swe_ws1920_1_1_wind_speed_computation

November 3, 2020

1 Tutorial 1.1. Wind Speed Computation

1.0.1 Description: The wind velocity is measured at a reference height and values are reported (generally) as the maxima of given duration. In this tutorial the yearly maxima are analyzed to fit a distribution from the given data. The design wind velocity is computed for the chosen return period. Some exercises are proposed. This is based upon 2.3 from J. D. Holmes., Wind loading of structures“, 2nd ed.

Students are advised to complete the exercises.

Project: Structural Wind Engineering WS 20-21 Chair of Structural Analysis @ TUM - R. Wüchner, M. Péntek, A. Kodakkal Author: anoop.kodakkal@tum.de, mate.pentek@tum.de
Created on: 30.11.2015
Last update: 28.10.2020

Contents:

1. Extreme wind estimation based on given data
2. Gumbel’s method
3. Gringorten’s method
4. The method of moments

[1]:
```
# import
import matplotlib.pyplot as plt
import numpy as np
from ipywidgets import interactive
```

Get the data files from ‘east_sale.dat’, data from J. D. Holmes., Wind loading of structures“, 2nd ed.

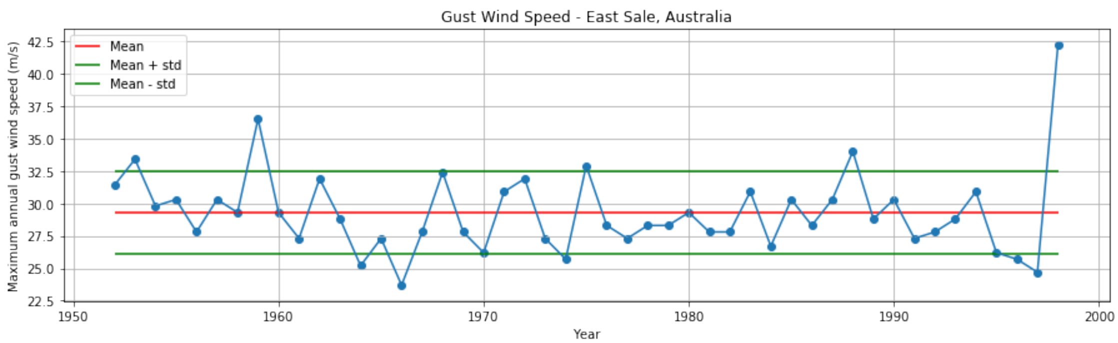
[2]:
```
winddata = np.loadtxt('east_sale.dat')
year = winddata[:,0]
maxgust = winddata[:,1]
```

Let us evaluate the mean and standard deviation of the wind data.

[3]:
```
gust_m = np.mean(maxgust)
gust_std = np.std(maxgust)
```

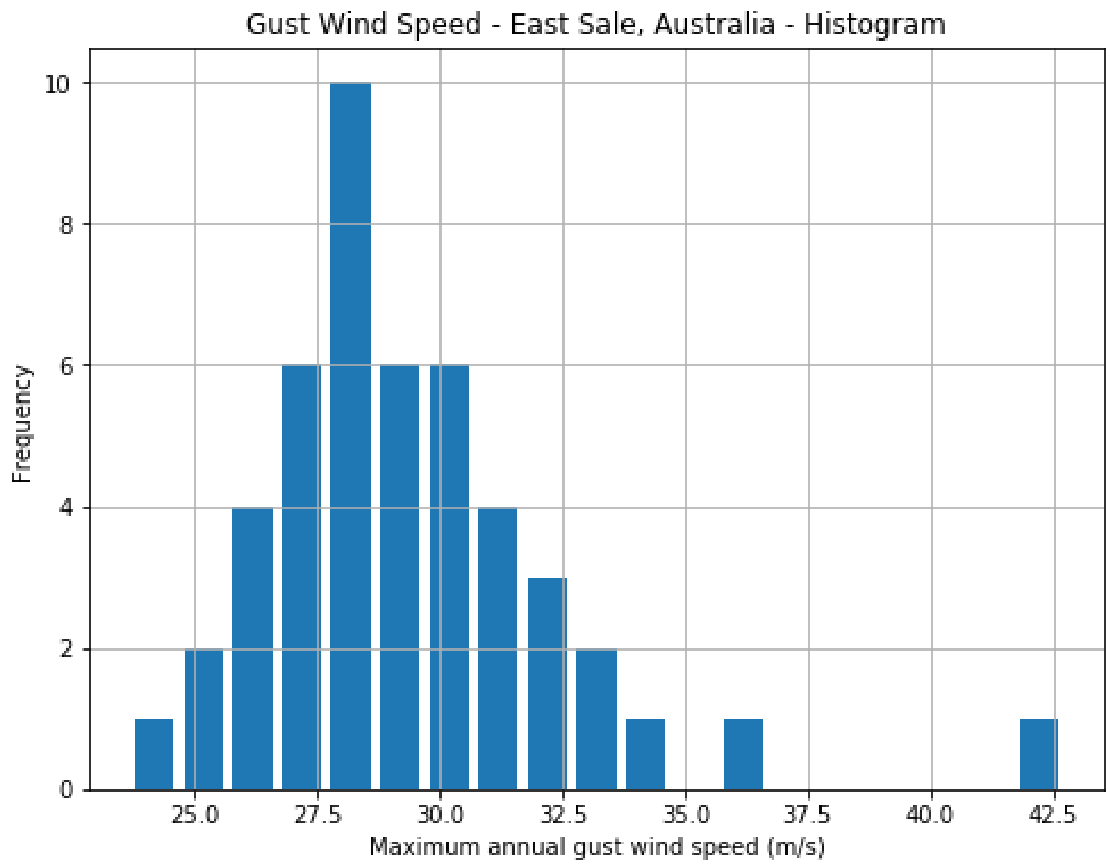
Let us look at the plot

[4]:
```
plt.figure(num=1, figsize=(15, 4))
plt.plot(year, maxgust)
plt.scatter(year, maxgust)
plt.hlines(gust_m, min(year), max(year), color='r', label = 'Mean')
plt.hlines(gust_m + gust_std, min(year), max(year), color='g', label = 'Mean +␣
↪std')
plt.hlines(gust_m - gust_std, min(year), max(year), color='g', label = 'Mean -␣
↪std')
plt.ylabel('Maximum annual gust wind speed (m/s)')
plt.xlabel('Year')
plt.title('Gust Wind Speed - East Sale, Australia')
plt.legend()
plt.grid(True)
```



To better understand the data, a histogram may be plotted. The histogram represents the frequency of data in each class interval. The inbuilt function for plotting histogram in python is used.

[5]:
```
plt.figure(num=2, figsize=(8, 6))
plt.hist(maxgust, bins=np.arange(min(maxgust), max(maxgust)+1), rwidth=0.8)
plt.ylabel('Frequency')
plt.xlabel('Maximum annual gust wind speed (m/s)')
plt.title('Gust Wind Speed - East Sale, Australia - Histogram')
plt.grid(True)
```



1.0.2 Extreme wind estimation based upon measured data

3 methods are used in this tutorial for extreme wind estimation

1. Gumbel's method
2. Gringorten's method
3. The method of moments

1. Gumbel's method A cumulative probability distribution function for extreme values of Type I (Gumbel) is considered

$$F_u(U) = e^{-e^{-\frac{U-u}{a}}}$$

The following steps are followed to obtain the extreme wind statistics in Gumbel's method.
step 1 The largest wind speed in each calendar year of the record is extracted U - 'maxgust'.
step 2 The series is ordered (ranked) $ 1,2,...m,...N $.
step 3 Each value is assigned a probability of non-exceedence $pm/(N+1)$.
step 4 A reduced variate is formed $ y = ln (ln(p))$.
step 5 A straight line is fit for the data reduced variate $ y $ and gust wind speed $ U $.
step 6 The slope a and mode u of the best fit line is obtained.

step 7 For a chosen return period an estimate for the maximum gust wind speed can be obtained as

$$U_R = u + a\{-ln[-ln(1-\frac{1}{R})]\}$$

What does the return period indicate?

step 1 The largest wind speed in each calendar year of the record is extracted (U) - 'maxgust' .

step 2 The series is ordered based on the absolute value of wind speed and ranked as $ rank = 1,2,...m,...N $, with N being the maximum rank

[6]:
```
gust_sorted = np.sort(maxgust)
max_rank = len(gust_sorted) # the highest rank
rank = np.arange(1, max_rank + 1) # the rank of ordered wind speed values.
```

step 3 Each wind speed value is assigned a **probability of non-exceedance** $pm/(N+1)$. The probability of non-exceedance is the expected chance that the wind speed value will not exceed the speed considered.

step 4 A reduced variate is formed $ y = ln (ln(p))$.

[7]:
```
gumbel_prob_nonexc = rank / (max_rank + 1)
gumbel_red_var = -np.log(-np.log(gumbel_prob_nonexc))
```

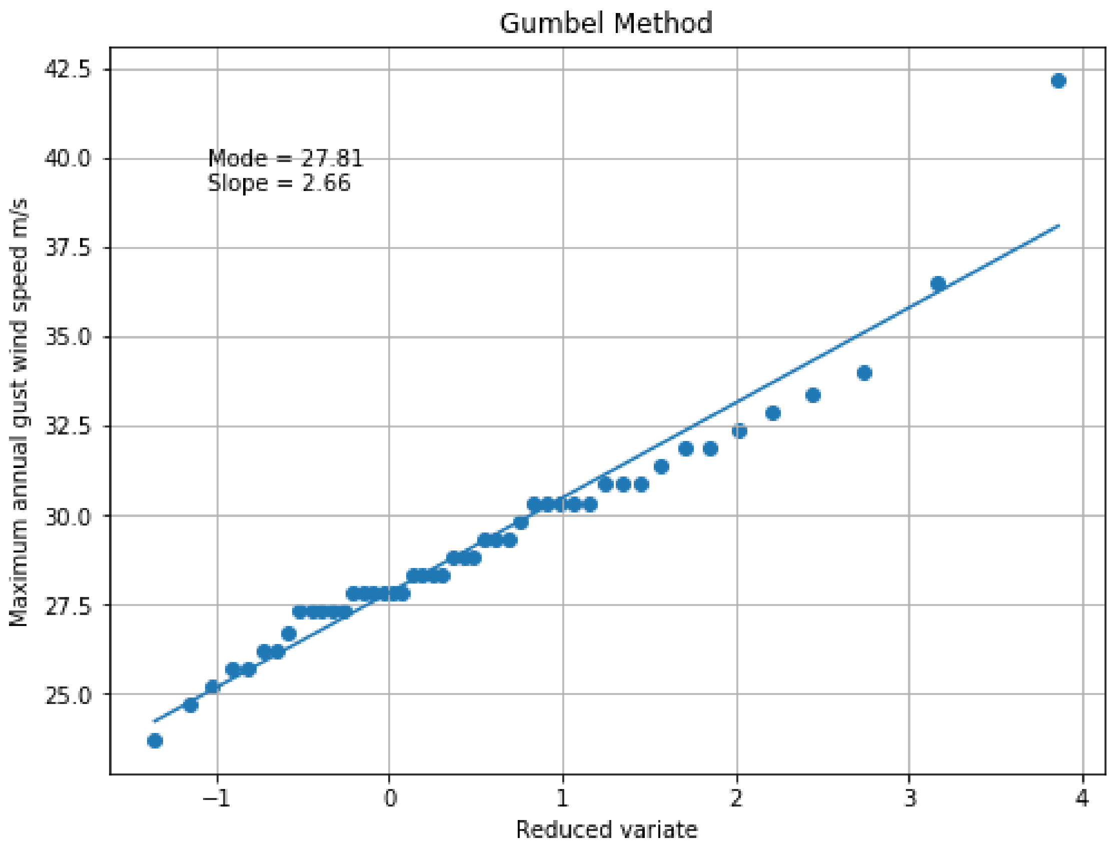
step 5 A straight line is fit for the data reduced variate $ y $ and gust wind speed $ U $. Check out the polyfit function

step 6 The slope a and mode u of the best fit line is obtained.

[8]:
```
[gumbel_slope, gumbel_mode] = np.polyfit(gumbel_red_var, gust_sorted, 1)
```

Let us look at the plot

[9]:
```
plt.figure(num=3, figsize=(8, 6))
plt.scatter(gumbel_red_var, gust_sorted)
x = np.linspace(min(gumbel_red_var), max(gumbel_red_var), 50)
plt.plot(x, gumbel_mode + gumbel_slope * x)
plt.text(0.1, 0.8,'Mode = ' + str(round(gumbel_mode,2)) +
         '\nSlope = ' + str(round(gumbel_slope,2)),
         transform=plt.gca().transAxes)
plt.ylabel('Maximum annual gust wind speed m/s')
plt.xlabel('Reduced variate')
plt.title('Gumbel Method')
plt.grid(True)
```



step 7 For a chosen return period an estimate for the maximum gust wind speed can be obtained as

$$U_R = u + a\{-ln[-ln(1-\frac{1}{R})]\}$$

The return period is the estimate of time in which the maximum annual gust wind speed will exceed the design wind speed. The return period is selected based on the design life of the structure.

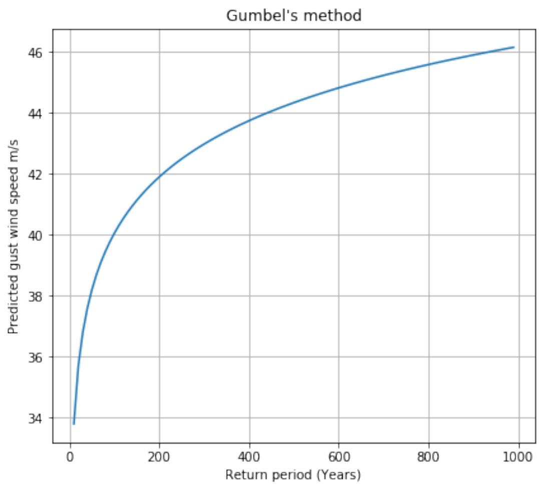
[10]:
```
return_period = np.arange(10, 1000, 10)
gumbel_predicted_gustwind = gumbel_mode + gumbel_slope * (-np.log(-np.log(1-1/
 ↪return_period)))
```

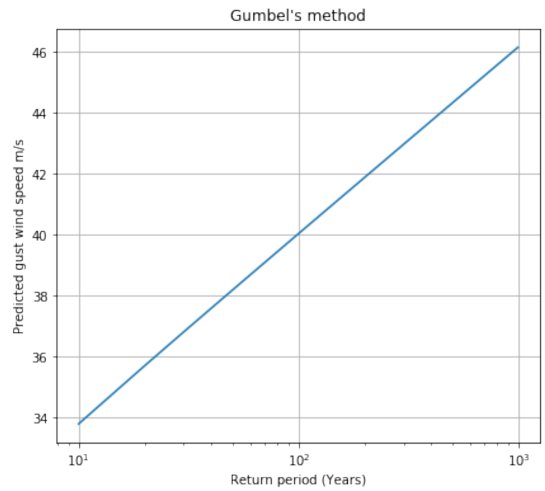
Let us look at the plot

[11]:
```
plt.figure(num=4, figsize=(15, 6))
plt.subplot(1,2,1)
plt.plot(return_period, gumbel_predicted_gustwind)
plt.ylabel('Predicted gust wind speed m/s')
plt.xlabel('Return period (Years)')
plt.title('Gumbel\'s method')
plt.grid(True)

plt.subplot(1,2,2)
```

```python
plt.plot(return_period, gumbel_predicted_gustwind)
plt.xscale('log')
plt.ylabel('Predicted gust wind speed m/s')
plt.xlabel('Return period (Years)')
plt.title('Gumbel\'s method')
plt.grid(True)
```





2. Gringorten's method The steps are similar to that of Gumbel's method, except that

step 3 Each value is assigned a probability of non-exceedence $p\frac{(m-0.44)}{(N+0.12)}$.

[12]:
```python
gringorten_prob_nonexc = (rank - 0.44) /(max_rank + 0.12)
gringorten_red_var = -np.log(-np.log(gringorten_prob_nonexc))
[gringorten_slope, gringorten_mode] = np.polyfit(gringorten_red_var, gust_sorted,1)
gringorten_predicted_gustwind = gringorten_mode + gringorten_slope * (-np.log(-np.log(1-1/return_period)))
```

[13]:
```python
plt.figure(num=5, figsize=(8, 6))
plt.scatter(gringorten_red_var, gust_sorted)
x = np.linspace(min(gringorten_red_var), max(gringorten_red_var), 50)
plt.plot(x, gringorten_mode + gringorten_slope * x)
plt.text(0.1, 0.8,'Mode = ' + str(round(gringorten_mode,2)) +
         '\nSlope = ' + str(round(gringorten_slope,2)),
         transform=plt.gca().transAxes)
plt.ylabel('Maximum annual gust wind speed m/s')
plt.xlabel('Reduced variate')
plt.title('Gringorten\'s method')
plt.grid(True)
```