Regression & Prediction

Theory and Practice with House Prices

Your Name

February 23, 2025

xΑΙ

Our Housing Journey

- Starting Simple:
 Foundations
- Expanding the Scope: Complexity
- Refining Precision:
 Optimization

- Facing Challenges: Pitfalls
- Insights & Next Steps: Applications

Focus

Unpacking King County house prices with rich theory and dual $\ensuremath{\mathsf{R}}/\ensuremath{\mathsf{Python}}$ implementations

Starting Simple

The Housing Puzzle

- Objective: Decode drivers of house prices in King County—size, location, features
- Regression: A statistical lens linking price (Y) to predictors like size (X)
- Purpose: Explain historical sales patterns and forecast future values for buyers and assessors

Simple Linear Regression: Theory & R

• Theory (p. 141): Models a straight-line relationship:

$$Y = b_0 + b_1 X + e$$

- b₀: Base price when size is zero;
 b₁: Price increase per sq ft; e:
 Random error
- Assumes linearity and
- independence—foundation of regression

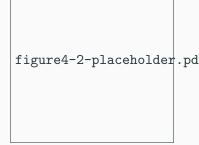


Figure 1: Price vs. Size Fit



Simple Linear Regression: Python

- Practice: Fits price to living space, revealing size's impact
- Key Insight: Positive slope shows larger homes fetch higher prices

```
# Python (p. 152
      adapted)
from sklearn.
     linear_
     model
     import
     LinearRegression
predictors = ['
     SqFtTotLiving
outcome = '
     AdjSalePrice
simple_lm =
     LinearRegression
simple_lm.fit(
     house
     predictors
```

3

5

Size as a core driver

Least Squares Theory

- Theory (p. 148): Finds the line minimizing residual sum of squares: $\sum_{i=1}^{n} (Y_i \hat{Y}_i)^2$
- How: Adjusts b₀ and b₁ to reduce prediction errors—optimal for linear fits
- History: Legendre (1805) and Gauss; computationally efficient but outlier-sensitive in small datasets

Expanding the Scope

Multiple Linear Regression: Theory & R

• **Theory (p. 150)**: Extends to multiple predictors:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + e$$

- Power: Captures combined effects—size, lot, bedrooms—assuming linearity
- Use: Explains complex housing dynamics



• Size adds \$229/sq ft

7

Multiple Linear Regression: Python

- Practice: Models housing with multiple factors
- Key Insight: Negative bedroom coef suggests smaller rooms hurt value



 Bedrooms vs. size tension

Factor Variables: Theory & R

- Theory (p. 163): Encodes categorical variables (e.g., property type) as binary dummies
- Why: Allows non-numeric factors in regression; compares to a reference level
- **Example**: Single-family vs. Townhouse effects

Type shifts price

Factor Variables: Python

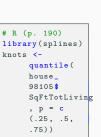
- Practice: Integrates property type into price model
- Key Insight: Townhouses may differ from single-family homes

```
# Python (p. 166
      adapted)
import pandas as
      pd
X = pd.get_
     dummies (
     house['
     PropertyType
     '], drop_
     first=True
# Drops first
     level (e.g
     Multiplex)
       as
      reference
```

• Baseline comparison

Nonlinear Fit: Theory & Splines in R

- Theory (p. 187): Nonlinear via polynomial segments joined at knots
- Why: Captures diminishing returns-e.g., small homes gain more per sq ft
- Advantage: Flexible fit without overfitting like high-order polynomials



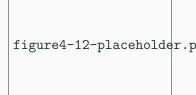
R (p. 190)

quantile (house_ 98105

p = c(.25, .5,

.75))

knots <-



11

Nonlinear Fit: Splines in Python

- Practice: Fits nonlinear price trends in zip 98105
- Key Insight: Better matches small vs. large home value shifts

```
# Python (p.
                                             190)
                                       import
                                             statsmodels
                                             .formula.
                                             api as smf
                                       formula = '
3
                                             AdjSalePrice
                                             u~ubs(
                                             SqFtTotLiving
                                             ,,,df=6,,,
                                             degree=3)
                                             +uSqFtLotu
                                             \pm_{11}
                                             Bathrooms
                                             +..Bedrooms
                                             0+0
                                             BldgGrade'
                                       model_spline =
                                             smf.ols(
```

Curves reflect reality

Refining Precision

Model Assessment: Theory

- Theory (p. 153): Measures prediction quality and fit
- **RMSE**: $\sqrt{\frac{\sum (y_i \hat{y}_i)^2}{n}}$ —average error magnitude
- R²: Proportion of variance explained (0-1); higher means better fit
- **Use**: Guides housing prediction accuracy

Cross-Validation: Theory

- Theory (p. 155): Validates model on unseen data via k-fold splits
- **Process**: Divide data, train on k-1, test on 1, repeat, average RMSE
- Why: Ensures predictions generalize beyond training sales—crucial for real estate

Model Selection: Theory & R

- Theory (p. 156): Balances fit vs. complexity—Occam's razor
- AIC: $2P + n \log(RSS/n)$ —penalizes extra predictors
- Goal: Optimal housing model without overkill



 Streamlined predictors

Model Selection: Python

- Practice: Automates predictor choice for housing
- Key Insight: Reduces noise, enhances prediction

```
# Python (p. 158
                                            adapted)
                                      from dmba import
                                            stepwise_
                                           selection
3
                                      predictors = ['
                                           SqFtTotLiving
                                           SqFtLot',
                                           'Bathrooms
                                           , ,
                                           Bedrooms',
                                           BldgGrade'
                                      def train model (
                                           vars):
                                     model =
                                           LinearRegression
```

• Focused fit

Weighted Regression: Theory & R

- Theory (p. 159): Weights adjust influence by reliability
- Why: Older sales less relevant—recent data gets priority
- Impact: Refines coefficients for current market



 Recent sales emphasized

Weighted Regression: Python

- Practice: Weights tune housing model
- **Key Insight**: Aligns predictions with market trends

```
# Python (p.
                                            160)
                                      house['Weight']
                                            = [int(
                                            date.split
                                            ('-') [O])
                                            for date
                                            in house.
                                            DocumentDate
                                           1 - 2005
                                      house_wt =
                                            LinearRegression
                                      house_wt.fit(
4
                                            house[
                                            predictors
                                           1. house[
                                            outcome],
                                            sample_
                                            weight=
```

• Fresher focus

Facing Challenges

Prediction Limits: Theory

- Theory (p. 161): Extrapolation beyond data fails—e.g., empty lots
- Intervals: Confidence for b_i , wider prediction for \hat{Y}_i
- Why: Uncertainty spikes outside training range—limits housing forecasts

Interpreting Coefficients: Theory

- Theory (p. 171): Coefficients mislead if predictors correlate
- Multicollinearity: Size and bedrooms overlap—unstable fits
- Confounding: Missing location skews results
- Interactions: Size's effect varies by zip—needs modeling

Diagnostics: Theory & R

- Theory (p. 176): Residuals reveal model flaws
- Outliers: Extreme sales (e.g.,
- **Heteroskedasticity**: Uneven errors signal gaps



R (p. 177) house 98105 <house[house \$

98105,]

Bathrooms

BldgGrade,

Bedrooms +

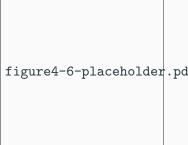


Figure 3: Influence Plot

21

Diagnostics: Python

- Practice: Identifies \$119,748 as partial sale anomaly
- Key Insight: Diagnostics ensure robust housing predictions

```
# Python (p.
                                           178)
                                      from statsmodels
                                           .stats.
                                           outliers_
                                           influence
                                           import
                                           OLSInfluence
                                      house 98105 =
                                           house[
                                           house['
                                           ZipCode']
                                           == 981057
4
                                      model = smf.ols(
                                           AdjSalePrice
                                           n~n
                                           SqFtTotLiving
                                           ⊔+⊔SqFtLot
```

Spots critical flaws

Insights & Next Steps

Housing Insights

- Findings: Linear ties price to size, splines capture nonlinear trends
- **Diagnostics**: Reveal quirks like partial sales—critical for accuracy
- Application: Real-world tool for buyers, sellers, and assessors

Key Takeaways

- Flexible: Evolves from simple lines to complex curves for housing
- Precise: RMSE and cross-validation ensure reliable price predictions
- Powerful: R and Python implementations unlock data-driven insights

Looking Ahead

- Resources: Statistical Learning (Hastie et al.), Time Series Forecasting (Shmueli)
- Next Steps: Dive into splines, time series for dynamic housing models
- Call: Blend theory and practice for smarter predictions