✅ **Notebook COMPLETO (SQL + Python) — tudo em 1 lugar**

🔵 **1) Criar a base no SQL**

(mesma TEMP VIEW que você já usa, mas só pra deixar o notebook completo)

```
%sql
CREATE OR REPLACE TEMP VIEW vw_base_mensal_grupos_tratamento AS
SELECT
  mes,
  cliente_id,
  segmento,
  uf,
  tempo_cliente_meses,
  tpv_total_medio,
  saldo_medio_conta,
  tpv_pix,
  receita_pix,
  saldo_conta,
  grupo_experimento,
  usuario_pix,
  participou_promocao,
  aceitou_pgto_pix
FROM vw_base_mensal_enriquecida
WHERE grupo_experimento IS NOT NULL;
```

🔵 **2) Criar a base do DiD (pré + pós) no SQL**

```
%sql
CREATE OR REPLACE TEMP VIEW did_base AS
SELECT
  mes,
  cliente_id,
  CAST(tpv_pix AS DOUBLE) AS tpv_pix,
```

```
  CASE WHEN grupo_experimento = 'tratamento' THEN 1 ELSE 0 END AS treated,
  CASE WHEN mes >= DATE '2024-06-01' THEN 1 ELSE 0 END AS post
FROM vw_base_mensal_grupos_tratamento
WHERE mes IN (DATE '2024-05-01', DATE '2024-06-01');
```

🔵 3) Deduplicar cliente-mês no SQL

```
%sql
CREATE OR REPLACE TEMP VIEW did_base_clean AS
SELECT
  mes,
  cliente_id,
  MAX(treated) AS treated,
  MAX(post) AS post,
  MAX(tpv_pix) AS tpv_pix
FROM did_base
GROUP BY mes, cliente_id;
```

🔵 4) Diferença-em-Diferenças (método clássico) — SQL

```
%sql
WITH grp AS (
  SELECT
    treated,
    post,
    AVG(tpv_pix) AS avg_tpv
  FROM did_base_clean
  GROUP BY treated, post
),
deltas AS (
  SELECT
    treated,
    MAX(CASE WHEN post = 1 THEN avg_tpv END)
```

```
    - MAX(CASE WHEN post = 0 THEN avg_tpv END) AS delta
  FROM grp
  GROUP BY treated
)
SELECT
  MAX(CASE WHEN treated = 1 THEN delta END)
 - MAX(CASE WHEN treated = 0 THEN delta END) AS did_tpv_pix
FROM deltas;
```

Esse é o seu DiD “simples” (o que deu ~2.116).

🔵 5) DiD via Regressão OLS — Python (no mesmo notebook)

```
import pandas as pd
import statsmodels.formula.api as smf

# Puxar a temp view diretamente
df = spark.sql("""
SELECT
  mes,
  cliente_id,
  CAST(tpv_pix AS DOUBLE) AS tpv_pix,
  CASE WHEN grupo_experimento = 'tratamento' THEN 1 ELSE 0 END AS treated,
  CASE WHEN mes >= DATE '2024-06-01' THEN 1 ELSE 0 END AS post
FROM vw_base_mensal_grupos_tratamento
WHERE mes IN (DATE '2024-05-01', DATE '2024-06-01')
""")

pdf = df.toPandas()

# Criar interação do DiD
pdf["did"] = pdf["treated"] * pdf["post"]
```

```
# Regressão OLS com erro-padrão clusterizado por cliente
model = smf.ols("tpv_pix ~ treated + post + did", data=pdf).fit(
    cov_type="cluster",
    cov_kwds={"groups": pdf["cliente_id"]}
)

print(model.summary())
```