CDF-t Downscaling of CMIP6 Wind Data for the Saint-Nazaire Offshore Wind Farm

Arefe Asadi

2025-11-14

1.Introduction:

Climate models are advanced computational tools designed to simulate the behavior of atmospheric and climatic systems on a global scale. Two primary types include General Circulation Models (GCMs), such as those within the CMIP6 framework, and reanalysis datasets like ERA5. GCMs provide long-term climate projections using complex physical equations and various greenhouse gas emission scenarios (e.g., SSPs), but they operate at a relatively low spatial resolution (typically over 100 km), making them suitable for decadal to centennial-scale climate change studies. In contrast, reanalysis datasets like ERA5 integrate real observations (from weather stations, satellites, etc.) with models to offer high-resolution (e.g., 0.25°) estimates of past and present atmospheric conditions.

However, these models and datasets differ significantly from local measurements. Local data, collected directly from ground-based instruments or regional stations, capture fine-scale details and site-specific features (e.g., coastal wind dynamics), which GCMs, due to their coarse resolution and systematic biases (consistent deviations from observations), cannot adequately represent. For instance, wind speed in a specific region, might be inaccurately estimated by large-scale models.

This discrepancy between global and local scales underscores the need for downscaling techniques. Downscaling refines raw data from large-scale models into more accurate local-scale information. The two main approaches are dynamical downscaling (using Regional Climate Models) and statistical downscaling (e.g., probabilistic methods like CDF-t). This study focuses on statistical probabilistic downscaling, which employs Cumulative Distribution Functions (CDFs) to align large-scale data with local observations. Methods such as the CDF-transform, introduced in Michelangeli et al. (2009), aim to reduce biases and predict local changes (e.g., wind speed reductions in the 21st century).

This project develops a complete and reproducible workflow for statistical downscaling of wind speed at the Saint-Nazaire Offshore Wind Farm, located on the French Atlantic coast. In this project, the lack of access to local wind speed data for the park region necessitated the use of ERA5 as a proxy for local observations, providing a reliable high-resolution dataset to bridge this gap. The project tests this approach using ERA5 data for local-scale validation and CMIP6 data (specifically the IPSL-CM6A-LR model) for large-scale inputs, demonstrating its potential to enhance climate analysis at finer scales. The central coordinates of the wind farm are 47.16° N and 2.62° W, and the spatial domain used in this study was constructed by adding and subtracting 0.5° from these values—resulting in boundaries of 47.66° N (north), 46.66° S (south), 3.12° W (west), and 2.12° E (east)—to fully encompass the site.

2. Methodology: CDF-Transform Downscaling

The core of this project's workflow is based on the probabilistic downscaling method known as the Cumulative Distribution Function Transform (CDF-t), introduced in the article "Probabilistic downscaling approaches: Application to wind cumulative distribution functions" by Michelangeli et al. (2009). This method extends the traditional Quantile-Matching (Q-matching) approach by directly modeling and providing local-scale Cumulative Distribution Functions (CDFs) through a non-parametric transformation applied to large-scale CDFs. Unlike classical statistical downscaling methods that generate continuous time series, CDF-t focuses on downscaling statistical characteristics (i.e., CDFs) of climate variables, such as wind speed anomalies, enabling the creation of realistic local-scale data while accounting for changes in the large-scale distribution over time.

In mathematical terms, CDF-t assumes the existence of a transformation T that translates a large-scale CDF into a local-scale one. For a given climate variable X (e.g., wind speed), let F_{Sh} be the CDF of observed local data during the historical calibration period, and F_{Gh} the CDF of large-scale model outputs (interpolated to the local site) for the same period. For a future period, F_{Gf} represents the large-scale CDF to be downscaled. The transformation is defined such that $F_{Sf}(x) = F_{Sh}(F_{Gh}^{-1}(F_{Gf}(x)))$, where F_{Sf} is the estimated local future CDF. This approach assumes stationarity in the relationship between large- and local-scale CDFs, a common assumption in statistical downscaling that should be interpreted cautiously in a changing climate. For values outside the range of the future dataset, a constant correction is applied to avoid unphysical extremes, as suggested by Déqué (2007).

This method is implemented in R using the CDFt package, which provides a non-parametric tool for downscaling or bias-correcting CDFs. The primary function, CDFt(ObsRp, DataGp, DataGf, npas=1000, dev=2), takes the following arguments:

- **ObsRp**: A vector of observed time series at the local scale, used to estimate the calibration local-scale CDF (F_{Sh}). In this project, ERA5 reanalysis data serves as a proxy for ObsRp due to the unavailability of direct local wind speed measurements at the Saint-Nazaire Offshore Wind Farm. The following parameters were set while downloading the dataset from Climate Data Store:
  - Product Type: Monthly Averaged Reanalysis
  - Variable: 10m u-component of wind, 10m v-component of wind
  - Year: 1980-1999
  - Month: all
  - Time: 00:00
  - Geographical area: North: 47.66°, West: -3.12°, South: 46.66°, East: -2.12°
- **DataGp**: A vector of large-scale time series for the historical calibration period, used to estimate the calibration large-scale CDF (F_{Gh}). The following parameters were set while downloading the dataset from Climate Data Store:
  - Temporal Resolution: Monthly
  - Experiment: Historical
  - Variable: Near-surface wind speed
  - Model: IPSL-CM6A-LR(France)
  - Year: 1980-1999
  - Month: all
  - Geographical area: North: 47.66°, West: -3.12°, South: 46.66°, East: -2.12°
- **DataGf**: A vector of large-scale time series for the period to be downscaled (e.g., future projections), used to estimate the large-scale CDF to downscale (F_{Gf}). In this workflow, this corresponds to CMIP6 IPSL-CM6A-LR future scenario data (e.g., under SSP scenarios). The following parameters were set while downloading the dataset from Climate Data Store:
  - Temporal Resolution: Monthly
  - Experiment: Historical
  - Variable: Near-surface wind speed
  - Model: IPSL-CM6A-LR(France)
  - Year: 2000-2014
  - Month: all
  - Geographical area: North: 47.66°, West: -3.12°, South: 46.66°, East: -2.12°
- **npas**: The number of "cuts" or points for empirically estimating quantiles (default: 1000), controlling the resolution of the CDF estimation.
- **dev**: A development coefficient (default: 2) that extends the range of data for quantile calculation by amplifying the difference between the means of DataGp and DataGf, ensuring the method captures

potential extremes.

Apart from the above datasets another dataset is considered for validation part:

- **ObsVal**: A vector of observed time series at the local scale, used to validate the downscaled CDF (F_{Sf}) and assess the model's accuracy for the period 2000-2014. ERA5 reanalysis data is again utilized as a proxy due to the lack of direct local wind speed measurements at Saint-Nazaire. The dataset was downloaded from Climate Data Store: with the following parameters:
  - Product Type: Monthly Averaged Reanalysis
  - Variable: 10m u-component of wind, 10m v-component of wind
  - Year: 2000-2014
  - Month: all
  - Time: 00:00
  - Geographical area: North: 47.66°, West: -3.12°, South: 46.66°, East: -2.12°

The calibration period (1980–1999) was selected because it lies entirely within the CMIP6 historical simulation window (1850–2014) and simultaneously overlaps with ERA5 reanalysis data. This ensures that both large-scale GCM historical outputs (DataGp) and local-scale observations (ObsRp) are available over the same interval, which is required for estimating the CDF-t transformation.

The validation period (2000–2014) was chosen because it corresponds to the remaining portion of the CMIP6 historical experiment that still overlaps with ERA5. CMIP6 historical simulations stop in 2014, so 2000–2014 is the latest fully consistent period for which both ERA5 (ObsVal) and CMIP6 historical/future boundary data (DataGf) coexist. Using a non-overlapping later period allows an independent evaluation of the downscaling performance outside the calibration window.

3.Data Processing

The required R packages for data processing, NetCDF handling, and statistical downscaling are loaded below.

```r
library(dplyr)      # Data manipulation (filter, mutate, join, arrange)
library(tidyr)      # Reshaping data for ggplot-friendly formats
library(viridis)    # Colorblind-friendly color palettes for ggplot
library(ggplot2)    # Plotting and visualization
library(ncdf4)      # Reading and handling NetCDF climate data
library(lubridate)  # Extracting and manipulating date components (year, month)
library(CDFt)       # CDF-transform method for statistical downscaling
```

The following NetCDF files are opened: ERA5 reanalysis for the calibration period (ObsRp) and validation period (ObsVal), and CMIP6 IPSL-CM6A-LR historical simulations for the corresponding past (DataGp, 1980–1999) and pseudo-future (DataGf, 2000–2014) periods.

```r
obsrp <- nc_open("C:/Users/arefe/Desktop/Personal Folder/My_Projects/downscaling/downscaling/obsrp/data_

datagp <- nc_open("C:/Users/arefe/Desktop/Personal Folder/My_Projects/downscaling/downscaling/datagp/sf

datagf <- nc_open("C:/Users/arefe/Desktop/Personal Folder/My_Projects/downscaling/downscaling/datagf/sf

obsval <- nc_open("C:/Users/arefe/Desktop/Personal Folder/My_Projects/downscaling/downscaling/obsval/da
```

For **ObsRp** the vector of observed time series at the local scale, The longitude and latitude are extracted to verify the spatial structure of the dataset.

```r
#  Extract longitude, latitude, and time
lon_obsrp  <- ncvar_get(obsrp, "longitude")
lat_obsrp  <- ncvar_get(obsrp, "latitude")
range(lon_obsrp)
```

```
## [1] -3.12 -2.12
```

```r
range(lat_obsrp)
```

```
## [1] 46.66 47.66
```

The extracted longitude and latitude ranges are consistent with the expected domain.

The time variable and the two wind components (u10 and v10) are extracted from the NetCDF file. Since u10 and v10 are stored as three-dimensional arrays (longitude $\times$ latitude $\times$ time), the wind speed is computed at each grid cell and time step using the standard formula

$$V = \sqrt{u^2 + v^2}$$

resulting in a 3-D wind-speed array.

```r
#  Extract time
time_obsrp <- ncvar_get(obsrp, "valid_time")

#  Extract u10 and v10 wind components
u10 <- ncvar_get(obsrp, "u10")
v10 <- ncvar_get(obsrp, "v10")

# u10 and v10 are 3-dimensional arrays: lon × lat × time

# Compute wind speed at each grid cell and each time step
ws_array <- sqrt(u10^2 + v10^2)
```

Since the ERA5 wind data are provided as a three-dimensional field (lon $\times$ lat $\times$ time) rather than a single point measurement, a spatial average was computed across all grid cells to obtain a single representative wind-speed value for each month, as required for producing a one-dimensional time series for the CDF-t algorithm.

```r
# Compute the spatial average wind speed for each time step
# (i.e., one mean value per month)
ws_obsrp <- apply(ws_array, 3, mean, na.rm = TRUE)
```

The time variable in NetCDF files is typically stored as raw numeric values that represent elapsed time from a predefined origin date. Because of this, converting these numeric values into human-readable calendar dates requires reading the time metadata and reconstructing the actual dates. The following lines perform this conversion:

```r
# Convert the time variable to calendar dates
time_units_obsrp <- ncatt_get(obsrp, "valid_time", "units")$value
print(time_units_obsrp)
```

```
## [1] "seconds since 1970-01-01"
```

In NetCDF conventions, the units string specifies both the time unit (e.g., seconds, hours, days) and the reference origin from which time is measured.

Based on the units string, the origin of the time scale is defined. Here, the origin is converted into R's POSIXct datetime format, which allows arithmetic operations on dates and times. The POSIXct format stores date and time in seconds with the number of seconds beginning at January 1, 1970, so a POSIXct date-time is essentially an single value on a timeline. Date-times prior to 1970, will be negative numbers.

The tz = "UTC" ensures that the conversion is done in Coordinated Universal Time, consistent with the NetCDF standard. In NetCDF climate files, time is almost always expressed in UTC (Coordinated Universal Time), not in a local time zone. Setting tz = "UTC" tells R to interpret the origin date using the same global time standard as the NetCDF file. This prevents unwanted time shifts that would occur if R attempted to convert the timestamp into the user's local time zone.

```r
# Create a POSIXct object for the epoch start
origin_obsrp <- as.POSIXct("1970-01-01 00:00:00", tz = "UTC")

# Add time offset to origin to get dates
dates_obsrp <- origin_obsrp + time_obsrp

# Convert to Date format
dates_obsrp <- as.Date(dates_obsrp)

# 7) Close the NetCDF file
nc_close(obsrp)
```

Having the time vector converted to the Date format and the extracted wind speed, a data frame of the local measurements (**ObsRp**) can be created.

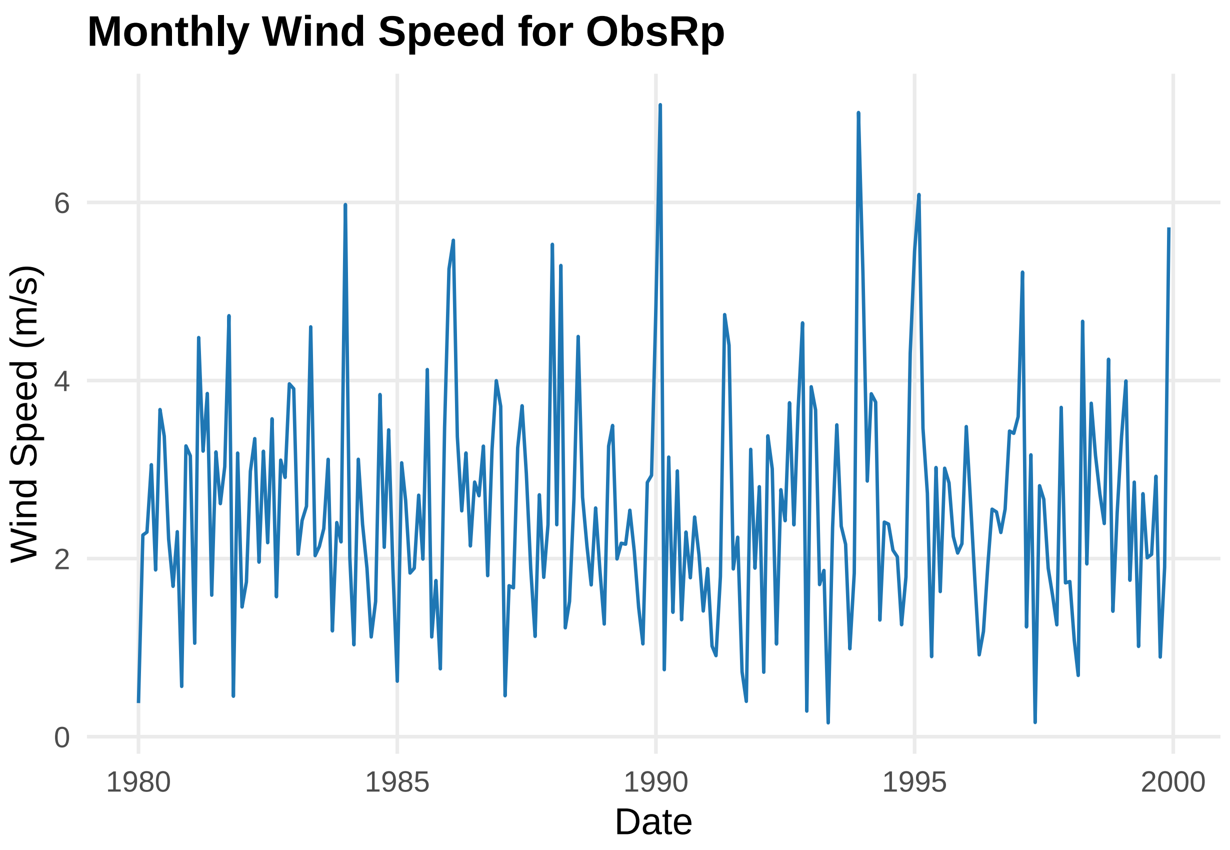
```r
# 8) Create a simple data frame
df_obsrp <- data.frame(date = dates_obsrp, ws = ws_obsrp)

# Initial check
head(df_obsrp, 3)
```

```
##         date        ws
## 1 1980-01-01 0.3778517
## 2 1980-02-01 2.2649159
## 3 1980-03-01 2.2975622
```

```r
tail(df_obsrp, 3)
```

```
##           date        ws
## 238 1999-10-01 0.8944554
## 239 1999-11-01 1.9021742
## 240 1999-12-01 5.7197141
```

```r
ggplot(df_obsrp, aes(x = date, y = ws)) +
  geom_line(color = "#1f77b4", linewidth = 0.6) +
  theme_minimal(base_size = 14) +
  labs(x = "Date", y = "Wind Speed (m/s)", title = "Monthly Wind Speed for ObsRp") +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank()
  )
```



The time series for the calibration period (**ObsRp**), provides a detailed view of wind variations measured in meters per second (m/s), with values ranging from 0 to over 6 m/s. The data reveals significant fluctuations, with noticeable peaks likely indicating severe weather events or seasonal changes, though no clear long-term increasing or decreasing trend is evident—suggesting natural variability over the 20-year period.

For **DataGp** the vector of large-scale time series The longitude and latitude are extracted to verify the spatial structure of the dataset.

```
# 2) Read coordinates, time, and surface wind variable
lon_datagp  <- ncvar_get(datagp, "lon")
lat_datagp  <- ncvar_get(datagp, "lat")

# Check longitude and latitude ranges
range(lon_datagp)
```

```
## [1] 357.5 357.5
```

```
range(lat_datagp)
```

```
## [1] 46.90141 46.90141
```

Although the download request specified a 0.5° × 0.5° spatial box around the site, CMIP6 outputs are provided on a coarser grid (approximately 1° resolution). Therefore, the dataset returns the nearest available model grid cell (357.5° E

$$\equiv$$

-2.5° W, obtained as 360° - 2.5°, and 46.90° N), rather than multiple points within the requested bounding box. This behavior is expected and confirms that the CMIP6 model provides only a single grid point for this location.

The time variable and the near surface wind field are extracted from the NetCDF file. The CMIP6 file stores time as “days since 1850-01-01”. To convert these values into calendar dates, the origin date is extracted from the units string, converted to POSIXct, and each time value is added as a number of elapsed days (× 86400 seconds). The result is a Date vector aligned with the model’s monthly wind-speed data.

```r
# Read time variable and wind field
time_datagp <- ncvar_get(datagp, "time")
ws_gp       <- ncvar_get(datagp, "sfcWind")

# Read time units
time_units_gp <- ncatt_get(datagp, "time", "units")$value
print(time_units_gp)
```

```
## [1] "days since 1850-01-01"
```

```r
origin_gp <- as.POSIXct("1850-01-01 00:00:00", tz = "UTC")
dates_gp  <- as.Date(origin_gp + time_datagp * 86400)
dates_gp <- as.Date(dates_gp)
```

A data frame is created to combine the converted dates with the corresponding CMIP6 wind-speed time series, allowing for easy inspection and further analysis.

```r
# Build a data frame containing the time series
df_gp <- data.frame(
  date = dates_gp,
  ws   = ws_gp
)

# Inspect results
head(df_gp, 3)
```

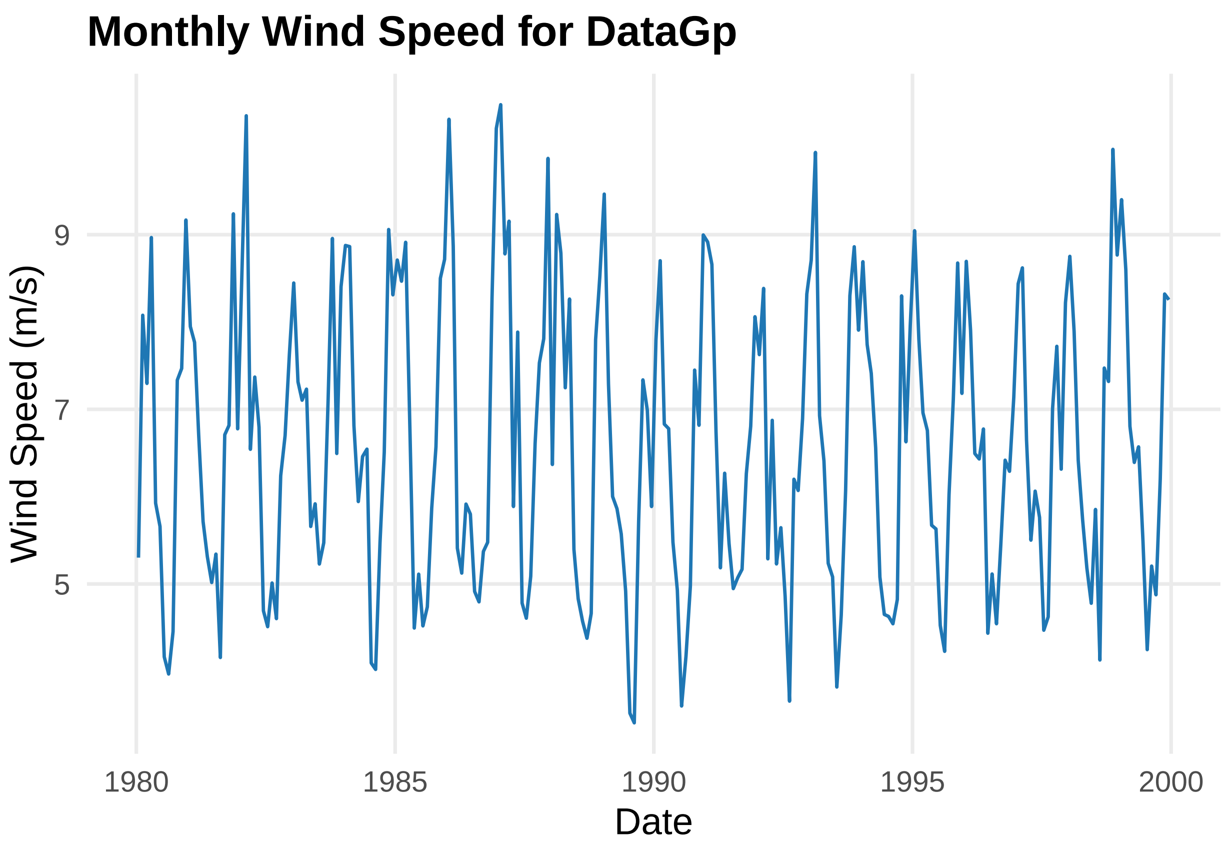
```
##         date       ws
## 1 1980-01-16 5.302005
## 2 1980-02-15 8.077882
## 3 1980-03-16 7.296510
```

```r
tail(df_gp, 3)
```

```
##           date       ws
## 238 1999-10-16 6.202218
## 239 1999-11-16 8.321090
## 240 1999-12-16 8.254923
```

```r
# Close the NetCDF file
nc_close(datagp)
```

```r
ggplot(df_gp, aes(x = date, y = ws)) +
  geom_line(color = "#1f77b4", linewidth = 0.6) +
  theme_minimal(base_size = 14) +
  labs(x = "Date", y = "Wind Speed (m/s)", title = "Monthly Wind Speed for DataGp") +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank()
  )
```



The time series for the historical large-scale model output (**DataGp**) shows monthly wind-speed variations simulated by the CMIP6 IPSL-CM6A-LR model for the 1980–1999 period. Compared with the ERA5-based calibration series (ObsRp), the CMIP6 historical simulation (DataGp) exhibits a noticeably smoother temporal structure. While the ERA5 series shows sharper fluctuations and more pronounced short-term variability—reflecting the high spatial and temporal resolution of reanalysis data—the DataGp series displays more moderate oscillations, with peaks and troughs that are less abrupt. This reduced variability is expected, as global climate models operate on a much coarser grid and therefore smooth out local extremes. Despite these differences, both series display similar overall seasonal patterns and neither shows evidence of a strong long-term trend over the 1980–1999 period.

For **DataGf** the CMIP6 future wind data (sfcWind) and its associated time variable are first extracted from the NetCDF file. Since the model encodes time as "days since YYYY-MM-DD", the origin date is retrieved from the units attribute and used to convert the numeric time values into calendar dates. The date and wind speed are then assembled into a data frame for subsequent analysis.

```r
## Read coordinates, time, and future wind field (sfcWind)

lon_datagf  <- ncvar_get(datagf, "lon")
lat_datagf  <- ncvar_get(datagf, "lat")
time_datagf <- ncvar_get(datagf, "time")
ws_gf       <- ncvar_get(datagf, "sfcWind")   # could be 1D or 2D depending on grid size

## Convert time values to calendar dates (time is in "days since YYYY-MM-DD")

time_units_gf <- ncatt_get(datagf, "time", "units")$value
print(time_units_gf)
```

```
## [1] "days since 1850-01-01"
```

```r
# Extract origin date from the units string
origin_str_gf <- sub(".*since ", "", time_units_gf)   # e.g., "1850-01-01"

# Convert to POSIXct
origin_gf <- as.POSIXct(paste0(origin_str_gf, " 00:00:00"), tz = "UTC")

# Convert days → seconds → calendar dates
dates_gf <- as.Date(origin_gf + time_datagf * 86400)

## Build a data frame

df_gf <- data.frame(
  date = dates_gf,
  ws   = ws_gf
)

head(df_gf,3)
```

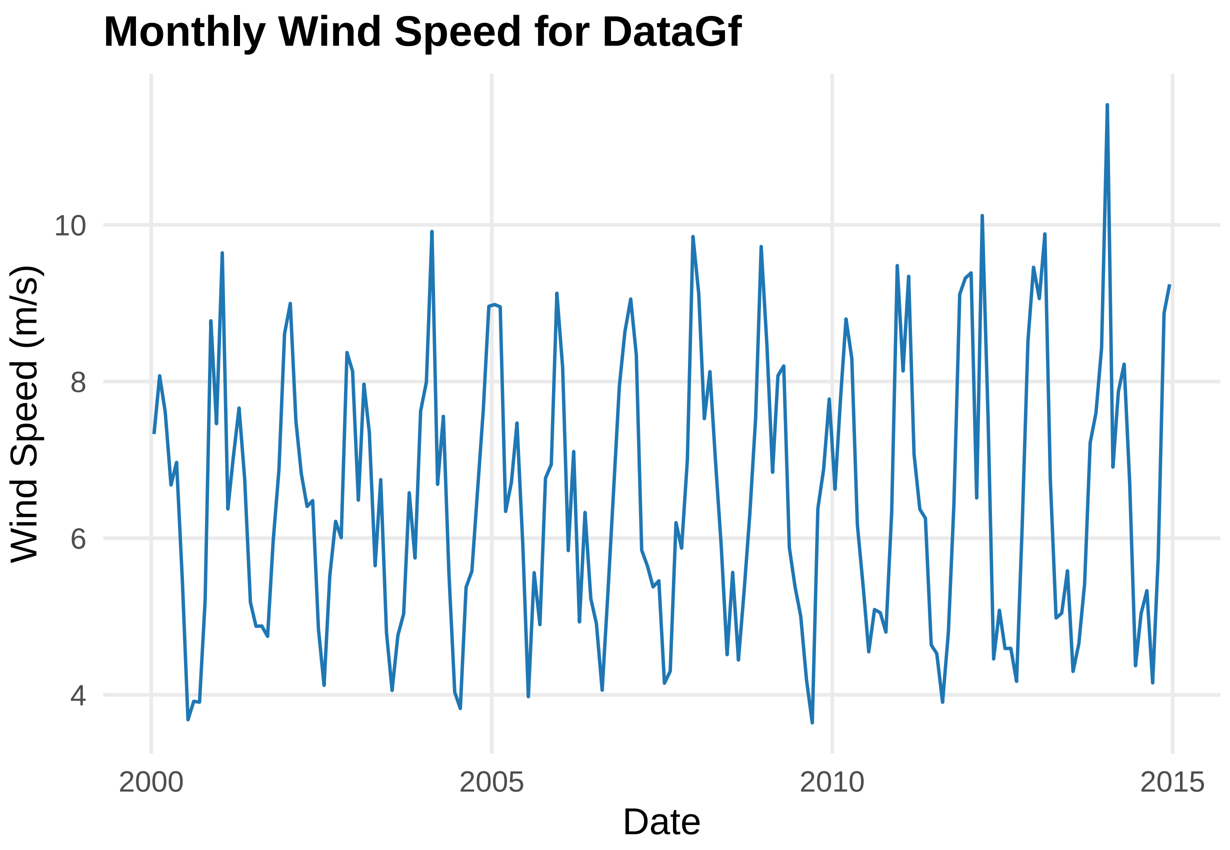
```
##         date       ws
## 1 2000-01-16 7.329406
## 2 2000-02-15 8.073177
## 3 2000-03-16 7.610505
```

```r
tail(df_gf,3)
```

```
##           date       ws
## 178 2014-10-16 5.766410
## 179 2014-11-16 8.871402
## 180 2014-12-16 9.241111
```

```r
## Close the NetCDF file

nc_close(datagf)
```

```r
ggplot(df_gf, aes(x = date, y = ws)) +
  geom_line(color = "#1f77b4", linewidth = 0.6) +
  theme_minimal(base_size = 14) +
  labs(x = "Date", y = "Wind Speed (m/s)", title = "Monthly Wind Speed for DataGf") +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank()
  )
```



The future wind-speed series (**DataGf**) shows clear seasonal fluctuations, with repeated peaks during windy months and noticeable drops during calmer periods. The pattern is highly variable, but no obvious long-term trend is visible over these 15 years.

The validation dataset (**ObsVal**) is extracted from the ERA5 reanalysis file by reading the longitude, latitude, time, and the two wind components (u10 and v10). Because ERA5 provides gridded data within the selected spatial box, the u10 and v10 variables form three-dimensional arrays (longitude × latitude × time). Wind speed is therefore computed at each grid cell and time step using the standard formula

$$V = \sqrt{u^2 + v^2}$$

and a spatial average is taken across all grid cells to obtain a single representative monthly value for the validation period.

The time variable is stored as seconds since 1970-01-01, so the origin date is defined accordingly and added to the raw time values to obtain calendar dates. The resulting wind-speed series is combined with the corresponding dates to create a validation data frame (df_obsval), which is used later to assess the performance of the downscaled CDF-t results.

```
## 1) Extract longitude, latitude, and time
lon_obsval  <- ncvar_get(obsval, "longitude")
lat_obsval  <- ncvar_get(obsval, "latitude")
range(lon_obsval)
```

```
## [1] -3.12 -2.12
```

```
range(lat_obsval)
```

```
## [1] 46.66 47.66
```

```r
time_obsval <- ncvar_get(obsval, "valid_time")

## 2) Extract u10 and v10 wind components
u10 <- ncvar_get(obsval, "u10")
v10 <- ncvar_get(obsval, "v10")

# At this stage, u10 and v10 are 3-dimensional arrays: lon × lat × time

## 3) Compute wind speed at each grid cell and each time step
ws_array <- sqrt(u10^2 + v10^2)

## 4) Compute a spatial average across all grid cells for each time step
# (i.e., a single representative wind-speed value per month)
ws_obsval <- apply(ws_array, 3, mean, na.rm = TRUE)

## 5) Convert the time variable to calendar dates
time_units_obsval <- ncatt_get(obsval, "valid_time", "units")$value
print(time_units_obsval)
```

```
## [1] "seconds since 1970-01-01"
```

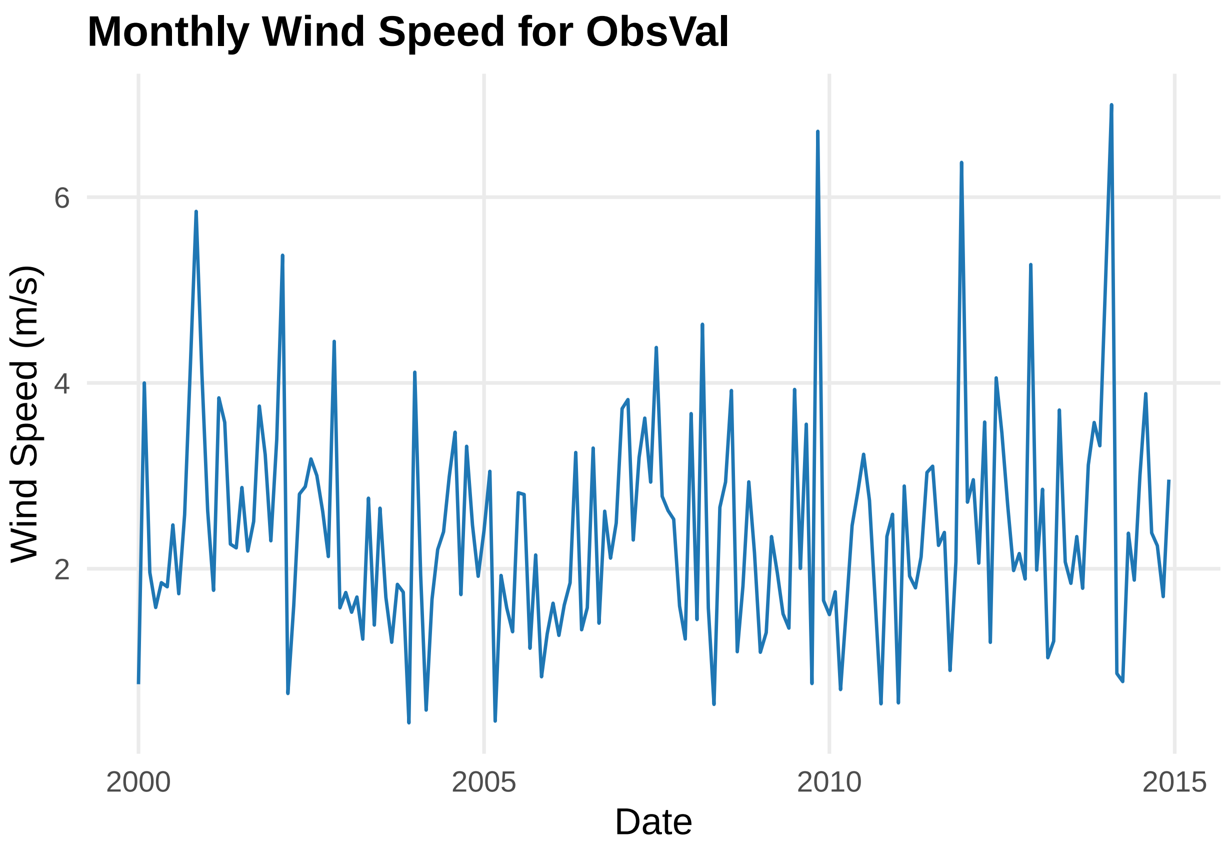
```r
origin_obsval <- as.POSIXct("1970-01-01 00:00:00", tz = "UTC")
dates_obsval  <- as.Date(origin_obsval + time_obsval)

## 6) Close the NetCDF file
nc_close(obsval)

## 7) Build a simple data frame containing the validation time series
df_obsval <- data.frame(date = dates_obsval, ws = ws_obsval)

# Quick inspection
head(df_obsval,3)
```

```
##         date        ws
## 1 2000-01-01 0.7572016
## 2 2000-02-01 4.0003285
## 3 2000-03-01 1.9602572
```

```r
tail(df_obsval,3)
```

```
##           date       ws
## 178 2014-10-01 2.248170
## 179 2014-11-01 1.700078
## 180 2014-12-01 2.959685
```

```r
ggplot(df_obsval, aes(x = date, y = ws)) +
  geom_line(color = "#1f77b4", linewidth = 0.6) +
  theme_minimal(base_size = 14) +
  labs(x = "Date", y = "Wind Speed (m/s)", title = "Monthly Wind Speed for ObsVal") +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank()
  )
```



The **ObsVal** time series (2000–2014) shows frequent month-to-month fluctuations, with several noticeable peaks and many periods of lower wind speeds. The overall pattern indicates high short-term variability typical of reanalysis data, with no clear long-term increasing or decreasing trend over the 15-year period.

To enable a month-by-month comparison between the ERA5 reanalysis data (ObsRp) and the CMIP6 historical model output (DataGp) over the 1980–1999 calibration period, both datasets were processed to extract the corresponding year and month from each timestamp. Since the two sources do not share identical day-of-month values (e.g., ERA5 timestamps typically occur on the first day of each month, while CMIP6 dates may fall mid-month), merging based on full dates would not correctly align the time series.

Using the extracted year and month fields, the two datasets were merged through an inner join, retaining only the months present in both series. A synthetic monthly date (set to the first day of each month) was then created to organize the merged dataset chronologically. This aligned dataset (hist_join) provides a consistent monthly basis for calibrating and comparing the large-scale model output with the observational proxy.

```r
hist_start <- as.Date("1980-01-01")
hist_end   <- as.Date("1999-12-31")

obs_hist <- df_obsrp %>%
  mutate(
    year  = year(date),
    month = month(date)
  )

gp_hist <- df_gp %>%
  mutate(
    year  = year(date),
    month = month(date)
  )
```

```r
head(obs_hist,3)
```

```
##         date        ws year month
## 1 1980-01-01 0.3778517 1980     1
## 2 1980-02-01 2.2649159 1980     2
## 3 1980-03-01 2.2975622 1980     3
```

```r
head(gp_hist,3)
```

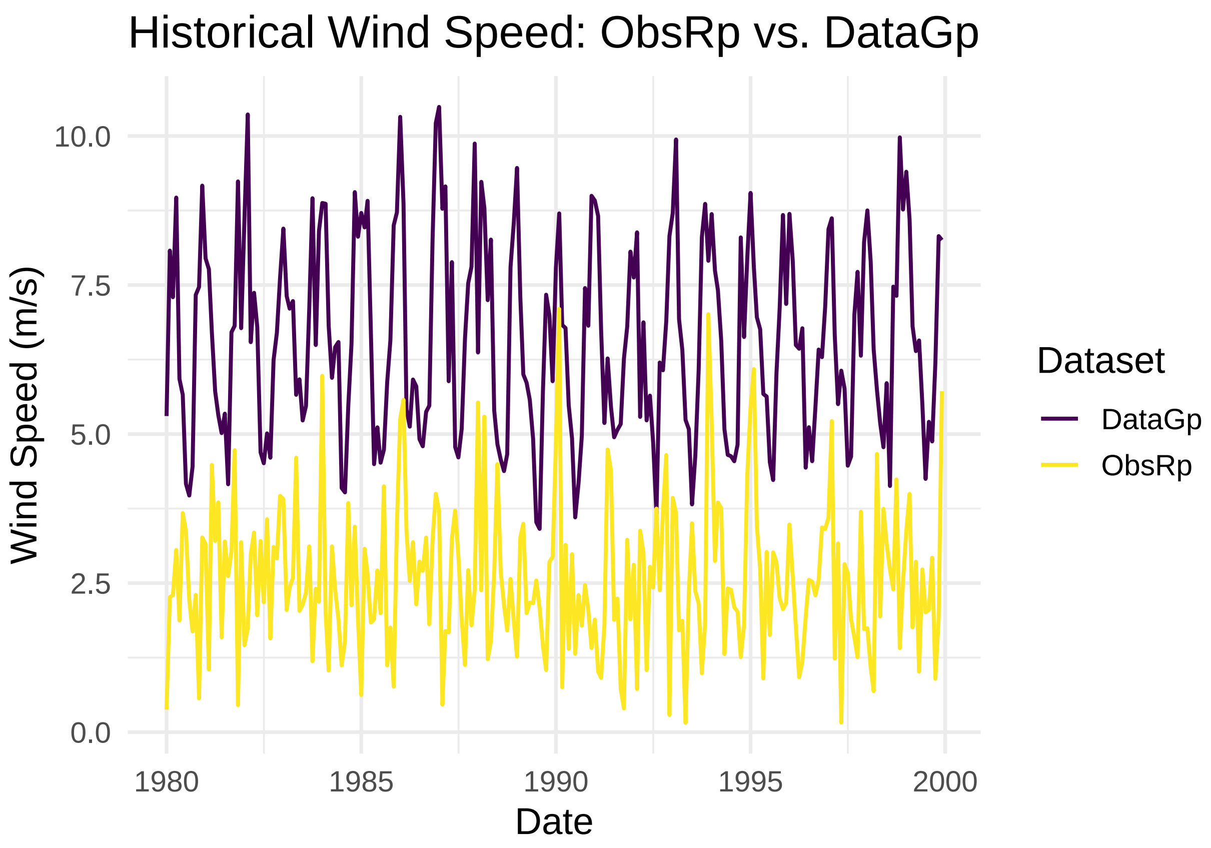
```
##         date       ws year month
## 1 1980-01-16 5.302005 1980     1
## 2 1980-02-15 8.077882 1980     2
## 3 1980-03-16 7.296510 1980     3
```

```r
hist_join <- inner_join(
  obs_hist %>% select(year, month, ws_obs = ws),
  gp_hist  %>% select(year, month, ws_gp  = ws),
  by = c("year", "month")
)

hist_join <- hist_join %>%
  mutate(date_month = as.Date(paste(year, month, "01", sep = "-"))) %>%
  arrange(date_month)

head(hist_join,3)
```

```
##   year month    ws_obs    ws_gp date_month
## 1 1980     1 0.3778517 5.302005 1980-01-01
## 2 1980     2 2.2649159 8.077882 1980-02-01
## 3 1980     3 2.2975622 7.296510 1980-03-01
```

```r
ObsRp  <- hist_join$ws_obs
DataGp <- hist_join$ws_gp
DataGf <- df_gf$ws
```

```r
# Build a data frame for the historical period
df_plot_hist <- data.frame(
  date   = hist_join$date_month,   # Aligned monthly dates
  ObsRp  = ObsRp,        # ERA5-based observational proxy
  DataGp = DataGp          # Historical GCM output
)

# Convert to long format for ggplot
df_long_hist <- pivot_longer(
  df_plot_hist,
  cols = c("ObsRp", "DataGp"),
  names_to = "series",
  values_to = "ws"
)

# Plot the historical comparison
ggplot(df_long_hist, aes(x = date, y = ws, color = series)) +
  geom_line(linewidth = 0.7) +
  scale_color_viridis_d(name = "Dataset") +
  theme_minimal(base_size = 14) +
  labs(
    title = "Historical Wind Speed: ObsRp vs. DataGp",
    x = "Date",
    y = "Wind Speed (m/s)"
  )
```



This figure shows a comparison between the observation-based reference dataset (ObsRp) and the historical GCM output (DataGp) over the 1980–1999 period. The GCM systematically overestimates wind speed relative to the observational reference, producing consistently higher values across all months. In addition to a higher mean level, the GCM also displays a larger amplitude of variability, with stronger seasonal oscillations compared to ObsRp. This persistent positive bias highlights the need for statistical downscaling—such as the CDF-t method—to correct the distribution and bring the GCM output closer to local observed conditions.

4.CDF-t Downscaling

The CDF-t algorithm was applied to the three aligned time series: ObsRp, DataGp, and DataGf. The function returns a list containing the empirical quantile grid (x), the calibrated historical CDFs (FRp and FGp), the future GCM CDF (FGf), and the downscaled future CDF (FRf). The component DS contains the downscaled monthly wind-speed values for the 2000–2014 period. These results provide the future local-scale projection consistent with both the observed historical distribution and the modeled large-scale changes.

```
res_cdft <- CDFt(ObsRp, DataGp, DataGf)

str(res_cdft)
```

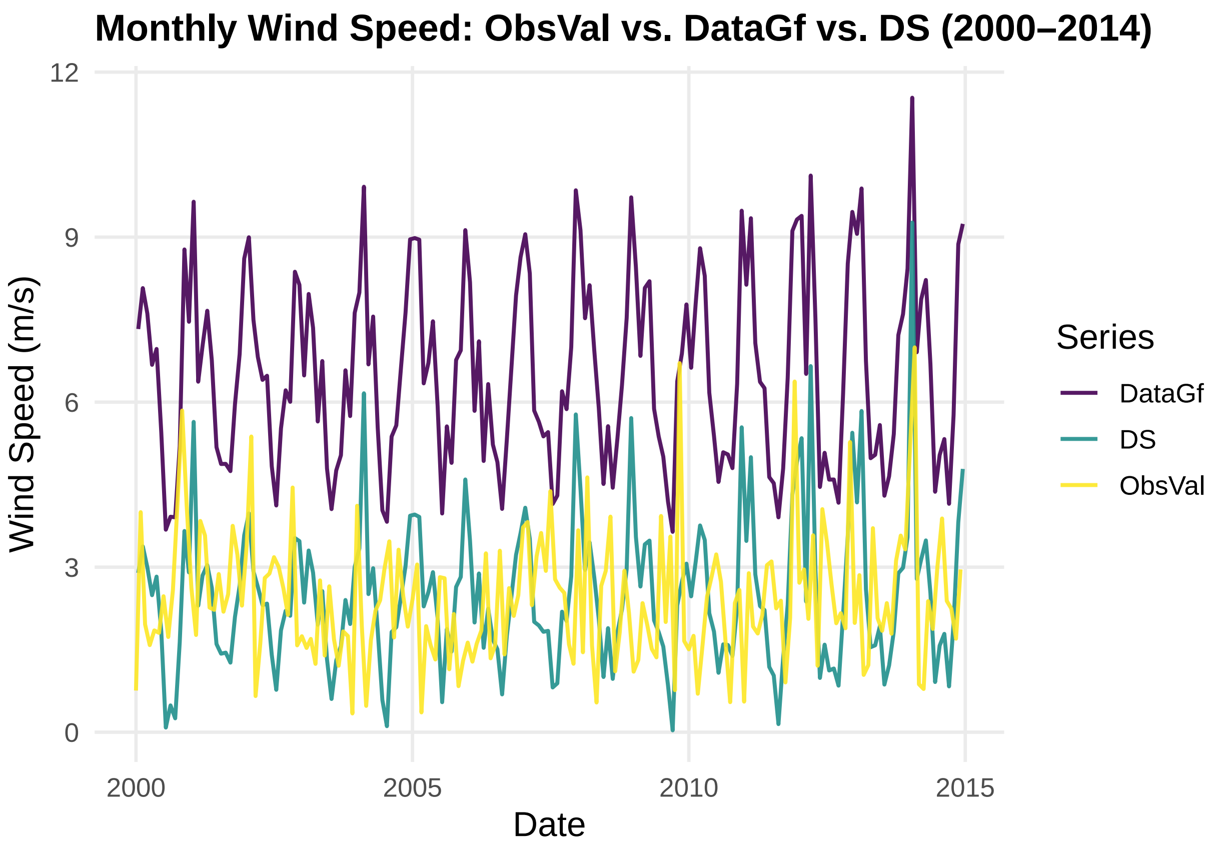
```
## List of 6
##  $ x  : num [1:1000] 0.0339 0.0455 0.0572 0.0688 0.0804 ...
##  $ FRp: num [1:1000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ FGp: num [1:1000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ FGf: num [1:1000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ FRf: num [1:1000] 0.00833 0.00833 0.00833 0.00833 0.00833 ...
##  $ DS : num [1:180] 2.9 3.38 3 2.49 2.83 ...
```

```
# Extract downscaled future series from CDFt output
DS <- res_cdft$DS   # downscaled monthly wind speeds for 2000-2014

# Build a data frame aligned with the GCM future dates
df_cdft <- data.frame(
  date = df_gf$date,  # same dates as future GCM series
  ws   = DS
)

# Quick check
head(df_cdft,3)
```

```
##         date       ws
## 1 2000-01-16 2.895070
## 2 2000-02-15 3.377435
## 3 2000-03-16 2.996623
```

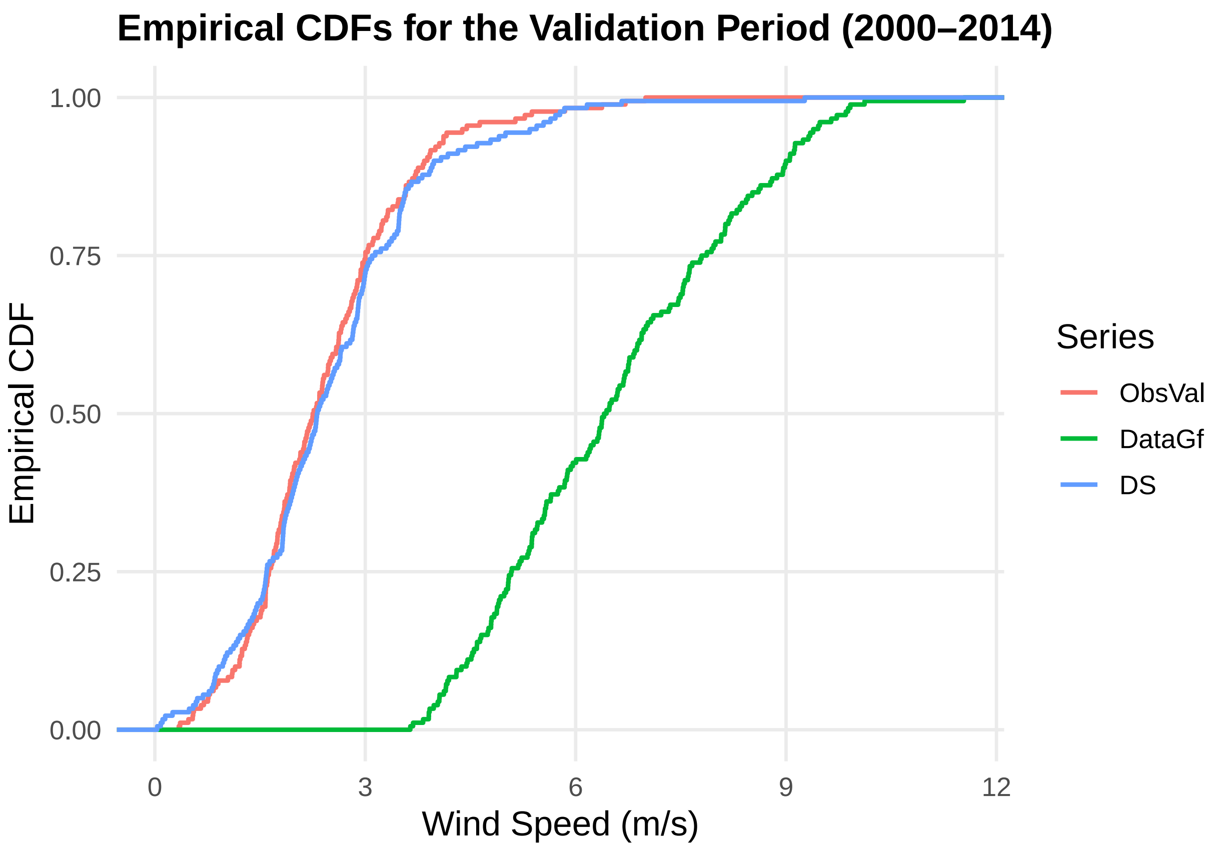
The CDF-t function returns the downscaled future series in the component DS. This vector is combined with the GCM future dates to construct a data frame (df_cdft) containing the CDF-t–corrected monthly wind-speed projections for 2000–2014.

```
# Put all future-period series into one long data frame
df_val_ts <- bind_rows(
  df_obsval %>% mutate(source = "ObsVal"),
  df_gf     %>% mutate(source = "DataGf"),
  df_cdft   %>% mutate(source = "DS")
)

# Time-series plot
ggplot(df_val_ts, aes(x = date, y = ws, color = source)) +
  geom_line(linewidth = 0.7, alpha = 0.9) +
  scale_color_viridis_d(option = "D") +
  theme_minimal(base_size = 13) +
  labs(
    x = "Date",
    y = "Wind Speed (m/s)",
    title = "Monthly Wind Speed: ObsVal vs. DataGf vs. DS (2000-2014)",
    color = "Series"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    panel.grid.minor = element_blank()
  )
```



For the validation period (2000–2014), the monthly wind-speed series for validation (**ObsVal**), the raw future GCM output(**DataGf**), and the CDF-t downscaled GCM series are plotted together. The DataGf series shows a clear positive bias, with consistently higher wind speeds than ObsVal. CDF-t effectively corrects this bias: DS distribution, seasonal variability, and amplitude closely match ObsVal. The downscaled series reproduces the observed interannual and intra-annual fluctuations while preserving the long-term structure of the GCM signal. This indicates that the CDF-t calibration successfully aligns the GCM wind-speed climatology with the observational reference.

```r
# Prepare data in long format for ECDF comparison
df_val_ecdf <- df_val_ts %>%
  mutate(source = factor(source,
                         levels = c("ObsVal",
                                    "DataGf",
                                    "DS")))

# Empirical CDF plot
ggplot(df_val_ecdf, aes(x = ws, color = source)) +
  stat_ecdf(size = 0.8) +
  theme_minimal(base_size = 13) +
  labs(
    x = "Wind Speed (m/s)",
    y = "Empirical CDF",
    title = "Empirical CDFs for the Validation Period (2000-2014)",
    color = "Series"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    panel.grid.minor = element_blank()
  )
```



The ECDF comparison shows that the CDF-t downscaled series closely reproduces the observed wind-speed distribution, while the raw GCM data exhibits a strong positive bias with substantially higher wind speeds across most quantiles.

5. Evaluation

To quantitatively evaluate the performance of the CDF-t downscaling, the Kolmogorov–Smirnov (KS) and Cramér–von Mises (CvM) statistics are computed between the downscaled future series (DS) and the observed validation series (ObsVal) over the 2000–2014 period.

Kolmogorov–Smirnov (KS) statistic

The KS statistic quantifies the maximum absolute difference between the empirical cumulative distribution functions (ECDFs) of two samples. Its value lies between 0 and 1, where 0 indicates identical distributions and 1 corresponds to the largest possible divergence. This makes the KS statistic a direct measure of how far two distributions can deviate at their most different point.

```
ObsVal    <- df_obsval$ws
ks_raw  <- KolmogorovSmirnov(S1 = DataGf,  S2 = ObsVal)
ks_cdft <- KolmogorovSmirnov(S1 = DS, S2 = ObsVal)

ks_raw
```

```
## [1] 0.8847491
```

```
ks_cdft
```

```
## [1] 0.05867434
```

The raw GCM data yields a KS statistic of 0.8847, which is close to the upper bound. This indicates a substantial mismatch between the raw modelled distribution and the observed validation data. After applying the CDF-t downscaling method, the KS statistic decreases to 0.0587, a value near zero. This sharp reduction demonstrates that the downscaled series closely reproduces the empirical distribution of the observations, confirming a strong improvement in distributional agreement.

Cramér–von Mises (CvM) statistic

The two-sample Cramér–von Mises statistic measures the integrated squared difference between the empirical CDFs of two samples across the entire support. Unlike the KS statistic, which captures only the maximum pointwise discrepancy, the CvM statistic reflects global distributional differences. Its value is always non-negative and equals zero only when the two empirical distributions coincide.

```r
# 3) Cramér-von Mises statistics using CDFt::CramerVonMisesTwoSamples
cvm_raw  <- CramerVonMisesTwoSamples(S1 = DataGf,  S2 = ObsVal)
cvm_cdft <- CramerVonMisesTwoSamples(S1 = DS, S2 = ObsVal)

cvm_raw
```

```
## [1] 27.12833
```

```r
cvm_cdft
```

```
## [1] 0.07552469
```

The raw GCM data produces a CvM statistic of 27.13, indicating a very large global discrepancy between the raw modelled distribution and the observed validation series. After applying the CDF-t downscaling, the CvM statistic falls dramatically to 0.0755. This near-zero value shows that the downscaled series closely matches the observed distribution across all quantiles. The comparison highlights a substantial improvement in overall distributional fidelity achieved by the downscaling procedure.

Conclusion

This study implemented a complete and reproducible CDF-t downscaling workflow to correct CMIP6 wind-speed projections over the Saint-Nazaire offshore wind-farm region. ERA5 reanalysis was used as a proxy for local observations, while CMIP6 historical (1980–1999) and future-boundary (2000–2014) data provided the large-scale inputs required for calibration and validation.

The CDF-t method successfully adjusted the distribution of the raw GCM series toward that of the observed local winds. Empirical CDF comparisons, together with the Kolmogorov–Smirnov and Cramér–von Mises statistics, showed that the downscaled future series is significantly closer to the observed validation data than the uncorrected GCM output. These results confirm that CDF-t improves the representation of local wind-speed variability and magnitude, and can therefore be a useful tool for bias correction and statistical downscaling in coastal and offshore wind-energy applications.