

# Projekt - Modelowanie zdarzeń ekstremalnych

Aleksander Mackiewicz-Kubiak

## Opis projektu

Projekt opiera się na badaniu i modelowaniu wartości ekstremalnych temperatur dla wybranej polskiej stacji meteorologicznej, oraz wybranych zależności z innymi stacjami jak i korelacji temperatury z innymi danymi meteorologicznymi. Dane do tego projektu pochodzą ze strony <https://danepubliczne.imgw.pl/datastore>. Moja analiza dotyczy wsi Korbielów, id 249190440, z województwa śląskiego na pograniczu z Słowacją. Pierwszą częścią projektu jest wyestymowanie na trzy sposoby 20 i 50 letniego poziomu zwrotu temperatury dla każdej z pór roku. Celem drugiej części będzie dopasowywanie i zwizualizowanie kopuł między moją stacją, a wybranymi innymi stacjami oraz analiza zmiennych z nimi powiązanych takich jak współczynnik Kendalla oraz współczynniki zależności ekstremalnej. Celem trzeciej, ostatniej części jest badanie struktur C-vine i D-vine służących do modelowania rozkładów wielowymiarowych na podstawie czterech danych meteorologicznych dla stacji Korbielów wraz z dopasowaniem modelu regresji kwantylowej.

## Część pierwsza

### Przygotowanie danych

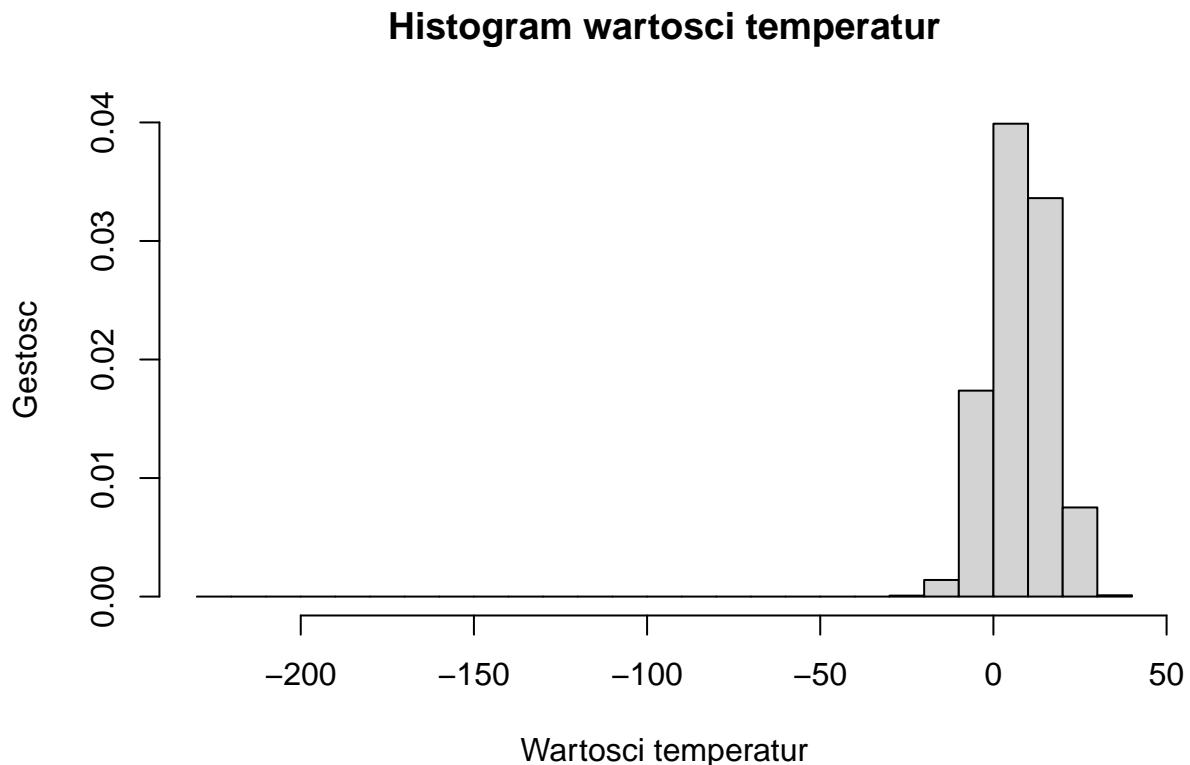
Pierwszy krok to wczytanie pliku z danymi. Plik zawiera dane dla wszystkich stacji, jednak mnie interesuje tylko jedna o numerze X249190440, więc tworze osobny dataframe tylko z temperaturami z tej stacji i dla uproszczenia dalszych komend zmieniam nazwę kolumny z temperaturami na "wartosci\_temp":

```
##   wartosci_temp      data
## 1      -4.56 2008-01-01 00:00
## 2      -4.70 2008-01-01 00:10
## 3      -4.59 2008-01-01 00:20
## 4      -4.57 2008-01-01 00:30
## 5      -4.67 2008-01-01 00:40
## 6      -4.65 2008-01-01 00:50
```

Następnie tworzę kolejnego dataframe'a, tym razem z podziałem kolumny z datami na trzy osobne kolumny: rok, miesiąc i dzień:

```
##   wartosci_temp year mth day
## 1      -4.56 2008    1    1
## 2      -4.70 2008    1    1
## 3      -4.59 2008    1    1
## 4      -4.57 2008    1    1
## 5      -4.67 2008    1    1
## 6      -4.65 2008    1    1
```

Sprawdzę teraz histogram wartości temperatur

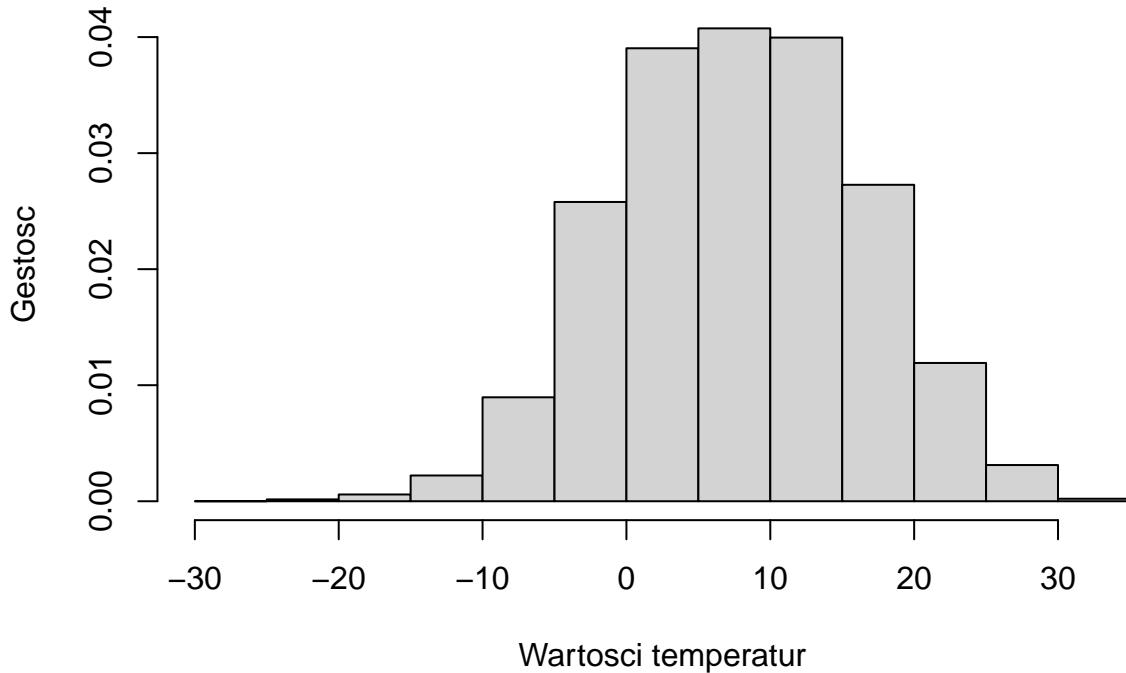


Histogram jest dziwny, wartości temperatur sięgają do -200 stopni , co jest niemożliwe. Sortując kolumnę z wartościami temperatur można odkryć, że istnieje pojedyncza wartość -222, która jest najprawdopodobniej błędem przy wpisywaniu:

```
##      wartosci_temp year mth day
## 7871      -222.94 2008    2   24
```

Zatem usuwam ją i ponownie sprawdzam histogram:

## Histogram wartosci



Histogram wygląda o wiele lepiej i naturalniej. Sprawdzę jeszcze średnia, wariancje i odchylenie standardowe wartości:

```
## [1] "Średnia = 7.836"  
## [1] "Wariancja = 71.608"  
## [1] "Odchylenie standardowe = 8.462"
```

Celem jest wyestymowanie poziomów zwrotów dla sezonów, więc podzielę dane na pory roku na podstawie miesięcy:

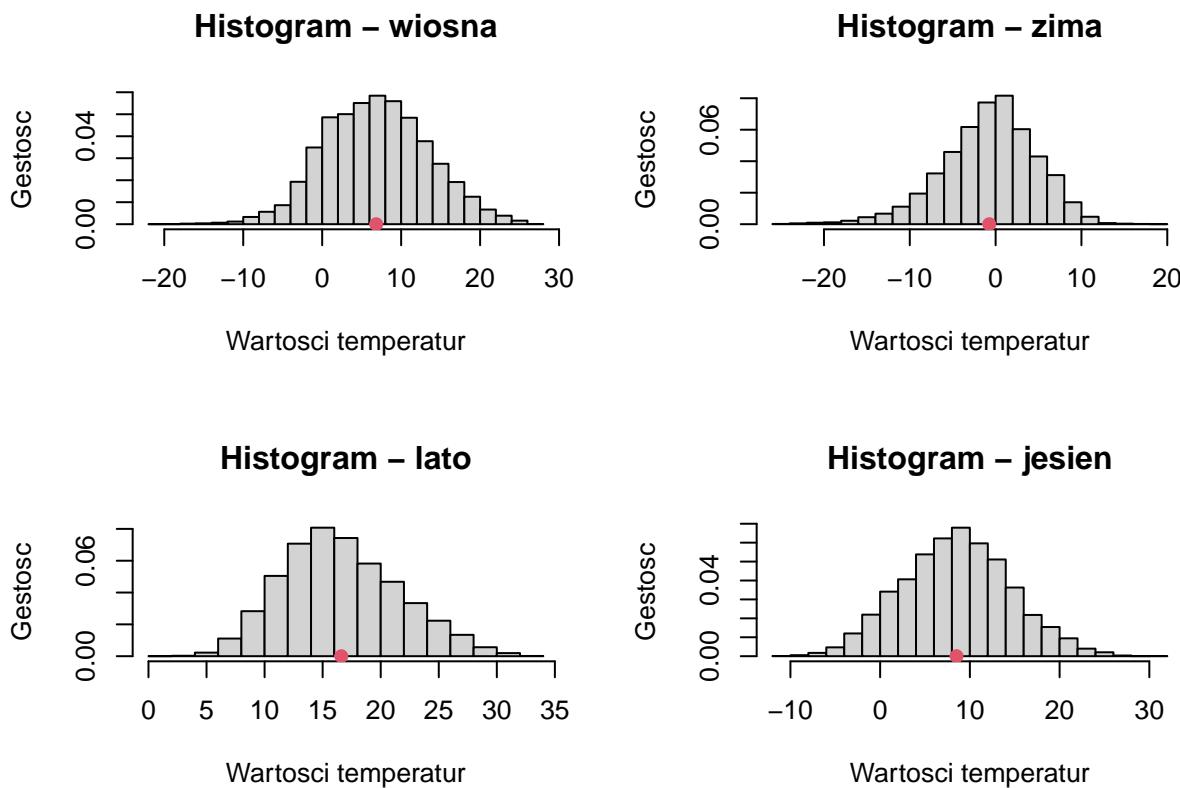
```
## List of 4  
## $ wiosna:'data.frame': 211968 obs. of 4 variables:  
##   ..$ wartosci_temp: num [1:211968] 5.15 5.3 5.44 5.59 5.49 ...  
##   ..$ year       : int [1:211968] 2008 2008 2008 2008 2008 ...  
##   ..$ mth        : int [1:211968] 3 3 3 3 3 3 3 3 3 ...  
##   ..$ day        : int [1:211968] 1 1 1 1 1 1 1 1 1 ...  
## $ zima  :'data.frame': 207936 obs. of 4 variables:  
##   ..$ wartosci_temp: num [1:207936] -4.56 -4.7 -4.59 -4.57 ...  
##   ..$ year       : int [1:207936] 2008 2008 2008 2008 2008 ...  
##   ..$ mth        : int [1:207936] 1 1 1 1 1 1 1 1 1 ...  
##   ..$ day        : int [1:207936] 1 1 1 1 1 1 1 1 1 ...  
## $ lato  :'data.frame': 211968 obs. of 4 variables:  
##   ..$ wartosci_temp: num [1:211968] 13.8 13.9 14.6 14.6 14.5 ...
```

```

##   ..$ year      : int [1:211968] 2008 2008 2008 2008 2008 2008 2008 2008 2008 2008 ...
##   ..$ mth       : int [1:211968] 6 6 6 6 6 6 6 6 6 6 ...
##   ..$ day       : int [1:211968] 1 1 1 1 1 1 1 1 1 1 ...
## $ jesien:'data.frame': 209664 obs. of  4 variables:
##   ..$ wartosci_temp: num [1:209664] 8.57 8.39 8.46 8.48 8.49 8.43 8.48 8.46 8.7 8.6 ...
##   ..$ year       : int [1:209664] 2008 2008 2008 2008 2008 2008 2008 2008 2008 2008 ...
##   ..$ mth       : int [1:209664] 9 9 9 9 9 9 9 9 9 9 ...
##   ..$ day       : int [1:209664] 1 1 1 1 1 1 1 1 1 1 ...

```

Histogramy dla tak podzielonych danych:



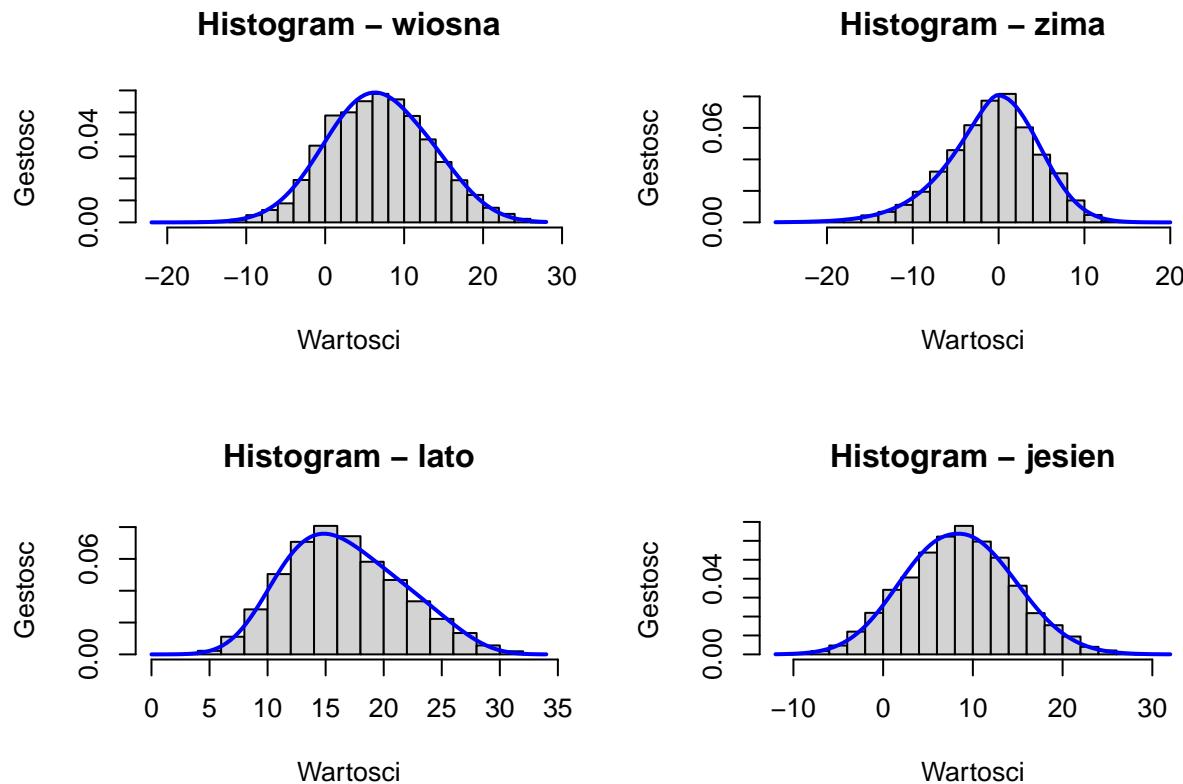
## Metoda 1

### Opis metody

Pierwszą metodą będzie zwykłe dopasowanie rozkładów do danych z każdego sezonu. W tym celu skorzystam z pakietu gamlls, który służy właśnie do dopasowywania rozkładów do danych empirycznych i wybierze najlepiej pasujący rozkład. Ocena dopasowania rozkładu będzie przy pomocy kryterium AIC - im mniejsza wartość AIC tym lepiej dopasowany rozkład. Następnie dla tak dopasowanych rozkładów wylicze poziomy zwrotu jako kwantyle tego rozkładu, pamiętając że używane tutaj dane to nie dane roczne a 10-minutowe, co wpływa na to który kwantyl będę wyznaczać.

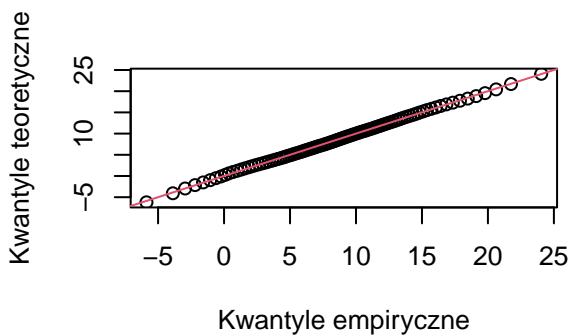
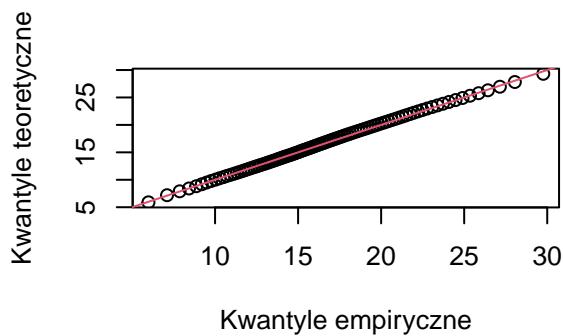
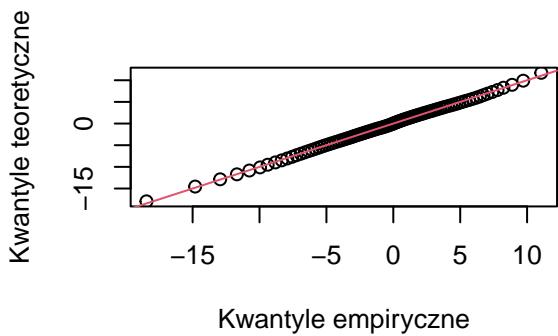
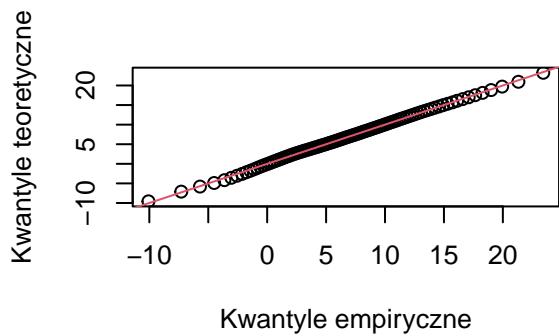
## Analiza

Przy użyciu funkcji `fitDist` z pakietu `gamlss` do każdej pory roku dopasowuje najlepszy rozkład względem kryterium AIC, a następnie wszystkie wyniki zapisuje w liście. By przekonać się o odpowiednim dopasowaniu tych rozkładów, możemy je nanieść na histogramy temperatur:



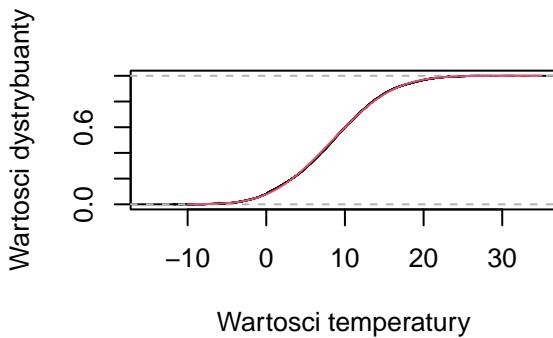
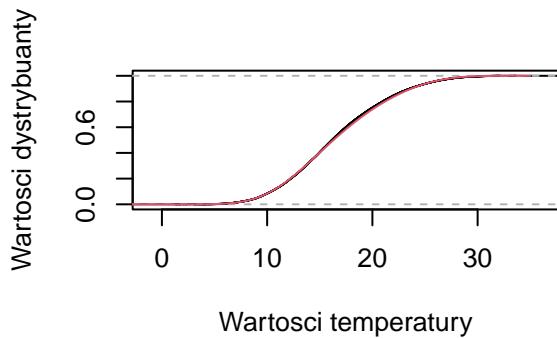
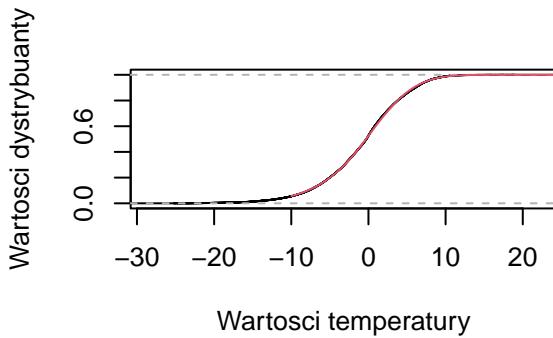
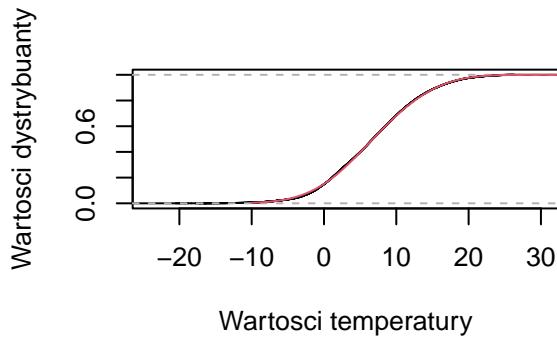
Po wykresach widać, że rozkłady są odpowiednie. By się jeszcze mocniej upewnić, sprawdzę wykresy Q-Q (wykresy kwantyl-kwantyl porównujące kwantyle empiryczne z danych do tych z rozkładu) dla każdej pory roku:

## Wykresy Kwantyl–Kwantyl



Wszystkie cztery wykresy wychodzą wręcz perfekcyjnie. Jako finalną wizualizację wyświetle jeszcze porównanie dystrybuanty empirycznej i teoretycznej (która patrząc na wykresy Q-Q też będzie się dobrze pokrywać):

## Wykresy dystrybuant



Skoro już jestem pewien dobrego dopasowania rozkładów, mogę przejść do wyliczenia poziomów zwrotu dla każdego sezonu. Chce wyznaczyć 20 i 50 letni poziom zwrotu. Moje dane jednak nie są roczne, a 10-minutowe, zatem nie mogę wziąć kwantylów 1/20 i 1/50, a potrzebuje wyliczyć ilość dni (czyli wartości) w ciągu tych lat dla każdego sezonu. Ilości dni wyglądają tak:

```
## wiosna  zima   lato jesien
## 13248  12960  13248  13104
```

Z metody nr 1 20 letnie poziomu zwrotu wyglądają tak:

```
## [1] "wiosna"
## [1] 31.96494
##
## [1] "zima"
## [1] 20.02608
##
## [1] "lato"
## [1] 34.53588
##
## [1] "jesien"
## [1] 35.24815
```

A 50 letnie tak:

```
## [1] "wiosna"
```

```
## [1] 33.83817
##
## [1] "zima"
## [1] 20.85248
##
## [1] "lato"
## [1] 35.01227
##
## [1] "jesien"
## [1] 36.3638
```

## Metoda 2

### Opis metody

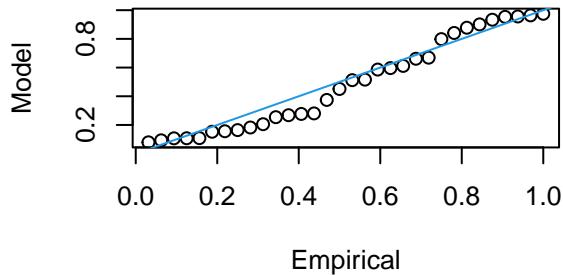
Drugą metodą będzie metoda maksimów blokowych (ang. Block maxima method). Polega ona na podziale naszych danych na równe przedziały, np. roczne, a następnie wyznaczeniu maksimów z każdego przedziału. Zgodnie z teorią zdarzeń esktremalnych jest ograniczona ilość rozkładów jakie takie maksima mogą przyjmować oraz że istnieje rozkład GEV (ang. generalized extreme value distribution), który jest uogólnieniem wszystkich tych rozkładów. Czyli po wyznaczeniu maksimów rocznych dla danych, dopasuje je do rozkładu GEV a następnie wyznacze kwantyle 20 i 50 letniego poziomu zwrotu. Ponieważ maksima użyte do dopasowania rozkładu będą roczne, to x-letni poziom zwrotu będzie zwyczajnym kwantylem rzędu  $1-1/x$ , czyli będą to kwantyle 0.95 i 0.98. Posłużą mi tutaj trzy pakiety: evir, ismev oraz fExtremes, które pozwolą mi dopasować rozkład GEV i go ocenić na podstawie różnych wykresów diagnostycznych (co jest ważne przy małej ilości danych na których będę tutaj operować).

### Analiza

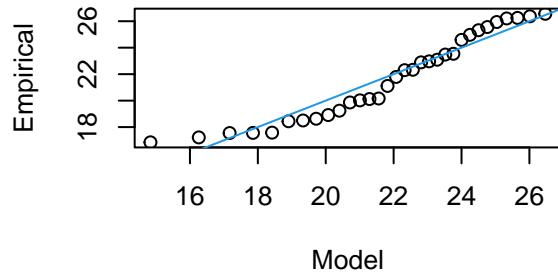
Dla każdego sezonu wyznaczam maksima blokowe(roczne), a następnie dopasowuję rozkład GEV i wyświetlам jego wykresy diagnostyczne:

wiosna

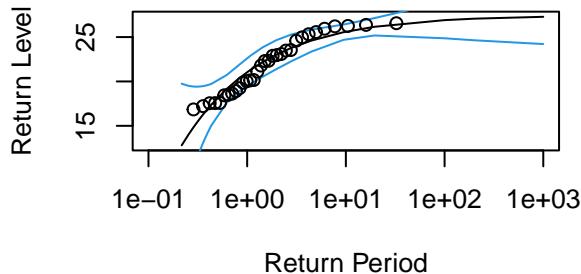
**Probability Plot**



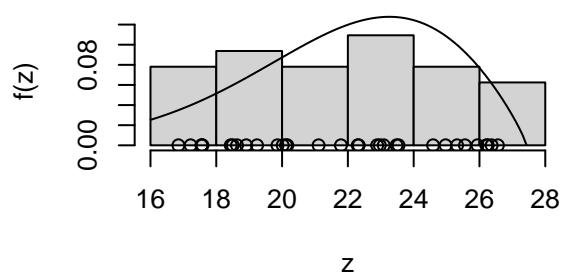
**Quantile Plot**



**Return Level Plot**

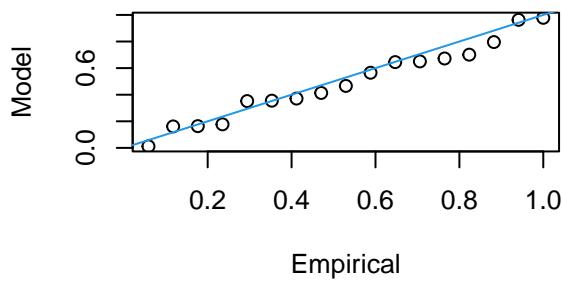


**Density Plot**

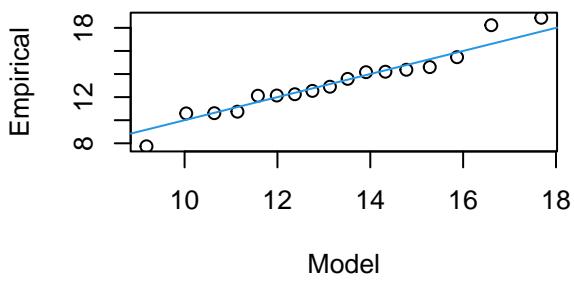


**zima**

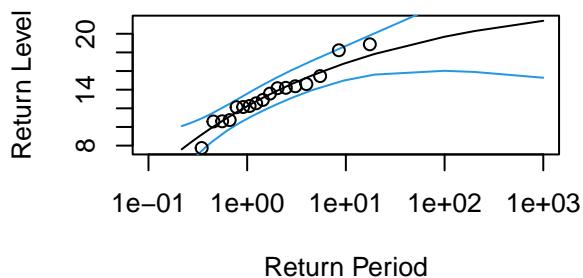
**Probability Plot**



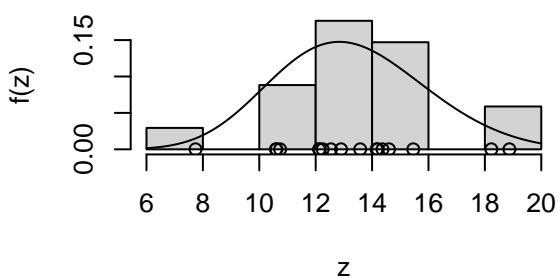
**Quantile Plot**



**Return Level Plot**

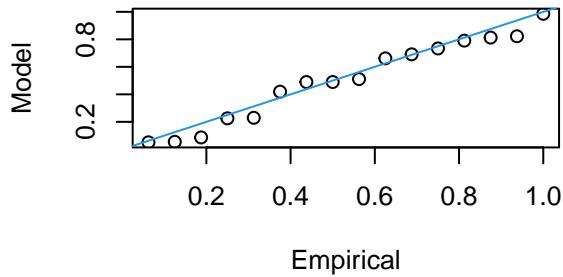


**Density Plot**

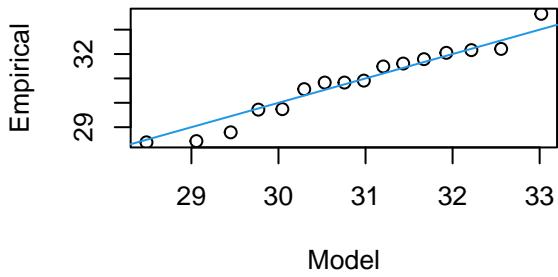


**lato**

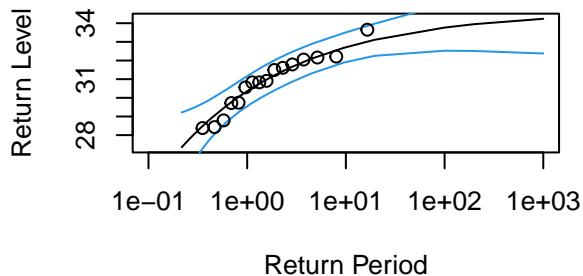
**Probability Plot**



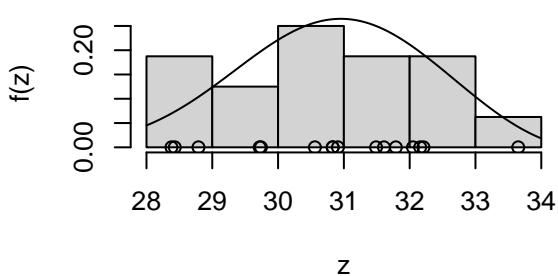
**Quantile Plot**



**Return Level Plot**

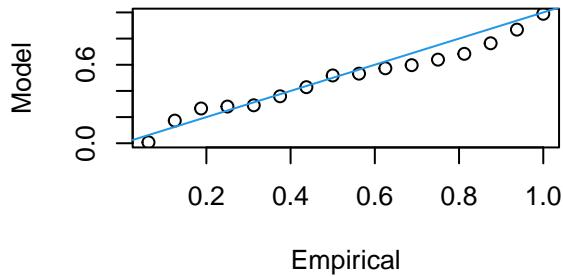


**Density Plot**

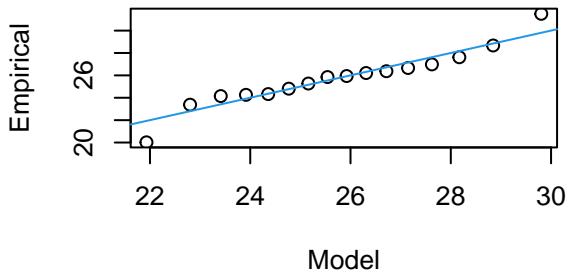


# jesien

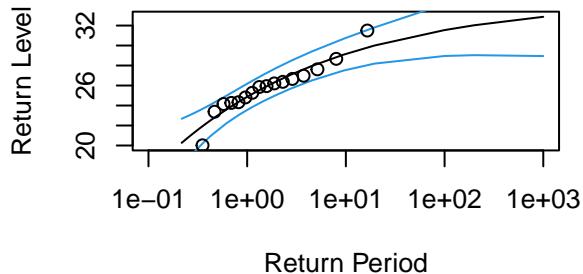
**Probability Plot**



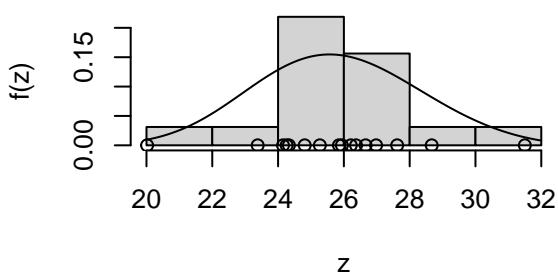
**Quantile Plot**



**Return Level Plot**



**Density Plot**



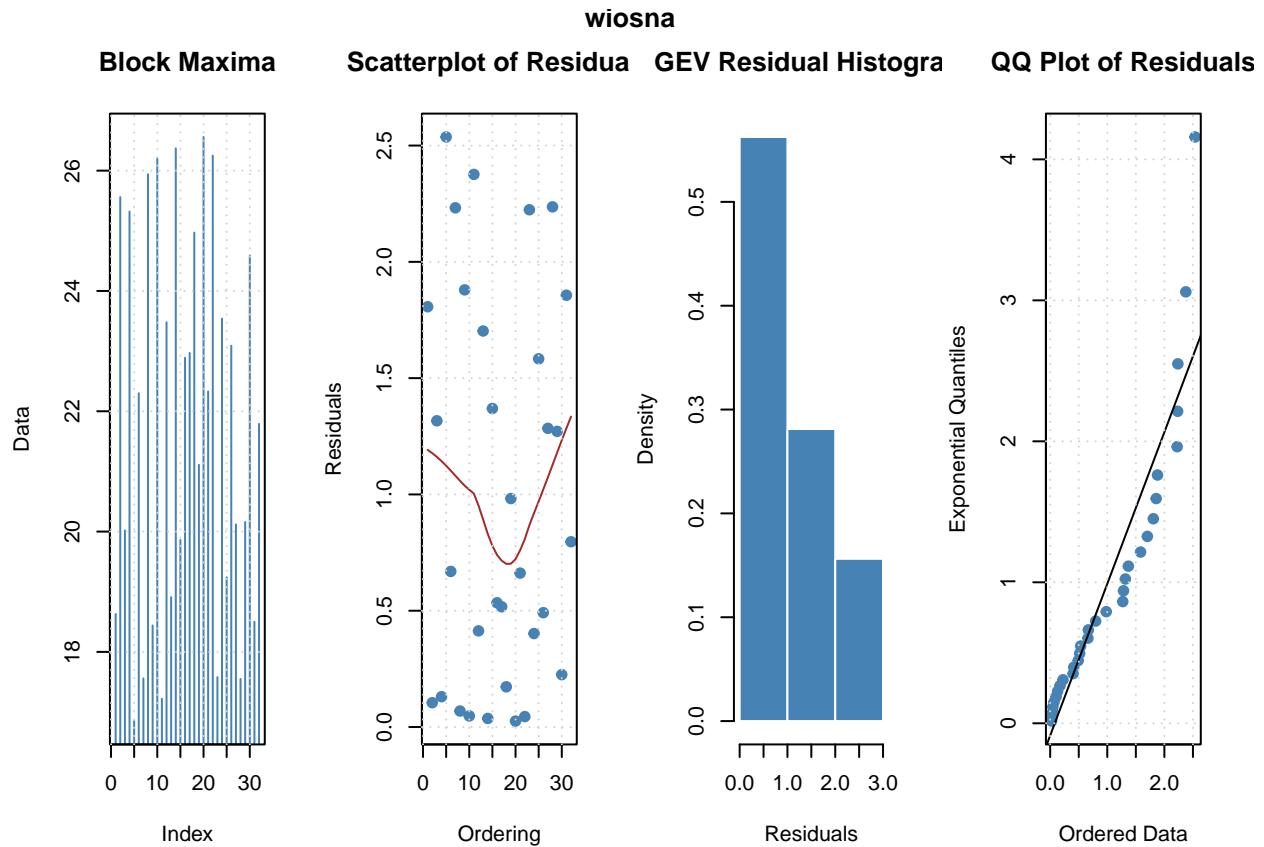
Wykresy, jak na dosyć skąpą ilość danych wyglądają satysfakcyjnie. Jednak dla pewności użyje drugiego pakietu fExtremes, i tam wbudowanego podsumowania dopasowania GEV, które zawiera inne wykresy diagnostyczne:

```
##  
## Title:  
##   GEV Parameter Estimation  
##  
## Call:  
##   fExtremes::gevFit(x = data, block = k_sezony[[i]])  
##  
## Estimation Type:  
##   gev mle  
##  
## Estimated Parameters:  
##           xi         mu        beta  
## -0.5427879 21.0480814  3.4656842  
##  
## Standard Deviations:  
##           xi         mu        beta  
## 0.3028943 0.7982739  0.7719286  
##  
## Log-Likelihood Value:  
##   80.95081  
##  
## Type of Convergence:
```

```

##    0
##
## Description
##   Wed Jun 12 15:29:36 2024

```



```

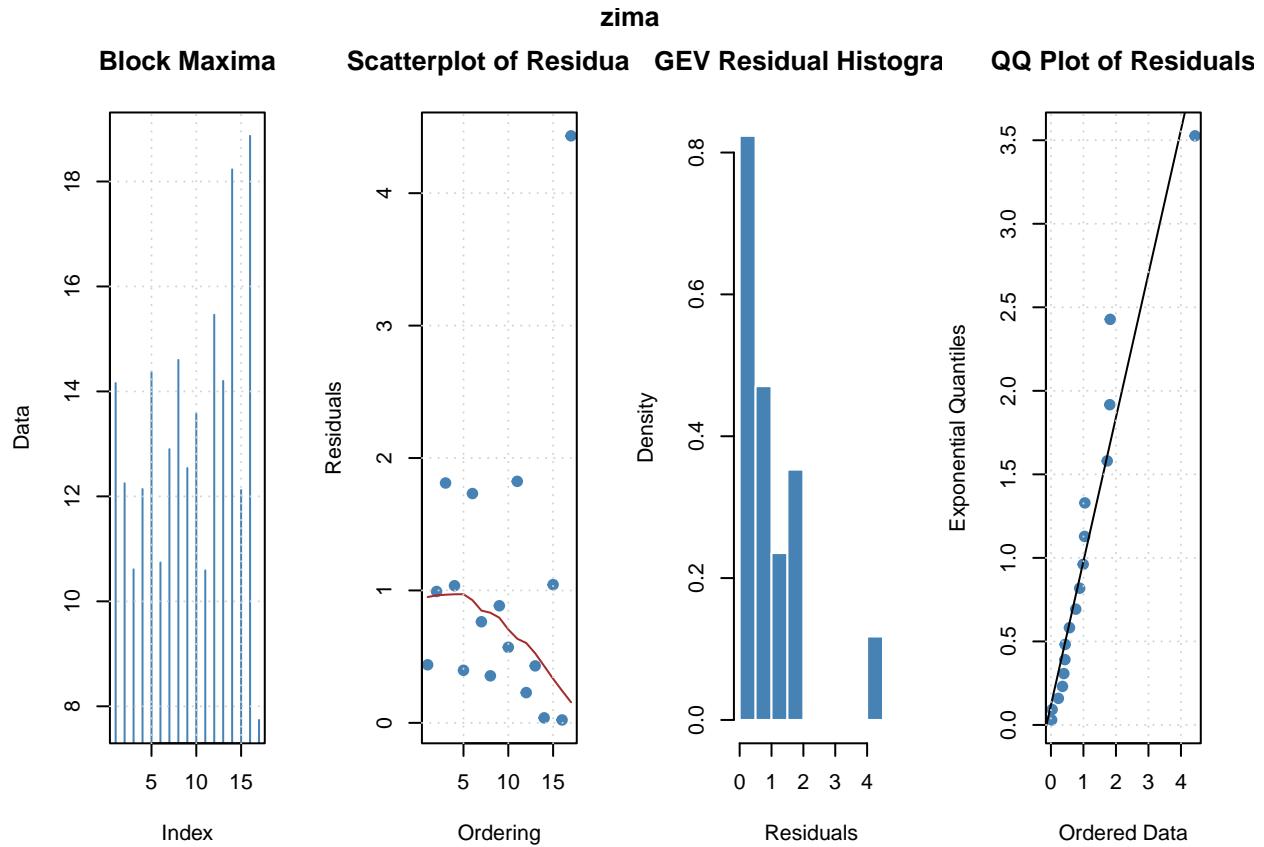
##
## Title:
##   GEV Parameter Estimation
##
## Call:
##   fExtremes::gevFit(x = data, block = k_sezony[[i]])
##
## Estimation Type:
##   gev mle
##
## Estimated Parameters:
##           xi         mu        beta
## -0.2162523 12.2302026  2.5559313
##
## Standard Deviations:
##           xi         mu        beta
## 0.1568556  0.6852178  0.4703567
##
## Log-Likelihood Value:
##   40.66781
##

```

```

## Type of Convergence:
##   0
##
## Description
##   Wed Jun 12 15:29:36 2024

```



```

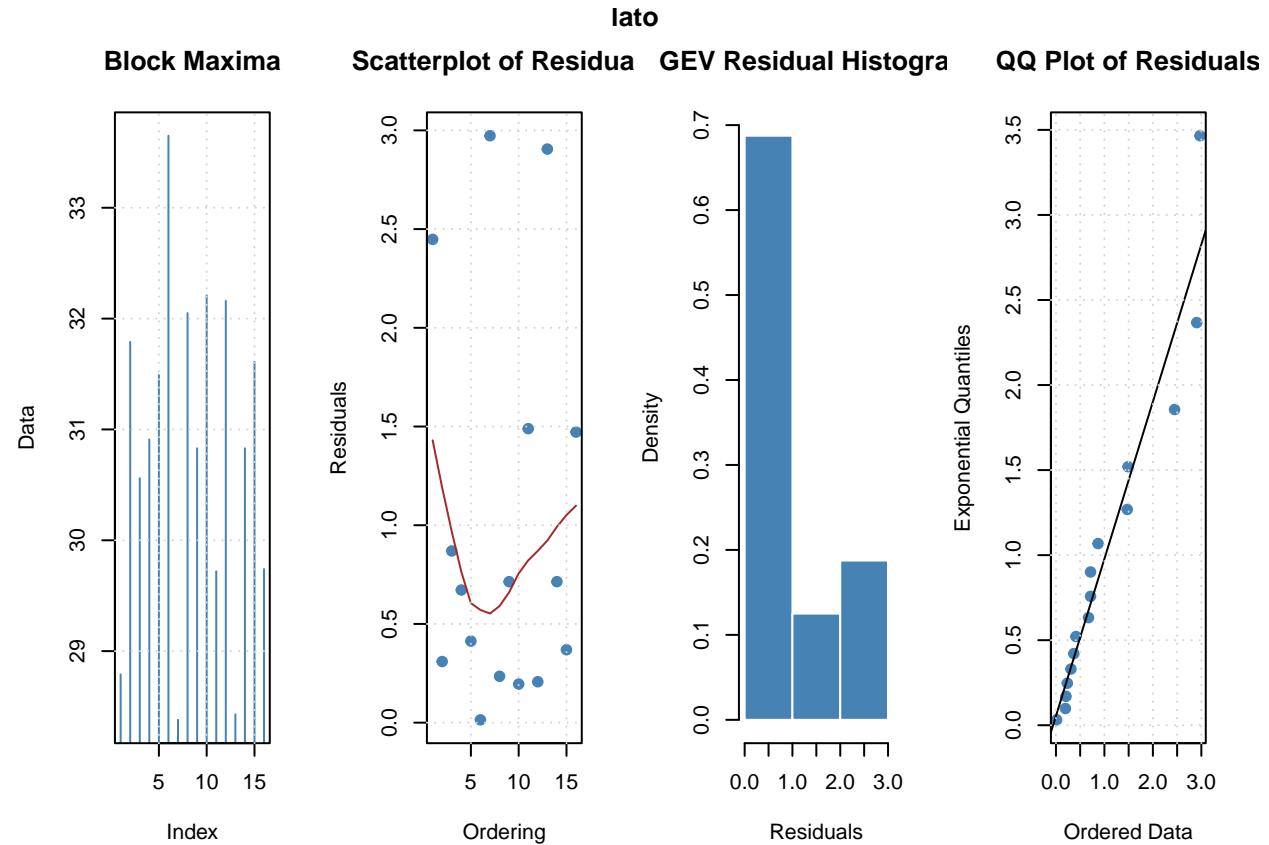
##
## Title:
##   GEV Parameter Estimation
##
## Call:
##   fExtremes::gevFit(x = data, block = k_sezony[[i]])
##
## Estimation Type:
##   gev mle
##
## Estimated Parameters:
##     xi      mu      beta
## -0.350019 30.356740  1.490294
##
## Standard Deviations:
##     xi      mu      beta
## 0.1671033 0.4114295 0.2971741
##
## Log-Likelihood Value:
##   28.44493

```

```

## 
## Type of Convergence:
##   0
##
## Description
##   Wed Jun 12 15:29:36 2024

```



```

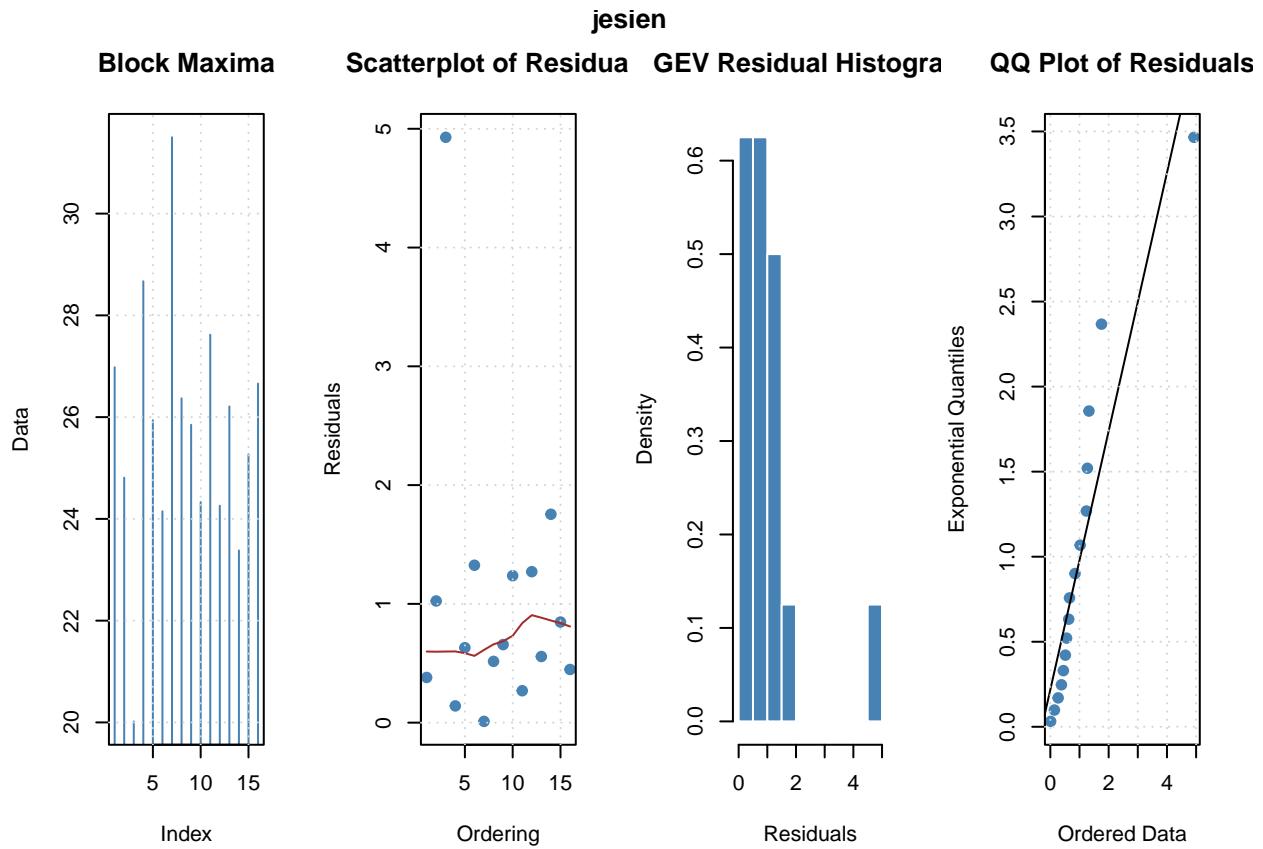
## 
## Title:
##   GEV Parameter Estimation
## 
## Call:
##   fExtremes::gevFit(x = data, block = k_sezony[[i]])
## 
## Estimation Type:
##   gev mle
## 
## Estimated Parameters:
##   xi      mu      beta
## -0.254911 24.869538  2.464108
## 
## Standard Deviations:
##   xi      mu      beta
## 0.1229932 0.6676864 0.4406820
## 
## Log-Likelihood Value:

```

```

##    37.10579
##
## Type of Convergence:
##    0
##
## Description
##    Wed Jun 12 15:29:36 2024

```



Wykresy ponownie nie są idealne, ale mogły być o wiele gorsze jak na ilość danych na których tutaj operujemy. Zatem akceptuje te wykresy i jakość dopasowania rozkładu, więc pozostaje mi tylko wyliczenie progów zwrotu poprzez kwantyle dopasowanego rozkładu GEV. Z metody 2 20 letnie poziomy zwrotu prezentują się tak:

```

## [1] "wiosna"
## [1] 25.54122
##
## [1] "zima"
## [1] 16.28461
##
## [1] "lato"
## [1] 33.10927
##
## [1] "jesien"
## [1] 28.62258

```

A 50 letnie tak:

```

## [1] "wiosna"
## [1] 26.18874
##
## [1] "zima"
## [1] 17.18688
##
## [1] "lato"
## [1] 33.52822
##
## [1] "jesien"
## [1] 29.53696

```

## Metoda 3

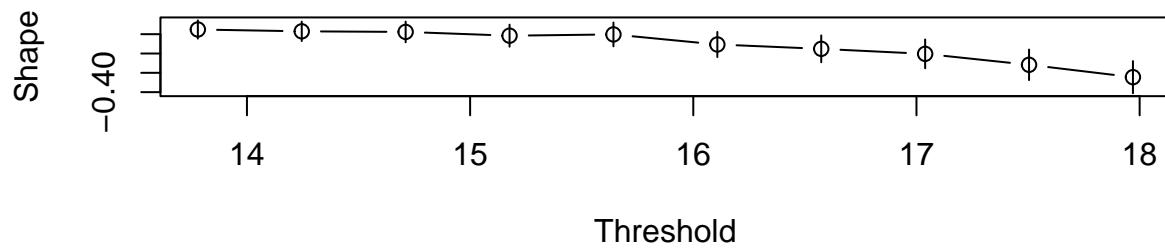
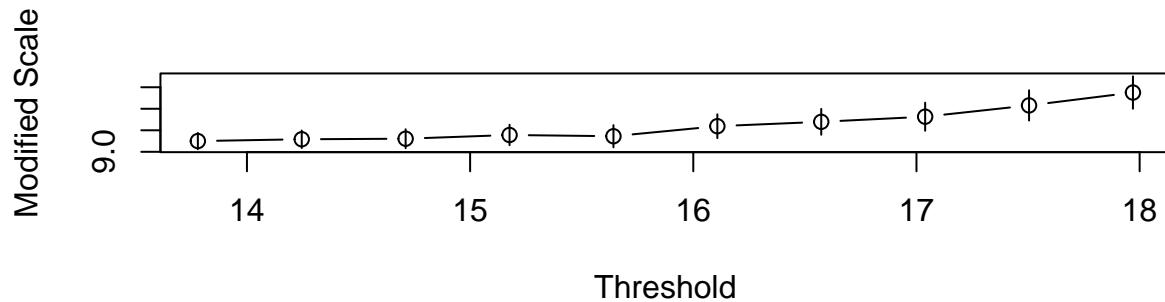
### Opis metody

Trzecią metodą będzie metoda przekroczeń progu (ang. Peak Over Threshold). Polega ona na badaniu danych ponad pewnym progiem, i na ich podstawie wyznaczenia poziomów zwrotów. Ponownie, zgodnie z teorią zdrażeń ekstremalnych, dla odpowiednio dobranego progu dane te można modelować za pomocą rozkładu GPD (ang. generalized Pareto distribution). Zatem wpierw zajmę się wyznaczeniem odpowiedniego progu u dla każdego sezony, wyznaczeniem danych pna tym progiem, dopasowaniem do nich rozkładu GPD i końcowo wyznaczeniem naszych kwantylów czyli poziomów zwrotu. Ponownie, jak w przypadku metody 1, dane to maksima 10-minutowe, zatem skorzystam z takich samych rzędów kwantylów co wtedy.

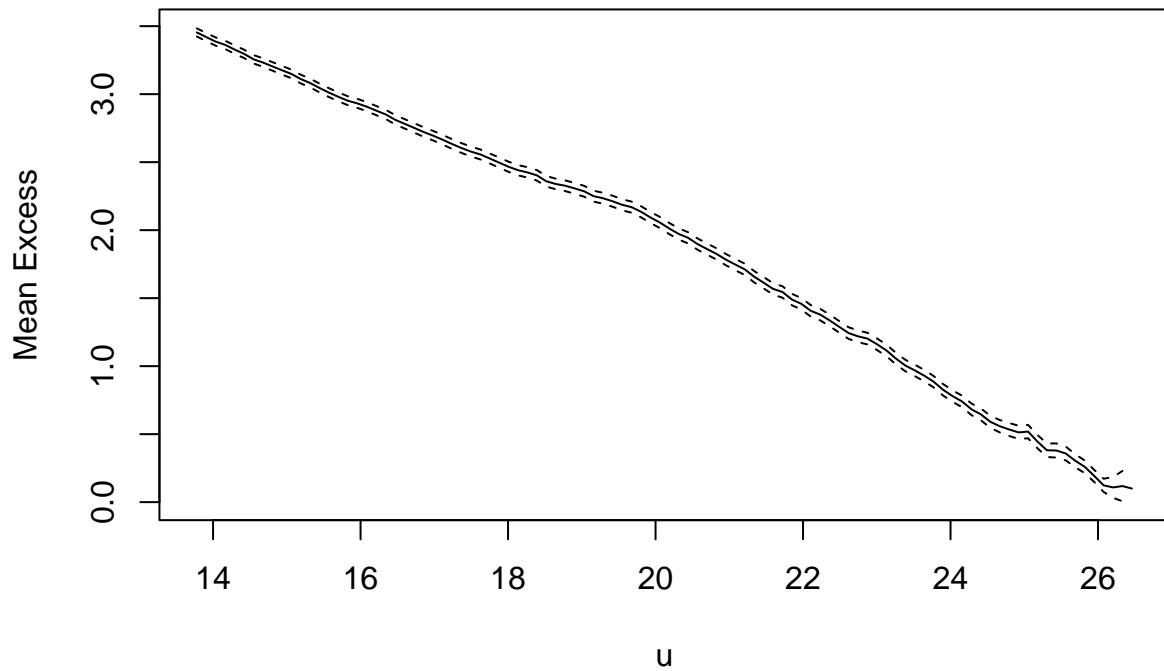
### Analiza

Pierwszym krokiem jest dopasowanie parametru u, który będzie progiem. Do tego potrzebujemy trzech wykresów bazujących na kwantylach 85 i 95 z oryginalnych danych. Pierwsze dwa wykresy to wykresy odpowiednio parametrów beta (modified scale) i xi (shape), a trzeci wykres to wykres średniej przekroczenia progu (Mean Excess):

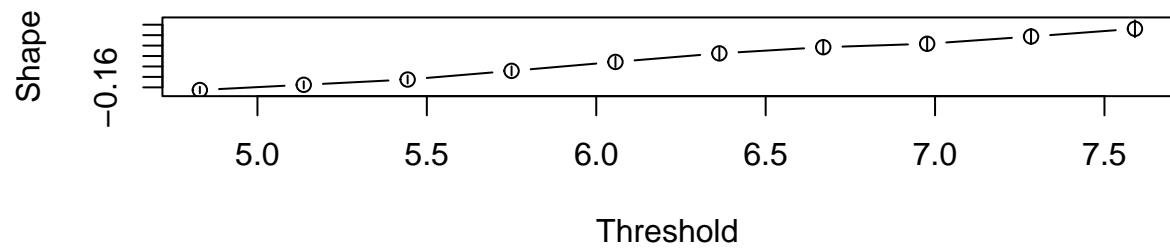
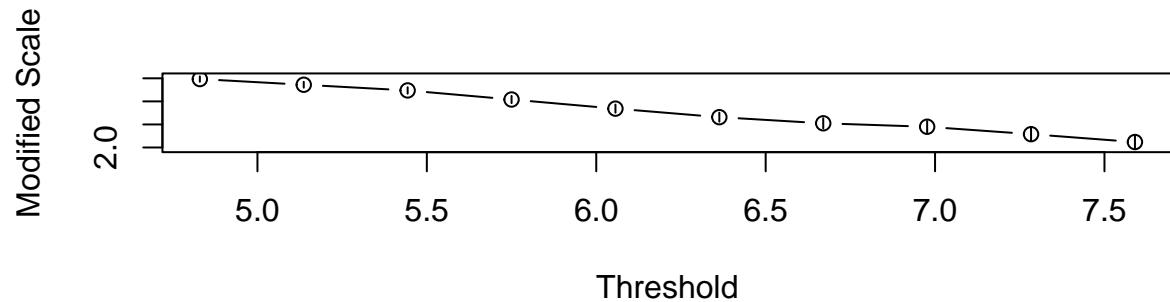
## wiosna



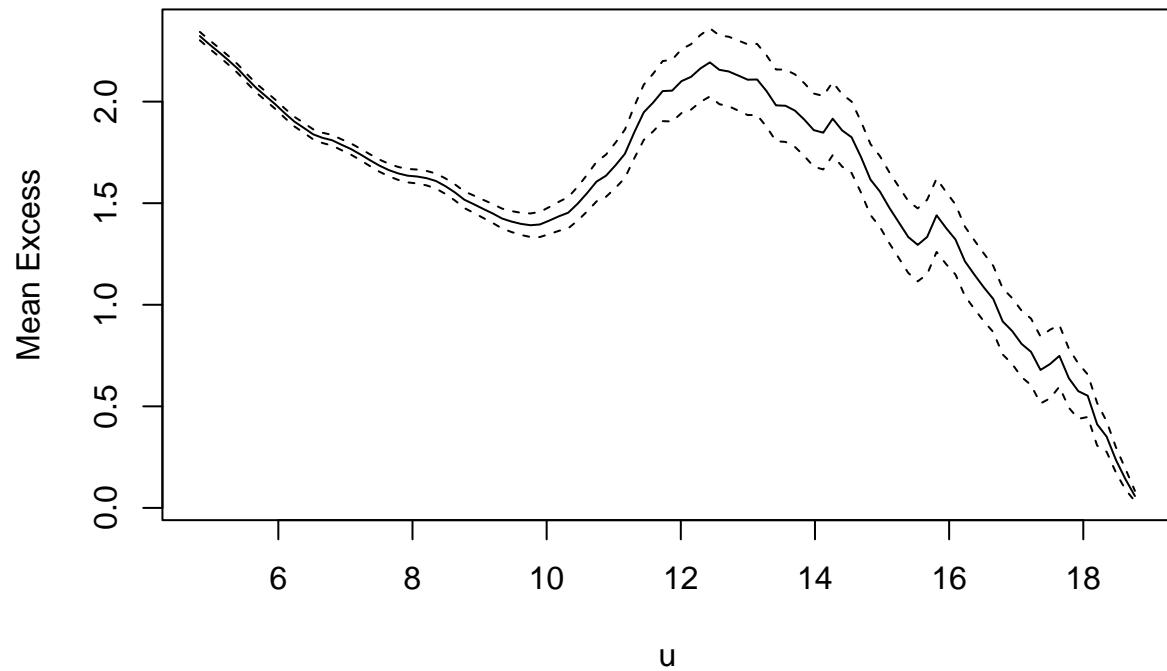
wiosna



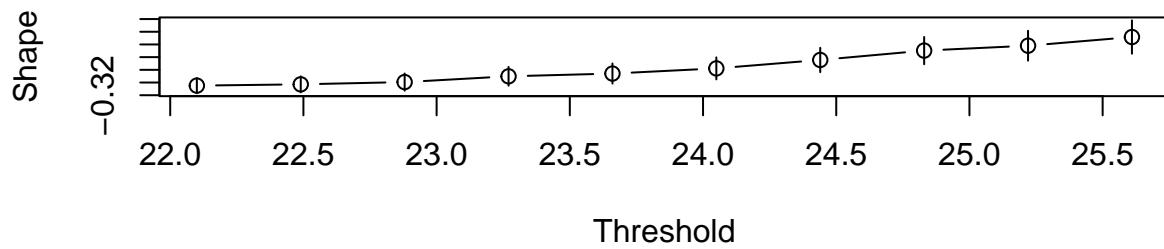
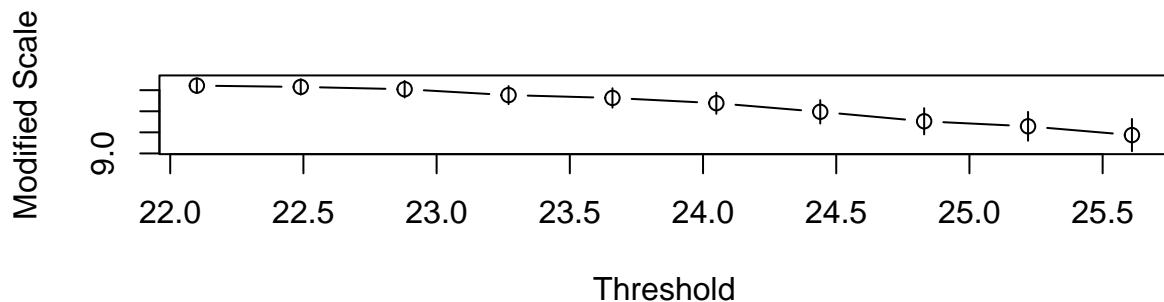
**zima**



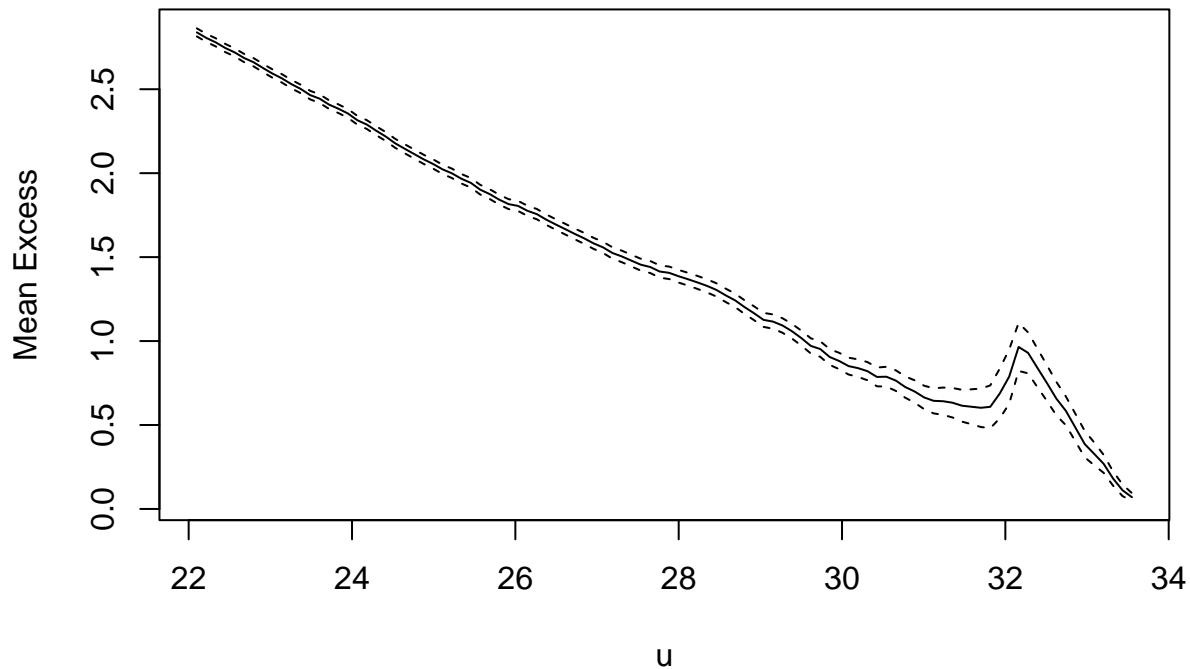
**zima**



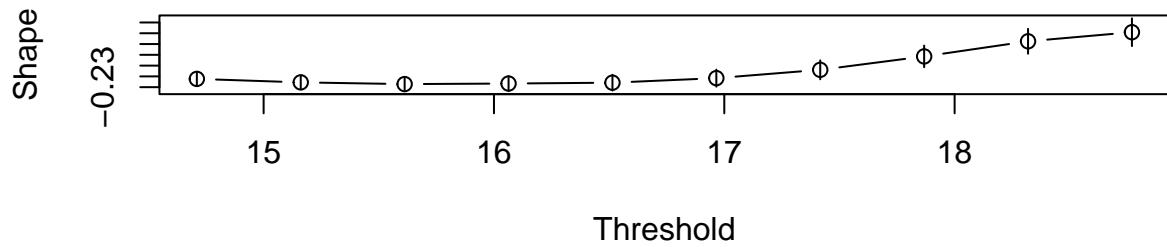
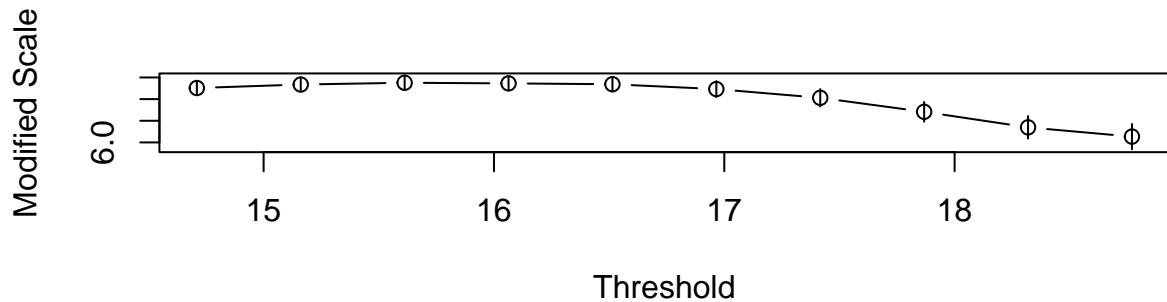
## **lato**



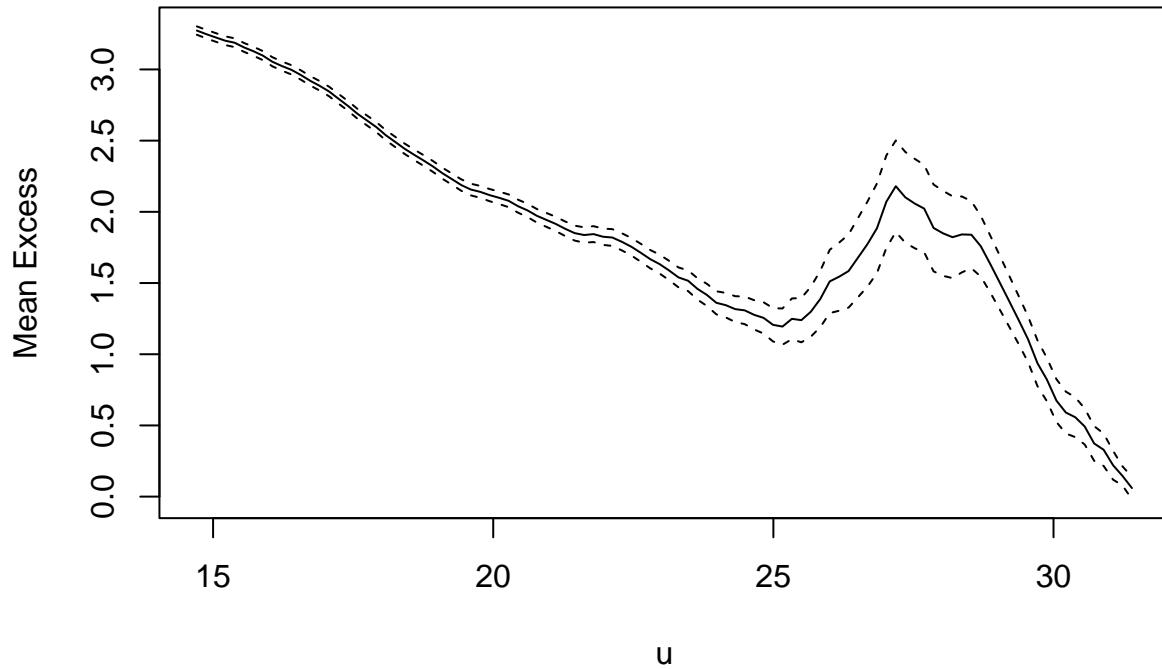
**lato**



## **jesien**



## jesien



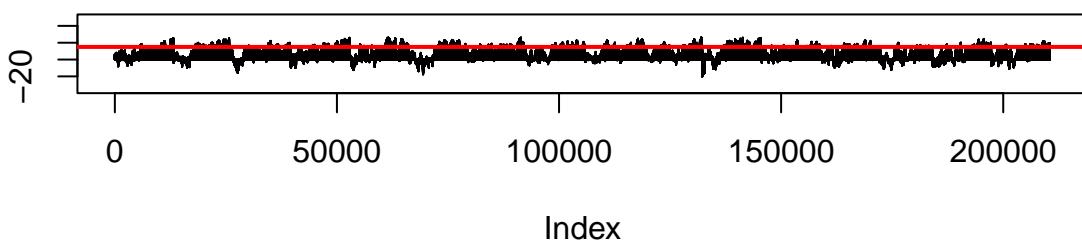
Z pierwszych dwóch wykresów szukamy takich wartości na osi x, dla których oba te wykresy są stałe. Na drugim wykresie ponownie szukamy wartości na osi x, tym razem takich dla których wykres przypomina funkcję liniową. Oba te warunki łączymy, dostając estymacje progu u. Dla moich wartości u tworzę dedykowaną listę i z wykresów odczytuje estymowane wartości:

```
## $wiosna
## [1] 15
##
## $zima
## [1] 6
##
## $lato
## [1] 24
##
## $jesień
## [1] 16
```

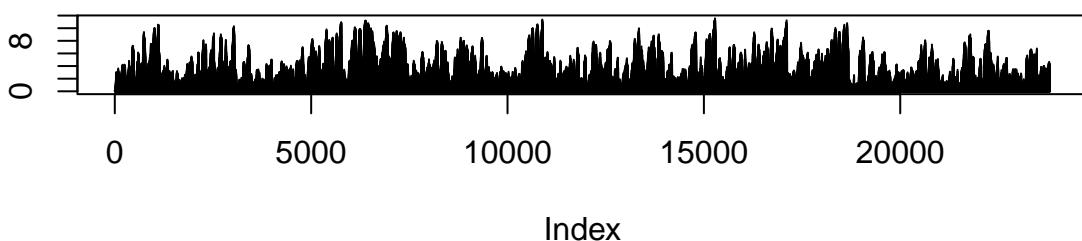
Dla ustalonych wartości u możemy zwizualizować dane z zaznaczonym progiem oraz wyświetlić dane do których będziemy dopasowywać rozkład GPD, czyli wartości ponad moim progiem:

### wiosna

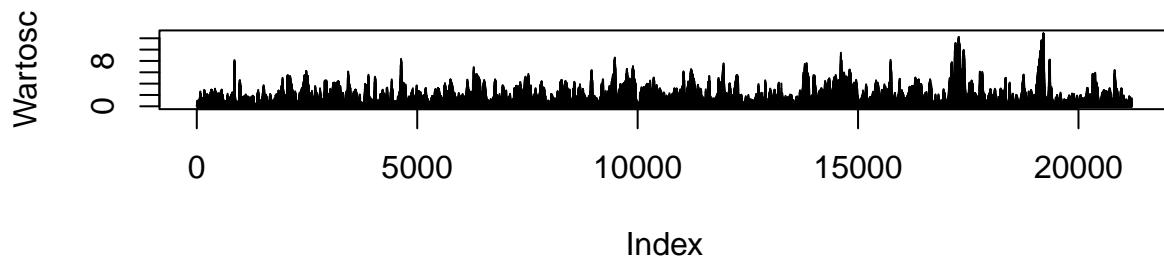
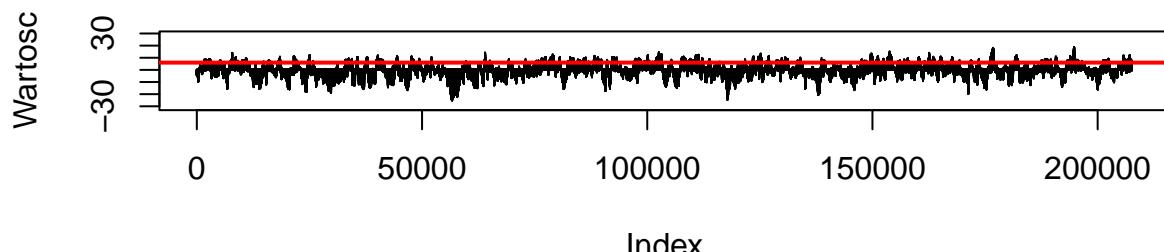
Wartosc

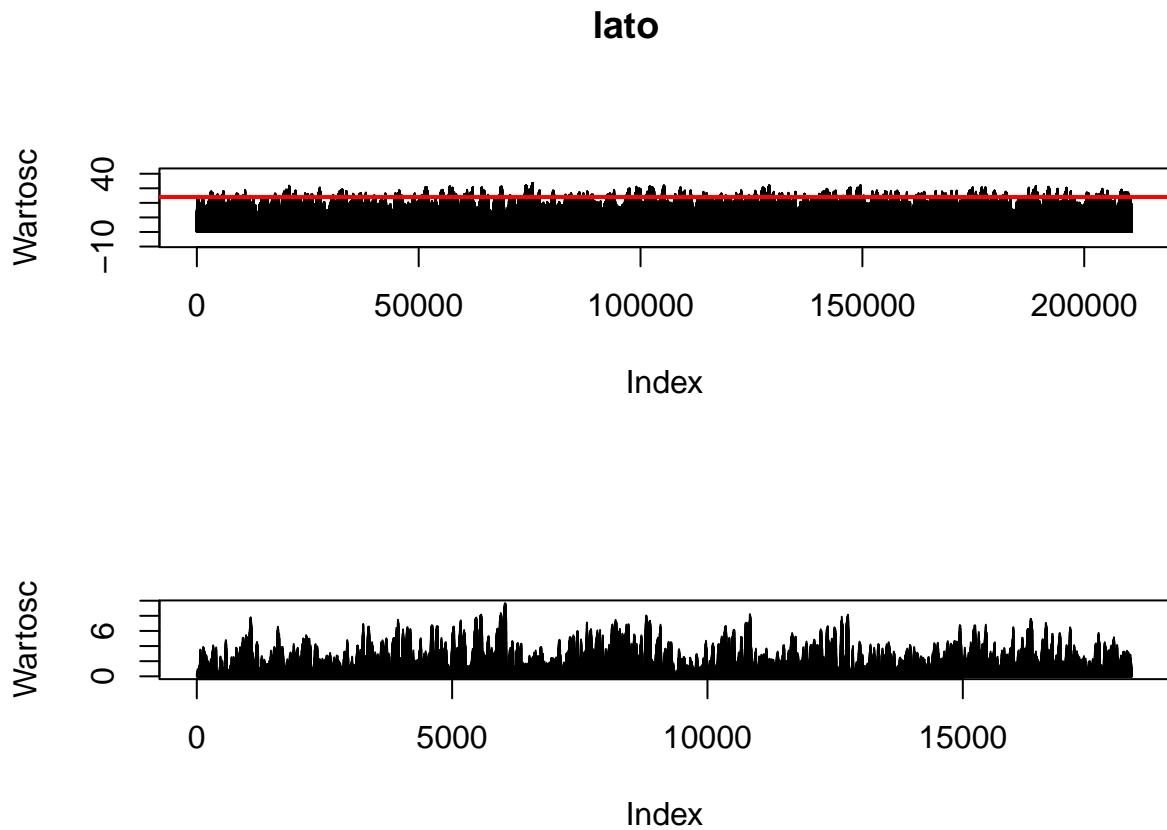


Wartosc

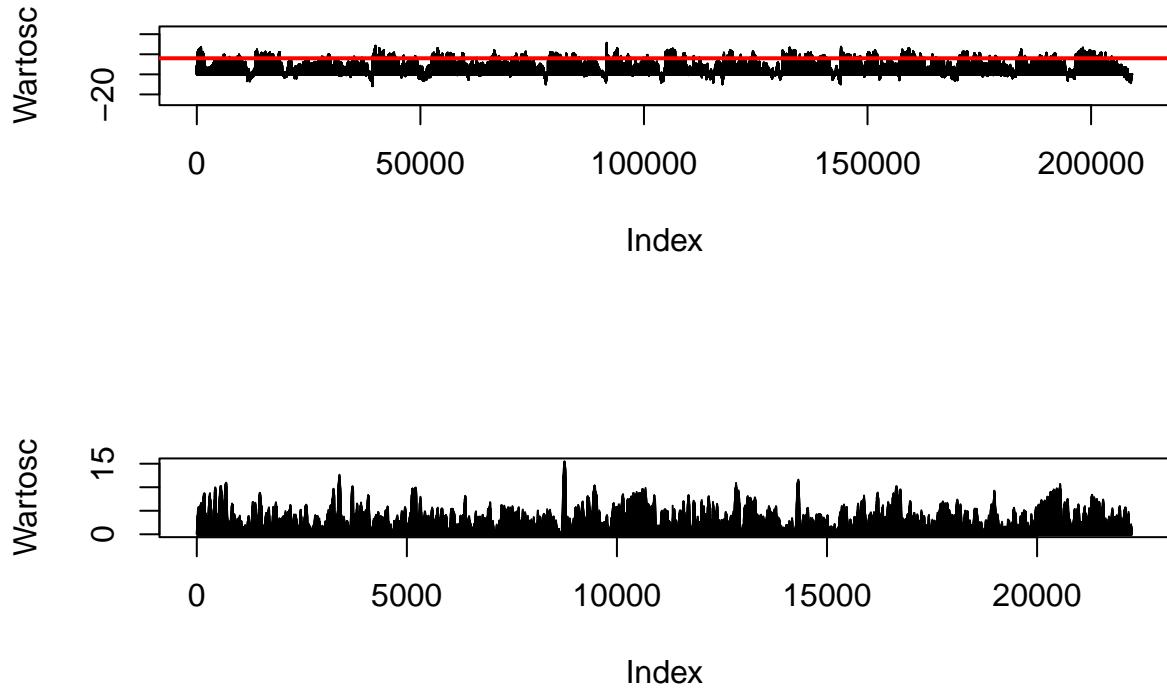


**zima**



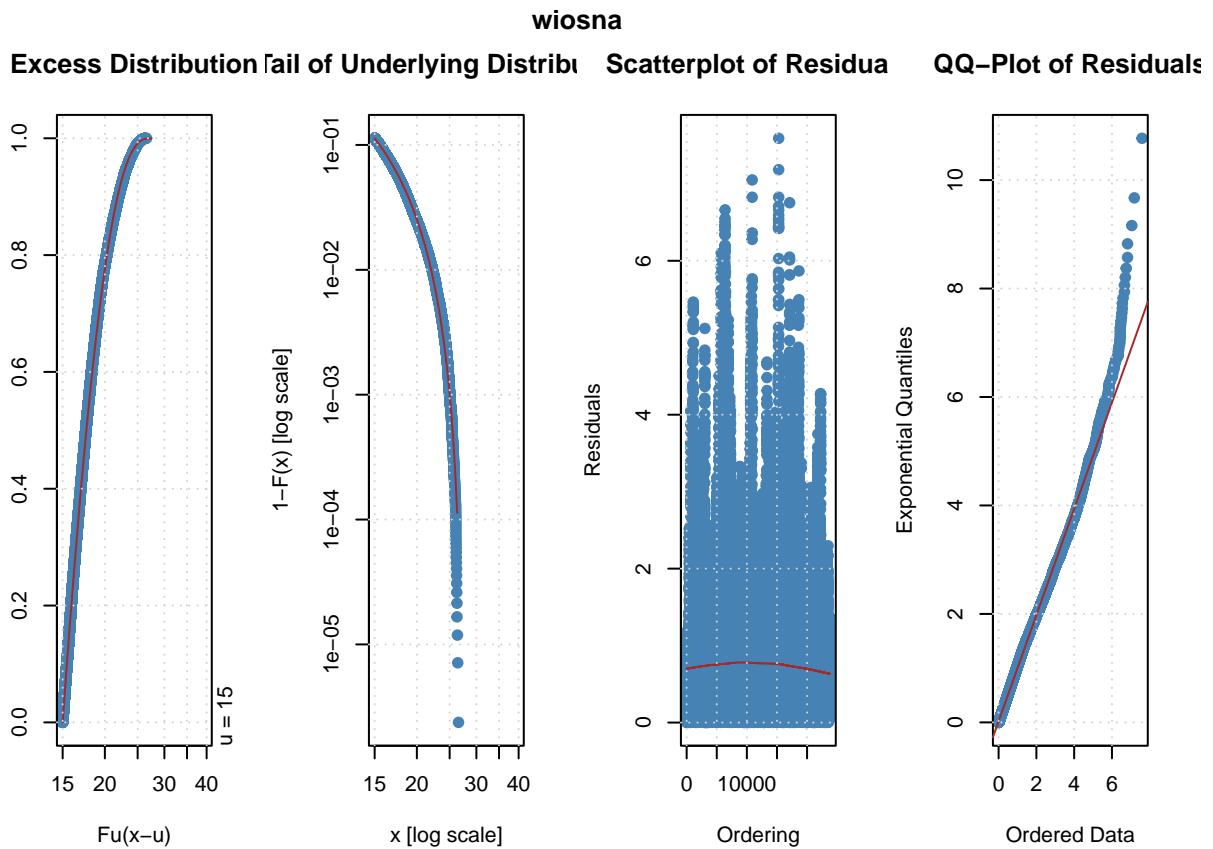


## jesien



Mając już zdefiniowane progi sprawdzę jakie będzie dopasowane rozkładu GPD do moich danych, przy pomocy wykresów diagnostycznych z pakietu fExtremes:

```
##  
## Title:  
## GPD Parameter Estimation  
##  
## Call:  
## fExtremes::gpdFit(x = data, u = u)  
##  
## Estimation Type:  
##   gpd mle  
##  
## Estimated Parameters:  
##     xi      beta  
## -0.3412607  4.2646592  
##  
## Standard Deviations:  
##     xi      beta  
## 0.005666138 0.035483120  
##  
## Log-Likelihood Value:  
##    50220.21  
##  
## Type of Convergence:  
##    0
```

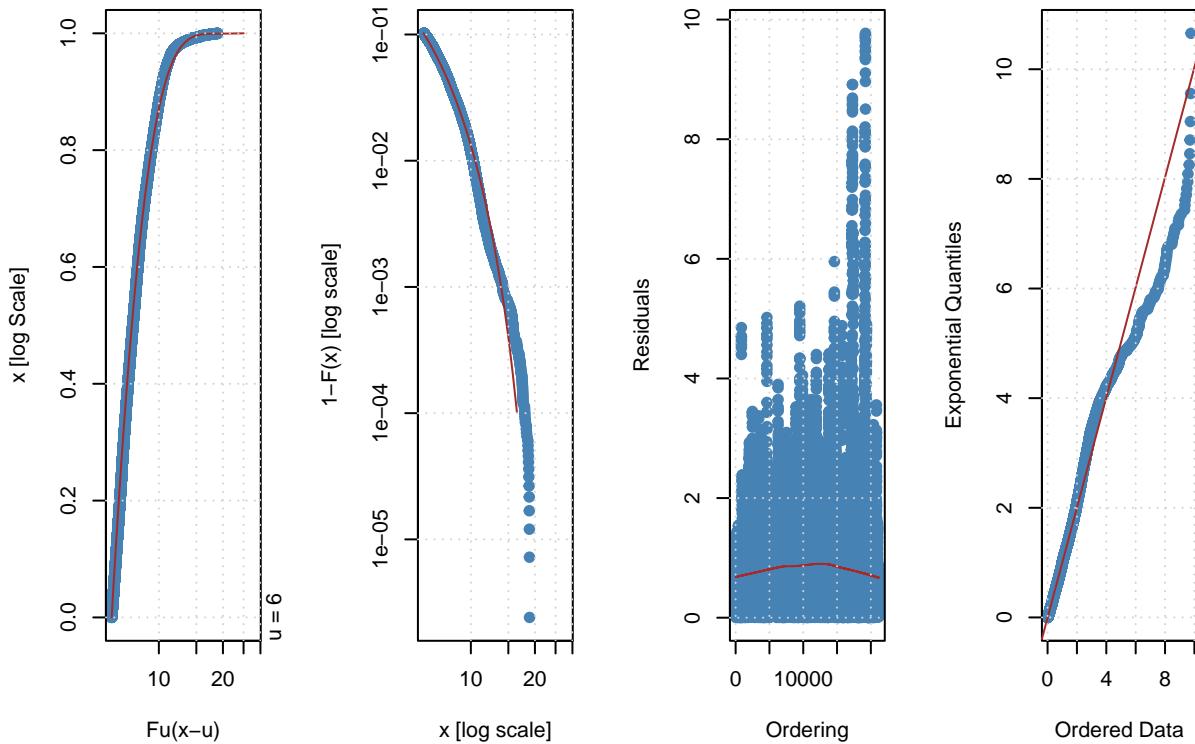


```
##
## Description
##   Wed Jun 12 15:29:44 2024 by user: winal
##
## Title:
##   GPD Parameter Estimation
##
## Call:
##   fExtremes::gpdFit(x = data, u = u)
##
## Estimation Type:
##   gpd mle
##
## Estimated Parameters:
##       xi      beta
## -0.1154554  2.1969762
##
## Standard Deviations:
##       xi      beta
## 0.004792893 0.018316143
##
## Log-Likelihood Value:
##   35461.81
##
## Type of Convergence:
```

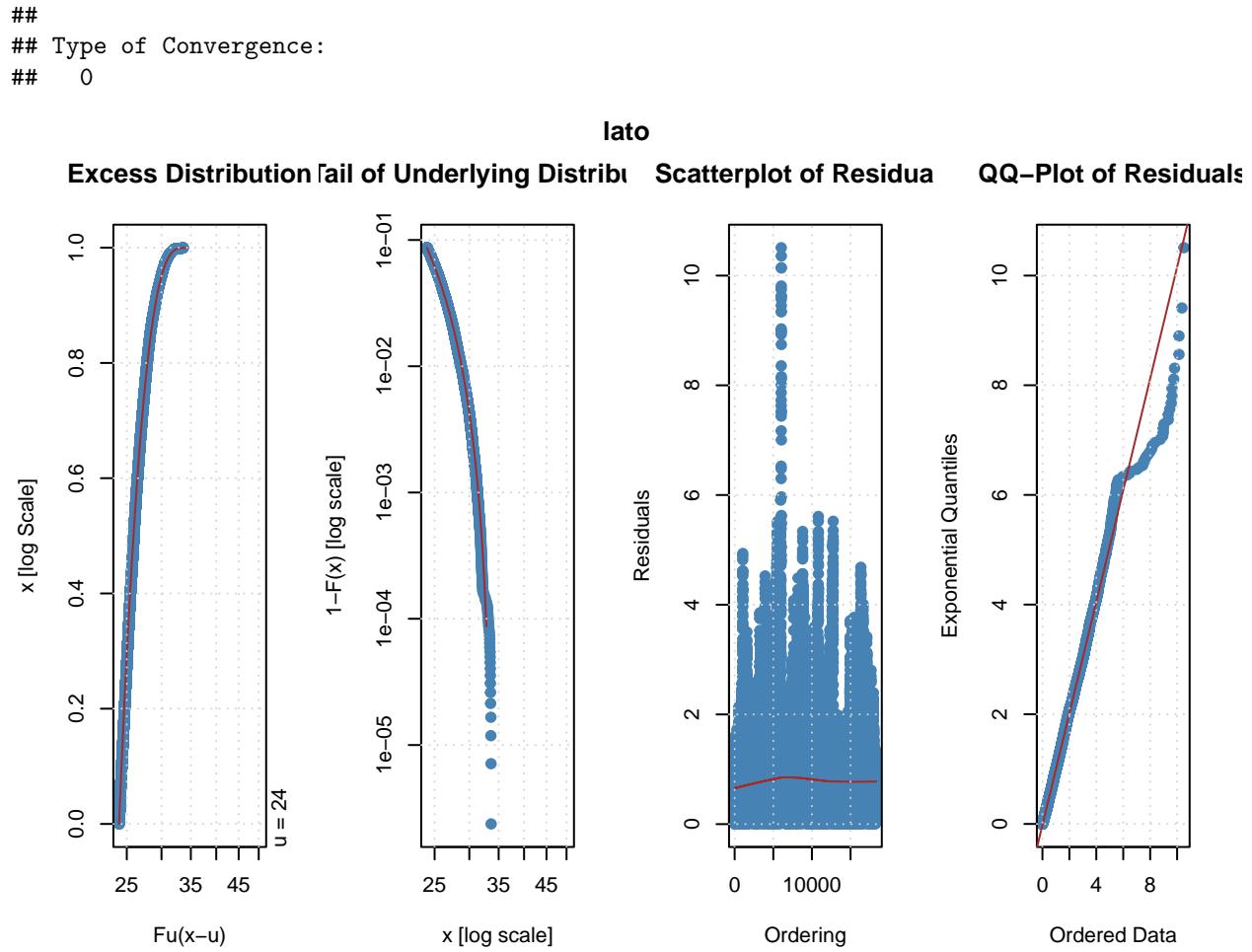
```
## 0
```

zima

Excess Distribution Tail of Underlying Distribution Scatterplot of Residuals QQ-Plot of Residuals



```
##  
## Description  
##   Wed Jun 12 15:29:45 2024 by user: winal  
##  
##  
## Title:  
##   GPD Parameter Estimation  
##  
## Call:  
##   fExtremes::gpdFit(x = data, u = u)  
##  
## Estimation Type:  
##   gpd mle  
##  
## Estimated Parameters:  
##       xi      beta  
## -0.3004539  3.0281874  
##  
## Standard Deviations:  
##       xi      beta  
## 0.004320329  0.025028192  
##  
## Log-Likelihood Value:  
## 33101.99
```



```

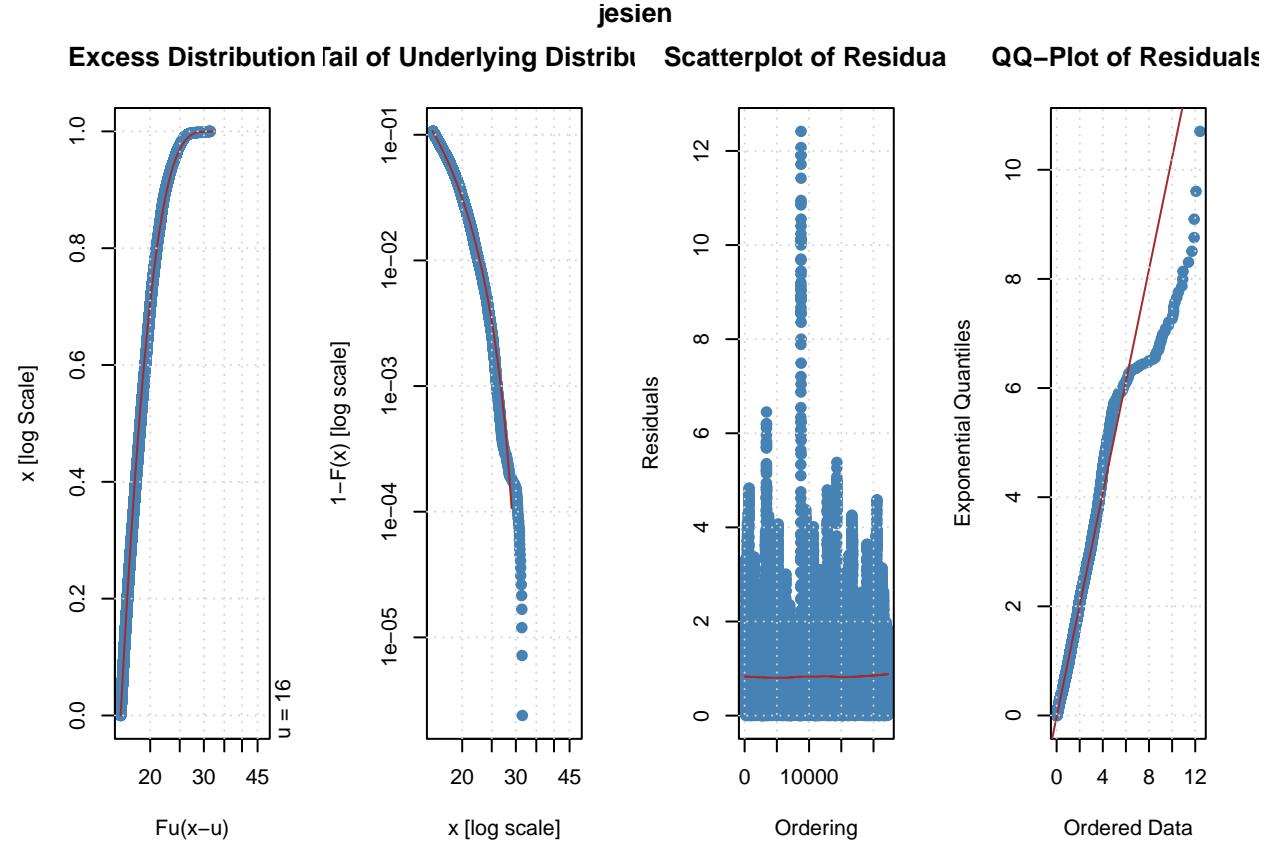
## 
## Description
##   Wed Jun 12 15:29:45 2024 by user: winal
## 
## 
## Title:
##   GPD Parameter Estimation
## 
## Call:
##   fExtremes::gpdFit(x = data, u = u)
## 
## Estimation Type:
##   gpd mle
## 
## Estimated Parameters:
##   xi      beta
## -0.2267173 3.7380452
## 
## Standard Deviations:
##   xi      beta
## 0.003517916 0.027704802
## 

```

```

## Log-Likelihood Value:
##      46568.93
##
## Type of Convergence:
##      0

```



```

##
## Description
##   Wed Jun 12 15:29:45 2024 by user: winal

```

Wykresy są momentami dosyć niepokojące, zwłaszcza Q-Q plot momentami mocno odbiegającymi od oczekiwanych kształtu, zatem jakość wyników przez tą metodę może być niesatisfakcjonująca. Zobaczmy więc te wyniki. Do dopasowanego uogólnionego rozkładu Pareto dla moich nadwyżek odczytuje i zapisuje interesujące mnie poziomy zwrotu. Z metody 3 20 letnie poziomy zwrotu prezentują się tak:

```

## [1] "wiosna"
## [1] 27.12603
##
## [1] "zima"
## [1] 19.15722
##
## [1] "lato"
## [1] 33.58578
##
## [1] "jesien"
## [1] 30.86852

```

A 50 letnie tak:

```
## [1] "wiosna"
## [1] 27.22558
##
## [1] "zima"
## [1] 19.74665
##
## [1] "lato"
## [1] 33.70441
##
## [1] "jesien"
## [1] 31.17225
```

Jak widać mimo nieidealnych dopasowań wyniki nie wydają się być skrajne lub niewłaściwe.

### Porównanie wyników części pierwszej:

Mając już wyniki z użyciem trzech różnych metod możemy je porównać do siebie. Wpierw spójrzę na estymacje 20-letnich poziomów zwrotu:

```
## [1] "wiosna"
## [1] 31.96494
## [1] 25.54122
## [1] 27.12603
##
## [1] "zima"
## [1] 20.02608
## [1] 16.28461
## [1] 19.15722
##
## [1] "lato"
## [1] 34.53588
## [1] 33.10927
## [1] 33.58578
##
## [1] "jesien"
## [1] 35.24815
## [1] 28.62258
## [1] 30.86852
```

Przy niektórych porach roku, takich jak lato, rozstrzał między wynikami jest względnie nieduży, bo raptem około jednego stopnia. Jednak wyniki dla jesieni są już specyficzne, bo oprócz różnicy o siedem stopni pomiędzy metodami, to jeszcze należy zauważyć, że metoda 1 pokazała tam poziom zwrotu wyższy niż w lecie, czyli cieplejszej porze roku. Ogólnie można zauważyć, że dla każdej poru roku metoda 1 daje najwyższe wyniki, a metoda 2 najniższe.

Zobaczę teraz wyniki dla 50 letniego poziomu zwrotu:

```
## [1] "wiosna"
## [1] 33.83817
```

```

## [1] 26.18874
## [1] 27.22558
##
## [1] "zima"
## [1] 20.85248
## [1] 17.18688
## [1] 19.74665
##
## [1] "lato"
## [1] 35.01227
## [1] 33.52822
## [1] 33.70441
##
## [1] "jesien"
## [1] 36.3638
## [1] 29.53696
## [1] 31.17225

```

Wnioski są tutaj analogiczne: metoda 1 daje najwyższe wyniki, metoda 2 daje najniższe wyniki oraz w lecie różnica między wynikami jest mała porównując ją do różnic w jesieni. Najważniejszy jest fakt, że każda wartość tutaj wyszła większa niż przy 20 letnim poziomu zwrotu, co jest zgodne z tym, że mają to być kwantyle wyższych rzędów.

## Część druga

### Przygotowanie danych

Do tej części potrzebuje dodatkowych danych z innego pliku, zawierające indeksy stacji z zapisami temperatur, ich nazwy oraz ich współrzędne geograficzne

```

##      LP.           ID          Nazwa          Rzeka
##  Min.   : 1.0   Min.   :249180010   Length:654      Length:654
##  1st Qu.:164.2  1st Qu.:249220062   Class  :character  Class  :character
##  Median :327.5  Median :250210145   Mode   :character  Mode   :character
##  Mean   :327.5  Mean   :260541990
##  3rd Qu.:490.8  3rd Qu.:253167588
##  Max.   :654.0  Max.   :354220195
##
## Szerokość geograficzna Długość geograficzna Wysokość n.p.m.
## Length:654          Length:654          Min.   : 0.0
## Class  :character    Class  :character    1st Qu.:130.0
## Mode   :character    Mode   :character    Median :222.0
##                           Mean   :308.3
##                           3rd Qu.:420.0
##                           Max.   :1991.0
##                           NA's   :39

```

Moje dane potrzebują na wstępie kilku zmian. Wpierw zamienie zapisy temperatur z maksimów 10-minutowych na maksima dobowe, dla zminimalizowania ilości danych do przetwarzania w kolejnych punktach. Oprócz tego, dla jeszcze większego zmniejszenia datasetu i upływnienia obliczeń zredukuje ilość stacji branych pod uwagę do 30. Wybór będzie podyktowany jak najmniejszą liczbą braków w danych.

Maksima dobowe dla mojej poprzednio wybranej stacji prezentują się tak:

```

##   wartosci_temp      data
## 1       -2.88 2008-1-1
## 2       -4.20 2008-1-2
## 3       -3.96 2008-1-3
## 4        0.34 2008-1-4
## 5        4.04 2008-1-5
## 6        2.92 2008-1-6

```

A lista indeksów stacji tak:

```

## [1] "X253230160" "X250180030" "X249180210" "X249190030" "X249180230"
## [6] "X249200370" "X254180010" "X249210070" "X249200920" "X253220270"
## [11] "X251200270" "X250170110" "X254200080" "X249210240" "X249190240"
## [16] "X253180150" "X354210185" "X249180160" "X253180040" "X350190550"
## [21] "X250170390" "X249190480" "X250200280" "X250200230" "X249220080"
## [26] "X250210240" "X250170330" "X253190220" "X250160520" "X250220120"

```

## Punkt 1

### Opis

Pierwszym punktem jest dopasowanie kopyuł opisujących zależności między parami stacji s0 i s1, gdzie s0 to moja poprzednio wybrana stacja Korbielów. Kopyuły to inne określenie dystrybuant wielowymiarowych, mających jednostajne rozkłady brzegowe. Kopyuły mają ograniczoną ilość “typów” czyli nazw, które różnią się od siebie ilością parametrów lub ich wartościami. Do ich estymacji tworze pętle iterującą po każdej stacji, wyznaczam wszystkie potrzebne informacje czyli nazwę wyznaczonej kopyuły (metodą parametryczną tzw. par i nieparametryczną tzw. npar), współczynnik Kendalla (statystyka opisująca zależności dwóch zmiennych losowych) oraz zależności ekstremalne, które zostaną użyte później. Wszystkie informacje zapisuje w tabelce, którą później łączę z informacjami o stacjach by móc zaprezentować wyniki na mapie.

### Dopasowanie kopyuł

Tabelka z danymi po operacji pętli prezentuje się następująco:

```

##           ID          kopuła.npar          kopuła.par
## 1 253230160      Rotated BB8 90 degrees Rotated BB8 90 degrees
## 2 250180030 Rotated Tawn type 2 90 degrees Rotated BB8 90 degrees
## 3 249180210                 Survival BB8          Survival BB8
## 4 249190030                 Survival BB8          Survival BB8
## 5 249180230                         Frank            Frank
##           Kendall.npar          Kendall.par
## 1    -0.106764047502332 -0.103747740062731
## 2   -0.0825255962712204 -0.0832103773633009
## 3    0.0706944836393473  0.0723560061971021
## 4    0.167953483141887  0.159894223773246
## 5    0.196248973962729  0.193933377240261

```

Teraz połączę ją z danymi o stacjach na podstawie ich ID:

	ID	kopuła.npar	kopuła.par	Kendall.npar	Kendall.par	Nazwa
## 1	249180160	Survival BB8	Survival BB8	0.12732139	0.13914980	BRENNNA

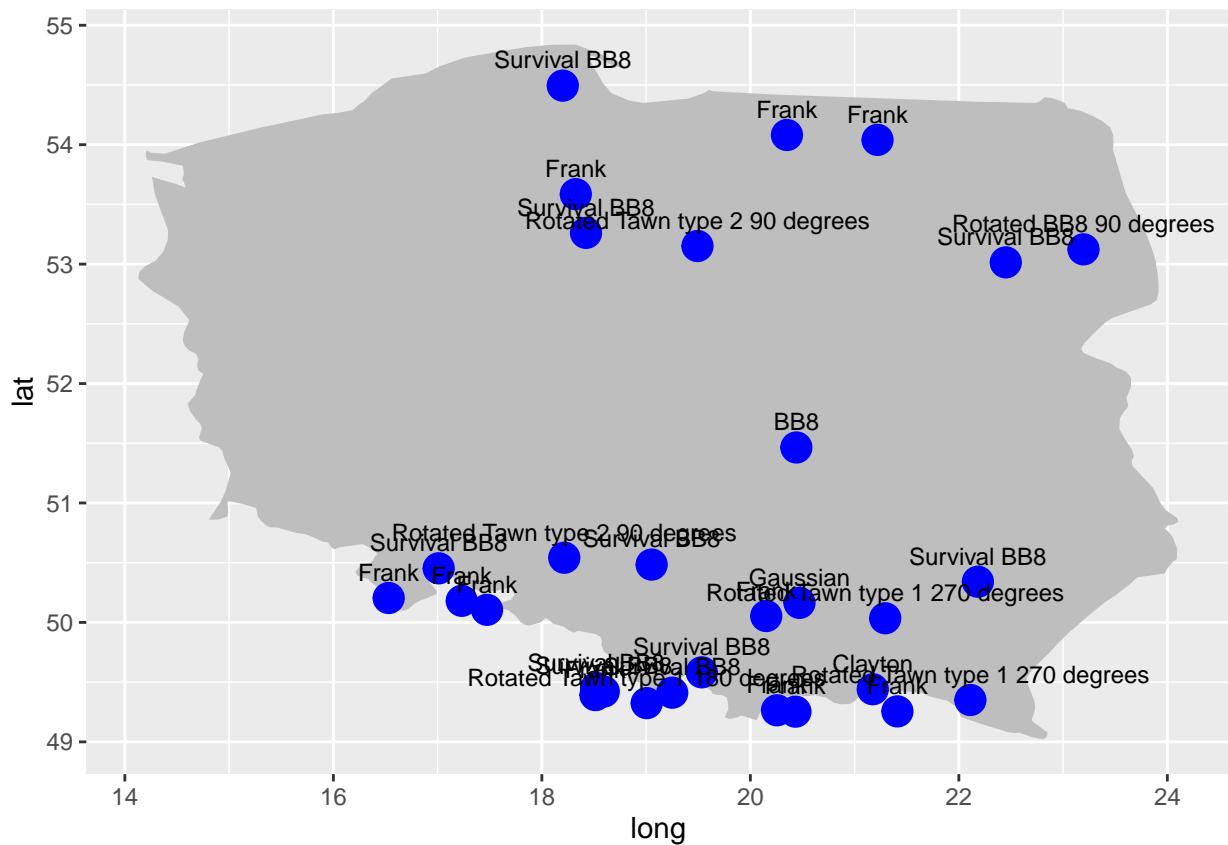
```

## 2 249180210 Survival BB8 Survival BB8 0.07069448 0.07235601      SZCZYRK
## 3 249180230          Frank        Frank 0.19624897 0.19393338      WISŁA
## 4 249190030 Survival BB8 Survival BB8 0.16795348 0.15989422 LIBERTÓW
## 5 249190240 Survival BB8 Survival BB8 0.10773515 0.10743522 LACHOWICE KRALE
##   Szerokość geograficzna Długość geograficzna
## 1           49.4513           18.5215
## 2           49.4211           18.5941
## 3           49.3917           18.5141
## 4           49.5821           19.5342
## 5           49.4118           19.2504

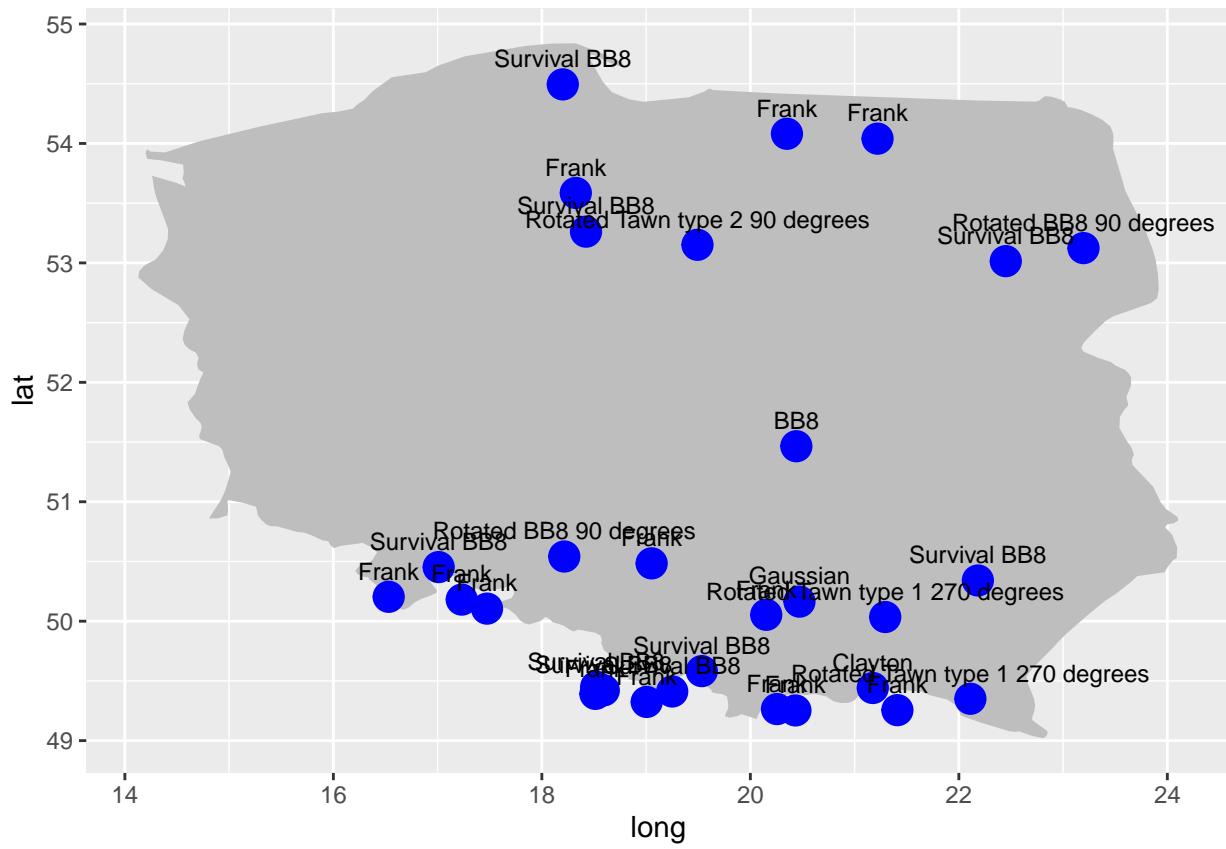
```

### Mapy z wynikami

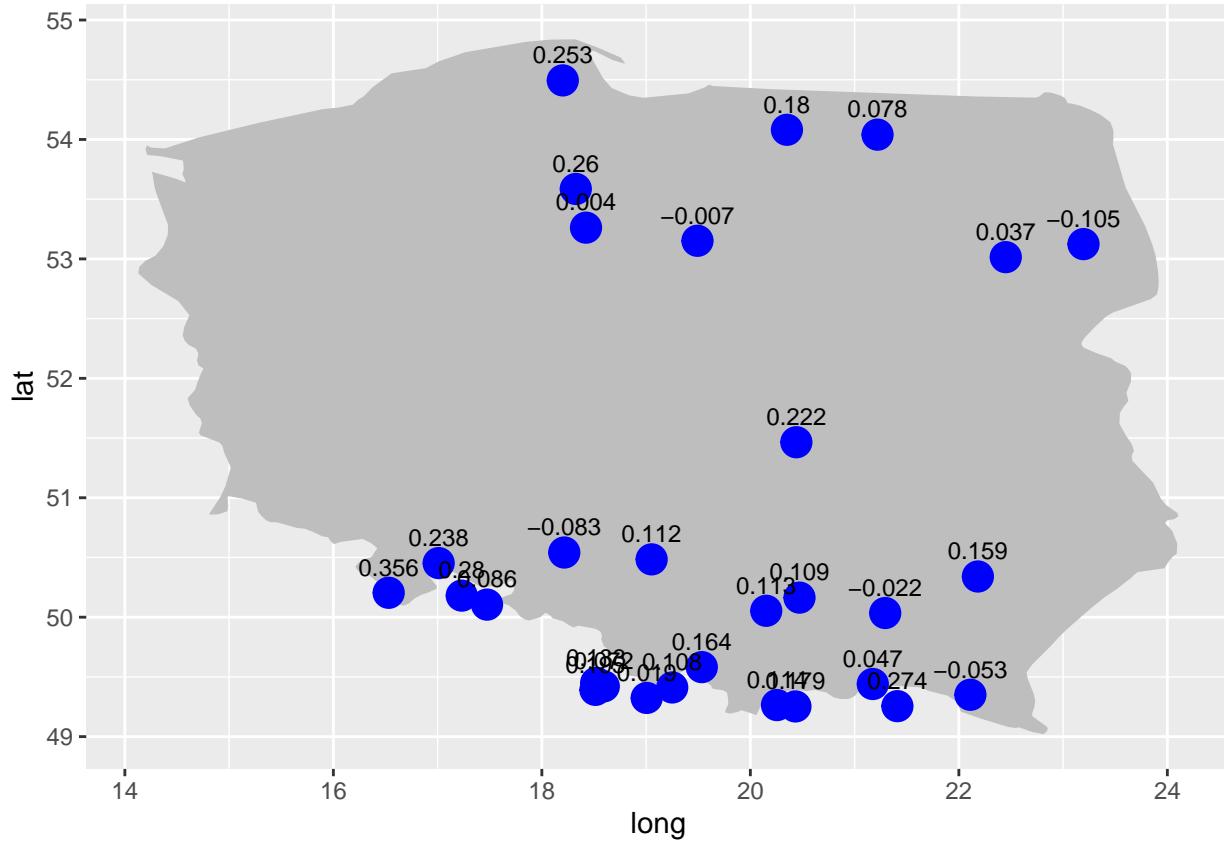
Pierwsza jest mapa stacji z nazwami wyestymowanych kopuł metodą nieparametryczną:



Kolejna jest mapa stacji z nazwami wyestymowanych kopuł metodą parametryczną:



A trzecią mapą są stacje z wyestymowanym dla nich współczynnikiem Kendalla (uśrednionym z obu metod):



### Opis najczęstszej kopuły

Mogę teraz sprawdzić najczęściej występującą kopułę dla naszych stacji, dla podejścia parametrycznego i nieparametrycznego:

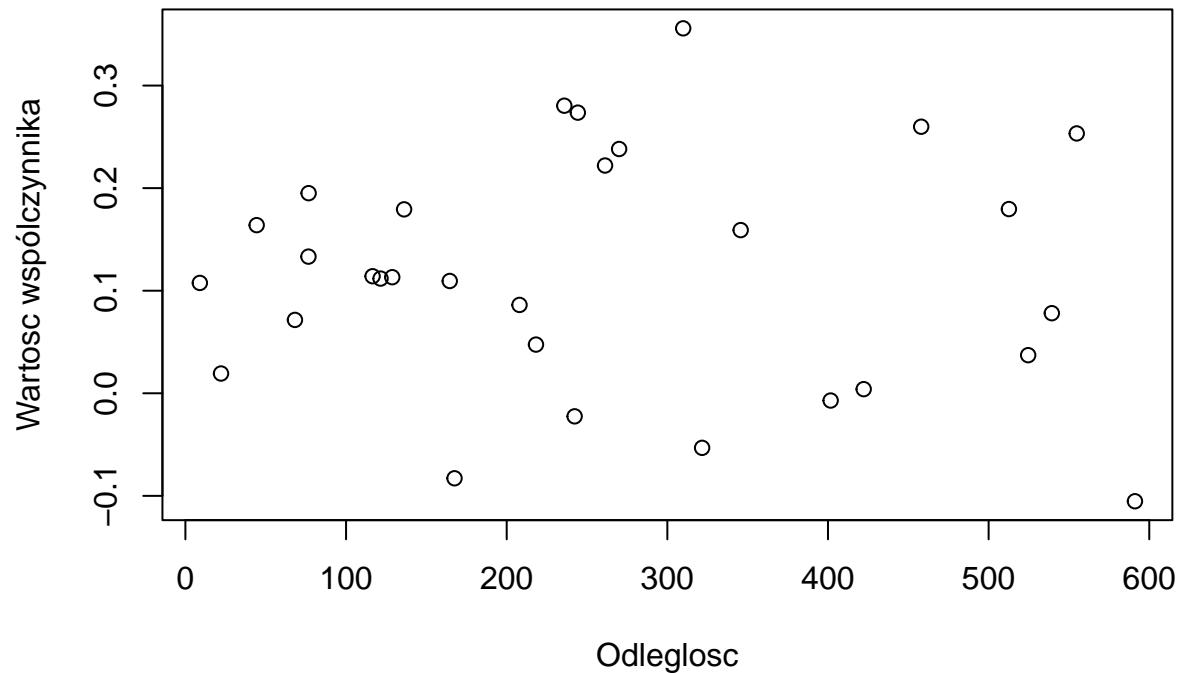
```
## [1] "Most occurring copula: Frank"
```

Najczęściej występującą kopułą dla obu podejść jest kopuła Franka. Jest to jedna z kopuł Archimedesa, oznacza symbolem  $F_{\theta}^{Fr}$  oparta na rozkładzie dwuwymiarowym Gaussa, z pojedynczym parametrem  $\theta$ . Oba współczynniki zależności ekstremalnej dla tej kopuły są równe 0. W przypadku gdy  $\theta=0$  to mamy szczególny przypadek, gdzie kopuła Franka pokrywa się z kopułą niezależności.

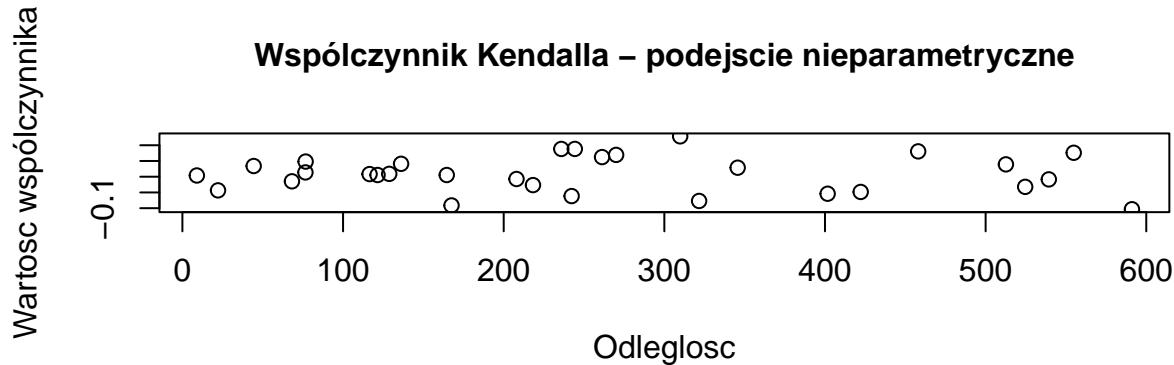
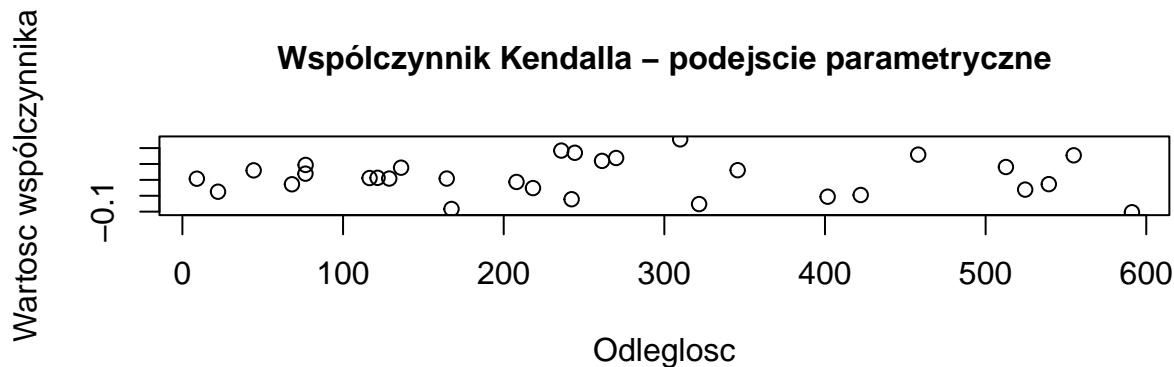
### Analiza współczynnika Kendalla

Mając wyliczony współczynnik Kendalla pozostało zrobić wykres jego zależności od odległości danej stacji od mojej oryginalnie wybranej. Wpierw uśrednie go z obu metod:

### **Współczynnik Kendalla – usredniony**



Z wykresu nie wynika jednoznaczna korelacja tych zmiennych. Jedynie widać, że dla mniejszych odległości wariancja współczynnika Kendalla jest mniejsza i wszystkie wartości są blisko siebie. Zobaczmy czy tak samo będzie jak rozdzielimy wartości ze względu na podejście:



Ponownie nie widać jakiejkolwiek zależności. Ponownie wariancja wydaje się być mniejsza dla bardzo małych odległości, jednak różnica wydaje się mniejsza niż przy uśrednionej wartości współczynnika Kendalla.

## Punkt 2

### Opis

Celem punktu drugiego jest wyestymowanie temperatury dla stacji najbliższej i najdalszej odległościowo od mojej wybranej stacji Korbielów, przy warunku, że w mojej stacji temperatura jest na poziomie 20 i 50 letniego poziomu zwrotu. By to zrobić ponownie dopasuje rozkład do mojej stacji, tym razem dla maksimów dobowych by mieć większą płynność obliczeń, następnie wyznacze poziomy zwrotu dla tych maximów i za pomocą kopuł wyznacze rozkłady warunkowe dla danych dwóch stacji i wyznacze dla nich te same kwantyle.

### Analiza

Rozkład dla maksimów dobowych to:

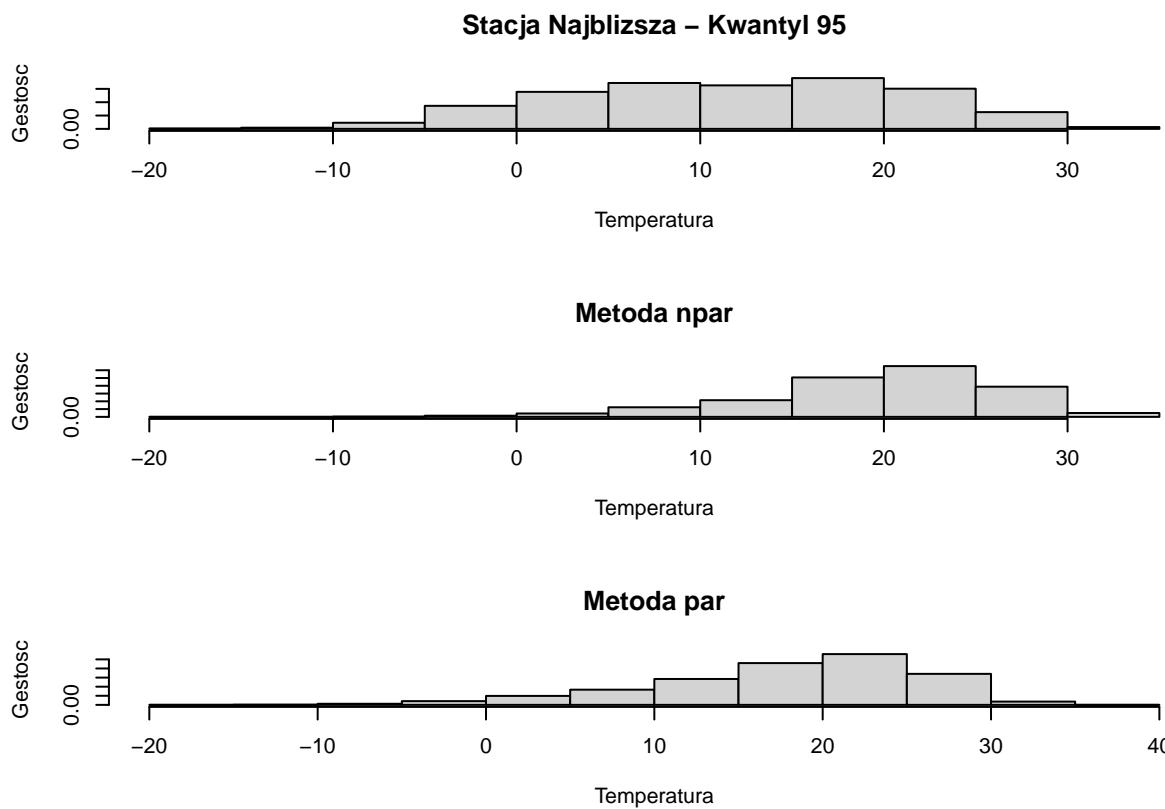
```
## [1] "SEP4"                      "skew exponential power type 4"
##   eta.mu eta.sigma   eta.nu   eta.tau
## 12.240647 2.710312 1.106608 1.508534
```

Poziomy zwrotu mojej stacji przy użyciu maksimów blokowych wynoszą:

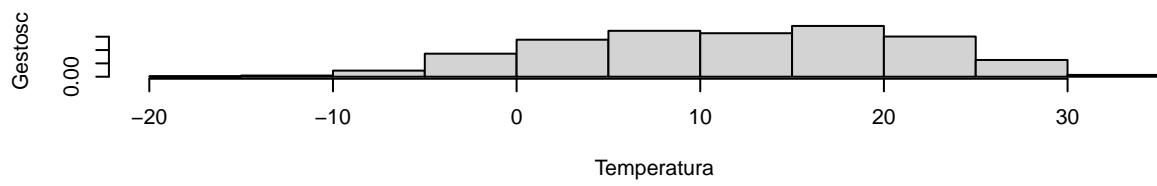
```
## [1] 34.04467 34.74064
```

Wyniki wychodzą dosyć wysokie, jednak należy pamiętać, że metoda z dopasowaniem rozkładu o ile najprostsza już poprzednio pokazała tendencje do zawyżania wyników. Jednak z powodu wygody i szybkości i tak zdecydowałem się jej użyć. Mając wyliczone poziomy zwrotu mojej stacji mogę sporządzić histogramy poziomów zwrotu dla pozostałych stacji:

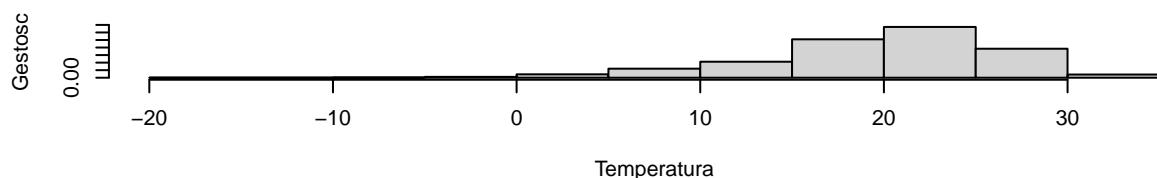
```
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[3,3] = 0
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[3,3] = 0
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[3,3] = 0
## |
```



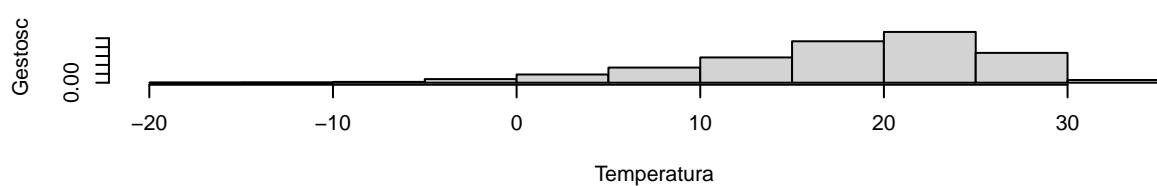
**Stacja Najblzsza – Kwantyl 98**



**Metoda npar**

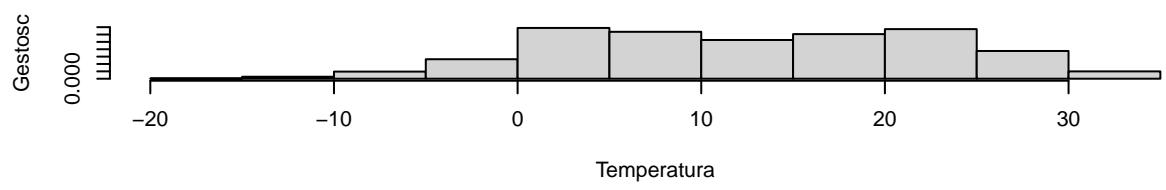


**Metoda par**

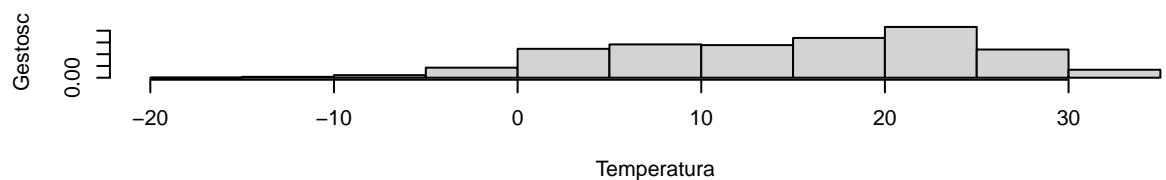


```
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[3,3] = 0 |
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[3,3] = 0 |
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0 |
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0 |
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0 |
## |
## procedura Lapack dgesv: system jest dokładnie osobliwy: U[4,4] = 0 |
## |=====
```

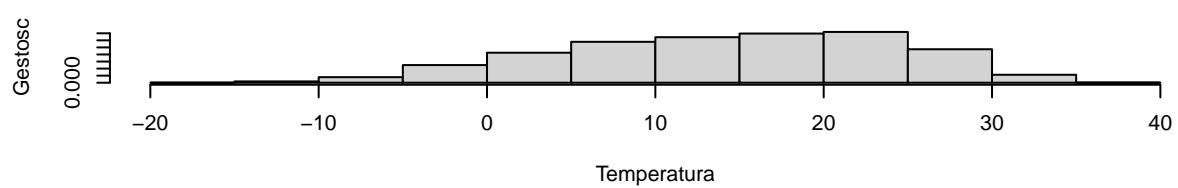
**Stacja Najdalsza – Kwantyl 95**

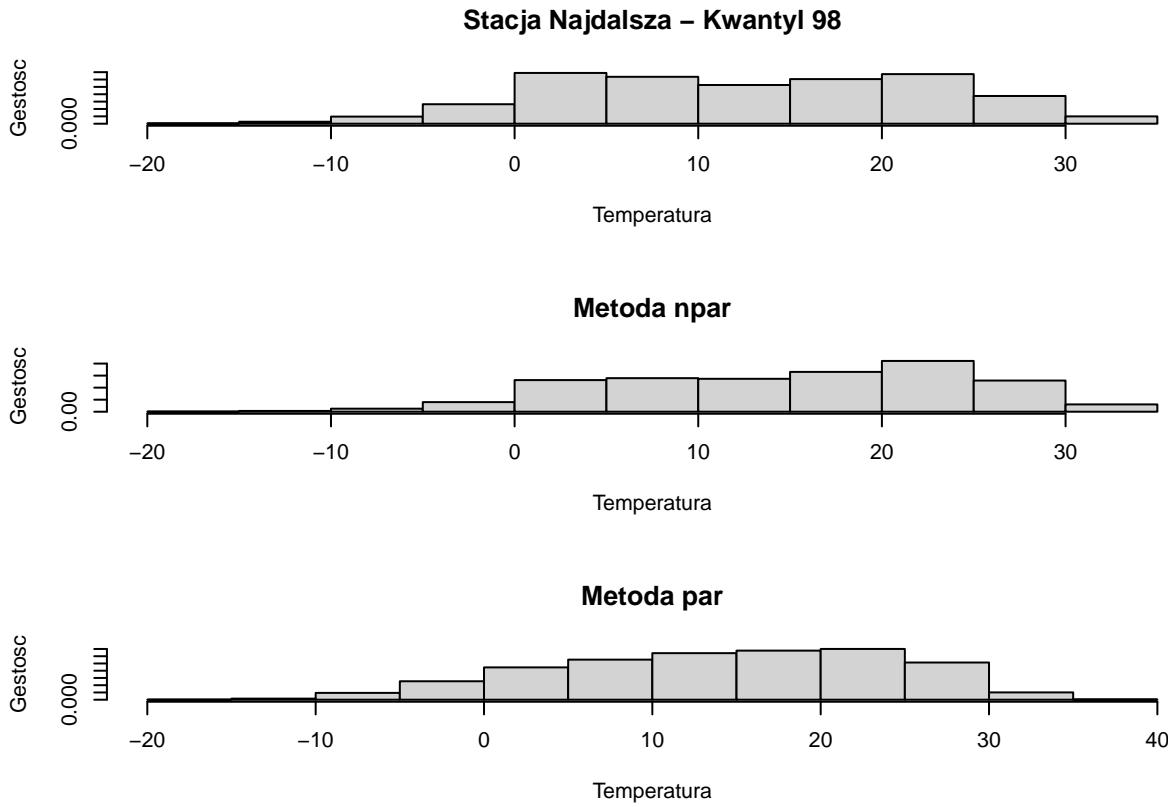


**Metoda npar**



**Metoda par**





Dla najbliższej stacji poziomy zwrotu wynoszą:

-metodą nieparametryczną

```
## $ '95%'  
## [1] 28.68024
```

```
## $ '98%'  
## [1] 30.05115
```

-metodą parametryczną

```
## $ '95%'  
## [1] 28.26688
```

```
## $ '98%'  
## [1] 29.62522
```

Dla najdalszej stacji poziomy zwrotu wynoszą:

-metodą nieparametryczną

```
## $ '95%'  
## [1] 28.97614
```

```
## $ '98%'  
## [1] 30.79
```

-metodą parametryczną

```
## $ '95%'  
## [1] 28.75099
```

```
## $ '98%'  
## [1] 30.45341
```

## Punkt 3

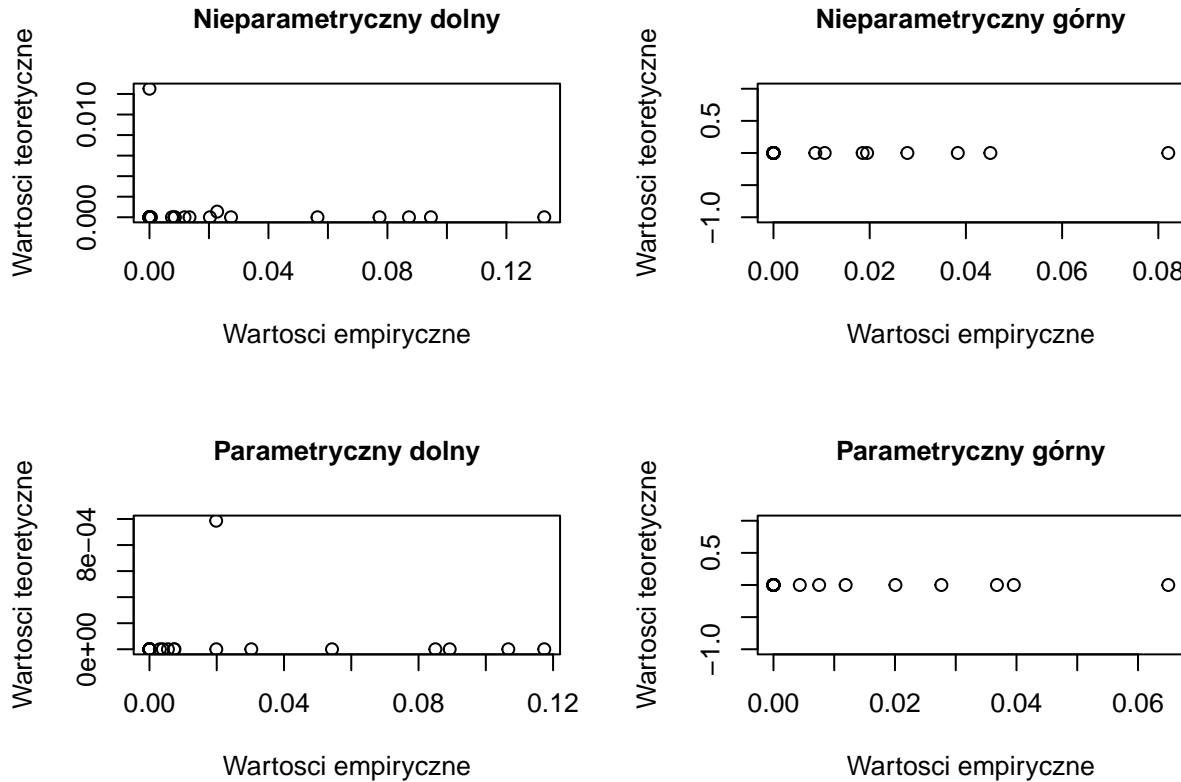
### Opis

Trzeci punkt polega na wyestymowaniu współczynników zależności ekstremalnej dla naszych dobranych kopuł, a następnie porównanie ich na wykresie z zależnościami teoretycznymi dla danych typów kopuł. Wszystkie te współczynniki zostały policzone i zapisane przy okazji punktu 1 i dopasowywania kopuł, zatem pozostało jedynie wyświetlić je na wykresie.

### Analiza

Wykresy współczynników zależności ekstremalnej:

#### **Współczynniki zależności ekstremalnej**



Wszystkie górne zależności teoretyczne są równe zero, jednak widać że z danych empirycznych były przypadki gdzie wychodziły wartości nieznacznie większe od zera. Dla dolnych zależności pojawiły się pojedyncze przypadki gdzie ich wartość nie powinna być zerem. Ponownie widać, że dane empiryczne dawały błędne niezerowe wyniki w niektórych przypadkach, i to większe niż w przypadku górnych zależności.

## Część trzecia

### Opis

Trzecia część projektu skupia się już nie na pojedynczej zmiennej, jaką jest temperatura powietrza, ale na czterech zmiennych i modelowaniu ich rozkładu wielowymiarowego jak i dopasowaniu do nich modelu regresji kwantylowej. Oba te kroki bazują na strukturach R-vine, czyli strukturach stworzonych z dwuwymiarowych kopuł służących do utworzenia rozkładu wielowymiarowego. Zatem moim pierwszym celem będzie dopasowanie po jednej ze struktur C-vine i D-vine do danych oraz krótkie opisanie ich. Następnie znajdę strukture najlepiej opisującą moje dane, i na tej podstawie przeprowadzę regresje kwantylową dla moich zmiennych i stworze model predykcyjny operujący na kwantylach z danych. Regresje przeprowadzę tworząc dwa modele: jeden metodą parametryczną i drugi nieparametryczną, a następnie porównam oba te modele ze sobą i zobaczę jakie wnioski o zależnościach między moimi zmiennymi z nich wynikają.

### Analiza

#### Data preprocessing

Pierwszym krokiem jest ponowne przygotowanie sobie maksimów dobowych, by mieć większą płynność w obliczeniach. Skorzystam już z tabelki stworzonej w części drugiej z maksimami dobowymi temperatur dla mojej stacji:

```
##   wartosci_temp      data
## 1      -2.88 2008-1-1
## 2     -4.20 2008-1-2
## 3     -3.96 2008-1-3
## 4      0.34 2008-1-4
## 5      4.04 2008-1-5
## 6      2.92 2008-1-6
```

A następnie do tej tabeli dodam na podstawie daty wszystkie pozostałe zmienne, również w postaci maksimów dobowych:

```
##      Data      Temperatura_powietrza Średnia_prędkość_wiatru
## Length:5783      Min.    :-16.350      Min.    : 0.000
## Class :character 1st Qu.:  5.463      1st Qu.: 3.600
## Mode  :character Median : 12.795      Median : 4.400
##                  Mean   : 12.271      Mean   : 4.701
##                  3rd Qu.: 19.337      3rd Qu.: 5.500
##                  Max.   : 33.650      Max.   :13.900
##                  NA's   :433        NA's   :275
##      Suma_opadów Temperatura_gruntu
## Min.    : 0.0000      Min.    : -11.860
## 1st Qu.: 0.0000      1st Qu.:  7.228
## Median : 0.1000      Median : 16.485
## Mean   : 0.4879      Mean   : 17.207
## 3rd Qu.: 0.4000      3rd Qu.: 24.750
## Max.   :24.8000      Max.   :1250.530
## NA's   :1057        NA's   :427
```

Widzę, że w każdej z tabel występują wartości NA, które zaburzą dalsze analizy, zatem od razu się ich pozbędę:

```

##      Data      Temperatura_powietrza Średnia_prędkość_wiatru
##  Length:4630      Min.   :-16.350      Min.   : 0.000
##  Class :character 1st Qu.:  5.572      1st Qu.: 3.600
##  Mode  :character Median : 12.745      Median : 4.400
##                  Mean   : 12.319      Mean   : 4.707
##                  3rd Qu.: 19.330      3rd Qu.: 5.500
##                  Max.   : 33.650      Max.   :13.900
##      Suma_opadów Temperatura_gruntu
##  Min.   : 0.0000      Min.   :-11.86
##  1st Qu.: 0.0000      1st Qu.:  7.49
##  Median : 0.1000      Median : 16.43
##  Mean   : 0.4892      Mean   : 17.52
##  3rd Qu.: 0.4000      3rd Qu.: 24.76
##  Max.   :24.8000      Max.   :1250.53

```

Kolejną obserwacją jest bardzo dziwne maksimum dla temperatury gruntu. Jest to niemożliwe by temperatura, podawana tutaj w stopniach cesjusza była aż tak wysoka. Jeżeli przyjrzeć się tej wartości w tej kolumnie posortowanych malejąco:

```
## [1] "Wartości Temperatury gruntu"
```

```

## [1] 1250.53 1222.62 267.04 221.37 198.13 165.30 163.07 162.28 160.53
## [10] 158.13 157.06 156.43 156.19 155.71 152.12 145.90 144.73 143.51
## [19] 142.45 140.54 139.05 137.45 133.28 132.86 128.84 126.15 121.13
## [28] 119.40 117.93 115.83 114.74 80.37 78.19 76.76 72.66 69.45
## [37] 68.54 43.25 42.69 42.05 41.91 41.21 40.86 40.35 39.99
## [46] 39.91 39.88 39.78 39.76 39.58

```

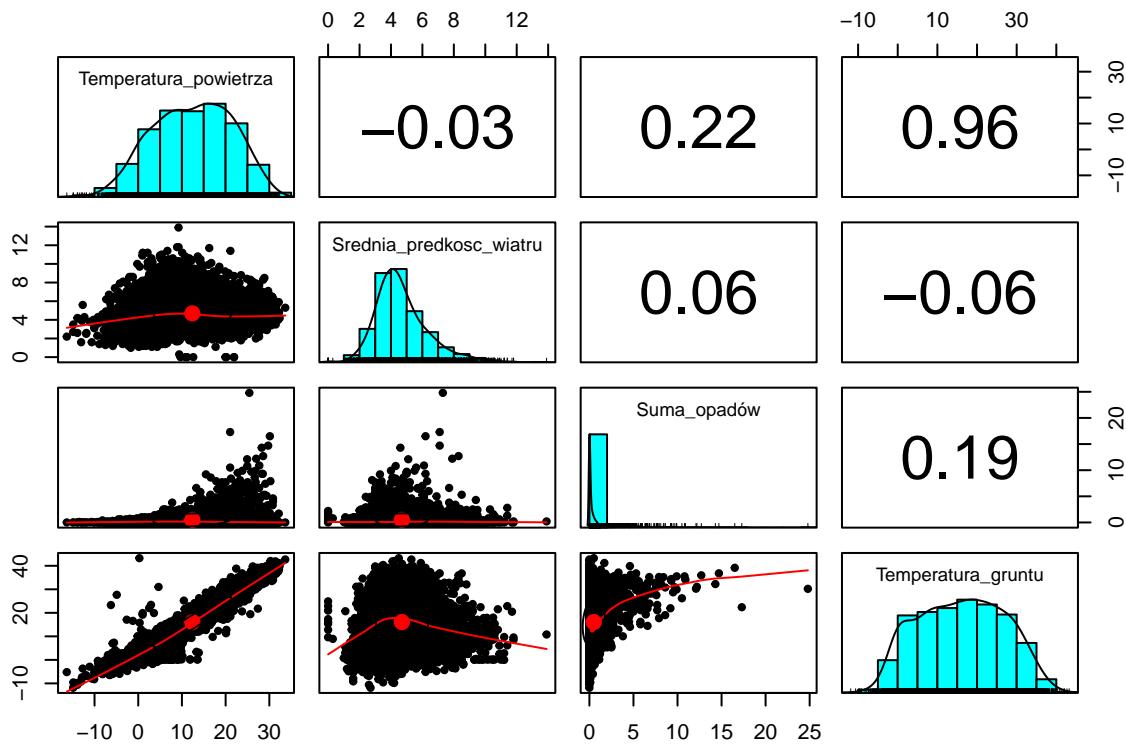
To widać, że wartości w obrębie 30-40 stopni są dosyć pospolite i bardzo stopniowo maleją. Pozostałe wartości wydają mi się być dziwne, zwłaszcza gdy przyrówna się je z temperaturą powietrza (jeżeli temperatura gruntu to 40 stopni to temperatura powietrza również jest wysoka, co zgadza się z logiczną korelacją tych wartości, a gdy przyjrzeć się wartościom 60+ to już temperatura powietrza przyjmuje najróżniejsze wartości, w tym ujemne). Z tych też powodów usunę wszystkie wartości temperatury gruntu powyżej 50:

```

##      Data      Temperatura_powietrza Średnia_prędkość_wiatru
##  Length:4593      Min.   :-16.35      Min.   : 0.000
##  Class :character 1st Qu.:  5.63      1st Qu.: 3.600
##  Mode  :character Median : 12.83      Median : 4.400
##                  Mean   : 12.38      Mean   : 4.692
##                  3rd Qu.: 19.37      3rd Qu.: 5.500
##                  Max.   : 33.65      Max.   :13.900
##      Suma_opadów Temperatura_gruntu
##  Min.   : 0.0000      Min.   :-11.86
##  1st Qu.: 0.0000      1st Qu.:  7.46
##  Median : 0.1000      Median : 16.25
##  Mean   : 0.4917      Mean   : 16.07
##  3rd Qu.: 0.4000      3rd Qu.: 24.54
##  Max.   :24.8000      Max.   : 43.25

```

Mając już odpowiednio przygotowane dane mogę jeszcze nanieść je na wykresy punktowe/histogramy i zobaczyć ich współczynnik korelacji:



Histogramy są ładne, jedynie specyficznie wygląda ten dla sumy opadów, gdzie dominuje wartość 0. Nie przejmuje się tym jednak, gdyż większość dni w roku faktycznie jest raczej bezdeszczowa. Współczynnik korelacji faktycznie potwierdza dużą zależność między temperaturami gruntu i powietrza, podczas gdy reszta zmiennych wydaje się być zdecydowanie słabiej skorelowana ze sobą.

### Struktury C-vine i D-vine

Pierwszym krokiem do stworzenia struktur vine jest stworzenie pseudoobserwacji z naszych danych. Metoda tworzenia jest dowolna, zatem ja korzystam z tej prostszej, nieparametrycznej wersji:

```
##   Temperatura_powietrza Średnia_prędkość_wiatru Suma_opadów Temperatura_gruntu
## 1          0.03983457          0.09958642    0.5346104      0.1242926
## 2          0.36569438          0.88365259    0.8231389      0.5385285
## 3          0.02960383          0.65509360    0.2333478      0.1173269
## 4          0.04386156          0.94884632    0.5346104      0.1394210
## 5          0.18676535          0.34294732    0.2333478      0.6138441
## 6          0.06323465          0.96821942    0.2333478      0.5095777
```

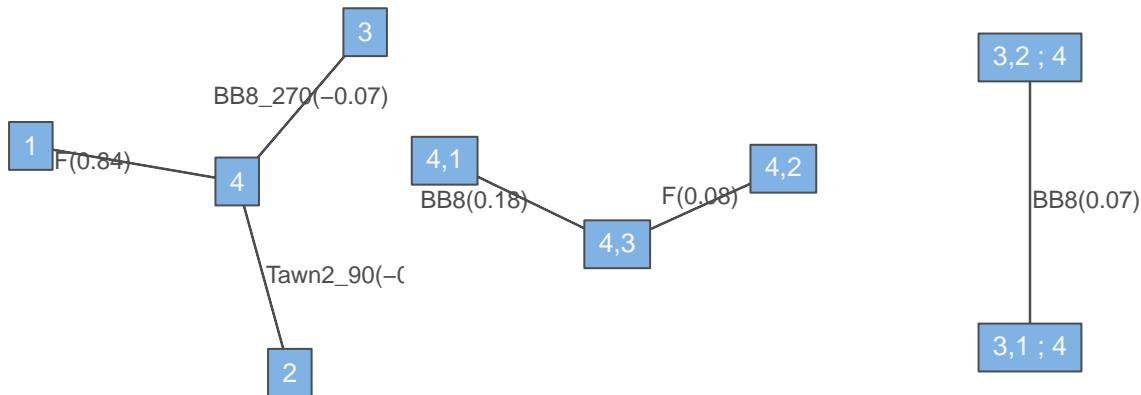
Do utworzenia struktur typu vine potrzebujemy odpowiedniej macierzy, która określa budowę tej struktury i definiuje połączenia między zmiennymi. Struktura C-vine to struktura opierająca się na jednej danej tzw. "korzeniu" (ang. root) i połączeniu reszty zmiennych z nią. Struktura D-vine z kolei opiera się na liniowym połączeniu zmiennych, ale bez żadnych cykli (dla przykładowych struktur będą pokazane grafy które lepiej ilustrują te zasady). Macierzy odpowiednich dla danej struktury jest więcej niż 1, i mają określone zasady, które spełniają powyższe warunki. Jednak ja nie chce tutaj dawać wzorów na nie, więc po prostu weźmy po jednej przykładowej macierzy dla każdej ze struktur, a później wyświetle macierz dla najlepiej dopasowanej struktury.

Wpierw zajmę się utworzeniem struktury C-vine, dla której przykładowa macierz wygląda tak:

```
##   1. 2. 3. 4.
## 1. 1  0  0  0
## 2. 2  2  0  0
## 3. 3  3  3  0
## 4. 4  4  4  4
```

Mając już macierz i pseudoobserwacje mogę dopasować za pomocą funkcji RVineCopSelect kopyły, które utworzą mi strukturę C-vine. Dobierz mi to najlepszą strukturę przy użyciu estymatora największej wiarygodności i kryterium AIC. Budowę struktur vine można przedstawić za pomocą grafów, w których wierzchołkami są danea krawędziami kopyły dwuwymiarowe dopasowane do danych z wierzchołków krawędzi. Mój dobrany C-vine wygląda tak:

**Tree 1**                    **Tree 2**                    **Tree 3**



Widać, że zmienna 4 pełni właśnie funkcje korzenia połącznego z każdą zmienną. Oprócz grafu można też sporządzić tabele zawierającą więcej szczegółów o naszej strukturze:

```
## tree    edge | family      cop    par    par2 | tau    utd    ltd
## -----
##   1     4,1 |      5       F  24.02  0.00 |  0.84    -    -
##           4,2 |    224  Tawn2_90 -1.61  0.05 | -0.04    -    -
##           4,3 |      40    BB8_270 -1.19 -0.95 | -0.07    -    -
##   2     3,1;4 |     10      BB8   3.64  0.44 |  0.18    -    -
##           3,2;4 |      5       F   0.77  0.00 |  0.08    -    -
##           3   2,1;3,4 |     10      BB8   1.17  0.97 |  0.07    -    -
## ---
```

```

## type: C-vine    logLik: 6285.62      AIC: -12551.24      BIC: -12486.92
## ---
## 1 <-> Temperatura_powietrza,    2 <-> Średnia_prędkość_wiatru,    3 <-> Suma_opadów,
## 4 <-> Temperatura_gruntu

```

Jest to jednak mniej czytelna reprezentacja tej struktury.

Teraz tworzę drugą strukturę, tak zwany D-vine. Ponownie wpierw należy przygotować macierz odpowiednią dla struktury tego typu. W moim przypadku używa macierzy:

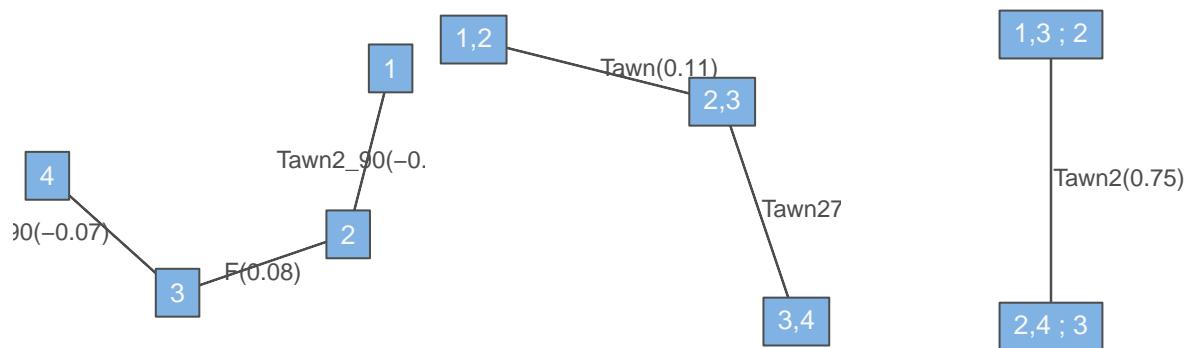
```

##   1. 2. 3. 4.
## 1.  4  0  0  0
## 2.  1  3  0  0
## 3.  2  1  2  0
## 4.  3  2  1  1

```

Ponownie mając macierz nie pozostaje nic innego jak dopasować strukturę do naszych pseudoobserwacji z danych. Graf dla przykładowego D-vine prezentuje się tak:

**Tree 1**                    **Tree 2**                    **Tree 3**



Ponownie pierwszy graf ładnie prezentuje liniowość o której wspominałem wyżej. Jeszcze pozostało pokazać tabelę ze szczegółami o kopułach dla D-vine:

## tree	edge	family	cop	par	par2	tau	utd	ltd
## 1	3,4	30	BB8_90	-1.19	-0.95	-0.07	-	-
##	2,3	5	F	0.76	0.00	0.08	-	-

```

##      1,2 | 224 Tawn2_90 -1.65  0.03 | -0.03   -   -
##      2 2,4;3 | 134 Tawn270 -1.67  0.05 | -0.04   -   -
##      1,3;2 | 104     Tawn  2.76  0.12 | 0.11  0.12   -   -
##      3 1,4;2,3 | 204     Tawn2  5.12  0.92 | 0.75  0.82   -   -
## ---
## type: D-vine    logLik: 5150.01    AIC: -10278.02    BIC: -10207.26
## ---
## 1 <-> Temperatura_powietrza,  2 <-> Średnia_prędkość_wiatru,  3 <-> Suma_opadów,
## 4 <-> Temperatura_gruntu

```

Po budowie grafów widać jasno różnice między tymi strukturami. Co więcej, kupyły dobrane między zmiennymi również się różnią dla obu tych przykładów, co jest naturalną konsekwencją ich różnej budowy, a co za tym idzie, innej kolejności zmiennych.

### Porównanie przykładowych struktur

Najlepszym porównaniem dla obu struktur będzie wyświetlenie kryterium AIC, na podstawie którego ocenimy, który model jest lepiej dopasowany:

```

## [1] "Struktura C-vine"

## [1] "AIC = -12551.2423060843"

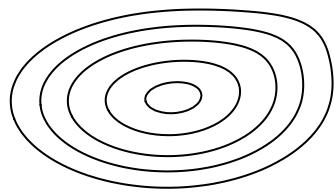
## [1] "Struktura D-vine"

## [1] "AIC = -10278.0196165404"

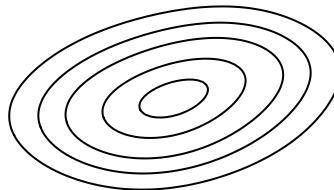
```

Jak widać przykładowa struktua C-vine jest lepiej dopasowana, gdyż ma mniejszą wartości kryterium AIC. Żeby bardziej szczegółowo porównać obie te struktury przyjrzę się dokładniej kopułom z których się składają. Jednak zamiast robić to za pomocą wcześniej prezentowanych tabel, mogę narysować ich wykresy konturowe:

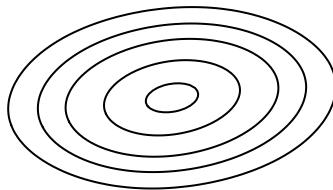
**2,1 ; 3,4**



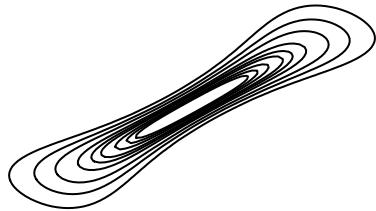
**3,1 ; 4**



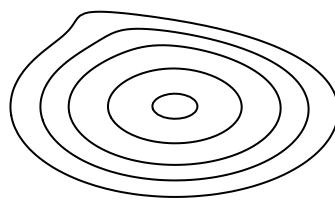
**3,2 ; 4**



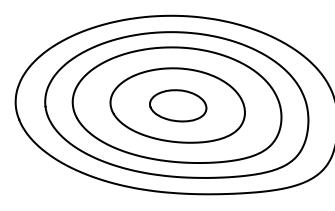
**4,1**

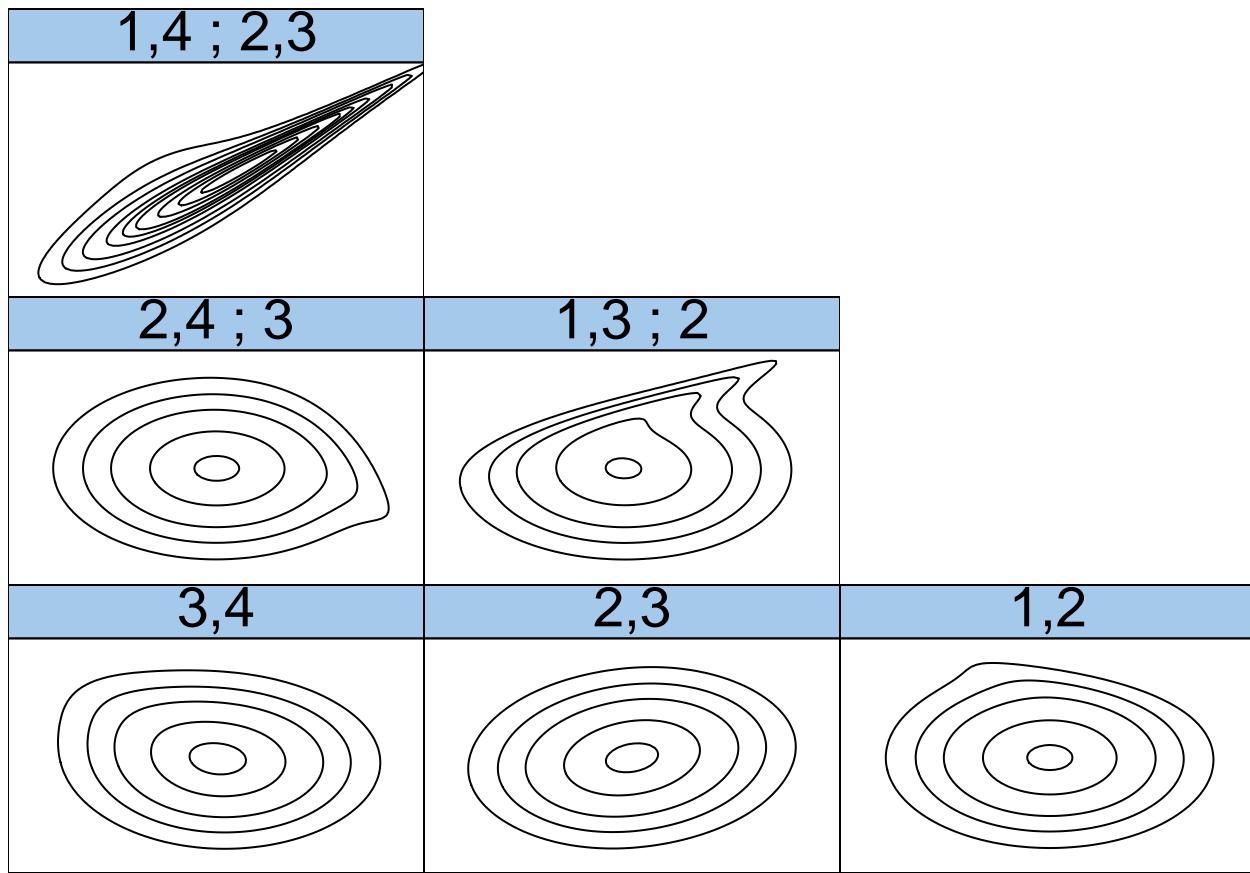


**4,2**



**4,3**





Porównując obie te figury, można dostrzec że kształty kopół odpowiadają sobie w tych strukturach, tylko zmieniają miejsce, czyli zmienne jakie modelują.

### Dobór najlepszej struktury

By dobrać najlepszą strukturę Vine dla naszych danych, skorzystam z funkcji RVineStructureSelect, która mi ją dobierze na podstawie wybranego kryterium porównawczego, w moim przypadku będzie to kryterium AIC:

```
## [1] "D-vine"
```

```
##      [,1] [,2] [,3] [,4]
## [1,]     3    0    0    0
## [2,]     1    1    0    0
## [3,]     4    2    2    0
## [4,]     2    4    4    4
```

Według tej funkcji najlepiej dopasowaną strukturą jest struktura D-vine z powyższą macierzą, chociaż różnica w porównaniu z przykładowym modelem C-vine jest minimalna. Możemy się upewnić czy jest faktycznie najlepsza, dopasowując jak wyżej struktury Vine zadaną przez tą macierz:

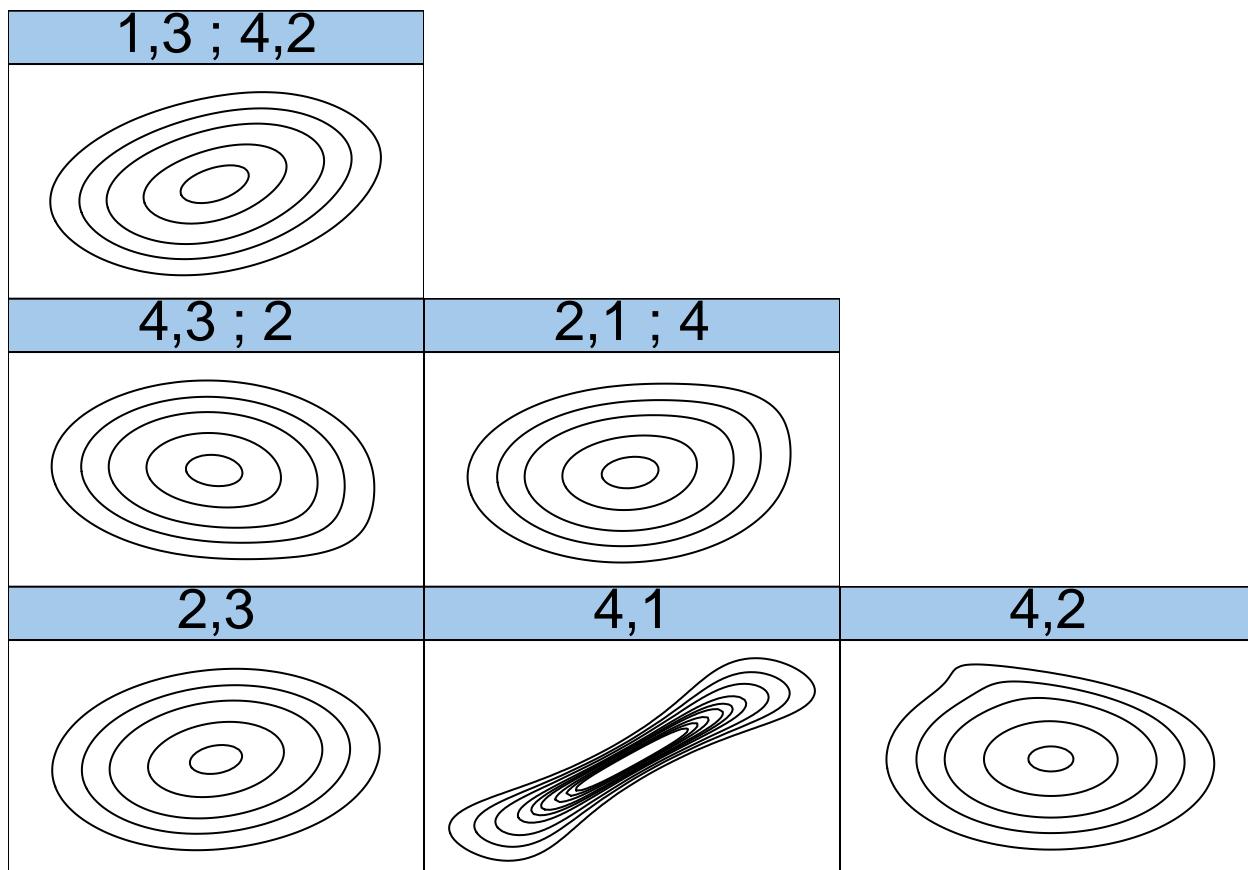
```
## [1] "Best DVine AIC value: -12553.2327902897"
```

Faktycznie jest to struktura o najmniejszym kryterium AIC. Możemy jeszcze dla zasady zobaczyć jej tabelę z kopułami oraz ich wykresy konturowe:

```

## tree      edge | family      cop   par   par2 | tau   utd   ltd
## -----
##    1      2,3 |      5        F   0.76  0.00 |  0.08  -  -
##          4,1 |      5        F  24.02  0.00 |  0.84  -  -
##          4,2 |    224  Tawn2_90 -1.61  0.05 | -0.04  -  -
##    2      4,3;2 |     40    BB8_270 -1.21 -0.94 | -0.08  -  -
##          2,1;4 |     10    BB8    1.23  0.93 |  0.08  -  -
##    3    1,3;4,2 |     10    BB8    3.67  0.43 |  0.18  -  -
##  ---
## type: D-vine  logLik: 6286.62    AIC: -12553.23    BIC: -12488.91
## ---
## 1 <-> Temperatura_powietrza,  2 <-> Średnia_prędkość_wiatru,  3 <-> Suma_opadów,
## 4 <-> Temperatura_gruntu

```



Jak widać kształty ponownie są analogiczne, tylko znów porozrzucane w innych miejscach.

### Regresja kwantylowa

Ponieważ z punktu wyżej wynikło, że najlepszą strukturą jest struktura D-vine to operować będzie na funkcji `vinereg`, która dopasowuje do moich danych model regresji kwantylowej ekskluzywnie ze strukturami D-vine. Pierwszym krokiem jest dopasowanie modelu, podanie mu zmiennych predykcyjnych (`x`) oraz zmiennej objaśnianej (`y`). Cały projekt skupia się na analizach wokół temperatury, więc to ona będzie tutaj zmienną objasnianą przez pozostałe zmienne wprowadzone w tej części. Zatem dopasowuje model parametryczny i patrze, które zmienne mają w nim największy wpływ na wartość temperatury powietrza:

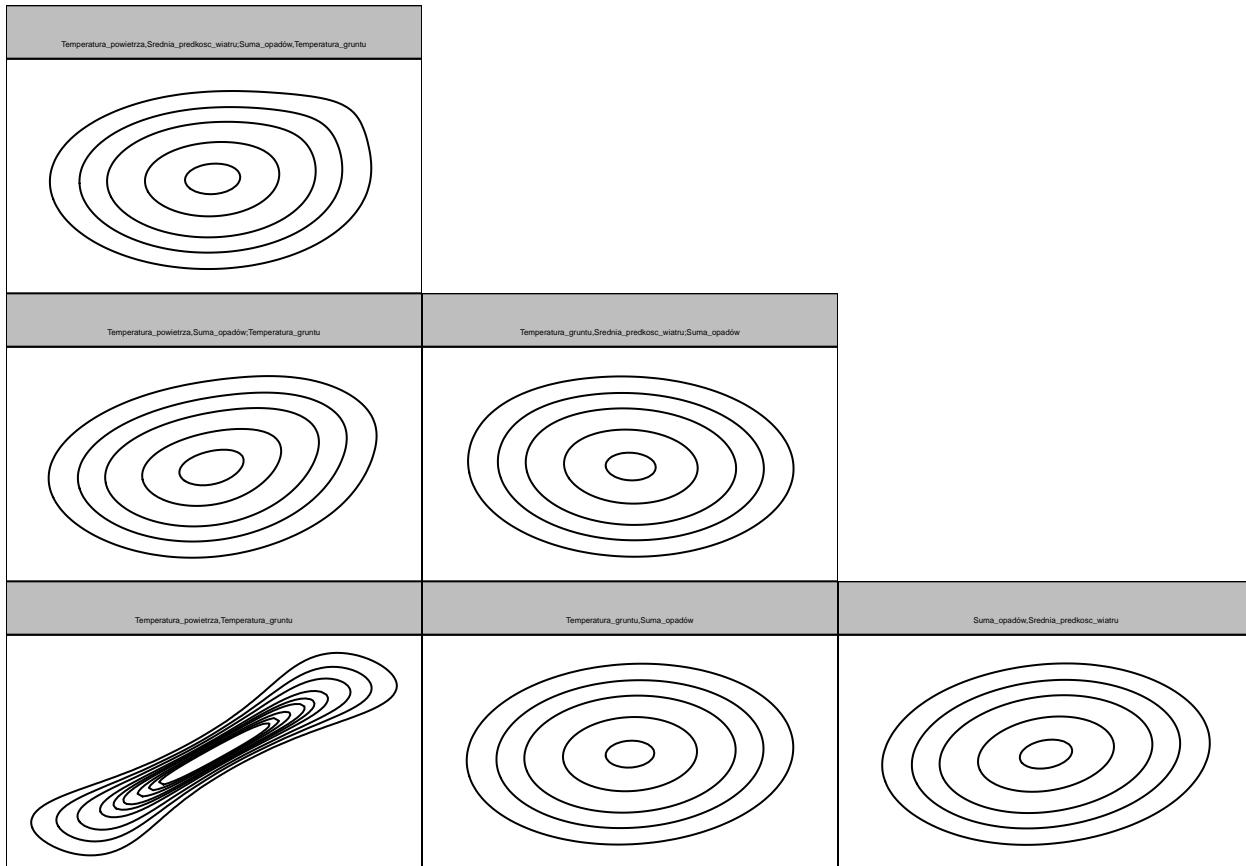
```
## [1] "Zmienne w kolejności z największym wpływem na zmienną objaśnianą:"
```

```
## [1] "Temperatura_gruntu"      "Suma_opadów"
## [3] "Średnia_prędkość_wiatru"
```

Czyli zmienną najmocniej wpływającą na wartość temperatury jest temperatura gruntu, a najmniej wpływa średnia prędkość wiatru. Zgadza się to z intuicyjnym myśleniem na temat tych zmiennych, jak i z współczynnikami korelacji wyświetlnymi przy obrabianiu danych. Ponieważ regresja kwantylowa z pakietu vinereg opiera się na strukturach D-vine, mogę ponownie przyjrzeć się ich budowie w postaci tabeli:

```
## # A data.frame: 6 x 11
##   tree edge conditioned conditioning var_types   family rotation parameters df
##   1    1      1, 2                   c,c     frank      0       24  1
##   1    2      2, 4                   c,c   gaussian      0      0.073  1
##   1    3      4, 3                   c,c     frank      0      0.77  1
##   2    1      1, 4                  2       c,c      bb8      0 1.77, 0.78  2
##   2    2      2, 3                  4       c,c      bb8     90 1.12, 0.82  2
##   3    1      1, 3                  4, 2       c,c      bb8      0 1.13, 0.98  2
##   tau loglik
## 0.843 5970.8
## 0.046 10.6
## 0.085 38.2
## 0.147 132.1
## -0.029 5.5
## 0.059 34.7
```

Oraz spojrzeć na wykresy koturowe kopuł między danymi:

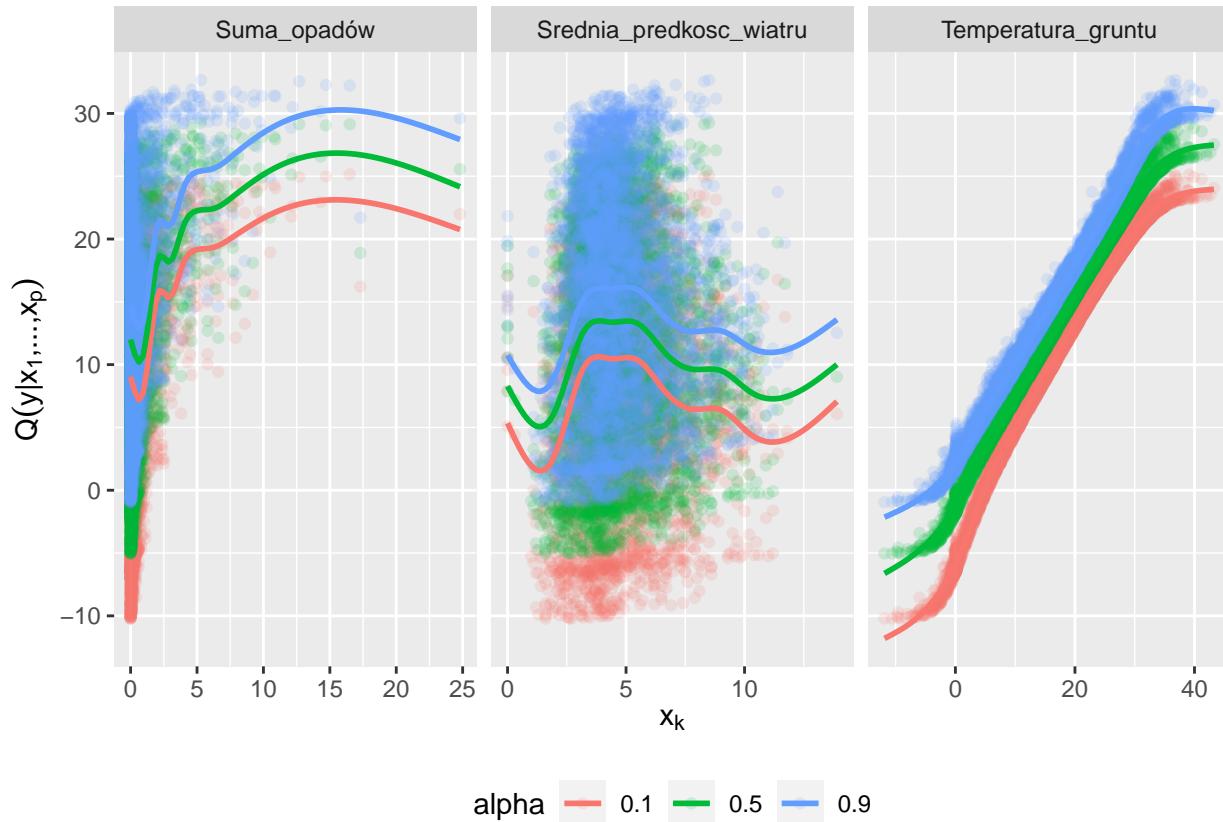


Ponownie mamy do czynienia z bardzo podobnymi kształtami co w poprzednich dopasowanych Vine'ach. Tym razem jednak jest to model regresji kwantylowej, który pozwala mi przewidywać kwantyle temperatury powietrza na podstawie wartości pozostałych zmiennych. Stwórzmy więc wartości predykcje modelu dla trzech rzędów kwantyle: 0.1, 0.5 i 0.9:

```
##          0.1          0.5          0.9
## 1 -3.745250  0.35716076  3.347728
## 2 11.824713 14.38742794 16.836287
## 3 -4.201134  0.02887948  3.115649
## 4 -2.064997  1.85989560  5.127810
## 5 12.766254 15.10503470 17.220857
## 6 10.436722 13.06386121 15.882402
```

Mając już nasze predykcje możemy je nanieść na wykres i zobaczyć zależności pomiędzy prawdziwymi kwantylami a tymi wyestymowanymi:

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Kształty wykresów mogą początkowo wydawać się dziwne, ale są one odpowiednie do rodzajów danych jakie tu wzięlem. Dogłębniejszą analizę wykresów zrobie przy podsumowaniu porównując wykresy modelu parametrycznego z nieparametrycznym.

Teraz powtórze wszystkie poprzednie kroki, tylko że dla modelu nieparametrycznego. Wpierw kolejno zmiennie z największym wpływem na zmienną objaśnianą:

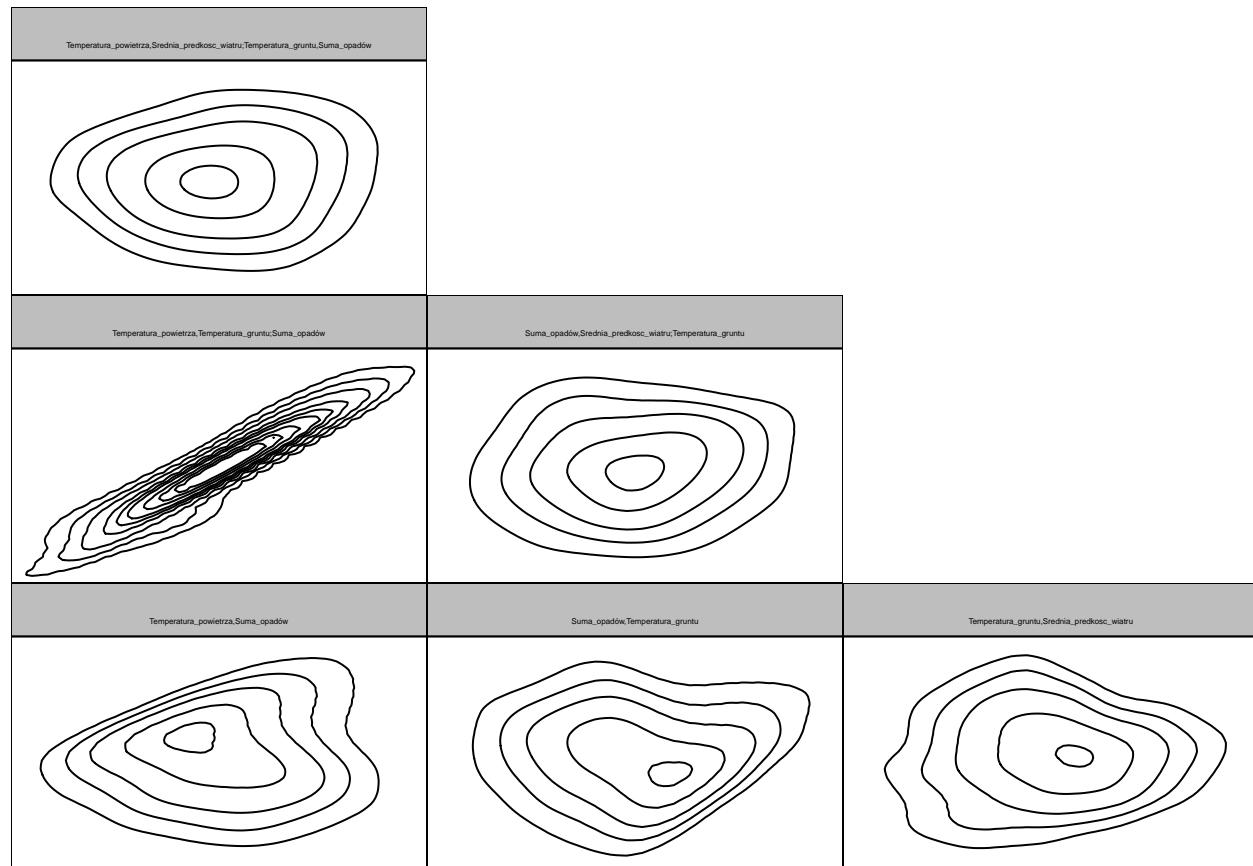
```
## [1] "Zmienne w kolejności z największym wpływem:"
```

```
## [1] "Suma_opadów"           "Temperatura_gruntu"
## [3] "Średnia_prędkość_wiatru"
```

Już tutaj jest zmiana w porównaniu do modelu parametrycznego - najbardziej wpływową zmienną w tym modelu jest suma opadów, a najmniej wpływową ponownie średnia prędkość wiatru. Dalej struktura vine dla modelu:

```
## # A data.frame: 6 x 11
##   tree edge conditioned conditioning var_types family rotation parameters df
##   1    1      1, 4             c,c    tll      0 [30x30 grid] 23
##   1    2      4, 2             c,c    tll      0 [30x30 grid] 19
##   1    3      2, 3             c,c    tll      0 [30x30 grid] 42
##   2    1      1, 2             4       c,c    tll      0 [30x30 grid] 83
##   2    2      4, 3             2       c,c    tll      0 [30x30 grid] 30
##   3    1      1, 3             2, 4   c,c    tll      0 [30x30 grid] 23
##   tau loglik
##   0.032    387
##  -0.002   321
##  -0.019   224
##  0.809   5887
##  0.068    77
##  0.035    77
```

Widać, że niektóre kolumny różnią się strukturą względem modelu parametrycznego, co nie powinno dziwić gdyż stosujemy tutaj zupełnie inne podejście. Co ciekawsze jest fakt, że każda z kopuł tutaj wyszła tego samego typu: typu tll. Wykresy konturowe powyższych kopuł:

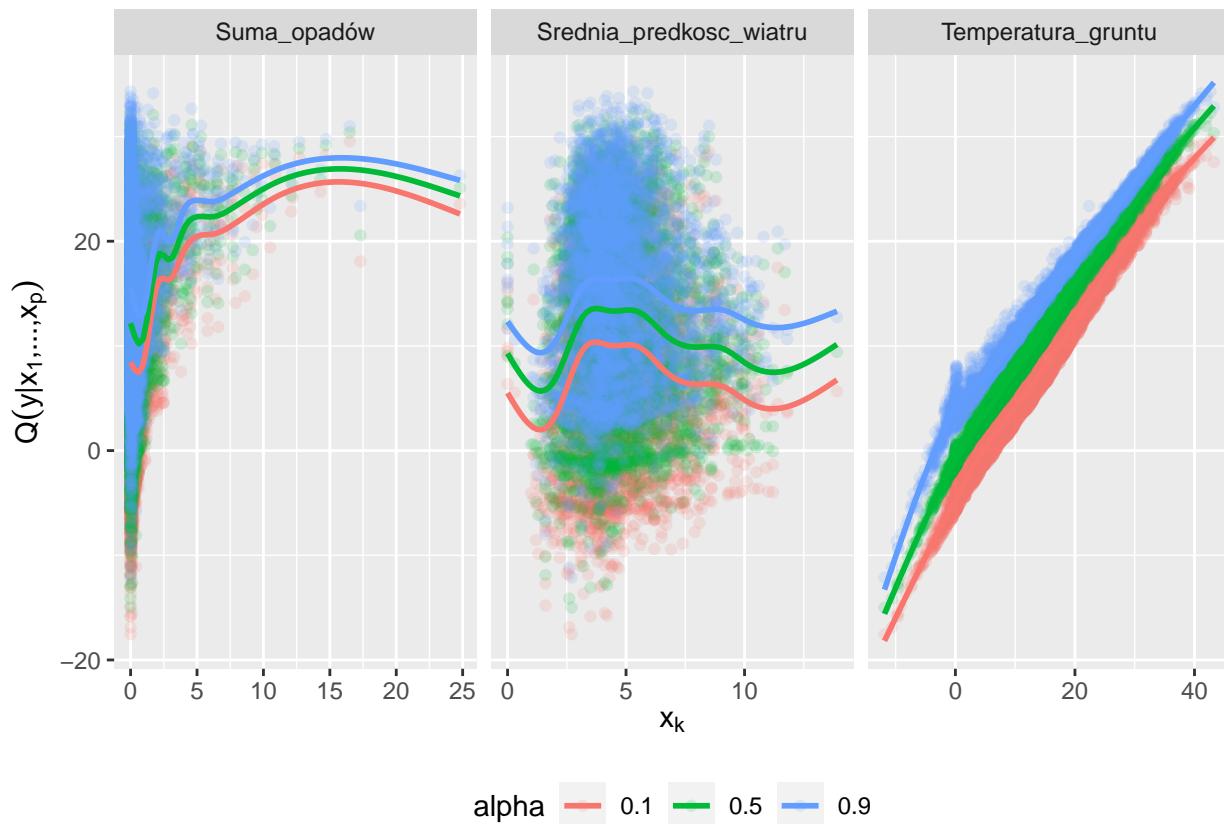


Na pierwszy rzut oka znowu widać to samo co poprzednio: podobne kształty tylko w innych kratkach ze zmieinnymi, jednak można spostrzec że wiele z kopuł tutaj jest o wiele mniej okrągłych i są bardziej niregularne niż te parametryczne. Zobaczmy czy na wykresie zależności również da się dostrzec jakieś zmiany. Wpierw predykcje nieparametrycznego modelu:

```
##          0.1      0.5      0.9
## 1 -3.213800  0.5215095  4.612466
## 2 11.802099 14.6622609 17.245961
## 3 -4.351306  0.1649089  5.175947
## 4 -2.152195  2.0347106  6.852501
## 5 11.383941 15.1340314 17.964000
## 6  9.021426 13.3592391 16.574020
```

I wykres zależności tych predykcji:

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Znów, na pierwszy rzut oka tak zwany różnice ciężko dostrzec. Dopiero jak się zestawi wykresy obu modeli obok siebie, można dostrzec pewne różnice, o których rozpisze się niżej w porównaniu.

## Porównanie i wnioski

Pierwszą rzeczą będzie porównanie i ocena modeli, na podstawie kryterium AIC:

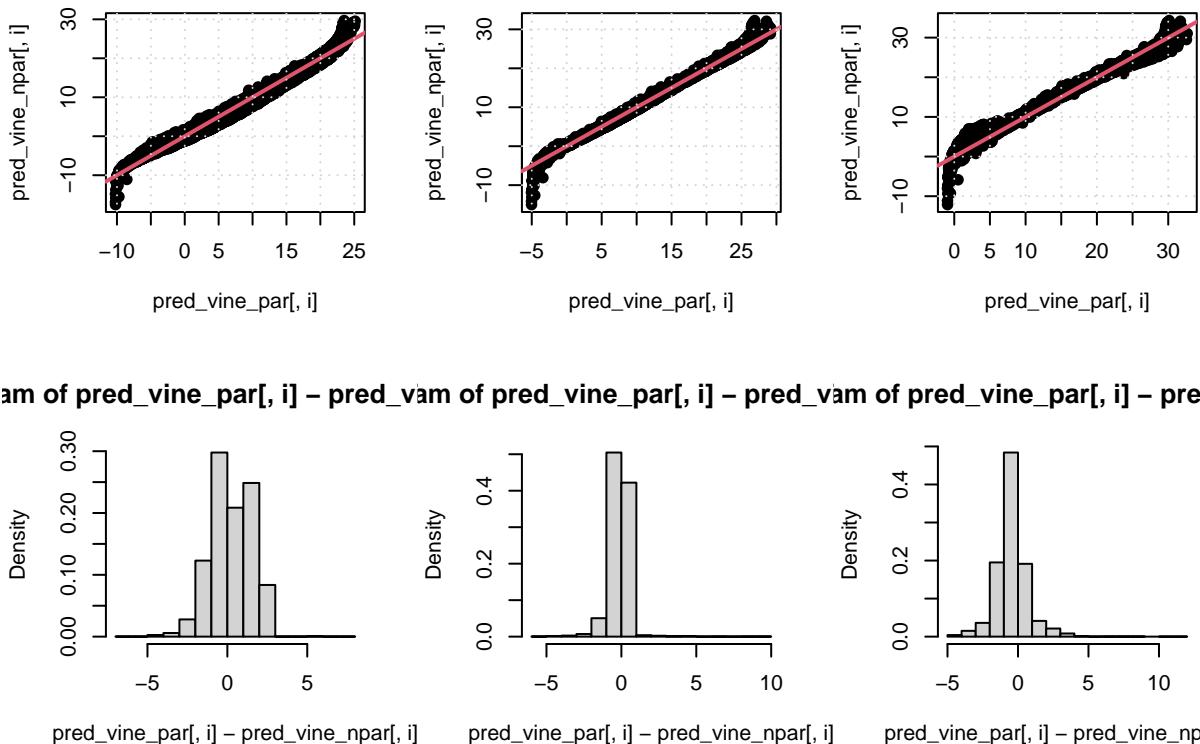
```
## [1] "Metoda parametryczna"
```

```

## [1] "AIC = 20594.531231748"
## [1] "Metoda nieparametryczna"
## [1] "AIC = 20416.0835412005"

```

Kryterium jest mniejsze dla modelu nieparametrycznego, zatem go traktuje jako lepiej dopasowany model. Następnie mogę zróbić bezpośrednie porównanie predykcji wartości modelu parametryczego i nieparametrycznego, oraz sprawdzić ich różnice na histogramie:

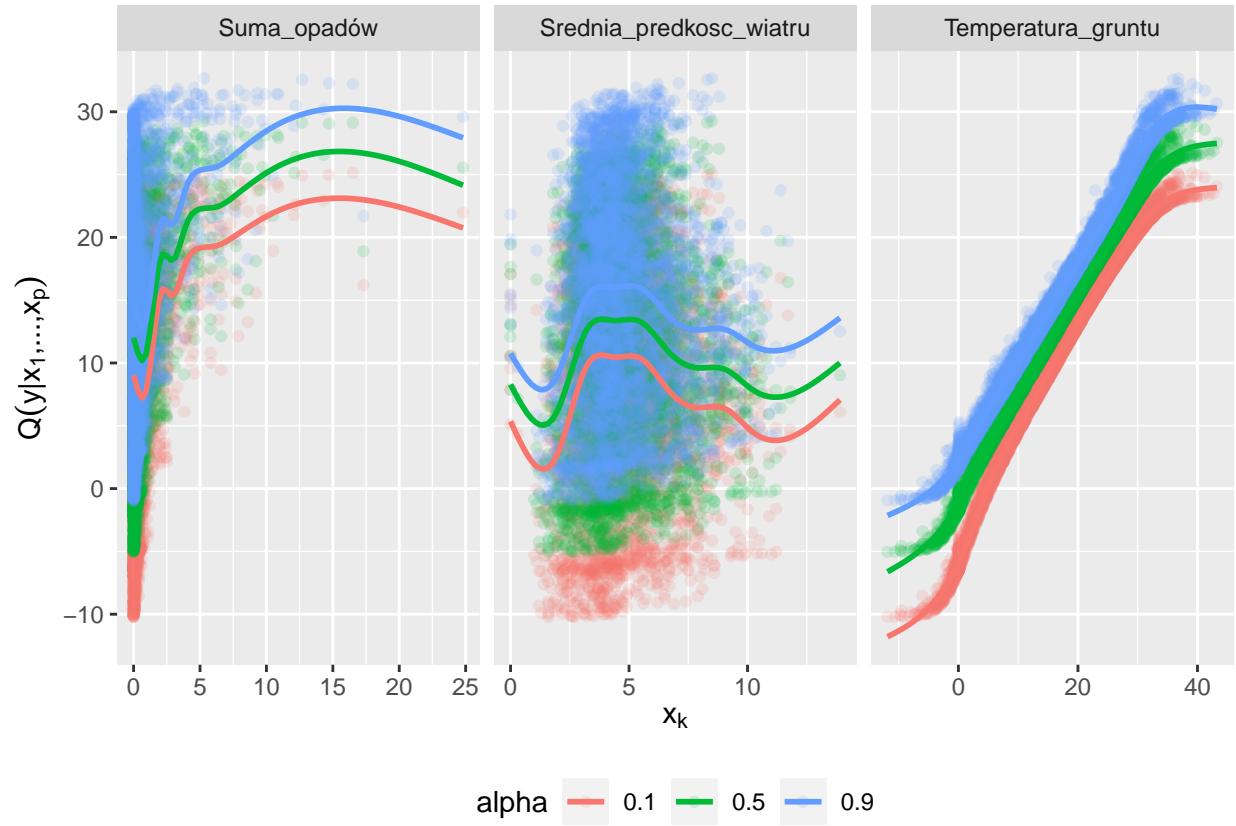


Z obu typów wykresu dla każdej zmiennej wynika, że predykcje metody parametrycznej zdają się mieć lekko większe wartości, ale różnica nie jest znacząca. Czyli skoro jest to model z większą wartością kryterium AIC, to możemy wnioskować, że model parametryczny przeszacowuje wartości temperatury, którym bliżej do predykcji modelu nieparametrycznego. Ostatnim porównaniem, z którego będę wnioskować to zestawienie wykresów zależności dla predykcji kwantylów zmiennych:

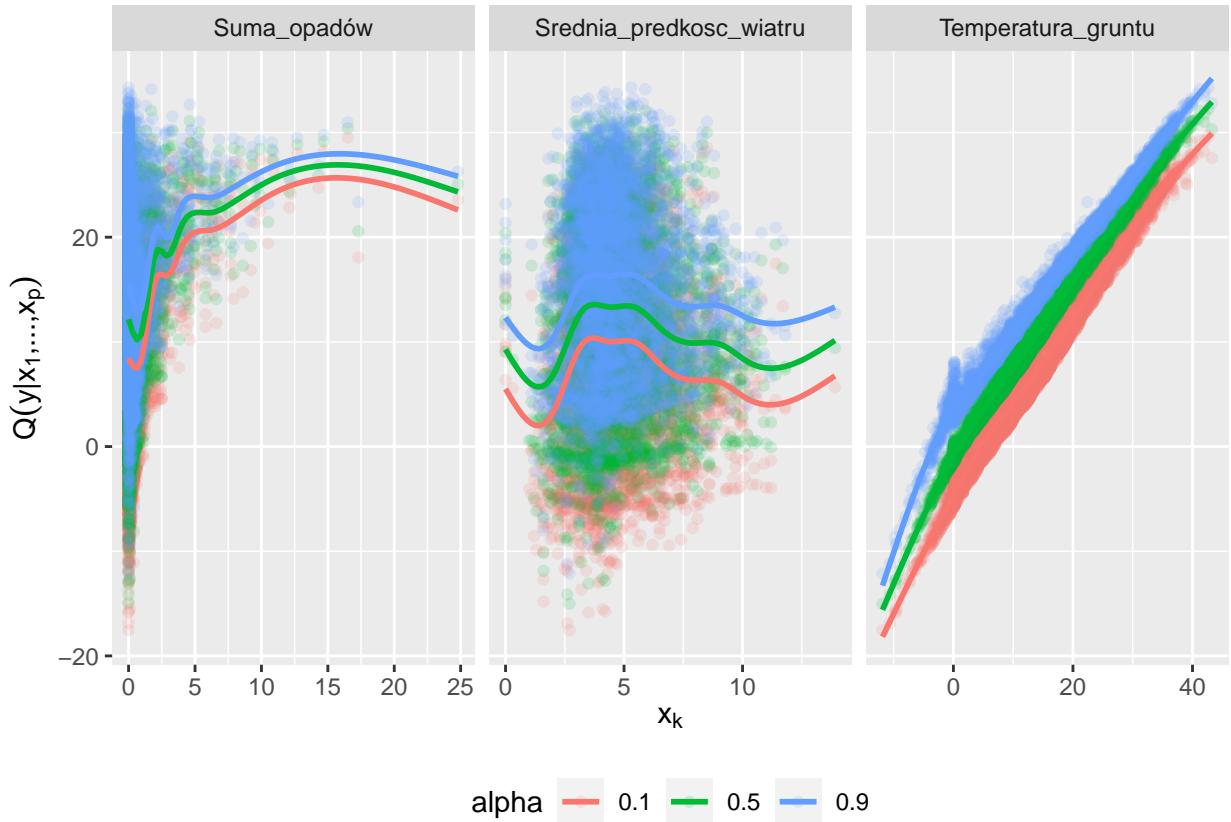
```

## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

```



```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Wykresy dla obu modeli prezentują się dosyć podobnie. Oba modele prezentują liniową zależność między temperaturą gruntu a naszą zmienną objaśnianą czyli temperaturą powietrza. Jest to satysfakcjonujący wynik, gdyż faktycznie te zmienne powinny być mocno i liniowo skorelowane. W przypadku pozostałych dwóch zmiennych ich korelacja z temperaturą powietrza jest widocznie nieliniowa. Suma opadów zdaje się mieć wpływ na wartości temperatury, co też zgadza się z logiką, gdyż deszcz często obniża temperaturę, podczas gdy zależność średniej prędkości wiatru i temperatury zdaje się być dosyć losowa. Zauważalny jest również większy rozstrzał danych i predykcji dla modelu parametrycznego, który może tłumaczyć dlaczego kryterium AIC pokazało go jako gorzej dopasowany model. Zatem można stwierdzić, że jeden typ kopuły występujący w modelu nieparametrycznym oraz jej mniej regularne kształty lepiej dopasowały się do moich danych i dały odrobine lepsze predykcje.