Pandas Cheat Sheet

Import Creating Data

```
import pandas as pd
```

Create Series:

```
pd.Series(["Blue", "Red", "White"])
```

Create DataFrame from dict:

```
pd.DataFrame({"col1": data1, "col2": data2})
```

Read CSV:

```
pd.read_csv("file.csv")
```

Export to CSV:

```
df.to_csv("output.csv")
```

Viewing Data

```
df.head() # First 5 rows
df.head(7) # First 7 rows
df.tail() # Last 5 rows
df.info() # Data types info
df.describe() # Statistics summary
df.dtypes # Column data types
df.columns # Column names
len(df) # Number of rows
df.shape # (rows, columns)
```

Selection Indexing

Select column:

```
df["column_name"]
```

Select by label (row):

```
df.loc[3] # Row with index 3
```

Select by position:

```
df.iloc[3] # 4th row (0-indexed)
```

Boolean indexing:

```
df[df["Odometer"] > 100000]
```

.loc uses labels, .iloc uses positions

Statistics Aggregation

```
df["col"].mean() # Average
df["col"].sum() # Sum
df["col"].max() # Maximum
df["col"].min() # Minimum
df["col"].std() # Standard deviation
df["col"].count() # Non-null count
df["col"].unique() # Unique values
df["col"].nunique() # Count unique
df["col"].value_counts() # Frequency
```

Grouping Crosstab

Group by column:

```
df.groupby(["Make"]).mean()
df.groupby(["Make"]).sum()
```

Multiple groups:

```
df.groupby(["Make", "Colour"]).count()
```

Crosstab (frequency table):

```
pd.crosstab(df["Make"], df["Doors"])
```

Data Manipulation

```
df["new_col"] = value # Add column
df.drop("col", axis=1) # Remove column
df.sample(frac=1) # Shuffle rows
df.reset_index() # Reset index
df.rename(columns={...}) # Rename cols
```

Use inplace=True to modify original

String Operations

```
df["col"].str.lower() # Lowercase
df["col"].str.upper() # Uppercase
df["col"].str.replace(old, new)
df["col"].str[:-2] # Slice string
df["col"].astype(int) # Change type
```

Example: Remove $ from prices

Handling Missing Data

Fill missing values:

```
df.fillna(value) # Fill with value
df["col"].fillna(df["col"].mean())
```

Remove missing values:

```
df.dropna() # Drop NaN rows
df.dropna(axis=1) # Drop NaN cols
```

NaN = Not a Number (missing value)

Apply Lambda Functions

Apply function to column:

```
df["col"].apply(function)
```

Lambda (anonymous function):

```
df["col"].apply(lambda x: x/1.6)
```

Example: Convert KM to miles

```
lambda x: x/1.6 # Divides each value
```

Visualization

```
%matplotlib inline # Jupyter magic
import matplotlib.pyplot as plt
df["col"].plot() # Line plot
df["col"].hist() # Histogram
df.plot.bar() # Bar chart
```

⚠ Column must be numeric to plot!

Common Issues Solutions

Problem: Can't plot Price column (TypeError)

Solution: Price stored as string ("$4,000.00")

```
df["Price"].str.replace(r"[\$\,\.]", "", regex=True)
df["Price"] = df["Price"].str[:-2] # Remove cents
df["Price"] = df["Price"].astype(int) # Convert
```

Always check data types with df.dtypes before operations!

Key Concepts

- **Series:** 1D labeled array (single column)
- **DataFrame:** 2D labeled table (rows columns)
- **Index:** Row labels (0, 1, 2... by default)
- **inplace=True:** Modifies original DataFrame (no assignment needed)
- **axis=0:** Operations down rows (columns), **axis=1:** Operations across columns (rows)

⚡ Pro Tips

- Chain methods: `df.groupby("Make").mean().sort_values("Price")`
- Use `numeric_only=True` in aggregations to avoid errors with non-numeric columns