Python in Excel: Quick Wins Reference Guide

1. Sample random rows	`df.sample() # Default` `df.sample(20) # Sample 20 rows`

2. Number of missing values in each column	`df.isna().sum() # Count missing per column`
	
	`(df.isna().sum() / len(df))\`
	`  .sort_values(ascending=False)  # % missing per column`

3. Descriptive statistics	`df.describe() # Default` `df.describe(percentiles=[.10, .50, .90]) # Custom percentiles`

4. Correlation matrix	`numeric_corr = df.select_dtypes(include="number").corr()  # Correlate all numeric variables`
	
	`sns.heatmap(numeric_corr, annot=True) # Visualize correlations`

5. Frequency tables	`df["col_1"].value_counts() # Frequency counts`
	
	`pd.crosstab(df["col_1"], df["col_2"],`
	`  normalize="all", margins=True) # Two-way proportion`

6. Resampling	`df_ts = df.set_index("date")  # Set datetime index`
	
	`df_ts.resample("M").sum().head() # Monthly totals`
	
	`(df_ts.resample("H").ffill() / 24).head(72) # Hourly forward-fill then scale`

7. Index number by group	`df["group_id"] = df.groupby("group_col").cumcount() + 1  # 1-based counter`

8. Leading and lagging variables	`df["lag_1"] = df["value"].shift(1) # Previous value`
	
	`df["lead_1"] = df["value"].shift(-1) # Next value`
	
	`df["pct_change_%"] = df["value"].pct_change() * 100 # % change`

9. Rolling and cumulative aggregations	`df["rolling_mean_7"] = df["value"].rolling(7).mean() # 7-period mean`
	
	`df["cum_sum"] = df["value"].cumsum() # Running total`
	
	`df["cum_mean"] = df["value"].expanding().mean() # Running average`
	
	`df["rolling_mean_7"] = df["value"].rolling(7, min_periods=1).mean() # Rolling w/ min`

10. Conditionally format plots	`sns.scatterplot(data=df, x="col_1", y="col_2", hue="group_col", alpha=.7) # Scatter w/ hue`
	
	`sns.barplot(data=df_mean, x="cat_col", y="metric", palette="Blues_r") # Bar chart with nonscaled palette`

11. Pairplot	`sns.pairplot(df) # Quick overview of all pair-wise relationships`
	
	`g = sns.pairplot(df, hue="group_col", diag_kind="kde", markers=["o", "s", "D"]) # Pairplot with hue, KDE diagonals, and custom markers`

	`g.fig.suptitle("Pairplot of Dataset by Group", y=1.02) # Add title`

12.Jitterplot	`sns.stripplot(data=df, x="cat_col_1", y="metric") # Default`
	
	`sns.stripplot(data=df, x="cat_col_1", y="metric", hue="cat_col_2", jitter=.2) # Custom jitter and hue`

13.Pairplot	`g = sns.FacetGrid(data=df, col="facet_col") # Column facets`
	`g.map(sns.scatterplot, "col_1", "col_2") # Map scatter`
	
	`g = sns.FacetGrid(data=df, col="facet_col", row="row_col", height=3) # Row + col facets`
	`g.map(sns.boxplot, "col_1") # Map boxplot`
	`g.set_titles(col_template="{col_name}", row_template="{row_name}") # Custom titles`

14.Jointplot	`sns.jointplot(data=df, x="col_x", y="col_y") # Default scatter + hist`
	
	`sns.jointplot(data=df, x="col_x", y="col_y", hue="group_col")  # Add hue by category`

15. Bubbleplot	`sns.scatterplot(data=df, x="col_1", y="metric", hue="group_col", size="size_col", alpha=.6)  # Bubble scatter`
	
	`sns.regplot(data=df, x="col_1", y="metric", scatter=False, color="black") # Trend line`