1.hypotéza

Denisa Mensatorisová a Adam Štuller

Ročná teplota rastie a s ňou rastie aj množstvo zrážok.

Prvá hypotéza sa týka závislosti teploty od množstva zrážok. Ako prvé overíme, či sa postupom času priemerné teploty zvyšujú. Ďalej spočítame priemerné množstvo zrážok pre jednotlivé roky a overíme závislosť priemerných ročných teplôt od ročného množstva zrážok.

```
all_data <- read.csv(file= "../data/all.csv")
```

```
all_data_split_date <- mutate(
    all_data,
    time = format(as_datetime(DATE), format = "%H:%M:%S"),
    date = format(as_date(DATE), format = "%Y-%m-%d"),
    month = month(DATE),
    year = year(DATE),
    md = substr(DATE, start = 6, stop = 10)
  )

data_temperature <- all_data_split_date %>% dplyr::select('DATE', 'TMP', 'time', 'date', 'year', 'month
data_temperature <- data_temperature[!is.na(data_temperature$TMP), ]
head(data_temperature)
```

```
##                  DATE TMP     time       date year month    md
## 1 2004-05-10T00:00:00   8 00:00:00 2004-05-10 2004     5 05-10
## 2 2004-05-10T01:00:00   8 01:00:00 2004-05-10 2004     5 05-10
## 3 2004-05-10T02:00:00   7 02:00:00 2004-05-10 2004     5 05-10
## 4 2004-05-10T03:00:00   6 03:00:00 2004-05-10 2004     5 05-10
## 5 2004-05-10T04:00:00   7 04:00:00 2004-05-10 2004     5 05-10
## 6 2004-05-10T05:00:00   7 05:00:00 2004-05-10 2004     5 05-10
```

Priemerná denná teplota

Graf zobrazuje priemerné denné teploty pre roky 1973 - 2020. Cez graf je vykreslený 95% interval spoľahlivosti lineárneho modelu. Ak by sme merania opakovali na inej vzorke, 95% regresných priamok bude v tomto intervale. Vidíme, že s pribúdajúcim časom sa lineárna priamka mierne zvyšuje, čo svedčí o narastajúcej teplote.

```
df_dayMean_tmp <- data_temperature %>% group_by(date) %>% summarise(tmp = na.omit(mean(TMP)), year = yea
```

```
## `summarise()` has grouped output by 'date'. You can override using the `.groups` argument.
```

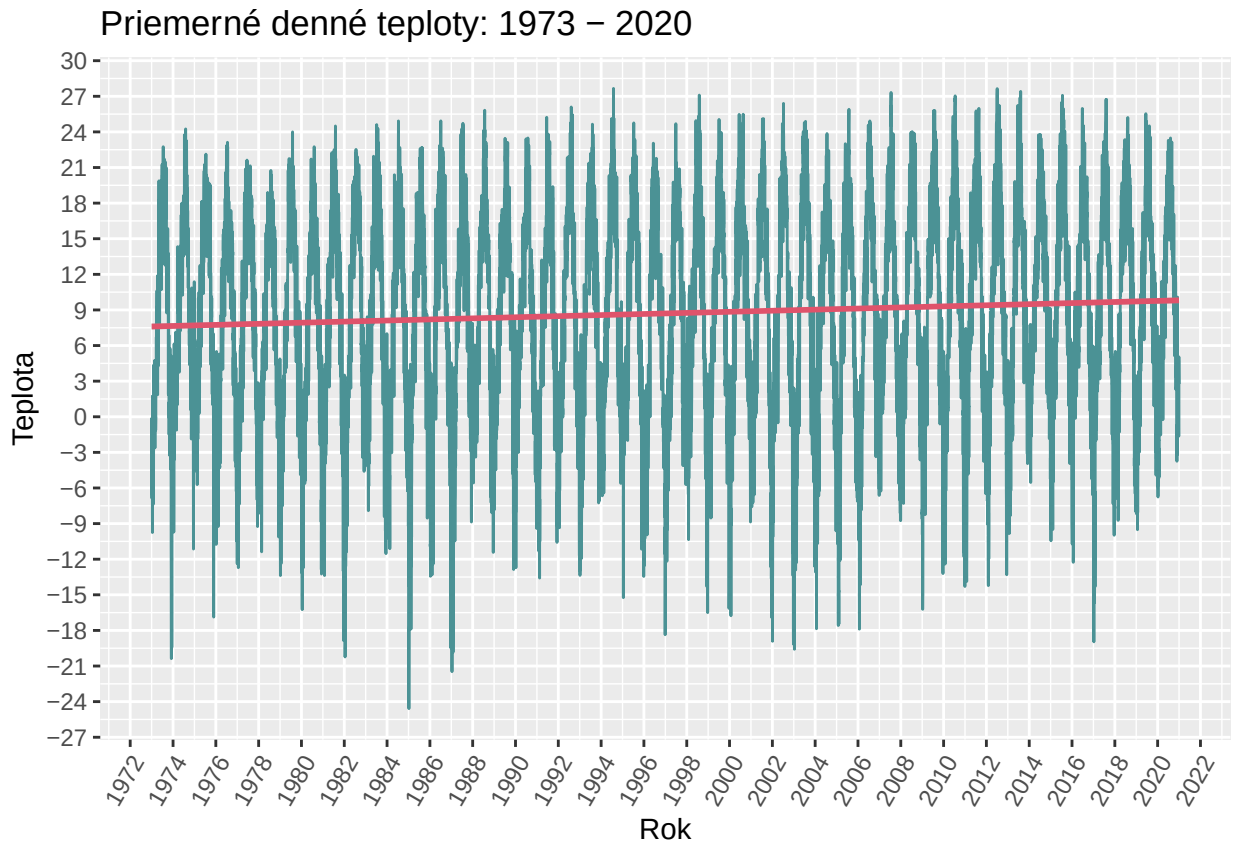
```
df_dayMean_tmp <- unique(df_dayMean_tmp)
head(df_dayMean_tmp)
```

```
## # A tibble: 6 x 4
## # Groups:   date [6]
##   date          tmp  year md
```

```
##   <chr>       <dbl> <int> <chr>
## 1 1973-01-01 -6.57   1973 01-01
## 2 1973-01-02 -5.62   1973 01-02
## 3 1973-01-03 -0.2    1973 01-03
## 4 1973-01-04 -0.125  1973 01-04
## 5 1973-01-05 -2.62   1973 01-05
## 6 1973-01-06 -6.29   1973 01-06
```

```
ggplot(df_dayMean_tmp, aes(x = as.Date(date), y = tmp)) +
  geom_line(color = "#4b9295") +
  geom_smooth(method = "lm", level = 0.95, color = 2, se = T) +
  labs(title = "Priemerné denné teploty: 1973 - 2020", x = "Rok", y = "Teplota") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(-30,30, by = 3))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
table_dayMean_tmp <- data.table(year = df_dayMean_tmp$year, md = df_dayMean_tmp$md, tmp = df_dayMean_tmp

# vytvorenie tabulky
# table_year_md_tmp <- dcast(table_dayMean_tmp, formula = year ~ md, value.var = 'tmp' ) # riadky = rok
table_year_md_tmp <- dcast(table_dayMean_tmp, formula = md ~ year, value.var = 'tmp' ) # riadky = dni,
head(table_year_md_tmp)
```

```
##       md      1973      1974   1975   1976   1977      1978       1979    1980
## 1: 01-01 -6.571429 -1.125000 -2.875 -1.125 -4.250 -3.250000  -1.857143  -2.375
```

```
## 2: 01-02 -5.625000  1.142857  0.750  1.625  0.125 -5.625000  -9.000000  -3.625
## 3: 01-03 -0.200000  1.250000 -3.875 -0.625  0.750 -2.125000  -9.428571 -10.000
## 4: 01-04 -0.125000  1.571429 -1.125 -0.875 -0.500  0.750000  -9.750000  -7.125
## 5: 01-05 -2.625000  1.125000  3.500 -4.875 -2.875 -6.142857 -11.500000 -13.125
## 6: 01-06 -6.285714  0.375000  6.250 -0.875 -3.125 -8.125000 -12.250000  -7.250
##      1981       1982       1983    1984       1985       1986    1987     1988
## 1: -0.750  0.9000000 -2.1625000  2.9125  -6.114286 -6.9875000  4.0750 1.062500
## 2:  0.000  2.1285714 -0.7250000  3.1500  -7.412500 -4.4750000  1.0625 1.987500
## 3:  1.625 -0.5000000 -1.0875000  0.7000 -10.937500 -1.8750000 -2.6375 2.237500
## 4:  3.500  0.8857143 -0.5500000  2.3750 -11.337500  0.2428571 -7.2875 3.050000
## 5: -0.375  3.5142857  3.0571429  0.7125 -14.212500 -6.8625000 -8.1375 4.257143
## 6: -3.625  2.7000000  0.3714286 -3.3875 -12.987500 -7.5750000 -3.4750 4.562500
##         1989    1990       1991      1992     1993       1994   1995     1996
## 1:  1.062500 -2.4875  0.7250000 -9.012500 -10.5750  1.1142857 -0.100  -0.5750
## 2: -1.050000 -1.8750 -0.3714286 -4.525000 -12.9875  2.0714286 -0.750   0.5750
## 3: -4.775000 -3.3500  0.8375000 -3.025000  -9.4000  0.3428571 -2.025  -2.5875
## 4: -6.537500 -8.0500  1.1625000 -8.950000  -9.1375 -0.4000000 -3.275  -3.5875
## 5: -7.437500 -4.5500  0.8875000  1.871429 -13.3625  0.3428571 -2.525  -6.0625
## 6: -4.433333 -8.9250 -2.1875000  2.114286 -10.8750  2.2666667 -3.800 -12.5500
##       1997      1998       1999     2000      2001       2002     2003     2004
## 1: -7.8375 0.3714286 -1.8875000 -12.1875 -4.250000  -8.728571  -6.3000   0.8000
## 2: -5.4750 2.6000000 -0.9625000  -7.7125 -2.733333  -1.950000  -4.2250  -2.6625
## 3: -2.4375 4.7142857  0.8142857  -5.1500  0.025000  -9.514286   2.3875 -10.3625
## 4:  0.4125 5.5600000  1.0285714  -3.4125 -0.475000 -18.912500   0.9000  -7.9000
## 5:  0.7750 1.3571429  0.3875000  -3.6125  1.062500 -14.000000  -1.6375  -9.5875
## 6: -1.8500 1.7166667  0.5000000 -10.4000  2.162500 -12.650000 -11.5750 -13.0750
##          2005       2006      2007      2008       2009       2010      2011
## 1: 0.08064516 -0.8733333 -1.353125 -6.181250  -8.343750  0.2033333 -5.772917
## 2: 1.90625000  1.6586207  1.593750 -3.962500  -7.806250 -0.7689655 -1.414583
## 3: 1.63225806  1.4093750  0.828125 -3.422581  -9.928125 -3.1464286 -8.381250
## 4: 3.46875000  0.4000000  0.959375 -5.951515  -9.218750 -7.6000000 -9.704167
## 5: 4.58125000  0.6218750  2.087500 -5.956250  -5.125000 -4.2312500 -5.385294
## 6: 4.40000000  2.0580645  4.859375 -2.846875 -11.175758 -1.1062500 -3.495833
##          2012      2013     2014       2015       2016       2017       2018
## 1: -3.0125000 -7.447887 2.270833 -4.1305556  -8.865278  -8.6296296  0.9423077
## 2:  0.5333333 -5.683333 2.659155 -2.5388889  -9.302778  -6.8269231  1.7058824
## 3:  0.8020833 -2.117647 3.426389  2.2555556  -9.531944  -1.6884615 -0.2745098
## 4:  1.4416667  4.743056 4.250704  0.1444444 -10.722222   0.3388889  1.1200000
## 5:  0.9083333  3.079167 6.088732 -2.3180556  -6.431944  -3.3196078  1.2941176
## 6:  2.5375000 -0.775000 5.994444 -4.0055556  -3.854167 -10.6734694  3.0800000
##         2019        2020
## 1: -1.773973 -0.22702703
## 2:  1.272603 -4.99305556
## 3: -2.446479 -5.84473684
## 4: -3.106944  1.71643836
## 5: -3.038356  0.04459459
## 6: -1.069444 -5.01388889
```

Priemerná mesačná teplota

Graf zobrazuje priemerné mesačné teploty pre jednotlivé roky 1973 - 2020. Tu je použitá metóda GAM - Generalized Additive Model, ktorá zachytáva aj sezónne a trendové zložky časového radu. Opäť aj tu je vidieť, že teplota sa mierne s časom zvyšuje.

```
# priemerna mesacna teplota
df_monthMean_tmp <- data_temperature %>% group_by(year,month) %>% summarise(tmp = na.omit(mean(TMP)), da
```

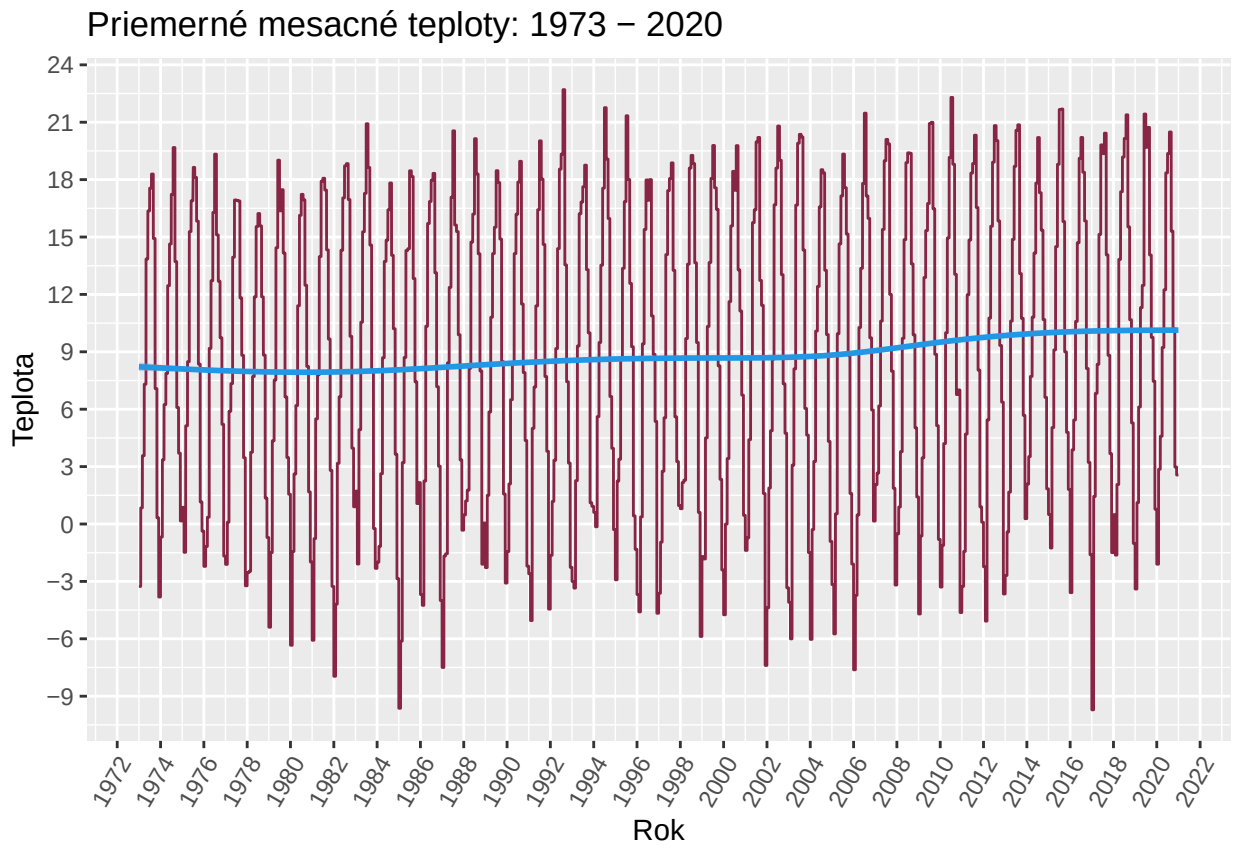
```
## `summarise()` has grouped output by 'year', 'month'. You can override using the `.groups` argument.
```

```
head(df_monthMean_tmp)
```

```
## # A tibble: 6 x 4
## # Groups:   year, month [1]
##    year month   tmp date
##   <int> <int> <dbl> <chr>
## 1  1973     1 -3.26 1973-01-01
## 2  1973     1 -3.26 1973-01-01
## 3  1973     1 -3.26 1973-01-01
## 4  1973     1 -3.26 1973-01-01
## 5  1973     1 -3.26 1973-01-01
## 6  1973     1 -3.26 1973-01-01
```

```
ggplot(df_monthMean_tmp, aes(x = as.Date(date), y = tmp)) +
  geom_line( color="#882545") +
  geom_smooth(color = 4, method = "gam") +
  labs(title = "Priemerné mesačné teploty: 1973 - 2020", x = "Rok", y = "Teplota") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(-30,30, by = 3))
```

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



```r
table_monthMean_tmp <- data.table(year = df_monthMean_tmp$year, month = df_monthMean_tmp$month, tmp = di
table_monthMean_tmp <- unique(table_monthMean_tmp)

# vytvorenie tabulky
table_year_month_tmp <- dcast(table_monthMean_tmp, formula = year ~ month, value.var = 'tmp' ) # riadky
# table_year_month_tmp <- dcast(table_monthMean_tmp, formula = month ~ year, value.var = 'tmp' ) # riad
head(table_year_month_tmp)
```

```
##    year          1          2         3       4        5        6        7
## 1: 1973 -3.2620087  0.84390244 3.5690377 7.306034 13.85294 16.36607 17.55462
## 2: 1974 -0.6818182  3.35585586 6.2426778 7.875000 12.45902 14.63636 17.23333
## 3: 1975  0.8713693 -1.47926267 5.1338912 8.484000 15.28448 16.89474 18.64179
## 4: 1976 -2.2139918 -1.17410714 0.3553719 9.181435 12.71660 16.28571 19.33333
## 5: 1977 -2.1106557  0.09502262 5.8943089 7.340336 13.94737 16.93724 16.91463
## 6: 1978 -2.5413223 -2.46846847 3.7500000 7.707627 11.87190 15.54852 16.22857
##           8        9       10        11         12
## 1: 18.29362 14.92511 7.075630 0.3259912 -3.8170213
## 2: 19.67769 13.71368 6.086957 3.7025862  0.1596639
## 3: 18.11837 15.82979 8.356275 1.1535433 -0.3734940
## 4: 15.11203 12.68908 9.733607 5.2094017 -1.6775510
## 5: 16.88618 11.81780 8.817073 3.4641350 -3.2304527
## 6: 15.58300 11.87500 8.831276 1.3559322 -0.7061224
```

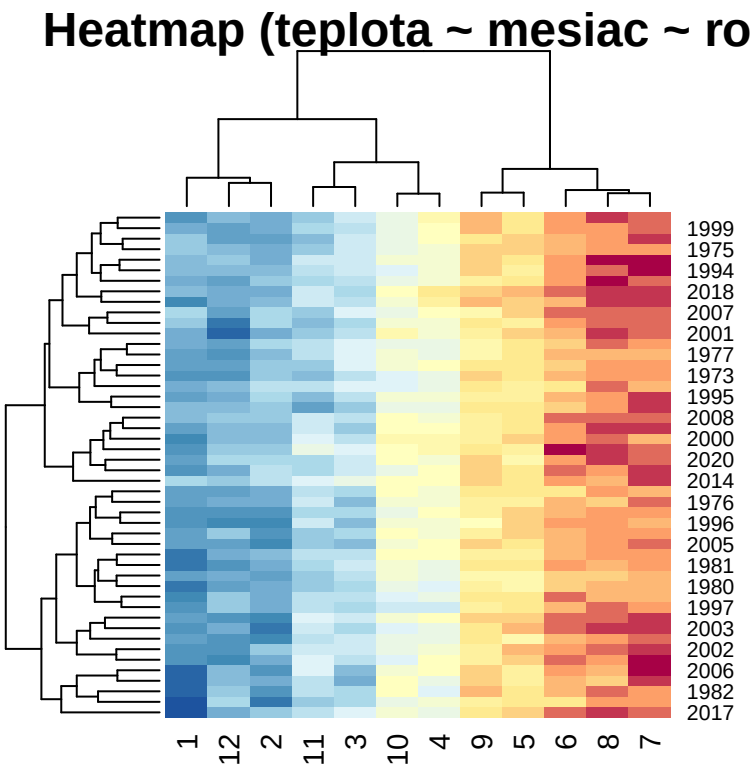
Heatmap

Na základe heatmapy tiež vidíme rozdelenie rokov podľa teploty. Najteplejší bol 7.mesiac a to v rokoch, ktoré majú tmavšiu bordovú farbu. Naopak najchladnejší bol január, v rokoch, ktoré majú tmavšiu modrú farbu.

```r
data_month_matrix <- as.matrix(table_year_month_tmp[,-1])
dim(data_month_matrix)
```

```
## [1] 48 12
```

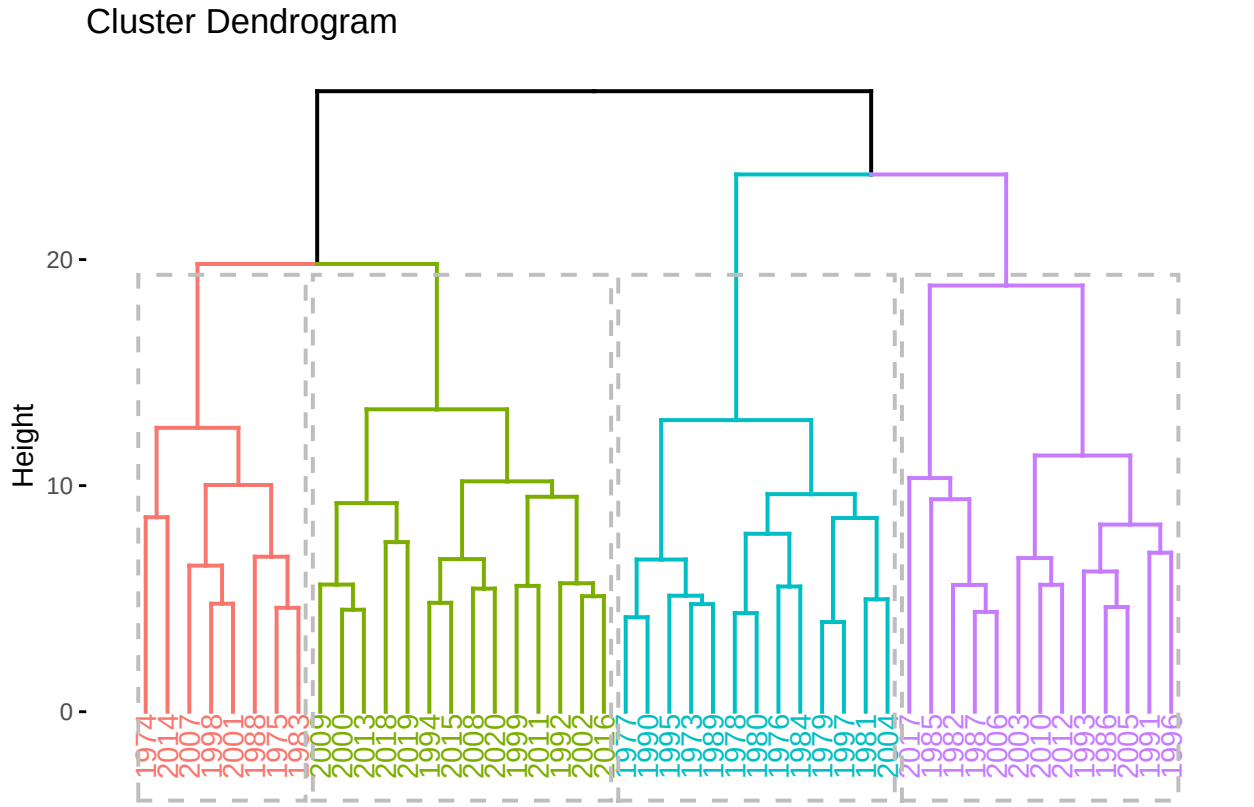
```r
# HEATMAP
heatmap(data_month_matrix,
        labRow = sort(table_year_month_tmp$year),
        scale = 'none',
        main = "Heatmap (teplota ~ mesiac ~ rok)",
        col = colorRampPalette(c("#1d539f", "#408ab5" ,"#74ADD1", "#ABD9E9" ,"#E0F3F8" ,"#FFFFBF", "#fe
```



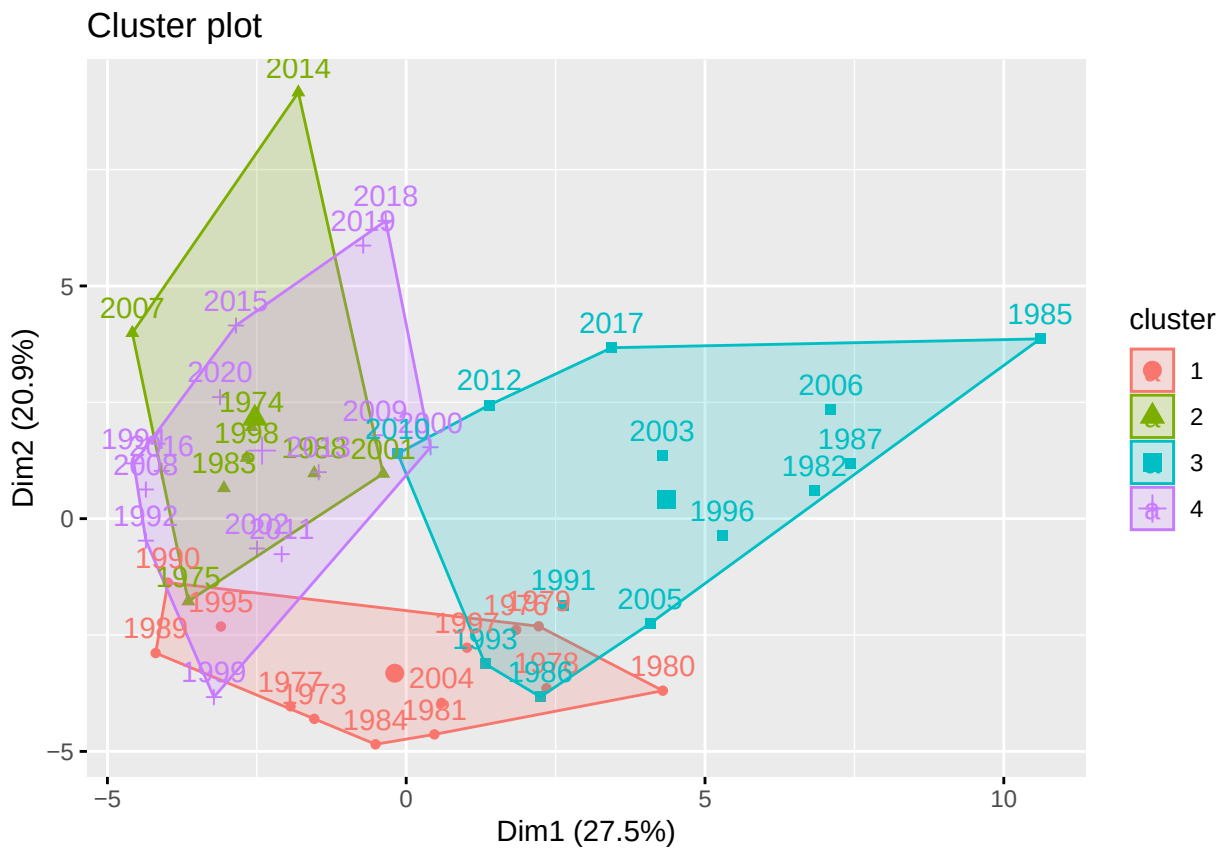
Hierarchické klastrovanie

Výstupom hierarchického zhlukovania je strom nazývaný dendrogram, ktorý zobrazuje sekvencie klastrov.Výška jednotlivých klastrov určuje stupeň podobnosti podľa stupnice na ľavej strane dendrogramu. Určili sme optimálny počet klastrov 4. Opäť podobné roky sú zoskupené do 1 klastra. V zelenom klasti sa nachádza najviac rokov nad 2000, z čoho môžme povedať, že teplota v aktuálnom tisícročí je podobná vo viacerých rokoch. Naopak väčšina 80-tych rokov je v modrom klastri teda tieto sú si teplotou dosť podobné.

```
rownames(data_month_matrix) <- table_year_month_tmp$year
res <- factoextra::hcut(dist(data_month_matrix), k = 4, stand = T)

fviz_dend(res, rect = TRUE)
```




```
fviz_cluster(res)
```



Klastrovanie metódou SOM (Self-Organized Map)

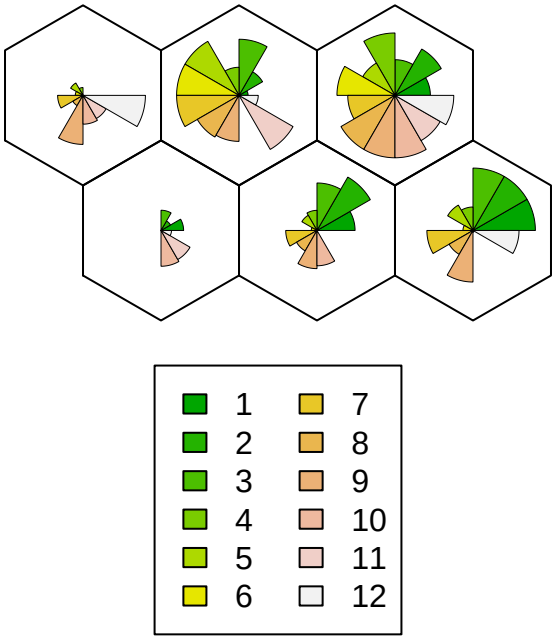
Zvolili sme mapu s rozmermi 3x2 polí, Euklidovskú metódu výpočtu vzdialenosti a tvar šesťuholníka, ktorý má viac susedov. Číslovanie polí je od 1 vľavo dole smerom doprava, najvyššie číslo má pole mapy vpravo hore.

Nastavenie ďalších parametrov súvisí s trénovaním SOM. Ak je parameter radius = 0 tak SOM je veľmi podobný K-Means algoritmu. Parameter radius by mal byť na začiatku cca 2/3 z rozmerov mapy - tu 3x2 = 6, čiže 4, postupne sa znižuje. Parameter rlen znamená koľkokrát sa dáta znovu načítajú a hodnotia v SOM. Tento parameter bol nastavený experimentálne sledovaním priebehu grafu "Changes". Hodnoty v grafe by sa mali znižovať a nakoniec by už mali oscilovať okolo finálnej hodnoty. Vtedy už tento parameter netreba ďalej meniť.

```
# SOM mapa -> roky a mesiace
set.seed(123)
som_grid <- kohonen::somgrid(xdim = 3, ydim = 2, topo = "hexagonal")

set.seed(123)
som_model <- kohonen::som(X = data_month_matrix, grid = som_grid,
              rlen = 200, alpha = c(0.05,0.01), keep.data = T, dist.fcts = "euclidean", radius = 4)

plot(som_model, type="changes")
```


```
graphics::plot(som_model, type = "codes", shape = "straight")
```

Codes plot




```
som_model$unit.classif # zadelenie rokov do tried
```

```
##  [1] 2 3 3 1 1 1 4 1 1 4 3 1 4 1 4 3 2 2 1 6 4 3 2 1 4 2 2 6 2 5 5 4 4 4 3 6 6 5
## [39] 6 5 6 3 6 6 5 6 6 6
```

```
table(som_model$unit.classif) # pocty v triedach
```

```
##
##  1  2  3  4  5  6
##  9  7  7  9  5 11
```

```
df <- data.frame(year = c(1973:2020), class = som_model$unit.classif)
df[order(df$class),]
```

```
##    year class
## 4  1976     1
## 5  1977     1
## 6  1978     1
## 8  1980     1
## 9  1981     1
## 12 1984     1
## 14 1986     1
## 19 1991     1
## 24 1996     1
## 1  1973     2
## 17 1989     2
## 18 1990     2
## 23 1995     2
```

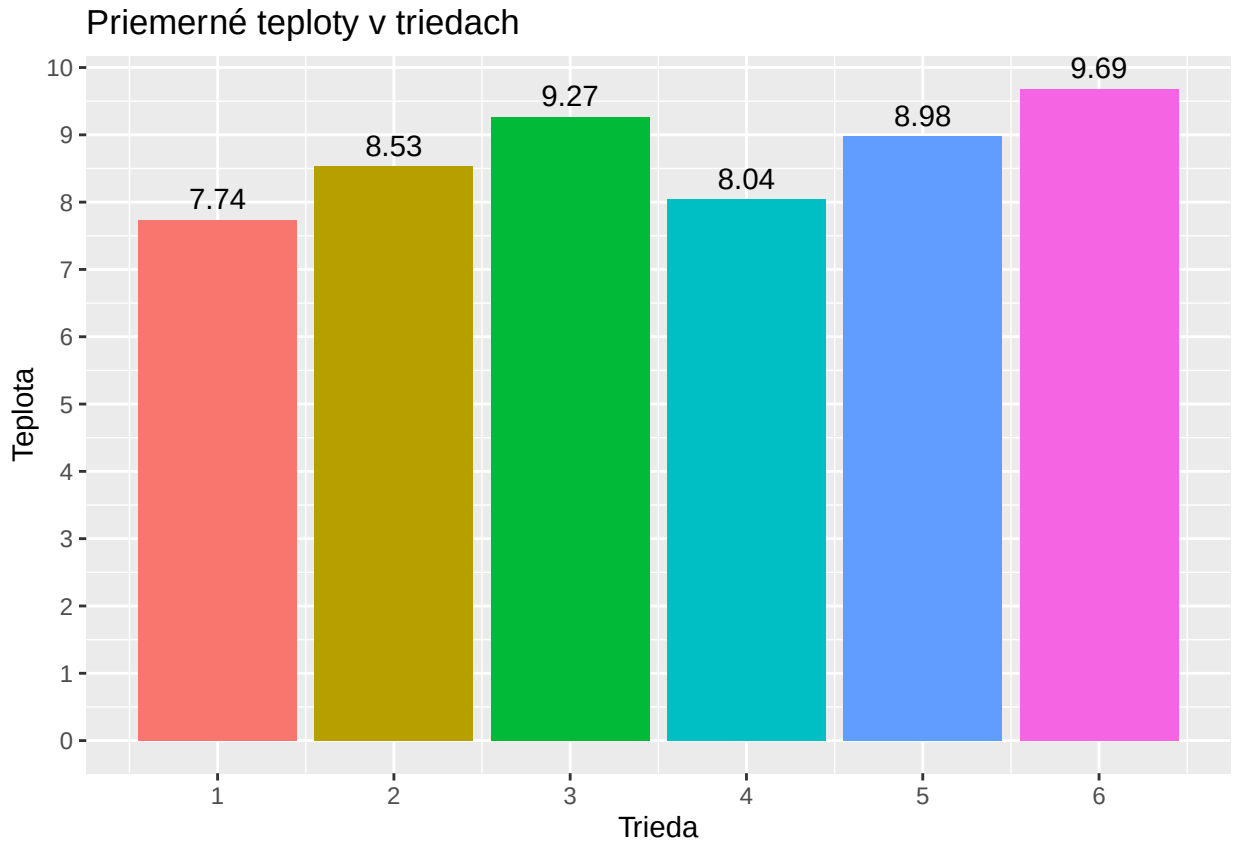
```
## 26 1998     2
## 27 1999     2
## 29 2001     2
## 2  1974     3
## 3  1975     3
## 11 1983     3
## 16 1988     3
## 22 1994     3
## 35 2007     3
## 42 2014     3
## 7  1979     4
## 10 1982     4
## 13 1985     4
## 15 1987     4
## 21 1993     4
## 25 1997     4
## 32 2004     4
## 33 2005     4
## 34 2006     4
## 30 2002     5
## 31 2003     5
## 38 2010     5
## 40 2012     5
## 45 2017     5
## 20 1992     6
## 28 2000     6
## 36 2008     6
## 37 2009     6
## 39 2011     6
## 41 2013     6
## 43 2015     6
## 44 2016     6
## 46 2018     6
## 47 2019     6
## 48 2020     6
```

```r
df_1 <- merge(x = df %>% filter(class == 1), y = df_monthMean_tmp, by = "year", all = F)
df_1 <- unique(df_1) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
mean_tmp_1 <- mean(df_1$mean_tmp)

df_2 <- merge(x = df %>% filter(class == 2), y = df_monthMean_tmp, by = "year", all = F)
df_2 <- unique(df_2) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
mean_tmp_2 <- mean(df_2$mean_tmp)

df_3 <- merge(x = df %>% filter(class == 3), y = df_monthMean_tmp, by = "year", all = F)
df_3 <- unique(df_3) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
mean_tmp_3 <- mean(df_3$mean_tmp)

df_4 <- merge(x = df %>% filter(class == 4), y = df_monthMean_tmp, by = "year", all = F)
df_4 <- unique(df_4) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
mean_tmp_4 <- mean(df_4$mean_tmp)

df_5 <- merge(x = df %>% filter(class == 5), y = df_monthMean_tmp, by = "year", all = F)
df_5 <- unique(df_5) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
```

```
mean_tmp_5 <- mean(df_5$mean_tmp)

df_6 <- merge(x = df %>% filter(class == 6), y = df_monthMean_tmp, by = "year", all = F)
df_6 <- unique(df_6) %>% group_by(year) %>% summarise(mean_tmp = mean(tmp))
mean_tmp_6 <- mean(df_6$mean_tmp)

tmp_class <- data.frame(class = c(1:6), mean_tmp = c(mean_tmp_1,mean_tmp_2,mean_tmp_3,mean_tmp_4,mean_tm
tmp_class
```

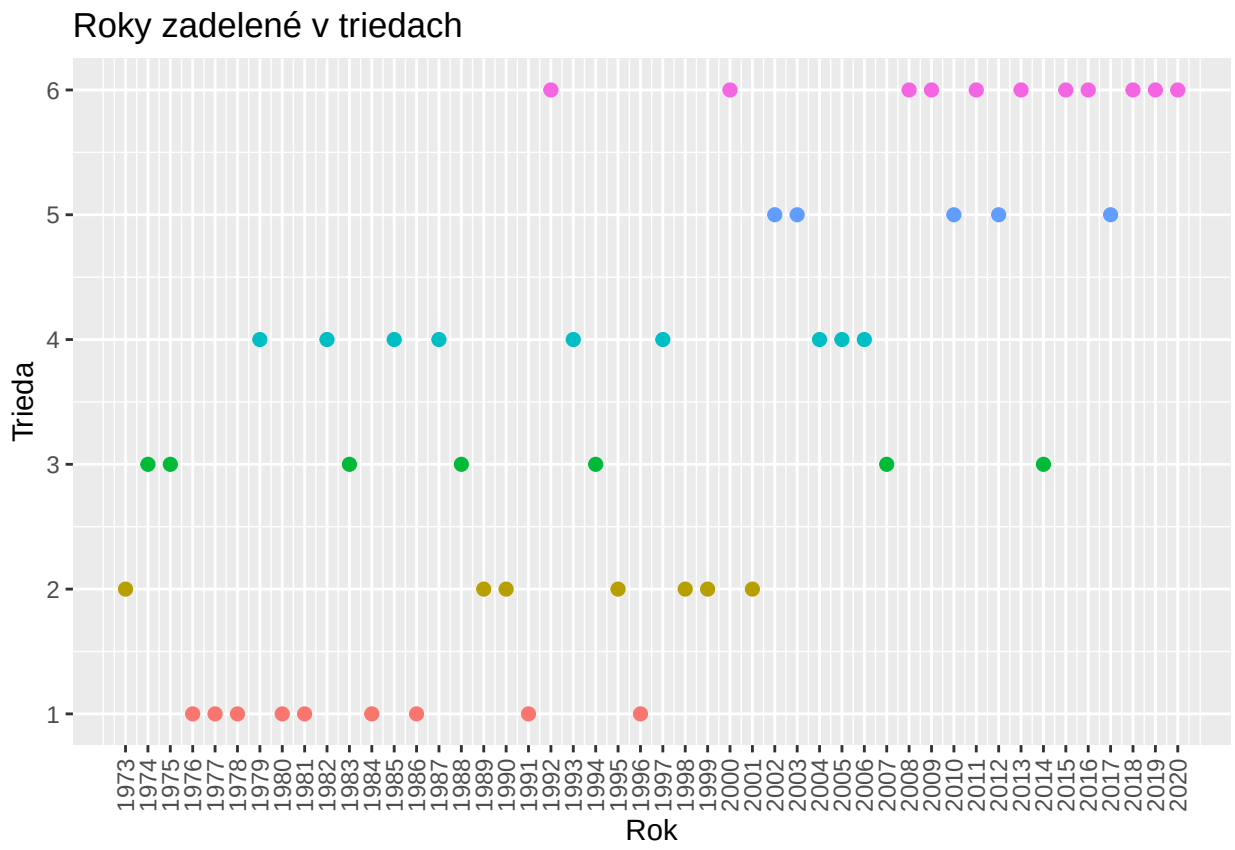
```
##   class mean_tmp
## 1     1 7.737836
## 2     2 8.533500
## 3     3 9.266730
## 4     4 8.041759
## 5     5 8.975162
## 6     6 9.686450
```

```
ggplot(tmp_class, aes(x = class, y = mean_tmp, fill = as.factor(class))) +
  geom_bar(stat = "identity") +
  labs(title = "Priemerné teploty v triedach", x = "Trieda", y = "Teplota") +
  theme(legend.position="none") +
  geom_text(mapping = aes(label = round(mean_tmp,2)), vjust = -0.5, size = 4) +
  scale_y_continuous(breaks = seq(0, 12, by = 1)) +
  scale_x_continuous(breaks = seq(1,6, by = 1))
```




```
ggplot(df, aes(x = year, y = class)) +
  geom_point(aes(color = as.factor(class)), size = 2) +
```

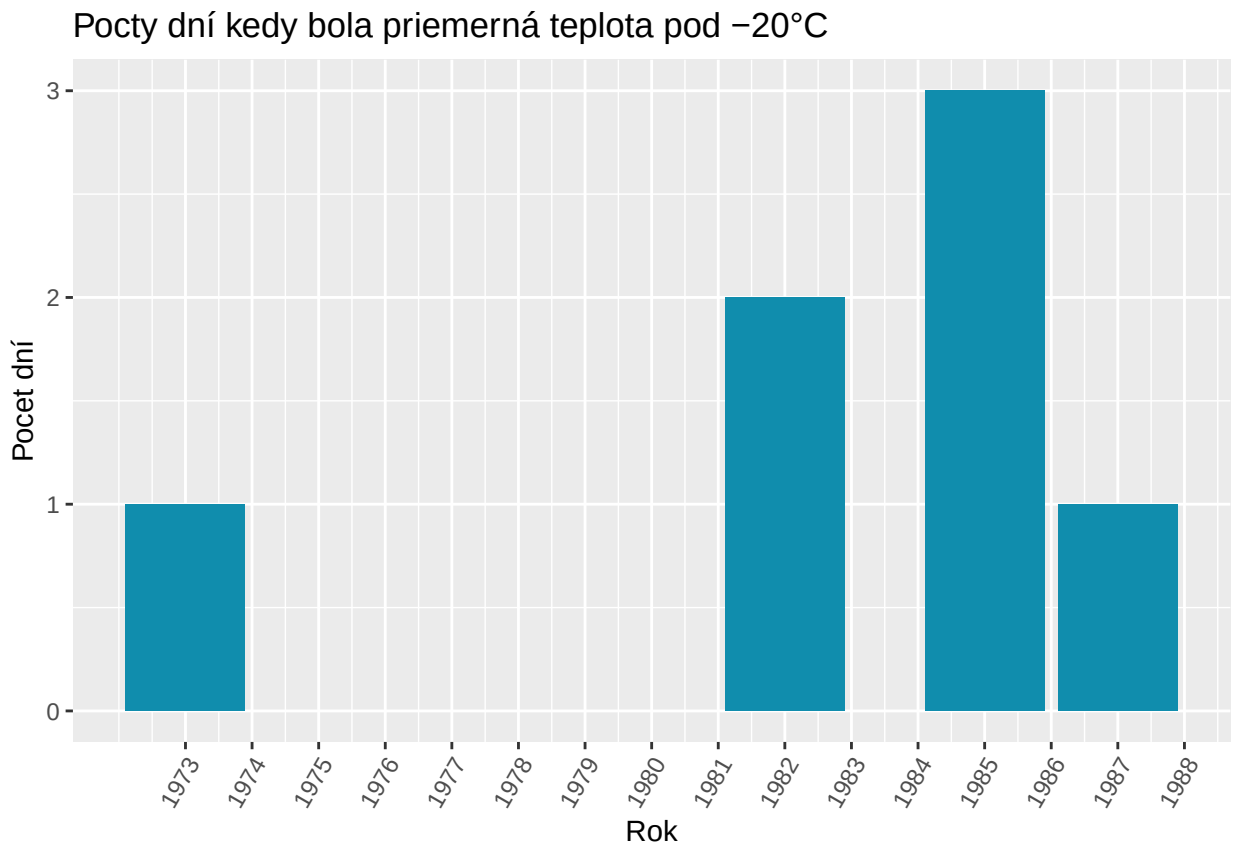
```
  scale_y_continuous(breaks = seq(1,6, by = 1)) +
  labs(title = "Roky zadelené v triedach", x = "Rok", y = "Trieda") +
  scale_x_continuous(breaks = seq(1973, 2020, by = 1)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  theme(legend.position="none")
```



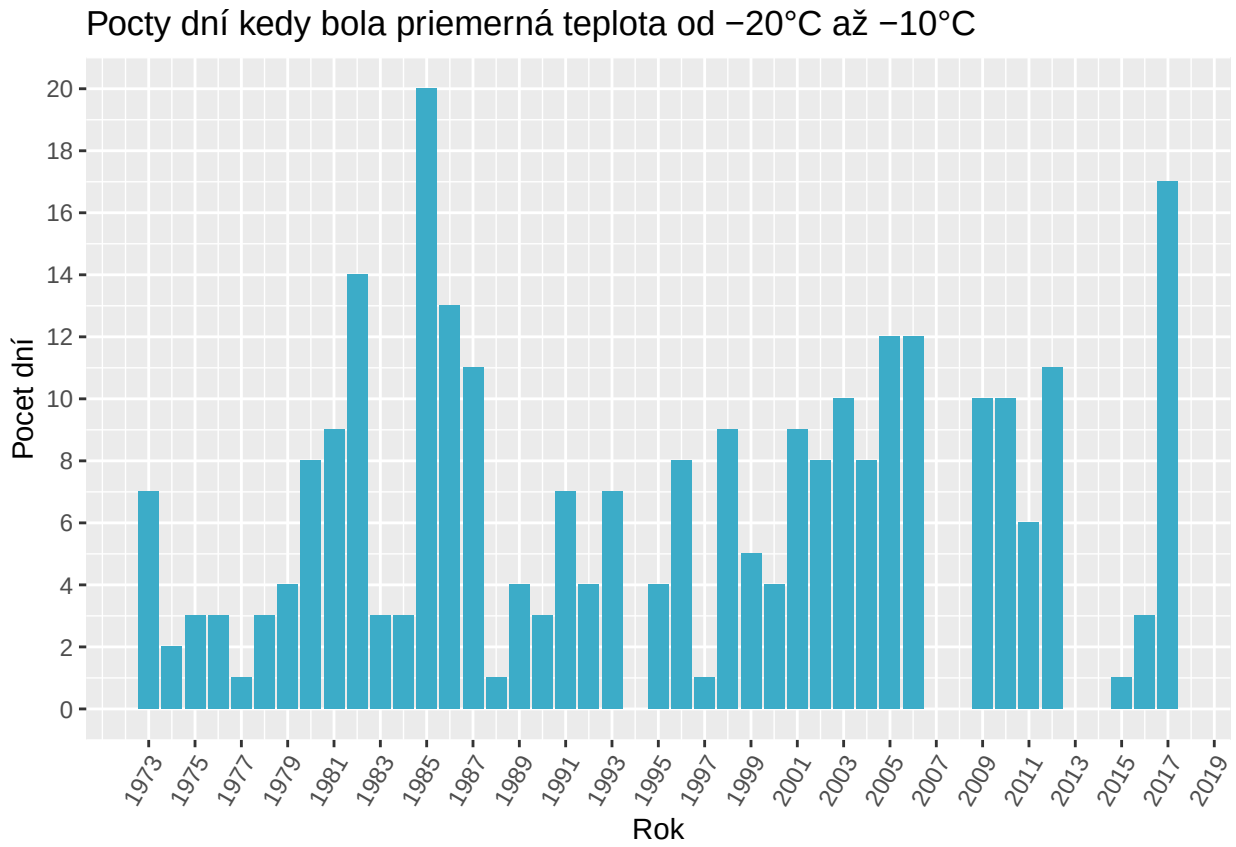
Na základe klastrovania pomocou SOM mapy sme zadelili jednotlivé roky do 6 tried podľa ich priemerných mesačných teplôt. Ako vidieť z grafov 6. trieda obsahuje väčšinu súčasných rokov nad 2010 a priemerná teplota v tejto triede je najvyššia, čo svedčí o tom, že v posledných rokoch sa teplota zvyšuje. Naopak tried č.1 obsahuje väčšinu 70-tych rokov a priemerná teplota rokov patriacich do tejto triedy je najnižšia.

```
add_group <- function(val) {

  if(val <= -20){
    return(as.numeric(1))
  }
  if(val <= -10){
    return(as.numeric(2))
  }
  if(val <= 0){
    return(as.numeric(3))
  }
  if(val <= 10){
    return(as.numeric(4))
  }
  if(val <= 20){
```

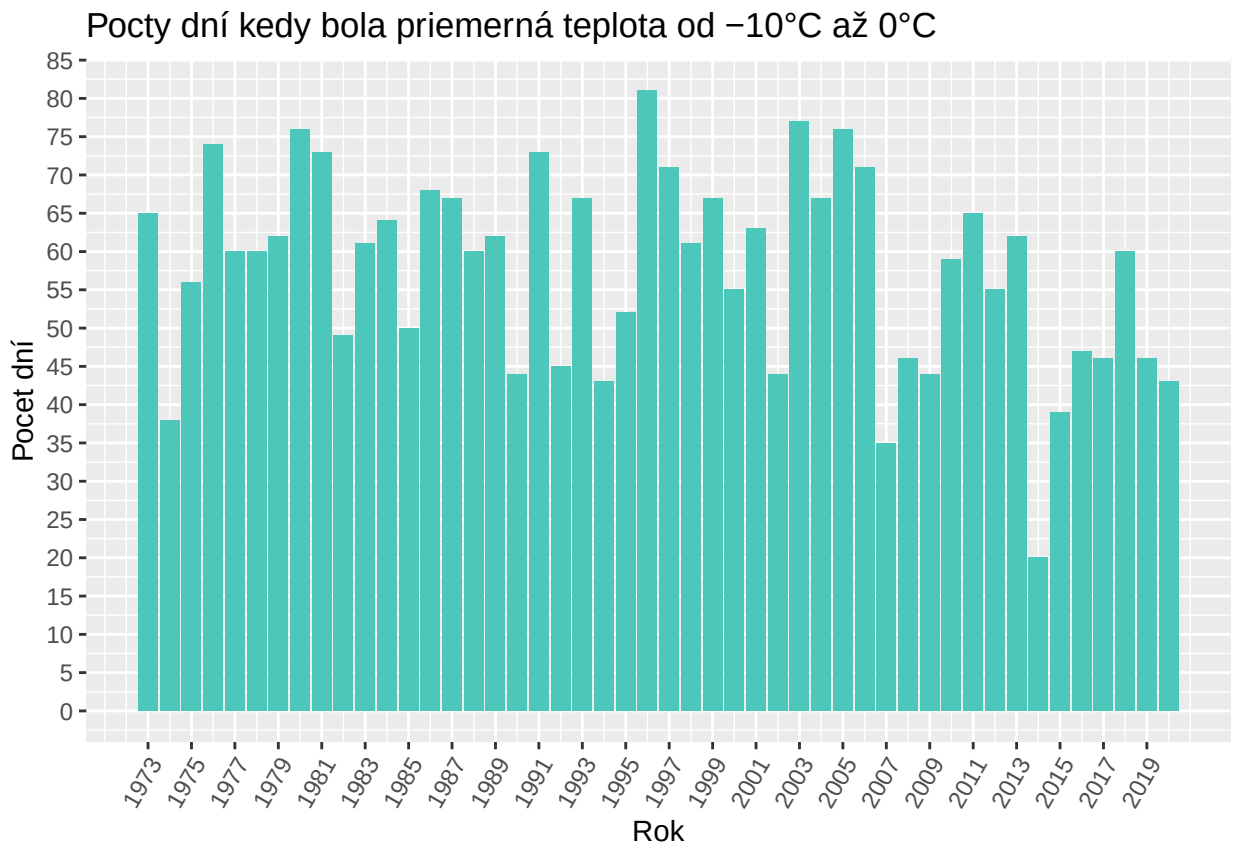
```
    return(as.numeric(5))
  }
  if(val > 20){
    return(as.numeric(6))
  }

}

df_dayMean_tmp <- df_dayMean_tmp %>% dplyr::mutate(
    group = map_dbl(tmp, add_group)
)

df_tmp_groups <- df_dayMean_tmp %>% group_by(year, group) %>% count()
df_tmp_groups$group <- factor(df_tmp_groups$group)

table_groups <- as.data.table(df_tmp_groups)


# vytvorenie tabulky - pocty dni
table_groups_tmp <- dcast(table_groups, formula = year ~ group, value.var = 'n' , fill = 0)


ggplot(data = df_tmp_groups %>% filter(group == 1), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#108dad") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,10, by = 1)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 1)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota pod -20°C"), y = "Počet dní", x = "Rok")
```



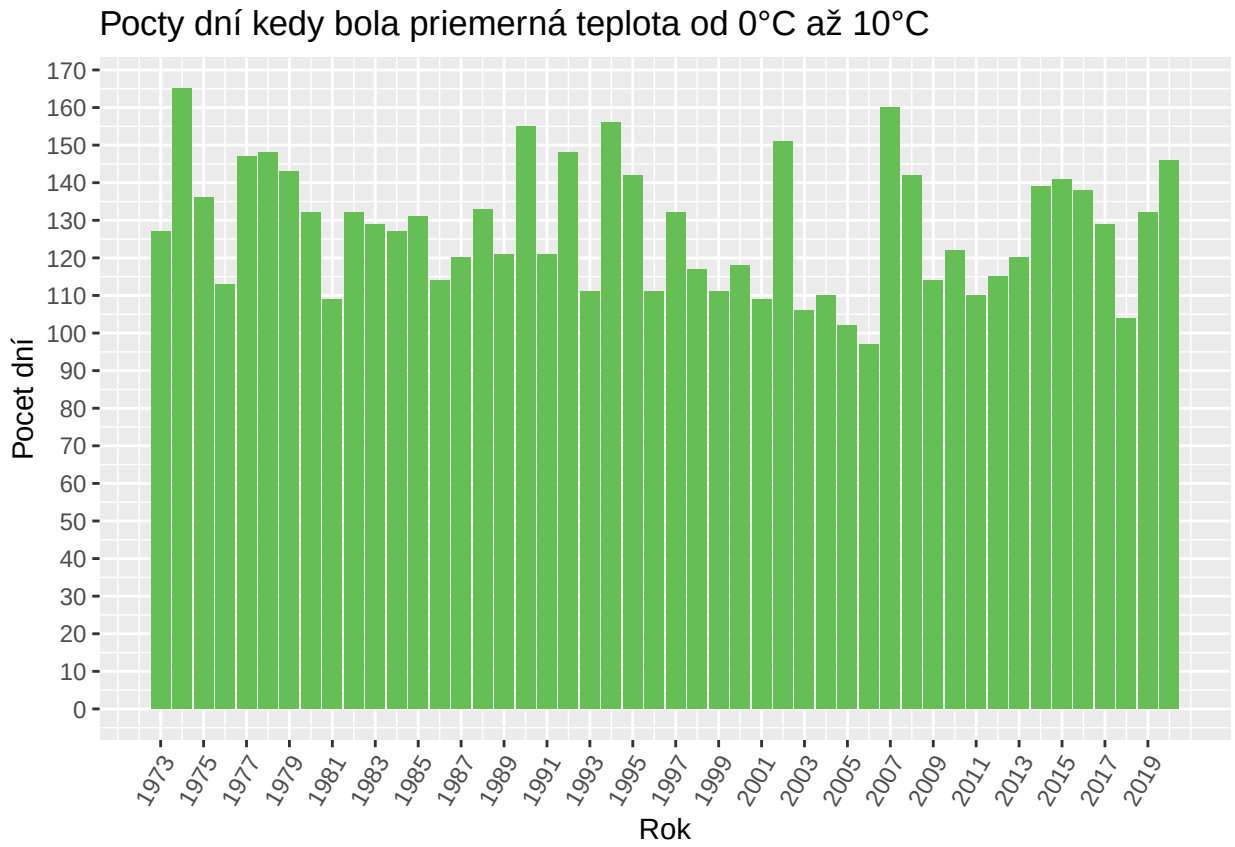

```
ggplot(data = df_tmp_groups %>% filter(group == 2), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#3cacc8") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,30, by = 2)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 2)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota od -20°C až -10°C"), y = "Počet dní", x = "
```



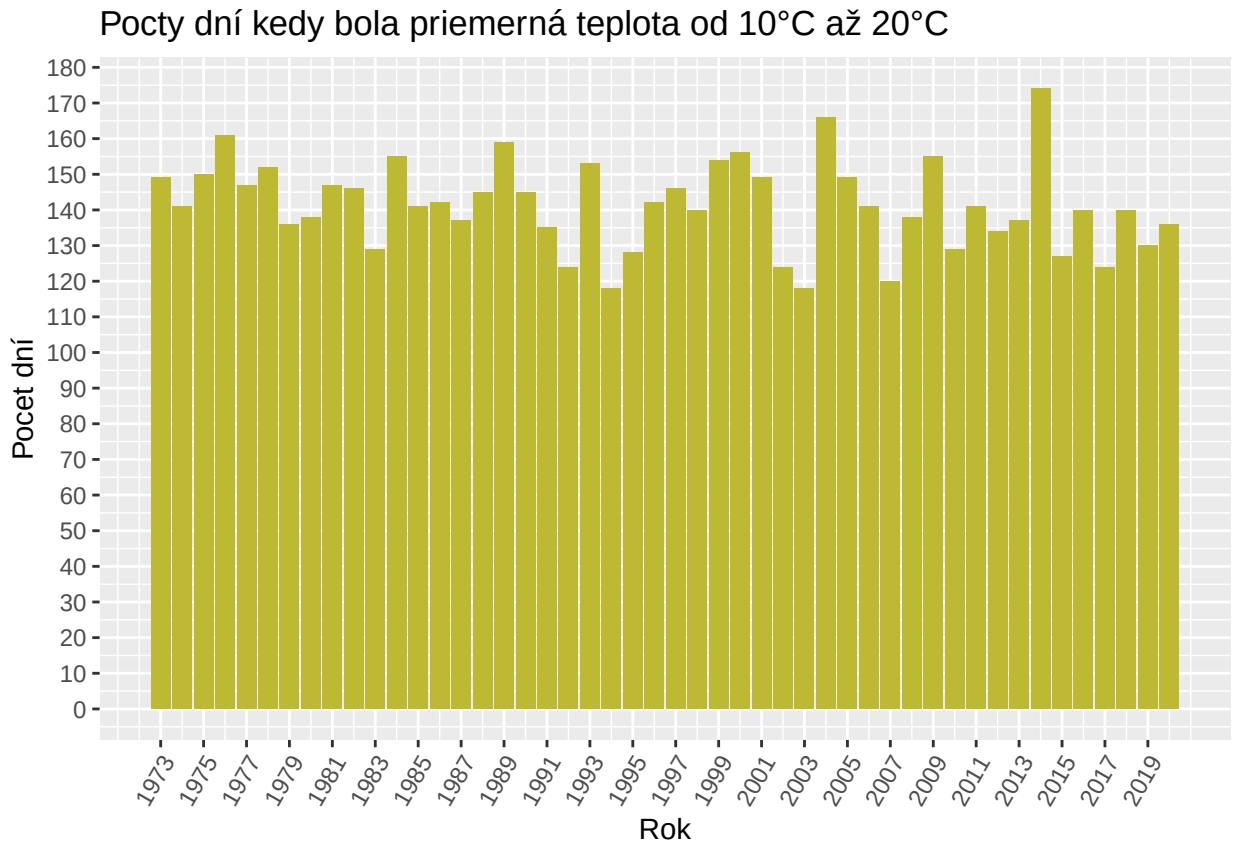

```
ggplot(data = df_tmp_groups %>% filter(group == 3), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#4dc7ba") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,100, by = 5)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 2)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota od -10°C až 0°C"), y = "Počet dní", x = "Rok
```



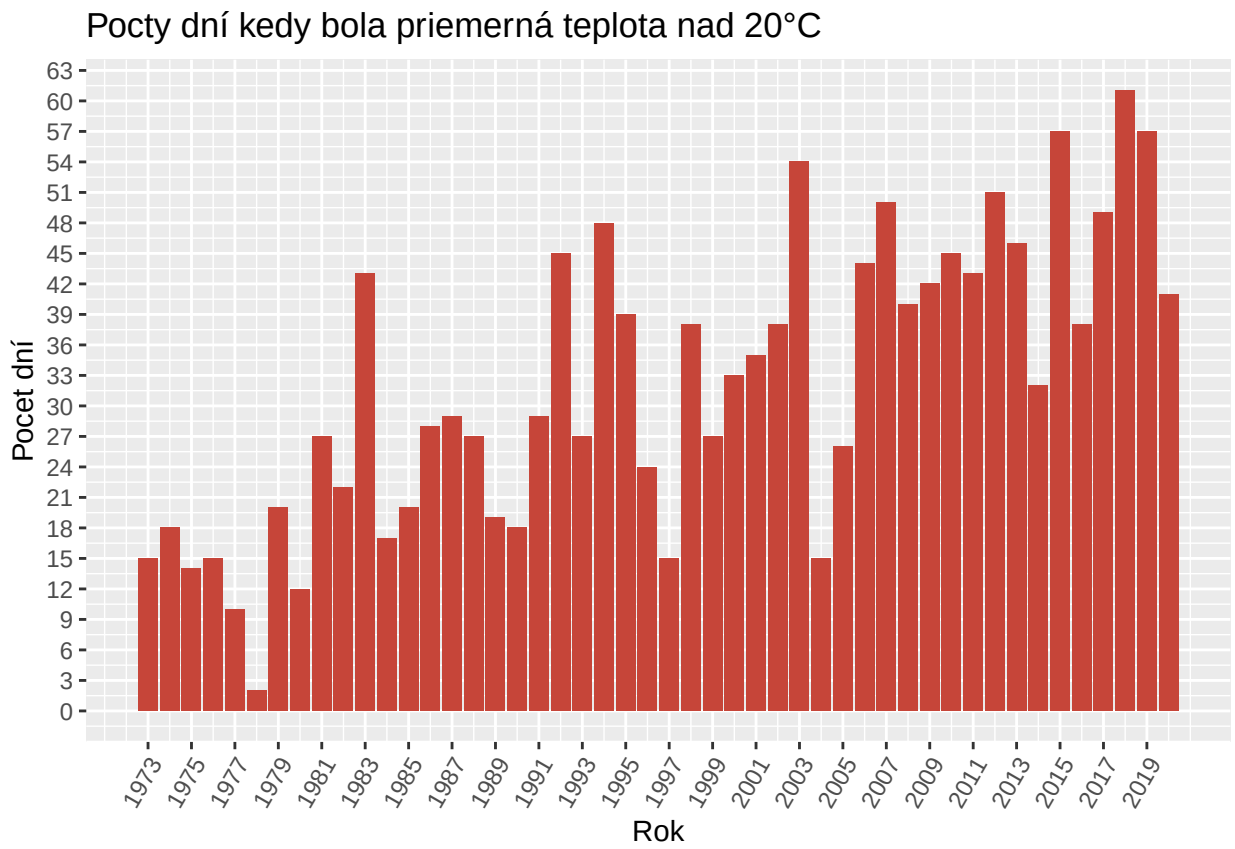

```
ggplot(data = df_tmp_groups %>% filter(group == 4), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#67be55") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,200, by = 10)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 2)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota od 0°C až 10°C"), y = "Počet dní", x = "Rok
```




```
ggplot(data = df_tmp_groups %>% filter(group == 5), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#bdb932") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,200, by = 10)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 2)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota od 10°C až 20°C"), y = "Počet dní", x = "Rok
```




```
ggplot(data = df_tmp_groups %>% filter(group == 6), aes(x=year, y=n, fill=group)) +
  geom_bar(stat="identity", fill = "#c64539") +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_y_continuous(breaks = seq(0,200, by = 3)) +
  scale_x_continuous(breaks = seq(1973,2020, by = 2)) +
  labs(title = paste("Počty dní kedy bola priemerná teplota nad 20°C"), y = "Počet dní", x = "Rok")
```



V grafoch je priemerná denná teplota rozdelená do 6 skupín nasledovne:

- pod -20°C
- od -20°C do -10°C
- od -10°C do 0°C
- od 0°C do 10°C
- od 10°C do 20°C
- nad 20°C

Pre každú skupinu a každý rok sú spočítané počty dní, kedy sa priemerná denná teplota nachádzala v danom intervale.

Z grafu pre poslednú skupinu (nad 20°C) je vidieť, že počet dní kedy teplota presahuje 20°C stúpa s pribúdajúcimi rokmi. Teda počet teplých dní sa časom zvyšuje.

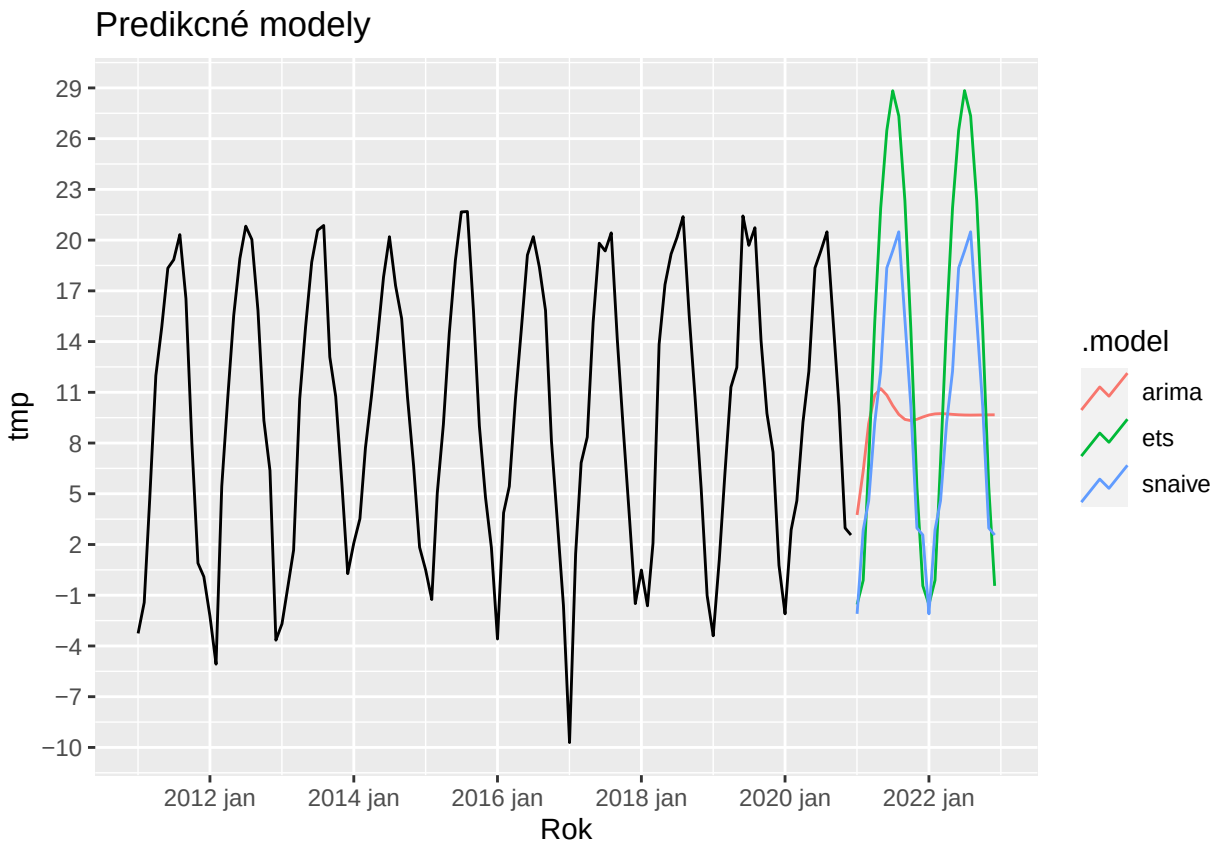
Predikčné modely na 2 roky:

1. ETS - Exponential smoothing state space model
2. ARIMA - Autoregressive integrated moving average
3. SNAIVE - Naive method

```
tmpts <- data_temperature %>%
  dplyr::mutate(
    year_month = yearmonth(paste(year, month))
  ) %>%
  dplyr::group_by(year_month) %>%
  dplyr::summarise(tmp = na.omit(mean(TMP))) %>%
  as_tsibble(
```

```
    index = year_month
    )

tmpts %>%
  model(
    ets = ETS(box_cox(tmp, 0.3)),
    arima = ARIMA(log(tmp)),
    snaive = SNAIVE(tmp)
  ) %>%
  forecast(h = "2 years") %>%
  autoplot(filter(tmpts, year(year_month) > 2010), level = NULL) +
  labs(title = "Predikčné modely", x = "Rok") +
  scale_y_continuous(breaks = seq(-10, 30, by = 3))
```

```
## Warning in log(tmp): NaNs produced
```



Lineárny model

H0: ai=0 H1: ai!=0. (vhodný je model, kde sú koeficienty štatisticky významne odlišné od 0)

V našom prípade chceme aby koeficienty sezónnych aj trendovej zložky boli rôzne od nuly, pretože vtedy existuje závislosť.

1. Sezónny model

Vidíme, že p-hodnoty konštanty(intercept) aj trendovej a sezónnej zložky modelu sú veľmi nízke (pod hladinou významnosti alfa = 0.05) teda sú štatisticky významné, teda sú dôležitou súčasťou modelu.

P-hodnota celého modelu je menšia ako hladina významnosti 0.05, teda celkovo model je významný. Teda môžeme povedať, že existuje štatisticky významná závislosť medzi teplotou a sezónnymi a trendovou zložkou časového radu teploty.

Hodnota výberového reziduálneho rozptylu (Residual standard error, RSE) je 1.783, čo je veľmi málo. Teda skutočné hodnoty teploty sa odchyľujú od odhadnutých hodnôt ležiacich na regresnej priamke približne o $\pm$ 1.783.

Hodnoty Multiple R-squared (koeficient determinácie) aj Adjusted R-squared sú vysoké. Až 95% variability dát je vysvetlených modelom. Teda model je veľmi dobrý, dobre popisuje dáta a zachytáva ich variabilitu.

1. Trendový model

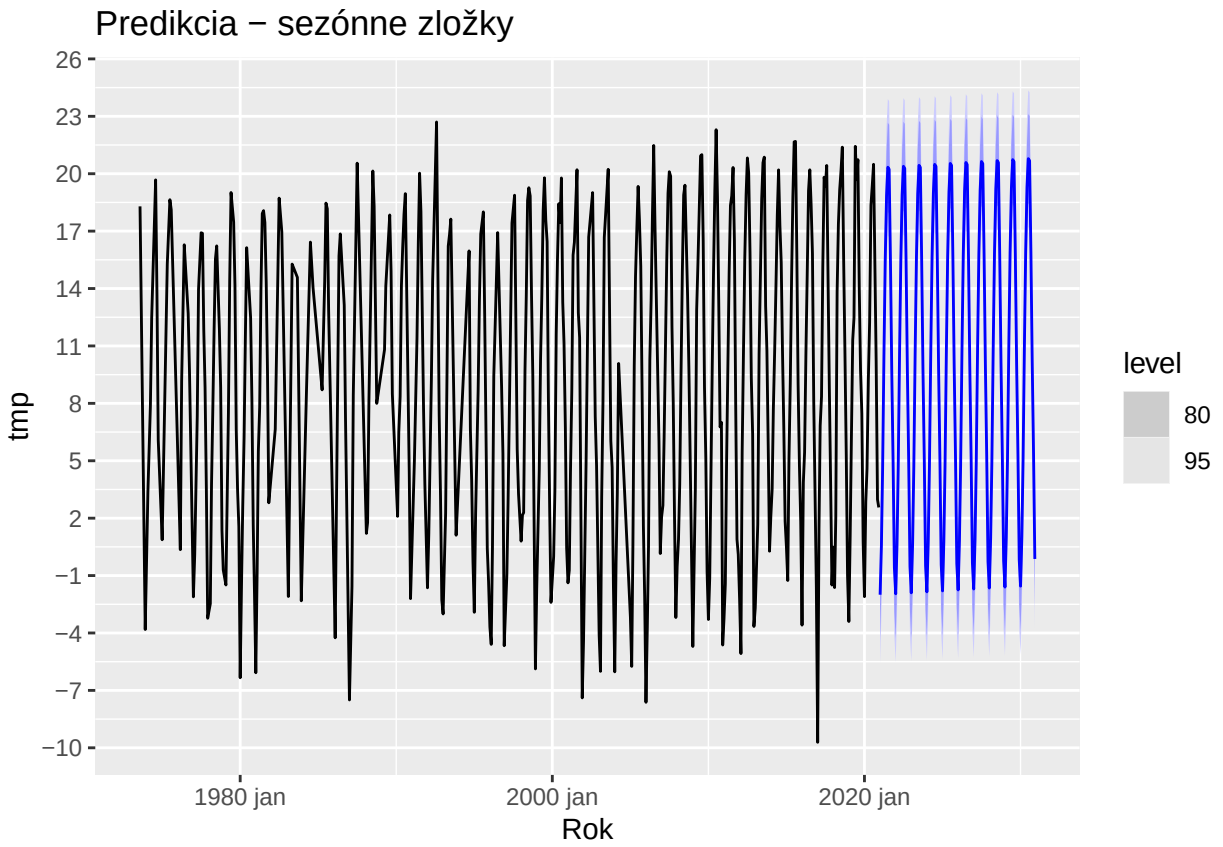
P-hodnota modelu je 0.56506, teda je väčšia ako 0.05, čo znamená, že závislosť teploty od trendovej zložky nie je štatisticky významná.

Nakoniec môžme vidieť hodnoty predikčného modelu trendovej aj sezónnych zložiek vykreslené do grafu. Trendová zložka je mierne stúpajúca čo svedčí o zvyšujúcej sa teplote v nasledujúcich rokoch, avšak táto priama závislosť nie je štatisticky významná.

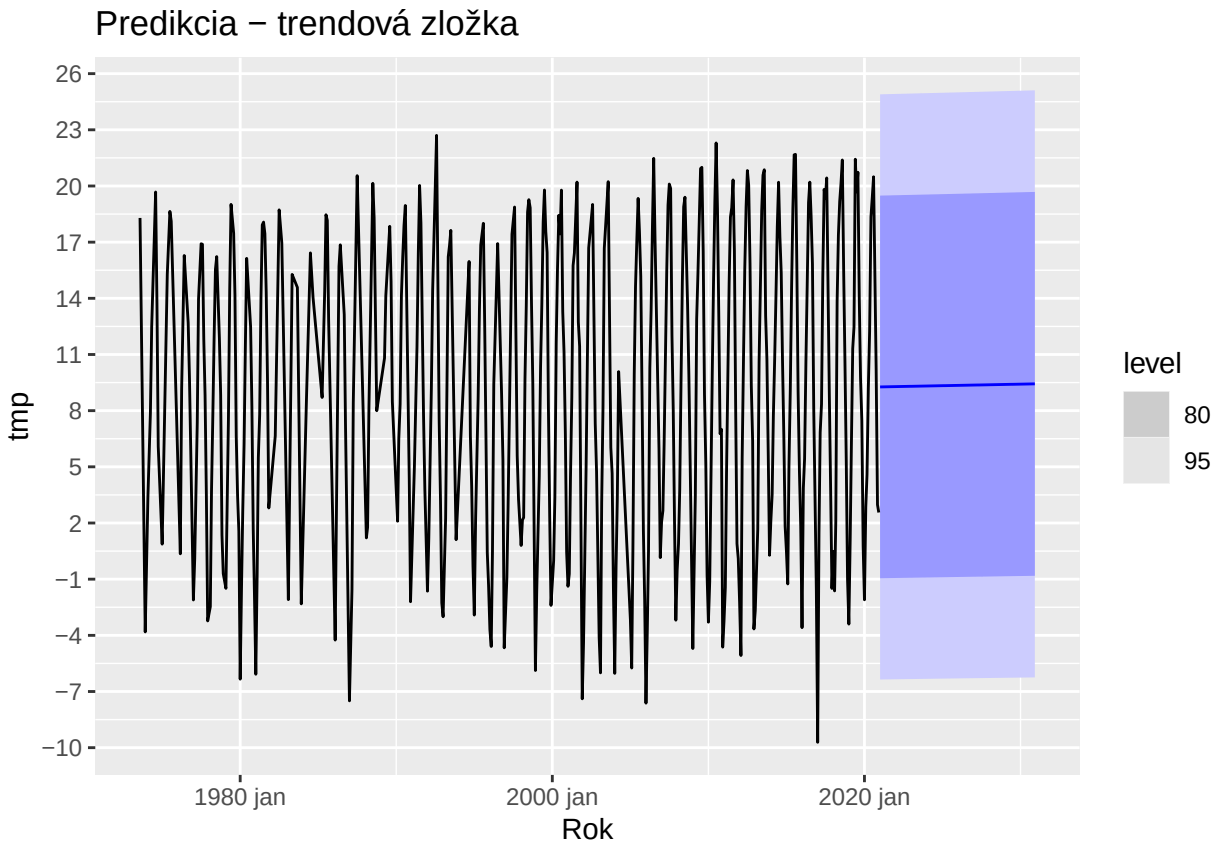
```
all_data %>%
  dplyr::mutate(
    year_month = yearmonth(DATE)
  ) %>%
  dplyr::group_by(year_month) %>%
  dplyr::summarise(tmp = na.omit(mean(TMP))) %>%
  as.data.frame %>%
  as_tsibble(
    index = year_month
    ) -> tsdf
```

```
## `summarise()` has grouped output by 'year_month'. You can override using the `.groups` argument.
```

```
tsdf %>%
  model(trend_model = TSLM(tmp ~ trend() + season())) -> season_m

tsdf %>%
  model(trend_model = TSLM(tmp ~ trend())) -> trend_m

season_m %>%
  forecast(h = "10 years") %>%
  autoplot(tsdf) +
  labs(title = "Predikcia - sezónne zložky", x = "Rok") +
  scale_y_continuous(breaks = seq(-10, 30, by = 3))
```




```
trend_m %>%
  forecast(h = "10 years") %>%
  autoplot(tsdf) +
  labs(title = "Predikcia - trendová zložka", x = "Rok") +
  scale_y_continuous(breaks = seq(-10, 30, by = 3))
```




```
report(season_m)
```

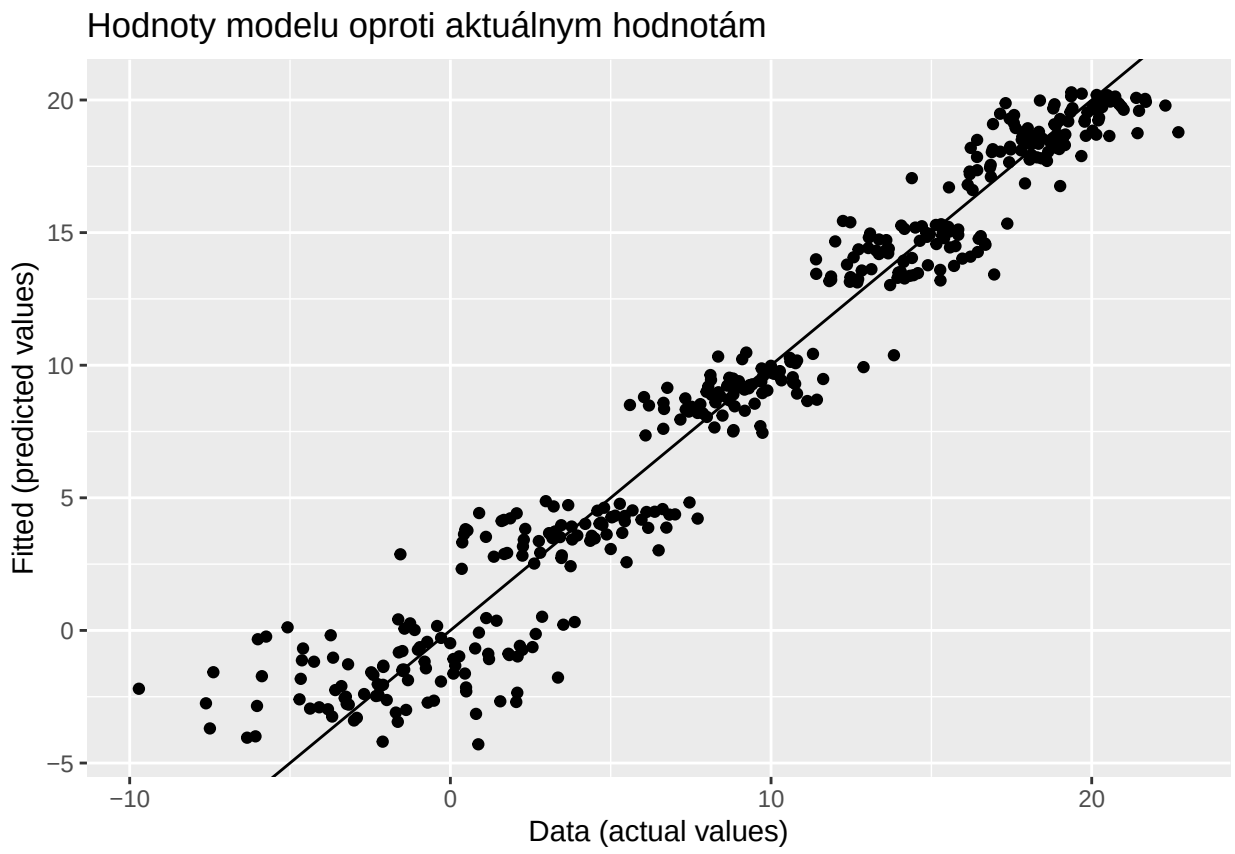
```
## Series: tmp
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.51258 -0.92440  0.02043  0.97387  5.16549
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.3689077  0.3715221 -11.759  < 2e-16 ***
## trend()         0.0041550  0.0005375   7.730 8.93e-14 ***
## season()year2   2.5613479  0.4432664   5.778 1.53e-08 ***
## season()year3   6.5576313  0.4399258  14.906  < 2e-16 ***
## season()year4  12.5142286  0.4327089  28.921  < 2e-16 ***
## season()year5  17.4727159  0.4326366  40.387  < 2e-16 ***
## season()year6  20.8286975  0.4496378  46.323  < 2e-16 ***
## season()year7  22.3146002  0.4496457  49.627  < 2e-16 ***
## season()year8  22.2015364  0.4401992  50.435  < 2e-16 ***
## season()year9  17.3326619  0.4437919  39.056  < 2e-16 ***
## season()year10 11.6594038  0.4345359  26.832  < 2e-16 ***
## season()year11  6.8829923  0.4460955  15.429  < 2e-16 ***
## season()year12  1.3753305  0.4530326   3.036  0.00256 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.783 on 397 degrees of freedom
## Multiple R-squared: 0.951,   Adjusted R-squared: 0.9495
## F-statistic: 641.5 on 12 and 397 DF, p-value: < 2.22e-16
```

```
report(trend_m)
```

```
## Series: tmp
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9151  -6.7190   0.2284   7.4203  13.9020
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.487634   0.858932   9.882   <2e-16 ***
## trend()     0.001369   0.002378   0.576    0.565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.941 on 408 degrees of freedom
## Multiple R-squared: 0.000812,    Adjusted R-squared: -0.001637
## F-statistic: 0.3316 on 1 and 408 DF, p-value: 0.56506
```

```
augment(season_m) %>%
  ggplot(aes(x = tmp, y = .fitted)) +
  geom_point() +
  labs(
    y = "Fitted (predicted values)",
    x = "Data (actual values)",
    title = "Hodnoty modelu oproti aktuálnym hodnotám"
  ) +
  geom_abline(intercept = 0, slope = 1)
```



Moving average - plávajúci priemer cez 12 mesiacov + lineárny model

Kĺzavý alebo plávajúci priemer je indikátor sledovania trendov založený na minulých hodnotách. Plávajúci priemer počítame z hodnôt teploty pre každý mesiac a rok, tieto hodnoty následne vstupujú do lineárneho modelu.

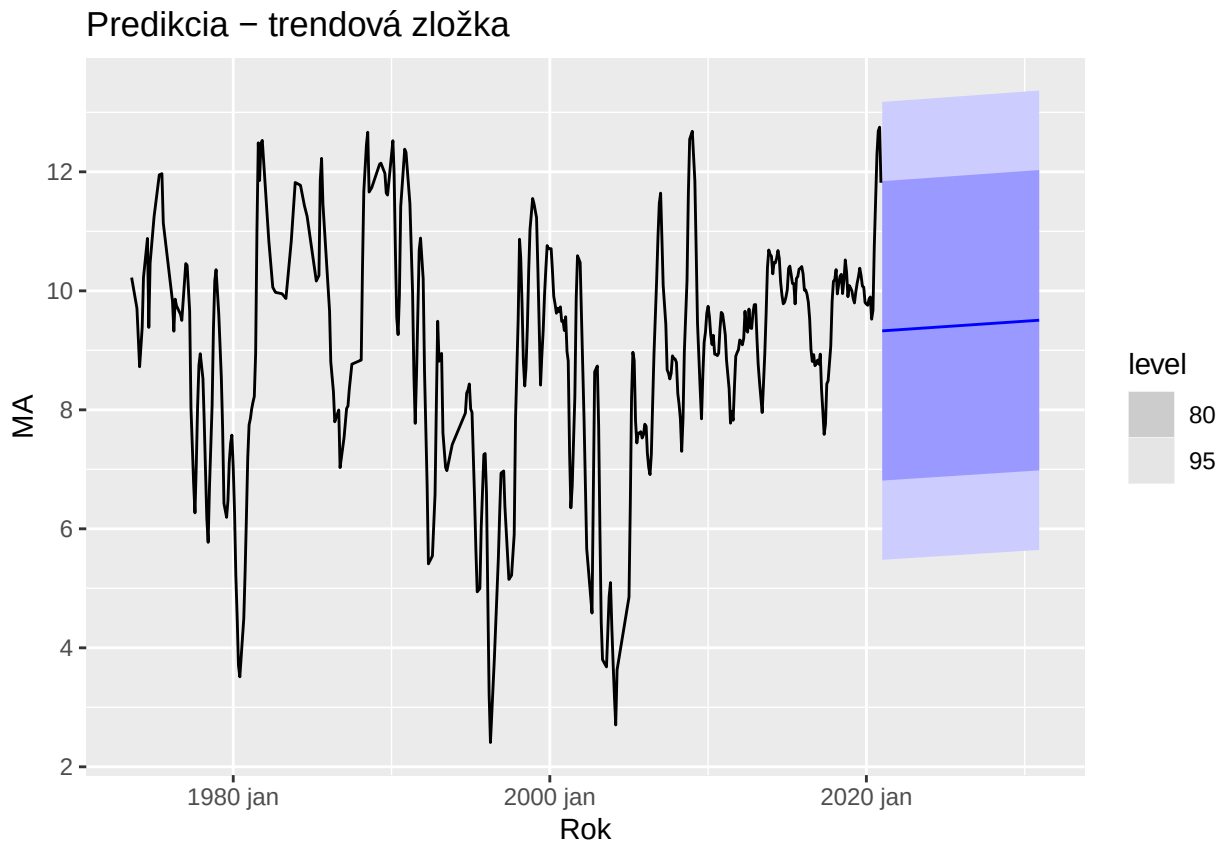
Výsledky modelu sú však o niečo horšie ako pre lineárny model. Na základe p-hodnoty ale model nezamietame a je štatisticky významný avšak zachytáva len necelé 2% variability dát.

Túto skutočnosť vidieť aj na grafe predikovaných hodnôt plávajúceho priemeru. Avšak trend je mierne stúpajúci.

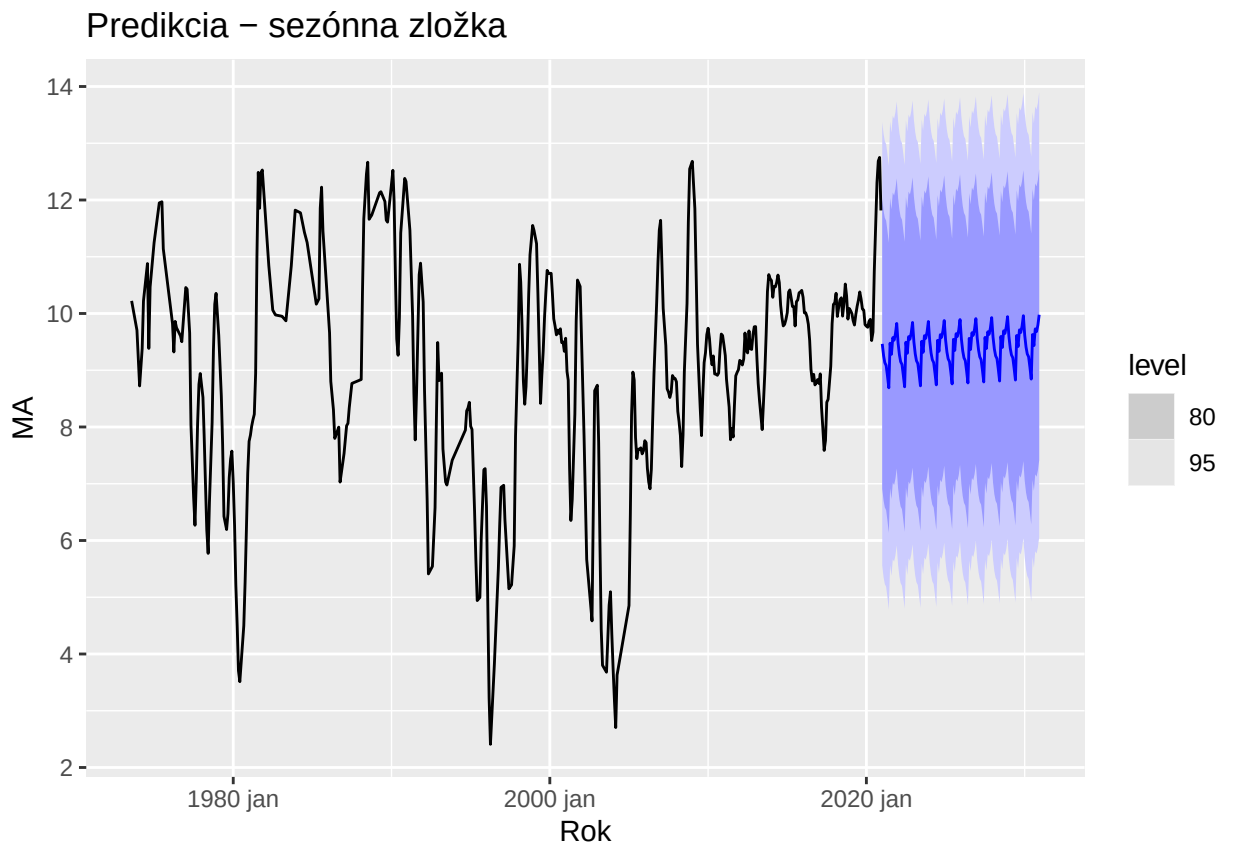
```
all_data %>%
  dplyr::mutate(
    year_month = yearmonth(DATE)
  ) %>%
  dplyr::group_by(year_month) %>%
  dplyr::summarise(tmp = na.omit(mean(TMP))) %>%
  as.data.frame %>%
  dplyr::mutate(
    MA = slider::slide_dbl(tmp, mean,
                .before = 5, .after = 6)
  ) %>%
  as_tsibble(
    index = year_month
    ) -> tsdf
```

```
## `summarise()` has grouped output by 'year_month'. You can override using the `.groups` argument.
```

```
tsdf %>%
  model(trend_model = TSLM(MA ~ trend())) -> m

tsdf %>%
  model(trend_model = TSLM(MA ~ trend() + season())) -> s_m


m %>%
  forecast(h = "10 years") %>%
  autoplot(tsdf) +
  labs(title = "Predikcia - trendová zložka", x = "Rok") +
  scale_y_continuous(breaks = seq(0, 20, by = 2))
```




```
s_m %>%
  forecast(h = "10 years") %>%
  autoplot(tsdf) +
  labs(title = "Predikcia - sezónna zložka", x = "Rok") +
  scale_y_continuous(breaks = seq(0, 20, by = 2))
```




```
report(m)
```

```
## Series: MA
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4714 -0.9863  0.2069  1.1442  3.9254
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.4689857  0.2115029   40.04   <2e-16 ***
## trend()     0.0015050  0.0005856    2.57   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.955 on 408 degrees of freedom
## Multiple R-squared: 0.01593, Adjusted R-squared: 0.01352
## F-statistic: 6.604 on 1 and 408 DF, p-value: 0.010529
```

Naive model

Výsledky naive modelu nie sú veľmi dobré nakoľko sú takmer rovnaké pre každý rok a z grafu nevidno žiadny stúpajúci trend.

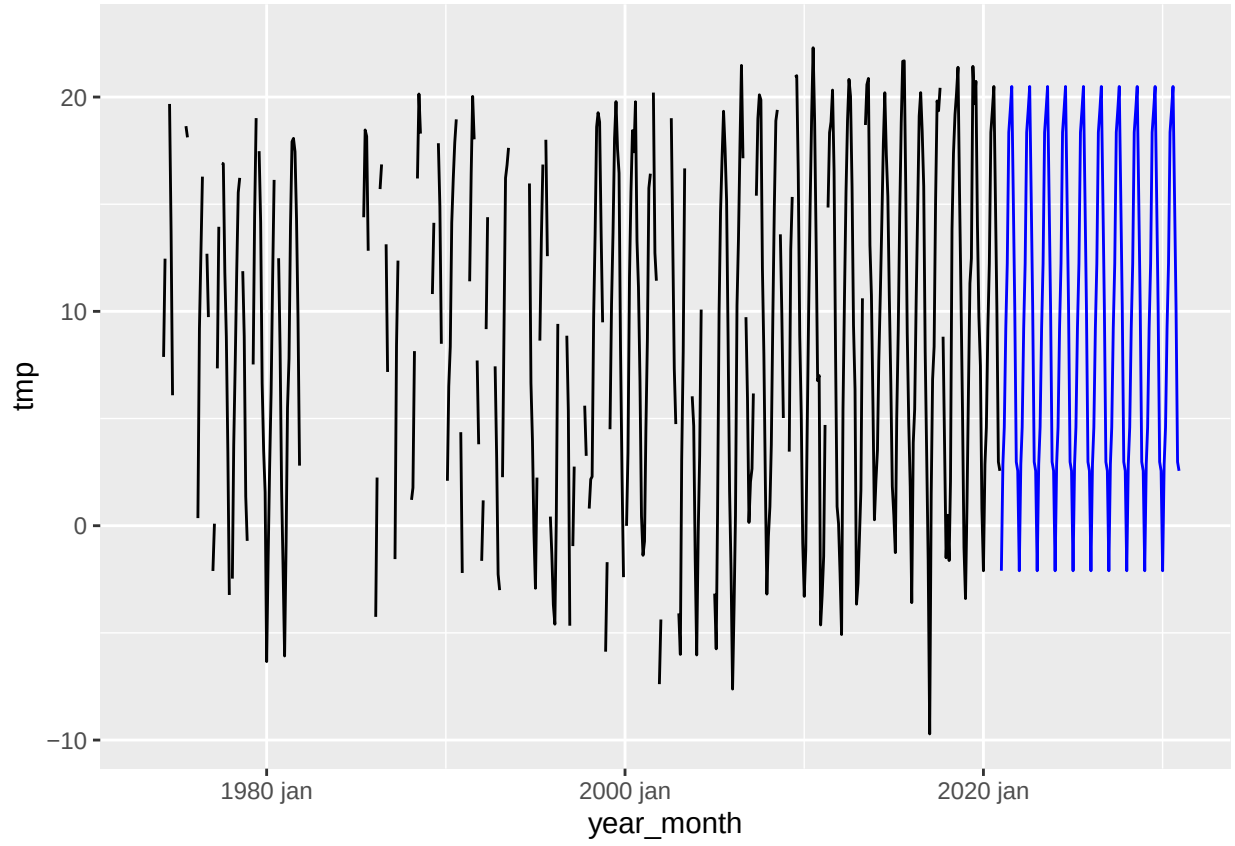
```
all_data %>%
  dplyr::mutate(
```

```
    year_month = yearmonth(DATE)
  ) %>%
  dplyr::group_by(year_month) %>%
  dplyr::summarise(tmp = na.omit(mean(TMP))) %>%
  as_tsibble(
    index = year_month
    )  %>%
  tsibble::fill_gaps()-> tsdf
```

```
## `summarise()` has grouped output by 'year_month'. You can override using the `.groups` argument.
```

```
tsdf %>%
  model(snaive = SNAIVE(tmp)) -> m

m %>%
  forecast(h = "10 years") %>%
  autoplot(filter(tsdf, year(year_month) > 2), level = NULL)
```

```
## `mutate_if()` ignored the following grouping variables:
## Column `year_month`
```




```
report(m)
```

```
## Series: tmp
## Model: SNAIVE
##
## sigma^2: 6.1217
```

```
all_data %>%
  dplyr::mutate(
    date = as_date(DATE)
  ) %>%
  select(date, LP, LP24) %>%
  separate(LP, c('lp_observation_period', 'lp_observation', NA, NA)) %>%
  filter(lp_observation_period == 12) %>%
  dplyr::mutate(lp_observation = map_dbl(lp_observation, process_col, 10)) %>%
  dplyr::select(date, lp_observation) %>%
  dplyr::group_by(date) %>%
  dplyr::summarise(LP12 = sum(lp_observation)) %>%
  as_tsibble(
    index = date
  ) %>%
  dplyr::filter(year(date)>0) %>%
  tsibble::fill_gaps() -> df_lp12


all_data %>%
  dplyr::mutate(
    date = as_date(DATE)
  ) %>%
  select(date, LP, LP24) %>%
  separate(LP, c('lp_observation_period', 'lp_observation', NA, NA)) %>%
  filter(lp_observation_period == "06") %>%
  dplyr::mutate(lp_observation = map_dbl(lp_observation, process_col, 10)) %>%
  dplyr::select(date, lp_observation) %>%
  dplyr::group_by(date) %>%
  dplyr::summarise(LP6 = sum(lp_observation)) %>%
  as_tsibble(
    index = date
  ) %>%
  dplyr::filter(year(date)>0) %>%
  tsibble::fill_gaps() -> df_lp6

all_data %>%
  dplyr::mutate(
    date = as_date(DATE)
  ) %>%
  select(date, LP24) %>%
  distinct(date, .keep_all = TRUE) %>%
  as_tsibble(
    index = date
  ) %>%
  tsibble::fill_gaps() -> df_lp24

merge(df_lp6, df_lp12, by = "date", all = TRUE) %>%
  merge(df_lp24, by = "date", all = TRUE) -> merged_df

merged_df %>%
  dplyr::mutate(
    LP = coalesce(LP12, LP6, LP24) %>% replace_na(0)
  ) %>%
```

```
  as_tsibble(
    index = date
  ) -> lp_df

lp_df %>%
  as.data.frame() %>%
  dplyr::mutate(
    year = year(date)
  ) %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(LP_SUM = na.omit(sum(LP))) %>%
  as.data.frame() %>%
  distinct(year, .keep_all = TRUE) %>%
  as_tsibble(
    index = year
    ) -> yearly_lp_df
```

Ďalej sa pozrieme na priemerné ročné teploty a počet zrážok.

Opäť pomocou výsledkov lineárneho modelu vieme určiť, či s rastúcou teplotou rastie aj počet zrážok.

Celková p-hodnota modelu je menšia ako 0.05, teda hypotézu o nulovosti koeficientov modelu zamietame. Závislosť medzi teplotou a množstvom zrážok je priama a je štatisticky významná.

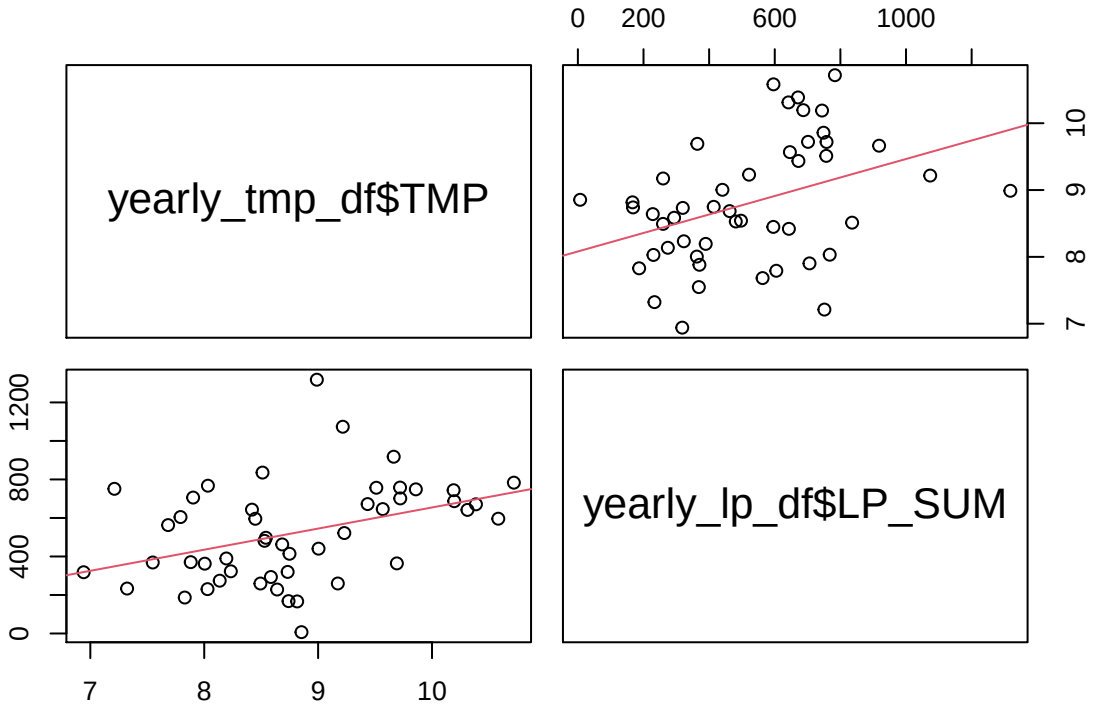
Teda našu hypotézu: **Ročná teplota rastie a s ňou rastie aj množstvo zrážok.** nezamietame.

```
data_temperature %>%
  as.data.frame() %>%
  dplyr::mutate(
    year = year(date)
  ) %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(TMP = na.omit(mean(TMP))) %>%
  as.data.frame() %>%
  distinct(year, .keep_all = TRUE) %>%
  as_tsibble(
    index = year
    ) -> yearly_tmp_df


model <- lm(yearly_tmp_df$TMP ~ yearly_lp_df$LP_SUM)
summary(model)
```

```
##
## Call:
## lm(formula = yearly_tmp_df$TMP ~ yearly_lp_df$LP_SUM)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90909 -0.55817  0.07419  0.59055  1.67601
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.0810232  0.2813621  28.721  < 2e-16 ***
## yearly_lp_df$LP_SUM 0.0013836  0.0004824   2.868  0.00621 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8606 on 46 degrees of freedom
## Multiple R-squared:  0.1517, Adjusted R-squared:  0.1333
## F-statistic: 8.228 on 1 and 46 DF,  p-value: 0.006208
```

```
pairs( ~ yearly_tmp_df$TMP + yearly_lp_df$LP_SUM, panel = function(x,y){
  points(x,y)
  abline(lm(y~x), col = 2)})
```



Zhodnotenie

Overovali sme hypotézu, že ročná teplota rastie a s ňou rastie aj množstvo zrážok.

Ako prvý krok sme sklastrovali teplotu aby sme videli nejaké zmeny v priebehu času. Podarilo sa nám vytvoriť niekoľko skupín v ktorých jasne vidieť, že priemerná teplota v skupinách sa líši.

Ďalej sme potvrdili hypotézu pomocou lineárneho modelu a dokázali sme, že priemerná ročná teplota rastie a s ňou rastie aj množstvo zrážok.