PHASE 5:

PROJECT TITLE:CREDIT CARD FRAUD DETECTION.

DATA SCIENCE:

1. Data science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data.
2. Data Science is about data gathering, analysis and decision-making.
3. Data Science is about finding patterns in data, through analysis, and make future predictions.

By using Data Science, companies are able to make:

1. Better decisions (should we choose A or B)
2. Predictive analysis (what will happen next?)
3. Pattern discoveries (find pattern, or maybe hidden information in the data)

A Data Scientist requires expertise in several backgrounds:

1. Machine Learning
2. Statistics
3. Programming (Python or R)
4. Mathematics
5. Databases

A Data Scientist must find patterns within the data. Before he/she can find the patterns, he/she must organize the data in a standard format.

By 2020, there will be around 40 zettabytes of data—that's 40 trillion gigabytes. The amount of data that exists grows exponentially. At any time, about 90 percent of this huge amount of data gets generated in the most recent two years, according to sources like IBM and SINTEF.

In fact, internet users generate about 2.5 quintillion bytes of data every day. By 2020, every person on Earth will be generating about 146,880 GB of data every day, and by 2025, that will be 165 zettabytes every year.

Simple data analysis can interpret data from a single source, or a limited amount of data. However, data science tools are critical to understanding big data and data from multiple sources in a meaningful way. A look at some of the specific data science applications in business illustrate this point and provide a compelling introduction to data science.

CREDIT CARD FRAUD: AN INTRODUCTION:-

Credit card fraud is an inclusive term for fraud committed using a payment card, such as a credit card or debit card.[1] The purpose may be to obtain goods or services or to make payment to another account, which is controlled by a criminal. The Payment Card Industry Data Security Standard (PCI DSS) is the data security standard created to help financial institutions process card payments securely and reduce card fraud.

Credit card fraud can be authorised, where the genuine customer themselves processes payment to another account which is controlled by a criminal, or unauthorised, where the account holder does not provide authorisation for the payment to proceed and the transaction is carried out by a third party. In 2018, unauthorised financial fraud losses across payment cards and remote banking totalled £844.8 million in the United Kingdom.

There are different types of credit card fraudulent activities occurs everywhere, includes,

1. Skimming
2. Dumpster diving
3. Phishing
4. Keystroke capturing
5. SIM Swap
6. Application fraud

7. Hacking. Etc

n conclusion, implementing robust credit card fraud detection systems is essential to safeguard financial transactions and protect both consumers and businesses from potential fraudulent activities.

ROLE OF DATA SCIENCE IN CREDIT CARD FRAUD DETECTION :-

Nowadays, data has become the most valuable asset in this sphere. Data science is a necessary requirement for banks to keep up with their rivals, attract more clients, increase the loyalty of existing clients, make more efficient data-driven decisions, empower their business, enhance operational efficiency, improve existing services/products and introduce new ones, reinforce security, and, as a result of all these actions, obtain more revenue. It is not surprising that the majority of all data science job demand comes from banking.

Data science allows the banking industry to successfully perform numerous tasks, including:

1. Investment risk analysis
2. Customer lifetime value prediction
3. Customer segmentation
4. Customer churn rate prediction
5. Personalized marketing
6. Customer sentiment analysis
7. Virtual assistants and chatbot

Data science plays a critical role in modern credit card fraud detection, revolutionizing how financial institutions safeguard their customers' assets. By harnessing the power of data analysis, machine learning, and real-time monitoring, data science helps identify and prevent fraudulent activities, providing a more secure and seamless experience for cardholders. In this introduction, we will explore the fundamental aspects of this role, highlighting how data science contributes to the ongoing battle against credit card fraud.

As the digital economy grows, so does the sophistication of fraudulent activities. Credit card fraudsters continually devise new tactics to exploit vulnerabilities, making it imperative for financial institutions to stay ahead of these threats..

DESIGN THINKING :

The goal of this project is to use machine learning techniques to improve the accuracy of detecting credit card fraudulent activities. The model will be trained on a dataset of credit card transactions and the goal is to improve the measure called Area Under the Precision-Recall Curve (AUPRC).

Design thinking is a non-linear, iterative process that teams use to understand users, challenge assumptions, redefine problems and create innovative solutions to prototype and test.
Here are some of the challenges that complicate the fraud detection process :

1. Changing fraud patterns over time : This one is the toughest to address since the fraudsters are always in the lookout to find new and innovative ways to get around the systems to commit the act. Thus it becomes all-important for the deep learning models to be updated with the evolved patterns to detect. This results in a decrease in the model's performance and efficiency. Thus the machine learning models need to keep updating or fail their objectives.
2. Class Imbalance : Practically only a small percentage of customers have fraudulent intentions. Consequently, there's an imbalance in the classification of fraud detection models (that usually classify transactions as either fraudulent or non-fraudulent) which makes it harder to build them. The fallout of this challenge is a poor user experience for genuine customers, since catching the fraudsters usually involves declining some legitimate transactions.
3. Model Interpretations: This limitation is associated with the concept of explainability since models typically give a score indicating whether a transaction is likely to be fraudulent or not — without explaining why.
4. Feature generation can be time-consuming :Subject matter experts can require long periods of time to generate a comprehensive feature set which slows down the fraud detection process.

PHASES OF DEVELOPMENT:

Data Source:

A data source is the location where data that is being used originates from. A data source may be the initial location where data is born or where physical information is first digitized, however even the most refined data may serve as a source, as long as another process accesses and utilizes it.

Data Pre-processing:

Data Pre-processing is the process of converting raw data into a format that is understandable and usable. It is a crucial step in any Data Science project to carry out an efficient and accurate analysis. It ensures that data quality is consistent before applying any Machine Learning or Data Mining techniques.

Feature Engineering:

Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.

Model Selection:

Model selection is the process of selecting the best model from all the available models for a particular business problem on the basis of different criterions such as robustness and model complexity.

Some of the important data science algorithms include regression, classification, clustering techniques, decision trees, random forests, and machine learning techniques like supervised, unsupervised, and reinforcement learning.

Model training:

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

Evaluation:

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

DESCRIBING DATASET:

A Dataset is a set or collection of data. This set is normally presented in a tabular pattern. Every column describes a particular variable. And each row corresponds to a given member of the data set, as per the given question.

Data sets describe values for each variable for unknown quantities such as height, weight, temperature, volume, etc., of an object or values of random numbers. The data set consists of data of one or more members corresponding to each row.

This is a short description about the data set given: (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)

Number of rows: 172792

Number of columns: 31

Total number of data entered: 5010968

DATA PRE-PROCESSING STEPS:

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

Data preprocessing is an important step in the data mining process that involves cleaning and transforming raw data to make it suitable for analysis. Some common steps in data preprocessing include:

Data Cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.
Data Integration: This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.
Data Transformation: This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.
Data Reduction: This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.
Data Discretization: This involves dividing continuous data into discrete categories or intervals. Discretization is often used in data mining and machine learning algorithms that require categorical data. Discretization can be achieved through techniques such as equal width binning, equal frequency binning, and clustering.
Data Normalization: This involves scaling the data to a common range, such as between 0 and 1 or -1 and 1. Normalization is often used to handle data with different units and scales. Common normalization techniques include min-max normalization, z-score normalization, and decimal scaling.
Data preprocessing plays a crucial role in ensuring the quality of data and the accuracy of the analysis results. The specific steps involved in data preprocessing may vary depending on the nature of the data and the analysis goals

MODEL TRAINING PROCESS:
Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range. The p urp ose of model training is to build the best mathematical representation of the relationship between data features and a target label (in supervised learning) or among the features themselves (unsupervised learning). Loss functions are a critical aspect of model training since they define how to optimize the machine learning algorithms. Depending on the objective, type of data and algorithm, data science practitioner use different type of loss functions. One of the popular examples of loss functions is Mean Square Error (MSE).

Data Collection

After defining the problem statement, it is necessary to investigate and gather data that can be used to feed the machine. This is an important stage in the process of creating an ML model because the quantity and quality of the data used will decide how effective the model is going to be. Data can be gathered from pre-existing databases or can be built from the scratch

Preparing The Data

The data preparation stage is when data is profiled, formatted and structured as needed to make it ready for training the model. This is the stage where the appropriate characteristics and attributes of data are selected. This stage is likely to have a direct impact on the execution time and results. This is also at the stage where data is categorized into two groups – one for training the ML model and the other for evaluating the model. Pre-processing of data by normalizing, eliminating duplicates and making error corrections is also carried out at this stage.

Assigning Appropriate Model / Protocols

Picking and assigning a model or protocol has to be done according to the objective that the ML model aims to achieve. There are several models to pick from, like linear regression, k-means and bayesian. The choice of models largely depends on the type of data that is being used. For instance, image processing convolutional neural networks would be the ideal pick and k-means would work best for segmentation.

Assigning Appropriate Model / Protocols

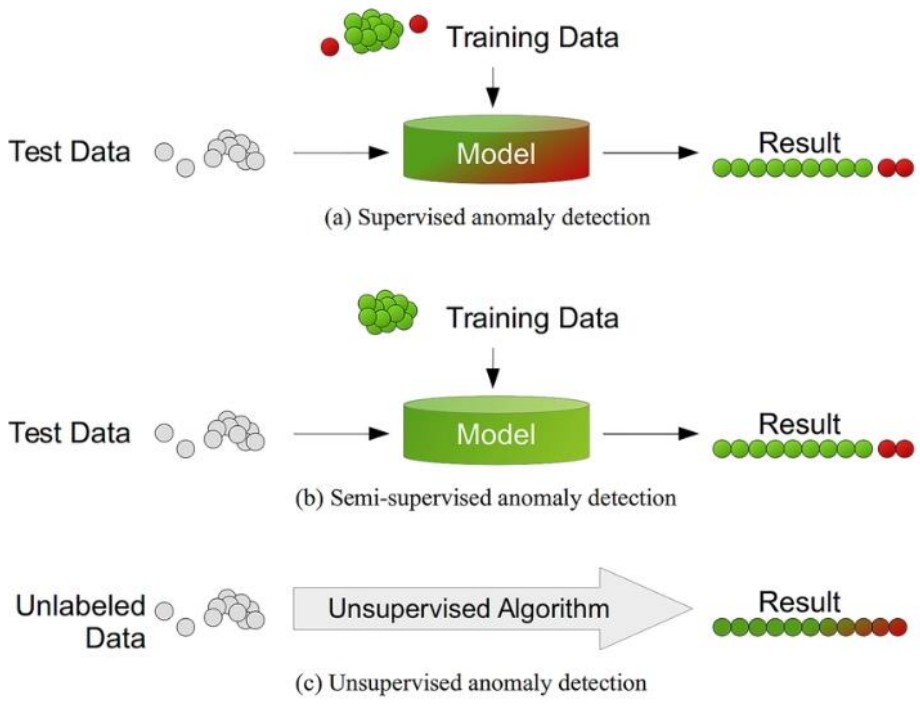
Picking and assigning a model or protocol has to be done according to the objective that the ML model aims to achieve. There are several models to pick from, like linear regression, k-means and bayesian. The choice of models largely depends on the type of data that is being used. For instance, image processing convolutional neural networks would be the ideal pick and k-means would work best for segmentation.

ANAMOLY DETECTION ALGORITHM:

Anomalies are data points that stand out amongst other data points in the dataset and do not confirm the normal behavior in the data. These data points or observations deviate from the dataset's normal behavioral patterns.

Anomaly detection is an unsupervised data processing technique to detect anomalies from the dataset. An anomaly can be broadly classified into different categories:

1. Outliers: Short/small anomalous patterns that appear in a non-systematic way in data collection.
2. Change in Events: Systematic or sudden change from the previous normal behavior.
3. Drifts: Slow, undirectional, long-term change in the data.



Anomalies detection are very useful to detect fraudulent transactions, disease detection, or handle any case studies with high-class imbalance. Anomalies detection techniques can be used to build more robust data science models.

1. Supervised Anamoly detection:

Supervised anomaly detection techniques require a data set that has been labeled as “normal” and “abnormal” and involves training a classifier. However, this approach is rarely used in anomaly detection due to the general unavailability of labelled data and the inherent unbalanced nature of the classes.

2. Semi-Supervised Anomalydetection:

Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning. It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model.

3. Un-supervised Anomaly detection:

Unsupervised learning is a type of machine learning that does not rely on labeled data to find patterns or clusters in the data. One of the applications of unsupervised learning is anomaly detection, which is the task of identifying outliers or abnormal instances in the data.

LOGISTIC REGRESSION ALGORITHMS:

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class or not. It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables. It is a powerful tool for decision-making. For example email spam or not.

It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

The output of logistic regression is always between (0, and 1), which is suitable for a binary classification task. The higher the value, the higher the probability that the current sample is classified as class=1, and vice versa.

He(x)=1/(1+e^(-ex))

As the formula above shows, n is the parameter we want to learn or train or optimize and Equation is the input data. The output is the prediction value when the value is closer to 1, which means the instance is more likely to be a positive sample(y=1). If the value is closer to 0, this means the instance is more likely to be a negative sample(y=0).

To optimize our task, we need to define a loss function(cost or objective function) for this task. In logistic regression, we use the log-likelihood loss function.

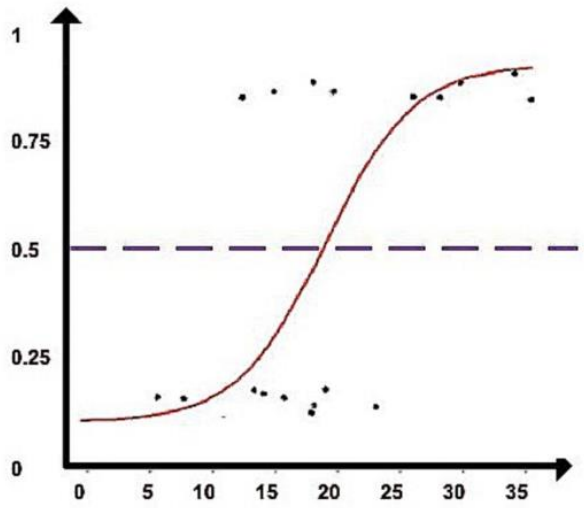
How logistic regression algorithm works:

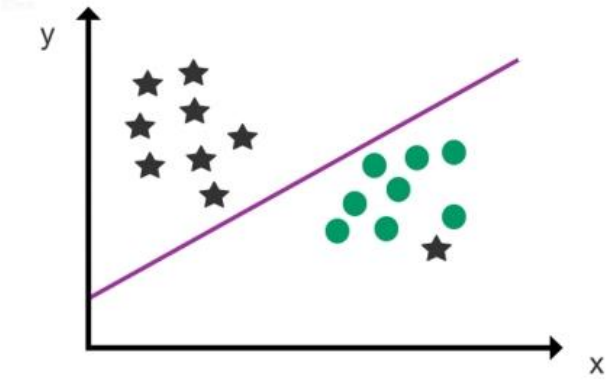
Machine learning generally involves predicting a quantitative outcome or a qualitative class. The former is commonly referred to as a regression problem. In the scenario of linear regression, the input is a continuous variable, and the prediction is a numerical value. When predicting a qualitative outcome (class), the task is considered a classification problem. Examples of classification problems include predicting what products a user will buy or if a target user will click on an online advertisement.

Not all algorithms fit cleanly into this simple dichotomy, though, and logistic regression is a notable example. Logistic regression is part of the regression family as it involves predicting outcomes based on quantitative relationships between variables. However, unlike linear regression, it accepts both continuous

and discrete variables as input and its output is qualitative. In addition, it predicts a discrete class such as “Yes/No” or “Customer/Non-customer”.

In practice, the logistic regression algorithm analyzes relationships between variables. It assigns probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into an expression of probability between 0 and 1.0. Probability is either 0 or 1, depending on whether the event happens or not. For binary predictions, you can divide the population into two groups with a cut-off of 0.5. Everything above 0.5 is considered to belong to group A, and everything below is considered to belong to group B.





DECISION TREE ALGORITHM:

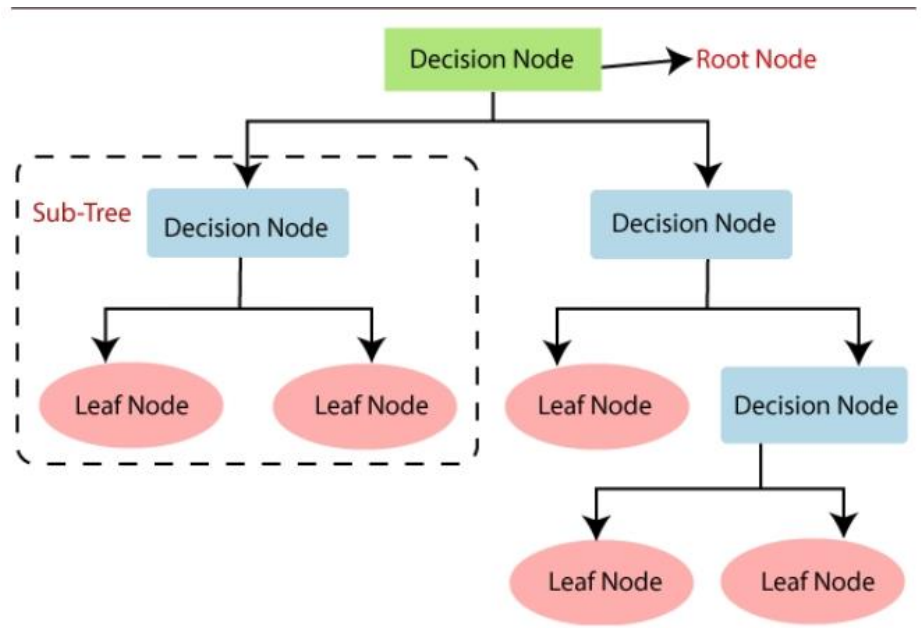
A decision tree is a non-parametric supervised learning algorithm for classification and regression tasks. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression tasks, providing easy-to-understand models.

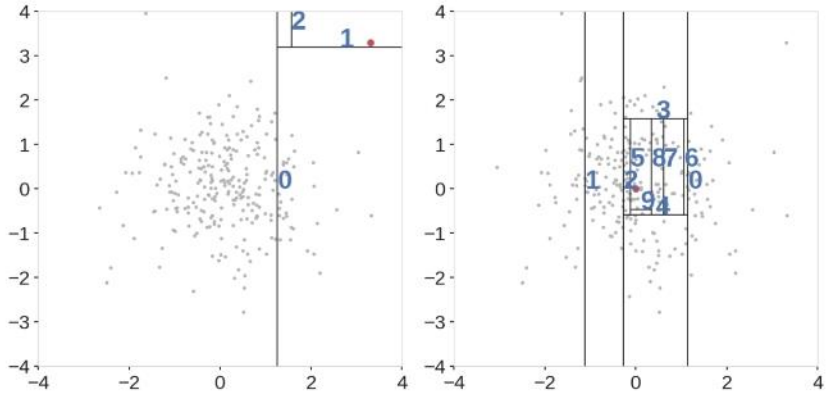
A decision tree is a hierarchical model used in decision support that depicts decisions and their potential outcomes, incorporating chance events, resource expenses, and utility. This algorithmic model utilizes conditional control statements and is non-parametric, supervised learning, useful for both classification and regression tasks. The tree structure is comprised of a root node, branches, internal nodes, and leaf nodes, forming a hierarchical, tree-like structure.

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

Some of the important terminologies in decision tree algorithm are:

1. Root Node: The initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.
2. Decision Nodes: Nodes resulting from the splitting of root nodes are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.
3. Leaf Nodes: Nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.
4. Sub-Tree: Similar to a subsection of a graph being called a sub-graph, a sub-section of a decision tree is referred to as a sub-tree. It represents a specific portion of the decision tree.
5. Pruning: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.
6. Branch / Sub-Tree: A subsection of the entire decision tree is referred to as a branch or sub-tree. It represents a specific path of decisions and outcomes within the tree.
7. Parent and Child Node: In a decision tree, a node that is divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes. The parent node represents a decision or condition, while the child nodes represent the potential outcomes or further decisions based on that condition.





(a) Anomaly point (b) Nominal point

CREDIT CARD FRAUD DETECTION:

Credit card fraud detection is a set of methods and techniques designed to block fraudulent purchases, both online and in-store. This is done by ensuring that you are dealing with the right cardholder and that the purchase is legitimate.

In this project we are using

1. PYTHON as programming language,
2. PYTHON PANDAS for working with data sets.
3. Google Colab for compiling.
4. Kaggle website for data sets about credit card fraud detection.

Now we are going to take a deep look into the processes involved in this project.

Data Understanding:

Data understanding involves accessing the data and exploring it using tables and graphics.

Data preparation is the process of preparing raw data so that it is suitable for further processing and analysis. Key steps for understanding the datasets of credit card fraud is collecting, cleaning, and labeling raw data into a form suitable for machine learning algorithms and then exploring and visualizing the data,

Importing dependencies:

It is necessary to import the software and requirements for further process, the requirements are

1. Importing numpy
2. Importing Python Pandas
3. Importing matplotlib
4. Importing Seaborn
5. Importing xgboost
6. Importing scipy

Data Analysis:

Data analysis is the process of cleaning, changing, and processing raw data and extracting actionable, relevant information that helps businesses make informed decisions. The procedure helps reduce the risks

inherent in decision-making by providing useful insights and statistics, often presented in charts, images, tables, and graphs.

Data Splitting:

Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model.

Data splitting is an important aspect of data science, particularly for creating models based on data. This technique helps ensure the creation of data models and processes that use data models – such as machine learning – are accurate.

Model Building:

Model building is an essential part of data analytics and is used to extract insights and knowledge from the data to make business decisions and strategies. In this phase of the project data science team needs to develop data sets for training, testing, and production purposes. These data sets enable data scientists to develop an analytical method and train it while holding aside some of the data for testing the model. Model building in data analytics is aimed at achieving not only high accuracy on the training data but also the ability to generalize and perform well on new, unseen data.

Cross Validation:

Cross-validation is a statistical technique used by data scientists for training and evaluating a machine learning model, with a focus on ensuring reliable model performance. To understand how cross-validation supports model development, we first need to understand how data is used to train and evaluate models.

Over Sampling:

Data Sampling is done by the following sampling techniques they are,

1. Random sampling: This data sampling method protects the data modeling process from bias toward different possible data characteristics. However, random splitting may have issues regarding the uneven distribution of data.
2. Stratified random sampling: This method selects data samples at random within specific parameters. It ensures the data is correctly distributed in training and test sets.
3. Nonrandomsampling:This approach is typically used when data modelers want the most recent data as the test set.

Hyper parameter tuning:

Hyperparameter tuning allows data scientists to tweak model performance for optimal results. This process is an essential part of machine learning, and choosing appropriate hyperparameter values is crucial for success. For example, assume you're using the learning rate of the model as a hyperparameter.

PROGRAM:

```
Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

From ipy widgets import interact,FloatSlider


from sklearn.preprocessing import StandardScaler, RobustScaler from
sklearn.model_selection importtrain_test_split,GridSearchCV from
sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,roc_curve,precision_recall_curve
```

```
from imblearn.under_sampling import TomekLinks

from imblearn.over_sampling import SMOTE


pd.options.display.max_columns=100
pd.options.display.max_rows=100
pd.options.display.width=100
plt.style.use('ggplot')

df=pd.read_csv('creditcard.csv') df.head()
```

OUTPUT1:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196

df.shape

OUTPUT2:

(284807,31)

df.isnull().sum().sum()

OUTPUT3:

0

df.describe()

OUTPUT4:

	Time	V1	V2	V3	V4	V5	V6	V7	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848(
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.1943
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.3216
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086.
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.0007

df.dtypes

OUTPUT5:

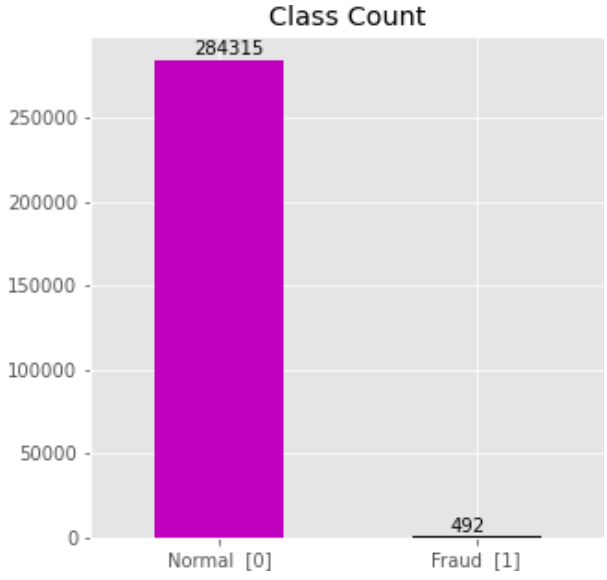
Time float64
V1 float64
V2 float64
V3 float64
V4 float64
V5 float64
V6 float64
V7 float64
V8 float64
V9 float64
V10 float64
V11 float64
V12 float64

```
V13       float64
V14       float64
V15       float64
V16       float64
V17       float64
V18       float64
V19       float64
V20       float64
V21       float64
V22       float64
V23       float64
V24       float64
V25       float64
V26       float64
V27       float64
V28       float64
Amount     float64
Class       int64
dtype: object
```

INPUT:

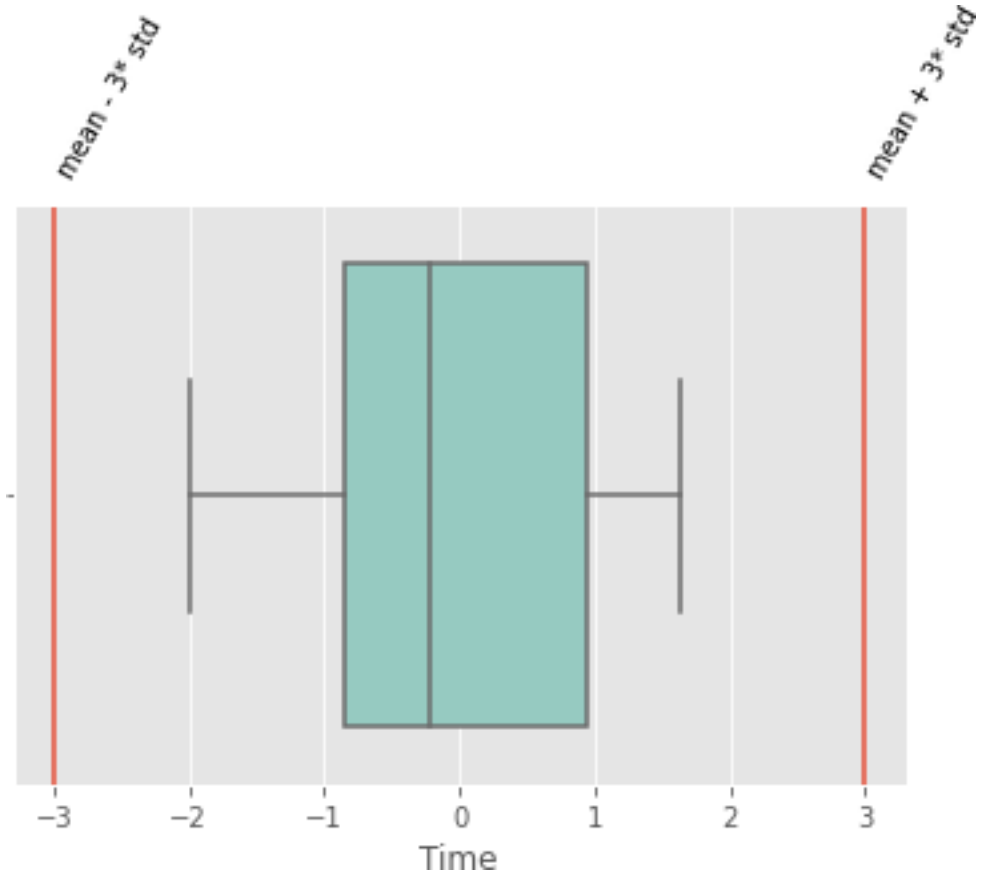
```
diff_class = df['Class'].value_counts() diff_class.plot(kind='bar',
color=['m', 'k'], figsize=(5, 5))
plt.xticks(range(2),['Normal[0]','Fraud[1]'],rotation=0)

 for i, v in enumerate(diff_class):
    plt.text(i-0.1,v+3000,str(v))
plt.title('Class Count') plt.show()
```

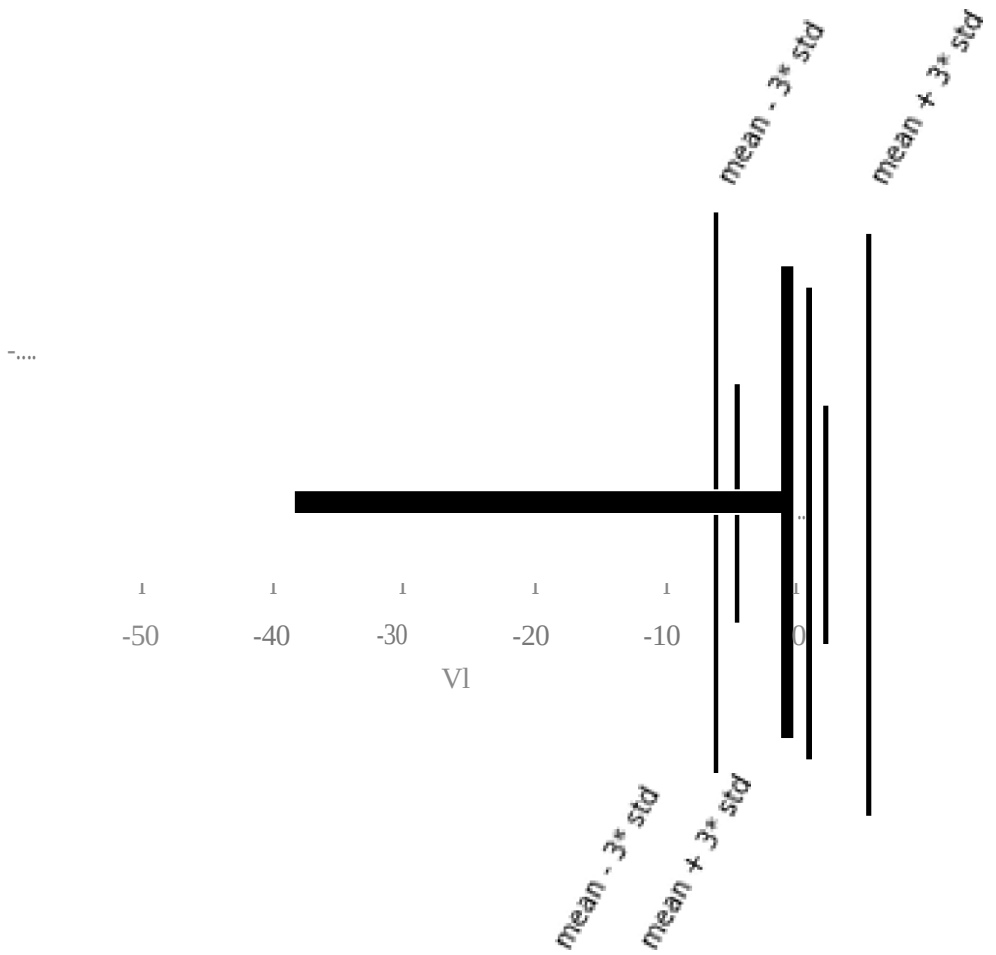
OUTPUT6:

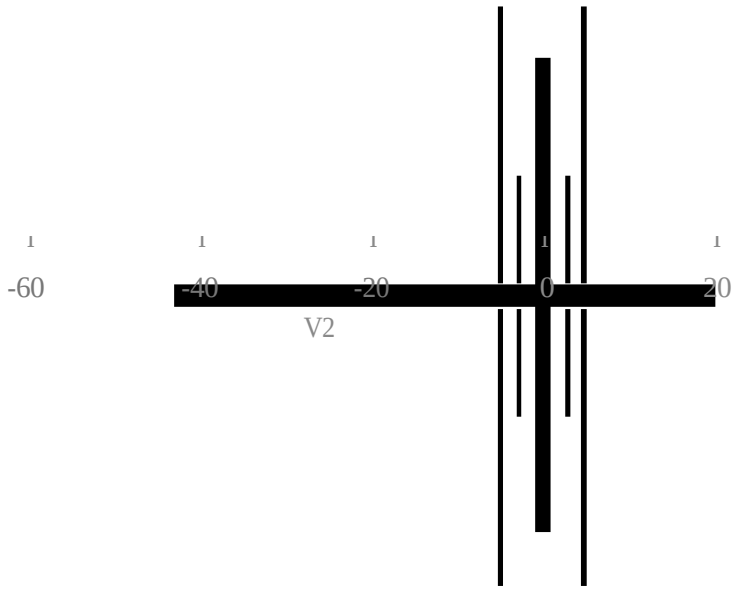


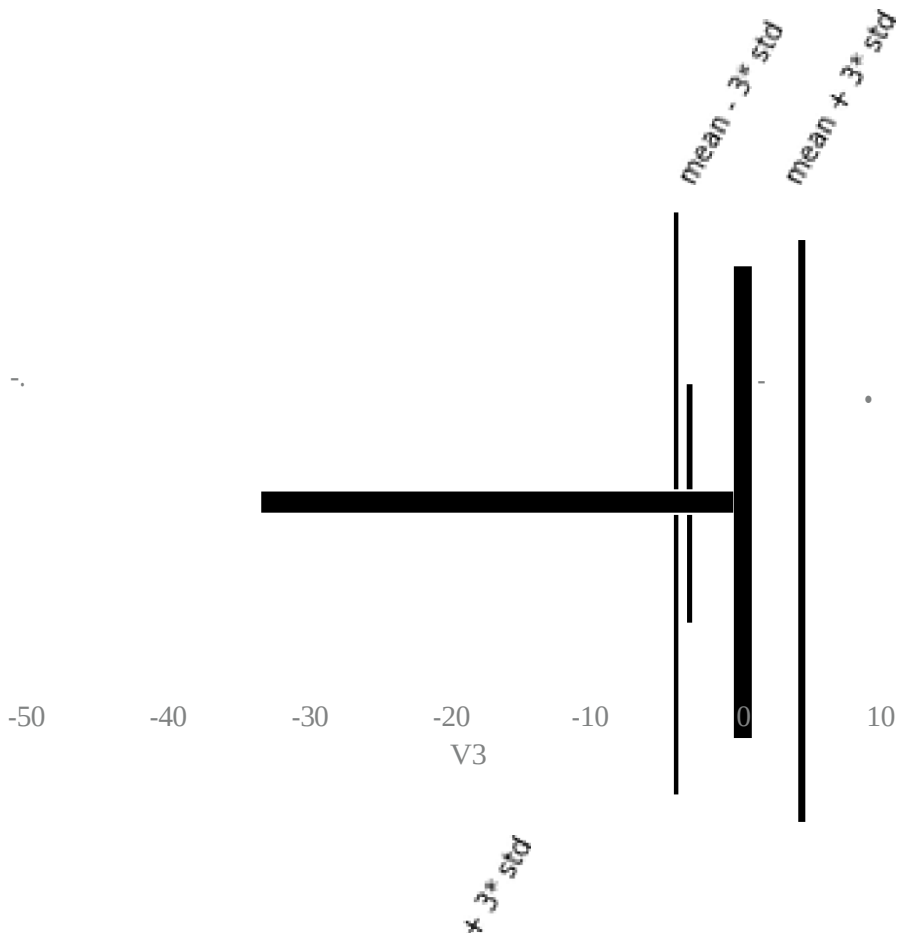
```
ss=StandardScaler()

df['Amount']=ss.fit_transform(df[['Amount']])
df['Time'] = ss.fit_transform(df[['Time']])


#Distributionofdifferentcolumns. for var


in df.columns[:-1]:


    sns.boxplot(df[var],hue=df['Class'],palette='Set3') mean =
    df[var].mean()
    std = df[var].std() plt.axvline(mean-
    3*std,0,1)
    plt.text(mean-3*std,-0.55,'mean-3*std',rotation=60) plt.axvline(mean + 3
    * std, 0, 1)
    plt.text(mean+3*std,-0.55,'mean+3*std',rotation=60) plt.show()
```

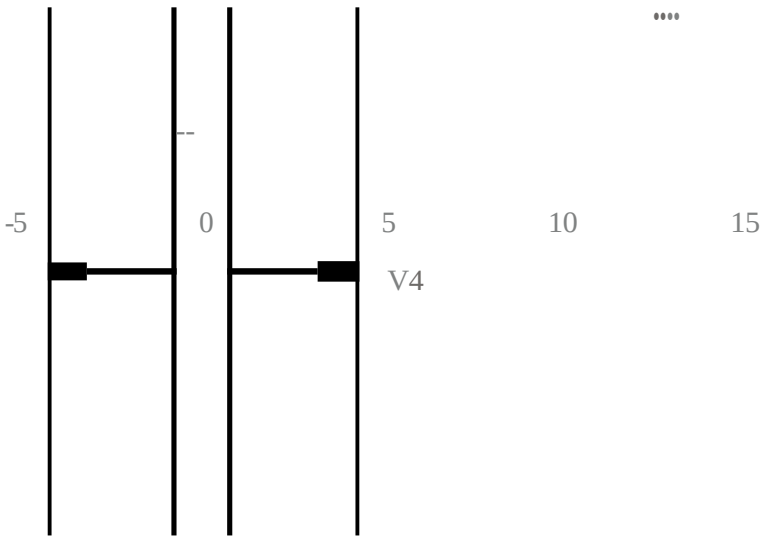
OUTPUTFORDATAINTRAININGANDDATASET:

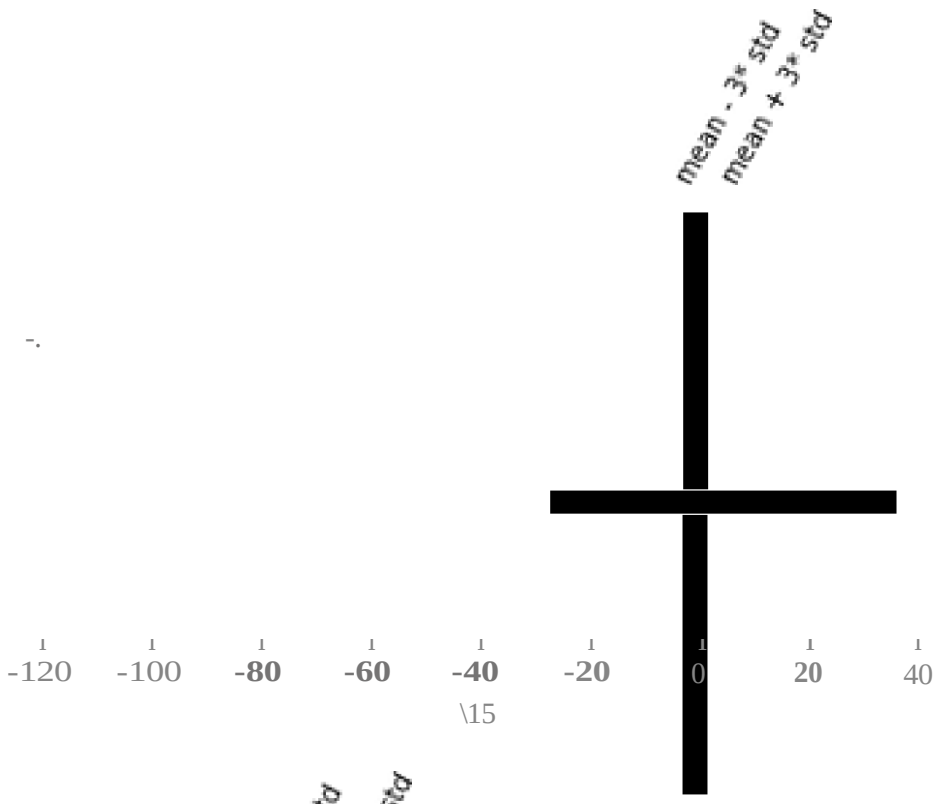



mean - 3* std
mean + 3* std
-50
-40
-30
-20
-10
0
V1
mean - 3* std
mean + 3* std

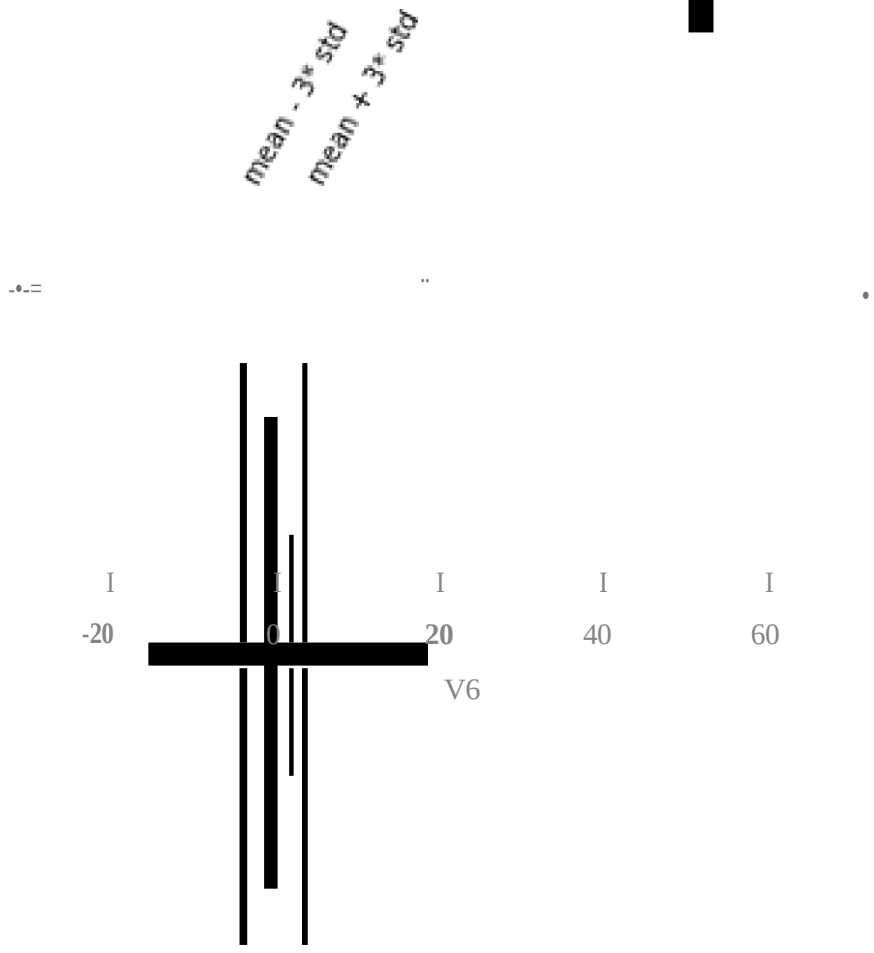

-60
-40
-20
0
20
V2


mean - 3* std
mean + 3* std
-50
-40
-30
-20
-10
0
10
V3

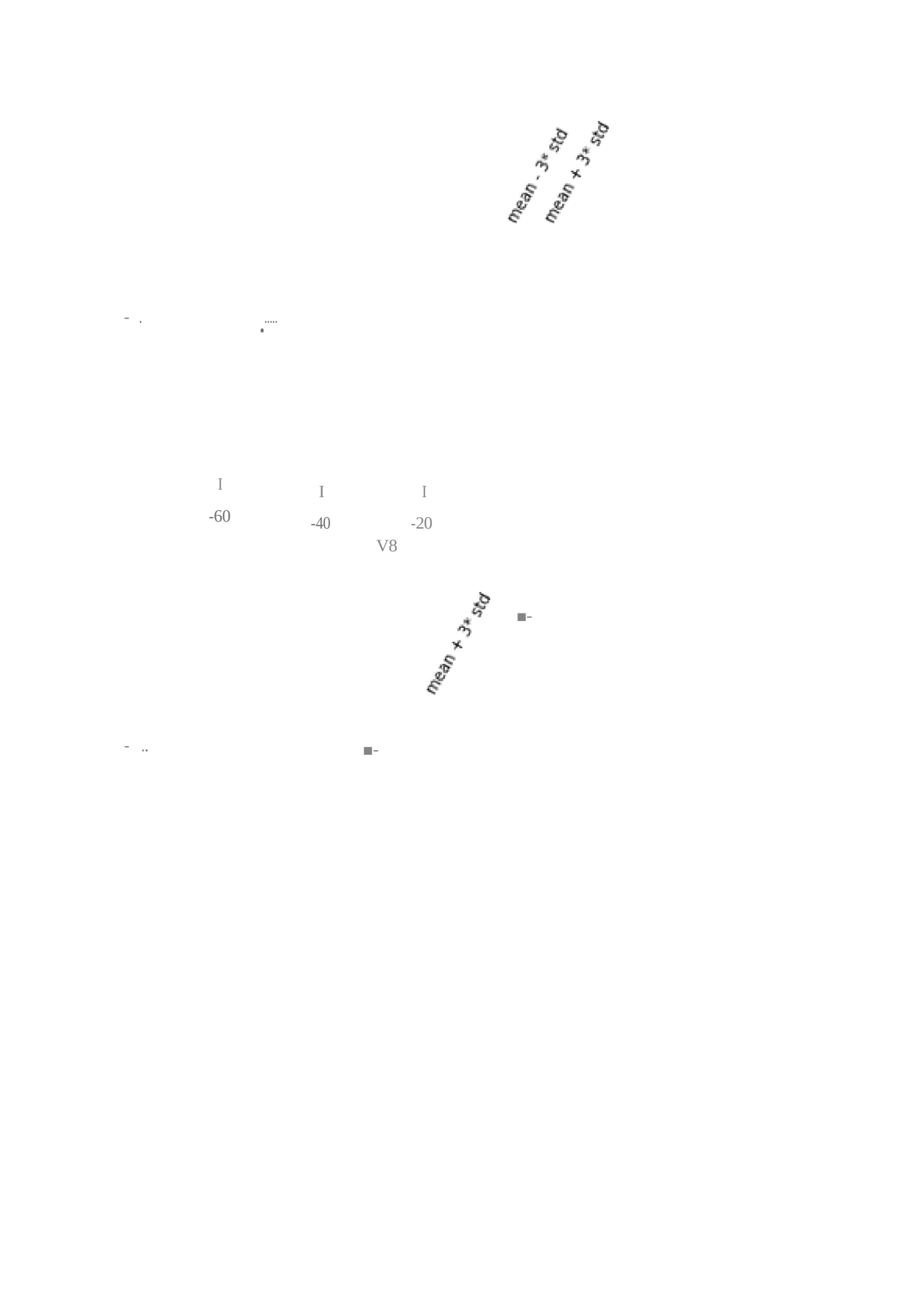

-5
0
5
10
15
V4


mean - 3* std
mean + 3* std
-120
-100
-80
-60
-40
-20
0
20
40

\15


mean - 3* std
mean + 3* std
-20
0
20
40
60

V6


mean - 3* std
mean + 3* std
I
I
I
-60
-40
-20
V8
mean + 3* std

I I

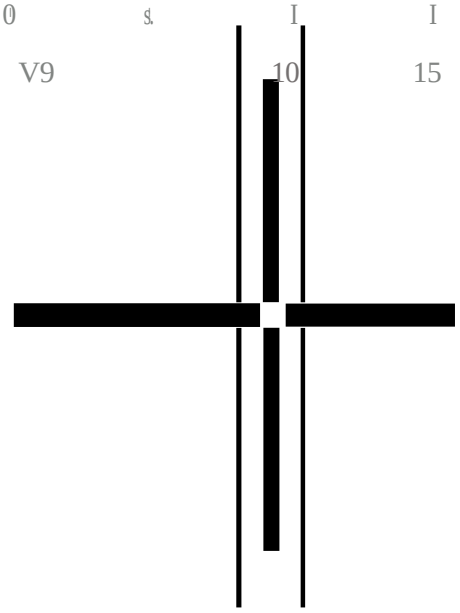
0 20

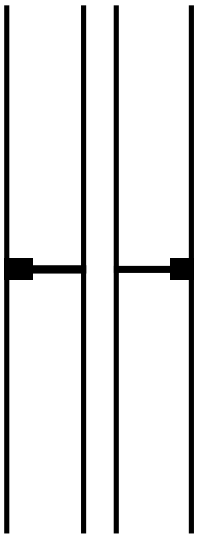
I

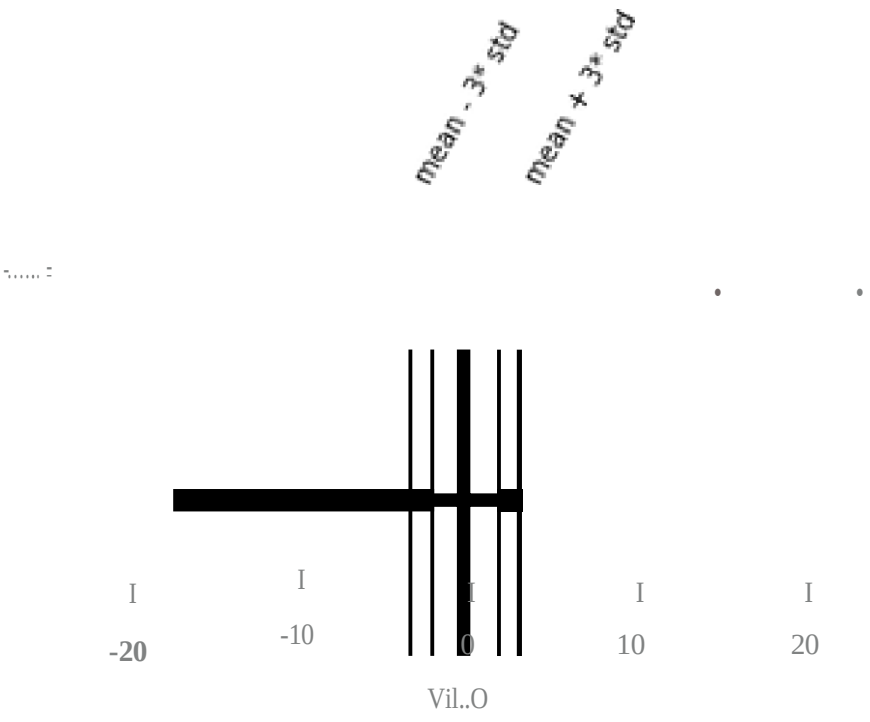
I

-10 -5 0 $ I I

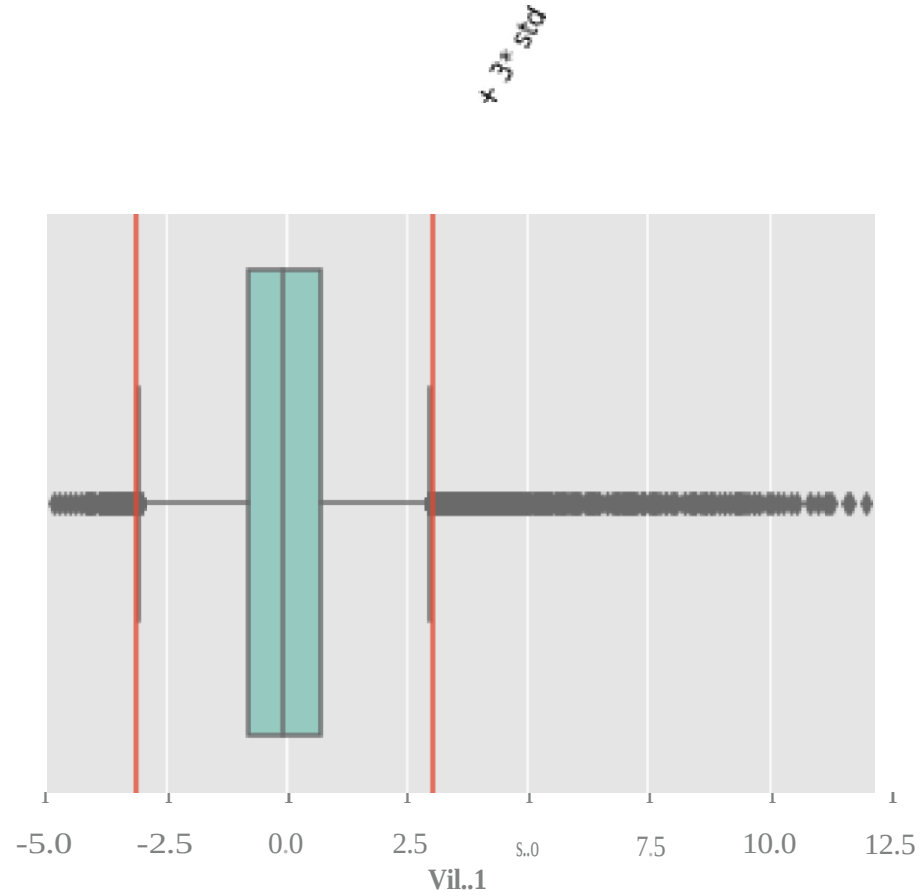
V9 10 15



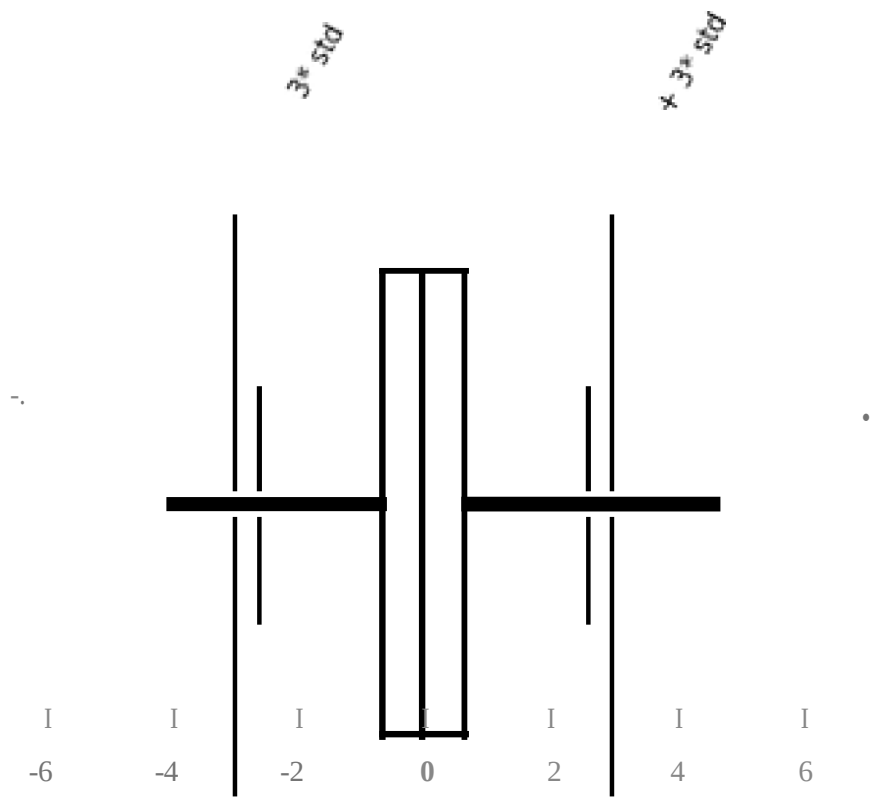



mean - 3* std
mean + 3* std
I
I
I
I
I
-20
-10
0
10
20

Vil..O

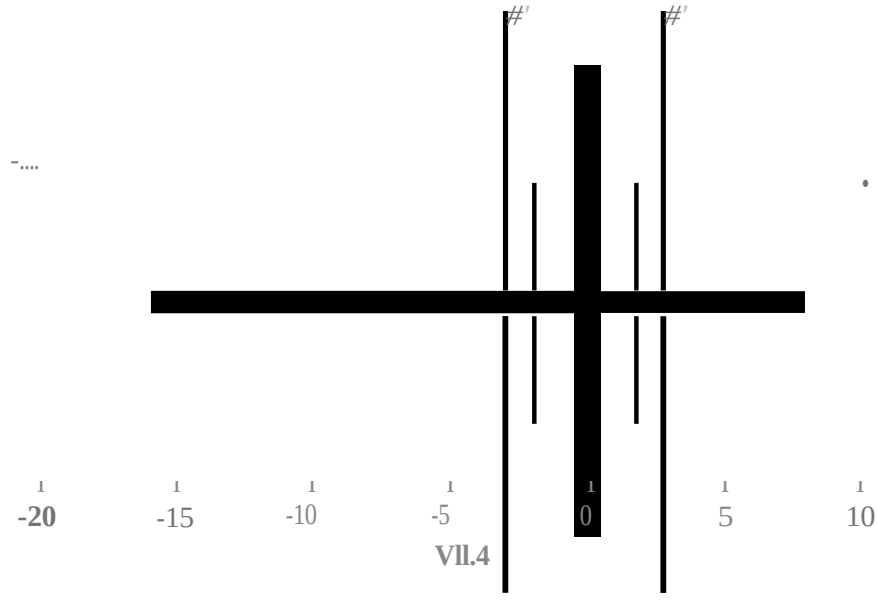

+ 3* std
-5.0
-2.5
0.0
2.5
7.5
10.0
12.5

Vil..1

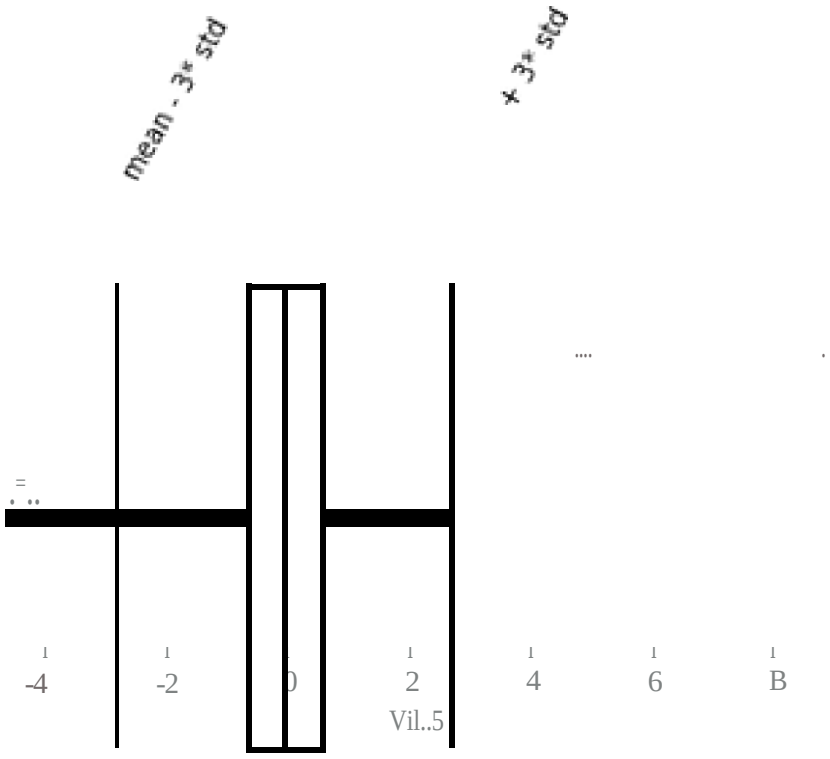


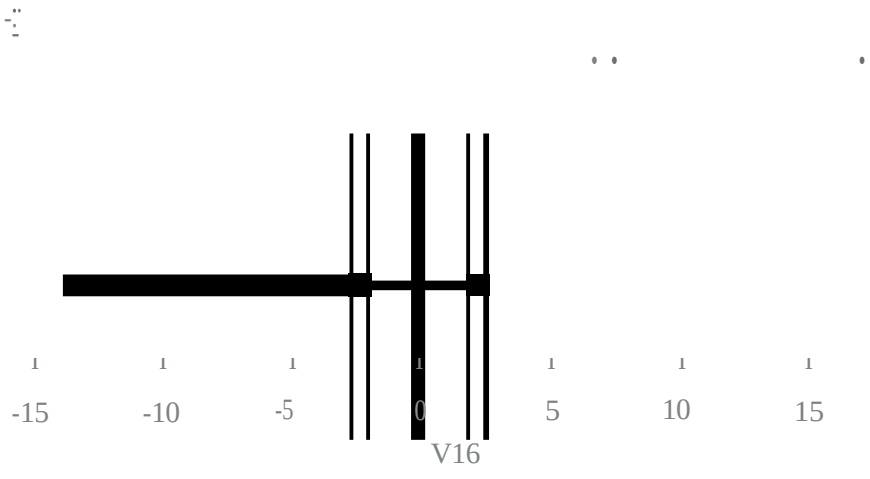
Vil..3

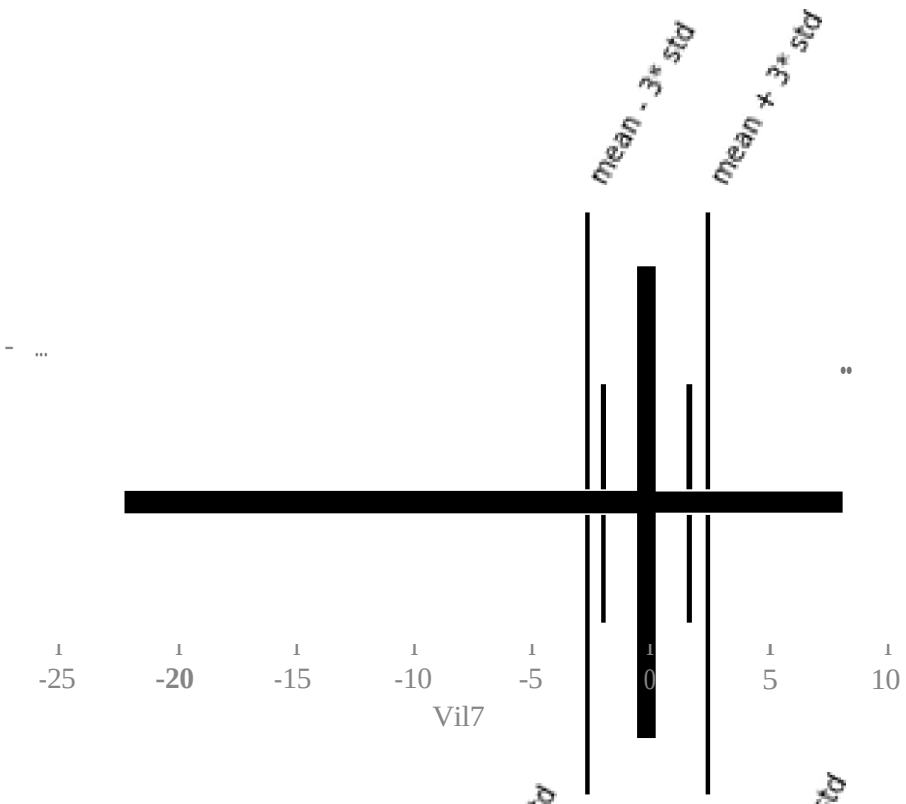
if *I*

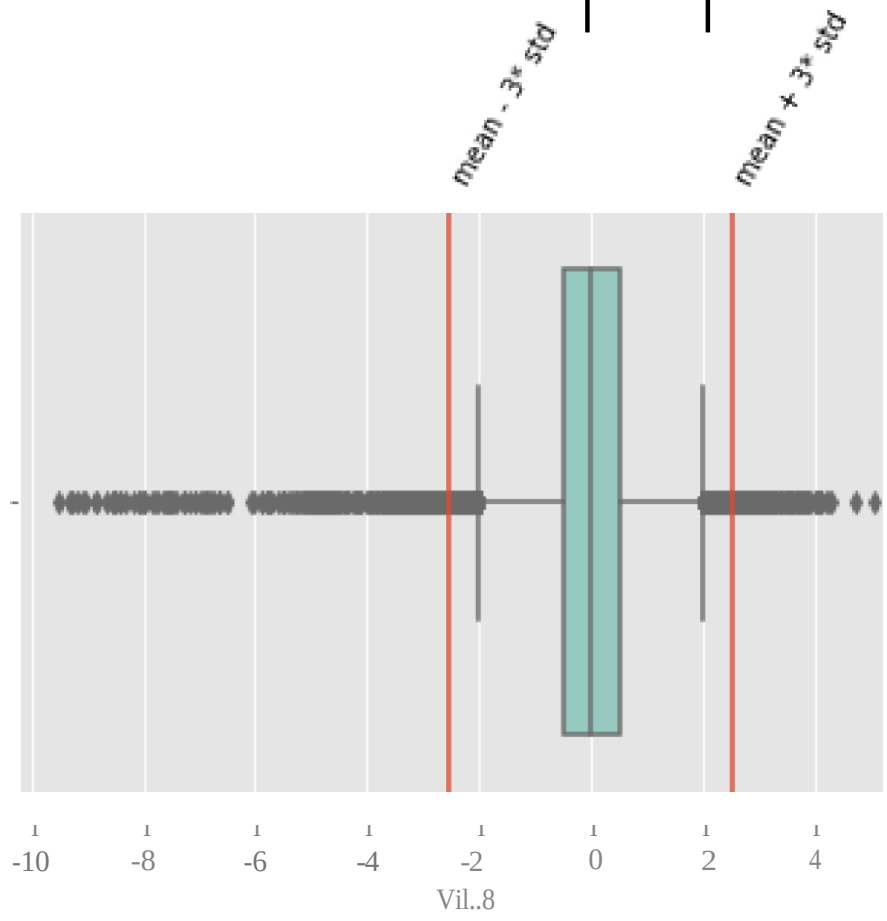


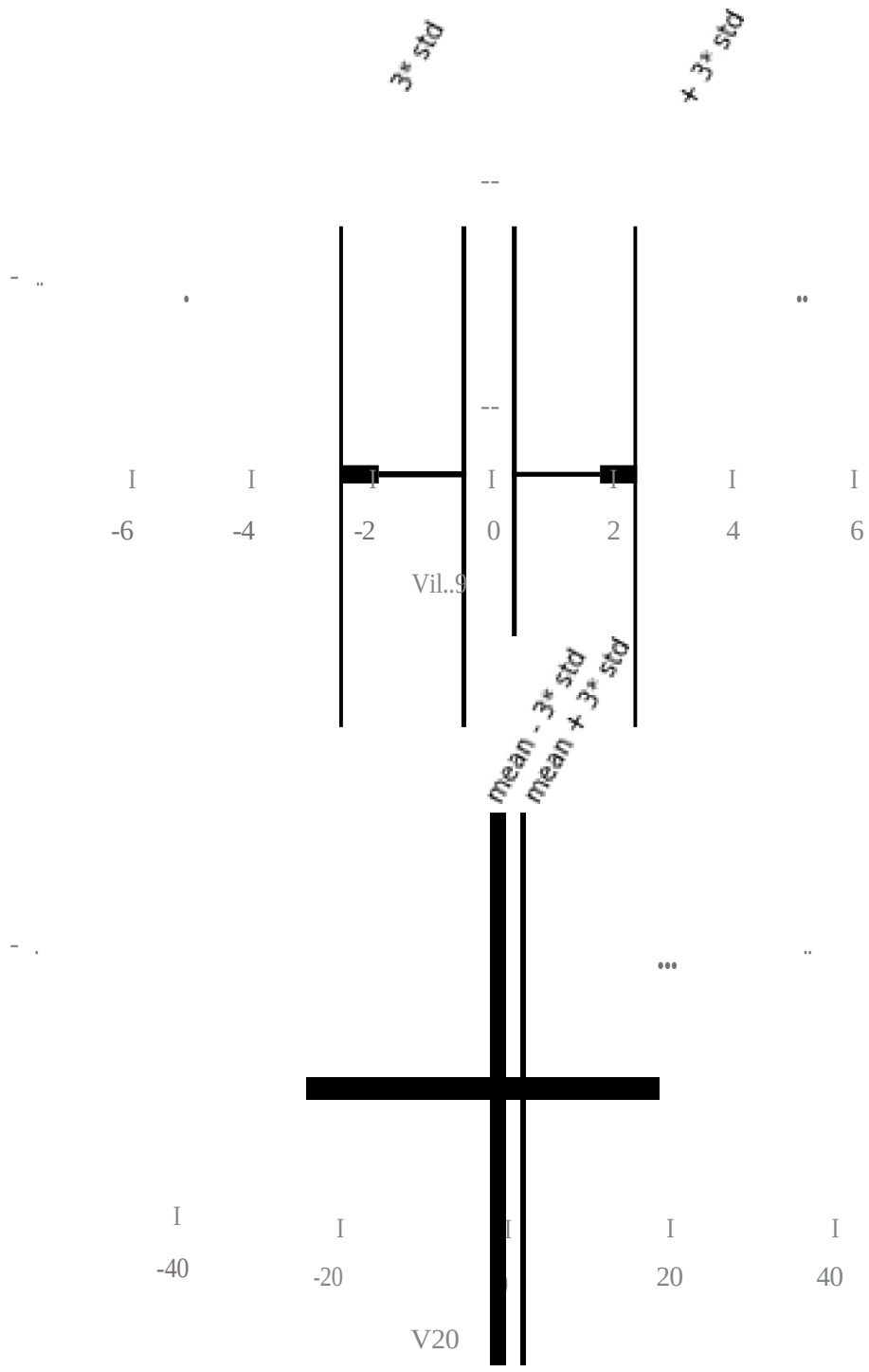
Vll.4

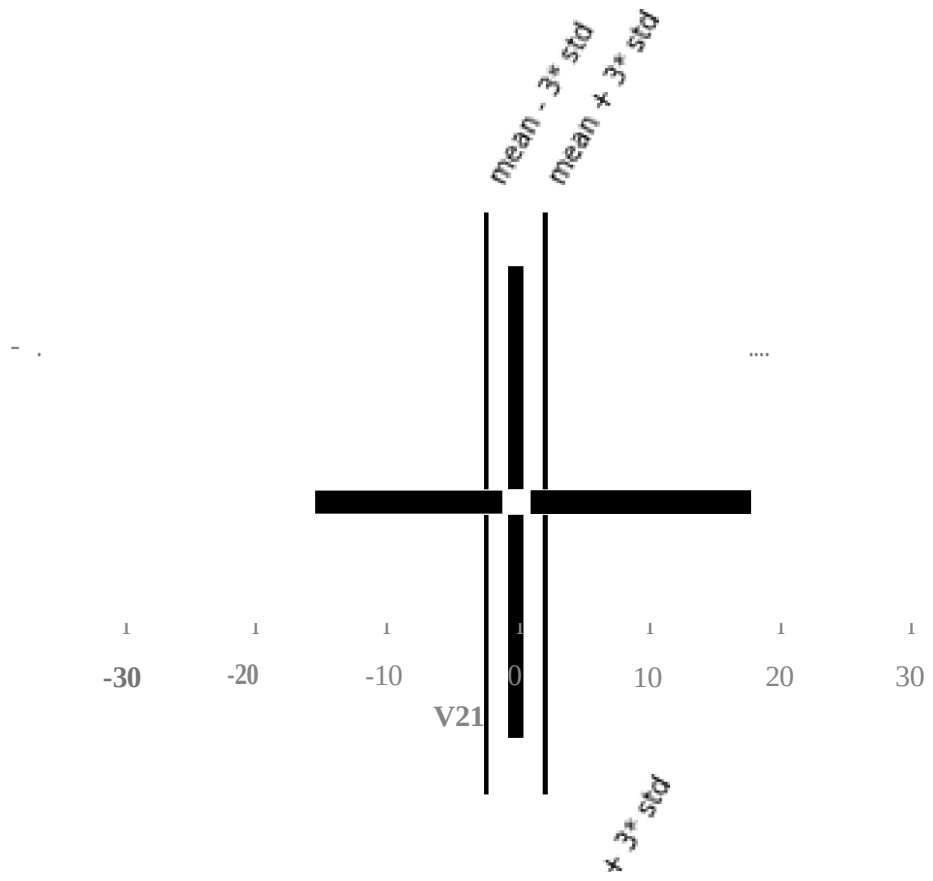

mean - 3* std
+ 3* std
-4
-2
0
2
4
6
B
Vil..5

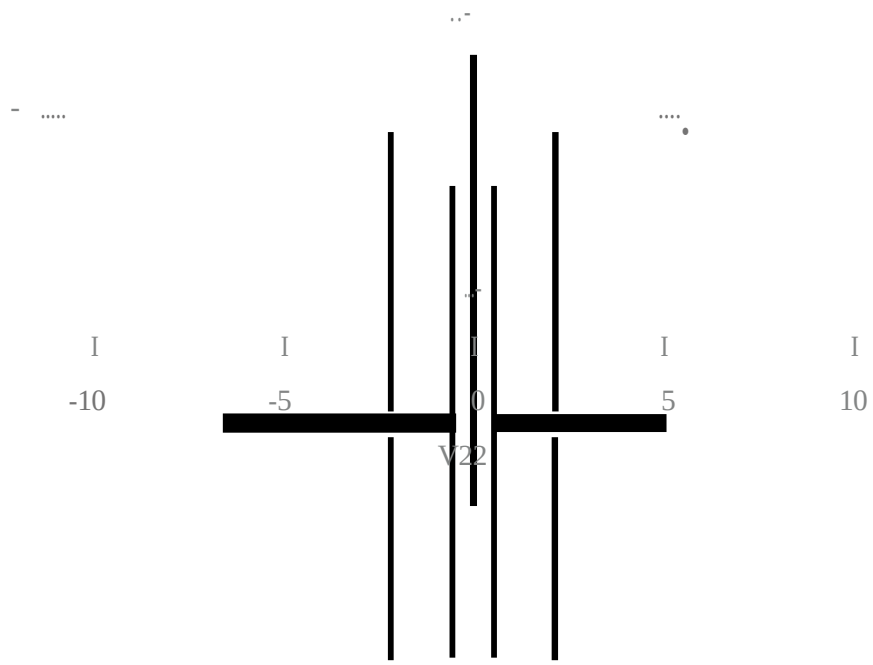

-15
-10
-5
0
5
10
15
V16

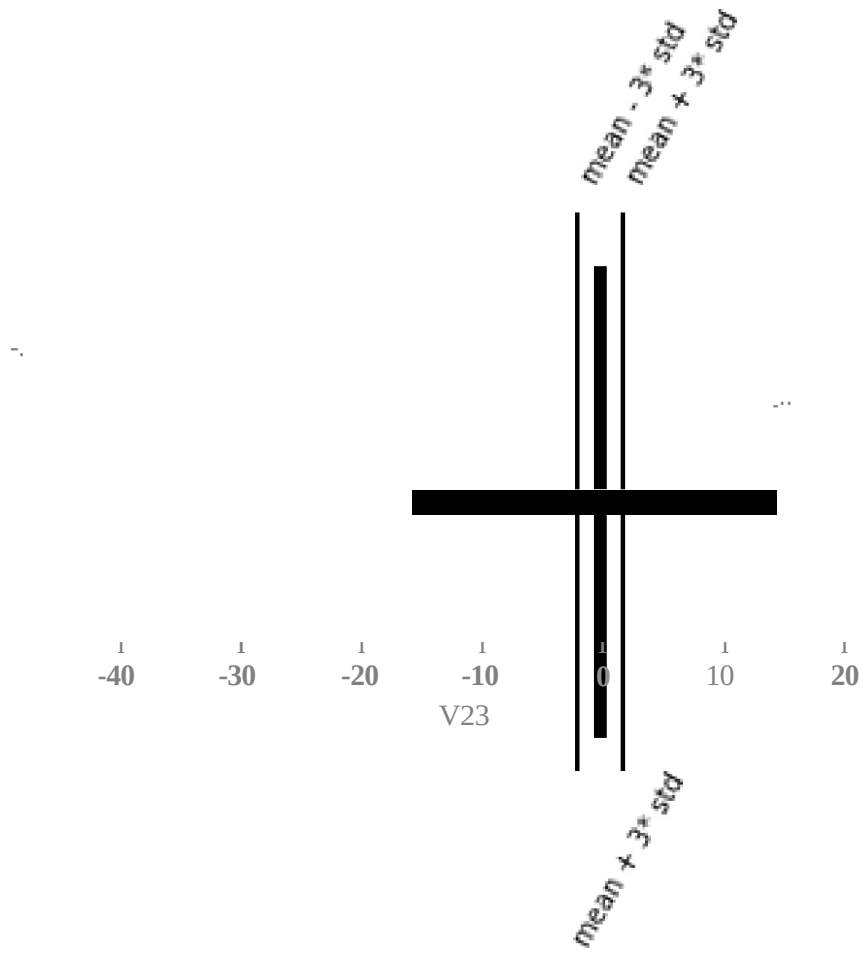

mean - 3* std
mean + 3* std
-25
-20
-15
-10
-5
0
5
10
Vil7

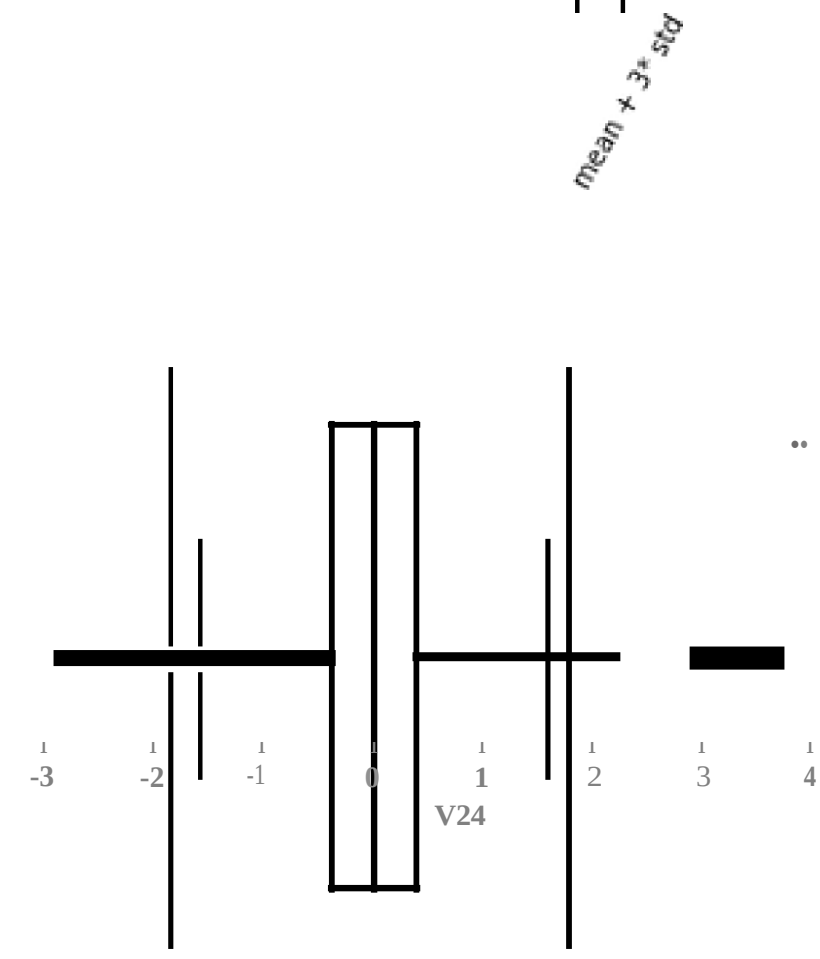

mean - 3* std
mean + 3* std
-10
-8
-6
-4
-2
0
2
4
Vil..8

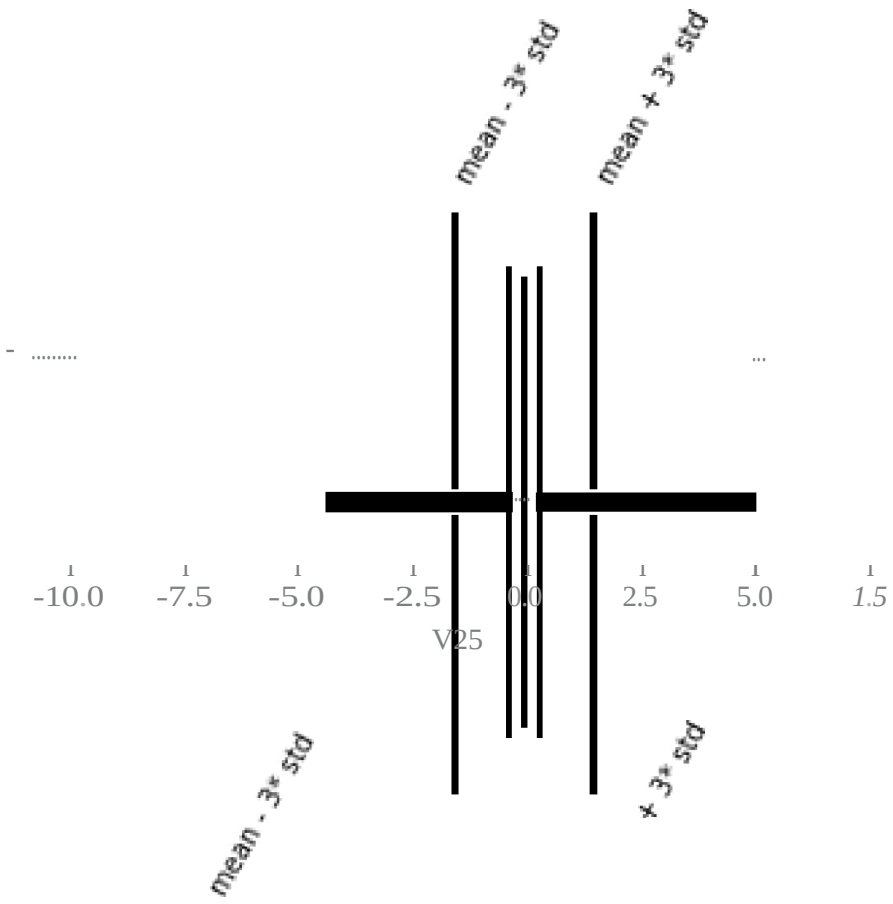

3* std
+ 3* std
-6
-4
-2
0
2
4
6
mean - 3* std
mean + 3* std
-40
-20
0
20
40
V20

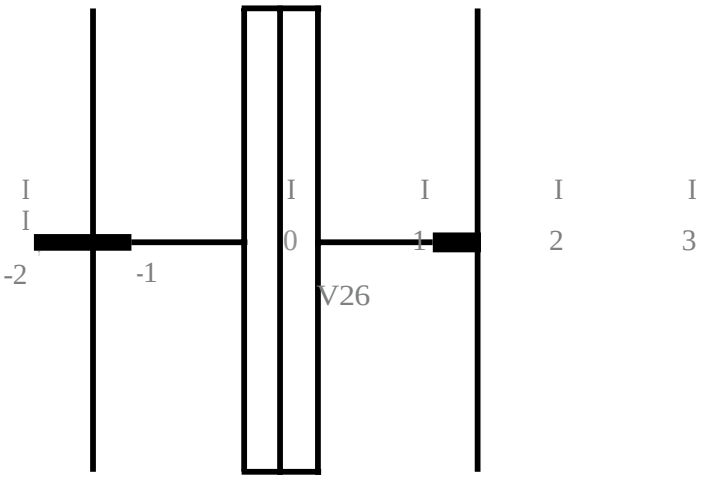

mean - 3* std
mean + 3* std
-30
-20
-10
0
10
20
30
V21
+ 3* std

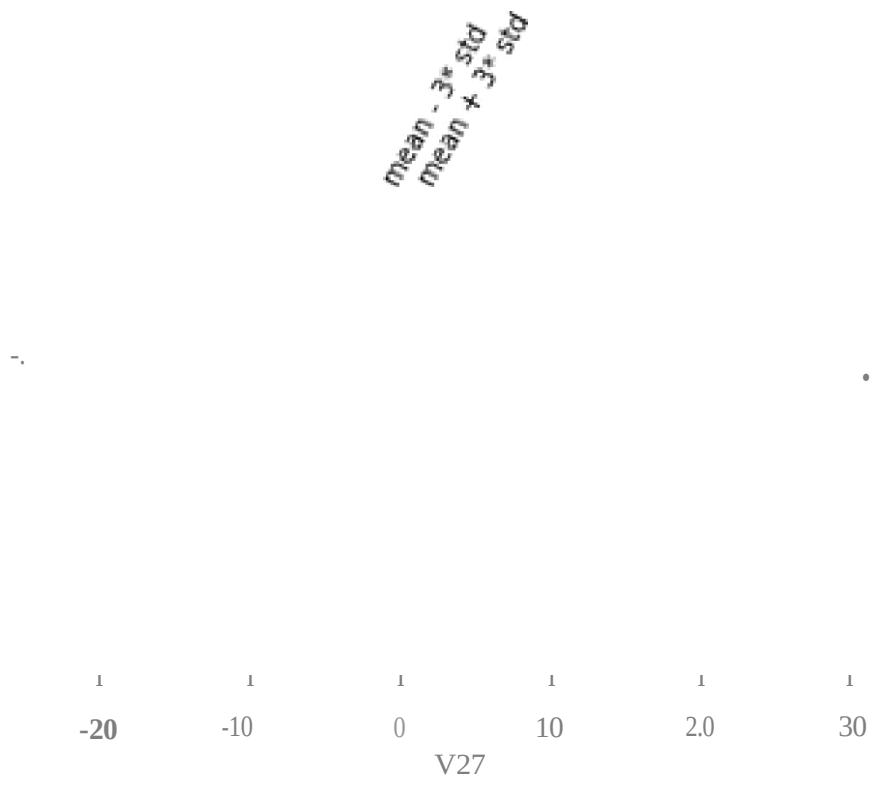

-10
-5
0
5
10
V22

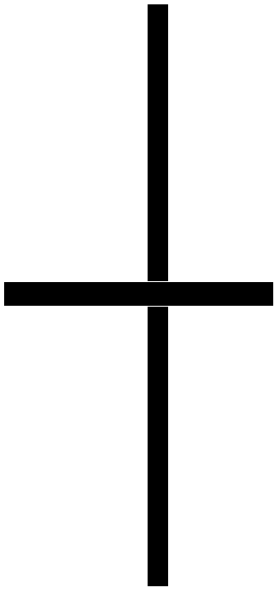

mean - 3* std
mean + 3* std
-40
-30
-20
-10
0
10
20
V23

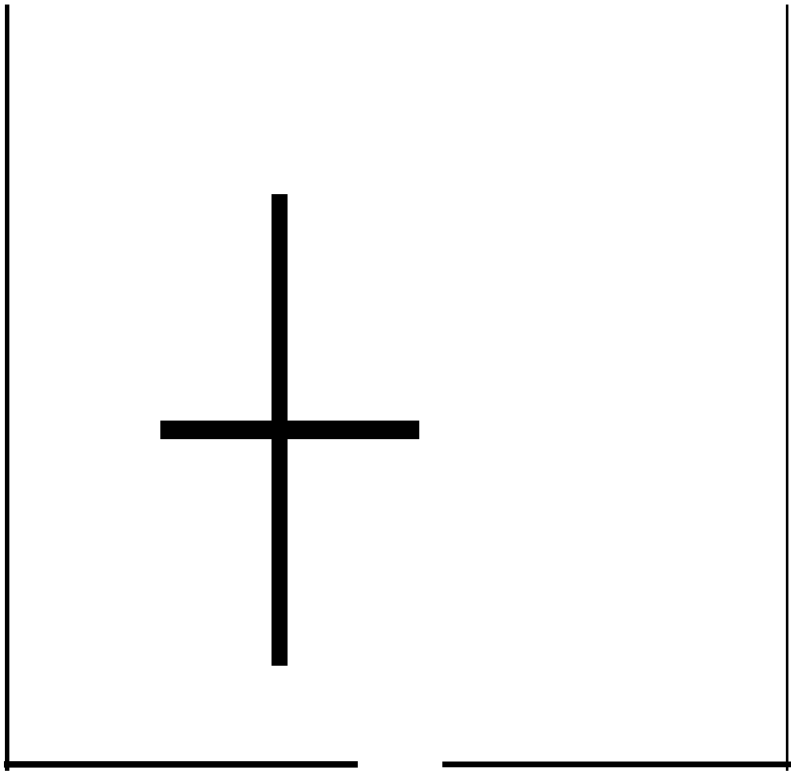

mean + 3* std
-3
-2
-1
0
1
2
3
4
V24


mean - 3* std
mean + 3* std
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0
1.5
V25
mean - 3* std
+ 3* std


-2
-1
0
1
2
3
V26


mean - 3* std
mean + 3* std
-20
-10
0
10
2.0
30
V27





I I I I

0 10 **20** 30

V28