	# MTTF	# Probability of Failure	# Reliability	# Number of Suspension	≡ Number of Fail
<u>Sample Data(1).</u>	0.48137	1197531	2575438	0.02794	0.97206	1129	44
<u>Sample Data(2).</u>	0.47368	1396343	3101719	0.02751	0.97249	1130	43
<u>Sample Data(3).</u>	0.47877	1302377	2831225	0.02737	0.97263	1147	43

Weibull++

--	--	--	--	--	--	--	--

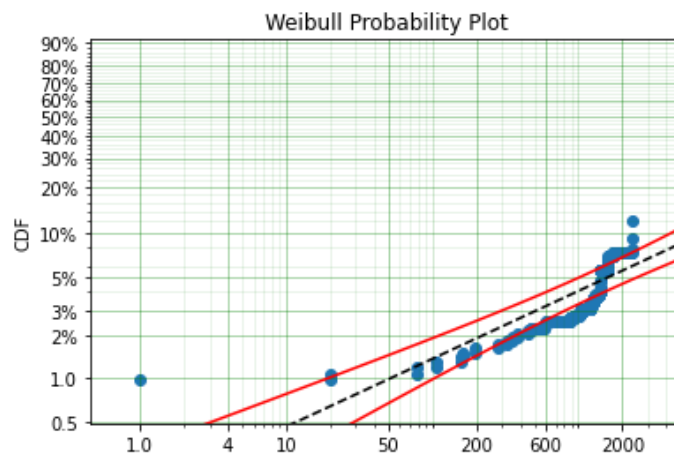
Aa Period	# Shape Parameter	# Scale Parameter	# MTTF	# Probability of Failure	# Reliability	# Number of Suspension	# Number of Fail
Sample Data(1).	0.48	1196754	2573277	0.0279	0.9721	1129	44
Sample Data(2).	0.47	1398258	3107226	0.0275	0.9725	1130	43
Sample Data(3).	0.48	1301719	2829379	0.0274	0.9726	1147	43

PLOT 비교

미니탭 출력 PLOT

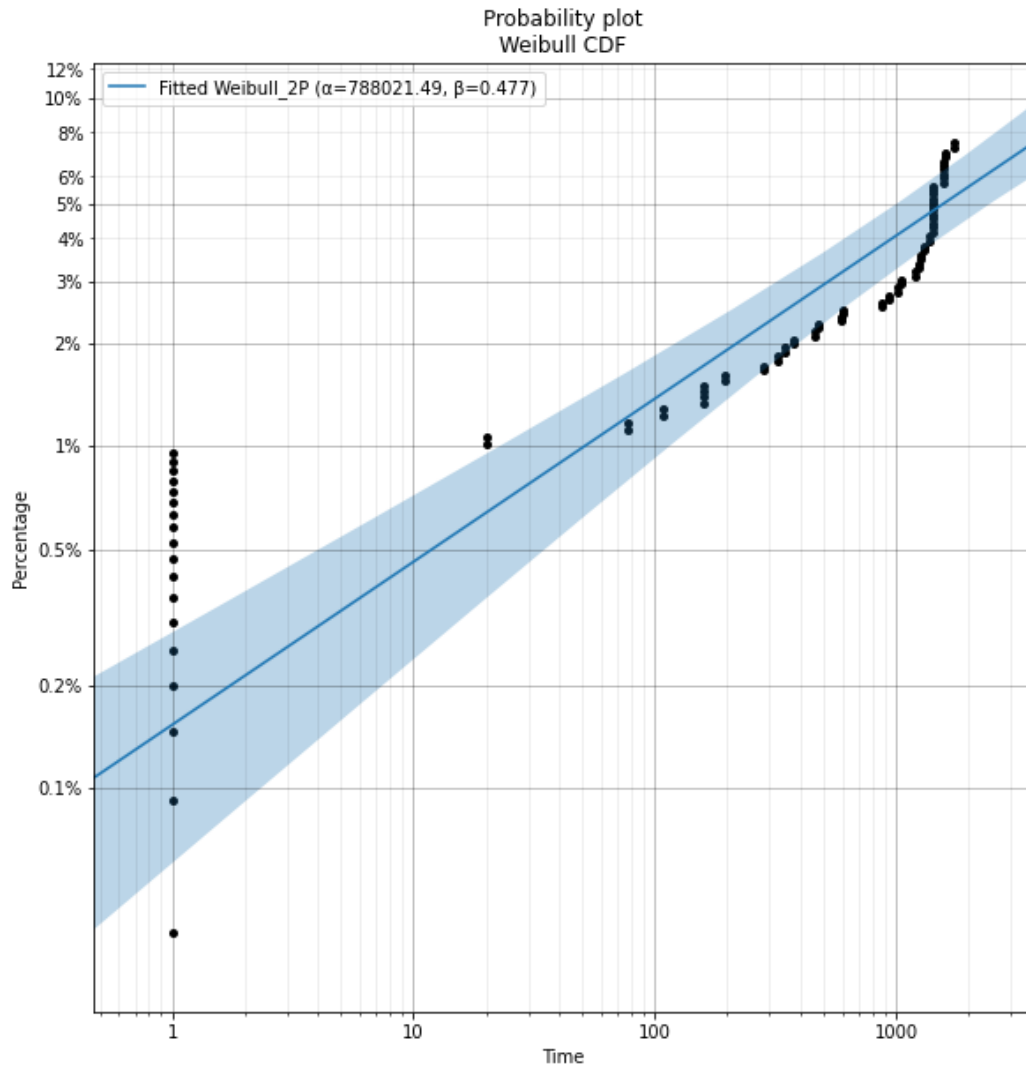
- 추가예정

파이썬 - Surpyval 출력 PLOT



- 플롯 저장은 가능하나, 수정은 불가능함

파이썬 - Reliability 출력 PLOT



- plot 수정 및 저장 가능
- 직관적인 그래프 도출 가능

일반적인 Weibull 분포가 아닌 Weibayes 기법 적용

- Weibayes small sample, zero and sudden death failure test and data analysis techniques in order to mathematically and graphically quantify the risk associated with assuming an incorrect value of the Weibull shape parameter, beta (beta).
- 다소 부정확할 수 있다는 점 감안하고 사용해야 함.

사용한 라이브러리

- weibull 라이브러리
 - 데이터프레임 → 리스트 자료형 변환 필요
 - <https://weibull.readthedocs.io/en/v0.0.11/>



사용 방법

```
# 라이브러리 설치
```

```
pip install weibull
```

```
# 파라미터 계산 및 플롯 출력
```

```
import weibull
```

```
result = weibull.Weibayes(data = 가동시간정보, confidence_level= 0.95(지정가능), beta = 지정가능)
```

```
result.plot()
```

출력된 PLOT

