Assignment 3

Ilaria Crippa

2024-01-24

Contents

1 Assignment 3

1.1 Multiple linear regression model

1.1.1 Graphical representation of the relationships among the variables

We now want to consider multiple variables and, in particular, at least one categorical variable. The dataset includes the categorical variable "station", which indicates the number of automatic weather station from the grid of the LDAPS (Local Data Assimilation and Prediction System) model for each datum. This variable has 25 levels.

```r
data$station <- factor(data$station)
levels(data$station)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25"
```

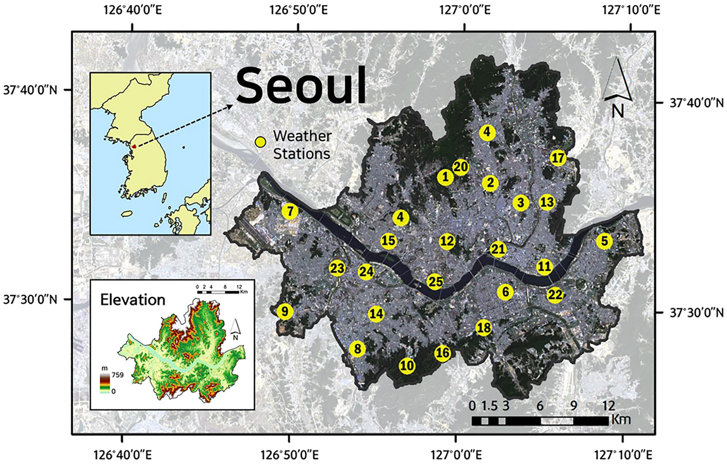


Figure 1: Location of automatic weather stations

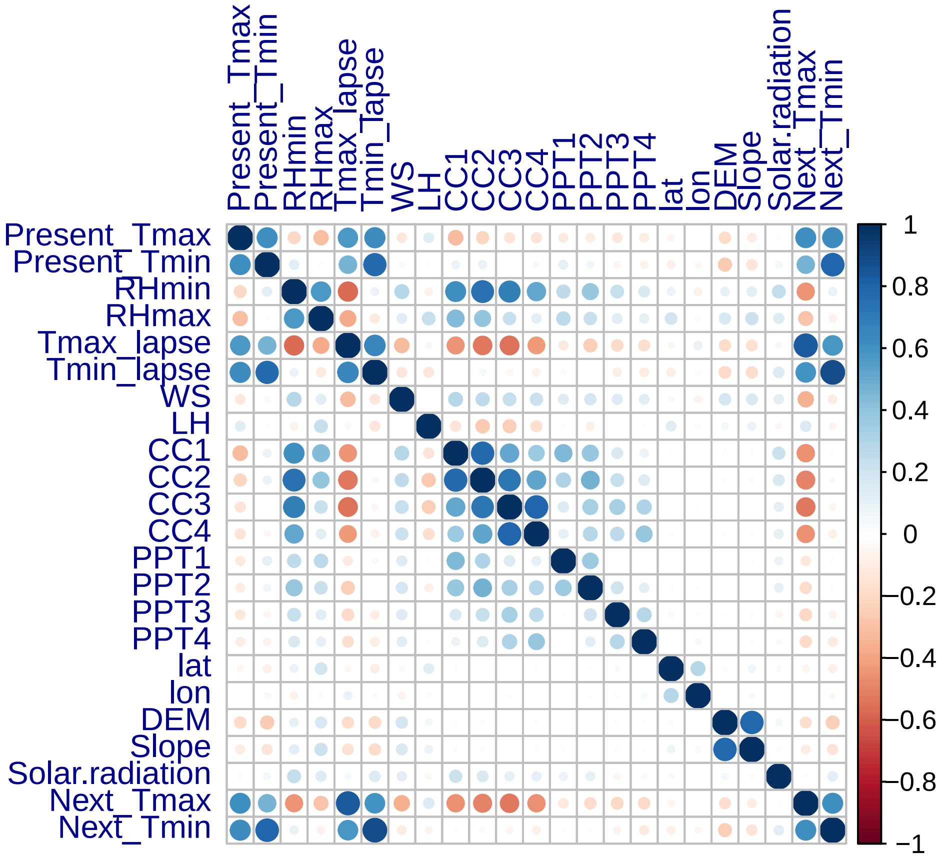
The correlation matrix among the continuous variables of the dataset shows that the variables with the highest correlation with the next day's maximum and minimum air temperatures are the next-day forecasts of the LDAPS climate model for maximum and minimum relative humidity, maximum and minimum air temperatures, wind speed, average cloud coverage and average precipitation levels. The red dots indicate a negative relationship, while the blue dots a positive linear relationship among the variables. We also note a quite strong inter-correlation between the predictors of relative humidity, cloud coverage and precipitation, which could lead to the rise of inter-collinearity issues in our model.

```
selected_variables <- c("Present_Tmax", "Present_Tmin", "RHmin", "RHmax", "Tmax_lapse", "Tmin_lapse", "
selected_data <- data[, selected_variables]
correlation_matrix <- cor(selected_data)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## corrplot 0.92 loaded
```

```
corrplot(correlation_matrix[, 1:23], tl.col = 'darkblue')
```

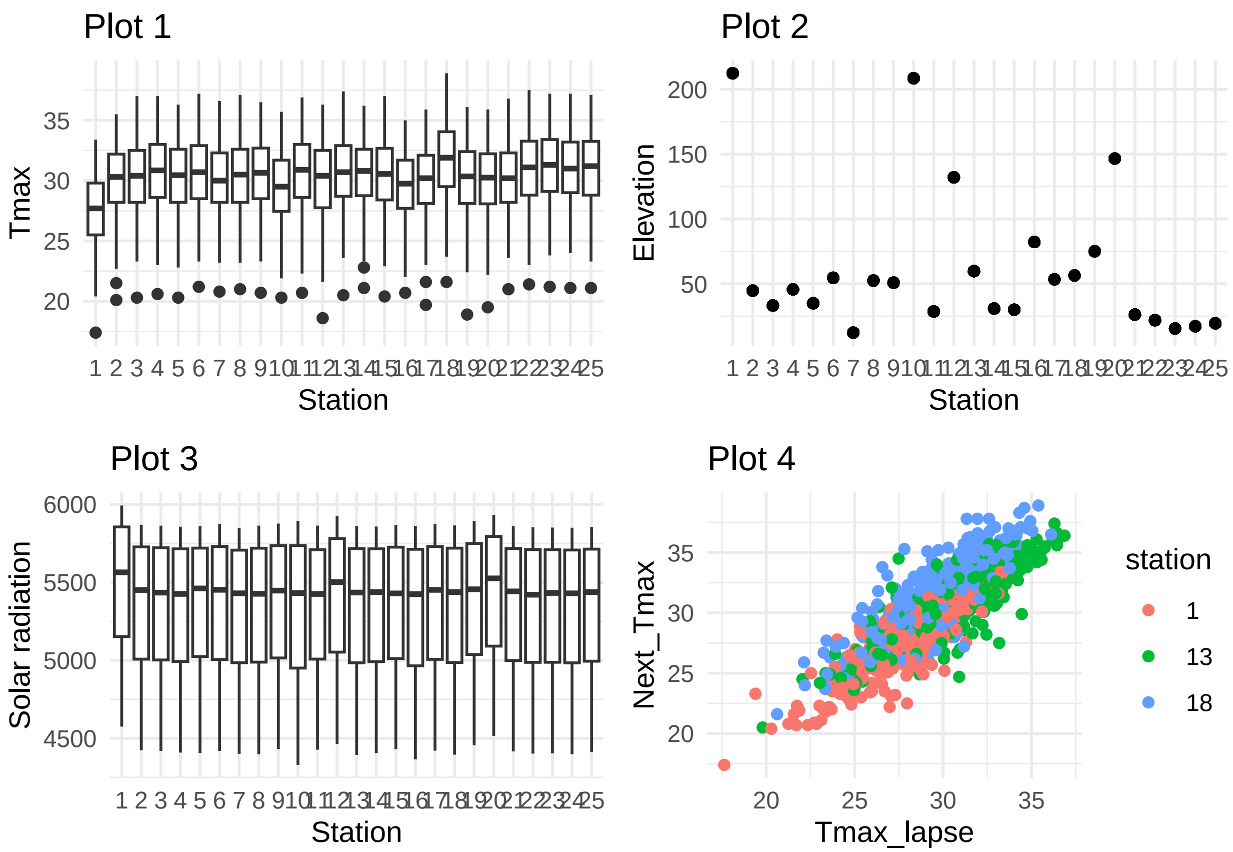


The following plots show the relationship between the factor "station" and other variables: "Plot 1" is a boxplot displaying the distribution of the next-day observed maximum temperature for every station. We can notice that the median of the maximum temperature for stations 1 and 18 does not overlap with the other box plots, as well as some inconsistency in station 10 (the median is lower than 30°C), as well as in stations 11 and from 22 to 25 (whose medians are a bit higher). We suspect that the discrepancies could be related to the elevation of these stations (see plot 2); this observation is valid in particular for stations 1 and 18 (high elevation) and stations 22, 23, 24 and 25 (low elevaation and close to the Han River). Plot 3, instead, shows that the distribution of solar radiation is fairly consistent. Plot 4 shows the correlation between the next-day maximum temperature forecasts and the observed next-day maximum temperatures for station 1 (whose values tend to be lower than the median of the observed maximum temperatures of 30.4°C), station 13 (whose values are close to the median) and station 18 (whose values are higher than the observed median).

```r
library(ggplot2)
library(gridExtra)
library(grid)

data$station <- as.character(data$station)
data$station <- factor(data$station, levels = as.character(1:25))
plot1 = ggplot(data, aes(x = station, y = Next_Tmax)) +
  geom_boxplot() +
  labs(title = "Plot 1", x = "Station", y = "Tmax") +
  theme_minimal()

plot2 = ggplot(data, aes(x = station, y = DEM)) +
  geom_point() +
  labs(title = "Plot 2", x = "Station", y = "Elevation") +
  theme_minimal()

plot3 = ggplot(data, aes(x = station, y = Solar.radiation)) +
  geom_boxplot() +
  labs(title = "Plot 3", x = "Station", y = "Solar radiation") +
  theme_minimal()

library(dplyr)
```

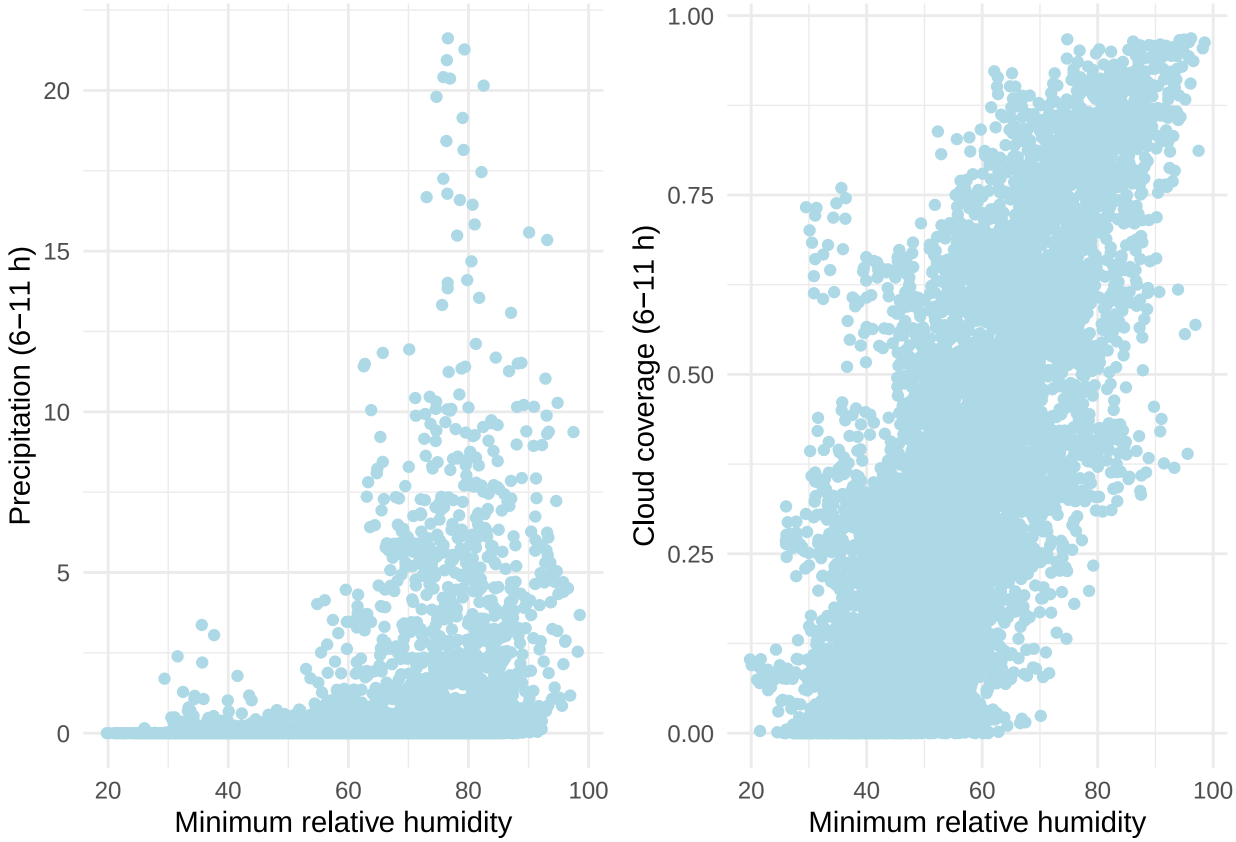
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
selected_levels <- c("1", "13", "18")
filtered_data <- data %>%
  filter(station %in% selected_levels)
plot4 = ggplot(filtered_data, aes(x = Tmax_lapse, y = Next_Tmax, color = station)) +
  geom_point() +
  labs(title = "Plot 4", x = "Tmax_lapse", y = "Next_Tmax", color = "station") +
  theme_minimal()

grid.arrange(plot1, plot2, plot3, plot4, ncol = 2, nrow=2)
```



Finally, we note the correlation between the forecasted minimum relative humidity and the forecasted precipitations and cloud coverage (between 6:00 and 11:00).

```
plot5 = ggplot(data, aes(x = RHmin, y = PPT2)) +
  geom_point(color = "lightblue") +
  labs(x = "Minimum relative humidity", y = "Precipitation (6-11 h)") +
  theme_minimal()

plot6 = ggplot(data, aes(x = RHmin, y = CC2)) +
  geom_point(color = "lightblue") +
  labs(x = "Minimum relative humidity", y = "Cloud coverage (6-11 h)") +
  theme_minimal()

grid.arrange(plot5, plot6, ncol = 2, nrow=1)
```



1.1.2 Multiple linear regression with all the predictors and an interaction term

In order to fit a multiple linear model with an interaction term with the variable “station”, we decrease the number of levels of the variable, by grouping the stations based on the elevation of their location: the four groups that are created - “Very low DEM station”, “Low DEM station”, “Medium DEM station” and “High DEM station” - correspond to an elevation between 0 and 25 meters, 25 and 50, 50 and 100 and 100 to 250 meters respectively.

```
breaks = c(-Inf, 25, 50, 100, 250)
labels <- c("Very low DEM station", "Low DEM station", "Medium DEM station", "High DEM station")
data$grouped_station <- cut(data$DEM, breaks = breaks, labels = labels)
levels(data$grouped_station)
```

```
## [1] "Very low DEM station" "Low DEM station"      "Medium DEM station"
## [4] "High DEM station"
```

To curb the negative effects of collinearity issues, we calculate the daily average values of cloud coverage and precipitation. Finally, we fit the model.

```
data$CC <- rowMeans(data[, c("CC1", "CC2", "CC3", "CC4")], na.rm = TRUE)
data$PPT = rowMeans(data[, c("PPT1", "PPT2", "PPT3", "PPT4")], na.rm = TRUE)

model0.2 = lm(Next_Tmax ~ Tmax_lapse + Tmin_lapse + RHmin + RHmax + CC + PPT + WS * grouped_station + L
summary(model0.2)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ Tmax_lapse + Tmin_lapse + RHmin + RHmax +
##     CC + PPT + WS * grouped_station + LH, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8957 -0.8674  0.0168  0.9148  6.1746
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          8.5096371  0.4137763  20.566  < 2e-16 ***
## Tmax_lapse                           0.6056927  0.0154524  39.197  < 2e-16 ***
## Tmin_lapse                           0.2645524  0.0159522  16.584  < 2e-16 ***
## RHmin                                0.0245350  0.0027828   8.817  < 2e-16 ***
## RHmax                               -0.0185384  0.0033422  -5.547 3.01e-08 ***
## CC                                  -4.0787022  0.1506390 -27.076  < 2e-16 ***
## PPT                                  0.1462919  0.0229188   6.383 1.84e-10 ***
## WS                                  -0.1755296  0.0199447  -8.801  < 2e-16 ***
## grouped_stationLow DEM station      -0.8087025  0.1813145  -4.460 8.31e-06 ***
## grouped_stationMedium DEM station   -0.2355638  0.1796308  -1.311  0.18977
## grouped_stationHigh DEM station     -0.7907901  0.1984608  -3.985 6.82e-05 ***
## LH                                   0.0092898  0.0006064  15.320  < 2e-16 ***
## WS:grouped_stationLow DEM station    0.0582122  0.0250210   2.327  0.02002 *
## WS:grouped_stationMedium DEM station  0.0117671  0.0251488   0.468  0.63987
## WS:grouped_stationHigh DEM station   0.0718496  0.0255074   2.817  0.00486 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.525 on 7573 degrees of freedom
## Multiple R-squared:  0.7603, Adjusted R-squared:  0.7598
## F-statistic:  1716 on 14 and 7573 DF,  p-value: < 2.2e-16
```

1.1.3 Interpret all the parameters and their uncertainties, also with the support of a graphical representation

Intercept: the estimated value of the next-day maximum air temperature is equal to 8.51°C, when all predictor variables are zero.

Tmax_lapse (0.6057): for each one-degree increase in the forecast of next-day maximum air temperature, the predicted maximum air temperature to be observed increases by approximately 0.6057 degrees, assuming all other predictors are constant.

Tmin_lapse (0.2646): for each one-degree increase in the forecast of next-day minimum air temperature, the predicted maximum air temperature to be observed increases by approximately 0.2646 units, assuming all other predictors are constant.

RHmin (0.0245): for each percentage increase in forecasted minimum relative humidity, the predicted maximum air temperature increases by approximately 0.0245 degrees, assuming all other predictors are constant.

RHmax (-0.0185): for each percentage increase in maximum relative humidity, the predicted maximum air temperature decreases by approximately 0.0185 units, assuming all other predictors are constant.

CC (-4.0787): a percentage increase in cloud coverage is associated with a significant decrease in the predicted maximum air temperature, by approximately 4.0787 degrees, assuming all other predictors are constant.

PPT (0.1463): for each percentage increase in average precipitation, the predicted maximum air temperature increases by approximately 0.1463 units, assuming all other predictors are constant.

WS (-0.1755): for each meter-per-second increase in wind speed, the predicted maximum air temperature decreases by approximately 0.1755 units, assuming all other predictors are constant.

Low DEM station (-0.8087): compared to stations that are located at very low elevations (between 0 and 25 meters), the predicted maximum air temperature is lower by approximately 0.8087 degrees for stations categorized as “Low elevation” stations, assuming all other predictors are constant.

Medium DEM station (-0.2356): there is no statistically significant difference in the predicted maximum air temperature for stations categorized as “Medium elevation” stations compared to the ones in the “Very low elevation” category, assuming all other predictors are constant.

High DEM station (-0.7908): compared to stations that are located at very low elevations, the predicted maximum air temperature is lower by approximately 0.7908 degrees for stations categorized as “High elevation” station”, assuming all other predictors are constant.

LH (0.0093): for each one-unit increase in latent heat flux (measure in (W/m2)), the predicted maximum air temperature increases by approximately 0.0093 degrees, assuming all other predictors are constant.

WS:Low DEM station (0.0582): the interaction effect suggests that the impact of wind speed on maximum air temperature is higher for stations categorized as “Low elevation” stations compared to “very low elevation” stations, which is the reference category.

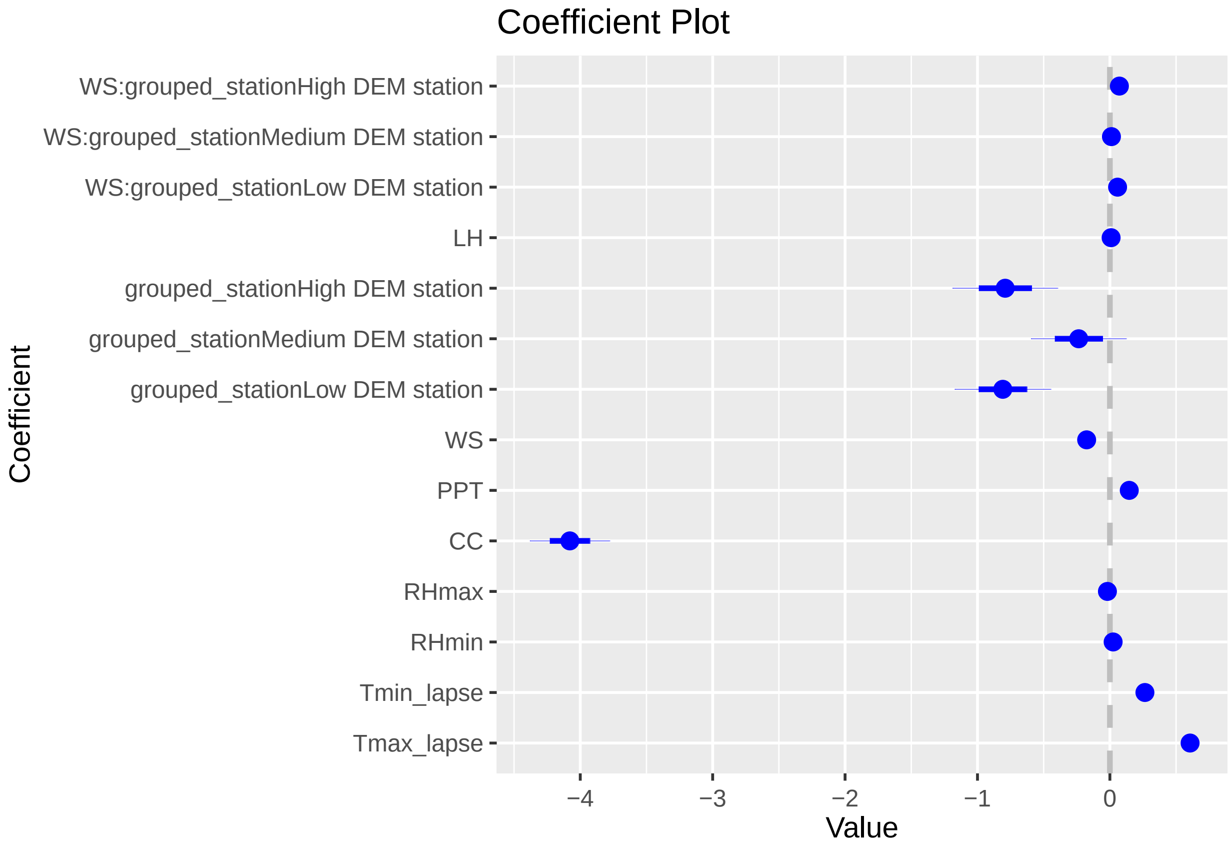
WS:Medium DEM station (0.0118): there is no statistically significant difference in the interaction effect for stations categorized as “Medium elevation” stations, compared to the reference category.

WS:High DEM station (0.0718): the interaction effect suggests that the impact of wind speed on maximum air temperature is higher for stations categorized as “High elevation” compared to “very low elevation”.

```r
library(coefplot)
```

```
## Warning: package 'coefplot' was built under R version 4.2.3
```

```r
plot8 = coefplot(model0.2, intercept = FALSE, only = FALSE)
plot8
```



1.1.4 Find and comment and R^2

```
sigma = summary(model0.2)$sigma
R_squared = summary(model0.2)$r.squared
sigma
```

```
## [1] 1.524986
```

```
R_squared
```

```
## [1] 0.76028
```

The residual standard error of the model, = 1.524986, implies that, on average, the actual values of the dependent variable (the observed next-day max temperature) deviate from the predicted values by approximately 1.52 degrees. The coefficient of determination, R^2, shows that the model predicts 76% of the variance in the dependent variable, that is predictable from the independent variables.

1.1.5 Potential collinearity issues and propose possible solutions

As already commented, collinearity concerns emerged with the cloud coverage and precipitation variables, as their data were collected at different times throughout the day. This problem was solved by averaging the collected values for each entry of the dataset. Additionally, we can see from the correlation matrix of the predictors, that the collinearity between cloud coverage and precipitation is high (0.76). A solution could be to drop the “PPT” variable.

```
selected_variables2 <- c("RHmin", "RHmax", "CC", "PPT")
correlation_matrix2 <- cor(model.matrix(model0.2)[,selected_variables2])
correlation_matrix2
```

```
##           RHmin     RHmax        CC       PPT
## RHmin 1.0000000 0.5783579 0.7581941 0.4387374
## RHmax 0.5783579 1.0000000 0.3513066 0.3206516
## CC    0.7581941 0.3513066 1.0000000 0.5387095
## PPT   0.4387374 0.3206516 0.5387095 1.0000000
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.3

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
vif_values = vif(model0.2)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
selected_variables3 = c("CC", "PPT")
vif_values[selected_variables3, ]
```

```
##         GVIF Df GVIF^(1/(2*Df))
## CC  3.452561  1        1.858107
## PPT 1.511232  1        1.229322
```

After dropping the "PPT" variable, the VIF value for cloud coverage decreases (from 3.45 to 2.97), while the coefficient of determination remains almost unchanged at 75.9%.

```
model0.3 = lm(Next_Tmax ~ Tmax_lapse + Tmin_lapse + RHmin + RHmax + CC + WS * grouped_station + LH, dat
summary(model0.3)$r.squared
```

```
## [1] 0.7589903
```

```
vif_values = vif(model0.3)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
selected_variables4 = c("CC")
vif_values[selected_variables4, ]
```

```
##            GVIF              Df GVIF^(1/(2*Df))
##        2.968936        1.000000        1.723060
```

1.1.6 Test each of the j to be 0 against one-sided or two-sided alternatives. Explicitly state and discuss the hypotheses tested and the conclusions.

We test each coefficient to be 0 against two-sided alternatives: the null hypothesis for each coefficient is that the true value is equal to zero, and the alternative hypothesis is that the true value is different from zero. As seen in the output of the model, all the coefficients are statistically significant (i.e. we reject the null hypothesis), apart from the coefficient for the group of the "Medium elevation" stations and the coefficient for the interaction between "Medium elevation" stations and wind speed.

We also have reasons to believe that the effect of an increase in cloud coverage results in a decrease in the maximum air temperature, in this case the test is one-sided (the alternative hypothesis is that the coefficient for "CC" is negative). From the result of the test, we can conclude that the coefficient is significant, because the p-value is lower than the significance level of = 0.05.

```
coef_CC <- -4.0787022
se_CC <- 0.1506390

#t-test statistic
t_stat_CC <- coef_CC / se_CC

#p-value for the one-sided test
p_value_CC <- pt(t_stat_CC, df = 7573)

cat("t-statistic:", t_stat_CC, "\n")
```

```
## t-statistic: -27.076
```

```
cat("p-value:", p_value_CC, "\n")
```

```
## p-value: 1.724227e-154
```

1.1.7 Test a group of regressors (motivate your choices) and all the regressors. Explicitly state and discuss the hypotheses tested and the conclusions.

We reduce the original model to test only a smaller group of regressors: we exclude the latent heat flux, the next-day forecast of minimum temperature and the maximum relative humidity, as they have a lower correlation to the dependent variable; additionally, we exclude the precipitation level, since we have observed some collinearity with the cloud coverage variable.

```
model_reduced <- lm(Next_Tmax ~ Tmax_lapse + RHmin + CC + WS * grouped_station, data = data)

anova_result <- anova(model_reduced, model0.2)

print(anova_result)
```

```
## Analysis of Variance Table
##
## Model 1: Next_Tmax ~ Tmax_lapse + RHmin + CC + WS * grouped_station
## Model 2: Next_Tmax ~ Tmax_lapse + Tmin_lapse + RHmin + RHmax + CC + PPT +
##     WS * grouped_station + LH
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1   7577 18704
## 2   7573 17612  4    1092.1 117.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic helps determine whether the overall fit of the full model is statistically different from a model with less predictors. In our case, it is equal to 117.4 and its associated p-value is highly significant, indicating that the addition of predictors in the full model significantly improves the fit compared to the reduced model.

1.1.8 Suppose you have information about a new observation of your regressors. Provide the prediction of your response variable with the associated uncertainty.

If we want to predict the maximum air temperature in a summer non-rainy day where the values of relative max and min humidity, cloud coverage, wind speed, latent heat, the forecasted min and max temperatures of the LDAPS model are at their median levels and the forecasts are produced by a station placed at an elevation between 100 and 250 meters, we see that the prediction is 30.32°C.

```
data2 = data.frame(RHmin = 55, RHmax = 90, CC = 0.30, PPT = 0, WS = 6.5, LH = 57, Tmax_lapse = 29.7, Tm
pred = predict(model0.2, newdata = data2, interval = "prediction", level = 0.95)
pred
```

```
##        fit    lwr      upr
## 1 30.31722 27.326 33.30845
```