Cwiczenie 2 - R - metody prezentacji danych

Michał Marosz

26 października 2015

Spis treści

Mozliwości pakietu R w kwestii prezentacji wyników są w praktyce ograniczone jedynie naszą wyobraźnią oraz dostepnością pakietów. Poniżej zaprezentowane zostaną jedynie podstawowe możliwości podstawowego pakietu *graphics*. W repozytoriach dostepne sa pakiety takie jak *ggplot* lub *lattice*. Można je zainstalować a poprzez to znacznie wzbogacić repertuar technik graficznej prezentacji wyników, co w klimatologii jest szczególnie istotne ze względu na ilość danych, z którymmi zazwyczaj przychodzi się nam zmierzyć oraz wielość możliwych wariantów wykonywanych analiz.

Wczytywanie danych

Dane w R, jak już wiecie, moga funkcjonować jako różne typy, wszelako najwszechstronniejszym w naszych analizach okazuje się być typ *dataframe*. Załadowanie do pamięci programu obiektu polega na wczytaniu go odpowiednim kodem. Przykładowo:

```
dane=read.table("air.txt", header=T )
```

wczytuje dane z pliku *air.txt* jako typ *dataframe* i jednoczesnie nadaje zmiennym nazwy zgodne z treścią pierwszego wiersza pliku tekstowego. Koniecznym jest wcześniejsze ustawienie parametru *Working directory* w celu uniknięcia wpisywania pełnego adresu lokalizacji pliku z danymi. W celu szybkiego sprawdzenia danych można wykonac polecenie *summary*

```
summary(dane)
```

```
##          DATA           YEAR            MC             TEMP
##  1951-01-01:  1   Min.   :1951   Min.   : 1.00   Min.   :-10.000
##  1951-02-01:  1   1st Qu.:1967   1st Qu.: 3.75   1st Qu.:  0.875
```

```
##  1951-03-01:  1   Median :1982   Median : 6.50   Median :  8.000
##  1951-04-01:  1   Mean   :1982   Mean   : 6.50   Mean   :  7.649
##  1951-05-01:  1   3rd Qu.:1998   3rd Qu.: 9.25   3rd Qu.: 14.900
##  1951-06-01:  1   Max.   :2014   Max.   :12.00   Max.   : 22.200
##  (Other)   :762
```

w celu szybszego odwoływania się do danych z obiektu *dataframe* (bez konieczności każdorazowego odwoływania się do pełnej ścieżki zmiennej) możemy zastosowac polecenie *attach*

```
attach(dane)
```

wówczas możemy napisac po prostu

```
mean(TEMP[which(MC==1)])
```

```
## [1] -2.790625
```

i obliczyc średnią temperaturę stycznia. Bez wykorzystanie funkcji *attach* nasz kod bylby nieco zagmatwany i wyglądałby tak:

```
mean(dane$TEMP[which(dane$MC==1)])
```

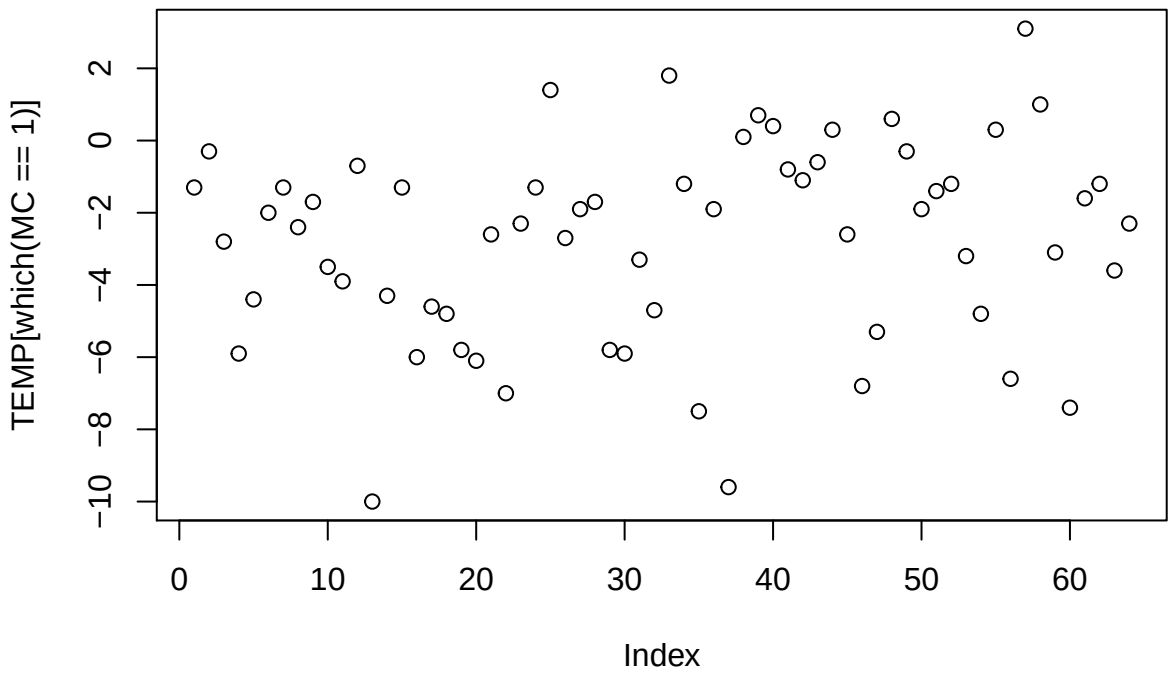
```
## [1] -2.790625
```

Wynik jest oczywiście tożsamy, jednak wieksza złozoność kodu niejednokrotnie powoduje niezamierzone błędy a ich poszukiwanie w gąszczu linii jest zazwyczaj żmudne i czasochłonne.

Wykresy liniowe

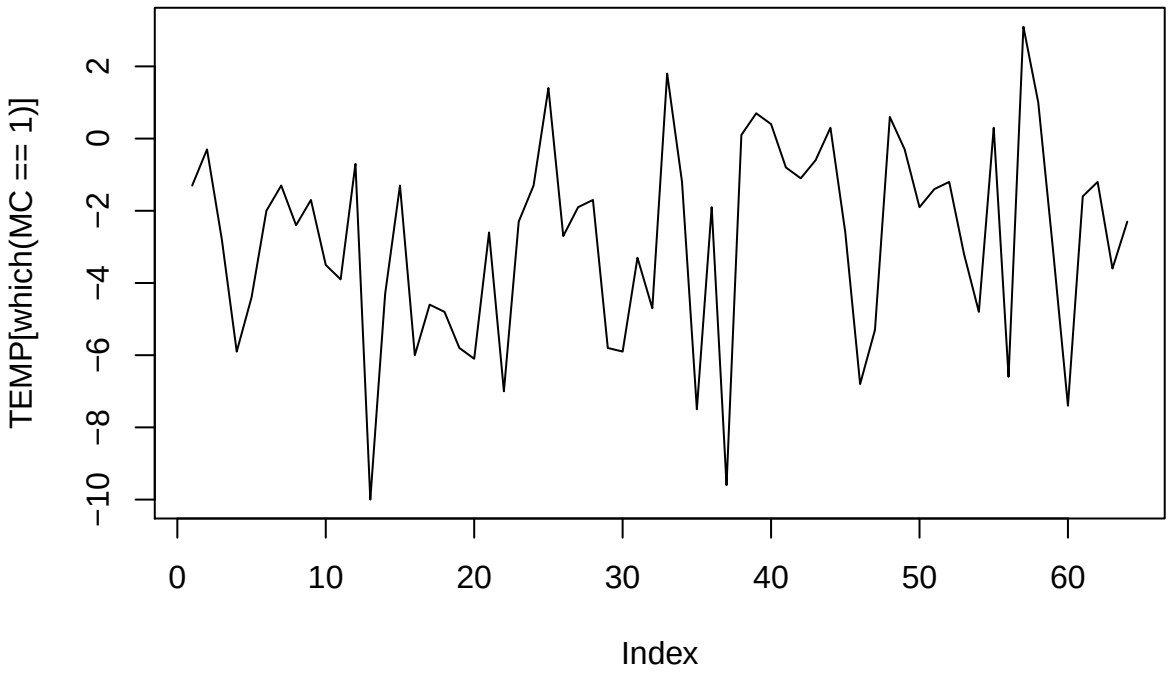
Wykresy liniowe sa jednymi z najcześciej stosowanych w klimatologii. Wprzypadku R generujemy je za pomoca polecenia *plot*. Przykładowo narysujmy przebieg wartości zmiennej *TEMP* dla stycznia

```
plot(TEMP[which(MC==1)])
```



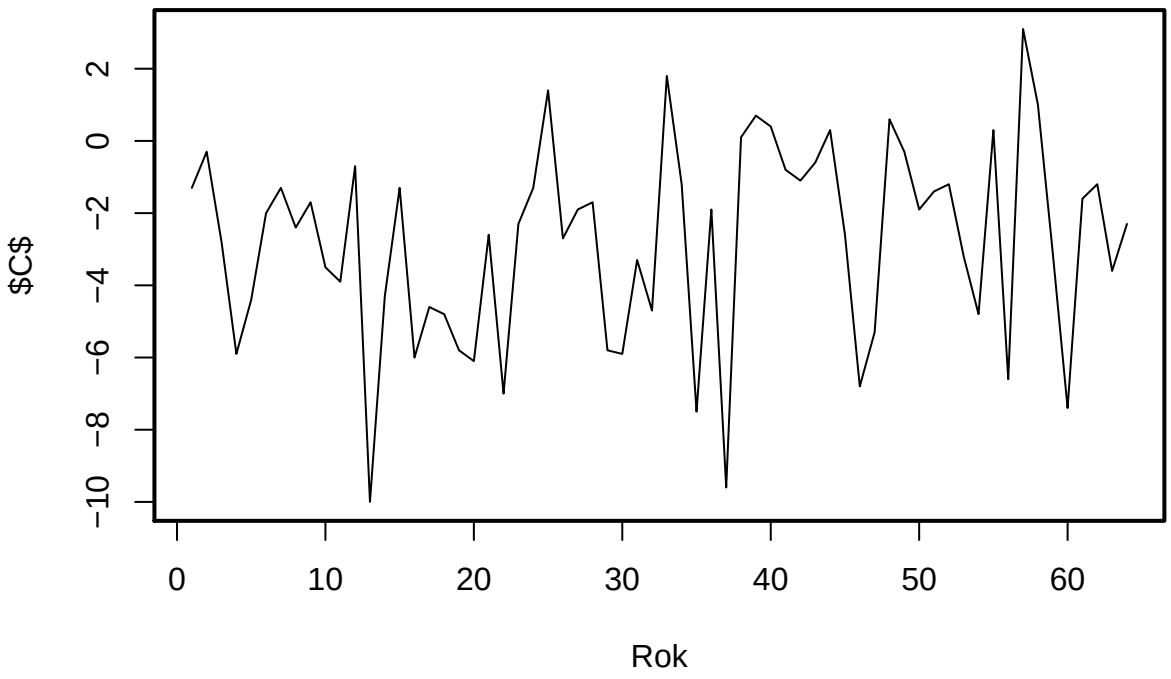
Hmmm, chyba nie o to do końca nam chodziło. Podstawowa wersja funkcji *plot* bez żadnych argumentów wywołuje wykres punktowy. Typ wykresy można zmieniać poprzez argument *type*

```
plot(TEMP[which(MC==1)], type="l")
```



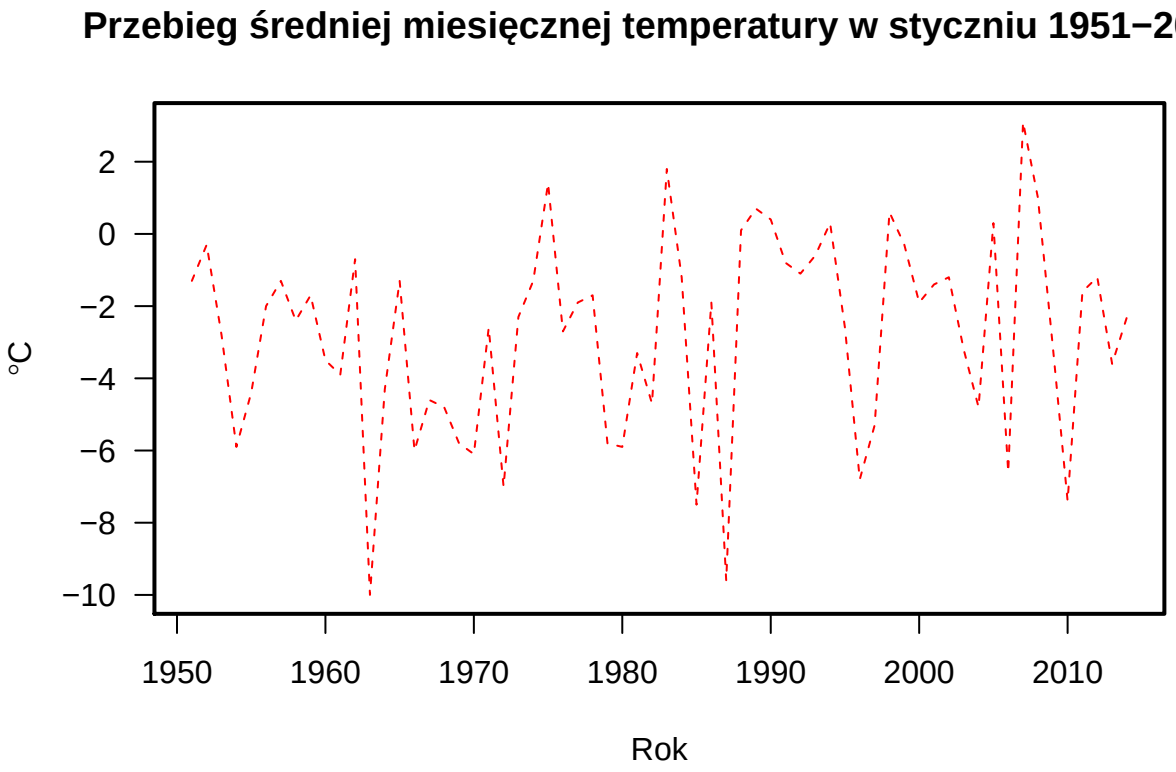
Dalsza praca nad wykresem w celu uczynienia go zdatnym do wykorzystania w publikacji będzie obejmować podpisanie osi, dodanie tytułu wykresy (chociaż w przypadku publikacji naukowych nie jest to zawsze wymagane - wszak podpisy znajduja sie w tytule), dodamy również kolejne lata jako wartości na osi x. Wszystkie te dodatkowe informacje mozna dodać w jednej linii a kolejne argumenty oddzielamy przecinkami. Jeżeli arumentów jest znaczna ilość warto podzielic linię na kilka wierszy. Zwiększy to czytelność kodu

```
plot(TEMP[which(MC==1)], type="l", xlab="Rok", ylab="$C$",
     main="Przebieg średniej miesięcznej temperatury w styczniu 1951-2014")
box(lwd=2)
```



Ostatnia linia kodu dodaje dodatkowe pogrubione obramowanie wykresu Wciąż nie mamy lat jako wartości na osi X. Dodamy je przekształcając nasz wykres w wykres rozrzutu.

```
plot(1951:2014,TEMP[which(MC==1)], type="l", xlab="Rok",
     ylab=expression(degree*C),
     main="Przebieg średniej miesięcznej temperatury w styczniu 1951-2014",
     las=1,
     col=2,
     lty=2
)
box(lwd=2)
```

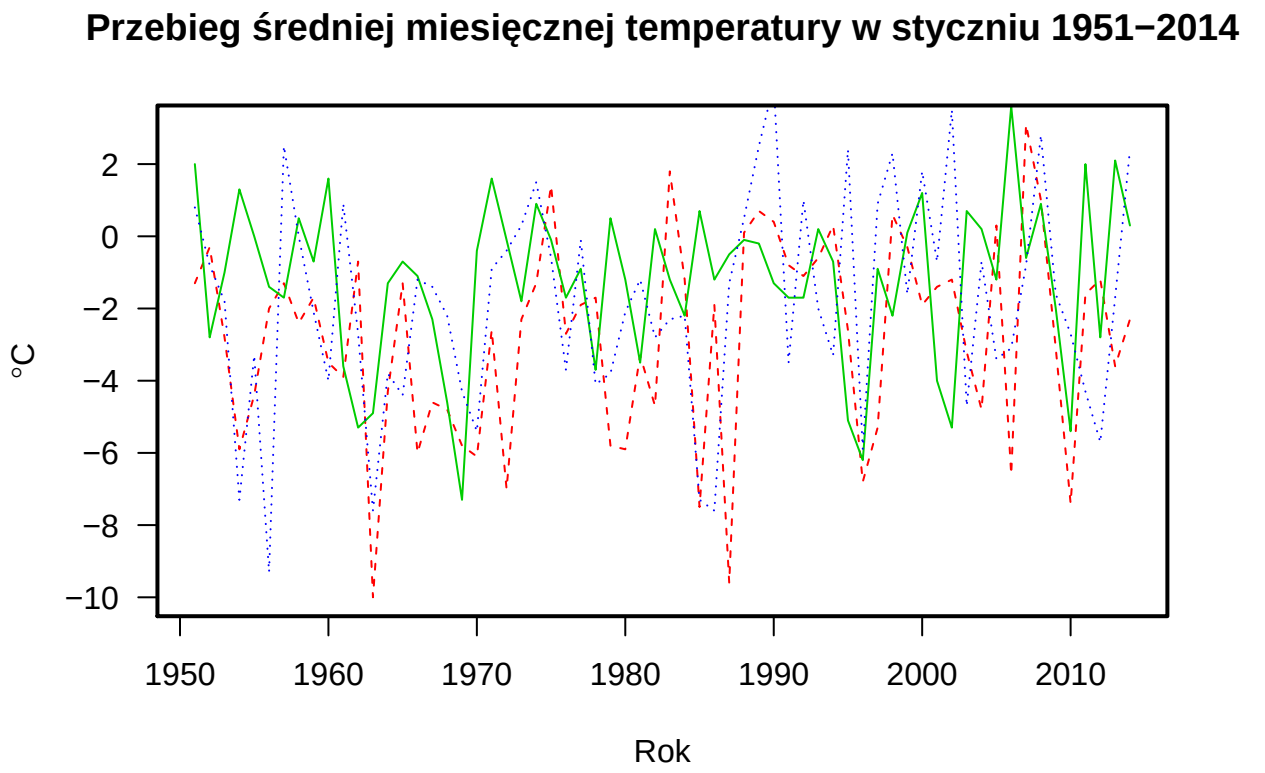


W powyższym przykładzie zastosowano dodatkowe parametry: *las* obracający etykiety, *col* ustalający kolor linii oraz *lty* określający typ linii. W tym wypadku 2 oznacza linię przerywaną, 1 - linię ciągłą, 3 - kropkowaną.

Do istniejących już wykresów można dodawac dodatkowe zmienne wykorzystując funkcję *lines*. Np. dodajmy przebieg wartości temperatury z grudnia (zieloną linią ciągłą) i lutego (niebieską linia kropkowaną)

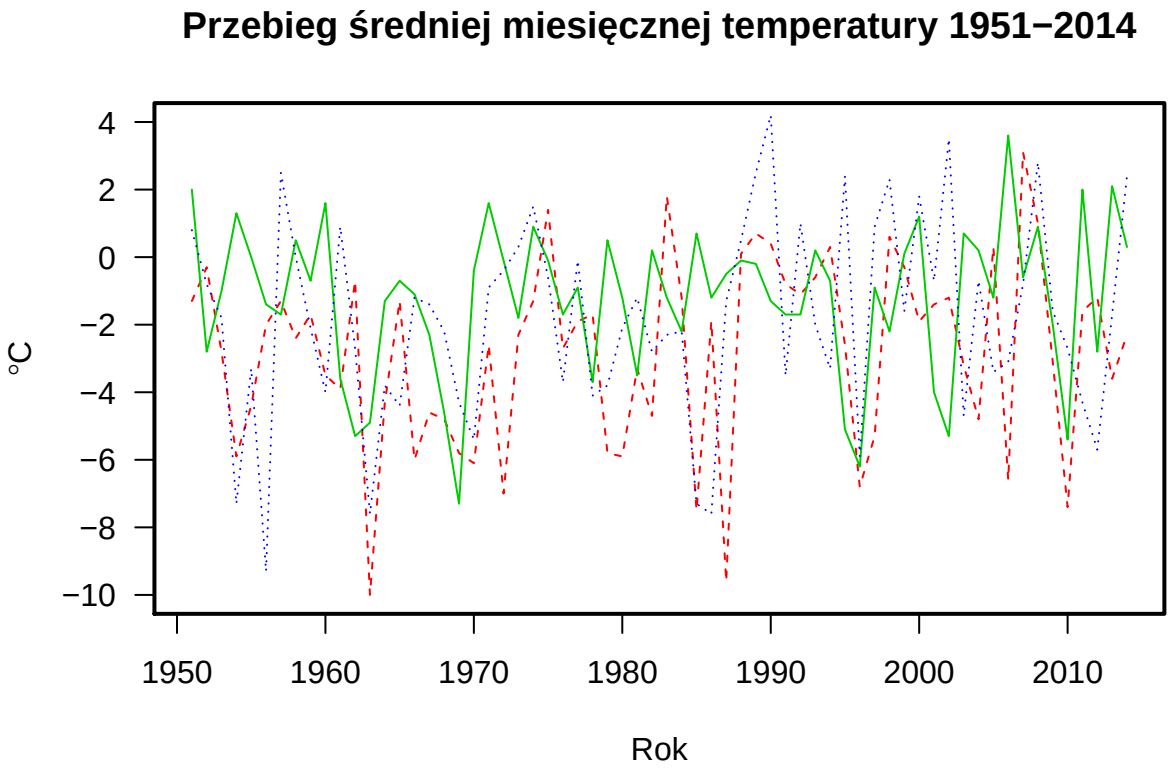
```
plot(1951:2014,TEMP[which(MC==1)], type="l", xlab="Rok",
     ylab=expression(degree*C),
     main="Przebieg średniej miesięcznej temperatury w styczniu 1951-2014",
     las=1, col=2, lty=2)
lines(1951:2014,TEMP[which(MC==12)], type="l",
     las=1, col=3, lty=1)

lines(1951:2014,TEMP[which(MC==2)], type="l",
```

```
     las=1, col=4, lty=3)
box(lwd=2)
```



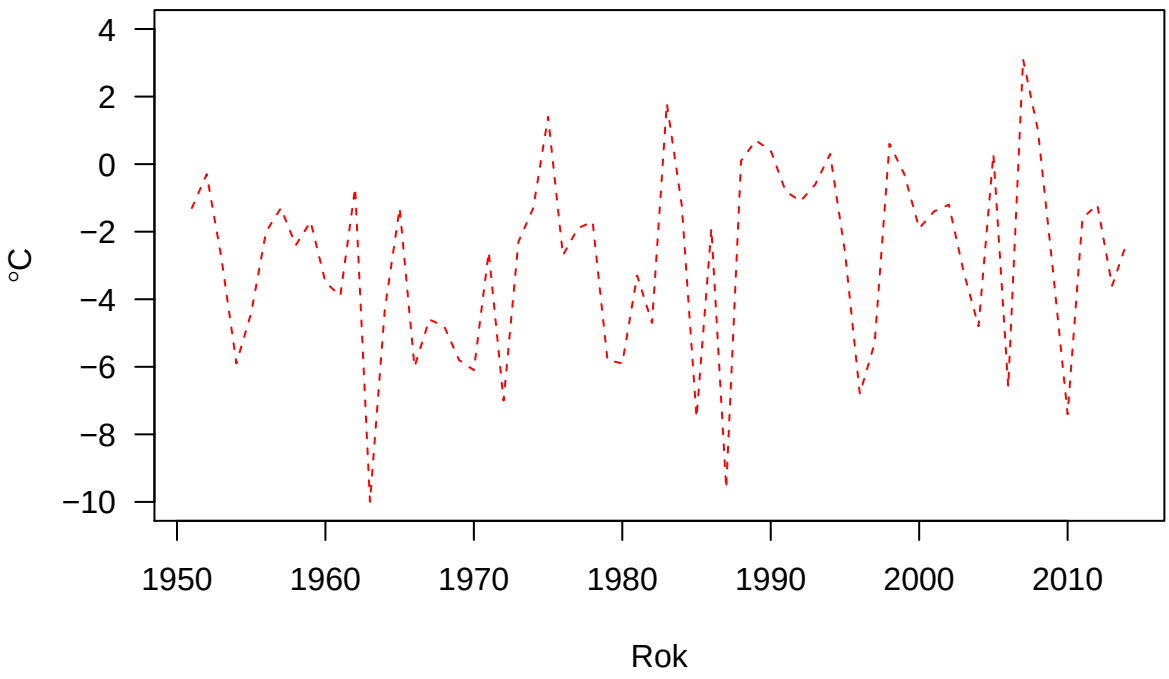
W przyapdku danych z lutego niestety rok 1990 nie zmieścil sie na wykresie. musimy zatem użyć dodatkowego parametru *ylim* pozwalającego na ustalenie zakresu wartości na osiach. Uzupełniony kod dla finalnego wykresu wyglądał będzie nastepująco:

```
plot(1951:2014,TEMP[which(MC==1)], type="l", xlab="Rok",
     ylab=expression(degree*C),
     main="Przebieg średniej miesięcznej temperatury 1951-2014",
     las=1, col=2, lty=2, ylim=c(-10, 4) )
lines(1951:2014,TEMP[which(MC==12)], type="l", las=1, col=3, lty=1)
lines(1951:2014,TEMP[which(MC==2)], type="l", las=1, col=4, lty=3)
box(lwd=2)
```



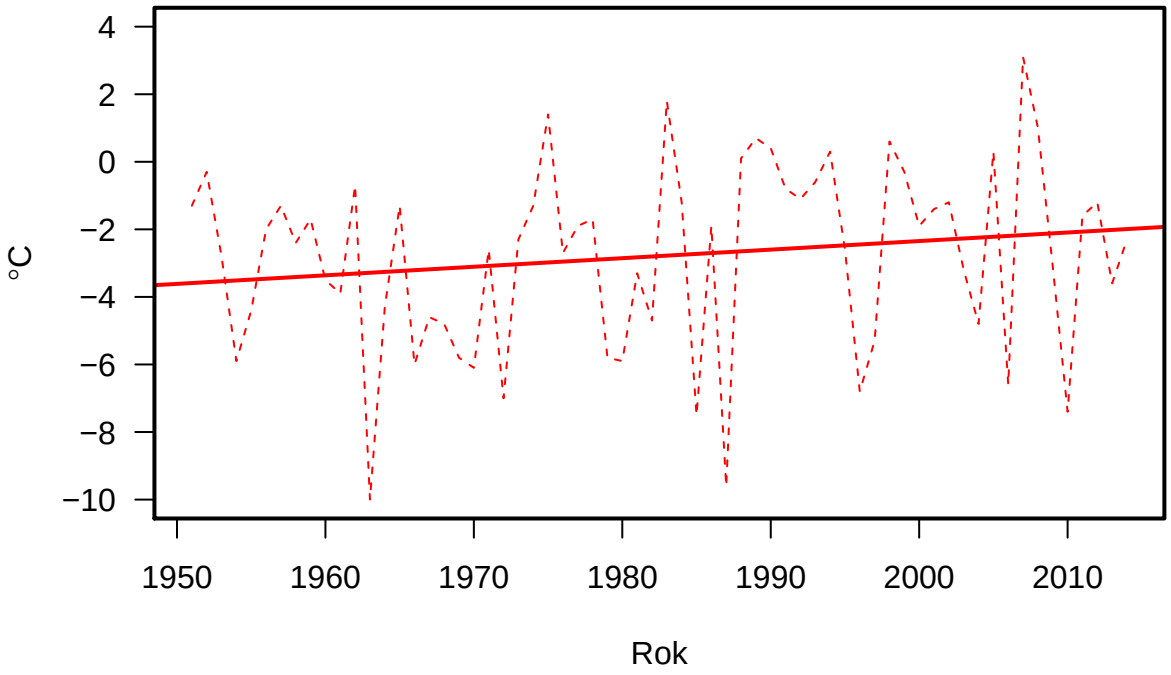
Wróćmy do wykresu dla stycznia

```
plot(1951:2014,TEMP[which(MC==1)], type="l", xlab="Rok",
     ylab=expression(degree*C),
     main="Przebieg średniej miesięcznej temperatury w styczniu 1951-2014",
     las=1, col=2, lty=2, ylim=c(-10, 4)
)
```



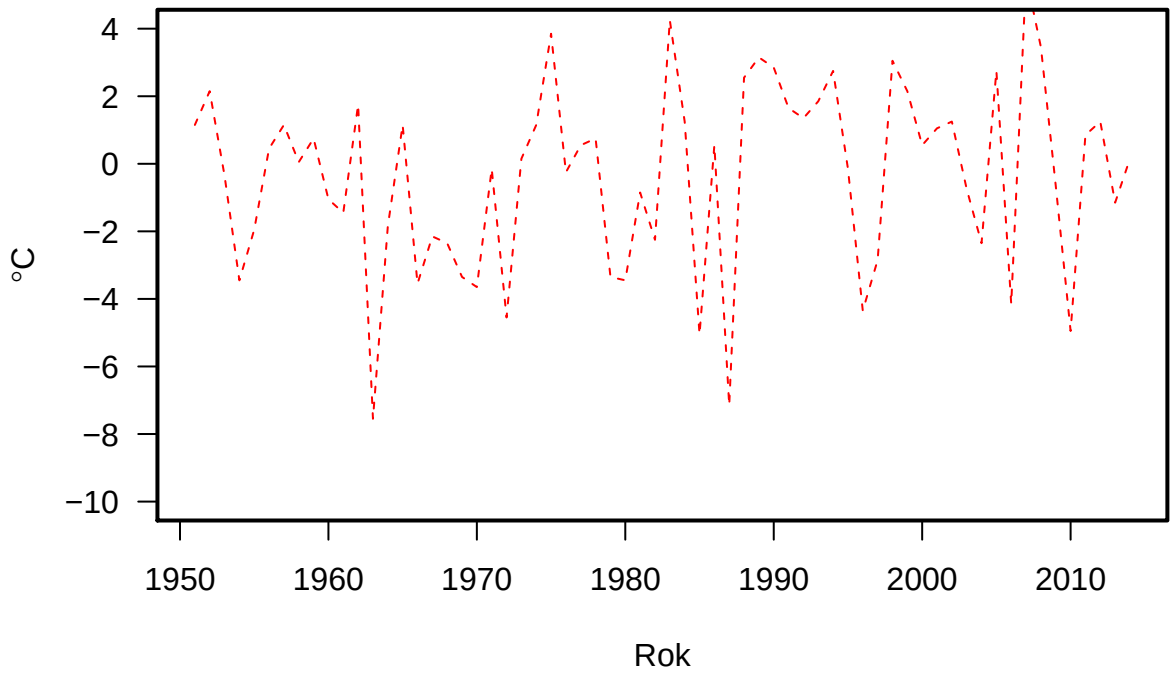
Do takiego wykresu można w bardzo łatwy sposób dodać linię trendu korzystając z polecenia *abline*

```
plot(1951:2014,TEMP[which(MC==1)], type="l", xlab="Rok",
     ylab=expression(degree*C),
     main="", las=1, col=2, lty=2, ylim=c(-10, 4) )
abline(lm(TEMP[which(MC==1)]~c(1951:2014)), col=2, lwd=2)
box(lwd=2)
```



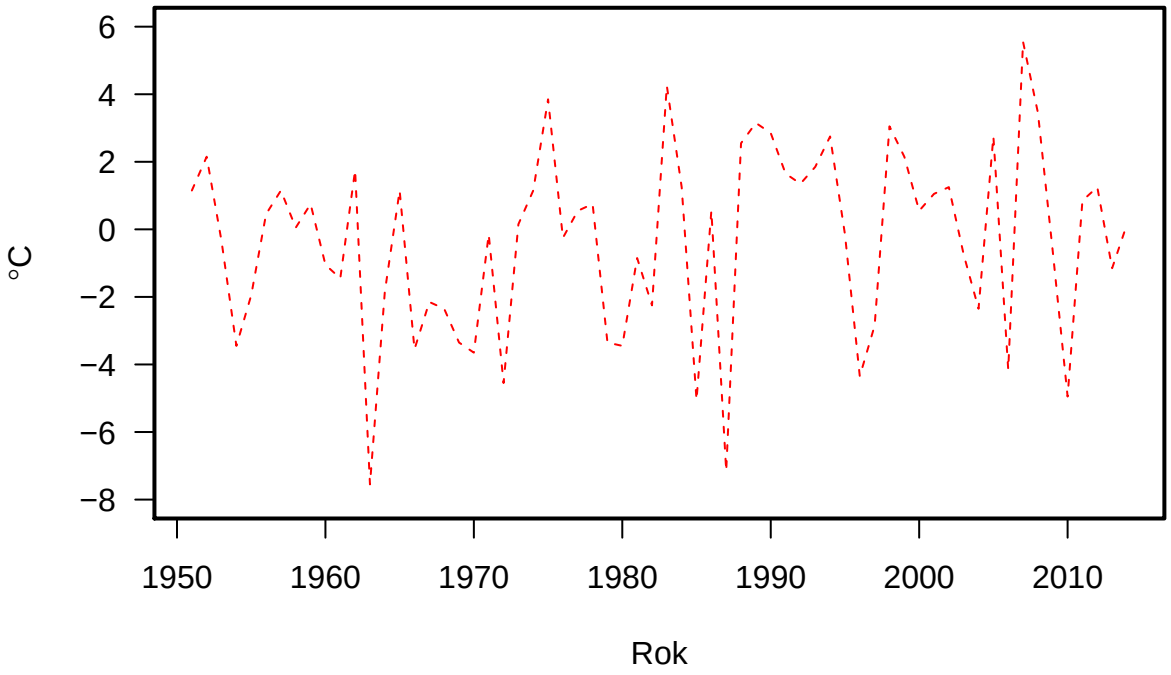
Czasami przydatne jest operaowanie na anomaliach. Wykresślmy zatem ta samą zmienną, ale obrazujący anomalie wartości temperatury w styczniu względem średniej z wielolecia 1971-2000. Można to zrobić w dwóch krokach obliczając najpierw wartość średnią jako obiekt lub bezpośrednio rozszerzając nieco kod odpowiedzialny za wykres.

```
plot(1951:2014,TEMP[which(MC==1)]-mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)]),
     type="l", xlab="Rok", ylab=expression(degree*C),
     main="", las=1, col=2, lty=2, ylim=c(-10, 4) )
box(lwd=2)
```



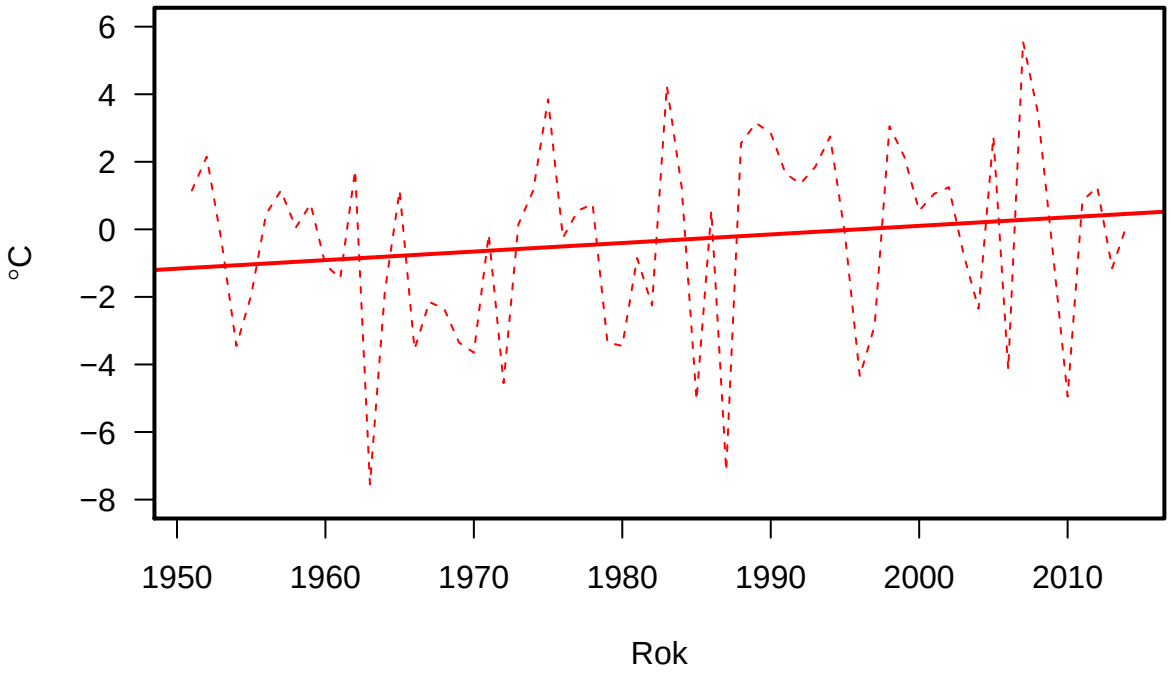
Wykres przesunał nam sie o wartośc średniej i niestety przestał się mieścić dlatego ponownie zmodyfikujemy argument *ylim*

```
plot(1951:2014,TEMP[which(MC==1)]-mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)]),
     type="l", xlab="Rok", ylab=expression(degree*C),
     main="", las=1, col=2, lty=2, ylim=c(-8, 6) )
box(lwd=2)
```



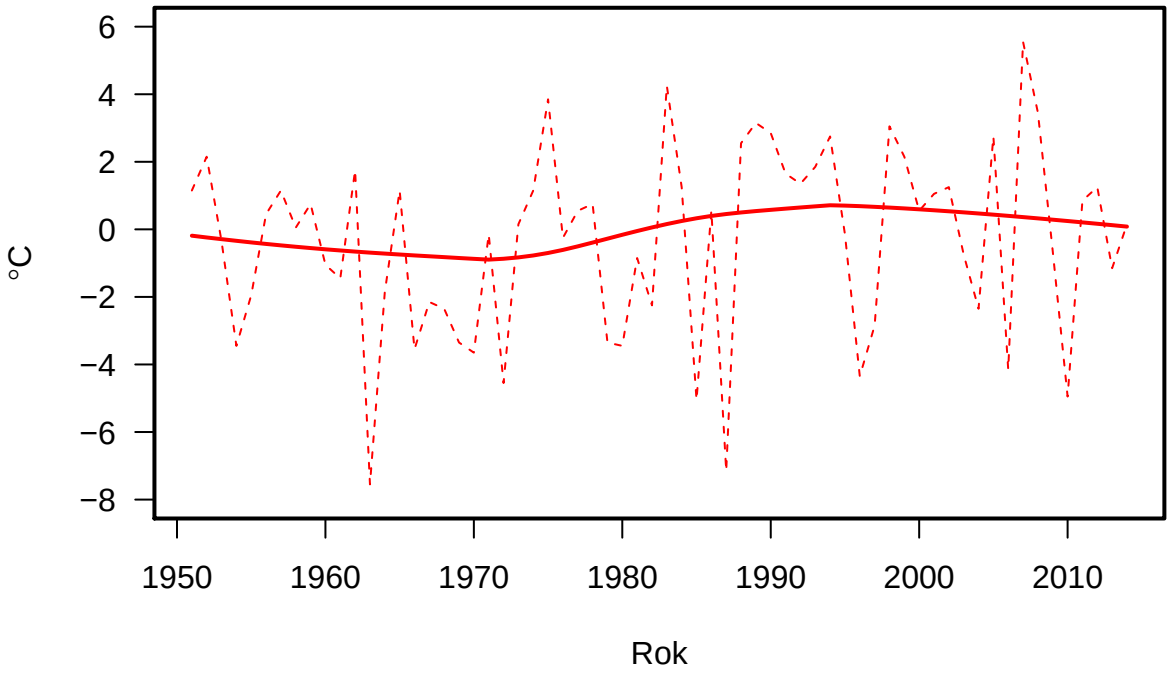
Wtkresy liniowe uzupełnić można jak już wcześniej wspomniałem o linię trendu

```
plot(1951:2014,TEMP[which(MC==1)]-
       mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)]),
     type="l", xlab="Rok", ylab=expression(degree*C),
     main="", las=1, col=2, lty=2, ylim=c(-8, 6) )
abline(lm(TEMP[which(MC==1)]-
            mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)])~c(1951:2014)),
       col=2, lwd=2)
box(lwd=2)
```



lub wygładzić np. funkcją *lowess*, która ma znaczna liczbę dodatkowych argumentów, ale w tym miejscu w celu ograniczenia zamieszania zastosujemy ją w najprostszej postaci.

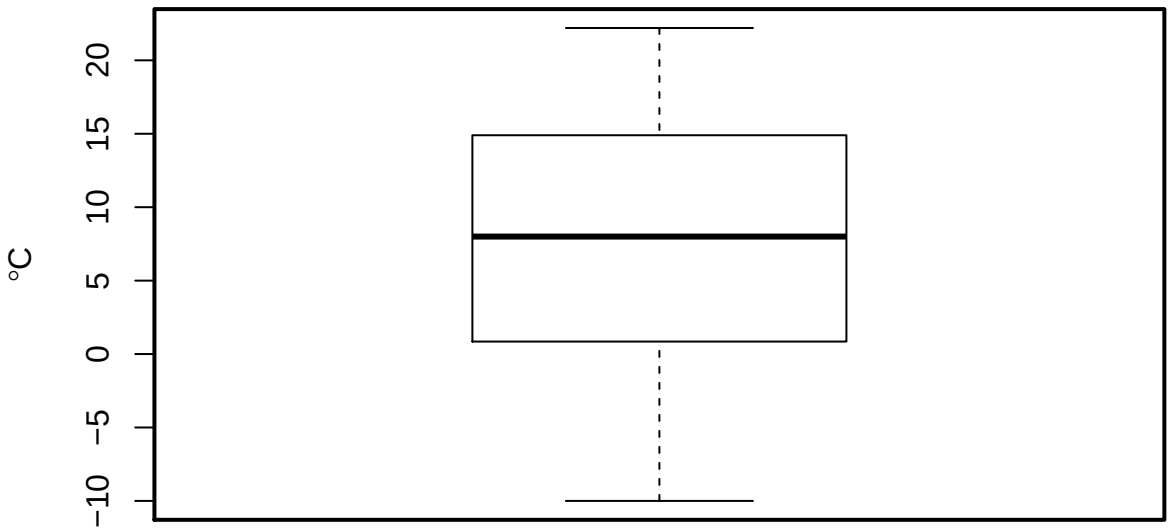
```
plot(1951:2014,TEMP[which(MC==1)]-
       mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)]),
     type="l", xlab="Rok", ylab=expression(degree*C),
     main="", las=1, col=2, lty=2, ylim=c(-8, 6) )
lines(lowess(1951:2014, TEMP[which(MC==1)]-
               mean(TEMP[which(MC==1 & YEAR>=1971 & YEAR<=2000)])),
      col=2, lwd=2)
box(lwd=2)
```



Wykresy pudełkowe

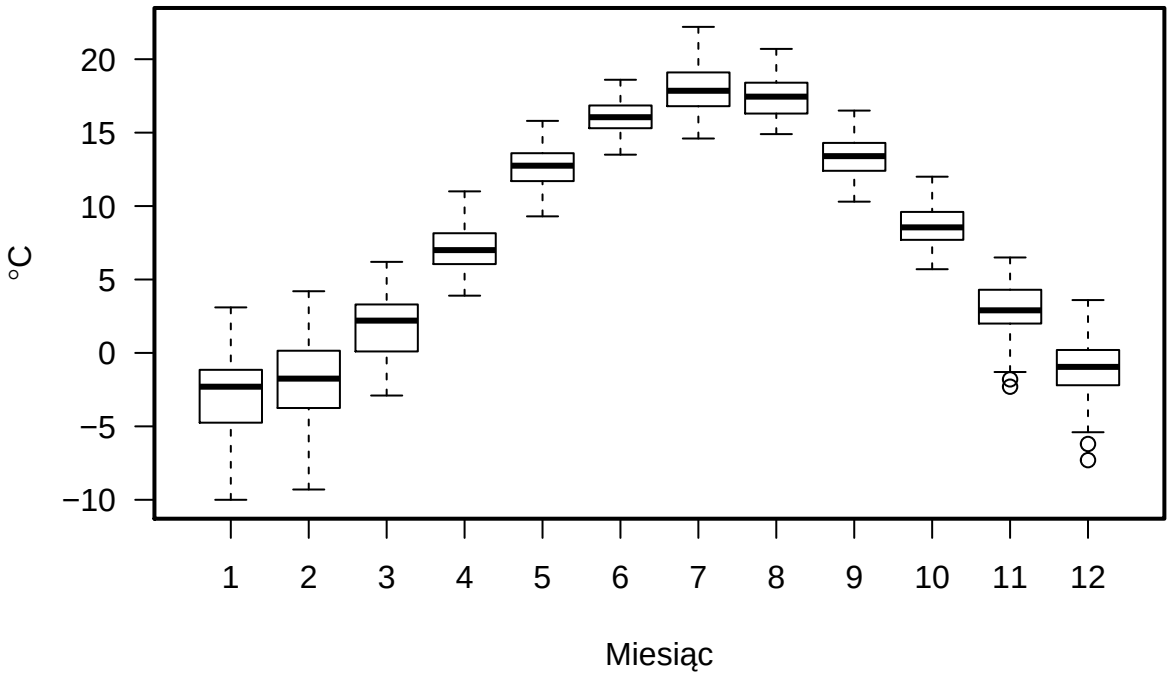
Jeżeli chcemy w szybki sposób przyjrzeć sie strukturze danych (typowi rozkładu, określonym wartościom) mozna posłużyć się wykresem pudełkowym *boxplot*.

```
boxplot(TEMP, ylab=expression(degree*C))
box(lwd=2)
```



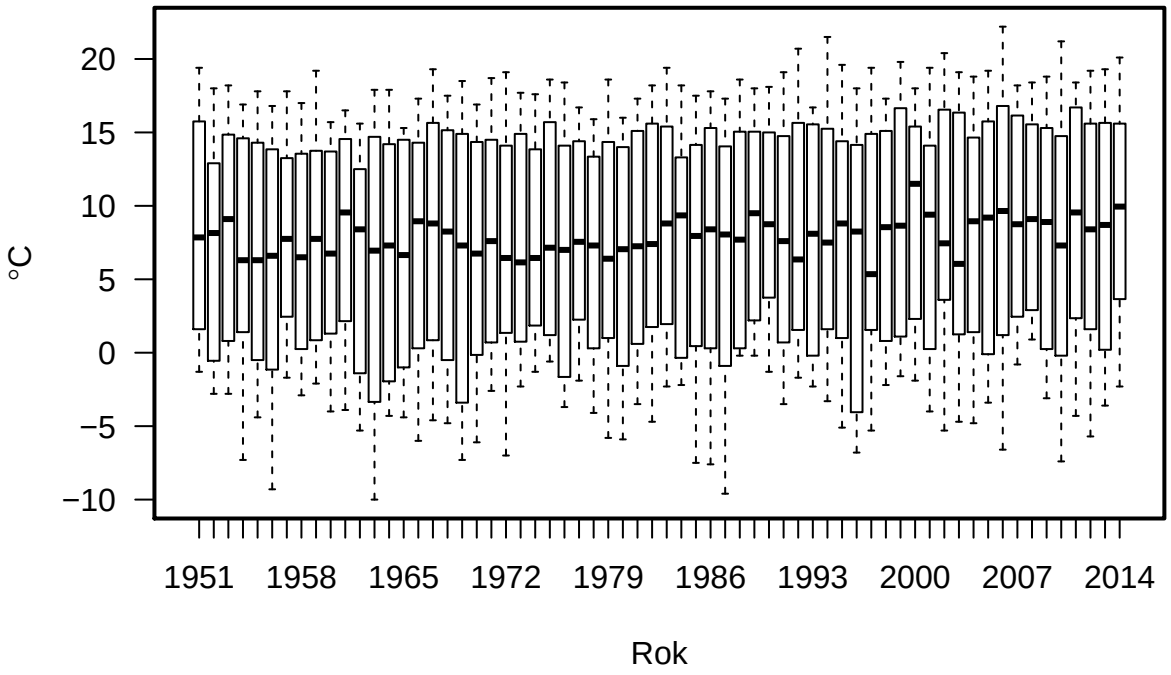
Jeżeli zalezy nam na analizie zmienności rocznej/cyklu rocznego, lub analizy według innego czynnika można uzupełnić komendę o *~CZYNNIK*, gdzie CZYNNIK oznacza zmienną grupującą. Przykładowo dla miesięcy:

```
boxplot(TEMP~MC, ylab=expression(degree*C), xlab="Miesiąc", las=1)
box(lwd=2)
```



lub lat:

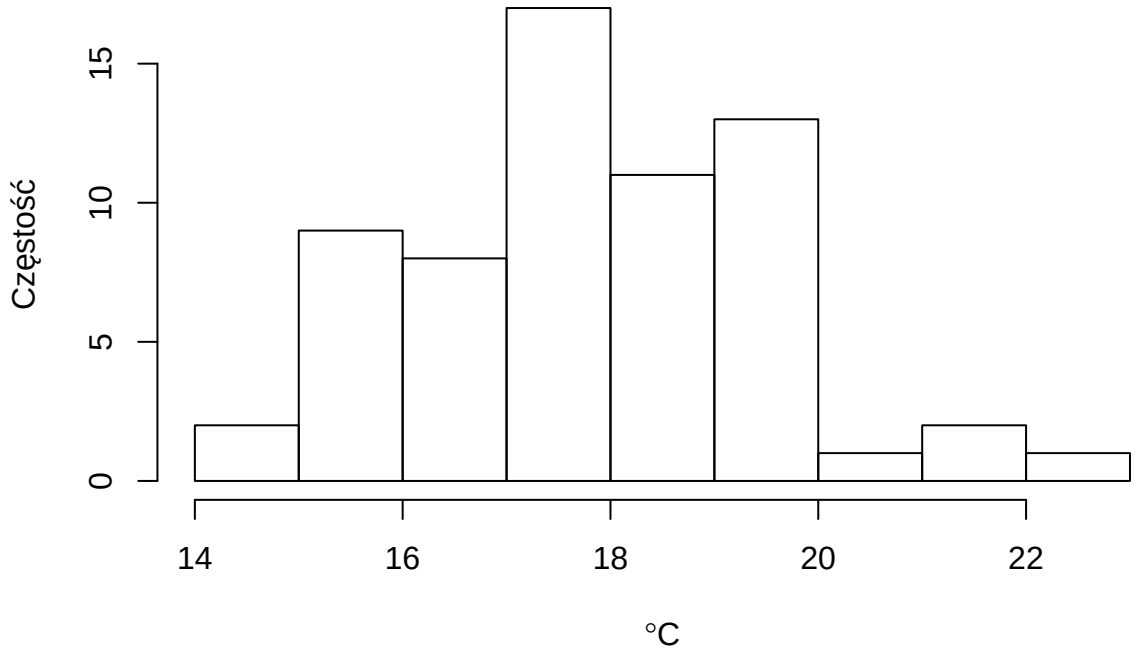
```
boxplot(TEMP~YEAR, ylab=expression(degree*C), xlab="Rok", las=1)
box(lwd=2)
```



Histogramy

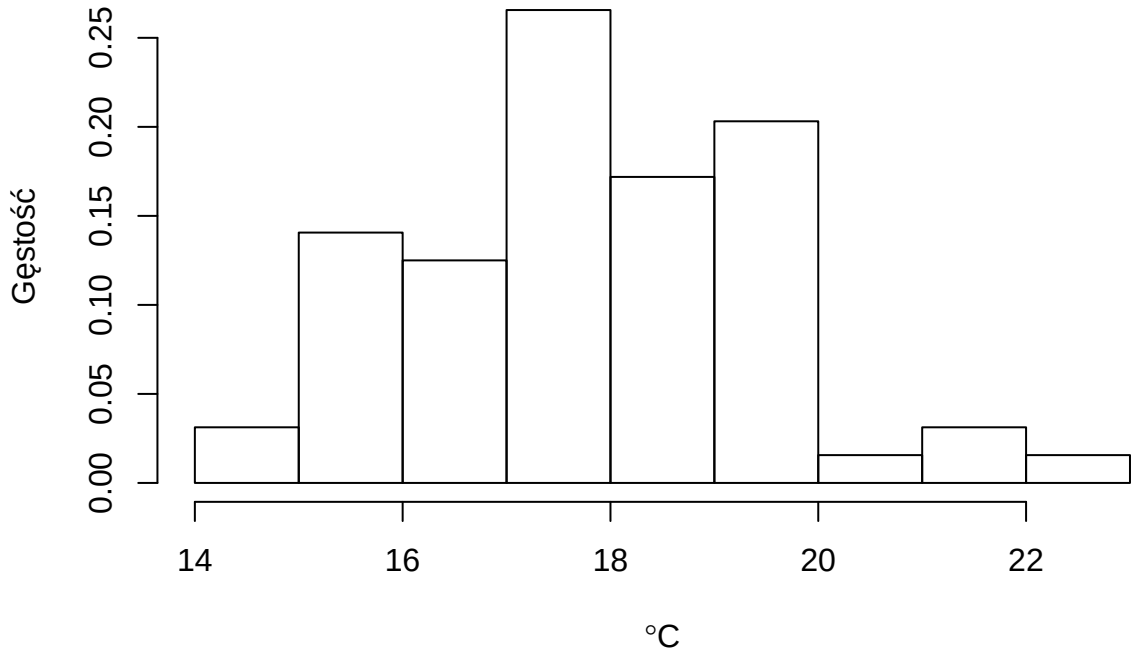
Analiza właściwości rozkładów bardzo często uzupełniana jest o ich graficzną prezentacje. Podstawową funkcją stosowaną w tym wypadku jest *hist*. Na przykład rozkład temperatury dla lipca.

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C), ylab="Częstość", main="")
```



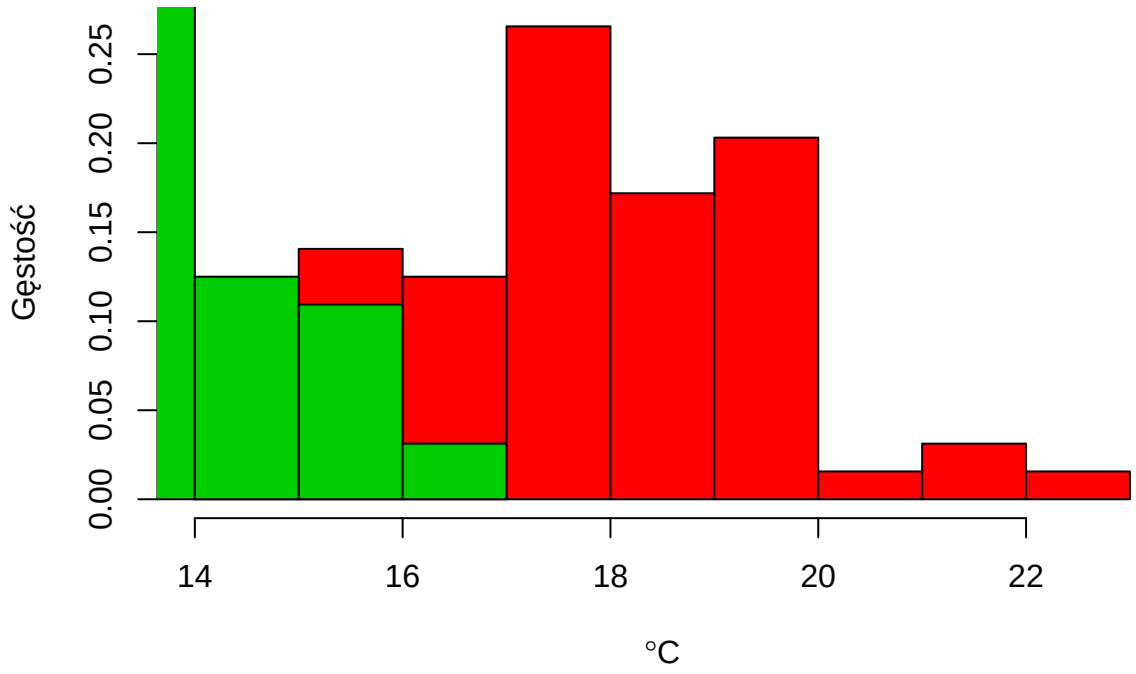
Wykres można dostosować poprzez dodanie kolorów poprzez argument *col*. Bardziej zaawansowane opcje kolorystyczne dostepne są poprzez wykorzystanie funkcji *rgb*, za pomocą której możemy określić w zasadzie dowolny kolor jak również przezroczystość, co jest pomocne w wyświetlaniu (w celach porównawczych) kilku histogramów na jednym wykresie. Nalezy jedynie pamiętać, aby w tym wypadku dodac kolejny argument *probs=T*, który spowoduje, że na osi y wyświetlana będzie gęstość prawdopodobieństwa a nie częstość.

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C), ylab="Gęstość",
     main="", prob=T)
```



dodatnie kolejnego histogramu nastepuje poprzez wpisanie argumentu $add{=}T$

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C),
     ylab="Gęstość", main="", prob=T, col=2)
hist(TEMP[which(MC==9)], xlab=expression(degree*C),
     ylab="", main="", prob=T, col=3, add=T)
```



Niestety zakres zmienności temperatur we wrześniu znacznie odbiega od tego dla lipca i dlatego w przypadku wrzesnia widzimy jedynie fragment histogramu. Faktyczny zakres zmienności można określić z wykorzystaniem funkcji *range*.

```
range(TEMP[which(MC==7)])
```

```
## [1] 14.6 22.2
```

```
range(TEMP[which(MC==9)])
```

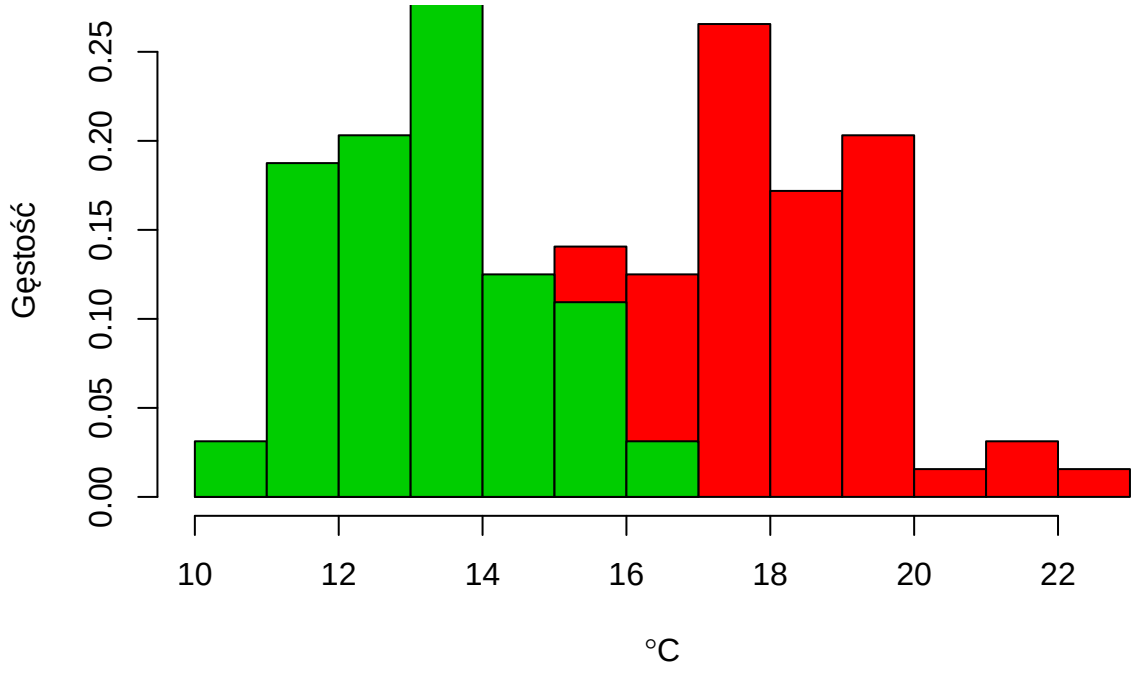
```
## [1] 10.3 16.5
```

Wyninki wskazują, że zakres zmienności osi x powinien obejmować przedział od 10 do 23. Ustalmy zatem wartości parametru *breaks*, gdzie podamy przedziały dla których będzie kreslony histogram

```
myBreaks=seq(10, 23, 1)
```

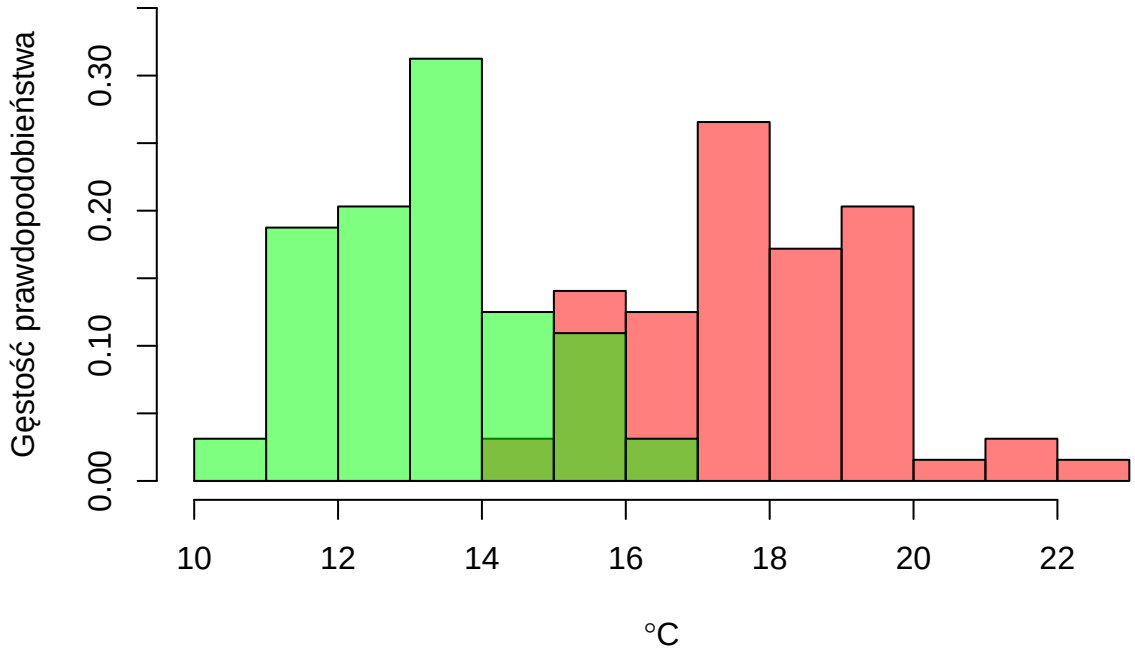
i ponownie *hist* tym razem z argumentem *breaks*

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C), ylab="Gęstość",
     main="", prob=T, col=2, breaks=myBreaks)
hist(TEMP[which(MC==9)], xlab=expression(degree*C), ylab="",
     main="", prob=T, col=3, add=T)
```



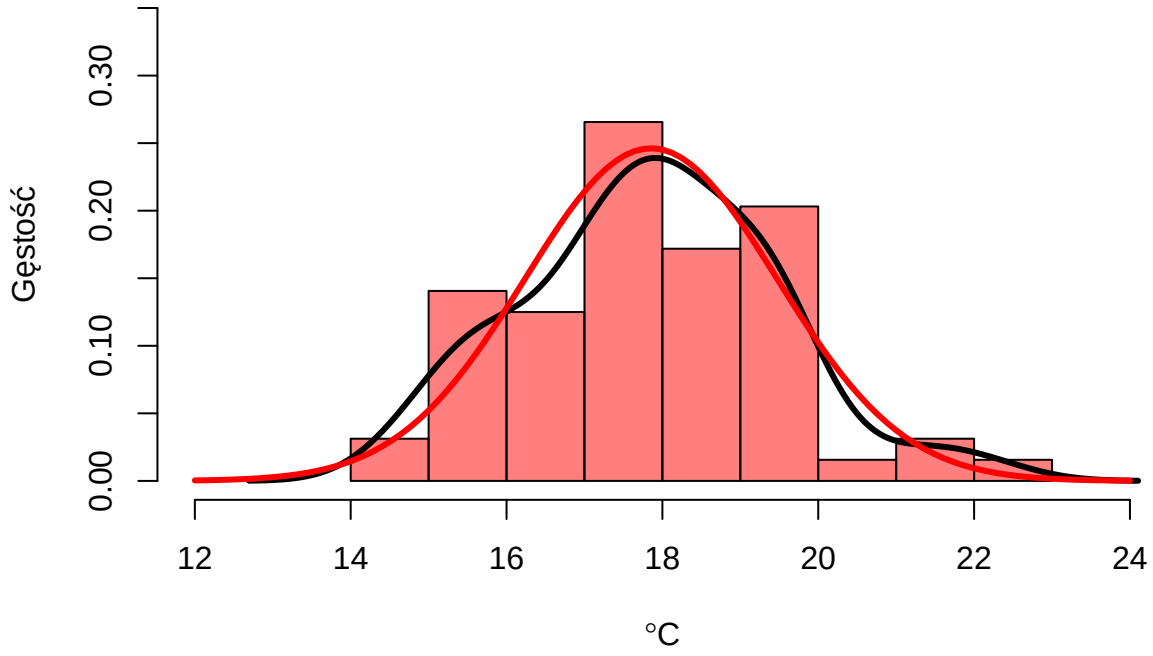
Wygląda lepiej, ale wciąż brak przezroczystości. Dodajmy ją. i jednoczesnie zwiekszmy zakres na osi y.

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C),
     ylab="Gęstość prawdopodobieństwa",
     main="", prob=T, col=rgb(1,0,0,0.5),
     breaks=myBreaks, ylim=c(0, 0.35))
hist(TEMP[which(MC==9)], xlab=expression(degree*C), ylab="",
     main="", prob=T, col=rgb(0,1,0,0.5), add=T)
```



Histogramy można uzupełnić o dopasowanie gęstości prawdopodobieństwa *density* lub wykres wybranego rozkladu teoretycznego np. rozkładu Gauss'a funkcją *curve*. Funkcja *density* posiada dodatkowe argumenty, ale tutaj zastosujemy ja w podstawowej postaci.

```
hist(TEMP[which(MC==7)], xlab=expression(degree*C),
     ylab="Gęstość",
     main="", prob=T, col=rgb(1,0,0,0.5),
     breaks=, ylim=c(0, 0.35), xlim=c(12, 24))
lines(density(TEMP[which(MC==7)]), lwd=3, col=1)


curve(dnorm(x, mean(TEMP[which(MC==7)]), sd(TEMP[which(MC==7)])),
      add=T, col=2, lwd=3)
```



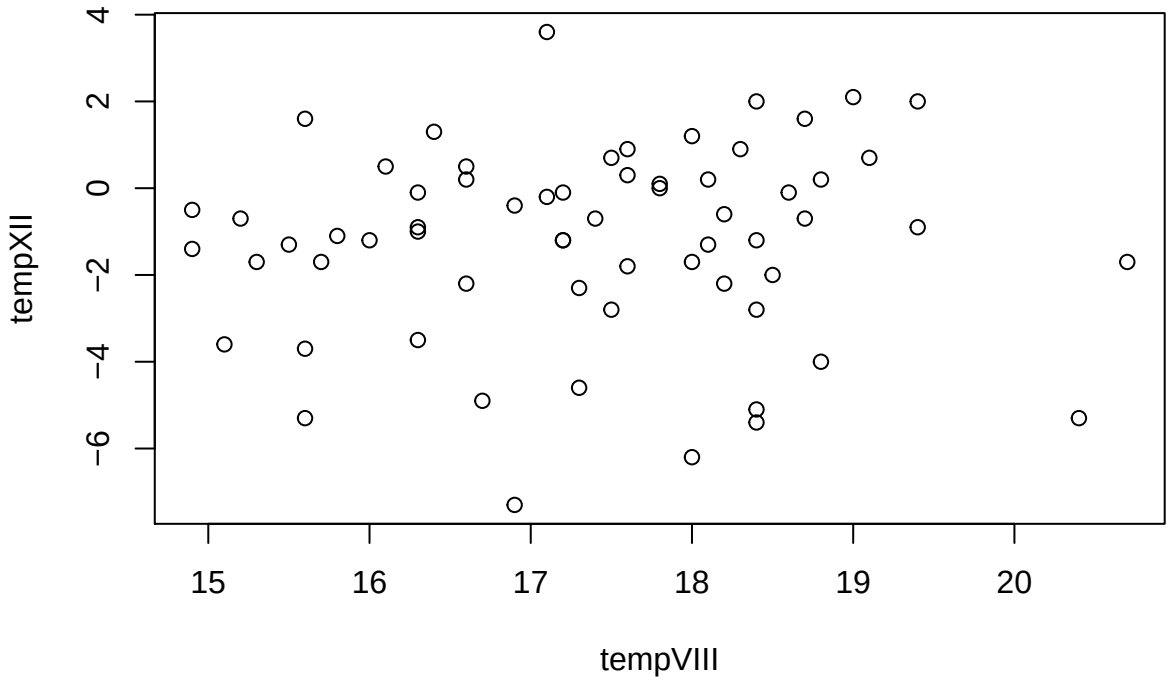
Wykresy rozrzutu

Wstepna analiza zależności między zmiennymi zazwyczaj polega na wykresleniu wykresu rozrzutu w celu wizualnej oceny typu zalezności i późniejszego doboru właściwych metod analizy. Założmy, że chcemy zweryfikowac lansowaną ostatnio w mediach hipotezę, że temepratura powietrza latem ma związek z temperaturą powietrza w miesiącach zimowych. Załóżmy również, że chcemy przeanalizowac sierpień i grudzień. W tym celu utworzymy dwie zmienne indeksując dane i wybierając jedynie te, które nas interesują.

```
tempVIII=TEMP[which(MC==8)]
tempXII=TEMP[which(MC==12)]
```

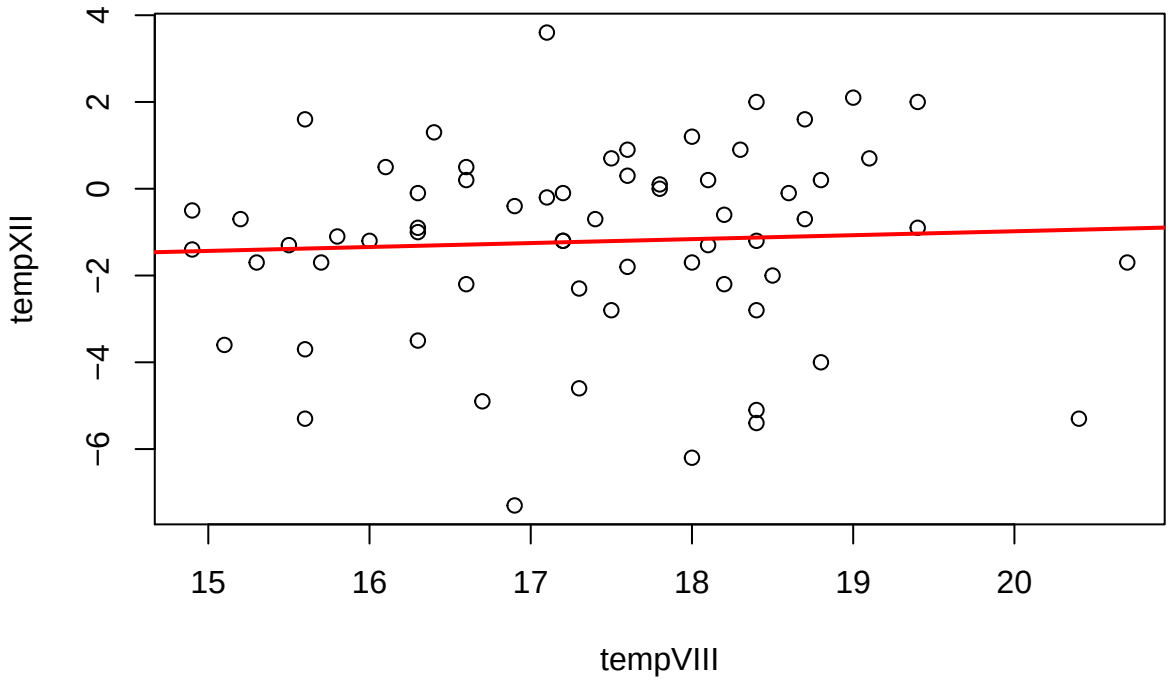
Wykres rozrzutu generujemy za pomocą polecenia *plot*

```
plot(tempVIII, tempXII)
```



Dodajmy linię regresji znanymi już nam poleceniami *abline* oraz *lm*.

```
plot(tempVIII, tempXII)

abline(lm(tempXII~tempVIII), col=2, lwd=2)
```



Korelacja między wartościami temperatury

```r
cor(tempVIII, tempXII)
```

```
## [1] 0.05414815
```

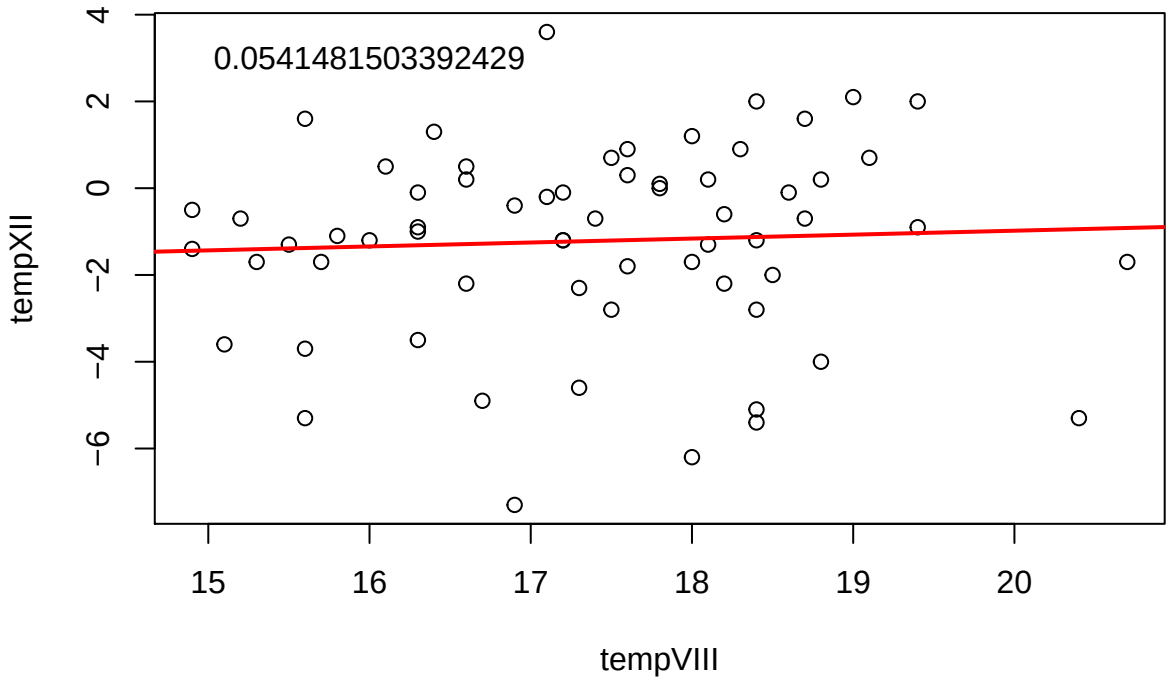
Jej istotność statystyczna

```r
cor.test(tempVIII, tempXII)
```

```
##
##  Pearson's product-moment correlation
##
## data:  tempVIII and tempXII
## t = 0.42699, df = 62, p-value = 0.6709
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1942466  0.2960174
## sample estimates:
##       cor
## 0.05414815
```

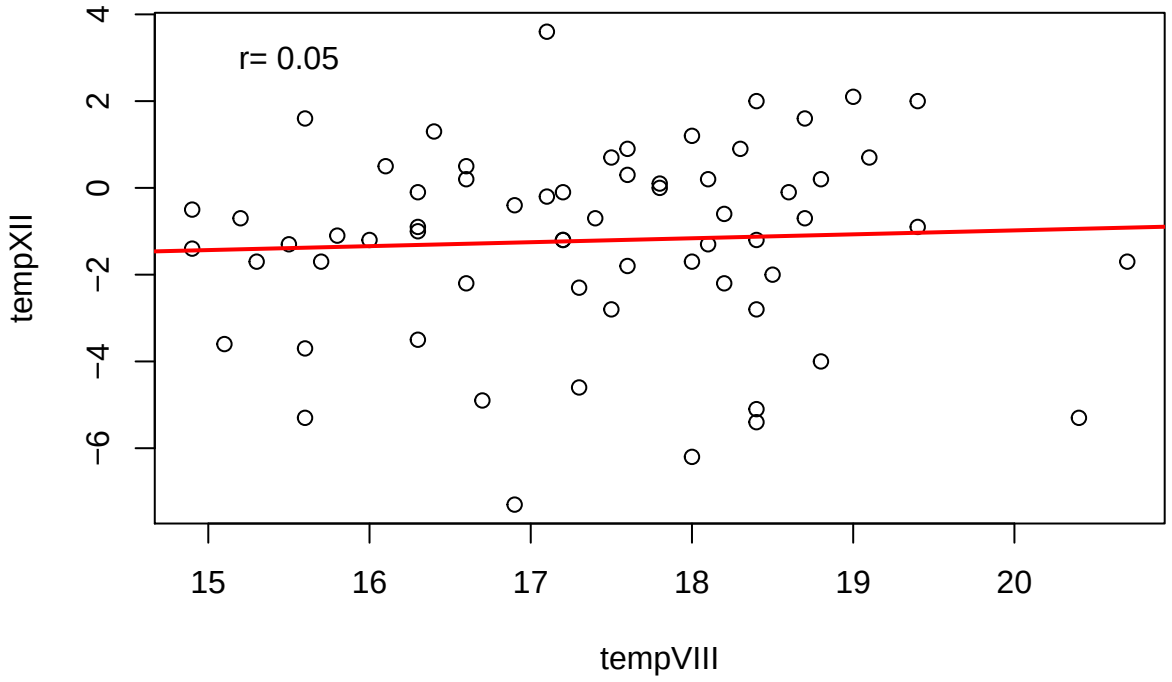
Wartość korelacji możemy wyświetlić bezpośrednio na wykresie korzystając z funkcji *text*

```r
plot(tempVIII, tempXII)
abline(lm(tempXII~tempVIII), col=2, lwd=2)
text(16,3, cor(tempVIII, tempXII))
```



Wynik niepotrzebnie składa sie z tylu cyfr. Można to poprawic stosując polecenie *round*

```
plot(tempVIII, tempXII)
abline(lm(tempXII~tempVIII), col=2, lwd=2)
text(15.5,3, paste("r=", round(cor(tempVIII, tempXII), digits=2)))
```

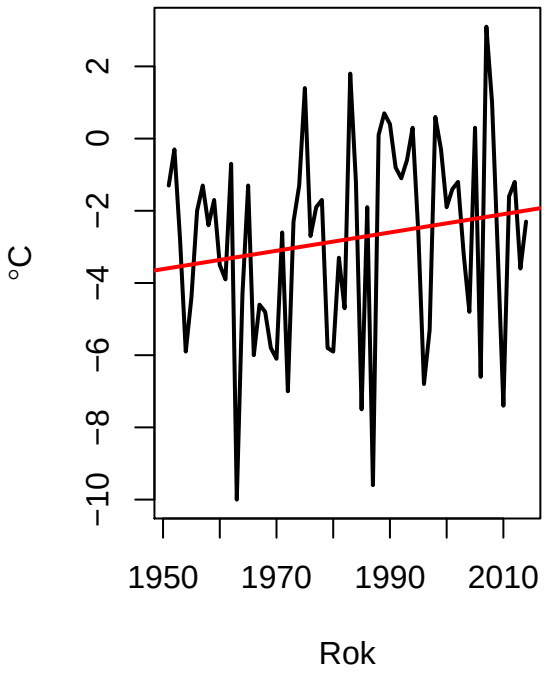


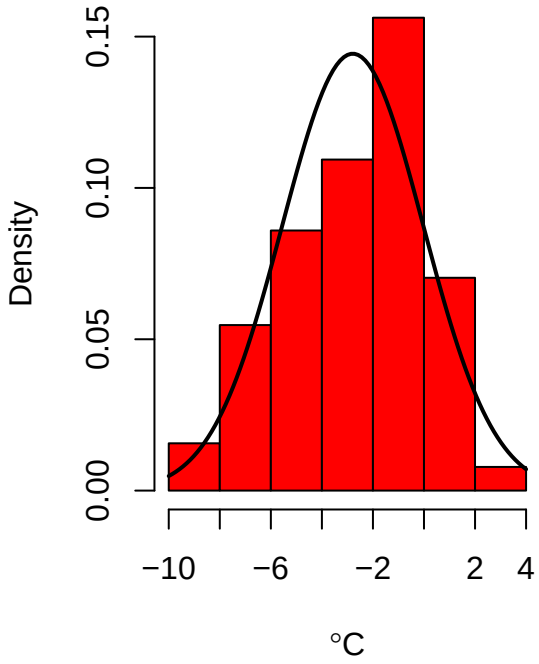
Jak widać, nie istnieje żaden związek między tymi zmiennymi.

Wiele wykresów na jednym

Bardzo często przygotowywanie i formatowanie wielu wykresów jest czynnością, która wymaga czasu a tego bynajmniej niegdy nie ma w zapasie. Również tutaj R ma dla nas miłą niespodziankę. Korzystając z parametru *par* można okreslić ile wykresów i w jakiej konfiguracji ma sie znaleść na wykresie wspólnym. Nalezy pamiętać aby po wygenerowaniu i zapisaniu takiego wykresu wprowadzić komendę *dev.off()*, która zresetuje te ustawienia, gdyż inaczej wszystkie kolejne wykresy będą generowane w ten sposób. Wykreślmy sobie wykres przebiegu temperatury stycznia i histogramu z dopasowaniem rozkładu Gauss'a

```
par(mfrow=c(1,2))
plot(1951:2014,TEMP[which(MC==1)], type="l", lwd=2,
     xlab="Rok", ylab=expression(degree*C),
     main=""
     )
abline(lm(TEMP[which(MC==1)]~c(1951:2014)), col=2, lwd=2)

hist(TEMP[which(MC==1)], col=2, main="", xlab=expression(degree*C), prob=T)
curve(dnorm(x, mean(TEMP[which(MC==1)]), sd(TEMP[which(MC==1)])), add=T, lwd=2)
```





OK, tyle tytułem wstępu.

Życze owocnego wkręcenia się w możliwości analizy i wizualizacji wyników w R.