

ПОСТРОЕНИЕ КРИВОЙ IS НА ОСНОВЕ МАКРОЭКОНОМИЧЕСКИХ ДАННЫХ ЭКОНОМИКИ РОССИИ 2011–2022 (I КВАРТАЛ) С ИСПОЛЬЗОВАНИЕМ PYTHON ДЛЯ ПРОВЕДЕНИЯ ПЕРВИЧНОГО АНАЛИЗА ДАННЫХ И ПОСТРОЕНИЯ РЕГРЕССИОННЫХ ЗАВИСИМОСТЕЙ КОМПОНЕНТ МОДЕЛИ

Щеголев Алексей Владимирович
Сотрудник, АО «НПЦ АСПЕКТ» Дубна
alexeyschegolev@yandex.ru

CONSTRUCTION OF THE IS CURVE BASED ON MACROECONOMIC DATA OF THE RUSSIAN ECONOMY 2011–2022 (I QUARTER) USING PYTHON FOR PRIMARY DATA ANALYSIS AND REGRESSION RELATIONSHIPS OF MODEL COMPONENTS

A. Shchegolev

Summary. This study conducts an analysis of Russia's macroeconomic data spanning from 2011 to the first quarter of 2022, aiming to construct the *IS* curve—an integral component of the *IS-LM* model. Leveraging Python programming tools, the research presents a methodology for primary data analysis and the establishment of regression dependencies for the elements of this model.

The dataset employed in this research comprises pivotal macroeconomic variables, including GDP, investments, and interest rates, allowing for an in-depth exploration of the impact of these factors on the Russian economy. Utilizing Python tools such as pandas, numpy, and matplotlib, the study conducts data analysis and constructs the *IS* curve, facilitating a better comprehension of the interrelationships among income, interest rates, and investment levels within the economy.

This article is intended for professionals in the field of economics, analysts, and researchers interested in employing programming for the analysis of macroeconomic data. The presented analysis extends new perspectives for understanding the factors influencing economic processes in Russia and demonstrates the application of contemporary data analysis tools in macroeconomics.

Keywords: IS curve, macroeconomic data, Python, primary data analysis, regression dependencies, GDP, investments, interest rates, libraries: pandas, numpy, matplotlib.

Аннотация. Данное исследование представляет собой анализ макроэкономических данных экономики России за период с 2011 по первый квартал 2022 года с целью построения кривой *IS* — одной из ключевых составляющих модели *IS-LM*. Применяя инструментальный язык программирования Python, статья предлагает методику первичного анализа данных и создания регрессионных зависимостей для компонентов данной модели. Используемый в исследовании набор данных содержит ключевые макроэкономические переменные, такие как ВВП, инвестиции и процентные ставки, предоставляя возможность провести детальный анализ влияния этих факторов на экономику России. С использованием инструментов Python, таких как pandas, numpy и matplotlib, проводится анализ данных и построение кривой *IS*, что позволяет лучше понять взаимосвязи между доходом, процентными ставками и уровнем инвестиций в экономике. Статья адресована специалистам в области экономики, аналитикам и исследователям, интересующимся применением программирования для анализа макроэкономических данных. Представленный анализ открывает новые горизонты для понимания факторов, влияющих на экономические процессы в России, и демонстрирует использование современных инструментов анализа данных в макроэкономике.

Ключевые слова: кривая *IS*, макроэкономические данные, Python, первичный анализ данных, регрессионные зависимости, ВВП, инвестиции, процентные ставки, библиотеки: pandas, numpy, matplotlib.

Введение

Глобальная экономическая среда постоянно меняется, представляя собой динамическую сеть взаимосвязанных факторов, влияющих на функционирование отдельных экономик. В этом контексте применение экономических моделей играет ключевую роль в понимании и прогнозировании экономических явлений. Мо-

дель *IS-LM*, основополагающая в макроэкономике, имеет особое значение, предоставляя понимание взаимодействия между доходом, процентными ставками и уровнем инвестиций в экономике [1–3]. Настоящее исследование затрагивает построение кривой *IS*, ключевого элемента модели *IS-LM*, в контексте экономической динамики России за период с 2011 по первый квартал 2022 года.

Россия, страна с многогранным экономическим ландшафтом, влияющим на глобальными рыночными силами, представляет поле для изучения сложностей экономических моделей. С использованием мощных возможностей языка программирования Python, данное исследование проводит детальный анализ макроэкономических данных России. Используемый в исследовании набор данных охватывает широкий спектр макроэкономических показателей, включая валовый внутренний продукт (ВВП), инвестиции и процентные ставки. Эти переменные служат основополагающими камнями для понимания экономической траектории и влияющих факторов в России.

Основная цель исследования заключается в построении кривой *IS* с использованием инструментов программирования на Python, таких как *pandas*, *numpy* и *matplotlib* [4]. Такой подход позволяет провести детальный анализ взаимосвязей между ВВП, инвестициями и процентными ставками, проясняя их совокупное влияние на экономику России. Путем применения методов регрессии исследование нацелено на определение и количественное описание этих взаимосвязей, обеспечивая более глубокое понимание того, как изменения в этих экономических переменных влияют друг на друга и, следовательно, на общую экономическую среду в России [5].

Материалы и методы исследования

Объектом исследования является макроэкономическая динамика Российской Федерации за период с 2011 по первый квартал 2022 года. Этот период был выбран из-за значимых экономических изменений, включая колебания в ценах на энергоносители, внутренние и внешние экономические факторы, а также изменения в макроэкономической политике страны.

Исследование сфокусировано на анализе макроэ-

кономических переменных, таких как ВВП, инвестиции и процентные ставки, и их взаимосвязи, чтобы более полно и точно понять динамику экономики России. Анализ этих данных призван раскрыть влияние фундаментальных экономических показателей на общую экономическую структуру страны (Рисунок 1).

Макроэкономические переменные представляют собой ключевые индикаторы здоровья экономики: ВВП является мерой общего экономического производства; инвестиции — индикатором будущего развития экономики; процентные ставки — фактором, влияющим на инвестиционные решения и распределение капитала.

Использование Python и связанных с ним библиотек позволяет провести анализ данных с высокой точностью и глубиной. *Pandas* и *NumPy* обеспечивают эффективную обработку и манипуляцию данными, *Matplotlib* и *Seaborn* — визуализацию результатов, а *Statsmodels* и *StatsPy* — статистический анализ и моделирование данных (Рисунок 2) [6–9].

Проведение изучения этих макроэкономических переменных и их взаимосвязей с использованием указанных инструментов позволяет создать более глубокое понимание динамики экономики России, ее уязвимостей и потенциальных тенденций, что является ключевым аспектом для понимания и прогнозирования экономического развития страны.

Для проведения анализа макроэкономических данных, необходимо прибегнуть к широкому спектру математических инструментов, включая статистику, алгебру, и методы численного анализа. В рамках данной работы ключевым становится метод статистического регрессионного моделирования и анализа.

Регрессионный анализ является мощным инструментом для изучения связей между переменными [10]. В кон-

	Год	Квартал	валовый внутренний продукт, млрд. руб. (Y)	Процентная ставка,% (i)	Сбережения, млрд. руб. (S)	Валовые частные внутренние инвестиции, млрд. руб (I)	i(m) - темп инфляции, %
0	2011	I	13024.799539	8.700000	6038.8	5178.5	3.81
1	2011	II	14434.822306	8.600000	6459.5	5770.9	1.14
2	2011	III	15745.583567	8.000000	6682.8	6781.8	-0.29
3	2011	IV	16908.795427	9.300000	7203.9	7253.1	1.35
4	2012	I	15182.776397	9.200000	6997.3	5989.8	1.46
5	2012	II	16435.995569	9.300000	7453.0	6401.0	1.73
6	2012	III	17715.787935	8.900000	7617.9	7674.5	1.89
7	2012	IV	18768.889727	9.400000	8119.0	8021.0	1.35
8	2013	I	16370.002897	8.800000	7831.7	6202.8	1.88
9	2013	II	17507.881781	9.600000	8354.3	6583.3	1.60
10	2013	III	19003.469311	9.400000	8568.5	7909.6	1.17
11	2013	IV	20104.347142	9.000000	9045.5	8431.3	1.65
12	2014	I	17311.393635	10.200000	8636.5	6175.3	2.33
13	2014	II	19044.189388	10.600000	9264.2	6902.6	2.44
14	2014	III	20544.004177	11.400000	9465.6	8356.2	1.39
15	2014	IV	22130.452874	9.500000	10072.7	9158.5	4.79
16	2015	I	18467.912804	15.000000	9266.9	6783.5	7.44
17	2015	II	19751.014988	12.000000	9815.8	7776.5	1.00
18	2015	III	21788.589405	11.500000	10037.8	9357.7	1.73

Рис. 1. Данные для проведения исследования

```
import pandas as pd
from IPython.display import display
from scipy.stats import pearsonr
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sympy import symbols, Eq, solve
import numpy as np
import matplotlib.pyplot as plt
from sympy import symbols, lambdify
```

Рис. 2. Импорт рабочих библиотек

тексте макроэкономических данных, линейная регрессия позволяет исследовать влияние одной переменной (например, ВВП) на другую (например, уровень инвестиций) и оценить силу и направление этой связи. Это происходит путем построения линейной модели, которая наилучшим образом соотносит зависимую переменную с одной или несколькими независимыми переменными. Множественная регрессия углубляет этот анализ, позволяя учесть несколько факторов одновременно, уточняя влияние каждого из них на зависимую переменную.

Помимо регрессионного анализа данные подверглись процедуре дефлирования, которое представляет собой процесс очистки данных от ошибок, несоответствий, пропущенных значений и выбросов. В контексте данного исследования по построению компоненты модели *IS-LM* на основе макроэкономических данных России, дефлирование является ключевым этапом [11].

Очистка данных позволяет создать более точную и достоверную основу для анализа. Имея чистые данные, исследователи снижают вероятность искажений результатов, обусловленных ошибками в исходных данных [12].

Основная важность дефлирования данных для данного исследования заключается в том, что точность результатов кривой товарного зависит от качества входных данных. Некорректные данные могут исказить результаты анализа и привести к неверным выводам о взаимосвязях макроэкономических переменных. Исключение выбросов или исправление ошибок в данных обеспечивает более точные и достоверные результаты при анализе взаимосвязей между доходом, процентными ставками и уровнем инвестиций, что важно для корректного построения *IS*.

Таким образом, регрессионный анализ и дефлирование данных, реализованные с помощью библиотек языка программирования Python, играют ключевую роль в обеспечении качественного анализа и построения модели, позволяя минимизировать возможные искажения, обусловленные ошибками в исходных данных, и повышая достоверность результатов исследования.

Результаты

На первом этапе проведения работы необходимо осуществить считывание данных из файла *Excel*, для ко-

торого используется функция `pd.read_excel()`. Ниже приведен пример кода, который загружает данные из файла *Excel* по указанному пути, представленные выше на рисунке 1 (Рисунок 3).

```
# Путь к файлу
file_path = 'data/IS.xlsx'

# Используем pd.read_excel() для чтения Excel файла
df = pd.read_excel(file_path)

# Теперь df содержит данные из файла IS.xlsx
# Отображение данных
display(df)
```

Рис. 3. Импорт данных из файла Excel

На следующем этапе необходимо провести подготовку данных к проведению исследования посредством дефлирования, а для этого требуется выполнить следующие шаги (Рисунок 4):

1. Разделить столбец « $i(m)$ — темп инфляции, %» на 100, чтобы получить темп инфляции в десятичном формате.
2. Создает столбец « $J(m)$ — цепной индекс», добавляя 1 к темпу инфляции.
3. Устанавливает начальное значение базисного индекса (предполагается, что для первой строки базисный индекс равен 1).
4. Затем выполняется итерация по строкам данных и рассчитывается базисный индекс как произведение предыдущего базисного индекса и $(1 + \text{темпы инфляции})$.

```
# Разделение столбца "i(m) - темп инфляции, %" на 100 и сохранение в столбце
df['i(m) - темп инфляции'] = df['i(m) - темп инфляции, %'] / 100

# Добавление 1 к столбцу "i(m) - темп инфляции" и сохранение в столбце "J(m) -
# цепной индекс"
df['J(m) - цепной индекс'] = df['i(m) - темп инфляции'] + 1

# Установка начального значения базисного индекса (базисный индекс для первой
# строки равен 1)
df.loc[0, 'GJ(m) - базисный индекс'] = 1

# Итерация по строкам для вычисления базисного индекса
for i in range(1, len(df)):
    df.loc[i, 'GJ(m) - базисный индекс'] = df.loc[i-1, 'GJ(m) - базисный индекс']
    * (1 + df.loc[i, 'i(m) - темп инфляции'])

# Отображение данных
display(df)
```

Рис. 4. Подготовка индекса дефлирования исходных данных

После проведения вышеуказанных действий возможно провести дефлирование данных (Рисунок 5).

```
# Создание столбца "валовый внутренний продукт, млрд. руб. (Y) Деф"
df['валовый внутренний продукт, млрд. руб. (Y) Деф'] = df['валовый внутренний
продукт, млрд. руб. (Y)'] / df['GJ(m) - базисный индекс']

# Создание столбца "Сбережения, млрд. руб. (S) Деф"
df['Сбережения, млрд. руб. (S) Деф'] = df['Сбережения, млрд. руб. (S)'] /
df['GJ(m) - базисный индекс']

# Создание столбца "Валовые частные внутренние инвестиции, млрд. руб. (I) Деф"
df['Валовые частные внутренние инвестиции, млрд. руб. (I) Деф'] = df['Валовые
частные внутренние инвестиции, млрд. руб. (I)'] / df['GJ(m) - базисный индекс']

# Отображение данных
display(df)
```

Рис. 5. Дефлирование исходных данных

Год	Квартал	валовой внутренний продукт, млрд. руб. (V)	Процентная ставка, % (i)	Сбережения, млрд. руб. (S)	Валовые частные внутренние инвестиции, млрд. руб. (I)	i(m) - темп инфляции, %	i(m) - темп инфляции	J(m) - цепной индекс	G(m) - базисный индекс	валовой внутренний продукт, млрд. руб. (V)	Сбережения, млрд. руб. (S) Деф	Валовые частные внутренние инвестиции, млрд. руб. (I) Деф
0	2011	I	13024,799539	6038,8	5178,5	3,81	0,0381	1,0381	1,000000	13024,799539	6038,800000	5178,500000
1	2011	II	14434,822306	6459,5	5770,9	1,14	0,0114	1,0114	1,011400	14272,120136	6386,691714	5705,853273
2	2011	III	15745,583567	6682,8	6781,8	-0,29	-0,0029	0,9971	1,008467	15613,385964	6626,692195	6724,861005
3	2011	IV	16908,795427	7203,9	7253,1	1,35	0,0135	1,0135	1,022081	16543,494494	7048,265531	7096,402605
4	2012	I	15182,776397	6997,3	5898,8	1,46	0,0146	1,0146	1,037004	14641,006030	6747,613797	5776,064642
5	2012	II	16435,995569	7453,0	6401,0	1,73	0,0173	1,0173	1,054944	15579,972775	7064,831370	6067,621844
6	2012	III	17715,787935	7617,9	7674,5	1,89	0,0189	1,0189	1,074882	16481,608343	7087,195029	7139,851961
7	2012	IV	18768,889727	8119,0	8021,0	1,35	0,0135	1,0135	1,089393	17228,757030	7452,773199	7362,814857
8	2013	I	16370,002897	7831,7	6202,8	1,88	0,0188	1,0188	1,109874	14749,428185	7056,380288	5588,743858
9	2013	II	17507,881781	8354,3	6583,3	1,60	0,0160	1,0160	1,127632	15526,241066	7408,713250	5838,165010
10	2013	III	19003,463311	8568,5	7909,6	1,17	0,0117	1,0117	1,140825	16657,655008	7510,792614	6933,228133
11	2013	IV	20104,347142	9045,5	8431,3	1,65	0,0165	1,0165	1,159649	17336,585312	7800,207653	7270,564456
12	2014	I	17311,393635	8636,5	6775,3	2,33	0,0233	1,0233	1,186668	14588,231417	7277,938639	5203,896773
13	2014	II	19044,189388	9264,2	6902,6	2,44	0,0244	1,0244	1,215623	15666,195227	7620,947412	5678,240064
14	2014	III	20544,004177	9465,6	8356,2	1,39	0,0139	1,0139	1,232520	16668,288708	7679,873516	6779,766637
15	2014	IV	22130,452874	10072,7	9158,5	4,79	0,0479	1,0479	1,291558	17134,695085	7798,875340	7091,048061
16	2015	I	18467,912804	9266,9	6783,5	7,44	0,0744	1,0744	1,387650	13908,769238	6678,125187	4888,480744
17	2015	II	19751,014988	9815,8	7776,5	1,00	0,0100	1,0100	1,401526	14092,502620	7003,649549	5548,593158
18	2015	III	21788,589405	10037,8	9357,7	1,73	0,0173	1,0173	1,425773	15281,950059	7040,251916	6563,247460
19	2015	IV	23079,842857	10628,4	10156,9	2,28	0,0228	1,0228	1,458280	15826,751716	7288,309932	6964,983925
20	2016	I	18885,121083	9490,6	7150,2	2,06	0,0206	1,0206	1,488321	12688,875947	6376,715592	4804,205406

Рис. 6. Подготовленные данные для исследования

В результате проведенных манипуляций с исходным DataFrame (Рисунок 1) мы получаем данные, подготовленные к реализации компонент модели кривой IS (Рисунок 6).

Код, приведенный ниже (Рисунок 7), выполняет анализ корреляции между различными экономическими показателями с использованием Python и его библиотек Pandas и SciPy [13]. Этот анализ поможет нам понять, как взаимосвязаны валовый внутренний продукт (Y), сбережения (S), валовые частные внутренние инвестиции (I) и процентная ставка (i).

В результате выполнения данного кода можно сделать следующие выводы:

- Валовый внутренний продукт (Y) и частные внутренние инвестиции (I) сильно и положительно коррелируют друг с другом.
- Существует также высокая положительная корреляция между валовым внутренним продуктом (Y) и сбережениями (S).
- Процентная ставка (i) не имеет сильной корреляции ни с одним из перечисленных показателей.

В целом, данные выводы указывают на важность учета взаимосвязи между внутренним продуктом, инве-

```
# Создание подмножества данных с выбранными столбцами
subset_df = df[columns_of_interest]

# Вычисление матрицы корреляции
correlation_matrix = subset_df.corr()

# Отображение тепловой карты корреляций
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Тепловая карта корреляций')
plt.show()
```

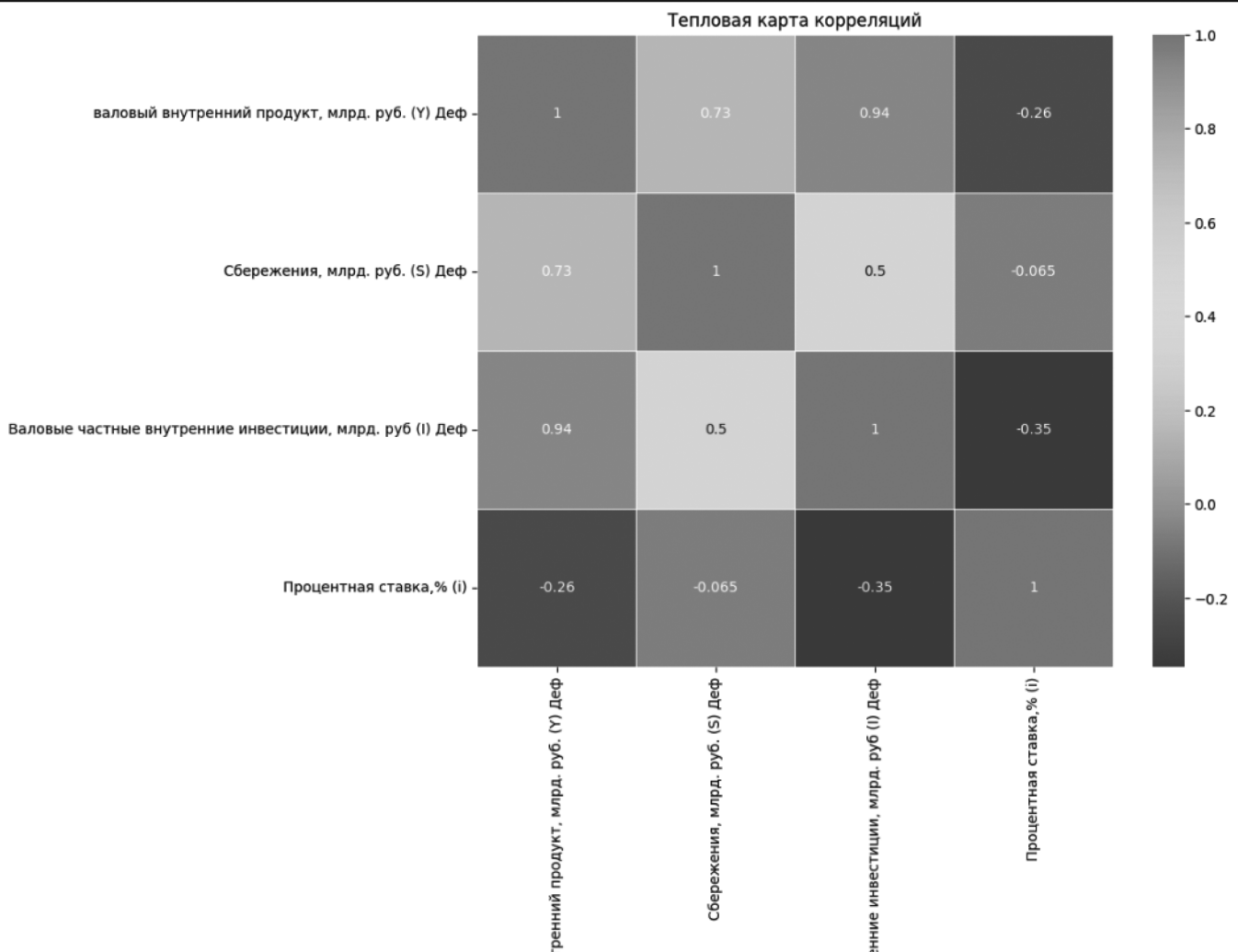


Рис. 7. Определение корреляционной зависимости между данными

стициями и сбережениями при формировании экономической политики в России. Они также подчеркивают, что процентная ставка не является главным фактором, влияющим на экономическое развитие, и что другие экономические факторы могут иметь более существенное значение.

Код на рисунке 8 выполняет анализ и построение линейных регрессионных моделей для трех важных экономических зависимостей в рамках исследования кривой *IS* (Investment-Savings curve). Подробно разберем каждую часть:

- **Уравнение $I(i) = ki + b$:**

1. В этой части кода анализируется зависимость между процентной ставкой (*i*) и частными внутренними инвестициями (*I*).
2. **X_i** представляет собой независимую переменную (процентную ставку), а **Y_i** — зависимую переменную (частные инвестиции).
3. С помощью библиотеки **statsmodels**, код создает линейную регрессионную модель и вычисляет коэффициенты **k_i** (наклон) и **b_i** (пересечение с осью *Y*) для этой зависимости.

- **Уравнение $S(Y) = kY + b$:**

1. Здесь анализируется зависимость между валовым внутренним продуктом (*Y*) и сбережениями (*S*).
2. **X_s** представляет собой независимую переменную (валовый внутренний продукт), а **Y_s** — зависимую переменную (сбережения).
3. Как и в первой части, код создает линейную регрессионную модель и вычисляет коэффициенты **k_s** и **b_s** .

- **Уравнение $I(S) = kS + b$:**

1. В этой части кода анализируется зависимость между сбережениями (*S*) и частными внутренними инвестициями (*I*).
2. **X_{is}** представляет собой независимую переменную (сбережения), а **Y_{is}** — зависимую переменную (частные инвестиции).
3. Код создает линейную регрессионную модель и вычисляет коэффициенты **k_{is}** и **b_{is}** .

Результатом выполнения данного кода станет уравнение:

```
# Уравнение I(i) = ki + b
X_i = df['Процентная ставка,% (i)'] # Независимая переменная (i)
Y_i = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] # Зависимая
# переменная (I(i))
X_i = sm.add_constant(X_i) # Добавление константы (intercept) к независимой
# переменной
model_i = sm.OLS(Y_i, X_i).fit() # Создание модели линейной регрессии
k_i = model_i.params[1] # Коэффициент k
b_i = model_i.params[0] # Коэффициент b

# Уравнение S(Y) = kY + b
X_s = df['валовый внутренний продукт, млрд. руб. (Y) Деф'] # Независимая
# переменная (Y)
Y_s = df['Сбережения, млрд. руб. (S) Деф'] # Зависимая переменная (S(Y))
X_s = sm.add_constant(X_s) # Добавление константы (intercept) к независимой
# переменной
model_s = sm.OLS(Y_s, X_s).fit() # Создание модели линейной регрессии
k_s = model_s.params[1] # Коэффициент k
b_s = model_s.params[0] # Коэффициент b

# Уравнение I(S) = kS + b
X_is = df['Сбережения, млрд. руб. (S) Деф'] # Независимая переменная (S)
Y_is = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] #
# Зависимая переменная (I(S))
X_is = sm.add_constant(X_is) # Добавление константы (intercept) к независимой
# переменной
model_is = sm.OLS(Y_is, X_is).fit() # Создание модели линейной регрессии
k_is = model_is.params[1] # Коэффициент k
b_is = model_is.params[0] # Коэффициент b

# Вывод уравнений
print(f"I(i) = {k_i:.2f} * i + {b_i:.2f}")
print(f"S(Y) = {k_s:.2f} * Y + {b_s:.2f}")
print(f"I(S) = {k_is:.2f} * S + {b_is:.2f}")
```

Рис. 8. Определение параметров уравнений декомпозиции кривой *IS*

$$I = -175.06 \cdot i + 8158.1 \quad (1)$$

А также линейная зависимость между уровнем сбережений населения и уровнем ВВП:

$$S = 0.18 \cdot Y + 4257.12 \quad (2)$$

$$I = 1.56 \cdot S - 4424.31 \quad (3)$$

Приведенные на рисунках 9–11 фрагменты выполняющей визуализацию зависимости между исследуемыми па-

раметрами модели, а также с помощью метода `summary()` позволяет получить обширный отчет о регрессии, который включает в себя информацию о коэффициентах регрессии, их статистической значимости и показателях качества модели, таких как R -квадрат.

Использование кода рисунков 9–11 были получены графики и показатели качества уравнений, представленные на рисунках 12–14.

```
# Определение независимой и зависимой переменных
X = df['валовой внутренний продукт, млрд. руб. (Y) Деф'] # Независимая
переменная (Y)
y = df['Сбережения, млрд. руб. (S) Деф'] # Зависимая переменная (S(Y))

# Добавление константы (intercept) к независимой переменной
X = sm.add_constant(X)

# Создание модели линейной регрессии
model = sm.OLS(y, X).fit()

# Вывод результатов регрессии
print(model.summary())

# График наблюдаемых данных
plt.scatter(X['валовой внутренний продукт, млрд. руб. (Y) Деф'], y,
label='Наблюдаемые данные', color='blue')

# График регрессионной линии
predicted_values = model.predict(X)
plt.plot(X['валовой внутренний продукт, млрд. руб. (Y) Деф'], predicted_values,
label=f'Регрессионная линия: S(Y) = {model.params[1]:.2f} * Y + {model.params[0]:.2f}', color='red')

plt.xlabel('Валовой внутренний продукт, млрд. руб. (Y) Деф')
plt.ylabel('Сбережения, млрд. руб. (S) Деф')
plt.title('Зависимость Сбережений от ВВП')
plt.legend()
plt.show()
```

Рис. 9. Определение коэффициентов уравнения $S(Y)$ и параметров его качества

```
# Определение независимой и зависимой переменных
X = df['Процентная ставка,% (i)'] # Независимая переменная (i)
y = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] # Зависимая
переменная (I(i))

# Добавление константы (intercept) к независимой переменной
X = sm.add_constant(X)

# Создание модели линейной регрессии
model = sm.OLS(y, X).fit()

# Вывод результатов регрессии
print(model.summary())

# График наблюдаемых данных
plt.scatter(y, X['Процентная ставка,% (i)'], label='Наблюдаемые данные',
color='blue')

# График регрессионной линии
predicted_values = model.predict(X)
plt.plot(predicted_values, X['Процентная ставка,% (i)'], label=f'Регрессионная
линия: I(i) = {model.params[1]:.2f} * i + {model.params[0]:.2f}', color='red')

plt.xlabel('Валовые частные внутренние инвестиции, млрд. руб (I) Деф')
plt.ylabel('Процентная ставка,% (i)')
plt.title('Зависимость Валовых частных внутренних инвестиций от Процентной
ставки')
plt.legend()
plt.show()
```

Рис. 10. Определение коэффициентов уравнения $I(i)$ и параметров его качества

```

# Определение независимой и зависимой переменных
X = df['Сбережения, млрд. руб. (S) Деф'] # Независимая переменная (S)
y = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] # Зависимая
переменная (I)

# Добавление константы (intercept) к независимой переменной
X = sm.add_constant(X)

# Создание модели линейной регрессии
model = sm.OLS(y, X).fit()

# Вывод результатов регрессии
print(model.summary())

# График наблюдаемых данных
plt.scatter(X['Сбережения, млрд. руб. (S) Деф'], y, label='Наблюдаемые данные',
color='blue')

# График регрессионной линии
predicted_values = model.predict(X)
plt.plot(X['Сбережения, млрд. руб. (S) Деф'], predicted_values,
label='Регрессионная линия: I = {model.params[1]:.2f} * S +
{model.params[0]:.2f}', color='red')

plt.xlabel('Сбережения, млрд. руб. (S) Деф')
plt.ylabel('Валовые частные внутренние инвестиции, млрд. руб (I) Деф')
plt.title('Зависимость Валовых частных внутренних инвестиций от Сбережений')
plt.legend()
plt.show()

```

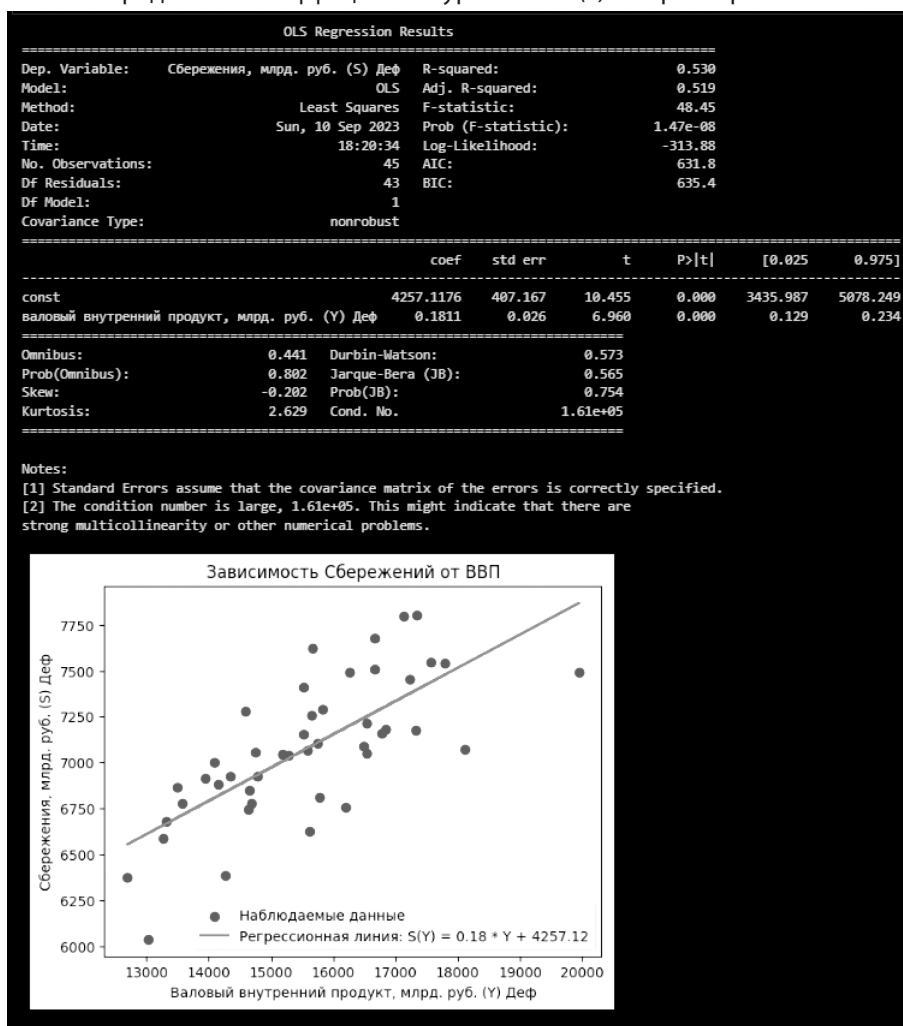
Рис. 11. Определение коэффициентов уравнения $I(S)$ и параметров его качества

Рис. 12. Результаты выполнения кода рисунка 9

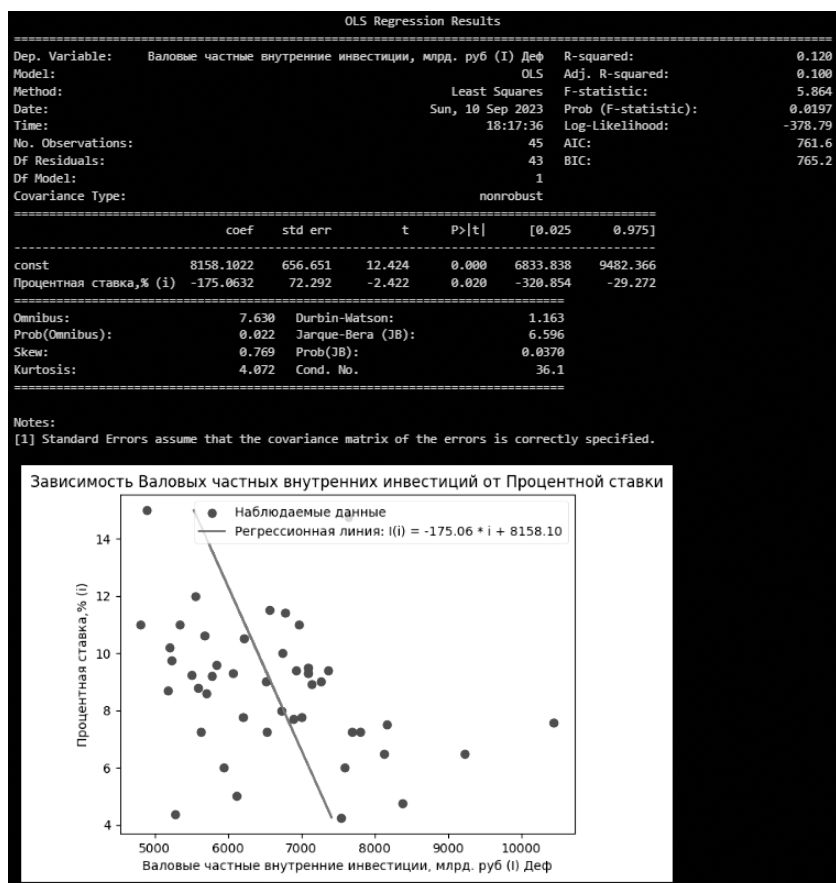


Рис. 13. Результаты выполнения кода рисунка 10

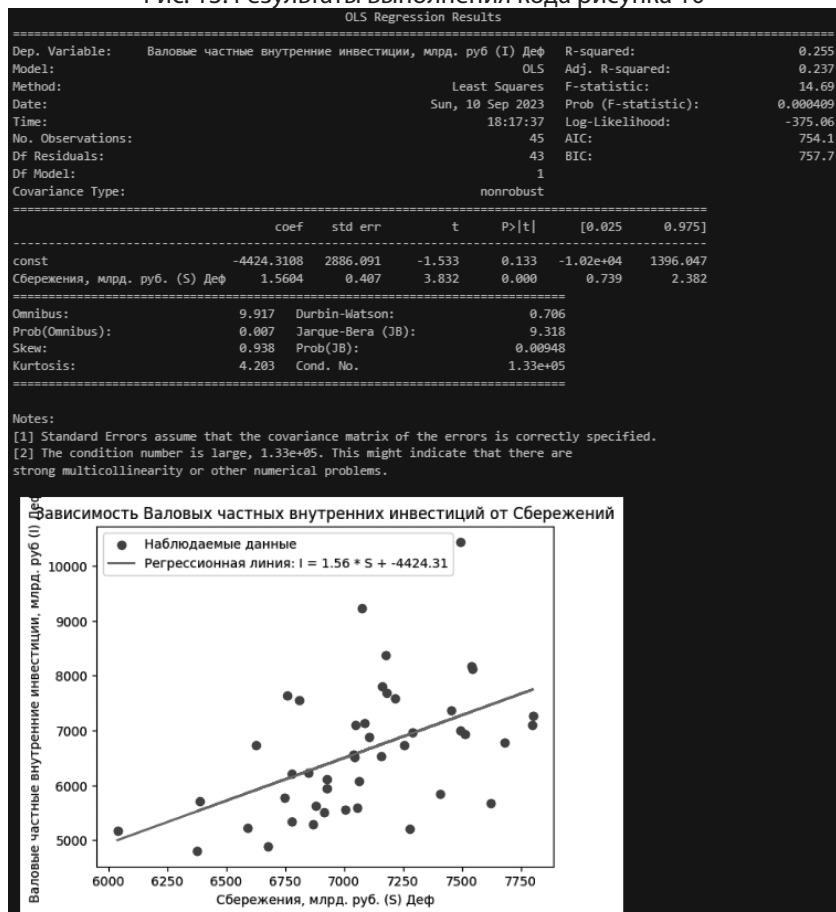


Рис. 14. Результаты выполнения кода рисунка 11

```

# Уравнение  $I(i) = ki + b$ 
X_i = df['Процентная ставка,% (i)'] # Независимая переменная (i)
Y_i = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] # Зависимая
переменная (I(i))
X_i = sm.add_constant(X_i) # Добавление константы (intercept) к независимой
переменной
model_i = sm.OLS(Y_i, X_i).fit() # Создание модели линейной регрессии
k_i = model_i.params[1] # Коэффициент k
b_i = model_i.params[0] # Коэффициент b

# Уравнение  $S(Y) = kY + b$ 
X_s = df['Валовый внутренний продукт, млрд. руб. (Y) Деф'] # Независимая
переменная (Y)
Y_s = df['Сбережения, млрд. руб. (S) Деф'] # Зависимая переменная (S(Y))
X_s = sm.add_constant(X_s) # Добавление константы (intercept) к независимой
переменной
model_s = sm.OLS(Y_s, X_s).fit() # Создание модели линейной регрессии
k_s = model_s.params[1] # Коэффициент k
b_s = model_s.params[0] # Коэффициент b

# Уравнение  $I(S) = kS + b$ 
X_is = df['Сбережения, млрд. руб. (S) Деф'] # Независимая переменная (S)
Y_is = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф'] #
Зависимая переменная (I(S))
X_is = sm.add_constant(X_is) # Добавление константы (intercept) к независимой
переменной
model_is = sm.OLS(Y_is, X_is).fit() # Создание модели линейной регрессии
k_is = model_is.params[1] # Коэффициент k
b_is = model_is.params[0] # Коэффициент b

# Вывод уравнений
print(f"I(i) = {k_i:.2f} * i + {b_i:.2f}")
print(f"S(Y) = {k_s:.2f} * Y + {b_s:.2f}")
print(f"I(S) = {k_is:.2f} * S + {b_is:.2f}")

```

Рис. 15. Уравнения декомпозиции кривой IS

С целью аналитического определения коэффициентов уравнения кривой IS объединим ранее проведенные расчетные операции в рамках одного фрагмента кода (рисунок 15).

В результате выполнения кода Рисунка 15 получим уравнения 1–3.

Используя данные результаты, можно перейти непосредственно к выражению уравнения, описывающего кривую товарного баланса, а также провести визуализацию всех компонент модели и самой кривой согласно классическому теоретическому представлению (рисунки 16, 17 и 18).

Обсуждение

Важным шагом была подготовка данных, включая дефлирование, очистку и подготовку для анализа. Использование библиотек Python, таких как Pandas, NumPy, и Matplotlib, сделало процесс анализа и визуализации данных более удобным и эффективным.

Результаты показали, что на основе полученных данных удалось построить компонент IS из состава модели IS-LM и проанализировать взаимосвязь между

уровнем дохода, процентными ставками и инвестициями в российской экономике. Установленные уравнения $S(Y)$, $I(i)$ и $I(S)$ позволили более детально изучить факторы, влияющие на решения о сбережениях и инвестициях в стране.

Декомпозиция кривой IS выявила важные компоненты, демонстрируя зависимость между сбережениями, инвестициями и уровнем дохода. Это позволяет предполагать, что изменения в уровне дохода могут оказать существенное влияние на инвестиционные решения.

Тем не менее, стоит отметить ограничения данного исследования, такие как ограниченный временной период данных и ограничения модели. Дальнейшие исследования могут включать больший объем данных и более сложные модели для получения более точных результатов.

В целом, данное исследование предоставляет важные инсайты в экономическую динамику России, отражая влияние уровня дохода и процентных ставок на инвестиции, что может быть полезно для разработки экономических стратегий и прогнозирования будущих тенденций в экономике страны.

Заключение

В данном исследовании проведен глубокий анализ экономических данных, используя мощные инструменты языка программирования Python. Цель заключалась в определении уравнения кривой IS и выявлении взаимосвязей между процентными ставками, валовым внутренним продуктом, сбережениями и инвестициями. Данное сочетание делает это исследование уникальным, так это интеграция научных методов с применением современных технологий.

В рамках исследования использовали библиотеки Python, такие как Pandas для обработки данных, StatsModels для статистического анализа и SymPy для символьных вычислений. Этот инструментарий позволил нам не только провести анализ данных, но и рассчитать коэффициенты уравнений и даже получить аналитическое выражение для кривой IS .

В результате исследования мы установили уравнение кривой IS и выявили, что оно имеет форму линейной зависимости между валовым внутренним продуктом и процентными ставками. Также мы показали, что изменения в сбережениях и инвестициях могут существенно влиять на равновесие в экономике.

Использование Python придало исследованию более высокую точность и эффективность, а также открыло новые возможности для анализа данных и моделирования экономических процессов. Этот опыт подчеркивает важность современных вычислительных инструментов в научных исследованиях и демонстрирует их применимость в экономической науке и практике.

Таким образом, исследование не только вносит вклад в понимание макроэкономических взаимосвязей, но и подчеркивает роль программирования и анализа данных в современной науке и экономике.

```
# Функция для вычисления коэффициентов уравнения  $I(i) = ki + b$ 
def calculate_I_i_coeffs(i_data, I_data):
    i = symbols('i')
    X_i = i_data
    Y_i = I_data
    X_i = sm.add_constant(X_i)
    model_i = sm.OLS(Y_i, X_i).fit()
    k_i = model_i.params[1]
    b_i = model_i.params[0]
    return k_i, b_i

# Функция для вычисления коэффициентов уравнения  $S(Y) = k_s Y + b_s$ 
def calculate_S_Y_coeffs(Y_data, S_data):
    Y = symbols('Y')
    X_s = Y_data
    Y_s = S_data
    X_s = sm.add_constant(X_s)
    model_s = sm.OLS(Y_s, X_s).fit()
    k_s = model_s.params[1]
    b_s = model_s.params[0]
    return k_s, b_s

# Функция для вычисления коэффициентов уравнения  $I(S) = k_s S + b_{is}$ 
def calculate_I_S_coeffs(S_data, I_data):
    S = symbols('S')
    X_is = S_data
    Y_is = I_data
    X_is = sm.add_constant(X_is)
    model_is = sm.OLS(Y_is, X_is).fit()
    k_is = model_is.params[1]
    b_is = model_is.params[0]
    return k_is, b_is

# Функция для вычисления уравнения кривой  $IS$ 
def calculate_IS_equation(i_data, Y_data, S_data, I_data):
    k_i, b_i = calculate_I_i_coeffs(i_data, I_data)
    k_s, b_s = calculate_S_Y_coeffs(Y_data, S_data)
    k_is, b_is = calculate_I_S_coeffs(S_data, I_data)

    i = symbols('i')
    S = symbols('S')
    Y = symbols('Y')

    I_i = k_i * i + b_i
    S_Y = k_s * Y + b_s
    I_S = k_is * S + b_is

    equation = Eq(I_i, k_is * (k_s * Y + b_s) + b_is)

    return equation

# Ваши данные
i_data = df['Процентная ставка, % (i)']
Y_data = df['валовый внутренний продукт, млрд. руб. (Y) Деф']
S_data = df['Сбережения, млрд. руб. (S) Деф']
I_data = df['Валовые частные внутренние инвестиции, млрд. руб (I) Деф']

# Вычисляем уравнение кривой  $IS$ 
IS_equation = calculate_IS_equation(i_data, Y_data, S_data, I_data)

# Решение уравнения относительно  $Y$ 
Y_symbol = symbols('Y')
solution = solve(IS_equation, Y_symbol)

# Вывод результата
print(f"Уравнение кривой  $IS$ :  $Y = \{solution[0]\}$ ")

✓ 0.2s
```

Уравнение кривой IS : $Y = 21022.6140395568 - 619.637152353685 * i$

Рис. 16. Определение параметров кривой IS

```

# Создаем фигуру с четырьмя графиками
fig, axes = plt.subplots(2, 2, figsize=(15, 10))

# Первый график (I(i) vs. i) с осями абсцисс и ординат поменяными местами
i_values = np.linspace(min(i_data), max(i_data), 100)
I_i_values = [float(k_i * i + b_i) for i in i_values]
axes[0, 0].plot(I_i_values, i_values, label='I(i) = k_i + b', color='blue') #
# Меняем местами x и y
axes[0, 0].set_xlabel('Валовые частные внутренние инвестиции, млрд. руб (I(i))')
axes[0, 0].set_ylabel('Процентная ставка, % (i)')
axes[0, 0].set_title('Уравнение I(i) = k_i + b')
axes[0, 0].legend()

# Второй график (кривая IS) с осями абсцисс и ординат поменяными местами
i_symbol = symbols('i')
IS_function = lambdify(i_symbol, solution[0], modules='numpy')
i_values = np.linspace(min(i_data), max(i_data), 100)
IS_values = IS_function(i_values)
axes[0, 1].plot(IS_values, i_values, label='IS curve', color='purple') # Меняем
# местами x и y
axes[0, 1].set_xlabel('Валовый внутренний продукт, млрд. руб. (Y)') # Изменяем
# подпись к оси X
axes[0, 1].set_ylabel('Процентная ставка, % (i)') # Изменяем подпись к оси Y
axes[0, 1].set_title('Уравнение кривой IS')
axes[0, 1].legend()

# Третий график (I(S) vs. S) с осями абсцисс и ординат поменяными местами
S_values = np.linspace(min(S_data), max(S_data), 100)
I_S_values = [float(k_is * S + b_is) for S in S_values]
axes[1, 0].plot(I_S_values, S_values, label='I(S) = kS + b', color='red') #
# Меняем местами x и y
axes[1, 0].set_xlabel('Валовые частные внутренние инвестиции, млрд. руб
# (I(i))') # Изменяем подпись к оси X
axes[1, 0].set_ylabel('Сбережения, млрд. руб. (S)') # Изменяем подпись к оси Y
axes[1, 0].set_title('Уравнение I(S) = kS + b')
axes[1, 0].legend()

# Четвертый график (S(Y) vs. Y)
Y_values = np.linspace(min(Y_data), max(Y_data), 100)
S_Y_values = [float(k_s * Y + b_s) for Y in Y_values]
axes[1, 1].plot(Y_values, S_Y_values, label='S(Y) = kY + b', color='green')
axes[1, 1].set_xlabel('Валовый внутренний продукт, млрд. руб. (Y)')
axes[1, 1].set_ylabel('Сбережения, млрд. руб. (S)')
axes[1, 1].set_title('Уравнение S(Y) = kY + b')
axes[1, 1].legend()

# Отображаем фигуру с графиками
plt.tight_layout()
plt.show()

```

Рис. 17. Фрагмент кода для представления визуализации фрагментов декомпозиции кривой

ЛИТЕРАТУРА

1. Acemoglu, D., Laibson, D.I. Optimal capital accumulation in general equilibrium: A closed-form solution. *Journal of Economic Theory*. — 2002. — Т. 105. — №. 1. — С. 1–45.
2. King, R.G., Plosser, C.I. Money, Credit, and Prices in a Real Business Cycle. *American Economic Review*. — 1984. — Т. 74. — №. 3. — С. 363–380.
3. Campante, F., Sturzenegger, F., Velasco A. *Advanced macroeconomics: an easy guide*. — LSE Press, 2021. — С. 418.
4. McKinney, Wes. *Python for Data Analysis* / Wes McKinney. — O'Reilly Media, Inc, 2017. 300 c.
5. Blanchard, O.J., Kahn, C.M. The solution of linear difference models under rational expectations // *Econometrica: Journal of the Econometric Society*. — 1980. — Т. 48. — №. 5. — С. 1305–1311.
6. Sahoo, K., Samal, A.K., Pramanik, J., Pani, S.K. Exploratory data analysis using Python // *International Journal of Innovative Technology and Exploring Engineering*. — 2019. — Т. 8. — №. 12. — С. 4727–4735.
7. VanderPlas, J. *Python Data Science Handbook*. / J. VanderPlas. — O'Reilly Media, Inc, 2016. 517 c.
8. Grus, J. *Data Science from Scratch*. / J. Grus. — O'Reilly Media, Inc, 2015. 369 c.
9. Raschka, S., Mirjalili, V. *Python Machine Learning*. / S. Raschka, V. Mirjalili. — Packt Publishing, 2017. 285 c.
10. Montgomery, D.C., Peck, E.A., Vining, G.G. *Introduction to Linear Regression Analysis*. / D.C. Montgomery, E.A. Peck, G.G. Vining — Wiley. 2012. 670 c.
11. Розанова, Н.М. *Макроэкономический анализ открытой экономики: учебник и практикум для вузов* / Н.М. Розанова. — Москва: Издательство Юрайт, 2024. — 193 с.
12. Крюкова А.С., Дубовик К.А. Дефляционная спираль и методы борьбы с ней // *актуальные вопросы современных научных исследований*. — 2023. — С. 22–24.
13. Саммерфилд, М. *Программирование на Python 3. Подробное руководство*. / М. Саммерфилд. — Пер. с англ. — СПб.: Символ-Плюс, 2009. — 608 с.

© Щеголев Алексей Владимирович (alexeyschegolev@yandex.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»

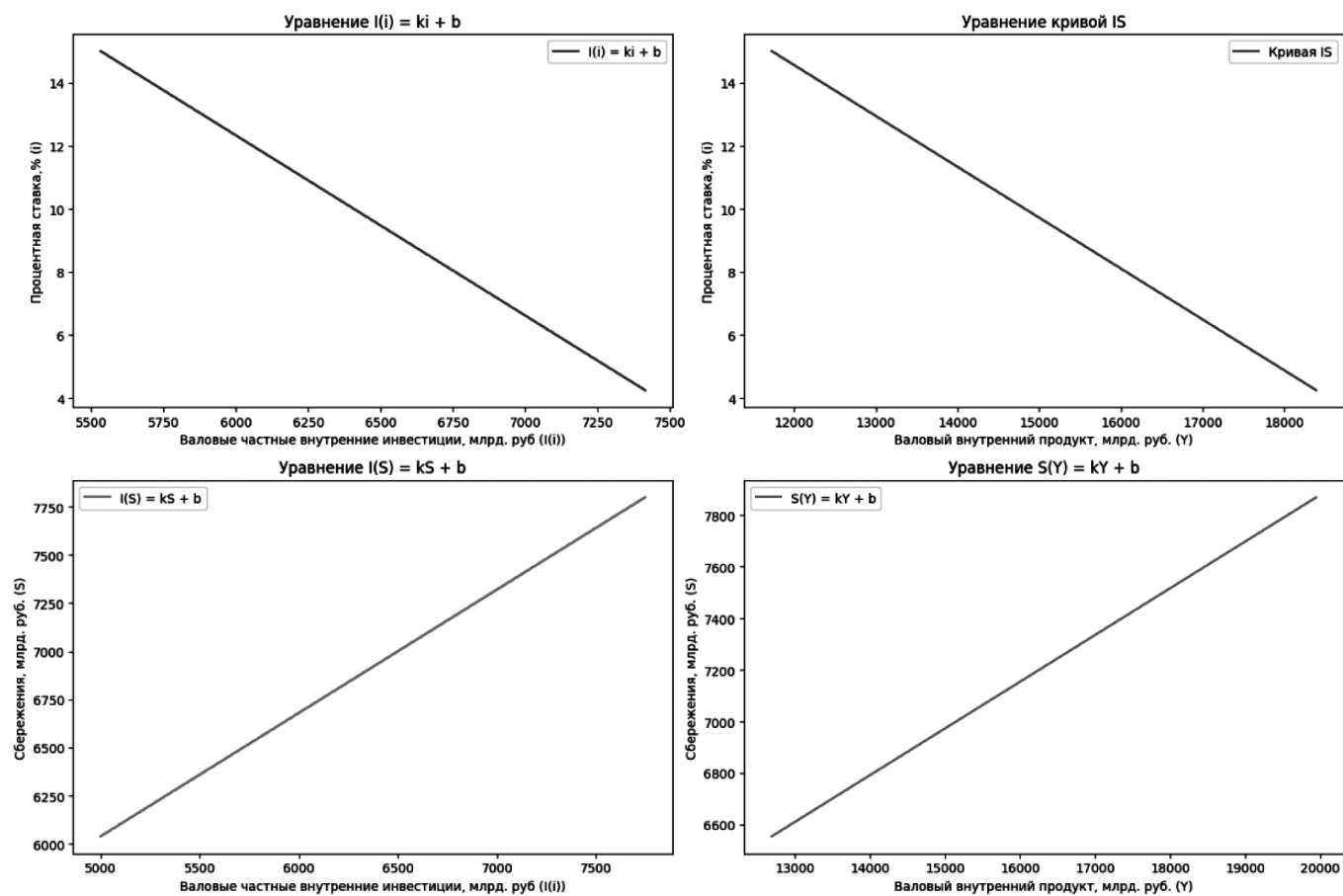


Рис. 18. Кривая IS и ее компоненты