Function Descriptions for `iCharmFastAPI.py`

This document explains each function in the FastAPI script in simple terms. The script runs a web API that processes climate data (like precipitation or temperature) from local or cloud sources, using a `metadata.csv` file to locate datasets. It provides data for a React frontend with Recharts to create graphs like time series or bar charts.

Utility Functions

`load_metadata(metadata_path: str)`

- **What it does**: Loads the `metadata.csv` file into a table.
- **How it works**: Reads the CSV file at the given path. If the file is missing, it shows an error. If there's another issue (like a bad format), it reports that too.
- **Output**: A table with dataset details (e.g., name, location, variable).
- **Use case**: Used by other functions to find dataset information.

`iso_times_from_coord(time_coord)`

- **What it does**: Turns dataset time values into standard date strings (YYYY-MM-DD).
- **How it works**: Takes time data from a dataset, handles different formats (like numpy, pandas, or cftime dates), and converts them to a list of date strings.
- **Output**: A list of dates, e.g., `["2025-01-01", "2025-01-02"]`.
- **Use case**: Ensures consistent date formats for graphs.

`choose_best_variable(ds: xr.Dataset, fallback: str = "precip")`

- **What it does**: Picks the best variable (like temperature or precipitation) from a dataset if none is specified.
- **How it works**: Checks dataset variables, preferring those related to anomalies, temperature, precipitation, or wind. If none match, it uses the fallback (default: "precip") or the first valid variable.
- **Output**: The variable name, e.g., `sst` or `cmorph`.
- **Use case**: Automatically selects the main data to analyze.

`load_dataset(metadata_path: str, dataset_name: str, variable: Optional[str], year: Optional[int], month: Optional[int], day: Optional[int])`

- **What it does**: Loads a dataset based on its name and metadata.
- **How it works**: Uses `metadata.csv` to find the dataset's location (local or cloud) and format (Zarr or NetCDF). Loads local files directly, handles cloud data (single files, daily files, or Zarr with kerchunk), selects the specified or default variable, and fixes time formats.
- **Output**: A dataset slice ready for analysis.
- **Note**: Needs `year` and `month` for cloud datasets with directory structures (e.g., CMORPH).
- **Use case**: Core function for accessing data before processing.

`is_multilevel(da: xr.DataArray)`

- **What it does**: Checks if a dataset includes vertical levels (like height or pressure).
- **How it works**: Looks for "level" or "plev" in the dataset's dimensions.
- **Output**: `True` if the dataset has levels, `False` otherwise.
- **Use case**: Helps handle datasets with multiple altitudes.

`select_time_safe(da, date_str: str)`

- **What it does**: Selects data for a specific date.
- **How it works**: Matches the given date (e.g., "2025-01-01") to the dataset's time, handling different date formats. If it fails, it picks the first available time.
- **Output**: A dataset slice for the nearest date.
- **Use case**: Ensures accurate date-based data selection.

Data Processing Functions

`compute_point_statistics(da: xr.DataArray, lat: float, lon: float, level: Optional[float], date: Optional[str])`

- **What it does**: Calculates statistics (min, max, mean, etc.) for a specific location.
- **How it works**: Selects data at the given latitude, longitude, and optional level (for multi-level datasets). Uses one day if a date is given, or all times otherwise. Computes min, 25th percentile, median, mean, 75th percentile, max, std, variance, skewness, and kurtosis.
- **Output**: A table with one row of stats, labeled by location (e.g., "(32.7, -117.2)"). Returns empty if no valid data.
- **Use case**: Provides summary stats for tables or bar charts in Recharts.

`compute_point_stats(da: xr.DataArray, lat: float, lon: float, level: Optional[float])`

- **What it does**: Gets a time series of values at a location with overall stats.
- **How it works**: Selects data at the given lat/lon and optional level, removes invalid values, and calculates the mean and standard deviation for all times.
- **Output**: A table with time, value, mean, and std columns, with time as the index.
- **Use case**: Ideal for Recharts line charts showing data over time.

`compute_monthly_mean_std(da: xr.DataArray, lat: float, lon: float, year: int, level: Optional[float])`

- **What it does**: Finds monthly averages and standard deviations for a specific year.
- **How it works**: Selects data at the given lat/lon and optional level, filters for the year, and computes mean and std for each month. Errors if no data for the year.
- **Output**: A table with months (e.g., "January") as the index and mean/std columns.
- **Use case**: Great for Recharts bar or line charts showing monthly trends.

`compute_monthly_mean_yearly_std(da, lat: float, lon: float, year: int, level: float)`

- **What it does**: Gets monthly averages for a year and std across all years.
- **How it works**: Selects data at the given lat/lon and optional level, computes monthly means for the year, and std for each month across all years.
- **Output**: A table with months as the index and mean/std columns.
- **Use case**: Useful for Recharts charts comparing a year to long-term trends.

`compute_seasonal_timeseries(da, lat: float, lon: float, month: int, start_year: int, end_year: int, level: float)`

- **What it does**: Gets values for a specific month over a range of years.
- **How it works**: Selects data at the given lat/lon and optional level, filters for the month and year range, and returns values for each year. Errors if no data.
- **Output**: A table with years as the index and values.
- **Use case**: Perfect for Recharts line charts showing yearly trends for a month.

API Endpoints for Web Frontend

`get_datasets() - GET /datasets`

- **What it does**: Lists all available datasets for the frontend.
- **How it works**: Reads `metadata.csv` and returns dataset details as JSON.
- **Output**: JSON array of dataset info (e.g., name, variable, units).
- **Use case**: Populates a dropdown in React for dataset selection.

`get_point_timeseries(dataset_name: str, lat: float, lon: float, level: Optional[float], variable: Optional[str], year: Optional[int], month: Optional[int], day: Optional[int]) - POST /point_timeseries`

- **What it does**: Returns a time series for a location to plot.
- **How it works**: Loads the dataset and gets time series data at the given lat/lon.
- **Output**: JSON array with time, value, mean, and std.
- **Use case**: Feeds Recharts line charts for time series visualization.

`get_point_statistics(dataset_name: str, lat: float, lon: float, level: Optional[float], date: Optional[str], variable: Optional[str], year: Optional[int], month: Optional[int], day: Optional[int]) - POST /point_statistics`

- **What it does**: Returns stats for a location.
- **How it works**: Loads the dataset and computes stats at the given lat/lon.
- **Output**: JSON array with one object of stats (min, max, etc.).
- **Use case**: Displays in tables or Recharts bar charts.

`get_monthly_mean_std(dataset_name: str, lat: float, lon: float, year: int, level: Optional[float], variable: Optional[str]) - POST /monthly_mean_std`

- **What it does**: Returns monthly averages and std for a year.
- **How it works**: Loads the dataset and computes monthly stats at the given lat/lon.
- **Output**: JSON array with month, mean, and std.
- **Use case**: Plots monthly trends in Recharts bar or line charts.

`get_monthly_mean_yearly_std(dataset_name: str, lat: float, lon: float, year: int, level: Optional[float], variable: Optional[str]) - POST /monthly_mean_yearly_std`

- **What it does**: Returns monthly averages for a year and std across all years.
- **How it works**: Loads the dataset and computes stats at the given lat/lon.
- **Output**: JSON array with month, mean, and std.
- **Use case**: Shows yearly vs. long-term trends in Recharts charts.

`get_seasonal_timeseries(dataset_name: str, lat: float, lon: float, month: int, start_year: int, end_year: int, level: Optional[float], variable: Optional[str]) - POST /seasonal_timeseries`

- **What it does**: Returns values for a specific month over multiple years.
- **How it works**: Loads the dataset and extracts data for the month and years.
- **Output**: JSON array with year and value.
- **Use case**: Plots yearly trends for a month in Recharts line charts.