

Lista 1 - Análise de Séries Temporais em Oceanografia

Lucas Salimene

Parte I – Visualização e edição dos dados oceanográficos

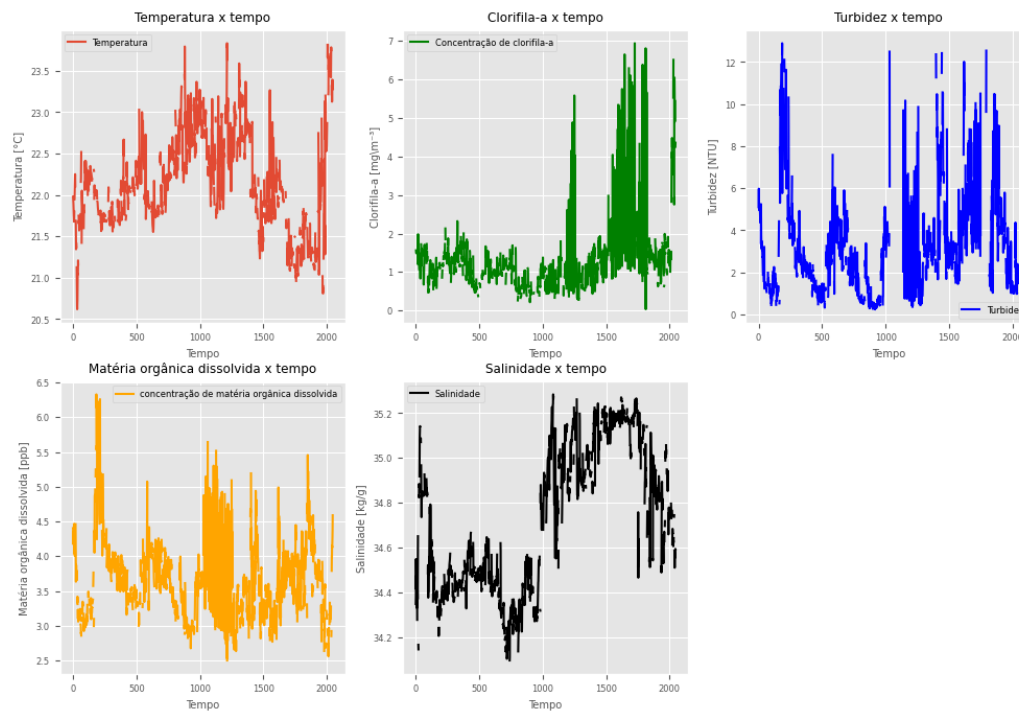


Figura 1: Variáveis disponíveis no dado do SIMCOSTA

Conforme se observa na figura 1, existe a necessidade de interpolação dos dados, pois existem lacunas nos mesmos.

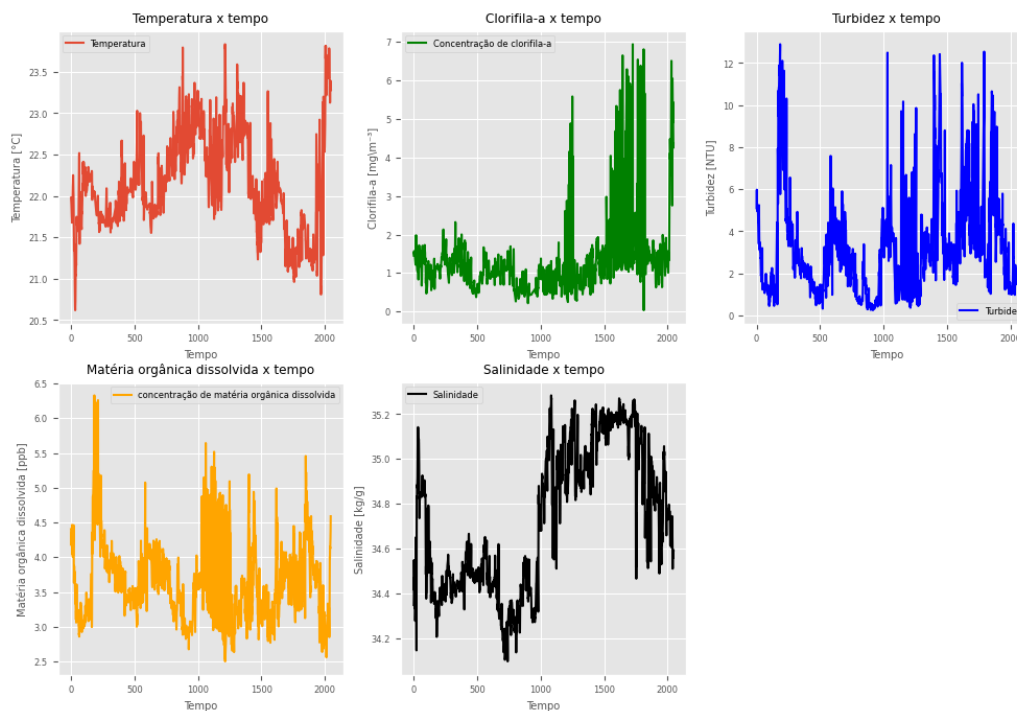


Figura 2: Variáveis disponível no dado do SIMCOSTA após interpolação linear

A figura 2 mostra os dados após uma interpolação linear utilizando o pacote pandas do Python.

```
pandas.DataFrame.interpolate
```

É notável que mesmo após a interpolação os dados apresentam pontos que não representam o comportamento da série e podem ter ocorrido devido a algum erro/processo que não seja do interesse. Com isso surge a necessidade da utilização de uma edição na série temporal.

Utilizando o *wild edit*, foi realizada a edição dos dados como pode ser visto na figura 3

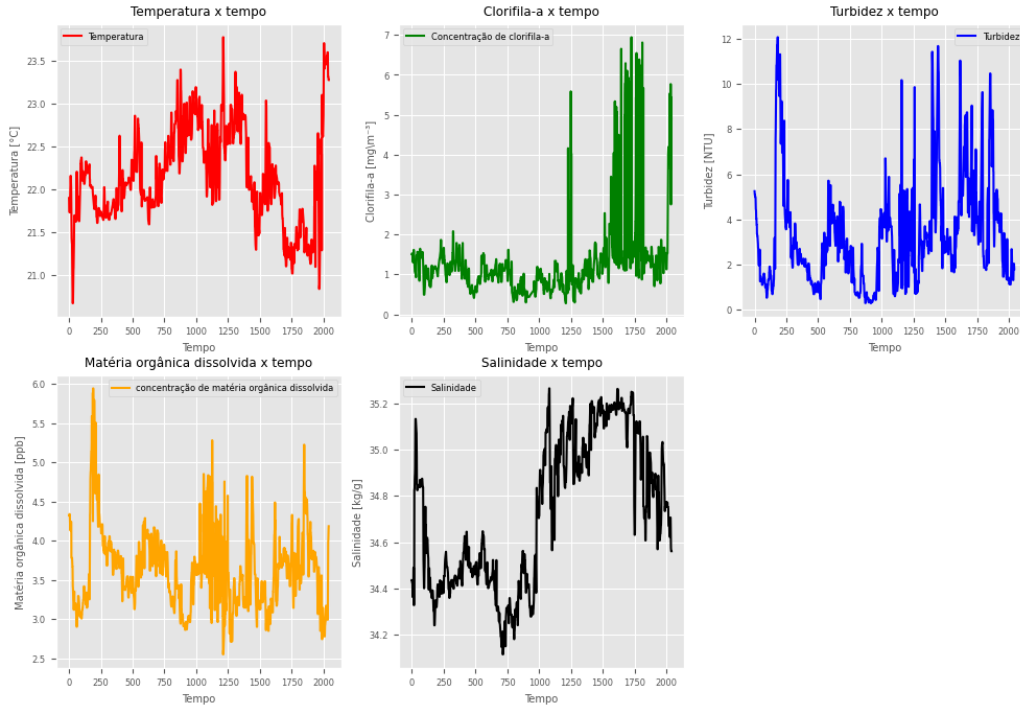


Figura 3: Variáveis editadas pelo conceito de *wild edit*

Os *spikes* foram removidos de forma que a série temporal original não seja muito afetada, com esse método funcionando com uma especie de janelamento dos dados, a figura 4 mostra uma comparação dos dados originais com os editados para a salinidade, onde é visível que o padrão da série se mantém com apenas os *spikes* sendo removidos, esse método pode induzir a erros que não facilmente detectáveis devido a forma que ele é feito. Dependendo da análise a ser realizada, outras edições ou diferentes métodos possam ser necessários.

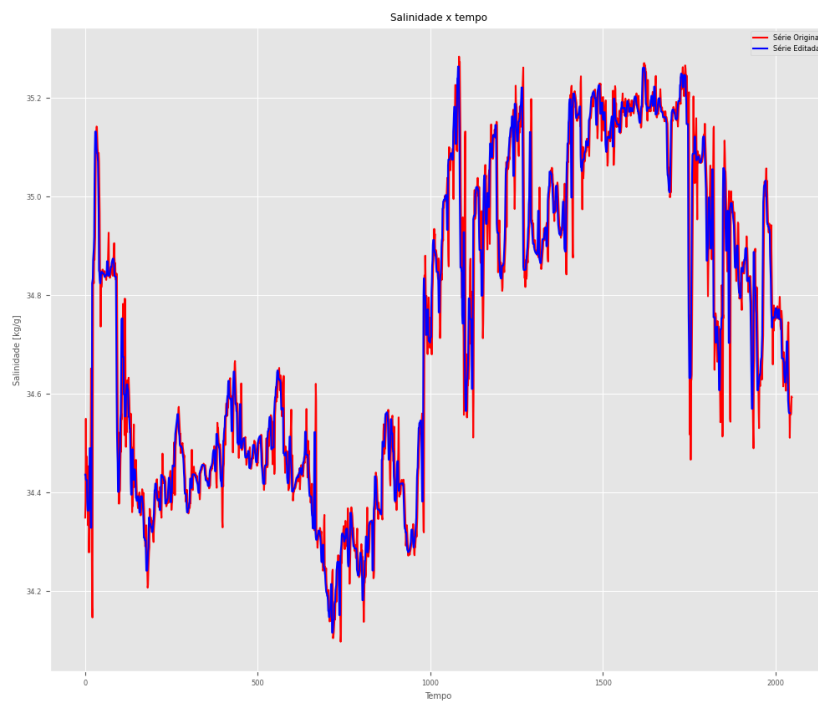


Figura 4: Comparação da salinidade sem edição e com edição

Parte II – Estatística básica

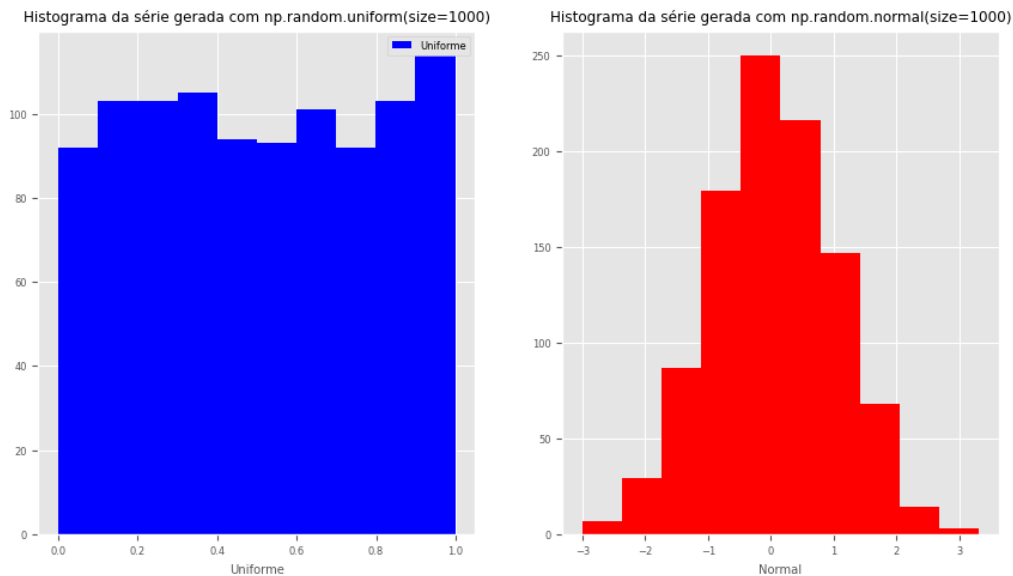


Figura 5: Histograma de funções aleatórias geradas com o modo uniforme e normal respectivamente

	Média	Desvio Padrão
Uniforme	0.506	0.291
Normal	0.035	0.98

Tabela 1: Média e desvio padrão para cada série aleatória

Devido a forma que as distribuições estatísticas são calculadas, os valores de média e desvio padrão estão dentro do esperados.

Para um sinal aleatório com distribuição normal com média de 14.7 e desvio padrão de 4.7, se pode utilizar a função da biblioteca NumPy para Python:

```
temprand = np.random.normal(loc=14.2, scale=4.7, size=10000)
```

Com isso se pode calcular o z-score com o pacote SciPy:

```
zscores = stats.zscore(temprand)
```

A média e o desvio padrão do z-score é apresentado na tabela 2

Média	Desvio Padrão
0	1

Tabela 2: Média e desvio padrão do z-score

A figura 6 apresenta o histograma do z-score

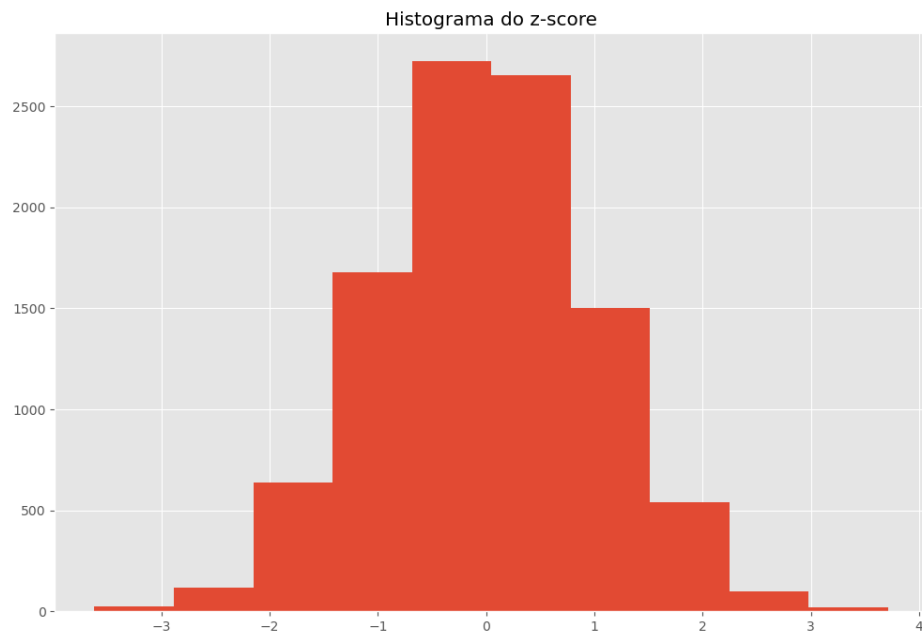


Figura 6: Histograma do z-score para uma função aleatória normal

Com os resultados apresentando o comportamento esperado para a padronização realizada. O gráfico de probabilidade pode ser feito utilizando a função:

```
probplot = stats.probplot(zscores, plot=plt)
```

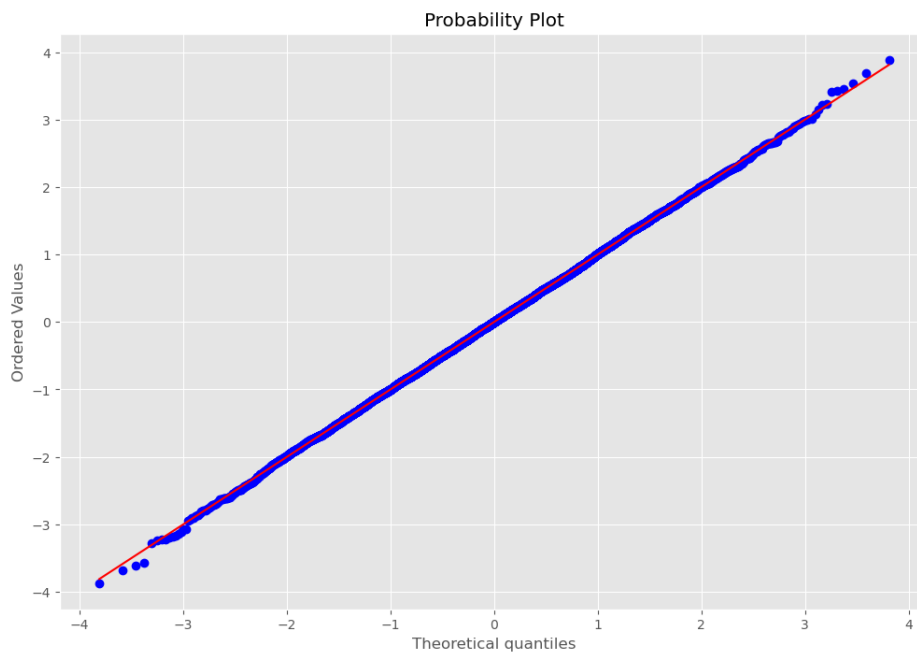


Figura 7: Gráfico de probabilidade normal da série

Pela figura 7, se observa que os dados gerados aleatoriamente e normalizados utilizando o z-score seguem a distribuição normal, com praticamente todos os pontos próximos a linha de regressão.

A função de densidade de probabilidade normal da série x variando de 0 a 30 pode ser calculada com:

```
a30 = np.arange(start=0, stop=30, step=0.1)
a30pdf = stats.norm.pdf(a30, 14.2, 4.7)
```

Onde a função `stats.norm.pdf()` calcula a densidade de probabilidade normal de x variando entre 0 e 30 com intervalo 0.1

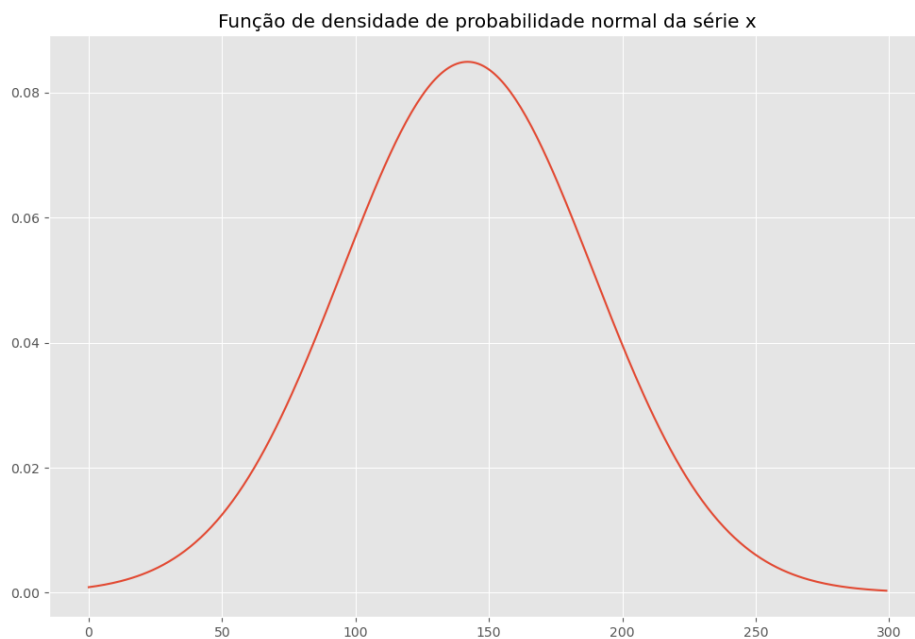


Figura 8: Função de densidade de probabilidade normal da série x

O z-score será calculada da mesma forma, com os resultados para a média e desvio apresentados na tabela ??szsorea30

Média	Desvio Padrão
0	1

Tabela 3: Média e desvio padrão do z-score da função de densidade de probabilidade normal

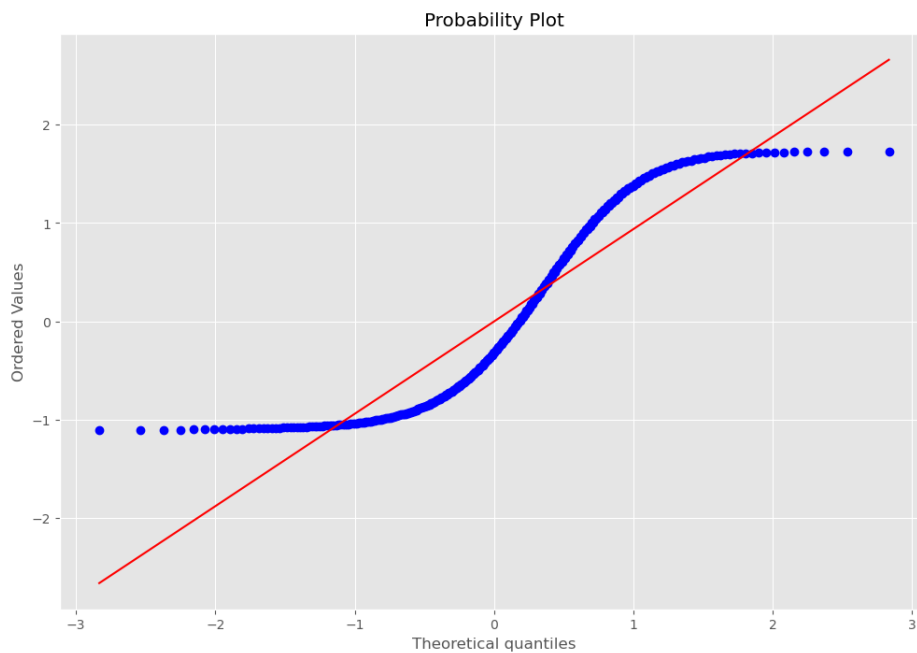


Figura 9: Gráfico de probabilidade normal da série

Pela figura 9 é possível ver que embora exista um padrão na orientação dos pontos com a linha de regressão, eles aparentam ter um comportamento não-linear em relação com a probabilidade normal, indicando que não seguem totalmente a distribuição normal.

O valor acumulado da probabilidade normal da série x variando de 0 a 30 pode ser calculada com:

```
a30cdf=stats.norm.cdf(a30,14.2,4.7)
```

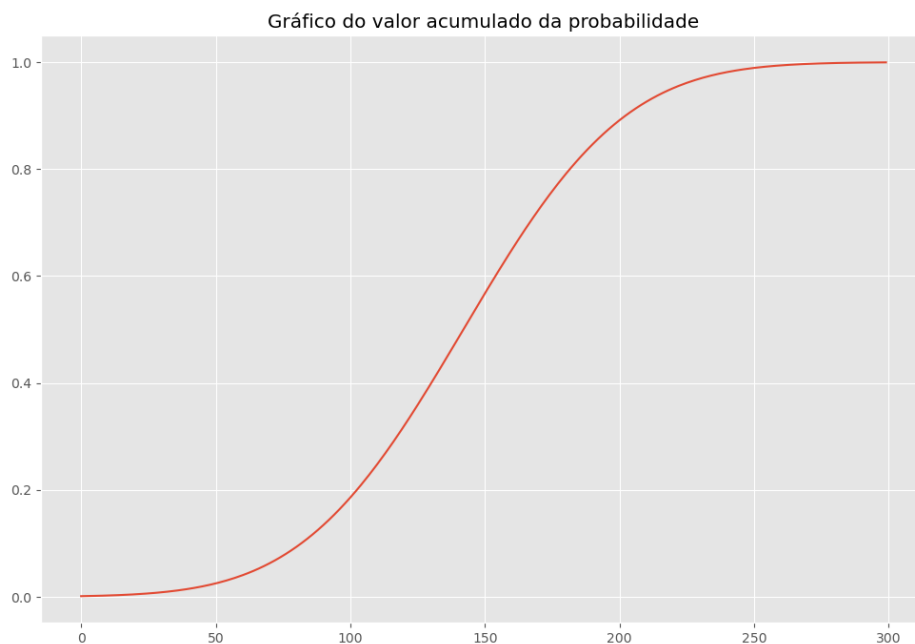


Figura 10: Gráfico do valor acumulado da probabilidade

A probabilidade de encontrar um valor de $x \leq 3$ é de 0.851% já para um valor de $x \geq 20$ é de 10.85%.

Utilizando a função:

```
_, pvalue = sm.lilliefors(a30pdf, dist='norm')
```

Se obtém um valor de 0.001, ou seja, a hipótese nula não pode ser rejeitada e os dados são originários de uma distribuição normal.