rSure! Here’s a sample project for a fictitious pharmaceutical company, Chisamba Pharmaceuticals, focusing on inferential statistics using confidence intervals and p-values. We’ll use a fictitious dataset and plotly.graph\_objects for visualization.

**Project Overview**

**Title:** Inferential Statistical Analysis for Clinical Trials at Chisamba Pharmaceuticals

**Objective:** To analyze the effectiveness of a new drug developed by Chisamba Pharmaceuticals using inferential statistics. The analysis will include calculating confidence intervals and p-values to determine the drug’s efficacy.

**Theory and Purpose**

**Inferential Statistics:** Inferential statistics allow us to make inferences about a population based on a sample. In clinical trials, this helps determine whether a new drug is effective.

**Confidence Interval:** A confidence interval gives a range of values within which we expect the true population parameter to lie, with a certain level of confidence (e.g., 95%).

**P-Value:** The p-value helps us determine the significance of our results. A p-value less than 0.05 typically indicates that the results are statistically significant.

**Expertise Acquired**

* Understanding of inferential statistics in the context of clinical trials.
* Ability to calculate and interpret confidence intervals and p-values.
* Proficiency in data visualization using plotly.graph\_objects.
* Skills in data storytelling to effectively communicate the results.

**Sample Code**

**Python**

import numpy as np

import pandas as pd

import plotly.graph\_objects as go

from scipy import stats

# Fictitious dataset

np.random.seed(42)

data = {

'patient\_id': range(1, 101),

'before\_treatment': np.random.normal(loc=50, scale=10, size=100),

'after\_treatment': np.random.normal(loc=55, scale=10, size=100)

}

df = pd.DataFrame(data)

# Calculate the difference

df['difference'] = df['after\_treatment'] - df['before\_treatment']

# Calculate mean and standard deviation

mean\_diff = df['difference'].mean()

std\_diff = df['difference'].std()

# Calculate confidence interval

confidence\_level = 0.95

degrees\_freedom = len(df) - 1

confidence\_interval = stats.t.interval(confidence\_level, degrees\_freedom, mean\_diff, std\_diff/np.sqrt(len(df)))

# Perform t-test

t\_stat, p\_value = stats.ttest\_rel(df['before\_treatment'], df['after\_treatment'])

# Data storytelling with Plotly

fig = go.Figure()

# Add traces

fig.add\_trace(go.Scatter(x=df['patient\_id'], y=df['before\_treatment'], mode='markers', name='Before Treatment'))

fig.add\_trace(go.Scatter(x=df['patient\_id'], y=df['after\_treatment'], mode='markers', name='After Treatment'))

# Add confidence interval

fig.add\_trace(go.Scatter(

x=[0, 100], y=[confidence\_interval[0], confidence\_interval[0]],

mode='lines', name='Lower Confidence Interval', line=dict(dash='dash')

))

fig.add\_trace(go.Scatter(

x=[0, 100], y=[confidence\_interval[1], confidence\_interval[1]],

mode='lines', name='Upper Confidence Interval', line=dict(dash='dash')

))

# Add layout

fig.update\_layout(

title='Clinical Trial Results for Chisamba Pharmaceuticals',

xaxis\_title='Patient ID',

yaxis\_title='Measurement',

legend\_title='Legend'

)

# Show plot

fig.show()

# Print results

print(f"Mean Difference: {mean\_diff}")

print(f"Confidence Interval: {confidence\_interval}")

print(f"P-Value: {p\_value}")

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

The graph generated by the code above shows the measurements before and after treatment for each patient. The confidence interval lines indicate the range within which we expect the true mean difference to lie. The p-value helps us determine the statistical significance of the observed difference.

By visualizing the data, we can effectively communicate the impact of the new drug. The confidence intervals and p-values provide a statistical basis for our conclusions, making the analysis robust and credible.

Feel free to modify the dataset and parameters to fit your specific needs! If you have any questions or need further assistance, just let me know.

**Project Overview**

**Theory and Purpose:** The purpose of this project is to analyze the dose-response relationship in a clinical trial conducted by Chisamba Pharmaceuticals. Dose-response analysis helps in understanding the effect of different doses of a drug on patients. This is crucial for determining the optimal dose that maximizes efficacy while minimizing side effects.

**Expertise Acquired:**

* Understanding of regression analysis and dose-response relationships.
* Proficiency in data visualization using plotly.graph\_objects.
* Skills in data storytelling to effectively communicate findings.

**Sample Code**

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from sklearn.linear\_model import LinearRegression

# Fictitious dataset

data = {

'Dose': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Response': [5, 7, 8, 15, 20, 25, 30, 35, 40, 45, 50]

}

df = pd.DataFrame(data)

# Regression analysis

X = df[['Dose']]

y = df['Response']

model = LinearRegression()

model.fit(X, y)

df['Predicted\_Response'] = model.predict(X)

# Plotting the data

fig = go.Figure()

# Scatter plot of actual data

fig.add\_trace(go.Scatter(

x=df['Dose'], y=df['Response'],

mode='markers',

name='Actual Response'

))

# Line plot of predicted data

fig.add\_trace(go.Scatter(

x=df['Dose'], y=df['Predicted\_Response'],

mode='lines',

name='Predicted Response'

))

# Adding titles and labels

fig.update\_layout(

title='Dose-Response Analysis',

xaxis\_title='Dose',

yaxis\_title='Response',

legend\_title='Legend'

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

The graph above illustrates the relationship between the dose of a drug and the response observed in patients. The scatter plot represents the actual responses recorded during the clinical trial, while the line plot shows the predicted responses based on our regression model.

**Key Takeaways:**

* **Trend Identification:** The positive slope of the regression line indicates a positive correlation between dose and response, suggesting that higher doses lead to increased responses.
* **Optimal Dose:** By analyzing the graph, we can identify the dose at which the response begins to plateau, indicating the optimal dose for maximum efficacy.
* **Model Accuracy:** The closeness of the actual data points to the regression line demonstrates the model’s accuracy in predicting responses.

This project not only showcases the application of regression analysis in a pharmaceutical context but also emphasizes the importance of data visualization and storytelling in conveying complex information effectively.

**Project Overview**

**Theory and Purpose:** Dose-response analysis is crucial in pharmaceutical research to understand the relationship between drug dose and its effect on patients. This analysis helps in determining the optimal dose that maximizes efficacy while minimizing side effects.

**Expertise Acquired:**

* Understanding of dose-response relationships.
* Proficiency in regression analysis techniques.
* Skills in data visualization using Plotly.
* Ability to interpret and communicate statistical findings effectively.

**Sample Code**

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from scipy.optimize import curve\_fit

# Fictitious dataset

data = {

'Dose': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Response': [0, 1.2, 2.3, 3.1, 4.5, 5.1, 5.8, 6.2, 6.5, 6.8, 7.0]

}

df = pd.DataFrame(data)

# Define the dose-response function

def dose\_response(dose, a, b, c):

return a \* np.exp(-b \* dose) + c

# Fit the model

popt, pcov = curve\_fit(dose\_response, df['Dose'], df['Response'])

# Generate predictions

df['Predicted\_Response'] = dose\_response(df['Dose'], \*popt)

# Plotting the data

fig = go.Figure()

# Scatter plot of actual data

fig.add\_trace(go.Scatter(x=df['Dose'], y=df['Response'], mode='markers', name='Actual Response'))

# Line plot of predicted data

fig.add\_trace(go.Scatter(x=df['Dose'], y=df['Predicted\_Response'], mode='lines', name='Predicted Response'))

# Adding titles and labels

fig.update\_layout(

title='Dose-Response Analysis for Chisamba Pharmaceutical Company',

xaxis\_title='Dose',

yaxis\_title='Response',

legend\_title='Legend'

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

The graph above illustrates the relationship between the dose of a drug and the patient’s response. The scatter plot represents the actual responses observed in the clinical trial, while the line plot shows the predicted responses based on our regression model.

**Key Takeaways:**

* **Optimal Dose Identification:** The curve helps identify the dose at which the response plateaus, indicating the optimal dose for maximum efficacy.
* **Model Accuracy:** The close fit between the actual and predicted responses suggests that our model accurately captures the dose-response relationship.
* **Clinical Implications:** Understanding this relationship aids in making informed decisions about dosage recommendations, potentially improving patient outcomes and minimizing adverse effects.

**Theory and Purpose of ANOVA**

**Analysis of Variance (ANOVA)** is a statistical method used to test whether there are any statistically significant differences between the means of three or more independent groups. It helps in determining if the observed variations among group means are due to actual differences or just random chance.

**Purpose**:

* To compare the means of multiple groups.
* To understand if at least one group mean is different from the others.
* To control for Type I errors that can occur when multiple t-tests are conducted.

**Expertise Acquired and Takeaways**

By conducting an ANOVA analysis, you will:

* Gain expertise in statistical analysis and hypothesis testing.
* Learn to interpret ANOVA results and understand their implications.
* Develop skills in data visualization and storytelling using tools like Plotly.

**Sample Project Code**

Here’s a sample project code using Python and Plotly to perform ANOVA on a fictitious dataset from Chisamba Pharmaceutical Company:

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from scipy import stats

# Fictitious dataset

np.random.seed(42)

data = {

'Group': np.repeat(['Drug A', 'Drug B', 'Drug C'], 30),

'Response': np.concatenate([

np.random.normal(50, 10, 30),

np.random.normal(55, 10, 30),

np.random.normal(60, 10, 30)

])

}

df = pd.DataFrame(data)

# Perform ANOVA

anova\_result = stats.f\_oneway(

df[df['Group'] == 'Drug A']['Response'],

df[df['Group'] == 'Drug B']['Response'],

df[df['Group'] == 'Drug C']['Response']

)

# Print ANOVA result

print(f'F-statistic: {anova\_result.statistic}, p-value: {anova\_result.pvalue}')

# Data visualization using Plotly

fig = go.Figure()

for group in df['Group'].unique():

fig.add\_trace(go.Box(

y=df[df['Group'] == group]['Response'],

name=group

))

fig.update\_layout(

title='ANOVA Analysis of Drug Responses',

yaxis\_title='Response',

xaxis\_title='Drug Type'

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

In this project, we analyzed the response of three different drugs (Drug A, Drug B, and Drug C) on patients. The ANOVA test was conducted to determine if there are statistically significant differences between the mean responses of these drugs.

**Outcome**:

* The F-statistic and p-value from the ANOVA test indicate whether the differences between the group means are statistically significant.
* The box plot visualizes the distribution of responses for each drug, helping to identify any apparent differences.

By interpreting the ANOVA results and visualizing the data, we can effectively communicate the findings and make informed decisions about the efficacy of the drugs tested.

**Project Overview**

**Objective**: To test the relationship between two categorical variables in a clinical trial dataset using the Chi-Square test and visualize the results.

**Dataset**: A fictitious dataset of patients in a clinical trial, including variables such as treatment type and outcome.

**Theory and Purpose**

**Chi-Square Test**: The Chi-Square test is a statistical method used to determine if there is a significant association between two categorical variables. It compares the observed frequencies in each category to the frequencies expected if there were no association.

**Purpose**: In clinical trials, understanding the relationship between treatment types and patient outcomes is crucial for evaluating the effectiveness of treatments.

**Expertise Acquired**

* **Statistical Analysis**: Understanding and applying the Chi-Square test.
* **Data Visualization**: Using plotly.graph\_objects to create informative and interactive visualizations.
* **Data Storytelling**: Interpreting and presenting the results in a meaningful way.

**Sample Code**

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from scipy.stats import chi2\_contingency

# Create a fictitious dataset

data = {

'PatientID': range(1, 101),

'Treatment': np.random.choice(['Drug A', 'Drug B'], 100),

'Outcome': np.random.choice(['Improved', 'Not Improved'], 100)

}

df = pd.DataFrame(data)

# Create a contingency table

contingency\_table = pd.crosstab(df['Treatment'], df['Outcome'])

# Perform Chi-Square test

chi2, p, dof, expected = chi2\_contingency(contingency\_table)

# Print results

print(f"Chi-Square Statistic: {chi2}")

print(f"P-Value: {p}")

print(f"Degrees of Freedom: {dof}")

print("Expected Frequencies:")

print(expected)

# Visualize the results

fig = go.Figure(data=[

go.Bar(name='Observed', x=contingency\_table.columns, y=contingency\_table.loc['Drug A'], marker\_color='indianred'),

go.Bar(name='Expected', x=contingency\_table.columns, y=expected[0], marker\_color='lightsalmon')

])

fig.update\_layout(

title='Chi-Square Test Results for Treatment Outcomes',

xaxis\_title='Outcome',

yaxis\_title='Frequency',

barmode='group'

)

fig.show()

**Data Storytelling**

From the graph, we can observe the comparison between the observed and expected frequencies of treatment outcomes. If the observed frequencies significantly differ from the expected frequencies, it suggests a potential relationship between the treatment type and the outcome.

**Takeaways**

* **Statistical Insight**: The Chi-Square test helps determine if there is a significant association between treatment and outcome.
* **Visualization**: Using plotly.graph\_objects provides a clear and interactive way to present the data.
* **Data Interpretation**: Effective data storytelling involves not just presenting the data but also interpreting the results to provide meaningful insights.

Sure! Let’s break this down into several parts: theory, purpose, expertise, takeaways, and the sample project code using plotly.graph\_objects.

**Theory of Survival Analysis**

Survival analysis is a branch of statistics that deals with the analysis of time-to-event data. The primary objective is to estimate the time until an event of interest occurs, such as death, relapse, or failure. Key concepts include:

* **Survival Function**: Probability that the event has not occurred by a certain time.
* **Hazard Function**: Instantaneous rate at which events occur, given no prior event.

**Purpose of Survival Analysis in Clinical Trials**

In clinical trials, survival analysis helps in:

* **Evaluating Treatment Efficacy**: Comparing the time-to-event between treatment groups.
* **Understanding Risk Factors**: Identifying factors that influence the time-to-event.
* **Handling Censored Data**: Managing incomplete data due to patients dropping out or the study ending before the event occurs.

**Expertise Acquired**

By working on a survival analysis project, you will gain expertise in:

* **Statistical Methods**: Kaplan-Meier estimator, Cox proportional hazards model.
* **Data Handling**: Managing censored data, time-dependent covariates.
* **Visualization**: Creating informative plots to convey findings.

**Takeaways from the Project**

* **Analytical Skills**: Improved ability to analyze and interpret time-to-event data.
* **Technical Proficiency**: Enhanced coding skills in Python, especially using libraries like plotly.
* **Data Storytelling**: Ability to present complex data in an understandable and engaging manner.

**Sample Project Code**

Here’s a sample project code using a fictitious dataset for Chisamba Pharmaceutical Company:

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from lifelines import KaplanMeierFitter

# Fictitious dataset

data = {

'patient\_id': range(1, 101),

'time\_to\_event': np.random.exponential(scale=365, size=100), # Time in days

'event\_occurred': np.random.binomial(1, 0.7, size=100) # 1 if event occurred, 0 if censored

}

df = pd.DataFrame(data)

# Kaplan-Meier Fitter

kmf = KaplanMeierFitter()

kmf.fit(durations=df['time\_to\_event'], event\_observed=df['event\_occurred'])

# Plotting the survival function

fig = go.Figure()

fig.add\_trace(go.Scatter(

x=kmf.survival\_function\_.index,

y=kmf.survival\_function\_['KM\_estimate'],

mode='lines',

name='Survival Function'

))

fig.update\_layout(

title='Survival Analysis for Chisamba Pharmaceutical Company',

xaxis\_title='Time (days)',

yaxis\_title='Survival Probability',

template='plotly\_white'

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

The graph generated from the above code shows the survival probability over time for patients in the clinical trial. Key points to emphasize:

* **Survival Curve**: The curve represents the probability of patients surviving (not experiencing the event) over time.
* **Drop-offs**: Sudden drops in the curve indicate times when multiple events occurred.
* **Censoring**: The curve accounts for censored data, ensuring accurate representation of survival probabilities

**1. Theory and Purpose of Meta-Analysis in Pharmaceutical and Clinical Trials**

**Meta-analysis** is a statistical technique that combines the results of multiple scientific studies to identify patterns, discrepancies, or other interesting insights. In the context of pharmaceutical and clinical trials, it helps in:

* **Aggregating Evidence**: Combining data from various studies to provide a more robust conclusion.
* **Identifying Trends**: Detecting trends that might not be apparent in individual studies.
* **Improving Precision**: Increasing the statistical power by pooling data.
* **Guiding Decision-Making**: Informing clinical guidelines and policy decisions based on comprehensive evidence.

**2. Sample Project Code**

Here’s a sample Python project using plotly.graph\_objects for a meta-analysis of clinical trials data from the fictitious Chisamba Pharmaceutical Company:

**Python**

import pandas as pd

import plotly.graph\_objects as go

# Sample fictitious dataset

data = {

'Study': ['Study 1', 'Study 2', 'Study 3', 'Study 4', 'Study 5'],

'Effect Size': [0.2, 0.5, 0.3, 0.4, 0.6],

'CI Lower': [0.1, 0.3, 0.2, 0.2, 0.4],

'CI Upper': [0.3, 0.7, 0.4, 0.6, 0.8]

}

df = pd.DataFrame(data)

# Create a forest plot

fig = go.Figure()

for i, row in df.iterrows():

fig.add\_trace(go.Scatter(

x=[row['CI Lower'], row['CI Upper']],

y=[row['Study'], row['Study']],

mode='lines',

line=dict(color='blue', width=2)

))

fig.add\_trace(go.Scatter(

x=[row['Effect Size']],

y=[row['Study']],

mode='markers',

marker=dict(color='red', size=10)

))

fig.update\_layout(

title='Forest Plot of Clinical Trials',

xaxis\_title='Effect Size',

yaxis\_title='Study',

yaxis=dict(tickmode='array', tickvals=df['Study']),

showlegend=False

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**3. Data Storytelling and Interpretation**

The forest plot above visualizes the effect sizes and confidence intervals (CI) of different studies. Here’s how to interpret it:

* **Effect Size**: The red markers represent the effect size of each study. A higher effect size indicates a stronger effect.
* **Confidence Intervals**: The blue lines represent the confidence intervals. Narrower intervals suggest more precise estimates.
* **Overall Trend**: By looking at the distribution of effect sizes and their confidence intervals, we can infer the overall effectiveness of the treatment being studied.

**4. Takeaways from the Project**

* **Understanding Variability**: The project helps in understanding the variability and consistency across different studies.
* **Critical Analysis**: It emphasizes the importance of critically analyzing the confidence intervals and effect sizes.
* **Data Visualization**: Using plotly.graph\_objects enhances the storytelling aspect by providing clear and interactive visualizations.
* **Evidence-Based Decisions**: The insights gained from the meta-analysis can guide evidence-based decisions in clinical practice and policy-making.

**Project Overview**

**Title:** Bayesian Statistics in Clinical Trials for Chisamba Pharmaceuticals

**Objective:** To demonstrate the application of Bayesian statistics in clinical trials, incorporating prior knowledge and evidence to make informed decisions about the efficacy of a new drug.

**Theory and Purpose**

**Bayesian Statistics:** Bayesian statistics is a mathematical approach that uses probability to represent uncertainty in statistical models. It incorporates prior knowledge (prior distribution) and updates this knowledge with new evidence (likelihood) to form a posterior distribution.

**Purpose:** The purpose of this project is to showcase how Bayesian methods can be used in clinical trials to continuously update the probability of a drug’s efficacy as new data is collected. This approach allows for more flexible and informed decision-making compared to traditional frequentist methods.

**Expertise Acquired**

* Understanding of Bayesian theory and its application in clinical trials.
* Ability to incorporate prior knowledge and evidence into statistical models.
* Proficiency in using plotly.graph\_objects for data visualization and storytelling.

**Fictitious Dataset**

Let’s assume we have a dataset from a clinical trial testing the efficacy of a new drug. The dataset includes:

* patient\_id: Unique identifier for each patient.
* treatment: Binary variable indicating whether the patient received the treatment (1) or placebo (0).
* outcome: Binary variable indicating the outcome of the treatment (1 for success, 0 for failure).

**Sample Code**

**Python**

import pandas as pd

import numpy as np

import plotly.graph\_objects as go

from scipy.stats import beta

# Generate a fictitious dataset

np.random.seed(42)

data = pd.DataFrame({

'patient\_id': range(1, 101),

'treatment': np.random.binomial(1, 0.5, 100),

'outcome': np.random.binomial(1, 0.6, 100)

})

# Prior knowledge: Assume prior success rate is 0.5 with 10 prior trials

prior\_successes = 5

prior\_failures = 5

# Calculate posterior distribution parameters

successes = data[data['treatment'] == 1]['outcome'].sum()

failures = data[data['treatment'] == 1]['outcome'].count() - successes

posterior\_successes = prior\_successes + successes

posterior\_failures = prior\_failures + failures

# Plot the prior and posterior distributions

x = np.linspace(0, 1, 100)

prior\_dist = beta(prior\_successes, prior\_failures).pdf(x)

posterior\_dist = beta(posterior\_successes, posterior\_failures).pdf(x)

fig = go.Figure()

fig.add\_trace(go.Scatter(x=x, y=prior\_dist, mode='lines', name='Prior Distribution'))

fig.add\_trace(go.Scatter(x=x, y=posterior\_dist, mode='lines', name='Posterior Distribution'))

fig.update\_layout(

title='Prior and Posterior Distributions of Treatment Success Rate',

xaxis\_title='Success Rate',

yaxis\_title='Density'

)

fig.show()

**Explanation and Takeaways**

1. **Prior Distribution:** Represents our initial belief about the success rate of the treatment before seeing the data. In this case, we assumed a prior success rate of 0.5 based on 10 prior trials.
2. **Posterior Distribution:** Updates our belief about the success rate after incorporating the new data from the clinical trial. The posterior distribution is more informative and reflects the combined knowledge from prior information and new evidence.
3. **Data Storytelling:** The graph shows how our understanding of the treatment’s success rate evolves as we incorporate new data. The posterior distribution is narrower and more peaked, indicating increased confidence in the estimated success rate.

This project demonstrates the power of Bayesian statistics in clinical trials, allowing for continuous learning and more informed decision-making. The use of plotly.graph\_objects helps in effectively communicating the results and insights from the analysis.

**Theory and Purpose of Meta-Analysis**

**Meta-analysis** is a statistical technique that combines the results of multiple scientific studies. The main purposes are:

* **Aggregating Evidence**: Combining data from multiple studies to increase the overall sample size and statistical power.
* **Identifying Trends**: Detecting patterns or trends that may not be apparent in individual studies.
* **Improving Precision**: Providing more precise estimates of the effects of treatments or interventions.
* **Guiding Decision-Making**: Informing clinical and policy decisions based on a comprehensive analysis of available evidence.

**Expertise Acquired and Takeaways**

By conducting this meta-analysis project, you will gain expertise in:

* **Data Collection and Management**: Handling and organizing data from multiple sources.
* **Statistical Analysis**: Applying meta-analytic techniques to synthesize data.
* **Data Visualization**: Using plotly.graph\_objects to create informative and interactive visualizations.
* **Data Storytelling**: Communicating the results effectively to stakeholders.

**Sample Project Code**

Here’s a sample Python project code using a fictitious dataset:

**Python**

import pandas as pd

import plotly.graph\_objects as go

from scipy.stats import norm

# Fictitious dataset

data = {

'Study': ['Study 1', 'Study 2', 'Study 3', 'Study 4', 'Study 5'],

'Effect\_Size': [0.2, 0.5, 0.3, 0.4, 0.6],

'Standard\_Error': [0.1, 0.15, 0.1, 0.2, 0.15]

}

df = pd.DataFrame(data)

# Calculate weights

df['Weight'] = 1 / df['Standard\_Error']\*\*2

# Calculate combined effect size

combined\_effect\_size = (df['Effect\_Size'] \* df['Weight']).sum() / df['Weight'].sum()

# Calculate combined standard error

combined\_se = (1 / df['Weight'].sum())\*\*0.5

# Confidence interval

ci\_lower = combined\_effect\_size - 1.96 \* combined\_se

ci\_upper = combined\_effect\_size + 1.96 \* combined\_se

# Create plot

fig = go.Figure()

# Add individual studies

for i, row in df.iterrows():

fig.add\_trace(go.Scatter(

x=[row['Effect\_Size']],

y=[row['Study']],

mode='markers',

marker=dict(size=10),

name=row['Study']

))

# Add combined effect size

fig.add\_trace(go.Scatter(

x=[combined\_effect\_size],

y=[-1],

mode='markers',

marker=dict(size=15, color='red'),

name='Combined Effect Size'

))

# Add confidence interval

fig.add\_trace(go.Scatter(

x=[ci\_lower, ci\_upper],

y=[-1, -1],

mode='lines',

line=dict(color='red', width=2),

name='95% CI'

))

fig.update\_layout(

title='Meta-Analysis of Clinical Trials',

xaxis\_title='Effect Size',

yaxis\_title='Study',

yaxis=dict(tickvals=list(range(-1, len(df))), ticktext=['Combined'] + df['Study'].tolist())

)

fig.show()

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

**Data Storytelling**

In this project, we aggregated data from five fictitious clinical trials to estimate the overall effect size of a treatment. The combined effect size is represented by the red marker, and the confidence interval is shown as a red line. This visualization helps in understanding the overall effectiveness of the treatment across different studies.

By synthesizing the data, we can make more informed decisions about the treatment’s efficacy, identify trends across studies, and improve the precision of our estimates. This approach is crucial for guiding clinical and policy decisions in the pharmaceutical industry.