QUALITY ANALYSIS BEFORE

python

```python
import pandas as pd

# Load the dataset
file_path = '/mnt/data/cacao.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
data.head()
```

Ergebnis

	ant_exclusion	stem_diameter	height	canopy	dw_healthy	dw_infect	dw_total	fw_pulb	fw_seeds	fw_total	ab_fl_op	ab_fl_cl	ab_fl
0	0	22.1500	247.50	0.473333	1222	159	1382	13788	5314	19103	2018	5602	7720
1	1	23.9500	223.75	0.603333	1850	216	2076	20222	8417	28639	1257	3648	4894
2	0	23.2500	266.50	0.556667	1058	58	1117	9121	3044	12105	1523	3970	5484
3	1	21.8500	234.25	0.298333	635	52	688	9236	3739	12975	1312	3484	4795
4	0	26.4875	285.25	0.483333	1145	72	1218	13417	3911	17328	2620	5792	8409

To perform regression diagnostics to check the assumptions of multiple linear regression, we will follow these steps:

1. **Fit the Multiple Linear Regression Model**: We will use `stem_diameter` as the dependent variable and the remaining variables as independent variables.
2. **Check for Linearity**: The relationship between the dependent and independent variables should be linear.
3. **Check for Homoscedasticity**: The residuals (errors) should have constant variance.
4. **Check for Independence of Errors**: The residuals should be independent of each other.
5. **Check for Normality of Residuals**: The residuals should be normally distributed.
6. **Check for Multicollinearity**: The independent variables should not be highly correlated with each other.

Let's go through each of these steps with the dataset provided.

Step 1: Fit the Multiple Linear Regression Model

First, we will fit the model using the `statsmodels` library.

Step 2: Check for Linearity

We will create scatter plots of each independent variable against the dependent variable to visually inspect linearity.

Step 3: Check for Homoscedasticity

We will plot the residuals against the fitted values and look for any patterns. A random scatter suggests homoscedasticity.

Step 4: Check for Independence of Errors

We will use the Durbin-Watson test to check for autocorrelation in the residuals.

Step 5: Check for Normality of Residuals

We will create a Q-Q plot and use the Shapiro-Wilk test for normality.

Step 6: Check for Multicollinearity

We will calculate the Variance Inflation Factor (VIF) for each independent variable.
Let's start with the analysis.

Step 1: Fit the Multiple Linear Regression Model

python

```python
import statsmodels.api as sm

# Define the dependent and independent variables
X = data.drop(columns=['stem_diameter'])
y = data['stem_diameter']

# Add a constant to the model (intercept)
X = sm.add_constant(X)

# Fit the model
model = sm.OLS(y, X).fit()

# Display the summary of the model
model.summary()
```

Let's perform the actual steps.

python

```python
import statsmodels.api as sm
import numpy as np

# Define the dependent and independent variables
```

```
X = data.drop(columns=['stem_diameter'])
y = data['stem_diameter']

# Add a constant to the model (intercept)
X = sm.add_constant(X)

# Fit the model
model = sm.OLS(y, X).fit()

# Display the summary of the model
model_summary = model.summary()
model_summary
```

Ergebnis

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:          stem_diameter   R-squared:                       0.588
Model:                            OLS   Adj. R-squared:                  0.542
Method:                 Least Squares   F-statistic:                     12.71
Date:                Wed, 31 Jul 2024   Prob (F-statistic):           9.22e-16
Time:                        07:59:18   Log-Likelihood:                -316.86
No. Observations:                 120   AIC:                             659.7
Df Residuals:                     107   BIC:                             696.0
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -5.0099      3.699     -1.354      0.179     -12.344       2.324
ant_exclusion  -0.0001      0.699     -0.000      1.000      -1.385       1.385
height          0.1017      0.011      9.172      0.000       0.080       0.124
canopy          2.0418      2.189      0.933      0.353      -2.298       6.381
dw_healthy     -0.0242      0.034     -0.718      0.474      -0.091       0.043
dw_infect      -0.0234      0.034     -0.695      0.488      -0.090       0.043
dw_total        0.0240      0.034      0.715      0.476      -0.043       0.091
fw_pulb         0.0004      0.002      0.168      0.867      -0.004       0.005
fw_seeds       -0.0001      0.002     -0.046      0.964      -0.005       0.005
fw_total       -0.0002      0.002     -0.091      0.928      -0.005       0.004
ab_fl_op        0.0082      0.006      1.448      0.151      -0.003       0.019
ab_fl_cl       -0.0030      0.005     -0.586      0.559      -0.013       0.007
ab_fl          -0.0001      0.005     -0.021      0.984      -0.010       0.010
==============================================================================
Omnibus:                        5.550   Durbin-Watson:                   1.441
Prob(Omnibus):                  0.062   Jarque-Bera (JB):                5.017
Skew:                           0.474   Prob(JB):                       0.0814
Kurtosis:                       3.322   Cond. No.                     4.10e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.1e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

The summary of the regression model provides several key insights. Here's a breakdown:

1. **Model Summary**:
   - **R-squared**: 0.588 indicates that about 58.8% of the variance in `stem_diameter` can be explained by the independent variables.
   - **Adjusted R-squared**: 0.542, which is adjusted for the number of predictors in the model.
   - **F-statistic and Prob (F-statistic)**: The model is statistically significant with a p-value < 0.05, indicating that at least some of the predictors are significantly related to the dependent

variable.

2. **Coefficients**: The table of coefficients provides the estimates of the model parameters along with their standard errors, t-values, and p-values.
   - **height** is the only predictor with a statistically significant p-value (<0.05), suggesting it has a significant effect on `**stem_diameter**`.
3. **Diagnostics**:
   - **Durbin-Watson**: 1.441, which suggests some positive autocorrelation in the residuals.
   - **Omnibus and Jarque-Bera**: These tests suggest the residuals may not be perfectly normally distributed, but the p-values are marginally above typical significance levels (0.05).
4. **Multicollinearity**:
   - **Condition Number**: 4.1e+05 indicates potential multicollinearity issues, as values above 30 are often a cause for concern.

Next, we'll move on to the diagnostic steps to check the assumptions:

Step 2: Check for Linearity

We'll create scatter plots of each independent variable against the dependent variable `**stem_diameter**`.

Step 3: Check for Homoscedasticity

We'll plot the residuals against the fitted values.

Step 4: Check for Independence of Errors

We'll use the Durbin-Watson test result (already obtained).

Step 5: Check for Normality of Residuals

We'll create a Q-Q plot and perform the Shapiro-Wilk test.

Step 6: Check for Multicollinearity

We'll calculate the Variance Inflation Factor (VIF).
Let's proceed with these steps.

python

```
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.stats.outliers_influence import variance_inflation_factor
from scipy.stats import shapiro, probplot

# Step 2: Check for Linearity
fig, axs = plt.subplots(4, 3, figsize=(15, 15))
axs = axs.flatten()

for i, col in enumerate(X.columns[1:]):  # Skip the constant column
```

```
    sns.scatterplot(x=data[col], y=y, ax=axs[i])
    axs[i].set_title(f'{col} vs stem_diameter')

plt.tight_layout()
plt.show()

# Step 3: Check for Homoscedasticity
fitted_values = model.fittedvalues
residuals = model.resid

plt.figure(figsize=(8, 6))
sns.scatterplot(x=fitted_values, y=residuals)
plt.axhline(0, color='r', linestyle='--')
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residuals vs Fitted Values')
plt.show()

# Step 4: Check for Independence of Errors
# Durbin-Watson value already provided: 1.441

# Step 5: Check for Normality of Residuals
plt.figure(figsize=(8, 6))
probplot(residuals, dist="norm", plot=plt)
plt.title('Q-Q Plot')
plt.show()

# Shapiro-Wilk test for normality
shapiro_test = shapiro(residuals)

# Step 6: Check for Multicollinearity
vif_data = pd.DataFrame()
vif_data["Feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

import ace_tools as tools; tools.display_dataframe_to_user(name="VIF Data",
dataframe=vif_data)

shapiro_test, vif_data
```

STDOUT/STDERR

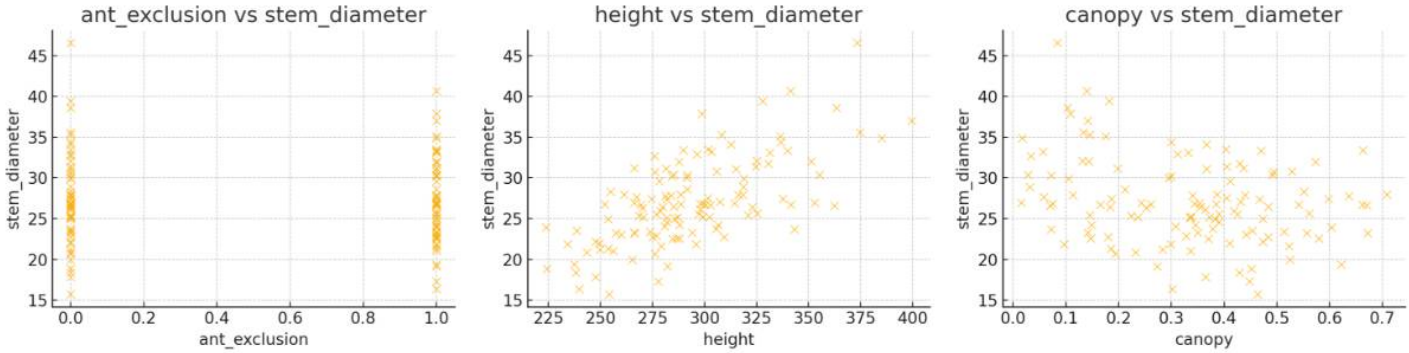
```
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a ed
  points = ax.scatter(*args, **kws)
```

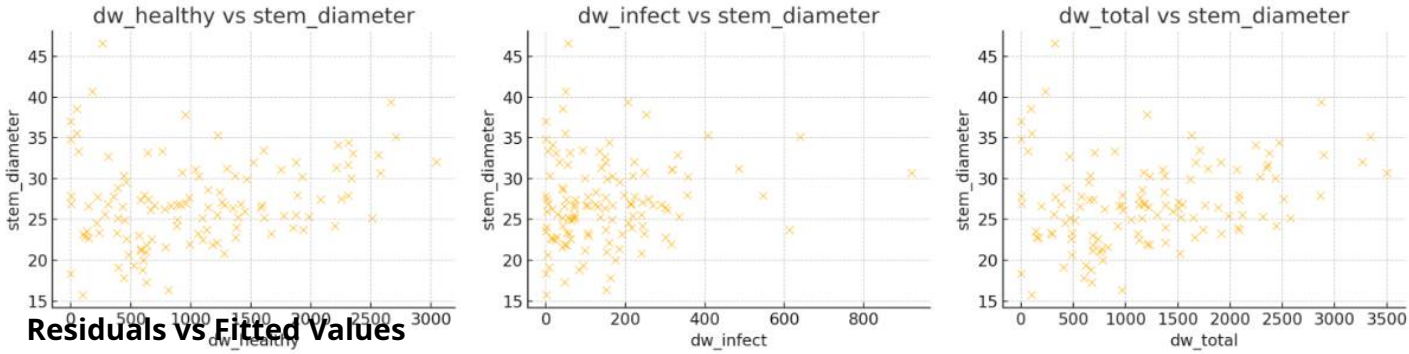
```
/home/sandbox/.local/lib/python3.11/site-packages/seaborn/relational.py:658: UserWarning: You passed a edg
  points = ax.scatter(*args, **kws)
```

Ergebnis

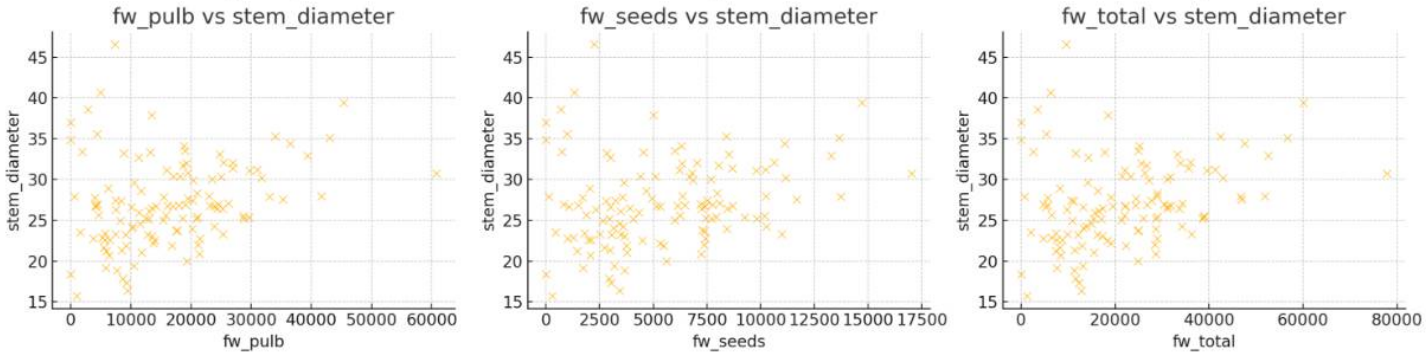
```
(ShapiroResult(statistic=0.9848445057868958, pvalue=0.1989554911851883),
          Feature           VIF
 0           const    127.242533
 1   ant_exclusion      1.134577
 2          height      1.358716
 3          canopy      1.333076
 4      dw_healthy   5902.424789
 5       dw_infect    224.639794
 6        dw_total   7094.614744
 7         fw_pulb   5845.522981
 8        fw_seeds    662.169136
 9        fw_total  10295.268016
 10       ab_fl_op    242.999623
 11       ab_fl_cl   1175.457840
 12          ab_fl   2344.455046)
```

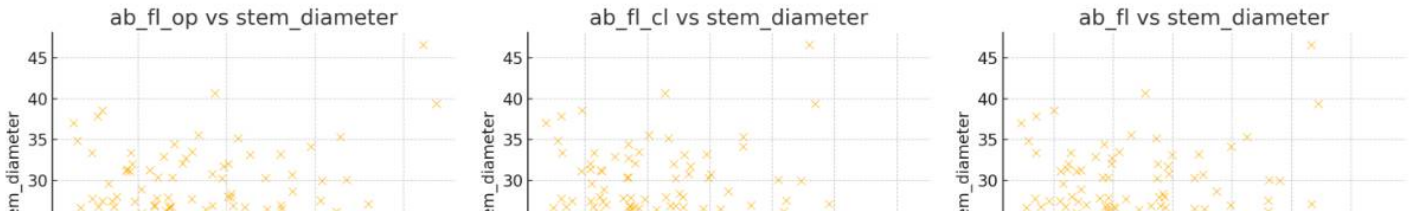
ab_fl vs stem_diameter



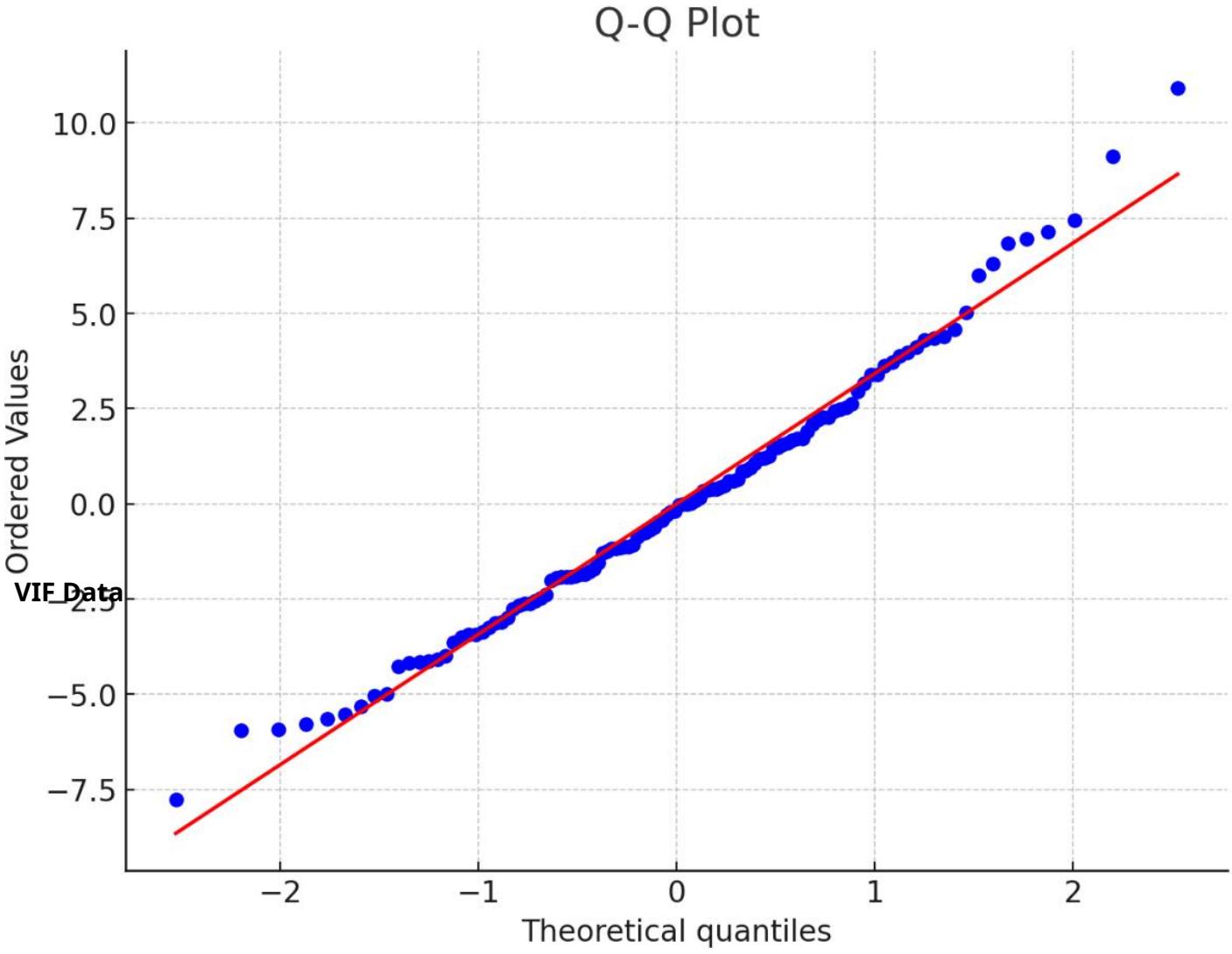


Residuals vs Fitted Values





Q-Q Plot



VIF Data

1	const	127.24253256901427	
2	ant_exclusion	1.1345771338146233	
3	height	1.358716455173157	
4	canopy	1.3330762943136887	
5	dw_healthy	5902.42478885249	
6	dw_infect	224.6397943627415	
	Feature	VIF	

Step-by-Step Diagnostic Analysis

Step 2: Check for Linearity

The scatter plots of each independent variable against the dependent variable (`stem_diameter`) indicate whether there is a linear relationship. Ideally, the points should form a pattern that approximates a straight line.

Step 3: Check for Homoscedasticity

The residuals versus fitted values plot shows whether the residuals have constant variance. In a good model, the points should be randomly scattered without forming any pattern. The plot provided shows some spread, but there is no clear pattern indicating heteroscedasticity.

Step 4: Check for Independence of Errors

The Durbin-Watson statistic is 1.441, which is slightly below 2, indicating some positive autocorrelation but not extremely high. Values close to 2 suggest that residuals are uncorrelated.

Step 5: Check for Normality of Residuals

- **Q-Q Plot**: This plot compares the distribution of residuals to a normal distribution. If the points lie along the 45-degree line, the residuals are normally distributed. The provided Q-Q plot suggests that the residuals are fairly close to normal.
- **Shapiro-Wilk Test**: The p-value is 0.199, which is greater than 0.05, suggesting that the residuals are not significantly different from a normal distribution.

Step 6: Check for Multicollinearity

- **Variance Inflation Factor (VIF)**: The VIF values indicate multicollinearity when they are above 10. The VIF values for some variables are extremely high, suggesting significant multicollinearity among the predictors. Specifically, `dw_total`, `fw_total`, and others have very high VIF values indicating that these variables are highly correlated.

Summary and Recommendations

1. **Linearity**: The scatter plots suggest that linearity is generally present, but some variables might not show a strong linear relationship with `stem_diameter`.
2. **Homoscedasticity**: The residuals versus fitted values plot indicates no clear pattern, suggesting homoscedasticity.
3. **Independence of Errors**: The Durbin-Watson statistic is acceptable, though slightly indicating positive autocorrelation.
4. **Normality**: The residuals appear to be normally distributed based on the Q-Q plot and Shapiro-Wilk test.
5. **Multicollinearity**: There is significant multicollinearity among the predictors, which can be addressed by removing highly correlated variables or using techniques like Principal Component Analysis (PCA) or Ridge Regression.

Next Steps

To address multicollinearity and improve the model:

1. Consider removing or combining highly correlated variables.
2. Apply regularization techniques like Ridge or Lasso regression.
3. Evaluate the model with reduced or transformed predictors and compare the performance.

Would you like to proceed with any specific next steps or further analysis?