# EXPLORATORY DATA ANALYSIS AND DATA CLEANING

## Data Loading

### Santa Maria Station

After downloading historical files from Webpage and save them in the current folder as follow: SantaMaria<year>.txt as well as download and save real-time data in the file 'SantaMariaRealTime.txt' , let's load data into tables. The url where Santa Maria data files are available is:

https://www.ndbc.noaa.gov/station_page.php?station=46011

```
historical = true;
dataRead2019 = loading('SantaMaria2019.txt',historical,'2019')
```

dataRead2019 = 16214×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2018 | 12 | 31 | 23 | 50 | 321 | 5.6000 | 7.0000 | 2.6700 |
| 2 | 2019 | 1 | 1 | 0 | 50 | 332 | 7.1000 | 8.4000 | 2.6400 |
| 3 | 2019 | 1 | 1 | 1 | 50 | 329 | 7.2000 | 9.1000 | 2.7400 |
| 4 | 2019 | 1 | 1 | 2 | 50 | 3 | 6.2000 | 8.4000 | 2.9400 |
| 5 | 2019 | 1 | 1 | 3 | 50 | 36 | 7.4000 | 8.4000 | 2.9100 |
| 6 | 2019 | 1 | 1 | 4 | 50 | 52 | 7.3000 | 8.7000 | 2.8900 |
| 7 | 2019 | 1 | 1 | 5 | 50 | 35 | 8.0000 | 9.7000 | 2.7800 |
| 8 | 2019 | 1 | 1 | 6 | 50 | 36 | 6.7000 | 8.5000 | 2.5000 |
| 9 | 2019 | 1 | 1 | 7 | 50 | 58 | 6.6000 | 8.2000 | 2.4200 |
| 10 | 2019 | 1 | 1 | 8 | 50 | 56 | 7.1000 | 8.7000 | 2.3400 |
| 11 | 2019 | 1 | 1 | 9 | 50 | 56 | 8.2000 | 10.1000 | 2.1800 |
| 12 | 2019 | 1 | 1 | 10 | 50 | 63 | 7.9000 | 9.7000 | 2.2400 |
| 13 | 2019 | 1 | 1 | 11 | 50 | 65 | 6.3000 | 7.7000 | 2.2000 |
| 14 | 2019 | 1 | 1 | 12 | 50 | 50 | 5.7000 | 7.2000 | 2.2800 |

⋮

```
historical = true;
dataRead2018 = loading('SantaMaria2018.txt',historical,'2018')
```

dataRead2018 = 8716×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2017 | 12 | 31 | 23 | 50 | 10 | 1.8000 | 2.1000 | 0.8600 |
| 2 | 2018 | 1 | 1 | 0 | 50 | 36 | 0.8000 | 1.4000 | 0.8500 |

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 2018 | 1 | 1 | 1 | 50 | 17 | 0.8000 | 1.1000 | 0.9200 |
| 4 | 2018 | 1 | 1 | 2 | 50 | 354 | 0.5000 | 0.9000 | 0.8700 |
| 5 | 2018 | 1 | 1 | 3 | 50 | 23 | 1.2000 | 1.6000 | 0.9200 |
| 6 | 2018 | 1 | 1 | 4 | 50 | 11 | 1.1000 | 1.3000 | 0.8500 |
| 7 | 2018 | 1 | 1 | 5 | 50 | 325 | 1.1000 | 1.6000 | 0.9000 |
| 8 | 2018 | 1 | 1 | 6 | 50 | 299 | 1.5000 | 1.8000 | 0.8000 |
| 9 | 2018 | 1 | 1 | 7 | 50 | 311 | 2.6000 | 3.1000 | 0.8100 |
| 10 | 2018 | 1 | 1 | 8 | 50 | 329 | 3.0000 | 3.6000 | 0.7400 |
| 11 | 2018 | 1 | 1 | 9 | 50 | 338 | 2.6000 | 3.2000 | 0.7500 |
| 12 | 2018 | 1 | 1 | 10 | 50 | 358 | 3.3000 | 3.9000 | 0.7200 |
| 13 | 2018 | 1 | 1 | 11 | 50 | 350 | 3.6000 | 4.2000 | 0.7000 |
| 14 | 2018 | 1 | 1 | 12 | 50 | 344 | 4.0000 | 4.8000 | 0.6800 |

⋮

```
historical = true;
dataRead2017 = loading('SantaMaria2017.txt',historical,'2017')
```

dataRead2017 = 8685×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2016 | 12 | 31 | 23 | 50 | 278 | 3.0000 | 4.7000 | 2.4200 |
| 2 | 2017 | 1 | 1 | 0 | 50 | 322 | 1.5000 | 2.5000 | 2.5200 |
| 3 | 2017 | 1 | 1 | 1 | 50 | 36 | 6.5000 | 8.2000 | 2.2200 |
| 4 | 2017 | 1 | 1 | 2 | 50 | 346 | 1.4000 | 2.2000 | 2.2700 |
| 5 | 2017 | 1 | 1 | 3 | 50 | 35 | 1.1000 | 2.0000 | 2.5200 |
| 6 | 2017 | 1 | 1 | 4 | 50 | 29 | 4.4000 | 5.7000 | 2.7100 |
| 7 | 2017 | 1 | 1 | 5 | 50 | 352 | 5.1000 | 6.4000 | 2.7800 |
| 8 | 2017 | 1 | 1 | 6 | 50 | 8 | 6.9000 | 8.4000 | 2.7100 |
| 9 | 2017 | 1 | 1 | 7 | 50 | 11 | 7.2000 | 8.3000 | 2.5500 |
| 10 | 2017 | 1 | 1 | 8 | 50 | 13 | 3.5000 | 4.6000 | 2.3600 |
| 11 | 2017 | 1 | 1 | 9 | 50 | 30 | 4.0000 | 5.2000 | 2.1800 |
| 12 | 2017 | 1 | 1 | 10 | 50 | 359 | 5.2000 | 6.3000 | 2.3400 |
| 13 | 2017 | 1 | 1 | 11 | 50 | 336 | 4.4000 | 6.3000 | 2.1300 |
| 14 | 2017 | 1 | 1 | 12 | 50 | 343 | 8.1000 | 9.9000 | 2.2700 |

⋮

```
historical = true;
dataRead2016 = loading('SantaMaria2016.txt',historical,'2016')
```

dataRead2016 = 8669×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2015 | 12 | 31 | 23 | 50 | 316 | 2.7000 | 4.0000 | 1.3700 |
| 2 | 2016 | 1 | 1 | 0 | 50 | 314 | 3.6000 | 4.8000 | 1.4200 |
| 3 | 2016 | 1 | 1 | 1 | 50 | 311 | 3.6000 | 4.6000 | 1.4500 |
| 4 | 2016 | 1 | 1 | 2 | 50 | 305 | 3.0000 | 4.3000 | 1.4500 |
| 5 | 2016 | 1 | 1 | 3 | 50 | 340 | 3.1000 | 4.2000 | 1.4600 |
| 6 | 2016 | 1 | 1 | 4 | 50 | 349 | 2.9000 | 3.6000 | 1.3400 |
| 7 | 2016 | 1 | 1 | 5 | 50 | 334 | 3.2000 | 4.3000 | 1.2600 |
| 8 | 2016 | 1 | 1 | 6 | 50 | 34 | 2.2000 | 3.4000 | 1.3100 |
| 9 | 2016 | 1 | 1 | 7 | 50 | 61 | 1.9000 | 3.0000 | 1.1500 |
| 10 | 2016 | 1 | 1 | 8 | 50 | 99 | 1.1000 | 1.9000 | 1.1800 |
| 11 | 2016 | 1 | 1 | 9 | 50 | 124 | 1.4000 | 2.6000 | 1.2700 |
| 12 | 2016 | 1 | 1 | 10 | 50 | 146 | 1.4000 | 2.4000 | 1.1700 |
| 13 | 2016 | 1 | 1 | 11 | 50 | 125 | 2.6000 | 3.4000 | 1.1900 |
| 14 | 2016 | 1 | 1 | 12 | 50 | 98 | 3.7000 | 4.6000 | 1.2400 |

⋮

```
historical = true;
dataRead2015 = loading('SantaMaria2015.txt',historical,'2015')
```

dataRead2015 = 8738×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2014 | 12 | 31 | 23 | 50 | 999 | 99 | 99 | 2.1000 |
| 2 | 2015 | 1 | 1 | 0 | 50 | 999 | 99 | 99 | 2.0900 |
| 3 | 2015 | 1 | 1 | 1 | 50 | 999 | 99 | 99 | 2.1000 |
| 4 | 2015 | 1 | 1 | 2 | 50 | 999 | 99 | 99 | 2.0700 |
| 5 | 2015 | 1 | 1 | 3 | 50 | 999 | 99 | 99 | 1.9500 |
| 6 | 2015 | 1 | 1 | 4 | 50 | 999 | 99 | 99 | 1.9300 |
| 7 | 2015 | 1 | 1 | 5 | 50 | 999 | 99 | 99 | 1.7100 |
| 8 | 2015 | 1 | 1 | 6 | 50 | 999 | 99 | 99 | 1.6200 |
| 9 | 2015 | 1 | 1 | 7 | 50 | 999 | 99 | 99 | 1.5300 |

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 2015 | 1 | 1 | 8 | 50 | 999 | 99 | 99 | 1.4700 |
| 11 | 2015 | 1 | 1 | 9 | 50 | 999 | 99 | 99 | 1.4300 |
| 12 | 2015 | 1 | 1 | 10 | 50 | 999 | 99 | 99 | 1.3800 |
| 13 | 2015 | 1 | 1 | 11 | 50 | 999 | 99 | 99 | 1.1700 |
| 14 | 2015 | 1 | 1 | 12 | 50 | 999 | 99 | 99 | 1.2300 |

⋮

```
dataRead2014 = loading('SantaMaria2014.txt',historical,'2014')
```

dataRead2014 = 8750×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2013 | 12 | 31 | 23 | 50 | 314 | 7.1000 | 8.7000 | 2.0000 |
| 2 | 2014 | 1 | 1 | 0 | 50 | 307 | 9.1000 | 10.7000 | 1.9500 |
| 3 | 2014 | 1 | 1 | 1 | 50 | 317 | 10.0000 | 11.6000 | 1.9200 |
| 4 | 2014 | 1 | 1 | 2 | 50 | 318 | 8.9000 | 11.3000 | 2.2300 |
| 5 | 2014 | 1 | 1 | 3 | 50 | 326 | 8.9000 | 11.2000 | 2.0400 |
| 6 | 2014 | 1 | 1 | 4 | 50 | 323 | 7.9000 | 9.8000 | 99.0000 |
| 7 | 2014 | 1 | 1 | 5 | 50 | 327 | 7.5000 | 9.2000 | 2.2500 |
| 8 | 2014 | 1 | 1 | 6 | 50 | 328 | 7.4000 | 8.4000 | 2.2400 |
| 9 | 2014 | 1 | 1 | 7 | 50 | 334 | 7.1000 | 99.0000 | 99.0000 |
| 10 | 2014 | 1 | 1 | 8 | 50 | 334 | 6.8000 | 8.1000 | 2.0900 |
| 11 | 2014 | 1 | 1 | 9 | 50 | 333 | 7.1000 | 8.2000 | 2.0900 |
| 12 | 2014 | 1 | 1 | 10 | 50 | 335 | 7.0000 | 8.7000 | 1.8900 |
| 13 | 2014 | 1 | 1 | 11 | 50 | 337 | 7.2000 | 8.4000 | 1.6400 |
| 14 | 2014 | 1 | 1 | 12 | 50 | 358 | 2.5000 | 3.3000 | 1.9200 |

⋮

```
historical = false;
dataReadRealTime = loading('SantaMariaRealTime.txt',historical,'2020')
```

dataReadRealTime = 6464×19 table

...

| | x_YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2020 | 4 | 9 | 12 | 10 | 220 | 4 | 5 | NaN |
| 2 | 2020 | 4 | 9 | 12 | 0 | 230 | 4 | 5 | NaN |

| | x_YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 2020 | 4 | 9 | 11 | 50 | 220 | 4 | 6 | 1.6000 |
| 4 | 2020 | 4 | 9 | 11 | 40 | 210 | 5 | 6 | NaN |
| 5 | 2020 | 4 | 9 | 11 | 30 | 220 | 6 | 7 | NaN |
| 6 | 2020 | 4 | 9 | 11 | 20 | 230 | 5 | 6 | NaN |
| 7 | 2020 | 4 | 9 | 11 | 10 | 230 | 6 | 7 | NaN |
| 8 | 2020 | 4 | 9 | 11 | 0 | 220 | 6 | 7 | NaN |
| 9 | 2020 | 4 | 9 | 10 | 50 | 220 | 6 | 7 | 1.4000 |
| 10 | 2020 | 4 | 9 | 10 | 40 | 220 | 6 | 7 | NaN |
| 11 | 2020 | 4 | 9 | 10 | 30 | 210 | 4 | 6 | NaN |
| 12 | 2020 | 4 | 9 | 10 | 20 | 210 | 5 | 6 | NaN |
| 13 | 2020 | 4 | 9 | 10 | 10 | 240 | 4 | 5 | NaN |
| 14 | 2020 | 4 | 9 | 10 | 0 | 250 | 5 | 6 | NaN |

⋮

```
historical = true;
dataRead2013 = loading('SantaMaria2013.txt',historical,'2013')
```

dataRead2013 = 8753×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2012 | 12 | 31 | 23 | 50 | 313 | 5.9000 | 7.1000 | 2.2700 |
| 2 | 2013 | 1 | 1 | 0 | 50 | 301 | 7.4000 | 8.8000 | 2.0000 |
| 3 | 2013 | 1 | 1 | 1 | 50 | 306 | 7.2000 | 8.8000 | 2.2400 |
| 4 | 2013 | 1 | 1 | 2 | 50 | 316 | 5.7000 | 6.6000 | 2.2200 |
| 5 | 2013 | 1 | 1 | 3 | 50 | 356 | 4.3000 | 5.1000 | 2.0700 |
| 6 | 2013 | 1 | 1 | 4 | 50 | 338 | 6.5000 | 7.7000 | 2.2400 |
| 7 | 2013 | 1 | 1 | 5 | 50 | 348 | 6.5000 | 8.0000 | 2.1400 |
| 8 | 2013 | 1 | 1 | 6 | 50 | 6 | 5.5000 | 6.5000 | 2.2800 |
| 9 | 2013 | 1 | 1 | 7 | 50 | 329 | 4.0000 | 4.9000 | 2.0900 |
| 10 | 2013 | 1 | 1 | 8 | 50 | 6 | 2.8000 | 3.6000 | 2.1400 |
| 11 | 2013 | 1 | 1 | 9 | 50 | 116 | 1.8000 | 2.9000 | 1.9800 |
| 12 | 2013 | 1 | 1 | 10 | 50 | 123 | 2.1000 | 3.2000 | 2.1000 |
| 13 | 2013 | 1 | 1 | 11 | 50 | 47 | 0.5000 | 1.5000 | 2.6000 |
| 14 | 2013 | 1 | 1 | 12 | 50 | 63 | 2.2000 | 3.3000 | 2.4700 |

⋮

```
historical = true;
dataRead2012 = loading('SantaMaria2012.txt',historical,'2012')
```

dataRead2012 = 8773×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2011 | 12 | 31 | 23 | 50 | 334 | 9.1000 | 10.6000 | 2.2400 |
| 2 | 2012 | 1 | 1 | 0 | 50 | 332 | 8.6000 | 10.0000 | 2.2400 |
| 3 | 2012 | 1 | 1 | 1 | 50 | 340 | 8.4000 | 10.3000 | 2.1500 |
| 4 | 2012 | 1 | 1 | 2 | 50 | 357 | 5.4000 | 6.4000 | 2.4100 |
| 5 | 2012 | 1 | 1 | 3 | 50 | 5 | 2.9000 | 3.7000 | 2.2600 |
| 6 | 2012 | 1 | 1 | 4 | 50 | 353 | 4.6000 | 5.4000 | 2.3000 |
| 7 | 2012 | 1 | 1 | 5 | 50 | 11 | 3.3000 | 4.5000 | 2.1200 |
| 8 | 2012 | 1 | 1 | 6 | 50 | 67 | 0.7000 | 1.7000 | 2.1600 |
| 9 | 2012 | 1 | 1 | 7 | 50 | 142 | 1.4000 | 2.5000 | 2.2100 |
| 10 | 2012 | 1 | 1 | 8 | 50 | 130 | 0.8000 | 1.9000 | 2.1100 |
| 11 | 2012 | 1 | 1 | 9 | 50 | 96 | 0.4000 | 1.1000 | 2.4100 |
| 12 | 2012 | 1 | 1 | 10 | 50 | 107 | 0.3000 | 1.5000 | 2.2500 |
| 13 | 2012 | 1 | 1 | 11 | 50 | 65 | 0.6000 | 1.7000 | 2.4400 |
| 14 | 2012 | 1 | 1 | 12 | 50 | 139 | 0.5000 | 1.3000 | 2.3300 |

⋮

```
historical = true;
dataRead2011 = loading('SantaMaria2011.txt',historical,'2011')
```

dataRead2011 = 8685×18 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2010 | 12 | 31 | 23 | 50 | 323 | 5.1000 | 6.1000 | 1.8700 |
| 2 | 2011 | 1 | 1 | 0 | 50 | 328 | 4.7000 | 5.6000 | 1.9800 |
| 3 | 2011 | 1 | 1 | 1 | 50 | 324 | 5.0000 | 5.9000 | 1.7300 |
| 4 | 2011 | 1 | 1 | 2 | 50 | 345 | 3.9000 | 4.6000 | 1.7800 |
| 5 | 2011 | 1 | 1 | 3 | 50 | 350 | 2.2000 | 3.0000 | 1.8800 |
| 6 | 2011 | 1 | 1 | 4 | 50 | 340 | 3.0000 | 4.2000 | 1.7200 |
| 7 | 2011 | 1 | 1 | 5 | 50 | 6 | 1.6000 | 2.2000 | 1.9400 |
| 8 | 2011 | 1 | 1 | 6 | 50 | 52 | 1.0000 | 1.4000 | 1.8900 |
| 9 | 2011 | 1 | 1 | 7 | 50 | 135 | 5.1000 | 6.5000 | 2.0900 |

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 2011 | 1 | 1 | 8 | 50 | 142 | 5.0000 | 6.3000 | 2.1000 |
| 11 | 2011 | 1 | 1 | 9 | 50 | 133 | 4.8000 | 5.8000 | 1.9100 |
| 12 | 2011 | 1 | 1 | 10 | 50 | 131 | 5.0000 | 6.1000 | 1.9000 |
| 13 | 2011 | 1 | 1 | 11 | 50 | 140 | 6.3000 | 7.7000 | 1.6800 |
| 14 | 2011 | 1 | 1 | 12 | 50 | 143 | 6.5000 | 7.7000 | 99.0000 |

⋮

```
historical = true;
dataRead2010 = loading('SantaMaria2010.txt',historical,'2010')
```

```
dataRead2010 = 7621×18 table
```

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2010 | 2 | 16 | 22 | 50 | 307 | 6.7000 | 8.1000 | 99.0000 |
| 2 | 2010 | 2 | 16 | 23 | 50 | 319 | 7.0000 | 8.5000 | 2.4500 |
| 3 | 2010 | 2 | 17 | 0 | 50 | 320 | 6.6000 | 7.8000 | 2.6400 |
| 4 | 2010 | 2 | 17 | 1 | 50 | 327 | 6.5000 | 7.5000 | 2.5800 |
| 5 | 2010 | 2 | 17 | 2 | 50 | 327 | 5.2000 | 6.5000 | 2.4600 |
| 6 | 2010 | 2 | 17 | 3 | 50 | 320 | 4.1000 | 5.4000 | 2.5600 |
| 7 | 2010 | 2 | 17 | 4 | 50 | 13 | 2.0000 | 2.5000 | 2.4200 |
| 8 | 2010 | 2 | 17 | 5 | 50 | 32 | 1.8000 | 2.4000 | 2.5500 |
| 9 | 2010 | 2 | 17 | 6 | 50 | 326 | 1.9000 | 2.8000 | 2.2000 |
| 10 | 2010 | 2 | 17 | 7 | 50 | 2 | 2.1000 | 2.7000 | 2.5200 |
| 11 | 2010 | 2 | 17 | 8 | 50 | 27 | 3.9000 | 4.6000 | 2.4000 |
| 12 | 2010 | 2 | 17 | 9 | 50 | 1 | 1.1000 | 1.7000 | 2.2700 |
| 13 | 2010 | 2 | 17 | 10 | 50 | 341 | 4.2000 | 5.1000 | 2.3500 |
| 14 | 2010 | 2 | 17 | 11 | 50 | 14 | 2.4000 | 3.2000 | 2.4900 |

⋮

## Visual Inspection of Data features

**SANTA MARÍA dataset**:

**Historical Files:** YY, MM, DD, hh, mm, WDIR, WSPD, GST, WVHT, DPD, APD, MWD, PRES, ATMP, WTMP, DEWP, VIS, TIDE

**Real Files:** YY, MM, DD, hh, mm, WDIR, WSPD, GST, WVHT, DPD, APD, MWD, PRES, ATMP, WTMP, DEWP, VIS, PITDY TIDE

Colour Leyend: Null values, Variable abailable only in Real Time Files

```
%Extracting Features available in both files and not completely null
dataUsed2018 = dataRead2018(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
```

```
dataUsed2018 = 8716×15 table
```

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2017 | 12 | 31 | 23 | 50 | 10 | 1.8000 | 2.1000 | 0.8600 |
| 2 | 2018 | 1 | 1 | 0 | 50 | 36 | 0.8000 | 1.4000 | 0.8500 |
| 3 | 2018 | 1 | 1 | 1 | 50 | 17 | 0.8000 | 1.1000 | 0.9200 |
| 4 | 2018 | 1 | 1 | 2 | 50 | 354 | 0.5000 | 0.9000 | 0.8700 |
| 5 | 2018 | 1 | 1 | 3 | 50 | 23 | 1.2000 | 1.6000 | 0.9200 |
| 6 | 2018 | 1 | 1 | 4 | 50 | 11 | 1.1000 | 1.3000 | 0.8500 |
| 7 | 2018 | 1 | 1 | 5 | 50 | 325 | 1.1000 | 1.6000 | 0.9000 |
| 8 | 2018 | 1 | 1 | 6 | 50 | 299 | 1.5000 | 1.8000 | 0.8000 |
| 9 | 2018 | 1 | 1 | 7 | 50 | 311 | 2.6000 | 3.1000 | 0.8100 |
| 10 | 2018 | 1 | 1 | 8 | 50 | 329 | 3.0000 | 3.6000 | 0.7400 |
| 11 | 2018 | 1 | 1 | 9 | 50 | 338 | 2.6000 | 3.2000 | 0.7500 |
| 12 | 2018 | 1 | 1 | 10 | 50 | 358 | 3.3000 | 3.9000 | 0.7200 |
| 13 | 2018 | 1 | 1 | 11 | 50 | 350 | 3.6000 | 4.2000 | 0.7000 |
| 14 | 2018 | 1 | 1 | 12 | 50 | 344 | 4.0000 | 4.8000 | 0.6800 |

⋮

```
dataUsed2019 = dataRead2019(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2017 = dataRead2017(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2016 = dataRead2016(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2015 = dataRead2015(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2014 = dataRead2014(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsedRealTime = dataReadRealTime(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WV
```

```
dataUsed2013 = dataRead2013(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2012 = dataRead2012(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2011 = dataRead2011(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
dataUsed2010 = dataRead2010(:, {'YY', 'MM', 'DD', 'hh', 'mm', 'WDIR', 'WSPD','GST', 'WVHT','DPD
```

## Visualization the data before any processing

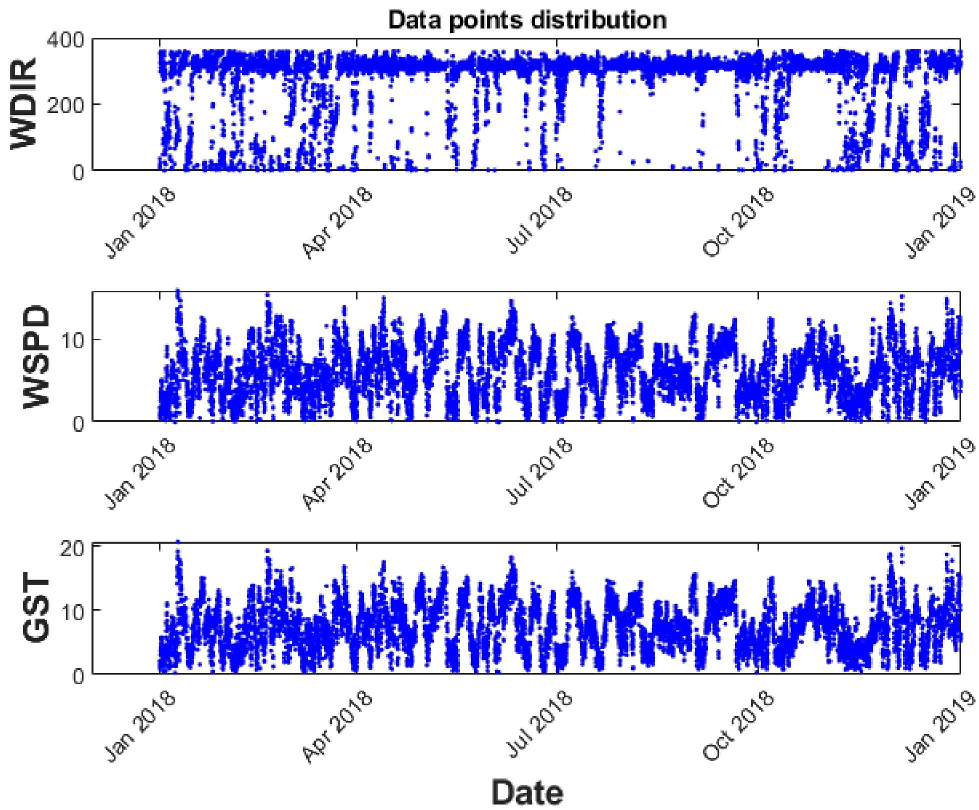
Firstly we are going to create an extra variable 'date' in order to represent each feature date-based.

### Extra feature

```
dataUsed2019.DATE = datetime(dataUsed2019.YY, dataUsed2019.MM, dataUsed2019.DD, dataUsed2019.hh
dataUsed2018.DATE = datetime(dataUsed2018.YY, dataUsed2018.MM, dataUsed2018.DD, dataUsed2018.hh
dataUsed2017.DATE = datetime(dataUsed2017.YY, dataUsed2017.MM, dataUsed2017.DD, dataUsed2017.hh
dataUsed2016.DATE = datetime(dataUsed2016.YY, dataUsed2016.MM, dataUsed2016.DD, dataUsed2016.hh
dataUsed2015.DATE = datetime(dataUsed2015.YY, dataUsed2015.MM, dataUsed2015.DD, dataUsed2015.hh
dataUsed2014.DATE = datetime(dataUsed2014.YY, dataUsed2014.MM, dataUsed2014.DD, dataUsed2014.hh
dataUsedRealTime.DATE = datetime(dataUsedRealTime.YY, dataUsedRealTime.MM, dataUsedRealTime.DD,
dataUsed2013.DATE = datetime(dataUsed2013.YY, dataUsed2013.MM, dataUsed2013.DD, dataUsed2013.hh
dataUsed2012.DATE = datetime(dataUsed2012.YY, dataUsed2012.MM, dataUsed2012.DD, dataUsed2012.hh
dataUsed2011.DATE = datetime(dataUsed2011.YY, dataUsed2011.MM, dataUsed2011.DD, dataUsed2011.hh
dataUsed2010.DATE = datetime(dataUsed2010.YY, dataUsed2010.MM, dataUsed2010.DD, dataUsed2010.hh
```

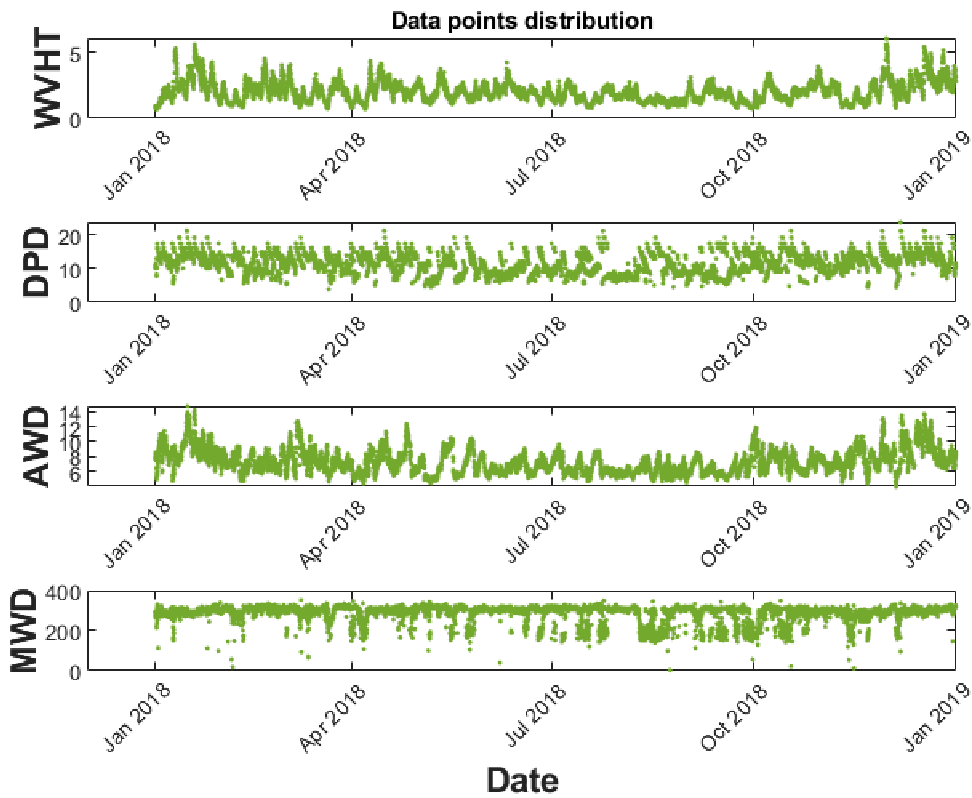
### Wind features

```
subplot(3,1,1)
plot(dataUsed2018.DATE, dataUsed2018.WDIR, '.','MarkerSize',5,"Color",'blue')
ylabel('WDIR',"FontSize",13,'FontWeight',"bold")
title('Data points distribution')
xtickangle(45)

subplot(3,1,2)
plot(dataUsed2018.DATE, dataUsed2018.WSPD, '.','MarkerSize',5,"Color",'blue')
ylabel('WSPD',"FontSize",13,'FontWeight',"bold")
xtickangle(45)

subplot(3,1,3)
plot(dataUsed2018.DATE, dataUsed2018.GST, '.','MarkerSize',5,"Color",'blue');
xlabel('Date',"FontSize",13,'FontWeight',"bold")
ylabel('GST',"FontSize",13,'FontWeight',"bold")
xtickangle(45)
```




## Waves features

```
clf
subplot(4,1,1)
plot(dataUsed2018.DATE, dataUsed2018.WVHT, '.','MarkerSize',5,"Color",[0.4660 0.6740 0.1880])
ylabel('WVHT',"FontSize",13,'FontWeight',"bold")
title('Data points distribution')
xtickangle(45)

subplot(4,1,2)
plot(dataUsed2018.DATE, dataUsed2018.DPD, '.','MarkerSize',5,"Color",[0.4660 0.6740 0.1880])
ylabel('DPD',"FontSize",13,'FontWeight',"bold")
xtickangle(45)

subplot(4,1,3)
plot(dataUsed2018.DATE, dataUsed2018.APD, '.','MarkerSize',5,"Color",[0.4660 0.6740 0.1880])
ylabel('AWD',"FontSize",13,'FontWeight',"bold")
xtickangle(45)

subplot(4,1,4)
plot(dataUsed2018.DATE, dataUsed2018.MWD, '.','MarkerSize',5,"Color",[0.4660 0.6740 0.1880])
xlabel('Date',"FontSize",13,'FontWeight',"bold")
ylabel('MWD',"FontSize",13,'FontWeight',"bold")
xtickangle(45)
```




## Temperatures and Pressure

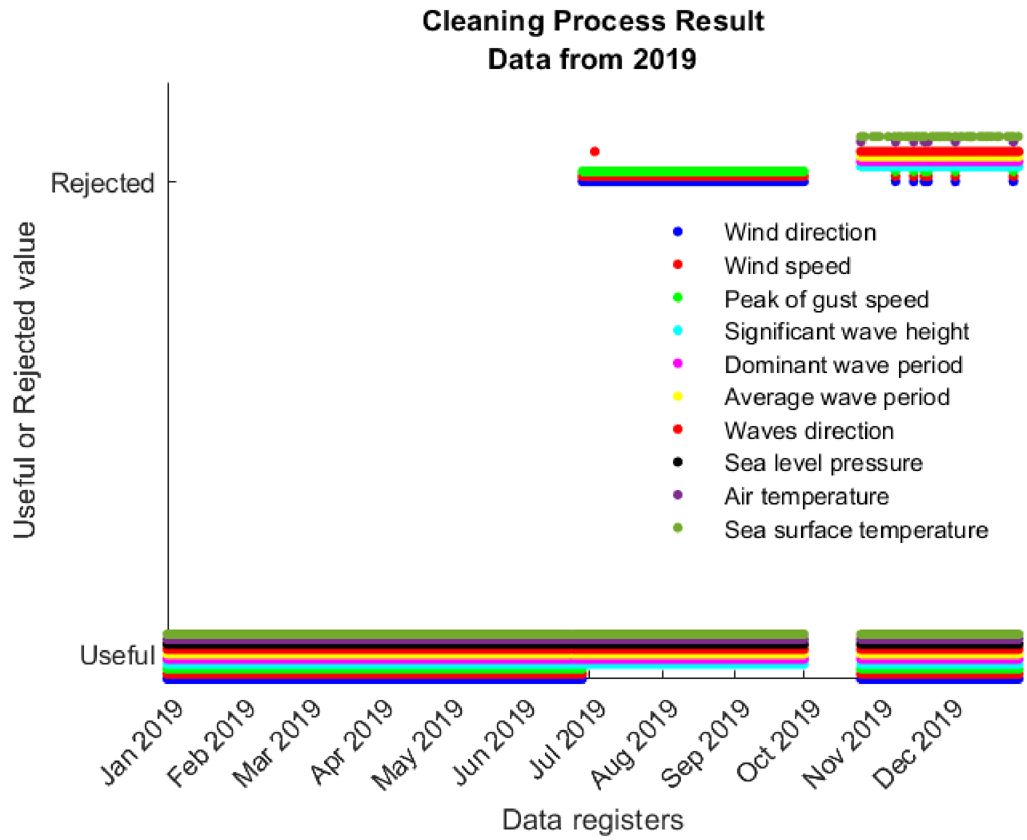
```
clf
subplot(3,1,1)
plot(dataUsed2018.DATE, dataUsed2018.PRES, '.','MarkerSize',5,"Color",'red')
ylabel('PRES',"FontSize",13,'FontWeight',"bold")
title('Data points distribution')
xtickangle(45)

subplot(3,1,2)
plot(dataUsed2018.DATE, dataUsed2018.ATMP, '.','MarkerSize',5,"Color",'red')
ylabel('ATMP',"FontSize",13,'FontWeight',"bold")
xtickangle(45)

subplot(3,1,3)
plot(dataUsed2018.DATE, dataUsed2018.WTMP, '.','MarkerSize',5,"Color",'red')
xlabel('Date',"FontSize",13,'FontWeight',"bold")
ylabel('WTMP',"FontSize",13,'FontWeight',"bold")
xtickangle(45)
```

## Data Quality Assurance : Data Cleaning

We are going to delete rows on datasets after selecting features which contain any missing value or series of number 9s . Features **DWP, VIS and TIDE** haven't been selected in data set combination because they are **completely null** ( there isn't any measure available for them). **Clean data** have been saved in variables with name like **data<year>.a**

## Missing values

```
historical = true;
[data2019, idxNaN2019]  = cleaning(dataUsed2019,historical);
plotCleaningResult(dataUsed2019, idxNaN2019,'2019')
```




We can see data points (rows of file) of every features . Each feature points are represented with a color captioned and points are separated between these valid point with a mesure value and points or register which are missing (corresponding to 'MM' in real time files and series of 9s in Historical files)

Code for plotting has been encapsulated in the function 'plotCleaningResult' in order to reuse it for different data files.

### Number of failed points with one or more missing measure

```
rejected2019 = size(dataUsed2019,1) - size(data2019, 1)
```

```
rejected2019 = 10411
```

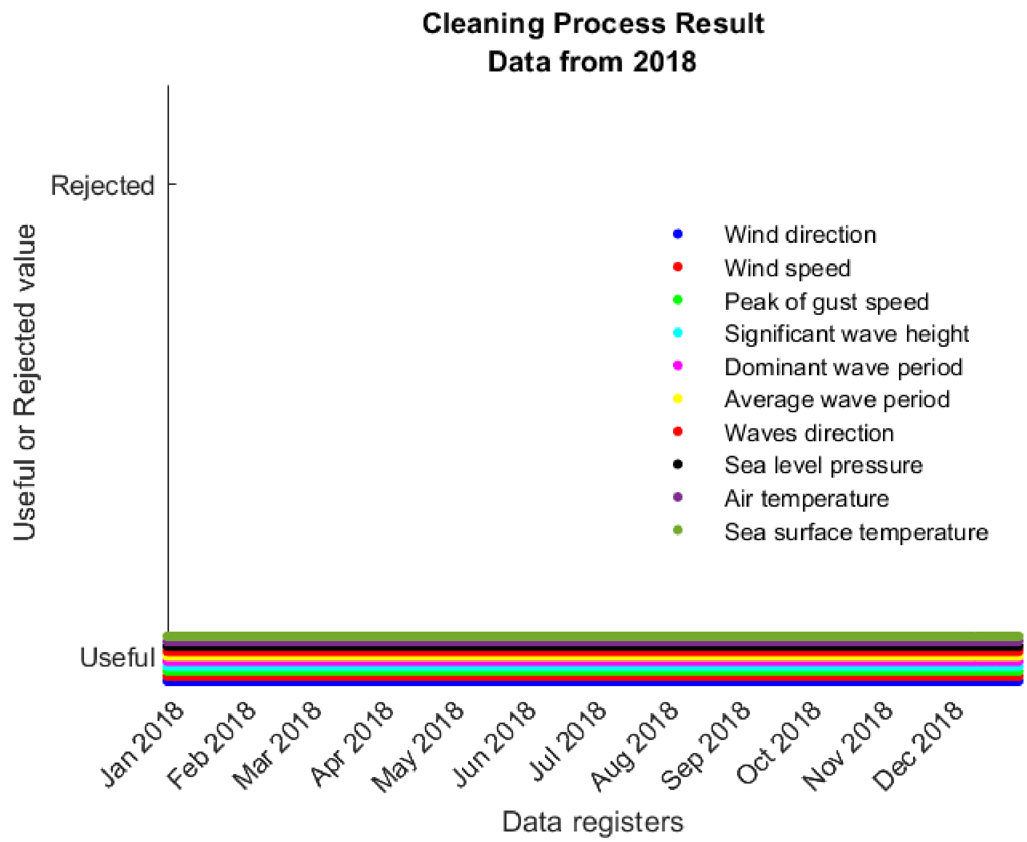
### Percentage of these failed points respect to total registers

```
percentage2019 = (size(dataUsed2019,1) - size(data2019, 1))/size(dataUsed2019,1) * 100
```

```
percentage2019 = 64.2099
```

```
dataUsed2018.DATE = datetime(dataUsed2018.YY, dataUsed2018.MM,dataUsed2018.DD, dataUsed2018.hh,
```

```
historical = true;
[data2018 , idxNaN ] = cleaning(dataUsed2018,historical);
plotCleaningResult(dataUsed2018,idxNaN,'2018')
```




**Number of failed points with one or more missing measure**

```
rejected2018 = size(dataUsed2018,1) - size(data2018, 1)
```

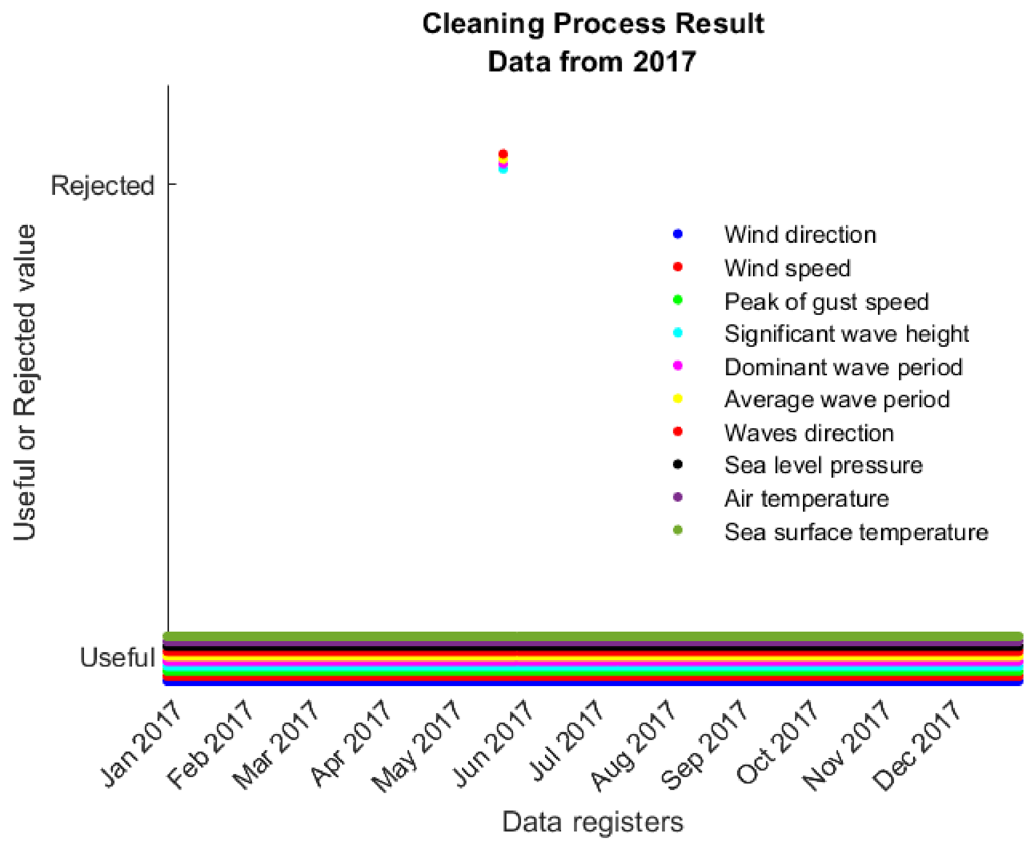
```
rejected2018 = 0
```

**Percentage of these failed points respect to total registers**

```
percentage2018 = (size(dataUsed2018,1) - size(data2018, 1))/size(dataUsed2018,1) * 100
```

```
percentage2018 = 0
```

```
historical = true;
[data2017, idxNaN2017] = cleaning(dataUsed2017,historical);
plotCleaningResult(dataUsed2017,idxNaN2017,'2017')
```




### Number of failed points with one or more missing measure

```
rejected2017 = size(dataUsed2017,1) - size(data2017, 1)
```

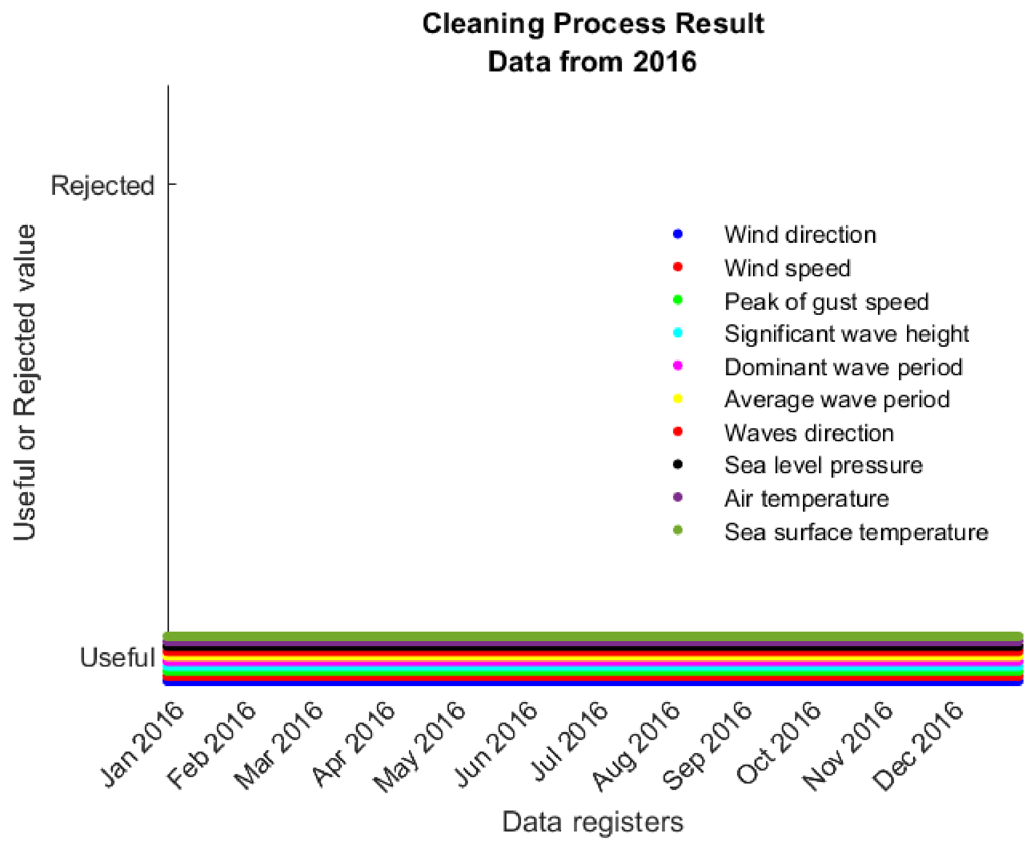
```
rejected2017 = 1
```

### Percentage of these failed points respect to total registers

```
percentage2017 = (size(dataUsed2017,1) - size(data2017, 1))/size(dataUsed2017,1) * 100
```

```
percentage2017 = 0.0115
```

```
historical = true;
[data2016, idxNaN2016] = cleaning(dataUsed2016,historical);
plotCleaningResult(dataUsed2016,idxNaN2016,'2016')
```




### Number of failed points with one or more missing measure

```
rejected2016 = size(dataUsed2016,1) - size(data2016, 1)
```

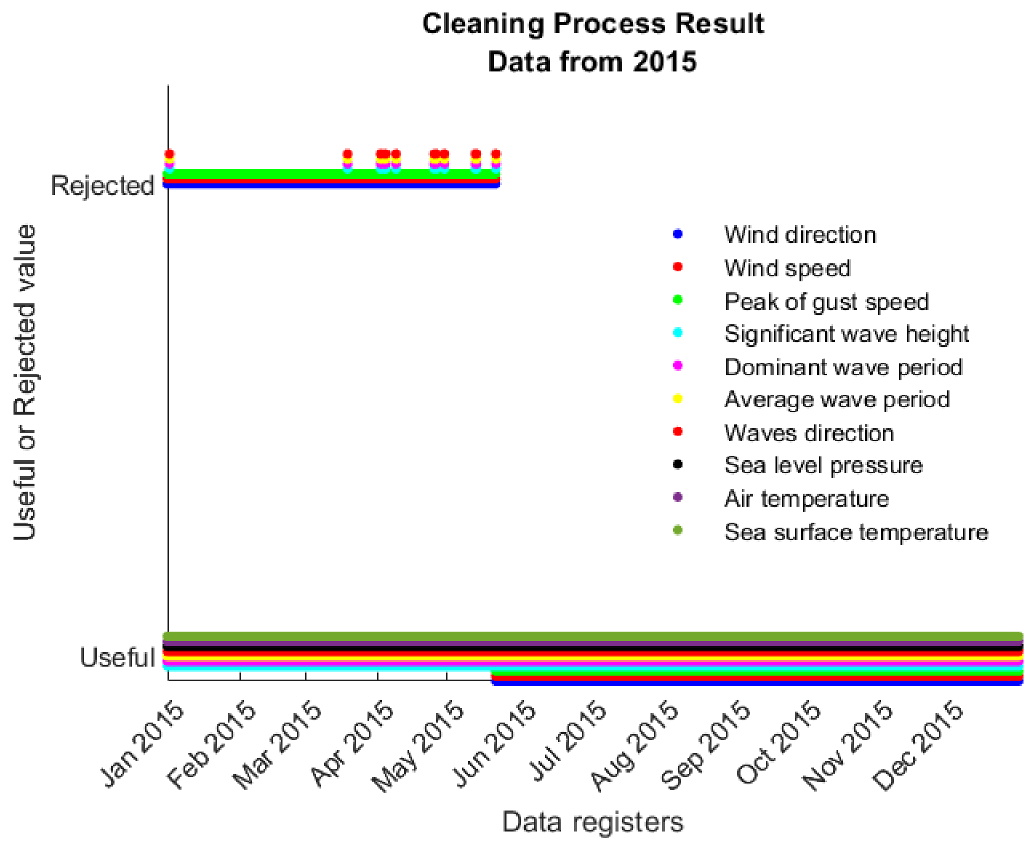
```
rejected2016 = 0
```

### Percentage of these failed points respect to total registers

```
percentage2016 = (size(dataUsed2016,1) - size(data2016, 1))/size(dataUsed2016,1) * 100
```

```
percentage2016 = 0
```

```
historical = true;
[data2015, idxNaN2015] = cleaning(dataUsed2015,historical);
plotCleaningResult(dataUsed2015,idxNaN2015, '2015')
```




**Number of failed points with one or more missing measure**

```
rejected2015 = size(dataUsed2015,1) - size(data2015, 1)
```

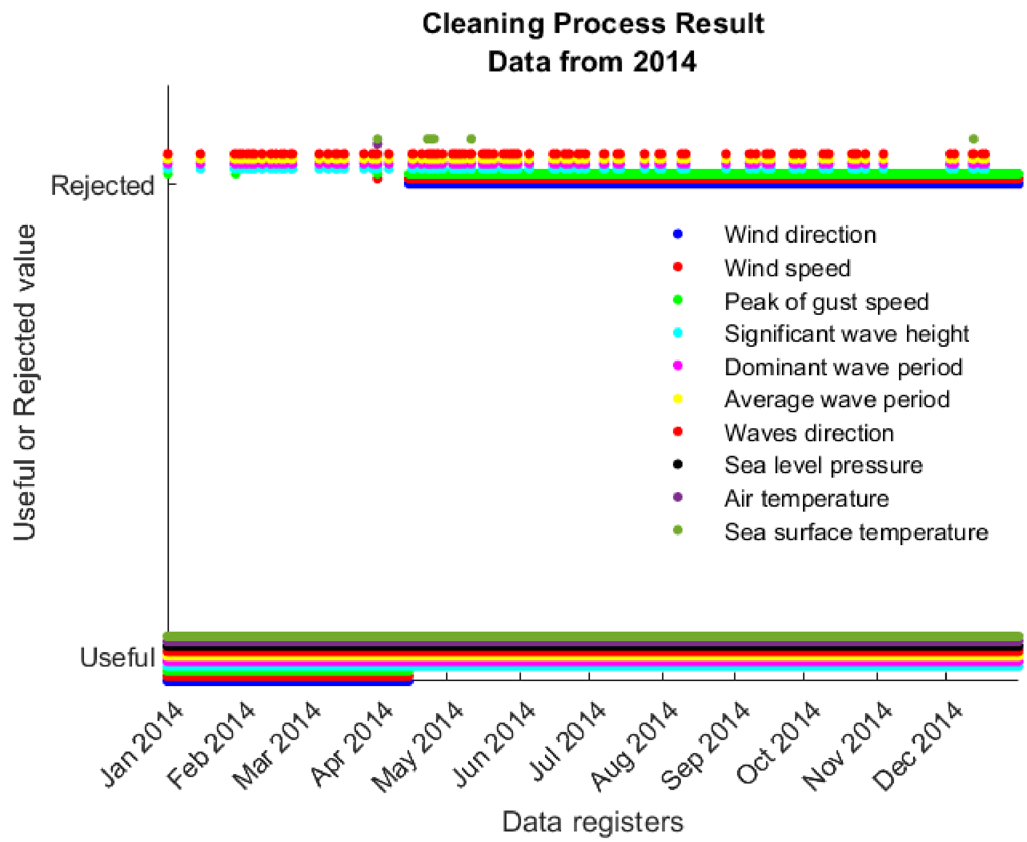
```
rejected2015 = 3380
```

**Percentage of these failed points respect to total registers**

```
percentage2015 = (size(dataUsed2015,1) - size(data2015, 1))/size(dataUsed2015,1) * 100
```

```
percentage2015 = 38.6816
```

```
historical = true;
[data2014, idxNaN2014] = cleaning(dataUsed2014,historical);
plotCleaningResult(dataUsed2014,idxNaN2014, '2014')
```




**Number of failed points with one or more missing measure**

```
rejected2014 = size(dataUsed2014,1) - size(data2014, 1)
```

```
rejected2014 = 6306
```

**Percentage of these failed points respect to total registers**

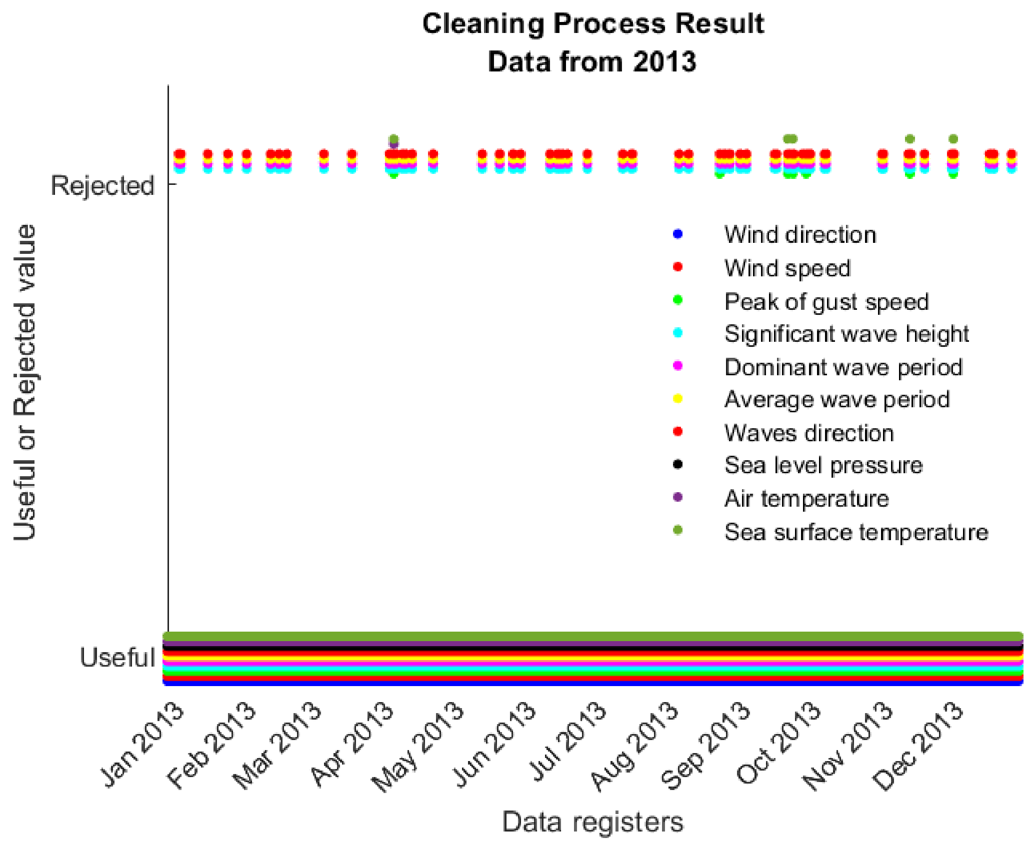
```
percentage2014 = (size(dataUsed2014,1) - size(data2014, 1))/size(dataUsed2014,1) * 100
```

```
percentage2014 = 72.0686
```

In the case of Data set from 2014 **70% approximately of data rows don't contain a wind direction measure,** which could be caused by a fault in the anemometer or any exceptional situation related to data storage**.**

So that, we are going to download data from 2013 to have more registers in case we could need them to train models with more training examples. ( **We add 2013 data load in the corresponding sections above)**

```
historical = true;
[data2013, idxNaN2013] = cleaning(dataUsed2013,historical);
plotCleaningResult(dataUsed2013,idxNaN2013, '2013')
```




**Number of failed points with one or more missing measure**

```
rejected2013 = size(dataUsed2013,1) - size(data2013, 1)
```

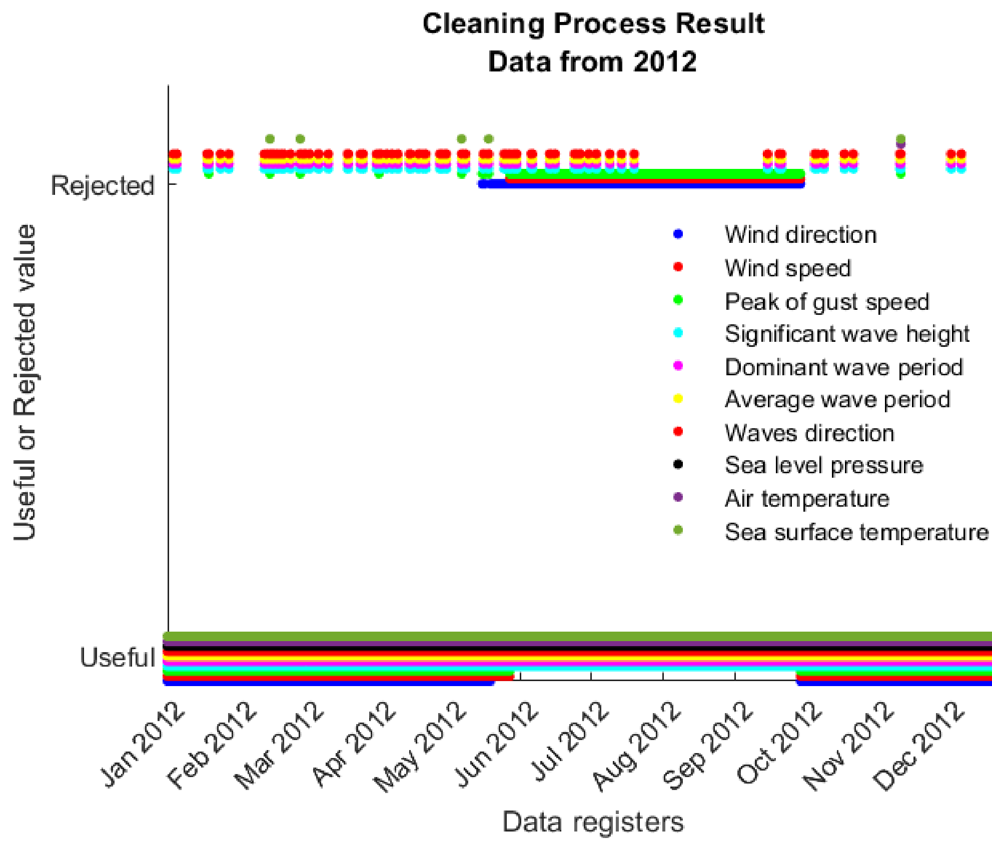
```
rejected2013 = 89
```

**Percentage of these failed points respect to total registers**

```
percentage2013 = (size(dataUsed2013,1) - size(data2013, 1))/size(dataUsed2013,1) * 100
```

```
percentage2013 = 1.0168
```

```
historical = true;
[data2012, idxNaN2012] = cleaning(dataUsed2012,historical);
plotCleaningResult(dataUsed2012,idxNaN2012, '2012')
```




**Number of failed points with one or more missing measure**

```
rejected2012 = size(dataUsed2012,1) - size(data2012, 1)
```

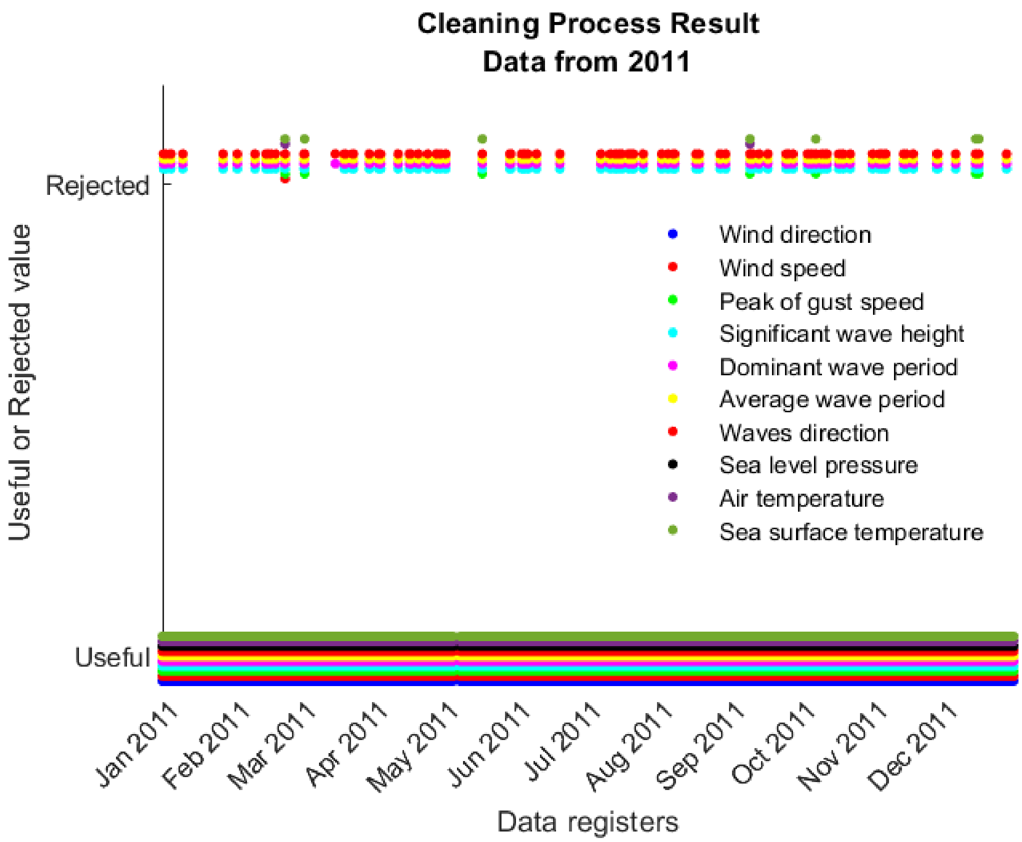
```
rejected2012 = 3283
```

**Percentage of these failed points respect to total registers**

```
percentage2012 = (size(dataUsed2012,1) - size(data2012, 1))/size(dataUsed2012,1) * 100
```

```
percentage2012 = 37.4216
```

```
historical = true;
[data2011, idxNaN2011] = cleaning(dataUsed2011,historical);
plotCleaningResult(dataUsed2011,idxNaN2011, '2011')
```




**Number of failed points with one or more missing measure**

```
rejected2011 = size(dataUsed2011,1) - size(data2011, 1)
```

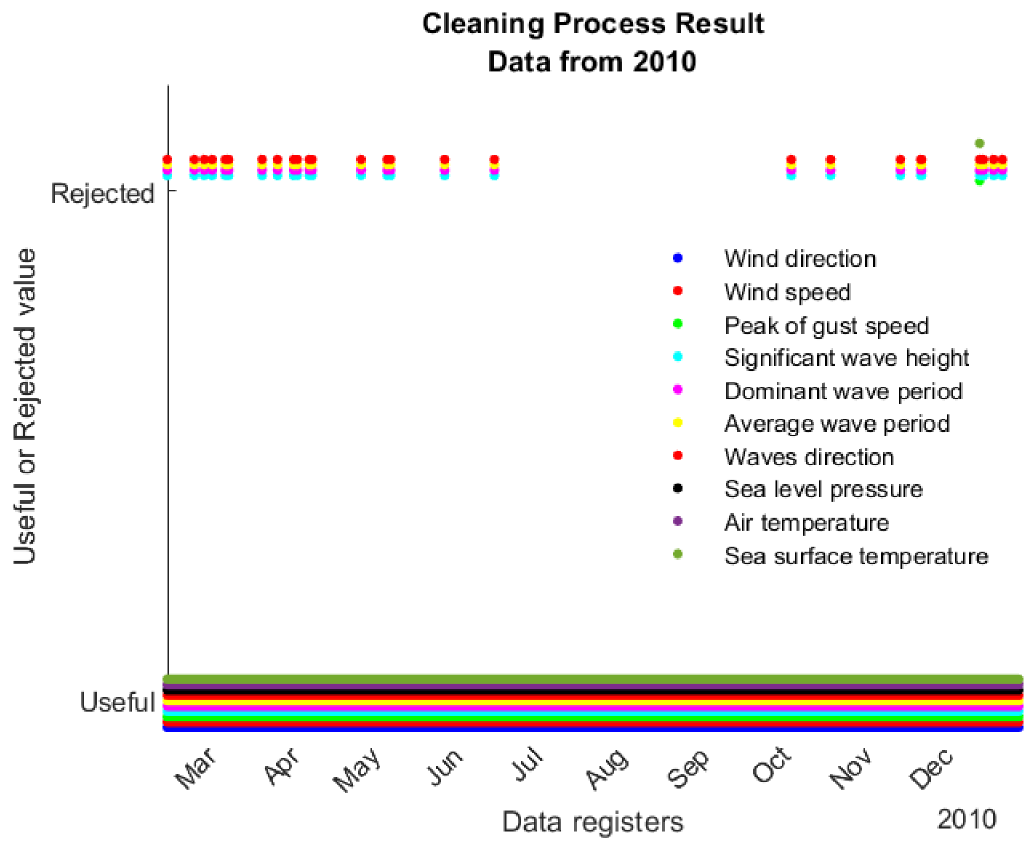
```
rejected2011 = 110
```

**Percentage of these failed points respect to total registers**

```
percentage2011= (size(dataUsed2011,1) - size(data2011, 1))/size(dataUsed2011,1) * 100
```

```
percentage2011 = 1.2666
```

```
historical = true;
[data2010, idxNaN2010] = cleaning(dataUsed2010,historical);
plotCleaningResult(dataUsed2010,idxNaN2010, '2010')
```




**Number of failed points with one or more missing measure**

```
rejected2010 = size(dataUsed2010,1) - size(data2010, 1)
```

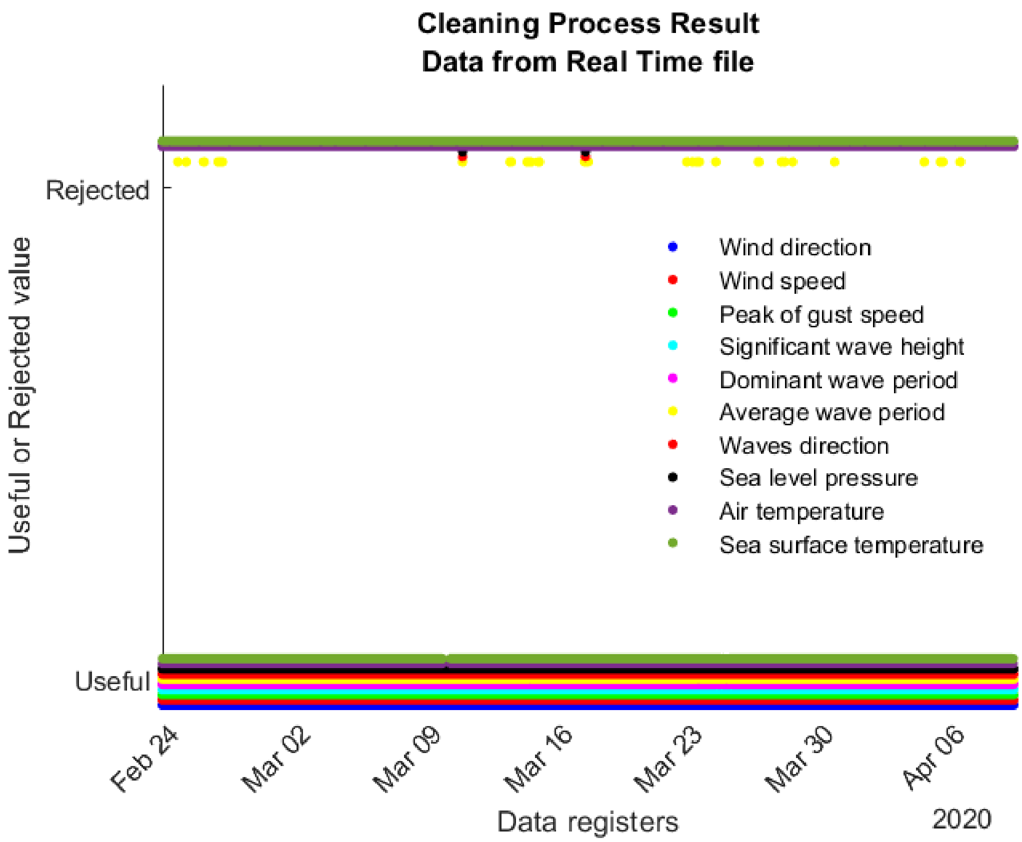
```
rejected2010 = 32
```

**Percentage of these failed points respect to total registers**

```
percentage2010 = (size(dataUsed2010,1) - size(data2010, 1))/size(dataUsed2010,1) * 100
```

```
percentage2010 = 0.4199
```

```
historical = false;
[dataRealTime, idxNaNReal] = cleaning(dataUsedRealTime, historical);
dataUsedRealTime = sortrows(dataUsedRealTime,'DATE','ascend');
plotCleaningResult(dataUsedRealTime,idxNaNReal, 'Real Time file')
```




**Number of failed points with one or more missing measure**

```
rejectedRT = size(dataUsedRealTime,1) - size(dataRealTime, 1)
```

```
rejectedRT = 5425
```

**Percentage of these failed points respect to total registers**

```
percentageRT = (size(dataUsedRealTime,1) - size(dataRealTime, 1))/size(dataUsedRealTime,1) * 10
```

```
percentageRT = 83.9264
```

```
rowData = size(dataUsed2019,1) +  size(dataUsed2018,1) +  size(dataUsed2017,1) + size(dataUsed2
cleanData = size(data2019,1) +  size(data2018,1) +  size(data2017,1) + size(data2016,1) +  size
passed = cleanData/rowData * 100
```

```
passed = 70.5407
```

**70.5407 % of data collected for training have passed the data quality assurance testing**

## Outliers detection

We will apply both boxplot and LDFO techniques to detect outliers and delete them rows with outliers in one or more variables

(in a separate Matlab file)

Performing data analysis for wind farm siting. Explore and discover patterns in wind and waves data wil conduce us to understand variables distribution and see what data can tell to us

# Univariate analysis

## Visualize data

Now that we have completed the review of data and we have deleted rows with missing measures or outliers on features selected, we are going to investigadte the data and obssserve the wind and waves characteristics of the site.

We will use for this purpose:

- Time series of each feature
- Summary stadistics
- Wind rose plots
- More detailed look into specifics

We want to observe the evolution of data for a complete year so we are going to choose 2018 because it's the bigger data set with less missing measures founded after cleaning process

Before developing the exploratory univariate analysis and checking data distribution, let's add again the variable DATA to dataset free of missing values and outliers

```
data2018 = data10_Real_clean(data10_Real_clean.YY == 2018,:);
data2018.DATE = datetime(data2018.YY, data2018.MM, data2018.DD, data2018.hh, data2018.mm,0)
```

data2018 = 8587×16 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2018 | 1 | 1 | 0 | 50 | 36 | 0.8000 | 1.4000 | 0.8500 |
| 2 | 2018 | 1 | 1 | 1 | 50 | 17 | 0.8000 | 1.1000 | 0.9200 |
| 3 | 2018 | 1 | 1 | 2 | 50 | 354 | 0.5000 | 0.9000 | 0.8700 |
| 4 | 2018 | 1 | 1 | 3 | 50 | 23 | 1.2000 | 1.6000 | 0.9200 |
| 5 | 2018 | 1 | 1 | 4 | 50 | 11 | 1.1000 | 1.3000 | 0.8500 |
| 6 | 2018 | 1 | 1 | 5 | 50 | 325 | 1.1000 | 1.6000 | 0.9000 |
| 7 | 2018 | 1 | 1 | 6 | 50 | 299 | 1.5000 | 1.8000 | 0.8000 |
| 8 | 2018 | 1 | 1 | 7 | 50 | 311 | 2.6000 | 3.1000 | 0.8100 |
| 9 | 2018 | 1 | 1 | 8 | 50 | 329 | 3.0000 | 3.6000 | 0.7400 |
| 10 | 2018 | 1 | 1 | 9 | 50 | 338 | 2.6000 | 3.2000 | 0.7500 |
| 11 | 2018 | 1 | 1 | 10 | 50 | 358 | 3.3000 | 3.9000 | 0.7200 |
| 12 | 2018 | 1 | 1 | 11 | 50 | 350 | 3.6000 | 4.2000 | 0.7000 |

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 13 | 2018 | 1 | 1 | 12 | 50 | 344 | 4.0000 | 4.8000 | 0.6800 |
| 14 | 2018 | 1 | 1 | 13 | 50 | 354 | 4.9000 | 5.7000 | 0.6700 |

⋮

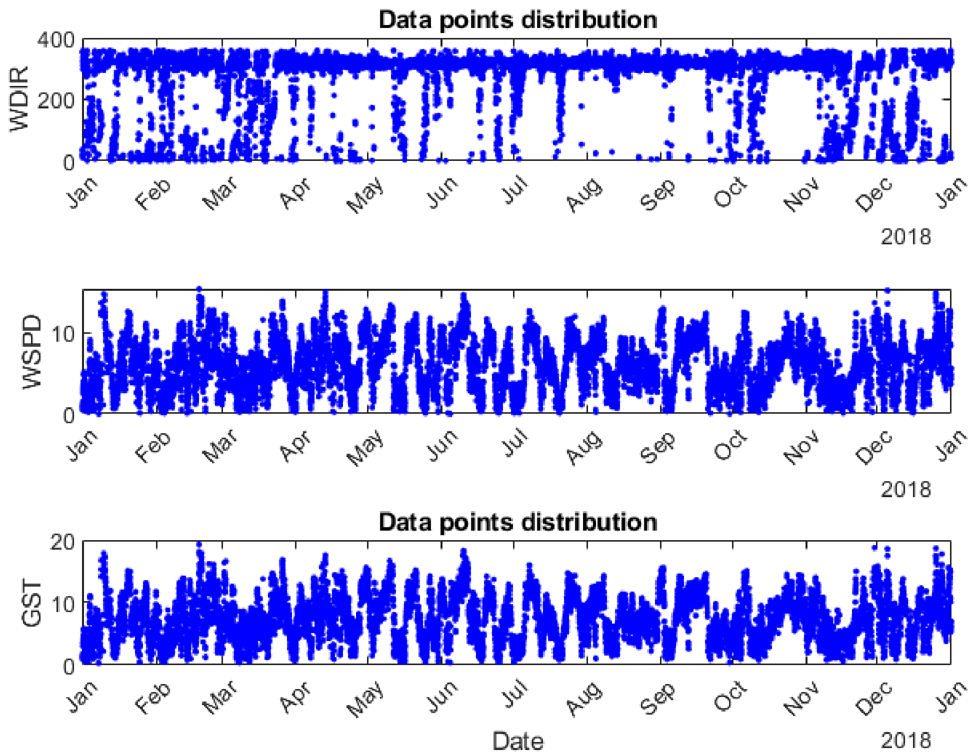
## Time series

Let's plot the same time -series for data saved at the beginning of the project and clean data to ensure that cleaning process haven't change data distribution and we haven't make any mistake that could have affected data

**Wind features**

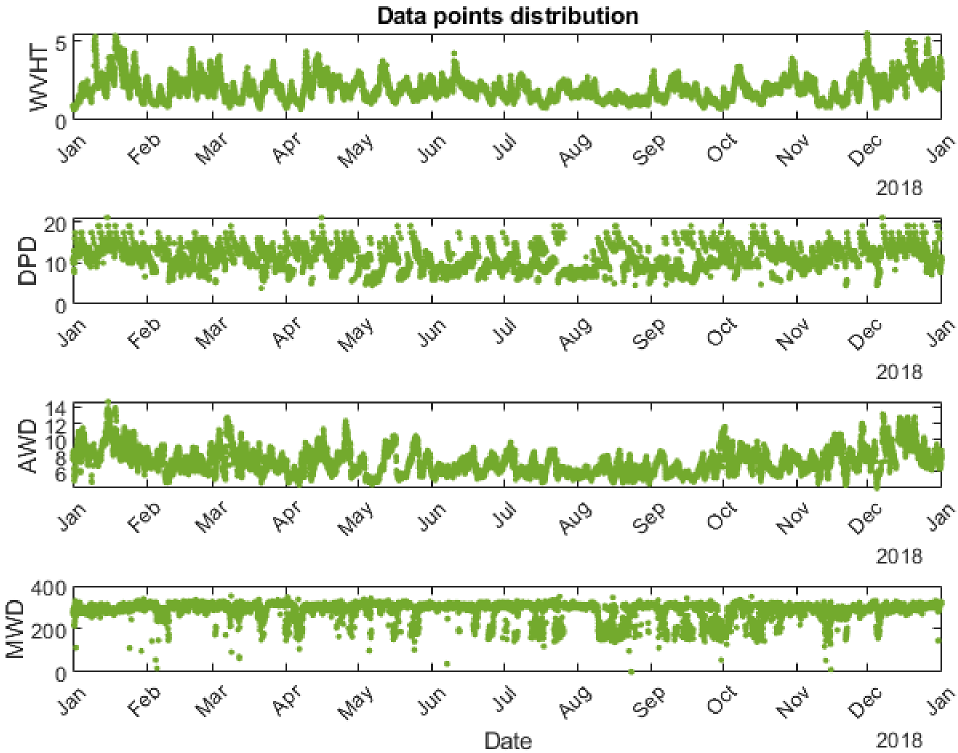
**Wind features**

```
subplot(3,1,1)
plot(data2018.DATE, data2018.WDIR, '.','MarkerSize',7,"Color",'blue')
ylabel('WDIR')
title('Data points distribution')
xtickangle(45)

subplot(3,1,2)
plot(data2018.DATE, data2018.WSPD, '.','MarkerSize',7,"Color",'blue')
ylabel('WSPD')
xtickangle(45)

subplot(3,1,3)
plot(data2018.DATE, data2018.GST, '.','MarkerSize',7,"Color",'blue');
xlabel('Date')
ylabel('GST')
title('Data points distribution')
xtickangle(45)
```



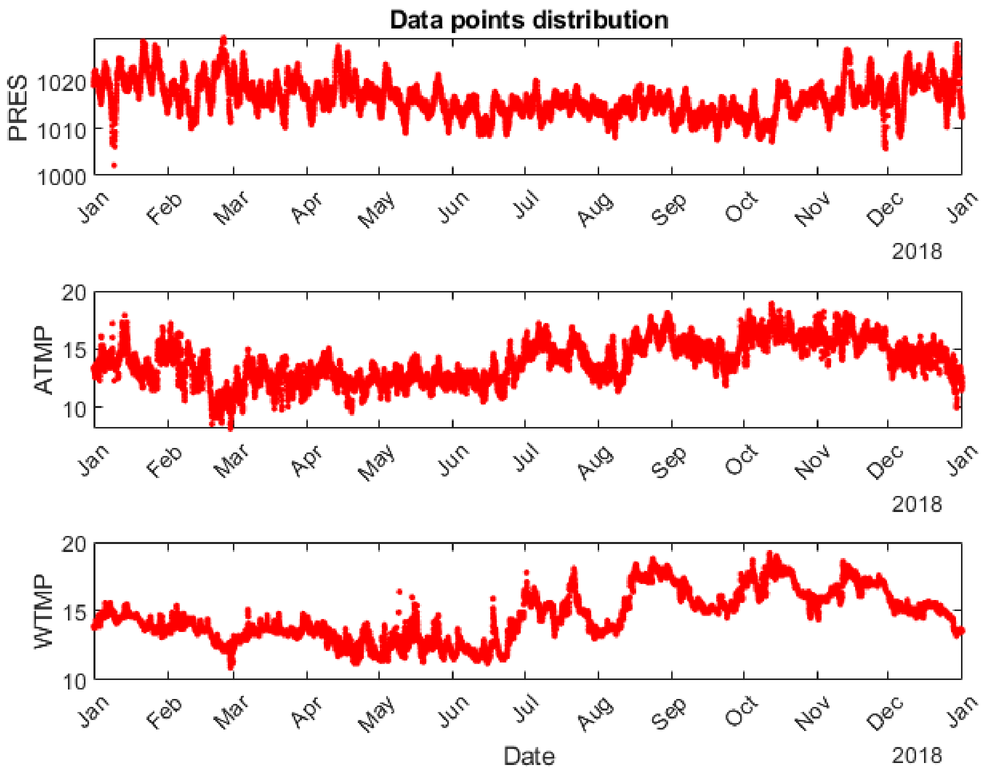

#### Waves features

```
clf
subplot(4,1,1)
plot(data2018.DATE, data2018.WVHT, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
ylabel('WVHT')
title('Data points distribution')
xtickangle(45)

subplot(4,1,2)
plot(data2018.DATE, data2018.DPD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
ylabel('DPD')
xtickangle(45)

subplot(4,1,3)
plot(data2018.DATE, data2018.APD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
ylabel('AWD')
xtickangle(45)

subplot(4,1,4)
plot(data2018.DATE, data2018.MWD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
xlabel('Date')
ylabel('MWD')
xtickangle(45)
```




## Pression and Temperatures

```
clf
subplot(3,1,1)
plot(data2018.DATE, data2018.PRES, '.','MarkerSize',7,"Color",'red')
ylabel('PRES')
title('Data points distribution')
xtickangle(45)

subplot(3,1,2)
plot(data2018.DATE, data2018.ATMP, '.','MarkerSize',7,"Color",'red')
ylabel('ATMP')
xtickangle(45)

subplot(3,1,3)
plot(data2018.DATE, data2018.WTMP, '.','MarkerSize',7,"Color",'red')
xlabel('Date')
ylabel('WTMP')
xtickangle(45)
```



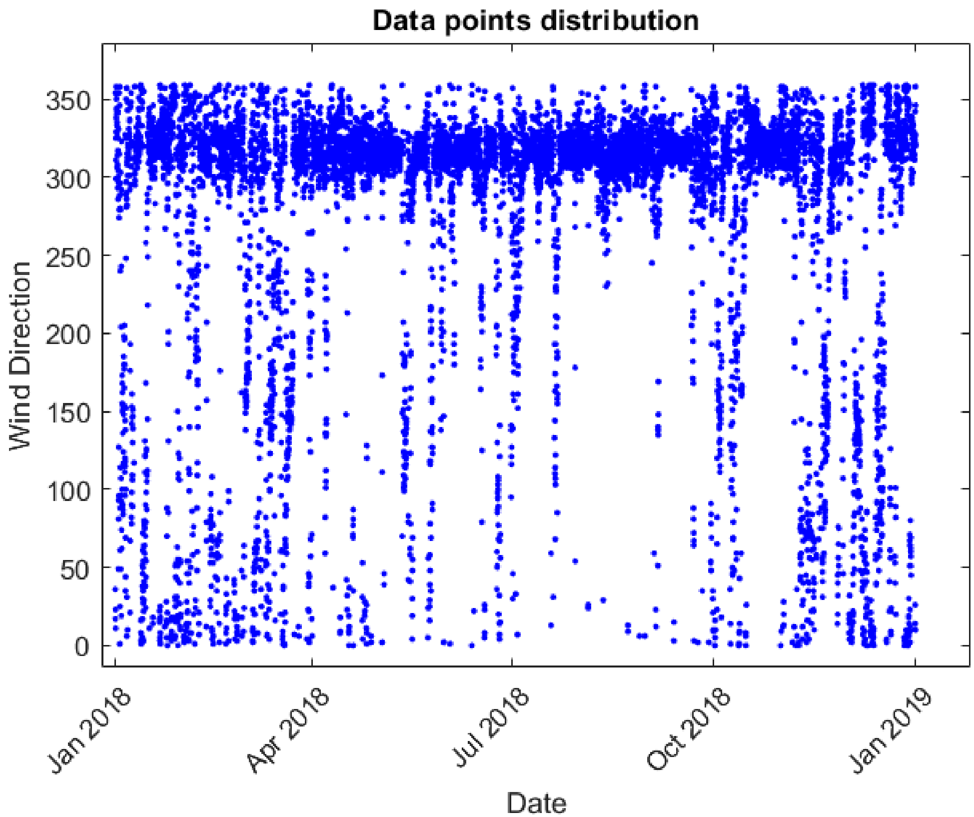

**Now , let's analize in more detail each feature by plotting each time-serie of each feature in separated figures.**

## Features time-series in separated plots

### Wind Features

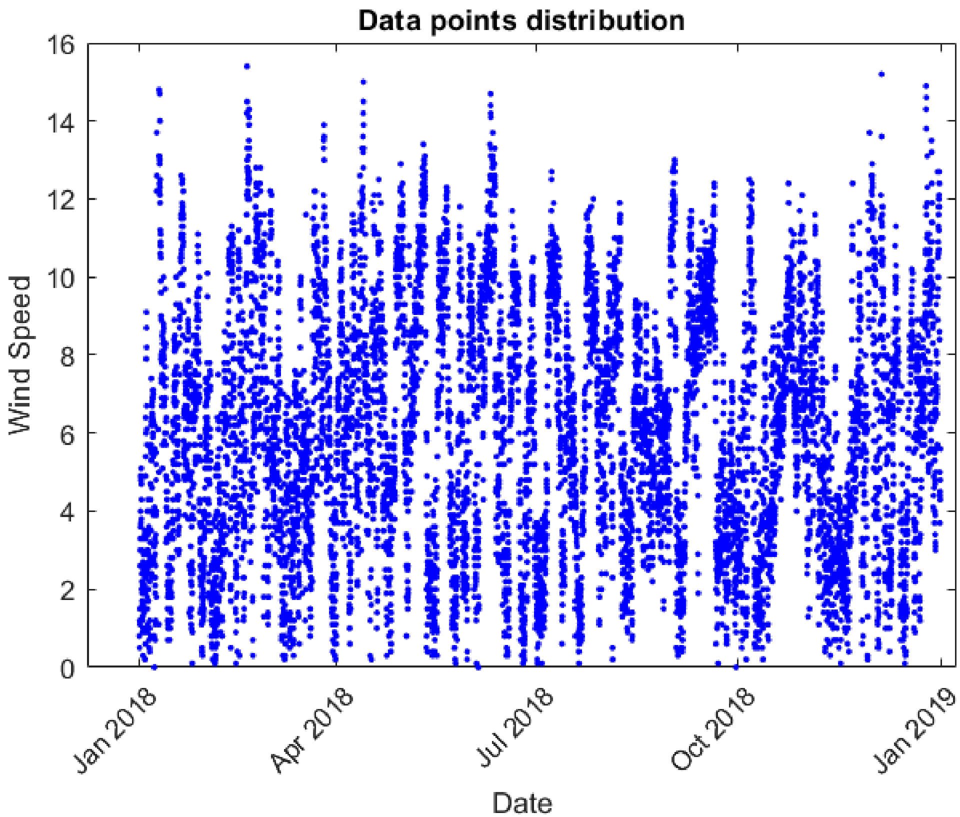
Wind direction  - WDIR

```
clf
plot(data2018.DATE, data2018.WDIR, '.','MarkerSize',7,"Color",'blue')
xlabel('Date')
ylabel('Wind Direction')
title('Data points distribution')
xtickangle(45)
xlim([datetime(2017,12,25,19,25,0)...
      datetime(2019,1,25,19,25,0)])
ylim([-14 386])
```



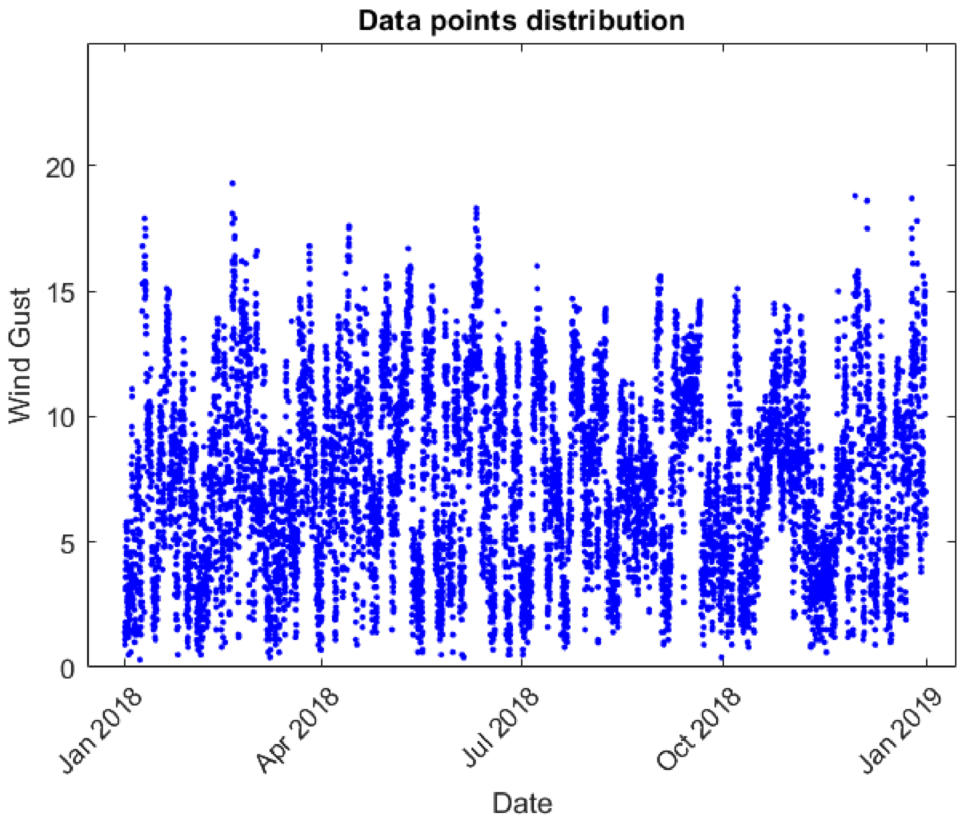

Wind speed - WSPD

```
plot(data2018.DATE, data2018.WSPD, '.','MarkerSize',7,"Color",'blue')
xlabel('Date')
ylabel('Wind Speed')
title('Data points distribution')
xtickangle(45)

xlim([datetime(2017,12,8,7,1,39)...
      datetime(2019,1,8,7,1,39)])
ylim([0 16.0])
```




Peak 5 or 8 ssecond of gust speed - GST

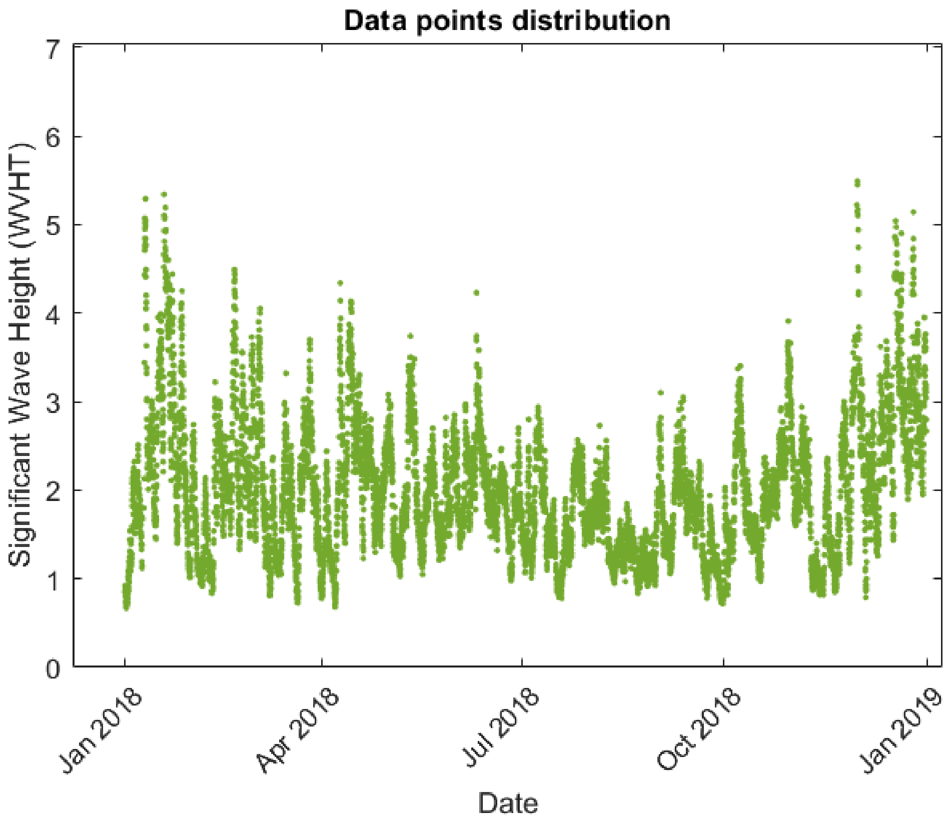
```
plot(data2018.DATE, data2018.GST, '.','MarkerSize',7,"Color",'blue');
xlabel('Date')
ylabel('Wind Gust')
title('Data points distribution')
xtickangle(45)

xlim([datetime(2017,12,14,20,33,8)...
      datetime(2019,1,14,20,33,8)])
ylim([0 24.9])
```




**Wave features**

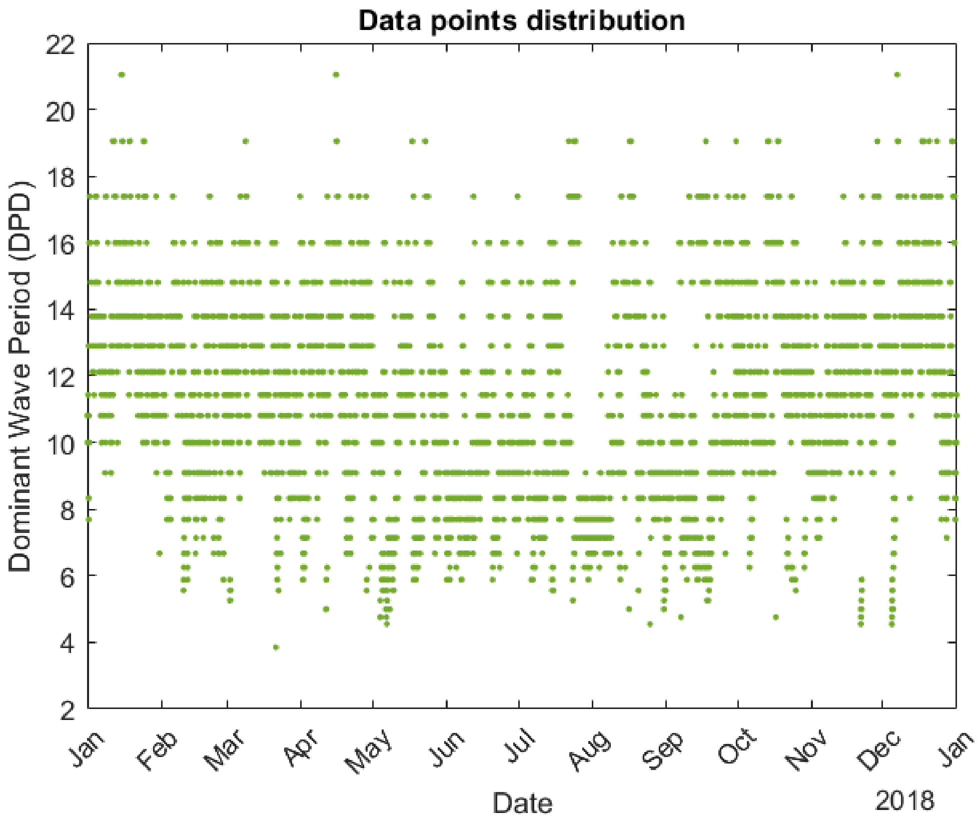
Significant Wave Height - WVHT

```
plot(data2018.DATE, data2018.WVHT, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
xlabel('Date')
ylabel('Significant Wave Height (WVHT)')
title('Data points distribution')
xtickangle(45)

xlim([datetime(2017,12,8,7,1,39)...
      datetime(2019,1,8,7,1,39)])
ylim([0 7.05])
```



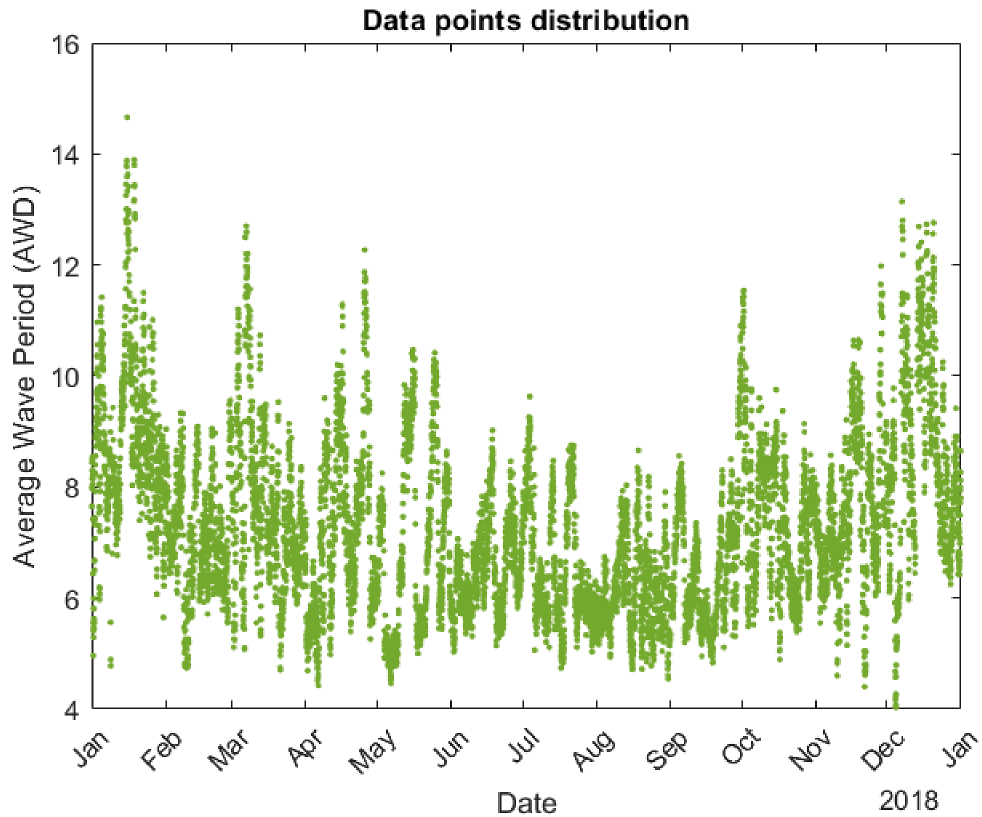

**Dominant Wave Period - DPD**

```
plot(data2018.DATE, data2018.DPD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
xlabel('Date')
ylabel('Dominant Wave Period (DPD)')
title('Data points distribution')
xtickangle(45)
```




**Average wave period - APD**

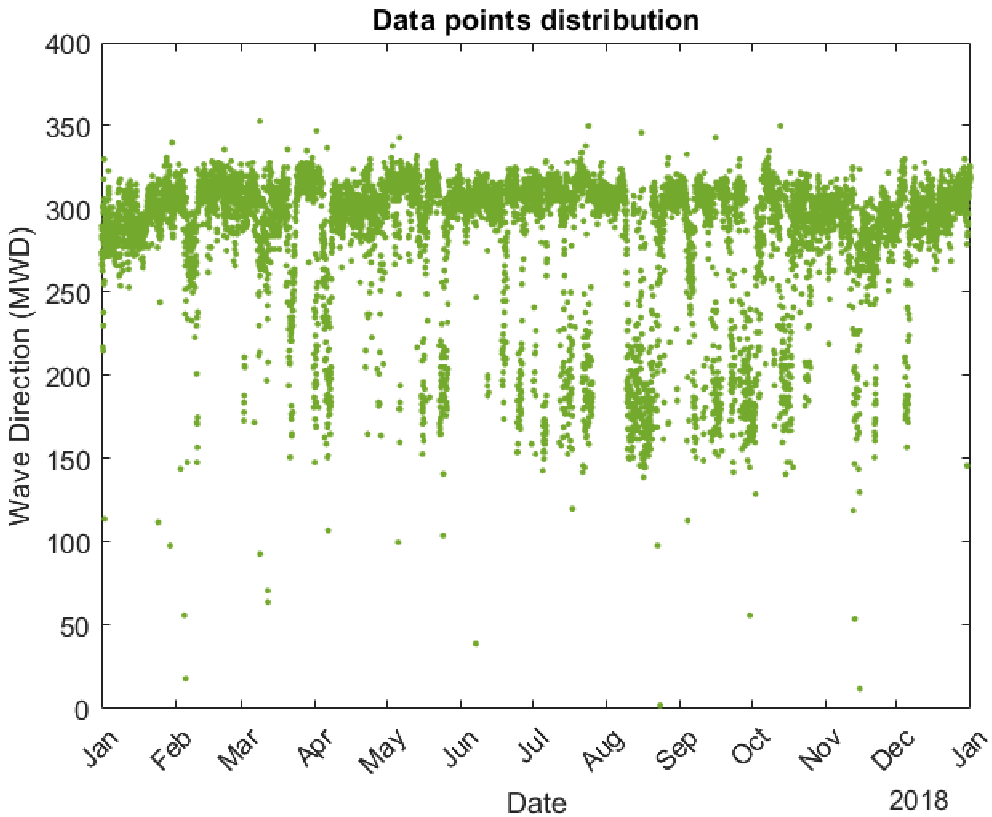
```
plot(data2018.DATE, data2018.APD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
xlabel('Date')
ylabel('Average Wave Period (AWD)')
title('Data points distribution')
xtickangle(45)
```




Obviously its's similar to dominant wave period values distribution but more concentrated in middle values between minimun (aprox. 5 ) and maximum

#### Wave direction - MWD

```
plot(data2018.DATE, data2018.MWD, '.','MarkerSize',7,"Color",[0.4660 0.6740 0.1880])
xlabel('Date')
ylabel('Wave Direction (MWD)')
title('Data points distribution')
xtickangle(45)
```

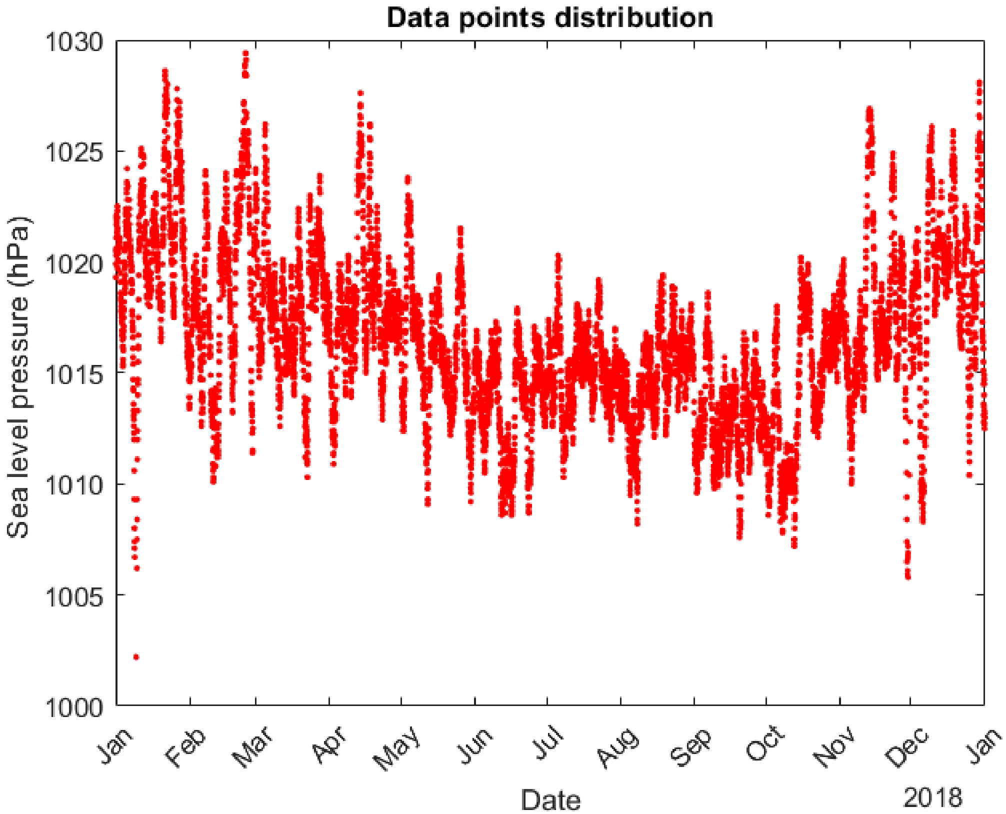



It is so similar to the wind direction graphic representation

**Pressure and Temperatures features**

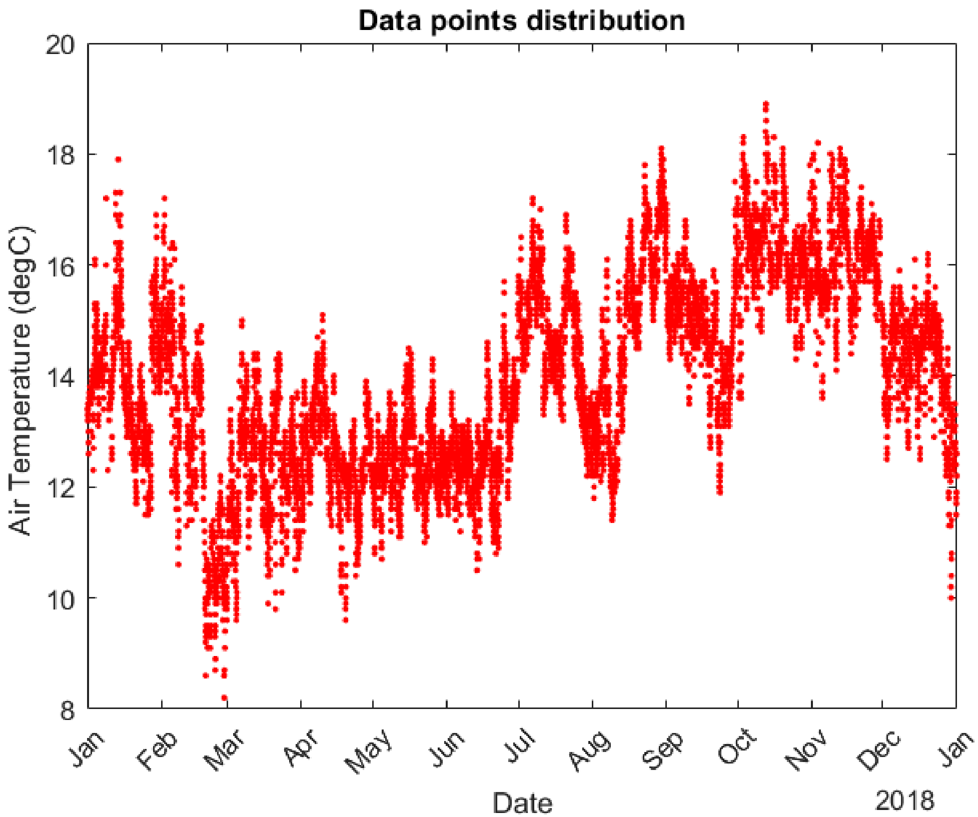
**Sea Level Pressure - PRES**

```
plot(data2018.DATE, data2018.PRES, '.','MarkerSize',7,"Color",'red')
xlabel('Date')
ylabel('Sea level pressure (hPa) ')
title('Data points distribution')
xtickangle(45)
```




**Air Temperature - ATMP**

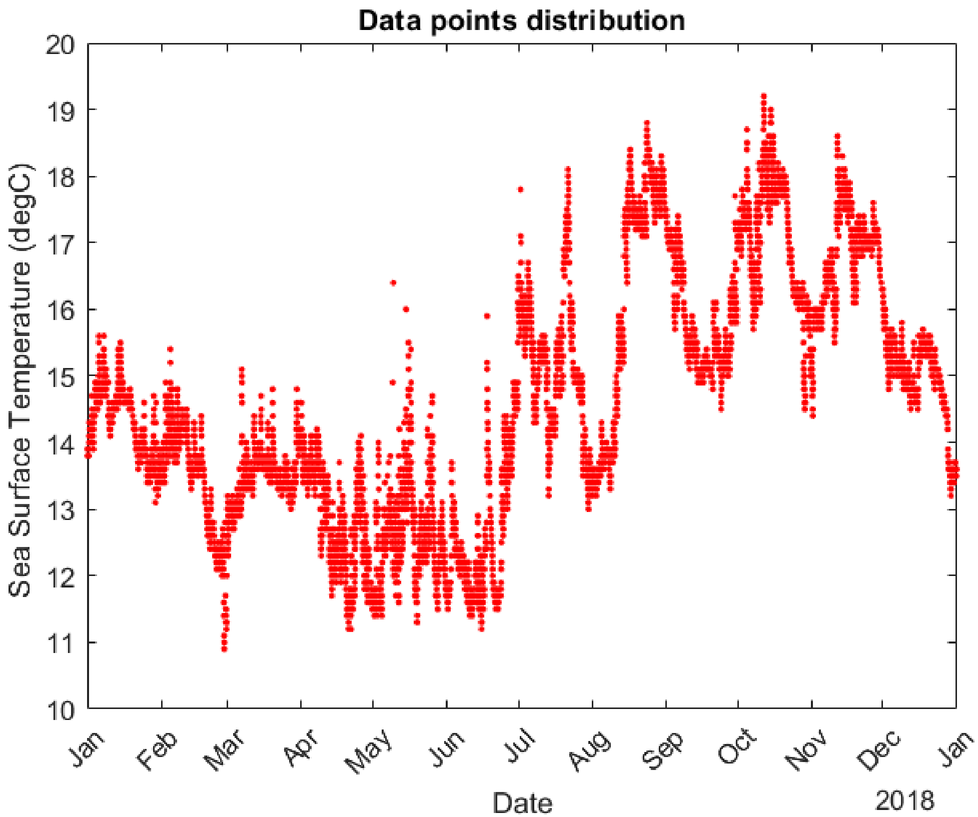
```
plot(data2018.DATE, data2018.ATMP, '.','MarkerSize',7,"Color",'red')
xlabel('Date')
ylabel('Air Temperature (degC) ')
title('Data points distribution')
xtickangle(45)
```




If we would plot a curve fixing represented points, we could separate figure in two subfigures . The first one from January to July where curve is convex and the second one from july to December where curve is concave. Its seems according to temperatures that we can distinguish two season periods. These observation could be of interest in order to divide data set for training if we considered to be beneficial to improve model accuracy.

**Sea Surface Temperature - WTMP**

```
plot(data2018.DATE, data2018.WTMP, '.','MarkerSize',7,"Color",'red')
xlabel('Date')
ylabel('Sea Surface Temperature (degC) ')
title('Data points distribution')
xtickangle(45)
```




Again if we consider the imaginary curve we can observe the same pattern in sea surface temperature as in air temperature graphic. The main difference is the range of values , which are higher on sea surface than in air.

## Statistical Analysis

Calculating descriptive basic statistics to overview and understand the winds and waves characteristics of the site. This will include:

- Measures of Central Tendency ( save in variables like <featureName>_cm
- Measures of Spread  (<featureName>_sm)
- Measures of Shape (<featureName>_spm)

Wind direction and waves direction : with circstat toolbox

### Wind speed

```
wspd_cm = centralMeasures(data2018.WSPD,2018)
```

```
wspd_cm = struct with fields:
       year: 2018
       mean: 6.0450
     median: 6
       mode: 6.4000
```

```
wspd_sm = spreadMeasures(data2018.WSPD,2018)
```

```
wspd_sm = struct with fields:
```

```
         year: 2018
        range: 15.4000
          std: 3.1328
     variance: 9.8145
    quantiles: [0 3.4000 6 8.5000 15.4000]
```

```
wspd_spm = shapeMeasures(data2018.WSPD,2018)
```

```
wspd_spm = struct with fields:
             year: 2018
    centralMoment: 5.1228
          maximun: 15.4000
          minimun: 0
```

## Peak of Gust Speed

```
gst_cm = centralMeasures(data2018.GST,2018)
```

```
gst_cm = struct with fields:
      year: 2018
      mean: 7.4615
    median: 7.3000
      mode: 7.9000
```

```
gst_sm = spreadMeasures(data2018.GST, 2018)
```

```
gst_sm = struct with fields:
         year: 2018
        range: 19
          std: 3.6458
     variance: 13.2916
    quantiles: [0.3000 4.5000 7.3000 10.2000 19.3000]
```

```
gst_spm = shapeMeasures(data2018.GST, 2018)
```

```
gst_spm = struct with fields:
             year: 2018
    centralMoment: 11.4506
          maximun: 19.3000
          minimun: 0.3000
```

## Significant Wave Height

```
wvht_cm =  centralMeasures(data2018.WVHT, 2018)
```

```
wvht_cm = struct with fields:
      year: 2018
      mean: 1.9779
    median: 1.8700
      mode: 1.6700
```

```
wvht_sm = spreadMeasures(data2018.WVHT, 2018)
```

```
wvht_sm = struct with fields:
         year: 2018
        range: 4.8200
          std: 0.7523
     variance: 0.5660
    quantiles: [0.6700 1.4200 1.8700 2.3900 5.4900]
```

```
wvht_spm = shapeMeasures(data2018.WVHT, 2018)
```

```
wvht_spm = struct with fields:
             year: 2018
    centralMoment: 0.4237
          maximun: 5.4900
          minimun: 0.6700
```

## Dominant wave period

```
dpd_cm =  centralMeasures(data2018.DPD, 2018)
```

```
dpd_cm = struct with fields:
      year: 2018
      mean: 11.5498
    median: 11.4300
      mode: 13.7900
```

```
dpd_sm = spreadMeasures(data2018.DPD, 2018)
```

```
dpd_sm = struct with fields:
         year: 2018
        range: 17.2000
          std: 3.0701
     variance: 9.4254
    quantiles: [3.8500 9.0900 11.4300 13.7900 21.0500]
```

```
dpd_spm = shapeMeasures(data2018.DPD, 2018)
```

```
dpd_spm = struct with fields:
             year: 2018
    centralMoment: 6.0984
          maximun: 21.0500
          minimun: 3.8500
```

## Average Wave Period

```
apd_cm =  centralMeasures(data2018.APD, 2018)
```

```
apd_cm = struct with fields:
      year: 2018
      mean: 7.2450
    median: 6.9900
      mode: 5.8000
```

```
apd_sm = spreadMeasures(data2018.APD, 2018)
```

```
apd_sm = struct with fields:
         year: 2018
        range: 10.6300
          std: 1.5534
     variance: 2.4130
    quantiles: [4.0300 6.0300 6.9900 8.1700 14.6600]
```

```
apd_spm = shapeMeasures(data2018.APD, 2018)
```

```
apd_spm = struct with fields:
             year: 2018
    centralMoment: 3.3068
```

```
maximun: 14.6600
minimun: 4.0300
```

## Waves Direction at DPD

```
mwd_cm =  centralMeasures(data2018.MWD, 2018)
```

```
mwd_cm = struct with fields:
      year: 2018
      mean: 284.9108
    median: 301
      mode: 311
```

```
mwd_sm = spreadMeasures(data2018.MWD, 2018)
```

```
mwd_sm = struct with fields:
         year: 2018
        range: 351
          std: 43.3946
     variance: 1.8831e+03
    quantiles: [2 283 301 311 353]
```

```
mwd_spm = shapeMeasures(data2018.MWD, 2018)
```

```
mwd_spm = struct with fields:
             year: 2018
    centralMoment: -1.5176e+05
          maximun: 353
          minimun: 2
```

## Sea Level Pressure

```
pres_cm =  centralMeasures(data2018.PRES, 2018)
```

```
pres_cm = struct with fields:
      year: 2018
      mean: 1.0165e+03
    median: 1.0161e+03
      mode: 1.0159e+03
```

```
pres_sm = spreadMeasures(data2018.PRES, 2018)
```

```
pres_sm = struct with fields:
         year: 2018
        range: 27.2000
          std: 3.7864
     variance: 14.3367
    quantiles: [1.0022e+03 1.0139e+03 1.0161e+03 1.0187e+03 1.0294e+03]
```

```
pres_spm = shapeMeasures(data2018.PRES, 2018)
```

```
pres_spm = struct with fields:
             year: 2018
    centralMoment: 22.2007
          maximun: 1.0294e+03
          minimun: 1.0022e+03
```

## Air temperature

```
atmp_cm =  centralMeasures(data2018.ATMP, 2018)
```

```
atmp_cm = struct with fields:
      year: 2018
      mean: 13.9682
    median: 13.9000
      mode: 12.4000
```

```
atmp_sm = spreadMeasures(data2018.ATMP, 2018)
```

```
atmp_sm = struct with fields:
         year: 2018
        range: 10.7000
          std: 1.7494
     variance: 3.0603
    quantiles: [8.2000 12.6000 13.9000 15.3000 18.9000]
```

```
atmp_spm = shapeMeasures(data2018.ATMP, 2018)
```

```
atmp_spm = struct with fields:
             year: 2018
    centralMoment: 0.2580
          maximun: 18.9000
          minimun: 8.2000
```

**Sea surface temperature**

```
wtmp_cm =  centralMeasures(data2018.WTMP, 2018)
```

```
wtmp_cm = struct with fields:
      year: 2018
      mean: 14.5809
    median: 14.4000
      mode: 13.6000
```

```
wtmp_sm = spreadMeasures(data2018.WTMP, 2018)
```

```
wtmp_sm = struct with fields:
         year: 2018
        range: 8.3000
          std: 1.7996
     variance: 3.2386
    quantiles: [10.9000 13.3000 14.4000 15.9000 19.2000]
```

```
wtmp_spm = shapeMeasures(data2018.WTMP, 2018)
```

```
wtmp_spm = struct with fields:
             year: 2018
    centralMoment: 1.6150
          maximun: 19.2000
          minimun: 10.9000
```

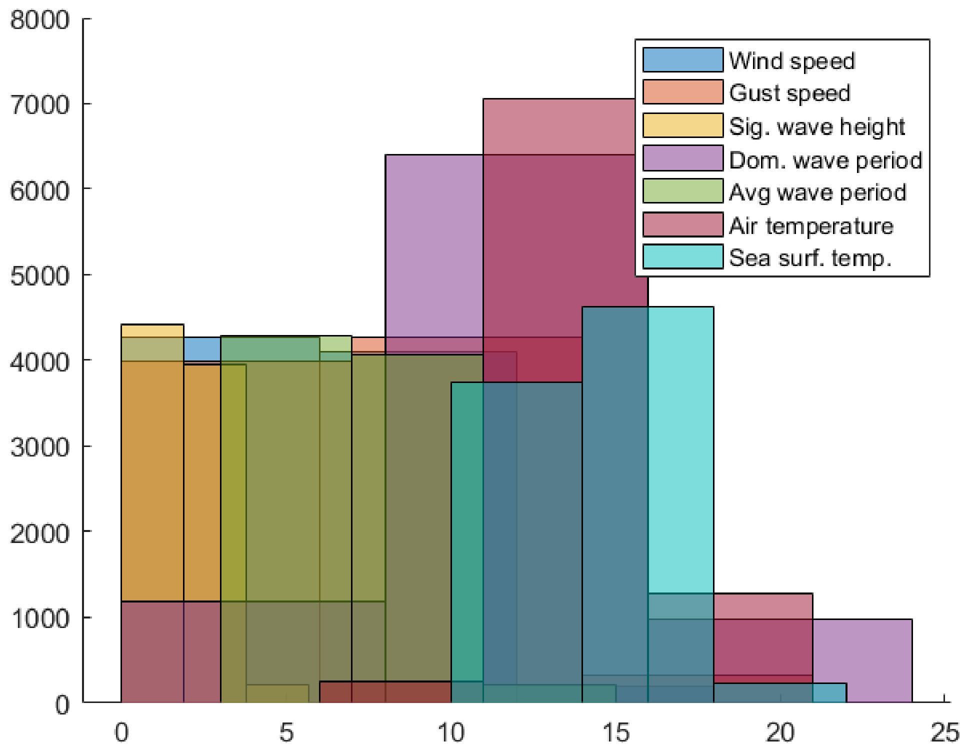
# Data Visualization

## Data Distribution approach

### Histogram with every feature distributions

```
clf
```

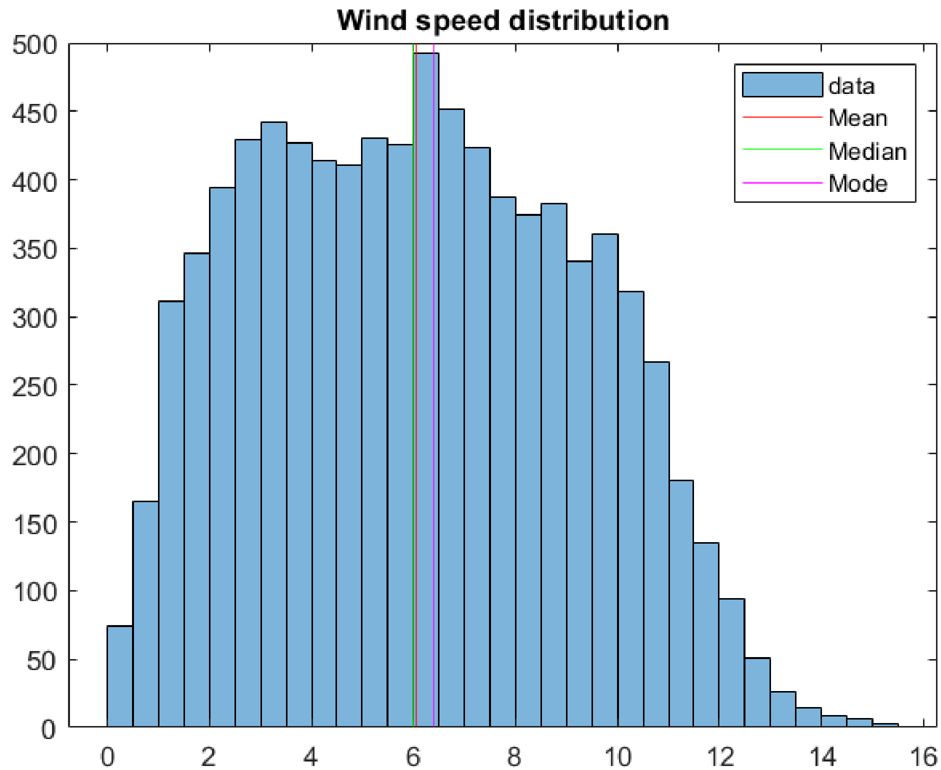
```
hold on
histogram(data2018.WSPD,3,"FaceAlpha",0.5,"Facecolor",[0, 0.4470, .7410])
histogram(data2018.GST,3,"FaceAlpha",0.5,"Facecolor",[.8500, .3250 .0980])
histogram(data2018.WVHT,3,"FaceAlpha",0.5,"FaceColor",[.9290 .6940 .1250])
histogram(data2018.DPD,3,"FaceAlpha",0.5,"FaceColor",[.4940 .1840 .5560])
histogram(data2018.APD,3,"FaceAlpha",0.5,"FaceColor",[.4660 .6740 .1880])
%histogram(data2018.PRES,5,"FaceAlpha",0.5,"FaceColor",[.3010 .7450 .9330])
histogram(data2018.ATMP,3,"FaceAlpha",0.5,"FaceColor",[.6350 .0780 .1840])
histogram(data2018.WTMP,3,"FaceAlpha",0.5,"FaceColor",[0 .75 .75])
legend("Wind speed", "Gust speed", "Sig. wave height", "Dom. wave period", "Avg wave period", .
    "Air temperature", "Sea surf. temp.")
```




### Wind speed - WSPD

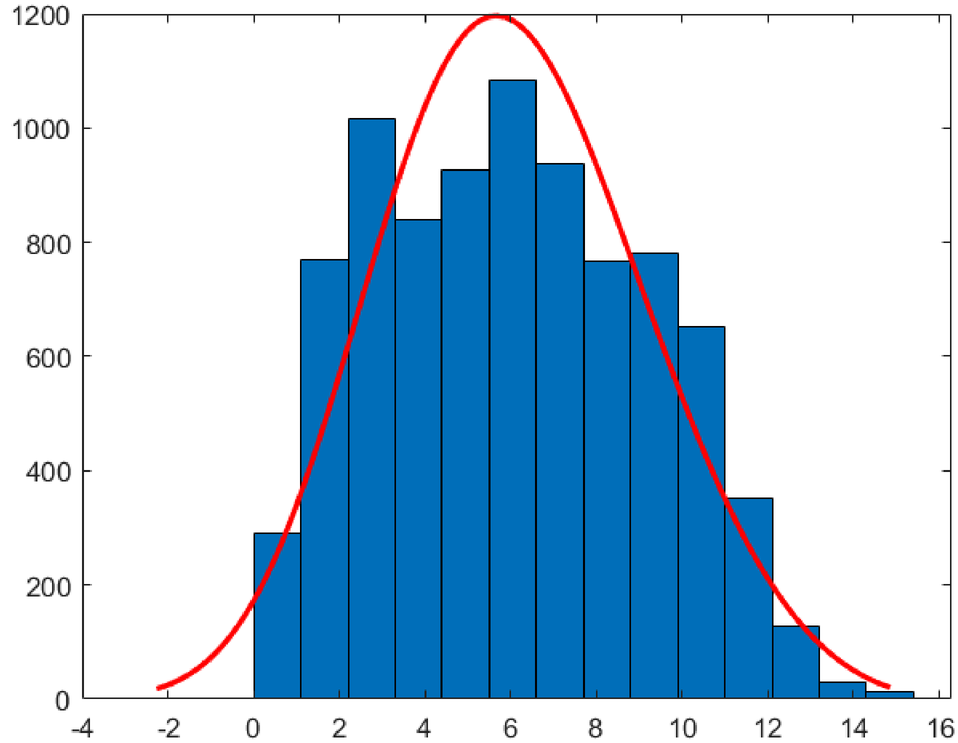
### Histogram with central and spread measures

```
figure1 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.WSPD, 'FaceAlpha',0.5);
hold on;
xline(wspd_cm.mean,'red');
xline(wspd_cm.median,'green');
xline(wspd_cm.mode,'magenta');
legend({'data','Mean','Median','Mode'})
title('Wind speed distribution');
hold off;
```

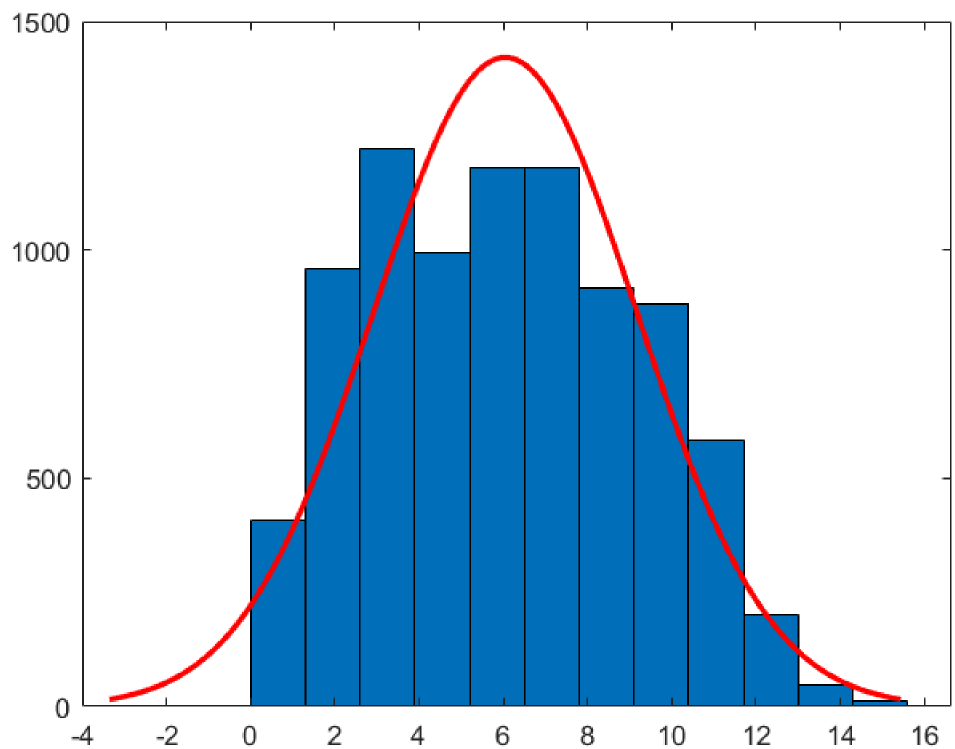



Median and mean are almost equal while mode is situated at right position. That means wind speed data is slightly biased to the left

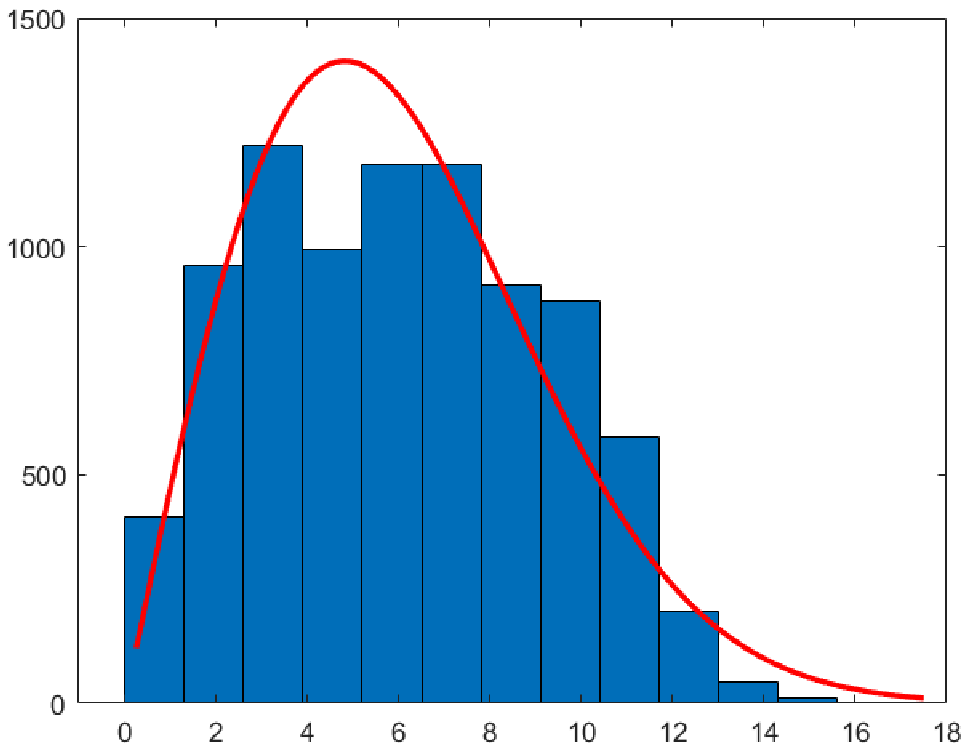
```
h2 = histfit(data2018.WSPD,14,'generalized extreme value');
```



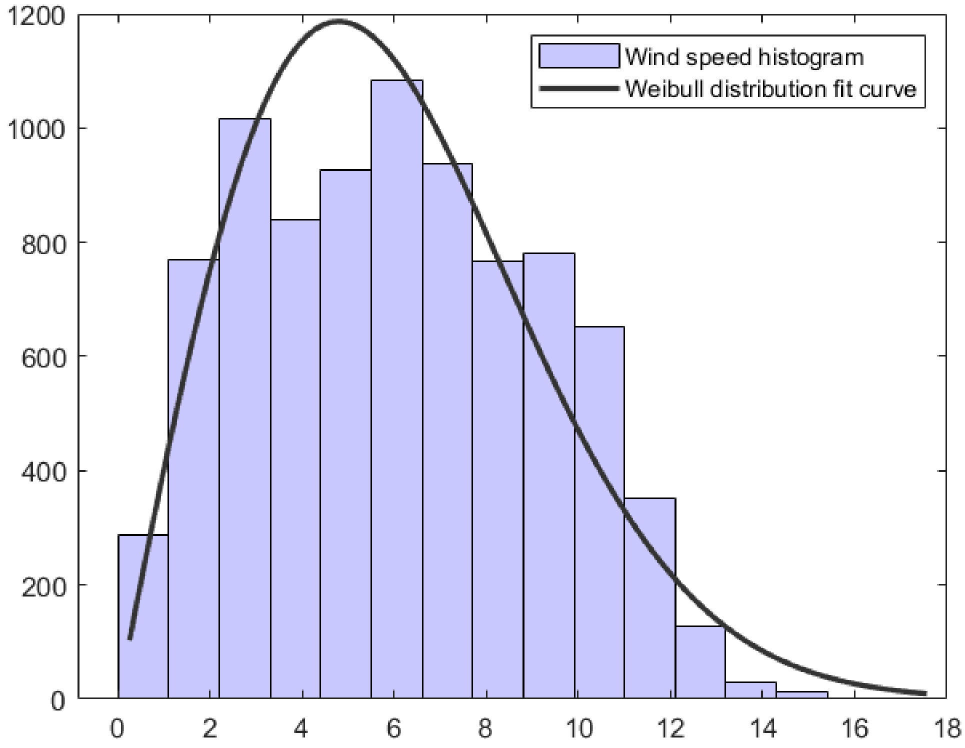

```
h3 = histfit(data2018.WSPD,12,'normal');
```



```
h4 = histfit(data2018.WSPD,12,'rayleigh');
```




```
clf
h5 = histfit(data2018.WSPD(data2018.WSPD > 0),14,'weibull');
h5(1).FaceColor = [.8 .8 1];
h5(2).Color = [.2 .2 .2];
h5(2).DisplayName = "Weibull distribution";
legend ("Wind speed histogram", "Weibull distribution fit curve");
```




For weibull distribution representation values equal to 0 must be excluded. There are 3 values 0 in wind speed feature:

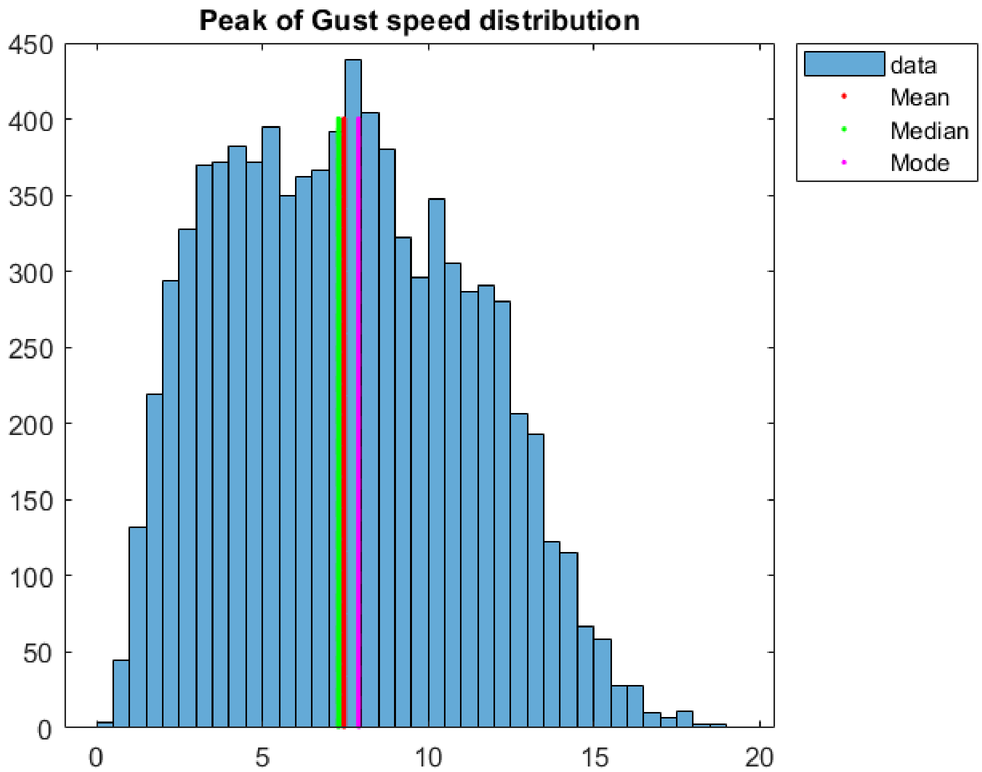
```
sum(data2018.WSPD == 0)
```

```
ans = 3
```

As we have observed by applying different statistics distribution to data (other distributions couln't be applied for data with zeros ) , the best that fit to wind speed data seems to be **rayleigh distribution and weibull distribution (**considering x >= 0 where x = wind speed variable).

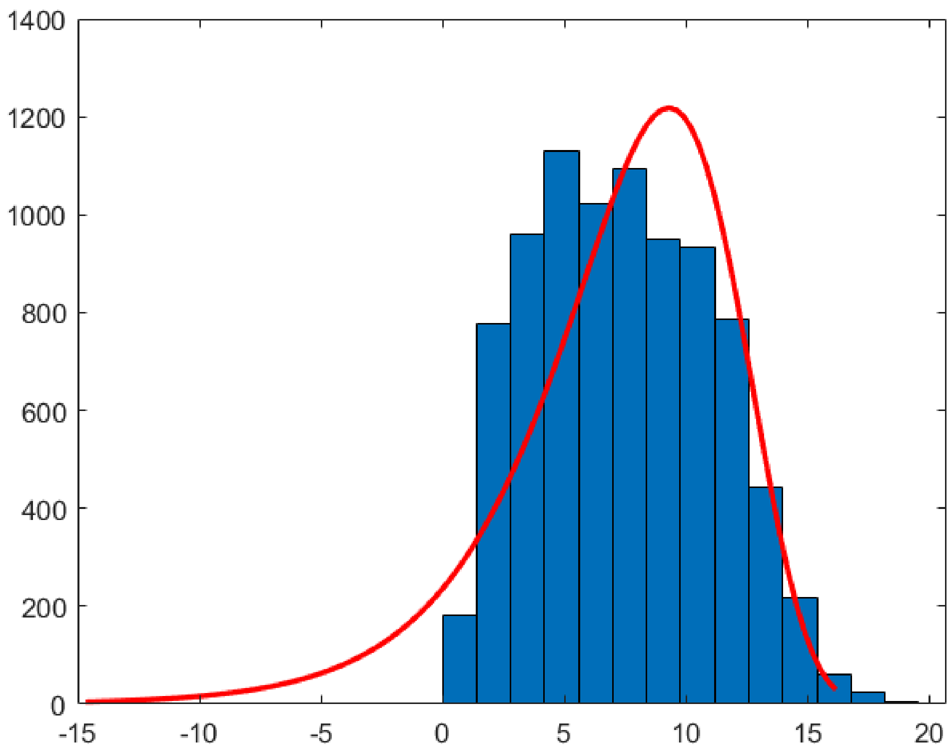
### Peak of Gust speed - GST

```
figure3 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.GST);
hold on;
scatter(repmat(gst_cm.mean,1,400), 1:1:400,'.','red');
scatter(repmat(gst_cm.median,1,400), 1:1:400,'.','green');
scatter(repmat(gst_cm.mode,1,400), 1:1:400,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Peak of Gust speed distribution');
```



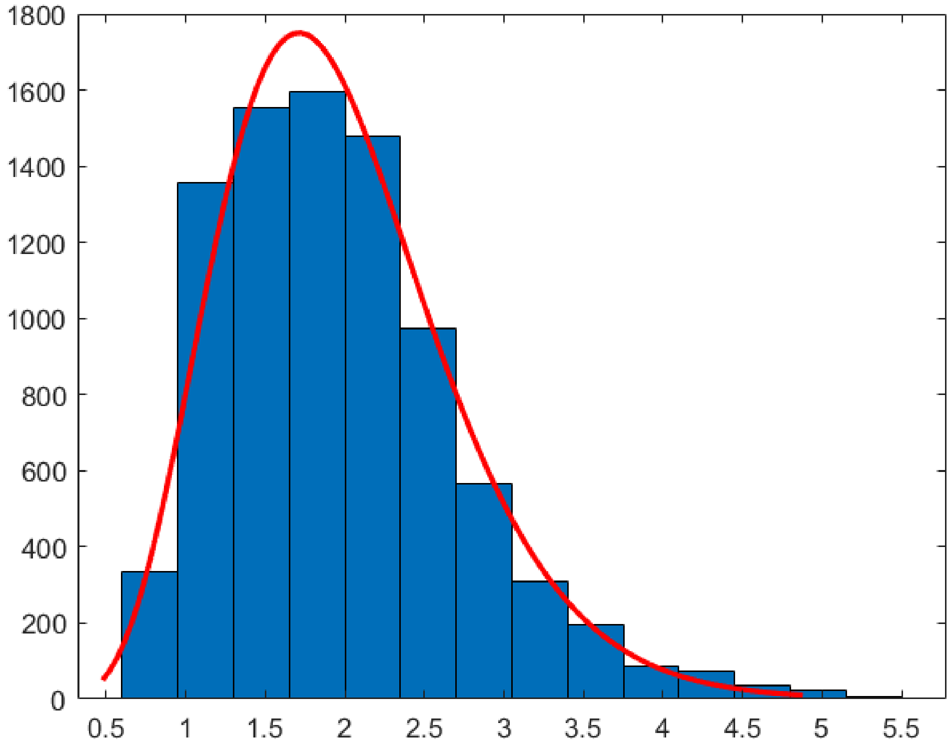

The distribution of Peak of Gust speed is almost **normal**

```
clf
h8 = histfit(data2018.GST,14, 'extreme value')
```



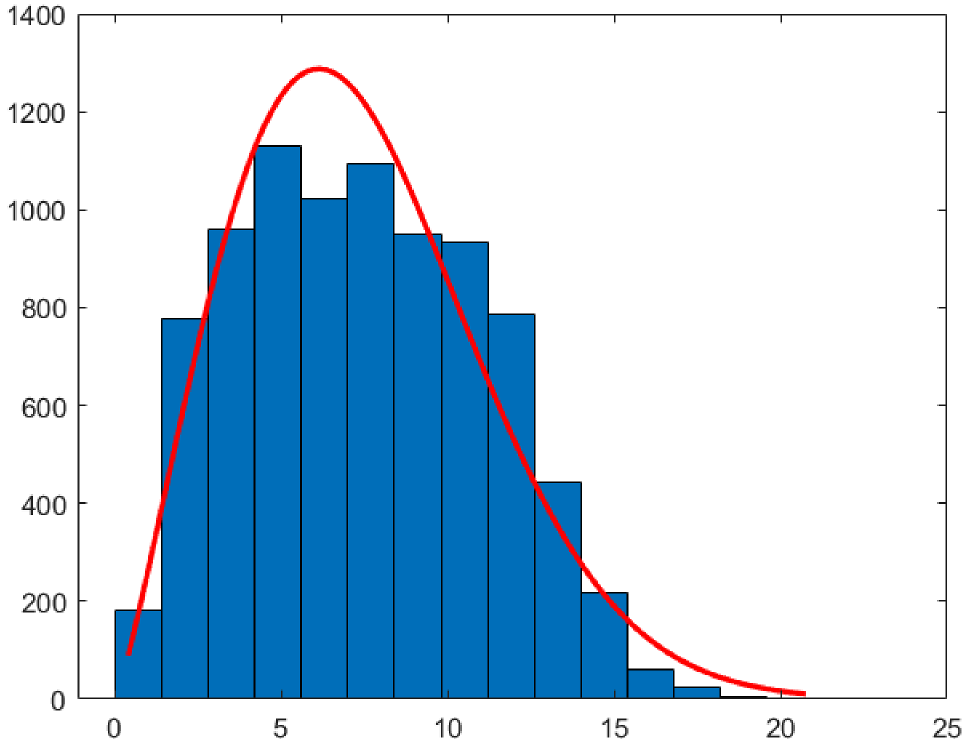

```
h8 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h9 = histfit(data2018.WVHT ,14, 'gamma')
```



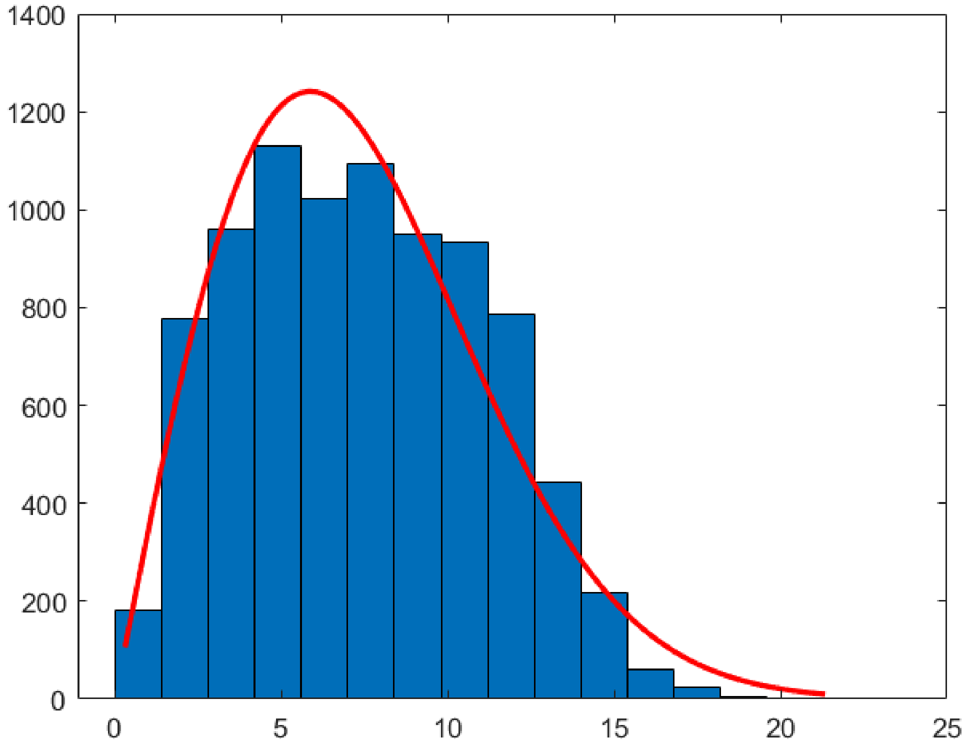

```
h9 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h10 = histfit(data2018.GST,14, 'nakagami')
```



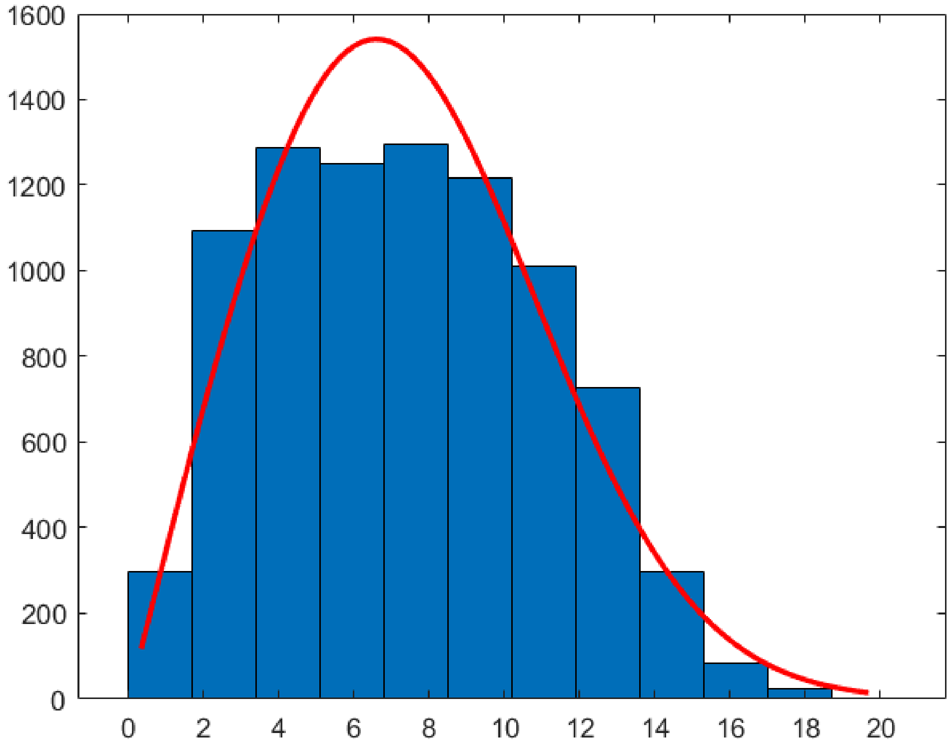

```
h10 =
  2×1 graphics array:

  Bar
  Line
```

```
h11 = histfit(data2018.GST,14, 'rayleigh')
```




```
h11 = 
  2×1 graphics array:

  Bar
  Line
```

```
clf
h12 = histfit(data2018.GST,12, 'rician')
```




```
h12 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h13 = histfit(data2018.GST,14, 'weibull')
```

```
h13 =
  2×1 graphics array:

  Bar
  Line
```

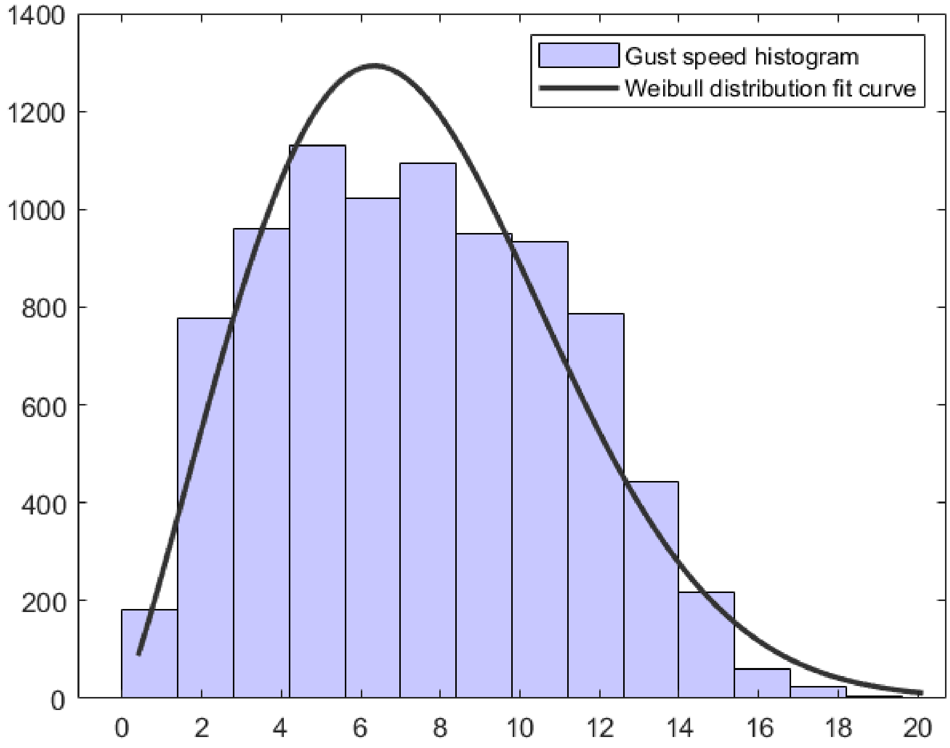
```
h13(1).FaceColor = [.8 .8 1];
h13(2).Color = [.2 .2 .2]
```

```
h13 =
  2×1 graphics array:

  Bar
  Line
```

```
h13(2).DisplayName = "Weibull distribution"
```

```
h13 =
  2×1 graphics array:

  Bar
  Line    (Weibull distribution)
```

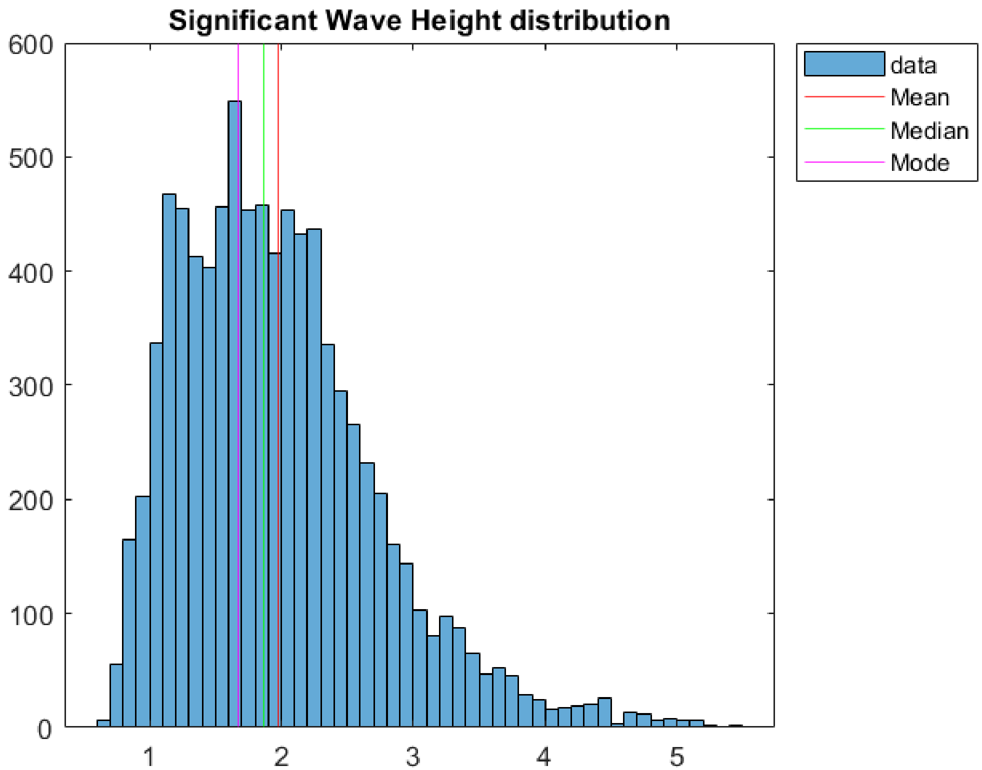
```
legend ("Gust speed histogram", "Weibull distribution fit curve")
```




At first glance, the **weibull distribution** (also raleigh) seems to be the one of the best distribution that best fits the **GST** values

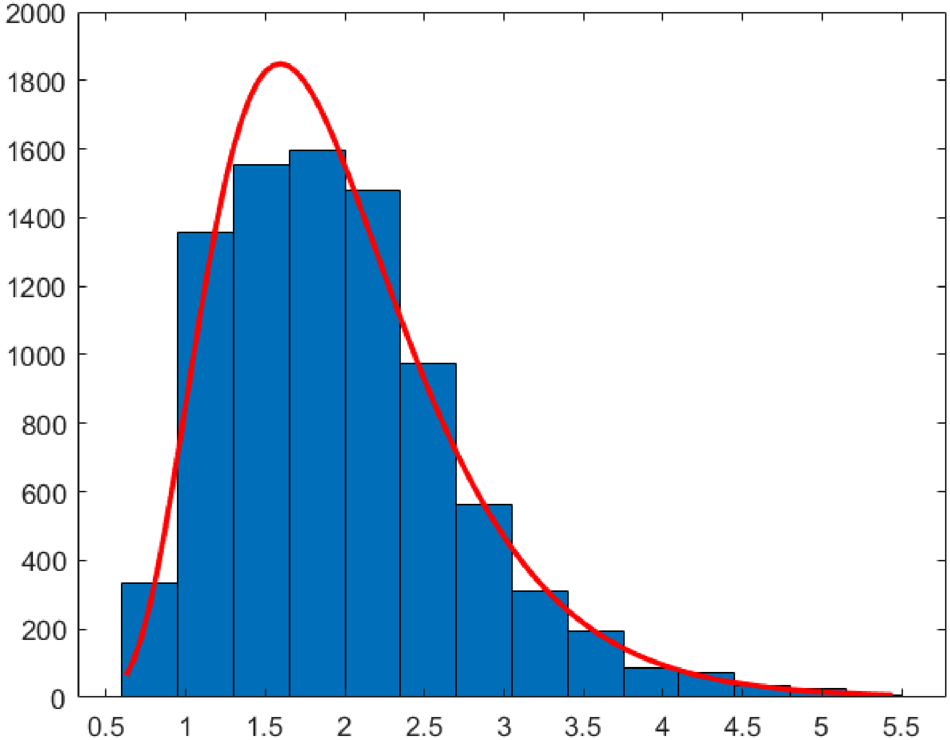
### Significant Wave Height - WVHT

```
figure4 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.WVHT);
hold on;
xline(wvht_cm.mean,'red');
xline(wvht_cm.median,'green');
xline(wvht_cm.mode,'magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Significant Wave Height distribution');
```




Median and mode are located to the left of right , that means WVHT data values is **positively biased** or **left-biased**

```
clf
h14 = histfit(data2018.WVHT,14, 'birnbaumsaunders')
```




```
h14 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h15 = histfit(data2018.WVHT,14, 'gamma')
```

```
h15 =
  2×1 graphics array:

  Bar
  Line
```

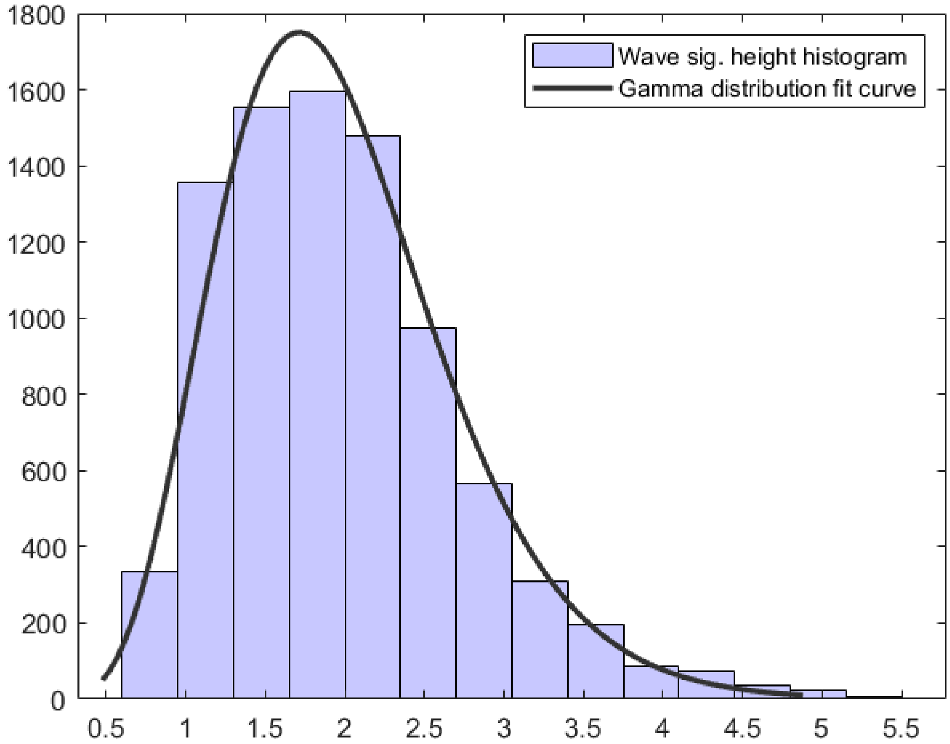
```
h15(1).FaceColor = [.8 .8 1];
h15(2).Color = [.2 .2 .2]
```

```
h15 =
  2×1 graphics array:

  Bar
  Line
```

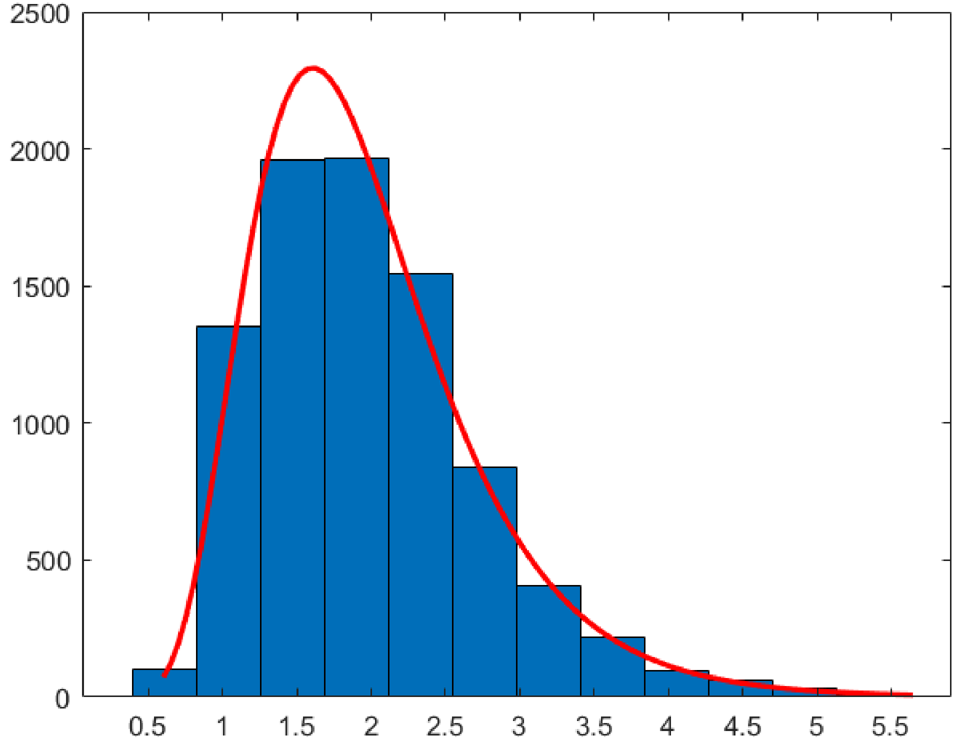
```
h15(2).DisplayName = "Gamma distribution"
```

```
h15 =
  2×1 graphics array:

  Bar
  Line    (Gamma distribution)
```

```
legend ("Wave sig. height histogram", "Gamma distribution fit curve")
```




```
clf
h16 = histfit(data2018.WVHT,12, 'lognormal')
```

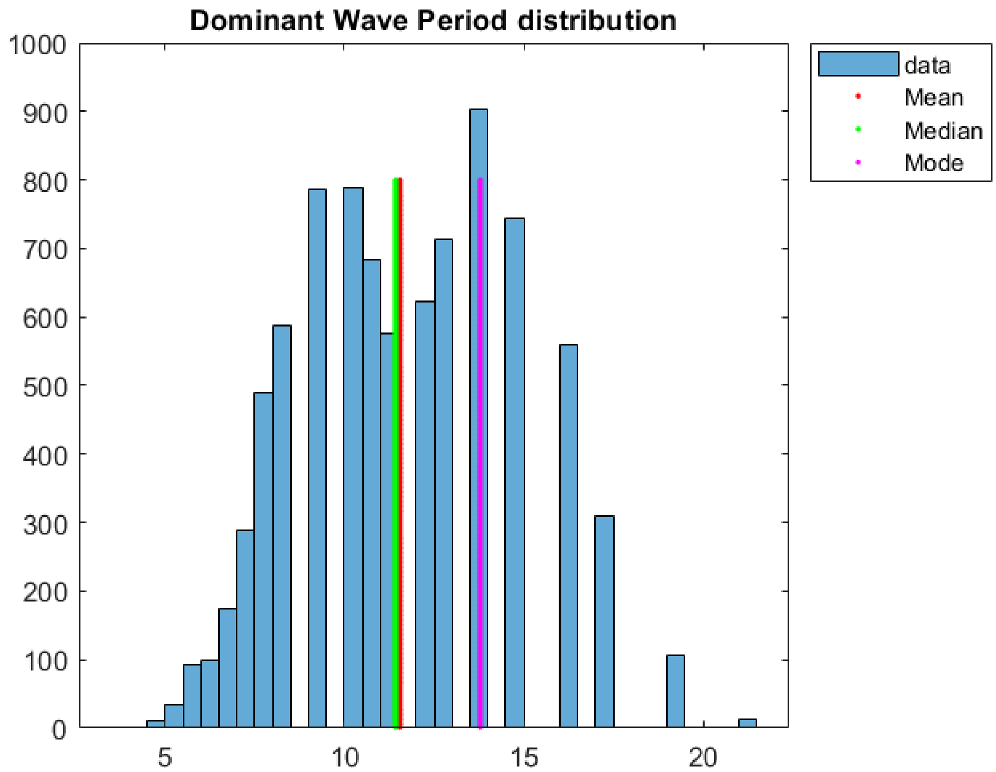



```
h16 =
  2×1 graphics array:

  Bar
  Line
```

the **gamma distribution** best fit to WVHT

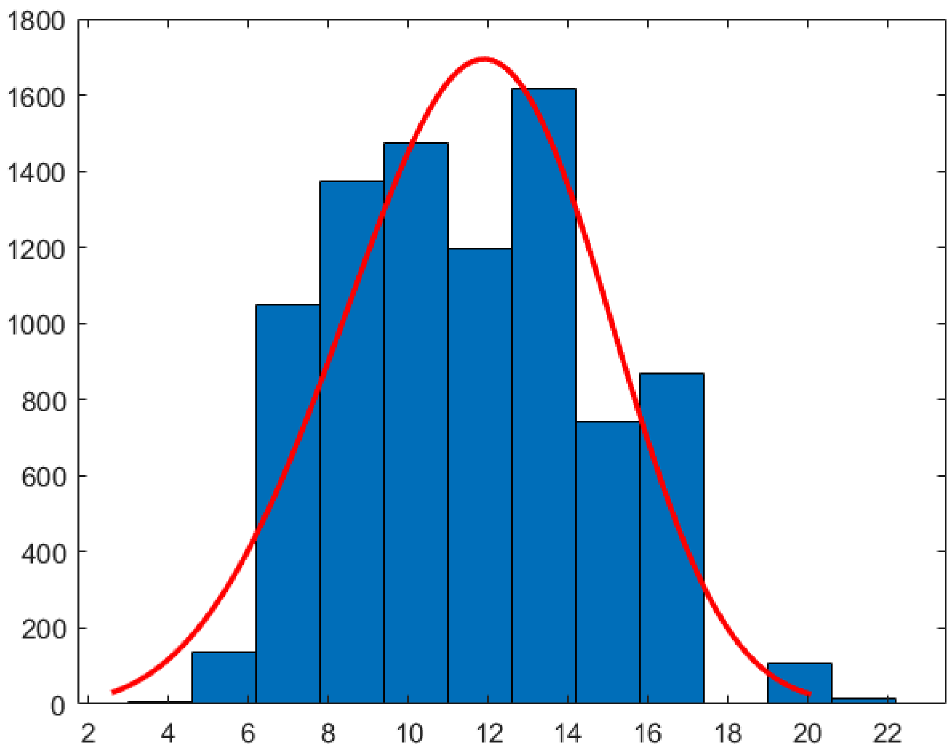
### Dominant Wave Period - DPD

```
figure5 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.DPD);
hold on;
scatter(repmat(dpd_cm.mean,1,800), 1:1:800,'.','red');
scatter(repmat(dpd_cm.median,1,800), 1:1:800,'.','green');
scatter(repmat(dpd_cm.mode,1,800), 1:1:800,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Dominant Wave Period distribution');
```




Median and mode are so separated and we can't say that data is so biased although is changeable (DPD takes discrete values)

```
clf
h17 = histfit(data2018.DPD,12,'weibull')
```




```
h17 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h17_2 = histfit(data2018.DPD,10,'gamma')
```

```
h17_2 =
  2×1 graphics array:

  Bar
  Line
```

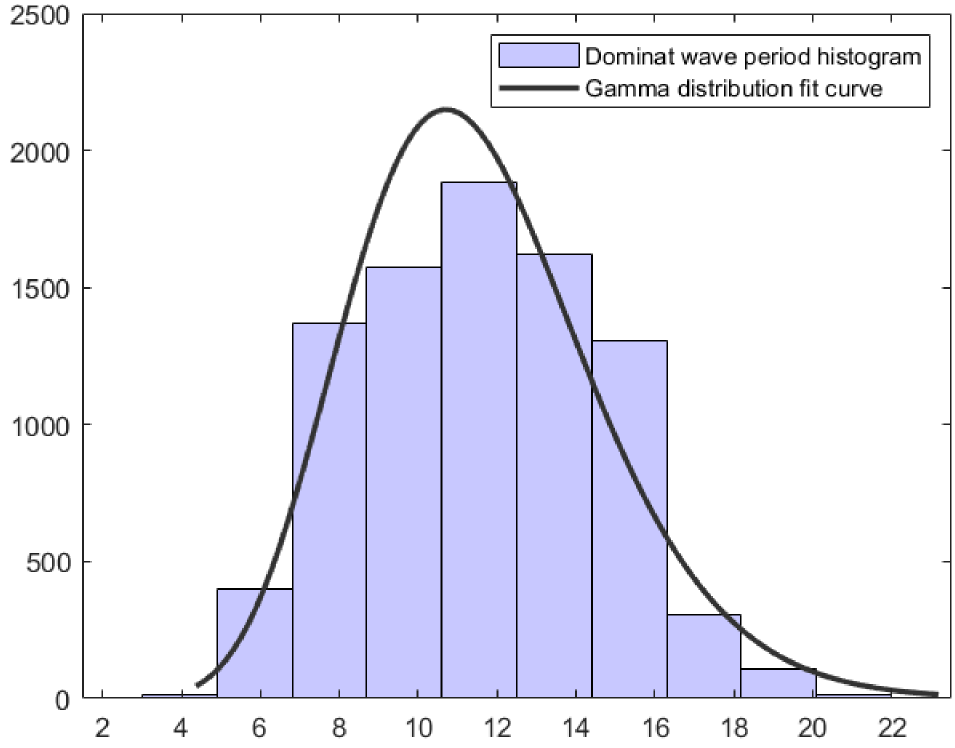
```
h17_2(1).FaceColor = [.8 .8 1];
h17_2(2).Color = [.2 .2 .2]
```

```
h17_2 =
  2×1 graphics array:

  Bar
  Line
```

```
h17_2(2).DisplayName = "Gamma distribution"
```

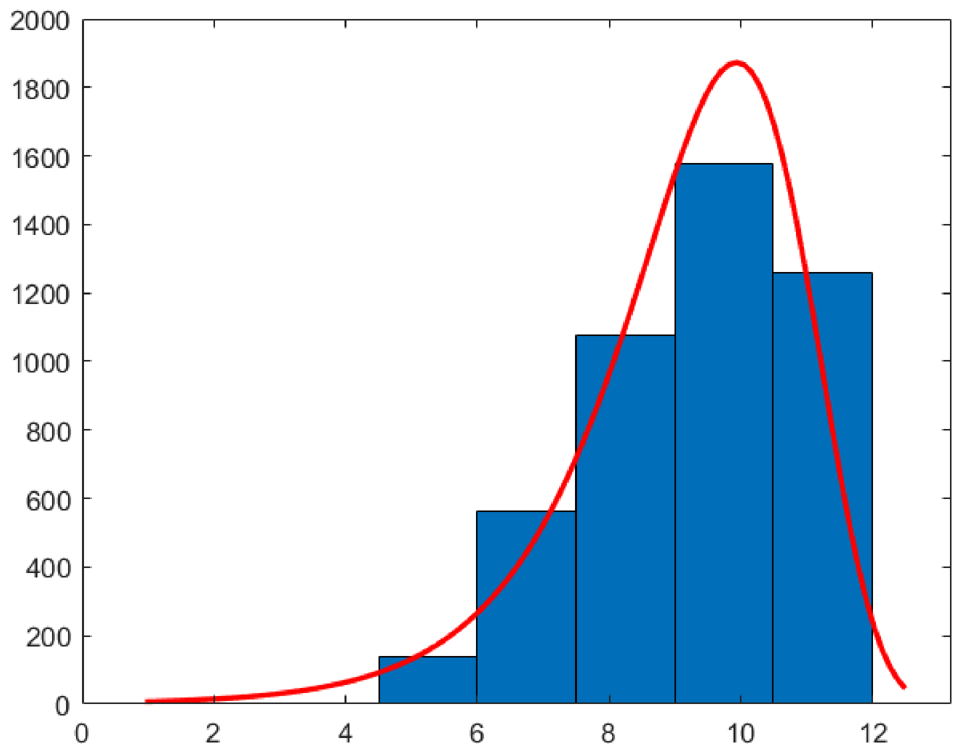
```
h17_2 =
  2×1 graphics array:

  Bar
  Line     (Gamma distribution)
```

```
legend ("Dominat wave period histogram", "Gamma distribution fit curve")
```



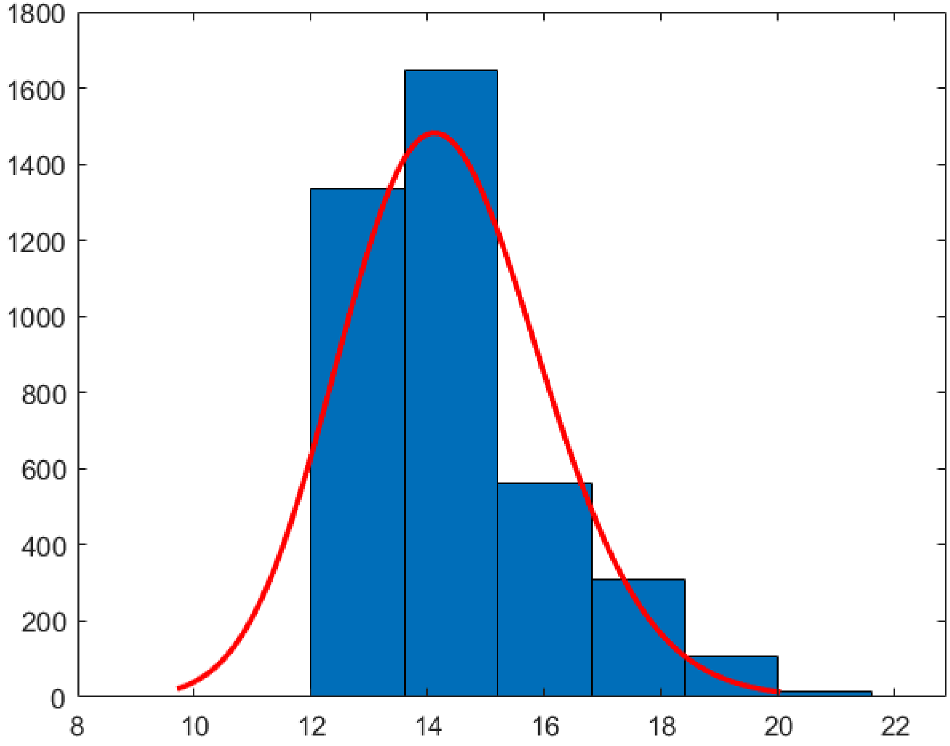

There is no a distribution that fit so well to data . It seems that the range 0-11 DPD values follows a distribution with less variance than data from 12 to maximun value which variance is higger.

```
h18 = histfit(data2018.DPD(data2018.DPD <= 12),6,'extreme value')
```




```
h18 =
  2×1 graphics array:

  Bar
  Line
```

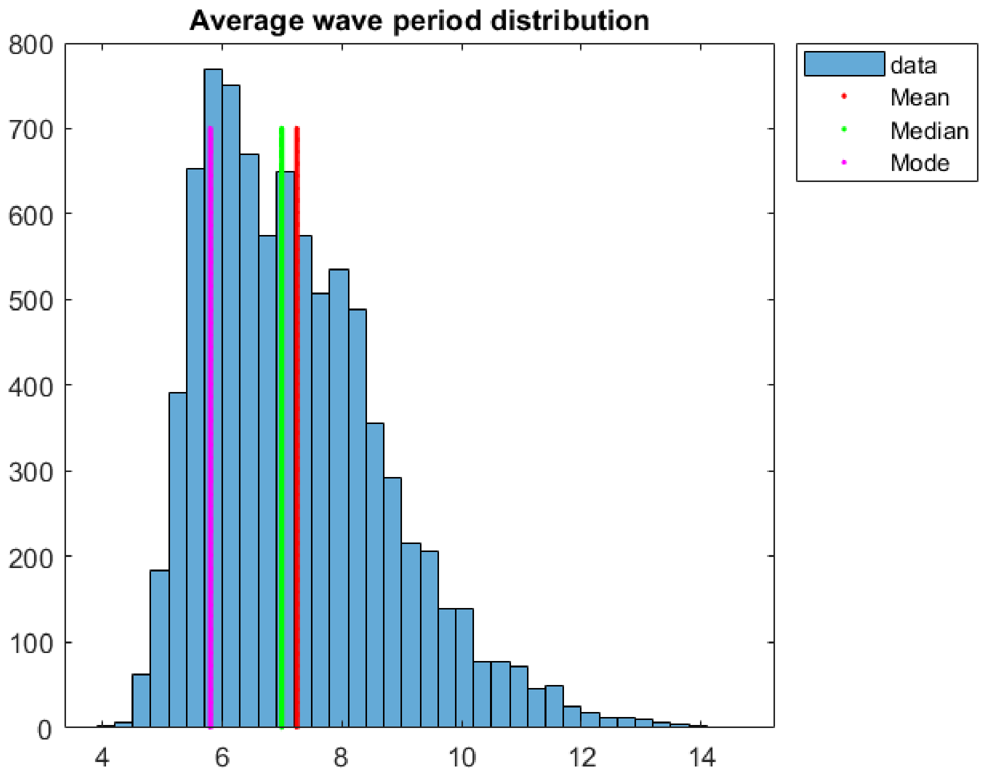
```
h18 = histfit(data2018.DPD(data2018.DPD > 12),6,'gamma')
```




```
h18 = 
  2×1 graphics array:

  Bar
  Line
```

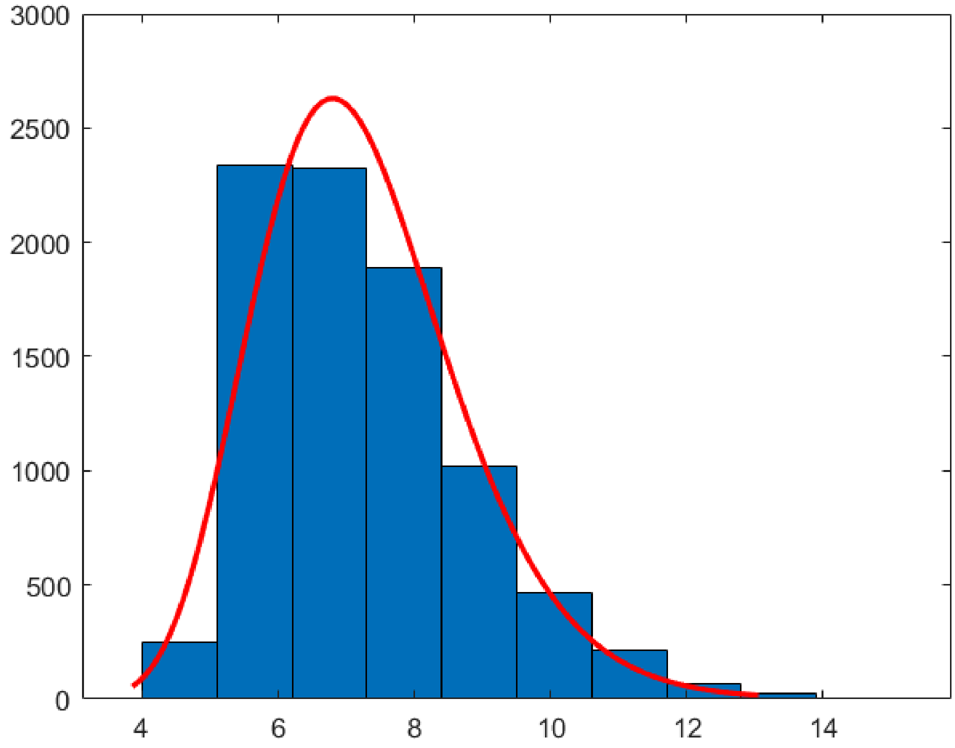
### Average Wave period - APD

```
figure6 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.APD);
hold on;
scatter(repmat(apd_cm.mean,1,700), 1:1:700,'.','red');
scatter(repmat(apd_cm.median,1,700), 1:1:700,'.','green');
scatter(repmat(apd_cm.mode,1,700), 1:1:700,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Average wave period distribution');
```



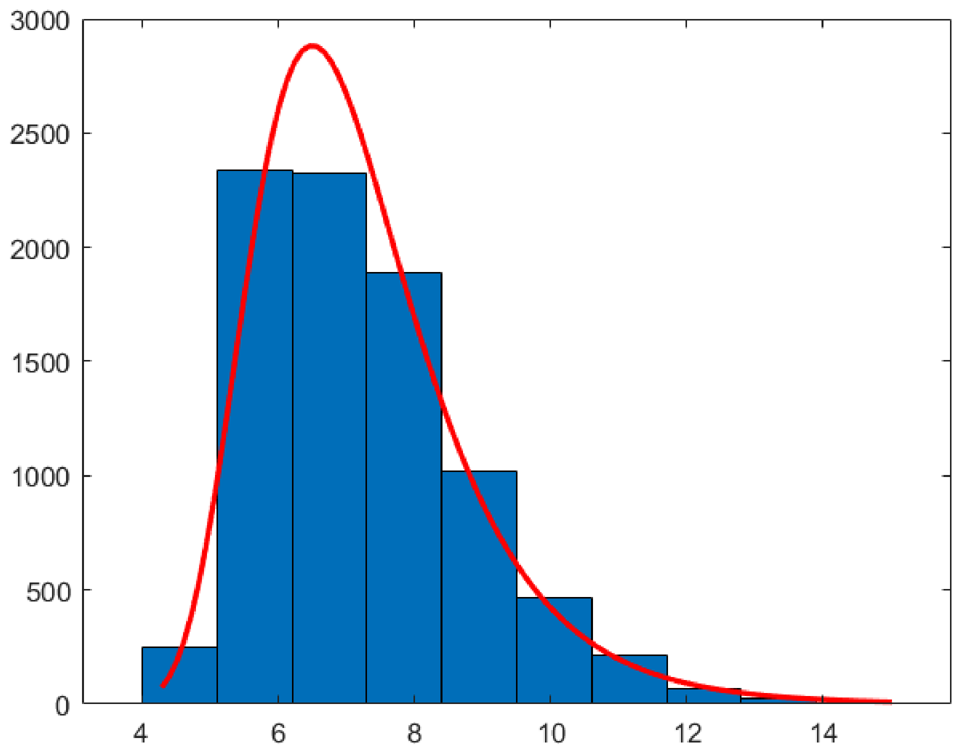

Meadian and mode are both at left of the median, so Average wave period values are positive asimetric

```
clf
h19 = histfit(data2018.APD,10,'birnbaumsaunders')
```



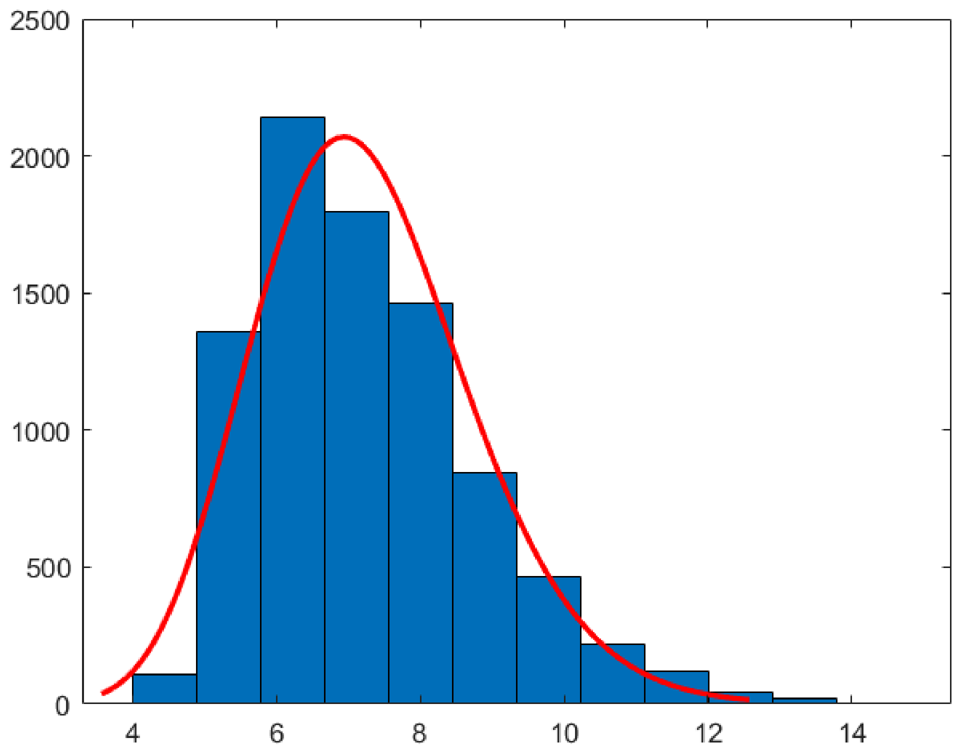

```
h19 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h20 = histfit(data2018.APD,10,'generalized extreme value')
```




```
h20 = 
  2×1 graphics array:

  Bar
  Line
```

```
clf
h21 = histfit(data2018.APD,12,'gamma')
```




```
h21 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h21_2 = histfit(data2018.APD,12,'inverse gaussian')
```

```
h21_2 =
  2×1 graphics array:

  Bar
  Line
```

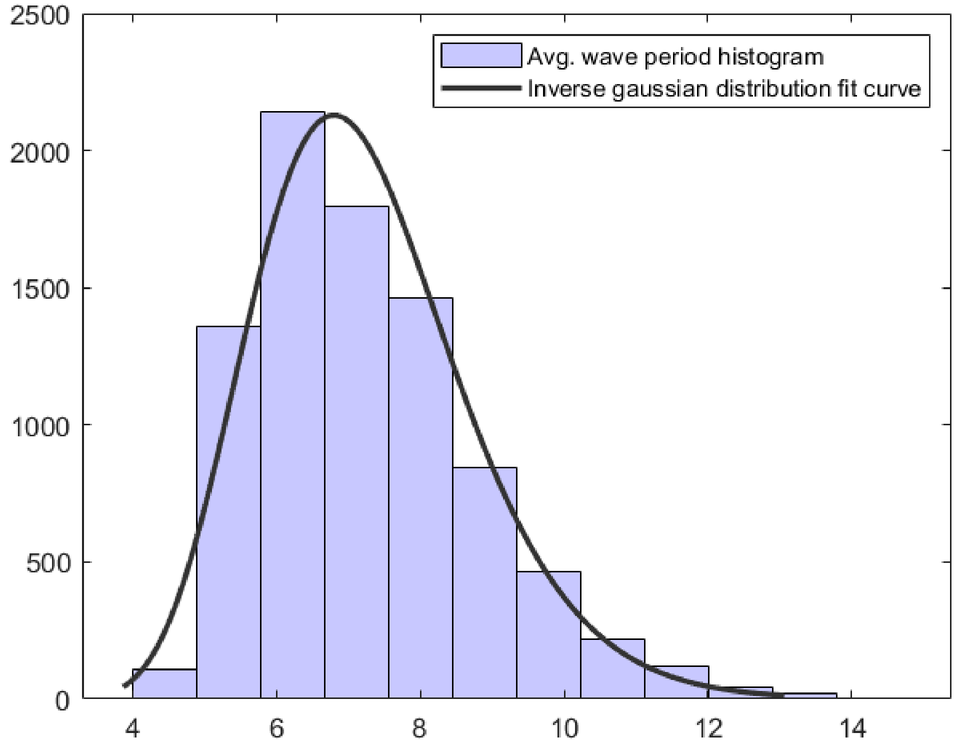
```
h21_2(1).FaceColor = [.8 .8 1];
h21_2(2).Color = [.2 .2 .2]
```

```
h21_2 =
  2×1 graphics array:

  Bar
  Line
```

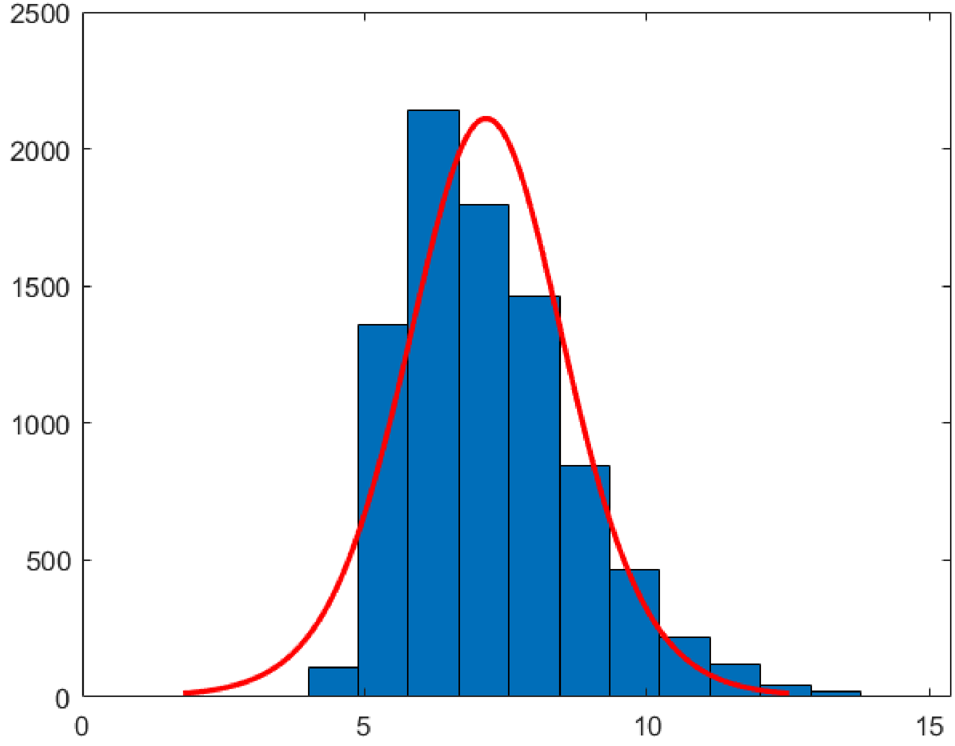
```
h21_2(2).DisplayName = "Inverse gaussian distribution"
```

```
h21_2 =
  2×1 graphics array:

  Bar
  Line    (Inverse gaussian distribution)
```

```
legend ("Avg. wave period histogram", "Inverse gaussian distribution fit curve")
```




```
h22 = histfit(data2018.APD,12,'tlocationscale')
```

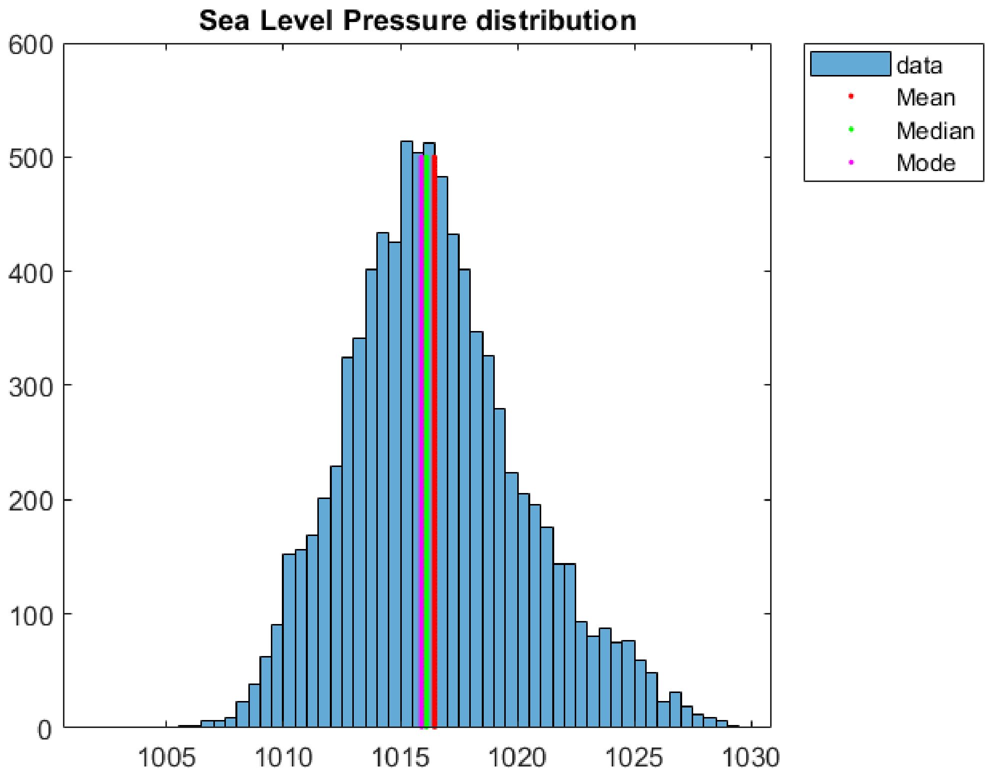



```
h22 = 
  2×1 graphics array:

  Bar
  Line
```

The **inverse gaussian** fits better to **Average Wave period , also gamma as was expected coul be a good aproximation**

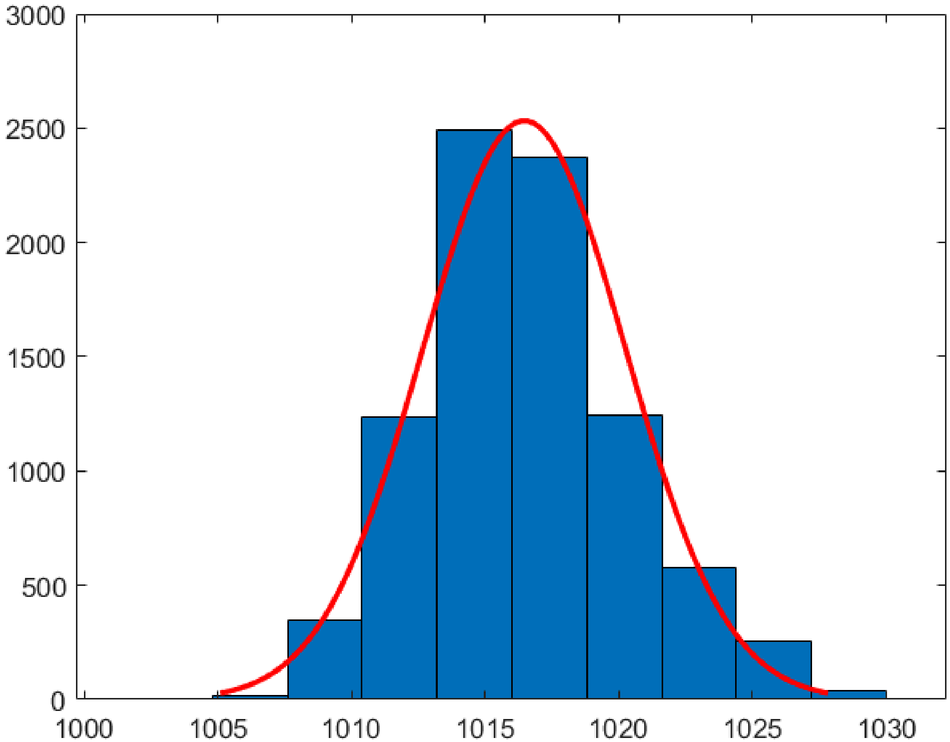
### Sea level pressure - PRES

```
figure8 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.PRES);
hold on;
scatter(repmat(pres_cm.mean,1,500), 1:1:500,'.','red');
scatter(repmat(pres_cm.median,1,500), 1:1:500,'.','green');
scatter(repmat(pres_cm.mode,1,500), 1:1:500,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Sea Level Pressure distribution');
```




Data is almost simetric as we can see. Mean , median and mode are relatively close to each other

```
clf
h24 = histfit(data2018.PRES,10,'normal')
```




```
h24 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h25 = histfit(data2018.PRES,12,'generalized extreme value')
```

```
h25 =
  2×1 graphics array:

  Bar
  Line
```

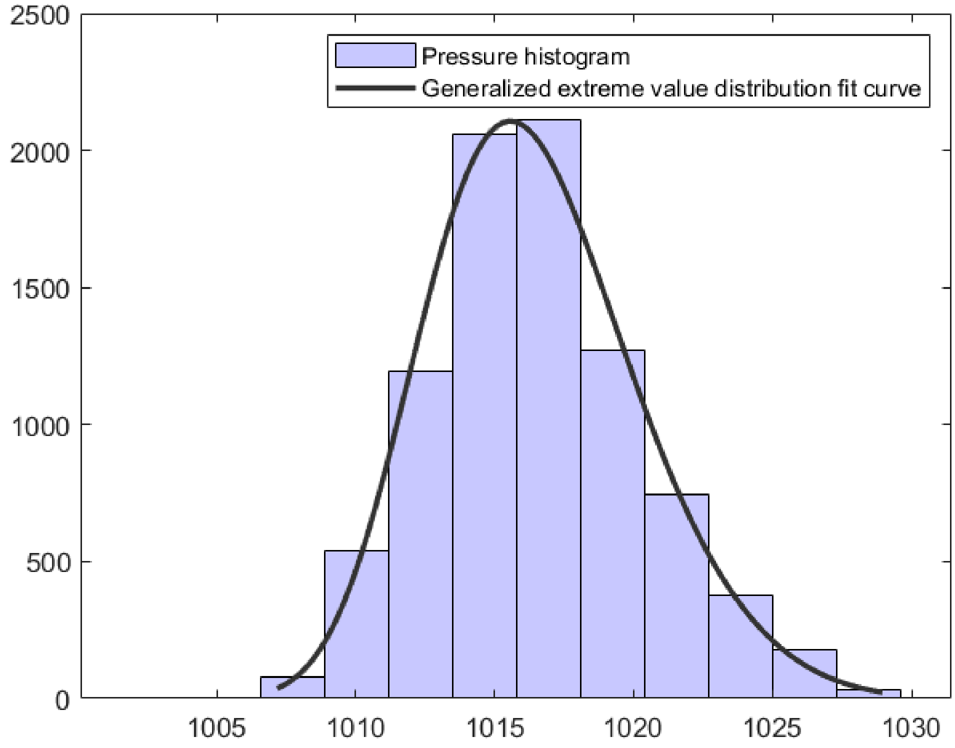
```
h25(1).FaceColor = [.8 .8 1];
h25(2).Color = [.2 .2 .2]
```

```
h25 =
  2×1 graphics array:

  Bar
  Line
```

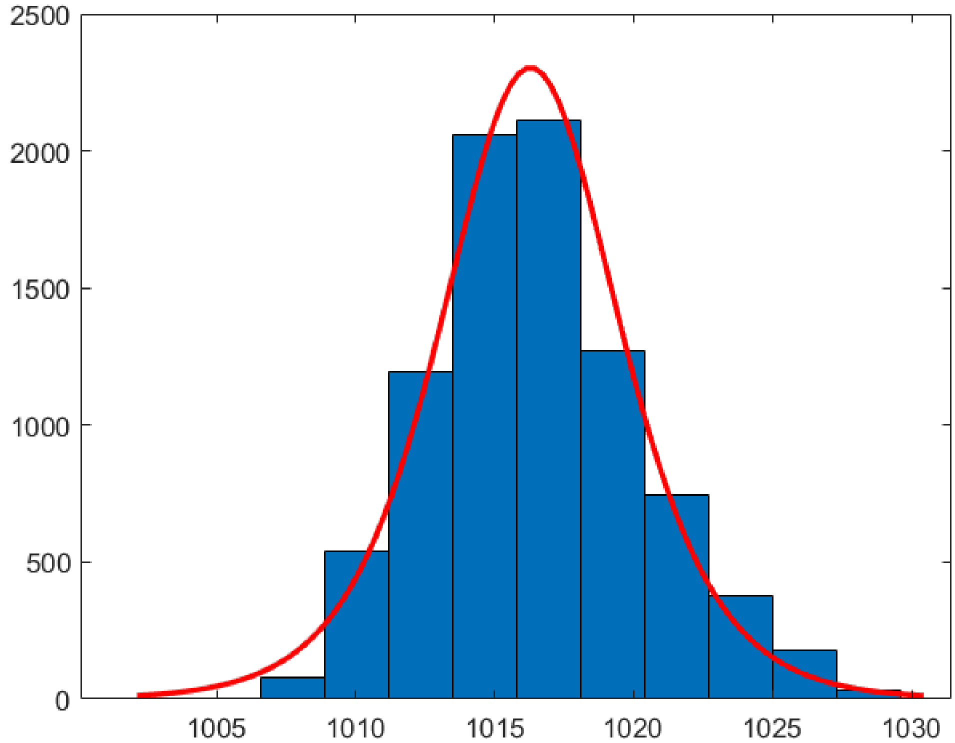
```
h25(2).DisplayName = "Generalized extreme value distribution"
```

```
h25 =
  2×1 graphics array:

  Bar
  Line    (Generalized extreme value distribution)
```

```
legend ("Pressure histogram", "Generalized extreme value distribution fit curve")
```

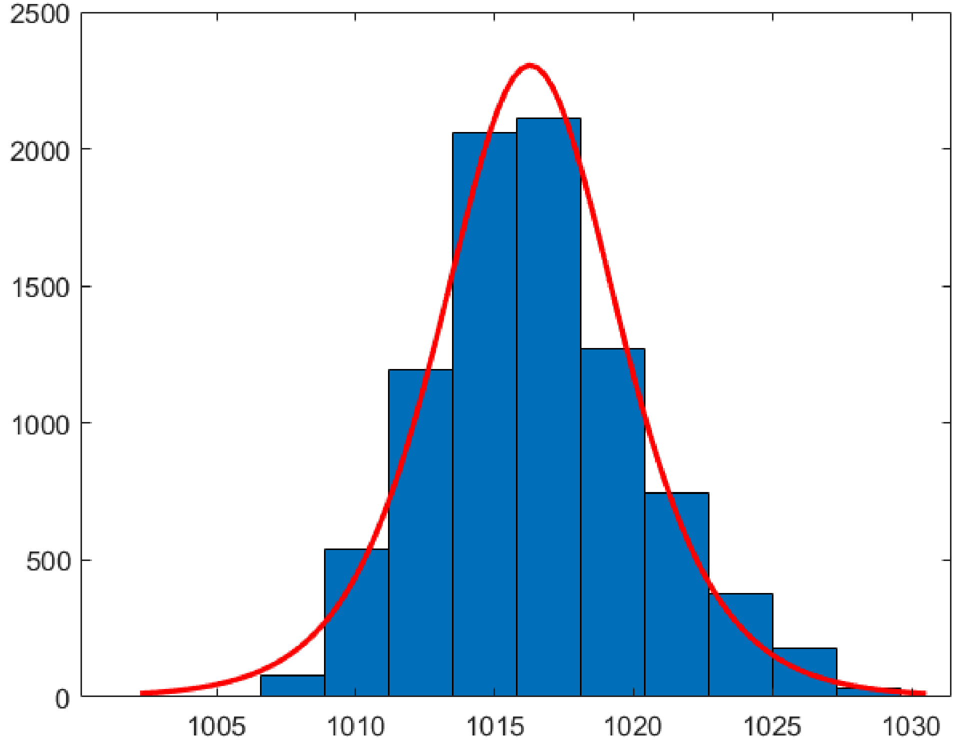



```
clf
h26 = histfit(data2018.PRES,12,'logistic')
```




```
h26 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h27 = histfit(data2018.PRES,12,'loglogistic')
```

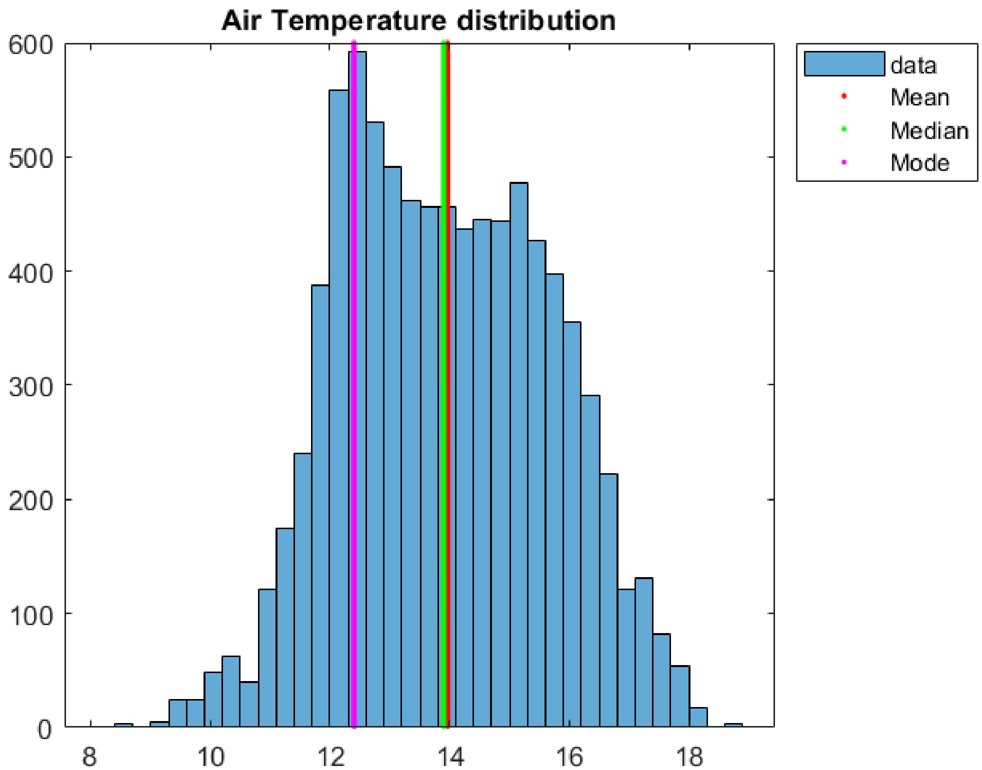



```
h27 = 
  2×1 graphics array:

  Bar
  Line
```

The **logistic distribution or loglogistic distribution** best fit to Sea Level Pressure

### Air Temperature - ATMP

```
figure9 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.ATMP);
hold on;
scatter(repmat(atmp_cm.mean,1,600), 1:1:600,'.','red');
scatter(repmat(atmp_cm.median,1,600), 1:1:600,'.','green');
scatter(repmat(atmp_cm.mode,1,600), 1:1:600,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Air Temperature distribution');
```




The appearence of air temperature histogram make us think that it is a bit biased to left or positively asimetric. But it doesn't follow a normal distribution in all histogram. The histogram bars fluctuate

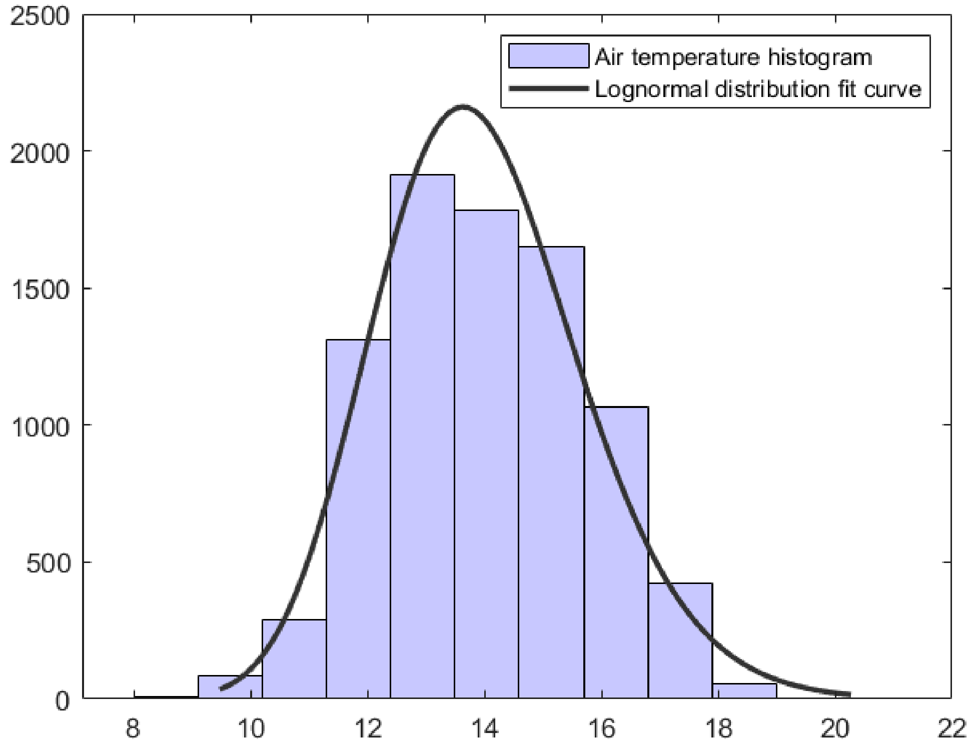
```
clf
h28 = histfit(data2018.ATMP,10,'lognormal')
```

```
h28 =
  2×1 graphics array:

  Bar
  Line
```

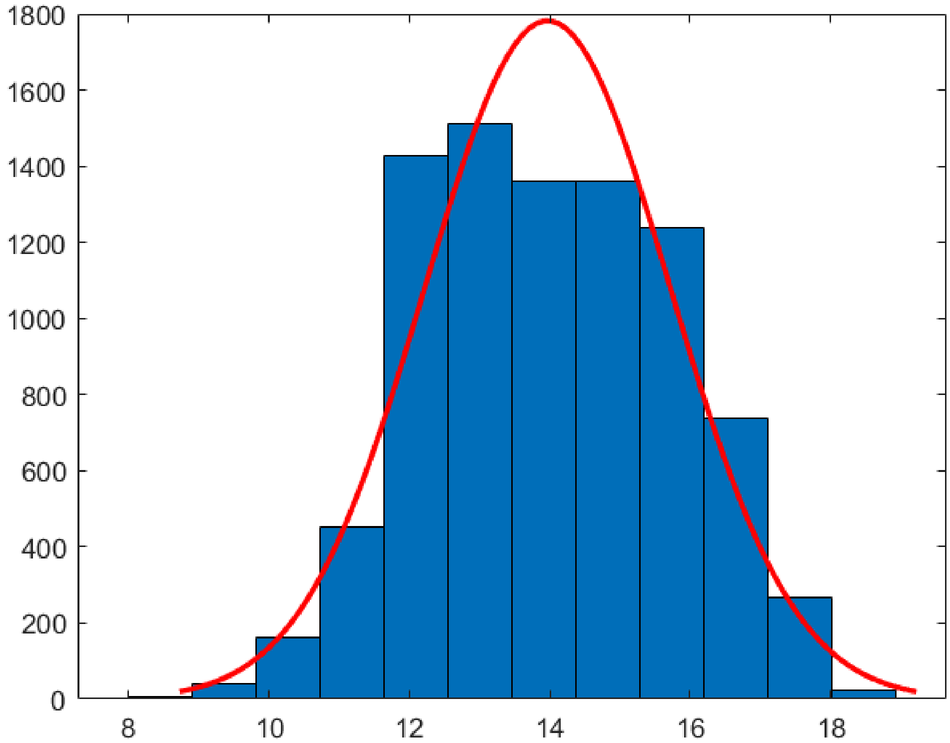
```
h28(1).FaceColor = [.8 .8 1];
h28(2).Color = [.2 .2 .2];
h28(2).DisplayName = "Lognormal distribution"
```

```
h28 =
  2×1 graphics array:

  Bar
  Line    (Lognormal distribution)
```

```
legend ("Air temperature histogram", "Lognormal distribution fit curve")
```

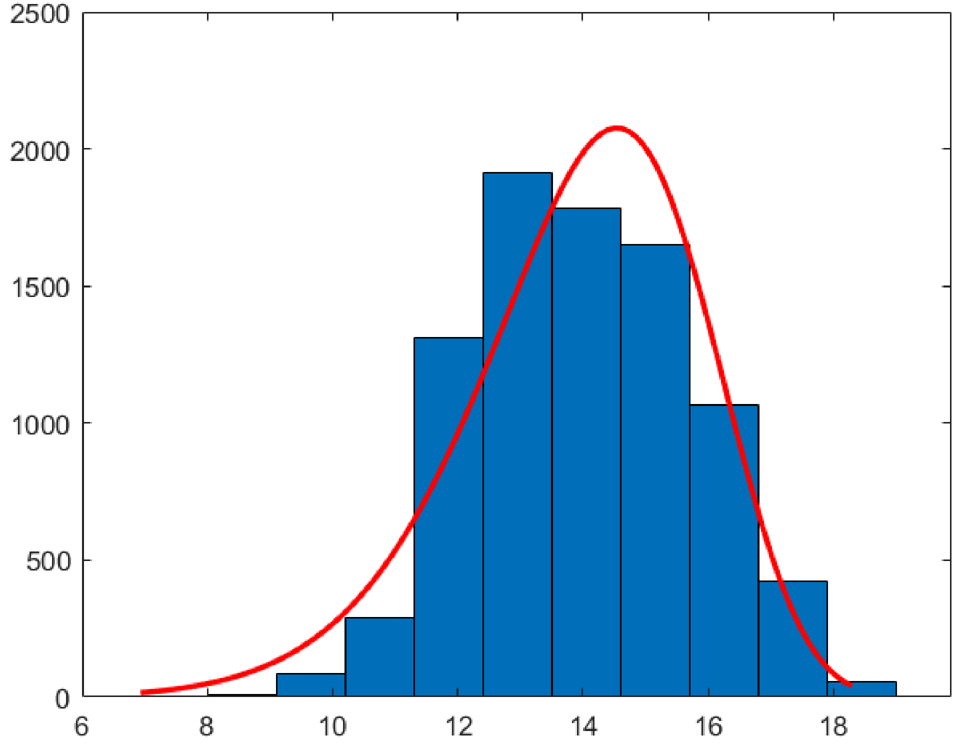



```
clf
h28 = histfit(data2018.ATMP,12,'rician')
```




```
h28 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h29 = histfit(data2018.ATMP,10,'weibull')
```

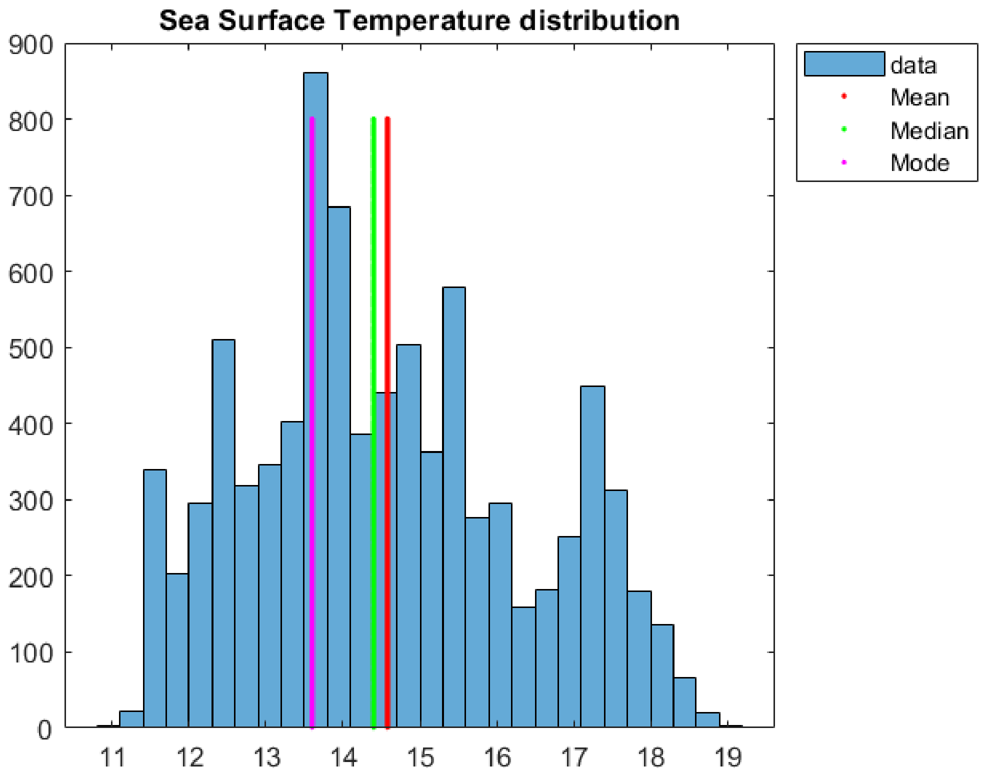



```
h29 =
  2×1 graphics array:

  Bar
  Line
```

The **lognormal distribution** is the most approximated distribution to real Air Temperature distribution

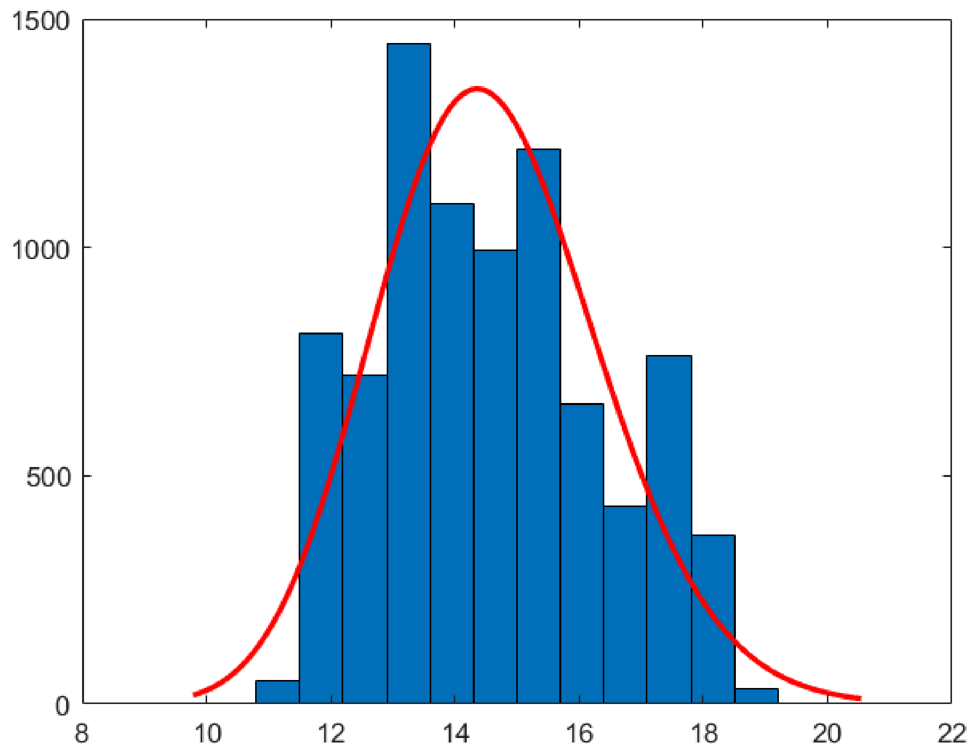
### Sea Surface Temperature - WTMP

```
figure10 = figure('Colormap',...
    [0 1 1;0.015873015873016 0.984126984126984 1;0.031746031746032 0.968253968253968 1;0.047619
histogram(data2018.WTMP);
hold on;
scatter(repmat(wtmp_cm.mean,1,800), 1:1:800,'.','red');
scatter(repmat(wtmp_cm.median,1,800), 1:1:800,'.','green');
scatter(repmat(wtmp_cm.mode,1,800), 1:1:800,'.','magenta');
legend({'data','Mean','Median','Mode'},'Location','bestoutside');
title('Sea Surface Temperature distribution');
```




Median and mode are located at the left of median , so we can say data is positevely asimetric

```
clf
h30 = histfit(data2018.WTMP,12,'gamma')
```




```
h30 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h31 = histfit(data2018.WTMP,10,'lognormal')
```

```
h31 =
  2×1 graphics array:

  Bar
  Line
```

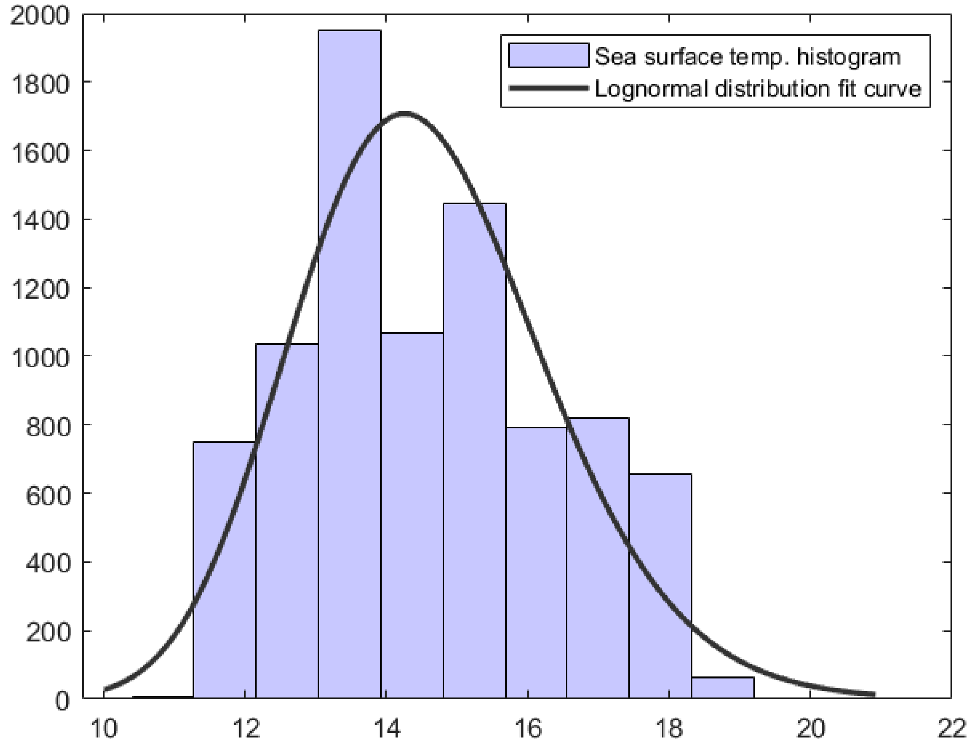
```
h31(1).FaceColor = [.8 .8 1];
h31(2).Color = [.2 .2 .2]
```

```
h31 =
  2×1 graphics array:

  Bar
  Line
```

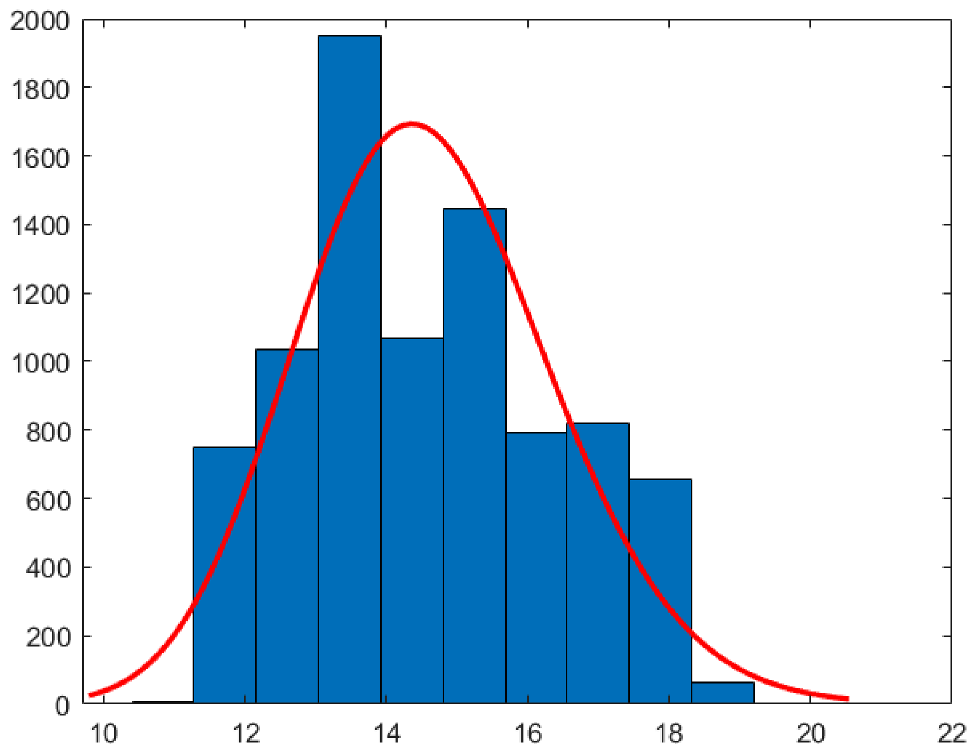
```
h31(2).DisplayName = "Lognormal distribution"
```

```
h31 =
  2×1 graphics array:

  Bar
  Line    (Lognormal distribution)
```

```
legend ("Sea surface temp. histogram", "Lognormal distribution fit curve")
```

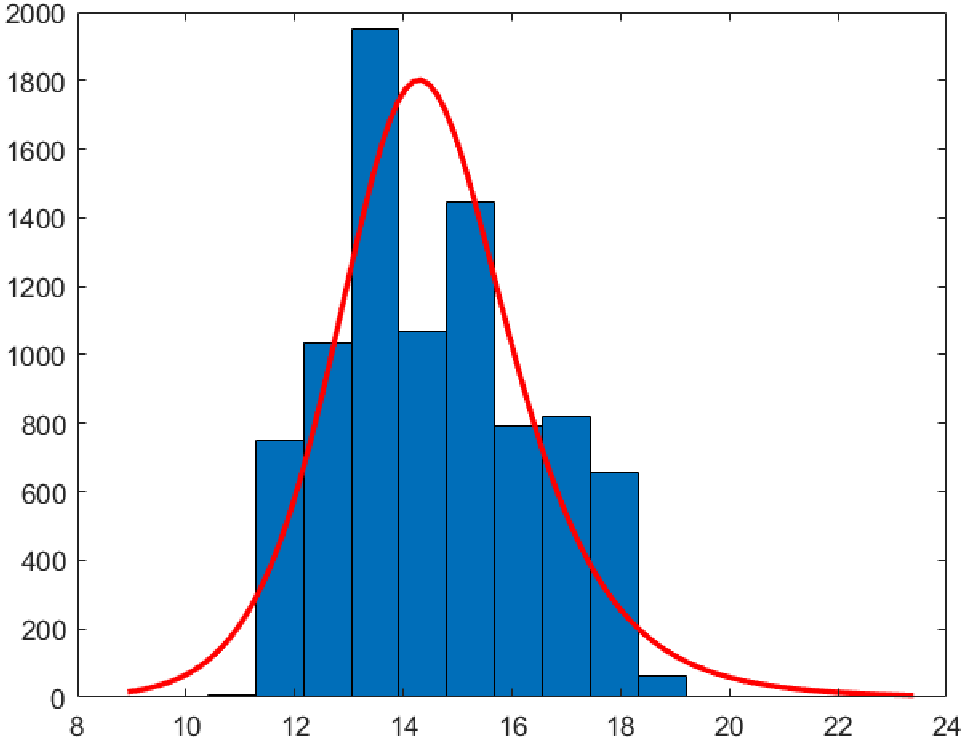



```
clf
h32 = histfit(data2018.WTMP,10,'gamma')
```




```
h32 =
  2×1 graphics array:

  Bar
  Line
```

```
clf
h33 = histfit(data2018.WTMP,10,'loglogistic')
```




```
h33 =
  2×1 graphics array:

  Bar
  Line
```

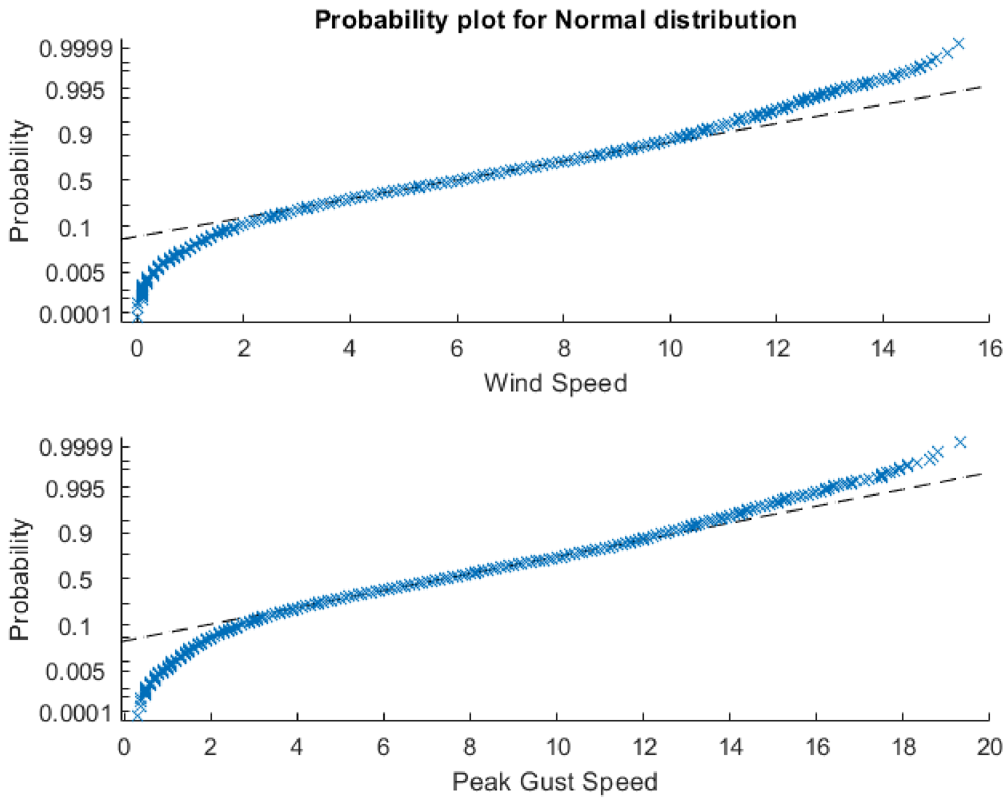
The best approach found is **the loglogistic distribution** although it doesn't represent the data irregular values that make bars go up and down between values 13 and 17 for **Sea surface temperature measure**

## Normal Probability plot

Comparision of each variable distribution relative to normal distribution **.** It is complementary to the previous subsection.

### WSPD and GST

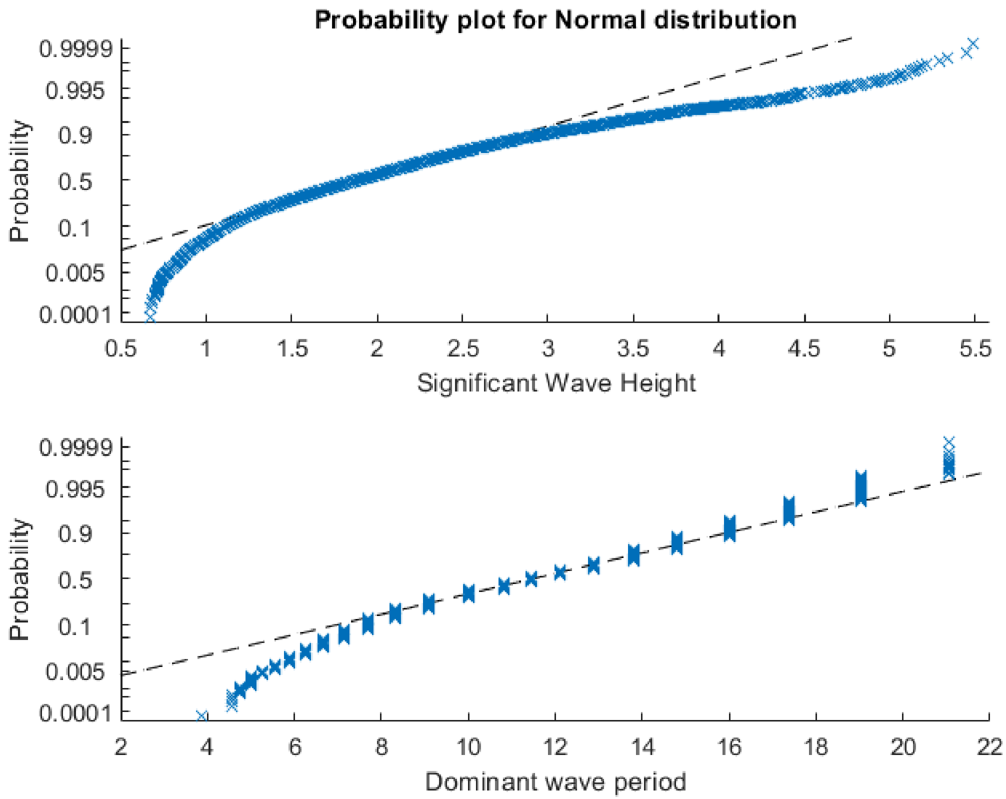
```
clf
subplot(2,1,1)
probplot('normal',data2018.WSPD);
xlabel("Wind Speed");

subplot(2,1,2)
probplot('normal',data2018.GST);
xlabel("Peak Gust Speed");
title('')
```



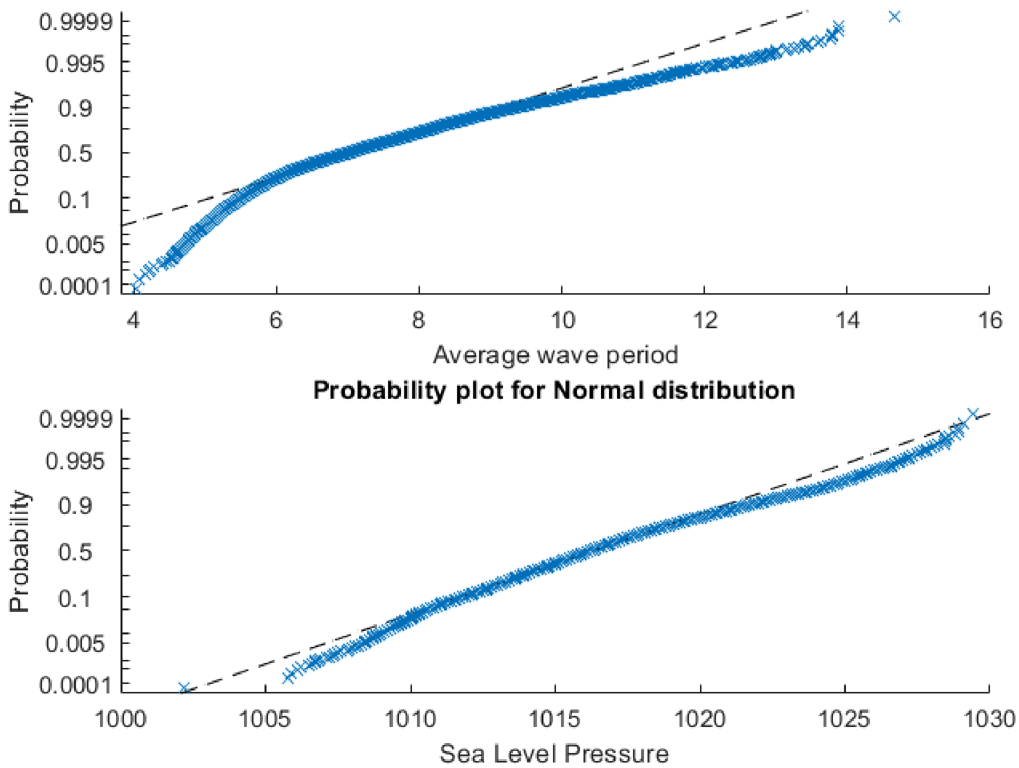

It seems that wind speed and peak of gust speed are so close to normal distribution for values in the interval [0,14]. Data is not too much deviated from normality. Then lets see the comparision with logistic distribution (best fitter function approached yet)

We can observe that probability distribution for wind direction follows the the trajectory of a cubic function.

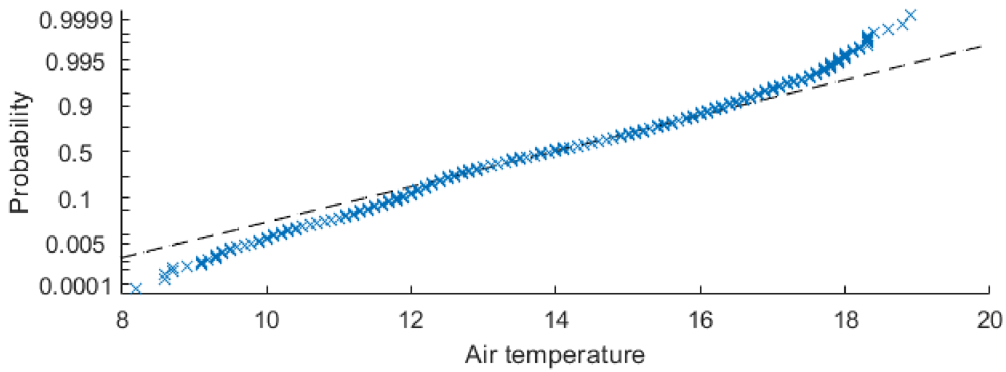
```
clf
subplot(2,1,1)
probplot('normal',data2018.WVHT);
xlabel("Significant Wave Height");

subplot(2,1,2)
probplot('normal',data2018.DPD);
xlabel("Dominant wave period");
title('');
```

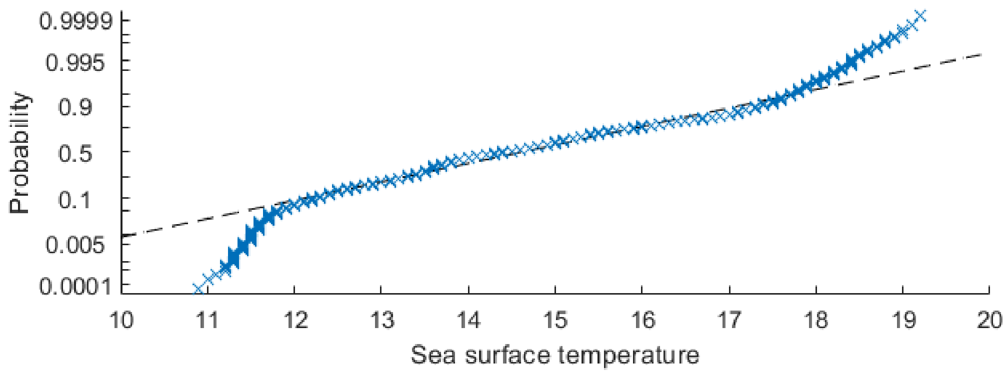



```
clf
hold on
subplot(2,1,1)
probplot('normal',data2018.APD);
xlabel("Average wave period");
title('');
subplot(2,1,2)
probplot('normal',data2018.PRES);
xlabel("Sea Level Pressure");
```




```
clf
subplot(2,1,1)
probplot('normal',data2018.ATMP);
xlabel("Air temperature");
title('');
subplot(2,1,2)
probplot('normal',data2018.WTMP);
xlabel("Sea surface temperature");
title('');
```




Checking for tidy data : the tidy data principle claims that data is tidy if each column correspond to a feature, each row correspond to a observation example and together form a data set table. This is relatively simple as we check looking the data frame if we haven't made any mistake that affected the data set read

```
data2018
```

data2018 = 8587×16 table

...

| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2018 | 1 | 1 | 0 | 50 | 36 | 0.8000 | 1.4000 | 0.8500 |
| 2 | 2018 | 1 | 1 | 1 | 50 | 17 | 0.8000 | 1.1000 | 0.9200 |
| 3 | 2018 | 1 | 1 | 2 | 50 | 354 | 0.5000 | 0.9000 | 0.8700 |
| 4 | 2018 | 1 | 1 | 3 | 50 | 23 | 1.2000 | 1.6000 | 0.9200 |
| 5 | 2018 | 1 | 1 | 4 | 50 | 11 | 1.1000 | 1.3000 | 0.8500 |
| 6 | 2018 | 1 | 1 | 5 | 50 | 325 | 1.1000 | 1.6000 | 0.9000 |
| 7 | 2018 | 1 | 1 | 6 | 50 | 299 | 1.5000 | 1.8000 | 0.8000 |
| 8 | 2018 | 1 | 1 | 7 | 50 | 311 | 2.6000 | 3.1000 | 0.8100 |
| 9 | 2018 | 1 | 1 | 8 | 50 | 329 | 3.0000 | 3.6000 | 0.7400 |
| 10 | 2018 | 1 | 1 | 9 | 50 | 338 | 2.6000 | 3.2000 | 0.7500 |
| 11 | 2018 | 1 | 1 | 10 | 50 | 358 | 3.3000 | 3.9000 | 0.7200 |

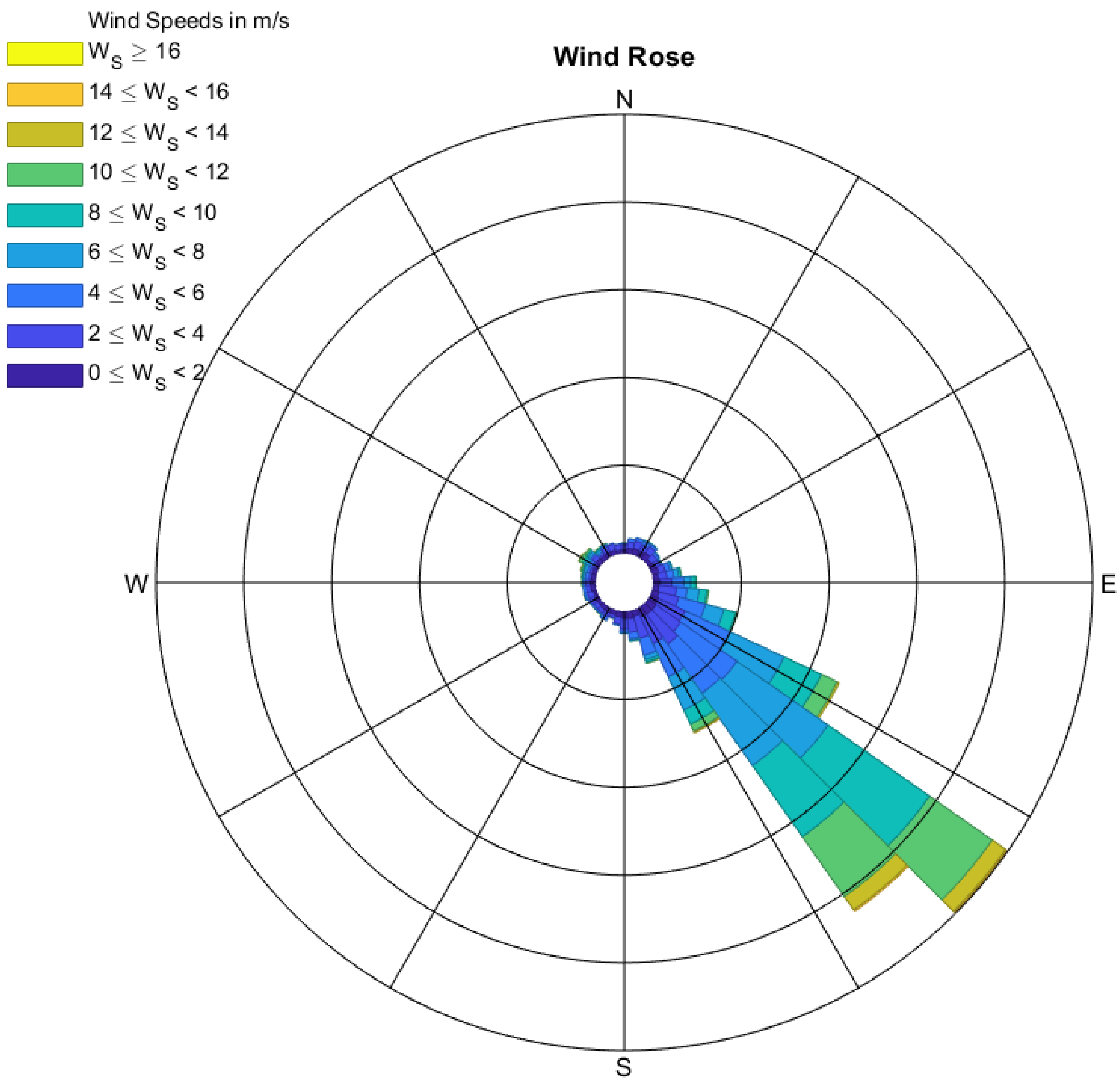
| | YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT |
|---|---|---|---|---|---|---|---|---|---|
| 12 | 2018 | 1 | 1 | 11 | 50 | 350 | 3.6000 | 4.2000 | 0.7000 |
| 13 | 2018 | 1 | 1 | 12 | 50 | 344 | 4.0000 | 4.8000 | 0.6800 |
| 14 | 2018 | 1 | 1 | 13 | 50 | 354 | 4.9000 | 5.7000 | 0.6700 |

⋮

# Bi-vriate Analysis

## Wind direction and Wind speed representation: Wind Rose

```
[figure_handle,count,speeds,directions,Table] = WindRose(data2018.WDIR, data2018.WSPD)
```




```
figure_handle = 
  Figure (24) with properties:
```

```
      Number: 24
        Name: ''
       Color: [1 1 1]
    Position: [448 160 640 640]
       Units: 'pixels'

  Show all properties
count = 36×9
    0.3610    0.6405    0.6405    0.3959    0.2562    0.0466         0         0 ···
    0.2562    0.4891    0.3843    0.2562    0.1281    0.0349         0         0
    0.3610    0.3610    0.2562    0.1281    0.0466         0         0         0
    0.2562    0.3028    0.0815    0.0466    0.0116         0         0         0
    0.1980    0.3727    0.1514    0.0233    0.0233         0         0         0
    0.3843    0.4192    0.1281    0.0349    0.0116         0         0         0
    0.3028    0.3610    0.2213    0.0466    0.0349    0.0116         0         0
    0.3144    0.4309    0.1863    0.0233         0         0         0         0
    0.2678    0.3610    0.1747    0.0233         0         0         0         0
    0.1863    0.2562    0.0699    0.0349    0.0116         0         0         0
      ⋮

speeds = 1×9
     0     2     4     6     8    10    12    14    16
directions = 36×1
     0
    10
    20
    30
    40
    50
    60
    70
    80
    90
      ⋮

Table = 40×12 cell
```

···

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 'Frequencie... | '' | 'Wind Speed... | '' | '' | '' | '' | '' |
| 2 | 'Direction ... | 'Avg. Direc... | '(0 , 2)' | '[2 , 4)' | '[4 , 6)' | '[6 , 8)' | '[8 , 10)' | '[10 , 12)' |
| 3 | '[355 , 5)' | 0 | 0.3610 | 0.6405 | 0.6405 | 0.3959 | 0.2562 | 0.0466 |
| 4 | '[5 , 15)' | 10 | 0.2562 | 0.4891 | 0.3843 | 0.2562 | 0.1281 | 0.0349 |
| 5 | '[15 , 25)' | 20 | 0.3610 | 0.3610 | 0.2562 | 0.1281 | 0.0466 | 0 |
| 6 | '[25 , 35)' | 30 | 0.2562 | 0.3028 | 0.0815 | 0.0466 | 0.0116 | 0 |
| 7 | '[35 , 45)' | 40 | 0.1980 | 0.3727 | 0.1514 | 0.0233 | 0.0233 | 0 |
| 8 | '[45 , 55)' | 50 | 0.3843 | 0.4192 | 0.1281 | 0.0349 | 0.0116 | 0 |
| 9 | '[55 , 65)' | 60 | 0.3028 | 0.3610 | 0.2213 | 0.0466 | 0.0349 | 0.0116 |
| 10 | '[65 , 75)' | 70 | 0.3144 | 0.4309 | 0.1863 | 0.0233 | 0 | 0 |
| 11 | '[75 , 85)' | 80 | 0.2678 | 0.3610 | 0.1747 | 0.0233 | 0 | 0 |
| 12 | '[85 , 95)' | 90 | 0.1863 | 0.2562 | 0.0699 | 0.0349 | 0.0116 | 0 |
| 13 | '[95 , 105)' | 100 | 0.1747 | 0.3028 | 0.0349 | 0.0116 | 0 | 0 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 14 | '[105 , ... | 110 | 0.2446 | 0.3261 | 0.0466 | 0.0116 | 0 | 0 |

⋮

mathworks.com/matlabcentral/fileexchange/47248-wind-rose

## Pairwise plots

The main objective of bi-variate analysis is to find out the relationship betweet two variables. We look for a pattern between any pair of variables so firstly, we are going to repreent each pair of features in scatter plots.

We want to forecast :

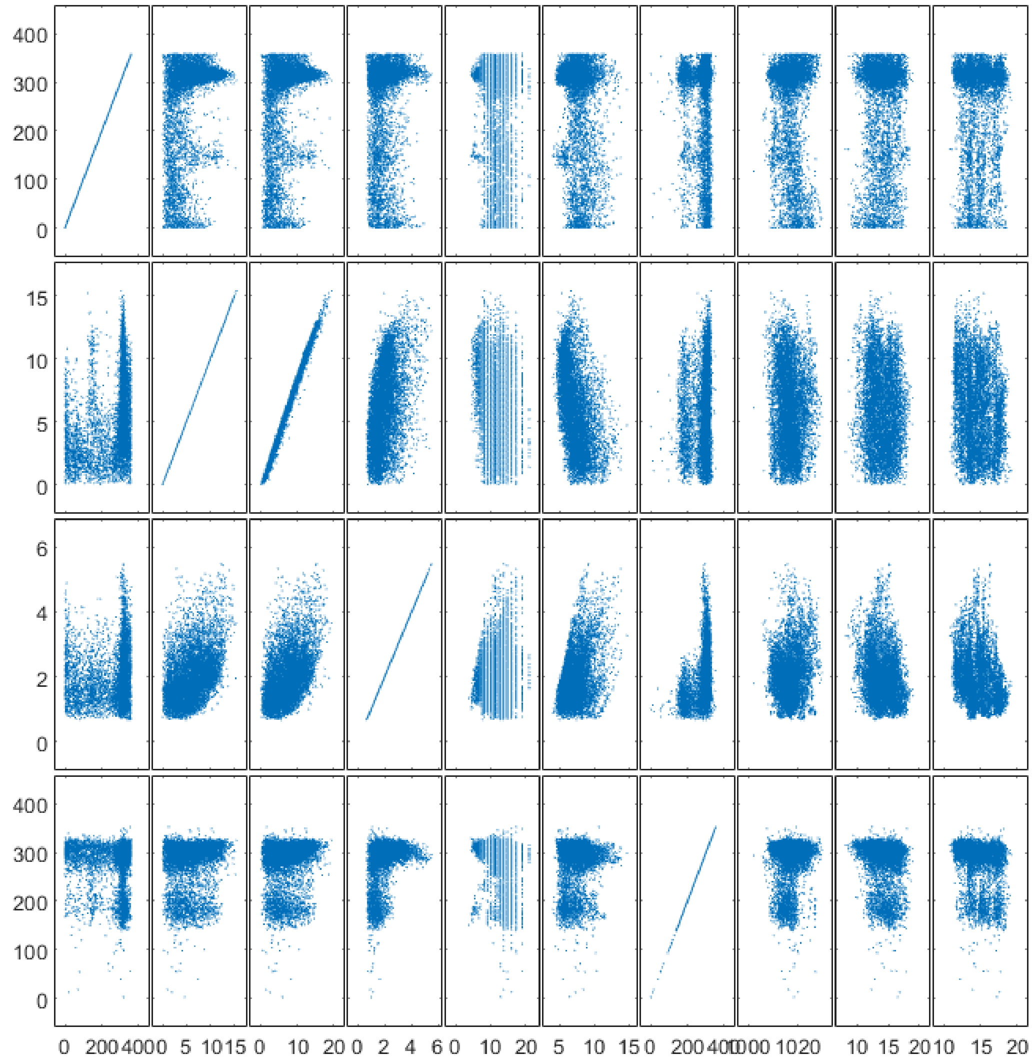
- Wind speed
- Wind direction
- Waves direction
- Significant Wave Height

These variables will be targets in models constructed to predict them.This lead us to analize the correlation etween each of these variables with the rest one. If a variable has little or no correlation with the variable target studied, it won't be cnsidered as input for target predictor model.

For Correlation Analysis we will use:

- Scatter plots to analyze correlation between a pair of varibles
- Heatmap

```
clf
figCorrelation1 = plotmatrix(table2array(data2018(:, {'WDIR','WSPD', 'GST','WVHT','DPD','APD','
```




```
figCorrelation1 =
  4×10 Line array:

    Line    Line    Line    Line    Line    Line    Line    Line    Line    Line
    Line    Line    Line    Line    Line    Line    Line    Line    Line    Line
    Line    Line    Line    Line    Line    Line    Line    Line    Line    Line
    Line    Line    Line    Line    Line    Line    Line    Line    Line    Line
```

## Wind and Waves direction univariate analysis

To analyse waves and wind direction data, we first need to convert degrees angles to radians because arguments functions are in radians

```
windDir2018 = circ_ang2rad(data2018.WDIR)
```

```
Unrecognized function or variable 'circ_ang2rad'.
```

```
wavesDir2018 = circ_ang2rad(data2018.MWD)
```

```
wdir_cm.mean = circ_mean(windDir2018)
```

```
wdir_cm = struct with fields:
    mean: -0.6936
```

```
mwd_cm.mean = circ_mean(wavesDir2018)
```

```
mwd_cm = struct with fields:
    mean: -1.1789
```

```
wdir_cm.median = circ_median(windDir2018)
```

```
wdir_cm = struct with fields:
      mean: -0.6936
    median: -0.7330
```

```
mwd_cm.median = circ_median(wavesDir2018)
```

```
mwd_cm = struct with fields:
      mean: -1.1789
    median: -1.0123
```

Let's calculate the length of mean vector. The closer is to 1, the more concentrted is data around the mean

```
wdir_cm.R = circ_r(windDir2018)
```

```
wdir_cm = struct with fields:
      mean: -0.6936
    median: -0.7330
         R: 0.6936
```

```
mwd_cm.R = circ_r(wavesDir2018)
```

```
mwd_cm = struct with fields:
      mean: -1.1789
    median: -1.0123
         R: 0.7846
```

```
wdir_sm.variance = circ_var(windDir2018)
```

```
wdir_sm = struct with fields:
    variance: 0.3064
```

```
mwd_sm.variance = circ_var(wavesDir2018)
```

```
mwd_sm = struct with fields:
    variance: 0.2154
```

```
wdir_sm.std = circ_std(windDir2018, [], [],'default' )
```

```
wdir_sm = struct with fields:
    variance: 0.3064
         std: 0.7828
```

```
mwd_sm.std = circ_std(wavesDir2018,[], [],'default')
```

```
mwd_sm = struct with fields:
    variance: 0.2154
         std: 0.6564
```

```
wdir_spm.centralMoment = circ_moment(windDir2018,[],0)
```

```
wdir_spm = struct with fields:
    centralMoment: 1
```

```
mwd_spm.centralMoment = circ_moment(wavesDir2018,[],0)
```

```
mwd_spm = struct with fields:
    centralMoment: 1
```

```
wdir_spm.skewness = circ_skewness(windDir2018)
```

```
wdir_spm = struct with fields:
    centralMoment: 1
         skewness: -0.0233
```

```
mwd_spm.skewness = circ_skewness(wavesDir2018)
```

```
mwd_spm = struct with fields:
    centralMoment: 1
         skewness: 0.2847
```

### Probability distributions

```
[mu kappa] = circ_vmpar(windDir2018)
```

```
Unrecognized function or variable 'windDir2018'.
```

```
[mu2 kappa2] = circ_vmpar(wavesDir2018)
```

```
[wdir_pdf anglesWind] = circ_vmpdf(windDir2018, mu, kappa)
```

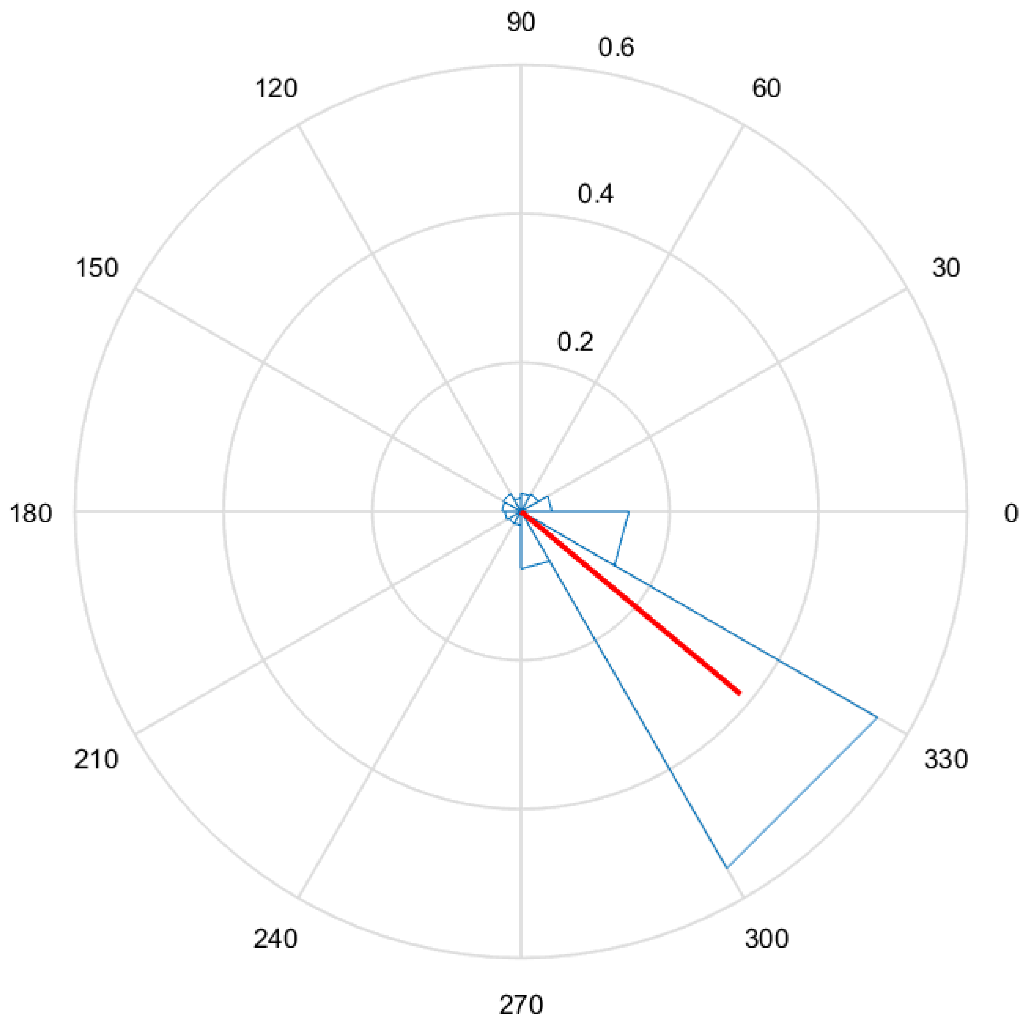
```
wdir_pdf = 8587×1
    0.1159
    0.2101
    0.3667
    0.1759
    0.2480
    0.5074
    0.4468
    0.4979
    0.4993
    0.4652
      ⋮

anglesWind = 8587×1
```

```
    0.6283
    0.2967
    6.1785
    0.4014
    0.1920
    5.6723
    5.2185
    5.4280
    5.7421
    5.8992
      ⋮
```

```
[waves_pdf angleWaves] = circ_vmpdf(wavesDir2018,mu2,kappa2)
```

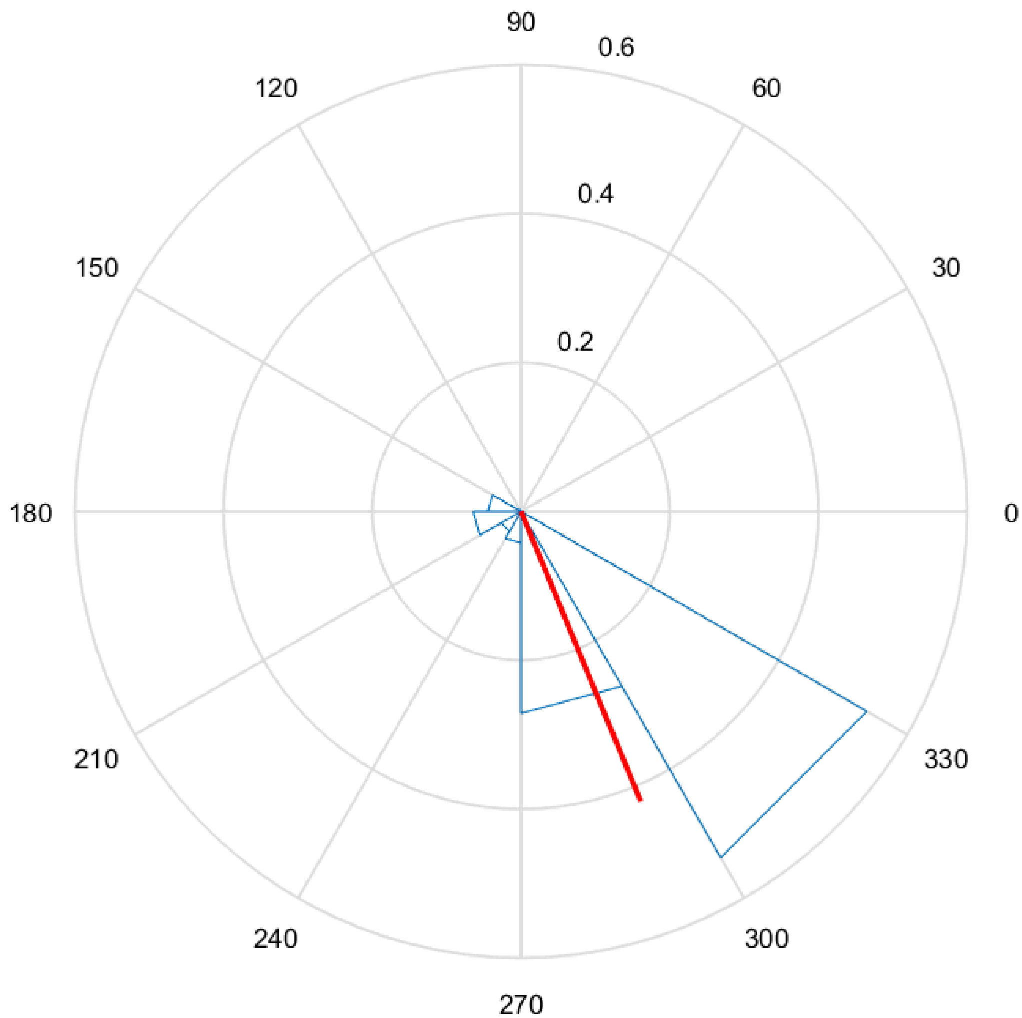
```
waves_pdf = 8587×1
    0.5353
    0.5272
    0.5970
    0.6132
    0.4735
    0.6097
    0.5879
    0.4639
    0.4639
    0.5014
      ⋮

angleWaves = 8587×1
    4.7822
    4.7647
    4.9567
    5.0615
    4.6600
    5.0265
    4.9218
    4.6426
    4.6426
    4.7124
      ⋮
```

```
clf
circ_plot(windDir2018,'hist',[], 12,true,true,'linewidth',2,'color','r')
```



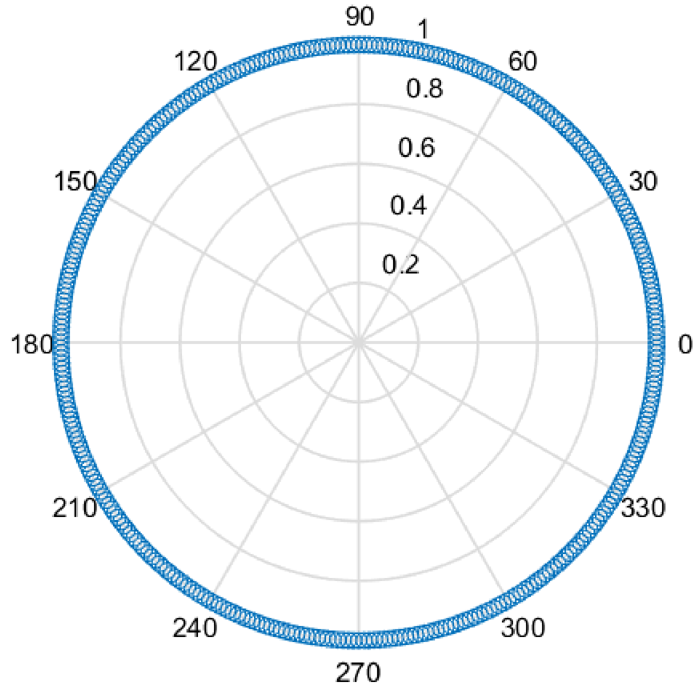

```
ans =
  Axes with properties:

             XLim: [-0.6000 0.6000]
             YLim: [-0.6900 0.6900]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Show all properties
```

```
clf
circ_plot(wavesDir2018,'hist',[], 12,true,true,'linewidth',2,'color','r')
```



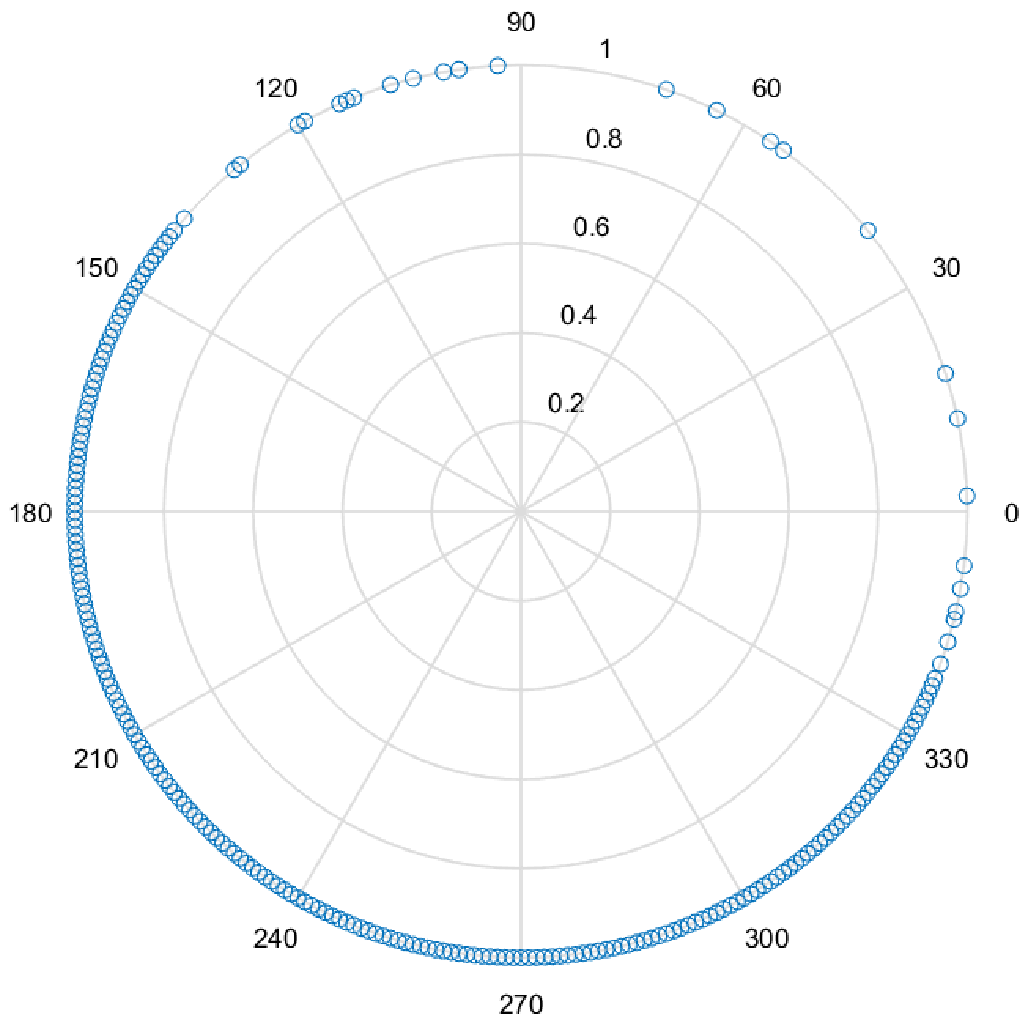

```
ans = 
  Axes with properties:

             XLim: [-0.6000 0.6000]
             YLim: [-0.6900 0.6900]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Show all properties
```

```
circ_plot(windDir2018)
```




```
ans = 
  Axes with properties:

             XLim: [-1 1]
             YLim: [-1.1500 1.1500]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Show all properties
```

```
circ_plot(wavesDir2018)
```




```
ans =
  Axes with properties:

             XLim: [-1 1]
             YLim: [-1.1500 1.1500]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Show all properties
```

**Wind direction and Waves direction features follow the Von Mises distribution also called "circular normal distribution"**

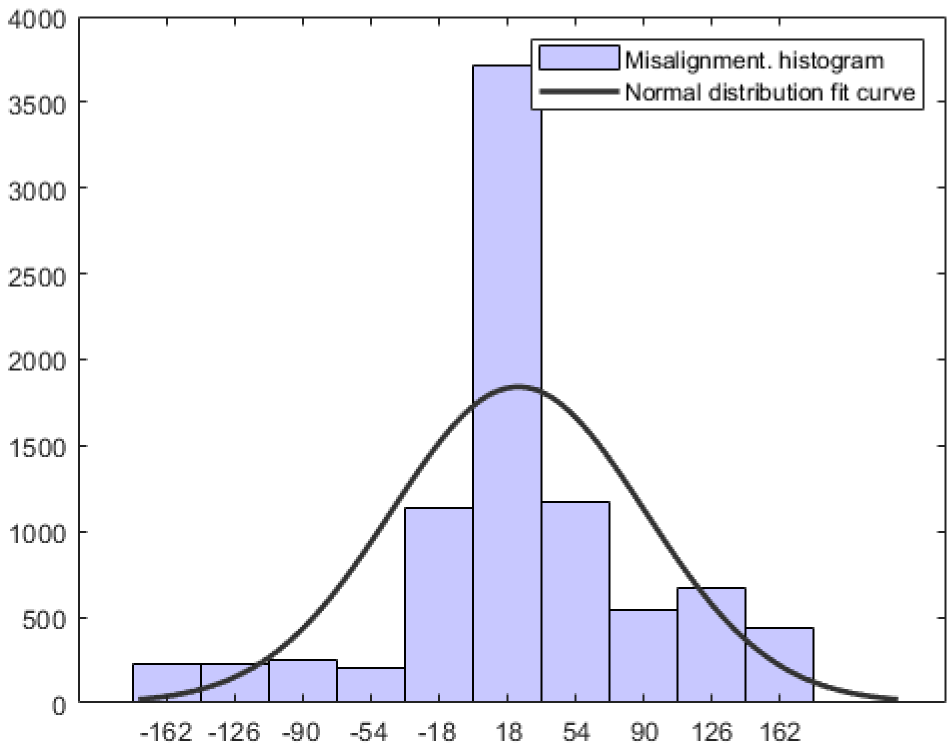
```
h32 = histfit(data2018.MIS, 10,'normal')
```

```
h32 =
  2×1 graphics array:

  Bar
  Line
```

```
h32(1).FaceColor = [.8 .8 1];
h32(2).Color = [.2 .2 .2];
h32(2).DisplayName = "Normal distribution"
```

```
h32 =
  2×1 graphics array:

  Bar
  Line    (Normal distribution)
```

```
legend ("Misalignment. histogram", "Normal distribution fit curve")
```




## Sources:

Exploratory Data Analysis - Mathwork

https://www.mathworks.com/help/stats/example.html?s_tid=srchtitle