



**ANÁLISE DE SÉRIES TEMPORIAS DO CONSUMO DE ENERGIA ELÉTRICA NO
SETOR INDUSTRIAL**

Guilherme Teixeira Santos

Prof. Dr. Manoel Ivanildo Silvestre Bezerra

Presidente Prudente

2021

Guilherme Teixeira Santos

**ANÁLISE DE SÉRIES TEMPORIAS DO CONSUMO DE ENERGIA ELÉTRICA NO
SETOR INDUSTRIAL**

Trabalho desenvolvido ao Curso de
Graduação em Estatística da FCT/Unesp
para aproveitamento na disciplina de
Análise de Séries Temporais.

Professor: Drº Manoel Ivanildo Silvestre
Bezerra

Presidente Prudente

2021

Sumário

1. INTRODUÇÃO.....	4
1.1 Objetivos.....	4
2. ANÁLISE DA SÉRIE	5
2.2 Análise gráfica e transformações.....	5
2.2 Avaliação dos pressupostos	6
2.3 Identificação dos modelos	7
2.4 Estimativas dos parâmetros	10
3. PREVISÃO.....	12
REFERÊNCIAS	15

1. INTRODUÇÃO

A energia elétrica no Brasil tem dois tipos de fornecimento e é composta por energias renováveis e não renováveis. A Eletrobras contribui para a promoção de estudos, projetos de construção e operação de usinas de geração, linhas de transmissão e distribuição de energia elétrica, assim como também gera contratos com comércio que partem dessas atividades, como a comercialização de energia elétrica. O consumo é utilizado por pessoas físicas ou jurídicas, que formam uma unidade ou um conjunto de unidades que necessitam de energia diariamente. Esta classe de consumo industrial é denominada unidade consumidora na qual atividade industrial é desenvolvida, e é definida na Classificação Nacional de Atividades Econômicas (CNAE).

1.1 Objetivos

O objetivo deste trabalho é fazer uma análise da série temporal do consumo de energia elétrica industrial no Brasil. Com as ferramentas adequadas, será possível realizar análises gráficas e também as devidas transformações para fins de ajustamento dos dados, em conjunto, os testes e as validações dos pressupostos deverão ser feitos para estimar os modelos. Por fim, os dois melhores modelos serão utilizados para prever o consumo de energia entre julho e dezembro de 2020 e avaliar qual o menor erro estimado afim de compará-los.

1.2 Ferramentas

A única ferramenta utilizada será o Python pois possui pacotes estatísticos que nos permitem fazer uma análise completa da série em questão.

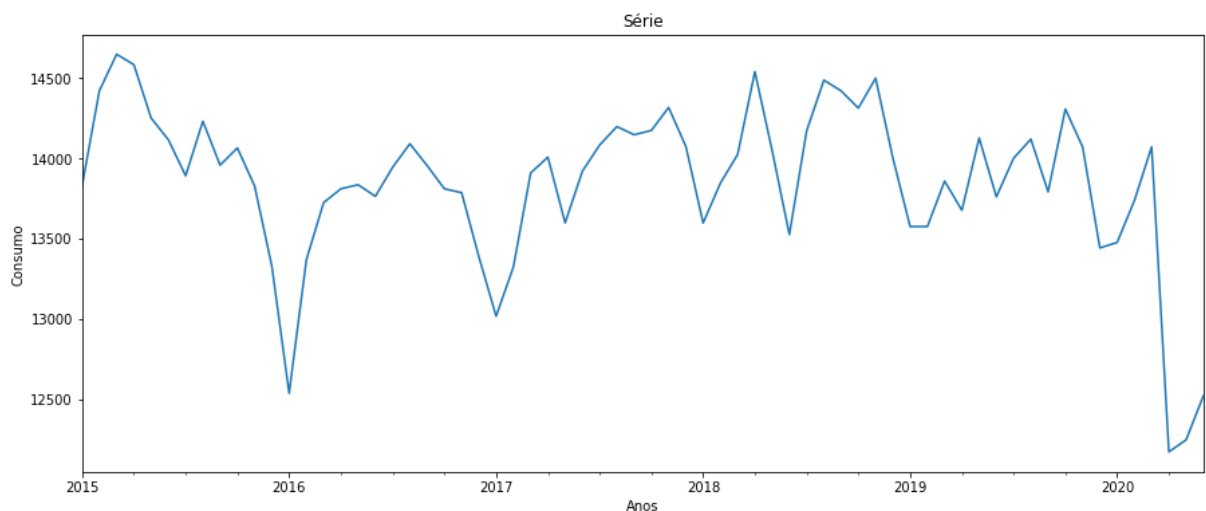
2. ANÁLISE DA SÉRIE

Os dados acessados do IPEA apresentam o consumo de energia entre janeiro de 2015 e março de 2021, totalizando 75 linhas. Nesta análise foram tomadas 67 observações, ou seja, o consumo de energia até junho de 2020. Para que fosse possível um bom funcionamento dos pacotes no Python, as informações foram indexadas com o comando `set_index`.

Primeiramente, é importante analisarmos o comportamento dos dados no tempo e validarmos importantes pressupostos.

2.2 Análise gráfica e transformações

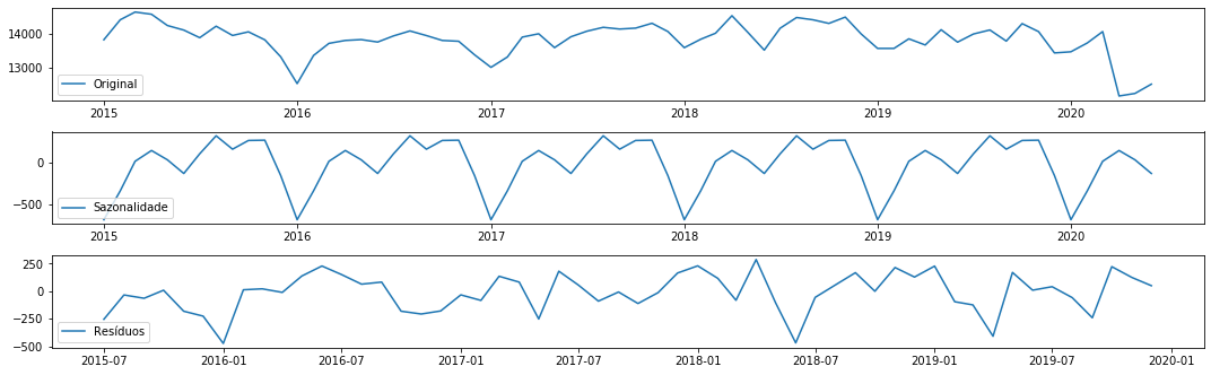
Figura 1. Gráfico da série original



Na figura 1, a série aparenta não haver tendência, mas é possível notar uma sazonalidade com periodicidade de 12 meses. Pode-se também notar uma brusca queda no consumo de energia todo início de ano.

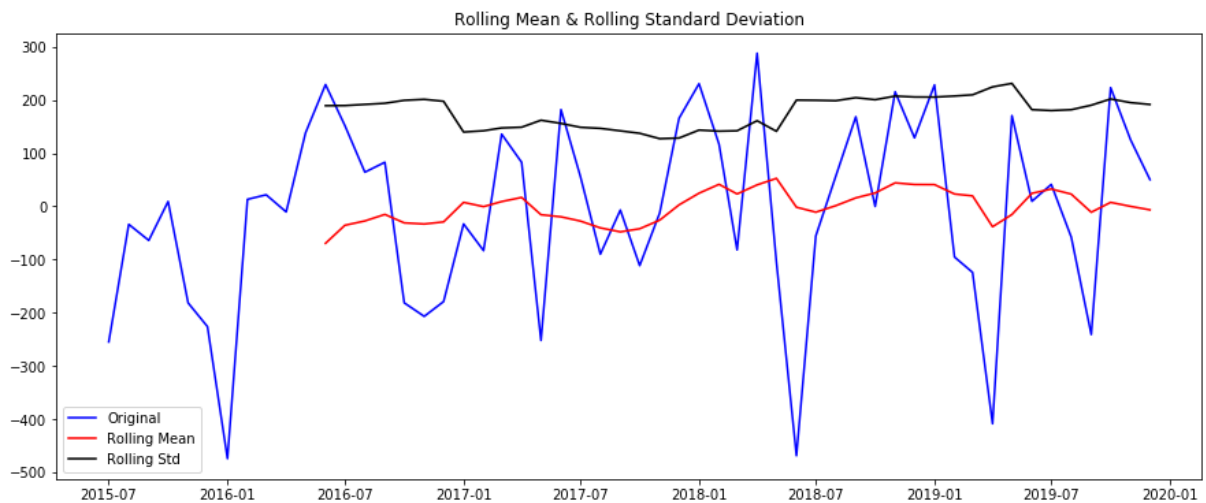
Com as questões levantadas, devemos aplicar a função de decomposição para retirar a sazonalidade e modelar os resíduos.

Figura 2. Gráfico da série original, da decomposição sazonal e residual



Após aplicarmos a decomposição residual, foi possível retirar parte da sazonalidade, pois a mesma fica ainda um pouco evidente, o que podemos ver na figura 3. Um ponto positivo são as médias e desvios móveis que se mostraram constantes ao longo do tempo, tendo fortes indícios de estacionariedade.

Figura 3. Gráfico dos resíduos decompostos



2.2 Avaliação dos pressupostos

Para que possamos prosseguir com a modelagem dos dados, um indispensável pressuposto é o de estacionariedade. Com os testes de Dickey Fuller Aumentado e Kwiatkowski-Phillips-Schmidt-Shin (KPSS) iremos comprovar ou não os processos estacionários.

Tabela 1. Teste de Dickey-Fuller Aumentado

Augmented Dickey-Fuller Test
ADF Statistic: -5.910
p-value < 0.001
Critical Values:
10% : -2.597
5% : -2.918
1% : -3.560
Result: The series is stationary

Tabela 2. Teste de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

KPSS Test
ADF Statistic: 0.339
p-value: 0.1
Critical Values:
1%: 0.739
5%: 0.463
10%: 0.347
Result: The series is stationary

Os testes de Dickey-Fuller e KPSS fornecem diferentes tipos de interpretação, mas a conclusão para os testes aplicados é a mesma.

Com um p-valor menor que 0.001 no Dickey-Fuller, não rejeitamos o pressuposto de estacionariedade, com 99% de confiança.

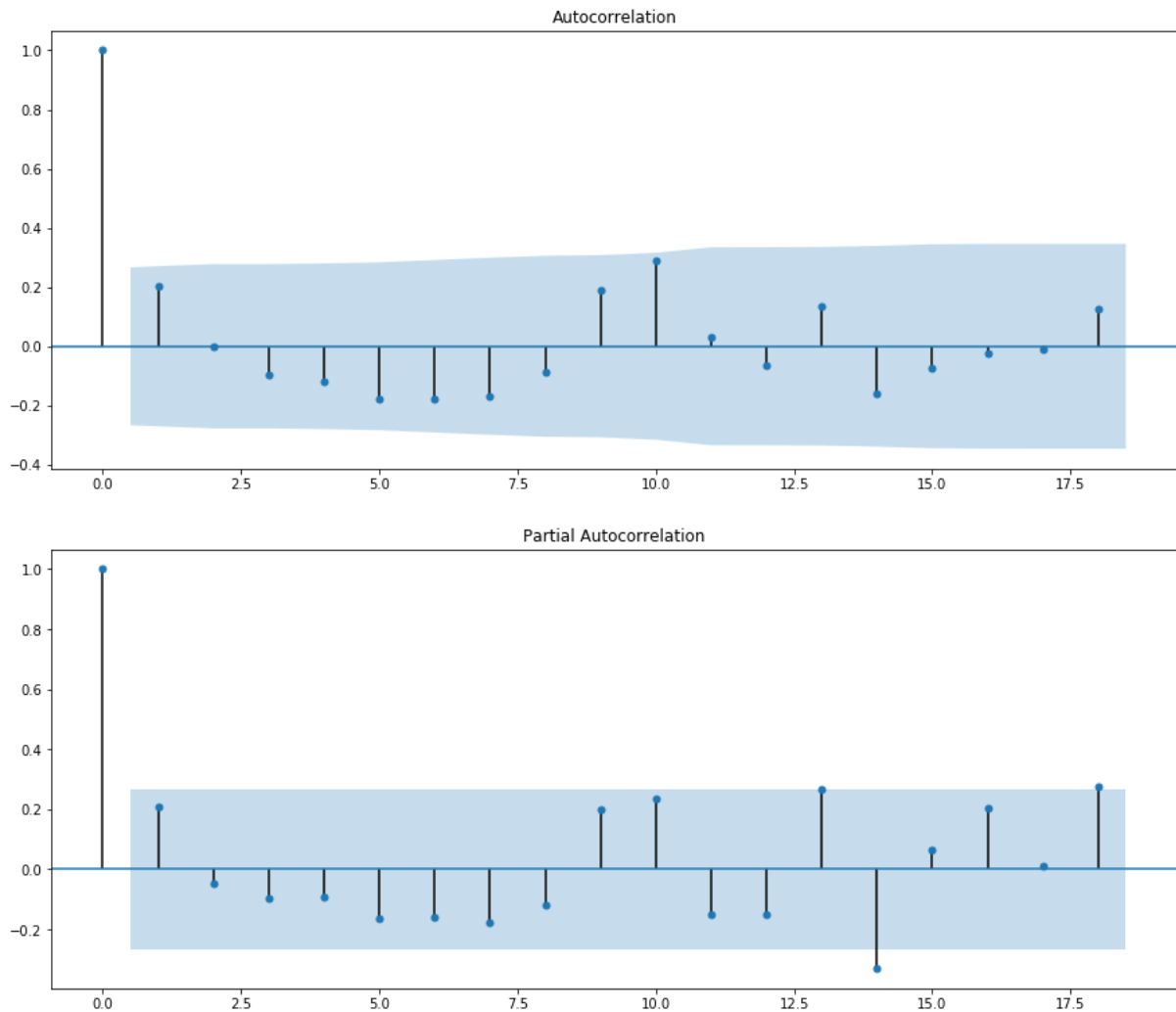
No KPSS foi obtido um p-valor de 0.1. Neste caso, como é maior que 0.05, temos também que a série é estacionária.

Com as proposições já comprovadas, podemos dar continuidade e identificar os modelos.

2.3 Identificação dos modelos

Através dos dados, podemos observar dois tipos de modelos, ARIMA e SARIMA. Iremos então, por meio dos resíduos decompostos analisar as autocorrelações e autocorrelações parciais para encontrar os parâmetros do modelo ARIMA.

Figura 4. Gráficos das autocorrelações e autocorrelações parciais dos resíduos

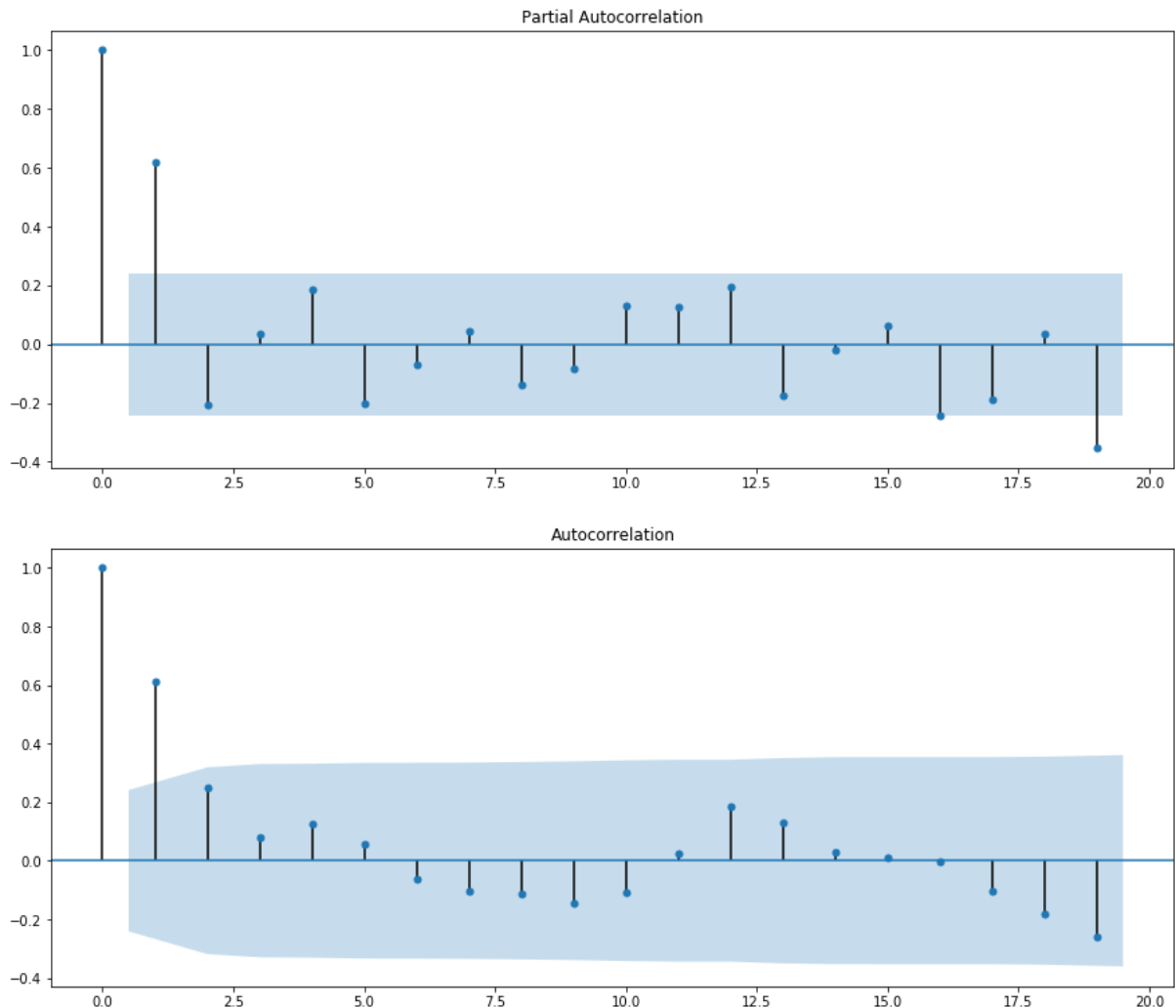


Analisando o gráfico das autocorrelações (AC), podemos notar que os “lags” apresentam ondas senoidais. No gráfico das autocorrelações parciais (ACP) vemos que o primeiro “lag” é significativo.

Essas observações apresentam características de um modelo ARIMA (0,0,1), que é também um MA (1).

No caso do modelo SARIMA, foram utilizados os dados sem a decomposição residual para a construção dos gráficos das autocorrelações.

Figura 5. Gráficos das autocorrelações e autocorrelações parciais



Podemos definir as ordens do modelo SARIMA através da Figura 5 e conjuntamente, também utilizar os valores dos parâmetros já definidos anteriormente do modelo ARIMA (0,0,1). Através do gráfico PACF é possível definir $p=1$, pois temos o primeiro “lag” acima do limite e com o gráfico ACF é possível definir $q=1$. Assim, o modelo para a sazonalidade pode ser dado por um ARIMA (1,0,1), ou ARMA (1,1).

Essas análises sugerem um modelo SARIMA (1,0,1)(0,0,1)₁₂, porém, a função `auto_arima` do Python indicou este modelo como o segundo melhor, tendo AIC=972.130. O melhor modelo fornecido pelo Python foi o SARIMA (1,0,0)(0,0,1)₁₂ com AIC=970.920, que posteriormente apresentou um erro quadrático médio também menor nas previsões.

Com isso, os modelos utilizados a seguir para as devidas análises, testes e previsões será o ARIMA (0,0,1) e o SARIMA (1,0,0)(0,0,1)₁₂.

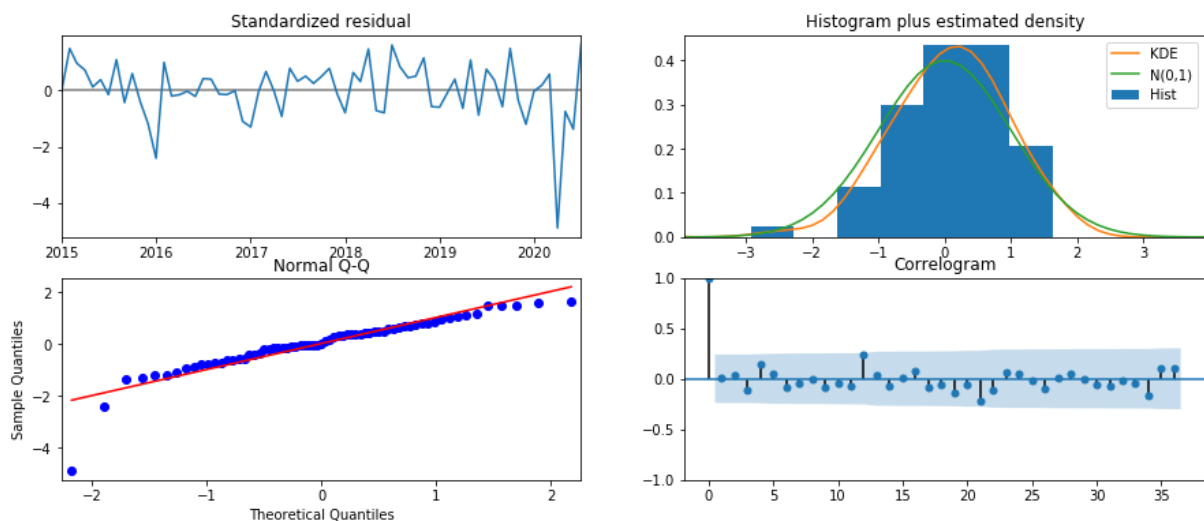
2.4 Estimativas dos parâmetros

Modelo ARIMA (0,0,1)

Figura 6. Estimativas e sumarização do modelo

SARIMAX Results						
Dep. Variable:		Consumo		No. Observations:		66
Model:		ARIMA(0, 0, 1)		Log Likelihood		-489.377
Date:		Sun, 01 Aug 2021		AIC		984.755
Time:		15:05:41		BIC		991.324
Sample:		01-01-2015		HQIC		987.351
- 06-01-2020						
Covariance Type:		opg				
	coef	std err	z	P> z	[0.025	0.975]
const	1.384e+04	103.057	134.339	0.000	1.36e+04	1.4e+04
ma.L1	0.5954	0.115	5.181	0.000	0.370	0.821
sigma2	1.6e+05	1.97e+04	8.134	0.000	1.21e+05	1.99e+05
Ljung-Box (Q):		49.53	Jarque-Bera (JB):		108.10	
Prob(Q):		0.14	Prob(JB):		0.00	
Heteroskedasticity (H):		2.36	Skew:		-1.71	
Prob(H) (two-sided):		0.05	Kurtosis:		8.25	

Figura 7. Gráfico dos resíduos padronizados, histograma dos resíduos, normal Q-Q plot e correlograma



Na Figura 6, temos uma forma muito bem resumida das estimativas dos parâmetros e também dos pressupostos necessários para uma previsão consistente. Pode-se notar que os parâmetros são significativos, na Figura 7 vemos os resíduos padronizados variando em torno

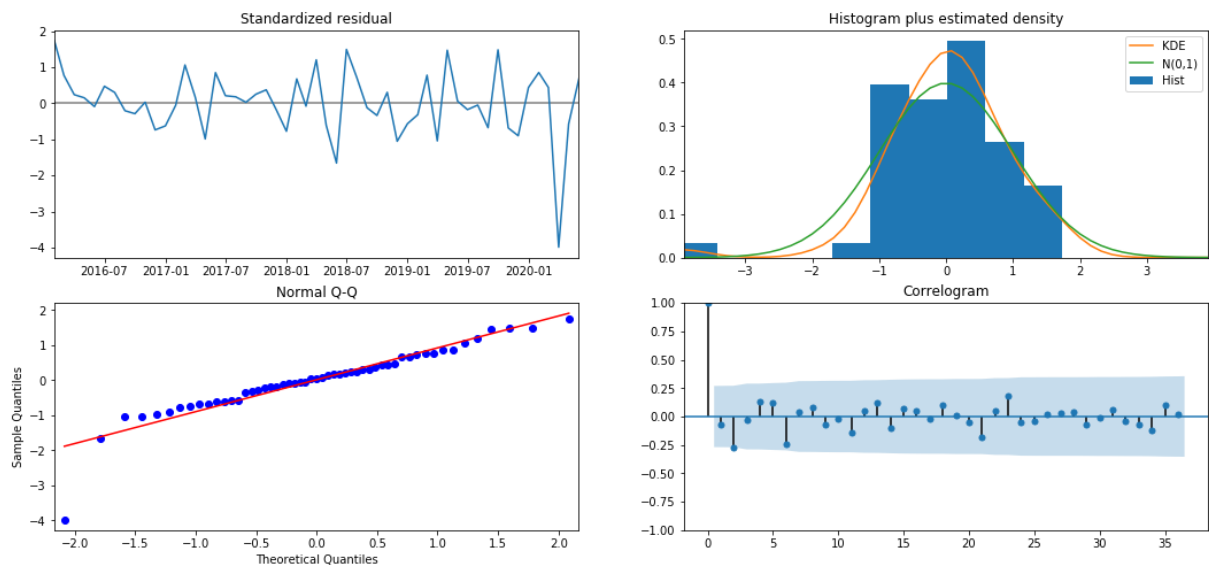
de zero e também apresentando uma distribuição aproximadamente normal no histograma, com todos os resíduos contidos no limite do correlograma.

Modelo SARIMA (1,0,0)(0,0,1)₁₂

Figura 8. Estimativas e sumarização do modelo

SARIMAX Results						
Dep. Variable:	Consumo		No. Observations:	66		
Model:	SARIMAX(1, 0, 0)x(0, 0, [1], 12)		Log Likelihood	-391.625		
Date:	Sun, 01 Aug 2021		AIC	789.250		
Time:	14:41:47		BIC	795.161		
Sample:	01-01-2015		HQIC	791.523		
- 06-01-2020						
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.9985	0.008	127.845	0.000	0.983	1.014
ma.S.L12	0.5472	0.304	1.797	0.072	-0.050	1.144
sigma2	1.697e+05	4.58e+04	3.702	0.000	7.98e+04	2.6e+05
Ljung-Box (Q):	36.46	Jarque-Bera (JB):	75.41			
Prob(Q):	0.63	Prob(JB):	0.00			
Heteroskedasticity (H):	3.27	Skew:	-1.36			
Prob(H) (two-sided):	0.02	Kurtosis:	8.17			

Figura 9. Gráfico dos resíduos padronizados, histograma dos resíduos, normal Q-Q plot e correlograma



Na Figura 8, vemos que somente o parâmetro MA (1) não é significativo, mas da mesma forma que o modelo ARIMA (0,0,1), fica evidente graficamente que os resíduos seguem uma distribuição aproximadamente normal.

Tabela 1. Testes de Box-Pierce e Box-Ljung

ARIMA (0,0,1)		
Lags=36	Box-Pierce	Box-Ljung
Statistics	30.819385	43.166905
P-Values	0.713253	0.191672

Tabela 2. Testes de Box-Pierce e Box-Ljung

SARIMA (1,0,0)(0,0,1) ₁₂		
Lags=36	Box-Pierce	Box-Ljung
Statistics	1.112159	1.524683
P-Values	1.0	1.0

Os testes de Box-Pierce nos fornecem p-valores todos acima de 0.05, o que comprova que de fato os resíduos tem distribuição ruído branco. o Box-Ljung também apresentou p-valores acima de 0.05 e com isso, temos que os resíduos são independentes e identicamente distribuídos.

3. PREVISÃO

A previsão dos meses de julho a dezembro de 2020 é realizada através do método “One-Step Ahead Forecast” utilizando o comando `get_prediction` do Python.

Tabela 3. Previsões do consumo em GWh para os 6 meses

Meses	Real	Previsão ARIMA com $IC_{95\%}$	Previsão SARIMA com $IC_{95\%}$
jul/2020	13864	13240 [12456 ; 14024]	12463 [11655 ; 13271]
ago/2020	14517	13844 [12932 ; 14756]	12434 [11292 ; 13576]
set/2020	14652	13844 [12932 ; 14756]	12263 [10866 ; 13661]
out/2020	15109	13844 [12932 ; 14756]	12578 [10966 ; 14190]
nov/2020	14672	13844 [12932 ; 14756]	12406 [10604 ; 14207]
dez/20	14589	13844 [12932 ; 14756]	12185 [10213 ; 14157]

Figura 10. Gráfico das previsões segundo o modelo ARIMA (0,0,1)

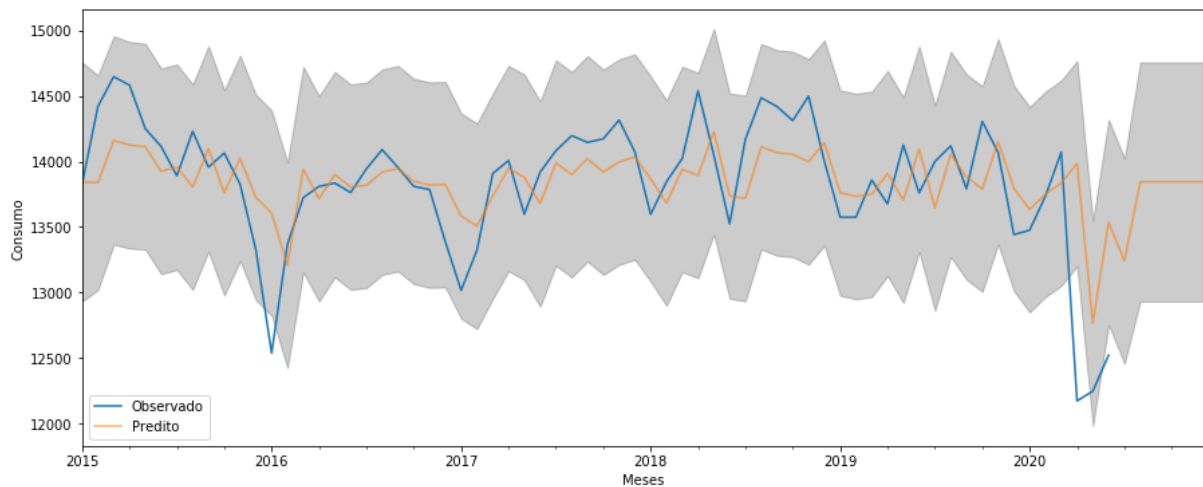
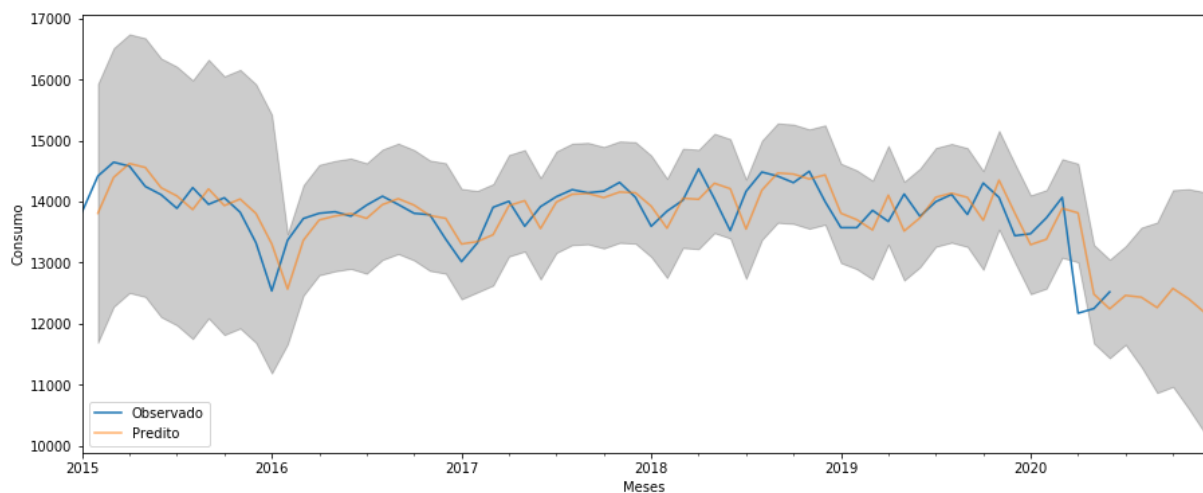


Figura 11. Gráfico das previsões segundo o modelo SARIMA (1,0,0)(0,0,1)₁₂



As Raízes do Erro Quadrático Médio Mínimo de Previsão (REQMMP) para os modelos são:

ARIMA= 849.53

SARIMA= 2210.42

Levando em conta todos os resultados apresentados, sabe-se que nenhum dos dois modelos violaram os pressupostos necessários para uma estimação adequada. O modelo ARIMA pôde prever melhor o consumo de energia nos meses subsequentes, com valores mais próximos dos reais, e na maioria das previsões, os intervalos de confiança cobriram os verdadeiros valores.

Por fim, também avaliamos o REQMMP que de fato evidencia um erro quadrático menor no modelo ARIMA e nos leva a crer que é o modelo mais ajustado para os dados do IPEA.

REFERÊNCIAS

pmdarima.arima.auto_arima. **pmdarima**. Disponível em: <https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html>. Acesso em 1 de ago. de 2021.

PULAGAGEM, Sushmitha. Time Series forecasting using Auto ARIMA in python. **Towards data science**, 2020. Disponível em: <<https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd>>. Acesso em 1 de ago. de 2021.

MAKLIN, Colin. ARIMA Model Python Example — Time Series Forecasting. **Towards data science**, 2019. Disponível em: <<https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7>>. Acesso em 1 de ago. de 2021.

PRABHAKARAN, Selva. ARIMA Model – Complete Guide to Time Series Forecasting in Python. **Machine learning plus**, 2019. Disponível em: <<https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>>. Acesso em 1 de ago. de 2021.

MELO, Carlos. Séries Temporais (Time Series) com Python. **Sigmoidal**, 2020. Disponível em: <<https://sigmoidal.ai/series-temporais-time-series-com-python/>>. Acesso em 1 de ago. de 2021.

LARROSA, Raphael. How to forecast sales with Python using SARIMA model. **Towards data science**, 2019. Disponível em: <<https://towardsdatascience.com/how-to-forecast-sales-with-python-using-sarima-model-ba600992fa7d>>. Acesso em 1 de ago. de 2021.

Guia completo para criar Time Series (com código em Python). **Vooo – Insights**, 2016. Disponível em: < <https://www.vooo.pro/insights/guia-completo-para-criar-time-series-com-codigo-em-python/>>. Acesso em 1 de ago. de 2021.