

CH01_Bank_Data_Python_Supplement

September 3, 2026

1 Chapter 1 — Bank data (Python supplement)

Condensed notebook for the **Python** / **Plotly** portion of *Chapter 1: Bank Data*.

Target graphics:

- Grouped bar chart of deposits by branch and account type (**Figure 1.42**)
- Formatted version with titles and axis labels (**Figure 1.43**)
- Reorganized by account type, then branch (**Figure 1.44**)
- Monday deposits only (**Figure 1.45**)

Data: Bank_data.csv in Data For Condensed Notebooks.

Dependencies:

- pandas
- plotly

1.1 Imports and display options

```
[1]: import pandas as pd
import plotly.express as px
```

```
[2]: # Set output options.
import plotly.io as pio
pio.renderers.default = "pdf+jupyterlab+notebook"
```

1.2 Loading and cleaning the data

```
[3]: csv_path = '../Data For Condensed Notebooks/Bank_data.csv'
df = pd.read_csv(csv_path)
```

```
[4]: df
```

```
[4]:
```

	Date	Weekday	Amount	AcctType	OpenedBy	Branch	Customer
0	1-Nov	Friday	5,000	IRA	New Accts	Central	Existing
1	1-Nov	Friday	14,571	CD	Teller	Central	New
2	1-Nov	Friday	500	Checking	New Accts	Central	Existing
3	1-Nov	Friday	15,000	CD	New Accts	Central	Existing
4	1-Nov	Friday	4,623	Savings	New Accts	North County	Existing

```

..      ...      ...      ...      ...      ...      ...
707  30-Nov  Saturday      7,000      IRA  New Accts  North County  Existing
708  30-Nov  Saturday      4,257      Savings  New Accts      Westside  Existing
709  30-Nov  Saturday        400  Checking      Teller      Central  Existing
710  30-Nov  Saturday     12,673  Checking  New Accts      Westside      New
711  30-Nov  Saturday      4,000  Checking      Teller  North County  Existing

```

[712 rows x 7 columns]

```
[5]: df.rename(columns={' Amount ': 'Amount'}, inplace=True)
```

```
[6]: # Replace commas, then convert Amount to numeric.
df['Amount'] = df['Amount'].str.replace(',', '')
df['Amount'] = pd.to_numeric(df['Amount'])
```

1.3 Bar charts — branch and account type

Basic grouped histogram, then formatted versions with titles and axis labels. ; **Figures 1.43–1.44** (formatted, branch vs. account type on x-axis).

Figure 1.42 (first plot)

```
[7]: fig = px.histogram(df, x='Branch', y='Amount',
                        color='AcctType', barmode='group')
fig.show()
```

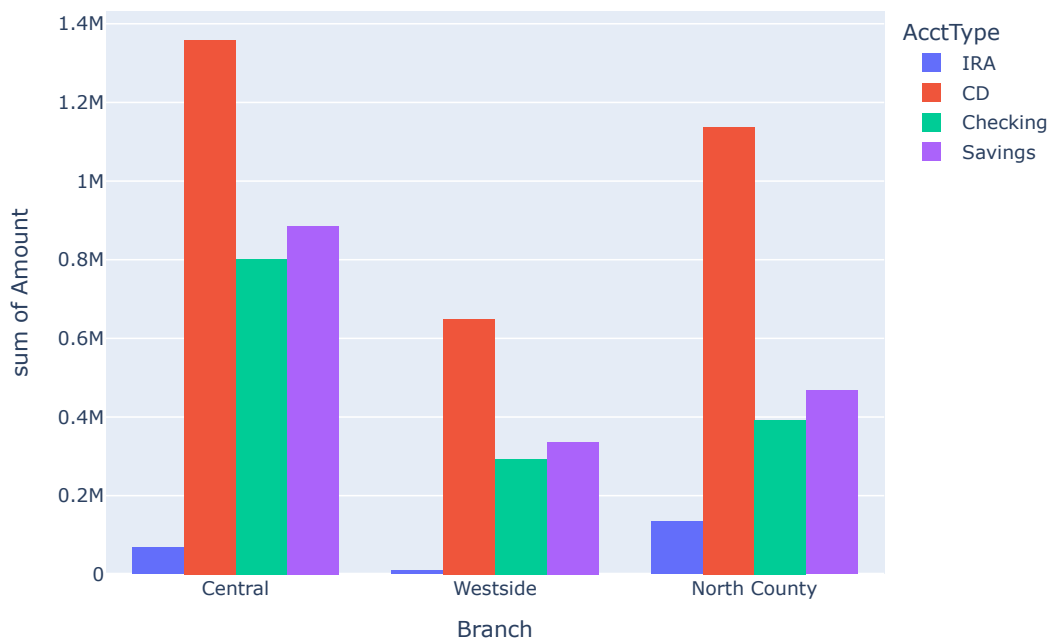


Figure 1.43 (formatted)

```
[8]: fig = px.histogram(df, x='Branch',
                        y='Amount',
                        color='AcctType',
                        barmode='group',
                        labels={'AcctType': 'Account Type'},
                        width=1000)
fig.update_layout(yaxis_title='Total Deposits')
fig.update_layout(title={'text': 'Deposits by Branch and Account Type',
                          'x': 0.5, 'xanchor': 'center'})
fig.show()
```

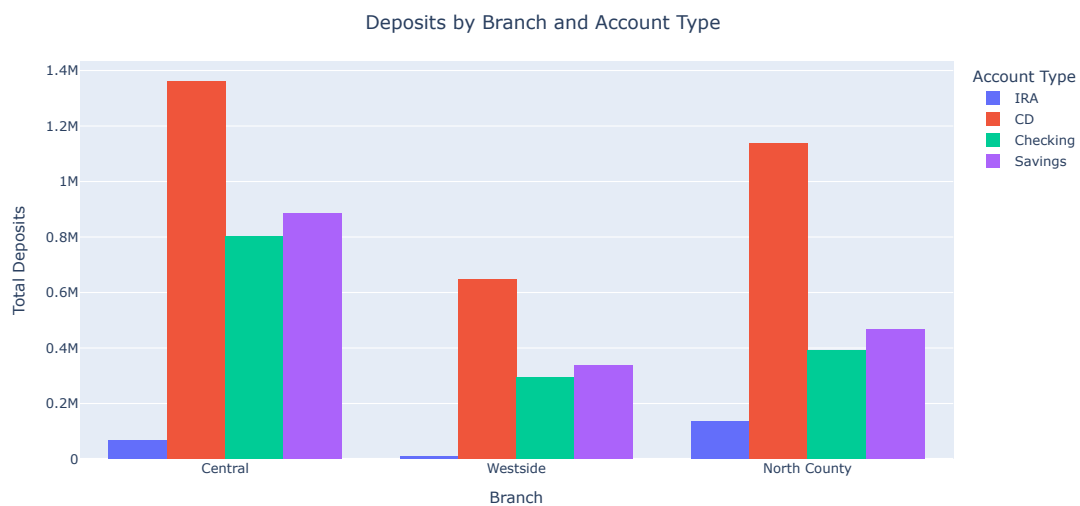
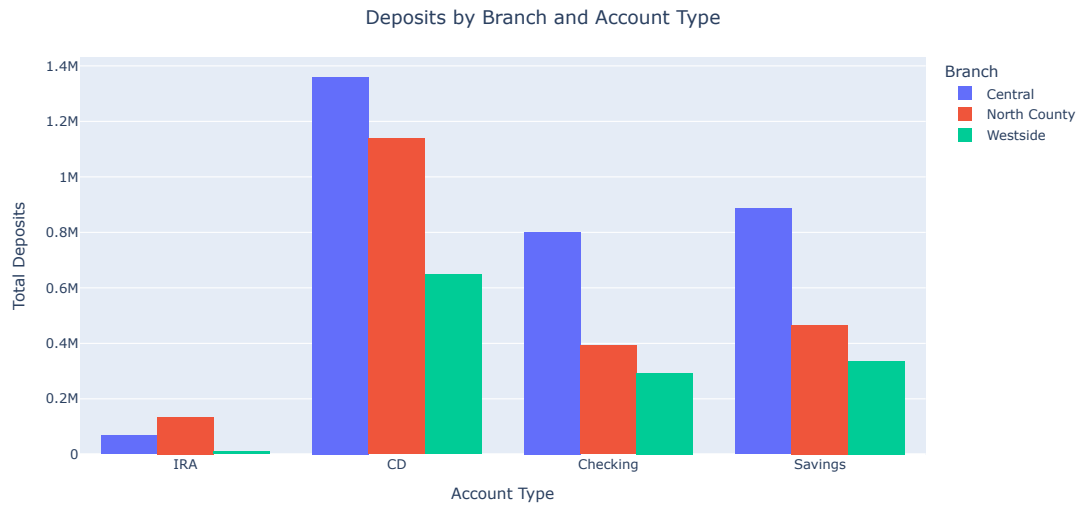


Figure 1.44 (organized by account type, then branch)

```
[9]: fig = px.histogram(df, x='AcctType',
                        y='Amount',
                        color='Branch',
                        barmode='group',
                        labels={'AcctType': 'Account Type'},
                        width=1000)
fig.update_layout(yaxis_title='Total Deposits')
fig.update_layout(title={'text': 'Deposits by Branch and Account Type',
                          'x': 0.5, 'xanchor': 'center'})
fig.show()
```



1.4 Monday deposits only

Filter to `Weekday == 'Monday'` and reuse the account-type layout. **Figure 1.45.**

```
[10]: df_monday = df[df['Weekday'] == 'Monday']
```

```
[11]: fig = px.histogram(df_monday, x='AcctType',
                        y='Amount',
                        color='Branch',
                        barmode='group',
                        labels={'AcctType': 'Account Type'},
                        width=1000)
fig.update_layout(yaxis_title='Total Deposits')
fig.update_layout(title={'text': 'Deposits by Branch and Account Type',
                        'x': 0.5, 'xanchor': 'center'})
fig.show()
```

