Question 2 – Quality Differentiated Coffee Markets

In this question, you will explore how futures prices for raw agricultural commodities influence price inflation for processed products. Specifically, you will analyze how the futures prices of Arabica and Robusta coffee beans affect the CPI for roasted ground coffee. Over the past year, retail coffee prices and futures prices for Arabica and Robusta have surged.

The basic theoretical framework can be described as follows:

High-quality coffee roasters purchase Arabica coffee beans at the world price, which we will assume is the futures price, F_t^A.

Let m_t^A denote the sum of 1) unit processing costs, 2) wholesale marketing margins, and 3) retail marketing margins for Arabica coffee. The period t retail price of coffee produced from Arabica beans can be expressed as

$$P_t^A = m_t^A + F_t^A.$$

Similarly, the period t retail price of coffee produced from Robusta beans can be expressed as

$$P_t^R = m_t^R + F_t^R.$$

Assume the consumption quantities for each coffee type are fixed. The CPI is a Laspeyres index (a fixed-weight price index) that measures the percentage change in spending on a fixed basket of goods relative to a base period $(t = 0)$. With fixed amounts of Arabica and Robusta coffee, the CPI for coffee in period t can be written as

$$CPI_t \;=\; \frac{F_t^A Q^A + F_t^R Q^R + m_t^A Q^A + m_t^R Q^R}{F_0^A Q^A + F_0^R Q^R + m_0^A Q^A + m_0^R Q^R}.$$

Let's divide the numerator and the denominator by $Q^A + Q^R$ and express the CPI in terms of expenditure shares,

$$S^i \;=\; \frac{Q^i}{Q^A + Q^R}.$$

Then

$$CPI_t \;=\; \frac{F_t^A S^A + F_t^R S^R + m_t^A S^A + m_t^R S^R}{F_0^A S^A + F_0^R S^R + m_0^A S^A + m_0^R S^R}.$$

To further simplify this expression, let's make the strong assumption that the weighted-average coffee processing margin, $m_t^A S^A + m_t^R S^R$, inflates at the same rate as overall CPI

for food. With this assumption, the previous equation can be rewritten as

$$CPI_t = \frac{F_t^A S^A + F_t^R S^R + CPI_t^{\text{Food}}}{F_0^A S^A + F_0^R S^R + CPI_0^{\text{Food}}}.$$

Finally, to conserve on notation let $z_0 = F_0^A S^A + F_0^R S^R + CPI_0^{\text{Food}}$. The coffee CPI expression can now be rewritten as

$$CPI_t = \frac{S^A}{z_0} F_t^A + \frac{S^R}{z_0} F_t^R + \frac{1}{z_0} CPI_t^{\text{Food}}.$$

By adding an intercept and an error term, we can regress the coffee CPI on the two coffee futures prices and the overall food CPI. The ratio of the coefficients on the F_t^A and F_t^R variables should reflect the relative shares of Arabica and Robusta beans. While this regression does not directly measure how shocks in futures prices pass through to retail coffee prices, statistically significant positive coefficients on the futures variables would indicate that weather and other shocks that affect coffee bean prices do have an impact on retail prices. Although this may seem intuitive, it is not uncommon for pass-through effects to be too small to detect (e.g., tomato prices and pasta sauce).

In [ ]:
```
# install & import libraries
!pip install pandas numpy matplotlib

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

Part A: Data Preparation and Visualization

In [2]:
```
# import the two price series.
arabica_data = pd.read_csv('data/kcz25_daily-nearby_historical-data-09-13-2025.c
robusta_data = pd.read_csv('data/rmx25_daily-nearby_historical-data-09-13-2025.c
```

In [3]:
```
# create time columns for filtering and aggregating later
arabica_data['Time'] = pd.to_datetime(arabica_data['Time'])
robusta_data['Time'] = pd.to_datetime(robusta_data['Time'])

# compute month and year column for aggregation later with CPI
arabica_data['Month'] = arabica_data['Time'].dt.month
arabica_data['Year'] = arabica_data['Time'].dt.year
robusta_data['Month'] = robusta_data['Time'].dt.month
robusta_data['Year'] = robusta_data['Time'].dt.year
```

In [4]:
```
# checking data structure
print("Arabica:")
print(arabica_data.head(1))
```

```
print("\nRobusta:")
print(robusta_data.head(1))
```

```
Arabica:
  Symbol       Time    Open   High    Low   Last  Change    %Chg  Volume  \
0  KCH00 2000-01-03  122.25  124.0  116.1  116.5    -9.4  -7.47%    1296

   Open Int  Month  Year
0     43996      1  2000

Robusta:
  Symbol       Time  Open  High   Low  Last  Change   %Chg  Volume  Open Int  \
0  RMH09 2008-08-01  2250  2291  2250  2279      19  0.84%     532      1362

   Month  Year
0      8  2008
```

In [5]:
```
# load CPI files
coffee_CPI = pd.read_csv('data/coffee_CPI.csv')
food_CPI = pd.read_csv('data/food_CPI.csv')

# rename columns
coffee_CPI = coffee_CPI.rename(columns={"CUSR0000SEFP01": "coffee_CPI"})
food_CPI = food_CPI.rename(columns={"CPIUFDSL": "food_CPI"})

# formtting date columns
coffee_CPI['Date'] = pd.to_datetime(coffee_CPI['observation_date'])
food_CPI['Date'] = pd.to_datetime(food_CPI['observation_date'])

# compute month and year column for aggregation
coffee_CPI['Month'] = coffee_CPI['Date'].dt.month
coffee_CPI['Year'] = coffee_CPI['Date'].dt.year
food_CPI['Month'] = food_CPI['Date'].dt.month
food_CPI['Year'] = food_CPI['Date'].dt.year
```

In [6]:
```
# merge coffee with the two CPI dataframes
# arabica
coffee_arabica = pd.merge(
    arabica_data,
    coffee_CPI[["Year", "Month", "coffee_CPI"]],
    on=["Year", "Month"],
    how="left")

all_arabica = pd.merge(
    coffee_arabica,
    food_CPI[["Year", "Month", "food_CPI"]],
    on=["Year", "Month"],
    how="left")

# robusta
coffee_robusta = pd.merge(
    robusta_data,
    coffee_CPI[["Year", "Month", "coffee_CPI"]],
    on=["Year", "Month"],
    how="left")
```

```
all_robusta = pd.merge(
    coffee_robusta,
    food_CPI[["Year", "Month", "food_CPI"]],
    on=["Year", "Month"],
    how="left")
```

In [7]:
```
# drop unused columns.
all_arabica = all_arabica.drop(columns=["Year", "Month", "Symbol", "Open", "Low"
all_robusta = all_robusta.drop(columns=["Year", "Month", "Symbol", "Open", "Low"
```

In [8]:
```
print(all_arabica.head(2))
print(all_robusta.head(2))
```

```
        Time    Last  coffee_CPI  food_CPI
0 2000-01-03  116.50         NaN     165.6
1 2000-01-04  116.25         NaN     165.6
        Time  Last  coffee_CPI  food_CPI
0 2008-08-01  2279     191.224   216.528
1 2008-08-04  2196     191.224   216.528
```

The Arabica data has missing coffee_CPI data, but we will address this in the next step.

1. Plot the coffee CPI and the overall food CPI on the same chart. Scale one series so both start at the same level. Comment briefly on whether coffee inflation has been faster or slower than overall food inflation since Jan-2020.

In [9]:
```
# if coffee_CPI data is not available, drop the row
all_arabica = all_arabica.dropna(subset=["coffee_CPI"])
all_robusta = all_robusta.dropna(subset=["coffee_CPI"])

# filter for after January 2020
arabica_2020 = all_arabica.loc[all_arabica["Time"] >= "2020-01-01"].reset_index(
robusta_2020 = all_robusta.loc[all_robusta["Time"] >= "2020-01-01"].reset_index(
```

In [10]:
```
print("before filtering:", all_arabica.shape)
print("after filtering:", arabica_2020.shape)
```

```
before filtering: (5695, 4)
after filtering: (1424, 4)
```

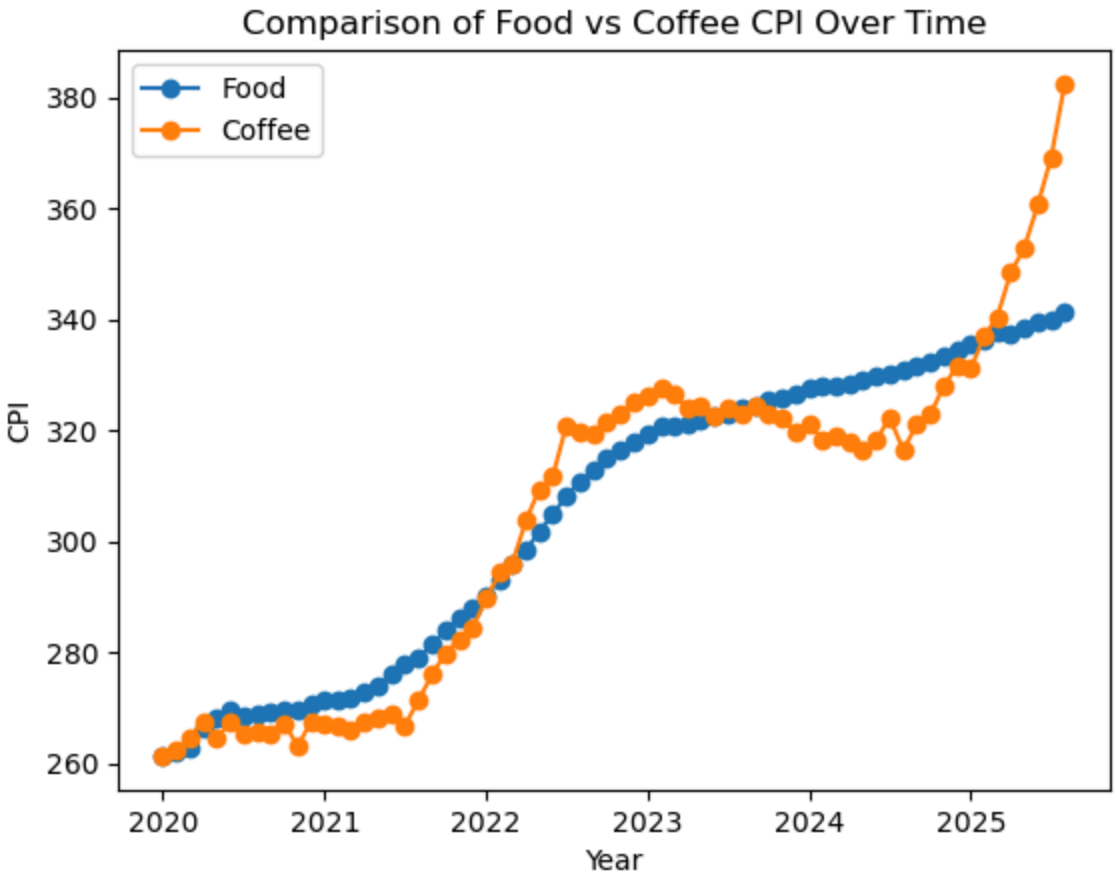
Plotting the coffee CPI and the overall food CPI.

In [11]:
```
# filter for after January 2020 for food and coffee CPI
food_2020 = food_CPI.loc[food_CPI["Date"] >= "2020-01-01"].reset_index(drop=True
coffee_2020 = coffee_CPI.loc[coffee_CPI["Date"] >= "2020-01-01"].reset_index(dro

# calculate coffee multiplier to have the same starting value
food_base = food_2020.loc[0, "food_CPI"]
coffee_base = coffee_2020.loc[0, "coffee_CPI"]

coffee_scale = food_base/coffee_base

print("Coffee Scale =", coffee_scale)
print("Food Base =",food_base)
```

```
# test
print("Coffee Base after Scaling =",coffee_base * coffee_scale)
```

```
Coffee Scale = 1.3651671891327064
Food Base = 261.293
Coffee Base after Scaling = 261.293
```

In [12]:
```
# plot

plt.figure()

# date on x, CPI on y
plt.plot(food_2020['Date'], food_2020['food_CPI'], label='Food',  marker='o')
plt.plot(coffee_2020['Date'], (coffee_2020['coffee_CPI']* coffee_scale), label='
plt.xlabel('Year')
plt.ylabel('CPI')
plt.title('Comparison of Food vs Coffee CPI Over Time')
plt.legend()
plt.show()
```



From 2020 to 2023, coffee has generally followed the same inflation trends as food overall. Between mid-2023 and mid-2024, coffee inflation slowed relative to food, briefly dipping below the overall food trend. However, from late 2024 into 2025, coffee CPI accelerated sharply, rising much faster than food CPI. This sharp acceleration likely reflects the severe drought in Brazil mentioned in the premise of the assignment.

2. Compute monthly inflation rates for Arabica futures and for the coffee CPI. Produce a scatterplot of coffee-CPI inflation (y-axis) against Arabica monthly inflation (x-axis).

In [13]:
```
arabica_2020.head()
```

Out[13]:

	Time	Last	coffee_CPI	food_CPI
0	2020-01-02	127.10	191.4	261.293
1	2020-01-03	126.35	191.4	261.293
2	2020-01-06	122.15	191.4	261.293
3	2020-01-07	122.40	191.4	261.293
4	2020-01-08	119.15	191.4	261.293

In [16]:
```
# set up arabica_monthly
arabica_monthly = all_arabica.groupby(['Time']).agg({
    'Last': 'mean',
    'coffee_CPI': 'first'  # CPI is monthly data, so just take first value
}).reset_index()

# get year-month for grouping (simpler / alternative approach than the last sect
arabica_monthly['Year_Month'] = arabica_2020['Time'].dt.to_period('M')

# group by month and take mean of daily prices within each month
arabica_monthly_final = arabica_monthly.groupby('Year_Month').agg({
    'Last': 'mean',
    'coffee_CPI': 'first'
}).reset_index()

print("Arabica monthly data:")
print(arabica_monthly_final.head())
```

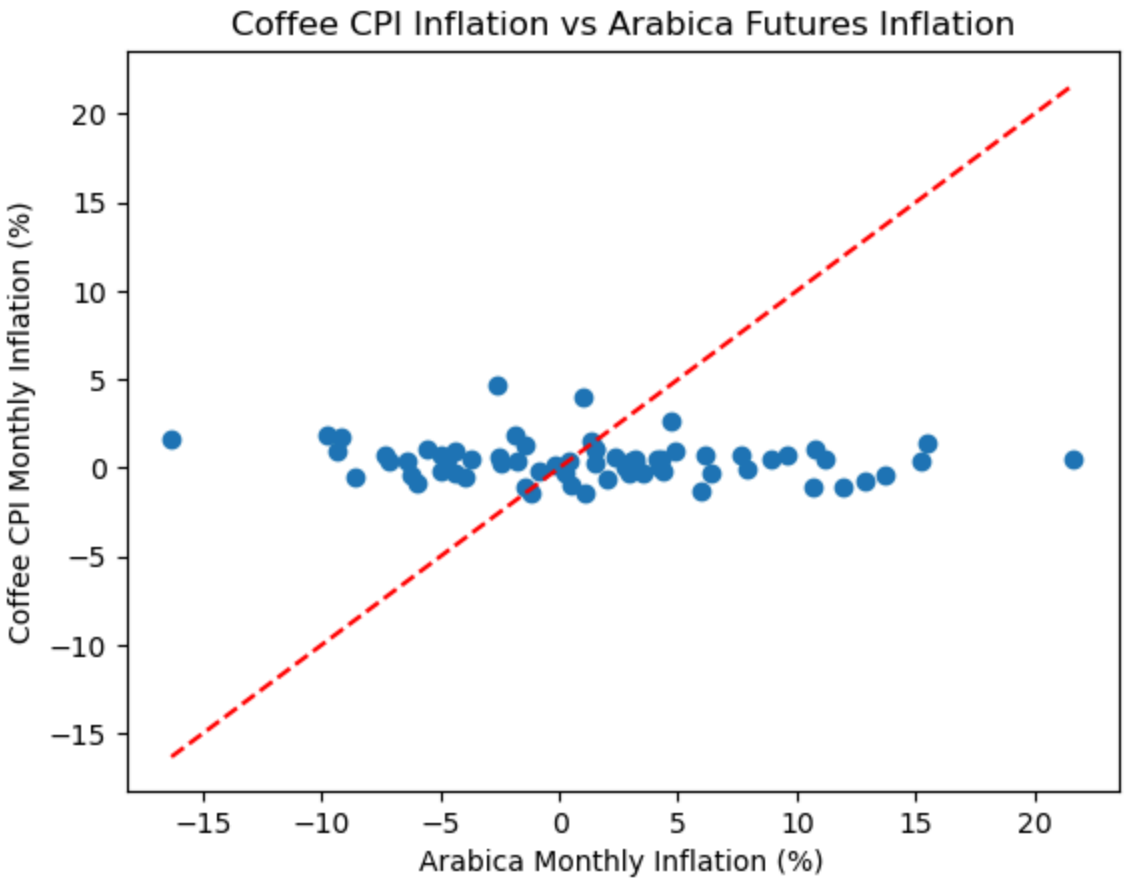
```
Arabica monthly data:
  Year_Month       Last  coffee_CPI
0    2020-01  64.871429       145.3
1    2020-02  63.944737       143.8
2    2020-03  59.390909       144.4
3    2020-04  63.923810       145.5
4    2020-05  65.212500       144.6
```

In [17]:
```
# calculate monthly inflation rates
# inflation rate = (current month price - previous month price) / previous month
# .pct_change() calculates % changes rowwise in sequence

arabica_monthly_final['arabica_inflation'] = arabica_monthly_final['Last'].pct_c
arabica_monthly_final['coffee_cpi_inflation'] = arabica_monthly_final['coffee_CP

# drop first row since it will have NaN inflation rates
arabica_inflation_data = arabica_monthly_final.dropna()
```

```
print("Monthly inflation data:")
print(arabica_inflation_data.head())
```

```
Monthly inflation data:
  Year_Month       Last  coffee_CPI  arabica_inflation  coffee_cpi_inflation
1    2020-02  63.944737       143.8          -1.428505             -1.032347
2    2020-03  59.390909       144.4          -7.121505              0.417246
3    2020-04  63.923810       145.5           7.632314              0.761773
4    2020-05  65.212500       144.6           2.015979             -0.618557
5    2020-06  60.447727       145.7          -7.306533              0.760719
```

In [19]:
```
# scatterplot
plt.figure()
plt.scatter(arabica_inflation_data['arabica_inflation'],
            arabica_inflation_data['coffee_cpi_inflation'])

# add 45-degree line for reference
x_min = arabica_inflation_data['arabica_inflation'].min()
x_max = arabica_inflation_data['arabica_inflation'].max()
y_min = arabica_inflation_data['coffee_cpi_inflation'].min()
y_max = arabica_inflation_data['coffee_cpi_inflation'].max()

# use the range that covers both axes
plot_min = min(x_min, y_min)
plot_max = max(x_max, y_max)

plt.plot([plot_min, plot_max], [plot_min, plot_max], 'r--')

plt.xlabel('Arabica Monthly Inflation (%)')
plt.ylabel('Coffee CPI Monthly Inflation (%)')
plt.title('Coffee CPI Inflation vs Arabica Futures Inflation')
plt.show()
```



If bean-market supply shocks were transmitted instantly and completely to retail coffee prices, the points would lie tightly around the 45° line. Using that benchmark, what do you conclude about the short-run pass-through from bean futures to retail coffee prices?

The scatterplot shows that Arabica monthly inflation and retail coffee CPI inflation do not align at all along the 45° line. This shows that short-run pass-through from Arabica futures to retail coffee prices is weak. There are a few possible explanations for this behavior:

1. **Hedging:** Retailers and processors may protect themselves from short-term bean price volatility by using futures contracts.
2. **Cost structure:** Coffee beans represent only a fraction of the total retail price. Processing, distribution, packaging, and marketing costs likely dominate, limiting the influence of bean price volatility.

Part B: Regressions

Commodity prices are often non-normal, so it's common to take logs before running regressions. Logging the variables also makes the coefficients directly interpretable as elasticities. Using this approach, regress the log of the coffee CPI on the logs of Arabica and Robusta futures prices and on the log of the overall food CPI.

In [20]:
```
# merge all data

all_arabica = all_arabica.rename(columns={"Last": "arabica_last"})
all_robusta = all_robusta.rename(columns={"Last": "robusta_last"})

merged = pd.merge(
    all_arabica,
    all_robusta[["Time", "robusta_last"]],
    on="Time",
    how="inner"
)

merged.head()
```

Out[20]:

	Time	arabica_last	coffee_CPI	food_CPI	robusta_last
0	2008-08-01	140.15	191.224	216.528	2279
1	2008-08-04	136.85	191.224	216.528	2196
2	2008-08-05	140.20	191.224	216.528	2280
3	2008-08-06	137.90	191.224	216.528	2260
4	2008-08-07	144.30	191.224	216.528	2280

In [21]:
```
# log transform
reg = merged.copy()
reg["log_coffee_CPI"] = np.log(merged["coffee_CPI"])
reg["log_food_CPI"]   = np.log(merged["food_CPI"])
reg["log_arabica"]    = np.log(merged["arabica_last"])
reg["log_robusta"]    = np.log(merged["robusta_last"])

reg.head()
```

Out[21]:

	Time	arabica_last	coffee_CPI	food_CPI	robusta_last	log_coffee_CPI	log_food_CPI
0	2008-08-01	140.15	191.224	216.528	2279	5.253446	5.37772
1	2008-08-04	136.85	191.224	216.528	2196	5.253446	5.37772
2	2008-08-05	140.20	191.224	216.528	2280	5.253446	5.37772
3	2008-08-06	137.90	191.224	216.528	2260	5.253446	5.37772
4	2008-08-07	144.30	191.224	216.528	2280	5.253446	5.37772

In [22]:
```
# coffee_CPI is the independent variable regressing onto arabica, robusta, and o
```

```
# regression controlling for heteroskedasticity
model = smf.ols("log_coffee_CPI ~ log_arabica + log_robusta + log_food_CPI", dat
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         log_coffee_CPI   R-squared:                       0.786
Model:                            OLS   Adj. R-squared:                  0.786
Method:                 Least Squares   F-statistic:                     4733.
Date:                Sat, 27 Sep 2025   Prob (F-statistic):               0.00
Time:                        02:12:46   Log-Likelihood:                 7571.5
No. Observations:                4235   AIC:                        -1.513e+04
Df Residuals:                    4231   BIC:                        -1.511e+04
Df Model:                           3
Covariance Type:                  HC1
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.6303      0.029     89.350      0.000       2.573       2.688
log_arabica    0.0871      0.004     23.443      0.000       0.080       0.094
log_robusta    0.0586      0.004     15.395      0.000       0.051       0.066
log_food_CPI   0.3272      0.007     49.453      0.000       0.314       0.340
==============================================================================
Omnibus:                      184.597   Durbin-Watson:                   0.007
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              159.523
Skew:                           0.408   Prob(JB):                     2.29e-35
Kurtosis:                       2.511   Cond. No.                         456.
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC1)
```

1) What does the relative size and significance of the Arabica and Robusta coefficients suggest about their respective weights in the coffee CPI?

Since we took the log of arabica and robusta prices, the coefficients reflect the elasticities of the two coffee types on the coffee CPI. The coefficients on Arabica (0.0871) and Robusta (0.0586) are both positive and significant, which shows that shocks in futures prices for both markets passes through to retail coffee prices.

The larger Arabica coefficient suggests that Arabica plays a stronger role in determining retail price changes than Robusta. In addition, by using the provided formula from the assignment introduction, we can take a ratio of the coefficients to see the relative shares of each coffee bean:

$$Arabica: \frac{0.0871}{0.0871 + 0.0586} * 100 = 59.78\%$$

$$Robusta: \frac{0.0586}{0.0871 + 0.0586} * 100 = 40.22\%$$

This is consistent with global consumption patterns, where Arabica is the dominant bean though Robusta captures a decent size of the market share, particularly in instant coffee and

value blends.

2) How large are the estimated price transmission elasticities overall, and what do they imply about how bean price changes pass through to retail coffee prices?

For interpretation, 1% increase in Arabica futures prices results in a 0.0871% increase in the coffee CPI, and a 1% increase in Robusta futures prices results in a 0.0586% increase in the coffee CPI. It's worth noting that since food CPI is used as a proxy for the inflation of the weighted-average coffee processing margin, we see that increases in processing and marketing margins dominate the effect on coffee CPI by at least 4 fold.

Part C: Tail Dependency

Arabica and Robusta coffee beans are usually not close substitutes, largely due to their distinct supply chains. However, with today's record-high coffee prices, substitution is expected to increase. If so, the correlation between their futures prices should be stronger when both prices are unusually high.

The figure below illustrates upper-tail dependency: for most prices (between the 10th and 90th percentiles) the correlation is moderate, but when both prices exceed the 90th percentile, their correlation tightens. If tail dependency were perfect, all high-price points would align on a single line.

1) Create a similar graph for Arabica and Robusta futures prices. Does the plot show visible evidence of upper-tail dependence between these two coffee markets?

In [23]:
```
shared_dates = set(all_arabica["Time"]) & set(all_robusta["Time"])

all_arabica = (all_arabica[all_arabica["Time"].isin(shared_dates)].reset_index(d
all_robusta = (all_robusta[all_robusta["Time"].isin(shared_dates)].reset_index(d

print(len(all_arabica), len(all_robusta))
```

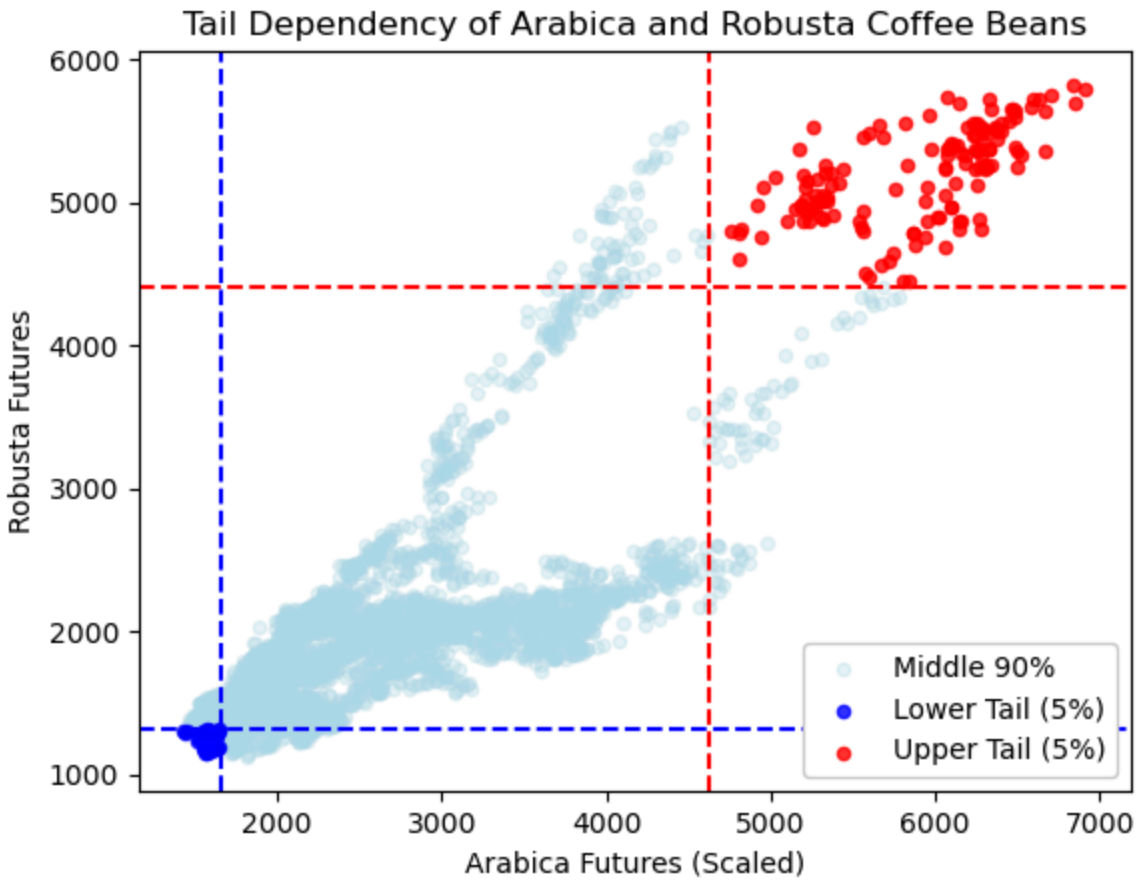
```
4235 4235
```

In [24]:
```
# scale Arabica up to Robusta prices
arabica_base = all_arabica.loc[0, "arabica_last"]
robusta_base = all_robusta.loc[0, "robusta_last"]

arabica_scale = robusta_base/arabica_base

print("arabica_base =", arabica_base)
print("robusta_base =",robusta_base)

# test
print("arabica_base after scaling =",arabica_base * arabica_scale)
```

```
arabica_base = 140.15
robusta_base = 2279
arabica_base after scaling = 2279.0
```

In [25]:

```python
# scatterplot

# first scale arabica to start at the same point as robusta
x = all_arabica['arabica_last'] * arabica_scale
y = all_robusta['robusta_last']

# set upper / lower tail thresholds; I chose 5% to clearly demonstrate the upper
x_low, x_high = np.percentile(x, [5, 95])
y_low, y_high = np.percentile(y, [5, 95])

# code that labels points where both arabica and robusta is in the upper/lower 5
both_low = (x <= x_low) & (y <= y_low)
both_up = (x >= x_high) & (y >= y_high)
middle = ~(both_low | both_up)

# plot
plt.figure()
plt.scatter(x[middle], y[middle], color="lightblue", alpha=0.3, s=20, label="Mid
plt.scatter(x[both_low], y[both_low], color="blue", alpha=0.8, s=20, label="Lowe
plt.scatter(x[both_up], y[both_up], color="red", alpha=0.8, s=20, label="Upper T

# add gridlines
plt.axvline(x_low, color="blue", linestyle="--")
plt.axvline(x_high, color="red", linestyle="--")
plt.axhline(y_low, color="blue", linestyle="--")
plt.axhline(y_high, color="red", linestyle="--")

# styling
plt.xlabel("Arabica Futures (Scaled)")
plt.ylabel("Robusta Futures")
plt.title("Tail Dependency of Arabica and Robusta Coffee Beans")
plt.legend(loc="lower right", framealpha=1)
plt.show()
```



Initially, I used a 10% tail which did not narrow the dataset down far enough to highlight the upper tail convergence. Using a 5% tail, we clearly see a convergence at the upper tails of the two coffee futures. This suggests that when prices are unusually high, Arabica and Robusta futures move together more strongly; the two types of coffee beans behave more like substitutes as prices increase.