COMPANION RESOURCE — APPENDIX B

Technical Specifications and Python Reference

Tested reference code from Chapter 9 and Appendix B — Policy by the Numbers

This appendix consolidates the model's statistical foundation into a practical reference: the data schema your pipeline expects, the model specification in compact form, and a tested, end-to-end Python implementation your technical team can adapt directly.

A.1 Data Schema Reference

Column	Type	Description
student_id	string	Unique student identifier for longitudinal linking
district_id	string	Required only for the hierarchical multi-district model
R	float	Test-score momentum, standardized (Z-score)
A	float	Attendance rate, 0 to 1
I	float	Instructional fidelity, 0 to 1
S	float	Demographic/subgroup factor, 0 to 1
D	int (0/1)	Subgroup indicator for equity-interaction modeling
V	float	Growth Velocity, R at time t minus R at time t-1
Y	int (0/1)	Outcome: 1 = success

Table A.1. Expected columns for the modeling dataset. Standardize R (Z-score) before fitting; A, I, and S arrive as naturally bounded proportions and typically don't require this step.

A.2 Model Specification, Compact Reference

Base model: $\text{logit}(P_i) = \beta 0 + \beta 1R + \beta 2A + \beta 3I + \beta 4S$

Equity-interaction model: $\text{logit}(P) = \beta 0 + \beta 1R + \beta 2A + \beta 3I + \beta 4S + D\cdot[(k_s-1)\beta 1R + (k_s-1)\beta 2A]$

Marginal effect of any predictor x_j: $\partial P/\partial x_j = \beta_j \cdot P(1-P)$

Hierarchical (district random intercept): $\text{logit}(P_{ij}) = \beta 0_j + \beta 1R_{ij} + \ldots$, where $\beta 0_j = \gamma 00 + u0_j$

A.3 Reference Implementation

Setup and normalization:

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.metrics import roc_auc_score, confusion_matrix
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Standardize R (Z-score) — see the caveat in A.1
df['R'] = (df['R_raw'] - df['R_raw'].mean()) / df['R_raw'].std()

# Interaction terms for the equity-adjusted model
df['D_R'] = df['D'] * df['R']
df['D_A'] = df['D'] * df['A']
```

Fitting the base model:

```
X = sm.add_constant(df[['R', 'A', 'I', 'S']])
base_model = sm.GLM(df['Y'], X, family=sm.families.Binomial())
base_result = base_model.fit()
print(base_result.summary())
```

Fitting the equity-interaction model and recovering k_s:

```
X_equity = sm.add_constant(df[['R', 'A', 'I', 'S', 'D_R', 'D_A']])
equity_model = sm.GLM(df['Y'], X_equity, family=sm.families.Binomial())
equity_result = equity_model.fit()

beta_R = equity_result.params['R']
beta_DR = equity_result.params['D_R']
k_s_estimated = 1 + (beta_DR / beta_R)
print(f"Estimated k_s: {k_s_estimated:.3f}")
print(equity_result.pvalues['D_R'])  # Wald test vs. k_s = 1.0
```

Computing exact marginal effects:

```
def marginal_effect(fitted_result, predictor, X):
    beta_j = fitted_result.params[predictor]
    p_hat = fitted_result.predict(X)
    return beta_j * p_hat * (1 - p_hat)
```

```
df['marginal_effect_A'] = marginal_effect(base_result, 'A', X)
print(df['marginal_effect_A'].describe())
```

Hierarchical model for multi-district data:

```
hierarchical_model = smf.mixedlm(
    "Y ~ R + A + I + S",
    data=df,
    groups=df["district_id"],
    re_formula="~1"
)
hierarchical_result = hierarchical_model.fit()
print(hierarchical_result.summary())
district_intercepts = hierarchical_result.random_effects
```

Minimum sample note: hierarchical models require enough districts (Level-2 units) to estimate the variance component τ^2 reliably — at least 5–10 districts, as a rule of thumb.

Diagnostics:

```
p_hat = base_result.predict(X)
auc = roc_auc_score(df['Y'], p_hat)

pred_class = (p_hat >= 0.5).astype(int)
cm = confusion_matrix(df['Y'], pred_class)
tn, fp, fn, tp = cm.ravel()
sensitivity = tp / (tp + fn)
specificity = tn / (tn + fp)

vif_data = pd.DataFrame()
vif_data['variable'] = ['R', 'A', 'I', 'S']
vif_data['VIF'] = [variance_inflation_factor(
    df[['R','A','I','S']].values, i) for i in range(4)]
```

A.4 Quick-Reference: Concept-to-Code Map

Book Concept	Chapter	Code Reference
Base model estimation (MLE via IRLS)	1, 9	A.3 — base model
Marginal effects ($\beta \cdot P(1-P)$)	9 §9.3	A.3 — marginal_effect()
Support Coefficient (k_s) as interaction	4, 7, 9 §9.4	A.3 — equity model
District-level partial pooling / GLMM	8, 9 §9.6	A.3 — hierarchical model
AUC, sensitivity, specificity	9 §9.7, 10	A.3 — diagnostics
Multicollinearity (VIF)	9 §9.7	A.3 — diagnostics

DATA PROVENANCE NOTE *The modeling approach documented here and in Chapter 9 was developed and internally validated against a real student dataset. That dataset is not published anywhere in this book and cannot be shared publicly, due to student privacy protections and internal data-sharing agreements. Every worked example in this book is an illustrative reconstruction.*
