Cheat Sheet

Day1 – Introduction

Data manipulation process: accessing, converting, transforming, cleaning, filtering, grouping and summarizing.

Exploratory data analysis: visualize data, detect trends/patterns. Focus on patterns

Python data structures: list: include mixed types, mutable; tuple: mixed types, immutable; dict: unique and hashable keys, mutable; set: unordered collection, immutable, no duplicates

Pandas.DataFrame: two-dimensional, rows organized by indexes, Columns are Series. Can be created by series, dictionaries

These concepts and methods lay the foundation for the rest of the course.

Day2&3 - Pandas I & II

Methods for inspecting df: describe(), info(), head(), sample()

Methods for cleaning and converting:

* Dealing with Nas: dropna(), fillna(value), or impute in following Pipeline
* Dealing with categorical variables: dummies/one-hot
* Discretize integers
* Dealing with specific rules: map(), apply()

Dealing with multiple datasets: merge on unique ids.

Common methods for EDA: value\_counts(), sort\_values(), and aggregating: groupby() with combination of aggregating funcs

Pivot table: pivot\_table(index, columns, values, aggfunc)

Connection: These EDA methods are important for inspecting data features，correlations, following with feature engineering and important for building models.

Day4 Univariate statistical analysis and visualization

**Data summary metrics:** centrality: mean, median, mode; range, variance, standard deviation, percentiles and quantiles.

**Visualization:** bar plot (categorical), pie charts (categorical), line plot (time trends), histogram (distribution), scatter plot (bivariate relationship), seaborn.pairplot (combination of various plots of multiple var pairs), subplot (combination of plots), stats.probplot (QQplot)

**QQplot** takes n ordered data points, sort from smallest to largest, analyze relationship between quantiles and quantiles from a theoretical distribution. Visualize the difference from normal distribution, visualizing the skewness of the distribution.

**Connection:** we need visualization to analyze the distribution and relationships between variables. We need more information than descriptive statistics as distribution will be different given the same descriptive statistics (Anscombe’s Quartet). These information helps us understand our data, contributing to feature engineering and building up models.

Day5 Correlation, ANOVA, Regression

**Parameters** are placeholders defined in the function declaration. Functions can accept zero or more parameters. **Positional arguments** are assigned to parameters in the same order they appear in the function definition. **Keyword arguments** are assigned to parameters by name, regardless of their order. **\*args** pass a number of positional arguments. **\*\*kwargs** pass several keyword arguments. \* unpacks positional arguments from a list/tuple. \*\* unpacks keyword arguments from a dict.

**Correlation** measures the direction and strength of linear relationship between variables

(-1<=r<=1). 图片包含 文本

描述已自动生成**Correlation does not imply causation.** use heatmap to visualize the correlation matrix. Boxplot visualize the means and the quartiles.

**t-test** to test if two means differ significantly.

**ANOVA** (analysis of variance) compares means across multiple groups to test if the means differ significantly. Default type II, use type III if there are interaction terms.

**Regression** models the linear additive relationships between variable and predictors, builds up predictions. Check significant coefficients and goodness-of-fit.

Ways to measure goodness-of-fit:

* R-squared
* Examine residuals (homoscedasticity and errors are normally distributed, centered at 0)
* Leverage plot, check outliers

Connection: ANOVA and t-tests are important when we have critical grouping variables and to do classifications. Help us find critical predictors. Correlation is foundational to understanding relationships within numeric vars, which regression models quantify. Regression models find significant predictors for building up models.

Day6 Categorical Data & Text Processing

Debugging: print key things, use debugger, try explaining the code to find errors, take a break, ask for help

Errors: syntax, runtime, semantic errors, logical errors, exceptions, warnings. Pay attention to logical and semantic errors, double check code logics.

Dealing with categorical data: convert categories to numbers astype(‘category’).cat.codes, one-hot encoding (pd.get\_dummies)

Understanding frequencies in different groups: contingency tables and chi-squared test

Ways to visualize the crosstab: use heatmap or mosaic based on crosstabs

chi-squared tests：quantitatively validate the differences across groups. The chi-square test sums the squares of the differences between the observed and expected values, normalized for the expected values. Null hypothesis: no difference in the value across groups.

Text analysis:

String functions in pandas: str.strip(), str.split(), str.extract(substring), str.upper()/lower(), str.contains(regex), str.match(regex)

Connection: dealing with categorical data in the dataset -> pipeline in the preprocessing phase; text analysis helps us transform non-numeric non-structured data to structured numeric variables, laying foundation for NLP.

Day 7 NLP

Preprocess text data:

* Normalize the text: upper/lower
* Clean the text: drop punctuations, spaces/tabs, special characters, r'[^\w\s]+', remove stop words
* Identify sentences and split sentences
* Tokenize the text: split the texts into words, further lemmatize
* Part of speech tagging: identify different parts of the speech
* Named entity recognition: identify and classify key entities information (locations/people/orgs/dates…)

NLP analytic techniques:

* Word frequencies: identify words best represent the text content, help guess what the text is mainly talk about.
* Word embeddings: word representation that allows words to be expressed in a continuous **vector space**. These vectors capture semantic meanings and relationships between words. Utilize pre-trained language model like all-MiniLM-L6-v2
  + Calculating similarity between words, Find a word similar to a target word
  + Find analogies
  + Visualize the semantic scales (bad-good/ weak-strong/…)
* Sentiment analysis: determine word sentiment (positive, negative, neutral)

Connection: NLP helps us deal with text data like customer reviews. The output of NLP like sentiment scores can also be predictors for machine learning models.