UNIVERSIDADE FEDERAL DO PAMPA
CURSO EM ENGENHARIA QUÍMICA

LARISSA VIEIRA GOMES

APLICAÇÃO DE METODOLOGIAS ESTATÍSTICAS E *MACHINE LEARNING* PARA VALIDAÇÃO DE ANALISADORES CONTÍNUOS PARA INDÚSTRIA DE CELULOSE

Bagé
2021

LARISSA VIEIRA GOMES

APLICAÇÃO DE METODOLOGIAS ESTATÍSTICAS E *MACHINE LEARNING* PARA VALIDAÇÃO DE ANALISADORES CONTÍNUOS PARA INDÚSTRIA DE CELULOSE

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia Química.

Orientador: Prof. Dr. Alexandre Denes Arruda

Coorientadora: Dra. Andressa Ápio

Bagé
2021

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

G633a Gomes, Larissa Vieira
Aplicação de metodologias estatísticas e *machine learning* para validação de analisadores contínuos para indústria de celulose / Larissa Vieira Gomes.
93 p.

Trabalho de Conclusão de Curso(Graduação)--
Universidade Federal do Pampa, ENGENHARIA QUÍMICA, 2021.
"Orientação: Alexandre Denes Arruda".

1. Indústria 4.0. 2. Celulose. 3. Estatística. 4. Analisadores virtuais. 5. *Machine learning*. I. Título.

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal do Pampa

LARISSA VIEIRA GOMES

APLICAÇÃO DE METODOLOGIAS ESTATÍSTICAS E MACHINE LEARNING PARA VALIDAÇÃO DE ANALISADORES CONTÍNUOS PARA INDÚSTRIA DE CELULOSE

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel Em Engenharia Química.

Trabalho de Conclusão de Curso defendido e aprovado em: 10 de maio de 2021.

Banca examinadora:

Prof. Dr. Alexandre Denes Arruda
Orientador
Unipampa

Dra. Andressa Apio
Coorientadora
LATOS

Prof. Dr. Rodolfo Rodrigues
UFSM

Prof. Dr. André Ricardo Felkl de Almeida
Unipampa

Assinado eletronicamente por **ANDRE RICARDO FELKL DE ALMEIDA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/05/2021, às 11:24, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Rodolfo Rodrigues**, **Usuário Externo**, em 10/05/2021, às 12:01, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **ALEXANDRE DENES ARRUDA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/05/2021, às 13:12, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Andressa Apio**, **Usuário Externo**, em 10/05/2021, às 17:17, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0520598** e o código CRC **1C5FC87E**.

AGRADECIMENTOS

À Deus, pois sem Ele nada seria possível.
À minha família, por todo apoio e incentivo.
Aos meus pais, que nunca mediram esforços e sempre acreditaram em mim.
À minha irmã, Thássia Cíntia Vieira Gomes, que sempre foi inspiração e me ajudou em momentos difíceis.
Ao meu orientador, Prof. Dr. Alexandre Denes Arruda, pela orientação e dedicação nesse tema que foi um grande desafio.
À minha coorientadora, Dra. Andressa Ápio, por toda a disponibilidade e assistência.
Aos meus amigos, pelo companheirismo e carinho.
À banca examinadora, composta pelo Prof. Dr. André Ricardo Felkl de Almeida e Prof. Dr. Rodolfo Rodrigues.
Aos meus professores, que contribuíram além da formação acadêmica.
À empresa, em especial, ao Alberto e Luciana, pelo fornecimento de dados para o desenvolvimento do trabalho.

RESUMO

Ferramentas computacionais são cada vez mais comuns no cotidiano de cientistas e engenheiros, uma vez que permitem a análise sistemática de grandes quantidades de dados de forma rápida. A qualidade dos dados interfere drasticamente na tomada de decisões, se tornando essencial para uma boa gestão industrial. Na indústria de papel e celulose, a coleta de dados pode ser fator de risco ao trabalhador ao realizar medições durante atividades, como a colheita florestal, o cozimento da madeira e secagem da polpa. Ainda, os dados obtidos em laboratório estão sujeitos a variações e erros. Isto posto, a utilização de sensores e medidores automáticos além de agregarem mais segurança ao processo produtivo, reduzem custos e aumentam a produtividade, confiabilidade e qualidade dos dados. O objetivo desse trabalho é validar instrumentos de medição *on-line* utilizando metodologias estatísticas e um modelo de *Machine Learning* (Regressão Linear, Árvores de decisão, Florestas Aleatórias) para obtenção das medições laboratoriais através dos dados dos instrumentos. A implementação das metodologias estatísticas foi realizada por meio da linguagem Python para a análise de dados e construção do analisador virtual através de *Machine Learning*. Os resultados estatísticos se mostraram satisfatórios, com alguns erros espúrios. Já o analisador virtual construído, apresentou resultados de predição pouco representativos, com R^2 de 62,3% após otimização, necessitando maior quantidade de dados para resultados estatisticamente satisfatórios.

Palavras-chave: Indústria 4.0. Celulose. Estatística. Analisadores virtuais. *Machine Learning*. Python.

ABSTRACT

Computational devices are more and more becoming the everyday tools that scientists and engineers turn to for efficiently and quickly handling large data sets. The quality of the data points allows for rapid decision making along the processes of industries. For example, in the paper milling and manufacturing process there is a high risk of variance and error in the quality assurance process, which typically occurs in laboratories and are performed by technicians. However, with the use of sensors and measuring devices these errors can be greatly reduced to produce greater returns on processes, cost reduction, resource allocation, and the statistical confidence and quality of data. The goal of this project is to validate the use of measuring devices on a network by using statistical modeling and machine learning, such as linear regression, decision trees, and random forest, to obtain the higher quality laboratory measurements and data points along the manufacturing process. The implementation of statistical methodologies was executed using the Python language for both statistical tools and construction of the virtual analyzer through Machine Learning. The statistical results were satisfactory, with some spurious errors. The soft sensor, on the other hand, presented poorly representative prediction results with an R^2 of 62,3% after optimization, requiring a greater amount of data for statistically satisfactory results.

Keywords: Industry 4.0. Pulp. Statistics. Soft Sensors. Machine Learning. Python.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas

BCG - Boston Consulting Group

CEP - Controle Estatístico de Processo

DEPEC - Departamento de Desenvolvimento de Extensão e Cultura

EIU - The Economist Intelligence Unit

FAO - Organização das Nações Unidas para Alimentação e Agricultura

I/O - Entrada/Saída (*Input/Output*)

LIMS - Laboratory Information Management System

MAE - Erro Médio Absoluto (*Mean Absolute Error*)

MES - Manufacturing Execution Systems

MQ1 - Máquina Linha de Processo 1

MQ2 - Máquina Linha de Processo 2

MQ3 - Máquina Linha de Processo 3

MR - Amplitude Móvel (*Moving Range*)

MSE - Erro Quadrático Médio (*Mean Square Error*)

NaN - Não é um número (*Not a Numbe*r)

PIMS - Plant Information Management System

RSME - Raiz do Erro Quadrático Médio (*Root Mean Square Error*)

WMS - Warehouse Management System

LISTA DE SIMBOLOS

CV - Coeficiente de Variação

LC - Limite Central

LIC - Limite Inferior de Controle

LSC - Limite Superior de Controle

N - Tamanho da Amostra

μ - Posição Central

σ - Desvio Padrão

$\bar{x}$ - Média Amostral

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Ferramentas computacionais são cada vez mais comuns no cotidiano de cientistas e engenheiros, uma vez que permitem a análise sistemática de grandes quantidades de dados e a rápida manipulação de arquivos com enormes volumes de informação. Desde os primórdios da era industrial, a necessidade de otimizar os meios de produção em massa se tornou essencial para o desenvolvimento de novas tecnologias (DEMINGOS, 2019; ABREU, 2017).

Entre os séculos XVIII e XIX surgiram a primeira e a segunda revolução industrial, que impulsionaram o crescimento da indústria têxtil e de ferro, avanços na indústria química, elétrica, de petróleo e de aço, e a produção em massa de bens de consumo. De 1950 a 2000, a Terceira Revolução Industrial foi marcada por transformações profundas na produção e pela rapidez do desenvolvimento de novas tecnologias (FUNDAÇÃO DOM CABRAL, 2016).

O termo Indústria 4.0, também conhecido como a quarta Revolução Industrial, surgiu na Alemanha em 2011, tendo como foco o desenvolvimento da alta tecnologia para a manufatura. De acordo com Albertin *et al.* (2017), o diferencial da Indústria 4.0 está no fato de que o processo de fabricação vai evoluindo de uma única célula automatizada para sistemas totalmente automatizados e integrados que se comunicam com outros, contribuindo para maior flexibilidade, velocidade, produtividade e qualidade dos sistemas produtivos.

Segundo a Fundação Dom Cabral (2016), o relatório da Boston Consulting Group (BCG) lista os pilares da quarta revolução, que são consideradas as principais tecnologias da Indústria 4.0, sendo essas: robôs automatizados, manufatura aditiva, simulação, integração horizontal e vertical de sistemas industriais, *big data e analytics,* nuvem, segurança cibernética e realidade virtual.

Tamás e Illés (2016) descreveram a essência do *Big Data* como a "determinação de probabilidades com métodos e procedimentos matemáticos" baseada em enormes quantidades de dados, o que permitirá que as decisões sejam tomadas sem conhecer os efeitos de causa. No contexto da Indústria 4.0, as enormes quantidades de dados se referem ao grande número de informações e dados relacionados à produção, que são produzidos pelo equipamento de fabricação inteligente durante o processo de produção (ALBERTIN *et al*., 2007).

Um exemplo da aplicação dessa tecnologia é o caso de uma mina de ouro africana que encontrou maneiras de capturar mais dados de seus sensores. Esses novos dados mostraram flutuações insuspeitadas nos níveis de oxigênio durante um processo chave, a lixiviação. A correção dessa flutuação resultou em um aumento da produção equivalente a US$ 20 milhões por ano (BAUR; WEE, 2015).

As indústrias de processos químicos correspondem a uma fatia da grande quantidade de dados gerada atualmente. Esses processos demandam grande velocidade e capacidade de análise de dados. Por exemplo, uma planta típica de produção de olefinas possui mais de 5.000 variáveis que devem ser monitoradas. Plantas industriais de grande escala chegam a 20.000 variáveis de processo. Além disso, como os dados são correlacionados, é necessário fazer avaliações simultâneas (CAMPOS *et al*., 2016).

A indústria de celulose apresenta características que a difere dos demais setores, isso porque possui alto nível de desenvolvimento tecnológico e plantas industriais com grande capacidade de produção (DEPARTAMENTO DE DESENVOLVIMENTO DE EXTENSÃO E CULTURA, 2016). O setor é formado por uma grande cadeia, que conta com fábricas de celulose, papel e insumos químicos dos fornecedores. Como se trata de uma cadeia muito grande, um enorme volume de dados é gerado e, qualquer avanço na otimização do processo impacta em resultados consideráveis na produção e custos totais (SERVIÇO NACIONAL DE APRENDIZAGEM NACIONAL DO PARANÁ, 2020).

Os seres humanos têm dificuldade de analisar, de modo simultâneo, problemas que envolvam mais de três variáveis, e isso se torna ainda mais acentuado quando os dados estão corrompidos com ruídos e incertezas. A necessidade de utilizar assistência computacional no processamento de dados tem se tornado essencial (CAMPOS *et al*., 2016).

Um exemplo de ferramenta computacional para análise de dados da indústria de processos é a linguagem Python, que permite a análise de grandes bases de dados de forma rápida e eficiente. A análise de dados em Python não só inclui a identificação de correlação, mas também o pré-processamento dos dados, a interpretação dos resultados, e a extração de informação útil para a tomada de decisão.

Além disso, a ferramenta de qualidade denominada Controle Estatístico de Processos (CEP), quando aplicada à produção, permite a redução sistemática da

variabilidade em características chave da qualidade, contribuindo para a melhoria da qualidade, da produtividade, da confiabilidade e do custo de produção (RIBEIRO; CATEN, 2012). Essa ferramenta combina métodos estatísticos que podem ser executados através da linguagem Python.

1.1 Metodologia do trabalho

O presente trabalho foi desenvolvido da seguinte forma:

● pesquisa bibliográfica - seleção de normas brasileiras (ABNT), artigos, trabalhos acadêmicos e correlatos visando a compreensão do estado da arte dos temas propostos e, assim, definir o objetivo bem como confirmar sua viabilidade;

● metodologias a serem utilizadas - definidos os objetivos, procedeu-se a uma análise de metodologias estatísticas a fim de identificar quais se aplicavam ao estudo de caso. As alternativas adotadas são descritas no decorrer do texto, e

● implementação das metodologias utilizando a linguagem de programação Python e interpretação dos resultados.

1.2 Organização do trabalho

O trabalho foi subdividido em seções de 1 a 5. Na primeira seção será abordada a introdução. Já nas seções seguintes, respectivamente, a revisão de literatura, metodologia, análise de resultados e considerações finais serão apresentadas.

2 OBJETIVOS GERAIS

O objetivo geral do presente trabalho foi validar analisadores contínuos para medições *on-line* de dados de processo, em uma fábrica de celulose, utilizando métodos estatísticos.

2.1 Objetivos específicos

a) Verificar aplicação de metodologias estatísticas para controle de processos:

- análise dos dados do processo mediante Estatística Geral;
- construção de Cartas de Controle; e
- desenvolvimento de um analisador virtual através de técnicas de *Machine Learning* (Regressão Linear, Árvores de Decisão e Florestas Aleatórias) para obtenção da predição dos dados laboratoriais através dos dados dos instrumentos.

b) Utilização da linguagem Python para automatizar a aplicação dos métodos estatísticos do item (a).

3 CONCEITOS GERAIS E REVISÃO DE LITERATURA

Na presente revisão bibliográfica, serão abordados tópicos sobre automatização das indústrias, os processos envolvidos na indústria de celulose, bem como as etapas de produção e sua automatização, métodos estatísticos para controle de processos e a linguagem de programação Python.

3.1 O processo de automatização

O processo de automatização das indústrias tem aumentado significativamente o volume de dados gerados, acarretando uma dificuldade de utilização das planilhas manuais convencionais. A velocidade e o volume com que estes dados vêm sendo criados é impactante. De acordo com Smolan e Erwitt (2012), até 2003 a humanidade havia gerado 5 exabytes de dados. A Harvard Business Review afirma que mais dados passam pela Internet por segundo do que a quantidade de dados armazenada durante 20 anos. Também afirma que a quantidade de dados gerados por dia estaria duplicando a cada 40 meses (MCAFEE; BRYNJOLFSSON, 2012). Um exemplo disso é a rede Walmart, que coleta mais de 2,5 petabytes de dados por hora nas transações de seus clientes. Estima-se que a quantidade de dados armazenados atualmente é de aproximadamente 4,4 zettabytes.

Em 2011, o periódico The Economist entrevistou executivos de grandes organizações em sua primeira pesquisa sobre o tema *Big Data* (THE ECONOMIST INTELLIGENCE UNIT, 2011). A pesquisa apurou que as organizações ainda tinham dificuldades com alguns aspectos básicos relacionados à coleta e administração de dados e à sua efetiva exploração. É necessário que ocorra um tratamento para posterior interpretação dos dados para geração de valor (TAN *et al*., 2015).

A transformação de dados em informação relevante além de melhorar o resultado final por meio da otimização de operações, também aumenta a confiabilidade do processo (VERHAPPEN, 2013).

3.2 Indústria Celulose

A indústria de celulose apresenta características que a difere dos demais setores, isso porque possui alto nível de desenvolvimento tecnológico com uma tecnologia relativamente acessível e globalizada; capital intenso em tecnologia; plantas industriais com grande capacidade de produção e base florestal plantada, além disso uma parcela significativa da indústria atua com processo produtivo integrado (DEPARTAMENTO DE DESENVOLVIMENTO DE EXTENSÃO E CULTURA, 2016).

De acordo com Vidal e Hora (2012), em virtude de suas especificidades produtivas e características da matéria-prima principal, pelo lado da oferta, essa indústria é extremamente concentrada, enquanto que pelo lado da demanda é polarizada conforme o tipo de fibra, seja ela curta ou longa.

Em 2015, conforme a FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS, a produção mundial de celulose para papel (*pulp for paper*) considerando-se os processos químico e semiquímico, pasta de alto rendimento e pastas de outras fibras totalizou 180,9 milhões de toneladas.

Em virtude da concentração deste segmento, observa-se que dez países se consolidam como principais produtores mundiais de celulose para papel, sendo responsáveis, em 2015, por mais de 82% da produção mundial (FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS, 2016), conforme ilustrado na Figura 1.

Figura 1 - Mapa do *ranking* dos principais produtores de celulose

Fonte: Food and Agriculture Organization of the United Nations (2016)

Em relação às movimentações do mercado internacional (exportações e importações), verifica-se que as exportações de celulose vêm aumentando ao longo do período, com exceção do ano de 2009 – ano que coincide com o agravamento da crise internacional – todos os outros anos apresentaram crescimento no volume exportado.

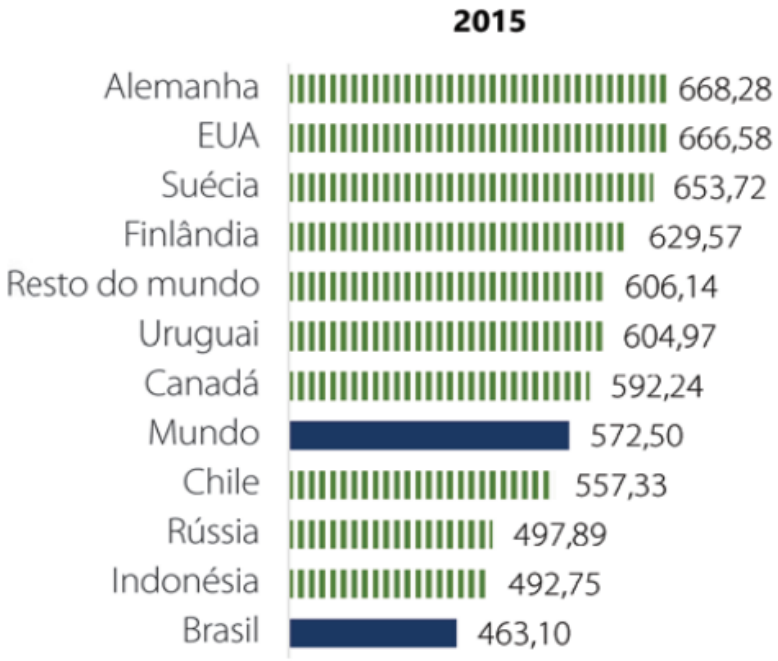
No período de 2005 a 2015, o crescimento foi de aproximadamente 29%, conforme a Figura 2.

Figura 2 - Evolução das exportações mundiais de celulose 2005 a 2015

Fonte: Food and Agriculture Organization of the United Nations (2016)

Na análise do preço médio da celulose por tonelada exportada, observa-se na Figura 3, que esse variou no ano de 2015 de US$ 463,10 a US$ 668,28 com o preço mundial médio de US$ 572,50 por tonelada exportada. O Brasil foi o país com o menor valor por tonelada, US$ 463,10.

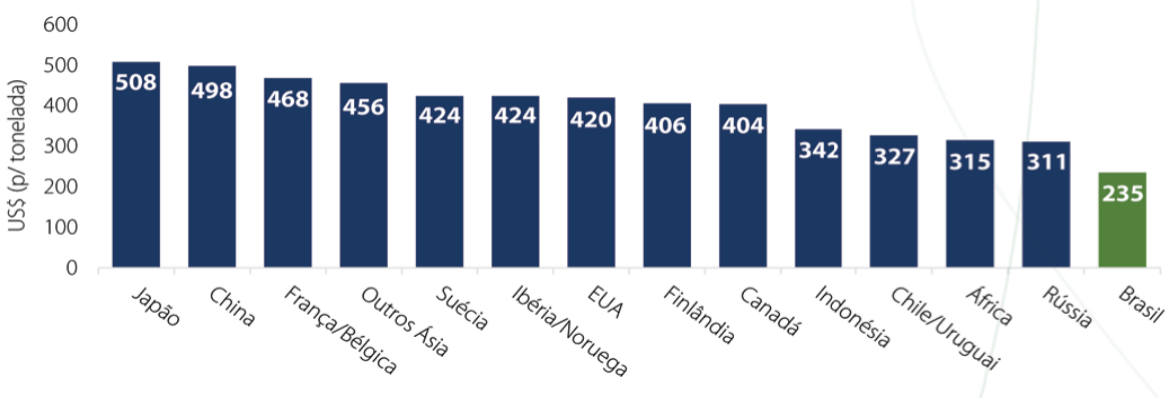
Figura 3 - Preço médio da celulose por tonelada exportada (em US$)



Fonte: Food and Agriculture Organization of the United Nations (2016)

Por outro lado, o preço mais competitivo do Brasil se deve principalmente pelo seu menor custo de produção global (Figura 4), em decorrência do clima favorável do país, além das características produtivas das empresas brasileiras, como a utilização de biotecnologia e de engenharia genética, que favorecem a produtividade brasileira, que é superior quando comparada com os demais países (DEPARTAMENTO DE DESENVOLVIMENTO DE EXTENSÃO E CULTURA, 2016).

Figura 4 - Custos de produção de celulose nos principais países produtores



Fonte: Departamento de Desenvolvimento de Extensão e Cultura (2016)

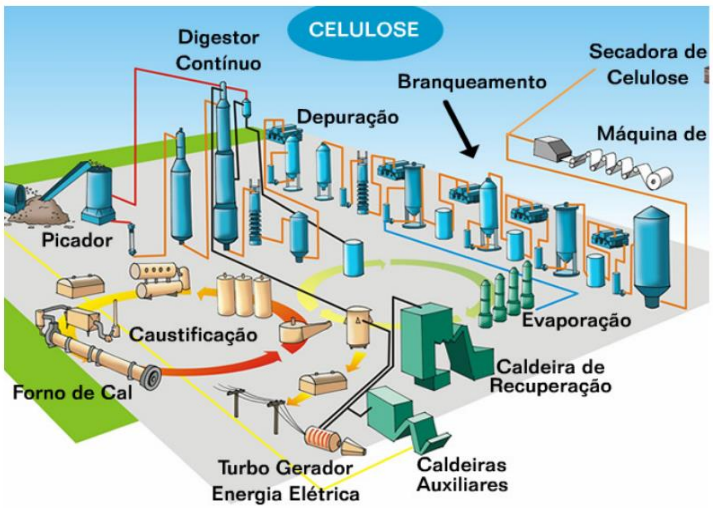
Observa-se, ainda, que o tempo de rotação dos eucaliptos de florestamento no Brasil também é menor, isso porque o eucalipto, principal fibra da celulose brasileira, leva em média 7 anos para crescer, enquanto que o pinus leva em média 15 a 20 anos (DEPARTAMENTO DE DESENVOLVIMENTO DE EXTENSÃO E CULTURA, 2016). Esses fatores contribuem para a maior produtividade brasileira, favorecendo o menor custo de produção do país.

Em 2015, o Brasil assumiu a primeira posição no ranking mundial de exportação, sendo o responsável por mais de 10,6 milhões de toneladas de celulose exportada, consolidando-se como um grande fornecedor global desse insumo. Na segunda posição segue o Canadá, responsável pela exportação de 9,3 milhões de toneladas. Os Estados Unidos se mantêm na terceira posição com 7 milhões de toneladas de celulose exportadas no mesmo período.

3.2.1 Produção da celulose

O processo de produção da celulose é baseado na transformação da madeira em material fibroso (pasta, polpa ou celulose industrial), inclui as seguintes etapas na ordem de produção: pátio de madeira; digestor; lavagem; deslignificação; depuração; branqueamento; secagem; evaporação; caldeira de recuperação; caustificação; forno de cal; conforme Figura 5.[1]

Figura 5 - Processo de produção da celulose



Fonte: Castro (2009, p. 10)

[1] Informação fornecida em Julho de 2020.

- **Pátio de madeira**: é a primeira etapa do processo, em que o objetivo é produzir cavacos com dimensões de aproximadamente 3 a 4 mm de espessura e 22 mm de comprimento, para alimentação do digestor. Os processos no pátio de madeira consistem em três etapas: descascamento, picagem e classificação.[2]

a) Descascamento

As cascas possuem um teor de fibras relativamente pequeno, e afetam negativamente as propriedades físicas do produto, portanto, a etapa de descascamento, tem por finalidade reduzir a quantidade de reagentes no processamento de madeira e facilitar a etapa de lavagem e peneiração. Além disso, as cascas residuais separadas das toras de madeira são picadas e enviadas junto com os rejeitos do peneiramento (finos) para a pilha de biomassa para geração de energia.[3]

b) Picagem

Consiste em reduzir as toras para as dimensões de 3 a 4 mm de espessura e 22 mm de comprimento, cujo tamanho facilita a penetração do licor de cozimento no digestor.[4]

c) Classificação

Após a picagem, os cavacos são classificados e separados de acordo com as dimensões padrões para o processamento. Os cavacos maiores retornam ao picador e os finos são incinerados na caldeira (CASTRO, 2009).

- **Digestor**: o cozimento resume-se em separar as fibras (celulose e hemicelulose) dos demais constituintes (lignina e extrativos) dos cavacos.[5] Os cavacos provenientes do pátio de madeira são alimentados ao silo (tempo de retenção de aproximadamente 18 minutos) e aquecidos com vapor à temperatura de 96ºC na saída do equipamento. O aquecimento expulsa o ar interno e promove o inchamento dos mesmos, o qual favorece a impregnação

[2] Informação fornecida em Julho de 2020.
[3] Informação fornecida em Julho de 2020.
[4] Informação fornecida em Julho de 2020.
[5] Informação fornecida em Julho de 2020.

do licor branco. Na saída do silo tem-se o primeiro contato com licor branco. As temperaturas são ajustadas entre 120 e 140 °C, esse controle é muito importante para assegurar o cozimento adequado.[6] O licor branco constitui-se de Hidróxido de Sódio, Sulfeto de Sódio e outros tipos de sais de sódio em pequenas quantidades (CASTRO, 2009).

- **Lavagem:** a lavagem prepara a polpa para deslignificação e, concomitantemente, inicia o circuito de envio do licor preto para a evaporação a fim de recuperar o máximo de álcali aplicado no cozimento. A composição básica do licor negro: 16% de sólidos; 37,4 g/L de carbonato de sódio e hidróxido de sódio; 7,4 g/L de sulfeto de sódio; 1,6 g/L de sulfato de sódio e 63,5 de hidróxido de sódio (total) (CASTRO, 2009).

- **Deslignificação:** o objetivo da deslignificação é reduzir o teor de lignina da polpa marrom proveniente do cozimento e em consequência reduzir o consumo de reagentes químicos na etapa seguinte, a depuração (CASTRO, 2009).

- **Depuração:** nesta etapa ocorre a eliminação dos rejeitos provenientes da madeira e de processo. A massa cozida é transferida para o sistema de depuração, que, por processo mecânico, separa os materiais estranhos (nós de madeira, pequenos palitos) das fibras.[7]

- **Branqueamento:** a finalidade do branqueamento é purificar a celulose removendo elementos que impediram o alvejamento completo, tais como resinas e extrativos da madeira, elementos não fibrosos e a lignina residual não dissolvida nas operações precedentes.[8]

- **Secagem**: na etapa da secagem ocorre a retirada da água e enfardamento da polpa branqueada, a fim de favorecer o transporte. A máquina secadora de celulose é do tipo folha flutuante, que seca a folha enquanto a mantém

[6] Informação fornecida em Julho de 2020.
[7] Informação fornecida em Julho de 2020.
[8] Informação fornecida em Julho de 2020.

flutuando em um colchão de vapor aquecido. A folha de celulose seca é tracionada e direcionada para a cortadeira, na qual realiza-se os cortes nas direções longitudinal e transversal. Subsequentemente, as folhas cortadas são empilhadas, prensadas em fardos, encapadas e identificadas, seguindo para o armazém de celulose.[9]

- **Evaporação**: a finalidade da evaporação é elevar a concentração do licor preto fraco, resultante do cozimento dos cavacos no digestor para uma concentração segura, possibilitando a incineração na caldeira de recuperação.[10]

- **Caldeira de recuperação**: é considerada uma das etapas mais importantes na fabricação de celulose, estando diretamente relacionada à viabilidade econômica de todo o processo (CASTRO, 2009). O processo consiste na incineração do licor preto previamente concentrado com a finalidade de recuperá-lo e gerar vapor para as turbinas. O controle de temperatura do licor preto de queima é em torno de 140ºC.[11] Após a queima, resta apenas a parte inorgânica constituída de carbonato de sódio (Na_2CO_3) e sulfeto de sódio (Na_2S), conhecido como "*smelt*". O *smelt* é dissolvido para formar o licor verde que é enviado para o processo de caustificação, onde recupera-se a soda cáustica (KLOCK; ANDRADE; HERNANDEZ, 2013).

- **Caustificação**: o principal objetivo da caustificação é converter o licor verde em licor branco com álcali efetivo (CASTRO, 2009), conforme segue:

Licor Verde + Cal + Água ⇆ Licor Branco + Lodo de Cal

$$Na_2S + NaCO_3 + CaO + H_20 \leftrightarrows (NaOH + Na_2S) + CaCO_3$$

- **Forno de Cal**: o forno de cal converte a lama de cal ($CaCO_3$) em cal (CaO), mantendo o carbonato residual na cal de saída do forno entre 2 a 4 %. Os principais parâmetros de controle são a manutenção da temperatura da

[9] Informação fornecida em Julho de 2020.
[10] Informação fornecida em Julho de 2020.
[11] Informação fornecida em Julho de 2020.

alimentação do forno entre 650 a 700 º C e residual de oxigênio entre 0,7 a 1,5 %.[12]

3.3 Automatização das indústrias de celulose

A Revista O Papel (2014) aborda a automatização das indústrias de celulose, e caracteriza alguns pontos elencados a seguir.

Eficiência operacional com custos reduzidos e qualidade do produto final são metas comuns à indústria de celulose e papel. As tecnologias de automatização e suas interfaces na manutenção contribuem de maneira significativa na conquista de tal objetivo. Mais do que uma contribuição, os sistemas de automação têm como principal objetivo justamente otimizar os processos produtivos, tornando-os mais eficientes e seguros, além de garantir um produto final de melhor qualidade.

Melhorias na qualidade do produto final e nos parâmetros ambientais são algumas das vantagens competitivas que a automação proporciona às indústrias de processamento. Outras vantagens evidenciadas por Marcelo de Oliveira, gerente geral industrial da unidade Aracruz da Fibria, são maior disponibilidade, estabilidade operacional, ritmo de produção e produtividade, além de redução de custos de produção e manutenção dos equipamentos e flexibilidade no sistema produtivo.

Para chegar aos resultados positivos oferecidos pela automação, todo o processo produtivo de celulose e papel – da floresta ao produto acabado – está em constante avanço tecnológico. Processos cada vez mais automatizados têm possibilitado um **controle mais eficiente**. “A demanda pela gestão industrial em tempo real torna a automação cada vez mais presente em toda a cadeia produtiva, beneficiando também áreas não ligadas diretamente ao processo fabril, as quais estão caminhando de forma acelerada para essa mesma direção”, avalia Gustavo Martins Galli, engenheiro eletricista da Lwarcel Celulose. Para exemplificar, ele cita sistemas como o de gerenciamento do manejo florestal (FS – *Forest System*), de análises laboratoriais (LIMS – *Laboratory Information Management System*) e de estoque de celulose (WMS – *Warehouse Management System*), os quais disponibilizam informações que, em conjunto com o PIMS (*Plant Information*

[12] Informação fornecida em Julho de 2020.

Management System) e com o MES (*Manufacturing Execution Systems*), permitem total rastreabilidade dos processos.

Galli informa ainda que o aumento da confiabilidade das redes de comunicação aproximou os dispositivos de I/O (*Input/Output*) aos equipamentos de campo. Painéis são instalados no campo, reduzindo o número de cabos até as salas elétricas. Com a descentralização dos painéis, as salas elétricas tiveram o tamanho reduzido. “Tudo isso contribui para diminuir os custos de instalação e pode ser considerado como uma solução consolidada no setor”, afirma o engenheiro eletricista. “A Lwarcel está alinhada com essa realidade e leva em conta tal solução nos novos projetos”, completa (REVISTA O PAPEL, 2014).

De acordo com Christiano Sousa, gerente executivo do Centro Técnico da Andritz, uma das grandes vantagens da automação industrial e do controle de processos reside na maior ênfase à flexibilidade e conversibilidade no processo de fabricação. “Os fabricantes estão cada vez mais exigentes quanto à capacidade de alternar facilmente a fabricação de uma ampla gama de produtos sem precisar reconstruir completamente as linhas de produção”, justifica ele sobre a tendência. Entre as razões para automatizar uma planta, Sousa cita (REVISTA O PAPEL, 2014):

- **Gestão das informações (automação integrada)**
  A automação é a base para que a empresa tenha as informações adequadas na definição das melhores estratégias e da gestão do negócio.
- **Aumento da produtividade**
  A automação industrial permite às empresas ciclos de operação mais rápidos e com maior eficiência.
- **Redução de custos**
  A automação industrial simplifica tarefas, reduzindo os custos de mão de obra, a ainda minimiza a criação de materiais e resíduos.
- **Melhoria da qualidade**
  Com a automação industrial, os processos podem ser cuidadosamente regulados e controlados, de modo que a qualidade do produto final não seja apenas confiável, mas também bastante melhorada.

- **Segurança**

  A automação industrial melhora efetivamente a segurança no trabalho e protege os operadores, cada vez mais envolvidos com os processos em salas de controle.

- **Competitividade**

  A fim de sobreviver na economia global de hoje, as empresas precisam manter-se competitivas. A automação industrial tem possibilitado às empresas ficar um passo à frente de seus concorrentes.

3.3.1 Pontos de coleta dados

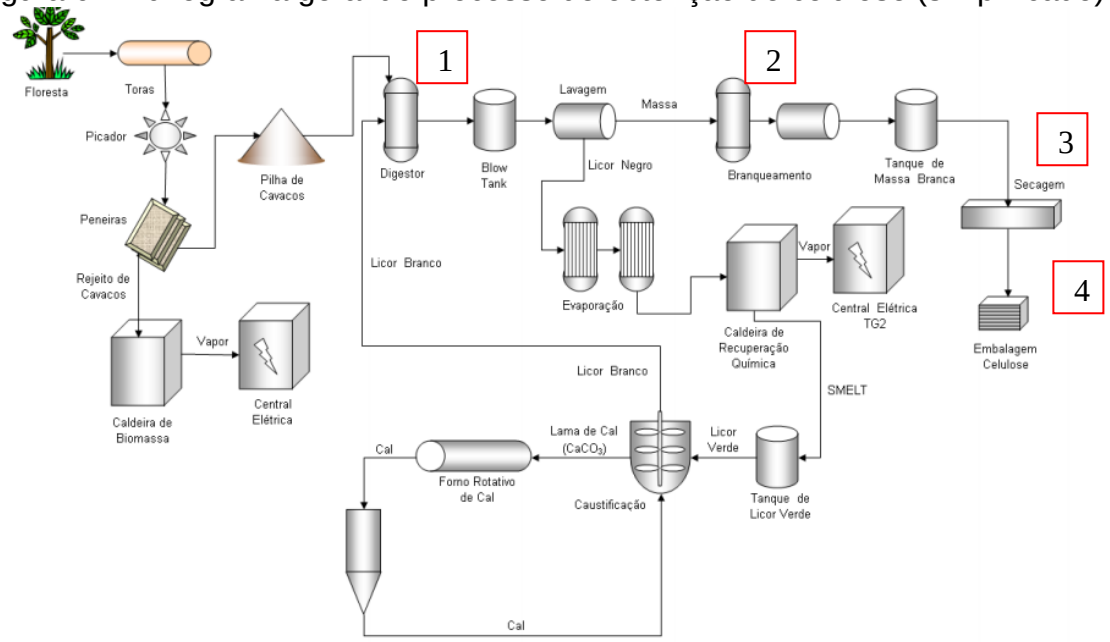
Na indústria de papel e celulose, a coleta de dados pode ser um fator de risco ao trabalhador ao realizar medições durante atividades, como a colheita florestal, o cozimento da madeira e secagem da polpa. Portanto, a utilização de sensores e medidores automáticos agregam mais segurança, precisão e acuracidade ao processo produtivo.

Esses equipamentos captam as variáveis, e transformam os parâmetros em dados, permitindo o envio das informações para o gerenciamento em tempo real. O processo de captação de dados inicia-se com a instalação de um instrumento com um sistema *on-line* de monitoramento, que é integrado ao equipamento de processo, fornecendo informações como pH, Alvura, Kappa, Consistência, Teor Seco, Gramatura, entre outras, tudo isso em tempo real no equipamento instalado nas plantas.

A gramatura é um parâmetro de qualidade relacionado à massa da folha de celulose por unidade de área. O número de Kappa está relacionado ao teor de lignina, sendo ela responsável pela coloração amarronzada da polpa. Já a alvura, é a medida do quão branco a polpa está após branqueamento. A consistência é definida pela razão entre o peso de massa de fibra seca e o peso total da polpa em meio aquoso, esta variável deve ser mais homogênea possível, pois interfere em outras variáveis da qualidade. O Teor Seco se trata de um parâmetro relacionado à umidade presente na massa (FOELKEL, 1976).

Os dados são coletados nos pontos cruciais do processo de produção da celulose. São eles: Digestor (1), Etapas de Branqueamento (2), Secagem (3) e Produto Final (4), conforme Figura 6.

Figura 6 - Fluxograma geral do processo de obtenção de celulose (simplificado)



Fonte: Adaptada de Lwarcel (2007)

3.4 Introdução à Estatística

De acordo com Crespo (1999), a Estatística é uma parte da Matemática que fornece métodos para a coleta, organização, descrição, análise e interpretação de dados e para a utilização dos mesmos na tomada de decisões.

A importância da estatística vem aumentando cada vez mais conforme a necessidade do ser humano em lidar com dados, previsões e tomadas de decisões.

> A Estatística ajudará em tal trabalho, como também na seleção e organização da estratégia a ser adotada no empreendimento e, ainda, na escolha das técnicas de verificação e avaliação da quantidade e da qualidade do produto e mesmo dos possíveis lucros e/ou perdas (DALPIAZ; GESSER, 2007, p. 4).

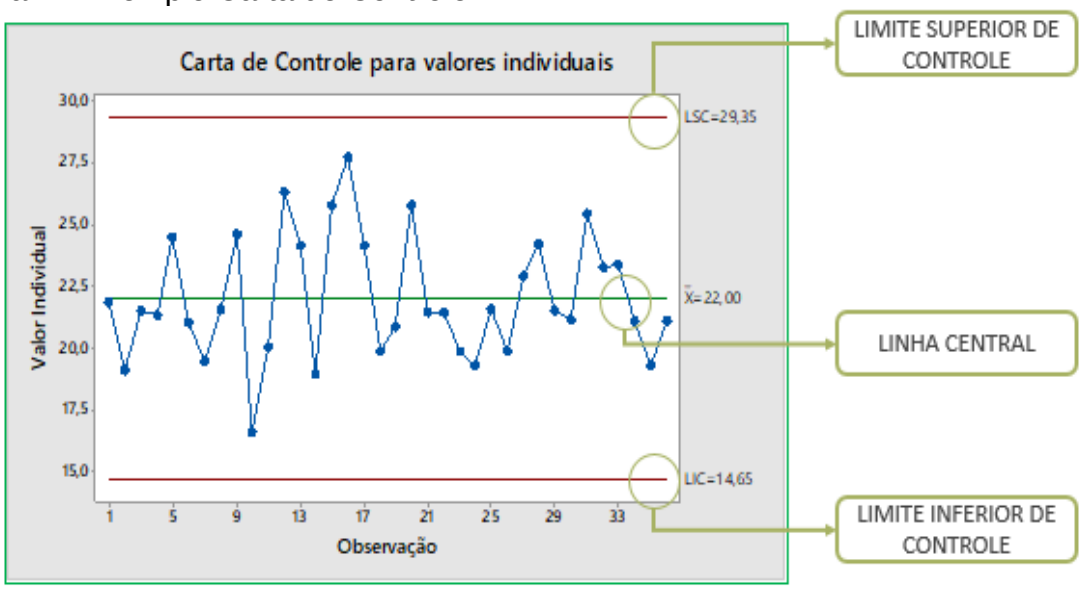
- **Controle Estatístico de Processo (CEP)**

O CEP é uma ferramenta utilizada para identificar as variabilidades do processo, através de algumas técnicas estatísticas ele quantifica e qualifica as variações do processo (MOURA; LINO; FERNANDES, 2008). Conforme Paese (2001), o CEP é aplicado na maioria dos casos para realizar o monitoramento de processos em tempo real, possibilitando que as tomadas de ações corretivas sejam realizadas com maior eficiência, evitando perdas na produtividade e em tempo de investigação dos problemas.

3.4.1 Gráficos de Controle

Os gráficos de controle são utilizados para identificar erros devido a causas especiais existentes em um processo. A partir dos dados obtidos na análise destes gráficos, é possível tomar decisões preventivas e controlar possíveis desvios de variabilidade no processo produtivo (MICHEL; FOGLIATTO, 2002). De acordo com Mayer (2004), os gráficos de controle, também chamados de cartas de controle permitem entender e visualizar resultados/saídas de processos. A Figura 7 representa um exemplo desses gráficos.

Figura 7 - Exemplo Carta de Controle



Fonte: Rodrigues (2020)

Estes gráficos determinam estatisticamente faixas denominadas limites de controle. Os limites são linhas que demarcam o gráfico, chamadas limite superior de controle (LSC), limite inferior de controle (LIC) e limite central (LC). Quando todos os pontos observados estiverem inseridos entre esses limites, de forma aleatória, considera-se que o processo está sob de controle. No entanto, se as observações estiverem fora dos limites de controle, o processo está fora de controle e, investigação e estudos são necessárias para detectar e eliminar as causas especiais no processo. Presume-se que uma causa especial ocorreu devido à existência destes valores extremos (OLIVEIRA, 2013).

As cartas de controle são classificadas em dois tipos: por variável ou por atributo. As cartas de controle por variáveis trabalham com dados de cunho quantitativo. Já as cartas de atributo analisam características qualitativas, o que

revela pouca informação sobre o processo, acarretando em uma baixa utilização no meio industrial (COSTA; EPPRECHT; CARPINETTI, 2004).

Dentre as cartas por variáveis, tem-se as cartas de valores individuais e amplitude móvel (X-mR). Este tipo é usado para monitorar a média e a variação do processo quando se tem dados contínuos que são observações individuais e não em subgrupos (MONTGOMERY, 2009). Caracteriza-se por dois gráficos: um gráfico baseado na Média (Carta Individual) e outro baseado na Amplitude Móvel (Carta mR), com duas observações sucessivas como base da estimativa da variabilidade do processo.

Antes de interpretar a carta individual, deve-se examinar a carta de Amplitudes Móveis (*MR Chart*) para determinar se a variação do processo está sob controle. Se a carta mR não estiver sob controle, os limites de controle da carta individual não serão precisos, entretanto o problema pode ser devido às causas especiais. Para identificação desses problemas, utiliza-se os testes de causas especiais para identificar padrões e tendências específicas nos dados de processo. Cada um dos testes detecta um padrão ou tendência específico, o que revela um aspecto diferente da instabilidade do processo.

O primeiro dos quatro testes diz que apenas um ponto acima do limite de controle, é uma situação atípica; o segundo teste possui maior sensibilidade e identifica mudanças no processo, é caracterizado por nove pontos em uma linha acima ou abaixo da linha central; o teste três diz que seis pontos em uma linha, todos crescentes ou todos decrescentes, indica uma tendência; por fim, o quarto teste diz que quatorze pontos em uma linha, alternando para cima e para baixo indica uma variação sistemática (SUPORTE MINITAB, 2018).

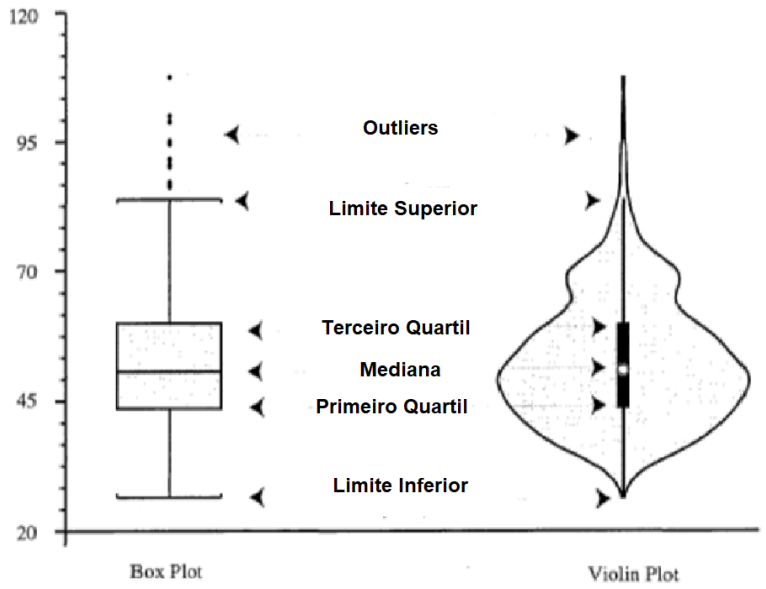
3.4.2 *Boxplots* e *Violinplots*

O *boxplot* é um gráfico frequentemente utilizado na pesquisa científica para a análise exploratória de variáveis quantitativas. É uma ferramenta que exibe a tendência central (mediana), dispersão (quartis 25% e 75%), valores de mínimo e máximo e valores discrepantes (*outliers*) (VALLADARES NETO, 2017).

Ainda dentro da análise exploratória de variáveis quantitativas, tem-se os gráficos *violinplots*. Este método é uma combinação do *boxplot* e de um gráfico de densidade *Kernel* rotacionado. A construção inicia-se com o *boxplot* indicando a

mediana e a distância interquartil, então é acrescentado um gráfico de densidade *Kernel* rotacionado em cada lado do *boxplot.* Sua forma é semelhante à de um violino, daí o nome recebido (HINTZE, 1998). A Figura 8 compara os gráficos *boxplot* e *violinplot.*

Figura 8 - Estrutura básica dos gráficos *boxplot* e *violinplot*



Fonte: Adaptada de Hintze (1998)

3.4.3 Analisadores virtuais

Analisadores virtuais são sistemas que utilizam algoritmos matemáticos para a obtenção de estimativas em tempo real de variáveis que não são medidas por sensores, utilizando suas correlações com dados disponíveis por medições em processo (FORTUNA; GRAZIANI; XIBILIA, 2005).

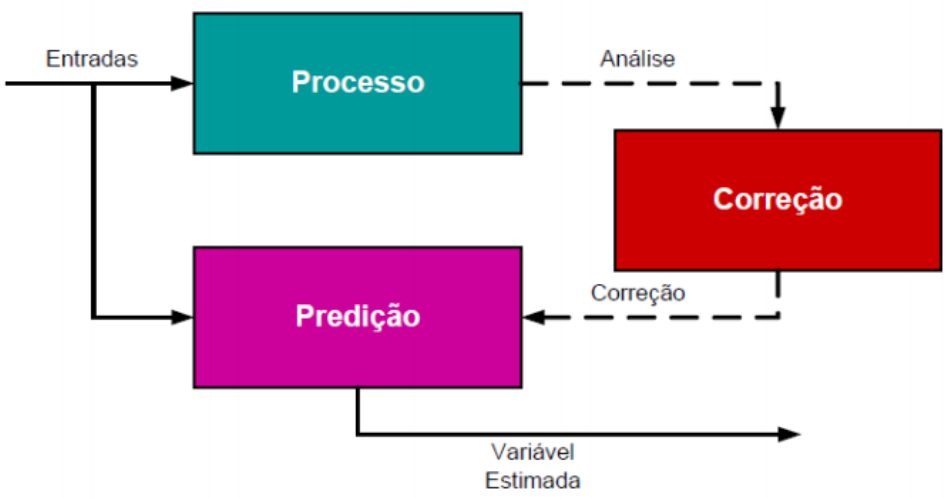
A utilização de analisadores virtuais em plantas industriais vem sendo estudada por diversos autores, se tornando uma prática comum para suprir a necessidade de medidas de variáveis obtidas por meio de análises pouco frequentes. Existem diversos trabalhos acerca da implementação de analisadores virtuais, por exemplo, para estimação da concentração de gasolina (C5) na corrente de topo e de butano (C4) na corrente de fundo de uma coluna debutanizadora (FORTUNA; GRAZIANI; XIBILIA, 2005), do índice de fluidez em um reator de polimerização e concentração de biomassa em fermentador contínuo (THAM *et al*., 1991) e viscosidade da corrente de alimentação de um secador do tipo spray-dryer (LIN; SOUZA; YOUNG, 2009).

Fortuna, Graziani e Xibilia (2005) destacam algumas vantagens da utilização de analisadores virtuais. São elas:

- alternativa de baixo custo em comparação com a instrumentação;
- podem trabalhar em paralelo com sensores físicos, sendo útil na identificação de falhas de instrumentação;
- são de fácil implementação e calibrados de acordo com as mudanças nos parâmetros do sistema;
- permitem a estimação de dados em tempo real.

A Figura 9 apresenta os constituintes básicos de um sistema de analisador virtual.

Figura 9 - Estrutura básica de um analisador virtual



Fonte: Facchin (2020, p. 20)

As entradas do algoritmo consistem nas variáveis secundárias do processo, que são correlacionadas no modelo matemático do analisador virtual, inserido no bloco de predição. O bloco correção é composto por uma estratégia de adaptação do modelo matemático, baseado nos dados obtidos pela análise laboratorial (CORSETTI, 2016).

Existem três tipos de abordagem para o desenvolvimento de sensores virtuais no que diz respeito ao modelo usado: físicos, estatísticos e baseados em inteligência artificial (KADLEC; GABRYS; STRANDT, 2009).

- **Analisadores virtuais utilizando *Machine Learning***

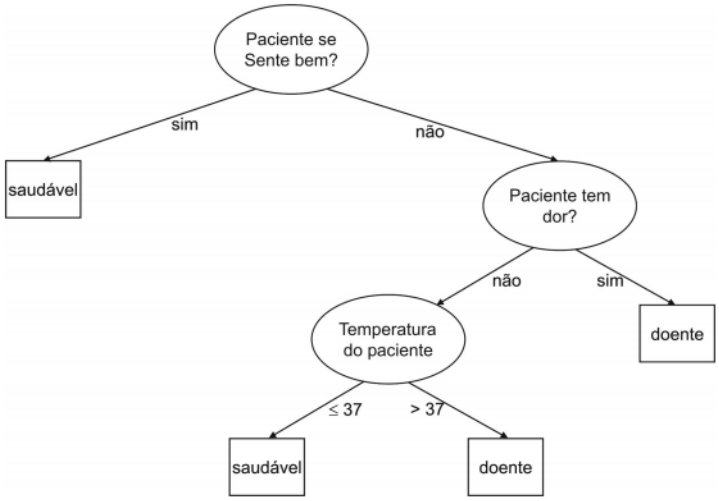
O *Machine Learning* é uma área da inteligência artificial voltada à construção de sistemas capazes de induzir hipóteses ou aproximar funções a partir da experiência acumulada em problemas anteriores (FACELI *et al.*, 2011). Uma ampla variedade de técnicas de inferência estatística e *Machine Learning* vem sendo empregadas no desenvolvimento de analisadores virtuais, entre eles árvores de decisão, florestas aleatórias e regressão.

a) Árvores de Decisão

As árvores de decisão são modelos hierárquicos de aprendizado supervisionado compostos internamente por nós de decisão e folhas terminais. Cada nó de decisão aplica um teste aos dados de entrada, de modo a seguir por um dos ramos que levará a um novo nó de decisão ou a uma folha terminal, determinando o valor da variável de interesse (COSTA-FILHO, 2019).

Na Figura 10 tem-se um exemplo ilustrativo de uma árvore de decisão para o diagnóstico de um paciente. Cada elipse representa o teste em um atributo para um dado conjunto de dados. Cada retângulo representa uma classe, ou seja, o diagnóstico. Para diagnosticar (classificar) um paciente, basta começar pela raiz, seguindo cada teste até que uma folha seja alcançada.

Figura 10 - Exemplo árvore de decisão simples



Fonte: Monard; Baranauskas (2003, p. 60)

b) Floresta Aleatória

O algoritmo floresta aleatória consiste em uma técnica de *Machine Learning* que agrupa diversas árvores de decisão criadas a partir de uma base de treinamento, de modo que o modelo resultante consolida os resultados de todas as árvores para predições (BREIMAN *et al.*, 2001).

c) Regressão Linear

A regressão linear é um dos conceitos estatísticos mais utilizados dentro do *Machine Learning,* caracteriza-se por ser uma reta traçada a partir de uma relação em um diagrama de dispersão. Este diagrama permite definir empiricamente se há um relacionamento linear entre as variáveis X e Y, e se o grau de relacionamento linear entre as variáveis é forte ou fraco (RODRIGUES, 2015).

3.4.4 Linguagem Python

Python é uma linguagem de programação interpretada, de alto nível e semântica simples. É uma das linguagens que mais tem crescido devido a sua compatibilidade com sistemas operacionais, capacidade de auxiliar outras linguagens e o incentivo à programação modularizada com reuso de códigos, devido ao suporte de módulos e pacotes. Programas como Dropbox, Reddit e Instagram são escritos em Python. A linguagem conquistou a comunidade cientifica e se tornou a mais popular no que concerne a análise de dados (CAELUM, 2020).

- **Principais Bibliotecas**

a) Bibliotecas Estatísticas e Numéricas

- **NumPy:** é um dos principais pacotes de processamento numérico. Ele oferece o código aglutinador para as estruturas de dados, os algoritmos e a biblioteca necessários à maioria das aplicações que envolvam dados numéricos em Python (MCKINNEY, 2019).

- **Pandas:** é uma biblioteca Python que fornece estruturas de dados de alto nível e uma grande variedade de ferramentas para análise. Os principais objetos são o DataFrame, uma estrutura de dados orientada a colunas, com Labels (rótulos) tanto para linhas quanto para colunas, e as series, um objeto array unidimensional com rótulo. O pandas disponibiliza uma funcionalidade sofisticada de indexação que facilita a reformatação, manipulação, agregações e a seleção de subconjuntos de dados (MCKINNEY, 2019).

- **SciPy**: é outra biblioteca essencial na análise de dados em Python. Composta por uma coleção de pacotes que estendem os recursos do NumPy. Alguns dos principais pacotes tem a finalidade de solucionar equações diferenciais, integração numérica, álgebra linear, matrizes, etc. (MCKINNEY, 2019).

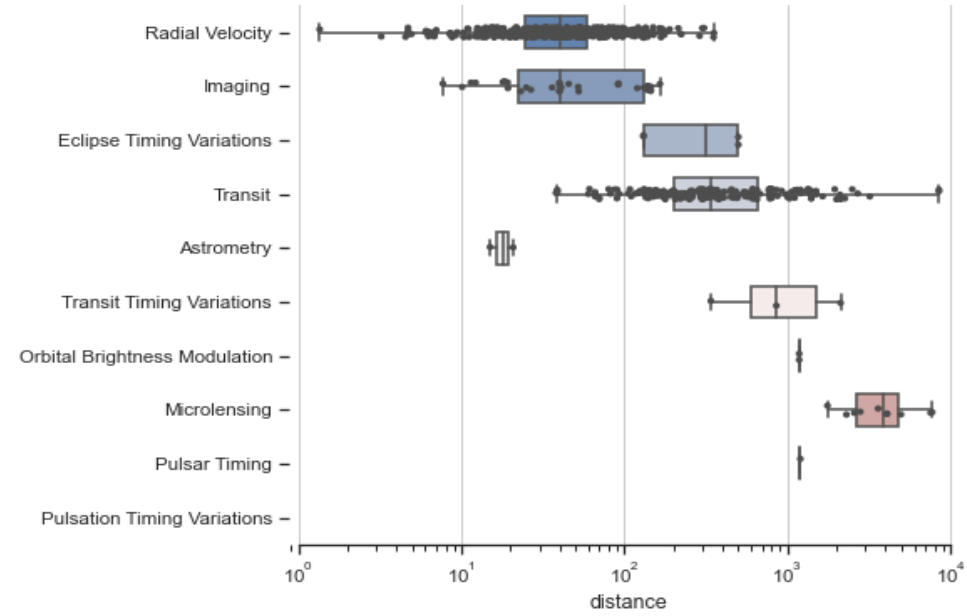
b) Bibliotecas de Visualização

- **Matplotlib:** é uma biblioteca destinada à criar diagramas e gráficos bidimensionais. É possível construir diversos tipos de gráficos, como histogramas, gráficos de dispersão e coordenadas não cartesianas (MCKINNEY, 2019).

- **Seaborn:** permite a construção de gráficos mais sofisticados, como gráficos de correlação, gráficos ternários e gráficos 3D (MCKINNEY, 2019).

A Figura 11 representa um gráfico de *boxplot* horizontal utilizando a biblioteca Seaborn.

Figura 11 - Exemplo gráfico criado utilizando o seaborn



Fonte: Seaborn Pydata (2020)

3.4.5 Estados da arte

Trabalhos recentemente publicados na literatura demonstram que as técnicas de *Machine Learning* e CEP têm sido aplicadas em diversas áreas industriais. O Quadro 1 exemplifica trabalhos que aplicaram estas técnicas.

Quadro 1 - Trabalhos com aplicação de *Machine Learning* e CEP

Autor(es) (Ano)	Objetivos	Métodos	Principais Resultados
Rodrigues (2014)	Propor o uso de analisadores virtuais como uma solução para obtenção do teor de enxofre em alta frequência e com baixo investimento inicial.	Modelos de Regressão Linear e Redes Neurais Artificiais	O modelo de Redes Neurais Artificiais demonstrou desempenho superior ao construído por Regressão Linear, se mostrando uma boa alternativa à compra de analisadores de processo.
Ferreira *et al.* (2018)	Analisar se os padrões de gramatura estão de acordo com o que se espera através do Controle Estatístico de Processo (CEP).	Cartas de Controle	Os padrões de gramatura dos itens alimentícios analisados estão estatisticamente satisfatórios.
Costa-Filho *et al.* (2020)	Evidenciar diferenças de precisão nas estimativas realizadas por modelos estatísticos de regressão e modelos de inteligência artificial.	Modelos de Regressão, Florestas Aleatorias, Máquina de Vetores de Suporte e Redes Neurais Artificiais	Os modelos de máquina de vetores de suporte e rede neural artificial obtiveram performances de generalização superiores aos demais modelos quanto aos indicadores estatísticos e dispersão de resíduos.
Farias (2018)	Construção de um analisador virtual capaz de predizer a pressão de fundo em poços de petróleo com o mínimo erro associado.	Modelos de Regressão Linear e Redes Neurais Artificiais	O método de Rede Neural foi o mais acurado para modelagem preditiva da pressão de fundo dos poços de petróleo.

Fonte: Autor (2021)

A análise do conteúdo destes trabalhos publicados permite identificar significativos resultados satisfatórios no uso das técnicas de *Machine Learning* e Cartas de Controle área das Engenharias.

4 METODOLOGIA

Neste item foram abordados a forma de obtenção dos dados utilizados, bem como a descrição dos métodos estatísticos aplicáveis ao caso, e seu desenvolvimento na linguagem Python.

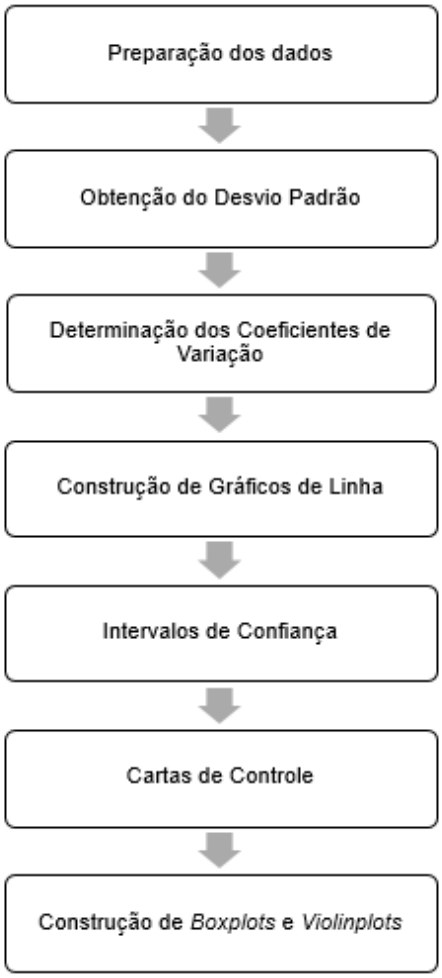
4.1 Obtenção dos dados de processo

A fábrica de celulose que fornece os dados de processo para este estudo, faz o levantamento das *tags* através do historiador de processos *PI DataLink*, utilizando o *plugin* da ferramenta para o Microsoft Excel. Este *plugin* permite que o intervalo de aquisição dos dados seja definido pelo usuário, bem como o regime de coleta. As informações utilizadas neste trabalho foram recebidas através planilhas enviadas por mensagem eletrônica.

4.2 Análise exploratória

A análise exploratória compreende a descrição dos dados estatisticamente e através de técnicas de visualização, para ressaltar os pontos mais importantes do conjunto de dados para uma análise posterior. Este processo considera a análise dos dados disponíveis por diferentes aspectos, a descrição e o condensamento das informações sem fazer nenhum tipo de julgamento referente aos dados. Essa parte compreendeu as etapas descritas na Figura 12:

Figura 12 - Fluxograma da metodologia de análise exploratória



Fonte: Autor (2021)

Preparação dos dados: a base de dados utilizada foram arquivos do tipo .xlsx e, portanto, precisam ser importados para um *DataFrame*, como pode ser observado na Figura 13.

Figura 13 - Algoritmo para importação de dados

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[ ]  df = pd.read_excel('dataLarissa.xlsx', header=1)
```

Fonte: Autor (2021)

A base de dados possui, para cada variável, a medição do instrumento e do laboratório em diferentes horas do dia. Uma amostra dos dados pode ser vista na Figura 14, onde é possível identificar que existem valores faltantes e colunas não nomeadas.

Figura 14 - Base de dados antes do pré-processamento

	Unnamed: 0	Kappa F1 lab	Kappa F1 Inst	Unnamed: 3	Consistência DA F1 lab	Consistência DA F1 Inst	Unnamed: 6	Unnamed: 7	Alvura P lab
0	Number of Values:	100.000000	NaN	NaN	Number of Values:	191.00	NaN	NaN	Number of Values:
1	2019-01-02 07:22:00	19.309999	18.780001	NaN	2019-01-02 09:08:00	10.13	10.04	NaN	2019-01-02 07:45:00
2	2019-01-08 07:41:00	15.800000	16.010000	NaN	2019-01-04 13:32:00	10.01	9.87	NaN	2019-01-08 08:03:00
3	2019-01-14 07:18:00	15.190000	15.720000	NaN	2019-01-08 09:13:00	10.09	10.03	NaN	2019-01-14 07:39:00
4	2019-01-21 07:28:00	17.490000	17.940001	NaN	2019-01-09 10:07:00	10.13	10.63	NaN	2019-01-22 07:46:00

Fonte: Autor (2021)

As etapas necessárias para pré-processamento dos dados incluíram a remoção de linhas e colunas que não possuíam valores numéricos, e renomeação das colunas.

Desvio Padrão Amostral: sua medida representa o quanto os dados se afastam da média, e pode ser obtida empiricamente através da Equação 1,

$$\sigma = \sqrt{\frac{(x_i - \bar{x})^2}{N-1}} \quad (1)$$

em que N é o número de amostras, x_i é o valor de uma amostra e $\bar{x}$ é o valor da média.

A função .describe(), vista na Figura 15, é utilizada para resumir brevemente a disposição estatística dos dados, através dela é possível obter o número de vezes que a variável aparece na lista, a média, desvio padrão, valores de máximo e mínimo e os quartis.

Figura 15 - Exemplo de utilização da função “describe”

```
df.describe()
```

	kappa_lab	kappa_inst	consistencia_lab	consistencia_inst	alvura_lab	alvura_inst
count	100.000000	100.000000	191.000000	191.000000	132.000000	132.000000
mean	16.754000	17.201100	9.968168	10.041518	89.812576	89.920303
std	0.824250	0.673642	0.278226	0.354330	0.399134	0.416737
min	14.670000	15.640000	9.050000	9.100000	88.779999	88.889999
25%	16.187500	16.732500	9.775000	9.820000	89.529999	89.599998
50%	16.785000	17.230000	10.000000	10.040000	89.849998	89.900002
75%	17.315000	17.682500	10.150000	10.205000	90.087502	90.199997
max	19.309999	19.370001	10.840000	11.040000	90.669998	91.599998

Fonte: Autor (2021)

Coeficiente de Variação: o coeficiente de variação é uma medida de dispersão relativa definida como a razão entre o desvio padrão e a média dado pela Equação 2,

$$CV = \frac{\sigma}{\bar{x}} \tag{2}$$

onde σ é o desvio padrão e $\bar{x}$ é o valor da média.

Gráficos de Linhas: são gráficos normalmente usados para avaliar alterações ao longo do tempo e facilitar a identificação de tendências ou de anomalias. O código em Python pode ser visto na Figura 16.

Figura 16 - Código utilizado para gerar gráficos de linha.

```
In [ ]: plt.plot(data['variável']['lab']),plt.plot(data['variável']['inst'])
        plt.legend(['Laboratório','Instrumento'])
        plt.xlabel('Time-stamp',size=15)
        plt.ylabel('Variável',size=15)
```

Fonte: Autor (2021)

Intervalos de Confiança: são obtidos empiricamente com 95% de confiança para o parâmetro populacional através da Equação 3. Em que, $\bar{x}$ é a média amostral, *n* o tamanho da amostra, σ representa o desvio padrão e μ é a posição central da distribuição,

$$\bar{x} - 1{,}96\,\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + 1{,}96\,\frac{\sigma}{\sqrt{n}} \tag{3}$$

Os intervalos de confiança são obtidos seguindo as funções descritas na Figura 17.

Figura 17 - Algoritmo para obtenção dos intervalos de confiança

```
In [ ]: variável_X=data['variávelX']['inst']-data['variávelX']['lab']
        mean, sigma= np.mean(variável_X), np.std(variável_X)
        print(mean+1.96*sigma/np.sqrt(len(variávelX)))
        print(mean-1.96*sigma/np.sqrt(len(variávelX)))
```

Fonte: Autor (2021)

Cartas de Controle: as equações válidas para a análise dos valores individuais e amplitudes móveis de acordo com Montgomery (2009), são as que seguem por meio das Equações 4, 5 e 6,

$$MR_i = |x_i - x_{i-1}| \tag{4}$$

sendo X_i e X_{i-1} valores monitorados.

Para a carta de controle de observações individuais, utiliza-se D_3 igual à 0 e D_4 como 3,267 (Anexo A). Dessa forma, os limites de controle da carta de amplitude média são calculados da seguinte forma:

$$\begin{aligned} LSC &= D_4\overline{MR} \\ LC &= \overline{MR} \\ LIC &= D_3\overline{MR} \end{aligned} \tag{5}$$

sendo MR a média das amplitudes móveis, D_3 e D_4 constantes.

Os parâmetros da carta de controle para unidades individuais são dados pela Equação 6.

$$LSC = \bar{x} + 3\frac{\overline{MR}}{D_2}$$

$$LC = \bar{x} \qquad (6)$$

$$LIC = \bar{x} - 3\frac{\overline{MR}}{D_2}$$

em que $\bar{x}$ é a média dos valores e D_2 é uma constante igual à 1,128 (Anexo A).

Não existem bibliotecas especificas para Python que construam cartas de controle. As mesmas são construídas combinando as bibliotecas demonstradas na Figura 18 e utilizando as Equações 4, 5 e 6.

Figura 18 - Bibliotecas requeridas para construção das cartas de controle

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statistics
```

Fonte: Autor (2021)

***Boxplots* e *Violinplots*:** estes gráficos foram gerados através da distribuição dos erros entre laboratório e instrumento, dado pela subtração do primeiro menos o segundo. Dessa forma, quanto mais próximo de zero estiver a média, menor é a diferença entre as medições. Os gráficos são obtidos seguindo as funções descritas na Figura 19.

Figura 19 - Algoritmos para obtenção dos *Boxplots* e *Violinplots*.

```
In [ ]: sns.boxplot(y=(data['alvura']['lab']-data['alvura']['inst']))
        sns.violinplot(y=(data['alvura']['lab']-data['alvura']['inst']))
```

Fonte: Autor (2021)

4.3 Desenvolvimento do Analisador Virtual

A aplicação das metodologias de *Machine Learning* para desenvolvimento de um analisador virtual que estime os dados do laboratório por meio dos dados disponíveis dos instrumentos de medição *on-line* foi utilizada como uma métrica de acuracidade dos dados de laboratório, além de ser uma estratégia avançada de controle de processos.

A Figura 20 apresenta uma sequência para a obtenção de um analisador virtual.

Figura 20 - Fluxograma de obtenção de um analisador virtual

Fonte: Adaptada de Kadlec *et al*. (2009)

- **Primeira inspeção de dados**

  Ao gerar a base de dados, realiza-se uma análise geral com o objetivo de reconhecer erros primários

- **Seleção de dados históricos**

  Seleciona-se os dados a serem treinados

- **Pré-processamento de dados**

  Esta etapa inclui a normalização dos dados, detecção de *outliers* e nova seleção de dados relevantes.

  A normalização dos dados é feita através da Equação 7.

$$Z = \frac{x - \min(x)}{max(x) - min(x)} \quad \textbf{(7)}$$

- **Seleção, treinamento e validação do modelo**

  Realiza-se a seleção do tipo de modelo (regressão linear, árvores de decisão e florestas aleatórias), divisão da base de dados e métrica de avaliação da eficiência do modelo e dos parâmetros utilizados.

- **Manutenção**

  Após o desenvolvimento e implementação do analisador virtual, ele deve ser mantido sob observação e ajustado regularmente.

Para criar o algoritmo de desenvolvimento do analisador virtual, utiliza-se o pacote de *Machine Learning* para Python chamado “Scikit-learn”. As funções deste pacote para as árvores de decisão, florestas aleatórias e regressão linear estão representadas na Figura 21.

Figura 21 - Funções para obtenção do analisador virtual

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
```

Fonte: Autor (2021)

5 RESULTADOS E DISCUSSÃO

Os resultados das análises estatísticas e construção do analisador virtual foram discutidos nesta seção.

5.1 Análise Exploratória

A primeira aplicação estatística para a determinação dos desvios de cada variável, pode ser observada através da Tabela 1.

Tabela 1 - Desvio padrão das variáveis de processo

Variável	Laboratório	Instrumento
Teor Seco MQ1	0,83	0,74
Teor Seco MQ2	0,81	0,71
Teor Seco MQ3	1,09	0,99
Gramatura MQ1	23,35	22,91
Gramatura MQ2	18,44	24,47
Gramatura MQ3	22,68	27,88
Kappa	0,82	0,67
Alvura	0,40	0,42
Consistência	0,28	0,35

Fonte: Autor (2021)

Analisando os resultados de desvio padrão observa-se uma alta dispersão no conjunto de dados para a variável Gramatura. Entretanto, um desvio padrão pode ser considerado grande ou pequeno, dependendo da ordem de grandeza da variável (SHIMAKURA, 2005).

Uma maneira de se expressar a variabilidade dos dados tirando a influência da ordem de grandeza da variável é através do coeficiente de variação, definido pela razão entre o desvio padrão e a média dos dados. A Tabela 2 dispõe os resultados dos coeficientes de variabilidade para as variáveis de processo, expressa em porcentagem.

Tabela 2 - Coeficientes de variabilidade das variáveis de processo, em %

	Laboratório	Instrumento
Teor Seco MQ1	0,92	0,82
Teor Seco MQ2	0,90	0,79
Teor Seco MQ3	1,22	1,10
Gramatura MQ1	1,79	1,74
Gramatura MQ2	1,43	1,90
Gramatura MQ3	1,74	2,15
Kappa	4,90	3,90
Alvura	0,44	0,46
Consistência	2,79	3,53

Fonte: Autor (2021)

A partir do coeficiente de variação pode-se avaliar a homogeneidade do conjunto de dados e, consequentemente, se a média é uma boa medida para representar estes dados. Além disso, permite comparações entre variáveis de naturezas e unidades de medida distintas.

De acordo com Pimentel (1985), valores inferiores a 10% são considerados baixos e, com isso, homogêneos. Analisando a Tabela 2, é possível verificar que todos os valores são inferiores a 10%, indicando que os dados são homogêneos e apresentam baixa variabilidade em relação à média.

Os gráficos de linhas gerados para identificação do comportamento do Teor Seco MQ1, MQ2 e MQ3, Gramatura MQ1, MQ2, MQ3, Kappa, Alvura e Consistência, estão dispostos nas Figuras 22 a 30, respectivamente.

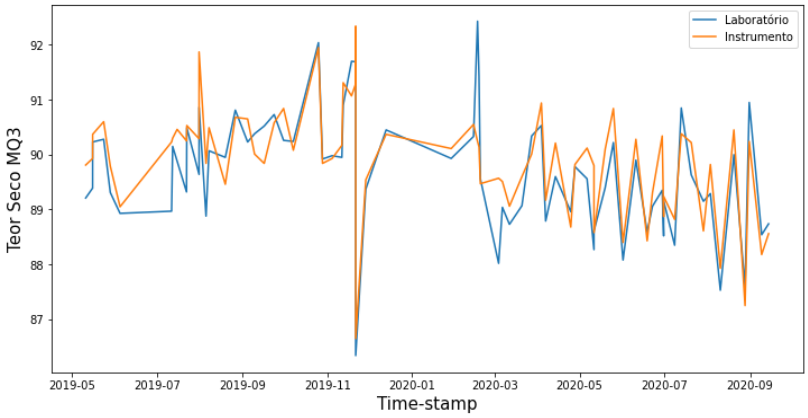
Figura 22 - Gráfico de linhas Teor Seco MQ1 – Laboratório e Instrumento

Fonte: Autor (2021)

Figura 23 - Gráfico de linhas Teor Seco MQ2 – Laboratório e Instrumento

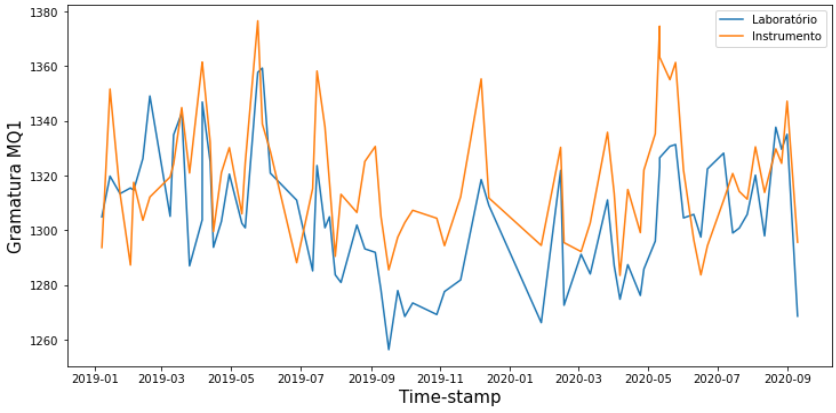
Fonte: Autor (2021)

Figura 24 - Gráfico de linhas Teor Seco MQ3 – Laboratório e Instrumento



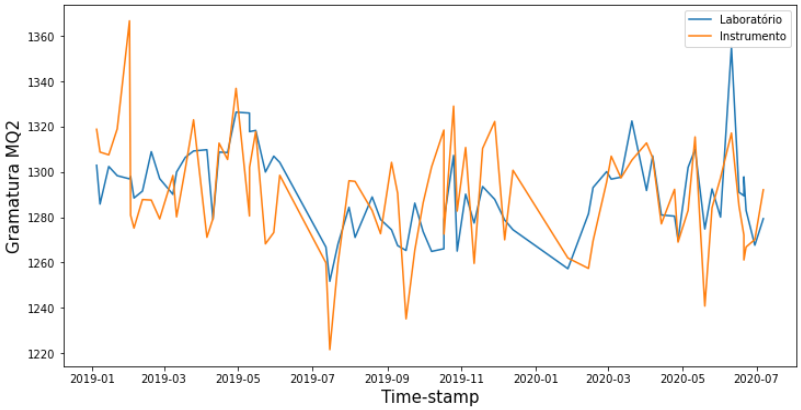
Fonte: Autor (2021)

Figura 25 - Gráfico de linhas Gramatura MQ1 – Laboratório e Instrumento



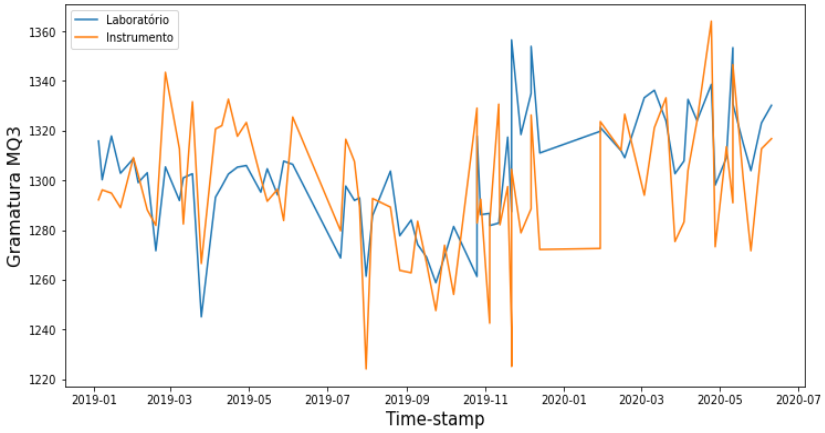
Fonte: Autor (2021)

Figura 26 - Gráfico de linhas Gramatura MQ2 – Laboratório e Instrumento



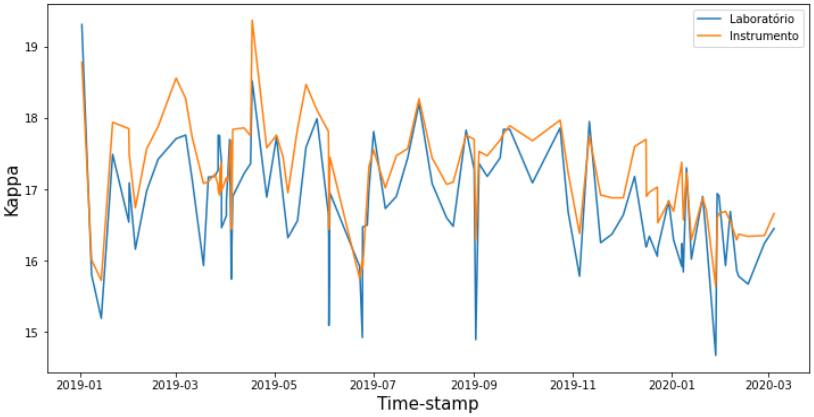
Fonte: Autor (2021)

Figura 27 - Gráfico de linhas Gramatura MQ3 – Laboratório e Instrumento



Fonte: Autor (2021)

Figura 28 - Gráfico de linhas Kappa – Laboratório e Instrumento



Fonte: Autor (2021)

Figura 29 - Gráfico de linhas Alvura – Laboratório e Instrumento

Fonte: Autor (2021)

Figura 30 - Gráfico de linhas Consistência – Laboratório e Instrumento

Fonte: Autor (2021)

Observa-se que os gráficos de linha apresentaram comportamento similar para variáveis de ambas medições, laboratório e instrumento. As Gramaturas MQ2 (Figura 26) e MQ3 (Figura 27) apresentaram valores irregulares, possivelmente por algum erro aleatório do instrumento e laboratório.

Os intervalos de confiança calculados para as variáveis de processo estão dispostos na Tabela 3.

Tabela 3 - Intervalos de confiança das variáveis de processo

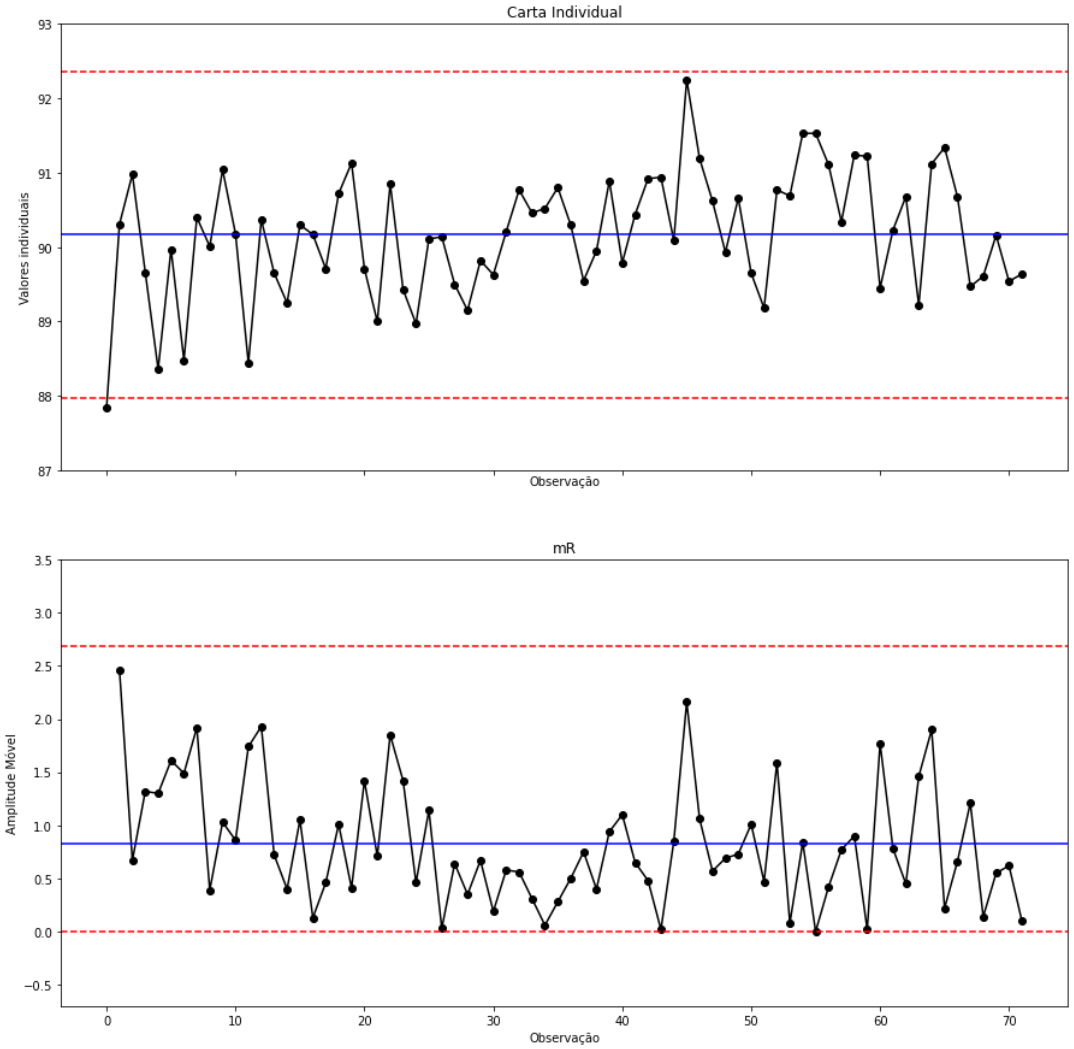
Variável	Laboratório	Instrumento
Teor Seco MQ1	90,17 ± 0,19	90,06 ± 0,17
Teor Seco MQ2	90,15 ± 0,18	90,09 ± 0,16
Teor Seco MQ3	89,71 ± 0,25	89,89 ± 0,23
Gramatura MQ1	1304,97 ± 5,36	1319,45 ± 5,25
Gramatura MQ2	1290,93 ± 4,22	1289,84 ± 5,61
Gramatura MQ3	1302,86 ± 5,20	1295,72 ± 6,39
Kappa	16,75 ± 0,16	17,20 ± 0,13
Alvura	89,81 ± 0,07	89,92 ± 0,07
Consistência	9,97 ± 0,04	10,04 ± 0,05

Fonte: Autor (2021)

Intervalos de confiança são usados para indicar a confiabilidade de uma estimativa, sua amplitude está associada à incerteza que temos acerca do parâmetro. Isto posto, as variáveis apresentaram resultados satisfatórios com baixas amplitudes, próximas à zero.

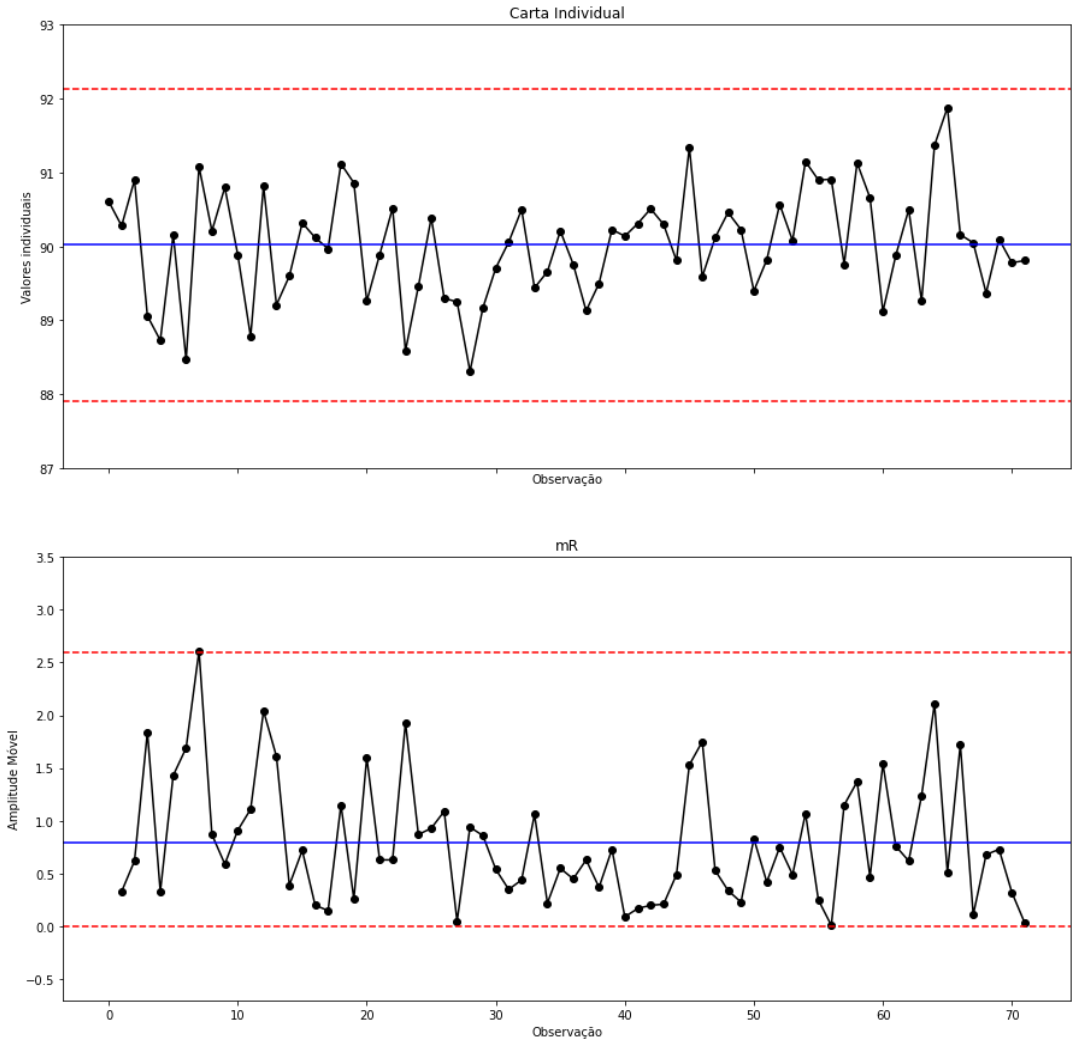
As cartas controle de média e amplitude móvel (mR) para a variável Teor Seco são apresentadas nas Figuras 31 a 36.

Figura 31 - Cartas de Controle para Teor Seco MQ1 laboratório



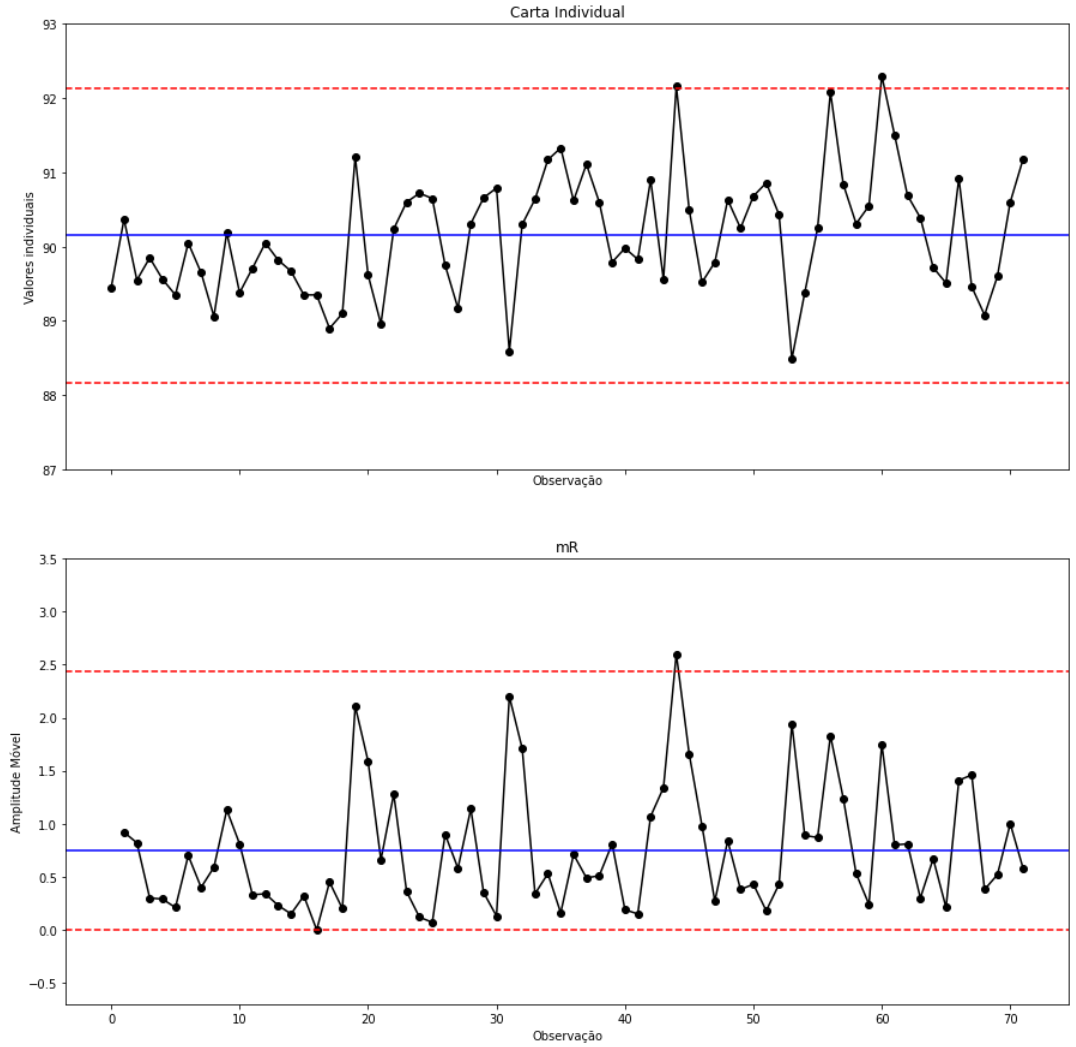
Fonte: Autor (2021)

Figura 32 - Cartas de Controle para Teor Seco MQ1 instrumento



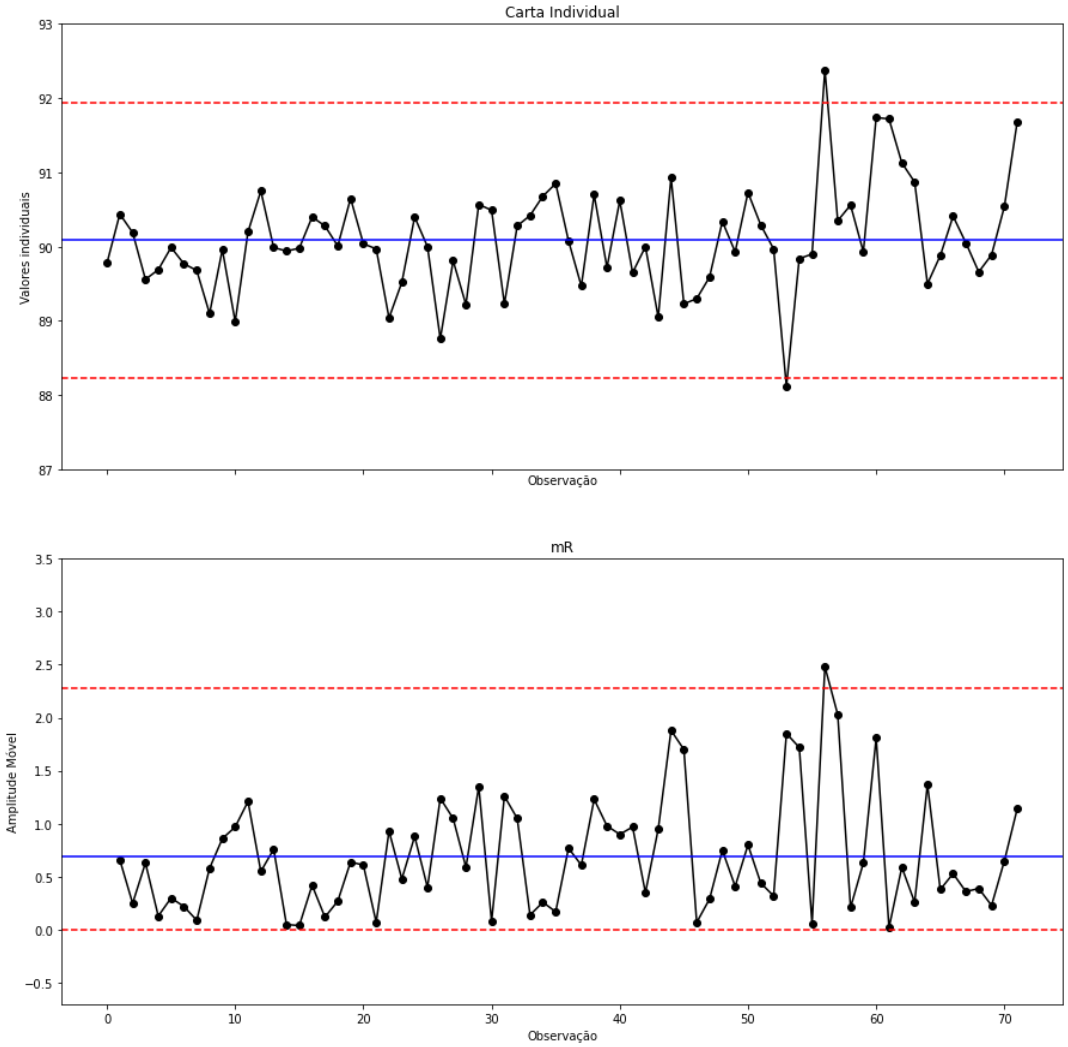
Fonte: Autor (2021)

Figura 33 - Cartas de Controle para Teor Seco MQ2 laboratório



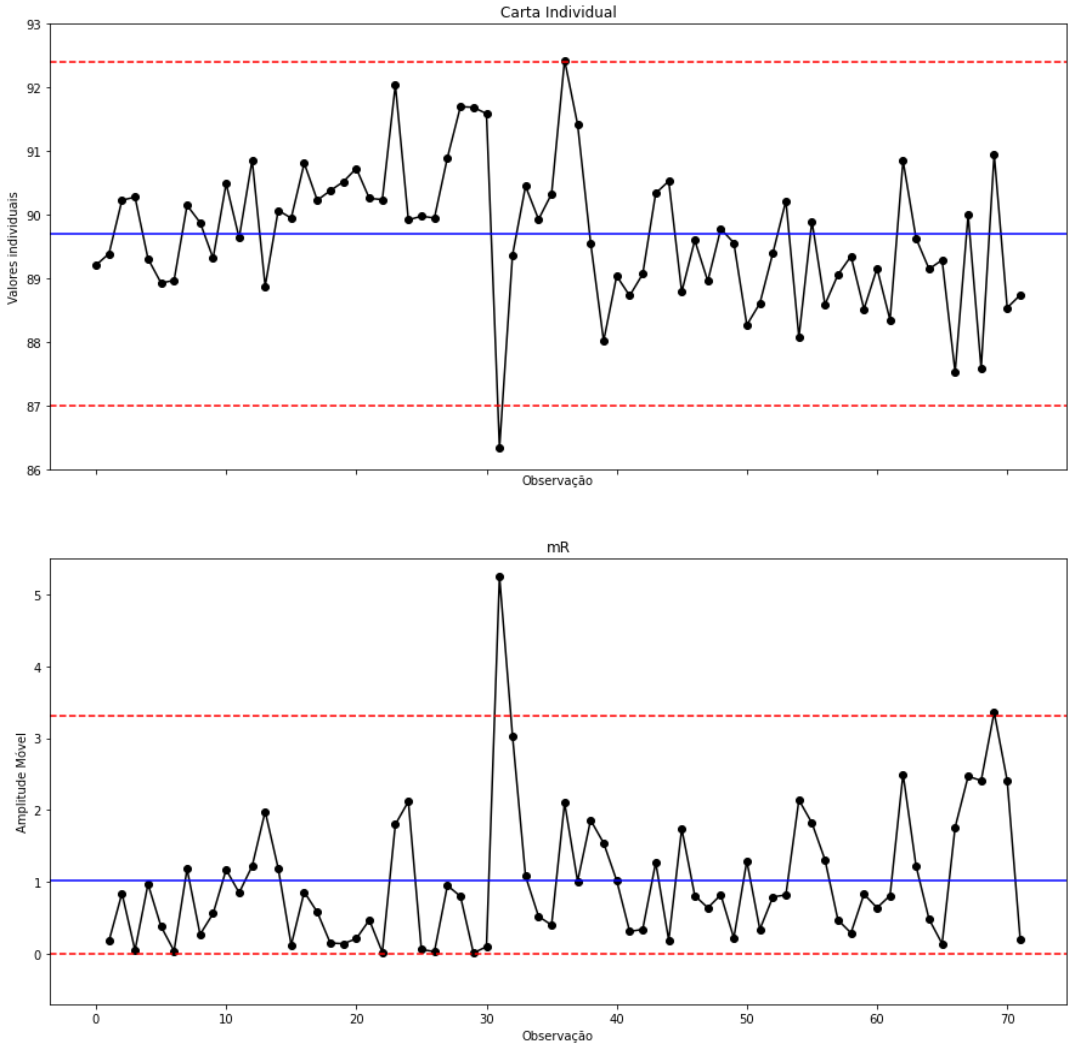
Fonte: Autor (2021)

Figura 34 - Cartas de Controle para Teor Seco MQ2 instrumento



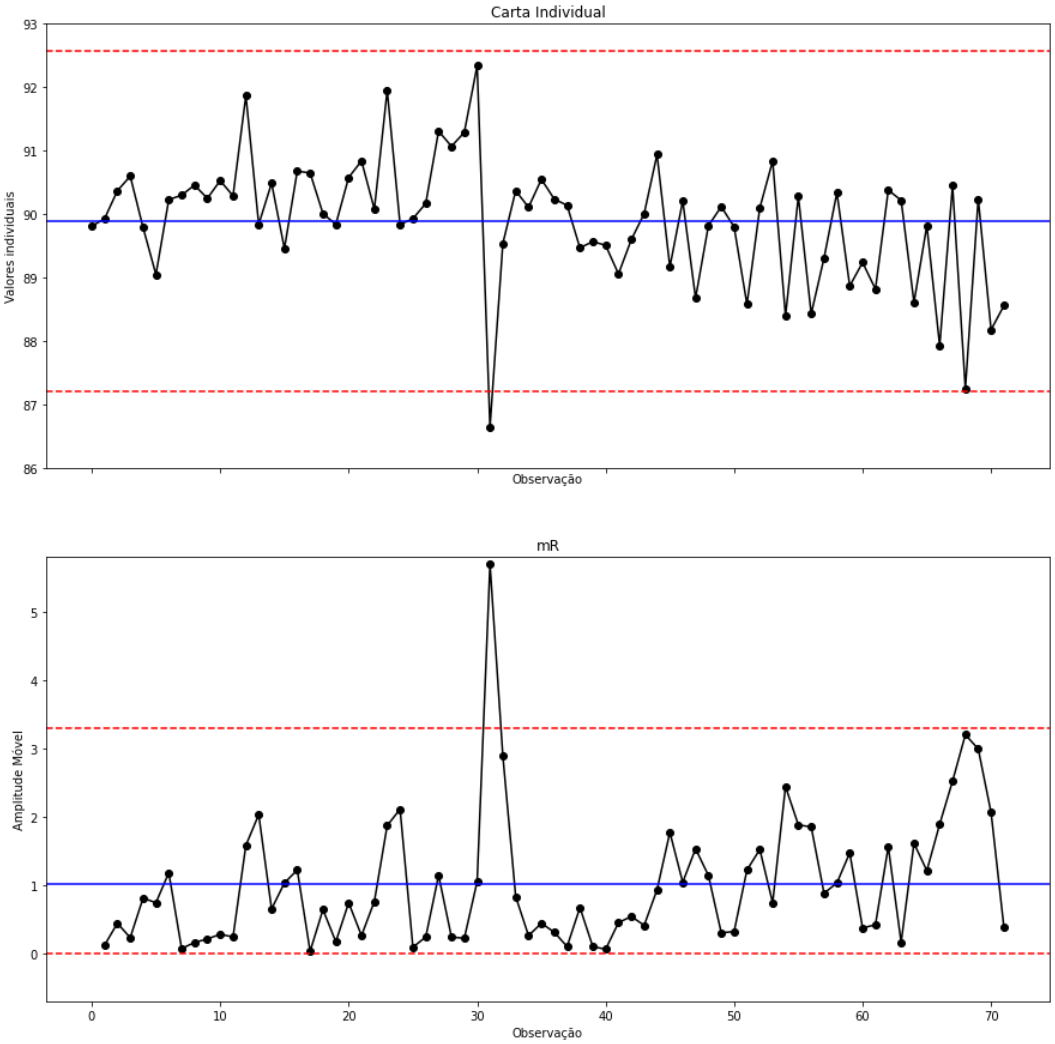
Fonte: Autor (2021)

Figura 35 - Cartas de Controle para Teor Seco MQ3 laboratório



Fonte: Autor (2021)

Figura 36 - Cartas de Controle para Teor Seco MQ3 instrumento



Fonte: Autores (2020)

As medidas do laboratório e instrumento para o Teor Seco MQ1, dispostas nas Figuras 31 e 32, respectivamente, apresentaram valores sob controle para as cartas de amplitude móvel. Analisando as cartas individuais, percebe-se que para Teor Seco MQ1 apenas o laboratório apresentou 1 valor fora de controle, sendo este o primeiro ponto do gráfico. Cada ponto fora do controle gera uma busca da causa através de análises das condições operacionais. Os resultados estatísticos dão partida para a análise, mas a explicação do que está acontecendo reside no próprio processo.

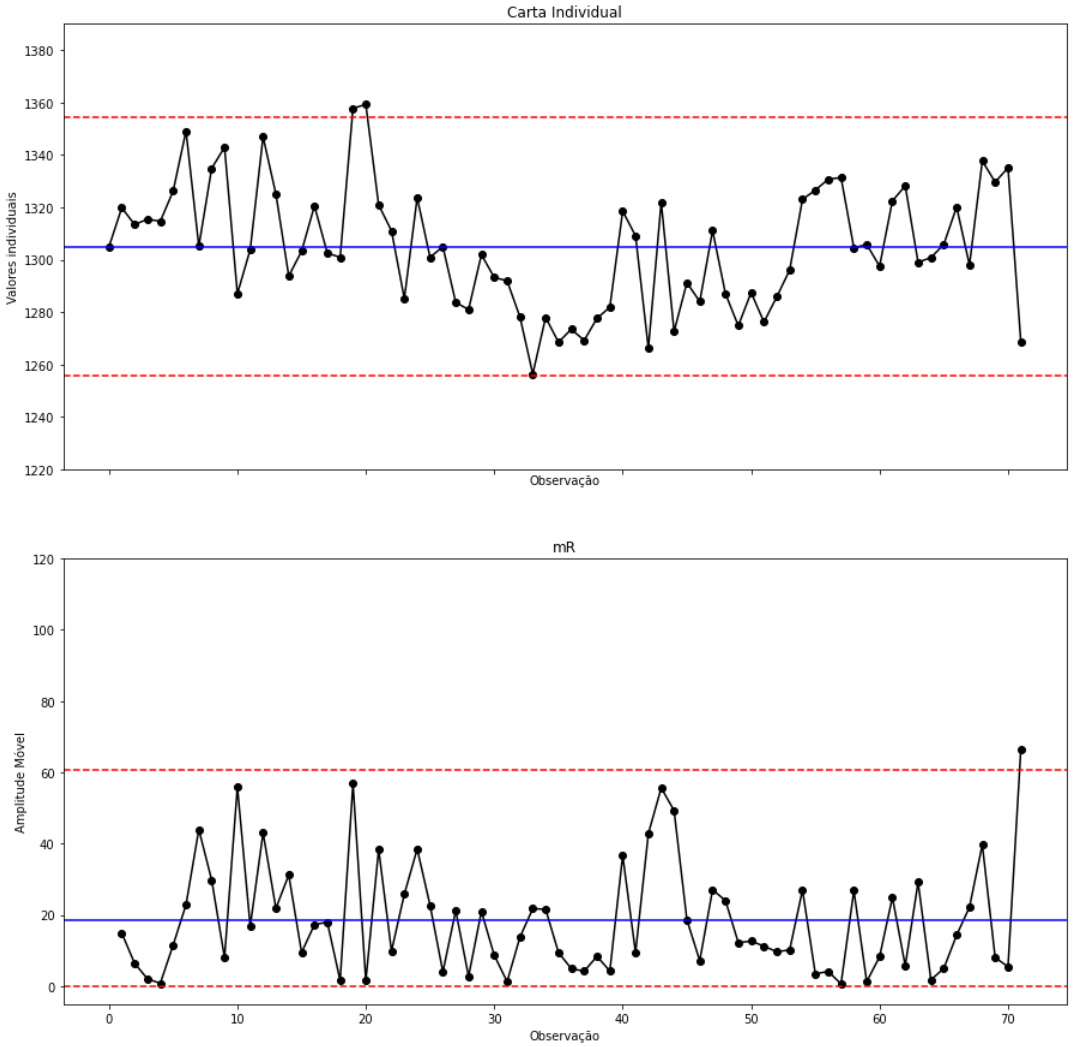
Para o Teor Seco MQ2 (Figuras 33 e 34), as cartas de amplitude móvel para laboratório e instrumento apresentam um valor em descontrole, entretanto, de acordo com as regras de interpretação de Sherwart (MONTGOMERY, 2009), um ponto fora do limite de controle se caracteriza como causas especiais atípicas, e não

como erros sistemáticos. As cartas individuais para laboratório e instrumento apresentaram 1 e 2 pontos fora de controle, respectivamente.

Observando as Figuras 35 e 36, constata-se, a partir das cartas de amplitude móvel do Teor Seco MQ3, que ocorreu erro devido às causas especiais. Já as cartas individuais da variável Teor Seco MQ3 demonstram 1 ponto fora de controle para as medições laboratoriais e instrumentais.

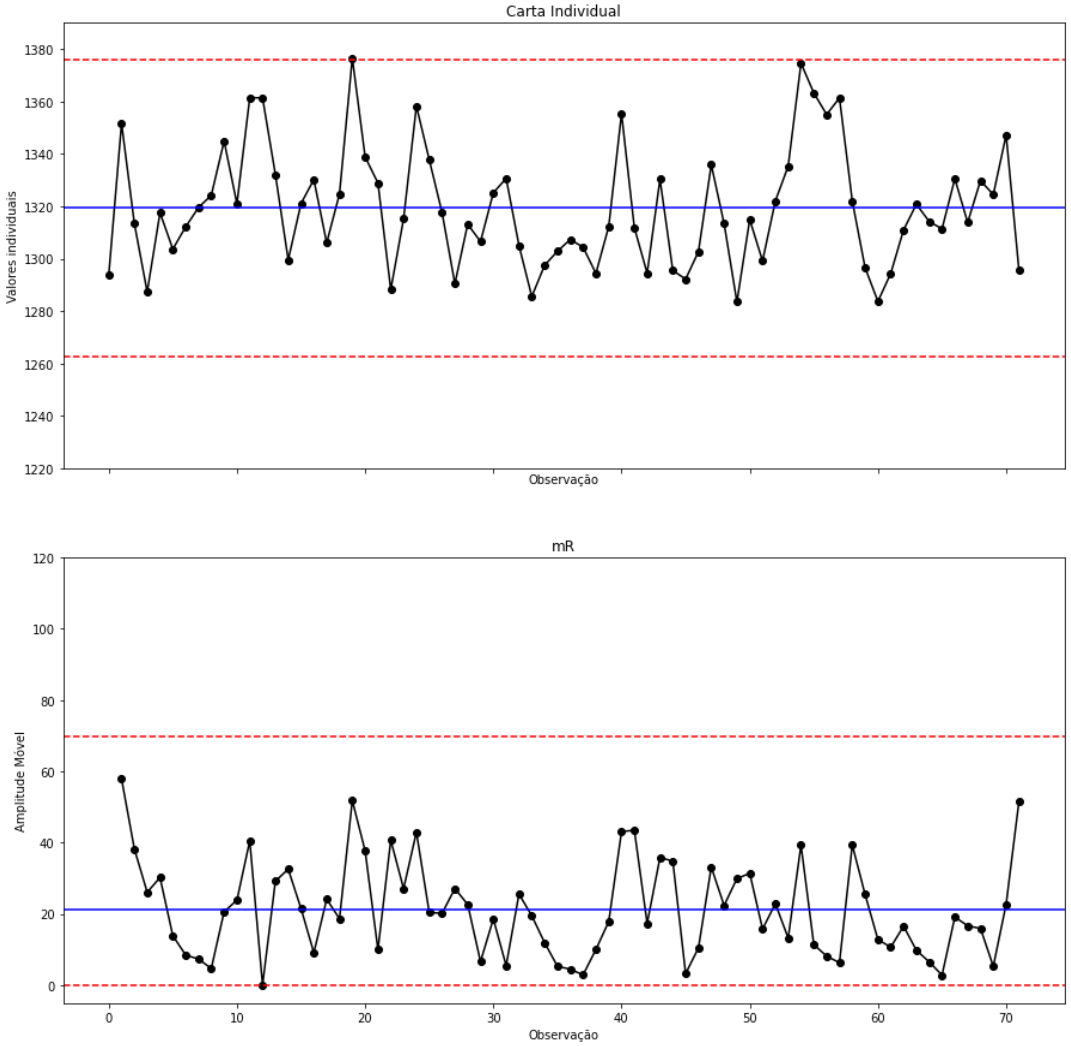
As cartas controle de média e amplitude móvel (mR) para a variável Gramatura são apresentadas nas Figuras 37 a 42.

Figura 37 - Cartas de Controle para Gramatura MQ1 laboratório



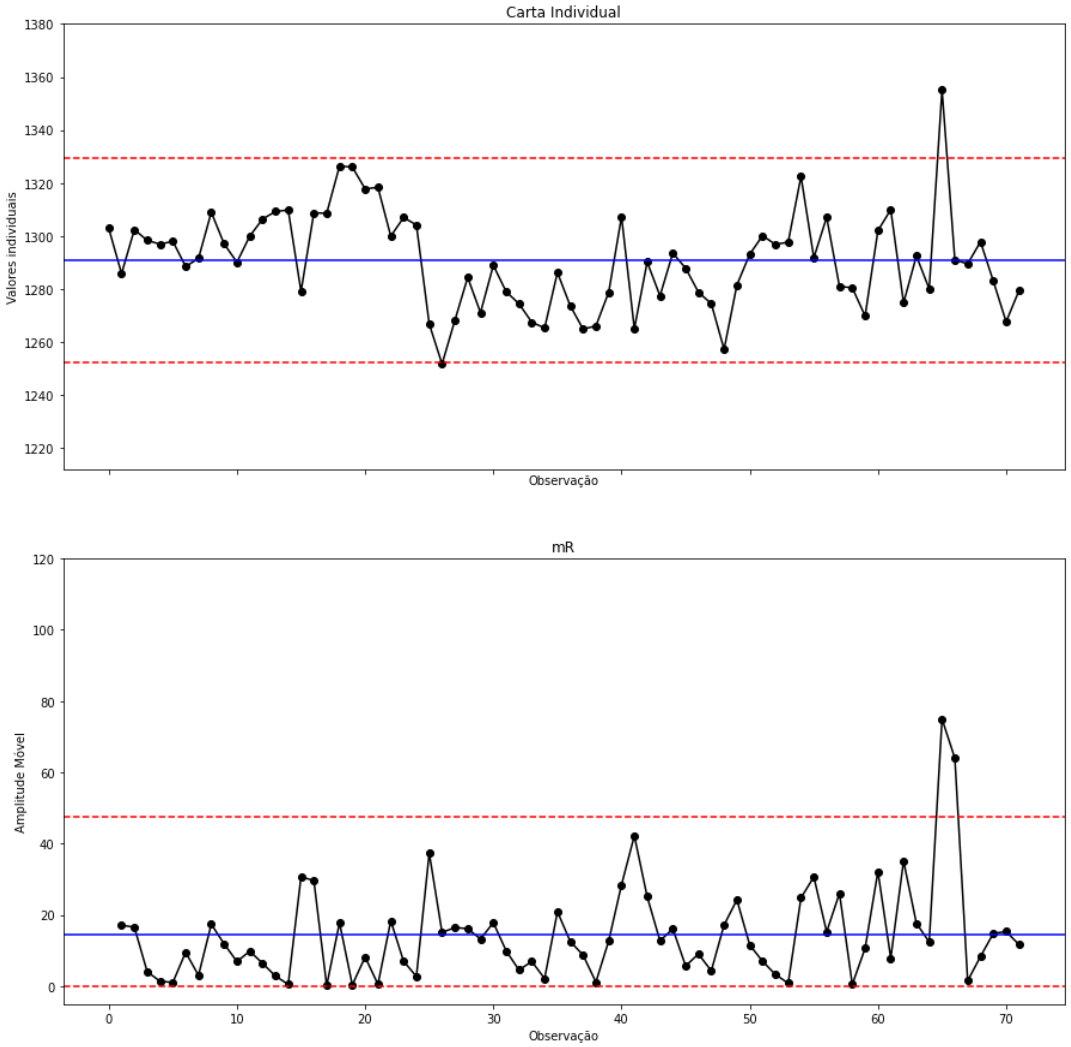
Fonte: Autor (2021)

Figura 38 - Cartas de Controle para Gramatura MQ1 instrumento



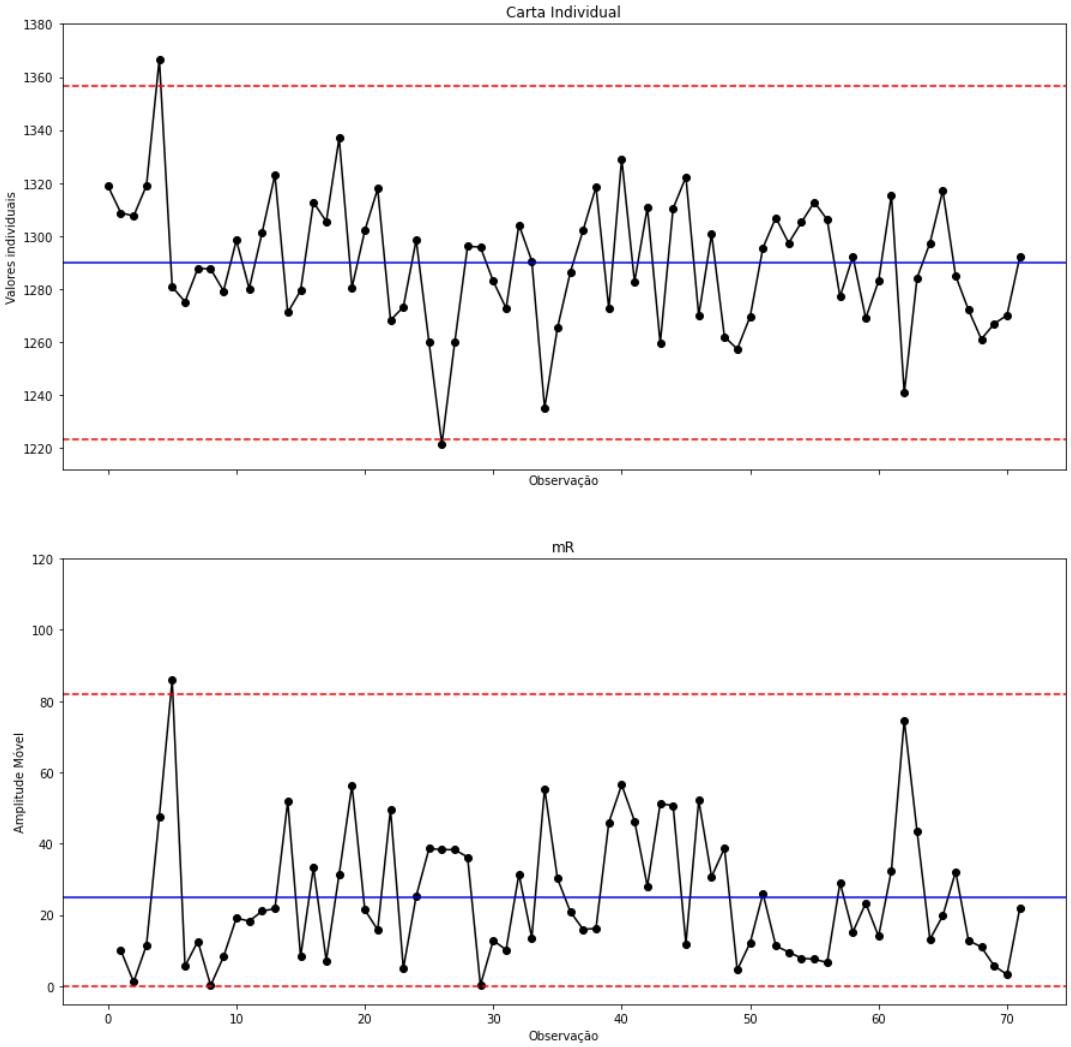
Fonte: Autor (2021)

Figura 39 - Cartas de Controle para Gramatura MQ2 laboratório



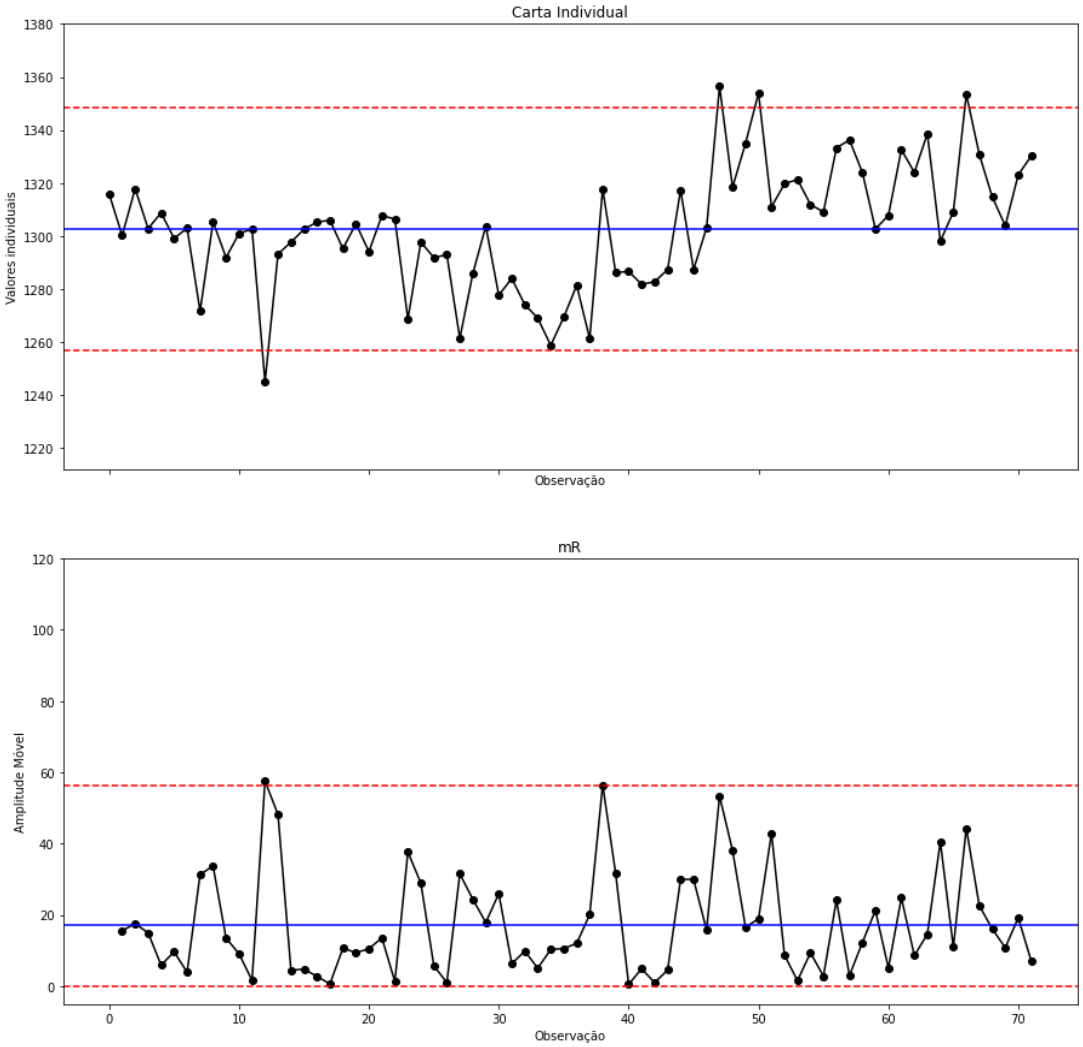
Fonte: Autor (2021)

Figura 40 - Cartas de Controle para Gramatura MQ2 instrumento



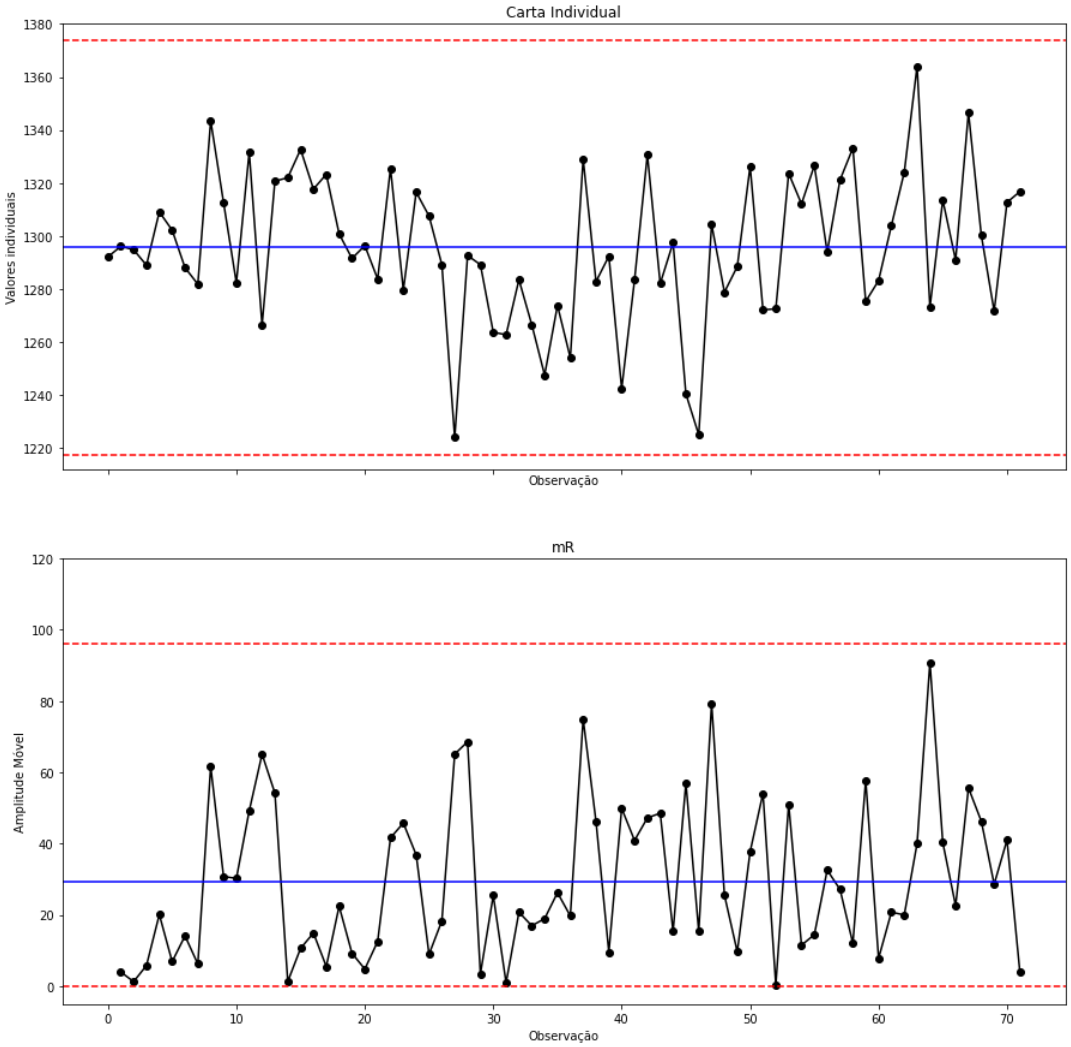
Fonte: Autor (2021)

Figura 41 - Cartas de Controle para Gramatura MQ3 laboratório



Fonte: Autor (2021)

Figura 42 - Cartas de Controle para Gramatura MQ3 instrumento



Fonte: Autor (2021)

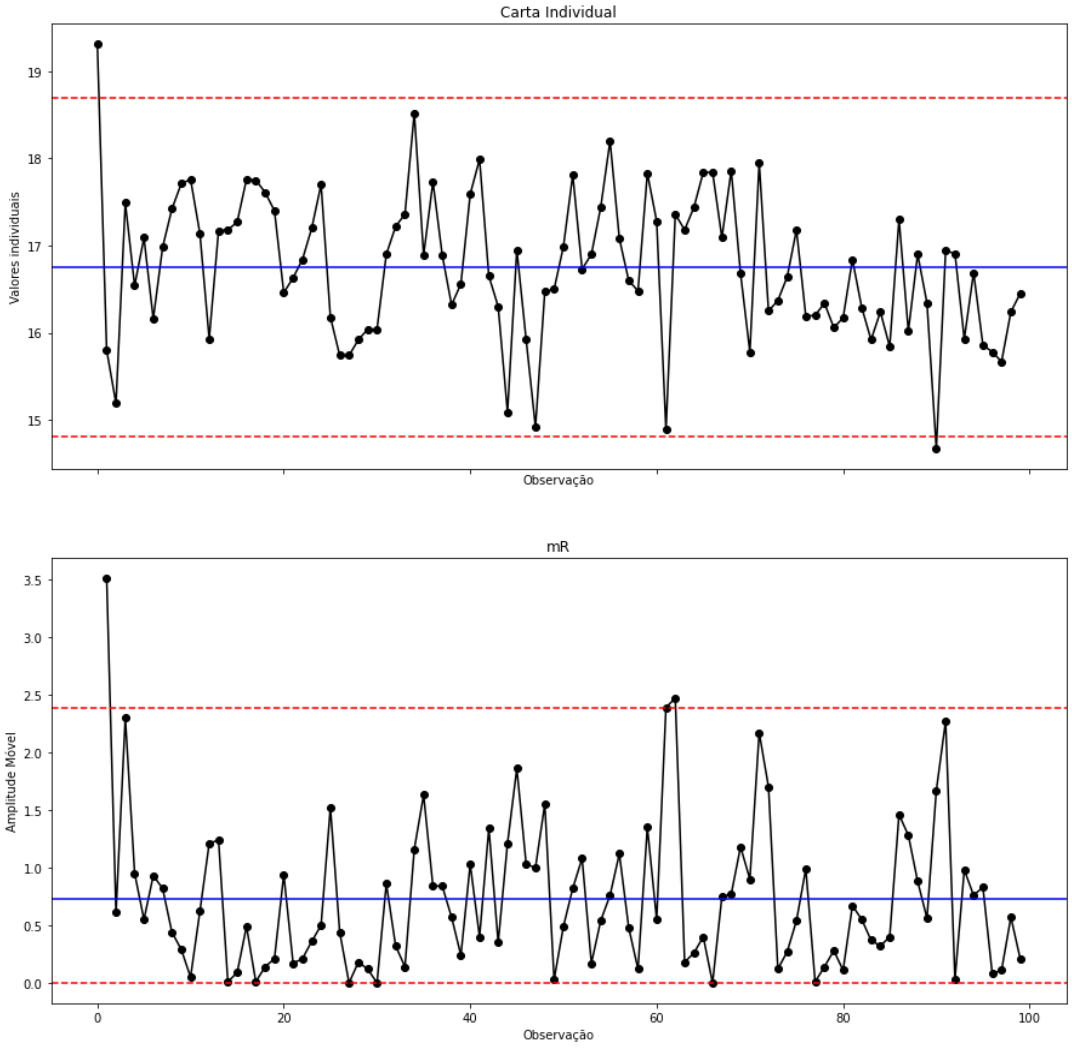
Com base na interpretação das cartas de amplitude média móvel para a Gramatura MQ1 (Figuras 37 e 38), o laboratório apresentou um valor fora de controle, se caracterizando como causas especiais atípicas. Os resultados das cartas individuais demonstram que 2 pontos estão fora de controle nas medições laboratoriais, e todos sob controle nas realizadas pelo instrumento.

As cartas mR para a Gramatura MQ2 dispostas nas Figuras 39 e 40, evidenciam que 2 pontos do laboratório e 1 medido pelo instrumento estão em descontrole, indicando que os limites calculados para as cartas individuais laboratoriais não são precisos, necessitando de estudo das possíveis causas. Já o instrumento revela erros por causas especiais atípicas. Considerando as cartas individuais, 1 e 2 pontos ficaram fora de controle para os métodos analítico e instrumental, respectivamente.

Analisando as Figuras 41 e 42, verifica-se a partir das cartas de amplitude móvel da Gramatura MQ3, que todos os pontos estão sob controle. Em relação às cartas individuais, somente o laboratório apresentou pontos fora de controle, com 4 pontos excedendo os limites.

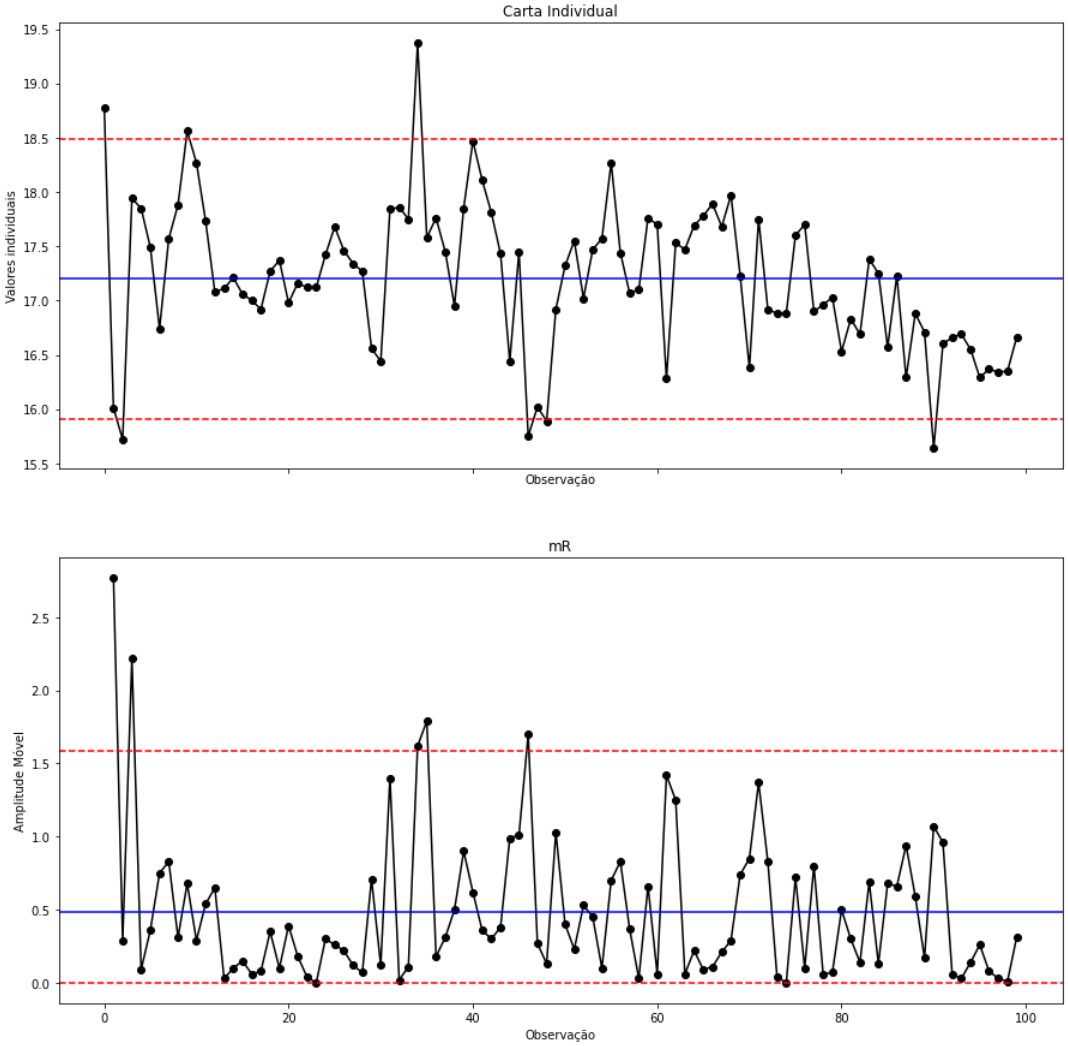
As cartas controle de média e amplitude móvel (mR) para as variáveis Kappa, Alvura e Consistência são apresentadas nas Figuras 43 a 44.

Figura 43 - Cartas de Controle para Kappa laboratório



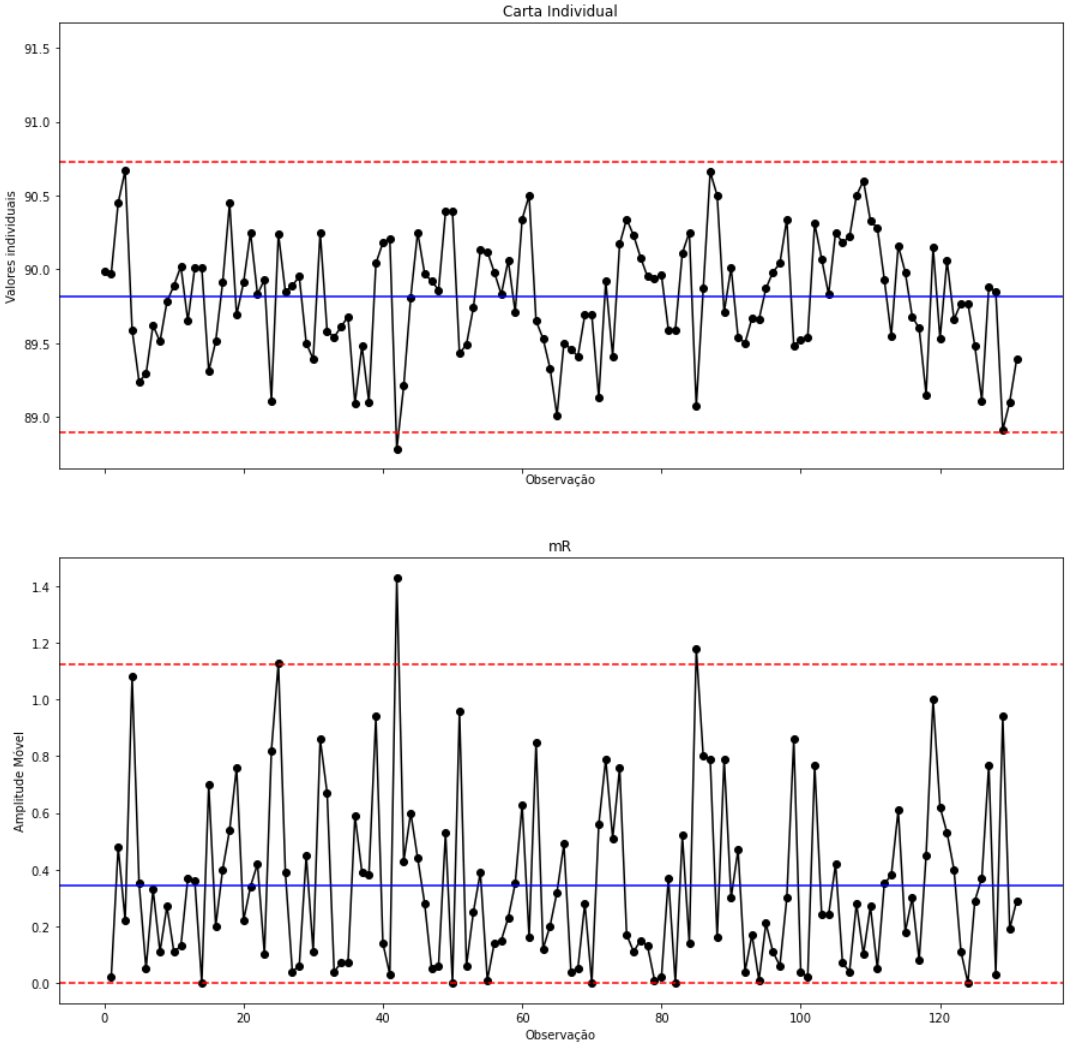
Fonte: Autor (2021)

Figura 44 - Cartas de Controle para Kappa instrumento



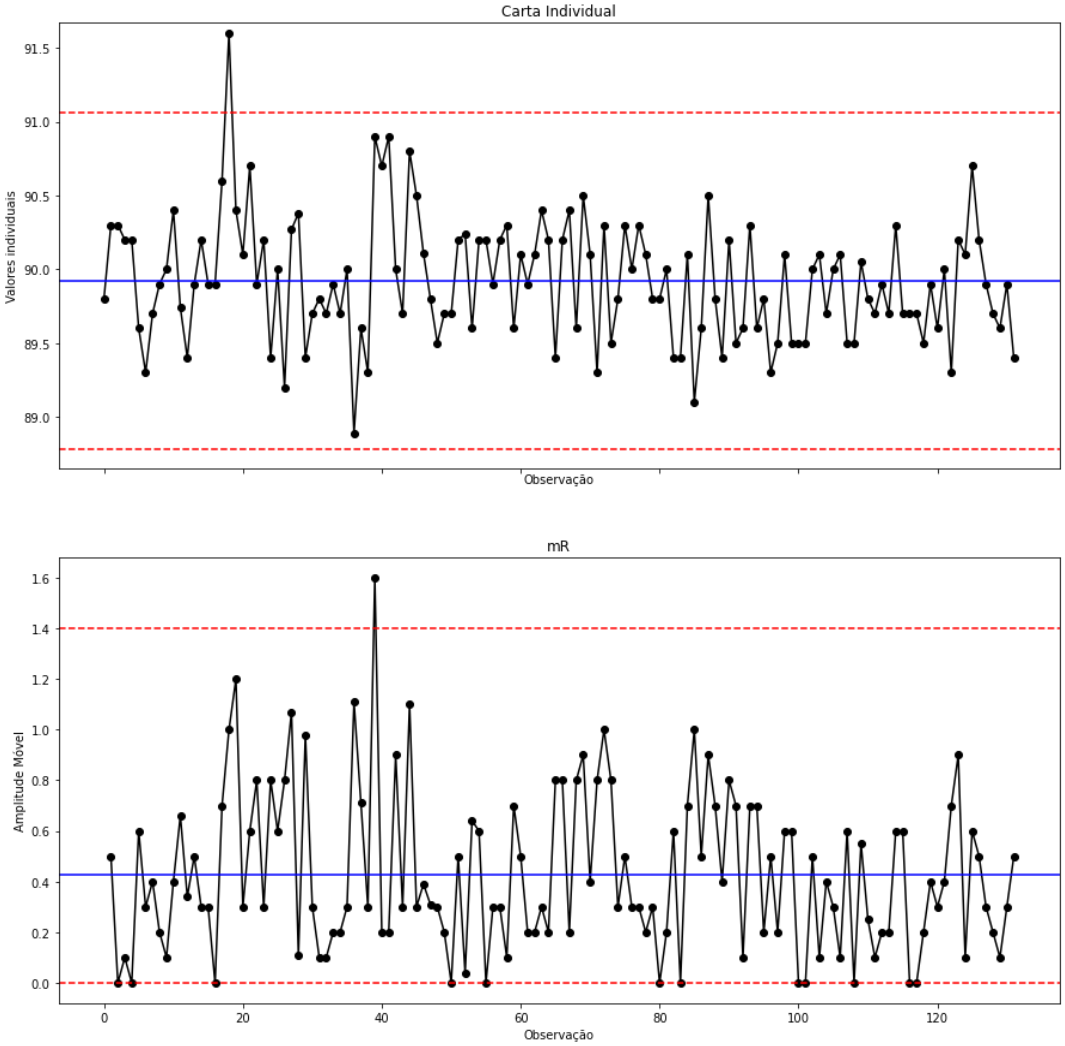
Fonte: Autor (2021)

Figura 45 - Cartas de Controle para Alvura laboratório



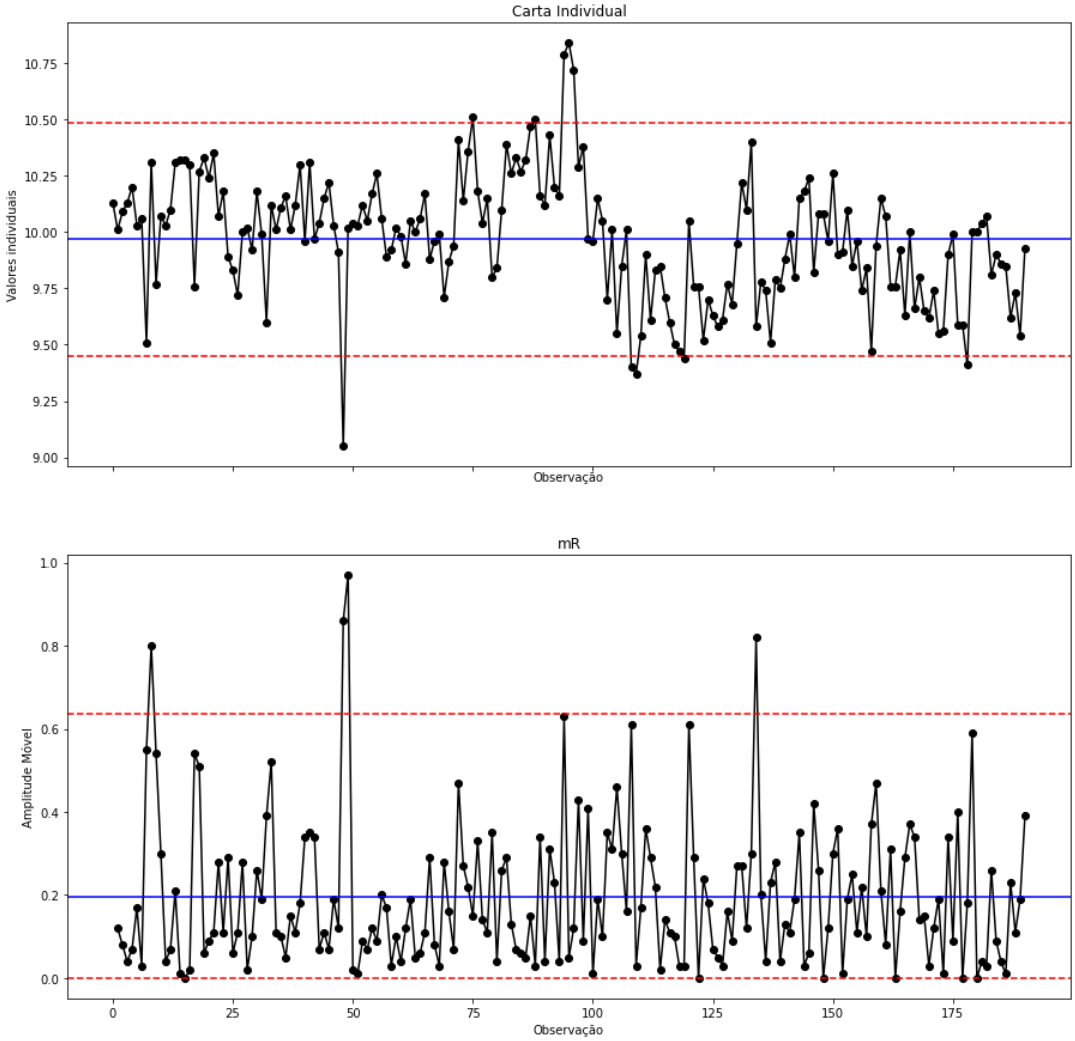
Fonte: Autor (2021)

Figura 46 - Cartas de Controle para Alvura instrumento



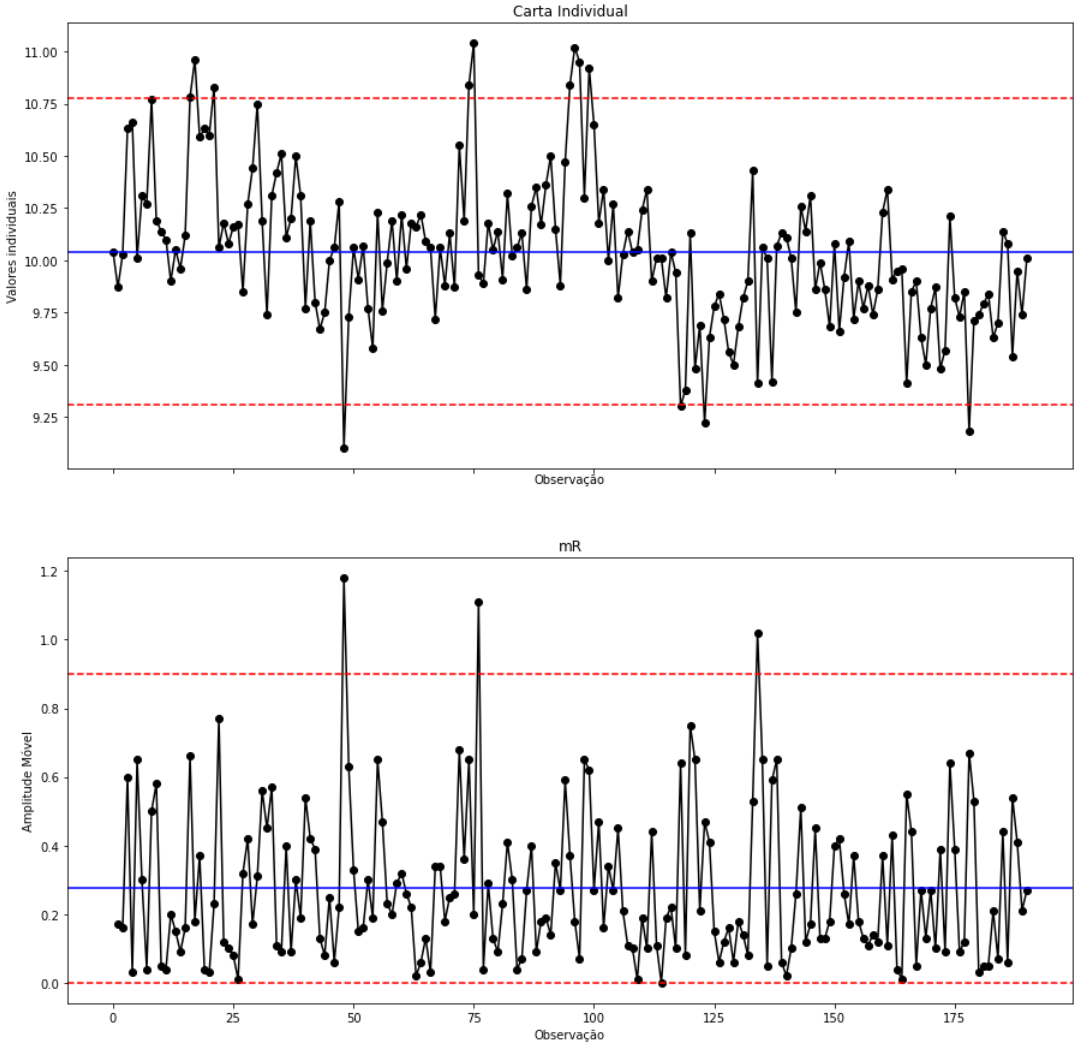
Fonte: Autor (2021)

Figura 47 - Cartas de Controle para Consistência laboratório



Fonte: Autor (2021)

Figura 48 - Cartas de Controle para Consistência instrumento



Fonte: Autor (2021)

As cartas de amplitude móvel para o Kappa (Figuras 43 e 44), apresentaram 2 e 4 pontos fora de controle, evidenciando baixa precisão nos limites de controle e necessidade de estudos de possíveis causas. Com relação às cartas individuais, o laboratório indica 2 pontos fora de controle, o que difere bastante dos 6 pontos encontrados nos dados do instrumento. Isso pode ser explicado pela variabilidade dos dados, que corroborou para limites de controle imprecisos e ocasionou mais pontos fora de controle do que efetivamente ocorreu.

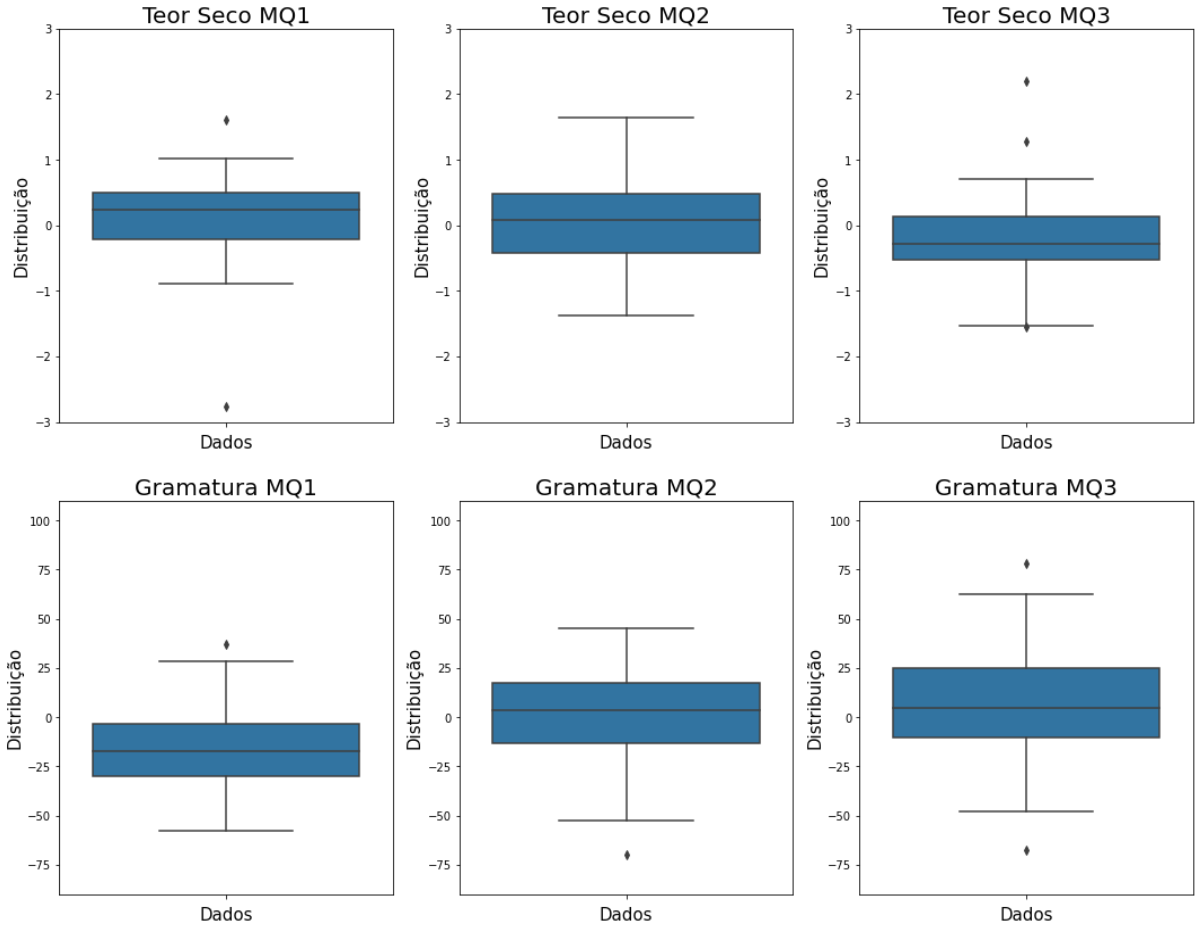
Os resultados das cartas de amplitude para Alvura podem ser vistos nas Figuras 45 e 46. As medições laboratoriais apresentaram 2 pontos acima do limite superior, logo necessitam de estudos de causas. A situação das medições instrumentais se classifica como atípica, com 1 ponto fora de controle. Observando as cartas individuais, é possível constatar que o laboratório e instrumento

apresentaram apenas 1 *outlier*. Ainda, é possível perceber uma maior homogeneidade na distribuição dos dados.

Assim como os dados de Kappa do instrumento (Figura 44), os dados para amplitude móvel das medições de laboratório e equipamento da Consistência (Figuras 47 e 48) estão fora de controle, indicando que os limites de controle não são precisos. As cartas individuais do laboratório e instrumento, apresentaram, respectivamente, 8 e 11 valores fora dos limites de controle. Estes valores podem ser explicados devido à baixa precisão dos limites, o que pode não condizer com a realidade.

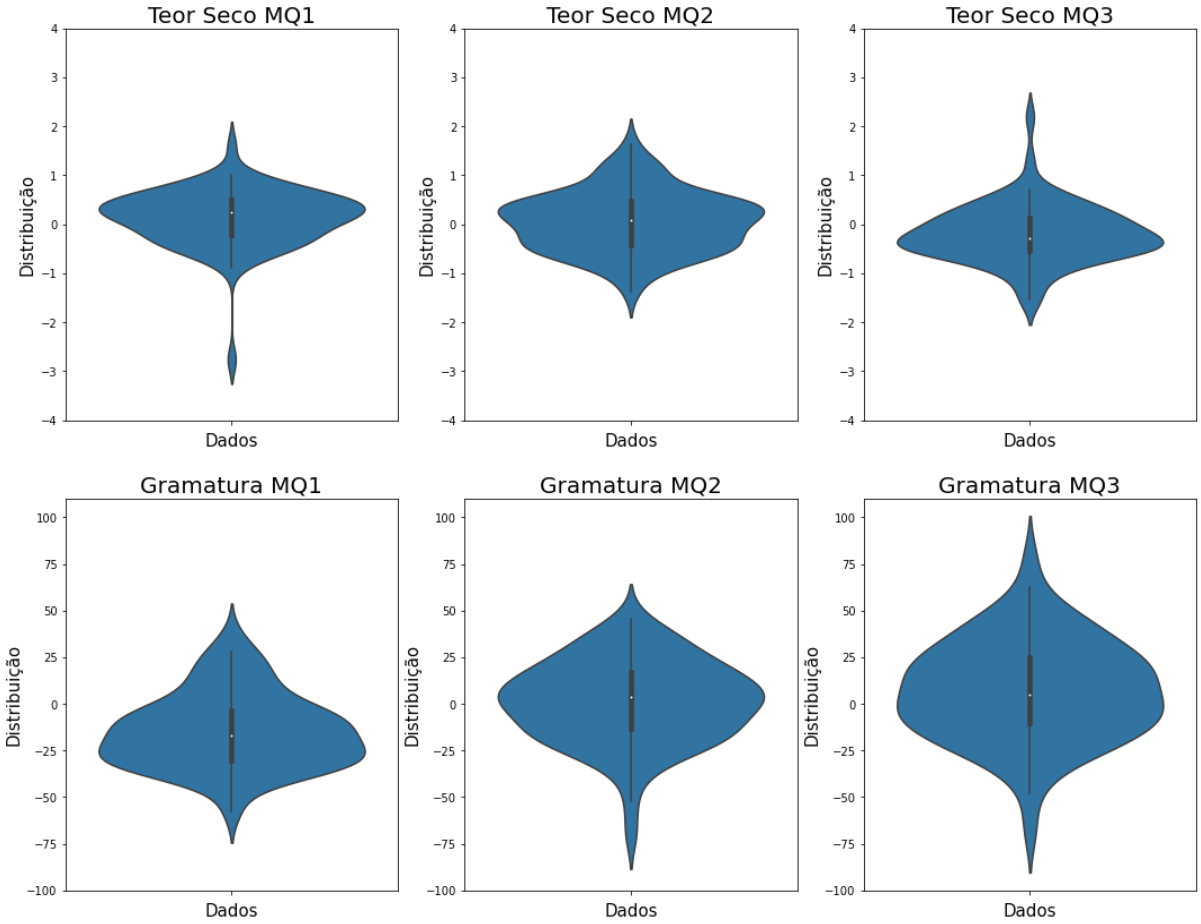
Os *Boxplots* e *Violinplots* para as variáveis Teor Seco MQ1, MQ2 e MQ3 e Gramatura MQ1, MQ2 e MQ3, Kappa, Alvura e Consistência são vistos nas Figuras 49 a 52.

Figura 49 - *Boxplots* para Teor Seco MQ1 a MQ3 e Gramatura MQ1 a MQ3



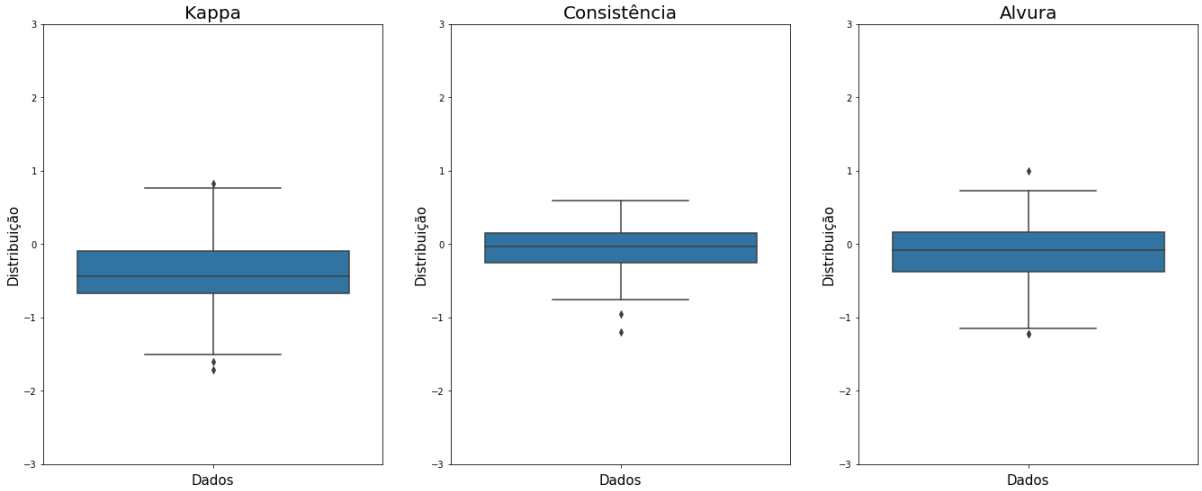
Fonte: Autor (2021)

Figura 50 - *Violinplots* para Teor Seco MQ1 a MQ3 e Gramatura MQ1 a MQ3



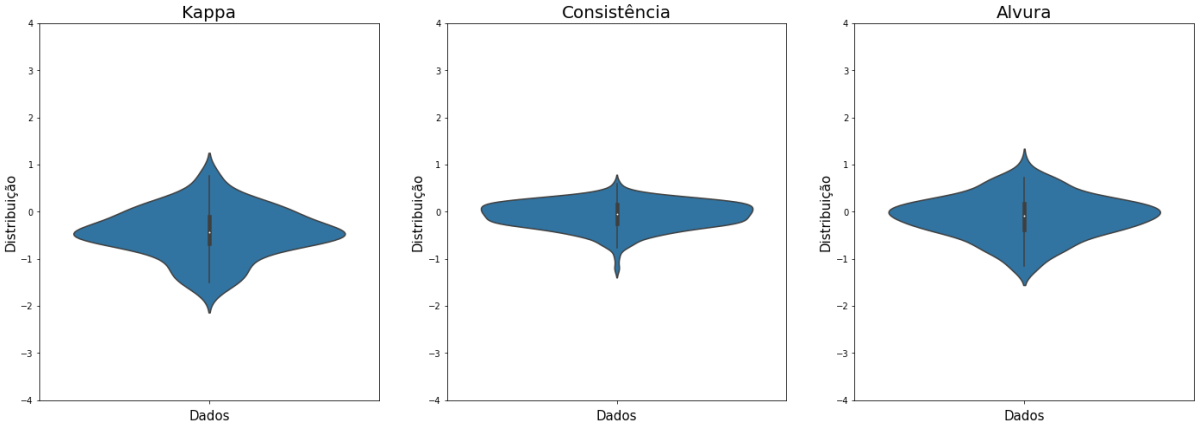
Fonte: Autor (2021)

Figura 51- *Boxplots* para Kappa, Consistência e Alvura



Fonte: Autor (2021)

Figura 52 - *Boxplots* para Kappa, Consistência e Alvura



Fonte: Autor (2021)

Analisando-se os gráficos das Figuras 49 a 52, é possível perceber que o pico das curvas se concentraram próximos ao zero na maior parte dos casos. Na Figura 49, a Gramatura MQ1 apresentou erro mais deslocado que as demais variáveis, em torno de -25, evidenciando considerável diferença entre dados. Além disso, os gráficos revelam que a variável com menor variabilidade é a Consistência (Figura 51). As medidas de dispersão ou variabilidade são simbolizadas pelas alturas da caixa e da haste dos *boxplots*. O tamanho da caixa representa o intervalo interquartílico, ou seja, a amostra compreendida entre o primeiro e terceiro quartis, sendo 50% da amostra total. Entre o primeiro quartil e a mediana estão 25%, os outros 25% ficam entre a mediana e terceiro quartil. O tamanho das hastes é definido pelos valores do intervalo entre o primeiro quartil e limite inferior, e do terceiro quartil ao limite superior.

Em relação às assimetrias, definidas pela posição relativa da mediana no espaço entre primeiro e terceiro quartis, são caracterizadas como positiva quando a linha da mediana está próxima ao primeiro quartil, e quando a posição é mais próxima ao terceiro quartil, os dados são assimétricos negativos (OLIVEIRA, 2019). O Teor Seco MQ1 e MQ2 apresentaram comportamento negativo, já a MQ3 possui uma assimetria positiva. As Gramaturas MQ1 e MQ2 apresentaram baixa assimetria, sendo a primeira ligeiramente positiva, e a segunda ligeiramente negativa. A Gramatura MQ3 obteve comportamento positivo (Figuras 49 e 50). A variável Kappa demonstra posição positiva, enquanto a Consistência e Alvura apresentam comportamento levemente negativo (Figuras 51 e 52). Os *violinplots* de Consistência e Alvura demonstram comportamento mais simétrico em relação às outras variáveis

(Figura 52), corroborando com a baixa variabilidade dos intervalos de confiança da Tabela 3.

Em termos de valores discrepantes, todas as variáveis contêm *outliers*, exceto o Teor Seco MQ2. Isso pode ser devido à erros espúrios de ambas técnicas de medição.

5.2 Analisador Virtual

Os modelos foram construídos a partir dos dados da variável Kappa, que apresentou melhores resultados estatísticos.

- **Regressão Linear**

Os indicadores da capacidade preditiva do modelo de Regressão Linear são parâmetros estatísticos que comparam o conjunto de dados observados ao conjunto correspondente de dados simulados ou preditos. Os resultados do modelo de Regressão Linear estão dispostos na Tabela 4.

Tabela 4 - Resultados modelo de Regressão Linear

Coeficiente	***R²***	**MAE**	**MSE**	**RMSE**
0,769	0,592	0,469	0,392	0,626

Fonte: Autor (2021)

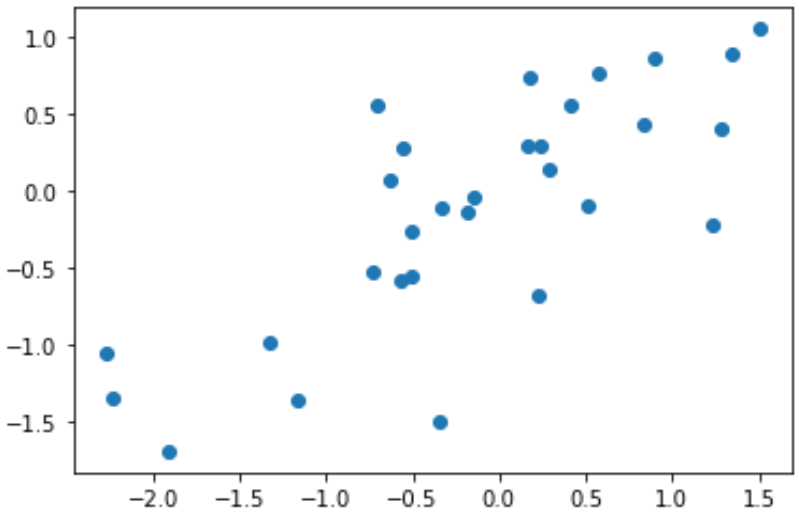
O parâmetro *R²* é utilizado para medir o grau de correlação linear do modelo predito. Nota-se que houve um nível de representatividade do modelo foi moderadamente boa visto que o modelo explica 59,6% da variável dependente. Observa-se que as métricas MSE e RMSE apresentam valores baixos, 0,392 e 0,626, respectivamente.

Interpreta-se o valor do parâmetro RMSE como uma medida do desvio médio entre observado e predito, este é o valor elevada ao quadrado. A verdadeira média dos desvios é descrita pelo MSE, que também representa o desvio médio entre observado e predito. Comparando o RMSE e o MSE, o primeiro dá um peso maior para desvios grandes (pois são elevados ao quadrado), enquanto o MSE dá um peso igual a todos os desvios.

Em diversos trabalhos, como de Zanata (2005), TakahashI (2006), Kano *et al.* (2000) e Piang (2005) *apud* Morais Junior (2011), a confiabilidade do analisador é expressa medindo o Erro Médio Quadrático (MSE), entre o valor estimado pelo analisador e o valor esperado.

Foi montado um diagrama de dispersão para visualizar os valores gerados pelo modelo. A Figura 53 mostra o resultado das predições.

Figura 53 - Diagrama de dispersão da predição do modelo Regressão Linear



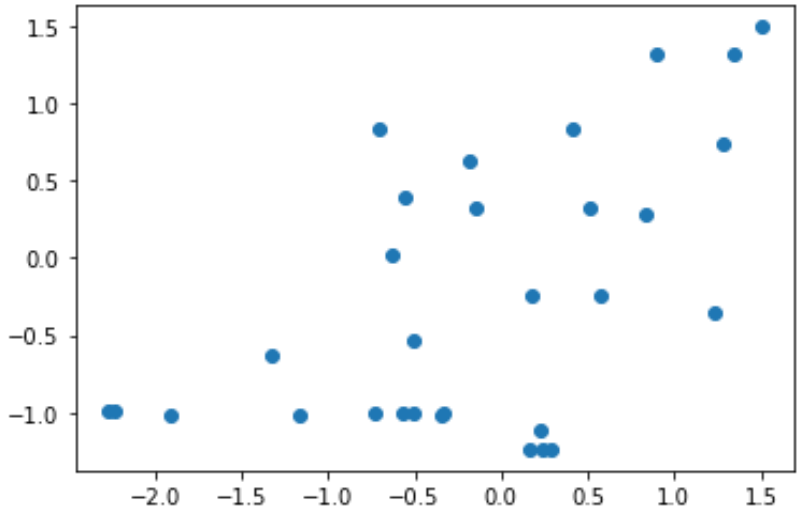
Fonte: Autor (2021)

Analisando-se o diagrama da Figura 53, é possível perceber que os valores preditos se mostraram dispersos. Uma função linear característica apresenta comportamento de uma reta crescente, o que não foi observado no resultado do modelo de Regressão Linear.

- **Árvores de Decisão**

O modelo de Árvores de Decisão apresentou R^2 de 0,198, se mostrando pouco correlativo. A Figura 54 demonstra o diagrama de dispersão gerado pelo modelo das Árvores de Decisão, observa-se que a distribuição dos dados é dispersa e não apresenta comportamento linear.

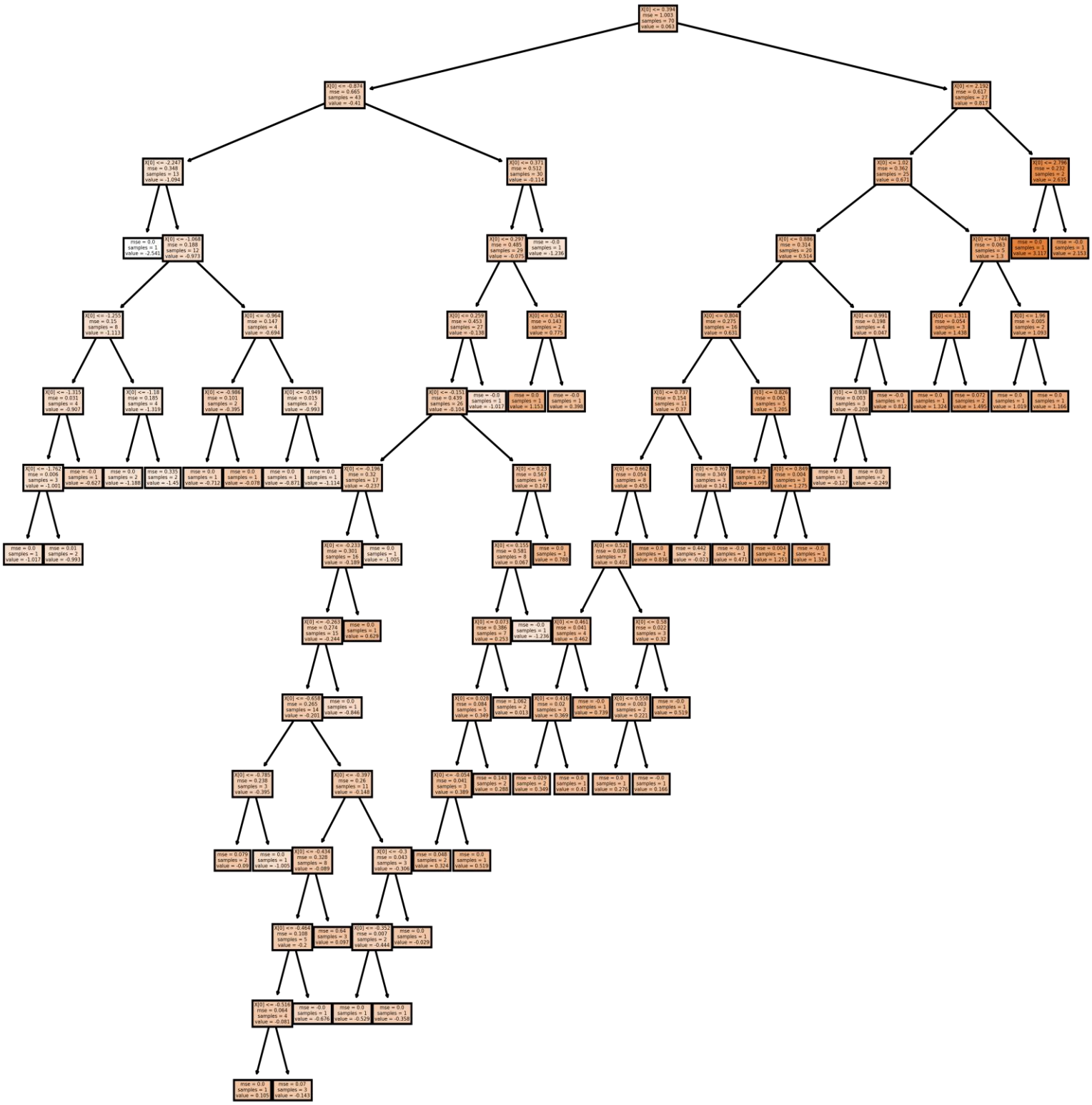
Figura 54 - Diagrama de dispersão da predição do modelo Árvores de Decisão



Fonte: Autor (2021)

A árvore de decisão gerada a partir do modelo pode ser vista na Figura 55.

Figura 55 - Resultado do modelo de Árvores de Decisão



Fonte: Autor (2021)

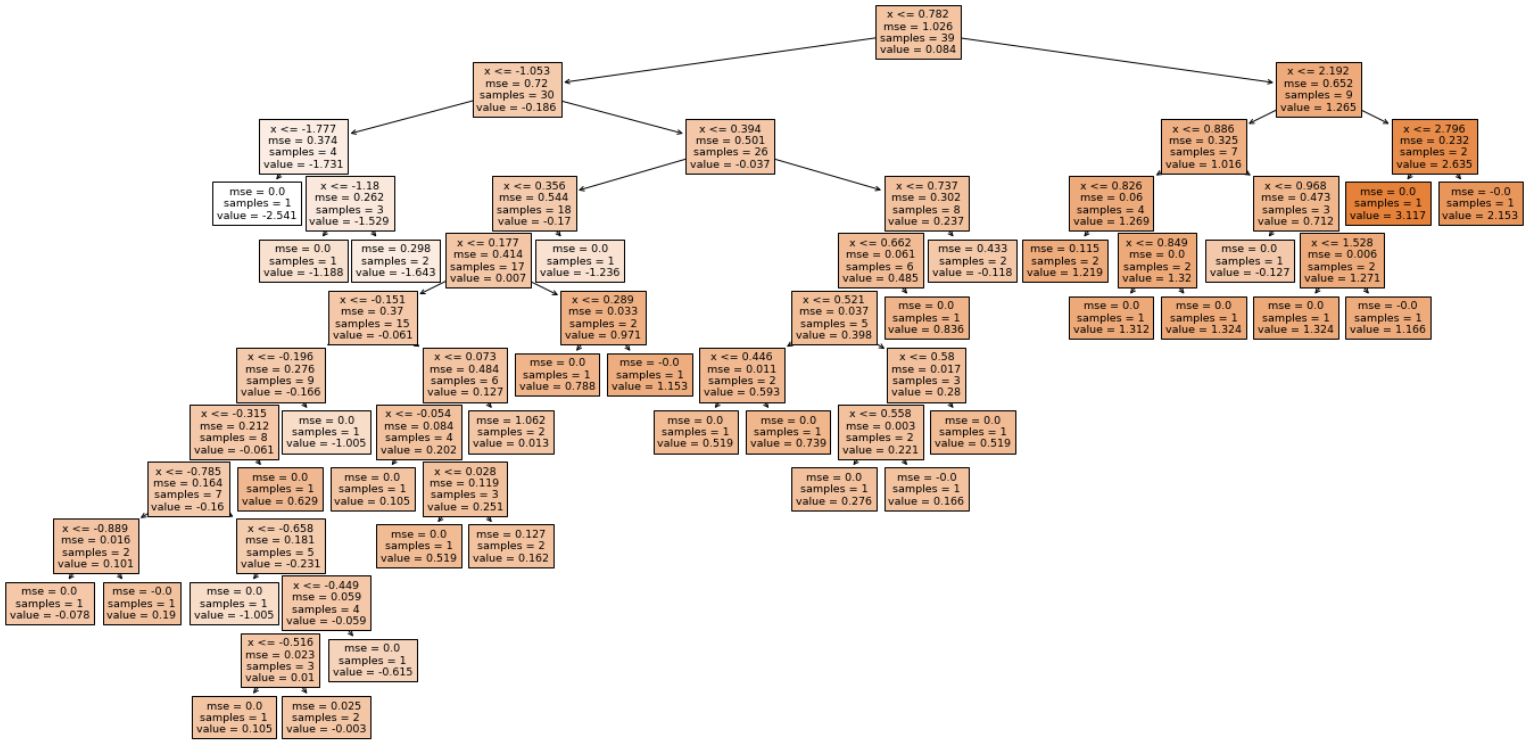
Os nós da árvore são subdivididos até que o modelo não identifique maiores ganhos de informação. Assim, chega aos nós terminais, que não possuem bifurcações. Analisando os nós dessa árvore, a primeira linha indica a variável e o valor para dividir esse nó. O MSE descreve o valor do erro quadrático médio do nó, enquanto que *samples* descreve o número de dados.

O caminho representado por menor ou igual a -0,874 e menor ou igual a 2,192 representam 43 e 27 conjuntos de dados, respectivamente. A partir da Árvore gerada na Figura 55, construiu-se a Floresta Aleatória.

- **Florestas Aleatórias**

O grau de representatividade do modelo de Florestas Aleatórias encontrado foi de 0,442. O resultado do modelo pode ser visto na Figura 56.

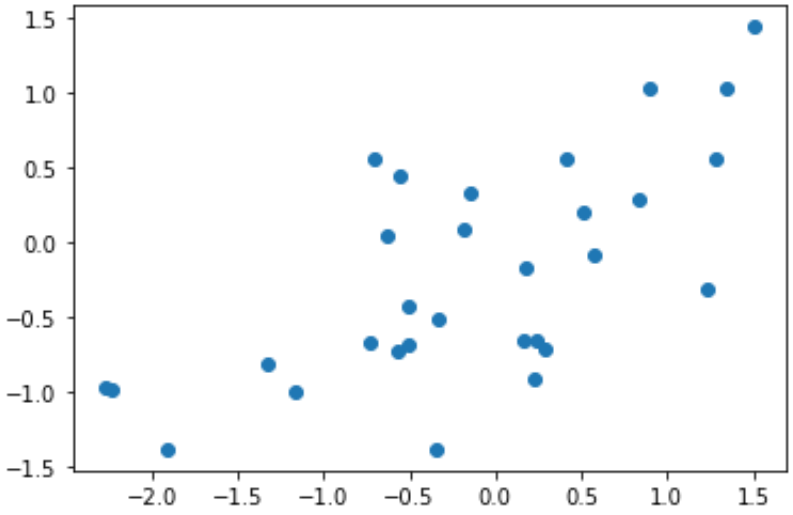
Figura 56 - Resultado do modelo de Florestas Aleatórias



Fonte: Autor (2021)

O caminho representado por menor ou igual a -0,874 e menor ou igual a 1,02 representam 24 e 20 conjuntos de dados, respectivamente. A Figura 57 demonstra o diagrama de dispersão gerada pelo modelo.

Figura 57 - Diagrama de dispersão da predição do modelo Florestas Aleatórias



Fonte**:** Autor (2021)

A partir da Figura 57 é possível perceber grande dispersão dos dados, o que corrobora com o resultado do parâmetro R^2 (0,442), indicando que o modelo não prevê satisfatoriamente os dados do laboratório.

5.3 Otimização via Busca em Grade

Com objetivo de testar as melhores combinações e obter melhor acurácia do modelo, otimizou-se o modelo de Florestas Aleatórias através da Busca em Grade, que testa todas as combinações possíveis e seleciona os resultados que obtiveram o menor erro. Os parâmetros analisados para otimização são dados conforme o Quadro 2. Já a Tabela 5 demonstra o intervalo de cada parâmetro e o melhor valor encontrado pelo método.

Quadro 2 - Parâmetros utilizados na otimização via Busca em Grade

Parâmetro	Significado
n_estimators	Número de árvores a serem geradas
max_features	Número de variáveis a serem consideradas ao procurar a melhor divisão
max_depth	Comprimento máximo de cada árvore
min_samples_split	Número mínimo de amostras necessárias para dividir um nó
min_samples_leaf	Número mínimo de amostras necessárias para estar em um nó da folha
bootstrap	Método de amostragem dos dados

Fonte: Autor (2021)

Tabela 5 - Valores encontrados pelo método de otimização via Busca em Grade

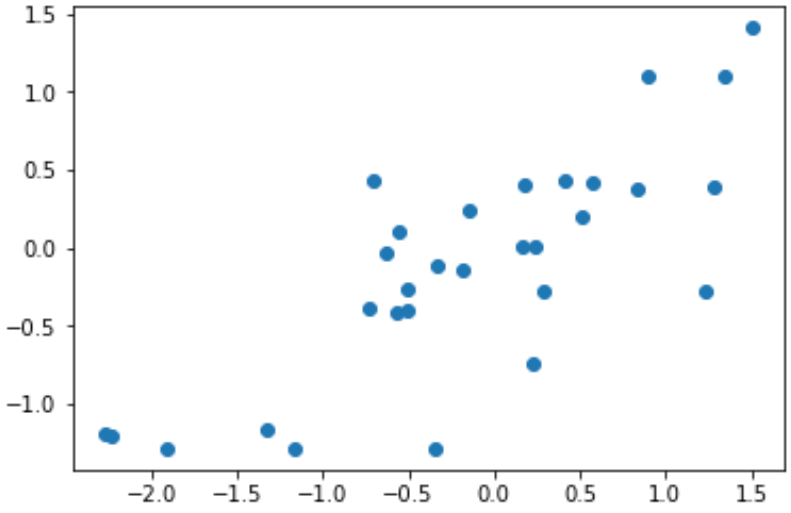
Parâmetro	Intervalo dos parâmetros	Melhores parâmetros
n_estimators	200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000	1600
max_features	auto, sqrt	auto
max_depth	10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None	None
min_samples_split	2, 5, 10	10
min_samples_leaf	1, 2, 4	2
bootstrap	True, False	True

Fonte: Autor (2021)

O modelo otimizado apresentou R^2 de 0,623, evidenciando um significativo aumento na representatividade do modelo, todavia, um bom R^2 deve ser acima de 0,70 para ser considerado estatisticamente satisfatório, o que não foi observado. A falta de predição dos modelos é explicada pela baixa quantidade de dados, que comprometeu a capacidade preditiva do analisador.

A Figura 58 mostra o resultado da otimização do modelo no digrama de dispersão.

Figura 58 - Resultado da otimização de modelo de Florestas Aleatórias



Fonte: Autor (2021)

6 CONSIDERAÇÕES FINAIS

A existência do grande volume de dados de processo na indústria moderna oferece oportunidades e desafios no que se refere à geração de informações necessárias à tomada de decisões estratégicas. Nesse sentido, o presente trabalho abordou técnicas de análise, exploração e visualização de dados no contexto do controle de processo.

Os resultados do desvio padrão amostral para a Gramatura se mostraram altos em relação às outras variáveis, entretanto a métrica de avaliação de homogeneidade, os coeficientes de variabilidade, apresentaram valores menores de que 10% em todas as variáveis de processo, se mostrando satisfatórios e homogêneos.

Os gráficos de linhas demonstraram comportamentos similares, exceto para as Gramaturas MQ2 e MQ3. Isto pode ser explicado devido a erros aleatórios do instrumento e laboratório.

A medida de dispersão e variabilidade, dada pelos intervalos de confiança, apresentaram valores satisfatórios, com baixa variabilidade das variáveis.

Em relação às cartas de controle, as variáveis Gramatura MQ1 e MQ3 (laboratório e instrumento) e Gramatura MQ2 (instrumento), Teor Seco MQ1, MQ2 e MQ3 (laboratório e instrumento) e Alvura (instrumento) evidenciam um processo, de forma geral, sob controle com alguns *outliers* devido à erros espúrios. As variáveis Gramatura MQ2 (laboratório), Kappa (laboratório e instrumento), Consistência (laboratório e instrumento) e Alvura (laboratório) demonstram instabilidade nas cartas de amplitudes móveis, ocasionando imprecisão dos limites de controle das cartas individuais e mais pontos discrepantes do que efetivamente ocorreu.

Analisando-se os *Boxplots* e *Violinplots,* conclui-se que as variáveis com maior e menor dispersão, respectivamente, são a Gramatura e a Consistência. A dispersão da Gramatura pode ser explicada pela alta grandeza, em torno de 1300 g/m², acarretando em variações de maior amplitude. Com relação aos valores discrepantes, todas as variáveis com exceção do Teor Seco, apresentaram *outliers*, possivelmente devido à erros espúrios do laboratório e instrumento.

O analisador virtual, construído através das técnicas Regressão Linear, Árvores de Decisão e Florestas Aleatórias, mostrou resultados preliminares de R^2 insatisfatórios. A otimização via Busca em Grade resultou em um R^2 de 0,623,

evidenciando um significativo aumento na representatividade do modelo de Florestas Aleatórias. Contudo, o valor ainda é considerado estatisticamente baixo para expressar uma predição representativa e confiável do analisador. A explicação para a falha na predição dos modelos reside na baixa quantidade de dados disponíveis das variáveis, impossibilitando o analisador de inferir o laboratório satisfatoriamente.

7 SUGESTÕES PARA TRABALHOS FUTUROS

Em decorrência da baixa quantidade de dados obtidos, recomenda-se para trabalhos futuros que as técnicas de *Machine Learning* para construção do analisador virtual sejam aplicadas à um conjunto de dados maior, com o objetivo de se obter um analisador de alta confiabilidade de predição.

REFERÊNCIAS

ABREU, C. *et al*. Indústria 4.0: Como as empresas estão utilizando a simulação para se preparar para o futuro. **Revista de Ciências Exatas e Tecnologia**, [*S. l.*], v. 12, n. 12, p. 49-53, 2017. Disponível em: https://revista.pgsskroton.com/index.php/rcext/article/view/5444. Acesso em: 10 nov. 2020.

ALBERTIN, M. *et al*. Principais inovações tecnológicas da indústria 4.0 e suas aplicações e implicações na manufatura*. In:* SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 24., 2017, Bauru. **Anais** [...]. Bauru: UNESP, 2017. Disponível em: https://www.researchgate.net/publication/321682376_PRINCIPAIS_INOVACOES_T. Acesso em: 10 nov. 2020.

BAUR, C.; WEE, D. **Manufacturing's next act.** McKinsey & Company, 2015. Disponível em: https://www.mckinsey.com/business-functions/operations/our-insights/manufacturings-next-act. Acesso em: 29 jun. 2020.

BREIMAN, L. Random forests. **Machine Learning**, [*S. l.*], v. 45, n. 1, p. 5-32, 2001. Disponível em: https://link.springer.com/article/10.1023/A:1010933404324#citeas. Acesso em: 12 out. 2020.

CAELUM. **Python e orientação a objetos**. São Paulo: Caelum, 2020. Disponível em: https://www.caelum.com.br/apostila-python-orientacao-a-objetos/. Acesso em 20 jun. 2020.

CAMPOS, B. C. *et al*. **Historiador de processos industriais**: análise e visualização de dados com o software elipse plant manager (epm). 2016. 93 f. Relatório (Bacharel em Engenharia Química), Universidade Federal de Minas Gerais, Belo Horizonte, 2016. Disponível em http://elipsecdn.blob.core.windows.net/public/REL_FINAL.pdf. Acesso em: 20 maio 2020.

CASTRO, H. F. **Processos químicos industriais II**. Papel e celulose. São Paulo: Universidade de São Paulo, 2009. Apostila 4. Disponível em http://sistemas.eel.usp.br/docentes/arquivos/5840556/434/apostila4papelecelulose.pdf. Acesso em: 13 jul. 2020.

CORREA, S. M. B. B. **Probabilidade e Estatística**. 2. ed. Belo Horizonte: PUC Minas Virtual, 2003. Disponível em: http://estpoli.pbworks.com/f/livro_probabilidade_estatistica_2a_ed. Acesso em: 22 out. 2020.

CORSETTI, A. R. **Desenvolvimento de um analisador virtual de teor de sólidos no processamento de proteína isolada de soja**. 2016. 42 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Química) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: https://lume.ufrgs.br/handle/10183/150747. Acesso em: 22 nov. 2020.

COSTA, A. B.; EPPRECHT, E. K.; CARPINETTI, L. R. **Controle estatístico da qualidade.** 2. ed. São Paulo: Atlas: 2004.

COSTA-FILHO, S. *et al*. Configuração de algoritmos de aprendizado de máquina na modelagem florestal: um estudo de caso na modelagem da relação hipsométrica. **Ciência Florestal**, Santa Maria , v. 29, n. 4, p. 1501-1515, Dec. 2019. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S198050982019000401501&lng=en&nrm=iso. Acesso em: 08 dez. 2020.

CRESPO, A. A. **Estatística fácil**. 17. ed. São Paulo: Saraiva 1999.

DALPIAZ, M; DEPINÉ, V. A.; GESSER, K. **Estatística.** Indaial: Grupo Uniasselvi, 2012. Disponível em: https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=6613. Acesso em: 19 out. 2020.

DEMINGOS, P. G. **ChemStruct:** um pacote em Python para análise estrutural de sistemas atômicos e auxílio em simulações de dinâmica molecular. 2019. 51 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Química) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019. Disponível em: https://lume.ufrgs.br/handle/10183/200343. Acesso em: 23 nov. 2020.

DEPARTAMENTO DE PESQUISAS E ESTUDOS ECONÔMICOS. Bradesco. Papel e Celulose**. Boletim do Panorama Setorial**, São Paulo, set. 2016.

ECONOMIST INTELLIGENCE UNIT. Big data. **Harnessing a game-changing asset**. Londres: The Economist, 2011. Disponível em: https://cdn.ymaws.com/edmcouncil.org/resource/resmgr/featured_documents/Press/EDMC_Big_Data_Harn_Dec11.pdf. Acesso em: 23 set. 2020.

FACCHIN, S. **Técnicas de análise multivariável aplicadas ao desenvolvimento de analisadores virtuais**. 2005. 140 f. Dissertação (Mestrado em Engenharia Química) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Engenharia Química, Porto Alegre, 2005. Disponível em: https://www.lume.ufrgs.br/handle/10183/8294. Acesso em: 22 out. 2020.

FACELI, K. *et al*. **Inteligência artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011. 378p.

FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. **Disponibiliza dados estatísticos referente ao cenário mundial de celulose e papel**, 2016. Disponível em: http://www.fao.org/faostat/en/. Acesso 20 jun. 2020.

FARIAS, N. **Desenvolvimento de analisador virtual para predição da pressão de fundo em poços de petróleo utilizando rede neural**. 2018. 32 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Química) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018. Disponível em: https://lume.ufrgs.br/handle/10183/193146. Acesso em: 30 mar. 2021.

FERREIRA, V. *et al*. Aplicação da carta de controle i-am para estudo de capabilidade em uma organização do setor de catering aéreo. *In:* ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. 28., Maceió, 2018. **Trabalho**. Maceió: ABEPRO, 2018. p. 1-21. Disponível em: http://www.abepro.org.br/biblioteca/TN_WIC_259_487_36148.pdf. Acesso em: 13 mar. 2021.

FOELKEL, C. E. B. Celulose kraft de Pinus spp. **Revista O Papel**, São Paulo, v. 38, n. 1, p. 49-67, 1976.

FORTUNA, L.; GRAZIANI, S.; XIBILIA, M. G. Soft sensors for product quality monitoring in debutanizer distillation columns. **Control Engineering Practice**, [*S. l*], v. 13, n. 4, p. 499 – 508, 2005. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0967066104000899?via%3Dihub. Acesso em 15 fev. 2021.

FUNDAÇÃO DOM CABRAL. O que seria a Indústria 4.0?. **Boletim: pesquisa sobre digitalização**. Nova Lima, fev. 2016. Disponível em: https://www.fdc.org.br/conhecimento-site/nucleos-de-pesquisa-site/centro-de-referencia-site/Materiais/O_que_seria_a_ind%C3%BAstria_4.0_-_Boletim_Fevereiro2016.pdf. Acesso em: 20 jun. 2020.

HINTZE, J. L.; NELSON, R. D. Violin plots: a box plot-density trace synergism. **The American Statistician**, [*S. l*], v. 52, n. 2, p. 181-184, 1998. Disponível em: http://www.stat.cmu.edu/~rnugent/PCMI2016/papers/ViolinPlots.pdf. Acesso em: 20 mar. 2021.

KADLEC, P.; GABRYS, B.; STRANDT, S. Data-driven soft sensors in the process industry. **Computers & Chemical Engineering**. v. 33, p. 795–814, 2009. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0098135409000076?via%3Dihub. Acesso em: 15 fev. 2021.

KLOCK, U.; ANDRADE, A.; HERNANDES, J. A. **Polpa e papel**. 3. ed. Curitiba: FUPEF, 2013. Série Didática. Disponível em: http://www.madeira.ufpr.br/disciplinasklock/polpaepapel/manualpolpa2013.pdf. Acesso em: 24 set. 2020.

LIN, T. I.; SOUZA, G.; YOUNG, B. Towards a viscosity and density correlation for dairy fluids - a soft sensor approach. **Computer Aided Chemical Engeneering**. [*S. l*.], v.27, 1371 - 1376 f. 2009. Disponível em: https://www.sciencedirect.com/science/article/pii/S1570794609706194. Acesso em: 10 mar. 2021.

LWARCEL CELULOSE E PAPEL LTDA. **Programa de redução de consumo de água da Lwarcel celulose**. Lençois Paulista: Lwarcel, 2007. Disponível em: http://az545403.vo.msecnd.net/uploads/2013/12/Lwarcel-Celulose-e-Papel.pdf. Acesso em: 20 nov. 2020.

MAYER, P. C. **Redução da Variabilidade em uma linha de produção de chapas de corpo de silos de grãos de corrugação 4'' através da implantação do controle estatístico do processo.** 2004. 98 f. Dissertação (Mestrado em Engenharia) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Engenharia, Porto Alegre, 2004. Disponível em: https://lume.ufrgs.br/handle/10183/6109. Acesso em: 17 fev. 2021.

MCAFEE, A. *et al*. Big data: the management revolution. **Harvard Business Review**, v. 90, n. 10, p. 60-68, 2012. Disponível em: https://hbr.org/2012/10/big-data-the-management-revolution. Acesso em: 10 out. 2020.

MCKINNEY, W. **Python para análise de dados:** tratamento de dados com Pandas, NumPy e IPython. São Paulo: Novatec, 2019.

MICHEL, R; FOGLIATTO, F.S. Projeto econômico de cartas adaptativas para monitoramento de processos. **Revista Gestão da Produção**. [*S. l*], v. 9, n.1, p 17-31, 2002. Disponível em: https://www.scielo.br/pdf/gp/v9n1/a03v9n1.pdf. Acesso em: 20 out. 2020.

MONARD, M.; BARANAUSKAS, J.. Indução de regras e árvores de decisão. *In:* REZENDE, S. O. (org.). **Sistemas inteligentes:** fundamentos e aplicações. Barueri: Manole, 2003. v. 1, p. 115-139. Disponível em: https://docplayer.com.br/78028436-Inducao-de-regras-e-arvores-de-decisao.html. Acesso em: 13 nov. 2020.

MONTGOMERY, D. **Introdução ao controle estatístico da qualidade**. 4. ed. Rio de Janeiro: LTC, 2009.

MORAIS JÚNIOR, A. A. **Elaboração de um analisador virtual utilizando sistema híbrido neuro-fuzzy para inferir a composição num processo de destilação.** 2011. 161 f. Dissertação (Mestrado em Engenharia Química) – Universidade Federal de Alagoas. Programa de Pós-Graduação em Engenharia Química, Maceió, 2011. Disponível em: http://www.repositorio.ufal.br/handle/riufal/1289. Acesso em: 15 mar. 2021.

MOURA. G. G.; LINO, H. S.; FERNANDES, S. M. Análise da metodologia de avaliação da capacidade dos processos de usinagem para implementação do CEP em ferramentas elétricas. *In*: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO. 15., Bauru, 2008. **Anais [...]**. Bauru: Revista Espacios, 2008.

OLIVEIRA, B. Boxplot: como interpretar?. **Operdata**, 2019. Disponível em: https://operdata.com.br/blog/como-interpretar-um-boxplot/. Acesso em: 10 fev. 2021.

OLIVEIRA, C. C. *et al*. **Manual para elaboração de cartas de controle para monitoramento de processos de medição quantitativos em laboratórios de ensaio.** São Paulo: Instituto Adolfo Lutz, 2013.

PAESE, C; CATEN, C. T; RIBEIRO, J. L. D. Aplicação da análise de variância na implantação do CEP. **Revista Produção**, v. 11, n. 1, p. 17- 26, 2001. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S010365132001000100002&lng=en&nrm=iso. Acesso em: 28 set. 2020.

PIMENTEL, F.; GARCIA, C. H. **Estatística aplicada à experimentação agronômica e florestais exposições com exemplos e orientações para uso de aplicativos**. Piracicaba: FEALQ, 2002.

REVISTA O PAPEL. **Automação e manutenção entre as vantagens competitivas dos players de celulose e papel.** São Paulo, jul. 2014. Disponível em: http://www.revistaopapel.org.br/noticia-anexos/1405689258_33297721c9e06c4747c6669b00d5880a_1037596495.pdf. Acesso em: 15 nov. 2020.

RIBEIRO, J. L.; CATEN, C. T. **Controle estatístico do processo**. Porto Alegre: FEENG/UFRGS, 2012. Disponível em: http://www.producao.ufrgs.br/arquivos/disciplinas/388_apostilacep_2012.pdf. Acesso em: 30 set. 2020.

RODRIGUES, B. H. S. **Métodos de construção de analisadores virtuais para estimação de teor de enxofre de hidrocarbonetos**. 2014. 60 f. Monografia (Especialização em Automação Industrial) – Universidade Federal de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica, Belo Horizonte, 2014. Disponível em: https://repositorio.ufmg.br/handle/1843/BUBD-9VMFLB. Acesso em: 15 mar. 2021.

RODRIGUES, L. Aprenda como aplicar o Controle Estatístico de Processo para a detecção de problemas. **Voitto**, 2019. Disponível em: https://www.voitto.com.br/blog/artigo/controle-estatistico-de-processo. Acesso em: 20 jun. 2020.

RODRIGUES, L. **Regressão linear simples**. São Paulo: Universidade de São Paulo, 2015. Apostila. Disponível em: https://www.ime.usp.br/~fmachado/MAE229/AULA10.pdf. Acesso em 25 nov. de 2020.

SALDANHA, P. *et al*. Analisando a aplicação do controle estatístico de processos na indústria química: um estudo de caso. **Revista Espacios**, [*S. l.*], v. 34, n. 11, p. 1-17, 2013. Disponível em: https://www.researchgate.net/publication/299505665_Analisando_a_aplicacao_do_controle_estatistico_de_processos_na_industria_quimica_um_estudo_de_caso. Acesso em: 23 set. 2020.

SEABORN. Statistical data visualization. **Seaborn Pydata**, 2020. Disponível em: https://seaborn.pydata.org/examples/horizontal_boxplot.html Acesso em: 20 nov. 2020.

SERVIÇO NACIONAL DE APRENDIZAGEM NACIONAL DO PARANÁ. **Tendências de inovação no mercado de papel e celulose**. Curitiba: SENAI, 2020. Disponível em: https://www.senaipr.org.br/tecnologiaeinovacao/blog/FreeComponent36128content450500.shtml. Acesso em: 10 fev. 2021.

SMOLAN, R; ERWITT, J. **The human face of Big Data.** Sausalito: Against All Odds Productions, 2012.

SOUZA, A. M. **Monitoração e ajuste de realimentação em processos produtivos multivariados.** Tese (Doutorado Engenharia de Produção) – Universidade Federal Santa Catarina. Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, 2000. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/79195/176087.pdf?sequence=1&isAllowed=y. Acesso em: 20 set. 2020.

SUPORTE MINITAB. **Interpretar os principais resultados para um Carta I-AM**, 2018. Disponível em: https://support.minitab.com/pt-br/minitab/18/help-and-how-to/quality-and-process-improvement/control-charts/how-to/variables-charts-for individuals/i-mr-chart/interpret-the-results/key-results/. Acesso em: 30 mar. 2021.

TAMÁS, P.; ILLÉS, B. Process improvement trends for manufacturing systems in Industry 4.0. **Academic Journal of Manufacturing Engineering**, v.14, n. 4, p. 1- 7, 2016. Disponível em: https://www.researchgate.net/publication/313359291_PROCESS_IMPROVEMENT_TRENDS_FOR_MANUFACTURING_SYSTEMS_IN_INDUSTRY_40. Acesso em: 23 set. 2020.

TAN, K. H. *et al*. Harvesting big data to enhance supply chain innovation capabilities: An analytic infrastructure based on deduction graph. **International Journal of Production Economics**, v. 165, p. 223-233, 2015. Disponível em: https://www.researchgate.net/publication/270595015_Harvesting_Big_Data_to_Enhance_Supply_Chain_Innovation_Capabilities_An_Analytic_Infrastructure_Based_on_Deduction_Graph. Acesso em: 23 set. 2020.

THAM, M. T. *et al.* Soft-sensors for process estimation and inferential control. **Journal of Process Control.** *[S. l.],* v. 1, n. 1, p. 3- 14, Jan. 1991. Disponível em: https://www.sciencedirect.com/science/article/pii/095915249187002F. Acesso em: 25 out. 2020.

VALLADARES NETO, J. *et al*. Boxplot: um recurso gráfico para a análise e interpretação de dados quantitativos. **Revista Odontológica do Brasil Central**, [*S. l.*], v. 26, n. 76, 2017. Disponível em: https://www.robrac.org.br/seer/index.php/ROBRAC/article/view/1132/897. Acesso em: 20 mar. 2021.

VERHAPPEN, I. Turning big data into information benefits the bottom line. **Offshore**, Dec. 12th, 2013. Disponível em: https://www.offshore-mag.com/business-briefs/equipment-engineering/article/16761293/turning-big-data-into-information-benefits-the-bottom-line. Acesso em: 20 jun. 2020.

VIDAL, A. C. F; HORA, A. B. **A indústria de papel e celulose** (org.). *In:* BNDES 60 anos: perspectivas setoriais. Rio de Janeiro: Banco de Desenvolvimento Econômico e Social, p.334-381. 2012. Disponível em: https://web.bndes.gov.br/bib/jspui/bitstream/1408/935/1/A%20ind%c3%bastria%20de%20papel%20e%20celulose_P-final.pdf. Acesso em: 25 out. 2020.

ANEXO

Anexo A - valores das constantes para construção dos limites de controle

Número de amostras (n)	A2	A3	B3	B4	D2	D3	D4
2	1,88	2,66	0	3,27	1,13	0	3,27
3	1,02	1,95	0	2,57	1,69	0	2,57
4	0,73	1,63	0	2,27	2,06	0	2,28
5	0,58	1,43	0	2,09	2,33	0	2,11
6	0,48	1,29	0,03	1,97	2,53	0	2,00
7	0,42	1,18	0,12	1,88	2,70	0,08	1,92
8	0,37	1,10	0,19	1,82	2,85	0,14	1,86
9	0,34	1,03	0,24	1,76	2,97	0,18	1,82
10	0,31	0,98	0,28	1,72	3,08	0,22	1,78
11	0,29	0,93	0,32	1,68	3,17	0,26	1,74
12	0,27	0,89	0,35	1,65	3,26	0,28	1,72
13	0,25	0,85	0,38	1,62	3,34	0,31	1,69
14	0,24	0,82	0,41	1,59	3,41	0,33	1,67
15	0,22	0,79	0,43	1,57	3,47	0,35	1,65
16	0,21	0,76	0,45	1,55	3,53	0,36	1,64
17	0,20	0,74	0,47	1,53	3,59	0,38	1,62
18	0,19	0,72	0,48	1,52	3,64	0,39	1,61
19	0,19	0,70	0,50	1,50	3,69	0,40	1,60
20	0,18	0,68	0,51	1,49	3,74	0,42	1,59

Fonte: Montgomery (2009, p. 83)

APÊNDICE

Apêndice A - Implementação em Python

Loading excel

```
"""
import basic libraries
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
```

```
df = pd.read_excel('Larissa.xlsx', header=1)
```

```
df.head()
```

```
df.keys()
```

```
df.drop(['Unnamed: 3','Unnamed: 7'], axis=1, inplace=True)
```

```
df.head()
```

```
df.columns = ['ts_kappa','kappa_lab','kappa_inst',
              'ts_consistencia','consistencia_lab','consistencia_inst',
              'ts_alvura','alvura_lab','alvura_inst']
```

```
df.head()
```

```
df.drop([0], inplace=True)
```

```
df
```

```
df.describe()
```

```
# dividir em dataframes diferentes
```

```
#import datetime as dt
data = 1
name_var = ['kappa','consist','alvura']
name_col = ['time-stamp', 'lab', 'inst']
n_var = len(name_var)
# size = '87'+5*'\d'+'$'

for i in range(n_var):
    # separar dataframes
    data[name_var[n]] = df.iloc[:,3*n:3*n+3]

    # trocar nome das colunas
    data[name_var[n]].columns = name_col

    # remover nans
    data[name_var[n]].dropna(inplace=True)

    # transformar em time-stamp
    data[name_var[n]].set_index('time-stamp', inplace=True)
```

```
data['alvura']
```

```
plt.figure(figsize=(12,6))
plt.plot(data['kappa']['lab']),plt.plot(data['kappa']['inst'],'m-')
plt.legend(['laboratório','instrumento'])
```

```
sns.distplot((data['kappa']['lab']-data['kappa']['inst']),bins=50);
```

```
sns.distplot((data['consist']['lab']-data['consist']['inst']),bins=50);
```

```
sns.distplot((data['alvura']['lab']-data['alvura']['inst']),bins=50);
```

```
plt.figure(figsize=(12,6))
plt.plot(data['consist']['lab']),plt.plot(data['consist']['inst'])
```

```
plt.figure(figsize=(12,6))
plt.plot(data['alvura']['lab']),plt.plot(data['alvura']['inst'])
```

```
data['kappa'].describe()
```

```
data['alvura'].describe()
```

```
data['consist'].describe()
```

```
sns.violinplot(y=data['alvura']['lab']-data['alvura']['inst'])
sns.violinplot(y=data['kappa']['lab']-data['kappa']['inst'])
```

```
sns.boxplot(y=data['alvura']['lab']-data['alvura']['inst'])
```

Cálculo intervalo de confiança

```
n_alvura = data['consist']['inst']
```

```
from scipy import stats
mean, sigma = np.mean(n_alvura), np.std(n_alvura)
#conf_int = stats.norm.interval(0.95, loc=mean, scale=sigma)
conf_int_b = stats.norm.interval(0.95, loc=mean, scale=sigma / np.sqrt(len(n_alvura)))
conf_int_b
```

```
print(mean - 1.96*sigma/np.sqrt(len(n_alvura)))
print(mean + 1.96*sigma/np.sqrt(len(n_alvura)))
```

```
print(mean)
print(1.96*sigma/np.sqrt(len(n_alvura)))
```

Control charts

```
x = pd.Series(data['kappa']['inst'].values)
x
```

```
# Define list variable for moving ranges
MR = [np.nan]

# Get and append moving ranges
i = 1
for ii in range(1, len(x)):
    MR.append(abs(x[i] - x[i-1]))
    i += 1

# Convert list to pandas Series objects
MR = pd.Series(MR)
MR
```

```
# Concatenate MR Series with and rename columns
dat = pd.concat([x,MR], axis=1).rename(columns={0:'x', 1:'mR'})
```

```
# Plot x and mR charts
fig, axs = plt.subplots(2, figsize=(15,15), sharex=True)

# x chart
axs[0].plot(dat['x'], linestyle='-', marker='o', color='black')
axs[0].axhline(statistics.mean(dat['x']), color='blue')
axs[0].axhline(statistics.mean(dat['x'])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128, color = 'red', linestyle ='dashed')
axs[0].axhline(statistics.mean(dat['x'])-3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128, color = 'red', linestyle ='dashed')
axs[0].set_title('Carta Individual')
axs[0].set(xlabel='Observação', ylabel='Valores individuais')

# mR chart
axs[1].plot(dat['mR'], linestyle='-', marker='o', color='black')
axs[1].axhline(statistics.mean(dat['mR'][1:len(dat['mR'])]), color='blue')
#axs[1].axhline(statistics.mean(dat['mR'][1:len(dat['mR'])])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])*0.853/1.128, color = 'red', linestyle ='dashed')
axs[1].axhline(3.267*statistics.mean(dat['mR'][1:len(dat['mR'])]), color='red', linestyle ='dashed')
#axs[1].axhline(0*statistics.mean(dat['mR'][1:len(dat['mR'])]), color='red', linestyle ='dashed')
axs[1].set_title('mR')
axs[1].set(xlabel='Observação', ylabel='Amplitude Móvel')
```

```
# Validate points out of control limits for x chart
i = 0
control = True
for unit in dat['x']:
    if unit > statistics.mean(dat['x'])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128 or unit < statistics.mean(dat['x'])-3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128:
        print('Unit', i, 'out of control limits!')
        control = False
    i += 1
if control == True:
    print('All points within control limits.')
```

```
# Validate points out of control limits for mR chart
i = 0
control = True
for unit in dat['mR']:
    if unit > statistics.mean(dat['mR'][1:len(dat['mR'])])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])*0.853/1.128:
        print('Unit', i, 'out of control limits!')
        control = False
    i += 1
if control == True:
    print('All points within control limits.')
```

machine learning

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import scale
```

```
var = 'kappa' #'alvura', kappa', 'consist' #0.13, 0.62, 0.48
X = data[var].drop('lab',axis=1)
y = data[var]['lab']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
```

```
# print the intercept
print(lm.coef_)
```

```
lm.score(X_test, y_test)
```

```
predictions = lm.predict(X_test)
```

```
plt.scatter(y_test,predictions)
```

```
sns.distplot((y_test-predictions),bins=50);
```

```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
lm.get_params(deep=True)
```

```
from sklearn.tree import DecisionTreeRegressor
dtree = DecisionTreeRegressor() #RandomForestRegressor(n_estimators=800)
dtree.fit(X_train,y_train)
```

```
dtree.fit(X_train,y_train)
dtree.score(X_test,y_test)
```

```
predictions = dtree.predict(X_test)
plt.scatter(y_test,predictions)
```

```
sns.distplot((y_test-predictions),bins=50);
```

```
X_test
```

```
from sklearn import tree
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (10,10), dpi=800)
tree.plot_tree(dtree, filled=True);
fig.savefig('rf_individualtree.png')
```

```

```

```
from sklearn.ensemble import RandomForestRegressor
```

```
rforest = RandomForestRegressor(n_estimators=800) #DecisionTreeRegressor()
```

```
rforest.fit(X_train,y_train)
```

```
plt.figure(figsize=(24,12))
tree.plot_tree(rforest.estimators_[0], feature_names = 'klab', filled = True);
```

```
rforest.score(X_test,y_test)
```

```
predictions = rforest.predict(X_test)
```

```
plt.scatter(y_test,predictions)
```

neural net

```
from sklearn.neural_network import MLPRegressor
regr = MLPRegressor(random_state=1, max_iter=500).fit(X_train, y_train)
```

```
prednn = regr.predict(X_test)
```

```
regr.score(X_test, y_test)
```

```
plt.scatter(y_test,prednn)
```

neural net gs

```
mlp = MLPRegressor(max_iter=100)
parameter_space = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}
```

```
from sklearn.model_selection import GridSearchCV

clf = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=3)
clf.fit(X_train, y_train)
```

```
print(clf.best_params_)
print(clf.best_score_)
# print("Train RSE:", np.round(train_mse,2))
# print("Test RSE:", np.round(test_mse,2))
```

```
predsnngs = clf.predict(X_test)
```

```
clf.score(X_test, y_test)
```

```
plt.scatter(y_test,predsnngs)
```

random forest gs

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state = 42)
from pprint import pprint
# Look at parameters used by our current forest
print('Parameters currently in use:\n')
pprint(rf.get_params())
```

```
from sklearn.model_selection import RandomizedSearchCV
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
print(random_grid)
```

```
# Use the random grid to search for best hyperparameters
# First create the base model to tune
rf = RandomForestRegressor()
# Random search of parameters, using 3 fold cross validation,
# search across 100 different combinations, and use all available cores
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)
# Fit the random search model
rf_random.fit(X_train,y_train)
```

```
rf_random.best_params_
```

```
def evaluate(model, test_features, test_labels):
    predictions = model.predict(test_features)
    errors = abs(predictions - test_labels)
    mape = 100 * np.mean(errors / test_labels)
    accuracy = 100 - mape
    print('Model Performance')
    print('Average Error: {:0.4f} degrees.'.format(np.mean(errors)))
    print('Accuracy = {:0.2f}%.'.format(accuracy))

    return accuracy
base_model = RandomForestRegressor(n_estimators = 10, random_state = 42)
base_model.fit(X_train,y_train)
base_accuracy = evaluate(base_model, X_train,y_train)
```

```
best_random = rf_random.best_estimator_
random_accuracy = evaluate(best_random, X_train,y_train)
```

```
predsrfb = best_random.predict(X_test)
```

```
best_random.score(X_test, y_test)
```

```
plt.scatter(y_test,predsrfb)
```

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df = pd.read_excel('Dados Larissa.xlsx', header=1)
```

In [ ]:
```
df.head()
```

In [ ]:
```
df.keys()
```

In [ ]:
```
df.drop(['Unnamed: 3' , 'Unnamed: 7' , 'Unnamed: 15' , 'Unnamed: 11', 'Unnamed: 19' ], axis=1, inplace=True)
```

In [ ]:
```
df.head()
```

In [ ]:
```
df.columns = ['ts_teorsecoMQ1','teorseco_labMQ1','teorseco_instMQ1','ts_gramaturaMQ1','gramatura_labMQ1','grama
```

In [ ]:
```
df.head()
```

In [ ]:
```
df.drop([0], inplace=True)
```

In [ ]:
```
df
```

In [ ]:
```
df.drop([1], inplace=True)
```

In [ ]:
```
df
```

In [ ]:
```
df.describe()
```

In [ ]:
```
df.dropna(axis=0, inplace=True)
```

In [ ]:
```
df
```

In [ ]:
```
df.describe()
```

In [ ]:
```
data = {}
name_var = ['teorsecoMQ1','gramaturaMQ1','teorsecoMQ2','teorsecoMQ3','gramaturaMQ2','gramaturaMQ3']
name_col = ['time-stamp','lab','inst']
n_var = len(name_var)
for n in range(n_var):
    data[name_var[n]] = df.iloc[:,3*n:3*n+3]
    data[name_var[n]].columns = name_col
    data[name_var[n]].dropna(inplace=True)
    data[name_var[n]].set_index('time-stamp',inplace=True)
```

In [ ]:
```
data['teorsecoMQ2']
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['gramaturaMQ1']['lab']),plt.plot(data['gramaturaMQ1']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Gramatura MQ1',size=15)
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['gramaturaMQ2']['lab']),plt.plot(data['gramaturaMQ2']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Gramatura MQ2',size=15)
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['gramaturaMQ3']['lab']),plt.plot(data['gramaturaMQ3']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Gramatura MQ3',size=15)
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['teorsecoMQ1']['lab']),plt.plot(data['teorsecoMQ1']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Teor Seco MQ1',size=15)
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['teorsecoMQ2']['lab']),plt.plot(data['teorsecoMQ2']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Teor Seco MQ2',size=15)
```

In [ ]:
```
plt.figure(figsize=(12,6))
plt.plot(data['teorsecoMQ3']['lab']),plt.plot(data['teorsecoMQ3']['inst'])
plt.legend(['Laboratório','Instrumento'])
plt.xlabel('Time-stamp',size=15)
plt.ylabel('Teor Seco MQ3',size=15)
```

In [ ]:
```
data['teorsecoMQ3'].describe()
```

In [ ]:
```
data['teorsecoMQ2'].describe()
```

In [ ]:
```
data['teorsecoMQ3'].describe()
```

In [ ]:
```
data['gramaturaMQ1'].describe()
```

In [ ]:
```
data['gramaturaMQ2'].describe()
```

In [ ]:
```
data['gramaturaMQ3'].describe()
```

In [ ]:
```
data.keys()
```

In [ ]:
```
sns.distplot((data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']),bins=50);
```

In [ ]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 14,))
sns.distplot((data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']),ax=axes[0][0],bins=50);
axes[0][0].set_title('Teor Seco MQ1',size=20)
axes[0][0].set_xlabel('Faixa de Erro',size=15)
axes[0][0].set_ylabel('Densidade',size=15)
axes[0][0].set_xlim([-4,4])
sns.distplot((data['teorsecoMQ2']['lab']-data['teorsecoMQ2']['inst']),ax=axes[0][1],bins=50);
axes[0][1].set_title('Teor Seco MQ2',size=20)
axes[0][1].set_xlabel('Faixa de Erro',size=15)
axes[0][1].set_ylabel('Densidade',size=15)
axes[0][1].set_xlim([-4,4])
sns.distplot((data['teorsecoMQ3']['lab']-data['teorsecoMQ3']['inst']),ax=axes[0][2],bins=50);
axes[0][2].set_title('Teor Seco MQ3',size=20)
axes[0][2].set_xlabel('Faixa de Erro',size=15)
axes[0][2].set_ylabel('Densidade',size=15)
axes[0][2].set_xlim([-4,4])
sns.distplot((data['gramaturaMQ1']['lab']-data['gramaturaMQ1']['inst']),ax=axes[1][0],bins=50);
axes[1][0].set_title('Gramatura MQ1',size=20)
axes[1][0].set_xlabel('Faixa de Erro',size=15)
axes[1][0].set_ylabel('Densidade',size=15)
axes[1][0].set_xlim([-100,100])
sns.distplot((data['gramaturaMQ2']['lab']-data['gramaturaMQ2']['inst']),ax=axes[1][1],bins=50);
axes[1][1].set_title('Gramatura MQ2',size=20)
axes[1][1].set_xlabel('Faixa de Erro',size=15)
axes[1][1].set_ylabel('Densidade',size=15)
axes[1][1].set_xlim([-100,100])
sns.distplot((data['gramaturaMQ3']['lab']-data['gramaturaMQ3']['inst']),ax=axes[1][2],bins=50);
axes[1][2].set_title('Gramatura MQ3',size=20)
axes[1][2].set_xlabel('Faixa de Erro',size=15)
axes[1][2].set_ylabel('Densidade',size=15)
axes[1][2].set_xlim([-100,100])
```

In [ ]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 14))
sns.violinplot(y=(data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']), ax=axes[0][0]);
axes[0][0].set_title('Teor Seco MQ1',size=20)
axes[0][0].set_xlabel('Dados',size=15)
axes[0][0].set_ylabel('Distribuição',size=15)
axes[0][0].set_ylim([-4,4])
sns.violinplot(y=(data['teorsecoMQ2']['lab']-data['teorsecoMQ2']['inst']), ax=axes[0][1]);
axes[0][1].set_title('Teor Seco MQ2',size=20)
axes[0][1].set_xlabel('Dados',size=15)
axes[0][1].set_ylabel('Distribuição',size=15)
axes[0][1].set_ylim([-4,4])
sns.violinplot(y=(data['teorsecoMQ3']['lab']-data['teorsecoMQ3']['inst']), ax=axes[0][2]);
axes[0][2].set_title('Teor Seco MQ3',size=20)
axes[0][2].set_xlabel('Dados',size=15)
axes[0][2].set_ylabel('Distribuição',size=15)
axes[0][2].set_ylim([-4,4])
sns.violinplot(y=(data['gramaturaMQ1']['lab']-data['gramaturaMQ1']['inst']),ax=axes[1][0]);
```

```
axes[1][0].set_title('Gramatura MQ1',size=20)
axes[1][0].set_xlabel('Dados',size=15)
axes[1][0].set_ylabel('Distribuição',size=15)
axes[1][0].set_ylim([-100,110])
sns.violinplot(y=(data['gramaturaMQ2']['lab']-data['gramaturaMQ2']['inst']),ax=axes[1][1]);
axes[1][1].set_title('Gramatura MQ2',size=20)
axes[1][1].set_xlabel('Dados',size=15)
axes[1][1].set_ylabel('Distribuição',size=15)
axes[1][1].set_ylim([-100,110])
sns.violinplot(y=(data['gramaturaMQ3']['lab']-data['gramaturaMQ3']['inst']),ax=axes[1][2]);
axes[1][2].set_title('Gramatura MQ3',size=20)
axes[1][2].set_xlabel('Dados',size=15)
axes[1][2].set_ylabel('Distribuição',size=15)
axes[1][2].set_ylim([-100,110])
```

In [ ]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 14))
sns.boxplot(y=(data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']), ax=axes[0][0]);
axes[0][0].set_title('Teor Seco MQ1',size=20)
axes[0][0].set_xlabel('Dados',size=15)
axes[0][0].set_ylabel('Distribuição',size=15)
axes[0][0].set_ylim([-3,3])
sns.boxplot(y=(data['teorsecoMQ2']['lab']-data['teorsecoMQ2']['inst']), ax=axes[0][1]);
axes[0][1].set_title('Teor Seco MQ2',size=20)
axes[0][1].set_xlabel('Dados',size=15)
axes[0][1].set_ylabel('Distribuição',size=15)
axes[0][1].set_ylim([-3,3])
sns.boxplot(y=(data['teorsecoMQ3']['lab']-data['teorsecoMQ3']['inst']), ax=axes[0][2]);
axes[0][2].set_title('Teor Seco MQ3',size=20)
axes[0][2].set_xlabel('Dados',size=15)
axes[0][2].set_ylabel('Distribuição',size=15)
axes[0][2].set_ylim([-3,3])
sns.boxplot(y=(data['gramaturaMQ1']['lab']-data['gramaturaMQ1']['inst']),ax=axes[1][0]);
axes[1][0].set_title('Gramatura MQ1',size=20)
axes[1][0].set_xlabel('Dados',size=15)
axes[1][0].set_ylabel('Distribuição',size=15)
axes[1][0].set_ylim([-90,110])
sns.boxplot(y=(data['gramaturaMQ2']['lab']-data['gramaturaMQ2']['inst']),ax=axes[1][1]);
axes[1][1].set_title('Gramatura MQ2',size=20)
axes[1][1].set_xlabel('Dados',size=15)
axes[1][1].set_ylabel('Distribuição',size=15)
axes[1][1].set_ylim([-90,110])
sns.boxplot(y=(data['gramaturaMQ3']['lab']-data['gramaturaMQ3']['inst']),ax=axes[1][2]);
axes[1][2].set_title('Gramatura MQ3',size=20)
axes[1][2].set_xlabel('Dados',size=15)
axes[1][2].set_ylabel('Distribuição',size=15)
axes[1][2].set_ylim([-90,110])
```

In [ ]:
```
df.info()
```

In [ ]:
```
gramaturamq1=data['gramaturaMQ1']['lab']
mean, sigma= np.mean(gramaturamq1), np.std(gramaturamq1)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(gramaturamq1)))
print(mean-1.96*sigma/np.sqrt(len(gramaturamq1)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(gramaturamq1)))
```

In [ ]:
```
gramaturamq2=data['gramaturaMQ2']['lab']
mean, sigma= np.mean(gramaturamq2), np.std(gramaturamq2)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(gramaturamq2)))
print(mean-1.96*sigma/np.sqrt(len(gramaturamq2)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(gramaturamq2)))
```

In [ ]:
```
gramaturamq3=data['gramaturaMQ3']['lab']
mean, sigma= np.mean(gramaturamq3), np.std(gramaturamq3)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(gramaturamq3)))
print(mean-1.96*sigma/np.sqrt(len(gramaturamq3)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(gramaturamq3)))
```

In [ ]:
```
teorsecomq1=data['teorsecoMQ1']['lab']
mean, sigma= np.mean(teorsecomq1), np.std(teorsecomq1)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(teorsecomq1)))
print(mean-1.96*sigma/np.sqrt(len(teorsecomq1)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(teorsecomq1)))
```

In [ ]:
```
teorsecomq2=data['teorsecoMQ2']['lab']
mean, sigma= np.mean(teorsecomq2), np.std(teorsecomq2)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(teorsecomq2)))
print(mean-1.96*sigma/np.sqrt(len(teorsecomq2)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(teorsecomq2)))
```

In [ ]:
```
teorsecomq3=data['teorsecoMQ3']['lab']
mean, sigma= np.mean(teorsecomq3), np.std(teorsecomq3)
```

In [ ]:
```
print(mean+1.96*sigma/np.sqrt(len(teorsecomq3)))
print(mean-1.96*sigma/np.sqrt(len(teorsecomq3)))
```

In [ ]:
```
print(mean)
print(1.96*sigma/np.sqrt(len(teorsecomq3)))
```

In [ ]:
```
def plot_comparison(x, title):
    fig, ax = plt.subplots(3, 1,figsize=(10, 12))
    sns.distplot((data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']),ax=ax[0],bins=200);
    ax[0].set_title('Histogram + KDE')
    sns.boxplot(y=(data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']), ax=ax[1]);
    ax[1].set_title('Boxplot')
    sns.violinplot(y=(data['teorsecoMQ1']['lab']-data['teorsecoMQ1']['inst']), ax=ax[2]);
    ax[2].set_title('Violin plot')
    fig.suptitle(title, fontsize=16)
    plt.show()
```

In [ ]:
```
sample_gaussian = np.random.normal
plot_comparison(sample_gaussian, 'Standard Normal Distribution')
```

In [ ]:
```
import statistics
x = pd.Series(data['teorsecoMQ3']['inst'].values)
MR=[np.nan]
i = 1
for d in range(1, len(x)):
    MR.append(abs(x[i] - x[i-1]))
    i += 1
MR = pd.Series(MR)
d = pd.concat([x,MR], axis=1).rename(columns={0:"x", 1:"mR"})
```

In [ ]:
```
# Plot x and mR charts
# Concatenate mR Series with and rename columns
dat = pd.concat([x,MR], axis=1).rename(columns={0:"x", 1:"mR"})
fig, axs = plt.subplots(2, figsize=(15,15), sharex=True)

# x chart
axs[0].plot(dat['x'], linestyle='-', marker='o', color='black')
axs[0].axhline(statistics.mean(dat['x']), color='blue')
axs[0].axhline(statistics.mean(dat['x'])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128, color = 'red', l
axs[0].axhline(statistics.mean(dat['x'])-3*statistics.mean(dat['mR'][1:len(dat['mR'])])/1.128, color = 'red', l
axs[0].set_title('Carta Individual')
axs[0].set(xlabel='Observação', ylabel='Valores individuais')


# mR chart
axs[1].plot(dat['mR'], linestyle='-', marker='o', color='black')
axs[1].axhline(statistics.mean(dat['mR'][1:len(dat['mR'])]), color='blue')
#axs[1].axhline(statistics.mean(dat['mR'][1:len(dat['mR'])])+3*statistics.mean(dat['mR'][1:len(dat['mR'])])*0.8
axs[1].axhline(3.267*statistics.mean(dat['mR'][1:len(dat['mR'])]), color='red', linestyle ='dashed')
axs[1].axhline(0*statistics.mean(dat['mR'][1:len(dat['mR'])]), color='red', linestyle ='dashed')
#axs[1].set_ylim(bottom=-2)
axs[1].set_title('mR')
axs[1].set(xlabel='Observação', ylabel='Amplitude Móvel')

axs[0].set_ylim([86,93])

axs[1].set_ylim([-0.7,5.8])
i = 0
control = True
for unit in d['x']:
    if unit > statistics.mean(d['x'])+3*statistics.mean(d['mR'][1:len(d['mR'])])/1.128 or unit < statistics.mea
        print('Unit', i, 'out of cotrol limits!')
        control = False
```

```
    i += 1
if control == True:
    print('All points within control limits.')
i = 0
control = True
for unit in d['mR']:
    if unit > statistics.mean(d['mR'][1:len(d['mR'])])+3*statistics.mean(d['mR'][1:len(d['mR'])])*0.8525 or uni
        print('Unit', i, 'out of control limits!')
```

In [ ]:
```
from sklearn.preprocessing import scale
X=scale(data['teorsecoMQ1'][['inst']])
y=scale(data['teorsecoMQ1']['lab'])
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [ ]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
```

In [ ]:
```
print(lm.intercept_)
```

In [ ]:
```
lm.score(X_test, y_test)
```

In [ ]:
```
lm.coef_
```

In [ ]:
```
predictions = lm.predict(X_test)
```

In [ ]:
```
plt.scatter(y_test,predictions)
```

In [ ]:
```
sns.distplot((y_test-predictions),bins=50);
```

In [ ]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

In [ ]:
```
#Arvore de decisão
dtree = DecisionTreeRegressor()
dtree.fit(X_train,y_train)
dtree.score(X_test,y_test)
```

In [ ]:
```
predictions = dtree.predict(X_test)
plt.scatter(y_test,predictions)
```

In [ ]:
```
sns.distplot((y_test-predictions),bins=50);

from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

In [ ]:
```
#Florestas aleatorias
Rforest = RandomForestRegressor()
Rforest.fit(X_train,y_train)
Rforest.score(X_test,y_test)
```

In [ ]:
```
predictions = Rforest.predict(X_test)
plt.scatter(y_test,predictions)
```

In [ ]:
```
sns.distplot((y_test-predictions),bins=50);

from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

In [ ]:
```
from sklearn.neural_network import MLPRegressor
regr = MLPRegressor(random_state=1, max_iter=500).fit(X_train, y_train)
prednn = regr.predict(X_test)

regr.score(X_test, y_test)
```

In [ ]:
```
plt.scatter(y_test,prednn)
```

In [ ]:

In [ ]:

In [ ]: