Spatio-temporal analysis of extreme wind velocities for infrastructure design

Dissertation submitted in partial fulfillment of the requirements

for the Degree of Master of Science in Geospatial Technologies

Jan 2020

Alexys Herleym Rodríguez Avellaneda

alexyshr@gmail.com

https://github.com/alexyshr

Supervised by:

Prof. Dr. Edzer Pebesma

Institute for Geoinformatics

University of Münster - Germany

Co-supervised by:

Prof. Dr. Juan C. Reyes

Department of Civil and Environmental Engineering

Universidad de los Andes - Colombia

Co-supervised by:

Prof. Dr. Sara Ribero

Information Management School

Universidade Nova de Lisboa - Portugal

---

ifgi
Institut für Geoinformatik
Universität Münster

NOVA IMS Information Management School

UNIVERSITAT JAUME I

Declaration of Academic Integrity

I hereby confirm that this thesis on *Spatio-temporal analysis of extreme wind velocities for infrastructure design* is solely my own work and that I have used no sources or aids other than the ones stated.

All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

February 24, 2020

_____

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

February 24, 2020

_____

Acknowledgements

Special thanks to Prof. Dr. **Edzer Pebesma**, first, for all the contributions to the open source community, considering that main work in this thesis was done using his R packages, especially `sf`, `stars` and `gstat`, and second, for all high level knowledge transmitted through the subjects *Spatial Data Science with R* and *Analysis of Spatio-Temporal Data*, which were the motivation and basis to carry out the investigation.

Special thanks to Prof. Dr. **Juan C Reyes** for his contribution in selecting the research topic, and great contributions in information, methodology and support.

Gratitude is extended to Dr. **Christoph Brox**, for being a support in difficult moments as the surgery and COVID-19 crisis, and in the same way to **Karsten Höwelhans**.

The author is thankful to:

Prof. Dr. **Edzer Pebesma**, Prof. Dr. **Juan C. Reyes**, and Prof. Dr. **Sara Ribero**, for supervising this work and spending their valuable time for discussions and feedback, it was really a huge advantage to have that support always available, and a pleasure to work beside them. Dr. **Adam Pintar**, for sharing its related POT-PP R Code, and for devoting much of his time to reviewing and commenting on my progress. The outstanding help of Dr. **Joaquín Huerta Guijarro**, for being receptive, friendly, and decidedly available to help. **European Union** 'Erasmus Mundus Grant', their funding allows me to fulfill this dream to go further with academic and professionals dreams. Engineer **Juan David Sandoval** for its helpful contributions. **Ligia Avellaneda** and **Nicolle Chaely**, mother and daughter of the author, for your love, prayers, and prized advice. Family members as **Elsa Manrique**, **Barbara Avellaneda**, and **Kevin Martinez**, for their really important source of motivation and accompaniment. To all the beautiful people that shared with the author different activities at **San Antonius Church of Münster**, with special mention of father **Alejandro Serrano Palacios** for his outstanding help and friendship which is permanently appreciated, and **choir friends**.

Table of Contents

List of Tables

List of Figures

Abstract

This research aims to create non-hurricane non-tornadic maps of extreme wind speeds for the *mean recurrence intervals* MRIs 700, 1700, and 3000 years, covering the Colombian territory. For infrastructure design, these maps are combined with existing hurricane wind speed studies, to be used as input loads due to wind.

For each station with non-thunderstorm wind speeds time histories in the input data, following (Pintar, Simiu, Lombardo, & Levitan, 2015), extreme wind speeds corresponding to each MRI are calculated using a *Peaks Over Threshold Poisson Process POT-PP* extreme value model, then wind velocities with the same MRI are *spatially interpolated* to generate continuous maps for the whole study area. The annual exceedance probability for all velocity values in 700, 1700 and 3000 years MRIs output maps are respectively 1/700, 1/1700 and 1/3000.

Regarding input data, not only time series of field measurements from IDEAM methodological stations are used, but also post-processed information coming from the Integrated Surface Database ISD, and ERA5 forecast reanalysis data. This condition demanded a comparison of the different data sources, in order to verify the feasibility in the use of ISD and ERA5, this is downscaling support. The result of the comparison showed little similarity between the different sources, but taking into account that complete and adequate measured data from IDEAM was not available. Before to apply POT-PP, ISD and IDEAM data sources were standardized to meet the requirement of three seconds (3-s) wind gust speed, ten (10) meters anemometer height, and terrain open space condition.

Due to the limitation in the classification of thunderstorm and non-thunderstorm data, it was not possible to take real advantage of POT-PP method, which was limited/restricted from non-homogeneous to homogeneous and from non-stationary to stationary, being equivalent to use the most common POT - generalized Pareto approach. Non-hurricane maps were created for data sources ISD and ERA5, using Kriging as spatial interpolation method. After the integration with previous hurricane studies, the results of ERA5 showed the most reliable final maps, despite limitations in the input data to guarantee downscaling support. ISD final map showed very high wind values, which are unlikely. These shortcomings may be corrected when complete IDEAM data-source and storm data classification are available.

A complete R tool was implemented to solve the whole process, which is based in copyrighted code for de-clustering and thresholding generously given by Dr Adam L. Pintar `adam.pintar@nist.gov` - National Institute of Standards and Technology NIST, U.S Department of Commerce.

List of Acronyms

AIS	Colombian Earthquake Engineering Association
ASCE	American Society of Civil Engineers
ASCE7-16	ASCE/SEI Design Loads Standard
CDF	Cumulative Distribution Function
EDA	Exploratory Data Analysis
ECMWF	European Centre for Medium-Range Weather Forecasts
ERA5	ECMWF climate reanalysis dataset
EVD	Extreme Value Distribution (GEVD, GEV)
GEVD	Generalized Extreme Value Distribution (EVD, GEV)
GEV	Generalized Extreme Value Distribution (GEVD, EVD)
GPD	Generalized Pareto Distribution
HF	Hazard Function
IDEAM	Institute of Hydrology, Meteorology and Environmental Studies
IDW	Inverse Distance Weighted
ISD	Integrated Surface Database
MRI	Mean Return Interval or Return Period
NSR	Seismic Resistant Norm
NOAA	National Oceanic and Atmospheric Administration
NetCDF	Network Common Data Form
NCEI	NOAA's National Centers for Environmental Information
P_e	Annual Exceedance Probability
PDF	Probability Distribution Function
P_n	Compound Exceedance Probability
POT	Peaks Over Threshold
PPF	Percent Point Function (Quantile)
PP	Poisson Process
Poisson-GPD	POT: 1D PP (time) and GPD (magnitude)
POT-PP	POT: 2D PP (time and magnitude)
RL	Return Level
RMSE	Root Mean Squared Error
SEI	Structural Engineering Institute
SQL	Structured Query Language
WGS84	World Geodetic System 1984

Chapter 1

Introduction

Extreme value models are used for estimating engineering design forces of *extreme events* like earthquakes, winds, rainfall, floods, etcetera (Beirlant, Goegebeur, Teugels, & Segers, 2004). Structures designed with these forces, holding a balance between safety and cost, will survive while being requested by an extreme event from a natural phenomenon (Castillo, Hadi, Balakrishnan, & Sarabia, 2005).

This research presents an application of extreme value analysis to estimate wind velocities for infrastructure design. Consequently, the main interest are probable future extreme events that structures need to be able to resist (Smith, 2004).

This research follows the methodological approach defined in chapter 26 of the ASCE7-16 standard (ASCE, 2017). ASCE7-16 considers design wind velocities for various mean recurrence intervals MRIs, depending on the risk category of the structure, as follows: MRI=700 years for risk category (RC) I and II, 1700 years for RC III, and 3000 for RC IV. A wind speed linked to a *mean recurrence interval - MRI* of *N-years* (N-years return period) is interpreted as the highest probable wind speed along the period of N-years (ASCE, 2017). The annual probability of equal or exceed that wind speed is 1/N, this is with a change of being equaled or exceeded only one time in the corresponding MRI period.

The development of this research (focused in non-hurricane data), covers three main areas, *downscaling support*, *temporal analysis*, and *spatial analysis*, and includes an integration process with *existing results of hurricane studies*.

Due to the specific characteristics of the study area where there is lack of historical wind measurements, it became necessary to look for alternative data sources: ISD, and ERA5 forecast data. This resulted in a downscaling issue that was confronted from a graphic comparison of all sources by matching stations, in the search of adequate *downscaling support* for the use of complementary data. Prior to the comparison process, ISD and IDEAM data sources were standardized to represent 3-second wind gust, 10 meters of anemometer height, and terrain open space roughness.

The *temporal analysis* method used to calculate the return levels at each station from the historical wind time series, is the Peaks Over Threshold POT using a non-homogeneous bi-

dimensional Poisson Process PP recommended by (ASCE, 2017) and developed in (Pintar et al., 2015). Main components of POT-PP model are de-clustering, thresholding, intensity function fitting, hazard curve, and return levels calculation. At each station with non-thunderstorm data, this model starts with a process of de-clustering choosing a suitable threshold level to leave for the analysis only the most extreme available values, and then, fit to the data an intensity function using maximum likelihood to find optimal parameters with the best goodness of fit. With the fitted model, and using the hazard curve, it was possible to calculate extreme wind velocities or return levels for required MRIs.

The integration of all these results allow to generate non-hurricane continuous maps of extreme winds velocities (using *Kriging* as *spatial analysis* method), which are combined with *existing wind extreme hurricane studies* to be used as input loads for the design of structures of different risk categories, i.e., less risky/important structures for short MRIs (700 and 1700 years), and highly important structures for the longest MRI of 3000 years.

1.1 Context and Background

To design a specific structure, horizontal forces (wind and earthquake) play a starring role. For Colombia, initially, wind forces are calculated considering a fixed velocity value of 100 km/h, later, a continuous map with a return period of 50 years was included in the official design standard. Afterwards, an additional map with a return period of 700 years was added (Vivienda, 2010).

In the context of this study, extreme wind analysis is concerned with statistical methods applied to very high values of wind velocity as random variable in a stochastic process, to allow statistical inference from historical data. Extreme analysis methods assess the probability of wind events that are more extreme than the ones previously registered and included in the input model of the maximum wind velocities ordered sample. Coles (2001) presents a detailed study about classical extreme value theory and threshold models. Asymptotic extreme value models arguments give a convenient representation of the stochastic behavior of maximum values (Coles, 2003).

According to Coles (2003), there are four main elements needed for a good analysis of extreme values: (a) appropriate selection of an asymptotic model; (b) use of all pertinent and available information; (c) properly estimation of uncertainty; and (d) considering non-stationary effects.

In general, there are two approaches to deal with extreme value analysis (Pintar et al., 2015), the classical approach, and peaks over threshold POT. In this research, POT is selected over classical approach to be able to use more samples for statistical estimation.

The classical approach or traditional method, as well known as *sample maxima* or *yearly maxima* is associated to a generalized extreme value distribution GEV (Fisher & Tippett, 1928; Gnedenko, 1943). GEV is a family of limit probability distributions including Gumbel, Fréchet and Weibull, unified in (Jenkinson, 1955; Mises, 1954). The GEV family *describes all limiting distributions of the centered and normalized sample maximum* (Coles, 2003).

POT models the values above a chosen high level, and in general, the POT method has two approaches (Smith, 2004): (a) exceedances over threshold associated to a Generalized Pareto Distribution GPD, onwards *POT-Poisson-GPD*; and (b) the exceedances over threshold associated to a non-homogeneous non-stationary bi-dimensional Poisson process POT-PP (a point process approach). POT-PP is considering to be more flexible than generalized Pareto approach (Coles, 2001), so for this study, POT-PP method is selected.

Selection of threshold level is relevant in POT. A low threshold (more exceedances) implies less variance and weak asymptotic support, but high bias. A high threshold (fewer exceedances) implies more variance and stronger asymptotic support, but low bias.

POT-Poisson-GPD models wind magnitudes over the threshold as a GPD, and time as a separated Poisson process (Pickands III & others, 1975). This method was used for the first time as statistical application in (Davison & Smith, 1990). The generalized Pareto family describes all possible limiting distributions of the distributions scaled above the threshold (Coles, 2003).

In POT-PP time and magnitude above the threshold are modeled using a two-dimensional Poisson process (Pickands, 1971). This method was applied for the first time in (Smith, 1989).

There are many techniques to estimate the parameters of extreme value models, i.e. graphical methods, estimators based on moments, order statistics, and likelihood based (Coles, 2003). Smith (1985) supported the use of likelihood methods due to *asymptotic normality* guaranteed when shape parameter is greater than 0.5, excluding light tailed distributions with finite end point. In addition, likelihood method is easy to evaluate and solve numerically, and the calculation of standard errors and confidence intervals is possible using asymptotic theory.

1.2 Problem Statement and Motivation

Wind forces are important for infrastructure design (Comarazamy, 2005). For a civil engineer main forces to consider for the design of a structure, for instance a bridge or a building, are (a) dead load due to the weight of the structure, and (b) live load due to earthquake and wind. For Colombia, the structure design standard has defined in great detail, all aspects related to seismic forces, and dead forces, but lack of detail in design wind velocities. Current wind velocities map is 20 years outdated, and is not appropriate for all types of structures, because it only includes two return periods.

It is well known that in recent years there have been accelerated changes in the climate of the planet, including issues related to winds. This aspect is reflected in frequent partial failures of structures due to wind forces (Council, 1994), and in some cases including with total losses (Rezapour & Baldock, 2014). Last five decades the way to assess wind loads in structural design has had remarkable changes (Roberts, 2012).

A complete study of extreme wind forces, need to address separately hurricane and non-hurricane data, to include in the product the integration of results from both fronts (ASCE,

2017). In the study area, hurricane winds are only present inland in the Caribbean Sea, therefore, only affects directly 'San Andres y Providencia' island - one (1) of thirty-three (33) states. In 1102 of 1103 municipalities (more than 99%), only non-hurricane winds are relevant. However, all municipalities located near to the northern onshore border may be impacted by side effects of hurricanes.

The national infrastructure design standard of Colombia, maintained by the Earthquake Colombian Association of Seismic Engineering AIS, uses km/h as official units for wind velocities. In this research km/h is always used, considering that output results will support the update of chapter B.6 (wind forces) of mentioned standard.

1.3 Knowledge Gap

Nowadays, methodologies to deal with the inference of extreme wind maps are quite mature and advanced, and many of them are already implemented and ready for use. For this reason, the main contribution of this research is not related to the theoretical foundations of the methods themselves, but to application of the method in a particular case where good quality data is not available (ADB, 2014). Thereby, the gaps in which this research aims to contribute are related to the use of alternative data sources, and how to meet the downscaling challenge considering the lack of field measurement data coming from weather stations.

1.4 Research Aim and Objectives

The main aim of this research is the estimation of wind extreme velocities to be used as input loads for the design of structures, considering their risk categories, and covering any place in the whole study area.

Specific objectives are:

1. Complement the lack of field measured wind data, with other sources of information, then, analyze and compare different time series, to select and use the best data source (or combination of sources), in case of downscaling support issue.
2. Select and apply a suitable probabilistic method to infer wind maps for infrastructure design.
3. Estimate extreme wind values for the stations in the selected input data source, considering non-hurricane approaches.
4. Allow the comparison of wind extreme values estimations, using different methods to verify and calibrate output results.
5. Generate continuous non-hurricane wind maps, using the most suitable spatial interpolation technique, considering the specific characteristics of the input data source.
6. Combine output maps from non-hurricane analysis with existing hurricane studies to obtain final maps for structural purposes.

1.5 Research Question

Main question of this research is directed to calculate future wind extreme velocities (return levels) for infrastructure design, then the research is:

What wind extreme velocities need to be used as load design forces for structures of different use category, in the study area?

1.6 Case Study

As mentioned before, case study in this research is Colombia a tropical country located in the northern part of South America. Its capital Bogotá is located in the center of the country at latitude $4.6°N$, and longitude $74.1°W$.

Despite the government Institute of Hydrology, Meteorology and Environmental Studies IDEAM maintains a network of weather stations, of which around 200 have anemometers measuring instant data every minute, it was impossible to obtain quality and complete measured data according to the needs of the present investigation, motivation to search for alternative data sources.

Nowadays in Colombia there are predefined requirements to design structures depending of its use category. The national standard for infrastructure design (Vivienda, 2010), following the American design standard (ASCE, 2017), covers the design of all types of structures with the mean recurrence intervals MRIs 700, 1700, and 3000 years. In this way this research aims to calculate wind extreme velocities that will be equaled or exceeded with a probability equal to $\frac{1}{MRI}$ in a given year, in other words, the velocities that will be equaled or exceeded only one time in mentioned periods. In terms of exposure time, understood as the time the structure will be in use, when the exposure time will be equal to those MRIs, the wind extreme velocities will have an occurrence compound probability of 67%.

Historically, hurricanes have only affected the Colombian Caribbean coast in a not very significant way, despite the fact that there have been significant events that have made landfall. The most likely areas to be affected by storms are the department of *La Guajira* and the island of *San Andrés* (Royero, 2011). In most cases the events that define the wind design loads in Colombia do not require hurricane data.

1.7 Outline

Main sections of thesis document are 1) Introduction, 2) Data, 3) Theoretical Framework, 4) Methodology, 5) Results and Discussion, 5) Conclusions, and 6) Annexes, from A to E.

After introduction, in second section *Data*, main information about data sources IDEAM, ISD, and ERA5 are described, including at the end additional details for ERA5 in Annex C. Annex D explains reasons for using PostgreSQL engine, and the database backup and restoration process.

Theoretical Framework section is dedicated to introduce statistical concepts that are basis for the investigation, both in **probability distributions** and in **extreme analysis**. Later, it is described in more detail, topics related to **extreme value analysis** (peaks over threshold with generalized Pareto POT-Poisson-GPD, and peaks over threshold with Poisson process POT-PP), and at the end, a summary report is done about **wind load requirements** for the study.

The *Methodology* chapter includes the processes needed to meet the objectives and answer the research question. Main components are data standardization, downscaling support, POT-PP, spatial interpolation, and integration with hurricane data.

Results and Discussion section shows, (1) all results for data standardization and comparison to support the downscaling issue, (2) all POT-PP results for one ISD station, (3) all output maps for ISD and ERA5 data sources including discussions of those finals results. These discussions are complemented by the *Conclusions* section.

To finalize the document, a series of appendices were created to facilitate the reproducibility of the research. Appendix A contains *research R code*. It is necessary to considering that the code provided by *Dr. Adam L. Pintar* to do the de-clustering and thresholding in POT-PP is not there because its publication and distribution is not authorized. Appendix B contains all *results in digital format*. Appendix C compliments the information needed to *download ERA5* data. Appendix D shows the use of PostgreSQL for data storage, and provides instructions for backup and restoring. Because the document for the thesis was done using package ´thesisdown´ (which is based in ´bookdown´) the most important *document R code* to create the thesis document, mainly graphics, is shown in Appendix E. Finally, in Appendix F F, a *user manual* is presented, in order to provide instruction to apply the same methodology in a different case study.

Chapter 2

Data

Input data was obtained from three different sources (a) Institute of Hydrology, Meteorology and Environmental Studies of Colombia IDEAM `http://www.ideam.gov.co`, (b) Integrated Surface Database ISD `https://www.ncdc.noaa.gov/isd`, and (c) climate reanalysis ERA5 `https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5`. Information about datasets, variables, and units is found in tables 2.1, 2.2, and 2.3 respectively.

Table 2.1: Datasets description

Institution	Dataset	Details
IDEAM	Historical records at weather stations	IDEAM is responsible for the installation, maintenance and management of all kind of weather stations located everywhere along the country
NOAA	ISD	ISD (Integrated Surface Database. NOAA's National Centers for Environmental Information - NCEI) Lite: A subset from the full ISD dataset containing eight common surface parameters in a fixed-width format free of duplicate values, sub-hourly data, and complicated flags.
ECMWF	ERA5	ERA5 is a reanalysis dataset with hourly estimates of atmospheric variables with horizontal resolution of 0.25^{o} (33 kilometers), i.e. cells equally spaced every 0.25 degrees

Table 2.2: Variables in all datasets

Dataset	Variables	Description
IDEAM	vv_aut_2	Instantaneous wind velocity each two (2) minutes
	vv_aut_10	Instantaneous wind velocity each ten (10) minutes
	vvmx_aut_60	Maximum wind velocity each sixty (60) minutes
ISD	v5	Maximum hourly five seconds (5-s) wind gust velocity
ERA5	fg10	10 meters wind gust since previous post-processing
	fsr	Forecast Surface Roughness

Table 2.3: Variables Units and Time

Variable	Units	Time	Stations
vv_aut_2	meters per second	Variable	20
vv_aut_10	meters per second	Variable	204
vvmx_aut_60	meters per second	Variable from 2001 until today. Irregular time series.	203
v5	meters per second	Variable from 1941 until today. Note: There is too much variability in time (start, end, and time range) for each station. Irregular time series.	101
fg10	meters per second	1979-Today	3381
fsr	meters per second	1979-Today	3381

Ideal data source to create extreme wind speeds maps should be field observed data from IDEAM, but the IDEAM data have the following deficiencies:

1. There are not enough number of stations around the study area to represent all the local wind variability in a huge country with multiple variety of climates and changing thermal floors.
2. There are uncertainties related to the way anemometers are registering data, then comparison with other data sources are needed to be able to do appropriate data standardization, needed as a prerequisite for the analysis.
3. There is no time continuity in the registration of IDEAM data. Historical time series are different and variable in each station.
4. The different wind related variables provided by IDEAM are not well documented and their comparative values for identical time periods are not consistent with each other.

Other logistical difficulties in accessing IDEAM data source are also highlighted:

1. It was not possible to obtain the wind hourly mean variable.
2. It was impossible for them to calculate and deliver a representative gust velocity, for example *5-minutes gust*.
3. There was also no access to complete raw data (instantaneous measurements every minute that they claim to have), to be able to calculate from them the variables required for the study.

Importance of ISD database for this study is based on the fact that post-processed ISD database has wind extreme values, and it was used to create extreme wind maps for United States. ISD allows comparison with IDEAM records to take better decisions in order to conduct data standardization. Despite that ERA5 data are not observed data (but forecast), its main advantage is their resolution (0.25 square decimal degrees) and availability.

2.1 IDEAM

Historical observed wind speeds from 203 stations in Colombia are managed by the official environmental authority IDEAM. Table 2.4 shows a sample of ten IDEAM stations. Figure 2.1 shows a map of IDEAM stations. Figure 2.2 shows data for IDEAM station "21205791".

Table 2.4: IDEAM Stations Sample

Name[Code]	Latitude	Longitude
EMAS - AUT [26155230]	5.09	-75.51
SAN BENITO - AUT [25025380]	9.16	-75.04
AEROPUERTO ALFONSO LOPEZ - [28025502]	10.44	-73.25
TIBAITATA - AUT [21206990]	4.69	-74.21
ELDORADO CATAM - AUT [21205791]	4.71	-74.15
LA LAGUNA DE CAJIBIO [26035090]	2.70	-76.60
SILOE - AUT [26085160]	3.43	-76.56
METROMEDELLIN - AUT [27015310]	6.33	-75.55
JARDIN BOTANICO - AUT [21205710]	4.67	-74.10
AEROPUERTO A.BONILLA - AUT [26075150]	3.53	-76.38

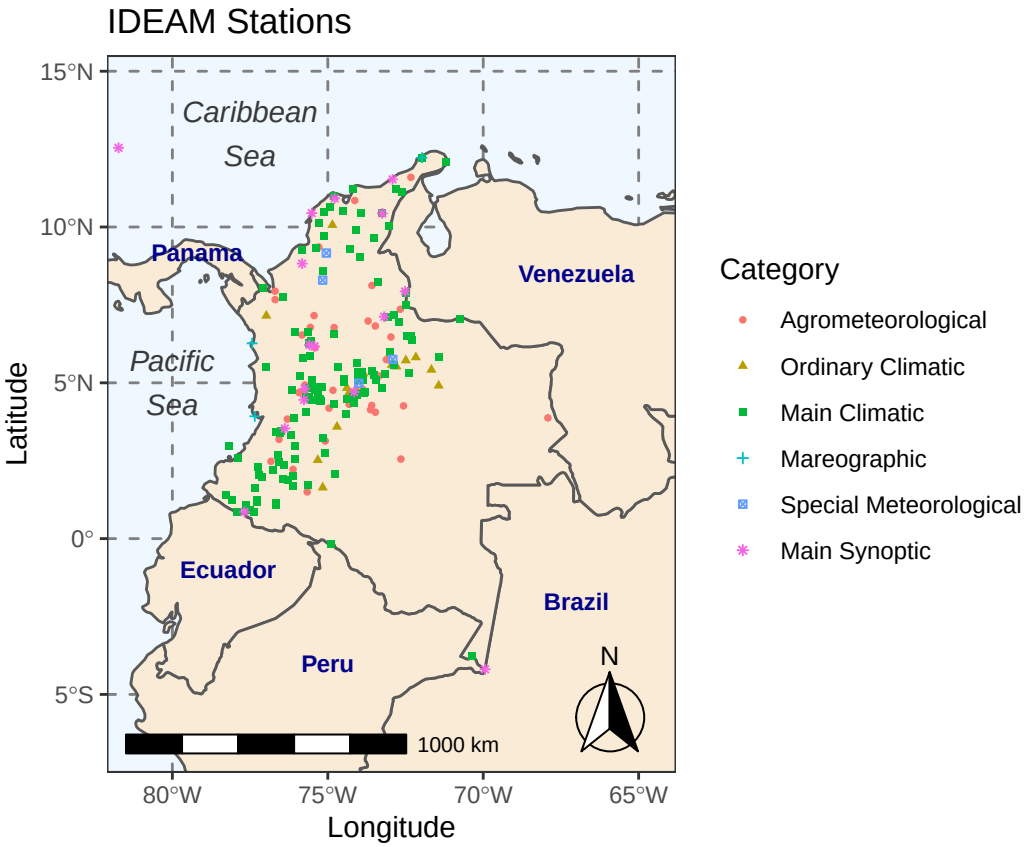

Figure 2.1: IDEAM Stations. Colombia

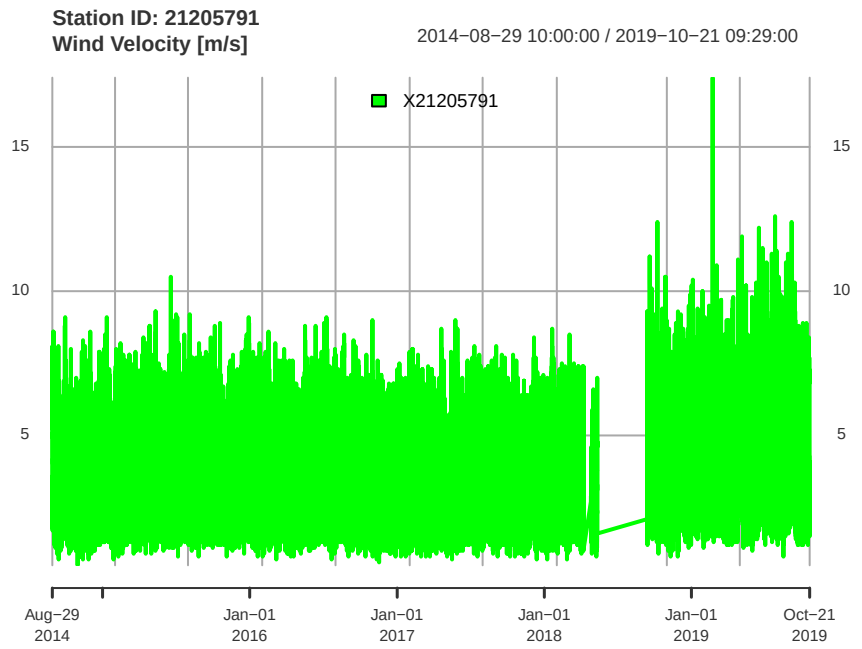

Figure 2.2: Time Series of IDEAM Station ELDORADO CATAM - AUT

2.2 ISD

ISD is a database with environmental variables, among them extreme wind speeds. ISD has data for the whole planet, and is based on observed data at meteorological stations in each country, which means that for Colombia is based on IDEAM data. Main advantage is data availability at neighbor countries and specialized post-processing made by NOAA's National Centers for Environmental Information NCEI in United States, which facilitates its use. Table 2.5 shows a sample of ten ISD stations. Figures 2.3 and 2.4 shows a map of ISD stations and data from ISD station "802590".

Table 2.5: ISD Stations Sample

Code	Name	Latitude	Longitude
804400	BARINAS	8.62	-70.22
800810	ALTO CURICHE	7.05	-76.35
801000	BAHIA SOLANO / JOSE MUTIS	6.18	-77.40
802590	ALFONSO BONILLA ARAGON INTL	3.54	-76.38
803150	BENITO SALAS	2.95	-75.29
801100	OLAYA HERRERA	6.22	-75.59
802190	GIRARDOT/SANTIAGO VILLA	4.28	-74.80
802410	LAS GAVIOTAS	4.55	-70.92
803000	GUAPI	2.58	-77.90
698704	AFWA ASSIGNED	4.22	-74.63

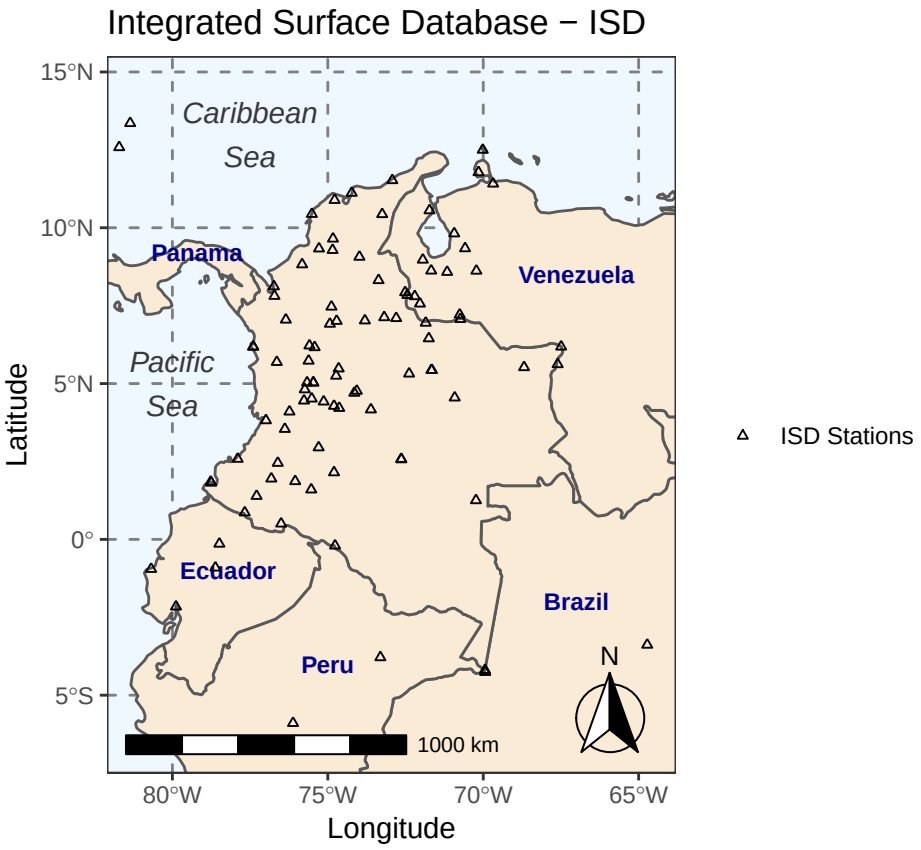

Figure 2.3: ISD Stations. Colombia and Surroundings countries

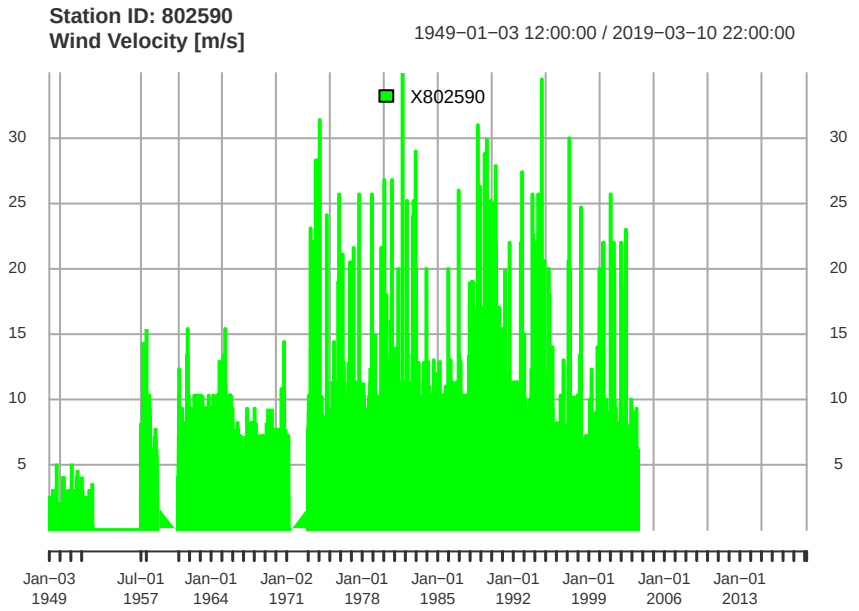

Figure 2.4: ISD Station ALFONSO BONILLA ARAGON INTL - Time Series

2.3 ERA5

ERA5 is forecast reanalysis data processed by the *European Centre for Medium-Range Weather Forecasts* ECMWF with wind speeds time series in square cells of 0.25 decimal degrees covering the whole planet. It was extracted a raster of 69 rows by 49 columns in format NetCDF. Cell centers represent ERA5 stations, with IDs from 1 (lon=-79, lat=12.5) to 3381 (lon=-67, lat=-4.5). Map in figure 2.5 shows ERA5 stations.

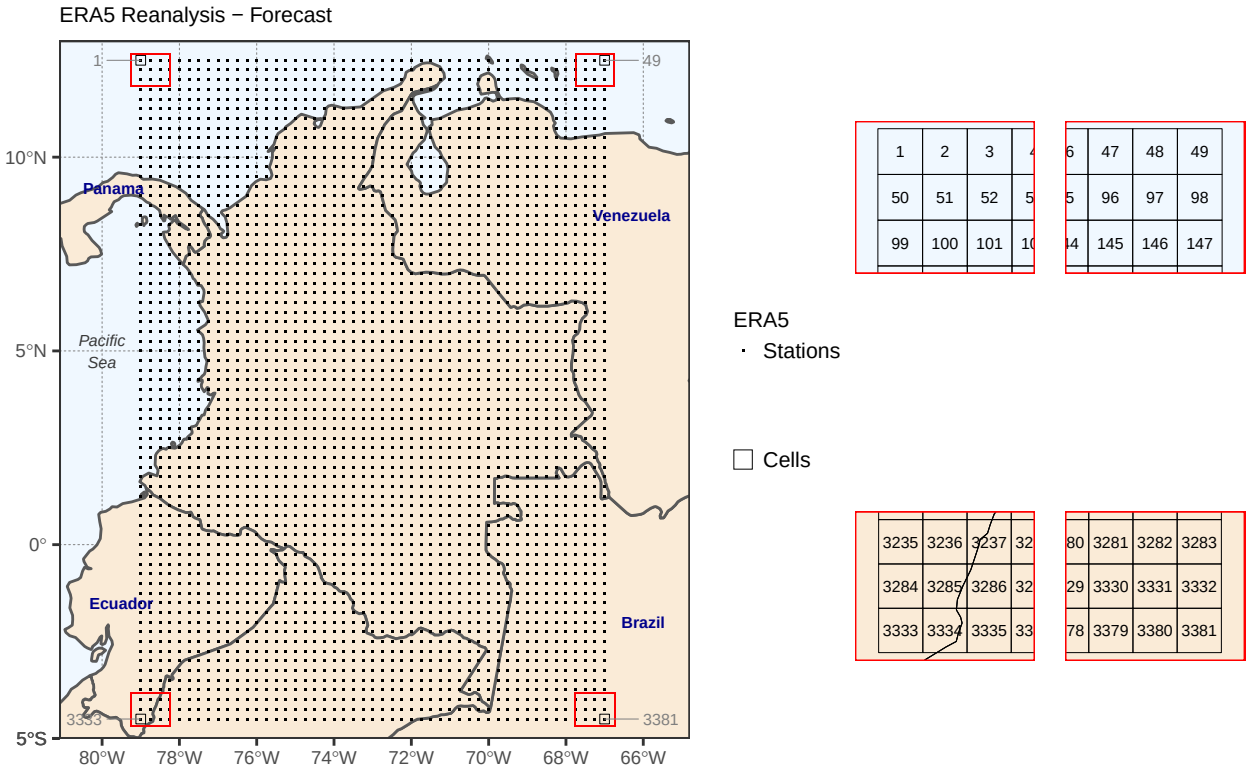

Figure 2.5: ERA5 Cells and Stations (Cells Centers)

2.4 Data Download and Data Organization

All data sources had different mechanisms for downloading. For IDEAM, the official procedure is through the e-mail *atencionalciudadano@ideam.gov.co*. For ISD all files are available in the FTP site `ftp://ftp.ncdc.noaa.gov/pub/data/noaa/isd-lite/`, organized in folders by years with *gzip* files inside; there are many files by station (one file for each year available), with names in the format *ID-99999-YYYY.gz*, where ID is the USAF-ISD station identifier, and YYYY is the year. ERA5 data request uses a Python scripts with data size limit for download. Files with all IDEAM and ISD stations are available in Annex A. For the Python code and commands to join NetCDF files of ERA5 data source see the Annex C. For data organization see the Annex D.

Chapter 3

Theoretical Framework

3.1 Probability Concepts

3.1.1 Probability Density Function PDF

PDF defines the probability that a continuous variable falls between two points. In PDF the probability is related to the area below the curve (integral) between two points, as for continuous probability distributions the probability at a single point is zero. The term density is related to the quantity of probability defined below each part of the curve, the higher the values of the curve, the higher the density and, consequently, the probability.

$$\int_{\mathrm{a}}^{\mathrm{b}} \mathrm{f(x)dx} = \mathrm{Pr[a \leq X \leq b]} \tag{3.1}$$

Equation (3.2) is the Gumbel PDF.

$$\mathrm{f(x)} = \frac{1}{\beta} \exp\left\{-\frac{\mathrm{x}-\mu}{\beta}\right\} \exp\left\{-\exp\left\{-\left(\frac{\mathrm{x}-\mu}{\beta}\right)\right\}\right\}, \quad -\infty < \mathrm{x} < \infty \tag{3.2}$$

where $\exp\{.\}$ is $\mathrm{e}^{\{.\}}$, β the scale parameter, and μ the location parameter. Location (μ) has the effect to shift the PDF to left or right along 'x' axis, thus, if location value is changed the effect is a movement to the left (small value for location), or to the right (big value for location). Scale has the effect to stretch ($\beta > 1$) of compress ($0 < \beta < 1$) the PDF, if scale parameter is close to zero it approaches a spike.

Figure 3.1 shows PDF with location (μ) = 100 and scale (β) = 40, using Equation (3.2). Figure 3.2 shows PDF with location (μ) = 100 and scale (β) = 40, using function `dgumbel` of the package `RcmdrMisc`.

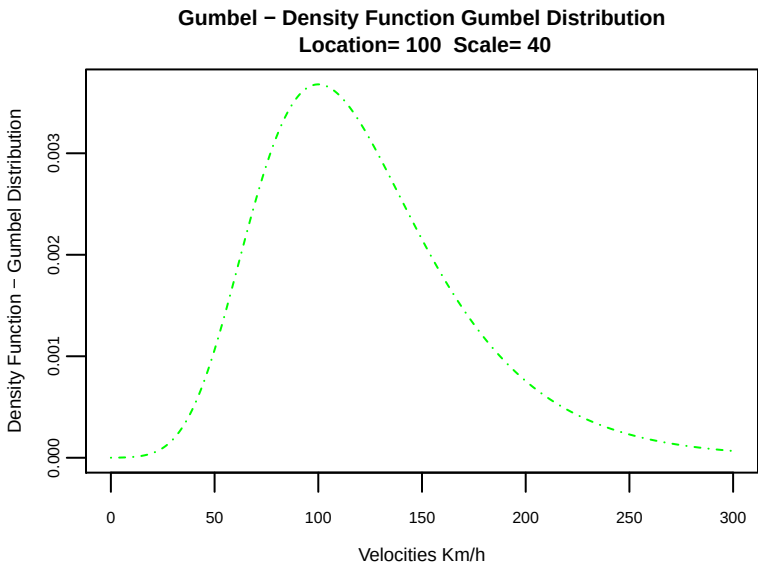

Figure 3.1: Gumbel PDF

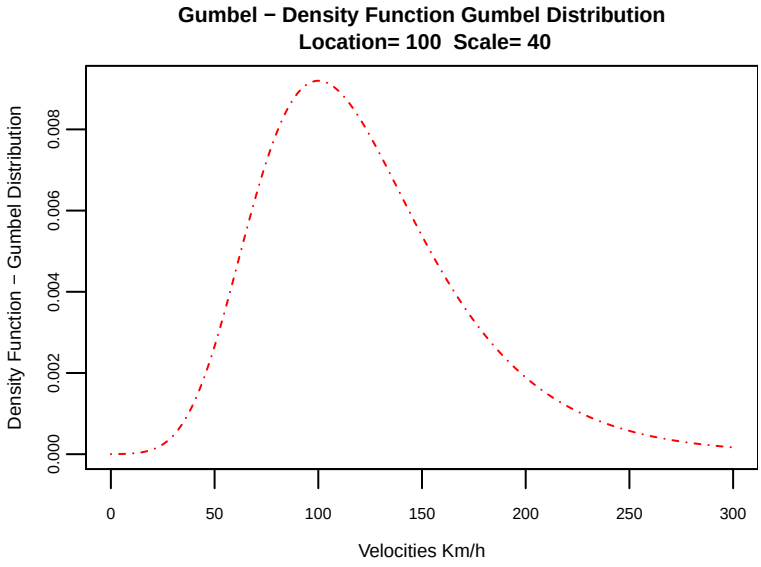

Figure 3.2: Gumbel PDF - dgumbel function

3.1.2 Cumulative Distribution Function CDF

CDF is the probability of taking a value less than or equal to x. That is

$$\mathrm{F(x)} = \mathrm{Pr[X < x]} = \alpha \tag{3.3}$$

For a continuous variable, CDF can be expressed as the integral of its PDF.

$$\mathrm{F(x)} = \int_{-\infty}^{\mathrm{x}} \mathrm{f(x)dx} \tag{3.4}$$

Equation (3.5) is the Gumbel CDF.

$$F(x) = \exp\left\{-\exp\left[-\left(\frac{x-\mu}{\beta}\right)\right]\right\}, \quad -\infty < x < \infty \tag{3.5}$$

Figure 3.3 shows Gumbel CDF with location (μ) = 100 and scale (β) = 40, using Equation (3.5). As previously done with PDF, similar result can be achieved using function `pgumbel` of package `RcmdrMisc`.

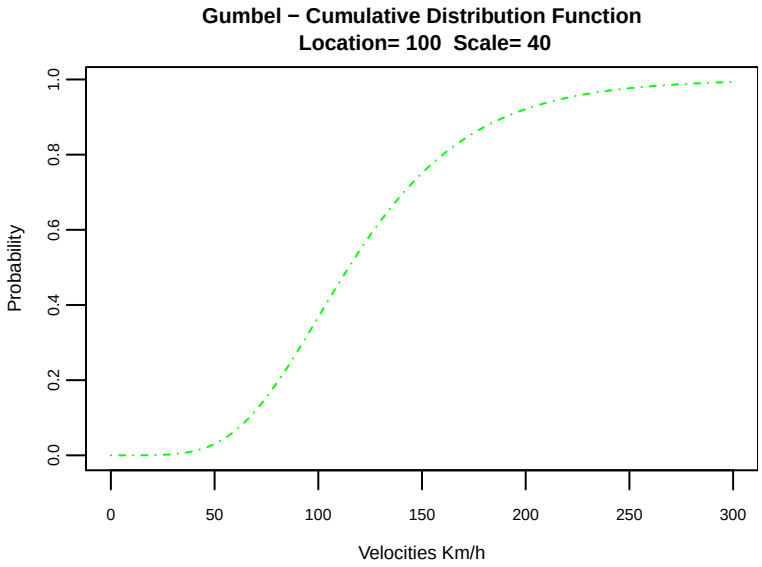

Figure 3.3: Gumbel CDF

3.1.3 Percent Point Function PPF

PPF is the inverse of CDF, also called the *quantile* function. This is, from a specific probability get the corresponding value x of the variable.

$$x = G(\alpha) = G(F(x)) \tag{3.6}$$

Equation (3.7) is the Gumbel PPF.

$$G(\alpha) = \mu - \beta\ln(-\ln(\alpha)) \quad 0 < \alpha < 1 \tag{3.7}$$

Figure 3.4 shows Gumbel PPF, using Equation (3.7). Similar result can be achieved using function `qgumbel` of package `RcmdrMisc`.

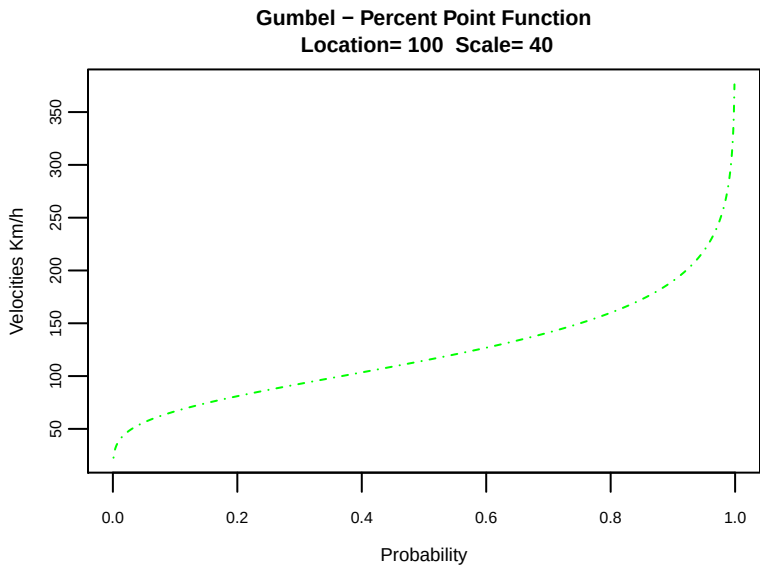

Figure 3.4: Gumbel PPF

3.1.4 Hazard Function HF

Figure 3.5 shows Gumbel HF, using Equation (3.9).

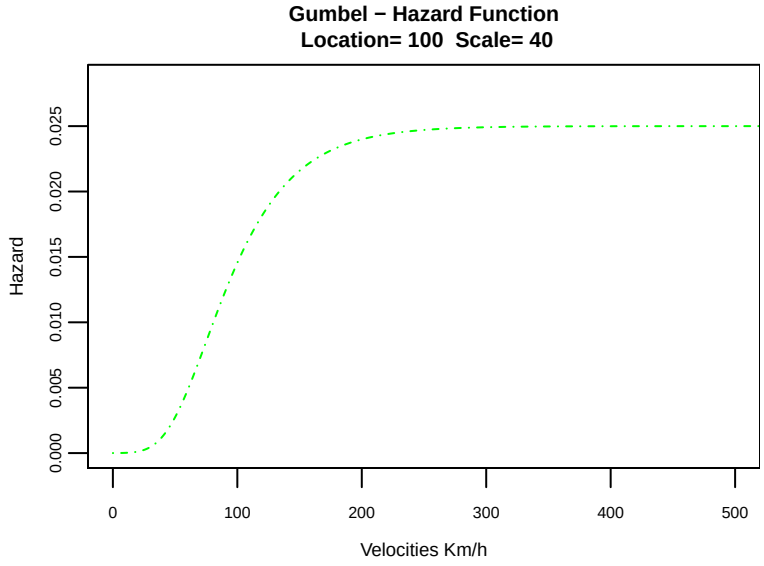

Figure 3.5: Gumbel HF

HF is the ratio between PDF and SF. SF is the survival function $S(x) = 1 - F(x)$, which defines the probability that a variable takes a value greater than x, $S(x) = Pr[X > x] = 1 - F(x)$.

$$\mathrm{h(x)} = \frac{\mathrm{f(x)}}{\mathrm{S(x)}} = \frac{\mathrm{f(x)}}{1 - \mathrm{F(x)}} \tag{3.8}$$

Equation (3.9) is the Gumbel HF.

$$\mathrm{h(x)} = \frac{1}{\beta}\frac{\exp(-(\mathrm{x}-\mu)/\beta)}{\exp(\exp(-(\mathrm{x}-\mu)/\beta))-1} \tag{3.9}$$

3.2 Statistical Concepts for Extreme Analysis

In order to approach the extreme value analysis, some statistical concepts are needed to understand the theoretical framework behind this knowledge area. This section introduces the concepts annual exceedance probability, mean recurrence interval MRI, exposure time, and compound probability for any given exposure time and MRI. As a hypothetical example, a simulated database of extreme wind speed will be used. This database is supposed to have 10.000 years of wind speeds.

3.2.1 Annual Exceedance Probability P_e

Using the previously described database, a question arises to calculate the probability to exceed the highest probable damage caused to any structure by the action of the winds from this simulated database. It is possible to conclude that there is only one event greater or equal (in this case equal) to the highest probable causing damage in 10.000 years, and it is the *highest wind*. If the database is sorted by wind magnitude in descending order, i.e. small winds to the right of the list as shown in Figure 3.6, the question is solved calculating the annual exceedance probability *Pe* with Equation (3.10).

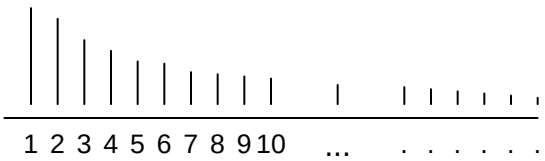

Figure 3.6: Sorted Wind Velocities by Magnitude

The annual exceedance probability P_e equals to the ratio between *event index after descending sorting* and *years of simulations*. The highest wind will be the first in the sorted list.

$$\mathrm{P_e} = \frac{\text{Event index after descending sorting}}{\text{Years of simulations}} = \frac{1}{10.000} = 0.001 = 0.01\% \tag{3.10}$$

Same exercise can be done with all winds to construct the annual exceedance probability curve, that in this case will represent the probability to equal or exceed different probable damage due to wind.

3.2.2 Mean Recurrence Interval MRI

Continuing with the previous section, if the inverse of the exceedance probability is taken, the return period (in years) is obtained. The return period or Mean Recurrence Interval MRI is associated with a specific return level (wind extreme velocity). MRI is the numbers of years (N) needed to obtain 63% of change that the corresponding return level will occur at least one time in that period. The return level is expected to be exceeded on average once every N-years.

The formula to calculate the annual exceedance probability of the return level depends on the MRI value as shown below:

$$\mathrm{P_e} = \begin{cases} 1 - \exp\left(-\dfrac{1}{MRI}\right), & \text{for MRI} < 10 \text{ years} \\ \dfrac{1}{MRI}, & \text{for MRI} \geq 10\text{years} \end{cases} \tag{3.11}$$

For a specific wind extreme event A, the probability that the event will occur in a period equal to MRI years is 63%. If we analyze for the same period a strongest wind extreme event B, its occurrence probability will be less than 67%. If the purpose of this research is to design infrastructure considering wind loads, the structure will be more resistant to wind if we design with stronger winds, this is high MRIs, and low annual exceedance probability. Common approach for infrastructure design, considering any type of load (earthquake, wind, etc.) is to choose high MRI according to the importance/use/risk/type of the structure. For highly important structures, like hospitals or coliseums, where the risk of collapse must be diminished, the MRI used to design is higher in comparison to common structures (for instance a normal house), which implies less risks for its use and importance.

3.2.3 Compound Exceedance Probability Pn

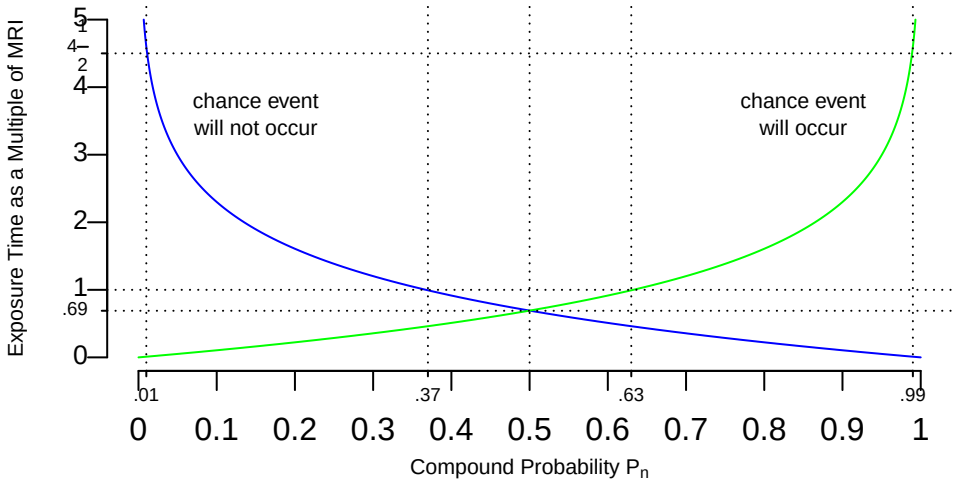

Figure 3.7: Compound Probability

It is possible to calculate a compound probability P_n, where n is the exposure period. The exposure period is the usage time of the structures that have been designed with an extreme

wind speed. P_n is the probability that the extreme wind speed will be equaled or exceeded at least one time in n years, and in this sense it is a probability of occurrence:

$$\mathrm{P_n} = 1 - \left(1 - \frac{1}{\mathrm{MRI}}\right)^{\mathrm{n}}, \text{ occurrence probability} \tag{3.12}$$

As a complementary probability, it is possible to calculate the compound non-occurrence probability as $\left(1 - \frac{1}{MRI}\right)^n$

If it is considering exposure time as a multiple of return period, the resulting Figure 3.7, shows that:

- When exposure time is .69% of the return period, then probability (occurrence and non-occurrence) will be 50%
- As was stated previously, when exposure time is equal to return period, then the probability that the extreme wind speed (return level) occur is 63%, and 37% for the non-occurrence probability.
- If exposure time is 4.5 times the return period, there is a 99% of change that the return level will occur.

The example discussed here was presented as an instrument to introduce important concepts, nonetheless, there are specialized approaches to deal with extreme value analysis which will be discussed in *Extreme Value Analysis Overview*.

3.3 Extreme Value Analysis Overview

Analysis of extreme values is related with statistical inference to calculate probabilities of extreme events. Main methods to analyze extreme data are *sample maxima*, and *Peaks Over Threshold POT*. The sample maxima method also known as epochal or block maxima, is the classical approach and uses the most extreme value for a specific frame of time, typically one year. POT is based in the selection of a single threshold value to do the analysis only with values above the threshold. There are different POT approaches depending on how the time and magnitude dimensions are analyzed, the most common one uses generalized Pareto for wind velocities and Poisson process for time (POT-Poisson-GPD), and the most flexible one uses Poisson process simultaneously for both dimensions (POT-PP).

In both methods (Epochal and POT), the main step is to fit wind velocities to an appropriate probability distribution model. Epochal uses a generalized extreme value distribution GEV, a family of distributions which include extreme value type I - Gumbel, extreme value type II - Fréchet, and Weibull. Gumbel is the most common used GEV. POT uses a generalized Pareto distribution (POT-Poisson-GPD), or an intensity function (POT-PP).

Distribution models are fitted based in the estimation of its parameters, commonly called location, scale, and shape, nonetheless each model has its own parameters names. There are different methods to estimate parameters, among them: (a) method of modified moments (Kubler, 1994), and L moments (Hosking & Wallis, 1997), (b) method of maximum likelihood MLE (Harris & Stocker, 1998), (c) probability plot correlation coefficient, and (d) elemental percentiles (for GPD and GEV).

Once candidate parameters are available, it is necessary to assess the goodness of fit of the selected model using tests like Kolmogorov-Smirnov (KS), or Anderson-Darling. Here a visual assessment is also useful using a probability plot or a kernel density plot with the fitted PDF overlaid.

The main use of the fitted model is the estimation of mean return intervals MRI, and extreme wind speeds (return levels),

$$\mathrm{MRI} = \frac{1}{1 - \mathrm{F(y)}} \tag{3.13}$$

with $F(y)$ as the CDF. If $1 - F(y)$ is the annual exceedance probability, MRI is its inverse; see (Simiu & Scanlan, 1996) for more details about MRI. If y is solved from Equation (3.13) using a given MRI of N-years, its value represents the Y_N wind speed return level. Refer to each specific method bellow for specific solutions to Y_N.

The CRAN Task View "Extreme Value Analysis" `https://cran.r-project.org/web/views/ExtremeValue.html` shows available **R** for block maxima, POT by GPD, and external indexes estimation approaches. Most important to consider are `evd`, `extremes`, `evir`, `POT`, `extremeStat`, `ismev`, and `Renext`.

3.3.1 Epochal (Sample Maxima)

To work with random variables of sample maximum values, used probability distribution function PDF is GEV:

$$\mathrm{H(y)} = \exp\left\{ -\left[1 + \xi \frac{\mathrm{x} - \mu}{\psi} \right]_+^{-\frac{1}{\xi}} \right\} \tag{3.14}$$

Where $\xi \neq 0$, $[...]_+ = max([...], 0)$, μ is the location parameter, $\psi > 0$ is a scale parameter, and ξ is a shape parameter. GEV combines in one unique family the Gumbel (medium-tailed) distribution (limit $\xi \to 0$), Fréchet (long-tailed) distribution ($\xi > 0$), and Weibull (short-tailed) distribution ($\xi < 0$).

3.3.2 Peaks Over Threshold using GPD and 1D Poisson Process POT-Poisson-GPD

In this model, (a) the magnitude of the observations above the threshold are assumed to be independent random variables with the same generalized Pareto as probability distribution, σ as scale, and ξ as tail length, and (b) corresponding times are assumed to follow a one dimensional homogeneous Poisson process with γ as parameter.

With the condition of exceeding some high threshold b, as a consequence $y = (x - b) > 0$, the CDF of y is the generalized Pareto distribution GPD:

$$\mathrm{F(y)} = 1 - \left[1 - \xi \frac{\mathrm{y}}{\sigma} \right]_+^{-\frac{1}{\xi}}, \tag{3.15}$$

where $[...]_+ = max([...], 0)$, b is the threshold. In both GPD (magnitude), and $1D$ Poisson process (time), it is not possible to differentiate between thunderstorm and non-thunderstorm

wind types. Pickands (1971) found a rigorous connection between epochal (GEV) and limits results of POT-Poisson-GPD, as parameter shape ξ in Equations (3.14) and (3.15) are exactly the same. The long-tailed case when $\xi > 0$, GPD behaves as usual Pareto distribution, for $\xi = 0$ (taking the limit $\xi \to 0$) it behaves as exponential distribution, and $\xi < 0$ the distribution has a finite upper endpoint at $-\frac{\sigma}{\xi}$.

The equation to calculate return levels (RL) Y_N corresponding to the N-years return period in POT-Poisson-GPD is:

$$\mathrm{Y_N} = \mathrm{G}\left(\mathrm{y}, 1 - \frac{1}{\lambda\,\mathrm{N}}\right) \tag{3.16}$$

where G is the quantile function (PPF), and the value of the probability passed to the G function, has to be modified with the λ parameter. λ is the number of wind speed events over the threshold per year.

3.3.3 Peaks Over Threshold Using a 2D Poisson Process POT-PP

According to (Pintar et al., 2015) the stochastic Poisson Process PP is mainly defined by its intensity function. As the intensity function is not uniform over the domain, the PP considered here is non-homogeneous, and due to the intensity function dependency of magnitude and time, it is also bi-dimensional. PP was described for the first time in (Pickands, 1971), then extended in (Smith, 1989).

Generic Equation (3.17) shows the intensity function, which is defined in the domain $D = D_t \cup D_{nt}$, and allow to fit the PP at each station to the observed data $\{t_i, y_i\}_{i=1}^{I}$, for all the times (t_i) of threshold crossing observations, and its corresponding wind speeds magnitudes (y_i). Thus, only data above the threshold (POT) are used.

$$\lambda\,(\mathrm{y}, \mathrm{t}) \begin{cases} \lambda_t(y), & \text{for t in thunderstorm period} \\ \lambda_{nt}(y), & \text{for t in non-thunderstorm period} \end{cases} \tag{3.17}$$

The specific intensity function of the PP is defined in (Smith, 2004) and is shown in Equation (3.18):

$$\frac{1}{\psi_\mathrm{t}}\left(1 + \zeta_\mathrm{t}\frac{\mathrm{y} - \omega_\mathrm{t}}{\psi_\mathrm{t}}\right)_+^{-\frac{1}{\zeta_\mathrm{t}}-1} \tag{3.18}$$

where, at a given time t, parameter *shape* $= \zeta_t$ controls the tail length of the intensity function, and the other two parameters ω_t and ψ_t define the location and scale of the intensity function.

Figure 3.8 represent the domain D of PP. In time, the domain represents the station service period from first sample t_1 to last sample t_4. D is the union of all thunderstorm periods D_t (from t_2 to t_3), and all non-thunderstorm periods D_{nt} (periods t_1 to t_2 and t_3 to t_4). In magnitude, only thunderstorm data above its threshold b_t, and only non-thunderstorm data above its threshold b_{nt} are used.

Thunderstorms and non-thunderstorms are modeled independently:

1. Observations in domain D follow a Poisson distribution with mean $\int_D \lambda(t, y)\, dt\, dy$
2. For each disjoint sub-domain D_1 or D_2 inside D, the observations in D_1 or D_2 are independent random variables.

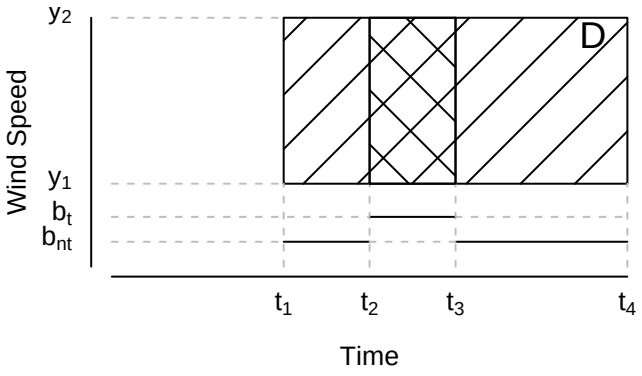

Figure 3.8: Domain off Poisson Process - PP

Visual representation of the intensity function for PP can be seen in Figure 3.9. In vertical axis, two surfaces were drawn representing independent intensity functions for thunderstorm $\lambda_t(y)$ and for non-thunderstorm $\lambda_{nt}(y)$. The volume under each surface for its corresponding time periods and peak over threshold velocities, is the mean of PP.

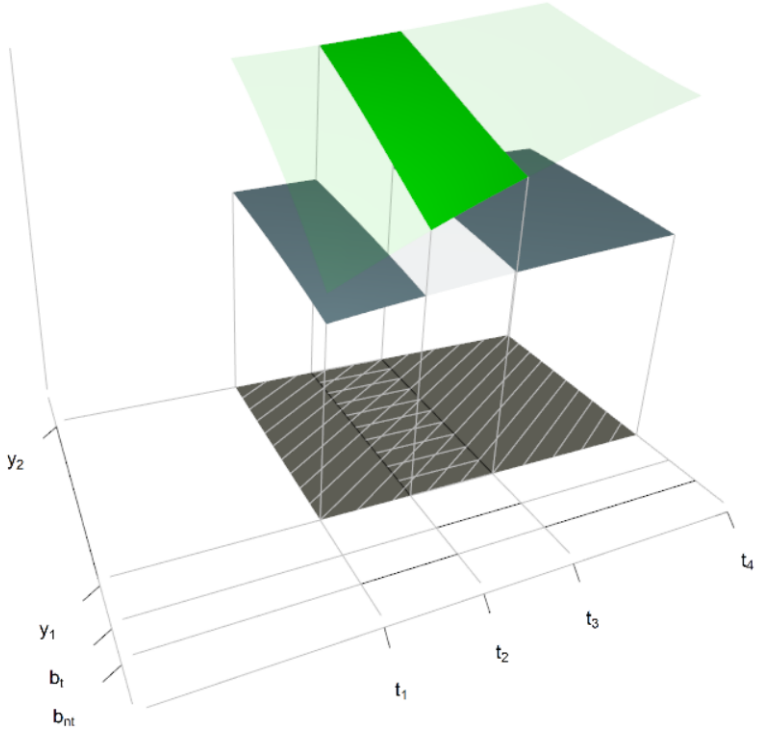

Figure 3.9: Volume Under Surfaces: Mean of PP

To fit the intensity function to the data, the method of maximum likelihood is used to

estimate its parameters, *scale* $= \psi$, *location* $= \omega$, and *shape* $= \zeta$, the selected vector of parameters η are the $\hat{\eta} = (\hat{\psi}, \hat{\omega}, \hat{\zeta})$ values that maximizes the function:

$$\mathrm{L}(\eta) = \left(\prod_{\mathrm{i}=1}^{\mathrm{I}} \lambda\left(\mathrm{y_i}, \mathrm{t_i}\right) \right) \exp\left\{ - \int_{\mathrm{D}} \lambda\left(\mathrm{y}, \mathrm{t}\right) \mathrm{dy}\, \mathrm{dt} \right\}. \tag{3.19}$$

The values of $\hat{\eta}$ need to be calculated using a numerical approach, because there is not analytical solution available.

Once the PP is fitted to the data, the model will provide extreme wind velocities (return levels), for different return periods (mean recurrence intervals).

A Y_N extreme wind velocity, called the return level (RL) belonging to the N-years return period, has an expected frequency to occur or to be exceeded (annual exceedance probability) $P_e = \frac{1}{N}$, and also has a probability that the event does not occur (annual non-exceedance probability) $P_{ne} = 1 - \frac{1}{N}$. Y_N will be the resulting value of the G (PPF or quantile) function using a probability equal to P_{ne}. $Y_N = quantile(y, p = P_{ne}) = G(y, p = P_{ne}) = PPF(y, p = P_{ne})$. Y_N can be understood as the wind extreme value expected to be exceeded on average once every N years.

For PP Y_N is defined in terms of the intensity function as the solution to the equation

$$\int_{\mathrm{Y_N}}^{\infty} \int_{0}^{1} \lambda\left(\mathrm{y}, \mathrm{t}\right) \mathrm{dydt} = \mathrm{A_t} \int_{\mathrm{Y_N}}^{\infty} \lambda_{\mathrm{t}}\left(\mathrm{y}\right)\, \mathrm{dy} + \mathrm{A_{nt}} \int_{\mathrm{Y_N}}^{\infty} \lambda_{\mathrm{nt}}\left(\mathrm{y}\right)\, \mathrm{dy} = \frac{1}{\mathrm{N}} \tag{3.20}$$

where A_t, is the multiplication of the average number of thunderstorm per year and the average length of a thunderstorm, taken to be 1 hour as defined in (Pintar et al., 2015), and $A_{nt} = 365 - A_t$. The average length of a non-thunderstorm event is variable, and it is adjusted for each station to guarantee that $A_{nt} + A_t = 365$. Value 365 is used only, if operations with time in the dataset are performed in days.

The same thunderstorm event in considered to occur if the time lag distance between successive thunderstorm samples is small than six hours, and for non-thunderstorm this time is 4 days. For PP, all the measurements belonging to the same event (thunderstorm or non-thunderstorm), need to be de-clustered to leave only one maximum value. In other words, the number of thunderstorm in the time series is one plus the number of time lag distances greater than 6 hours, and above 4 days for non-thunderstorm.

3.4 Wind Loads Requirements

As the output maps of this research will be used as input loads for infrastructure design, the methodology used for its creation, need to be consistent with Colombian official wind loads requirements. Colombian structure design code, from now the *design standard*, was created and is maintained by the Colombian Association of Seismic Engineering - AIS.

The design standard is mainly based in *minimum design loads and associated criteria for buildings and other structures ASCE7-16* standard (ASCE, 2017). The ASCE7-16 standard defines the minimum requirements for design wind loads in pages 733 to 747. Wind speeds

requirements of ASCE7-16 are based in the combination of independent non-hurricane analysis, and hurricane wind speeds simulations models. The focus of this research will be the analysis of non-hurricane wind data, however, existing results of hurricane studies will be used to present final maps with both components. In ASCE7-16, for non-hurricane wind speed, the procedure is mainly based on (Pintar et al., 2015).

ASCE7-16 (page 734), requires the calculation of wind extreme return levels for specific return periods according to the risk category of the structure to be designed, as follows: risk category I - 300 years, risk category II - 700 years, risk category III - 1700 years, risk category IV - 3000 years. In addition, extreme wind speeds for those MRI need to correspond to 3 seconds gust speeds at 33 ft. (10 meters) above the ground and exposure category C. Below is a description of the risk categories:

- Risk IV - This are 'indispensable buildings' that involve substantial risk. These structures that can handle toxic or explosive substances.
- Risk III - There is substantial risk because these structures that can handle toxic or explosive substances, can cause a serious economic impact, or massive interruption of activities if they fail.
- Risk II - Category 'by default', and correspond to structures not classified in others categories.
- Risk I - This structures represent low risk for life of people.

To standardize wind speeds to gust speeds ASCE7-16 proposes the curve Durst, see (C. S. Durst, 1960), and Figure 3.10. Durst curve is only valid for open terrain conditions, and it shows in axis y the gust factor $\frac{V_t}{V_{3600}}$, a ratio between any wind gust (maximum speeds) averaged at t seconds, V_t, and the hourly averaged wind speed V_{3600}, and in the axis x the duration t of the gust in seconds. Be aware that curve values in Figure 3.10 are approximated values taken visually from the original curve.

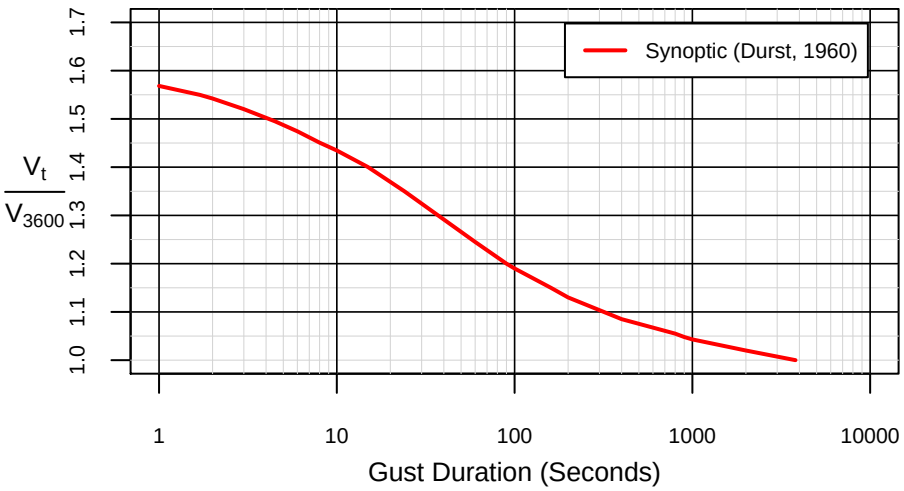

Figure 3.10: Durst Curve

Chapter 4

Methodology

This research is focus in non-hurricane data, with three main elements: *data*, *temporal analysis* with POT-PP, and *spatial analysis* to do spatial interpolation and create return levels RL maps for MRIs of 700, 1700, and 3000 years. Core steps (1, and 3 to 7) need to be done in an iterative process station by station as is shown in Figure 4.1.

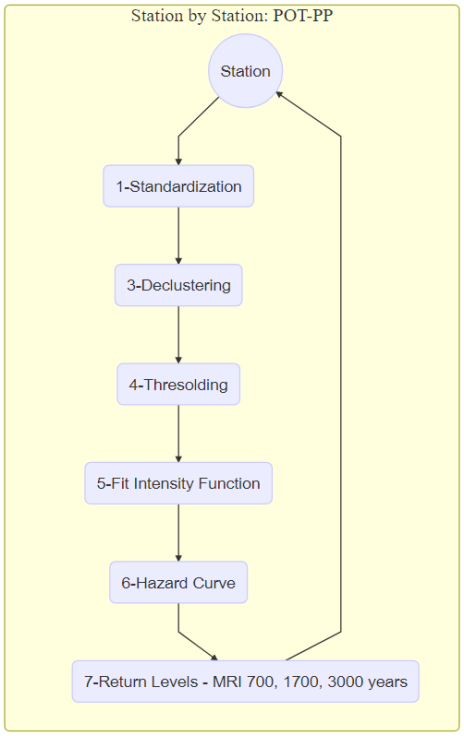

Figure 4.1: Iterative Process in Methodology

Figure 4.2 shows the methodological scheme where the main elements mentioned are highlighted using shaded boxes. Steps 1 to 8 are the most representative, but step 2 is a data verification process that can be done once (in bulk) for all stations.

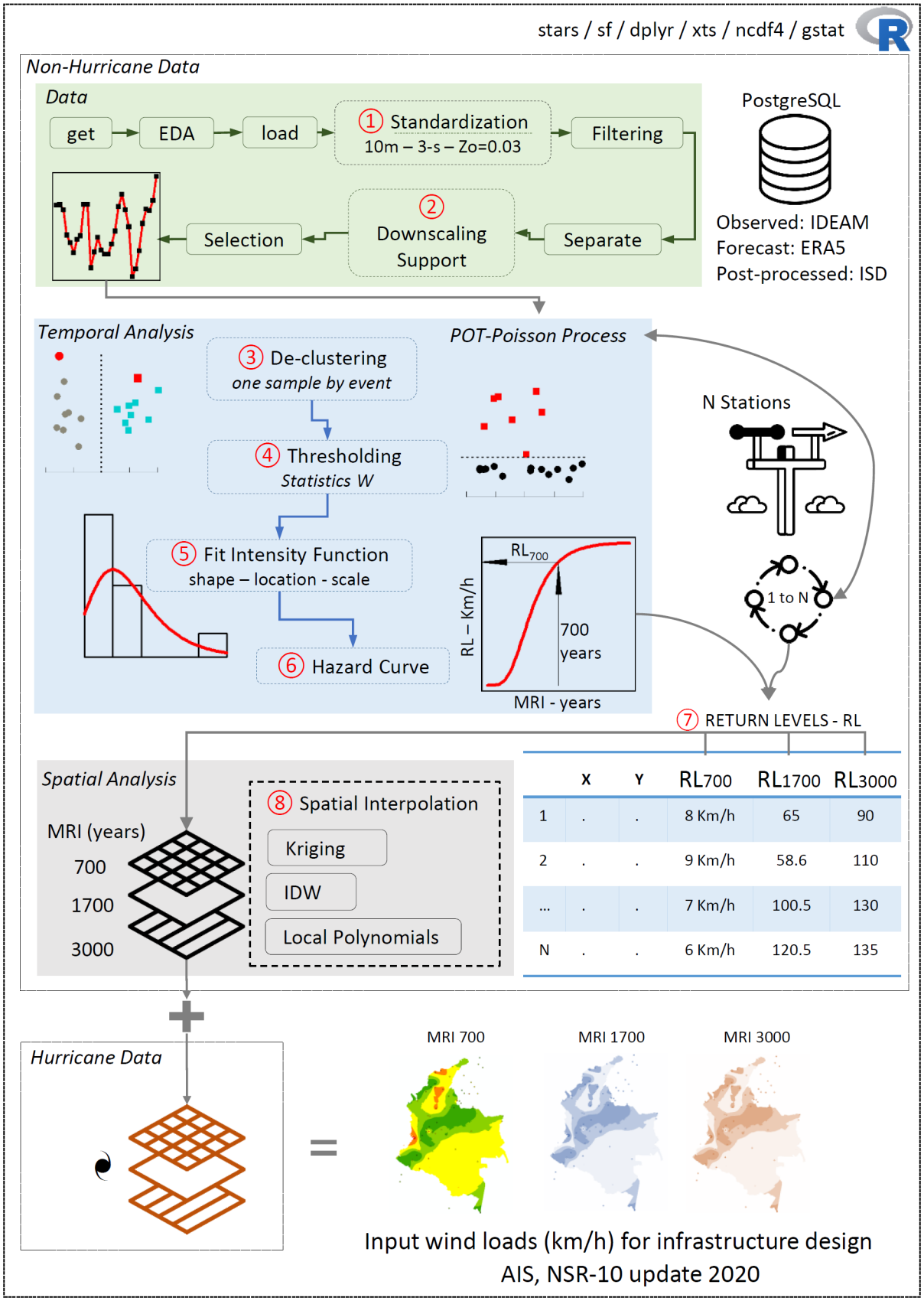

Figure 4.2: Methodology

Once the iterative cycle ends and the RL are calculated in all the stations, continuous surfaces will be created, one for 700 years, next for 1700 years, and finally for 3000 years. An additional element, is the integration with existing hurricane studies to produce final maps, that will be used as input loads for infrastructure design, and will be part of the design standard

4.1 Data Standardization

Analysis of extreme wind speeds requires data standardization as initial step. All input data must be standardized to represent three important conditions: a) anemometer height of 10 meters, b) open space terrain roughness (exposition C), and c) averaging time of 3-seconds wind gust. ASCE (2017) defines exposition C as areas with few obstructions, and exposition D refers to perfect open space.

Parallel to the standardization activity described below, it is also important to consider for all stations involved in the analysis:

- *Separating*: As far as possible, identify each record of the time series, as thunderstorm (t) or non-thunderstorm (nt)
- *Filtering*: Remove wind speeds above $200\frac{km}{h}$ and data pertaining to hurricane events, because the procedure with hurricane requires a different approach and need to be done independently

4.1.1 Anemometer Height (10 m)

According to the protocol for field data collection and location of methodological stations (IDEAM, 2005), the anemometer (wind sensor) in installed always to a fixed height of 10 meters from the surface, as is shown in Figure 4.3; therefore, no height correction is needed.

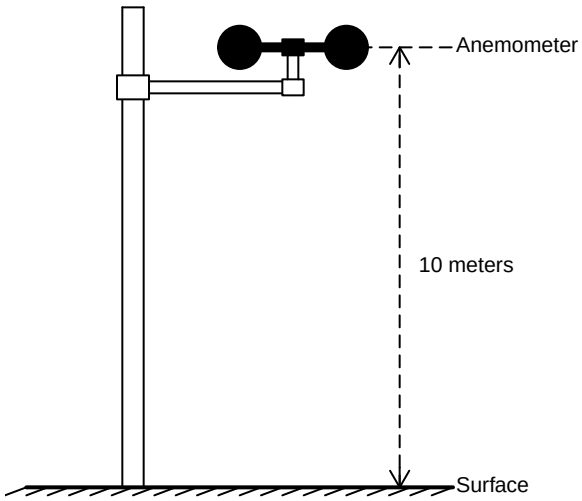

Figure 4.3: Anemometer height: 10 meters

4.1.2 Surface Roughness at Open Terrain

Due to the effects that the terrain has on wind speed, a correction should be applied if the station is located in a geographical space considered "not open terrain". When terrain is open, the roughness corresponds to 0.03 meters. There are some alternative methodologies to calculate the roughness; for example, (Masters, Vickery, Bacon, & Rappaport, 2010) uses the station data, but the separation of the measurements should not exceed one minute (something difficult to obtain), (Lettau, 1969) uses the empirical Equation (4.1) (recommended in ASCE7-16 page 743, equation C26.7-1) to calculate roughness z_o, which was used here,

$$\mathrm{z_0 = 0.5\ H_{ob}\ \frac{S_{ob}}{A_{ob}}} \tag{4.1}$$

where H_{ob} is the average height of the obstacles, S_{ob} is the average vertical area perpendicular to the wind of the obstacles, and A_{ob} is the average area of the terrain occupied by each obstruction. The empirical exponent α, gradient height z_g, and exposure coefficient K_z, are used to calculate the correction factor $F_{exposition}$, for z_0 units are in meters.

$$\mathrm{\alpha = 5.65 * z_0^{-0.133}} \tag{4.2}$$

$$\mathrm{z_g = 450 * z_0^{0.125}} \tag{4.3}$$

$$\mathrm{K_z = 2.01 * \left(\frac{z}{z_g}\right)} \tag{4.4}$$

$$\mathrm{F_{exposition} = \frac{0.951434}{K_z}} \tag{4.5}$$

According to (NIST, 2012), calculation of roughness needs to be weighted according to the predominance of wind magnitude in eight directions (north, south, east, west, north-east, north-west, south-east, and south-west) around the station location. The calculation in each direction can be done using a detailed aerial photo or satellite image of the station, including a radius of 800 meters. Figure 4.4 shows the wind percentages in mentioned directions for a generic station. Figure 4.5 shows the satellite image for *Vanguardia* ISD station (USAF:802340), located in *Villavicencio* airport, with four (south, north, east, and west) 45 degree sectors highlighted

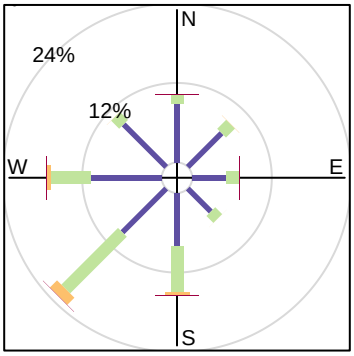

Figure 4.4: Wind Rose with Wind Percentages

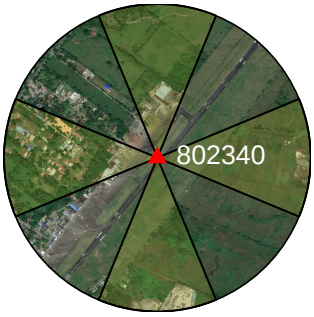

Figure 4.5: Digital Imagery for 'Vanguardia' ISD Station (USAF:802340)

Figure 4.6 shows extreme conditions for roughness, open space in left image (ISD Station 804070) with roughness value of 0.03, closed space in center image (ISD Station 803000) with roughness value of 0.1, and a typical example where mentioned Lettau procedure is needed because roughness is different in each direction, in right image. Lettau Equation (4.1) need to be applied to each direction and then the final z_o value is the weighted average, using historical wind percentage. See Figure 4.7 showing the strokes made to calculate the different areas for two Colombian stations, in red the area occupied by the obstacles, and in blue the perpendicular area (Triana, 2019). Information about wind percentage per direction at each station were obtained from (IDEAM, 1999).

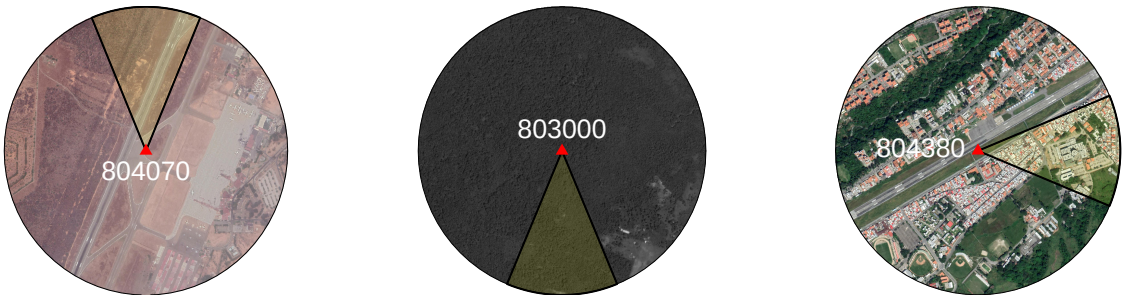

Figure 4.6: Roughness. Open (L), Closed (C), and Lettau (R).

Figure 4.7: Lettau Calculation

4.1.3 Averaging Time: 3-s Gust

To transform hourly mean wind velocity V_{3600}, to 3-s gust velocity V_3, ASCE7-16 recommends the use of (C. S. Durst, 1960). In curve Durst the axis x represents the duration t of the gust, what is done is to look there for the value 3 seconds, and read the corresponding gust factor $\frac{V_t}{V_{3600}}$ in axis Y. For instance, using variable V_{3600} from IDEAM data source, the gust factor for 3-s gust is 1.51.

$$\mathrm{V_t} = \mathrm{V_{3\,seconds}} = (\text{gust factor})\,\mathrm{V_{3600}} \tag{4.6}$$

It is valid only for open terrain conditions. Durst curve shows in axis y the gust factor $\frac{V_t}{V_{3600}}$, a ratio between any wind gust averaged at t seconds, V_t, and the hourly averaged wind speed V_{3600}, and in the axis x the duration t of the gust in seconds.

4.2 Downscaling Support

Where it is necessary to complement the local/regional wind analysis, with data from ISD (output data of a model for extreme winds), and ERA5 reanalysis dataset (large scale forecast data), it is required to probe by means of *comparisons* (exploratory data analysis and/or statistical measures) that those sources (modeled and forecast) are similar to IDEAM field measurements.

The proposed mechanism in the search for downscaling support is, (a) the creation of *common time series graphs*, where time series from all data sources are expected to be similar, and (b) the elaboration of *scatter plots graphics*, which are generated matching two sources in time (sorted in ascending order by wind velocity). By visual inspection is possible to evaluate data similarity between sources, when all the points fall very close to a 45-degree line. In both cases, the strategy for station matching, could be one of the following:

1. *Manual matching*, doing a detailed analysis station by station (only for ISD and IDEAM). While it is true that ISD is based on IDEAM, their names and locations are somewhat different, for this reason, it is necessary to read information available from each source, and decide station by station, about its correspondence.
2. *Intersection matching*, between ISD and IDEAM point stations and ERA5 cells. All ISD and IDEAM stations falling inside a ERA5 cell, will be compared between them.

4.3 Temporal Analysis (POT-PP)

Similar to how the adjustment of statistical data to a normal distribution is done to make inferences, in extreme value analysis only some part of the data (those that are extreme - over a high threshold - POT) needs to be fitted to a PP considering extreme deviations from the mean. While in the first case (normal distribution) the inferences are for events similar to the samples, in this case, when working with extreme value theory, the inferences will be for more extreme events than any previously observed or measured.

In summary, POT means only to work with extreme values, and PP means to adjust data to a PDF, which depends on an intensity function $\lambda(t, y)$, where t is time, y is wind extreme velocity. As shown in Figure 3.8, in a POT-PP approach with domain D, all the observations follow a Poisson distribution with mean $\int_D \lambda(t, y)\, dt\, dy$. Main advantage of POT-PP is that it is designed to consider storm and not-storm events independently (for each disjoint sub-domain D_1 or D_2 inside D, the observations in D_1 or D_2 are independent random variables), but in the end use them both for the inferences,

$$\mathrm{PDF} = \mathrm{f(t, y|\eta)} = \frac{\lambda(\mathrm{t, y})}{\int_{\mathrm{D}} \lambda(\mathrm{t, y})\, \mathrm{dt\, dy}} \tag{4.7}$$

4.3.1 De-clustering

To make the assumptions of PP more justifiable, it is important to have only one sample per event: the highest one. For instance, if a hypothetical storm started at 11:30 in the morning and ended at 12:30 in the afternoon, and the time series for that event has thirty wind measurements (one each two minutes), it is necessary to leave only the stronger or maximum value; this process is called de-clustering. In Figure 4.8, two thunderstorm clusters are shown, and only red samples are used to fit the PP. POT-PP defines that all the adjacent observations separated by six hours (6) or less in the case of thunderstorm events, and four (4) days or less, in the case of non-thunderstorm events, belong to the same cluster.

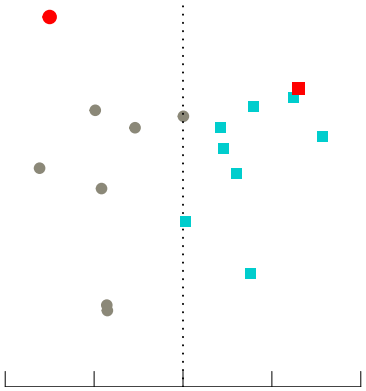

Figure 4.8: De-clustering in PP

4.3.2 Thresholding

As the POT model requires to work only with the most extreme values in the time series, it is necessary to select a threshold to filter out small values. Bias is high when a low threshold is selected (many exceedances) because the asymptotic support is weak; opposite situation happens for high thresholds where variance is potentially high. According to (Davison & Smith, 1990), it is necessary to select a threshold value, consistent with model structure. Figure 3.8 represents the thresholding process in a generic dataset, where only points above (red squares) the dotted horizontal line (the threshold) will be used for the model.

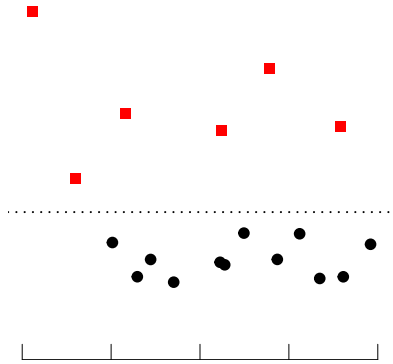

Figure 4.9: POT - Thresholding

The procedure to choose the best thresholds pairs, one for thunderstorm, and other for non-thunderstorm, is based in the W transformation. POT-PP needs selection of the best threshold pairs b_t and b_{nt} (see Figure 3.8) that produces the optimal fit. Measurement of this threshold fitting is done through W statistic. If wind variable y, in a POT-PP approach, has a $CDF = U = F(y)$, then $F(y)$ is distributed as uniform between 0 and 1 *uniform(0,1)*, meaning that the transformation $W = -log(1 - U)$ is an exponential random variable with mean one (1).

$$\mathrm{CDF} = \mathrm{U} = \mathrm{F(y)} = \mathrm{P(y \leq Y)} = \frac{\int_{\mathrm{b}}^{\mathrm{Y}} \lambda(\mathrm{y,t})\,\mathrm{dy}}{\int_{\mathrm{b}}^{\infty} \lambda(\mathrm{y,t})\,\mathrm{dy}} \tag{4.8}$$

W-statistic is done comparing the ordered result of applying $W = -log(1 - U)$ to the data (the axis y in Figure 4.10) with the theoretical quantiles of an exponential variable with uniform distribution between 0 and 1 (axis x in same figure). W-statistic is the highest vertical distance between the 45º line and the points in the graphic. The best thresholds pairs return the minimum value for W-statistics after testing, in an iterative process, with many threshold pairs combinations.

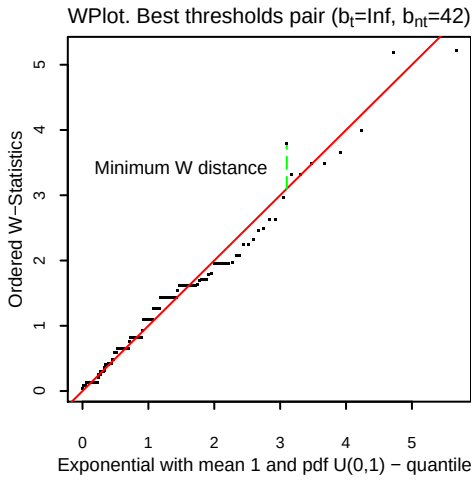

Figure 4.10: POT - Thresholding W Statistic

4.3.3 Exclude No-Data Periods

PP requires to remove long periods of time when stations were not recording or failing. Proposed time in (Pintar et al., 2015) is 180 days, namely, to remove all the gaps from the time series larger than six months.

4.3.4 Fit Intensity Function

Probability density function PDF, and cumulative distribution function CDF, of the PP, depend of the intensity function, and are shown in Equation (4.7), and Equation (4.8), respectively.

To facilitate the estimation of the parameters for the PP intensity function, parameter $shape = \zeta_t$ is taken to be zero in Equation (3.18), then doing the limit, the resulting intensity function is the same as the GEV type I or Gumbel distribution,

$$\frac{1}{\psi_{\mathrm{t}}} \exp\left\{\frac{-(\mathrm{y}-\omega_{\mathrm{t}})}{\psi_{\mathrm{t}}}\right\} \tag{4.9}$$

In this study, used intensity functions are:

$$\lambda\,(\mathrm{y},\mathrm{t}) = \begin{cases} \dfrac{1}{\psi_s} \exp\left(\dfrac{-(y-\omega_s)}{\psi_s}\right), & \text{for t in thunderstorm period} \\ \dfrac{1}{\psi_{nt}} \exp\left(\dfrac{-(y-\omega_{nt})}{\psi_{nt}}\right), & \text{for t in non-thunderstorm period} \end{cases} \tag{4.10}$$

As is shown in 4.11, the fitting process involve finding the best group of parameters of the intensity function, in such a way that the red curve (PDF of the PP, based in intensity function) be as tight as possible to the shape of the data histogram. As is described in *POT-PP*, optimal parameters to do the fitting process of the intensity function are calculated using *maximum likelihood*.

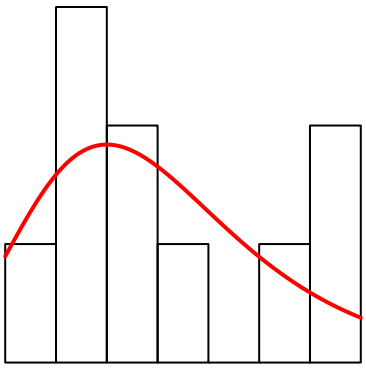

Figure 4.11: POT - PP Intensity Function Fitting Process

4.3.5 Hazard Curve and Return Levels RL

A hazard curve is shown in Figure 4.12, where axis x represents annual exceedance probability $P_e = \frac{1}{N}$, and axis y represents the return level RL Y_N for the corresponding N-years return period. It is possible to obtain the extreme return wind velocity level for any given return period going from axis x to axis y through the curve.

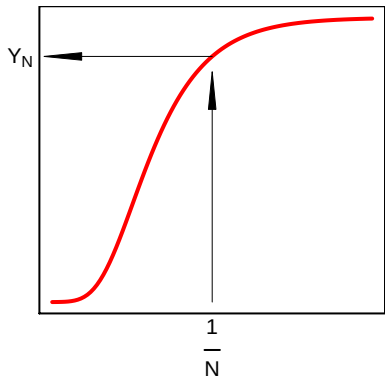

Figure 4.12: POT - PP Hazard Curve

In this research POT-PP includes only time series classified as non-thunderstorm, and this implies that the intensity function to be used (Equation (4.10)) does not differentiate between wind types (thunderstorm and non-thunderstorm), i.e., the intensity function is not a function of time t.

Hazard curve can be created solving Y_N in Equation (3.20) for a specific value of N in years (MRI). As a bad estimation caused by the deficiencies in the available information for the case study, the average duration time of non-thunderstorm events by year is considered to be one year, i.e., the parameter A_{nt} in Equation (3.20) is equal to one, and A_t is equal to zero (units in years).

Considering that the intensity function is not a function of time, equation (4.11) can be used replacing directly the parameters of the intensity function (PP) and the return periods (N), to create the hazard curve and get RL:

$$\mathrm{Y_N} = \frac{\psi}{\zeta}\left[-\log\left(\frac{\mathrm{N}-1}{\mathrm{N}}\right)\right]^{-\zeta} - \frac{\psi}{\zeta} + \omega \tag{4.11}$$

As for this research $\zeta = 0$ in selected intensity function (Equation (4.10)), return levels Y_N can be calculated with the Gumbel quantile function using $(1 - \frac{1}{N})$ as probability. This alternative approach is only valid when the analysis of POT-PP includes only one type of event (thunderstorm or non-thunderstorm). The connection between the intensity function of PP and the Gumbel function (variant of the GEV) is described in (Johnson, Kotz, & Balakrishnan, 1995, p. 75)

4.4 Spatial Interpolation

Probabilistic (Kriging) and deterministic (IDW, local polynomials) techniques are used to create maps for return levels with same return period. Interpolation with Kriging requires verification of minimum technical requirements to ensure proper use of the method, particularly:

- Structural analysis, which includes data normality check, for example with Kolmogorov Smirnov or Shapiro Wilk goodness of fit tests, and if needed, data transformation to ensure data normality, e.g. using Box-Cox, and in addition, trend analysis to verify the need for trend modeling in subsequent steps.
- Semivariance Analysis: Use of available tools like cloud semivariogram, experimental semivariogram, directional semivariograms to verify isotropy or anisotropy, and different theoretical semivariograms, to ensure the best model for spatial autocorrelation, as a preliminary step to interpolation.
- Kriging Predictions: Use of different types of Kriging predictors, like simple, ordinary, universal, based on the results of the structural analysis.
- Cross Validation: Use of statistics like root mean square, average standard error, mean standardized, and root mean square standardized, that allow to measure the quality of the predictions and the magnitude of the errors.

Possible advantage of deterministic methods, is a better assessment of the local variability of spatial autocorrelation. It can also be considered with IDW or local polynomials a detailed assessment of structural analysis and cross validation. At the end of the spatial interpolation analysis all the predictions can be compared to select the most suitable result.

Main references in this research related to this matter, using *R software* are (E. Pebesma & Graeler, 2019), (Pebesma, 2004), and (Gräler, Pebesma, & Heuvelink, 2016). For the implementation of spatial statistics using vector or raster format, see (E. Pebesma, 2019a), (E. Pebesma, 2019b), and (Pebesma, 2018).

4.5 Integration with Hurricane Data

ASCE7-16 proposes the equation C26.5-2 for combination of statistically independent events, of non-hurricane and hurricane wind speed data.

$$\mathrm{P_e(y > Y_N) = 1 - P_{NH}(y < Y_N)P_H(y < Y_N)} \tag{4.12}$$

where $P_e(y > Y_N)$ is the annual exceedance probability for the combined wind hazards, $P_{NH}(y < Y_N)$ is the annual non-exceedance probability for non-hurricane winds, and $P_H(y < Y_N)$ is the annual non-exceedance probability for hurricane winds.

To understand Equation (4.12), it is important to remember that to calculate return level Y_N, for a given N-year return period, the exceedance probability $\frac{1}{N}$ of Y_N is calculated. Then, the non-exceedance probability for Y_N is $\left(1 - \frac{1}{N}\right)$. The procedure consists in the creation of a

new hazard curve, calculating all $P_e(y > Y_N)$ values for different Y_N return levels, combining hazard curves from non-thunderstorm and thunderstorm data.

Equation (4.12) can be expressed only in terms of exceedance probabilities, $P_e = 1 - (1 - P_{nh})(1 - P_h)$, where P_{nh} is the the annual exceedance probability for non-hurricane winds, and P_h is the annual exceedance probability for hurricane winds. A graphical explanation of the procedure to calculate the combined P_e for the return level $30\frac{Km}{h}$, is shown in Figure 4.13. For each cell in the study area, it is necessary to calculate a new combined hazard curve, this is, all the P_e values corresponding all different return levels (Figure 4.13).

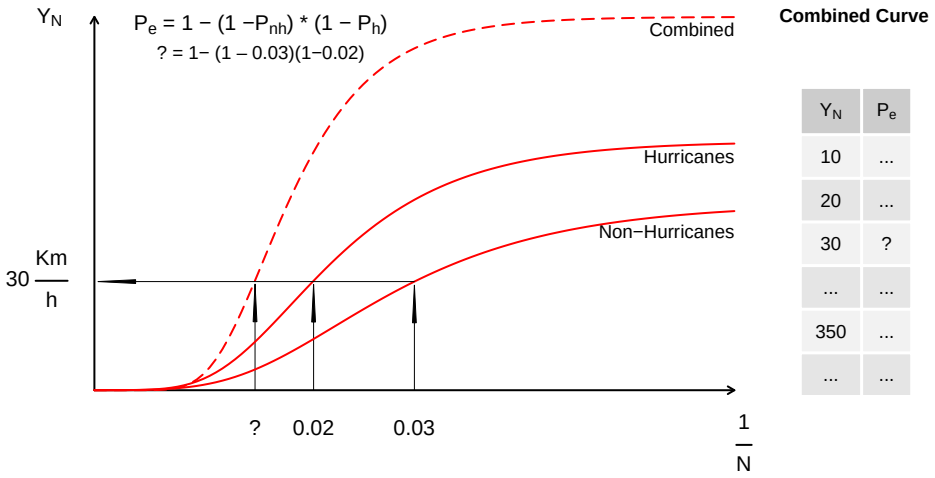

Figure 4.13: Integration of Hurricane and Non-Hurricane Data

The procedure followed in this research to generate the final maps, requires that hurricane and non-hurricane wind maps have already been generated in raster format for main return periods (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, and 7000 years). As many maps as possible are required for different return periods to estimate detailed enough hazard curves from return values (cell values).

The main elements of the implemented algorithm are: (a) select the cell size for the integrated map as the *maximum* cell size of the input maps (hurricane and non-hurricane), (b) create an empty final raster with cell size from previous step, (c) recreate input maps using zonal statistics, where zone is the empty raster, to leave only the *maximum* value of the input cells that fall within each cell in zone, (c) for each available tile in integrated map, recreate non-hurricane and hurricane hazard curves (using result from previous step), (d) use Equation (4.12) to calculate the integrated return level for each tile of the final map, and (e) create a final raster for each main return period.

An alternative approach for non-hurricane and hurricane data integration is apply Equation (4.12) with non-hurricane hazard curves obtained at each station (before spatial interpolation) in combination with hurricane hazard curves from existing studies (obtained at same location of non-hurricane stations), and finally apply a spatial interpolation process. Main disadvantage of this procedure is that no non-hurricanes wind maps will be generated. Main advantage is that the computation time and the complexity of the integration algorithm are

comparatively low, since the integration is only done in the location of the stations and not in all cells.

For the selection of the method to apply in this section, it is recommended to analyze among other aspects: (a) differences in the spatial resolution (e.g. cell size) of both types of studies need to be evaluated, because it can impact the quality of the final product, (b) prediction errors resulting from spatial interpolation can be amplified at specific cells locations, (c) creation of the algorithm, i.e. use of `st_apply` function of `stars` package (E. Pebesma, 2019b) (apply functions to raster dimensions) to avoid cell-by-cell calculations, and (d) computer processing time.

Chapter 5

Results and Discussion

This section has four main elements. First, the data source comparison (post standardization process) to face the downscaling issue, then, the process of fitting POT-PP to the ISD station 801120, next, non-hurricane maps (ISD and ERA5) and final maps will be displayed, and finally, a detailed discussion of the results and future work is highlighted.

5.1 Data Standardization and Downscaling Support

Looking for a statistical justification in the use of ISD (model) and ERA5 databases (forecast), as input data for this study, and considering the *downscaling approach* presented previously, data sources ISD and IDEAM were standardized to enable comparison. Standardization consisted of transforming the data to be equivalent to V_3 (3-s gust), 10 meters of anemometer height, and open space roughness. In the comparison process, it was checked if the velocity values (standardized) in the three sources, for equal stations and dates, were similar in magnitude.

5.1.1 Data Standardization

None of the sources required anemometer height standardization. (Lettau, 1969) was used for roughness standardization of ISD and IDEAM, applying the method station by station. Gust velocities standardization was done using Durst curve. To obtain V_3 from Durst curve, it was required to start from V_{3600} (average hourly speed), or from a different wind gust speed, for instance V_5 (5-s gust). Some elements that are relevant to each data source are described below.

ERA5:

- Variable *10m wind gust - 10fg* of ERA5 data source does not need any standardization, because it comes standardized from the source.

ISD:

- Wind velocity from ISD comes from source as V_5, that is, five seconds gust wind velocity. To standardize from V_5 to V_3, using Durst curve, the correction factor is 1.03.
- Wind velocity V_5 from 74 ISD stations, was standardized station by station, using procedure described in sections *Surface Roughness at Open Terrain*, and *Averaging Time 3-s Gust*.

IDEAM:

- It was not possible to obtain the *average hourly speed* V_{3600} directly from the institute (IDEAM), see Table 2.2, but it was calculated from received variables: a *good* estimator from *instantaneous wind velocity each 2 minutes VV_AUT_2*, and a *poor* estimator from *instantaneous wind velocity each 10 minutes VV_AUT_10*.
- The Durst curve with V_{3600} was used to calculate gust speeds. To standardize from V_{3600} to V_3, the correction factor is 1.51.

5.1.2 Data Comparison

The available IDEAM data allowed two comparison processes, with quality data for few stations, and with low quality data but available for all stations. In both cases to make the use of ISD and ERA5 viable, its time series are expected to be as similar as possible to IDEAM (field measurements). As was described in methodology section, to verify similarity two types of graphics were constructed: **time series overlay**, and **scatter plot graphics**.

Quality data comparison of instantaneous wind velocity each 2 minutes (VV_AUT_2)

Despite the fact that the ISD database (for Colombia) is based on the measured data of the IDEAM stations, their identifiers are completely different, and their names and locations in most cases have significant differences. In order to compare ISD and IDEAM sources, a manual station-by-station procedure was required to match the stations of each source. The IDEAM variable instantaneous wind velocity each 2 minutes (VV_AUT_2) was available for twenty (20) stations, of which only twelve (12) were *perfectly equivalent* to ISD stations (Table 5.1 and map in left panel of Figure 5.1)

VV_AUT_2 dataset was transformed to V_{3600} (average hourly speed) averaging all 20 values available per hour. For twelve matching stations, wind velocity V_{3600} was standardized station by station using procedure described in *Surface Roughness at Open Space* section and *Averaging Time 3-s Gust*. For the same twelve ISD and IDEAM standardized stations, a comparison was done against matching ERA5 stations (the corresponding cell in ERA5 that has within ISD and IDEAM locations).

The stations described in each row of the Table 5.1 were compared by generating scatter plots and common time series graphics. For the stations IDEAM 28025502 and ERA5 416 corresponding to the tenth row of the mentioned table, there was high correspondence between sources. Unfortunately, in the stations corresponding to the other eleven rows, downscaling support was not reflected.

Table 5.1: Quality Data Comparison

ISD ID	IDEAM ID	ERA5 ID, (col,row), [lon,lat]
803980	48015050	3320, (37, 68), [-70, -4.25]
803700	52055230	2309, (6, 48), [-77.75, 0.75]
802110	26125061	1582, (14, 33), [-75.75, 4.5]
802100	26125710	1533, (14, 32), [-75.75, 4.75
801120	23085270	1240, (15, 26), [-75.5, 6.25]
801100	27015330	1240, (15, 26), [-75.5, 6.25]
800970	16015501	909, (27, 19), [-72.5, 8]
800940	23195502	1102, (24, 23), [-73.25, 7]
800630	13035501	749, (14, 16), [-75.75, 8.75]
800360	28025502	416, (24, 9), [-73.25, 10.5]
800350	15065180	221, (25, 5), [-73, 11.5]
800280	29045190	312, (18, 7), [-74.75, 11]

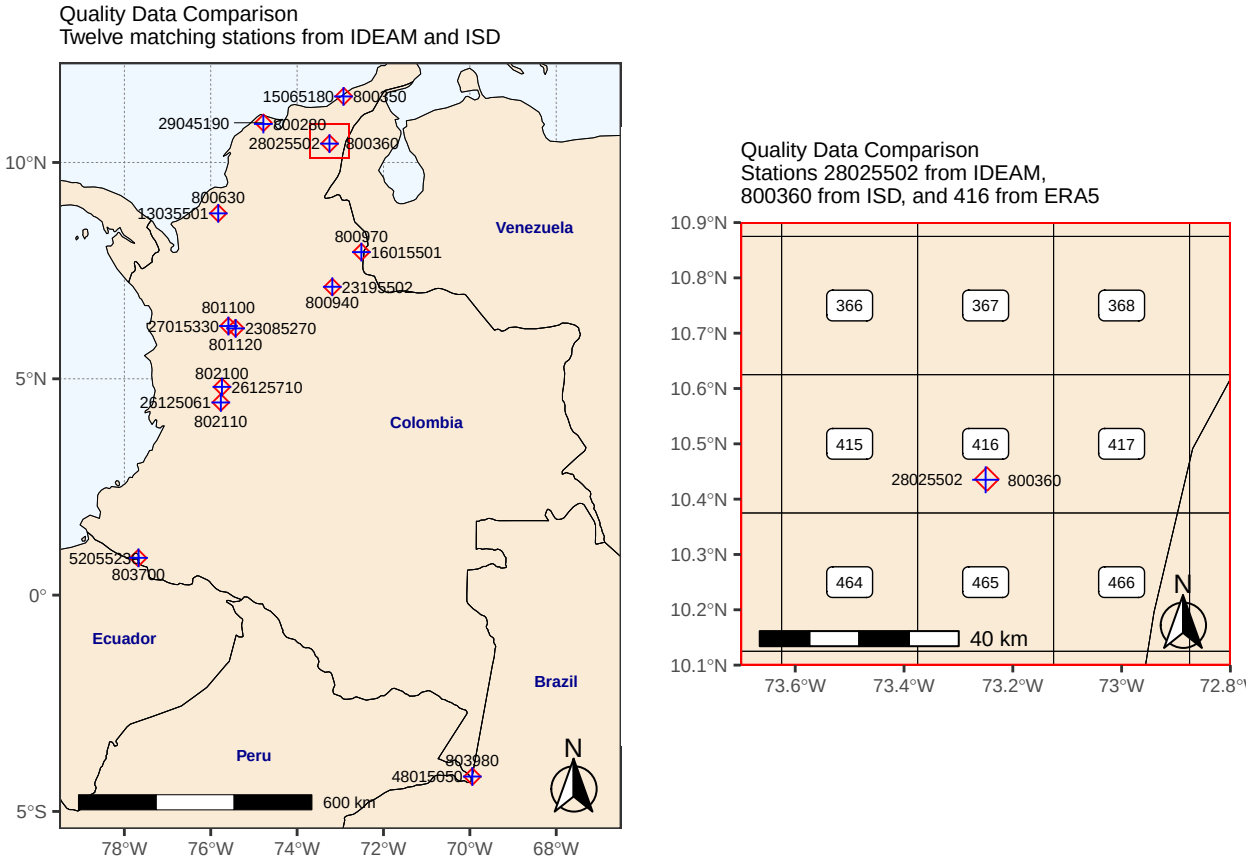

Figure 5.1: IDEAM VV_AUT_2 - Quality Data Comparison

The stations corresponding to the map in the right panel of Figure 5.1 and scatter plot in Figure 5.2, show high correspondence between sources because green regression line (empirical) is very similar to 45° line (theoretical). Axis x in m/s correspond to IDEAM station

28025502, and axis y (with same units) contains ERA5 416 station (cell with center point in $-73.25°$ longitude, and $10.5°$ latitude). The points in the upper part of Figure 5.2 that move away from the global trend of the correlation, correspond to erroneous field measurements (sensor failure) of the IDEAM meteorological station. The graph shows some statistics and coefficients of the linear regression model; the value **P** is equivalent to the p-value for the slope of the regression line, i.e. the probability of observing any value equal or larger than t-statistic.

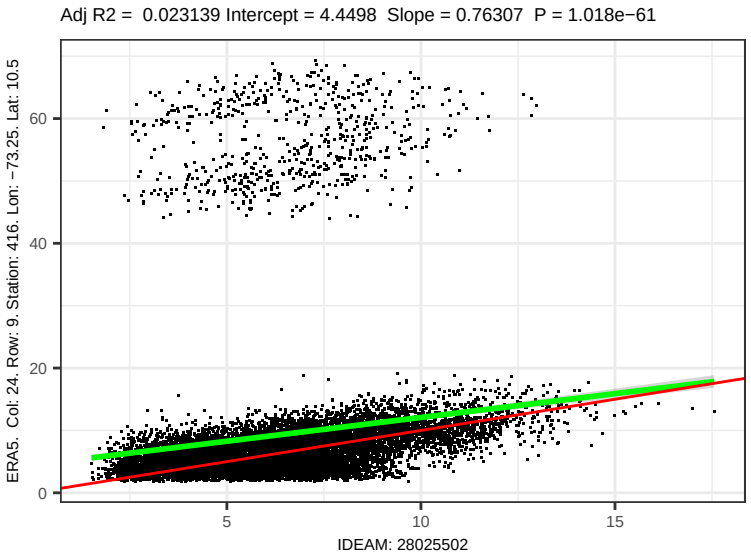

Figure 5.2: Quality Data Comparison. High Similarity Between Sources

Non-quality data comparison of instantaneous wind velocity each 10 minutes (VV_AUT_10)

IDEAM variable instantaneous wind velocity each 10 minutes (VV_AUT_10) was available for 204 stations, despite V_{3600} calculated from this source is not an accurate or quality estimator, the comparison results are shown in Figure 5.3. Downscaling support was 'Good' comparing IDEAM and/or ISD stations with twenty-three (23) ERA5 stations (2261, 1971, 2066, 2020, 2260, 1875, 2213, 2637, 1442, 1583, 1501, 1582, 1381, 1493, 1485, 1397, 1338, 1055, 511, 1644, 515, 221, 1038), and 'Very Good' comparing IDEAM and/or ISD with five (5) ERA5 stations (265, 360, 78, 312, 416).

Table 5.2 shows in each row compared stations with 'Very Good' downscaling results. Figure 5.4 shows an example of a very good match in the time series plot for the ERA5 station 78 vs IDEAM stations 15075501 and 15079010. Figure 5.5 shows four different very good matching scatter plots: (a) IDEAM 15015120 vs ERA5 265, (b) IDEAM 29004520 vs ERA5 312, (c) IDEAM 15079010 vs ERA5 78, and (d) IDEAM 15075501 vs ERA5 78. Red line in each graphic represent the desired $45°$ line, where all points should fall, if the data sources would be exactly the same (theoretical behavior when there is equivalence of sources), and green line represents the linear regression line (empirical or real behavior when making the comparison).

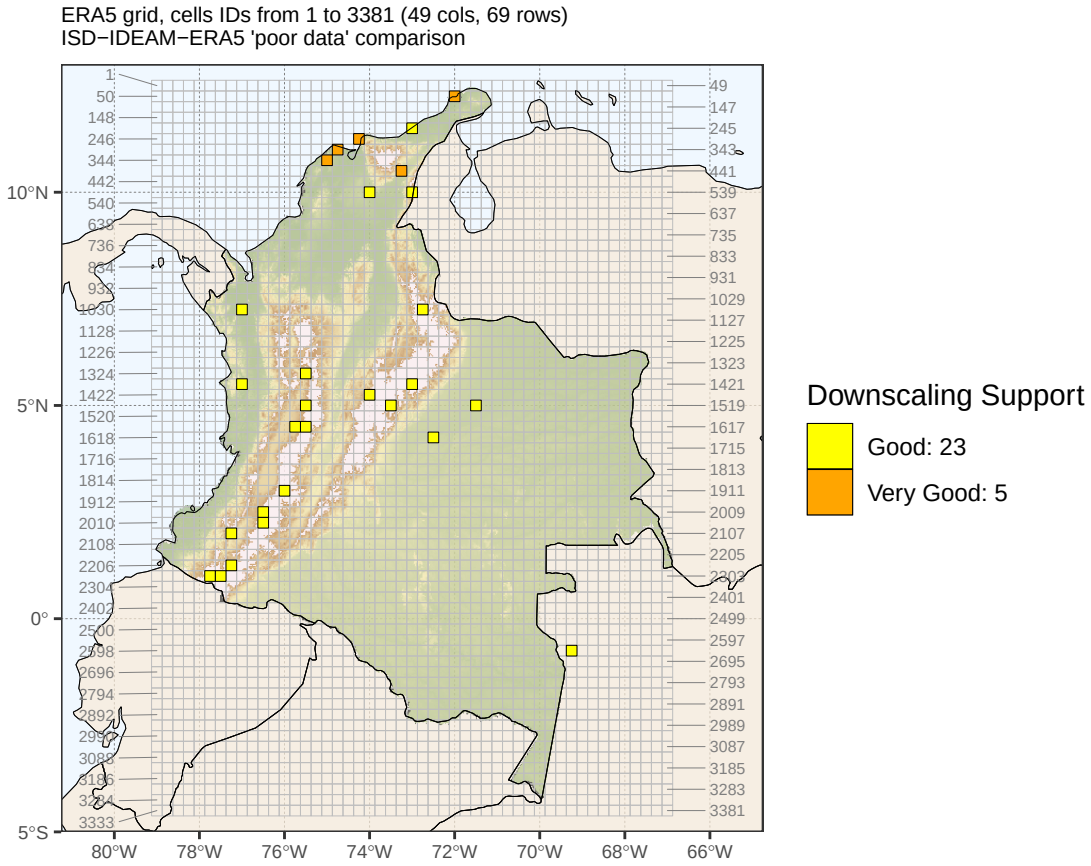

Figure 5.3: IDEAM VV_AUT_10. Non-Quality Data Comparison

Table 5.2: Non-Quality Comparison. Very good downscaling support.

ISD ID	IDEAM ID	ERA5: ID, (col,row), [lon,lat]
NA	16015501	78, (29, 2), [-72, 12.25]
NA	15079010	78, (29, 2), [-72, 12.25]
NA	15075501	78, (29, 2), [-72, 12.25]
NA	15015120	265, (20, 6), [-74.25, 11.25]
NA	29004520	312, (18, 7), [-74.75, 11]
800280	29045190	312, (18, 7), [-74.75, 11]
NA	29045000	360, (17, 8), [-75, 10.75]
NA	28025502	416, (24, 9), [-73.25, 10.5]
800360	28035060	416, (24, 9), [-73.25, 10.5]

Note that although the Table 5.2 has nine records, each for a different IDEAM station, there are only five ERA5 stations because some of them are repeated, for example station 78 that appears three times. Additionally, there are only two ISD stations. The value 'NA' means that for the corresponding ERA5 and IDEAM station (same row), there is not an ISD station located inside the ERA5 cell space ($0.25° * 0.25°$). Velocities in the axis y of Figure 5.4 and all the axis in Figure 5.5 are in m/s. The regression model coefficients are shown for the green regression lines in Figure 5.5: adjusted R2, line intercept, line slope, and the

probability $Pr(>|t|)$ (the significance of the estimate increases as p-value decreases).

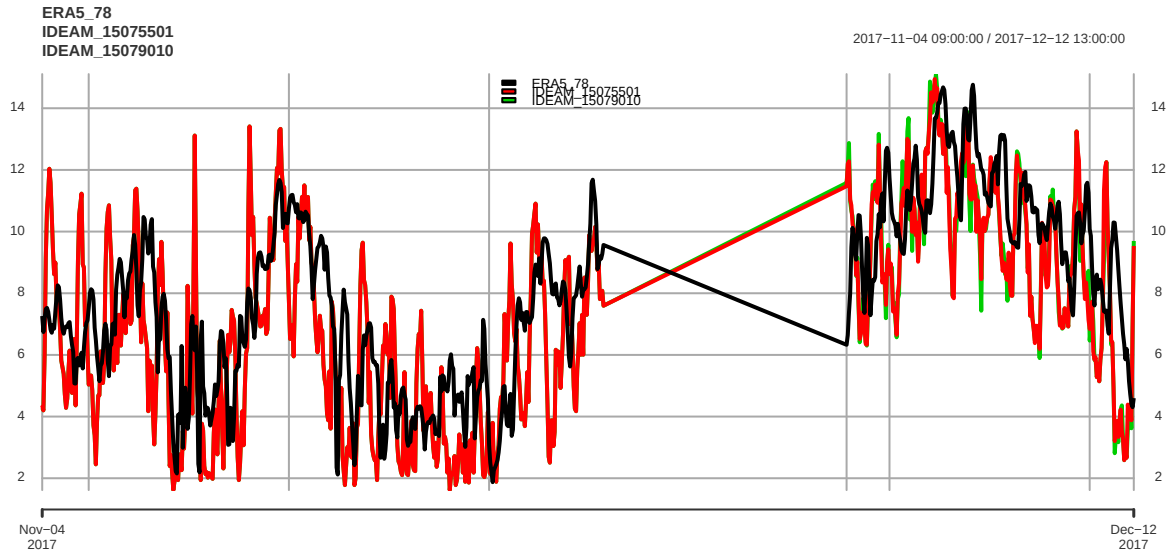

Figure 5.4: Time Series Graphic for 'Very Good' Downscaling Support

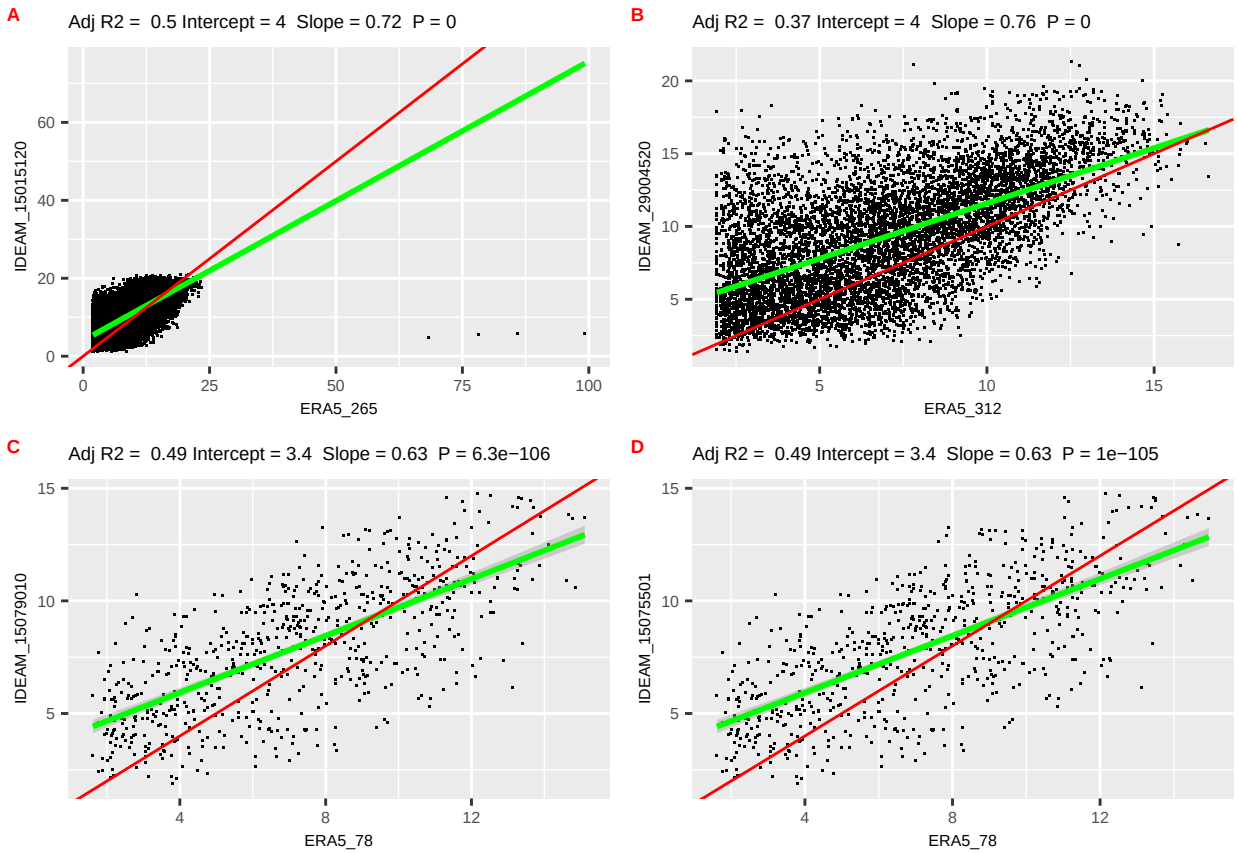

Figure 5.5: Scatter Plots for 'Very Good' Downscaling Support

5.2 POT-PP for ISD Station 801120

Figure 5.6 shows the satellite image (source Google Earth) of ISD station 801120, located in the international airport 'José María Córdova', municipality of Rio Negro (Antioquia, Colombia), with latitude 6.125°, and longitude −75.423° WGS84 coordinates. Red circle represents an influence radius of 800 meters. Table 5.3 shows different calculations related to correction factors applied to this station, using procedure described in sections *Surface Roughness at Open Terrain*, and *Averaging Time 3-s Gust*.

Figure 5.6: Location of ISD Station 801120

Table 5.3: Corrections Factors for ISD Station 801120

Variable	Value
Roughness - Z_o	0.05
Empirical exponent - α	8.38
Gradient height - z_g	310.56
Exposure coefficient - K_z	0.88
$F_{exposition}$	1.07
Gust factor for V_3	1.03

5.2.1 Raw Data, De-clustering, and Thresholding

As storm information is not available for any of the data sources, all the data for the station was classified as *non-thunderstorm*. According to *POT-PP* method described in *Methodology*, the first process applied to original time series *raw data*, is *De-clustering*, and then, *Thresholding*.

Non-thunderstorm raw data for ISD station 801120 has 2931 records, from 1986-12-06 12:00:00 to 2019-03-01 12:00:00, corresponding to a total amount time in days of 11739, and to an average number of events per year of 18.9, which means that the average duration of an event is 19.3 days (average size in days of a cluster). After *De-clustering* and *Thresholding* processes, the number of records decreases to 181. Time series graphics are related in

Figure 5.7, showing the data before (left) and after (right) applying the mentioned processes. Detailed yearly statistics are reported in Table 5.4, also including summary for before (left), and after (right).

Table 5.4: Yearly statistics of raw data and de-clustered data for ISD station 801120

	Raw Data				Declustered Data			
Year	Count	Mean	Min	Max	Count	Mean	Min	Max
1986	63	45.2	27.9	163.3	7	106.4	43.8	163.3
1987	192	36.1	26.7	87.6	10	61.0	45.0	87.6
1988	234	43.8	26.7	90.4	23	64.2	45.0	90.4
1989	256	44.2	27.9	103.6	19	64.4	45.0	103.6
1990	250	44.9	26.7	103.6	21	67.2	45.0	103.6
1991	149	38.7	26.7	127.5	20	58.6	45.0	127.5
1992	126	35.2	26.3	81.7	9	52.6	43.8	81.7
1993	109	36.3	26.3	79.7	13	53.5	43.8	79.7
1994	124	36.8	26.7	79.7	12	56.1	45.0	79.7
1995	89	33.3	26.7	111.5	2	77.7	43.8	111.5
1996	70	35.6	26.7	87.6	6	65.7	43.8	87.6
1997	71	36.6	26.7	119.5	4	86.9	49.0	119.5
1998	65	33.8	27.9	61.4	2	54.6	47.8	61.4
1999	48	31.7	26.7	47.8	1	47.8	47.8	47.8
2000	69	33.4	26.7	87.6	3	68.3	55.8	87.6
2001	62	29.9	26.7	39.8	0	NA	NA	NA
2002	94	33.3	26.7	71.7	5	54.2	43.8	71.7
2003	78	31.5	26.7	71.7	1	71.7	71.7	71.7
2004	60	31.9	26.7	51.8	2	48.4	45.0	51.8
2005	59	33.3	26.7	94.4	2	69.1	43.8	94.4
2006	55	32.6	26.7	164.1	1	164.1	164.1	164.1
2007	25	29.8	26.7	39.0	0	NA	NA	NA
2008	13	36.1	26.7	96.4	1	96.4	96.4	96.4
2009	36	31.6	26.7	82.1	1	82.1	82.1	82.1
2010	31	43.0	27.9	119.5	8	83.0	61.4	119.5
2011	32	29.2	26.7	41.0	0	NA	NA	NA
2012	82	31.9	26.7	87.6	4	64.5	43.0	87.6
2013	91	29.7	26.7	37.0	0	NA	NA	NA
2014	95	30.1	26.7	47.8	1	47.8	47.8	47.8
2015	129	30.3	26.7	51.8	1	51.8	51.8	51.8
2016	33	30.7	26.7	87.6	1	87.6	87.6	87.6
2017	18	31.3	26.7	67.7	1	67.7	67.7	67.7
2018	22	31.0	26.7	39.8	0	NA	NA	NA
2019	1	28.7	28.7	28.7	0	NA	NA	NA

It can be seen in the Table 5.4 that de-clustered data has zero records for some years. This situation is due to that all the data for each one of those years (2001, 2007, 2011, 2013, 2018, and 2019), belonged to a cluster that started the previous year or finished the next year, and the unique chosen maximum value (the value representative for the cluster) was found in the previous or next year, but not in mentioned years of zero records.

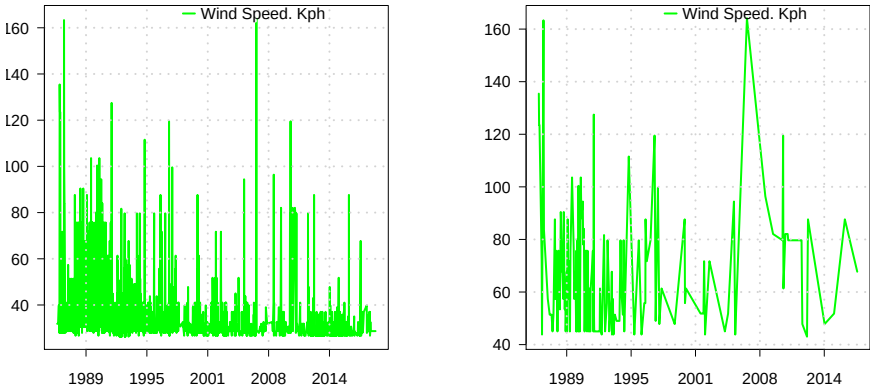

Figure 5.7: Non-Storm Time Series. ISD Station 801120. Raw Data(L). De-clustered(R)

Using de-clustered data, it is only necessary to calculate optimal threshold for non-thunderstorm data, because there are no records classified as thunderstorm in any data source. Many non-thunderstorm thresholds were tested, to choose the best one using the W statistic, as described in section *Thresholding* of the *Methodology*.

Figure 5.8 shows a very good fit in resulting W-Statistic plot, for optimal non-thunderstorm threshold $b_{nt} = 42\frac{km}{h}$, with a minimum W distance of 0.47, for ISD station 801120, where empirical values (black points) are very close or similar to theoretical values (red line).

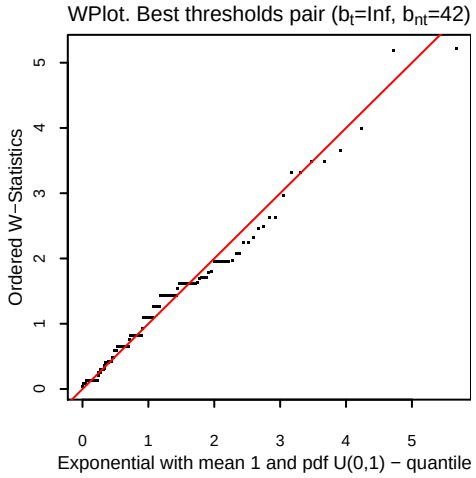

Figure 5.8: POT - Thresholding. ISD Station 801120

5.2.2 Fitted PDF and CDF, and Goodness of Fit

Equation (3.18), defined in section *POT-PP* of the *Methodological Framework*, was used as intensity function $\lambda(t, y) = \lambda_{nt}(y)$. When shape ζ_t is equal to zero (as it is in this study) an equivalent intensity function is described in Equation (4.10) defined in terms of the parameters location (ω_t), and scale (ψ_t). Related PDF and CDF functions are referenced

in Equations (4.7), where the domain D constraint the data above the threshold b, and the time to a non-thunderstorm period, and (4.8) respectively.

- Intensity function: $\frac{1}{\psi_{nt}} \exp\left(\frac{-(y-\omega_{nt})}{\psi_{nt}}\right)$
- PDF: $f(t,y) = \frac{\lambda(t,y)}{\int_D \lambda(t,y)\,dt\,dy}$
- CDF: $F(t,y) = P(y \leq Y) = \frac{\int_b^Y \lambda(y,t)\,dy}{\int_b^\infty \lambda(y,t)\,dy}$

After fitting the intensity function to the domain D, the resulting parameters for ISD station 801120, are location ω_t equal to -55.62, and scale ψ_t equal to 23.4. Figure 5.9 shows the histogram and fitted PDF in panel A, Q-Q plot (theoretical quantiles against empirical ones) in panel B, empirical cumulative distribution against fitted CDF in panel C, and P-P plot (theoretical probabilities against empirical ones) in panel D. In all four panels, it can be seen that there is a very good visual correspondence between empirical data (points and histogram) and theoretical adjustment (lines).

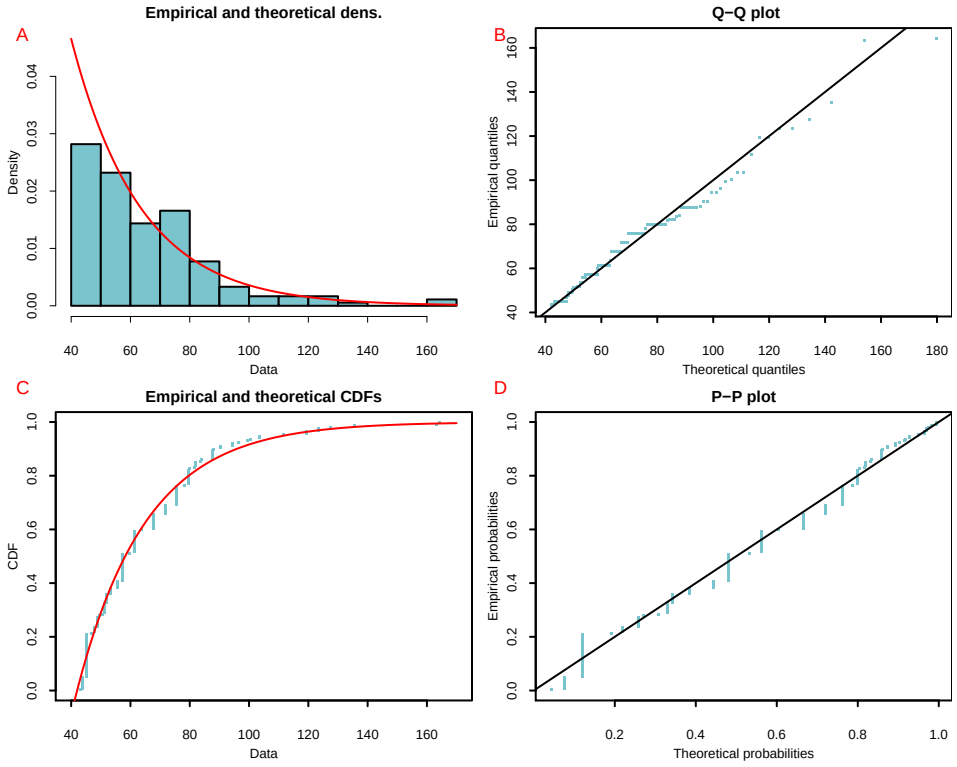

Figure 5.9: Goodness of Fit Graphic Diagnosis. Station 801120

Results of formal goodness of fit statistics for 'Kolmogorov-Smirnov D', 'Cramer-von Mises T' and 'Anderson-Darling A' are 0.089, 0.21, and 1.68 respectively. For a proposed null hypothesis, which indicates that the data conforms to a POT-PP, all resulting p-values using statistics D, T and A, confirm that there is no statistical evidence to reject stated hypothesis. Resulting p-value for statistic D is 0.11. Another available criterion to measure the quality

of the fitted process are 'Akaike's Information Criterion', and 'Bayesian Information Criterion', with values 1505.2, and 1508.4 respectively. The Root Mean Square Error (RMSE), calculated using theoretical versus empirical CDF, is 0.023.

5.2.3 Hazard Curve and Return Levels RL

Hazard curve is the solution to Equation (3.20), but eliminating from it elements related to thunderstorms the equation is simplified to $A_{nt} \int_{Y_N}^{\infty} \lambda_{nt}(y)\, dy = \frac{1}{N}$, where A_{nt} is the average time of non-thunderstorm events by year, and Y_N is the return level or extreme wind velocity, corresponding to the N-years return period or MRI. Replacing in this equation the intensity function $lambda_{nt}$, and solving Y_N for all possible values of MRI, will provide hazard curve displayed in Figure 5.10.

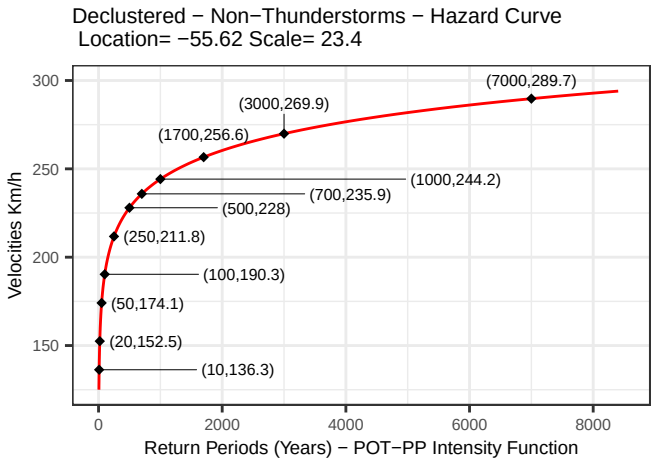

Figure 5.10: Hazard Curve. Station 801120

Table 5.5: Return Levels -RL for Typical Mean Return Intervals - MRI. ISD station 801120

MRI (years)	Return Level (km/h)
10	136.30
20	152.48
50	174.10
100	190.32
250	211.76
500	227.98
700	235.85
1000	244.20
1700	256.61
3000	269.90
7000	289.73

Return levels of interest for this research, correspond to 700, 1700 and 3000 years of MRI, however, due to the mechanism of *Integration with Existing Hurricane Studies*, described in *Methodology*, it is necessary to extract for all stations values related to typical return periods, as shown in the Table 5.5.

5.2.4 Comparison with POT-Poisson-GPD and Common Extreme Value Distributions

To enable a comparison between (a) POT-PP (previous section), (b) *POT-Poisson-GPD*, and (c) the fitting process of common extreme value distributions (GPA, GEV, GUM) without using POT method, i.e., using the generic concept of *Hazard Function HF* (see *Theoretical Framework*), a whole automation process was done to calculate return levels and errors using mentioned alternatives, bearing in mind that in all cases *maximum likelihood* was used to calculate the parameters.

To this day (Feb 2020), there is no R package available that implements POT-PP, in contrast, there are many packages that implement POT-Poisson-GPD, this reflects that globally, extreme value analyzes are mainly done with the latter. The following SIX R packages were used: (a) `extRemes` (Gilleland, 2019), (b) `ismev` (Janet E. Heffernan with R port & Alec G. Stephenson., 2018), (c) `evd` (Stephenson, 2002), (d) `Renext` (Deville & IRSN, 2016), (e) `evir` (Pfaff & McNeil, 2018), and (f) `fExtremes` (Wuertz, Setz, & Chalabi, 2017).

Resulting return levels and errors using mentioned packages are reported in Table 5.6. Similarly, return levels calculated from the adjustment of the probability distributions GPA, GEV, and Gumbel are shown it Table 5.7.

Table 5.6: POT-Poisson-GPD. Return Levels in km/h

PACKAGE	RETURN LEVELS FOR TYPICAL MRIs											ERROR
	10	20	50	100	250	500	700	1000	1700	3000	7000	RMSE
extRemes	155.6	169.3	187.2	200.4	217.6	230.3	236.4	242.8	252.2	262.1	276.6	0.057
ismev	155.5	169.3	187.1	200.4	217.5	230.1	236.2	242.6	252.0	261.9	276.4	0.057
evd	155.6	169.3	187.2	200.4	217.6	230.3	236.4	242.7	252.2	262.1	276.6	0.057
Renext Renouv	155.6	169.3	187.2	200.4	217.6	230.3	236.4	242.7	252.2	262.1	276.6	0.057
evir	155.0	168.5	185.8	198.6	215.1	227.3	233.1	239.2	248.2	257.6	271.3	0.058
fExtremes	155.5	169.3	187.2	200.4	217.5	230.2	236.3	242.6	252.0	261.9	276.5	0.057
Renext 2 parameters	200.8	203.9	206.5	207.8	208.9	209.4	209.6	209.7	209.9	210.1	210.3	0.337

Table 5.7: Common Extreme Value Distributions. Return Levels in km/h

EVD		RETURN LEVELS FOR TYPICAL MRIs											ERROR
	NAME	10	20	50	100	250	500	700	1000	1700	3000	7000	RMSE
gpa	Generalized Pareto	149.6	160.6	174.2	183.9	195.8	204.2	208.2	212.2	218.0	223.9	232.2	0.048
gev	Generalized Extreme Value	172.5	198.8	239.2	274.8	329.5	377.8	403.5	432.7	479.9	536.0	631.7	0.058
gum	Gumbel	140.9	152.1	167.0	178.2	193.0	204.3	209.7	215.5	224.1	233.3	247.0	0.067

5.3 Wind Maps

Maps in this section correspond to: (a) existing hurricane maps from previous studies, (b) non-hurricane wind maps created in this study with POT-PP (ERA5 and ISD), and (c) final maps (integrated results of hurricane and non-hurricane studies) for ERA5 and ISD.

5.3.1 Existing Hurricane Maps

The Colombian consulting firm *Ingeniar Ltda*, following the methodology described in (CIMNE, 2015), and (CIMNE, 2017), has provided raster wind maps for return periods 10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, and 7000 years, resulting from the probabilistic study of winds due to hurricanes in the Colombian Caribbean Sea and the surrounding continental zone. Figure 5.11 shows three of mentioned maps.

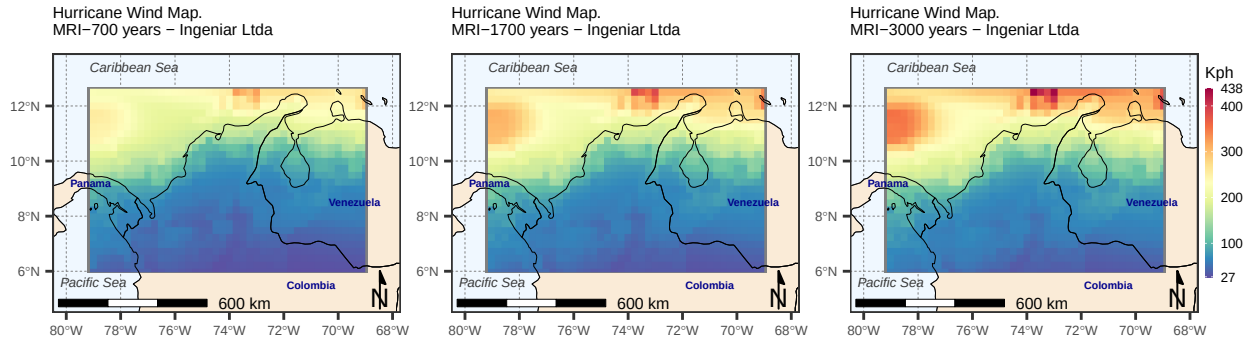

Figure 5.11: Ingeniar Hurricane Wind Maps

5.3.2 Non-Hurricane Maps

Using POT-PP to calculate RL in ISD stations, continuous maps covering the study area were created using *Spatial Interpolation Techniques* as described in *Methodology*.

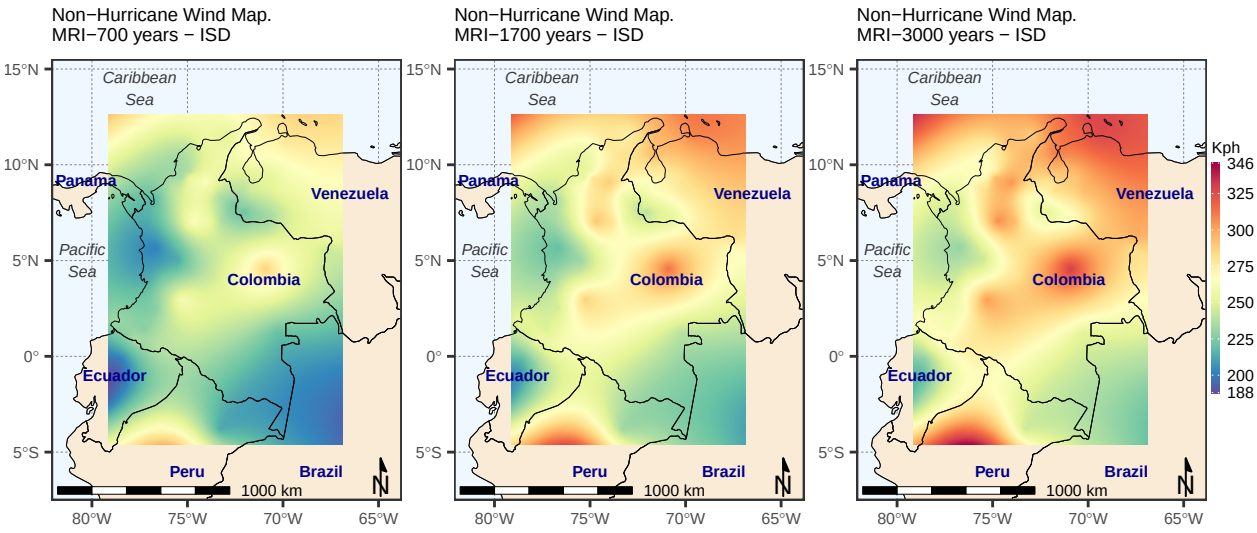

Figure 5.12: ISD Non-Hurricane Wind Maps

The comparison process between measured data and ERA5 data did not give good results, but this is mainly due to inadequate data received from IDEAM. It was decided to execute the process POT-PP in the ERA5 variable *ten meters wind gust fg10*, but remembering that once adequate field data is obtained, the downscaling support must be verified to guarantee the feasibility of the results. No interpolation process was required because the calculated RLs at each station, represent predefined square cells of 0.25° decimal degrees.

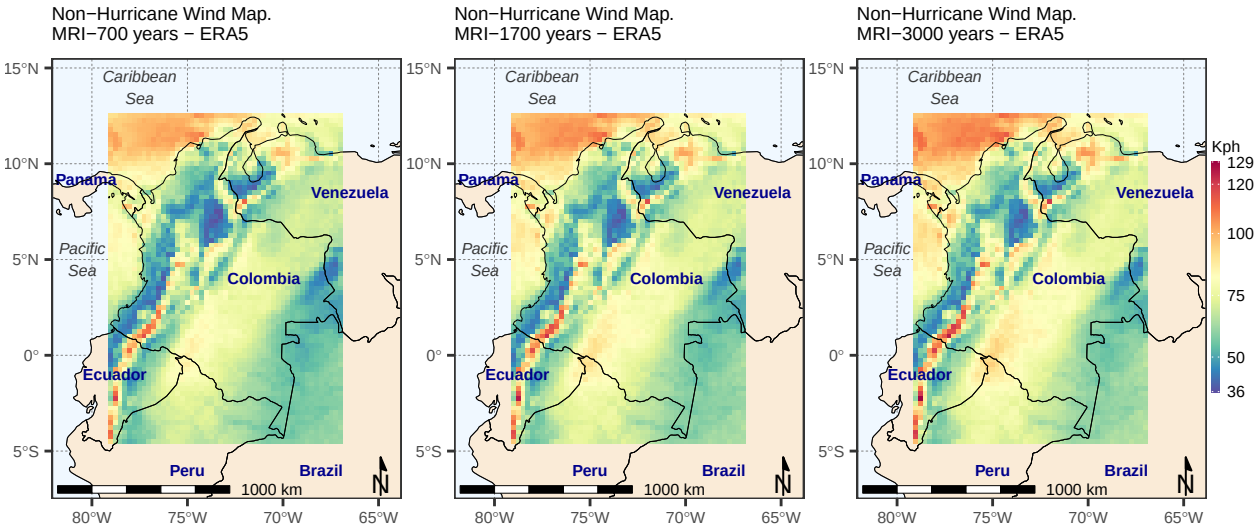

Figure 5.13: ERA5 Non-Hurricane Wind Maps

5.3.3 Combined Maps

Following the procedure described in *Integration with Hurricane Data*, final wind maps were created, combining existing data for hurricane studies, and non-hurricane maps produced in this study.

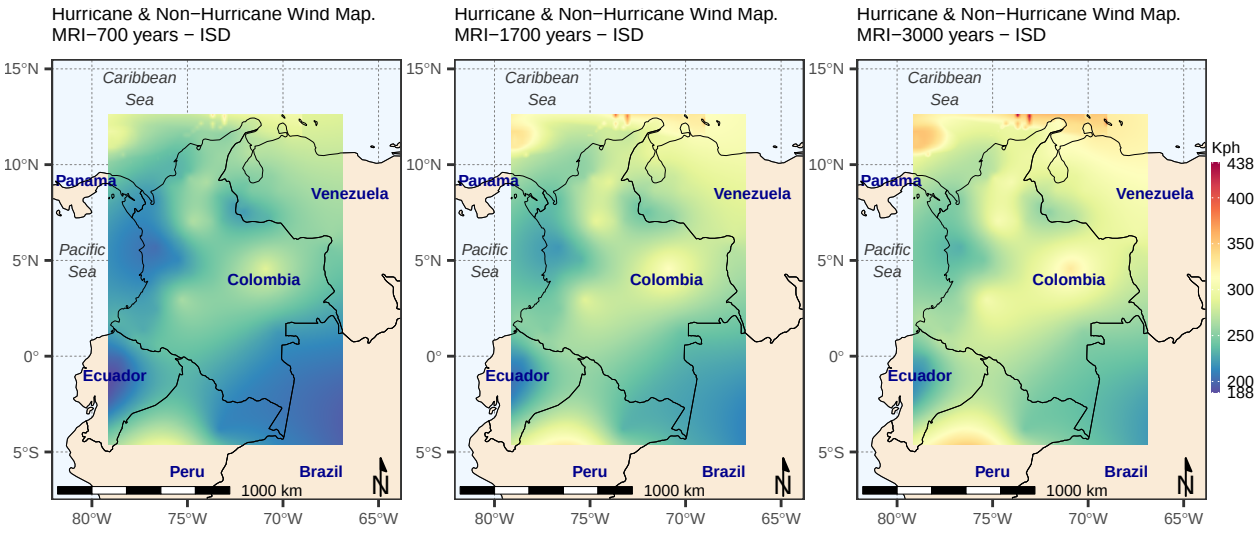

Figure 5.14: ISD Hurricane & Non-Hurricane Wind Maps

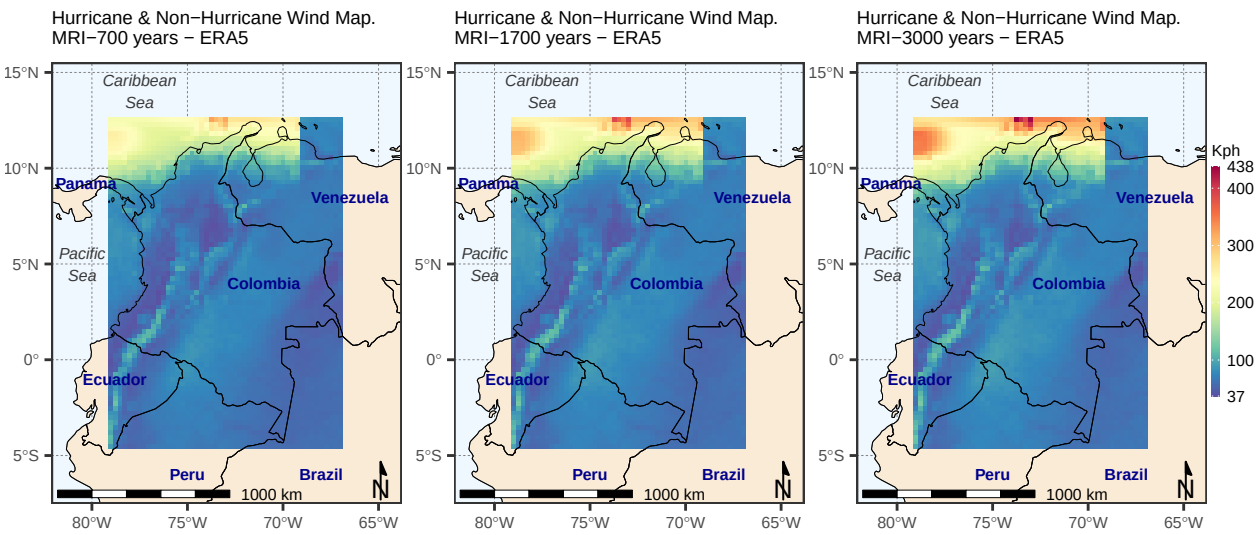

Figure 5.15: ERA5 Hurricane & Non-Hurricane Wind Maps

5.4 Final Discussion and Future Work

The main difference between POT-PP and POT-Poisson-GPD is that in the latter, wind quantities are adjusted to a GPD, and the time is adjusted to a Poisson Process (1D), while in the former, time and magnitude are adjusted to a Poisson Process (2D). If there is no weather classification available (storm and no storm) in the wind time series, POT-PP loses its advantages and resembles in potential and scope to POT-Poisson-GPD, because the intensity function varying only in magnitude, becomes similar to a GPD. For this reason, POT-PP method is really useful, if historical classification of time series is available, record by record, in storm and non-storm.

Regarding the comparison of the data, it must be remembered that the basis of comparison, that is, the one that represents the truth in the field - IDEAM field measurements, was not fully available, what disturbed the process since before starting it. On the other hand, there are many uncertainties with respect to the model that represents the ISD database, because first, the available documentation does not specify whether it is an average or a gust data, second, the comparative graphs showed that ISD database did not represent a continuous variable (vertical or horizontal stripes in scatter plots), and finally, the comparisons against IDEAM never showed good results.

With respect to ERA5 database, although the comparative results showed greater similarity, it should be remembered that each record in the time series does not represent a point value, on the contrary, it represents a square cell of 0.25 decimal degrees. The IDEAM stations with which the comparison was made, can fall into any location of the cell, and constitute only a very local condition, that is not represented by an average forecast for the whole cell size. This considers that Colombia is a tropical region with a widely diverse territory (mountains, plains, rivers, forests, etc.) and climate. So the possible similarity between IDEAM and ERA5 is limited by this condition.

This classification by time of historical series, is useful because it allows to define more precisely the average rate of events per year (Poisson process rate), which in POT-PP is represented by the average amount of events time per year, this is, components A_t and A_{nt} of Equation (3.20) $A_t \int_{Y_N}^{\infty} \lambda_t(y)\, dy + A_{nt} \int_{Y_N}^{\infty} \lambda_{nt}(y)\, dy = \frac{1}{N}$, used to calculate return levels Y_N.

By the lack of thunderstorm and non-thunderstorm information, it is impossible to calculate which part of the annual time belongs to storm and which to non-storm. As the available data were all assumed as non-thunderstorm data, this average time of events per year will always result in a fixed wrong value of 365 days, the maximum possible value. For ISD, this condition is reflected in high and unlikely final results. However, this condition did not affect the results in the ERA5 database in the same way. Although also in ERA5, all the data were classified as "non-thunderstorm", and the average time of events was always 365 days, an additional condition made the final result more realistic. Contrary to what happens in the ISD database, where time series have many gaps and there is lack of information, used ERA5 database has the full time series, hour by hour, from 1979 to 2019. Following the theory behind POT-PP, this implies that there is only one cluster for the whole time series, which would leave a single data after the de-clustering, canceling the entire subsequent process. Our proposed solution was to work only with one data per week, the maximum, which implies that the de-clustering process will have no effect, since it is based on 4-day gaps, resulting in more events above the threshold, exactly 48 events for each of the 40 years of history, which translates into greater averaging of the final wind data.

One of the objectives of the investigation was to compare different methods in the calculation of return levels, and that was achieved using in all cases, both POT-Poisson-GPD, and POT-PP. The Poisson for POT-Poisson-GPD, does not accept data classification by time (storm or non-storm), because it is one-dimensional and data must represent a single general type of event: wind. Despite that, it is important to emphasize that the shortcomings in the calculation of the Poisson process rate, similarly affected the application of both methods, so in all cases, the results have the same limitation, and the use of POT-Poisson-GPD does not represent best quality in the results.

In relation to the use of PostgreSQL to store time series, seeking to decrease RAM load to avoid memory overflow or slow processes, it can be concluded that although it is an efficient solution, it is also a delayed solution in terms of its implementation effort (process text files, load them, and unstack them in the database). In addition, replication of research for future work is more difficult by the need to install the database engine and restore the database backup, compared to the simple use of text files. It is recommended to use a database for analysis and processing with *R* and `tidyverse` lazy `tibble` data-frame technologies, when data sources are already stored in a database as PostgreSQL.

Final recommendation to optimize memory management when time series are processed are: (a) to work directly with text files, but make a good code design avoiding to load all time series into memory at the same time, i.e., serial processing file by file, (b) make sure to remove used objects from memory `rm("object")`, (c) make memory garbage collection `gc()` to frees up unwanted objects, or (d) reuse the same variables in each cycle of the process, to use always the same previously allocated memory spaces. Next chunk of code shows some

important memory management options.

```
#List objects in current environment
  ls()
#Remove all objects in current environment
  rm(list=ls())
#After removing object, apply garbage collection
  gc()
#List object sorted by size
  sort(sapply(ls(),function(x){object.size(get(x))}))
#Remove an specific object
  rm("name")
#Review some memory values
  library(pryr)
  mem_used()
  memory.size()
  memory.limit()
#Change memory limit
  memory.limit(size = value)
```

For big NetCDF files, as in this study where variable *fg10* (3-s wind gust) stored 3.381 (49 columns x 69 rows) cells and in each cell 356.472 hours, representing 1.205'231.832 different wind records in a file size of 3.53 GB, it is recommended to get specific needed slices instead of loading the entire file into memory. For instance, use parameters `start`, and `count` of function `ncvar_get` (R package `ncdf4`) to get desired NetCDF records, as shown in code below where time series of first ERA5 cell/station (upper-left) is assigned to variable *firstcelltimeseries*.

```
library(ncdf4)
ncname <- "outfile_nc4c_zip9.nc"
ncin <- nc_open(filename)
#First column
  firstcol=1
#First row
  firstrow=1
#First time
  firsttime = 1
#Number of cols to get
  ncols=1
#Number of rows to get
  nrows=1
#Total number of time records (time dimension)
  timedim <-  dim(ncvar_get(ncin, "time"))
#Number of time records to get
  ntime = timedim
  firstcelltimeseries = ncvar_get(nc=ncin, varid='fg10',
                        start=c(firstcol,firstrow,firsttime),
                        count=c(ncols,nrows,ntime))
```

To improve the quality of extreme wind analysis in future research, the inclusion of seasonal effects is recommended. This can be done in two ways, first, using a separate POT-PP for each season, and second, model the Poisson process parameters (location, scale and shape) as sinusoidal functions of time. Finally, it is possible to include more formal statistics (not only graphics) using reliable and complete measured data to face the downscaling challenge.

Chapter 6

Conclusions

The conclusions of this study are:

- Final maps using ERA5 forecast database, representing return periods of 700, 1700, and 3000 years, are the extreme velocities needed as input load for the design of structures of different use category in the study area. Nevertheless, by one hand, full data from the IDEAM source is needed to enable the validation of downscaling support, on the other hand, it is essential to include in the study the classification of thunderstorm and non-thunderstorm data to achieve more realistic results, and finally, an additional conservative calibration process is needed, where to each municipality is assigned only a wind velocity in order to define final values that will be part of the structure design norm.
- In the absence of wind field measurements, alternatives data sources as ISD and ERA5 can be a viable source of data to calculate extreme wind events, but always must be searched for statistic or graphic support for the downscaling issue, and at the end a process of calibration is needed for each particular case.
- A powerful **R** tool was implemented to apply extreme value analysis using POT-PP method allowing the calculation of comparative results with POT-Poisson-GPD approach (using existing packages).
- Results of this research could be the starting point of a process to update wind maps in countries with information issues.
- Output results of this research will contribute to reduce the risk of infrastructure collapse, representing a favorable impact in human lives, material losses, and disaster prevention.

For a detailed analysis of the results, refer to *Results and Discussion* section, and for a discussion about the project and its relevant topics, refer to *Final Discussion*.

References

ADB. (2014). *Guidelines for wind resource assessment: Best practices for countries initiating wind development*. Asian Development Bank. Retrieved from https://www.ebook.de/de/product/30686652/guidelines_for_wind_resource_assessment.html

ASCE. (2017). *Minimum design loads and associated criteria for buildings and other structures (asce7-16)*. American Society of Civil Engineers. Retrieved from https://www.ebook.de/de/product/35017614/american_society_of_civil_engineers_minimum_design_loads_and_associated_criteria_for_buildings_and_other_structures_7_16.html

Beirlant, J., Goegebeur, Y., Teugels, J., & Segers, J. (2004). *Statistics of extremes: Theory and applications*. John Wiley & Sons, Ltd. http://doi.org/10.1002/0470012382

Berhane, F. (2016, April). Working with databases in r. Wep Page. https://datascienceplus.com/. "Data Management in R". Retrieved from https://datascienceplus.com/working-with-databases-in-r/

Castillo, E., Hadi, A. S., Balakrishnan, N., & Sarabia, J.-M. (2005). Extreme value and related models with applications in engineering and science.

CIMNE, I. (2015). *Update on the probabilistic modelling of natural risks at global level: Global risk model* (technical report). The United Nations Office for Disaster Risk Reduction - UNISDR. Retrieved from https://www.preventionweb.net/english/hyogo/gar/2015/en/bgdocs/CIMNE-INGENIAR,%202014a.pdf

CIMNE, I., ITEC. (2017). *Metodología de modelación probabilista de riesgos naturales* (technical report No. ERN-CAPRA-T1-3). CAPRA- Probabilistic Risk Assessment Initiative. Retrieved from https://ecapra.org/sites/default/files/documents/ERN-CAPRA-R6-T1-3%20-%20Modelos%20de%20Evaluaci%C3%B3n%20de%20Amenazas.pdf

Coles, S. (2001). *An introduction to statistical modeling of extreme values*. Springer London. http://doi.org/10.1007/978-1-4471-3675-0

Coles, S. (2003). Extreme values in finance, telecommunications, and the environment. In B. F. inkenstädt & H. Rootzén (Eds.), (pp. 79–100). Chapman; Hall/CRC. Retrieved from https://www.amazon.com/Telecommunications-Environment-Monographs-Statistics-Probability/dp/1584884118?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1584884118

Comarazamy, D. (2005). *Disaster mitigation in health facilities. Wind effects. Structural issues.* Pan American Health Organization. Retrieved from http://www.disaster-info.net/viento/english/guiones/structural.pdf

Council, N. R. (1994). Hurricane Hugo, Puerto Rico, the Virgin Islands, and Charleston, South Carolina, September 17-22, 1989. In (pp. 247–257). Washington, DC: National Academies Press. http://doi.org/10.17226/1993

C. S. Durst, B. A., O. B.E. (1960). Wind speeds over short periods of time. *The Meteorological Magazine*, *89*(1056), 181–187. Retrieved from https://www.depts.ttu.edu/nwi/Pubs/ReportsJournals/ReportsJournals/Windspeeds.pdf

Davison, A. C., & Smith, R. L. (1990). Models for exceedances over high thresholds. *Journal of the Royal Statistical Society. Series B (Methodological)*, *52*(3), 393–442. Retrieved from http://www.jstor.org/stable/2345667

Deville, Y., & IRSN. (2016). *Renext: Renewal method for extreme values extrapolation.* Retrieved from https://CRAN.R-project.org/package=Renext

Fisher, R. A., & Tippett, L. H. C. (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society*, *24*(2), 180–190. http://doi.org/10.1017/s0305004100015681

Gilleland, E. (2019). *ExtRemes: Extreme value analysis.* Retrieved from https://CRAN.R-project.org/package=extRemes

Gnedenko, B. (1943). Sur la distribution limite du terme maximum dUne serie aleatoire. *The Annals of Mathematics*, *44*(3), 423. http://doi.org/10.2307/1968974

Gräler, B., Pebesma, E., & Heuvelink, G. (2016). Spatio-temporal interpolation using gstat. *The R Journal*, *8*(1), 204–218. Retrieved from https://journal.r-project.org/archive/2016/RJ-2016-014/index.html

Harris, J. W., & Stocker, H. (1998). Maximum likelihood method. In *Handbook of mathematics and computational science* (p. 824). Springer-Verlag.

Hosking, J. R. M., & Wallis, J. R. (1997). *Regional frequency analysis.* Cambridge University Press. http://doi.org/10.1017/cbo9780511529443

IDEAM. (1999, June). Aeronautical information. Annual wind regime. Web Page. Retrieved from http://bart.ideam.gov.co/cliciu/rosas/viento.htm

IDEAM. (2005). *Protocolo toma de datos de campo y emplazamiento de estaciones meteorológicas.*

Janet E. Heffernan with R port, O. S. functions written by, & Alec G. Stephenson., R. documentation provided by. (2018). *Ismev: An introduction to statistical modeling of extreme values.* Retrieved from https://CRAN.R-project.org/package=ismev

Jenkinson, A. F. (1955). The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Quarterly Journal of the Royal Meteorological Society*,

81(348), 158–171.

Johnson, L., Kotz, S., & Balakrishnan, N. (1995). Limiting distributions of extremes. *Continuous Univariate Distributions*, *2*.

Kubler, J. (1994). *Computational Statistics &Amp; Data Analysis*, *18*(4), 473–474. Retrieved from https://EconPapers.repec.org/RePEc:eee:csdana:v:18:y:1994:i:4:p:473-474

Lettau, H. (1969). Note on aerodynamic roughness-parameter estimation on the basis of roughness-element description. *Journal of Applied Meteorology*, *8*(5), 828–832. http://doi.org/10.1175/1520-0450(1969)008%3C0828:NOARPE%3E2.0.CO;2

Masters, F. J., Vickery, P. J., Bacon, P., & Rappaport, E. N. (2010). Toward objective, standardized intensity estimates from surface wind speed observations. *Bulletin of the American Meteorological Society*, *91*(12), 1665–1682. http://doi.org/10.1175/2010bams2942.1

Mises, R. von. (1954). La distribution de la plus grande de n valeurs. In (ed.), selected papers (vol. II, pp. 271-294). Providence, ri. *American Mathematical Society*.

Müller, K., & Wickham, H. (2019). *Tibble: Simple data frames*. Retrieved from https://CRAN.R-project.org/package=tibble

NIST. (2012, February). Standardized extreme wind speed database for the United States. Web Page. Retrieved from https://www.itl.nist.gov/div898/winds/NIST_TN/nist_tn.htm

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, *10*(1), 439–446. http://doi.org/10.32614/RJ-2018-009

Pebesma, E. (2019a). *Sf: Simple features for r*. Retrieved from https://CRAN.R-project.org/package=sf

Pebesma, E. (2019b). *Stars: Spatiotemporal arrays, raster and vector data cubes*.

Pebesma, E., & Graeler, B. (2019). *Gstat: Spatial and spatio-temporal geostatistical modelling, prediction and simulation*. Retrieved from https://CRAN.R-project.org/package=gstat

Pebesma, E. J. (2004). Multivariable geostatistics in S: The gstat package. *Computers & Geosciences*, *30*, 683–691.

Pfaff, B., & McNeil, A. (2018). *Evir: Extreme values in r*. Retrieved from https://CRAN.R-project.org/package=evir

Pickands, J. (1971). The two-dimensional poisson process and extremal processes. *Journal of Applied Probability*, *8*(4), 745–756. http://doi.org/10.2307/3212238

Pickands III, J., & others. (1975). Statistical inference using extreme order statistics. *The Annals of Statistics*, *3*(1), 119–131.

Pintar, A. L., Simiu, E., Lombardo, F. T., & Levitan, M. L. (2015). *Simple guide for evaluating and expressing the uncertainty of NIST MeasuremenMaps of non-hurricane non-tornadic wind speeds with specified mean recurrence intervals for the contiguous united states using a two-dimensional poisson process extreme value model and local regressiont results*. National Institute of Standards; Technology.

Rezapour, M., & Baldock, T. E. (2014). Classification of hurricane hazards: The importance of rainfall. *Weather and Forecasting*, *29*(6), 1319–1331. http://doi.org/10.1175/waf-d-14-00014.1

Roberts, S. (2012). *Wind Wizard: Alan G. Davenport and the Art of Wind Engineering*. Princeton University Press. Retrieved from https://books.google.de/books?id=e2eYDwAAQBAJ

Royero, J. C. O. (2011). Exposure of the colombian caribbean coast, including san andrés island, to tropical storms and hurricanes, 19002010. *Natural Hazards*, *61*(2), 815–827. http://doi.org/10.1007/s11069-011-0069-1

Simiu, E., & Scanlan, R. H. (1996). *Wind effects on structures : Fundamentals and applications to design* (3rd ed.). New York : John Wiley. Retrieved from http://lib.ugent.be/catalog/rug01:001267836

Smith, R. L. (1985). Maximum likelihood estimation in a class of nonregular cases. *Biometrika*, *72*(1), 67–90.

Smith, R. L. (1989). Extreme value analysis of environmental time series: An application to trend detection in ground-level ozone. *Statistical Science*, *4*(4), 367–377. http://doi.org/10.1214/ss/1177012400

Smith, R. L. (2004). Extreme values in finance, telecommunications, and the environment (chapman & hall/crc monographs on statistics and applied probability). In B. F. inkenstädt & H. Rootzén (Eds.), (pp. 1–78). Chapman; Hall/CRC. Retrieved from https://www.amazon.com/Telecommunications-Environment-Monographs-Statistics-Probability/dp/1584884118?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1584884118

Stephenson, A. G. (2002). Evd: Extreme value distributions. *R News*, *2*(2), 0. Retrieved from https://CRAN.R-project.org/doc/Rnews/

Triana, J. D. S. (2019, September). Contrato no. 01-03-2019 ais-jdst. Actividades profesionales necesarias para actualizar el mapa de velocidades básicas de viento para el documento ais 100-19. PDF.

Vivienda, M. de. (2010). *Reglamento Colombiano de Construccción Sismo Resistente - NSR-10*. Carrera 20 # 84-14, oficina 502, Bogotá.: Asociación Colombiana de Ingeniería Sísmica. Comisión Asesora Permanente.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10). http://doi.org/10.18637/jss.v059.i10

Wickham, H. (2019). *Tidyverse: Easily install and load the 'tidyverse'*. Retrieved from `https://CRAN.R-project.org/package=tidyverse`

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. `http://doi.org/10.21105/joss.01686`

Wickham, H., François, R., Henry, L., & Müller, K. (2019). *Dplyr: A grammar of data manipulation*. Retrieved from `https://CRAN.R-project.org/package=dplyr`

Wuertz, D., Setz, T., & Chalabi, Y. (2017). *FExtremes: Rmetrics - modelling extreme events in finance*. Retrieved from `https://CRAN.R-project.org/package=fExtremes`

Appendix A

Research R Code - Digital Files

Table A.1: Research R Code. ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password).

Folder Tree - Ftp Links	Description
code	Folder with R code. ALL CODE CREATED BY DR. ADAM PINTAR IS NOT PUBLISHED.
|-pot_pp	Folder with POT-PP R code. Based in Dr Adam Pintar code (respected copyright).
| |-function_lib.r	POT-PP Functions. Author of de-clustering and thresholding functions is Dr Adam Pintar.
| |-plot_nt.r	Plot non-thunderstorm graphics.
| |-plot_t.r	Plot thunderstorm graphics.
| |-plot_t_nt.r	Plot graphics with thunderstorm and non-thunderstorm data, in simultaneous.
| |-stats_graphs_dnt.r	Statistics and graphics for non-thunderstorm de-clustered data.
| |-stats_graphs_dt.r	Statistics and graphics for thunderstorm de-clustered data.
| |-stats_raw_data.r	Statistics for raw data.
| |-stats_raw_data_nt.r	Statistics for non-thunderstorm raw data.
| |-stats_raw_data_t.r	Statistics for thunderstorm raw data.
| |-tnt_csv_1perday.r	Create CSV (thunderstorm and non-thunderstorm) with one data (the maximum) per day.
| |-era5	Folder with specific code for ERA5 data.
| | |-pot_pp_era5.r	POT-PP for ERA5 data. Based in Dr Adam Pintar code.
| | |-maps	Folder with specific code to calculate return levels and plot maps for ERA5 data.
| | | |-return_levels.r	Calculate return levels for ERA5 data.
| | | |-plot_maps.r	Join return levels to cells and plot ERA5 maps.
| |-isd	Folder with specific code for ISD data.
| | |-pot_pp_isd.r	POT-PP for ISD data. Based in Dr Adam Pintar code.
| | |-maps	Folder with code to calculate return levels, do spatial interpolation, and plot maps. ISD data.
| | | |-rl_10_nh.r	Calculate return levels and do spatial interpolation. MRI 10, non-hurricane data.
| | | |-rl_20_nh.r	Calculate return levels and do spatial interpolation. MRI 20, non-hurricane data.
| | | |-rl_50_nh.r	Calculate return levels and do spatial interpolation. MRI 50, non-hurricane data.
| | | |-rl_100_nh.r	Calculate return levels and do spatial interpolation. MRI 100, non-hurricane data.
| | | |-rl_250_nh.r	Calculate return levels and do spatial interpolation. MRI 250, non-hurricane data.
| | | |-rl_500_nh.r	Calculate return levels and do spatial interpolation. MRI 500, non-hurricane data.
| | | |-rl_700_nh.r	Calculate return levels and do spatial interpolation. MRI 700, non-hurricane data.
| | | |-rl_1000_nh.r	Calculate return levels and do spatial interpolation. MRI 1000, non-hurricane data.
| | | |-rl_1700_nh.r	Calculate return levels and do spatial interpolation. MRI 1700, non-hurricane data.
| | | |-rl_3000_nh.r	Calculate return levels and do spatial interpolation. MRI 3000, non-hurricane data.
| | | |-rl_7000_nh.r	Calculate return levels and do spatial interpolation. MRI 7000, non-hurricane data.
| | | |-rl_combined.r	Integrate return levels from hurricane and non-hurricane data.
| | | |-plot_maps.r	Plot ISD maps.
|-downscaling	Folder with code to compare all data sources, looking for downscaling support.
| |-qualitydata	Folder with code to compare using quality data from IDEAM (variable VV_AUT_2).
| | |-VV_AUT_2_1.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (1).
| | |-VV_AUT_2_2.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (2).
| | |-VV_AUT_2_3.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (3).
| |-nonqualitydata	Folder with code to compare using non-quality data from IDEAM (variable VV_AUT_10).
| | |-VV_AUT_10.r	All stations from ISD or IDEAM that intersects one ERA5 cell are compared.

Appendix B

Results - Digital Files

Table B.1: Results. Digital Files in FTP site ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password).

Folder Tree - Ftp Links	Description
downscaling	Downscaling Support
|-qualitydata	Quality data comparison (graphics in PDF)
|-nonqualitydata	Non quality data comparison (graphics in PDF)
|-ideam_stations.csv	IDEAM Stations
pot_pp	POT-PP input and output files
|-era5	ERA5 files
| |-FittedModel_*.pdf	ERA5 POT-PP output graphics. See Table B.4.
| |-fitted_model_result.xlsx	Return levels ERA5 (all stations). See Table B.5.
| |-raw_data_station_*_fitted.xlsx	ERA5 POT-PP output parameters by station. See Table B.2.
| |-raw_data_station_*_statistics.xlsx	ERA5 POT-PP time (year, month, week) statistics by station. See Table B.3.
| |-maps	ERA5 raster and vector output data
| | |-era5grid_left_right.*	ERA5 stations shapefile (IDs from left to right, then down)
| | |-era5grid_left_right_pol.*	ERA5 cells shapefile (IDs from left to right, then down)
| | |-era5grid_up_down.*	ERA5 stations shapefile (IDs from top to down, then right)
| | |-era5grid_up_down_pol.*	ERA5 cells shapefile (IDs from top to down, then right)
| | |-rl4326_points_nh_combined.*	ERA5 stations shapefile with all return levels
| | |-combined	ERA5 final wind maps (non-hurricanes + hurricanes). See Table B.6.
| | |-nonhurricanes	ERA5 POT-PP non-hurricane wind maps. See Table B.6.
|-isd	ISD files
| |-01 estaciones - 76 ok isd.txt	ISD list of used stations
| |-01 estaciones - isd - error.txt	One ISD station not working
| |-FittedModel_*.pdf	ISD POT-PP output graphics. See Table B.4.
| |-fitted_model_result.xlsx	Return levels ISD (all stations). See Table B.5.
| |-isd_stations.xlsx	ISD stations
| |-raw_data_station_*_fitted.xlsx	ISD POT-PP output parameters by station. See Table B.2.
| |-raw_data_station_*_statistics.xlsx	ISD POT-PP time (year, month, week) statistics by station. See Table B.3.
| |-maps	ISD raster and vector output data
| | |-rl_nh_h_combined_allcells4326.*	ISD stations shapefile with all return levels
| | |-combined	ISD final wind maps (non-hurricanes + hurricanes). See Table B.7.
| | |-nonhurricanes	ISD POT-PP non-hurricane wind maps. See Table B.7.
| |-raw_data	ISD non-thunderstorm time series (standardized)
|-correction_factors_isd_ideam.xlsx	Correction factors for standardization (ISD and IDEAM)
final_presentation.pdf	Slides for final thesis defense
final_document.pdf	Thesis final report
outfile_nc4c_zip9.nc	ERA5 data. Variable fg10 (3-seconds wind gust)

Table B.2: Content of the output Excel Book 'raw_data_station_*_fitted.xlsx', where '*' is replaced by the Station ID. One file by station.

Excel Sheet Name	Description	Important
nt_evd-fgev_fGumbel	Non thunderstorm. Fitting Gumbel using evd::fgev	Do not use
nt_bbmle-mle2	Non thunderstorm. Fitting Gumbel using bbmle::mle2	Do not use
nt_nll-optim	Non thunderstorm. Fitting Gumbel using negative likelihood and stats::optim	Do not use
nt_fitdistrplus-fitdist	Non thunderstorm. Fitting Gumbel using fitdistrplus::fitdist	Do not use
nt_extRemes	Non thunderstorm. Calculation of return levels POT-GPD, using extRemes::fevd	Do not use
nt_distLquantile_quant	Non thunderstorm. Calculation of return levels and RMSE (POT-GPD and EVDs), using extremeStat::distLquantile	To compare with POT-PP
nt_distLquantile_parameters	Non thunderstorm. Calculation of fitting parameters POT-GPD and EVD, using extremeStat::distLquantile	To compare with POT-PP
nt_distLextreme_returnlev	Non thunderstorm. Calculation of return levels POT-GPD and EVD, using extremeStat::distLextreme	To compare with POT-PP
nt_distLextreme_parameter	Non thunderstorm. Calculation of fitting parameters POT-GPD and EVD, using extremeStat::distLextreme	To compare with POT-PP
nt_POT-GPD-Equivalent	Non thunderstorm. For POT-PP and using POT-GPD equivalent. Calculation of Goodness of Fit and RMSE	Use as Goodness of Fit of POT-PP

Table B.3: Content of the output Excel Book 'raw_data_station_*_statistics.xlsx', where '*' is replaced by the Station ID. One file by station.

Excel Sheet Name	Description
all_years	Raw data time series statistics by year
all_months	Raw data time series statistics by month
all_weeks	Raw data time series statistics by week
all_gaps30days	Raw data gaps of 30 days of more
nt_years	Non-thunderstorm time series statistics by year
nt_months	Non-thunderstorm time series statistics by month
nt_weeks	Non-thunderstorm time series statistics by week
nt_gaps30days	Non-thunderstorm gaps of 30 days of more
IMP.VALS	Main statistics of dataset after de-clustering and thresholding
declu_nt_years	Non-thunderstorm time series statistics by year, after de-clustering and thresholding
declu_nt_months	Non-thunderstorm time series statistics by month, after de-clustering and thresholding
declu_nt_weeks	Non-thunderstorm time series statistics by week, after de-clustering and thresholding
declu_nt_gaps30days	Non-thunderstorm gaps of 30 days of more, after de-clustering and thresholding

Table B.4: Content of the output graphics PDF file 'Fitted-Model_*.pdf', where '*' is replaced by the Station ID. One file by station.

Graphic	Description
Page 1	Time Series Plot for Raw Data
Page 2	Time Series Plot for Non-Thunderstorm ('nt')
Page 3	Log-Likelihood(Gumbel) - Optim (nll-optim)
Page 4	De-clustered - Non-Thunderstorm - fitdistrplus-fitdist(gumbel)
Page 5	De-clustered Non-Thunderstorm ('nt') Time Series
Page 6	W-Statistic Plot for best pair of thresholds

Page 7	De-clustered - Non-Thunderstorm - Density Function from Intensity Function of Poisson Process
Page 8	De-clustered - Non-Thunderstorm - POT-GPD Equivalent
Page 9	De-clustered - Non-Thuderstorm - Cumulative Distribution Function from Intensity Function of Poisson Process
Page 10	De-clustered - Non-Thunderstorms - Hazard Curve. Poisson Process Intensity Function
Page 11	De-clustered - Non-Thunderstorms - Hazard Curve. Gumbel like tail Intensity Function of Poisson Process
Page 12	Non-Thunderstorms. Gumbel Density Function, but using parameters of Poisson Process
Page 13	Non-Thunderstorms. Gumbel Cumulative Distribution, but using parameters of Poisson Process
Page 14	De-clustered Non-Thunderstorm. Fitted Gumbel density function using parameters of Poisson Process
Page 15	De-clustered - Non-Thunderstorms - Hazard Curve. Gumbel Quantile Function using parameters of Poisson Process

Table B.5: Content of the output Excel Book 'fitted_model_result.xlsx' (sheet pp_pintar). One file by dataset (ISD, ERA5).

Column ID	Columns Name	Important	Description
1	id		Consecutive Row ID
2	t_thresh	Not Available	Threshold for thunderstorm data
3	t_mu_location	Not Available	Location for thunderstorm data
4	t_psi_scale	Not Available	Scale for thunderstorm data
5	nt_thresh		Threshold for non-thunderstorm data
6	nt_mu_location		Location for non-thunderstorm data
7	nt_psi_scale		Scale for non-thunderstorm data
8	distance_w		Minimum W distance to choose best threshold pairs
9	station		Station ID
10 to 20	t_MRI_poissonprocessintfunc	Not Available	Thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using Poisson Process Intensity Function
21 to 31	t_MRI_gumbeltailintfunc	Not Available	Thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using PP Gumbel Tail Intensity Function
32 to 42	t_MRI_gumbelquantilefunc	Not Available	Thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using PP Gumbel Quantile Function
43 to 53	nt_MRI_poissonprocessintfunc	Used to create maps!	Non-thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using Poisson Process Intensity Function
54 to 64	nt_MRI_gumbeltailintfunc	Do not use	Non-thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using PP Gumbel Tail Intensity Function
65 to 75	nt_MRI_gumbelquantilefunc	Do not use	Non-thunderstorm Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using PP Gumbel Quantile Function
76 to 86	tnt_MRI_poissonprocessintfunc	Not Available	Combined (t and nt) Return levels for MRIs (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, 7000), using Poisson Process Intensity Function

Table B.6: ERA5 Output Maps

File	Description
rl_nonhurricanes_4326_10_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 10 years
rl_nonhurricanes_4326_20_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 20 years
rl_nonhurricanes_4326_50_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 50 years
rl_nonhurricanes_4326_100_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 100 years
rl_nonhurricanes_4326_250_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 250 years
rl_nonhurricanes_4326_500_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 500 years

rl_nonhurricanes_4326_700_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 700 years
rl_nonhurricanes_4326_1000_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 1000 years
rl_nonhurricanes_4326_1700_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 1700 years
rl_nonhurricanes_4326_3000_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 3000 years
rl_nonhurricanes_4326_7000_st.tif	ERA5 POT-PP non-hurricane wind map. MRI 7000 years
rl_combined_4326_10_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 10 years
rl_combined_4326_20_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 210 years
rl_combined_4326_50_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 50 years
rl_combined_4326_100_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 100 years
rl_combined_4326_250_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 250 years
rl_combined_4326_500_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 500 years
rl_combined_4326_700_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 700 years
rl_combined_4326_1000_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 1000 years
rl_combined_4326_1700_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 1700 years
rl_combined_4326_3000_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 3000 years
rl_combined_4326_7000_st.tif	ERA5 final wind map (hurricane + non-hurricane). MRI 7000 years

Table B.7: ISD Output Maps

File	Description
nh_10.tif	ISD POT-PP non-hurricane wind map. MRI 10 years
nh_20.tif	ISD POT-PP non-hurricane wind map. MRI 20 years
nh_50.tif	ISD POT-PP non-hurricane wind map. MRI 50 years
nh_100.tif	ISD POT-PP non-hurricane wind map. MRI 100 years
nh_250.tif	ISD POT-PP non-hurricane wind map. MRI 250 years
nh_500.tif	ISD POT-PP non-hurricane wind map. MRI 500 years
nh_700.tif	ISD POT-PP non-hurricane wind map. MRI 700 years
nh_1000.tif	ISD POT-PP non-hurricane wind map. MRI 1000 years
nh_1700.tif	ISD POT-PP non-hurricane wind map. MRI 1700 years
nh_3000.tif	ISD POT-PP non-hurricane wind map. MRI 3000 years
nh_7000.tif	ISD POT-PP non-hurricane wind map. MRI 7000 years
isd_combined_4326_10_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 10 years
isd_combined_4326_20_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 210 years
isd_combined_4326_50_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 50 years
isd_combined_4326_100_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 100 years
isd_combined_4326_250_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 250 years
isd_combined_4326_500_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 500 years
isd_combined_4326_700_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 700 years
isd_combined_4326_1000_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 1000 years
isd_combined_4326_1700_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 1700 years
isd_combined_4326_3000_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 3000 years
isd_combined_4326_7000_st.tif	ISD final wind map (hurricane + non-hurricane). MRI 7000 years

Appendix C

ERA5 Data Download

The European Center for Medium-Range Weather Forecasts - ECMWF had implemented the Climate Data Storage - CDS `https://cds.climate.copernicus.eu/`, where all its datasets can be downloaded, however there is a straightforward way to get ERA5 data through Python library CDSAPI. Before to use CDSAPI, it is necessary to research about names and meanings of ERA5 variables using the official *data documentation* web page `https://confluence.ecmwf.int/display/CKB/ERA5%3A+data+documentation`, or the *parameter database* `https://apps.ecmwf.int/codes/grib/param-db`, that includes all ECWMF data sources, not only ERA5.

Next block of code shows the use of CDSAPI (Python) to download ERA5 variable *10fg - 10 meters wind gust*, from 1979 to 1991 for Colombia area. The most important keywords there, allow to define ERA5 data source and product type, variable name *'variable'*, format NetCDF *'format'*, area of interest *'area'*, using WGS88 coordinates in the format *north, west, south, east*, cell size *'grid'* in decimal degrees, and all the keywords related to time *'year'*, *'month'*, *'day'*, *'time'*.

```
import cdsapi
c = cdsapi.Client()
c.retrieve('reanalysis-era5-single-levels',{
 'product_type':'reanalysis',
 'format':'netcdf',
 'variable':'10m_wind_gust_since_previous_post_processing',
 'year':['1979','1980','1981','1982','1983','1984','1985','1986','1987','1988','1989','1990','1991'],
 'month':['01','02','03','04','05','06','07','08','09','10','11','12'],
 'time':['00:00','01:00','02:00','03:00','04:00','05:00','06:00','07:00','08:00','09:00','10:00','11:00',
        '12:00','13:00','14:00','15:00','16:00','17:00','18:00','19:00','20:00','21:00','22:00','23:00'],
 'day':['01','02','03','04','05','06','07','08','09','10','11','12','13','14','15','16','17','18',
        '19','20','21','22','23','24','25','26','27','28','29','30','31'],
 'area':[12.5, -79.1, -4.5, -66.8], # North, West, South, East.
 'grid':[0.25,0.25]},
'10fg_1979_1991_netcdf_.25x.25.nc')
```

Table C.1 shows all Python scripts used to download the variables *10fg* (10 meters wind gust) and *fsr* (forecast surface roughness) for the study area along the period 1979 to 2019. Last file in the table holds a summary of commands to manipulate NetCDF files.

After downloading separate NetCDF files, there are different tools available, to manipulate them, for instance, Climate Data Operators - CDO `https://code.mpimet.mpg.de/projects/cdo/`, NetCDF command line utilities `https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_working_with_netcdf_files.html`, and NetCDF operator - NCO `http://nco.sourceforge.net/`.

Table C.1: Python Code to Get ERA5 data. NetCDF Commands. ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password is needed).

Folder Tree - Ftp Links	Description
downloadingEra5	Python scripts to download ERA5 data
|-10fg	10fg: 10 meters wind gust ERA5 variable
| |-10fg_1979_1991_netCDF_0.25x0.25.py	Get 10fg variable from 1979 to 1991
| |-10fg_1992_2004_netCDF_0.25x0.25.py	Get 10fg variable from 1992 to 2004
| |-10fg_2005_2013_netCDF_0.25x0.25.py	Get 10fg variable from 2005 to 2013
| |-10fg_2014_2018_netCDF_0.25x0.25.py	Get 10fg variable from 2014 to 2018
| |-10fg_2019_oct_netCDF_0.25x0.25.py	Get 10fg variable from Jan to Sep of 2019
|-fsr	fsr: forecast surface roughness ERA5 variable
| |-fsr_1979_1991_netCDF_0.25x0.25.py	Get fsr variable from 1979 to 1991
| |-fsr_1992_2004_netCDF_0.25x0.25.py	Get fsr variable from 1992 to 2004
| |-fsr_2005_2013_netCDF_0.25x0.25.py	Get fsr variable from 2005 to 2013
| |-fsr_2014_2018_netCDF_0.25x0.25.py	Get fsr variable from 2014 to 2018
| |-fsr_2019_oct_netCDF_0.25x0.25.py	Get fsr variable from Jan to Oct of 1991
|-netcdfcommands.txt	Commands to join NETCDF files

Next CDO command, will join by time all NetCDF files inside a folder, where *-f* defines the file format, *-b* defines the data format, and *-z* defines the compression level (from 1 to 9). The resulting file name is *outfile_nc4c_zip9.nc*. Then with *cdo griddes* is possible to review output file content.

```
cdo -f nc4c -b F32 -z zip_9 mergetime *.nc outfile_nc4c_zip9.nc
cdo griddes outfile_nc4c_zip9.nc
```

Using NetCDF command line utilities, next commands will explore the content of a file, and change file format.

```
ncdump -h file.nc
nccopy -k 'nc4' -d 9 file.nc file_d9.nc
```

In Linux, next NCO commands will extract variable p0001 from file.nc to p0001.nc, rename variables p0001 to fsr in file p0001.nc, and view the result.

```
ncks -v p0001 file.nc p0001.nc
ncrename -v p0001,fsr p0001.nc
ncview p0001.nc
```

Appendix D

Database Storing

To optimize memory usage when running R code, some information related to IDEAM and ISD data sources, like stations and wind time series, were stored in **PostgreSQL database**, thanks to the possibility of lazy evaluation with **tidyverse** R package `dplyr`, see (Wickham, 2014), and (Wickham et al., 2019), which do not load the entire dataset in memory from the beginning, this is, *"it delays the actual operation until necessary and loads data onto R from the database only when we need it"* (Berhane, 2016). In parallel, it was used throughout the investigation, the **tidyverse** R package `tibble`, which allow lazy and surly data-frames in R, namely *"they do less ..., and complain more"* (Müller & Wickham, 2019). As a reference, there is an R package called `tidyverse`, which installs and loads all related packages (`ggplot2`, `tibble`, `tidyr`, `readr`, and `dplyr`), see (Wickham, 2019), and (Wickham, Averick, et al., 2019).

D.1 Loading Time Series from Text Files to PostgreSQL

R Package `dplyr` can be used to load data-frames created with text files, to tables inside databases, using for instance, `copy_to` function, see (Berhane, 2016), nonetheless an alternative procedure was done with *MS-DOS* commands, and *SQL* scripts using *pgsql* (terminal-based front-end to PostgreSQL). This procedure is described below for IDEAM data VV_AUT_10 - instantaneous wind velocity each ten (10) minutes, see Table 2.2.

There is one text file for each station time series in VV_AUT_10. There are a total of 204 stations, that is 204 files. The file name follows the format *VV_AUT_10@*.data*, where * is replaced by the station identifier, as can be seen below.

```
VV_AUT_10@31095030.data
VV_AUT_10@29065130.data
VV_AUT_10@29065140.data
...
```

Below is the partial content of a time series file (VV_AUT_10@31095030.data). It has two

columns (Fecha and Valor) separated by character |.

```
Fecha|Valor
2008-10-29 10:30:00|0.9
2008-10-29 10:40:00|1.4
2008-10-29 10:50:00|1.2
2008-10-29 11:00:00|1.5
...
```

1. Install PostgreSQL 10.5
2. Create database with next credentials

Table D.1: PostgreSQL Database Credentials

Credential	Value
dbname	winddata
host	localhost
pot	5432
user	user1
password	user1

```
-- Database: winddata
-- DROP DATABASE winddata;
CREATE DATABASE winddata
    WITH
    OWNER = user1
    ENCODING = 'UTF8'
    LC_COLLATE = 'English_United States.1252'
    LC_CTYPE = 'English_United States.1252'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

3. Inside PostgreSQL create *Temp* table with the following SQL code. This table will temporarily store the information of each station text file.

```
CREATE TABLE public."TEMP" (
  txtfecha VARCHAR(30),
  valor NUMERIC(6,2) ) WITH (oids = false);

ALTER TABLE public."TEMP"
  OWNER TO user1;
```

4. Inside PostgreSQL create *VV_AUT_10* table with following SQL code. This table will permanently store, one under the other, the content of all files with time series (204 files), identifying with an additional column the station ID of each record.

```
CREATE TABLE public."VV_AUT_10" (
  id BIGSERIAL,
  estationid BIGINT,
  txtfecha VARCHAR(30),
  valor NUMERIC(6,2),
  "timestamp" TIMESTAMP(0) WITHOUT TIME ZONE ) WITH (oids = false);

ALTER TABLE public."VV_AUT_10"
  OWNER TO user1;
```

5. Use next MS-DOS script that will take files content and load them into the database,

temporarily to *Temp*, then to *VV_AUT_10*.

```
REM 1: consecutive id
REM 2: file name
REM 3: station id
echo --(%1 of 204)>>"load_vv_aut_10.logdos"
echo --(%1 of 204)
echo Empty table TEMP
echo psql -L "load_vv_aut_10.logquery" -c "DELETE FROM public.\"TEMP\""
 "postgresql://user1:user1@127.0.0.1/winddata" >>"load_vv_aut_10.logdos"
psql -L "load_vv_aut_10.logquery" -c "DELETE FROM public.\"TEMP\""
 "postgresql://user1:user1@127.0.0.1/winddata" >>"load_vv_aut_10.logdos"
echo Copy from FILE to TEMP
echo psql -L "load_vv_aut_10.logquery" -c "COPY public.\"TEMP\" FROM 'modified\%2'
 (FORMAT CSV, DELIMITER'|', HEADER)" "postgresql://postgres:postgres@127.0.0.1/winddata"
 >>"load_vv_aut_10.logdos"
psql -L "load_vv_aut_10.logquery" -c "COPY public.\"TEMP\" FROM 'modified\%2'
 (FORMAT CSV, DELIMITER'|', HEADER)" "postgresql://postgres:postgres@127.0.0.1/winddata"
 >>"load_vv_aut_10.logdos"
echo Insert from TEMP into VV_AUT_10
echo psql -L "load_vv_aut_10.logquery" -c "INSERT INTO public.\"VV_AUT_10\"(estationid, txtfecha, valor,
 timestamp) SELECT %3, txtfecha, valor, TO_TIMESTAMP(txtfecha, 'YYYY-MM-DD HH24:MI:SS') FROM
 public.\"TEMP\"" "postgresql://postgres:postgres@127.0.0.1/winddata" >>"load_vv_aut_10.logdos"
psql -L "load_vv_aut_10.logquery" -c "INSERT INTO public.\"VV_AUT_10\"(estationid, txtfecha, valor,
 timestamp) SELECT %3, txtfecha, valor, TO_TIMESTAMP(txtfecha, 'YYYY-MM-DD HH24:MI:SS') FROM
 public.\"TEMP\"" "postgresql://postgres:postgres@127.0.0.1/winddata" >>"load_vv_aut_10.logdos"
echo HECHO! >>"load_vv_aut_10.logdos"
echo HECHO!
echo .. >>"load_vv_aut_10.logdos"
echo ..
echo .>>"load_vv_aut_10.logdos"
echo .
```

6. Copy previous MS-DOS commands into a text file called *load_vv_aut_10.bat*. To use this file three parameters need to be passed: 1) consecutive id, 2) file name, and 3) station id. All results (after call it) are stored in a log file named *load_vv_aut_10.logdos*. All SQL commands including the *pgsql* answer/result after code execution, are stored in a log file named *load_vv_aut_10.logquery*. The script, first, keep a consecutive record of the procedure inside the log file, second, empty the contents of the *Temp* table, then, copy from text file to table *Temp*, and finally, from *Temp* inserts into *VV_AUT_10*. In all script steps, detailed output logs are sent to files *load_vv_aut_10.logdos*, and *load_vv_aut_10.logquery*.

7. Call previous file using MS-DOS commands as shown below, where first parameters are a consecutive identifier, second is time series file name, and third one is the station identifier.

```
 call load_vv_aut_10.bat 1 VV_AUT_10_57025020.csv 57025020
 call load_vv_aut_10.bat 2 VV_AUT_10_31095030.csv 31095030
 call load_vv_aut_10.bat 3 VV_AUT_10_29065130.csv 29065130
 call load_vv_aut_10.bat 4 VV_AUT_10_29065140.csv 29065140
 ...
```

Consider following recommendations

- All files (204 files representing time series and load_vv_aut_10.bat) need to be stored in same Windows folder, for instance, 'c:\load_data'
- Using Windows Command Prompt, use *cd* command to locate the terminal inside the

folder

```
c:
cd load_data
```

- Use *call* command described in last step, once for each station time series file.
- Review content of log file load_vv_aut_10.logdos, to detect and correct possible loading errors.

Following ISD and IDEAM information were loaded to the database using previous procedure, see Table 2.2:

- VV_AUT_10
- ALL_VVMX_AUT_60
- ISD_LITE

The previously listed tables have **stacked** stations information, namely, with the same number of columns, the records of the first station occupy a certain number of rows, and the station identifier is repeated in each row, and below them, increasing the number of rows, the information of other stations is stored.

Stacked tables were transform to **unstacked** versions, namely, one column for each station identifier. Stacked table *ALL_VVMX_AUT_60* was transformed to unstacked table *IDEAM_VVMX_60*, using next chunk of SQL code.

```
CREATE table ideam_vvmx_60 AS
SELECT * FROM crosstab(
'SELECT DISTINCT
  timestamp,
  estacionid,
  valor
FROM
  public."ALL_VVMX_AUT_60"
ORDER BY timestamp, estacionid',
'SELECT DISTINCT
  estacionid
FROM
  public."ALL_VVMX_AUT_60"
ORDER BY estacionid')
AS ct(mydatetime timestamp, "11105020" real, "11135030" real, "12015100" real, "12015110" real,
 "13035501" real, "13085050" real, "14015080" real, "15015120" real, "15065180" real, "15065190" real,
 "15065501" real, "15075150" real, "15075501" real, "15079010" real, "15085050" real, "16015110" real,
 "16015120" real, "16015130" real, "16015501" real, "16055120" real, "17015010" real, "17035010" real,
 "21015030" real, "21015040" real, "21015050" real, "21015060" real, "21015070" real, "21055070" real,
 "21105030" real, "21115010" real, "21115170" real, "21115180" real, "21145080" real, "21185090" real,
 "21195160" real, "21195170" real, "21195190" real, "21205012" real, "21205710" real, "21205791" real,
 "21205910" real, "21205940" real, "21206280" real, "21206600" real, "21206790" real, "21206920" real,
 "21206930" real, "21206940" real, "21206950" real, "21206960" real, "21206980" real, "21206990" real,
 "21215150" real, "21215160" real, "21215170" real, "21215180" real, "21215190" real, "21235030" real,
 "21255160" real, "21255170" real, "22025040" real, "22075050" real, "23035030" real, "23065180" real,
 "23065190" real, "23085260" real, "23085270" real, "23105060" real, "23105070" real, "23125160" real,
 "23125170" real, "23195190" real, "23195230" real, "23195240" real, "23195502" real, "24015110" real,
 "24015380" real, "24025090" real, "24035360" real, "24035370" real, "24035380" real, "24035390" real,
 "24035410" real, "24035430" real, "24055070" real, "24055080" real, "25025000" real, "25025002" real,
 "25025030" real, "25025280" real, "25025340" real, "25025350" real, "25025360" real, "25025380" real,
 "26015010" real, "26015030" real, "26035090" real, "26035100" real, "26055100" real, "26055110" real,
 "26055120" real, "26075120" real, "26075150" real, "26085160" real, "26085170" real, "26095320" real,
 "26105240" real, "26105250" real, "26115090" real, "26125061" real, "26125290" real, "26125300" real,
 "26125710" real, "26135290" real, "26135300" real, "26135310" real, "26135320" real, "26135330" real,
```

```
    "26145090" real, "26155220" real, "26155230" real, "26155240" real, "26185030" real, "26185050" real,
    "26225060" real, "26255030" real, "27015280" real, "27015290" real, "27015300" real, "27015310" real,
    "27015320" real, "27015330" real, "28025120" real, "28025130" real, "28025502" real, "28035060" real,
    "29004520" real, "29015000" real, "29015040" real, "29035000" real, "29035200" real, "29045000" real,
    "29045150" real, "29045190" real, "29065000" real, "29065120" real, "29065130" real, "31095030" real,
    "32105080" real, "35025080" real, "35025090" real, "35025110" real, "35035100" real, "35035110" real,
    "35035130" real, "35075070" real, "35075080" real, "35085060" real, "35085070" real, "35085080" real,
    "35095120" real, "35095130" real, "35165000" real, "35185010" real, "35195060" real, "35215020" real,
    "35215030" real, "35225030" real, "35235040" real, "35235050" real, "36015020" real, "37015030" real,
    "44015060" real, "44015070" real, "44035040" real, "44035050" real, "46015030" real, "46035010" real,
    "47035030" real, "48015040" real, "48015050" real, "51025060" real, "51025080" real, "52015050" real,
    "52025080" real, "52025090" real, "52035040" real, "52045080" real, "52055150" real, "52055160" real,
    "52055170" real, "52055210" real, "52055220" real, "52055230" real, "53045040" real, "53075020" real,
    "54077210" real, "55015010" real, "56019010" real, "57025020" real, "1111500036" real,
    "2612500038" real, "3706500109" real)
```

Next list shows unstacked tables inside the database:

- IDEAM_VVMX_60 (correspond to ALL_VVMX_AUT_60)
- ISD_LITE_UNSTACK (correspond to ISD_LITE)

In addition, stations catalogs of IDEAM and ISD were loaded as tables. Main fields are *identifier*, *name*, *latitude* and *longitude*:

- IDEAM_ALL_STATIONS
- ISD_ALL_STATIONS

Next chunk of code shows the use of `dplyr` and `tibble` packages with PostgreSQL tables

```
    library(dplyr)
    con1 = src_postgres(dbname = "winddata", host = "localhost",
                        port = 5432, user = "user1", password = "user1")
    originalfields3 = c("id", "usaf", "station_name", "latitud", "longitud")
    originalfields3 = paste(originalfields3, collapse= ", ", sep = "")
    query3 = paste("select", originalfields3,
                   "from isd_all_stations where usaf_isd_dataua != ''", sep=" ")
    isd_stations = as_tibble(tbl(con1, sql(query3)))
    library(sf)
    isd_stations = st_as_sf(isd_stations, coords = c("longitud", "latitud"), crs = 4326)
```

Object *isd_stations* belongs to classes *sf*, *tbl_df*, *tbl*, and *data.frame*. Its description is shown below using the object name.

```
        class(isd_stations)
```

```
[1] "sf"         "tbl_df"     "tbl"        "data.frame"
```

```
        isd_stations
```

```
Simple feature collection with 100 features and 3 fields
geometry type:  POINT
dimension:      XY
bbox:           xmin: -81.711 ymin: -5.894 xmax: 0 ymax: 13.357
epsg (SRID):    4326
proj4string:    +proj=longlat +datum=WGS84 +no_defs
# A tibble: 100 x 4
      id usaf   station_name                geometry
   <int> <chr>  <chr>                     <POINT [°]>
 1 13848 690186 SAN JOSE DEL GUAVIA    (-72.633 2.567)
```

```
 2 14080 698689 MARANDUA AB            (-68.686 5.524)
 3 14081 698704 AFWA ASSIGNED          (-74.633 4.217)
 4 21913 789820 REINA BEATRIX INTL    (-70.015 12.501)
 5 21918 800000 BOGUS COLOMBIAN                (0 0)
 6 21919 800010 GUSTAVO ROJAS PINILLA (-81.711 12.584)
 7 21921 800020 EL EMBRUJO            (-81.358 13.357)
 8 21925 800090 SIMON BOLIVAR          (-74.231 11.12)
 9 21926 800220 RAFAEL NUNEZ          (-75.513 10.442)
10 21927 800280 ERNESTO CORTISSOZ      (-74.781 10.89)
# ... with 90 more rows
```

D.2 Database Backup

Next chunks of commands were used to create database backup.

D.2.1 Schema Backup

```
pg_dump.exe -f dump-C-E-o-s_winddata_2020_03.sql -C -E UTF8 -o -s -h localhost -U postgres winddata
```

D.2.2 Data Backup

Make a backup with option Fc (to load with pg_restore). Be aware of the use of option "copy" (not the option insert –D).

```
pg_dump.exe -Fc -a -E UTF8 -h localhost -U postgres -f dump-Fc-a-E_winddata_2020_03.dump winddata
```

D.2.3 Create Table of Contents (TOC) File

It is possible to load data from a backup file to the database, using a TOC text file containing the list of tables to be restored. The user can edit this file to decide which tables to restore or change the restoration order. TOC file created with next command, will have the load order by default (this command only works with backups created with the options Fc o Ft)

```
pg_restore.exe -l -f dump-Fc-a-E_winddata_2020_03.TOC dump-Fc-a-E_winddata_2020_03.dump
```

The content of default TOC file (dump-Fc-a-E_winddata_2020_03.TOC) as result from previous command is shown below

```
; Archive created at 2020-03-25 15:38:55
;     dbname: winddata
;     TOC Entries: 17
;     Compression: -1
;     Dump Version: 1.13-0
;     Format: CUSTOM
;     Integer: 4 bytes
;     Offset: 8 bytes
;     Dumped from database version: 10.5
;     Dumped by pg_dump version: 10.5
; Selected TOC Entries:
2865; 0 259855 TABLE DATA public ALL_VVMX_AUT_60 user1
2868; 0 259883 TABLE DATA public TEMP user1
2867; 0 259879 TABLE DATA public VV_AUT_10 user1
```

```
2875; 0 260005 TABLE DATA public ideam_all_stations user1
2876; 0 260092 TABLE DATA public ideam_vvmx_60 user1
2874; 0 259958 TABLE DATA public isd_all_stations user1
2871; 0 259898 TABLE DATA public isd_lite user1
2873; 0 259926 TABLE DATA public isd_lite_copy user1
2869; 0 259892 TABLE DATA public isd_lite_string user1
2877; 0 260096 TABLE DATA public isd_lite_unstack user1
2883; 0 0 SEQUENCE SET public ALL_VVMX_AUT_60_id_seq user1
2884; 0 0 SEQUENCE SET public ISD_LITE_ID_seq user1
2885; 0 0 SEQUENCE SET public VV_AUT_10_id_seq user1
2886; 0 0 SEQUENCE SET public isd_lite_new_id_seq user1
```

D.3 Database Restore

D.3.1 Schema Load

Before to restore the schema from file dump-C-E-o-s_winddata_2020_03.sql, it is possible to manually edit it, considering special user needs:

- Force to remove the database, if it is already created within PostgreSQL.
- Enable extensions for particular needs, for instance PostGis, DBLink, pgRouting, plpgsql

In addition, it is important to remember:

- Database name: winddata
- Database user: user1
- Log in the database as user *postgres*
- In next command, first *postgres* is the name of database, and second one, is the user name.

```
psql.exe -f dump-C-E-o-s_winddata_2020_03.sql postgres postgres
```

D.3.2 Load Data

To load data is necessary to use TOC file, which can be edited to change load order, or load only specific tables. To proceed with next command, please use the user *postgres*

```
pg_restore.exe -d winddata -Fc -a -L dump-Fc-a-E_winddata_2020_03.TOC -h localhost -U postgres
 -e dump-Fc-a-E_winddata_2020_03.dump
```

D.3.3 Restore Individual Tables

It is possible to restore/load individual tables using next command, where "-n" refers to schema and "-t" refers to table name. Next command will restore table *ideam_vvmx_60* from *public* schema (all tables belong to public schema)

```
pg_restore -d winddata -Fc -a -h localhost -U postgres -n public -t ideam_vvmx_60 -v
 -e dump-Fc-a-E_winddata_2020_03.dump
```

Appendix E

Thesis Document R Code

This appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` or `echo=FALSE, message=FALSE, warning=FALSE` chunks tag) to help with readability and/or setup.

In Chapter 2 - Data:

1. Install/Load Packages

```
# List of packages required for this analysis
  pkg <- c("dplyr", "sf", "ggplot2","rnaturalearth", "rnaturalearthdata", "ggspatial", "kableExtra", "ncdf4", "stars", "magick", "RcmdrMisc",
           "knitr", "ggrepel", "grid", "gridExtra", "cowplot", "xts", "bookdown", "lubridate", "devtools")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
  new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
  if (length(new.pkg))
    install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages (thesisdown will load all of the packages as well)
  library(thesisdown)
  library(dplyr)
  library(sf)
  library(ggplot2)
  library(rnaturalearth)
  library(rnaturalearthdata)
  library(ggspatial)
  library(knitr)
  library(kableExtra)
  library(ncdf4)
  library(stars)
  library(magick)
  library(RcmdrMisc)
  library(ggrepel)
  library(grid)
  library(gridExtra)
  library(cowplot)
  library(xts)
  library(lubridate)
```

2. Load IDEAM and ISD Stations

```
#Load IDEAM and ISD Stations
con1 = src_postgres(dbname = "winddata", host = "localhost", port = 5432, user = "user1", password = "user1")

#Get Ideam Stations Table
originalfields4 = c("objectid", "codigo1", "nombre", "latitud", "longitud", "categoria")
originalfields4 = paste(originalfields4, collapse= ", ", sep = "")
query4 = paste("select", originalfields4, "from ideam_all_stations", "where inpqrs2 = 'YES'", sep=" ")
ideam_stations = as_tibble(tbl(con1, sql(query4)))
Encoding(ideam_stations$categoria) <- "UTF-8"
Encoding(ideam_stations$nombre) <- "UTF-8"

originalfields3 = c("id", "usaf", "station_name", "latitud", "longitud")
```

```
originalfields3 = paste(originalfields3, collapse= ", ", sep = "")
query3 = paste("select", originalfields3, "from isd_all_stations where usaf_isd_dataua != ''", sep=" ")
isd_stations = as_tibble(tbl(con1, sql(query3)))
#Create simple features from Ideam Stations
ideam_stations = st_as_sf(ideam_stations, coords = c("longitud", "latitud"), crs = 4326)
#Create simple features from ISD stations
isd_stations = st_as_sf(isd_stations, coords = c("longitud", "latitud"), crs = 4326)
```

3. Plot IDEAM Stations

```
#Plot IDEAM Stations
theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))


colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

ggplot(data = world) +
  geom_sf(fill= "antiquewhite") +
  geom_text(data= world_points[venezuela,],aes(x=-67, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.2, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-67, y=-2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 13, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = ideam_stations, size=1, aes(shape=categoria, color=categoria), show.legend = "point") +
  scale_color_discrete(name = 'Category', labels = c("Agrometeorological", "Ordinary Climatic", "Main Climatic", "Mareographic",
   "Special Meteorological", "Main Synoptic")) +
  scale_shape_discrete(name = 'Category', labels = c("Agrometeorological", "Ordinary Climatic", "Main Climatic", "Mareographic",
   "Special Meteorological", "Main Synoptic")) +
  annotation_scale(location = "bl", width_hint = 0.5) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.05, "in"), pad_y = unit(0.05, "in"),
   style = north_arrow_fancy_orienteering) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("IDEAM Stations") +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.5), panel.background = element_rect(fill = "aliceblue"))
#Plote time series - one IDEAM Station
originalfields = c("21205791")
newfields = paste ("X", originalfields, sep="")
originalfields = paste('"', originalfields, '"', sep = "")
newfields = paste('"', newfields, '"', sep = "")

fiedls_query = paste(originalfields, "as", newfields, sep = " ")
fiedls_query = c(paste('"', "mydatetime", '"', sep = ""), fiedls_query)
fiedls_query = paste (fiedls_query, "", sep= "", collapse=", ")

wherestring = c("21205791")
wherestring = paste('"', wherestring, '"', sep = "")
wherestring = paste(wherestring, "IS NOT NULL", sep = " ")
wherestring = paste(wherestring, collapse = " OR " , sep = " ")
query = paste("select", fiedls_query, "from ideam_vvmx_60", "where", wherestring, sep=" ")

all_vvmx_aut_60 = as_tibble(tbl(con1, sql(query)))
timestamp_all_vvmx_aut_60 <- as.POSIXct(as_tibble(select(all_vvmx_aut_60, mydatetime))$mydatetime,format="%Y-%m-%d %H:%M:%S", tz="UTC")

statideam_xts = na.omit(xts(x=select(all_vvmx_aut_60, "X21205791"), order.by = timestamp_all_vvmx_aut_60))

plot.xts(statideam_xts, main = "Station ID: 21205791\nWind Velocity [m/s]", major.ticks="years", format.labels = "%b-%d\n%Y", legend.loc = "top",
  col="green", cex.main=0.3, cex=0.4, cex.axis=0.9, mar = c(2.5,1,0,1), oma = c(0,0,0,0))
```

4. Plot ISD Stations

```
#Plot ISD Stations
theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

ggplot(data = world) +
```

```
  geom_sf(fill= "antiquewhite") +
  geom_text(data= world_points[venezuela,],aes(x=-67, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.2, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-67, y=-2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 13, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = isd_stations, size=1, aes(color= "ISD Stations"), shape=2, show.legend = "point") +
  scale_color_manual(values = c("ISD Stations" = "black"), name="") +
  annotation_scale(location = "bl", width_hint = 0.5) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.05, "in"), pad_y = unit(0.05, "in"),
   style = north_arrow_fancy_orienteering) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("Longitude") +
  ylab("Latitude") +
  ggtitle("Integrated Surface Database - ISD") +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.5), panel.background = element_rect(fill = "aliceblue"))
#Plot - ISD Station
originalfields1 = c("802590")
newfields1 = paste ("X", originalfields1, sep="")
originalfields1 = paste('"', originalfields1, '"', sep = "")
newfields1 = paste('"', newfields1, '"', sep = "")
fiedls_query1 = paste(originalfields1, "as", newfields1, sep = " ")
fiedls_query1 = c(paste('"', "mydatetime", '"', sep = ""), fiedls_query1)
fiedls_query1 = paste (fiedls_query1, "", sep= "", collapse=", ")

wherestring1 = c("802590")
wherestring1 = paste('"', wherestring1, '"', sep = "")
wherestring1 = paste(wherestring1, "IS NOT NULL", sep = " ")
wherestring1 = paste(wherestring1, collapse = " OR " , sep = " ")
query1 = paste("select", fiedls_query1, "from isd_lite_unstack", "where", wherestring1, sep=" ")

isdlite = as_tibble(tbl(con1, sql(query1)))

timestamp_isdlite <- as.POSIXct(as_tibble(select(isdlite, mydatetime))$mydatetime,format="%Y-%m-%d %H:%M:%S", tz="UTC")

statisd_xts = na.omit(xts(x=select(isdlite, "X802590"), order.by = timestamp_isdlite))

plot.xts(statisd_xts, main = "Station ID: 802590\nWind Velocity [m/s]", major.ticks="years", format.labels = "%b-%d\n%Y", legend.loc = "top",
 col="green", cex.main=0.3, cex=0.4, cex.axis=0.9, mar = c(2.5,1,0,1), oma = c(0,0,0,0))
```

5. Load ERA5 Stations

```
#Load ERA5 NetCDF dataset - variable fg10
ncname <- "outfile_nc4c_zip9"
filename <- paste("./data/", ncname, ".nc", sep = "")
ncin <- nc_open(filename)
lon <- ncvar_get(ncin, "longitude")
nlon = dim(lon)
lat <- ncvar_get(ncin, "latitude")
nlat = dim(lat)
ntime <-  dim(ncvar_get(ncin, "time"))
variablename <- "fg10"
fg10.units <- ncatt_get(ncin, variablename, "units")
fg10.units
lonlat.unstack <- expand.grid(lon=as.numeric(lon), lat=as.numeric(lat))
#Create ERA5 centers (point with lat, lon, and value, as cell index)
era5colpoints = st_as_sf(lonlat.unstack, coords=1:2, crs=st_crs(4326))
era5colpoints$value = 1:(nlon*nlat)
#Define stars object to match with ERA5 bounding box.
#Cell centers of stars object, need to be same cell centers of ERA5
pointsbbox = st_bbox(era5colpoints)
cellsize = lonlat.unstack$lon[2]- lonlat.unstack$lon[1]
mybbox = st_bbox(c(pointsbbox$xmin - (cellsize/2), pointsbbox$xmax + (cellsize/2), pointsbbox$ymax + (cellsize/2), pointsbbox$ymin - (cellsize/2)),
 crs = st_crs(4326))
era5colraster.st = st_rasterize(era5colpoints, st_as_stars(mybbox, nx = nlon, ny = nlat, values = era5colpoints$value))
#Load ERA5 polygon vectors, representing cells of ERA5
file_era5_sf_pol = "./data/era5grid_left_right_pol.shp"
era5_4326_sf_pol = st_read(dsn=file_era5_sf_pol)
pts <- do.call(rbind, st_centroid(st_geometry(era5_4326_sf_pol)))
x = pts[,1]
y = pts[,2]
era5_4326_sf_pol$x = x
era5_4326_sf_pol$y = y
era5_4326_sf_pol_filter_corners = era5_4326_sf_pol %>% filter(DN %in% c(1, 49, 3333, 3381))
era5_4326_sf_pol_filter_corners_left = era5_4326_sf_pol %>% filter(DN %in% c(1, 3333))
era5_4326_sf_pol_filter_corners_right = era5_4326_sf_pol %>% filter(DN %in% c(49, 3381))
```

6. Plot ERA5 Stations

```
#Plot ERA5 Stations
theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))
```

```
colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

big = ggplot(data = world) +
  geom_sf(fill= "antiquewhite") +
  geom_sf(data=era5colpoints, size=0.1, aes(color = "Stations"), shape=".",  show.legend = "point")+
  scale_color_manual(values = c("Stations" = "black"), name="ERA5", guide = guide_legend(override.aes = list(fill= c(NA), linetype = c("blank"),
   shape = c("."))) ) +
  geom_sf(data = era5_4326_sf_pol_filter_corners, color = "black", aes(fill="Cells"), size=0.1, alpha=1, show.legend = "polygon") +
  scale_fill_manual(values = c("Cells" = NA), name="", guide = guide_legend(override.aes = list(fill = c(NA), shape = c(NA), size=0.1))) +
  geom_rect(mapping=aes(xmin=-79.252968100, xmax=-78.247031900, ymin=11.832846362, ymax=12.667153638), color="red", alpha=0, size=0.1) +
  geom_rect(mapping=aes(xmin=-67.752968100, xmax=-66.747031900, ymin=11.832846362, ymax=12.667153638), color="red", alpha=0, size=0.1) +
  geom_rect(mapping=aes(xmin=-79.258632089, xmax=-78.241367911, ymin=-4.671851259, ymax=-3.828148741), color="red", alpha=0, size=0.1) +
  geom_rect(mapping=aes(xmin=-67.758632089, xmax=-66.741367911, ymin=-4.671851259, ymax=-3.828148741), color="red", alpha=0, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.3, y=8.5, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.7, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.5, y=-1.5, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75.5, y=-5.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-66, y=-2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_text_repel(data = era5_4326_sf_pol_filter_corners_left, size=2, aes(x=x, y=y, label = DN), direction="y", segment.size=0.1,
   segment.color= "grey50", color="grey50", nudge_x=-1, hjust=1, box.padding=0.1) +
  geom_text_repel(data = era5_4326_sf_pol_filter_corners_right, size=2, aes(x=x, y=y, label = DN), direction="y", segment.size=0.1,
   segment.color= "grey50", color="grey50", nudge_x=+1, hjust=0, box.padding=0.1) +
  coord_sf(xlim = c(-81.1, -64.8), ylim = c(-5, 13), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("ERA5 Reanalysis - Forecast") +
  theme(plot.title = element_text(size=8)) +
  theme(axis.text.x= element_text(size=7)) +
  theme(axis.text.y= element_text(size=7)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(legend.title = element_text(size=8)) +
  theme(legend.text=element_text(size=8)) +
  theme(legend.key.size = unit(0.5,"line")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -10))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10)))

corner1lt = ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_sf(data = era5_4326_sf_pol, colour="black", fill=NA, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"),  size=0.1)+
  geom_sf_text(data = era5_4326_sf_pol, aes(label = DN), size=2) +
  coord_sf(xlim = c(-79.252968100, -78.247031900), ylim = c(11.832846362, 12.667153638), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("") +
  theme(panel.grid = element_blank()) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(axis.text.x = element_blank(), axis.text.y = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(plot.margin=grid::unit(c(0,0.2,0,0),"cm")) +
  theme(panel.border = element_rect(colour = "red"))+
  theme(axis.ticks.length=unit(0, "null")) +
  theme(axis.title.x=element_blank()) +
  theme(axis.title.y=element_blank()) +
  theme(plot.title = element_blank())

corner2rt = ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_sf(data = era5_4326_sf_pol, colour="black", fill=NA, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"),  size=0.1)+
  geom_sf_text(data = era5_4326_sf_pol, aes(label = DN), size=2) +
  coord_sf(xlim = c(-67.752968100, -66.747031900), ylim = c(11.832846362, 12.667153638), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("") +
  theme(panel.grid = element_blank()) +
```

```
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(axis.text.x = element_blank(), axis.text.y = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(plot.margin=grid::unit(c(0,0,0,0.2),"cm")) +
  theme(panel.border = element_rect(colour = "red"))+
  theme(axis.ticks.length=unit(0, "null")) +
  theme(axis.title.x=element_blank()) +
  theme(axis.title.y=element_blank()) +
  theme(plot.title = element_blank())

corner3lb = ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_sf(data = era5_4326_sf_pol, colour="black", fill=NA, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"),  size=0.1)+
  geom_sf_text(data = era5_4326_sf_pol, aes(label = DN), size=2) +
  coord_sf(xlim = c(-79.258632089, -78.241367911), ylim = c(-4.671851259, -3.828148741), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("") +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(axis.text.x = element_blank(), axis.text.y = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(plot.margin=grid::unit(c(0,0.2,0,0),"cm")) +
  theme(panel.border = element_rect(colour = "red"))+
  theme(axis.ticks.length=unit(0, "null")) +
  theme(axis.title.x=element_blank()) +
  theme(axis.title.y=element_blank()) +
  theme(plot.title = element_blank())

corner4rb = ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_sf(data = era5_4326_sf_pol, colour="black", fill=NA, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"),  size=0.1)+
  geom_sf_text(data = era5_4326_sf_pol, aes(label = DN), size=2) +
  coord_sf(xlim = c(-67.758632089, -66.741367911), ylim = c(-4.671851259, -3.828148741), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("") +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(axis.text.x = element_blank(), axis.text.y = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(plot.margin=grid::unit(c(0,0,0,0.2),"cm")) +
  theme(panel.border = element_rect(colour = "red"))+
  theme(axis.ticks.length=unit(0, "null")) +
  theme(axis.ticks.margin=unit(0, "null")) +
  theme(axis.title.x=element_blank()) +
  theme(axis.title.y=element_blank()) +
  theme(plot.title = element_blank())

grid.arrange(big, arrangeGrob(corner1lt, corner2rt, corner3lb, corner4rb), ncol=2, widths=c(2.2,1))
```

In Chapter 3 - Theoretical Framework:

1. Install/Load Packages

```
# List of packages required for this analysis
  pkg <- c("RcmdrMisc")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
  new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
  if (length(new.pkg))
    install.packages(new.pkg, repos = "http://cran.rstudio.com")
# Load packages (thesisdown will load all of the packages as well)
  library(RcmdrMisc)
  library(knitr)
```

2. Plot Gumbel PPF

```
par(mar=c(2.5,2.5,2,0))
par(oma=c(0,0,0,0))
par(mgp=c(1.5,0.5,0))
location = 100
scale = 40
.x <- seq(0, 300, length.out=1000)
pdfG <- function(x) {
  1/location *exp(-(x-location)/scale)*exp(-exp(-(x-location)/scale))
  }
.y = pdfG(.x)
plot(.x, .y, col="green", lty=4, xlab="Velocities Km/h", ylab="Density Function - Gumbel Distribution", cex.axis = 0.5, cex.lab= 0.6, cex.main=0.7,
 main=paste("Gumbel - Density Function Gumbel Distribution\n", "Location=", round(location,2), " Scale=", round(scale,2)), type="l", cex.sub=0.6)
par(mar=c(2.5,2.5,2,0))
par(oma=c(0,0,0,0))
par(mgp=c(1.5,0.5,0))
location = 100
scale = 40
.x <- seq(0, 300, length.out=1000)
dfG = dgumbel(.x, location=location, scale=scale)
plot(.x, dfG, col="red", lty=4, xlab="Velocities Km/h", ylab="Density Function - Gumbel Distribution", cex.axis = 0.5, cex.lab= 0.6, cex.main=0.7,
 main=paste("Gumbel - Density Function Gumbel Distribution\n", "Location=", round(location,2), " Scale=", round(scale,2)), type="l", cex.sub=0.6)
```

3. Plot Gumbel CDF

```
par(mar=c(2.5,2.5,2,0))
par(oma=c(0,0,0,0))
par(mgp=c(1.5,0.5,0))
location = 100
scale = 40
.x <- seq(0, 300, length.out=1000)
cdfG <- function(x) {
  exp(-exp(-(x-location)/scale))
  }
.y = cdfG(.x)
plot(.x, .y, col="green", lty=4, xlab="Velocities Km/h", ylab="Probability", cex.axis = 0.5, cex.lab= 0.6, cex.main=0.7, type="l",
 main=paste("Gumbel - Cumulative Distribution Function\n", "Location=", round(location,2), " Scale=", round(scale,2)), cex.sub=0.6)
```

4. Plot Gumbel PPF

```
par(mar=c(2.5,2.5,2,0))
par(oma=c(0,0,0,0))
par(mgp=c(1.5,0.5,0))
location = 100
scale = 40
.x <- seq(0, 1, length.out=1000)
ppfG <- function(x) {
  location - (scale*log(-log(x)))
  }
.y = ppfG(.x)
plot(.x, .y, col="green", lty=4, ylab="Velocities Km/h", xlab="Probability", cex.axis = 0.5, cex.lab= 0.6, cex.main=0.7, cex.sub=0.6,
 main=paste("Gumbel - Percent Point Function\n", "Location=", round(location,2), " Scale=", round(scale,2)), type="l")
```

5. Plot Gumbel HF

```
par(mar=c(2.5,2.5,2,0))
par(oma=c(0,0,0,0))
par(mgp=c(1.5,0.5,0))
location = 100
scale = 40
.x <- seq(0, 1500, length.out=1000)
hfG <- function(x) {
  (1/scale)*(exp(-(x-location)/scale))/(exp(exp(-(x-location)/scale))-1)
  }
.y = hfG(.x)
plot(.x, .y, col="green", lty=4, xlab="Velocities Km/h", ylab="Hazard", cex.axis = 0.5, cex.lab= 0.6, cex.main=0.7, cex.sub=0.6,
 main=paste("Gumbel - Hazard Function\n", "Location=", round(location,2), " Scale=", round(scale,2)), type="l", xlim=c(0,500))
```

6. Compound Exceedance Probability - Pn

```
par(mar=c(3,3,0,0))
par(oma=c(0,0,0,0))
par(mgp=c(2,1,0))
plot(1, type="n", xlab=expression(paste("Compound Probability ", P[n])), ylab="Exposure Time as a Multiple of MRI",
 xlim=c(0,1), ylim= c(0,2500), xaxt ="n", yaxt="n", bty="n", cex.lab=0.7)

y1 = c(0, 500,1000,1500,2000,2500)
text(y=y1,  x=par("usr")[1], labels = y1/500, srt = 0, pos = 2, xpd = TRUE, cex=0.8)
y1 = 500*0.69
text(y=y1,  x=par("usr")[1], labels = ".69", srt = 0, pos = 2, xpd = TRUE, cex=0.6)
y1 = 2250
```

```
text(y=y1,  x=par("usr")[1], labels = expression(paste("4",frac(1,2))), srt = 0, pos = 2, xpd = TRUE, cex=0.6)

npn <- function(x) (1-(1/500))^x #Event will not occur
n = seq(from=0, to=2500, by=1)
mynpn = npn(n)

lines(x=mynpn,y=n, col= "blue")

pn <- function(x) 1-(1-(1/500))^x #Event will occur
mypn= pn(n)
lines(x=mypn,y=n, col= "green")

text(x=c(0.01, 0.37,0.63, 0.99),  par("usr")[3], labels = c(".01",".37",".63",".99") , srt = 0, pos = 1, xpd = TRUE, cex=0.6)

axis(1, at= seq(from=0, to=1, by= 0.1), labels=seq(from=0, to=1, by= 0.1), tick=TRUE, col.axis="black", cex=0.8)

axis(2, at=c(0, 500,1000,1500,2000,2500),labels=FALSE, tick=TRUE, col.axis="black")
axis(2, at=c(345, 2250),labels=FALSE, tick=TRUE, col.axis="black", tck=-0.015)
axis(1, at=c(0.01,.37,.63,0.99),labels=FALSE, tick=TRUE, col.axis="black", tck=-0.015)

abline(v=c(0.01, 0.37,0.5, 0.63,0.99), lty="dotted")
abline(h=c(345,500, 2250), lty="dotted")
text(x=0.15,  y=1800, labels = "chance event\nwill not occur", cex=0.7)
text(x=0.85,  y=1800, labels = "chance event\nwill occur", cex=0.7)
```

In Chapter 4 - Methodology:

1. Combined Hazard Curve

```
plotit<-function(){

  par(mar=c(2,2,0,0))
  par(oma=c(0,0,0,0))
  par(bg=NA)
  plot(1, xlab='', ylab='', type='n', yaxt='n', xaxt='n', tck=0, xlim=c(0,200), ylim=c(0,0.05), bg = 'transparent', bty="n")
  arrows(0,0,0,0.05, length=0.04)
  arrows(0,0,200,0, length=0.04)
  text(x = par("usr")[2] - 5,  y = par("usr")[3] - 0.005, labels = expression(frac(1, N)), xpd = NA, srt = 0, cex = 0.7)
  text(x = par("usr")[1] - 6,  y = 0.05, labels = expression(Y[N]), xpd = NA, srt = 0, cex = 0.7)
  text(x = 50.2,  y = par("usr")[3] - 0.003, labels = "?", xpd = NA, srt = 0, cex = 0.7)
  text(x = 68.5,  y = par("usr")[3] - 0.003, labels = "0.02", xpd = NA, srt = 0, cex = 0.7)
  text(x = 100,  y = par("usr")[3] - 0.003, labels = "0.03", xpd = NA, srt = 0, cex = 0.7)
  text(x = par("usr")[1] - 10,  y = 0.015, labels = expression(paste("30 ", frac(Km, h))), xpd = NA, srt = 0, cex = 0.7)
  text(x = par("usr")[2] - 2,  y = 0.048, labels = "Combined", xpd = NA, srt = 0, pos = 2, cex = 0.6)
  text(x = par("usr")[2] - 2,  y = 0.031, labels = "Hurricanes", xpd = NA, pos = 2, srt = 0, cex = 0.6)
  text(x = par("usr")[2] - 2,  y = 0.021, labels = "Non-Hurricanes", xpd = NA, pos = 2, srt = 0, cex = 0.6)
  myexp = expression(paste(P[e], " = 1 - (1 -", P[nh], ") * (1 - ", P[h], ")"))
  text(x = par("usr")[1] + 60,  y = 0.049, labels = myexp, xpd = NA, srt = 0, cex = 0.7)
  text(x = par("usr")[1] + 60,  y = 0.045, labels = "? = 1- (1 - 0.03)(1-0.02)", xpd = NA, srt = 0, cex = 0.6)
  location = 65
  scale = 20
  .x <- seq(0, 1500, length.out=1000)
  hfG <- function(x) {
    (1/scale)*(exp(-(x-location)/scale))/(exp(exp(-(x-location)/scale))-1)
  }
  curve(hfG, add=T, col="red", lwd=1, lty=5)
  Arrows (x0=50.2, y0=0, x1=50.2, y1=(hfG(50.2)-0.003), arr.type="triangle", arr.width=0.04, lwd=0.1)

  location = 80
  scale = 30
  .x <- seq(0, 1500, length.out=1000)
  curve(hfG, add=T, col="red", lwd=1)
  Arrows (x0=68.5, y0=0, x1=68.5, y1=(hfG(68.5)-0.003), arr.type="triangle", arr.width=0.04, lwd=0.1)

  location = 100
  scale = 40
  .x <- seq(0, 1500, length.out=1000)
  curve(hfG, add=T, col="red", lwd=1)
  Arrows (x0=100, y0=0, x1=100, y1=(hfG(100)-0.003), arr.type="triangle", arr.width=0.04, lwd=0.1)
  Arrows (x0=100, y0=hfG(100), x1=7, y1=hfG(100) , arr.type="triangle", arr.width=0.04, lwd=0.1)
}

z.plot1<-function(){plotit()}
mydataframe = data.frame(v = c(10, 20, 30, "...", 350, "..."), Pe = c("...", "...", "?", "...", "...", "..."))
names(mydataframe) <- c(expression(Y[N]), expression(P[e]))
tt <- ttheme_default(base_size = 7, colhead=list(fg_params = list(parse=TRUE)))
tbl <- tableGrob(mydataframe, rows=NULL, theme=tt)
plot_grid(z.plot1, tbl, ncol = 2, rel_widths = c(4,1), labels=c("", "Combined Curve"), label_size = 7, hjust=-0.13)
```

In Chapter 5 - Results and Discussion:

1. Downscaling Support - Quality Data Comparison

```
#Downscaling Support - Sources Comparison
#
#Important Note for next line of code (seven lines bellow):
#
#Not run when Knitting. Run it only in RStudio before Knit process, that is why next line of code is out-commented (using #).
#This code generates some graphics in format RDS, used inside Results and Discussion chapter. It is not necessary to run again,
#unless current RDS files already generated, are not usable for changes in R packages versions (possible situation for future)
#Output of this code goes to the folder '.../index/data/' of the 'thesisdown' root folder.
#
#source('./code/comparing_sources_pqrs_20199050080932_VV_AUT.r')
#
#
#IDEAM VV_AUT_2 - Quality Data Comparison
#
con1 = src_postgres(dbname = "winddata", host = "localhost", port = 5432, user = "user1", password = "user1")

#Get Ideam Stations Table
originalfields4 = c("objectid", "codigo1", "nombre", "latitud", "longitud", "latitud", "longitud", "categoria")
newnames4 = c("objectid", "codigo1", "nombre", "latitud", "longitud", "y", "x", "categoria")
originalfields4 = paste(originalfields4, " as ", newnames4, sep="")
originalfields4 = paste(originalfields4, collapse= ", ", sep = "")
query4 = paste("select", originalfields4, "from ideam_all_stations", "where codigo1 IN (48015050, 52055230, 26125061, 26125710, 23085270, 27015330,
 16015501, 23195502, 13035501, 28025502, 15065180, 29045190)", sep=" ")
ideam_stations = as_tibble(tbl(con1, sql(query4)))
Encoding(ideam_stations$categoria) <- "UTF-8"
Encoding(ideam_stations$nombre) <- "UTF-8"
originalfields3 = c("id", "usaf", "station_name", "latitud", "longitud", "latitud", "longitud")
newnames3 = c("id", "usaf", "station_name", "latitud", "longitud", "y", "x")
originalfields3 = paste(originalfields3, " as ", newnames3, sep="")
originalfields3 = paste(originalfields3, collapse= ", ", sep = "")
query3 = paste("select", originalfields3, "from isd_all_stations where usaf IN ('803980', '803700', '802110', '802100', '801120', '801100',
 '800970', '800940', '800630', '800360', '800350', '800280')", sep=" ")

isd_stations = as_tibble(tbl(con1, sql(query3)))
ideam_stations = st_as_sf(ideam_stations, coords = c("longitud", "latitud"), crs = 4326)
isd_stations = st_as_sf(isd_stations, coords = c("longitud", "latitud"), crs = 4326)

ideam_stations3 <- ideam_stations %>% filter(codigo1 %in% c(15065180, 29045190, 28025502))
ideam_stationso <- ideam_stations %>% filter(codigo1 %in% c(48015050, 52055230, 26125061, 26125710, 23085270, 27015330, 16015501, 23195502,
 13035501))
isd_stations3 <- isd_stations %>% filter(usaf %in% c('800350', '800280', '800360'))
isd_stationso <- isd_stations %>% filter(usaf %in% c('803980', '803700', '802110', '802100', '801120', '801100', '800970', '800940', '800630'))


file_era5_st = "./data/era5grid_left_right.tif"
era5_4326_st = read_stars(file_era5_st)
era5_4326_st = setNames(era5_4326_st, "Station")

file_col_st = "./data/col2.tif"
col_4326_st = read_stars(file_col_st)
col_4326_st = setNames(col_4326_st, "col_4326_st")

file_era5_sf_point = "./data/era5grid_left_right.shp"
era5_4326_sf_point = st_read(dsn=file_era5_sf_point)

file_era5_sf_pol = "./data/era5grid_left_right_pol.shp"
era5_4326_sf_pol = st_read(dsn=file_era5_sf_pol)

file_col_sf_pol = "./data/COLOMBIA.shp"
col_4326_sf_pol = st_read(dsn=file_col_sf_pol)

img_stack_col=stack(file_col_st)
img_stack_col.crop = crop(img_stack_col, extent(col_4326_sf_pol))
img_stack_col.mask = mask(img_stack_col.crop, col_4326_sf_pol)

myPalette <- colorRampPalette(rev(brewer.pal(11, "YlGn")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(0, 255))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

k = era5_4326_sf_pol

big <-ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-68.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-3.7, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
```

```
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  geom_rect(mapping=aes(xmin=-73.7, xmax=-72.8, ymin=10.1, ymax=10.9), color="red", alpha=0, size=0.1) +
  geom_sf(data = ideam_stations, size=2, shape=23, color= "red", show.legend = "point") +
  geom_text_repel(data = ideam_stationso, size=2, aes(x=x, y=y, label = codigo1), direction="x", segment.size=0.1) +
  geom_text_repel(data = ideam_stations3, size=2, aes(x=x, y=y, label = codigo1), direction="x", segment.size=0.1, nudge_x=-0.5) +
  geom_sf(data = isd_stations, size=2, shape=3, color= "blue", show.legend = "point") +
  geom_text_repel(data = isd_stationso, size=2, aes(x=x, y=y, label = usaf), direction="y", segment.size=0.1) +
  geom_text_repel(data = isd_stations3, size=2, aes(x=x, y=y, label = usaf), direction="x", segment.size=0.1, nudge_x=0.5) +
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.2, "cm"), line_width = 0.5, text_cex = 0.5) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.05, "in"), pad_y = unit(0.05, "in"), height = unit(1, "cm"),
   width = unit(1, "cm"), style = north_arrow_fancy_orienteering) +
  coord_sf(xlim = c(-79.5, -66.5), ylim = c(-5.4, 12.3), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Quality Data Comparison\nTwelve matching stations from IDEAM and ISD") +
  theme(plot.title = element_text(size=8)) +
  theme(axis.text.x= element_text(size=7)) +
  theme(axis.text.y= element_text(size=7)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -10))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10)))

small <- ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1) +
  geom_sf(data = k, colour="black", fill=NA, size=0.1) +
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.5, y=9.2, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-79.2, y=-1, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=3, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 4) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 4) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"),  size=0.1)+
  geom_sf(data = ideam_stations, size=3, shape=23, color= "red", show.legend = "point") +
  geom_sf_text(data = ideam_stations, aes(label = codigo1), size=2, position = position_nudge(x = -0.11)) +
  geom_sf(data = isd_stations, size=3, shape=3, color= "blue", show.legend = "point") +
  geom_sf_text(data = isd_stations, aes(label = usaf), size=2, position = position_nudge(x = +0.09)) +
  geom_sf_label(data = k, aes(label = DN), size=2) +
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.2, "cm")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.05, "in"), pad_y = unit(0.05, "in"), height = unit(1, "cm"),
   width = unit(1, "cm"), style = north_arrow_fancy_orienteering) +
  coord_sf(xlim = c(-73.7, -72.8), ylim = c(10.1, 10.9), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Quality Data Comparison\nStations 28025502 from IDEAM, \n800360 from ISD, and 416 from ERA5") +
  theme(plot.title = element_text(size=8)) +
  theme(axis.text.x= element_text(size=7)) +
  theme(axis.text.y= element_text(size=7)) +
  theme(panel.border = element_rect(colour = "red"))+
  #theme(plot.background = element_rect(color = "black")) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.5), panel.background = element_rect(fill = "aliceblue"))
grid.arrange(big, small, ncol=2)
#
#Quality Data Comparison. High similarity between sources
#
dat1 <- readRDS("data/comparison13.rds")
dat1
```

2. Downscaling Support - Non-Quality Data Comparison

```
#
#IDEAM VV_AUT_10 - Non Quality Data Comparison
#
myPalette <- colorRampPalette(rev(brewer.pal(11, "YlGn")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(0, 255))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

pts <- do.call(rbind, st_centroid(st_geometry(era5_4326_sf_pol)))
x = pts[,1]
y = pts[,2]

era5_4326_sf_pol$x = x
era5_4326_sf_pol$y = y
```

```
era5_4326_sf_pol_filter_good = era5_4326_sf_pol %>% filter(DN %in% c(2261, 1971,2066,2020,2260,1875,2213,2637,1442,1583,1501,1582,1381,1493,1485,
 1397,1338,1055,511,1644,515,221,1038))

era5_4326_sf_pol_filter_very_good = era5_4326_sf_pol %>% filter(DN %in% c(265,360,78,312,416))

era5_4326_sf_pol_col1 = era5_4326_sf_pol %>% filter(DN %in% c(1, 50, 148, 246, 344, 442, 540, 638, 736, 834, 932, 1030, 1128, 1226, 1324, 1422,
 1520, 1618, 1716, 1814, 1912, 2010, 2108, 2206, 2304, 2402, 2500, 2598, 2696, 2794, 2892, 2990, 3088, 3186, 3284, 3333))

era5_4326_sf_pol_col49 = era5_4326_sf_pol %>% filter(DN %in% c(49, 147, 245, 343, 441, 539, 637, 735, 833, 931, 1029, 1127, 1225, 1323, 1421, 1519,
 1617, 1715, 1813, 1911, 2009, 2107, 2205, 2303, 2401, 2499, 2597, 2695, 2793, 2891, 2989, 3087, 3185, 3283, 3381))

ggplot(data = world) +
  geom_sf(fill= "antiquewhite",  size=0.1, alpha=0.7) +
  ggRGB(img_stack_col.mask, r = 1, g = 2, b = 3, ggLayer = TRUE, alpha=0.8)+
  sc+
  geom_sf(data = era5_4326_sf_pol, colour="grey", fill=NA, size=0.1) +
  geom_sf(data = era5_4326_sf_pol_filter_good, aes(fill = "Good: 23"), colour="black", size=0.1, alpha=1, show.legend = "polygon") +
  geom_sf(data = era5_4326_sf_pol_filter_very_good, aes(fill = "Very Good: 5"), colour="black", size=0.1, alpha=1, show.legend = "polygon") +
  scale_fill_manual(values = c("Good: 23" = "yellow", "Very Good: 5" = "orange"), name="Downscaling Support") + #, label=c("dd")) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  geom_text_repel(data = era5_4326_sf_pol_col1, size=2, aes(x=x, y=y, label = DN), direction="y", segment.size=0.1, segment.color= "grey50",
   color="grey50", nudge_x=-1, hjust=1, box.padding=0.1) +
  geom_text_repel(data = era5_4326_sf_pol_col49, size=2, aes(x=x, y=y, label = DN), direction="y", segment.size=0.1, segment.color= "grey50",
   color="grey50", nudge_x=+1, hjust=0, box.padding=0.1) +
  coord_sf(xlim = c(-81.25, -64.75), ylim = c(-5, 13), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("ERA5 grid, cells IDs from 1 to 3381 (49 cols, 69 rows)\nISD-IDEAM-ERA5 'poor data' comparison") +
  theme(plot.title = element_text(size=8)) +
  theme(axis.text.x= element_text(size=7)) +
  theme(axis.text.y= element_text(size=7)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -10))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10)))
#
#Non Quality Data Comparison. Time Series Graphic for 'Very Good' Downscaling Support
#
tl = readRDS(file="./data/eracomparisonideam_188_78_plotxts.rds")
tl
#
#Non Quality Data Comparison: Scatter plots for 'Very Good' Downscaling Support
#
par(mfrow=c(3,2))
par(mar=c(0,0,0,0))
par(oma=c(0,0,0,0))

cl = readRDS(file="./data/eracomparisonideam_129_1_2_265_plotgg.rds")
cr = readRDS(file="./data/eracomparisonideam_195_1_2_312_plotgg.rds")
bl = readRDS(file="./data/eracomparisonideam_188_1_3_78_plotgg.rds")
br = readRDS(file="./data/eracomparisonideam_188_1_2_78_plotgg.rds")

plot_grid(cl, cr, bl, br, ncol = 2, labels=c("A", "B", "C", "D"), label_size = 7, label_colour = "red")
```

3. POT-PP for ISD Station 801120

```
#POT-PP in one ISD Station
#
#Output of this code goes to the folder '.../index/data/' of the 'thesisdown' root folder.
source('./code/pp_one_station.r')
#
#Non-Thunderstorm Time Series for ISD station 801120. Left: Raw Data. Right: De-clustered Data
#
plot_grid(myprint1, NULL, myprint5, ncol = 3,  rel_widths = c(1,0.26,1))
#
#POT - Thresholding
#
dat <- readRDS("data/myprint6.rds")
dat
#
#Graphic Diagnosis Of Goodness of Fit. Station 801120
#
replayPlot(myprint8)
add_label <- function(xfrac, yfrac, label, pos = 4, ...) {
  u <- par("usr")
  x <- u[1] + xfrac * (u[2] - u[1])
  y <- u[4] - yfrac * (u[4] - u[3])
  text(x, y, label, pos = pos, ...)
}
par(xpd = TRUE)
add_label(-0.06, 0.01, "A", cex=0.6, col="red")
add_label(0.47, 0.01, "B", cex=0.6, col="red")
add_label(-0.06, 0.54, "C", cex=0.6, col="red")
add_label(0.47, 0.54, "D", cex=0.6, col="red")
#
#Hazard Curve. Station 801120
#
```

```
x= 1/paP
y= yvels
df = data.frame(x = x, y = y)

x1= tipicalReturnPeriods
y1= veocitiesfortypicalreturnperiodsP

df2 = data.frame(x = x1, y = y1)

df2_more1700 <- df2 %>% filter(x >= c(1700))
df2_less1700 <- df2 %>% filter(x < c(1700))

ggplot(data=df, aes(x=x, y=y, group=1)) +
  geom_line(color="red")+
  geom_point(data=df2, aes(x=x, y=y, group=1), shape = 18, color = "black", fill="lightgray") +
  xlim(0,8500) +
  ylim(125, 300) +
  geom_text_repel(data=df2_more1700, size=2, aes(x=x, y=y, label = paste0("(",x,",",round(y, digits=1),")")),  direction="y", segment.size=0.1,
   nudge_y=15) +
  geom_text_repel(data=df2_less1700, size=2, aes(x=x, y=y, label = paste0("(",x,",",round(y, digits=1),")")),  direction="x", segment.size=0.1,
   nudge_x=1) +
  xlab("Return Periods (Years) - POT-PP Intensity Function") +
  ylab("Velocities Km/h") +
  ggtitle(paste("Declustered - Non-Thunderstorms - Hazard Curve", "\n", "Location=", round(z6,2),  "Scale=", round(z7,2))) +
  theme(plot.title = element_text(size=8)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(axis.title =element_text(size=7))
```

4. Wind Maps

```
#
#Hurricane Wind Maps
#
file_rl_h_4326_700_st = "./data/hurricanemaps/h_700.tif"
rl_h_4326_700_st = read_stars(file_rl_h_4326_700_st)
rl_h_4326_700_st = setNames(rl_h_4326_700_st, "Kph")

file_rl_h_4326_1700_st = "./data/hurricanemaps/h_1700.tif"
rl_h_4326_1700_st = read_stars(file_rl_h_4326_1700_st)
rl_h_4326_1700_st = setNames(rl_h_4326_1700_st, "h_1700")

file_rl_h_4326_3000_st = "./data/hurricanemaps/h_3000.tif"
rl_h_4326_3000_st = read_stars(file_rl_h_4326_3000_st)
rl_h_4326_3000_st = setNames(rl_h_4326_3000_st, "h_3000")

myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(27, 438), breaks=c(27, 100, 200, 300, 400, 438))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

c700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_h_4326_700_st, aes(fill = Kph, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-69.4, y=8.5, label=name), color="darkblue", fontface="bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.1, y=9.2, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=5.5, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 13.4, label = "Caribbean Sea", fontface = "italic", color = "grey22", size = 1.8) +
  annotate(geom = "text", x = -79, y = 5.6, label = "Pacific Sea", fontface = "italic", color = "grey22", size = 1.8) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1) +
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-80.5, -67.7), ylim = c(4.5, 13.9), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane Wind Map. \nMRI-700 years - Ingeniar Ltda") +
  theme(plot.title = element_text(size=6)) +
  theme(axis.text.x= element_text(size=5)) +
  theme(axis.text.y= element_text(size=5)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
```

```
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10))) +
  theme(legend.title = element_text(size = 6)) +
  theme(legend.text = element_text(size = 5)) +
  theme(legend.background = element_blank()) +
  theme(legend.key.width = unit(0.1, "cm")) +
  theme(legend.key.height = unit(0.5, "cm"))

legend <- get_legend(c700)

c700 = c700 + theme(legend.position = "none")
c1700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_h_4326_1700_st, aes(fill = h_1700, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-69.4, y=8.5, label=name), color="darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.1, y=9.2, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=5.5, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 13.4, label = "Caribbean Sea", fontface = "italic", color = "grey22", size = 1.8) +
  annotate(geom = "text", x = -79, y = 5.6, label = "Pacific Sea", fontface = "italic", color = "grey22", size = 1.8) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-80.5, -67.7), ylim = c(4.5, 13.9), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane Wind Map. \nMRI-1700 years - Ingeniar Ltda") +
  theme(plot.title = element_text(size=6)) +
  theme(axis.text.x= element_text(size=5)) +
  theme(axis.text.y= element_text(size=5)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10))) +
  theme(legend.position = "none")

c3000 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_h_4326_3000_st, aes(fill = h_3000, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-69.4, y=8.5, label=name), color="darkblue", fontface="bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[panama,],aes(x=-79.1, y=9.2, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=5.5, label=name), color = "darkblue", fontface = "bold", size=1.5, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 13.4, label = "Caribbean Sea", fontface = "italic", color = "grey22", size = 1.8) +
  annotate(geom = "text", x = -79, y = 5.6, label = "Pacific Sea", fontface = "italic", color = "grey22", size = 1.8) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1) +
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-80.5, -67.7), ylim = c(4.5, 13.9), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane Wind Map. \nMRI-3000 years - Ingeniar Ltda") +
  theme(plot.title = element_text(size=6)) +
  theme(axis.text.x= element_text(size=5)) +
  theme(axis.text.y= element_text(size=5)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -10))) +
  theme(legend.position = "none")
plot_grid(c700, c1700, c3000, legend, ncol=4, rel_widths=c(1, 1, 1, 0.13))
#
#ISD Non-Hurricane Wind Maps
#
file_rl_nh_4326_700_st = "./data/isdmaps/nh_700.tif"
rl_nh_4326_700_st = read_stars(file_rl_nh_4326_700_st)
rl_nh_4326_700_st = setNames(rl_nh_4326_700_st, "Kph")

file_rl_nh_4326_1700_st = "./data/isdmaps/nh_1700.tif"
rl_nh_4326_1700_st = read_stars(file_rl_nh_4326_1700_st)
rl_nh_4326_1700_st = setNames(rl_nh_4326_1700_st, "nh_1700")

file_rl_nh_4326_3000_st = "./data/isdmaps/nh_3000.tif"
rl_nh_4326_3000_st = read_stars(file_rl_nh_4326_3000_st)
rl_nh_4326_3000_st = setNames(rl_nh_4326_3000_st, "nh_3000")
```

```
myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(188, 346),
                           breaks=c(188, 200, 225, 250, 275, 300, 325, 346))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

c700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nh_4326_700_st, aes(fill = Kph, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-700 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.title = element_text(size = 6)) +
  theme(legend.text = element_text(size = 6)) +
  theme(legend.background = element_blank()) +
  theme(legend.key.width = unit(0.1, "cm"))

legend <- get_legend(c700)

c700 = c700 + theme(legend.position = "none")

c1700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nh_4326_1700_st, aes(fill = nh_1700, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-1700 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

c3000 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nh_4326_3000_st, aes(fill = nh_3000, x = x, y = y)) +
  sc+
```

```
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1) +
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-3000 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

plot_grid(c700, c1700, c3000, legend, ncol=4, rel_widths=c(1, 1, 1, 0.1))
#
#ERA5 Non-Hurricane Wind Maps
#
file_rl_nonhurricanes_4326_700_st = "./data/era5maps/rl_nonhurricanes_4326_700_st.tif"
rl_nonhurricanes_4326_700_st = read_stars(file_rl_nonhurricanes_4326_700_st)
rl_nonhurricanes_4326_700_st = setNames(rl_nonhurricanes_4326_700_st, "Kph")

file_rl_nonhurricanes_4326_1700_st = "./data/era5maps/rl_nonhurricanes_4326_1700_st.tif"
rl_nonhurricanes_4326_1700_st = read_stars(file_rl_nonhurricanes_4326_1700_st)
rl_nonhurricanes_4326_1700_st = setNames(rl_nonhurricanes_4326_1700_st, "nh_1700")

file_rl_nonhurricanes_4326_3000_st = "./data/era5maps/rl_nonhurricanes_4326_3000_st.tif"
rl_nonhurricanes_4326_3000_st = read_stars(file_rl_nonhurricanes_4326_3000_st)
rl_nonhurricanes_4326_3000_st = setNames(rl_nonhurricanes_4326_3000_st, "nh_3000")

myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(36, 129), breaks=c(36, 50, 75, 100, 120, 129))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

c700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nonhurricanes_4326_700_st, aes(fill = Kph, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-700 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.title = element_text(size = 6)) +
  theme(legend.text = element_text(size = 6)) +
```

```
  theme(legend.background = element_blank()) +
  theme(legend.key.width = unit(0.1, "cm"))

legend <- get_legend(c700)

c700 = c700 + theme(legend.position = "none")

c1700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nonhurricanes_4326_1700_st, aes(fill = nh_1700, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-1700 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

c3000 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_nonhurricanes_4326_3000_st, aes(fill = nh_3000, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Non-Hurricane Wind Map. \nMRI-3000 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

plot_grid(c700, c1700, c3000, legend, ncol=4, rel_widths=c(1, 1, 1, 0.1))
#
#ISD Hurricane & Non-Hurricane Wind Maps
#
file_rl_combined_4326_700_st = "./data/isdmaps/isd_combined_4326_700_st.tif"
rl_combined_4326_700_st = read_stars(file_rl_combined_4326_700_st)
rl_combined_4326_700_st = setNames(rl_combined_4326_700_st, "Kph")

file_rl_combined_4326_1700_st = "./data/isdmaps/isd_combined_4326_1700_st.tif"
rl_combined_4326_1700_st = read_stars(file_rl_combined_4326_1700_st)
rl_combined_4326_1700_st = setNames(rl_combined_4326_1700_st, "c_1700")

file_rl_combined_4326_3000_st = "./data/isdmaps/isd_combined_4326_3000_st.tif"
rl_combined_4326_3000_st = read_stars(file_rl_combined_4326_3000_st)
rl_combined_4326_3000_st = setNames(rl_combined_4326_3000_st, "c_3000")

myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(188, 438), breaks=c(188, 200, 250, 300, 350, 400, 438))

theme_set(theme_bw())
```

```
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

c700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_700_st, aes(fill = Kph, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-700 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.title = element_text(size = 6)) +
  theme(legend.text = element_text(size = 6)) +
  theme(legend.background = element_blank()) +
  theme(legend.key.width = unit(0.1, "cm"))

legend <- get_legend(c700)

c700 = c700 + theme(legend.position = "none")

c1700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_1700_st, aes(fill = c_1700, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-1700 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

c3000 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_3000_st, aes(fill = c_3000, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
```

```
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-3000 years - ISD") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

plot_grid(c700, c1700, c3000, legend, ncol=4, rel_widths=c(1, 1, 1, 0.1))
#
#ERA5 Hurricane & Non-Hurricane Wind Maps
#
file_rl_combined_4326_700_st = "./data/era5maps/rl_combined_4326_700_st.tif"
rl_combined_4326_700_st = read_stars(file_rl_combined_4326_700_st)
rl_combined_4326_700_st = setNames(rl_combined_4326_700_st, "Kph")

file_rl_combined_4326_1700_st = "./data/era5maps/rl_combined_4326_1700_st.tif"
rl_combined_4326_1700_st = read_stars(file_rl_combined_4326_1700_st)
rl_combined_4326_1700_st = setNames(rl_combined_4326_1700_st, "c_1700")

file_rl_combined_4326_3000_st = "./data/era5maps/rl_combined_4326_3000_st.tif"
rl_combined_4326_3000_st = read_stars(file_rl_combined_4326_3000_st)
rl_combined_4326_3000_st = setNames(rl_combined_4326_3000_st, "c_3000")

myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))
sc <- scale_fill_gradientn(colours = myPalette(100), limits=c(37, 438), breaks=c(37, 100, 200, 300, 400, 438))

theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
world_points<- st_centroid(world)
world_points <- cbind(world, st_coordinates(st_centroid(world$geometry)))

colombia = world_points$name == "Colombia"
panama = world_points$name == "Panama"
peru= world_points$name == "Peru"
brazil= world_points$name == "Brazil"
venezuela= world_points$name == "Venezuela"
ecuador= world_points$name == "Ecuador"

c700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_700_st, aes(fill = Kph, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-700 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.title = element_text(size = 6)) +
  theme(legend.text = element_text(size = 6)) +
  theme(legend.background = element_blank()) +
  theme(legend.key.width = unit(0.1, "cm"))

legend <- get_legend(c700)
```

```
c700 = c700 + theme(legend.position = "none")

c1700 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_1700_st, aes(fill = c_1700, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-1700 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

c3000 = ggplot(data = world) +
  geom_sf(fill= "antiquewhite", size=0.1) +
  geom_stars(data = rl_combined_4326_3000_st, aes(fill = c_3000, x = x, y = y)) +
  sc+
  geom_text(data= world_points[venezuela,],aes(x=-66.5, y=8.5, label=name), color="darkblue", fontface="bold", size=2, check_overlap=FALSE) +
  geom_text(data= world_points[panama,],aes(x=-80.3, y=9.2, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[ecuador,],aes(x=-78.8, y=-1, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[peru,],aes(x=-75, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[brazil,],aes(x=-68, y=-6, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  geom_text(data= world_points[colombia,],aes(x=-71, y=4, label=name), color = "darkblue", fontface = "bold", size=2, check_overlap = FALSE) +
  annotate(geom = "text", x = -77.5, y = 14, label = "Caribbean\nSea", fontface = "italic", color = "grey22", size = 2) +
  annotate(geom = "text", x = -80.5, y = 5, label = "Pacific\nSea", fontface = "italic", color = "grey22", size = 2) +
  geom_sf(data = st_cast(world, "MULTILINESTRING"), size=0.1)+
  annotation_scale(location = "bl", width_hint = 0.5, height = unit(0.1, "cm"), line_width = 0.1, text_cex = 0.5, pad_x = unit(0.03, "in"),
   pad_y = unit(0.03, "in")) +
  annotation_north_arrow(location = "br", which_north = "true", pad_x = unit(0.01, "in"), pad_y = unit(0.02, "in"), height = unit(0.5, "cm"),
   width = unit(0.5, "cm"), style = north_arrow_minimal) +
  coord_sf(xlim = c(-82.1, -63.8), ylim = c(-7.5, 15.5), expand = FALSE) +
  xlab("") +
  ylab("") +
  ggtitle("Hurricane & Non-Hurricane Wind Map. \nMRI-3000 years - ERA5") +
  theme(plot.title = element_text(size=7)) +
  theme(axis.text.x= element_text(size=6)) +
  theme(axis.text.y= element_text(size=6)) +
  theme(panel.grid.major = element_line(color = gray(.5), linetype = "dashed", size = 0.1)) +
  theme(panel.background = element_rect(fill = "aliceblue")) +
  theme(plot.margin=unit(c(0,0,0,0),"cm")) +
  theme(plot.background = element_rect(fill = "transparent", color = NA))+
  theme(axis.text.x = element_text(margin =  margin(t =2, b = -14))) +
  theme(axis.text.y = element_text(margin =  margin(r =2, l = -16))) +
  theme(legend.position = "none")

plot_grid(c700, c1700, c3000, legend, ncol=4, rel_widths=c(1, 1, 1, 0.1))
```

Appendix F

User Manual

Software requirements:

- Software R version 3.6.2 (2019-12-12). Platform x86_64-w64-mingw32. Arch x86_64. Os mingw32. System x86_64, mingw32. Svn rev 77560. Language R. Nickname Dark and Stormy Night.
- RStudio - Version 1.2.5033 - 2009-2019. "Orange Blossom" (330255dd, 2019-12-04)
- Sixty (60) **R** packages. Use next chunk of code to install and load R packages.

```
# List of packages required for this analysis
  pkg <- c("actuar", "bbmle", "bookdown", "cowplot", "devtools", "DiagrammeR", "dplyr", "evd", "evir", "evmix", "extRemes", "extremeStat",
  "fitdistrplus", "geoR", "ggmap", "ggplot2", "ggrepel", "ggspatial", "grid", "gridExtra", "gstat", "ismev", "kableExtra", "knitr", "lattice",
  "lmom", "lmomco", "lubridate", "magick", "maptools", "mapview", "MASS", "ncdf4", "openair", "plot3D", "plotly", "POT", "quantmod", "raster",
  "RcmdrMisc", "RColorBrewer", "Renext", "rgdal", "rgl", "rnaturalearth", "rnaturalearthdata", "RPostgreSQL", "RStoolbox", "sf", "shape",
  "sp", "SpatialExtremes", "stars", "thesisdown", "tibble", "tidyr", "viridis", "xls", "xlsx", "xts")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
  new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
  if (length(new.pkg))
    install.packages(new.pkg, repos = "http://cran.rstudio.com")
#Load packages
  library(actuar)
  library(bbmle)
  library(bookdown)
  library(cowplot)
  library(devtools)
  library(DiagrammeR)
  library(dplyr)
  library(evd)
  library(evir)
  library(evmix)
  library(extRemes)
  library(extremeStat)
  library(fitdistrplus)
  library(geoR)
  library(ggmap)
  library(ggplot2)
  library(ggrepel)
  library(ggspatial)
  library(grid)
  library(gridExtra)
  library(gstat)
  library(ismev)
  library(kableExtra)
  library(knitr)
  library(lattice)
  library(lmom)
  library(lmomco)
  library(lubridate)
  library(magick)
  library(maptools)
  library(mapview)
  library(MASS)
  library(ncdf4)
```

```
library(openair)
library(plot3D)
library(plotly)
library(POT)
library(quantmod)
library(raster)
library(RcmdrMisc)
library(RColorBrewer)
library(Renext)
library(rgdal)
library(rgl)
library(rnaturaleearth)
library(rnaturaleearthdata)
library(RPostgreSQL)
library(RStoolbox)
library(sf)
library(shape)
library(sp)
library(SpatialExtremes)
library(stars)
library(thesisdown)
library(tibble)
library(tidyr)
library(viridis)
library(xls)
library(xlsx)
library(xts
```

F.1 Data Standardization

After calculation of correction factors for each station (IDEAM or ISD), see *Data Standardization* in *Methodology* section, procedure to modify original wind velocity values using R code, can be done by different ways.

There is one text file for each station time series in IDEAM variables VV_AUT_2 (instantaneous wind velocity each two seconds), and VV_AUT_10 (instantaneous wind velocity each ten seconds), see 2.2. There are twelve files corresponding to twelve different stations. File names follow the format *VV_AUT_2@*.data*, where * is replaced by the station identifier, as can be seen below. Same format applies to files in variable VV_AUT_10.

```
VV_AUT_2@48015050.data
VV_AUT_2@52055230.data
VV_AUT_2@26125061.data
...
```

Below is the content of a time series file (VV_AUT_2@48015050.data). It has two columns (Fecha and Valor) separated by character |.

```
Fecha|Valor
2019-01-04 00:00:00|0.2
2019-01-04 00:02:00|0.4
2019-01-04 00:04:00|0.4
...
```

Next code from IDEAM variable VV_AUT_2 (wind velocity each two seconds), read all twelve time series text files from folder *./data/manual/VV_AUT_2/*, calculate hourly mean, apply a 3-s gust correction factor of 1.52, and roughness corrections factors stored in the column *fc_zo* of *stationssample* data-frame, nonetheless, this code is not efficient in terms of memory management because it loads all times series to memory using the object *ldf*, and load all standardized time series to memory using object *ldf_hourlymean*.

```
path_vv_aut_2 = "./data/manual/VV_AUT_2/"
filenames = list.files(path_vv_aut_2, pattern = "VV_AUT_2@")
#Be aware that ldf is a list of dataframes and you need to do [[]], to go inside it
#Create a list dataframes in ldf
#To access each station time series, you need to use ldf[[integer]]
ldf = lapply(filenames, function(x) {
  dat = read.table(paste0(path_vv_aut_2, x), header=TRUE, sep="|", stringsAsFactors=FALSE)
  # Add column names
    names(dat) = c('fecha', 'valor')
  #Take control of possible bad original values
    dat$valor[dat$valor > 300] = NA
    dat$valor[dat$valor < 1] = NA
  #Get station id from file name
    dat$station_id = substring(x,10,nchar(x)-5)
  #Add datetime object
    dat$mydatetime = as.POSIXct(dat$fecha,format="%Y-%m-%d %H:%M:%S", tz="UTC")
    return(dat)})
#Stations IDs and Roughness corrections factors
  stationssample = data.frame(
    ideam_id = as.character(
      c(48015050, 52055230, 26125061, 26125710, 23085270, 27015330,
        16015501, 23195502, 13035501, 28025502, 15065180, 29045190)),
    fc_zo =
      c(1.197230052, 1.219771719, 1.102205474 , 1.18154867, 1.113341504, 1.29241596,
        1.102205474, 1.177562503, 1.114968586, 1.184849704, 1.5, 1.224744381),
    stringsAsFactors=FALSE)
#Calculate hourly mean from original time series, and apply correction factors
  ldf_hourlymean <- NULL
  for(station in ldf){
    library(xts)
    library(dplyr)
    select <- dplyr::select
    #Create XTS object removing NA values
      myxts = na.omit(xts(x=select(station, valor), order.by = station$mydatetime))
    #Conversion to 3-s gust using Durst curve
    #Gust factor from Durst curve equal to 1.52
      myxts$valor = myxts$valor * 1.52
    #Correction factor by Roughness
      fczo = stationssample$fc_zo[stationssample$ideam_id == station$station_id[1]]
      myxts$valor = myxts$valor * fczo
    #Name of time series is the string "X" + Station ID
      colnames(myxts) = paste0("X", station$station_id[1])
    #Extract index values of last observation of each hour
      endhour = endpoints(myxts,on="hours")
    #Calculate hourly mean
      myxtshour = xts::period.apply(myxts, INDEX=endhour, FUN=mean)
    #Rounding time series to previous hour
      index(myxtshour)=trunc(index(myxtshour),"hours")
    #Store all standardized time series in ldf_hourlymean
      ldf_hourlymean <- cbind(ldf_hourlymean, myxtshour)}
```

Next code from ISD data source stored in unstacked PostgreSQL table *isd_lite_unstack*, load twelve stations (using WHERECLAUSE to filter) and apply roughness correction factors. Correction factor values are identical because listed ISD stations are equivalent to the stations listed in the code above. An efficient management of memory is made in the code due to the use of lazy `tibble` data-frames from PostgreSQL database, see Annex D *Database Storing*.

```
#Stations IDs and Roughness corrections factors
stationssample = data.frame(
  isd_usaf_id= as.character(
    c(803980, 803700, 802110, 802100, 801120, 801100, 800970, 800940, 800630, 800360, 800350, 800280)),
  fc_zo =
```

```
        c(1.197230052, 1.219771719, 1.102205474 , 1.18154867, 1.113341504, 1.29241596,
          1.102205474, 1.177562503, 1.114968586, 1.184849704, 1.5, 1.224744381),
      stringsAsFactors=FALSE)
    #List of field to read from table, formated with "X" character as column prefix name,
    #and double quotes
      originalfields1 = stationssample$isd_usaf_id
      newfields1 = paste ("X", originalfields1, sep="")
      originalfields1 = paste('"', originalfields1, '"', sep = "")
      newfields1 = paste('"', newfields1, '"', sep = "")
    #Roughness correction factor
      fczo = stationssample$fc_zo
    #Apply roughness correction factor, and
    #force to NULL all values below one
      fiedls_query1 = paste("CASE WHEN", originalfields1, "< 1", "THEN NULL ELSE", originalfields1,
       "*", fczo, "END AS", newfields1, sep = " ")
    #Join datatime field "mydatetime" to the query
      fiedls_query1 = c(paste('"', "mydatetime", '"', sep = ""), fiedls_query1)
      fiedls_query1 = paste (fiedls_query1, "", sep= "", collapse=", ")
    #Construct WHERECLAUSE
      wherestring1 = stationssample$isd_usaf_id
      wherestring1 = paste('"', wherestring1, '"', sep = "")
      #Only take values greater than one and not null
      wherestring1 = paste(wherestring1, ">= 1 AND", wherestring1, "IS NOT NULL", sep = " ")
      wherestring1 = paste(wherestring1, collapse = ") OR (" , sep = " ")
      wherestring1 = paste("(", wherestring1, ")", sep ="")
    #Final query
      query1 = paste("select", fiedls_query1, "from isd_lite_unstack", "where", wherestring1, sep=" ")
    #View content of query1, but first split SQL command to show inside PDF document
      mycat <- function(text){
        text2 = gsub(pattern = "CASE", replacement = "\n CASE", x = text)
        text3 = gsub(pattern = "from", replacement = "\n from", x = text2)
        cat(gsub(pattern = "OR", replacement = "\n OR", x = text3))}
      mycat(query1)
```

```
select "mydatetime",
 CASE WHEN "803980" < 1 THEN NULL ELSE "803980" * 1.197230052 END AS "X803980",
 CASE WHEN "803700" < 1 THEN NULL ELSE "803700" * 1.219771719 END AS "X803700",
 CASE WHEN "802110" < 1 THEN NULL ELSE "802110" * 1.102205474 END AS "X802110",
 CASE WHEN "802100" < 1 THEN NULL ELSE "802100" * 1.18154867 END AS "X802100",
 CASE WHEN "801120" < 1 THEN NULL ELSE "801120" * 1.113341504 END AS "X801120",
 CASE WHEN "801100" < 1 THEN NULL ELSE "801100" * 1.29241596 END AS "X801100",
 CASE WHEN "800970" < 1 THEN NULL ELSE "800970" * 1.102205474 END AS "X800970",
 CASE WHEN "800940" < 1 THEN NULL ELSE "800940" * 1.177562503 END AS "X800940",
 CASE WHEN "800630" < 1 THEN NULL ELSE "800630" * 1.114968586 END AS "X800630",
 CASE WHEN "800360" < 1 THEN NULL ELSE "800360" * 1.184849704 END AS "X800360",
 CASE WHEN "800350" < 1 THEN NULL ELSE "800350" * 1.5 END AS "X800350",
 CASE WHEN "800280" < 1 THEN NULL ELSE "800280" * 1.224744381 END AS "X800280"
 from isd_lite_unstack where ("803980" >= 1 AND "803980" IS NOT NULL)
 OR ("803700" >= 1 AND "803700" IS NOT NULL)
 OR ("802110" >= 1 AND "802110" IS NOT NULL)
 OR ("802100" >= 1 AND "802100" IS NOT NULL)
 OR ("801120" >= 1 AND "801120" IS NOT NULL)
 OR ("801100" >= 1 AND "801100" IS NOT NULL)
 OR ("800970" >= 1 AND "800970" IS NOT NULL)
 OR ("800940" >= 1 AND "800940" IS NOT NULL)
 OR ("800630" >= 1 AND "800630" IS NOT NULL)
 OR ("800360" >= 1 AND "800360" IS NOT NULL)
 OR ("800350" >= 1 AND "800350" IS NOT NULL)
 OR ("800280" >= 1 AND "800280" IS NOT NULL)
```

```
    #Connect to Database
      library('RPostgreSQL')
```

```
Loading required package: DBI
```

```
    pg = dbDriver("PostgreSQL")
    con1 = dbConnect(pg, user="user1", password="user1", host="localhost", port=5432, dbname="winddata")
  #Create tibble dataframe with applied correction factors
    isdlite = tbl(con1, sql(query1))
    class(isdlite)
```

```
[1] "tbl_PostgreSQLConnection" "tbl_dbi"
[3] "tbl_sql"                  "tbl_lazy"
[5] "tbl"
```

```
    select(isdlite, paste0("X", stationssample$isd_usaf_id[6:12]))
```

```
# Source:   lazy query [?? x 7]
# Database: postgres 10.0.5 [user1@localhost:5432/winddata]
   X801100 X800970 X800940 X800630 X800360 X800350 X800280
     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1      NA      NA      NA      NA      NA      NA    1.22
 2      NA      NA      NA      NA      NA      NA    6.25
 3      NA      NA      NA      NA      NA      NA    6.25
 4      NA      NA      NA      NA      NA      NA    2.57
 5      NA      NA      NA      NA      NA      NA    3.80
 6      NA      NA      NA      NA      NA      NA    2.57
 7      NA      NA      NA      NA      NA      NA    3.80
 8      NA      NA      NA      NA      NA      NA    1.22
 9      NA      NA      NA      NA      NA      NA    5.02
10      NA      NA      NA      NA      NA      NA    1.22
# ... with more rows
```

Finally, recommended option to apply correction factors is shown using variable `VV_AUT_10` (instantaneous wind velocity each ten seconds) from IDEAM data-source. It is based in an Excel file with all correction factors by each station, and time series in text files. In column total correction factor F_{total} of Table F.1, integrates correction factors due to surface roughness F_e, and gust F_{gust}. As it is described in *Data Standardization*, Lettau (1969) is applied to calculate F_e, using roughness Z_o, gradient height Z_g, empirical exponent α, and exposure coefficient K_z. For F_{gust}, Durst curve is used, see *Averaging Time 3-s Gust*. Despite it is not using lazy `tibble` data-frames in PostgreSQL, next code uses efficiently memory resources, as it loads only one file time series to memory in a serial iterative process.

Table F.1: Excel Sheet with Corrections Factors

Station ID	Z_o	Source	Z_g	α	K_z	F_e	F_{gust}	F_{total}
789820	0.03	ISD Neighbors	290.3	9	0.95	1.05	1.03	1.08
804250	0.18	ISD Neighbors	362.7	7.1	0.73	1.3	1.03	1.34
...	...	...	...	...	...	...	...	...
800010	0.23	ISD Colombia	375.5	6.85	0.7	1.36	1.03	1.4
800090	0.06	ISD Colombia	315.6	8.24	0.87	1.09	1.03	1.13
...	...	...	...	...	...	...	...	...
11105020	0.1	IDEAM	337.4	7.67	0.8	1.18	1.51	1.79
12015100	0.05	IDEAM	309.4	8.41	0.89	1.07	1.51	1.61
...	...	...	...	...	...	...	...	...

```
  library(xlsx)
#Read file with correction factors
  fc <-read.xlsx(file="./data/manual/correction_factors_isd_ideam.xlsx", sheetName="fc", header=TRUE,
                 stringsAsFactors = F)
#In next folder all text files time series from variable VV_AUT_10 are stored
  path_vv_aut_10 = "./data/manual/VV_AUT_10/"
#Read files inside previous folder, with pattern VV_AUT_10@ in the name
```

```
  filenames = list.files(path_vv_aut_10, pattern = "VV_AUT_10@")
#Variable to hold station IDs without roughness correction factor
  mymissing = NULL
#Do a loop for each 'filename' inside 'filenames' variable
  for(filename in filenames){
    #Read current filename from disk into memory, variable 'dat'
      dat = read.table(paste0(path_vv_aut_10, filename), header=TRUE, sep="|", stringsAsFactors=FALSE)
    # Modify column names
      names(dat) = c('fecha', 'valor')
    #Take control of possible bad original values
      dat$valor[dat$valor > 300] = NA
      dat$valor[dat$valor < 1] = NA
    #Get station id from current file name
      station_id = substring(filename, 11, nchar(filename)-5)  #this is for vv_aut_2
    #Add datetime object
      dat$mydatetime = as.POSIXct(dat$fecha,format="%Y-%m-%d %H:%M:%S", tz="UTC")
    library(xts)
    library(dplyr)
    select <- dplyr::select
    #Create XTS object removing NA values
      myxts = na.omit(xts(x=select(dat, valor), order.by = dat$mydatetime))
    #Get correction factor by Roughness
      fexpo = fc$fexpo[fc$station_id == as.integer(station_id)]
    #Keep the record of stations without correction factors in Excel file
      if (length(fexpo) == 0) {mymissing = c(mymissing, station$station_id[1])
          cat(paste("Station ", station$station_id[1], "not found in correction factors file", "\n"))}
    #Get correction factor by gust
    #All stations are reported in Excel file for gust correction factor
      fgust = fc$fgust[fc$station_id == as.integer(station$station_id[1])]
    #Apply correction factors
      myxts$valor = myxts$valor * fexpo * fgust
    #Name of time series is the string "X" + Station ID
      colnames(myxts) = paste0("X", station$station_id[1])
    #Extract index values of last observation of each hour
      endhour = endpoints(myxts,on="hours")
    #Calculate hourly mean
      myxtshour = xts::period.apply(myxts, INDEX=endhour, FUN=mean)
    #Rounding time series to previous hour
      index(myxtshour)=trunc(index(myxtshour),"hours")
    #Hourly corrected time series are ready!
    #Proceed to apply next step in methodology: POT-PP
    #...}
```

F.2 Downscaling Support

Table F.2 shows specific code used to compare all data sources. Complete R code report can be found in Annex A - Research R Code - Digital Files, Table A.1.

Table F.2: Downscaling Support R Code. ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password).

Folder Tree - Ftp Links	Description
code	Folder with R code. ALL CODE CREATED BY DR. ADAM PINTAR IS NOT PUBLISHED.
|-downscaling	Folder with code to compare all data sources, looking for downscaling support.
|-qualitydata	Folder with code to compare using quality data from IDEAM (variable VV_AUT_2).
| |-VV_AUT_2_1.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (1).
| |-VV_AUT_2_2.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (2).
| |-VV_AUT_2_3.r	Using predefined list of matching stations (ISD vs IDEAM). ERA5 match is by intersection (3).
|-nonqualitydata	Folder with code to compare using non-quality data from IDEAM (variable VV_AUT_10).
|-VV_AUT_10.r	All stations from ISD or IDEAM that intersects one ERA5 cell are compared.

F.2.1 Quality Data Comparison

Procedure to run data comparison using VV_AUT_2 IDEAM variable:

For this procedure, needed R files are VV_AUT_2_1.r, VV_AUT_2_2.r, and VV_AUT_2_3.r, see Table F.2.

1. Install R version 3.6.2, RStudio Version 1.2.5033, and 60 R packages dependencies (see chunk of code at beginning of this manual)
2. Verify files and variables according to descriptions and recommendations of following list (from 1 to 5), then execute file **VV_AUT_2_1.r**.

Files to run quality data comparison are inside the folder *.../downscaling/qualitydata/*. Main file to run this process is *VV_AUT_2_1.r*, and inside it, next list of variables and code need to be configured:

1. *stationssample*

```
stationssample = data.frame(
  isd_usaf_id= as.character(
    c(803980, 803700, 802110, 802100, 801120, 801100,
      800970, 800940, 800630, 800360, 800350, 800280)),
  ideam_id = as.character(
    c(48015050, 52055230, 26125061, 26125710, 23085270, 27015330,
      16015501, 23195502, 13035501, 28025502, 15065180, 29045190)),
  fc_zo =
    c(1.197230052, 1.219771719, 1.102205474 , 1.18154867, 1.113341504, 1.29241596,
      1.102205474, 1.177562503, 1.114968586, 1.184849704, 1.5, 1.224744381),
  stringsAsFactors=FALSE)
```

Inside file *VV_AUT_2_1.r*, all needed correction factors are hard coded in variable *stationssample*. If a more suitable calculation of correction factors is done, those values can be updated inside this data-frame.

2. *path_vv_aut_2*

```
path_vv_aut_2 = "./data/manual/VV_AUT_2/"
```

Inside folder *path_vv_aut_2*, non-standardized time series text files must be stored (one file per station history). Be aware that all wind velocities must not be standardized. Twelve files are inside this folder, see complete list below.

```
VV_AUT_2@13035501.data
VV_AUT_2@15065180.data
VV_AUT_2@16015501.data
VV_AUT_2@23085270.data
VV_AUT_2@23195502.data
VV_AUT_2@26125061.data
VV_AUT_2@26125710.data
VV_AUT_2@27015330.data
VV_AUT_2@28025502.data
VV_AUT_2@29045190.data
VV_AUT_2@48015050.data
VV_AUT_2@52055230.data
```

Files names must follow this convention **VV_AUT_2@*.data**, where * is the IDEAM

station ID. Content of each file must follow structure shown below.

```
Fecha|Valor
2015-05-30 17:20:00|0.2
2015-05-30 17:22:00|0.1
2015-05-30 17:26:00|0.1
2015-05-30 17:28:00|0.2
2015-05-30 17:30:00|0.4
2015-05-30 17:32:00|0.2
2015-05-30 17:34:00|0.2
...
```

The first line must contain the text *Fecha/Valor* representing two column names, first column *Fecha* has the date time in the format "YYYY-MM-DD HH:MM:SS" (time zone UTC), and second column *Valor* must have the non-standardized wind velocity in kilometers per hour. Separation of each column value must be the / character.

3. *con1*

```
con1 = dbConnect(pg, user="user1", password="user1", host="localhost", port=5432, dbname="winddata")
```

In variable *con1* all PostgreSQL database credentials are defined. Tables to use are *isd_all_stations*, and *ideam_all_stations*, corresponding to ISD and IDEAM stations catalogs, and *isd_lite_unstack* corresponding to ISD unstacked time series. Verify values in *con1* according to current database configuration.

4. *VV_AUT_2_2.r* and *VV_AUT_2_3.r*

```
source('VV_AUT_2_2.r')
...
source('VV_AUT_2_3.r')
```

Inside files *VV_AUT_2_2.r* and *VV_AUT_2_3.r* all code to read ERA5 dataset is defined. The most important line there, defines the location of NetCDF file *outfile_nc4c_zip9.nc* with variable *fg10* (3-s wind gust). Be sure that *ncname* (shown below) variable is pointing to the right place where NetCDF file is stored.

```
(ncname <- "../data/outfile_nc4c_zip9")
```

5. *somePDFPath*

```
somePDFPath = paste(paste("isdideamera5", i, statideam, sep = "_"), "pdf", sep=".")
```

Variable *somePDFPath* holds the name of the PDF file with the quality data comparison between IDEAM, ISD and ERA5. After execution, there will be a file for each station provided in data frame *stationssample*. Inside each PDF file, comparison time series graphics are provided, as well as scatter plots. Using graphics inside this file, it is possible to visually define if exist downscaling support to use model or forecast data (ISD and ERA5), by the comparison against measure data (IDEAM). See Table B.1 in Annex *Results - Digital Files*, for a description of all digital output files of this research.

F.2.2 Non-quality Data Comparison

Procedure to run data comparison using VV_AUT_10 IDEAM variable:

For this procedure, needed R file is VV_AUT_10.r, see Table F.2.

1. Install R version 3.6.2, RStudio Version 1.2.5033, and 60 R packages dependencies (see chunk of code at beginning of this manual)
2. Verify files and variables according to descriptions and recommendations of following list (from 1 to 6), then execute file **VV_AUT_10.r**.

Inside the folder *.../downscaling/nonqualitydata/*, it is possible to find files to run non-quality data comparison. Main file to run this process is *VV_AUT_10.r*, and inside it, next list of variables and code need to be configured:

1. *con1*

```
con1 = dbConnect(pg, user="user1", password="user1", host="localhost", port=5432, dbname="winddata")
```

In variable *con1* all PostgreSQL database credentials are defined. Tables to use are *isd_all_stations*, and *ideam_all_stations*, corresponding to ISD and IDEAM stations catalogs, and *isd_lite_unstack* corresponding to ISD unstacked time series. Verify values in *con1* according to current database configuration.

2. *stationssample*

```
stationssample = data.frame(
  isd_usaf_id= as.character(
    c(803980, 803700, 802110, 802100, 801120, 801100,
      800970, 800940, 800630, 800360, 800350, 800280)),
  ideam_id = as.character(
    c(48015050, 52055230, 26125061, 26125710, 23085270, 27015330,
      16015501, 23195502, 13035501, 28025502, 15065180, 29045190)),
  fc_zo =
    c(1.197230052, 1.219771719, 1.102205474 , 1.18154867, 1.113341504, 1.29241596,
      1.102205474, 1.177562503, 1.114968586, 1.184849704, 1.5, 1.224744381),
  stringsAsFactors=FALSE)
```

Inside file *VV_AUT_2_1.r*, all needed correction factors are hard coded in variable *stationssample*. If a more suitable calculation of correction factors is done, those values can be updated inside this data-frame.

3. *ncname*

```
ncname <- "../data/outfile_nc4c_zip9"
```

Variable *ncname*, defines the location of NetCDF file *outfile_nc4c_zip9.nc* with variable *fg10* (3-s wind gust). Be sure that *ncname* variable is pointing to the right place where mentioned NetCDF file is stored.

4. *fc*

```
fc <-read.xlsx(file="correction_factors_isd_ideam.xlsx", sheetName="fc", header=TRUE,
               stringsAsFactors = F)
```

Variable *fc* must point to Excel file where correction factors and dependent variables are stored for each analyzed ISD and IDEAM station. This file must have columns *Station ID*, roughness Z_o, *Source*, gradient height Z_g, empirical exponent α, exposure coefficient K_z, exposition correction factor F_e, gust correction factor F_{gust}, and total correction factor F_{total}, as is shown in Table F.1. For a detailed explanation, see *Data Standardization*.

5. *path_vv_aut_10*

```
path_vv_aut_10 = "./data/manual/VV_AUT_10/"
```

Non standardized time series text files for IDEAM data VV_AUT_10 - instantaneous wind velocity each ten (10) minutes, must be stored inside folder *path_vv_aut_10*, one file per station history. Be aware that all wind velocities must not be standardized. There are a total of 204 stations, that is 204 files. The file name follows the format *VV_AUT_10@*.data*, where * is replaced by the station identifier, as can be seen below.

```
VV_AUT_10@31095030.data
VV_AUT_10@29065130.data
VV_AUT_10@29065140.data
...
```

Content of each file must follow structure shown below. The first line must contain the text *Fecha|Valor*, representing two column names, first column *Fecha*, has the date time in the format "YYYY-MM-DD HH:MM:SS" (time zone UTC), and second column *Valor* must have the non-standardized wind velocity in kilometers per hour. Separation of each column value must be the | character.

```
Fecha|Valor
2008-10-29 10:30:00|0.9
2008-10-29 10:40:00|1.4
2008-10-29 10:50:00|1.2
...
```

6. *somePDFPath*

```
somePDFPath = paste(paste("isdideamera5", i, era5_intersect_ideam["ideam_station",i], sep = "_"),
                    "pdf", sep=".")
```

Variable *somePDFPath* holds the name of the output PDF file with non-quality data comparison between IDEAM, ISD and ERA5. After execution, there will be a file for each match between ERA5 cells and ISD and/or IDEAM intersecting stations. Inside each PDF file, comparison time series graphics will be generated, as well as scatter plots. Using graphics inside this file, it is possible to visually define if exist downscaling support to use model or forecast data (ISD and ERA5), by the comparison against measure data (IDEAM). See Table B.1 *Results - Digital Files*, for a description of all digital output files of this research.

F.3 POT-PP

F.3.1 ISD

Table F.3 shows specific POT-PP R code used for ISD stations. Complete R code report can be found in Table A.1 of Annex A.

Table F.3: R Code POT-PP ISD. ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password).

Folder Tree - Ftp Links	Description
code	Folder with R code. ALL CODE CREATED BY DR. ADAM PINTAR IS NOT PUBLISHED.
|-pot_pp	Folder with POT-PP R code. Based in Dr Adam Pintar code (respected copyright).
|-function_lib.r	POT-PP Functions. Author of de-clustering and thresholding functions is Dr Adam Pintar.
|-plot_nt.r	Plot non-thunderstorm graphics.
|-plot_t.r	Plot thunderstorm graphics.
|-plot_t_nt.r	Plot graphics with thunderstorm and non-thunderstorm data, in simultaneous.
|-stats_graphs_dnt.r	Statistics and graphics for non-thunderstorm de-clustered data.
|-stats_graphs_dt.r	Statistics and graphics for thunderstorm de-clustered data.
|-stats_raw_data.r	Statistics for raw data.
|-stats_raw_data_nt.r	Statistics for non-thunderstorm raw data.
|-stats_raw_data_t.r	Statistics for thunderstorm raw data.
|-tnt_csv_1perday.r	Create CSV (thunderstorm and non-thunderstorm) with one data (the maximum) per day.
|-isd	Folder with specific code for ISD data.
|-pot_pp_isd.r	POT-PP for ISD data. Based in Dr Adam Pintar code.
|-maps	Folder with code to calculate return levels, do spatial interpolation, and plot maps. ISD data.
|-rl_10_nh.r	Calculate return levels and do spatial interpolation. MRI 10, non-hurricane data.
|-rl_20_nh.r	Calculate return levels and do spatial interpolation. MRI 20, non-hurricane data.
|-rl_50_nh.r	Calculate return levels and do spatial interpolation. MRI 50, non-hurricane data.
|-rl_100_nh.r	Calculate return levels and do spatial interpolation. MRI 100, non-hurricane data.
|-rl_250_nh.r	Calculate return levels and do spatial interpolation. MRI 250, non-hurricane data.
|-rl_500_nh.r	Calculate return levels and do spatial interpolation. MRI 500, non-hurricane data.
|-rl_700_nh.r	Calculate return levels and do spatial interpolation. MRI 700, non-hurricane data.
|-rl_1000_nh.r	Calculate return levels and do spatial interpolation. MRI 1000, non-hurricane data.
|-rl_1700_nh.r	Calculate return levels and do spatial interpolation. MRI 1700, non-hurricane data.
|-rl_3000_nh.r	Calculate return levels and do spatial interpolation. MRI 3000, non-hurricane data.
|-rl_7000_nh.r	Calculate return levels and do spatial interpolation. MRI 7000, non-hurricane data.
|-rl_combined.r	Integrate return levels from hurricane and non-hurricane data.
|-plot_maps.r	Plot ISD maps.

Procedure to run POT-PP in ISD stations:

1. Install R version 3.6.2, RStudio Version 1.2.5033, and 60 R packages dependencies (see chunk of code at beginning of this manual)
2. Verify files and variables according to descriptions and recommendations of following list (from 1 to 4), then execute file **pot_pp_isd.r**.

Files to run POT-PP in ISD stations are inside the folder *. . . /pot_pp/isd/*. Main file to run this process is **pot_pp_isd.r**, and inside it, next list of variables need to be configured.

1. *inputpath*

```
inputpath="./raw_data/"
```

Inside this folder *inputpath*, standardized time series text files must be stored, one file per station history. Be aware that all wind velocities must be already standardized, see *Data Standardization* section in *Methodology*. Below, an example of the content of this folder is

shown.

```
raw_data_station_800740.txt
raw_data_station_800770.txt
...
```

Files names must follow this convention **raw_data_station_*.txt**, where * must be replaced by the station ID. Its content must follow structure shown below.

```
"date_time" "kph" "thunder_flag"
"1950/06/22 12:00:00 GMT"  122.60934 "nt"
"1951/06/13 12:00:00 GMT"   203.9521 "nt"
"1951/08/02 12:00:00 GMT"  30.553136 "nt"
"1963/02/26 12:00:00 GMT"  26.585196 "nt"
...
```

The first line must contain the text *"date_time" "mph" "thunder_flag"*, representing three column names separated by spaces. First column *"date_time"* has the date time in the format "YYYY/MM/DD HH:MM:SS" (time zone UTC). Second column *kph* must have the standardized wind velocity in kilometers per hour. Third column *"thunder_flag"* must have a flag with one of two possible values *"nt"* or *"t"*, representing *non-thunderstorm* or *thunderstorm* classification respectively for the current wind value. For current research third column always had the flag *"nt"* (all stations, all rows). Separation of each column value must be *space* character. Values corresponding to *"date_time"* and *"thunder_flag"* must be enclosed by double quotes.

2. *estaciones*

```
estaciones <- read.delim(paste0(inputpath, "01 estaciones.txt"),
  header = FALSE, sep = "\t")
```

Variable *estaciones*, must point to a text file, named *'01 estaciones.txt'*, and stored inside *inputpath* folder, with the IDs of ISD stations to be processed. Each line of the file must have one ISD station ID. This file must have the list of stations to process, one station per row. If the intention is to run the process for only one station, the content of this file must have a single line, with the corresponding station ID. The code repeats the POT-PP procedure for each line of this file. Below, an example of its content is shown.

```
800740
800770
...
```

Be aware that for each line in this file *01 estaciones.txt*, one text file with wind historical time series information, named **raw_data_station_*.txt**, where * represents the same station ID, must be stored inside *inputpath* folder. See first element, lines up, in this list of variables.

3. *outputpath*

```
outputpath = "./isd/"
```

Variable *outputpath*, should point to the folder where all output files will be stored, after

running POT-PP process. The following list describes the main files to be generated, where * will be replaced by correspondent station ID.

- *FittedModel_*.pdf*: ISD POT-PP output graphics. See Table B.4.
- *fitted_model_result.xlsx*: Return levels ISD (all stations). See Table B.5.
- *raw_data_station_*_fitted.xlsx*: ISD POT-PP output parameters by station. See Table B.2.
- *raw_data_station_*_statistics.xlsx*: ISD POT-PP time (year, month, week) statistics by station. See Table B.3.

Table F.4 shows **input** and **output** files for ISD stations, after running POT-PP. See Annex B - Results - Digital Files, Table B.1, for a complete report of research files.

Table F.4: POT-PP ISD Input and Output Files

Folder Tree - Ftp Links	Description
pot_pp	POT-PP input and output files
|-isd	ISD files
| |-01 estaciones - 76 ok isd.txt	ISD list of used stations
| |-01 estaciones - isd - error.txt	One ISD station not working
| |-FittedModel_*.pdf	ISD POT-PP output graphics. See Table B.4.
| |-fitted_model_result.xlsx	Return levels ISD (all stations). See Table B.5.
| |-isd_stations.xlsx	ISD stations
| |-raw_data_station_*_fitted.xlsx	ISD POT-PP output parameters by station. See Table B.2.
| |-raw_data_station_*_statistics.xlsx	ISD POT-PP time (year, month, week) statistics by station. See Table B.3.
| |-maps	ISD raster and vector output data
| | |-rl_nh_h_combined_allcells4326.*	ISD stations shapefile with all return levels
| | |-combined	ISD final wind maps (non-hurricanes + hurricanes). See Table B.7.
| | |-nonhurricanes	ISD POT-PP non-hurricane wind maps. See Table B.7.
| |-raw_data	ISD non-thunderstorm time series (standardized)
|-correction_factors_isd_ideam.xlsx	Correction factors for standardization (ISD and IDEAM)

4. Linked R code

Main file **pot_pp_isd.r**, runs supplemental code using the R command *source*. Be sure that all R code files listed in next chunk of code, are pointing to the right location. See Table F.3 with the description of POT-PP ERA5 complementary R files. In the Table A.1 of the Annex A it is possible to see all R files related to this research.

```
#Library of POT-PP functions, including Dr Adam Pintar R Code (not
#published because this is copyrighted)
  source('./code/function_lib.R')
#Raw Data (whole dataset) Statistics and Send to CSV
  source('./code/stats_raw_data.r')
#Non Thunderstorm - Create Raw Data Statistics and Send to CSV
  source('./code/stats_raw_data_nt.r')
#Thunderstorm - Create Raw Data Statistics and Send to CSV
  source('./code/stats_raw_data_t.r')
#Write "t" to csv, but changing to one data per day (the maximum)
#Write "nt" to csv, but changing to one data per day (the maximum)
  source('./code/tnt_csv_1perday.r')
#Statistics and graphics for de-clustered non-thunderstorm
  source('./code/stats_graphs_dnt.r')
#Statistics and graphics for de-clustered thunderstorm
  source('./code/stats_graphs_dt.r')
#Plots for thunderstorm
  source('./code/plot_t.r')
```

```
#Plots for non-thunderstorm
  source('./code/plot_nt.r')
#Plots for non-thunderstorm and thunderstorm
  source('./code/plot_t_nt.r')
```

ISD Maps

Main output file of POT-PP analysis is *fitted_model_result.xlsx*, which contains return levels for typical return periods. Inside this Excel book, sheet *pp_pintar*, use columns *43 to 53*, to create extreme wind maps. Name of those columns are *nt_*_poissonprocessintfunc*, corresponding to non-thunderstorm return levels using Poisson Process Intensity Function, where * is replaced for 10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000, and 7000 years. A detailed description of all columns of *fitted_model_result.xlsx*, can be found in Table B.5.

To create non-hurricane maps using Kriging, from columns 43 to 53 of file *fitted_model_result.xlsx*, use R files inside folder *.../code/pot_pp/isd/maps/*, see Table F.3. For instance, to create non-hurricane map with return levels for 10 years MRI, use file *rl_10_nh.r*, for 50 MRI use *rl_50_nh.r*, and in the same way for other return periods. Once GeoTIFF images have been created with previous R files, use *plot_maps.r* to plot maps using `ggplot2` R package.

Inside files *rl_*_nh.r* different types of geostatistical related analysis are coded, mainly using `gstat`, `sf`, and `stars` R packages, see (E. Pebesma & Graeler, 2019), (E. Pebesma, 2019a), and (E. Pebesma, 2019b) respectively:

- Experimental semivariogram
- Directional semivariograms
- Theoretical semivariograms
- Graphics of semivariance models
- First-order trend modeling
- Graphics of semivariance models - first order trend
- Ordinary Kriging estimation
- Simple Kriging estimation
- Universal Kriging estimation
- Graphics of Kriging predictions & errors
- Cross validation - 'leave-one-out'
- Cross validation - 'N-fold'
- Cross validation - comparison statistics
- Final map - predictions
- Final map - errors
- IDW & cross validation

Procedure to create non-hurricane maps:

1. Run the procedure previously described in this manual, to execute POT-PP in ISD stations. File *fitted_model_result.xlsx* will be created with all return levels for different return periods.

2. From file *fitted_model_result.xlsx*, create file *rlisd.xlsx* with 20 columns according to Table F.5. See Table B.5 for columns descriptions. Keep same number of records in *rlisd.xlsx* compared with *fitted_model_result.xlsx*.

Table F.5: Creation of File rlisd.xlsx

File fitted_model_result.xlsx		File rlisd.xlsx	
Column ID	Column Name	Column ID	Column Name
1	id	1	id
2	t_thresh	2	t_thresh
3	t_mu_location	3	t_mu_location
4	t_psi_scale	4	t_psi_scale
5	nt_thresh	5	nt_thresh
6	nt_mu_location	6	nt_mu_location
7	nt_psi_scale	7	nt_psi_scale
8	distance_w	8	distance_w
9	station	9	station
43	nt_10_poissonprocessintfunc	10	nt_10_poissonprocessintfunc
44	nt_20_poissonprocessintfunc	11	nt_20_poissonprocessintfunc
45	nt_50_poissonprocessintfunc	12	nt_50_poissonprocessintfunc
46	nt_100_poissonprocessintfunc	13	nt_100_poissonprocessintfunc
47	nt_250_poissonprocessintfunc	14	nt_250_poissonprocessintfunc
48	nt_500_poissonprocessintfunc	15	nt_500_poissonprocessintfunc
49	nt_700_poissonprocessintfunc	16	nt_700_poissonprocessintfunc
50	nt_1000_poissonprocessintfunc	17	nt_1000_poissonprocessintfunc
51	nt_1700_poissonprocessintfunc	18	nt_1700_poissonprocessintfunc
52	nt_3000_poissonprocessintfunc	19	nt_3000_poissonprocessintfunc
53	nt_7000_poissonprocessintfunc	20	nt_7000_poissonprocessintfunc

3. Enable access to PostgreSQL database with ISD stations information. See Annex D *Database Storing*. Connection information to the spatial database is shown in Table F.6.

Table F.6: PostgreSQL Database Credentials

Credential	Value
dbname	winddata
host	localhost
pot	5432
user	user1
password	user1

4. Be sure that *rlisd.xlsx* file is stored in same folder as *rl_700_nh.r* file, this is, *.../code/pot_pp/isd/maps/*

5. Run file *rl_*_nh.r*, where * corresponds to the desired MRI year (10, 20, 50, 100, 250, 500, 700, 1000, 1700, 3000). For all the different geostatistical analysis implemented inside the file, the *spatial analysis expert* must review and interpret each partial result, and chose the best semivariance model and prediction map to use as final wind map.

6. Repeat previous step for all typical MRI

F.3.2 ERA5

Table F.7: R Code POT-PP ERA5. ftp://ftp.geocorp.co/windthesis/. User anonymous@geocorp.co (no password).

Folder Tree - Ftp Links	Description
code	Folder with R code. ALL CODE CREATED BY DR. ADAM PINTAR IS NOT PUBLISHED.
|-pot_pp	Folder with POT-PP R code. Based in Dr Adam Pintar code (respected copyright).
|-function_lib.r	POT-PP Functions. Author of de-clustering and thresholding functions is Dr Adam Pintar.
|-plot_nt.r	Plot non-thunderstorm graphics.
|-plot_t.r	Plot thunderstorm graphics.
|-plot_t_nt.r	Plot graphics with thunderstorm and non-thunderstorm data, in simultaneous.
|-stats_graphs_dnt.r	Statistics and graphics for non-thunderstorm de-clustered data.
|-stats_graphs_dt.r	Statistics and graphics for thunderstorm de-clustered data.
|-stats_raw_data.r	Statistics for raw data.
|-stats_raw_data_nt.r	Statistics for non-thunderstorm raw data.
|-stats_raw_data_t.r	Statistics for thunderstorm raw data.
|-tnt_csv_1perday.r	Create CSV (thunderstorm and non-thunderstorm) with one data (the maximum) per day.
|-era5	Folder with specific code for ERA5 data.
|-pot_pp_era5.r	POT-PP for ERA5 data. Based in Dr Adam Pintar code.
|-maps	Folder with specific code to calculate return levels and plot maps for ERA5 data.
|-return_levels.r	Calculate return levels for ERA5 data.
|-plot_maps.r	Join return levels to cells and plot ERA5 maps.

Table F.7 shows specific POT-PP R code used for ERA5 stations. Complete R code report can be found in Annex A - Research R Code - Digital Files, Table A.1.

Stations in ERA5 dataset correspond to its cell centers, from ID equal to 1 (top, left cell), counting next cells to the right, then bottom, up to 3381 (bottom, right cell). See Figure F.1.

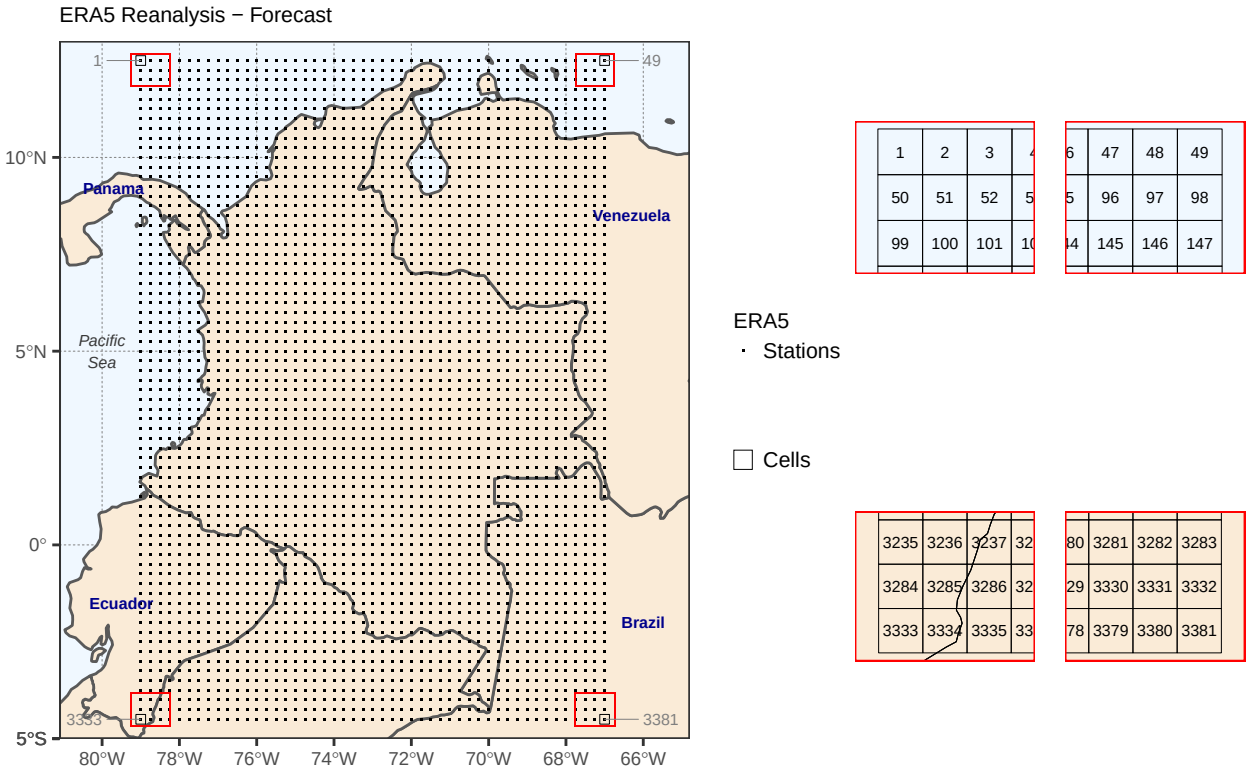

Figure F.1: ERA5 Cells and Stations

Procedure to run POT-PP in ERA5 stations:

1. Install R version 3.6.2, RStudio Version 1.2.5033, and 60 R packages dependencies (see chunk of code at beginning of this manual)
2. Verify files and variables according to descriptions and recommendations of following list (from 1 to 2), then execute file **pot_pp_era5.r**.

Files to run POT-PP in ERA5 stations are inside the folder *.../pot_pp/era5/*. Main file to run this process is **pot_pp_era5.r**, and inside it, next list of variables need to be configured.

1. *inputpathnetcdf*

```
inputpathnetcdf = "./data/"
```

Inside this folder *inputpathnetcdf*, file *"outfile_nc4c_zip9.nc"*, with variable 3-s wind gust *fg10*, must be stored. Be aware that ERA5 dataset does not need any type of standardization, as it comes standardized from source. See Annex C for a detailed procedure to download ERA5 information from Climate Data Storage - CDS `https://cds.climate.copernicus.eu/`. 2. *outputpath*

```
outputpath = "./era5/"
```

Variable *outputpath*, should point to the folder where all output files will be stored, after running POT-PP process. Following list describes main files to be generated, where * will be replaced by correspondent station ID.

- *FittedModel_*.pdf*: ERA5 POT-PP output graphics. See Table B.4.
- *fitted_model_result.xlsx*: Return levels ERA5 (all stations). See Table B.5.
- *raw_data_station_*_fitted.xlsx*: ERA5 POT-PP output parameters by station. See Table B.2.
- *raw_data_station_*_statistics.xlsx*: ERA5 POT-PP time (year, month, week) statistics by station. See Table B.3.

Table F.8: POT-PP ERA5 Input and Output Files

Folder Tree - Ftp Links	Description
pot_pp	POT-PP input and output files
|-era5	ERA5 files
|-FittedModel_*.pdf	ERA5 POT-PP output graphics. See Table B.4.
|-fitted_model_result.xlsx	Return levels ERA5 (all stations). See Table B.5.
|-raw_data_station_*_fitted.xlsx	ERA5 POT-PP output parameters by station. See Table B.2.
|-raw_data_station_*_statistics.xlsx	ERA5 POT-PP time (year, month, week) statistics by station. See Table B.3.
|-maps	ERA5 raster and vector output data
|-era5grid_left_right.*	ERA5 stations shapefile (IDs from left to right, then down)
|-era5grid_left_right_pol.*	ERA5 cells shapefile (IDs from left to right, then down)
|-era5grid_up_down.*	ERA5 stations shapefile (IDs from top to down, then right)
|-era5grid_up_down_pol.*	ERA5 cells shapefile (IDs from top to down, then right)
|-rl4326_points_nh_combined.*	ERA5 stations shapefile with all return levels
|-combined	ERA5 final wind maps (non-hurricanes + hurricanes). See Table B.6.
|-nonhurricanes	ERA5 POT-PP non-hurricane wind maps. See Table B.6.

Table F.8 shows **input** and **output** files for ERA5 stations, after running POT-PP. See Table B.1 in Annex B *Results - Digital Files* for a complete report of research files.

3. Linked R code

Main file **pot__pp__era5.r**, runs supplemental code using the R command *source*. Be sure that all R code files listed in next chunk of code, are pointing to the right location. See Table F.7 with the description of POT-PP ERA5 complementary R files. In Annex A - Research R Code - Digital Files, Table A.1 it is possible to see all R files related to this research.

```
#Library of POT-PP functions, including Dr Adam Pintar R Code (not
#published because this is copyrighted)
  source('./code/function_lib.R')
#Raw Data (whole dataset) Statistics and Send to CSV
  source('./code/stats_raw_data.r')
#Non Thunderstorm - Create Raw Data Statistics and Send to CSV
  source('./code/stats_raw_data_nt.r')
#Thunderstorm - Create Raw Data Statistics and Send to CSV
  source('./code/stats_raw_data_t.r')
#Write "t" to csv, but changing to one data per day (the maximum)
#Write "nt" to csv, but changing to one data per day (the maximum)
  source('./code/tnt_csv_1perday.r')
#Statistics and graphics for de-clustered non-thunderstorm
  source('./code/stats_graphs_dnt.r')
#Statistics and graphics for de-clustered thunderstorm
  source('./code/stats_graphs_dt.r')
#Plots for thunderstorm
  source('./code/plot_t.r')
#Plots for non-thunderstorm
  source('./code/plot_nt.r')
#Plots for non-thunderstorm and thunderstorm
  source('./code/plot_t_nt.r')
```