Method: 2P Weibull distribution and Weighted Least Squares Regression

Contents

Statistical evaluation of failure stresses.

The procedure is as follows:

- Sort the stress at failure for each specimen in increasing order of magnitude: σ_i, where i is the rank number.
- Calculate the probability of failure $P_{f,i}$ for each measured value. Use the following estimator: $P_{f,i} = \frac{i-0.5}{n}$ (*Hazen's probability estimator*) or $P_{f,i} = \frac{i}{n+1}$ (*Mean rank probability estimator*) where n is the number of tested samples.
- Put the Weibull distribution into linearized form. Enter the measured data into the Weibull mesh.
- Fit the data by Weighted Least Squares Regression . Find the slope, the intercept, and the coefficient of determination R^2, coefficient of variation COV and the Anderson Darling goodness of fit metric p_{AD}.
- Calculate the confidence interval CI.
- Determine the Weibull parameters β (shape parameter) and θ (scale parameter).
- Determine the desired fractile value of the bending tensile strength f_y using the regression line and using the confidence interval. In the case of the confidence interval, the target goal seek is used.
- Plot the cumulative distribution function $F(x)$ and the density function $f(x)$.

> **Note:** You can use the 🚀 **`Live Code`** button in the top right to activate the interactive features and use Python interactively!

Once the "Live Code" is enabled it is advised to **"Run All"** cells first to load all the necessary packages and functions.

Afterwards, any changes can be made in the input form and when the **"Evaluate"** button is clicked the changes are recorded.

Finally, the last two cells can be run individually by clicking on the **"Run"** button to produce the Weibull plots.

▼ Hide code cell source

```
import numpy as np
import pandas as pd
from scipy.stats import linregress, t, weibull_min
import matplotlib.pyplot as plt
from matplotlib.ticker import ScalarFormatter
from matplotlib.ticker import FuncFormatter
from scipy.optimize import fsolve
import ipywidgets as widgets
from IPython.display import display, HTML, clear_output
from scipy.special import gamma
%config InlineBackend.figure_format = 'svg'
```

run | run all | add cell | clear

Definition of useful functions:

- Convert failure stress to equivalent failure stress for a constant load given a reference time period.
- Calculation of standard error.
- Confidence interval calculation
- Weibull pdf and cdf distributions
- Calculationf of Anderson Darling goodness of fit metric p_{AD}

▶ Show code cell source

Print to PDF

Interactive Data Input

This widget allows users to input datasets, set analysis parameters, and optionally convert failure stress values.

What can be done:

Choose the number of datasets (default: 3).
Enter dataset names and values (comma-separated).
Set target values:

- stress fractile (default: 5%)
- confidence interval: Enter the alpha value. (default: 95%)
- x-limits (default: lower limit = min_stress_level - 20, upper limit =max_stress_level + 20)

Select a probability estimator:

- *(i - 0.5) / n* (*Hazen's probability estimator*)
- *i / (n + 1)* (*Mean rank probability estimator*)

Convert failure stress (optional):

- Toggle conversion of failure stress to equivalent failure stress for a selected reference period ON/OFF, by clicking Yes/No respectively.
- Enter time-to-failure values.
- Choose a reference time period (1s, 3s, 5s, 60s).

How It Works:

1️⃣ Click **"Confirm"** to generate input fields for the failure stress datasets.
2️⃣ *(Optional)* Click **"Yes"** and enter the time-to-failure values corresponding to each failure stress dataset.
3️⃣ Enter values and click **"Evaluate"** to process the data.

> **Note:** The script checks for errors in the input or mismatch in the dimensions between the time-to-failure datasets and the failure stress datasets.

▶ Show code cell source

Enter the data separated by commas, with decimal point (e.g. "1.44, 2.33, 4.22, 3.01,...")

Data protection declaration: The data entered will not be saved or transmitted over the network.

Number of Datasets: 3

Confirm

Name 1: At scratch Values 1: 59.94805695, 36.96298947, 33.85559 34.92163864, 33.31065808, 38.22288

Name 2: Not at scratch Values 2: 92.37645599, 75.78975356, 75.16784 61.57474021, 112.1285123, 59.01209

Name 3: SC-air-HT500 Values 3: 42.89434463, 88.71811493, 50.50840 141.0013163, 117.6306604, 41.14746

Target stress fractile: 0,05

Target confidence interval: 0,1

Lower x limit: 0

Upper x limit: 0

Probability Estimator:

(i-0.5)/n i/(n+1)

Convert to equivalent constant failure stress for a reference time period?

No Yes

Select reference time period[s]: 5

Time to failure 1: 4.322750691, 1.490313627, 1.355389249, 1.566979218, 1.390874312, 3.15951534, 1.998810295, 8.315062848,

Time to failure 2: 3.784500028, 2.904653487, 2.786599839, 2.849313211, 2.400164791, 4.561799845, 2.30234895, 3.575816102,

Time to failure 3: 5.733701498, 4.358551369, 1.612755329, 2.368135054, 2.909189821

Evaluate

```
Selected Probility Estimator: (i-0.5)/n
At scratch; n = 15 samples: [59.94805695 36.96298947 33.85559988 37.41138
 38.22288871 34.1066492  35.29675867 36.84531616 33.28913956 37.73474803
 42.89434463 50.50840873 41.14746787]
Not at scratch; n = 14 samples: [ 92.37645599  75.78975356  75.16784915
 112.1285123   59.01209856  96.03848041  88.71811493  62.87593174
 141.0013163  117.6306604   47.1         61.9        ]
SC-air-HT500; n = 29 samples: [ 92.37645599  75.78975356  59.94805695  36
```

```
 75.16784915  37.41138875  34.92163864  74.98678347  33.31065808
 38.22288871  34.1066492   61.57474021  35.29675867  36.84531616
112.1285123   33.28913956  37.73474803  59.01209856  96.03848041
 42.89434463  88.71811493  50.50840873  62.87593174 141.0013163
117.6306604   41.14746787  58.94676758  76.46729593]

Target stress fractile: 5.0%
Target confidence interval: 95%
Default values for Lower x limit and Upper x limit.

Conversion of failure stress: Yes, Reference time period [s]: 5
At scratch; n = 15; Time to failure values[s]: [4.322750691, 1.490313627,
The equivalent failure stress for 5 seconds is [49.76 28.71 26.14 29.15 2
 34.1  40.47 32.12]
Not at scratch; n = 14; Time to failure values[s]: [3.784500028, 2.904653
The equivalent failure stress for 5 seconds is [ 76.05  61.37  60.71  60.
 119.13  97.7   37.66  50.13]
SC-air-HT500; n = 29; Time to failure values[s]: [3.784500028, 2.90465348
The equivalent failure stress for 5 seconds is [ 76.05  61.37  49.76  28.
  30.24  29.49  49.27  27.41  29.16  93.39  25.52  29.25  47.1   78.78
  34.1   72.51  40.47  50.89 119.13  97.7   32.12  47.13  61.93]
```

Start of Statistical evaluation

▼ Hide code cell source

```
if 'stress_values' not in globals() or 'stress_values' not in locals
    print("Please enter all the necessary input data in the above in
else:
    plt.figure(figsize=(10, 6))
    i = 0;
    min_stress_data = 100000
    max_stress_data = -10000

    # Initialize an empty dictionary to store the scale & shape valu
    Weibull_distribution_parameters = {}
    Regression_line_parameters = {}
    for dataset, stress_data in stress_values.items():

        if stress_data.size > 0 and np.any(stress_data != 0):
```

run | run all | add cell | clear

At scratch (n=15)

Fractile [%]	Stress [MPa]	95% CI lower [MPa]	95% CI upper [MPa]
0.8%	9.93	3.17	14.55
5%	15.62	7.84	19.76
50%	29.5	26.32	32.3
Selected 5.0%	15.62	7.84	19.76

At scratch (n=15)

Min Stress [MPa]	Max Stress [MPa]	Mean Stress [MPa]	Coeff. of variation [%]	Goodness of fit, p_{AD}
25.52	49.76	31.04	27.47	0.01

Regression line for "**At scratch**" (**n=15**) is: **y = 4.09x - 14.22; R^2= 0.515**

Not at scratch (n=14)

Fractile [%]	Stress [MPa]	95% CI lower [MPa]	95% CI upper [MPa]
0.8%	14.79	10.18	19.03
5%	27.53	22.01	32.09
50%	65.87	61.87	70.21
Selected 5.0%	27.53	22.01	32.09

Not at scratch (n=14)

Min Stress [MPa]	Max Stress [MPa]	Mean Stress [MPa]	Coeff. of variation [%]	Goodness of fit, p_{AD}
37.66	119.13	68.24	36.51	34.42

Regression line for "**Not at scratch**" (**n=14**) is: **y = 2.98x - 12.87; R^2= 0.864**

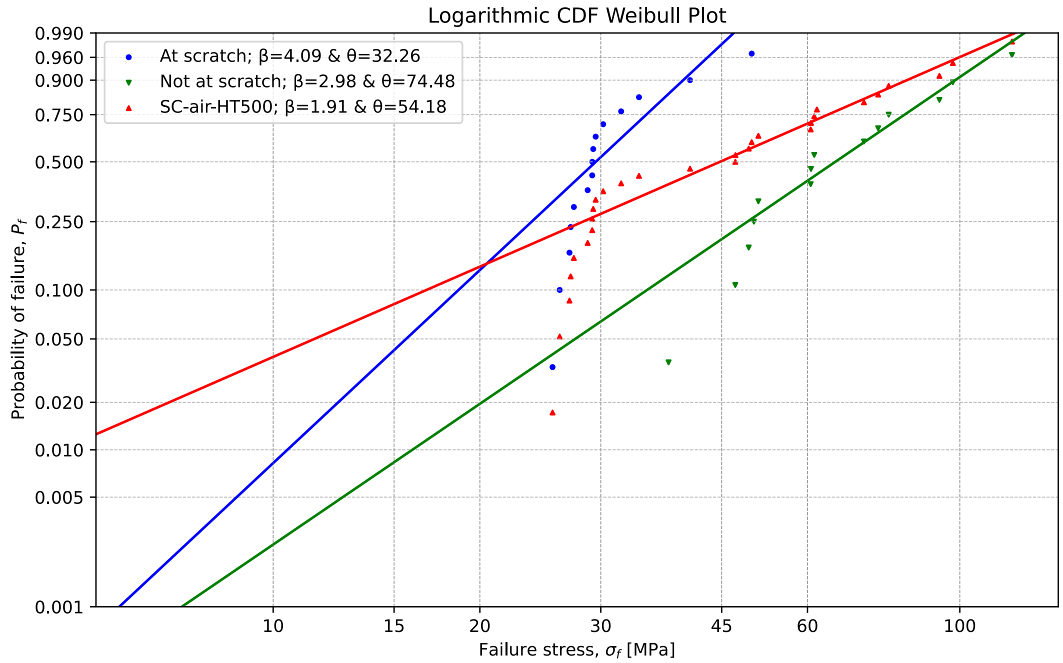
SC-air-HT500 (n=29)

Fractile [%]	Stress [MPa]	95% CI lower [MPa]	95% CI upper [MPa]
0.8%	4.36	2.31	6.6
5%	11.48	7.85	14.78
50%	44.74	40.55	49.37
Selected 5.0%	11.48	7.85	14.78

SC-air-HT500 (n=29)

Min Stress [MPa]	Max Stress [MPa]	Mean Stress [MPa]	Coeff. of variation [%]	Goodness of fit, p_{AD}
25.52	119.13	49.73	54.38	0.59

Regression line for "**SC-air-HT500**" (**n=29**) is: **y = 1.91x - 7.64; R^2= 0.733**

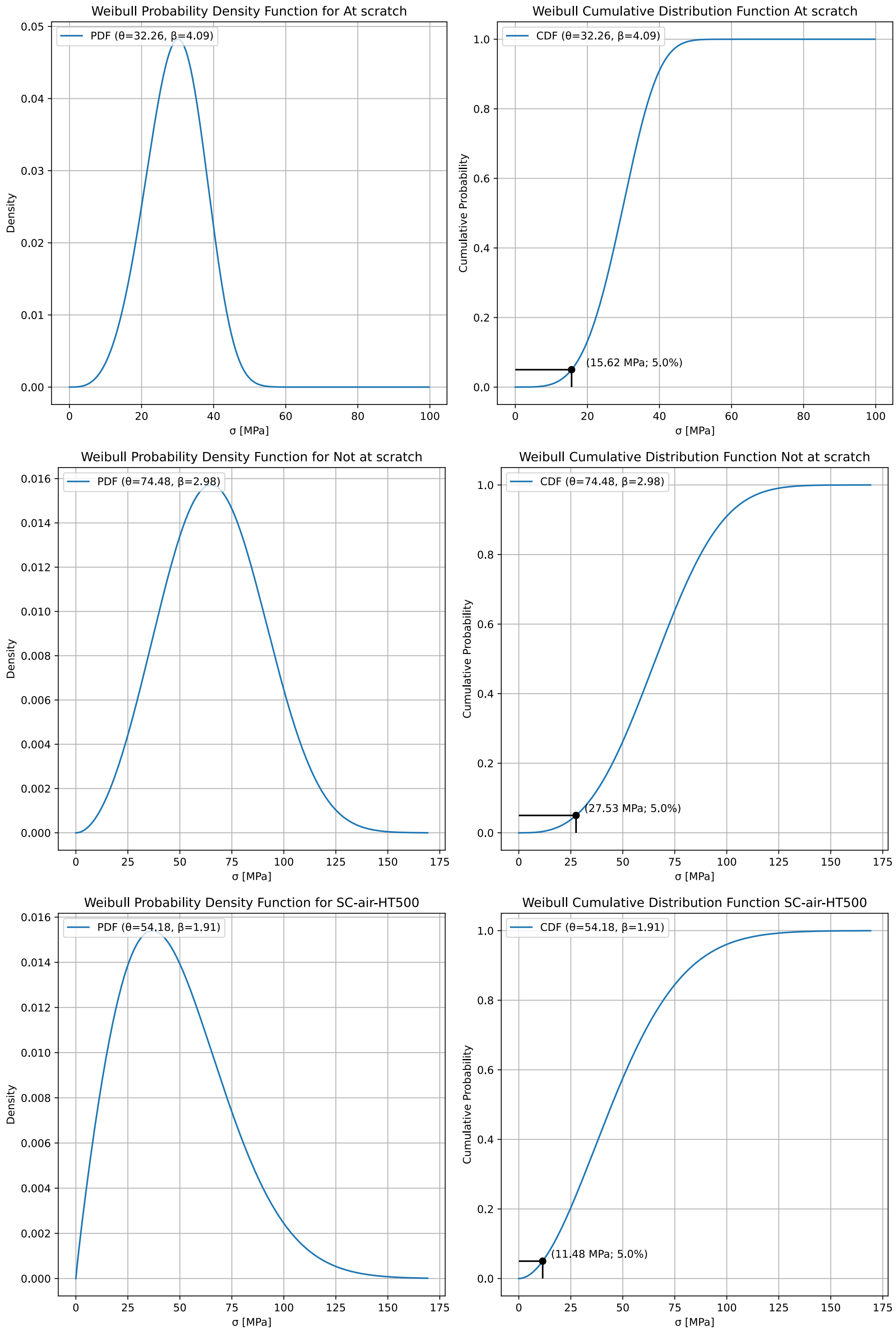


Plot the cumulative distribution function $F(x)$ and the density function $f(x)$.

▼ Hide code cell source

```
if 'stress_values' not in globals() or 'stress_values' not in locals(
    print("Please enter all the necessary input data in the above inp
else:
    for dataset, stress_data in stress_values.items():
            if stress_data.size > 0 and np.any(stress_data != 0):
                if convert_to_equivalent_stress == "Yes" and len(time
                    stress_data = equivalent_stress_values[dataset]
                # Creating a range of x values
                x = np.linspace(0,max(stress_data)+50,1000)
                # Calculate PDF and CDF
                pdf_values = weibull_pdf(x, Weibull_distribution_para
                cdf_values = weibull_cdf(x, Weibull_distribution_para
                # Calculate the x% fractile
                target_percentile = 1 - target_P_f
```

run | run all | add cell | clear


Weibull Probability Density Function for At scratch
PDF (θ=32.26, β=4.09)
Density
σ [MPa]
Weibull Cumulative Distribution Function At scratch
CDF (θ=32.26, β=4.09)
Cumulative Probability
(15.62 MPa; 5.0%)
σ [MPa]
Weibull Probability Density Function for Not at scratch
PDF (θ=74.48, β=2.98)
Density
σ [MPa]
Weibull Cumulative Distribution Function Not at scratch
CDF (θ=74.48, β=2.98)
Cumulative Probability
(27.53 MPa; 5.0%)
σ [MPa]
Weibull Probability Density Function for SC-air-HT500
PDF (θ=54.18, β=1.91)
Density
σ [MPa]
Weibull Cumulative Distribution Function SC-air-HT500
CDF (θ=54.18, β=1.91)
Cumulative Probability
(11.48 MPa; 5.0%)
σ [MPa]