Environmental Data and Analysis

Dana Golden

Environmental and Natural Resource Economics - **December 9, 2024**

Presentation Outline

What is Economics about Really?

- Is economics about the stock market?

What is Economics about Really?

- Is economics about the stock market?
- Is economics about money?

What is Economics about Really?

- Is economics about the stock market?
- Is economics about money?
- Is economics about the economy? Consumers? Business and profits? Policy?

What is Economics about Really?

- Is economics about the stock market?
- Is economics about money?
- Is economics about the economy? Consumers? Business and profits? Policy?
- Economics is about a particular kind of scarcity
- Economics is about how individuals make decisions about tradeoffs between finite resources and infinite desires

What is Economics about Really?

- Is economics about the stock market?
- Is economics about money?
- Is economics about the economy? Consumers? Business and profits? Policy?
- Economics is about a particular kind of scarcity
- Economics is about how individuals make decisions about tradeoffs between finite resources and infinite desires
- Economic decisions are about tradeoffs, human decision and human interaction

What is an Economically Interesting Question: Research

- Economically interesting research questions need human action
- Someone needs to make a decision about something, particularly something requiring optimization of resources

What is an Economically Interesting Question: Research

- Economically interesting research questions need human action
- Someone needs to make a decision about something, particularly something requiring optimization of resources
- Economic questions are motivated by theory, we are not regression jockeys

What is an Economically Interesting Question: Research

- Economically interesting research questions need human action
- Someone needs to make a decision about something, particularly something requiring optimization of resources
- Economic questions are motivated by theory, we are not regression jockeys
- Economically interesting questions are also novel and non-obvious
- If there is economic agreement and lots of evidence that two variables in a particular way, new evidence isn't particularly interesting
- It might be cool to analyze a new policy, but without an interesting connection to theory, not particularly interesting

Characteristics of Good Research Questions

- Specific: Good research questions are specific enough that they can be answered and focus on situations with many assumptions
- Falsifiable: Your hypothesis should be able to be proven wrong, evidence could be found either way
- Data-driven: Understand what data exist and what estimation might look like before starting research
- Novel: Good research questions do things that have not been done before but usually build strongly on existing literature and models
- Literature-driven: Good research should be aware of techniques used in literature and how they can be improved

Examples of Environmental Economics Research Questions

- What is the impact of policy uncertainty on decisions of coal generators?
- What is the best technique for valuing water rights in the Western United States?
- How do carbon border adjustment mechanisms impact international trade?
- What are the macroeconomic impacts of carbon taxes on economig growth?
- What is the impact of increasing transmission on generation investment in electricity markets?

What is an Economically Interesting Question: Business

- In business, economically interesting questions are much more broad

What is an Economically Interesting Question: Business

- In business, economically interesting questions are much more broad
- In particular, in business, theories of human behavior are less important than empirical predictions
- Forecasting is almost anathema to traditional economics but is central to analysis in business

What is an Economically Interesting Question: Business

- In business, economically interesting questions are much more broad
- In particular, in business, theories of human behavior are less important than empirical predictions
- Forecasting is almost anathema to traditional economics but is central to analysis in business
- Research in economics requires interesting complex new techniques
- Research in business often uses boring techniques because they work and are easy to explain

What is an Economically Interesting Question: Business

- In business, economically interesting questions are much more broad
- In particular, in business, theories of human behavior are less important than empirical predictions
- Forecasting is almost anathema to traditional economics but is central to analysis in business
- Research in economics requires interesting complex new techniques
- Research in business often uses boring techniques because they work and are easy to explain
- Business analysis often focuses on price or demand forecasting, understanding customer behavior, and cost-benefit analysis

What a Request from a Manager Looks Like

- Hey Dave, I was wondering if you could come up with characteristics of customers who buy to predict who buys?

What a Request from a Manager Looks Like

- Hey Dave, I was wondering if you could come up with characteristics of customers who buy to predict who buys?
- Hi Sarah! Could you run the numbers on projections for revenues and costs for next quarter?

What a Request from a Manager Looks Like

- Hey Dave, I was wondering if you could come up with characteristics of customers who buy to predict who buys?
- Hi Sarah! Could you run the numbers on projections for revenues and costs for next quarter?
- Could you come up with a valuation for this project?

What a Request from a Manager Looks Like

- Hey Dave, I was wondering if you could come up with characteristics of customers who buy to predict who buys?
- Hi Sarah! Could you run the numbers on projections for revenues and costs for next quarter?
- Could you come up with a valuation for this project?
- Often very vague and very open ended with faster timelines and more specialized data
- Explain-ability is very important. Overly complex models are frowned upon.
- Results should be succinct, actionable, and visualized

Examples of Environmental Economics Business Questions

- What is the net present value of this wind turbine? How sensitive are your estimates to interest rates?
- How should we think about environmental damage and mitigate the costs of damage we cause?
- How should we price this flood insurance product? How will those prices change with interest rates?
- How will climate change and the renewable energy transition impacts the markets for our products?

Structure of a Research Paper

- Abstract: Broad overview of paper and analysis
- Introduction: What is the paper about? Why is it important?
- Literature Review: What have others done? How are you different?
- Model: How do you think this industry or phenomenon works? Who are the players? What motivates them? What economic theory are you testing or building on?
- Data: Where is your data from? What makes it interesting? What features characterize the data?
- Estimation Strategy: How do you plan to estimate your model? What statistical models will you use?
- Results: What exactly came of your estimation strategy?

Knowing what Data you Need

- Data is almost always a major constraint on any research project
- Look at data used by others in field and replicate, replicate, replicate
- Start by thinking about level of granularity needed
  - Granularity is hard, you want to make sure that you answer the question you're asking
- Ask what in the data answers the question you have
- Ask what data are missing. How does this impact your model?

Sources of Environmental Data

- Federal sources:
  - EPA: Data on emissions and environmental quality
  - EIA: Data on energy and energy markets
  - ERS/NASS: Data on agriculture and rural environment
  - NREL: Data on technical side and potential side of renewable energy
- Private data sources
  - Bloomberg NEF
  - S&P Capital IQ Global
  - ISO's
- International data sources
  - UN
  - EU
  - OECD

Sources of Commodity Market Data

Key Sources for Commodity Market Data:

- **Government and International Organizations:**
  - **U.S. Energy Information Administration (EIA):**
    - Energy prices, production, and consumption data.
    - Example: eia.gov
  - **U.S. Geospatial Services (USGS):**
    - Data on metals.
    - Example: Lithium Data
  - **U.S. Department of Agriculture (USDA):**
    - Data on agricultural commodities.
    - Example: NASS Quickstats
  - **World Bank Commodity Price Data (Pink Sheet):**
    - Historical prices of key commodities.
    - Example: worldbank.org
  - **International Energy Agency (IEA):**
    - Energy market statistics and forecasts.
    - Example: iea.org

Finding Commodity Market Data

- **Private Data Providers:**
  - **S&P Global Platts:** Pricing and analytics for commodities.
  - **Bloomberg Terminal:** Real-time commodity prices and analysis.
  - **Refinitiv (formerly Thomson Reuters):** Commodity market data and analytics.
- **Exchanges:**
  - **Chicago Mercantile Exchange (CME):** Futures and options data.
  - **Intercontinental Exchange (ICE):** Energy and agricultural commodities.
  - **London Metal Exchange (LME):** Base metal prices.
- **Academic and Open Data:**
  - **Quandl:** Data for academic and research purposes.
  - **UN Comtrade Database:** International trade statistics.

What is an API?

- API stands for Application Programming Interface.
- It allows applications to communicate with one another.
- Data APIs provide programmatic access to datasets via HTTP requests.

Key Components:

- **Endpoint:** URL where the API can be accessed.
- **Parameters:** Inputs to customize the request (e.g., date range).
- **Authentication:** Methods like API keys or OAuth to access data securely.

API Visual

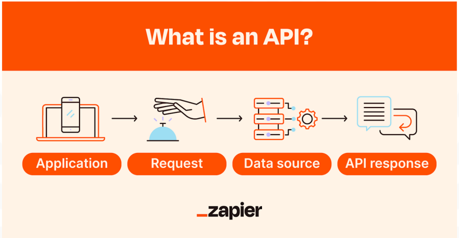

Figure 1: Application to Data

Steps to Access Data via API

1. **Identify the API:**
   - Choose a suitable API for the data you need (e.g., EIA, SEC).
2. **Obtain API Access:**
   - Sign up for an account, if required.
   - Obtain an API key for authentication.
3. **Understand API Documentation:**
   - Study the available endpoints, parameters, and response formats.
4. **Make the API Request:**
   - Use tools like Python (requests library) or Postman.
   - Example HTTP request: `GET https://api.example.com/data?key=API_KEY`
5. **Parse the Response:**
   - Decode JSON or XML response and extract the data.
6. **Store and Use the Data:**
   - Save the data in your desired format (CSV, database, etc.).

Example: Making an API Request in Python

Python Code for API Request:

```python
import requests

# Define the API endpoint and parameters
url = "https://api.example.com/data"
params = {
    "key": "YOUR_API_KEY",
    "date": "2024-12-01",
    "format": "json"
}
```

Example: Making an API Request in Python

Python Code for API Request:

```python

# Check the response
if response.status_code == 200:
    data = response.json()
    print("Data:", data)
else:
    print("Error:", response.status_code)
```

Example: Making an API Request in R

R Code for API Request:

```r
# Load required libraries
library(httr)
library(jsonlite)

# Define the API endpoint and parameters
url <- "https://api.example.com/data"
params <- list(
    key = "YOUR_API_KEY",
    date = "2024-12-01",
    format = "json"
)

# Make the GET request
response <- GET(url, query = params)
```

Example: Making an API Request in R

R Code for API Request:

```

# Check the response status
if (status_code(response) == 200) {
    # Parse the JSON response
    data <- fromJSON(content(response, as = "text"))
    print("Data:")
    print(data)
} else {
    cat("Error: ", status_code(response), "\n")
}
```

Getting an API Key from EIA

1. **Visit the EIA Developer Portal:** Navigate to EIA API Registration.
2. **Create an Account:** Register with a valid email address and create a password.
3. **Request an API Key:** After logging in, request an API key. The key will be emailed to you.
4. **Secure Your Key:** Store your API key securely. Treat it like a password; do not share it publicly.
5. **Key Usage:** Use the API key to authenticate your queries to the EIA API endpoints.

Tip: Always read EIA's API terms of use to ensure compliance.

Tips for Writing EIA API Queries

- **Understand the API Documentation:** Read the official EIA API documentation to know available endpoints, parameters, and data structure.
- **Choose the Right Endpoint:** Select the API endpoint that matches your data needs (e.g., electricity, petroleum, natural gas).
- **Use Filters:** Apply filters like geography, time, and series IDs to narrow down your data request.
- **Test Your Query:** Use tools like Postman or curl to test and debug your API queries.
- **Optimize for Efficiency:** Avoid excessively large queries by limiting your data request to only what you need.
- **Handle Errors Gracefully:** Write scripts to manage errors such as invalid keys, rate limits, or missing data.

Pro Tip: Check the "Examples" section in the EIA API documentation for sample queries.

Example of an EIA API Query

Scenario: Fetch total electricity generation data for 2023.

API Query:

https://api.eia.gov/v2/electricity/data/? $api_key = YOUR_API_KEYdata[] = generationfrequency = monthlystart = 2023 - 01end = 2023 - 12$

Explanation:

- `api_key`: Replace with your unique API key.
- `data[]`: Specify the type of data (e.g., `generation`).
- `frequency`: Set the data frequency (e.g., `monthly`).
- `start` and `end`: Define the time range.

Result: Returns monthly electricity generation data for 2023.

Getting an API Key from PJM

1. **Visit the PJM API Portal:** Navigate to the PJM API Portal.
2. **Register for an Account:** Click on "Sign Up" and provide the necessary information to create a PJM tools account.
3. **Email Verification:** Check your email for a verification link from PJM and follow the instructions to verify your account.
4. **Request API Access:** After logging in, request access to the Data Miner API by subscribing to the relevant API products.
5. **Obtain Your API Key:** Once approved, retrieve your unique API key from your profile on the PJM API Portal.

Tips for Writing PJM API Queries

- **Review API Documentation:** Familiarize yourself with the PJM Data Miner API Guide to understand available endpoints and parameters.
- **Select Appropriate Endpoints:** Choose endpoints that align with your data requirements, such as real-time LMPs or load forecasts.
- **Apply Filters:** Use query parameters to filter data by criteria like date ranges, geographic zones, or specific data feeds.
- **Test Queries:** Utilize tools like Postman to test and refine your API requests before integrating them into applications.
- **Manage Rate Limits:** Be aware of PJM's rate limits (e.g., non-members may not exceed 6 data connections per minute) to avoid service interruptions.

Pro Tip: Regularly check PJM's Data Miner page for updates and new data feeds.

Example Data Miner Query

Scenario: Retrieve real-time five-minute Locational Marginal Prices (LMPs) for a specific date.

API Query:

`https://api.pjm.com/api/v1/rt`$_f ivemin_h rl_l mps?startRow = 1 rowCount = 5000 sort = datetime_b eginning_e ptorder = ascdatetime_b eginning_e pt = 2023 - 12 - 01T00 : 00to2023 - 12 - 01T23 : 59format = csv$

Explanation:

- `startRow` and `rowCount`: Define the pagination of results.
- `sort` and `order`: Specify sorting by date and ascending order.
- `datetime_beginning_ept`: Set the date range for December 1, 2023.
- `format`: Request data in CSV format.

Note: Include your API key in the request header as specified in the PJM API documentation.

Tidyverse

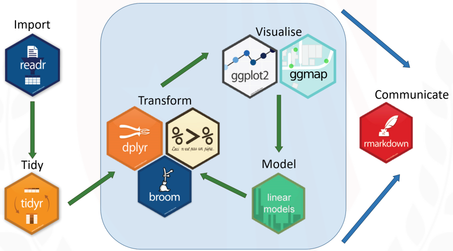

Figure 2: Wake up babe. New Tidyverse package just dropped...

Introduction to Data Cleaning with dplyr and tidyverse

- **dplyr**: A grammar of data manipulation, part of the `tidyverse`.
- **tidyverse**: A collection of R packages designed for data science.
- **Common tasks:**
  - Filtering rows
  - Selecting columns
  - Creating or modifying variables
  - Handling missing data
  - Summarizing and grouping
- **Why use dplyr?**
  - Simple, readable syntax
  - Handles large datasets efficiently

Common dplyr Functions

- `filter()`: Subset rows based on conditions.
- `select()`: Choose specific columns.
- `mutate()`: Add or modify variables.
- `arrange()`: Sort rows.
- `summarize()`: Aggregate data.
- `group_by()`: Group data for summarization.
- `rename()`: Rename columns.

Pipe Operator (%>%): Use to chain operations for clean, readable code.

Example

```
data %>% filter(col1 > 10) %>% select(col1, col2)
```

Example 2: Handling Missing Data

Scenario: Remove rows with missing values in critical columns.
Code:

```
data <- data filter(!is.na(Salary)) mutate(Salary =
replace(Salary, is.na(Salary), median(Salary, na.rm = TRUE)))
```

Explanation:

- `filter(!is.na(Salary))`: Remove rows where `Salary` is missing.
- `mutate()`: Replace missing values in `Salary` with the median.

Example 3: Summarizing and Grouping Data

Scenario: Calculate average salary by department.
Code:

```
summary <- data group_by(Department) summarize(Average_Salary
= mean(Salary, na.rm = TRUE))
```

Explanation:

- `group_by(Department)`: Group data by Department.
- `summarize()`: Calculate the average salary for each group.

Resources for Learning More about Tidyverse

- **Official Documentation:** dplyr.tidyverse.org
- **Cheatsheet:** Data Transformation Cheatsheet
- **Books:**
  - *R for Data Science* by Hadley Wickham and Garrett Grolemund
  - *Data Manipulation with R* by Jaynal Abedin
- **Practice:** DataCamp and Kaggle

Data Requirements for the Hedonic Pricing Model

Key Components:

- **Dependent Variable:**
  - Price of the good (e.g., property price, product price).
- **Independent Variables:**
  - **Structural Attributes:** Characteristics of the good (e.g., size, age, features).
  - **Locational Attributes:** Geographic location, proximity to amenities, or environmental factors.
  - **Environmental/Contextual Attributes:** Non-market factors (e.g., air quality, noise levels).
- **Data Quality:**
  - Accurate measurement of prices and attributes.
  - Sufficient variability in attributes for robust estimation.
- **Sample Size:**
  - A large sample size is ideal to capture heterogeneity.

Example Equation for the Hedonic Pricing Model

Hedonic Pricing Equation:

$$P_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki} + \epsilon_i$$

- P_i: Price of the good i (e.g., house price).
- $X_{1i}, X_{2i}, \ldots, X_{ki}$: Attributes of the good.
- β_0: Intercept term.
- β_k: Coefficients for each attribute X_k, representing marginal implicit prices.
- ϵ_i: Error term capturing unobserved factors.

Example:

$$P_{\text{house}} = \beta_0 + \beta_1(\text{Size}) + \beta_2(\text{Age}) + \beta_3(\text{Proximity to park}) + \epsilon$$

Model Estimation Code in R

Estimating a Hedonic Pricing Model:

```
# Load required library
library(tidyverse)

# Example dataset
data <- tibble(
    price = c(300000, 250000, 400000, 350000),
    size = c(1500, 1200, 1800, 1600),
    age = c(10, 20, 5, 15),
    park_proximity = c(1, 0, 1, 0) # 1 = Near park, 0
        = Not near
)

# Fit the Hedonic Pricing Model
model <- lm(price ~ size + age + park_proximity, data
   = data)
```

Model Estimation Code in R

Estimating a Hedonic Pricing Model:

```r
# Summarize the results
summary(model)
```

Explanation:

- **price:** Dependent variable (e.g., house price).
- **size, age, park_proximity:** Independent variables (attributes).
- **lm():** Fits a linear regression model.
- **summary():** Provides coefficient estimates and model diagnostics.

Overview of Logistic Regression

What is Logistic Regression?

- A statistical model used to predict the probability of a binary outcome (e.g., success/failure, yes/no).
- Dependent variable Y is binary (e.g., $Y = 1$ or $Y = 0$).

Logistic Regression Equation:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k)}}$$

- $P(Y = 1|X)$: Probability of the positive outcome given predictors X.
- $\beta_0, \beta_1, \ldots, \beta_k$: Model parameters to be estimated.

Applications:

- Credit scoring (approve/deny loan).
- Resource valuation (e.g., probability of resource availability).

Logistic Regression for Non-Market Valuation

Why Use Logistic Regression?

- Predicts the probability that an individual accepts or rejects a bid for a non-market good.
- Useful for estimating willingness to pay (WTP).

Model Framework:

- Dependent Variable (Y):
  - $Y = 1$: Accept the bid (e.g., willing to pay).
  - $Y = 0$: Reject the bid (e.g., not willing to pay).
- Independent Variables:
  - Bid amount (e.g., \$5, \$10, \$20).
  - Demographics (e.g., income, education).
  - Attitudes or preferences (e.g., environmental concern).

Applications:

- Valuation of clean air, water quality, park access, etc.

Logistic regression visualized

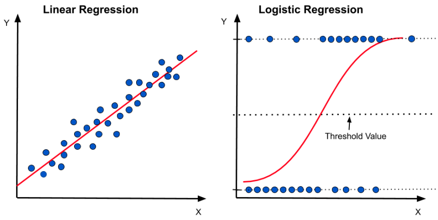

Figure 3: Linear vs. Logistic

Example: Logistic Regression for Individual Valuation

Example Research Question:

- What is the probability that an individual is willing to pay $X for improved water quality in a local lake?

Example Logistic Regression Model:

$$\log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1(\text{Bid}) + \beta_2(\text{Income}) + \beta_4(\text{EnvConcern})$$

- $Y = 1$: Respondent is willing to pay the bid amount.
- Bid: Amount offered in the survey.
- Income: Respondent's household income.
- EnvConcern: Level of environmental concern (e.g., Likert scale).

WTP Estimation:

- Mean WTP can be derived as:

$$\textbf{Mean WTP} = -\frac{\beta_0 + \beta_2(\text{Avg Income}) + \beta_4(\text{Avg EnvConcern})}{\beta_1}$$

Example Code: Logistic Regression for WTP in R

R Code for Non-Market Valuation:

```r
# Load required libraries
library(tidyverse)

# Example dataset
data <- tibble(
    accept = c(1, 0, 1, 0, 1),  # 1 = Willing to pay,
        0 = Not willing
    bid = c(5, 10, 15, 20, 5),
    income = c(50000, 45000, 60000, 40000, 55000),
    education = c(16, 14, 18, 12, 16),
    env_concern = c(4, 3, 5, 2, 4)  # 1-5 Likert scale
)
```

Example Code: Logistic Regression for WTP in R

R Code for Non-Market Valuation:

```
# Fit the logistic regression model
model <- glm(accept ~ bid + income + education + env_
    concern,
                data = data,
                family = binomial)

# Summarize the results
summary(model)
```

Example Code: Logistic Regression for WTP in R

R Code for Non-Market Valuation:

```r

# Calculate Mean WTP (example)
beta <- coef(model)
avg_income <- mean(data$income)
avg_education <- mean(data$education)
avg_env_concern <- mean(data$env_concern)
mean_wtp <- -(beta[1] + beta[3] * avg_income +
              beta[4] * avg_education +
              beta[5] * avg_env_concern) / beta[2]
cat("Estimated Mean WTP:", mean_wtp, "\n")
```

Example Code: Logistic Regression for WTP in R: Explanation

- **accept:** Dependent variable (willingness to pay).
- **bid, income, education, env_concern:** Independent variables.
- **glm():** Fits a logistic regression model.
- **Mean WTP:** Derived from the model coefficients.

Visualization of Predicted Probabilities

Predicted Probability of Bid Acceptance:

- Visualization of the probability that respondents accept a bid.
- Helps identify trends in willingness to pay.

R Code for Visualization:

```r
ggplot(new_data, aes(x = bid, y = predicted_probability)) +
  geom_line(color = "blue") +
  labs(title = "Predicted Probability of Accepting Bid",
       x = "Bid Amount ($)",
       y = "Predicted Probability") +
  theme_minimal()
```

Production Function Estimation for Environmental Valuation

What is Production Function Estimation?

- A method to estimate the relationship between inputs (e.g., labor, capital, environmental quality) and outputs (e.g., agricultural yield, fisheries production).
- Useful for valuing environmental goods by estimating their contribution to economic production.

Example Applications:

- Valuing water quality improvements in agricultural production.
- Estimating the effect of air quality on labor productivity.

General Form:

$$Q = f(K, L, E) + \epsilon$$

Example Code: Production Function Estimation in R

R Code for Estimating a Cobb-Douglas Production Function:

```r
# Load required library
library(tidyverse)

# Example dataset
data <- tibble(
    output = c(100, 150, 200, 250, 300),    # Output (
        e.g., crop yield)
    capital = c(10, 15, 20, 25, 30),        # Capital
        input (e.g., machinery)
    labor = c(5, 7, 9, 11, 13),             # Labor
        input (e.g., workers)
    water_quality = c(8, 8.5, 9, 9.5, 10)   #
        Environmental input
)
```

Example Code: Production Function Estimation in R

R Code for Estimating a Cobb-Douglas Production Function:

```
# Log-transform the data
data <- data %>%
    mutate(log_output = log(output),
           log_capital = log(capital),
           log_labor = log(labor),
           log_water_quality = log(water_quality))
```

Production Function Estimation

```
Summarize the results summary(model)
Calculate marginal productivity of water quality
beta_water <- coef(model)["log_water_quality"]
MeanQuality<-mean(data$water_quality)
MeanOutput<-mean(data$output)
marginal_productivity <- beta _water * ) /MeanQuality
cat("MP Quality:", marginal_productivity, "\n")
```

Explanation:

- **output:** Dependent variable (e.g., crop yield).
- **capital, labor, water_quality:** Independent variables.
- **log-transform:** Transforms data for Cobb-Douglas estimation.
- **Marginal Productivity:** Value of an additional unit of environmental input.

Basic Time-Series Forecasting

What is Time-Series Forecasting?

- Predicting future values based on past observations of a variable.
- Assumes temporal dependence between data points.

Key Concepts:

- **Stationarity:** The statistical properties of the series do not change over time.
- **Autocorrelation:** Correlation between observations at different time lags.
- **Lag Order:** Number of past observations used to predict future values.

Common Models:

Common Time-series Models

- **Autoregressive (AR):** Depends on its own past values.
- **Moving Average (MA):** Depends on past forecast errors.
- **ARIMA:** Combines AR and MA with differencing to handle non-stationarity.

AR Visualized

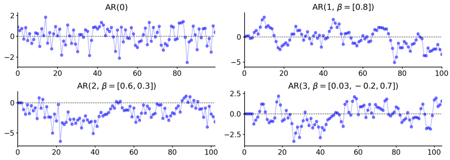

Figure 4: Go ahead. Add more terms...

Forecasting Natural Gas Futures Prices with AR(1)

AR(1) Model:

$$P_t = \alpha + \beta P_{t-1} + \epsilon_t$$

- P_t: Futures price at time t.
- P_{t-1}: Futures price at time $t-1$ (lagged price).
- α: Intercept term.
- β: Autoregressive coefficient, measuring dependence on the previous value.
- ϵ_t: White noise error term.

Forecasting Natural Gas Futures Prices with AR(1)

Steps:

1. Obtain historical natural gas futures prices from EIA.
2. Test for stationarity using the Augmented Dickey-Fuller (ADF) test.
3. Fit an AR(1) model to the data.
4. Use the model to forecast future prices.

Applications:

- Energy market price prediction.
- Risk management and portfolio optimization.

Example Code: AR(1) Model for Natural Gas Prices in R

R Code for AR(1) Estimation:

```r
# Load libraries
library(tidyverse)
library(forecast)

# Example dataset: Simulated natural gas futures
   prices
set.seed(123)
data <- tibble(
    date = seq.Date(from = as.Date("2023-01-01"), by =
        "month", length.out = 36),
    price = cumsum(rnorm(36, mean = 0.5, sd = 2)) +
        100
)
```

Example Code: AR(1) Model for Natural Gas Prices in R

R Code for AR(1) Estimation:

```
# Convert to time series object
gas_prices <- ts(data$price, start = c(2023, 1),
    frequency = 12)

# Test for stationarity
adf_test <- tseries::adf.test(gas_prices)
print(adf_test)

# Fit AR(1) model
ar_model <- arima(gas_prices, order = c(1, 0, 0))
summary(ar_model)
```

Example Code: AR(1) Model for Natural Gas Prices in R Explanation

- **Stationarity:** Checked using the ADF test.
- **AR(1) Model:** Fit using the 'arima()' function.
- **Forecast:** Predicted prices for the next 12 months.

Example Code: AR(1) Model for Natural Gas Prices in R

R Code for AR(1) Estimation:

```
# Forecast future prices
forecast_values <- forecast(ar_model, h = 12)
print(forecast_values)

# Plot the forecast
autoplot (forecast_values) + labs (title = "Natural gas futures price forecast", x = "Time",
    y = "Price ($)")
```

Example Code: AR(1) Model for Natural Gas Prices in R Explanation

- **Stationarity:** Checked using the ADF test.
- **AR(1) Model:** Fit using the ‘arima()‘ function.
- **Forecast:** Predicted prices for the next 12 months.

Forecasting with Exponential Smoothing

What is Exponential Smoothing?

- A forecasting method that applies exponentially decreasing weights to past observations.
- Suitable for time series with trends and seasonality.

Types of Exponential Smoothing:

- **Simple Exponential Smoothing (SES):**
  - For series with no trend or seasonality.
  - Formula: $\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$.
- **Holt's Linear Trend Method:**
  - For series with a trend.
- **Holt-Winters Method:**
  - For series with both trend and seasonality.

Key Parameters Exponential Smoothing

- α: Smoothing parameter for the level.
- β: Smoothing parameter for the trend.
- γ: Smoothing parameter for the seasonality.

Exponential Smoothing Visualized

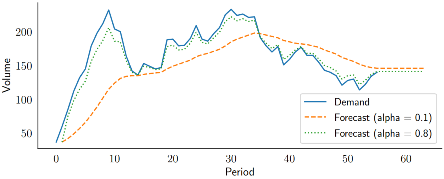

Figure 3.2: Simple smoothing

Figure 5: Exponential Smoothing

Example Code: Exponential Smoothing in R

R Code for Exponential Smoothing:

```
# Load required libraries
library(forecast)

# Simulated time series data
set.seed(123)
data <- ts(rnorm(50, mean = 100, sd = 10), start = c
    (2023, 1), frequency = 12)

# Simple Exponential Smoothing
ses_model <- ses(data, h = 12)
summary(ses_model)

# Holt's Linear Trend Method
holt_model <- holt(data, h = 12)
summary(holt_model)
```

Example Code: Exponential Smoothing in R

R Code for Exponential Smoothing:

```r

# Holt-Winters Seasonal Method
hw_model <- hw(data, h = 12, seasonal = "additive")
summary(hw_model)

# Plot forecasts
autoplot(data) +
  autolayer(ses_model, series = "SES", PI = FALSE) +
  autolayer(holt_model, series = "Holt", PI = FALSE) +
  autolayer(hw_model, series = "Holt-Winters", PI =
      TRUE) +
  labs(title = "Exponential␣Smoothing␣Forecasts",
       x = "Time",
       y = "Value") +
  theme_minimal()
```

Exponential Smoothing in R Explanation

- **SES:** Suitable for series without trend or seasonality.
- **Holt:** Accounts for trends in the data.
- **Holt-Winters:** Captures both trends and seasonality.
- **autoplot():** Visualizes forecasts with historical data.

Thank You So Much!

List of References