```
import pandas as pd

# Load the data from the CSV file
df = pd.read_csv('22070521106_CA1_EDA.csv')

# Fill any missing values with 0
df.fillna(0, inplace=True)
```

```
# Display the first 5 rows to get a quick look at the data
print("Data head:")
print(df.head())

# Show column data types and non-null values
print("\nDataFrame info:")
df.info()

# Get descriptive statistics for all numerical columns
print("\nDescriptive statistics:")
print(df.describe())
```

```
2   2  2022  Andaman and Nicobar Islands          35
3   3  2022  Andaman and Nicobar Islands          35
4   4  2022  Andaman and Nicobar Islands          35

              district_name  district_code          region population_group  \
0  North and Middle Andaman            632  Eastern Region            Rural
1            South Andamans            602  Eastern Region            Rural
2            South Andamans            602  Eastern Region       Semi-urban
3            South Andamans            602  Eastern Region            Urban
4  North and Middle Andaman            632  Eastern Region            Rural

   no_of_offices  no_of_accounts  deposit_amount
0             10             108             729
1             13             106             775
2             10              64             463
3             36             301            4620
4              0               0               0

DataFrame info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14037 entries, 0 to 14036
Data columns (total 11 columns):
```

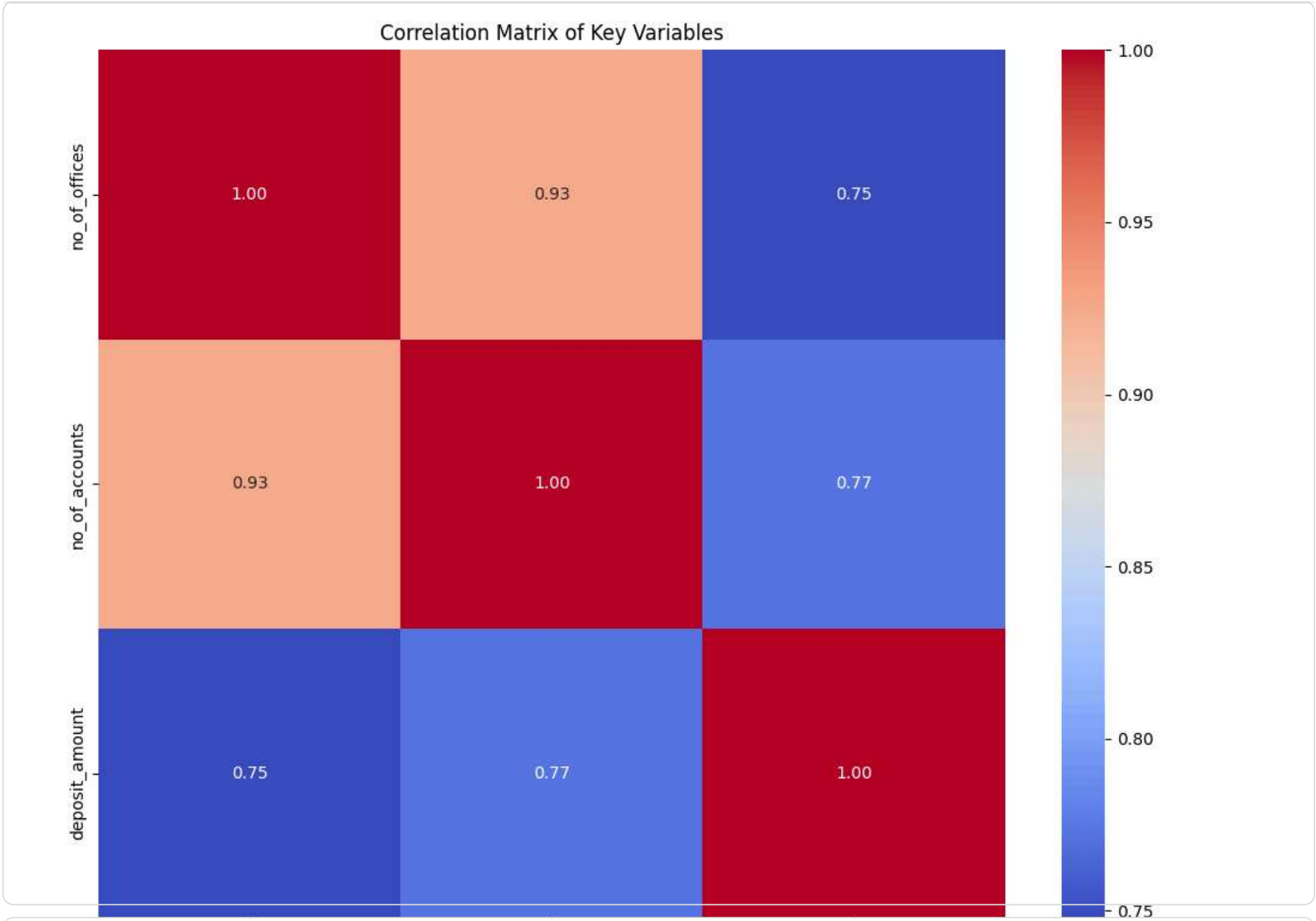
```
 2   state_name        14037 non-null  object
 3   state_code        14037 non-null  int64
 4   district_name     14037 non-null  object
 5   district_code     14037 non-null  int64
 6   region            14037 non-null  object
 7   population_group  14037 non-null  object
 8   no_of_offices     14037 non-null  int64
 9   no_of_accounts    14037 non-null  int64
 10  deposit_amount    14037 non-null  int64
dtypes: int64(7), object(4)
memory usage: 1.2+ MB

Descriptive statistics:
                id          year    state_code  district_code  no_of_offices  \
count  14037.000000  14037.000000  14037.000000   14037.000000   14037.000000
mean    7062.753224   2020.504310     18.226473     355.351500      43.268718
std     4088.358741      1.117906      9.973987     205.731486     105.355999
min        0.000000   2019.000000      1.000000       1.000000       0.000000
25%     3524.000000   2020.000000      9.000000     169.000000       0.000000
50%     7053.000000   2021.000000     19.000000     365.000000       0.000000
75%    10601.000000   2022.000000     27.000000     528.000000      54.000000
max    14159.000000   2022.000000     38.000000     734.000000    2807.000000

       no_of_accounts  deposit_amount
count    14037.000000    1.403700e+04
mean       599.058844    4.189880e+03
std       1577.467136    3.394547e+04
min          0.000000    0.000000e+00
25%          0.000000    0.000000e+00
50%          0.000000    0.000000e+00
75%        781.000000    2.386000e+03
max      52981.000000    1.400625e+06
```

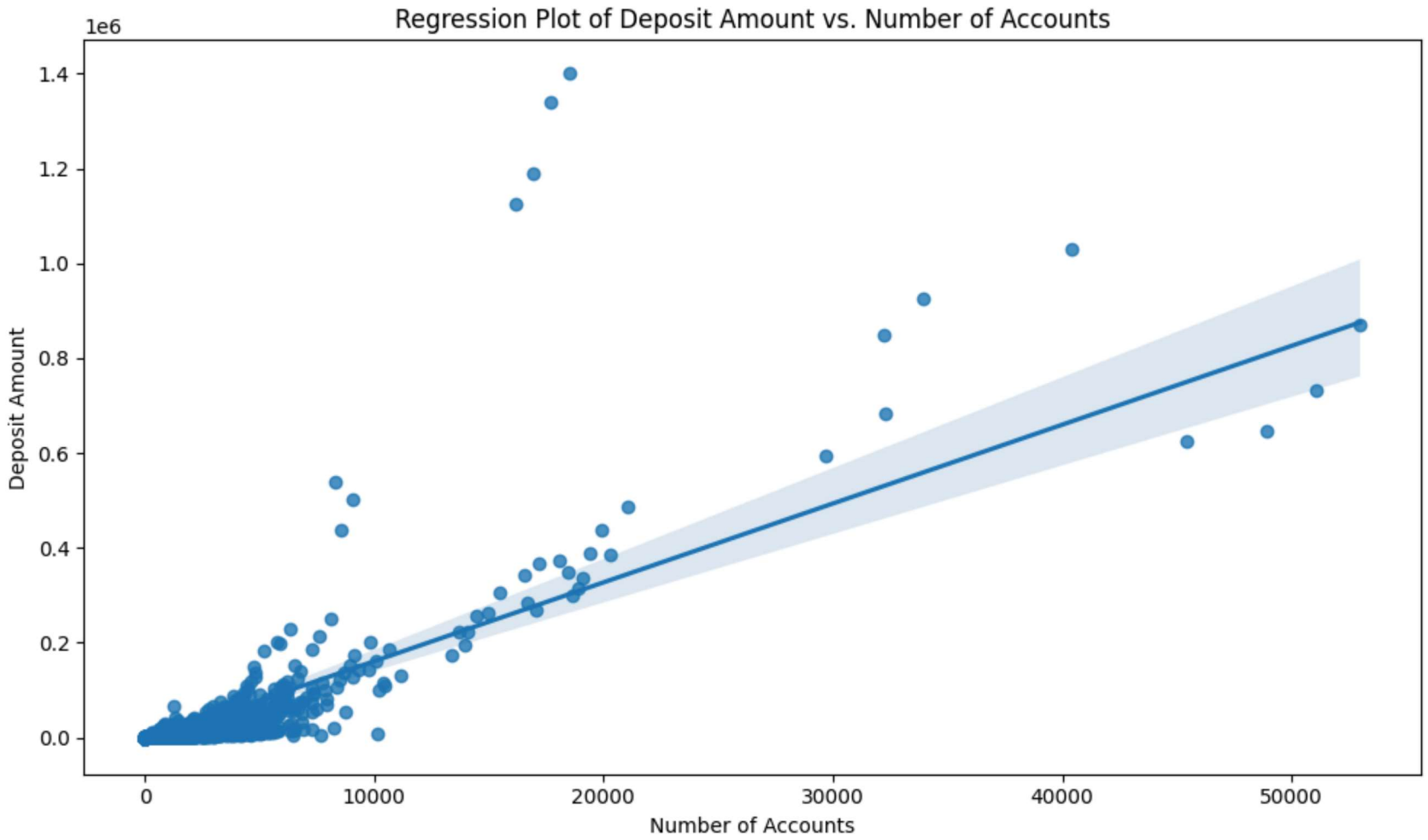
```
import seaborn as sns
import matplotlib.pyplot as plt

# Generate and save a correlation heatmap
plt.figure(figsize=(10, 8))
correlation_matrix = df[['no_of_offices', 'no_of_accounts', 'deposit_amount']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Key Variables')
plt.tight_layout()
plt.savefig('correlation_heatmap.png')
plt.show()
```




```
# Generate and save a scatter plot with a regression line
plt.figure(figsize=(10, 6))
sns.regplot(x='no_of_accounts', y='deposit_amount', data=df)
plt.title('Regression Plot of Deposit Amount vs. Number of Accounts')
plt.xlabel('Number of Accounts')
```

```
plt.ylabel('Deposit Amount')
plt.tight_layout()
plt.savefig('regression_plot.png')
plt.show()
```




```
import statsmodels.formula.api as smf

# Define the model formula
formula = 'deposit_amount ~ no_of_accounts + no_of_offices'

# Fit the linear model to the data
model = smf.ols(formula=formula, data=df).fit()
```

```
# f) Print the results of the model
print("\nLinear Regression Model Summary:")
print(model.summary())

# g) Get the number of observations from the model summary
num_observations = int(model.nobs)
print(f"\n- The regression was run on {num_observations} observations.")

# h) Get the R-squared value and explain what it tells you
r_squared = model.rsquared
print(f"\n- The R-squared of this regression is {r_squared:.4f}.")
print("  This value represents the proportion of the variance in the dependent variable ('deposit_amount') that is predictable from the

# i) Determine if 'size' (no_of_offices and no_of_accounts) is a statistically significant predictor
alpha = 0.05
p_value_offices = model.pvalues['no_of_offices']
p_value_accounts = model.pvalues['no_of_accounts']

print("\n- Statistical Significance of Predictors:")
print(f"  P-value for no_of_offices: {p_value_offices:.4f}")
print(f"  P-value for no_of_accounts: {p_value_accounts:.4f}")

if p_value_offices < alpha:
    print("  'no_of_offices' is a statistically significant predictor.")
else:
    print("  'no_of_offices' is not a statistically significant predictor.")

if p_value_accounts < alpha:
    print("  'no_of_accounts' is a statistically significant predictor.")
else:
    print("  'no_of_accounts' is not a statistically significant predictor.")

# j) Get the regression equation
intercept = model.params['Intercept']
coeff_offices = model.params['no_of_offices']
coeff_accounts = model.params['no_of_accounts']

print("\n- Regression Equation:")
print(f"  deposit_amount = {intercept:.4f} + ({coeff_offices:.4f} * no_of_offices) + ({coeff_accounts:.4f} * no_of_accounts)")
```

```
Linear Regression Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:         deposit_amount   R-squared:                       0.605
```

```
Model:                            OLS   Adj. R-squared:                  0.605
Method:                 Least Squares   F-statistic:                 1.075e+04
Date:                Mon, 22 Sep 2025   Prob (F-statistic):               0.00
Time:                        14:14:50   Log-Likelihood:            -1.5984e+05
No. Observations:               14037   AIC:                         3.197e+05
Df Residuals:                   14034   BIC:                         3.197e+05
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept      -6229.4462    194.684    -31.998      0.000   -6611.053   -5847.840
no_of_accounts    12.1389      0.302     40.198      0.000      11.547      12.731
no_of_offices     72.7416      4.521     16.088      0.000      63.879      81.604
==============================================================================
Omnibus:                    36446.114   Durbin-Watson:                   1.715
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       1104134702.099
Skew:                          29.863   Prob(JB):                         0.00
Kurtosis:                    1375.678   Cond. No.                     1.83e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.83e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

- The regression was run on 14037 observations.

- The R-squared of this regression is 0.6050.
  This value represents the proportion of the variance in the dependent variable ('deposit_amount') that is predictable from the indepen

- Statistical Significance of Predictors:
  P-value for no_of_offices: 0.0000
  P-value for no_of_accounts: 0.0000
  'no_of_offices' is a statistically significant predictor.
  'no_of_accounts' is a statistically significant predictor.

- Regression Equation:
  deposit_amount = -6229.4462 + (72.7416 * no_of_offices) + (12.1389 * no_of_accounts)
```