Module 7: Regression for health outcomes

Data analysis with Python for health specialists

Yaé Ulrich Gaba

2026

AIRINA Labs

From correlation to prediction

- Chapter 6: “Is there a *difference*?” (hypothesis testing)
- Chapter 7: “Can we *predict* one variable from another?”

Regression is the workhorse of clinical research:

- Every epidemiological study
- Every clinical trial analysis
- Every risk prediction model

Key phrase: “after adjusting for” = multiple regression.

Simple linear regression

Q: How does systolic BP change with age?

```
from scipy import stats

slope, intercept, r, p, se = stats.linregress(
    df["age"], df["systolic_bp"])
print(f"SBP = {intercept:.1f} + {slope:.2f} x Age")
print(f"R-squared: {r**2:.3f}")
```

Slope = 0.7 means: on average, SBP increases by **0.7 mmHg per year of age**.

Caution
This does *not* mean aging *causes* higher BP. It means the two are linearly associated in this sample.

Multiple linear regression

Q: Predict SBP from age, BMI, and smoking *simultaneously*.

```
import statsmodels.api as sm

X = sm.add_constant(df[["age", "bmi", "smoking"]])
model = sm.OLS(df["systolic_bp"], X).fit()
print(model.summary())
```

Interpreting coefficients:

- Age coef $\approx$ 0.6: +0.6 mmHg per year, *after adjusting for BMI and smoking*
- BMI coef $\approx$ 0.8: +0.8 mmHg per BMI unit, *after adjusting for age and smoking*
- Smoking coef $\approx$ 5: smokers have 5 mmHg higher SBP, *after adjusting for age and BMI*

Confounders: the coffee trap

```
# Unadjusted: coffee appears to predict heart disease
r, p = stats.pearsonr(coffee, heart_disease_risk)
# r = 0.35, p < 0.001 -- significant!

# Adjusted: coffee effect disappears when controlling for smoking
X = sm.add_constant(df[["coffee", "smoking"]])
model = sm.OLS(df["hd_risk"], X).fit()
# Coffee p = 0.72 (not significant)
# Smoking p < 0.001 (the real cause)
```

Lesson

Without adjusting for smoking, you would wrongly conclude coffee increases heart disease risk. Always think about confounders in observational studies.

Checking regression assumptions

```
fig, axes = plt.subplots(1, 3, figsize=(15, 4))

# 1. Residuals vs fitted (linearity + homoscedasticity)
axes[0].scatter(model.fittedvalues, model.resid, alpha=0.4)
axes[0].axhline(0, color="red", linestyle="--")

# 2. Histogram of residuals (normality)
axes[1].hist(model.resid, bins=25, edgecolor="black")

# 3. Q-Q plot (normality)
stats.probplot(model.resid, dist="norm", plot=axes[2])
```

4 assumptions: linearity, independence, homoscedasticity, normality of residuals.

Logistic regression: binary outcomes

Q: Predict diabetes (yes/no) from age, BMI, and glucose.

```
# statsmodels (for odds ratios and p-values)
X = sm.add_constant(pima[["age", "bmi", "glucose"]])
logit = sm.Logit(pima["outcome"], X).fit()
print(logit.summary())

# Odds ratios with 95% CI
import numpy as np
or_df = pd.DataFrame({
    "OR": np.exp(logit.params),
    "CI_low": np.exp(logit.conf_int()[0]),
    "CI_high": np.exp(logit.conf_int()[1]),
    "p": logit.pvalues
}).round(4)
```

Interpreting odds ratios

Variable	OR	Interpretation
Glucose	1.04	+1 mg/dL $\rightarrow$ 4% higher odds of diabetes
BMI	1.08	+1 kg/m^2 $\rightarrow$ 8% higher odds
Age	1.02	+1 year $\rightarrow$ 2% higher odds

- OR > 1: higher risk
- OR < 1: lower risk (protective)
- OR $= 1$: no association

All holding other variables constant.

Confusion matrix: evaluating the model

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

sensitivity = tp / (tp + fn)  # recall
specificity = tn / (tn + fp)
ppv = tp / (tp + fp)          # precision
npv = tn / (tn + fn)
```

Clinical priority

Screening test: prioritize high sensitivity (don't miss cases).

Confirmatory test: prioritize high specificity (don't give false alarms).

Survival analysis: Cox model

```
from lifelines import CoxPHFitter

cph = CoxPHFitter()
cph.fit(surv_df, duration_col="time",
        event_col="event", formula="treatment")
cph.print_summary()

hr = np.exp(cph.params_["treatment"])
print(f"Hazard ratio: {hr:.3f}")
# HR = 0.65 -> treatment reduces hazard by 35%
```

Hazard ratios are the standard in oncology trials, cardiovascular studies, and any time-to-event endpoint.

What you can do after this module

1. Fit simple and multiple linear regression with clinical interpretation
2. Identify and control for confounders
3. Fit logistic regression and interpret odds ratios
4. Evaluate classification with sensitivity, specificity, PPV, NPV
5. Run survival analysis with Kaplan-Meier and Cox models

Next: Module 8 — Machine learning for clinical prediction

Exercises (Hour 3)

1. **Simple logistic:** Predict diabetes from glucose only; report OR
2. **Multiple logistic:** Add age, BMI, BP, pedigree; report all ORs with 95% CIs
3. **Thresholds:** Try classification thresholds of 0.3, 0.5, 0.7; compare sensitivity/specificity
4. **Mini-project:** Build a complete diabetes risk prediction notebook with confusion matrix and clinical interpretation