DataSci 347

Midterm Examination

PRACTICE TEST

CRITICAL INFORMATION

The actual midterm exam will use the **SAME DATASET**!

The exam will test the **SAME CORE SKILLS**!

However, the specific questions will be different.

This practice test prepares you for the **types** of analyses you'll need to perform, but the actual exam will ask different specific questions using this dataset.

Exam Day Instructions

- You will complete the exam **in class** on your laptop
- You may use **any Python environment** (Jupyter, Colab, VS Code, PyCharm, Spyder, etc.)
- Ensure your Python environment has: `pandas`, `numpy`, `matplotlib`, `seaborn`, `statsmodels`, `sklearn`
- When finished, upload your code file (.py or .ipynb) to **Canvas**
- Bring your laptop **charged** with software already installed

Study Strategy

- Work through this practice test **completely**
- Understand **why** you're doing each step, not just **how**
- Practice interpreting coefficients and p-values
- Review model assumptions and diagnostics
- Master LASSO variable selection with cross-validation

Contents

1 Learning Objectives

By completing this practice test, you will demonstrate your ability to:

1. **Perform Exploratory Data Analysis (EDA)**
   - Create and interpret summary statistics by groups
   - Generate appropriate visualizations (histograms, boxplots, scatterplots)
   - Describe patterns and variability in data
2. **Build and Interpret Linear Regression Models**
   - Fit simple and multiple linear regression models
   - Interpret coefficients in context
   - Test for statistical significance
   - Compare models and explain differences
3. **Work with Categorical Predictors**
   - Include categorical variables (like State) in regression models
   - Interpret dummy variable coefficients
   - Make predictions for specific factor levels
4. **Assess Model Assumptions**
   - Create and interpret residual plots
   - Check normality assumptions with Q-Q plots
   - Diagnose potential violations
5. **Perform Variable Selection**
   - Use LASSO regression with cross-validation
   - Select optimal regularization parameters
   - Build parsimonious final models
6. **Communicate Statistical Findings**
   - Write concise interpretations
   - Present results clearly and accurately
   - Draw appropriate conclusions

2 Background

The outbreak of the novel Corona virus disease 2019 (COVID-19) was declared a public health emergency of international concern by the World Health Organization (WHO) on January 30, 2020. Upwards of 112 million cases have been confirmed worldwide, with nearly 2.5 million associated deaths. Within the US alone, there have been over 500,000 deaths and upwards of 28 million cases reported.

Governments around the world have implemented and suggested a number of policies to lessen the spread of the pandemic, including mask-wearing requirements, travel restrictions, business and school closures, and even stay-at-home orders. The global pandemic has impacted the lives of individuals in countless ways, and though many countries have begun vaccinating individuals, the long-term impact of the virus remains unclear.

The impact of COVID-19 on a given segment of the population appears to vary drastically based on the socioeconomic characteristics of the segment. In particular, differing rates of infection and fatalities have been reported among different racial groups, age groups, and socioeconomic groups. One of the most important metrics for determining the impact of the pandemic is the **death rate**, which is the proportion of people within the total population that die due to the disease.

Research Questions

There are two main goals for this case study:

1. Number of deaths vary drastically across State. We want to find out how State relates to the death rate.
2. COVID-19 seems to target elder people's lives. Is there evidence in our data to show that proportion of elder people indeed relates to the death at county level?

3 Data Preparation

To make our case study here simple and manageable in a timely fashion, we have assembled a subset of data called: `covid_county_midterm.csv`. It collects county level death rate, labeled as `log_death_rate`, as well as a subset of demographic information based on two cleaned datasets:

- `covid_county.csv`: County-level socioeconomic information that combines 4 datasets: Income (Poverty level and household income), Jobs (Employment type, rate, and change), People (Population size, density, education level, race, age, household size, and migration rates), County Classifications
- `covid_rates.csv`: Daily cumulative numbers on infection and fatality for each county

3.1 Death Rate Calculation

Understanding the Response Variable:

What is a good way to measure COVID death rate? There are quite a number of counties with a very low or no number of deaths. We have created a new measurement of death rate as follows:

- The total number of deaths for each county is gathered by November 1st, 2020

- The death rate is calculated as:

$$\text{death\_rate} = \frac{\text{deaths} + 1}{\text{population} + 2}$$

- We then apply the logarithm: `log_death_rate` $= \log(\text{death\_rate})$

Why this transformation?

- Adding 1 to deaths avoids $\log(0)$ for counties with no deaths
- Log transformation helps normalize the distribution
- Coefficients can be interpreted as percentage changes

3.2 Read and Prepare Data

We are ready to read the data `covid_county_midterm.csv` into Python. To simplify the analyses further, we created a subset called `covid_county_sub` for us to use in the entire case study.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import Lasso, LassoCV
from sklearn.preprocessing import StandardScaler
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings('ignore')

# Set visualization style
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

# Read the data
covid_county = pd.read_csv("covid_county_midterm.csv")

# Create subset with relevant variables
covid_county_sub = covid_county[['log_death_rate', 'State',
                                 'Deep_Pov_All', 'PovertyAllAgesPct',
                                 'PerCapitaInc', 'UnempRate2020',
                                 'PctEmpFIRE', 'PctEmpServices',
                                 'PopDensity2010', 'Age65AndOlderPct2010'
    ,
                                 'WhiteNonHispanicPct2010',
                                 'HiCreativeClass2000']].copy()

# Display basic information
print(f"Dataset shape: {covid_county_sub.shape}")
print(f"\nMissing values:\n{covid_county_sub.isnull().sum()}")
print(f"\nBasic statistics:\n{covid_county_sub.describe()}")
```

Listing 1: Data Loading and Preparation in Python

Pedagogical Note: Always start by understanding your data:

- Check dimensions (rows, columns)
- Identify missing values
- Review summary statistics
- Understand variable types (numeric, categorical)

4 Exploratory Data Analysis (EDA)

During the course of pandemic, we have witnessed that many policies are carried out at state level. For example, when to reopen after the March lockdown. To see the variability of death rates among states, we will conduct thorough exploratory analysis.

Question 1: State-Level Death Rate Analysis (3 parts)

Skills tested: Data aggregation, visualization, descriptive statistics

a) Create the **median** `log_death_rate` by State. Show a **sorted bar plot** (not histogram) of the median `log_death_rate` by State, displaying only the top 15 states. Use no more than three sentences to summarize what this plot reveals about variation across states.

Hint: Use `groupby()` to calculate medians, then `sort_values()` and plot with `plt.barh()` or `sns.barplot()`.

b) Create a violin plot of `log_death_rate` by State (show all states or a meaningful subset). Use no more than two sentences to describe what the violin plot reveals about the **distribution shape** within states that the boxplot might not show.

Hint: `sns.violinplot()` shows the full distribution density.

c) Calculate and report the **coefficient of variation** (CV) of the average `log_death_rate` across states. What does this CV value tell you about the relative variability? (1-2 sentences)

Hint: $CV = \frac{\text{standard deviation}}{\text{mean}}$

Learning Point: Different visualizations reveal different aspects of data:

- **Histograms**: Show distribution of a single variable
- **Boxplots**: Show median, quartiles, and outliers by group
- **Violin plots**: Show full distribution shape by group
- **Bar plots**: Compare summary statistics across categories

Choose the visualization that best answers your research question!

5 Statistical Analyses

In the following analyses, we try to find out factors related to `log_death_rate`.

5.1 Model 1 (fit1): Poverty and log_death_rate

While age is often discussed, socioeconomic factors like poverty also play a crucial role. Let us start with a simple regression of `log_death_rate` vs. `PovertyAllAgesPct`.

```
# Prepare data (remove missing values)
data_fit1 = covid_county_sub[['log_death_rate',
                              'PovertyAllAgesPct']].dropna()

X = data_fit1[['PovertyAllAgesPct']]
y = data_fit1['log_death_rate']

# Add constant for intercept
X_with_const = sm.add_constant(X)

# Fit model
fit1 = sm.OLS(y, X_with_const).fit()
print(fit1.summary())

# Visualize the relationship
plt.figure(figsize=(10, 6))
plt.scatter(X, y, alpha=0.5)
plt.plot(X, fit1.predict(X_with_const), color='red', linewidth=2)
plt.xlabel('Poverty Rate (All Ages %)')
plt.ylabel('Log Death Rate')
plt.title('Simple Linear Regression: Poverty vs Log Death Rate')
plt.show()
```

Listing 2: Simple Linear Regression: Poverty

Question 2: Simple Regression (2 parts)

Skills tested: Hypothesis testing, coefficient interpretation

a) Is `PovertyAllAgesPct` a significant variable at .05 level in this analysis? Report the p-value and the coefficient. Interpret the coefficient in context.

Hint: A coefficient of 0.05 means for every 1% increase in poverty rate, log_death_rate increases by 0.05.

b) Report the R-squared value. What percentage of variability in `log_death_rate` is explained by `PovertyAllAgesPct` alone?

5.2 Model 2 (fit2): Poverty and log_death_rate controlling for State

Question 3: Multiple Regression with Categorical Variable (5 parts)

Skills tested: Multiple regression, categorical predictors, model comparison, prediction

How do `PovertyAllAgesPct` and State collectively affect `log_death_rate`? In `fit2`, run a linear model of `log_death_rate` vs State and `PovertyAllAgesPct` (without interactions).

a) Is `PovertyAllAgesPct` significant at .05 level in this model? Report the coefficient and p-value.

b) How do you interpret the coefficient of `PovertyAllAgesPct` on the `log_death_rate` in `fit2`? Compare and explain: Why is the effect of `PovertyAllAgesPct` different in `fit1` vs `fit2`? (No more than 4 sentences).

Key concept: This addresses **confounding** and **omitted variable bias**.

c) Perform an F-test to determine if State is significant in this model at .05 level. Report the F-statistic and p-value.

Hint: Compare `fit2` (with State) to `fit1` (without State) using `anova_lm()`.

d) Report the R-squared values for both `fit1` and `fit2`. How much additional variance is explained by adding State to the model?

e) Create a residual plot (residuals vs fitted values) and a Q-Q plot for `fit2`. Based on these plots, are the linear model assumptions reasonably met? Specifically comment on:

- Linearity
- Homoscedasticity (constant variance)
- Normality of residuals

Use no more than three sentences.

```
# Prepare data
data_fit2 = covid_county_sub[['log_death_rate', 'State',
                              'PovertyAllAgesPct']].dropna()

# Create dummy variables for State
X_fit2 = pd.get_dummies(data_fit2[['State', 'PovertyAllAgesPct']],
                        columns=['State'], drop_first=True)
y_fit2 = data_fit2['log_death_rate']

# Add constant
X_fit2_const = sm.add_constant(X_fit2)

# Fit model
```

```
fit2 = sm.OLS(y_fit2, X_fit2_const).fit()
print(fit2.summary())

# F-test: Compare fit2 to fit1
from statsmodels.stats.anova import anova_lm
print("\nF-test for State significance:")
print(anova_lm(fit1, fit2))

# Model diagnostics
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Residual plot
axes[0].scatter(fit2.fittedvalues, fit2.resid, alpha=0.5)
axes[0].axhline(y=0, color='r', linestyle='--')
axes[0].set_xlabel('Fitted Values')
axes[0].set_ylabel('Residuals')
axes[0].set_title('Residuals vs Fitted Values')

# Q-Q plot
stats.probplot(fit2.resid, dist="norm", plot=axes[1])
axes[1].set_title('Normal Q-Q Plot')

plt.tight_layout()
plt.show()
```

Listing 3: Multiple Linear Regression with State

Understanding Confounding:

When we add State to the model, the coefficient of Poverty may change because:

- Some states have higher poverty rates AND higher death rates for other reasons
- By controlling for State, we isolate the effect of poverty *within* states
- This reveals the true relationship between poverty and death rate

This is why **multiple regression** is more powerful than simple regression!

5.3 Final Model (fit.final)

In this section, using all possible variables available in `covid_county_sub`, we will build a final parsimonious model to identify a set of important variables that are related to the `log_death_rate`. We will not fine-tune the final model.

As you have seen, State effect explains a large portion of variability in `log_death_rate`, we will lock State in all the analyses.

Question 4: Variable Selection with LASSO (5 parts)

Skills tested: LASSO regression, cross-validation, final model interpretation

Important Remark: You are going to run LASSO to pick up a few variables in addition to State. In case you can't get LASSO to work, go to part b) directly and use the following set of variables to get your `fit.final`: State, `PovertyAllAgesPct`, `PopDensity2010`, `Age65AndOlderPct2010`, `UnempRate2020`. (Note: this is not necessarily the LASSO output.)

a) Use LASSO with cross-validation to pick up a few variables in addition to State. List all variables with non-zero coefficients from your LASSO output. To be specific, let us control the following settings to get consistent results:

- Use `random_state=42` to control the cross-validation splits
- Use 10-fold cross validations
- Force State in all the LASSO models (by not penalizing State dummies)
- Pick the final set of variables using `alpha` corresponding to minimum cross-validation error (not 1se for this practice)

Hint: You'll need to use `LassoCV` and carefully handle the State dummy variables.

b) Run a final model `fit.final` of `log_death_rate` vs State and the set of variables obtained from your LASSO output. Also include `Age65AndOlderPct2010` regardless of whether it is in your LASSO output or not (since our research question focuses on age). Report the summary of `fit.final`.

c) Is State significant at .05 level in this model? Is `Age65AndOlderPct2010` significant at .05 level in this model? Report both p-values.

d) Among all the **continuous variables** in `fit.final` (excluding State), which variable has the largest absolute coefficient? What does this suggest about its relative importance? (2-3 sentences)

Note: Compare standardized coefficients for fair comparison across different scales.

e) Assume all linear model assumptions are met. Write a brief summary of your findings based on `fit.final`. Your summary should address:

- Which variables are significantly associated with log death rate?
- What does this tell us about COVID-19 impact across counties?
- Answer the original research question about elderly population

(No more than 5 sentences)

```python
from sklearn.linear_model import LassoCV
from sklearn.preprocessing import StandardScaler

# Prepare full data
data_full = covid_county_sub.dropna()

# Separate State and continuous variables
continuous_vars = ['Deep_Pov_All', 'PovertyAllAgesPct', 'PerCapitaInc',
                   'UnempRate2020', 'PctEmpFIRE', 'PctEmpServices',
                   'PopDensity2010', 'Age65AndOlderPct2010',
                   'WhiteNonHispanicPct2010', 'HiCreativeClass2000']

# Create dummy variables for State
state_dummies = pd.get_dummies(data_full['State'], prefix='State',
                               drop_first=True)

# Combine continuous variables and state dummies
X_full = pd.concat([data_full[continuous_vars], state_dummies], axis=1)
y_full = data_full['log_death_rate']

# Standardize ONLY continuous variables (not state dummies)
scaler = StandardScaler()
X_full[continuous_vars] = scaler.fit_transform(X_full[continuous_vars])

# Create penalty weights: 0 for State dummies (don't penalize), 1 for
    others
n_continuous = len(continuous_vars)
n_states = state_dummies.shape[1]
# Unfortunately, sklearn's Lasso doesn't support per-feature penalties
    easily
# Alternative approach: Fit LASSO on continuous vars, then add State

# Simpler approach for this exercise:
# Fit LASSO on continuous variables
lasso_cv = LassoCV(cv=10, random_state=42, max_iter=10000, n_alphas=100)
lasso_cv.fit(data_full[continuous_vars], y_full)

# Get selected variables (non-zero coefficients)
selected_vars = [continuous_vars[i] for i, coef in enumerate(lasso_cv.
    coef_)
                 if abs(coef) > 1e-10]
print("Selected continuous variables:", selected_vars)
print(f"Optimal alpha: {lasso_cv.alpha_}")

# Now build final model with State + selected variables +
    Age65AndOlderPct2010
final_vars = selected_vars.copy()
if 'Age65AndOlderPct2010' not in final_vars:
    final_vars.append('Age65AndOlderPct2010')

print(f"\nFinal model variables: State + {final_vars}")
```

Listing 4: LASSO Regression for Variable Selection

```python
# Prepare data for final model
data_final = covid_county_sub[['log_death_rate', 'State'] +
                              final_vars].dropna()

# Create design matrix
X_final = pd.get_dummies(data_final[['State'] + final_vars],
                         columns=['State'], drop_first=True)
y_final = data_final['log_death_rate']

# Add constant
X_final_const = sm.add_constant(X_final)

# Fit final model
fit_final = sm.OLS(y_final, X_final_const).fit()
print(fit_final.summary())

# For standardized coefficients comparison
X_final_std = X_final.copy()
# Standardize only continuous variables
for var in final_vars:
    if var in X_final_std.columns:
        X_final_std[var] = (X_final_std[var] - X_final_std[var].mean()) /
   X_final_std[var].std()

X_final_std_const = sm.add_constant(X_final_std)
fit_final_std = sm.OLS(y_final, X_final_std_const).fit()
print("\n=== Standardized Coefficients ===")
print(fit_final_std.params[final_vars].abs().sort_values(ascending=False))
```

Listing 5: Fit Final Model

Why LASSO?

LASSO (Least Absolute Shrinkage and Selection Operator) helps us:

- Avoid overfitting by penalizing model complexity
- Perform automatic variable selection (sets some coefficients to exactly zero)
- Build more interpretable, parsimonious models
- Handle correlated predictors better than stepwise selection

Key insight: LASSO finds variables that are *independently* predictive, controlling for all others.

6 Comprehensive Review Questions

Conceptual Questions for Exam Preparation

Review these concepts - they may appear on the midterm:

1. **Model Comparison:** When comparing nested models, which test do you use? How do you interpret the p-value?
2. **Coefficient Interpretation:** In a model with log-transformed outcome, how do you interpret a coefficient of 0.03 on a predictor?
3. **Categorical Variables:** If a state has 50 levels and we include it in a regression, how many dummy variables are created? Why?
4. **Model Assumptions:** Name the four main assumptions of linear regression. How do you check each one?
5. **R-squared vs Adjusted R-squared:** Which should you use when comparing models with different numbers of predictors? Why?
6. **P-values:** What does it mean if a variable has p-value $= 0.03$ when using $\alpha = 0.05$?
7. **Confounding:** What is a confounding variable? How does multiple regression help address it?
8. **LASSO Tuning:** What is the role of the regularization parameter (alpha or lambda)? What happens when it's very large? Very small?

7 Summary and Final Reminders

Key Takeaways for Exam Success

- The actual exam will use this **SAME dataset** (`covid_county_midterm.csv`)
- The exam will test the **SAME core statistical skills** you practiced here
- The **specific questions will be different** - don't just memorize answers!
- Understand the **why** behind each analysis, not just the code

- Practice interpreting results in **plain English**
- Be able to justify your modeling choices
- Know how to check and interpret model diagnostics
- Ensure your Python environment is set up and tested before exam day

You've got this! Work through this practice systematically and you'll be well-prepared.

End of Practice Test

Questions? Review materials? Visit office hours!