

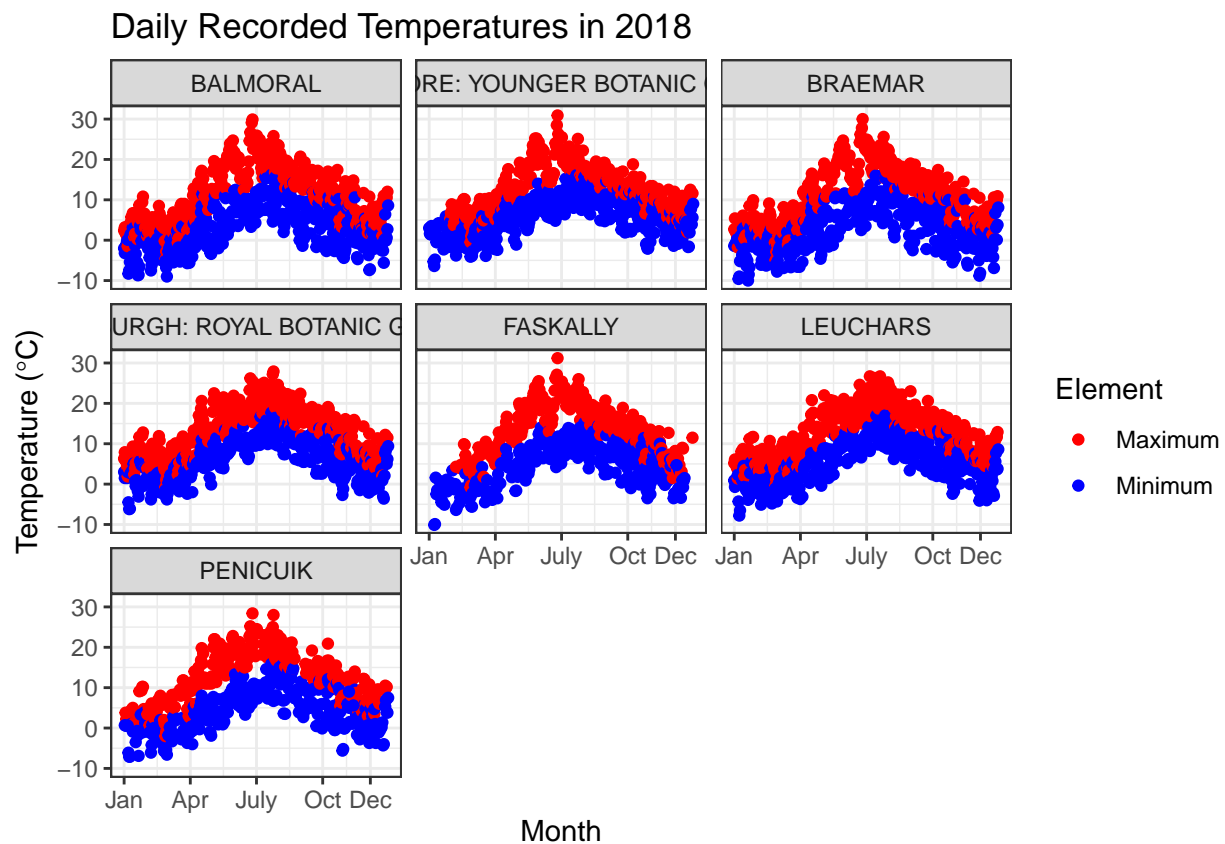
StatComp Project 2: Scottish weather

Conor Cassidy (s2077927)

Seasonal variability

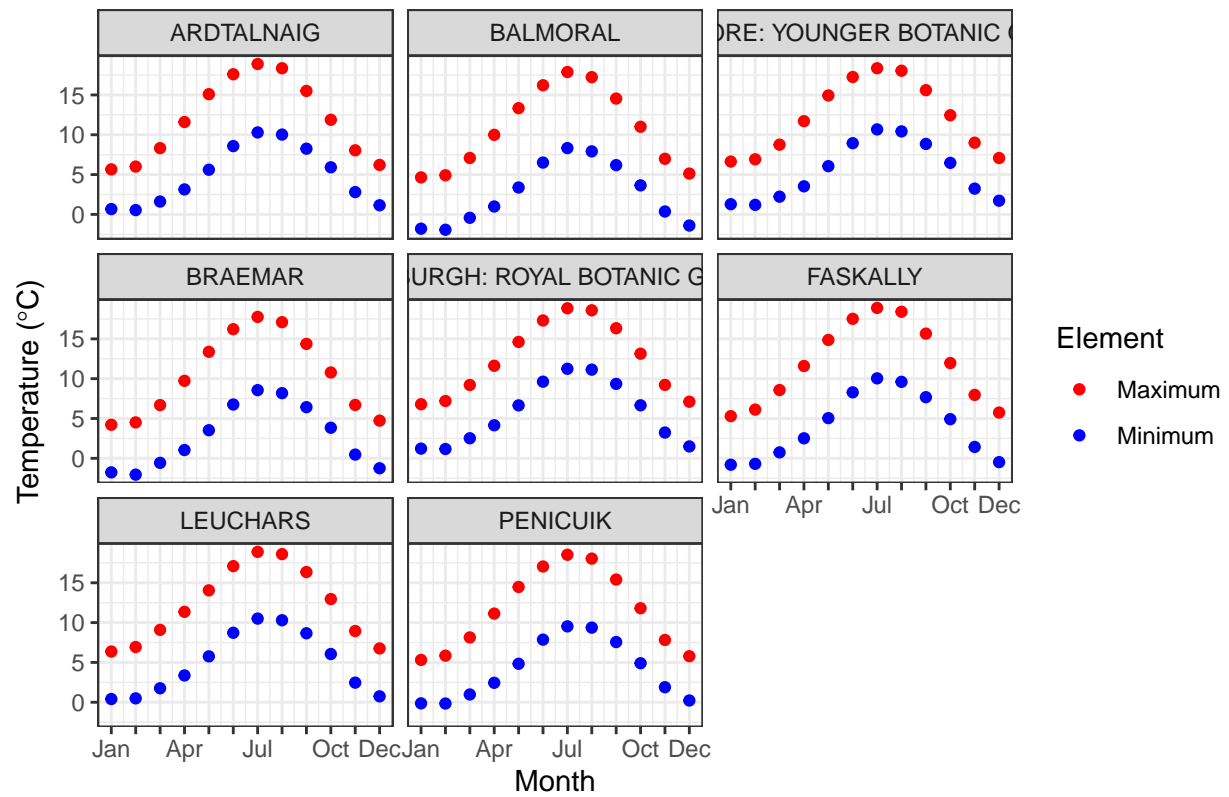
In this first section of the report, we investigate the seasonal variability of Scottish weather. To do so, we use a subset of data from The Global Historical Climatology Network.¹ This data contains daily observations for the level of precipitation measured as well as the minimum and maximum temperature recorded at 8 weather stations around Scotland from 1 January 1960 to 31 December 2018. Note, however, that this data is not complete and there are some observations missing from the various stations. One example of this is that there is no data observed for the Ardtalnaig station in 2018 and hence why it does not have a panel in two of the figures below.

We investigate the seasonality using the following series of plots to visualise patterns in the data. For simplicity, we let winter be the months of January, February, March, October, November, and December. Similarly, let the summer months be April, May, June, July, August, and September.

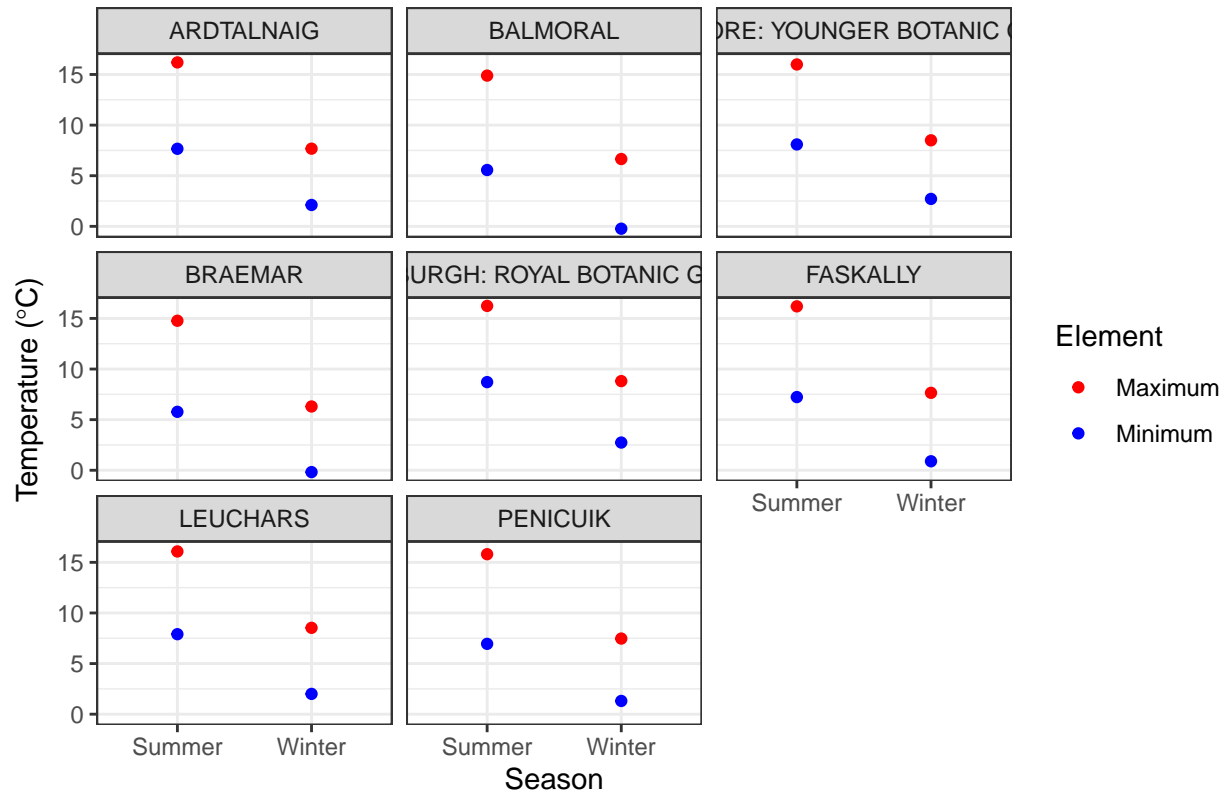


¹<https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily>

Monthly Average Temperatures



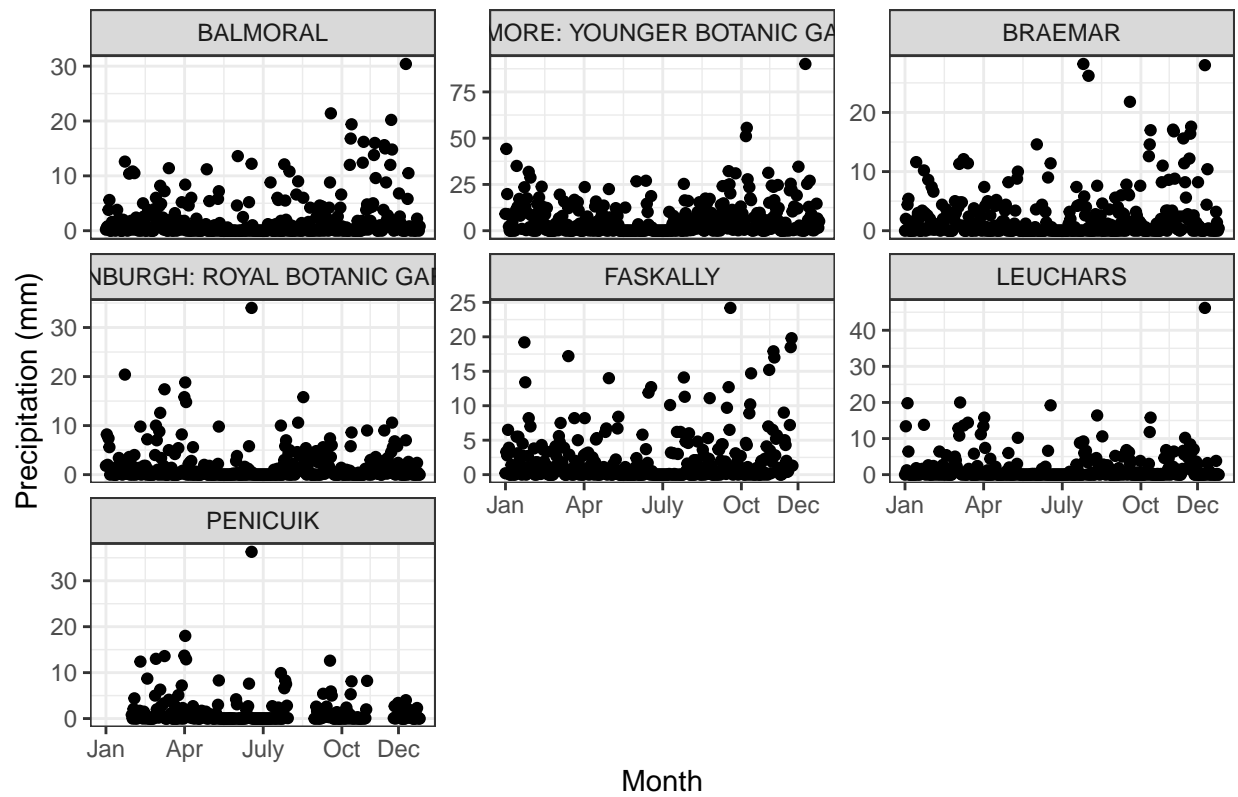
Season Average Temperatures



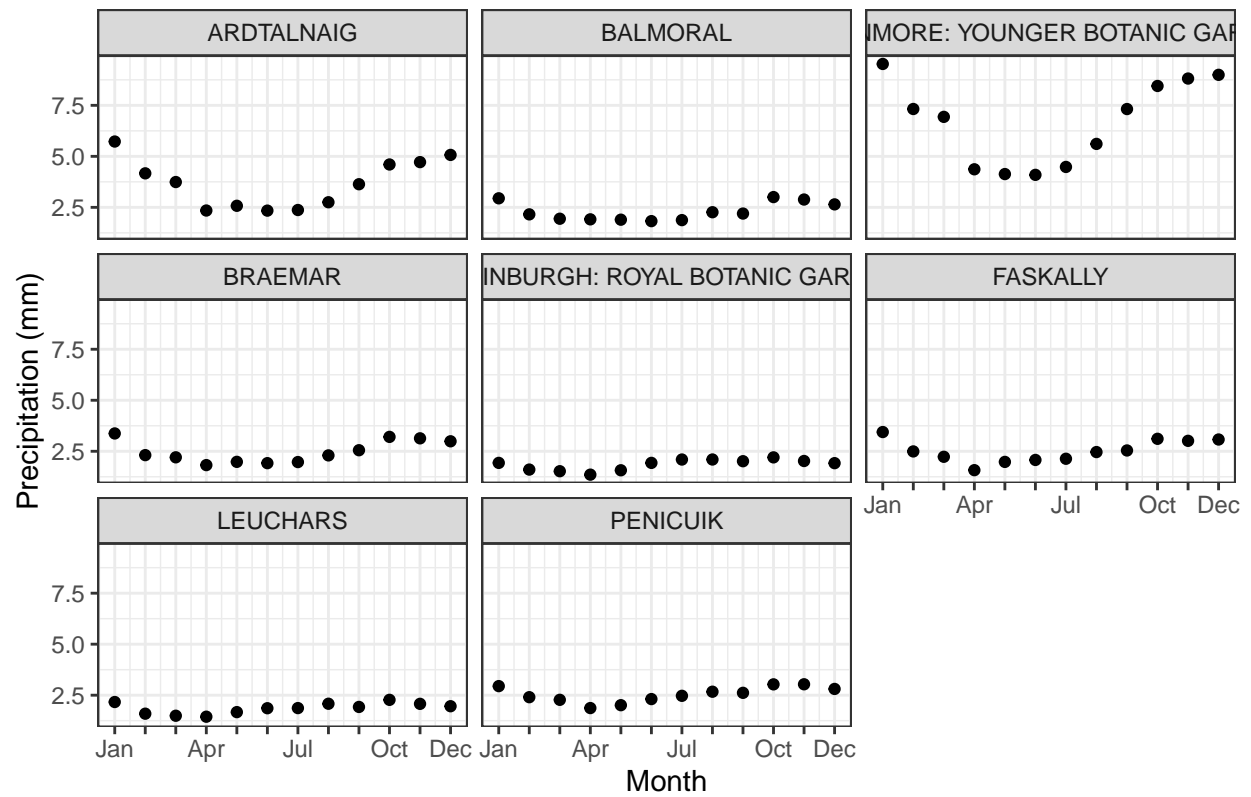
The above plots on the temperature recorded zoom out in averages from a daily recorded minimum and maximum temperature, to average monthly temperatures, and to make the difference even more abundantly clear, we finally show the average recorded temperatures for both seasons. There is a noticeable seasonality in the daily recorded temperatures in 2018. Both the minimum and the maximum temperatures appear to be higher in summer months and lower in the winter months. This is further accentuated in the monthly averages plot which clearly shows the seasonality in temperature at each station. Temperatures start and end the year low, with a unimodal peak in the summer months between these two troughs in winter months. Finally, the difference in temperature in summer and winter is clear to see from the third plot of seasonal averages. The minimum and maximum recorded temperatures are both lower, on average, in the winter than in the summer. The findings above make intuitive sense: on average, a winter day is colder than a summer day.

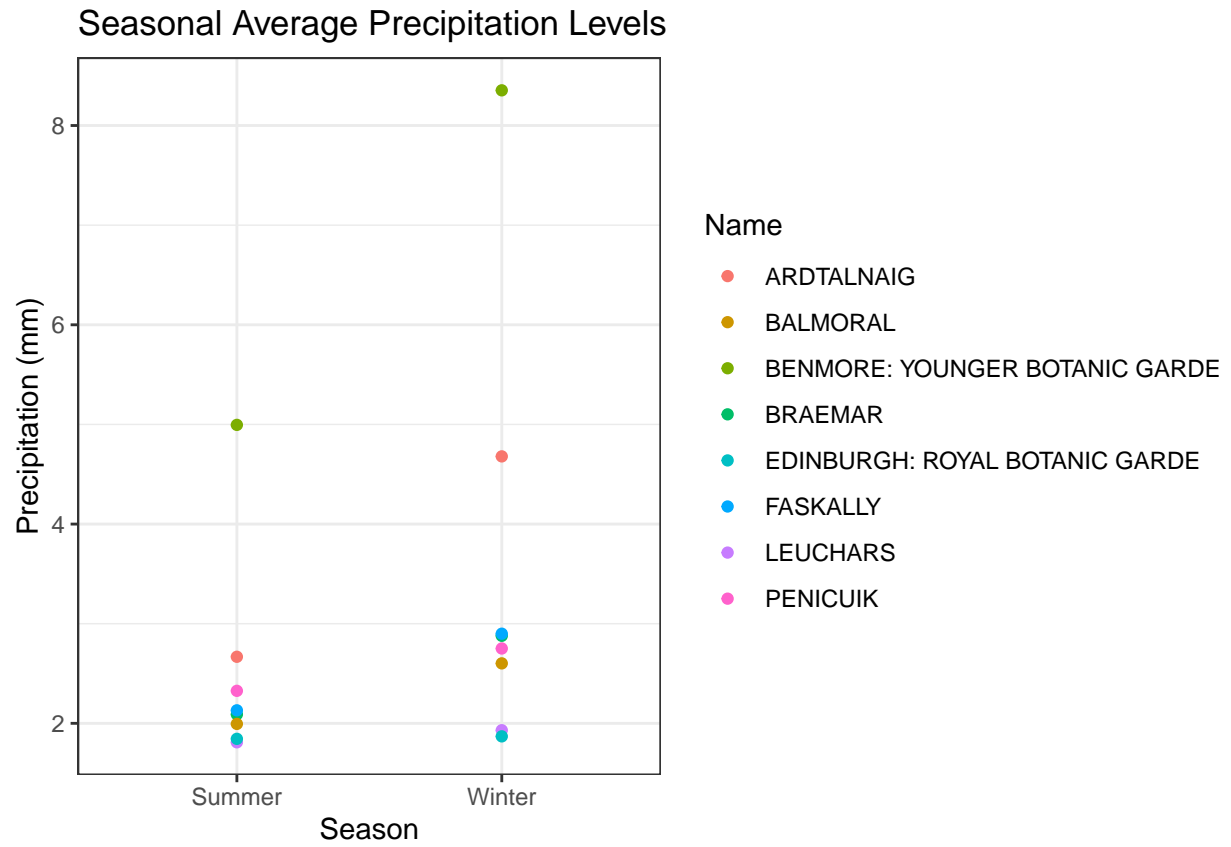
We now investigate seasonality in precipitation in the following series of plots.

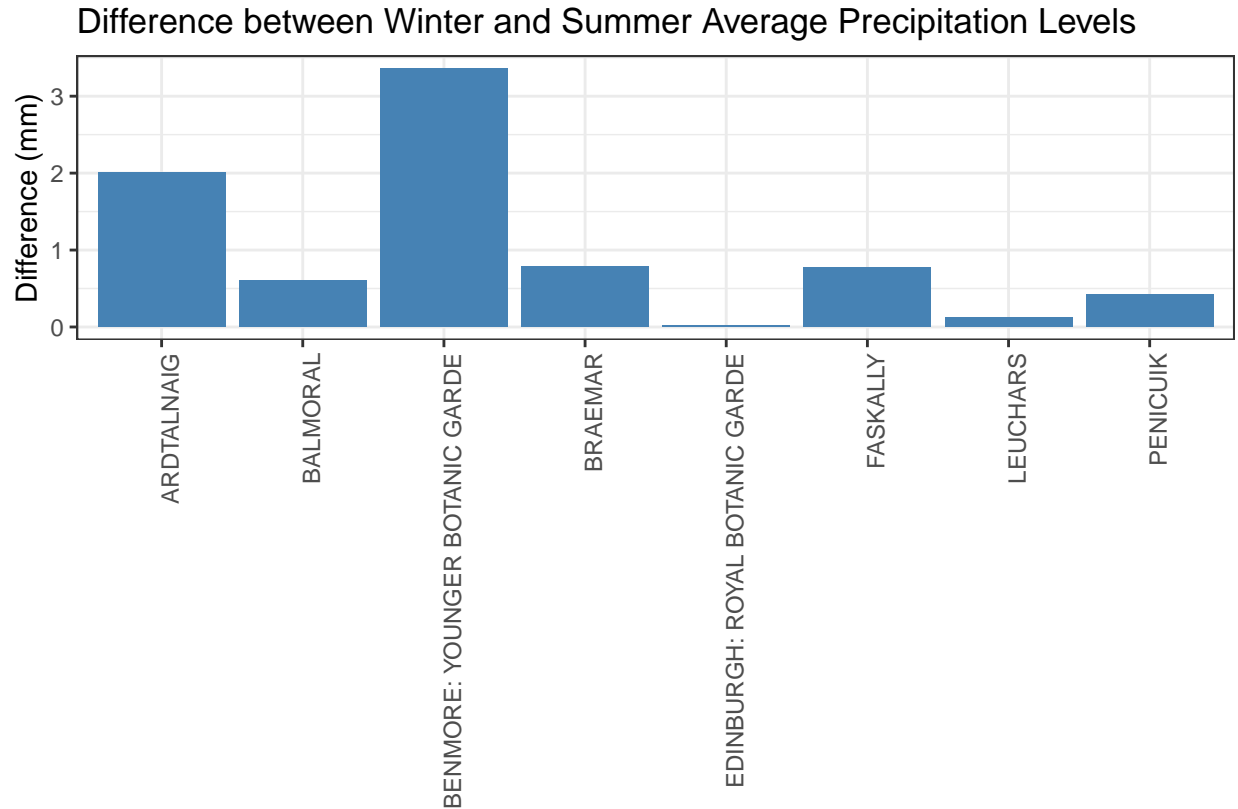
Daily Recorded Precipitation in 2018



Monthly Average Precipitation Levels







Though we now measure precipitation, again in the plots above we start by plotting daily recorded levels in 2018, then show the average level for each month, and then seasonal averages. We also plot the difference between seasonal average precipitation levels here. We note that in the first plot of the daily recorded precipitation, the scale for the Precipitation axis (y-axis) is not the same for each plot. This is case due to the discrepancy in the value of outliers for each station so that we could still get a good idea of the precipitation levels at each station. In the daily recorded values plot for 2018, it is very difficult to argue for any seasonality in the observations: the level of precipitation seems to be mostly dominated by zeros, and the days with some precipitation are not at particularly higher values on average during either winter or summer days.

This is not strictly the case in the second plot, however, which shows the monthly average precipitation levels at each station. Here, there appears to be some observable seasonality in the stations that observe higher average precipitation levels. Namely, the stations Benmore: Younger Botanic Gardens and Ardtalnaig seem to display a trend of seasonality where there is more precipitation in the winter months than the summer months. This is most pronounced for the Benmore station where the largest difference between peak and trough precipitation is roughly 5mm difference between the winter and summer. This difference is again present in the third plot which shows the seasonal average precipitation for each station. We can again see a large seasonal difference in rainfall in the two stations with the highest average recorded precipitation. However, for the stations with much lower average precipitation, there is no discernible seasonality in precipitation. The precipitation appears to be much the same in both summer and winter. Most stations only have up to 1mm in difference, typically below 0.5mm difference, but the two aforementioned stations with higher average precipitation levels have more than 3mm or 2mm difference in the average precipitation between the seasons. This is made clear in the final plot which shows the difference in seasonal average temperatures. Here though, we could only argue that there is little or no sign of seasonality for two of the stations: Edinburgh's Younger Botanic Gardens and Leuchars.

Thus to summarise, there is an argument to made for seasonality in precipitation levels for all but two of

the stations. The stations with lower levels of average precipitation have more persistent and less seasonal precipitation levels, whereas the stations with higher average levels of precipitation appear to have a seasonal difference, with more precipitation measured in the winter than the summer.

To test the seasonality of precipitation, we perform a Monte carlo test for the hypotheses:

H_0 : The precipitation distribution is the same in winter as in summer

H_1 : The winter and summer precipitation distributions have different expected values

We use the following test statistic $T = |\text{winter average} - \text{summer average}|$. This test is performed for each station and we calculate the estimated p-value, Monte Carlo standard deviation, and the confidence interval for the estimated p-value. For details see the Analysis code in the Code appendix. The results are shown in the following table:

	p_value	MC_SD	CI_lwr	CI_upr
BRAEMAR	0.0000000	0.0000000	0.0000000	0.0036821
BALMORAL	0.0000000	0.0000000	0.0000000	0.0036821
ARDTALNAIG	0.0000000	0.0000000	0.0000000	0.0036821
FASKALLY	0.0000000	0.0000000	0.0000000	0.0036821
LEUCHARS	0.0350350	0.0058144	0.0236390	0.0464311
PENICUIK	0.0000000	0.0000000	0.0000000	0.0036821
EDINBURGH: ROYAL BOTANIC GARDE	0.6546547	0.0150360	0.6251846	0.6841247
BENMORE: YOUNGER BOTANIC GARDE	0.0000000	0.0000000	0.0000000	0.0036821

From the table above, we reject the null hypothesis that the precipitation has the same distribution in the winter and summer at the 1% significance level for all but two of the stations. We reject the null hypothesis at seven of the stations at the 5% significance level. We conclude this due to the fact that the estimated p-values, and their confidence intervals, are less than the respective significance levels (0.01 and 0.05). This gives us strong evidence against the hypothesis of no seasonality in the precipitation levels for seven of the eight stations in Scotland. As a result, we would accept the alternative hypothesis that the winter and summer precipitation distributions have different expected values.

There is, however, one station with quite a different conclusion: the Edinburgh Royal Botanic Gardens station has an estimated p-value of 0.65 (2 d.p.). This means that we fail to reject the null hypothesis at any conventional significance level. That is to say, there is insufficient evidence against the equality of the distribution of precipitation levels in summer and winter for the Edinburgh station. This result is hardly surprising though, given how close the average levels of precipitation in winter and summer are for the Edinburgh station. The difference is roughly 0.02 (2 d.p.) (recall the final precipitation plot above) which indicates that the distribution of precipitation will roughly have the same average value in winter and summer.

Spatial weather prediction

In this second part of the report, we are interested in estimating models used for spatial prediction of precipitation in Scotland. To do so, we use data from 8 stations around Scotland to model the square root of the monthly averaged precipitation values on a set of covariates relating to the stations spatial coordinates, elevation, and time of year the recordings were made. In order to better capture the effect of seasonality, discussed more in the first part of the report, we augment the models sequentially with suitable cos and sin functions. The reason for modelling the square root of the monthly average precipitation is that the precipitation values are very skewed, with variance increasing with the mean value. Using the square root of the monthly averages helps alleviate that issue, making a constant-variance model more plausible.

Estimation and prediction

In total, there are 5 models we estimate:

$$M0 : \sqrt{\text{Monthly average precipitation}} \sim \text{Intercept} + \text{Longitude} + \text{Latitude} + \text{Elevation} + \text{Monthly Average DecYear}$$

$$M1 : \sqrt{\text{Monthly average precipitation}} \sim \text{Intercept} + \text{Longitude} + \text{Latitude} + \text{Elevation} + \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M2 : \sqrt{\text{Monthly average precipitation}} \sim \text{Intercept} + \text{Longitude} + \text{Latitude} + \text{Elevation} + \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M3 : \sqrt{\text{Monthly average precipitation}} \sim \text{Intercept} + \text{Longitude} + \text{Latitude} + \text{Elevation} + \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M4 : \sqrt{\text{Monthly average precipitation}} \sim \text{Intercept} + \text{Longitude} + \text{Latitude} + \text{Elevation} + \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

The dependent variable in these models, as outlined above, is the square root of the monthly average precipitation. This allows us to model and hence make predictions of the average precipitation level in a month. The covariates of Longitude, Latitude, and Elevation relate the spatial location and coordinates of the stations the dependent variable was measured from. We include the monthly averaged decimal year variable as a covariate to account for the time of year the observations were made, also measured in the same time frame as the dependent variable: one month. We then sequentially augment the models with sin and cos functions to better account for seasonality in the precipitation levels.

If we use the entirety of the data to fit each of these models, we can obtain estimates of the coefficients of each covariate. This is performed in the Analysis code element of the Code appendix. To three decimal places, these models are estimated as:

$$M0 : \sqrt{\text{Monthly average precipitation}} \sim 2.477 - 0.546 \times \text{Longitude} - 0.136 \times \text{Latitude} + 0.000 \times \text{Elevation} + 0.002 \times \text{Monthly Average DecYear}$$

$$M1 : \sqrt{\text{Monthly average precipitation}} \sim 2.490 - 0.547 \times \text{Longitude} - 0.135 \times \text{Latitude} + 0.000 \times \text{Elevation} + 0.002 \times \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M2 : \sqrt{\text{Monthly average precipitation}} \sim 2.477 - 0.547 \times \text{Longitude} - 0.135 \times \text{Latitude} + 0.000 \times \text{Elevation} + 0.002 \times \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M3 : \sqrt{\text{Monthly average precipitation}} \sim 2.470 - 0.547 \times \text{Longitude} - 0.135 \times \text{Latitude} + 0.000 \times \text{Elevation} + 0.002 \times \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

$$M4 : \sqrt{\text{Monthly average precipitation}} \sim 2.471 - 0.547 \times \text{Longitude} - 0.135 \times \text{Latitude} + 0.000 \times \text{Elevation} + 0.002 \times \text{Monthly Average DecYear} + \gamma_{c,1} \cos(2\pi t)$$

We can see that coefficients on our estimated models are very close to one another. This is to be expected given the similarity of the models and the size of the data set that they are fitted on.

Observing these estimated models allows us to gain some interesting insights into the precipitation in Scotland. We can see that the estimated coefficient on Elevation is 0 in all models. This implies that the elevation of the stations alone is not estimated to have a significant impact on the dependent variable, the square root of monthly average precipitation. In other words, changes in the elevation do not result in significant changes in precipitation levels after controlling for the effects of the other covariates in the model. Rather surprisingly, the models all have a negative coefficient on the Latitude. This is interesting as it suggests that if we increase the latitude, i.e. move farther North, we predict that a lower level of precipitation, *ceteris paribus*. This is counter to what one might have anticipated. The coefficient on longitude being negative is more aligned with what anyone with experience of Scottish weather might have anticipated: if we increase the longitude, that is move eastward, we predict less precipitation. It makes sense for this to be the case as the West coast of Scotland is typically much wetter than the East coast.

Note that it is much more challenging to provide a meaningful or interesting interpretation for the coefficients of the sin and cos covariates, hence the lack of discussion regarding them.

Assessment: Station and season differences

The reason for estimating these 5 models in the first place is because we want to be able to predict precipitation levels in new locations across Scotland. It is thus of vital importance to understand and measure how well our models predict precipitation at unobserved locations. Moreover, due to the seasonality of precipitation discussed in the previous section, it is of interest to investigate how well our models predict precipitation throughout the year.

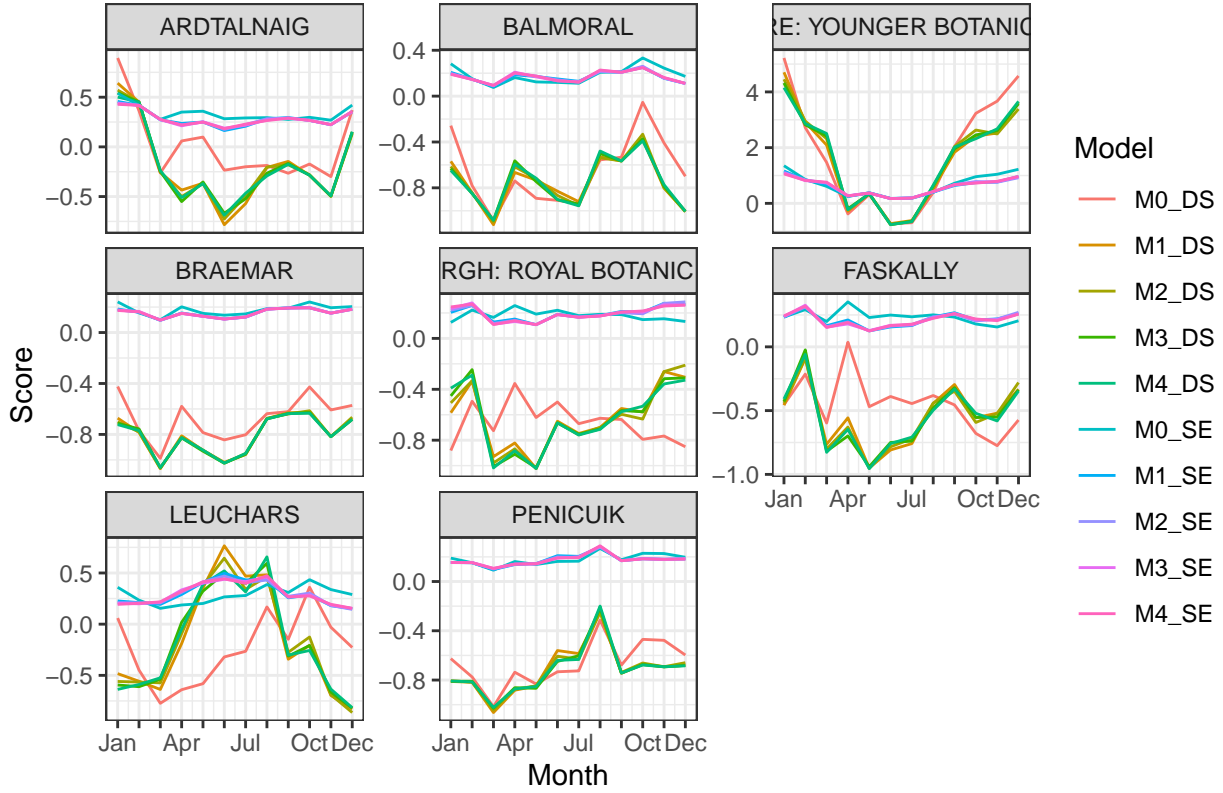
To undertake this investigation, we perform a stratified cross-validation that groups the data by weather station and computes some prediction scores for each station, as well as the overall cross-validated average scores, aggregated to the 12 months of the year. For the prediction scores, we use the Squared Error and Dawid-Sebastiani scores.

In practice, this stratified cross-validation involves training (or fitting) each of the 5 models on a subset of the data corresponding to using 7 out of 8 weather stations. We then test each of the 5 models model on the remaining data (i.e., the one weather station that was not included in the training set). This process is repeated multiple times, using a different weather station for testing each time, to obtain an estimate of the model's performance on new, unseen data. This ensures that each model is evaluated on its ability to predict the precipitation levels of new locations.

For the technical details of how this was performed, please see the Code appendix, Analysis code section. We now present the findings of this stratified cross-validation, answering our two main questions in turn:

- are the models equally good at predicting the different stations precipitation levels?
- is the prediction accuracy the same throughout the year?

Model Comparison by Station



In the above figure, we can see one plot for each station corresponding to the case when that station was used as the validation station in the stratified cross-validation. It shows the Squared Error (SE) and Dawid-Sebastiani (DS) scores for each of the 5 models (M0, M1, etc.). It is noted that these are negatively oriented scores, which means that a lower score indicates a better predictive performance. This figure thus allows us to compare the models predictive performance for each omitted station and hence help to answer our first question: are the models equally good at predicting the different stations precipitation levels?

We can see that the SE scores are similar for all of the models in each station. This does not allow for many interesting insights. However, the station with the highest SE scores for each of its models is the case when Benmore: Younger Botanic Gardens is the validation station. This is seen by the fact that most of the SE scores are no bigger than 0.5 for the other validation stations, but the SE scores for each of the models is much higher, around 1 in the winter months, in the case when Benmore is used as the validation station. This indicates that the models are not equally good at predicting different stations precipitation levels.

Moreover, when we observe the DS scores from the plot, the inequality in predictive performance for these models in different stations is clearer to see. One such example is again in the case where the models poorly predict the precipitation for the Benmore station. The DS scores for most other stations are at the very least negative throughout the year or if they do reach a positive score, they are below 1. However the DS scores for the Benmore station are much higher, and are even around values of 4 in the winter months. This indicates a very poor predictive performance when we train the models on the other 7 stations and attempt to predict precipitation for Benmore. One reason for this poor predictive performance could be the discrepancy between the distribution of precipitation at the Benmore station and the other 7 stations. This was noted in the preliminary investigation of precipitation for each of the stations where it was noted that Benmore exhibited more seasonality and also had higher average precipitation levels than the other stations.

Another such example is for the Leuchars station. Here we can see an inverted pattern from that of the Benmore station. In the summer months, the DS scores are much higher in the case where Leuchars is the validation station compared to all of the other stations. This again indicates poor predictive performance

from the models in predicting precipitation in Leuchars, particularly in the summer.

From the above arguments, we can conclude that the models are not equally as good at predicting precipitation for each station. This is most easily seen by the discrepancy in the predictive scores for each model when comparing which station is used as the validation station.

In the analysis above, there were hints of differing predictive performance of the models throughout the year. We next move on to investigate this more in an attempt to answer our second question: is the prediction accuracy the same throughout the year?



The above figure shows two plots displaying the average DS and SE scores, respectively, for each model across all of the stratified cross-validation folds. Each score is aggregated to the twelve months of the year. This plot reveals a clear trend in the scores: the scores are much lower in the summer months than the winter months. This is the case for both scores as they follow a very similar trend. This clearly shows that the models are not equally good at predicting precipitation throughout the year. Instead, the models have much better predictive accuracy in summer months compared to the winter months, as indicated by the lower scores in the summer months.

Code appendix

Function definitions

```
# Conor Cassidy (s2077927)
# Test statistic
```

```

#'
#' Calculates the test statistic of the absolute difference of precipitation in the summer and winter.
#'
#' @param data data frame; description.
#' @param perm vector; if it is null there is no permutation, otherwise it corresponds to the permutati

test_stat <- function(data, perm = NULL) {
  prcp <- data$Value
  season <- data$Season

  if (!is.null(perm)) {
    prcp <- c(prcp[perm], prcp[-perm])
  }

  winter_avg <- mean(prcp[season == "Winter"])
  summer_avg <- mean(prcp[season == "Summer"])
  return(abs(winter_avg - summer_avg))
}

#' p-value confidence interval
#'
#' A function that finds the Monte Carlo standard deviation and confidence interval for estimated p-val
#'
#' @param phat vector; the estimated p-values from the Monte-Carlo permutation test.
#' @param N numeric; The number of permutations
#' @param alpha numeric; significance level with default of 0.05.

p_value_CI <- function(phat, N, alpha = 0.05) {
  z_alpha_2 <- qnorm(1 - alpha / 2)
  CI_lwr <- ifelse(phat > 0, phat - z_alpha_2 * sqrt((phat * (1 - phat)) / N), ifelse(phat == 0, 0, NA))
  CI_upr <- ifelse(phat > 0, phat + z_alpha_2 * sqrt((phat * (1 - phat)) / N), ifelse(phat == 0, 1 - 0.
  MC_SD <- sqrt((phat * (1 - phat) / N))
  return(list(CI_lwr = CI_lwr, CI_upr = CI_upr, MC_SD = MC_SD))
}

#' p-value Monte-Carlo permutations
#'
#' Performs and returns the estimated p-values, standard deviations, and confidence interval after perf
#' Monte Carlo permutation test on the null hypothesis that the distribution of precipitation is the sa
#' winter and summer, for each station.
#'
#'
#' @param data data frame; the data to be used in the permutation test
#' @param N numeric; The number of permutations

p_val <- function(data, N) {
  n_summer <- data %>%
    filter(Summer == TRUE) %>%
    count(Summer) %>%
    pull(n)
  n_winter <- data %>%

```

```

    filter(Summer == FALSE) %>%
    count(Summer) %>%
    pull(n)

summer <- data %>%
  filter(Summer == TRUE)
summer_avg <- mean(summer$Value)

winter <- data %>%
  filter(Summer == FALSE)
winter_avg <- mean(winter$Value)

t_stat <- abs(winter_avg - summer_avg)

t_null <- numeric(N)
for (i in seq_len(N)) {
  if (i == 1) {
    # permutation corresponding to the observed case
    xindex <- 1:n_winter
  } else {
    # random permutations under H0
    xindex <- sample(seq_len(n_winter + n_summer), size = n_winter, replace = FALSE)
  }
  t_null[i] <- test_stat(
    data = data,
    perm = xindex
  )
}
t_stat <- t_null[1]
phat <- (mean(t_null[-1] >= t_stat))

p_CI <- p_value_CI(phat, N)

return(list(p_value = phat, MC_SD = p_CI$MC_SD, CI_lwr = p_CI$CI_lwr, CI_upr = p_CI$CI_upr))
}

#' Estimating models
#'
#' Defines and fits each of the 5 models defined in the report given any input data.
#'
#' @param data data frame; contains the training data that should be used to estimate and fit the model.

estimate_models <- function(data) {
  # Model M0
  predictor_M0 <- formula(Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear_avg)
  model_M0 <- lm(predictor_M0, data = data)

  # Model M1
  predictor_M1 <- formula(Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear_avg +
    cos(2 * pi * DecYear_avg) + sin(2 * pi * DecYear_avg))
  model_M1 <- lm(predictor_M1, data = data)

```

```

# Model M2
predictor_M2 <- formula(Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear_avg +
  cos(2 * pi * DecYear_avg) + sin(2 * pi * DecYear_avg) + cos(4 * pi * DecYear_avg) + sin(4 * pi * DecYear_avg))
model_M2 <- lm(predictor_M2, data = data)

# Model M3
predictor_M3 <- formula(Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear_avg +
  cos(2 * pi * DecYear_avg) + sin(2 * pi * DecYear_avg) + cos(4 * pi * DecYear_avg) + sin(4 * pi * DecYear_avg))
model_M3 <- lm(predictor_M3, data = data)

# Model M4
predictor_M4 <- formula(Value_sqrt_avg ~ Longitude + Latitude + Elevation + DecYear_avg +
  cos(2 * pi * DecYear_avg) + sin(2 * pi * DecYear_avg) + cos(4 * pi * DecYear_avg) + sin(4 * pi * DecYear_avg))
model_M4 <- lm(predictor_M4, data = data)

# return a list where each element contains information on the fit of the model.
return(list(M0 = model_M0, M1 = model_M1, M2 = model_M2, M3 = model_M3, M4 = model_M4))
}

#' Training and testing splits
#'
#' Creates two subsets of a provided data frame that will be stored in a list: one is the data frame containing
#' the data the model should be fitted and estimated from and the other is a data frame containing the data
#' that should be used to evaluate predictive performance.
#'
#' @param data data frame; contains the all of the data that should be split.
#' @param test_name character; the name of the station that is to be the validation station (i.e. to be
#' the testing data).

train_test_data <- function(data, test_name) {
  train_data <- data %>%
    filter(data$Name != `test_name`)

  test_data <- data %>%
    filter(data$Name == `test_name`)

  return(list(train_data = train_data, test_data = test_data))
}

```

Analysis code

The following code shows how we perform the Monte Carlo permutation, as referred to in the first section of the report.

```

# load the data relevant for the precipitation.
prcp_data <- subset(ghcnd, Element == "PRCP")

# add a logical variable indicating if the season is summer
prcp_data$Summer <- prcp_data$Season == "Summer"

# initialise a vector for the results of the permutation test
results <- list()

```

```

# loop for each station to be used as the
for (name in unique(prcp_data$Name)) {

  # filter the data to be used - only use the relevant station
  test <- prcp_data %>%
    filter(Name == name)

  # apply the p-val function from the functions.R file to perform and returns the estimated p-values, s
  p_val_result <- p_val(test, N = 1000)
  results[[name]] <- p_val_result
}

# display the results in a table
results_df <- do.call(rbind, lapply(results, as.data.frame))
knitr::kable(results_df)

```

Now we move onto the second half of the project focusing on spatial weather prediction. We first deal with the data that we will use, and then estimate the models defined in the report using this data. The coefficients of each model is finally displayed in a table.

```

# get the relevant data for the precipitation
prcp_data <- subset(ghcnd, Element == "PRCP")

# edit this data to make it monthly and have the required variables - similar to before
prcp_monthly <- prcp_data %>%
  group_by(Name, Year, Month) %>%
  summarise(avg_prcp = mean(Value), DecYear_avg = mean(DecYear), .groups = "drop") %>%
  mutate(Value_sqrt_avg = sqrt(avg_prcp))

prcp_data_summarized <- prcp_data %>%
  group_by(Name, Year, Month) %>%
  summarise(Latitude = first(Latitude),
            Longitude = first(Longitude),
            Elevation = first(Elevation), .groups = "drop") %>%
  ungroup()

prcp_monthly_models <- prcp_monthly %>%
  left_join(prcp_data_summarized, by = c("Name", "Year", "Month"))

# get the coefficients for each model
M0_coeffs <- round(estimate_models(prcp_monthly_models)$M0$coefficients, 3)
M1_coeffs <- round(estimate_models(prcp_monthly_models)$M1$coefficients, 3)
M2_coeffs <- round(estimate_models(prcp_monthly_models)$M2$coefficients, 3)
M3_coeffs <- round(estimate_models(prcp_monthly_models)$M3$coefficients, 3)
M4_coeffs <- round(estimate_models(prcp_monthly_models)$M4$coefficients, 3)

# this is used to add NAs to the empty elements of the shorter vectors so they are all the same length
max_len <- max(length(M0_coeffs), length(M1_coeffs), length(M2_coeffs), length(M3_coeffs), length(M4_coeffs))

M0_coeffs <- c(M0_coeffs, rep(NA, max_len - length(M0_coeffs)))
M1_coeffs <- c(M1_coeffs, rep(NA, max_len - length(M1_coeffs)))
M2_coeffs <- c(M2_coeffs, rep(NA, max_len - length(M2_coeffs)))
M3_coeffs <- c(M3_coeffs, rep(NA, max_len - length(M3_coeffs)))

```



```

M4_coefs <- c(M4_coefs, rep(NA, max_len - length(M4_coefs)))

# make one data frame with all the coefficients for the models
model_coefs <- data.frame(M0_coefs, M1_coefs, M2_coefs, M3_coefs, M4_coefs)

# display the result in a table
knitr::kable(model_coefs)

```

	M0_coefs	M1_coefs	M2_coefs	M3_coefs	M4_coefs
(Intercept)	2.477	2.490	2.477	2.470	2.471
Longitude	-0.546	-0.547	-0.547	-0.547	-0.547
Latitude	-0.136	-0.135	-0.135	-0.135	-0.135
Elevation	0.000	0.000	0.000	0.000	0.000
DecYear_avg	0.002	0.002	0.002	0.002	0.002
cos(2 * pi * DecYear_avg)	NA	0.183	0.183	0.183	0.183
sin(2 * pi * DecYear_avg)	NA	-0.132	-0.132	-0.132	-0.132
cos(4 * pi * DecYear_avg)	NA	NA	0.031	0.031	0.031
sin(4 * pi * DecYear_avg)	NA	NA	0.001	0.001	0.001
cos(6 * pi * DecYear_avg)	NA	NA	NA	-0.007	-0.007
sin(6 * pi * DecYear_avg)	NA	NA	NA	0.029	0.028
cos(8 * pi * DecYear_avg)	NA	NA	NA	NA	0.014
sin(8 * pi * DecYear_avg)	NA	NA	NA	NA	0.012

Now in the assessment of the models, we perform the stratified cross-validation as outlined in the report.

```

# get a list of all the station names
station_names <- unique(prcp_monthly_merged$Name)

# create an empty list to store all of the estimated models & scores for each fold of the stratified cr
all_fold_fit_list <- list()

# perform a loop to use each station as the validation station once
for(name in station_names) {
  # get training and testing data
  train <- train_test_data(prcp_monthly_merged, test_name = `name`)$train_data
  test <- train_test_data(prcp_monthly_merged, test_name = `name`)$test_data

  # estimate the 5 models on the training data
  models <- estimate_models(train)

  # create a list to store the predictions for each model
  pred_list <- vector("list", length = length(models))

  # now loop over each model and predict for the remaining validation station
  for(modID in seq_along(models)) {
    pred <- predict.lm(models[[modID]], newdata = test, se.fit = TRUE)
    pred_list[[modID]] <- pred

    # we now want to compute the scores for these predictions and append them to the model information st
    resid_var_est <- sum((models[[modID]]$residuals)^2) / models[[modID]]$df.residual
    sd <- sqrt((pred_list[[modID]]$se.fit)^2 + resid_var_est)
  }
}

```

```

se <- proper_score("se", obs = test$Value_sqrt_avg, mean = pred_list[[modID]]$fit)
ds <- proper_score("ds", obs = test$Value_sqrt_avg, mean = pred_list[[modID]]$fit, sd = sd)

pred_list[[modID]]$se <- se
pred_list[[modID]]$ds <- ds
}

all_fold_fit_list[[name]] <- pred_list
}

# we now initialise some vectors to store each models scores (for each fold)
M0_SE <- c()
M1_SE <- c()
M2_SE <- c()
M3_SE <- c()
M4_SE <- c()

M0_DS <- c()
M1_DS <- c()
M2_DS <- c()
M3_DS <- c()
M4_DS <- c()

# now we put the scores into their respective vectors
for (name in station_names) {
  M0_SE <- append(M0_SE, all_fold_fit_list[[name]][[1]]$se, after = length(M0_SE))
  M1_SE <- append(M1_SE, all_fold_fit_list[[name]][[2]]$se, after = length(M1_SE))
  M2_SE <- append(M2_SE, all_fold_fit_list[[name]][[3]]$se, after = length(M2_SE))
  M3_SE <- append(M3_SE, all_fold_fit_list[[name]][[4]]$se, after = length(M3_SE))
  M4_SE <- append(M4_SE, all_fold_fit_list[[name]][[5]]$se, after = length(M4_SE))

  M0_DS <- append(M0_DS, all_fold_fit_list[[name]][[1]]$ds, after = length(M0_DS))
  M1_DS <- append(M1_DS, all_fold_fit_list[[name]][[2]]$ds, after = length(M1_DS))
  M2_DS <- append(M2_DS, all_fold_fit_list[[name]][[3]]$ds, after = length(M2_DS))
  M3_DS <- append(M3_DS, all_fold_fit_list[[name]][[4]]$ds, after = length(M3_DS))
  M4_DS <- append(M4_DS, all_fold_fit_list[[name]][[5]]$ds, after = length(M4_DS))
}

# append the data frame with these scores
prcp_monthly_merged <- prcp_monthly_merged %>%
  ungroup() %>%
  mutate(M0_SE = M0_SE, M1_SE = M1_SE, M2_SE = M2_SE, M3_SE = M3_SE, M4_SE = M4_SE, M0_DS = M0_DS, M1_DS = M1_DS, M2_DS = M2_DS, M3_DS = M3_DS, M4_DS = M4_DS)

# now summarise these scores into a data frame of the monthly averages for each score/model
prcp_month_sum <- prcp_monthly_merged %>%
  group_by(Month) %>%
  summarise(M0_SE = mean(M0_SE), M1_SE = mean(M1_SE), M2_SE = mean(M2_SE), M3_SE = mean(M3_SE), M4_SE = mean(M4_SE), M0_DS = mean(M0_DS), M1_DS = mean(M1_DS), M2_DS = mean(M2_DS), M3_DS = mean(M3_DS), M4_DS = mean(M4_DS))

```

Finally, in order to plot the results of this stratified cross-validation we manipulate the data into forms that are easier to work with.

```
# summarise the data in a way suitable for making station comparisons
```

```
compare_station_scores <- prcp_monthly_merged %>%
```

```
  group_by(Name, Month) %>%
```

```
  summarise(M0_SE = mean(M0_SE), M1_SE = mean(M1_SE), M2_SE = mean(M2_SE), M3_SE = mean(M3_SE), M4_SE =
```

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

```
## '.groups' argument.
```

```
# Melt the data frame to make it easier to plot
```

```
compare_station_scores <- melt(compare_station_scores, id.vars = c("Name", "Month"),
```

```
  measure.vars = c("M0_DS", "M1_DS", "M2_DS", "M3_DS", "M4_DS",  
                    "M0_SE", "M1_SE", "M2_SE", "M3_SE", "M4_SE"),
```

```
  variable.name = "Model", value.name = "Score")
```

```
# Melt the data frame in a different way to make it easier to plot the year round performance of each s
```

```
prcp_month_sum_melted <- melt(prcp_month_sum, id.vars = "Month",
```

```
  measure.vars = c("M0_DS", "M1_DS", "M2_DS", "M3_DS", "M4_DS",  
                    "M0_SE", "M1_SE", "M2_SE", "M3_SE", "M4_SE"),
```

```
  variable.name = "Model", value.name = "Score")
```