StatComp Project 2: Scottish weather

Chenyao Yu (s2156882)

Introduction

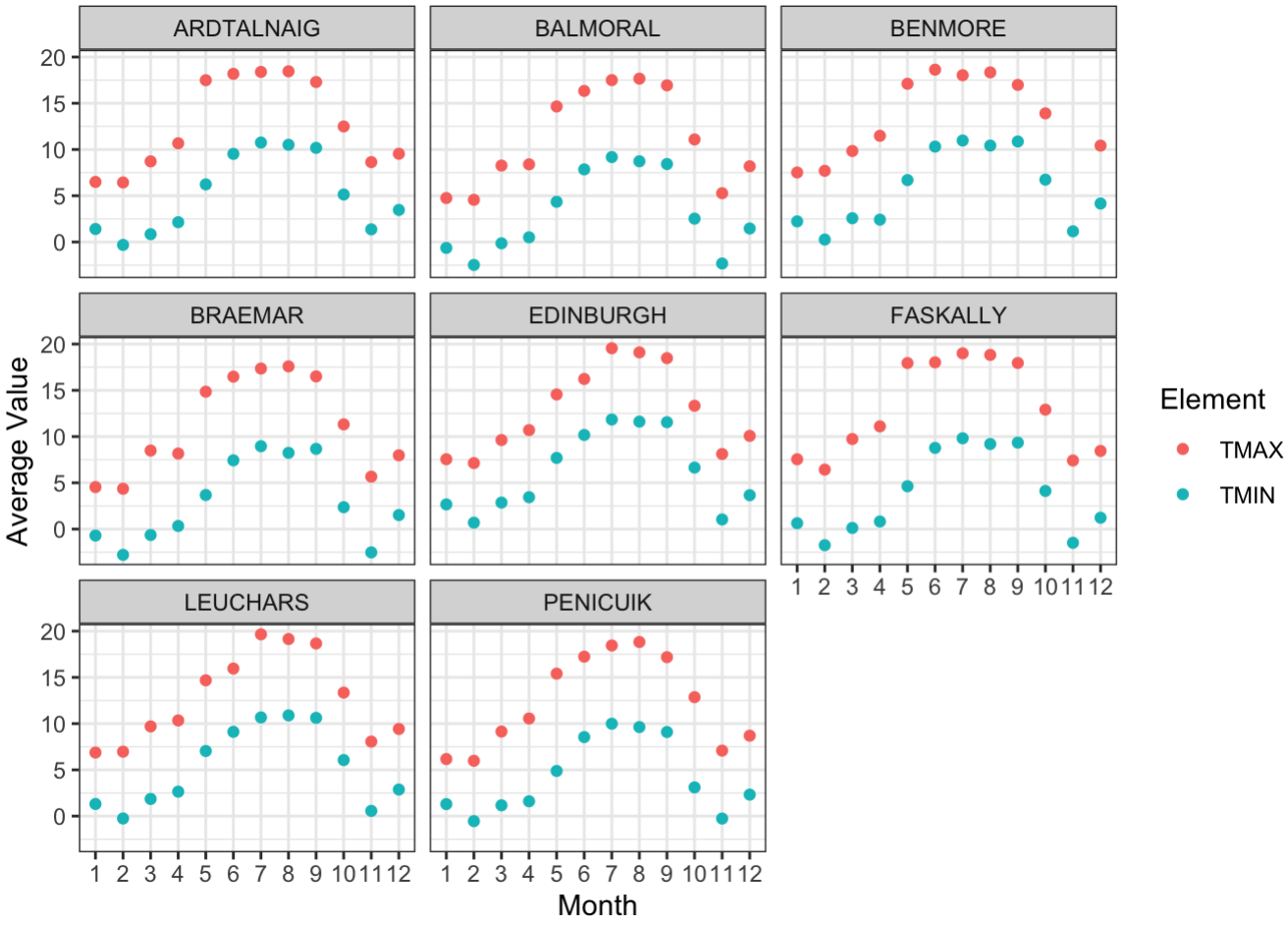
This report aims to consider weather conditions (including precipitation) from eight weather stations in Scotland and then build a model to predict the new precipitation at new locations. The data here covers the time period from 1 January 1960 to 31 December 2018.

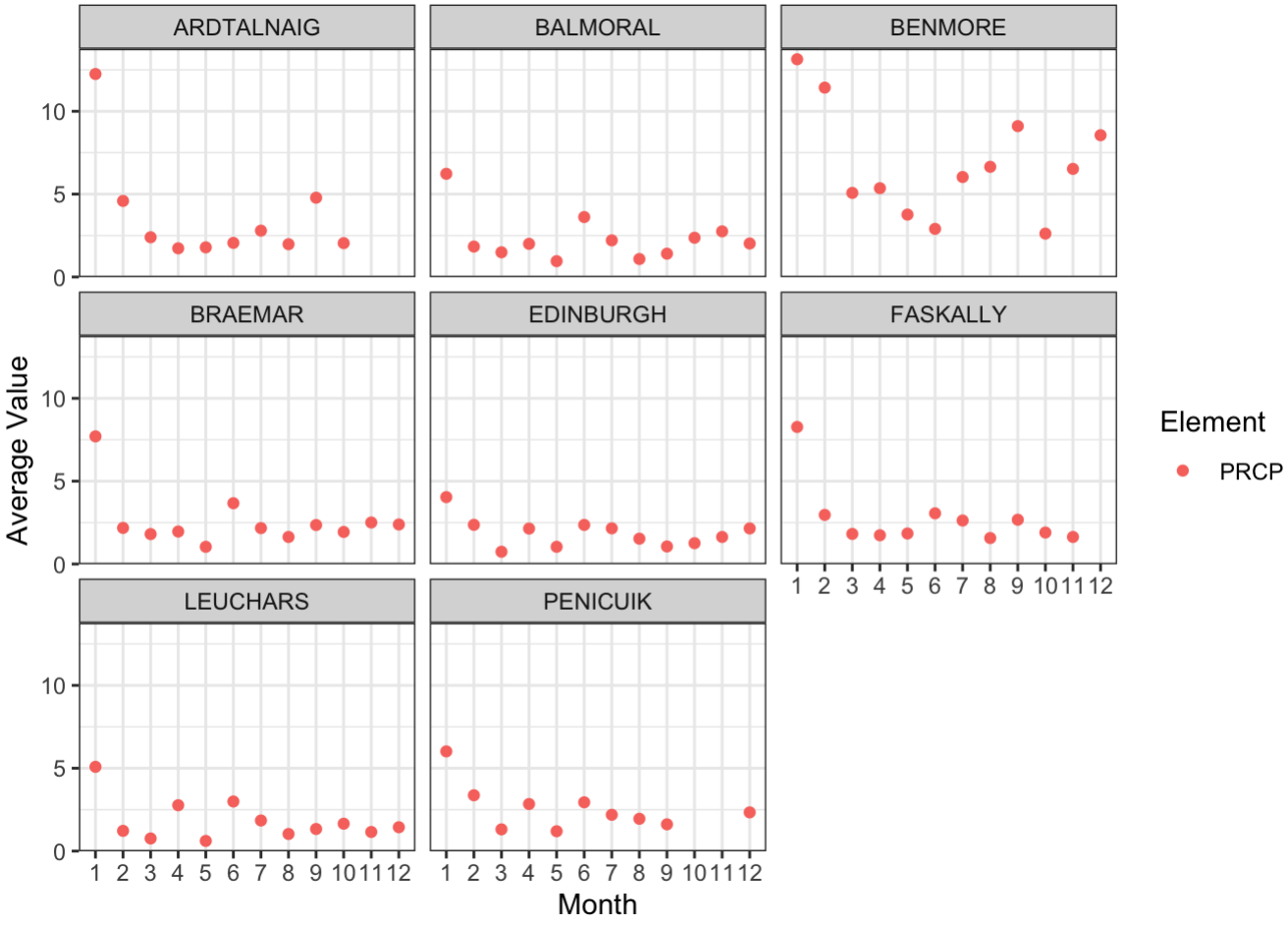
```
# load data
data(ghcnd_stations, package = "StatCompLab")
data(ghcnd_values, package = "StatCompLab")

ghcnd_stations$Name[7] <- "EDINBURGH"
ghcnd_stations$Name[8] <- "BENMORE"
```

Seasonal variability

The daily precipitation data is more challenging to model than temperature because of the mix of zeros and positive values.
We first joined the ghcnd_values and ghcnd_stations data based on ID, then selected the records where the Element column is “TMIN” or “TMAX”, grouped by Name, Element, and Month columns, calculated the mean of the Value column, and finally plotted the scatter plot of TMIN and TMAX against the months for eight stations. Then do the same for plotting the scatter plot of precipitation against the months for eight stations.

Thus the graph below shows the temperature and precipitation data in order to show the behavior.





We plot the average temperature (TMAX and TMIN) and average precipitation (PRCP) across all the stations in Scotland for the year 2016. We can see from the first plot that TMAX and TMIN both have clear seasonal effects; the average temperatures are higher in summer while lower in winter. Besides, the seasonal pattern is more obvious for TMAX, which is generally much less in winter than in summer. However, for precipitation, the seasonal effect is not obvious if any. The average precipitation in 2016 does not have significant difference between in winter and in summer due to the second graph.

Then we construct a Monte Carlo permutation test for the hypotheses:
H_0: The rainfall distribution is the same in winter as in summer
versus
H_1: The winter and summer distributions have different expected values
And here let winter be {Jan, Feb, Mar, Oct, Nov, Dec}, and let summer be {Apr, May, Jun, Jul, Aug, Sep}. Then
The next step is to determine whether the difference in precipitation between summer and winter is significant for each weather station. To achieve this, the weather station data is merged with the station information, and the data for each month is labeled as either summer or winter. The code then iterates through each weather station, calculating the precipitation in winter and summer at that station. To test the significance of the difference between these values, the absolute difference between the winter and summer averages is used as a test statistic (T = |winter average − summer average|). The code uses 1000 Monte Carlo permutations to calculate the p-value and confidence interval of this test statistic for each weather station. The resulting information can be used for further analysis.

```
ghcnd <- ghcnd_values %>%
  mutate(Summer = Month %in% 4:9) %>%
  merge(ghcnd_stations, by = "ID")

df1 <- data.frame()

# set winter and summer
for (station in ghcnd_stations$Name) {
  winter <- ghcnd %>%
    filter(Element == "PRCP", Summer == FALSE, Name == station) %>%
    pull(Value)
  winter_index_n <- length(winter)

  summer <- ghcnd %>%
    filter(Element == "PRCP", Summer == TRUE, Name == station) %>%
    pull(Value)
  summer_index_n <- length(summer)

  all_data <- ghcnd %>%
    filter(Element == "PRCP", Name == station)
  all_index_n <- nrow(all_data)
  all_index <- rownames(all_data)

  # have t test
  t_stat <- abs(mean(winter) - mean(summer))
  t_stat_vec <- c()
  sd_t_stat_perm<-c()

  # Monte Carlo 1000 times
  for (i in 1:1000) {
    samp <- sample(all_index_n, winter_index_n)
    winter_index_mc <- all_index[samp]
    summer_index_mc <- all_index[-samp]

    winter_mc <- all_data %>%
      filter(row.names(.) %in% winter_index_mc) %>%
      pull(Value)
    summer_mc <- all_data %>%
      filter(row.names(.) %in% summer_index_mc) %>%
      pull(Value)

    t_stat_mc <- abs(mean(winter_mc) - mean(summer_mc))
    t_stat_vec <- c(t_stat_vec, t_stat_mc)
  }
  p_val = mean(t_stat_vec >= t_stat)
  sd_p <- sqrt(p_val*(1-p_val)/1000)
  sd_t_stat_perm <- c(sd_t_stat_perm, sd_p)

  # calculate p-value and confidence intervals by the function
  df1 <- rbind(
    df1,
    data.frame(station_name = station,
               p_val = p_val,
               p_val_ci_lower =
                 p_value_CI(x = sum(t_stat_vec >= t_stat),
```

```
                      N = 1000)$lower,
              p_val_ci_upper =
                p_value_CI(x = sum(t_stat_vec >= t_stat),
                      N = 1000)$upper,
              sd = sd_t_stat_perm)
  )
}
```

The results of the Monte Carlo permutation test is:

Monte Carlo permutation test for each station

station_name	p_val	p_val_ci_lower	p_val_ci_upper	sd
BRAEMAR	0.000	0.0000	0.0037	0.0000
BALMORAL	0.000	0.0000	0.0037	0.0000
ARDTALNAIG	0.000	0.0000	0.0037	0.0000
FASKALLY	0.000	0.0000	0.0037	0.0000
LEUCHARS	0.030	0.0211	0.0425	0.0054
PENICUIK	0.000	0.0000	0.0037	0.0000
EDINBURGH	0.656	0.6260	0.6848	0.0150
BENMORE	0.000	0.0000	0.0037	0.0000

We can see from the table that apart from the stations LEUCHARS and ROYAL BOTANIC GARDE EDINBURGH, all other stations have zero Monte Carlo p-values and standard deviations. In particular, ROYAL BOTANIC GARDE EDINBURGH has a quite big p value of 0.6. Hence, we cannot reject the null hypothesis that the precipitation distribution is the same in winter as in summer for station ROYAL BOTANIC GARDE EDINBURGH because its Monte Carlo p-value is greater than the significance level of 5%. However the p value for LEUCHARS is 0.03, which is less than the significance level of 5%, thus for all other 7 weather stations (except EDINBURGH), we can conclude that the winter and summer distributions have different expected values. The standard deviations for all stations are not too high, which implies that the data is relatively reliable and does not have any outliers or extreme values. Lastly, from the Confidence intervals of the chart shown above, the narrow width of the confidence intervals further strengthens the validity of the test results.

Spatial weather prediction

Next, a new variable named Value_sqrt_avg is created, which represents the square root of the average precipitation for each station in each month. Specifically, the code first filters the precipitation data with the “PRCP” element from the ghcnd_values dataset, then groups the data by station ID, year, and month, calculates the square root of the average precipitation for each group, and calculates the average year value for each group (represented by DecYear). Finally, the calculation results are merged with the ghcnd_stations dataset by station ID for further analysis.

Estimation and prediction

Then define and estimate models for the square root of the monthly averaged precipitation values in Scotland. First, define a basic model M0, which includes covariates such as longitude, latitude, elevation, and time, to capture seasonal variability. Then, by adding covariates cos(2πkt) and sin(2πkt) of frequency k = 1, 2, …, we can also model seasonal variability, defining models M1, M2, M3, and M4, where the time variable t is defined to be DecYear. Lastly, according to the Model definition, five precipitation models are defined and computed, and stored in m0, m1, m2, m3, and m4.

```
# transform data for regression analysis
ghcnd_month <- ghcnd_values %>%
  filter(Element == "PRCP") %>%
  group_by(ID, Year, Month) %>%
  summarise(Value_sqrt_avg = sqrt(mean(Value)),
            DecYear = mean(DecYear)) %>%
  ungroup() %>%
  merge(ghcnd_stations, by = "ID")

# using the model in function
m0 <- precipitation_estimate(k = 0)
m1 <- precipitation_estimate(k = 1)
m2 <- precipitation_estimate(k = 2)
m3 <- precipitation_estimate(k = 3)
m4 <- precipitation_estimate(k = 4)
summary(m0)
```

```
##
## Call:
## lm(formula = formula0, data = ghcnd_month, subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94594 -0.32642 -0.02307  0.30481  1.95082
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.477e+00  1.394e+00   1.777   0.0757 .
## Longitude   -5.464e-01  1.073e-02 -50.930  < 2e-16 ***
## Latitude    -1.360e-01  2.018e-02  -6.737 1.79e-11 ***
## Elevation    4.022e-04  7.689e-05   5.231 1.75e-07 ***
## DecYear      2.411e-03  3.968e-04   6.076 1.31e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4902 on 5442 degrees of freedom
## Multiple R-squared:  0.3424, Adjusted R-squared:  0.3419
## F-statistic: 708.3 on 4 and 5442 DF,  p-value: < 2.2e-16
```

```
summary(m1)
```

```
##
## Call:
## lm(formula = paste(formula0, formula_k), data = ghcnd_month,
##     subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.98638 -0.30492 -0.01175  0.29335  1.89615
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.490e+00  1.319e+00   1.888   0.0591 .
## Longitude                   -5.467e-01  1.015e-02 -53.862  < 2e-16 ***
## Latitude                    -1.350e-01  1.909e-02  -7.069 1.76e-12 ***
## Elevation                    4.040e-04  7.274e-05   5.554 2.92e-08 ***
## DecYear                      2.376e-03  3.754e-04   6.329 2.67e-10 ***
## I(cos(2 * pi * 1 * DecYear))  1.828e-01  8.902e-03  20.536  < 2e-16 ***
## I(sin(2 * pi * 1 * DecYear)) -1.320e-01  8.870e-03 -14.885  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4637 on 5440 degrees of freedom
## Multiple R-squared:  0.4117, Adjusted R-squared:  0.411
## F-statistic: 634.5 on 6 and 5440 DF,  p-value: < 2.2e-16
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = paste(formula0, formula_k), data = ghcnd_month,
##     subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.98846 -0.30883 -0.01187  0.29641  1.89402
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.477e+00  1.318e+00   1.879 0.060239 .
## Longitude                   -5.467e-01  1.014e-02 -53.911  < 2e-16 ***
## Latitude                    -1.349e-01  1.907e-02  -7.075 1.69e-12 ***
## Elevation                    4.042e-04  7.267e-05   5.562 2.80e-08 ***
## DecYear                      2.382e-03  3.750e-04   6.351 2.31e-10 ***
## I(cos(2 * pi * 1 * DecYear))  1.829e-01  8.894e-03  20.567  < 2e-16 ***
## I(sin(2 * pi * 1 * DecYear)) -1.321e-01  8.862e-03 -14.903  < 2e-16 ***
## I(cos(2 * pi * 2 * DecYear))  3.088e-02  8.889e-03   3.474 0.000516 ***
## I(sin(2 * pi * 2 * DecYear))  7.686e-04  8.866e-03   0.087 0.930922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4633 on 5438 degrees of freedom
## Multiple R-squared:  0.413,  Adjusted R-squared:  0.4121
## F-statistic: 478.3 on 8 and 5438 DF,  p-value: < 2.2e-16
```

```
summary(m3)
```

```
##
## Call:
## lm(formula = paste(formula0, formula_k), data = ghcnd_month,
##     subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01405 -0.30872 -0.01268  0.29559  1.86840
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.470e+00  1.317e+00   1.876 0.060694 .
## Longitude                    -5.467e-01  1.013e-02 -53.958  < 2e-16 ***
## Latitude                     -1.349e-01  1.906e-02  -7.080 1.62e-12 ***
## Elevation                     4.044e-04  7.261e-05   5.570 2.67e-08 ***
## DecYear                       2.385e-03  3.747e-04   6.364 2.12e-10 ***
## I(cos(2 * pi * 1 * DecYear))  1.828e-01  8.887e-03  20.568  < 2e-16 ***
## I(sin(2 * pi * 1 * DecYear)) -1.321e-01  8.855e-03 -14.915  < 2e-16 ***
## I(cos(2 * pi * 2 * DecYear))  3.072e-02  8.882e-03   3.459 0.000546 ***
## I(sin(2 * pi * 2 * DecYear))  5.964e-04  8.859e-03   0.067 0.946329
## I(cos(2 * pi * 3 * DecYear)) -7.072e-03  8.868e-03  -0.797 0.425214
## I(sin(2 * pi * 3 * DecYear))  2.854e-02  8.872e-03   3.217 0.001305 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4629 on 5436 degrees of freedom
## Multiple R-squared:  0.4142, Adjusted R-squared:  0.4131
## F-statistic: 384.3 on 10 and 5436 DF,  p-value: < 2.2e-16
```

```
summary(m4)
```

```
##
## Call:
## lm(formula = paste(formula0, formula_k), data = ghcnd_month,
##     subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.00162 -0.30627 -0.01235  0.29501  1.88086
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.471e+00  1.317e+00   1.877  0.06056 .
## Longitude                    -5.467e-01  1.013e-02 -53.968  < 2e-16 ***
## Latitude                     -1.350e-01  1.906e-02  -7.083 1.59e-12 ***
## Elevation                     4.042e-04  7.259e-05   5.568 2.70e-08 ***
## DecYear                       2.385e-03  3.747e-04   6.366 2.09e-10 ***
## I(cos(2 * pi * 1 * DecYear))  1.829e-01  8.886e-03  20.578  < 2e-16 ***
## I(sin(2 * pi * 1 * DecYear)) -1.320e-01  8.853e-03 -14.916  < 2e-16 ***
## I(cos(2 * pi * 2 * DecYear))  3.070e-02  8.880e-03   3.457  0.00055 ***
## I(sin(2 * pi * 2 * DecYear))  6.781e-04  8.857e-03   0.077  0.93897
## I(cos(2 * pi * 3 * DecYear)) -7.045e-03  8.867e-03  -0.795  0.42691
## I(sin(2 * pi * 3 * DecYear))  2.849e-02  8.870e-03   3.212  0.00133 **
## I(cos(2 * pi * 4 * DecYear))  1.356e-02  8.900e-03   1.524  0.12763
## I(sin(2 * pi * 4 * DecYear))  1.223e-02  8.838e-03   1.384  0.16656
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4628 on 5434 degrees of freedom
## Multiple R-squared:  0.4146, Adjusted R-squared:  0.4134
## F-statistic: 320.8 on 12 and 5434 DF,  p-value: < 2.2e-16
```

	M0_estimate	M1_estimate	M2_estimate	M3_estimate	M4_estimate
(Intercept)	2.4769477	2.4899466	2.4768245	2.4704214	2.4712330
Longitude	-0.5464155	-0.5466656	-0.5466570	-0.5466809	-0.5466744
Latitude	-0.1359610	-0.1349611	-0.1349497	-0.1349466	-0.1349674
Elevation	0.0004022	0.0004040	0.0004042	0.0004044	0.0004042
DecYear	0.0024109	0.0023757	0.0023820	0.0023850	0.0023852
I(cos(2 * pi * 1 * DecYear))	NA	0.1828141	0.1829270	0.1827885	0.1828520
I(sin(2 * pi * 1 * DecYear))	NA	-0.1320307	-0.1320672	-0.1320662	-0.1320498
I(cos(2 * pi * 2 * DecYear))	NA	NA	0.0308837	0.0307235	0.0307010
I(sin(2 * pi * 2 * DecYear))	NA	NA	0.0007686	0.0005964	0.0006781
I(cos(2 * pi * 3 * DecYear))	NA	NA	NA	-0.0070723	-0.0070451
I(sin(2 * pi * 3 * DecYear))	NA	NA	NA	0.0285383	0.0284900
I(cos(2 * pi * 4 * DecYear))	NA	NA	NA	NA	0.0135609
I(sin(2 * pi * 4 * DecYear))	NA	NA	NA	NA	0.0122274

We can see from the regression results that the estimated coefficients for the first four variables (`Longitude` , `Latitude` , `Elevation` and `DecYear`) are all quite similar among the five models. The estimated coefficients for longitude and latitude are both negative and significant, indicating that they have a negative effect on the response variable. And those for elevation and Decyear are positive and significant, indicating that they have a positive effect on the response variable. And by checking the p-values, they are relatively small, suggests that the result is statistically significant. Lastly, by checking the value of R^2, they ranges from 0.34-0.41 which means that the models explain between 34% to 41% of the variance in the response variable. Therefore, further analysis of important predictors is needed.

Assessment: Station and season differences

For each weather station, we first estimate the models from the subset data without this station, and then make prediction of `Value_sqrt_avg` on the subset data with this station only. So we can get the prediction scores (SE and DS scores) for each station in terms of each model (M0 to M4). We take average of each type of scores to get the average assessment score for each model-station and present them in the table and plot. And for the assessment scores for season differences, we use the data from station cross validation then group by month to calculate and compare the mean value.

```
df_as_station <- data.frame()

# consider each weather station
for (name in ghcnd_stations$Name) {
  df <- data.frame()
  # For every Model
  for (i in 0:4) {
    mm0 <- precipitation_estimate(k = i, subset = (ghcnd_month$Name != name))

 # make prediction of Value_sqrt_avg
    pred_m0 <- predict(mm0,
                       newdata = ghcnd_month %>%
                         filter(Name == name),
                       se.fit = T)

 # calculate proper score for SE assessment
    score_m0_se <- proper_score("se",
                            obs = ghcnd_month$Value_sqrt_avg[ghcnd_month$Name == nam
e],
                            mean = pred_m0$fit)
 # calculate proper score for DS assessment
     score_m0_ds <- proper_score("ds",
                            obs = ghcnd_month$Value_sqrt_avg[ghcnd_month$Name == nam
e],
                            mean = pred_m0$fit,
                            sd = sqrt(pred_m0$se.fit ^ 2 +
                                        sum(mm0$residuals ^ 2) /mm0$df.residual))

     df <- rbind(df,
                 ghcnd_month %>%
                   filter(Name == name) %>%
                   transmute(model = paste0("M", i), Year, Month, se = score_m0_se, d
s = score_m0_ds))
  }

  df_as_station <- rbind(
    df_as_station,
    df %>%
      mutate(station_name = name)
  )
}
```

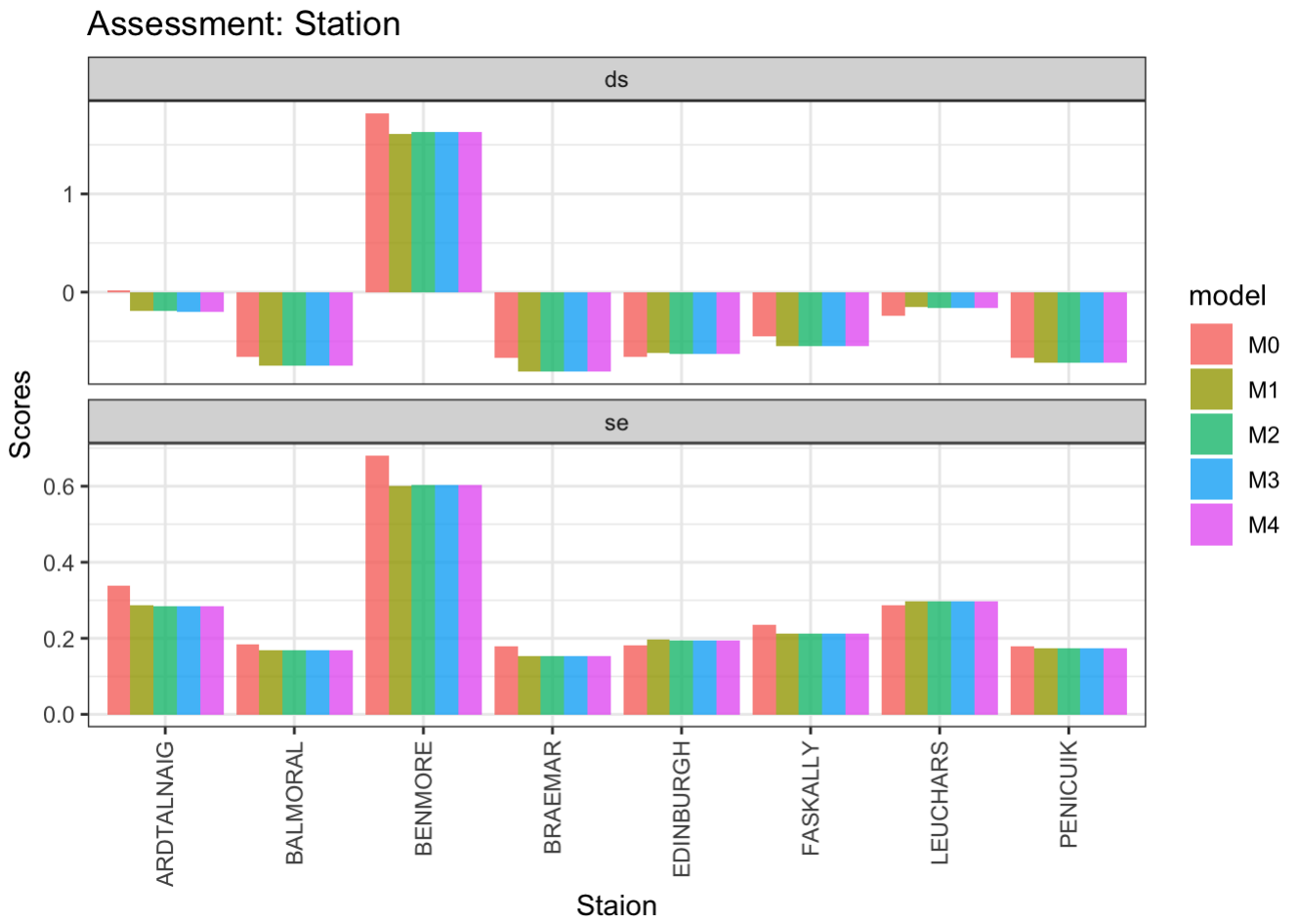
Assessment: Station (SE scores)

station_name	se_M0	se_M1	se_M2	se_M3	se_M4
ARDTALNAIG	0.3394	0.2857	0.2850	0.2838	0.2836
BALMORAL	0.1831	0.1684	0.1686	0.1687	0.1686
BENMORE	0.6792	0.6015	0.6027	0.6021	0.6023
BRAEMAR	0.1788	0.1545	0.1544	0.1538	0.1535

station_name	se_M0	se_M1	se_M2	se_M3	se_M4
EDINBURGH	0.1811	0.1957	0.1950	0.1950	0.1949
FASKALLY	0.2344	0.2133	0.2121	0.2114	0.2114
LEUCHARS	0.2867	0.2974	0.2965	0.2964	0.2962
PENICUIK	0.1801	0.1748	0.1743	0.1743	0.1743

Assessment: Station (DS scores)

station_name	ds_M0	ds_M1	ds_M2	ds_M3	ds_M4
ARDTALNAIG	0.0136	-0.1929	-0.1957	-0.2013	-0.2019
BALMORAL	-0.6543	-0.7455	-0.7451	-0.7448	-0.7455
BENMORE	1.8181	1.6145	1.6275	1.6278	1.6295
BRAEMAR	-0.6718	-0.8065	-0.8074	-0.8104	-0.8113
EDINBURGH	-0.6605	-0.6241	-0.6276	-0.6277	-0.6282
FASKALLY	-0.4500	-0.5448	-0.5504	-0.5537	-0.5537
LEUCHARS	-0.2372	-0.1561	-0.1598	-0.1599	-0.1609
PENICUIK	-0.6651	-0.7169	-0.7197	-0.7197	-0.7201



From the table and plot above, we can see that the score assessments are not equally effective in predicting the various stations. For the Dawid-Sebastiani scores, station BENMORE stands out with the only positive value, while for the Squared Error, BENMORE has an extremely high score compared to the others, indicating that the model performs poorly in predicting this particular station. Additionally, overall, the higher frequency models (M1 to M4) have lower scores than M0 for each station.

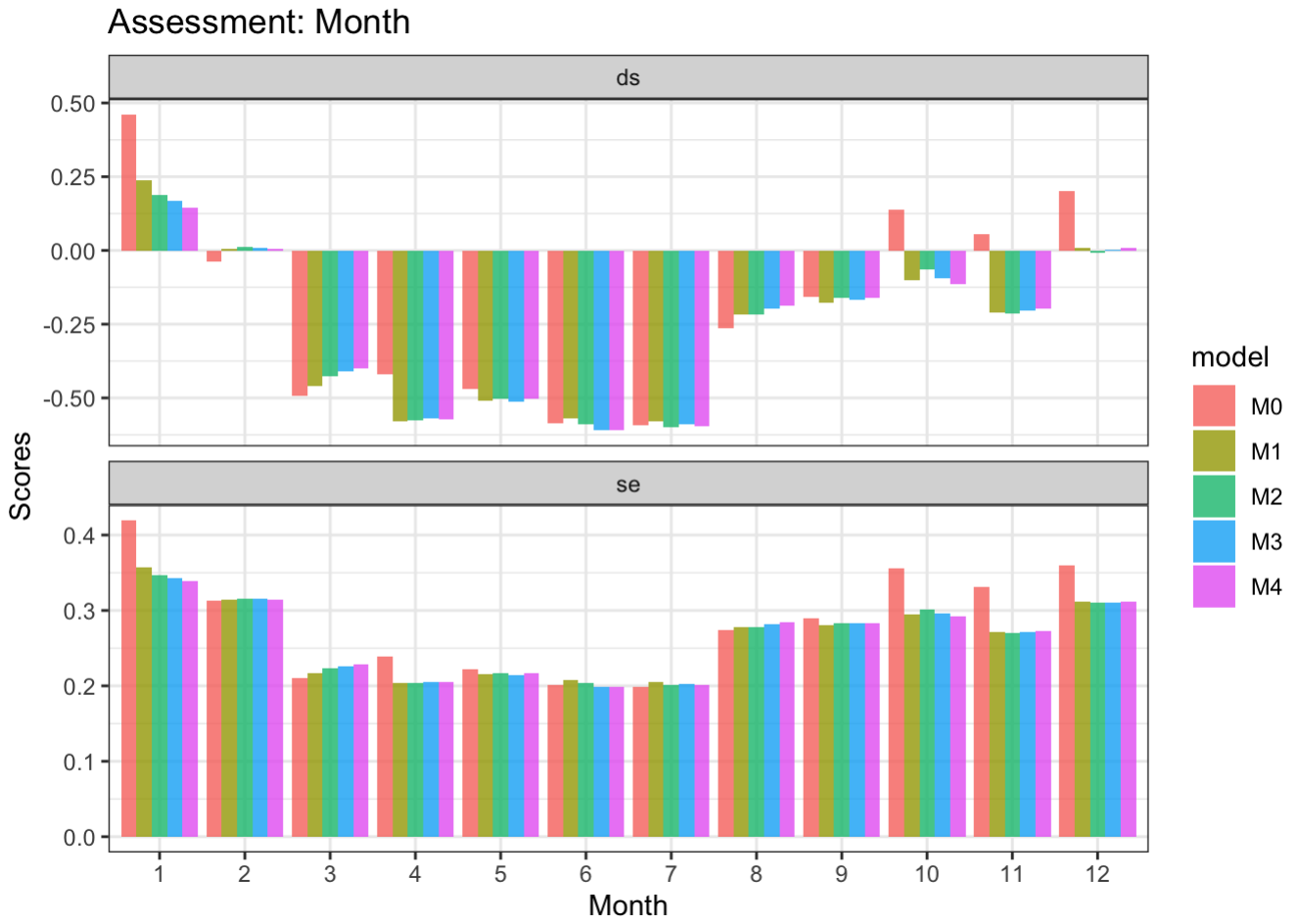
Assessment: Month (SE scores)

Month	se_M0	se_M1	se_M2	se_M3	se_M4
1	0.4189	0.3566	0.3469	0.3431	0.3384
2	0.3130	0.3143	0.3150	0.3153	0.3143
3	0.2100	0.2171	0.2233	0.2263	0.2286
4	0.2389	0.2035	0.2036	0.2052	0.2045
5	0.2226	0.2155	0.2165	0.2142	0.2165
6	0.2014	0.2079	0.2033	0.1989	0.1992
7	0.1989	0.2046	0.2009	0.2028	0.2010
8	0.2736	0.2785	0.2782	0.2824	0.2846
9	0.2896	0.2810	0.2837	0.2825	0.2835
10	0.3554	0.2950	0.3016	0.2958	0.2926

Month	se_M0	se_M1	se_M2	se_M3	se_M4
11	0.3317	0.2710	0.2702	0.2716	0.2724
12	0.3598	0.3123	0.3098	0.3104	0.3118

Assessment: Month (DS scores)

Month	ds_M0	ds_M1	ds_M2	ds_M3	ds_M4
1	0.4602	0.2384	0.1895	0.1697	0.1440
2	-0.0368	0.0056	0.0105	0.0095	0.0063
3	-0.4928	-0.4611	-0.4273	-0.4111	-0.3984
4	-0.4207	-0.5787	-0.5775	-0.5697	-0.5731
5	-0.4682	-0.5082	-0.5035	-0.5139	-0.5032
6	-0.5845	-0.5694	-0.5909	-0.6109	-0.6096
7	-0.5928	-0.5807	-0.5980	-0.5892	-0.5974
8	-0.2642	-0.2173	-0.2178	-0.1968	-0.1865
9	-0.1578	-0.1774	-0.1597	-0.1666	-0.1607
10	0.1392	-0.1018	-0.0655	-0.0955	-0.1133
11	0.0537	-0.2098	-0.2132	-0.2038	-0.1984
12	0.2003	0.0091	-0.0067	0.0013	0.0099



When it comes to assessing seasonal differences, we observe the same trend as in station assessment: the higher frequency models (M1 to M4) generally have lower scores than M0 for each station. In addition, the bar plot clearly shows a U-shaped pattern for both Squared Error and Dawid-Sebastiani scores, indicating that the scores are typically lower in summer and higher in winter. Therefore, we can conclude that the models perform better at predicting precipitation in summer than in winter.

Code appendix

Function definitions

```
# Chenyao Yu (s2156882)

# Place your function definitions that may be needed in the report.Rmd, including fun
ction documentation.
# You can also include any needed library() calls here

#' Construct Bayesian CI for Monte Carlo p-values
#'
#' @param x How many times we observe a randomised test statistic as extreme as or mo
re extreme than the observed test statistic
#' @param N Total number of replications
#' @param level The confidence level required
#'
#' @return A list contains upper bar and lower bar of the CI

p_value_CI <- function(x, N, level = 0.95) {
  return(list(lower = qbeta((1 - level) / 2,
                            shape1 = 1 + x,
                            shape2 = 1 + N - x),
              upper = qbeta(1 - (1 - level) / 2,
                            shape1 = 1 + x,
                            shape2 = 1 + N - x)))
}

#' Estimate model for the square root of monthly average precipitation
#'
#' @param k Frequency for adding covariates to capture seasonal variability
#' @param subset Specification of the rows to be used: defaults to all rows
#'
#' @return An object of class "lm"

precipitation_estimate <- function(k = 0, subset = NULL) {
  formula0 <- "Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear"

  m0 <- lm(formula0,
           data = ghcnd_month, subset = subset)

  formula_k <- paste(
    "+ I(cos(2 * pi *",
    1:k,
    "* DecYear)) + I(sin(2 * pi *",
    1:k,
    "* DecYear))", collapse = " ")

  mk <- lm(paste(formula0,
                 formula_k),
           data = ghcnd_month, subset = subset)

  if (k == 0) {
    return(m0)
  } else {
    return(mk)
```

```
  }
}
```