Climate Project: Prediction of Soil Temperature

Jonas Graf

16 - August - 2022

Contents

1 Data Setup

```r
# Libraries: install
if (!require(tidyverse)) install.packages("tidyverse", repos =
"http://cran.us.r-project.org")
if (!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if (!require(cowplot)) install.packages("cowplot", repos =
"http://cran.us.r-project.org")
if (!require(magick)) install.packages("magick", repos = "http://cran.us.r-project.org")
if (!require(data.table)) install.packages("data.table", repos =
"http://cran.us.r-project.org")
if (!require(ggplot2)) install.packages("ggplot2", repos =
"http://cran.us.r-project.org")
if (!require(kableExtra)) install.packages("kableExtra", repos =
"http://cran.us.r-project.org")
if (!require(stringr)) install.packages("stringr", repos =
"http://cran.us.r-project.org")
if (!require(tidyr)) install.packages("tidyr", repos = "http://cran.us.r-project.org")
if (!require(forcats)) install.packages("forcats", repos =
"http://cran.us.r-project.org")
if (!require(formatR)) install.packages("formatR", repos =
"http://cran.us.r-project.org")
if (!require(ranger)) install.packages("ranger", repos = "http://cran.us.r-project.org")
if (!require(lubridate)) install.packages("lubridate", repos =
"http://cran.us.r-project.org")

# Libraries: load
library(tidyverse)
library(caret)
library(cowplot)
library(magick)
library(data.table)
library(dplyr)
library(ggplot2)
library(kableExtra)
library(tidyr)
library(stringr)
library(forcats)
library(formatR)
library(ranger)
library(lubridate)


# German climate datasets, weather station Bremen (ID 691):
# Source: Deutscher Wetterdienst

# Downloading environment file
dl <- tempfile()
download.file("https://opendata.dwd.de/climate_environment/CDC/observations_-
germany/climate/daily/kl/historical/tageswerte_KL_00691_18900101_20211231_hist.zip",
    dl)

env <- fread(text = gsub("::", "\t", readLines(unzip(dl,
"produkt_klima_tag_18900101_20211231_00691.txt"))),
    col.names = c("STATIONS_ID", "MESS_DATUM", "QN_3", "FX",
        "FM", "QN_4", "RSK", "RSKF", "SDK", "SHK_TAG", "NM",
        "VPM", "PM", "TMK", "UPM", "TXK", "TNK", "TGK", "eor"))

env <- env %>%
    select(c("MESS_DATUM", "RSK", "SDK", "VPM", "TMK", "UPM",
        "TXK", "TNK", "TGK"))

# Downloading soil temperature file
dl <- tempfile()
download.file("https://opendata.dwd.de/climate_environment/CDC/observations_-
germany/climate/daily/soil_temperature/historical/tageswerte_EB_00691_19510101_20211231_-
hist.zip",
    dl)

soil <- fread(text = gsub("::", "\t", readLines(unzip(dl,
"produkt_erdbo_tag_19510101_20211231_00691.txt"))),
    col.names = c("STATIONS_ID", "MESS_DATUM", "QN_2", "V_TE002M",
        "V_TE005M", "V_TE010M", "V_TE020M", "V_TE050M", "eor"))

soil <- soil %>%
    select(c("MESS_DATUM", "V_TE005M"))

# Linking soil and env datasets
climate <- left_join(soil, env, by = "MESS_DATUM")

# Removing measurement dates with a missing value ('-999')
climate$label = ifelse(climate$RSK == "-999" | climate$SDK ==
    "-999" | climate$VPM == "-999" | climate$TMK == "-999" |
    climate$UPM == "-999" | climate$TXK == "-999" | climate$TNK ==
    "-999" | climate$TGK == "-999" | climate$V_TE005M == "-999",
    0, 1)

climate <- climate[climate$label == 1, ]
climate <- climate[, c(1:10)]

# Rounding temperatures
climate$V_TE005M <- round(climate$V_TE005M, 0)
climate$TMK <- round(climate$TMK, 0)
climate$TXK <- round(climate$TXK, 0)
climate$TNK <- round(climate$TNK, 0)
climate$TGK <- round(climate$TGK, 0)
climate$UPM <- round(climate$UPM, 0)

# Partitioning 80/20 (i.e., 5 fold) according to the Pareto
# Principle https://en.wikipedia.org/wiki/Pareto_principle
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = climate$V_TE005M, times = 1,
    p = 0.2, list = FALSE)
edx <- climate[-test_index, ]
temp <- climate[test_index, ]

# Make sure variable values in validation set are also in
# edx set
validation <- temp %>%
    semi_join(edx, by = "TMK") %>%
    semi_join(edx, by = "SDK") %>%
    semi_join(edx, by = "VPM") %>%
    semi_join(edx, by = "UPM") %>%
    semi_join(edx, by = "RSK") %>%
    semi_join(edx, by = "TXK")

# Add rows removed from validation set back into edx set
removed <- anti_join(temp, validation)
edx <- rbind(edx, removed)
rm(climate, env, soil, test_index, dl, removed, temp)
```

2 Introduction

The overarching aim of this project is to predict soil temperatures using daily climate and soil observations captured at the weather station Bremen, Germany (station identification number 691). The data are provided by the *Deutscher Wetterdienst*. Bremen is a city situated in Northwestern Germany (marked red on map).

Map Credit: TUBS, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons.

A detailed description of the data can be found here:
https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/DESCRIPTION_obsgermany_climate_daily_kl_historical_en.pdf
https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/soil_temperature/historical/DESCRIPTION_obsgermany_climate_daily_soil_temperature_historical_en.pdf

These data include >20,000 daily measurements (e.g., air temperature, relative humidity) captured between 1951 and 2021. Predicting soil temperature may be useful for farmers in order to find the optimal time point of planting (https://extensionpublications.unl.edu/assets/pdf/g2122.pdf).

Three machine learning models will be fitted: linear, k nearest neighbors, and random forest. The goal is to achieve a Root Mean Squared Error (RMSE) below 1.0. The RMSE will be calculated using the following equation:

$$\boxed{\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i}(\hat{y}_i - y_i)^2}}$$

The random forest model approach reached an RMSE of ~1.2. However, the model missed the overarching aim of the project (RMSE < 1.0).

3 Methods & Analysis

3.1 Descriptive data exploration

First, an exploration of the training data is warranted.

3.1.1 Variables: overview

Here, an overview of the abbreviations, meanings and units of the variables is provided.

Abbreviation	Meaning	Unit
MESS_DATUM	date of measurement	yyyymmdd
V_TE005M	daily soil temperature in 5cm depth	°C
RSK	daily precipitation height	mm
SDK	daily sunshine duration	h
VPM	daily mean of vapor pressure	hPa
TMK	daily mean of temperature	°C
UPM	daily mean of relative humidity	%
TXK	daily maximum of temperature at 2m height	°C
TNK	daily minimum of temperature at 2m height	°C
TGK	daily minimum of air temperature at 5cm above ground	°C

3.1.2 First six data entries: overview

The head function in R provides us a first descriptive overview of the variables.

MESS_DATUM	V_TE005M	RSK	SDK	VPM	TMK	UPM	TXK	TNK	TGK
19510101	-3	2.4	0.0	5.2	-1	89	1	-9	-9
19510102	-2	4.2	0.0	5.6	0	90	2	-2	-3
19510103	-1	0.0	1.9	6.0	1	90	3	-3	-4
19510104	-1	1.4	2.0	6.1	2	90	2	0	-1
19510105	0	7.5	0.0	7.9	5	92	6	1	1
19510106	1	6.2	0.0	7.7	4	95	7	3	3

The overview revealed that the class of the 'MESS_DATUM' variables needs to be formatted from numeric to date.

3.1.3 Classes: overview

MESS_DATUM	V_TE005M	RSK	SDK	VPM	TMK	UPM	TXK	TNK	TGK
integer	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric

3.1.4 Reformatting date column & creating month of measurement variable

In addition to the reformatting, a variable containing the month of measurement needs to be created. Soil temperatures likely follow a seasonal trend. Hence, a month of measurement variable may become useful for the analysis.

```
edx$MESS_DATUM <- ymd(edx$MESS_DATUM)
edx$month_m <- format(edx$MESS_DATUM, "%m")
validation$MESS_DATUM <- ymd(validation$MESS_DATUM)
validation$month_m <- format(validation$MESS_DATUM, "%m")
```

3.1.5 First, last, and count of measurements

Here, the dates of the first/last measurement is provided. Further, the table below depicts the total count of observations within the training dataset.

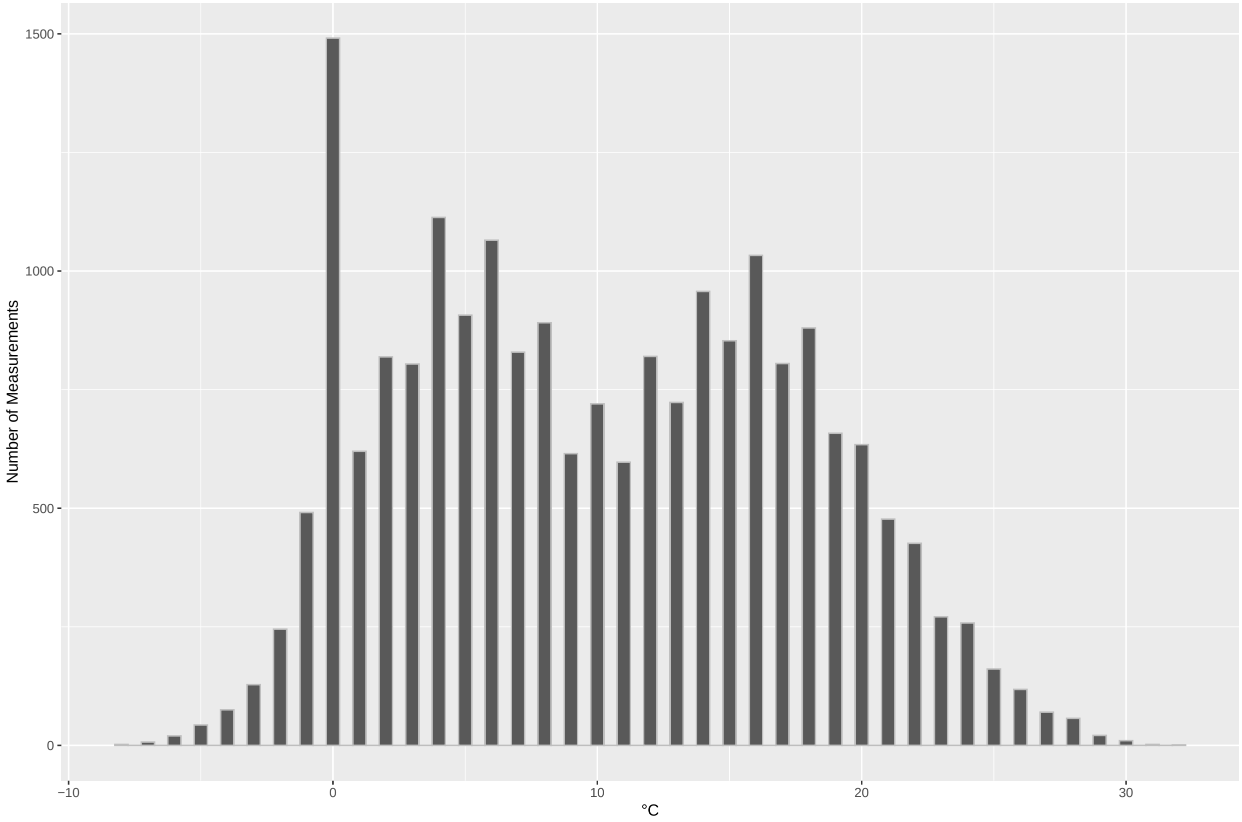
```
data.frame(First = min(edx$MESS_DATUM), Last = max(edx$MESS_DATUM),
    Count = dim(edx)[1]) %>%
    kable(format = "simple", align = "c")
```

First	Last	Count
1951-01-01	2021-12-31	20717

3.2 Visualization

Next, providing a visualization of the formatted training data is warranted.

3.2.1 Distribution: soil temperature



The histogram of the soil temperature reveals that 0°C measurement are most frequent. This may reflect the physical phenomenon enthalpy of fusion. In short, an additional amount of energy is necessary in order for water to change from frozen to liquid. In case of water, 333 kJ are necessary for ice (e.g., within the soil) to melt; this is the same amount of energy necessary for heating ice from -40°C to 0°C. However, if enthalpy of fusion makes our predictions more difficult remains subject of speculation.

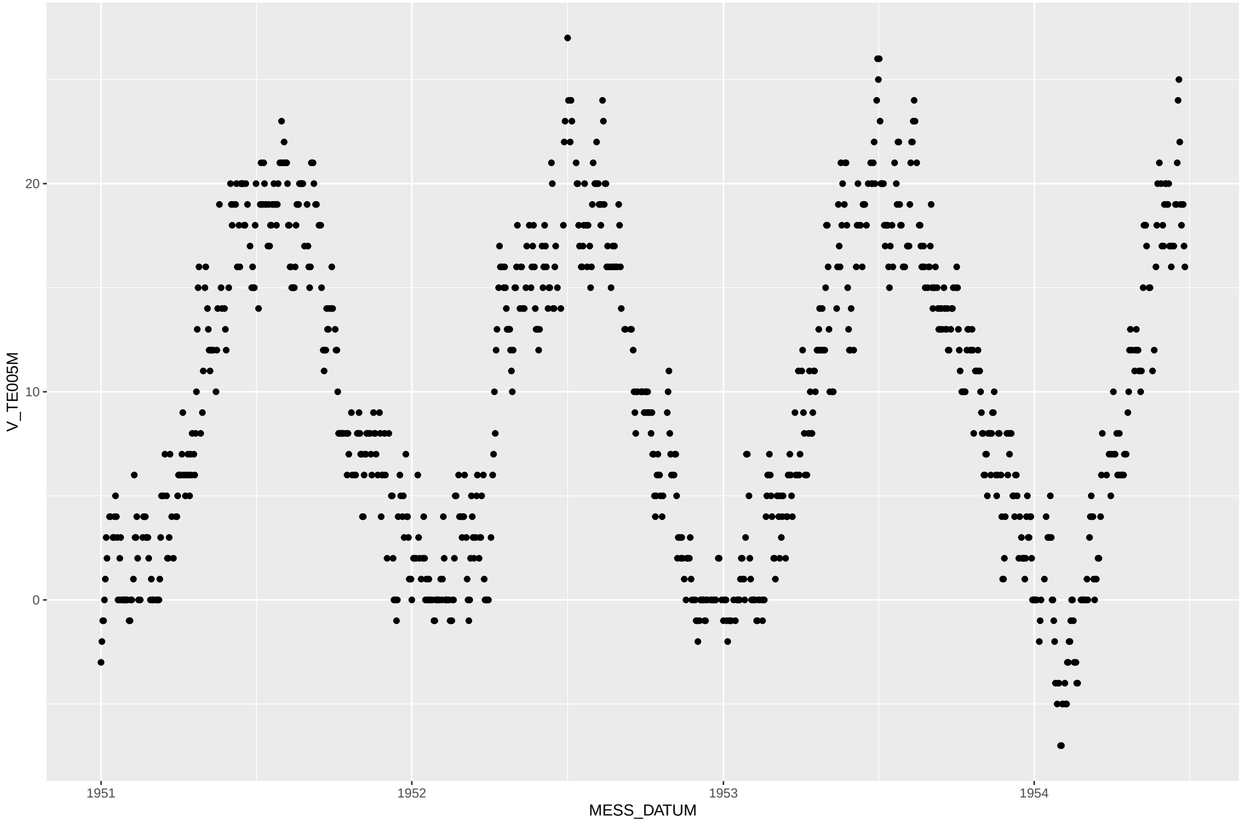
The soil temperature values center around a median of 10°C.

```
median(edx$V_TE005M)
```

```
## [1] 10
```

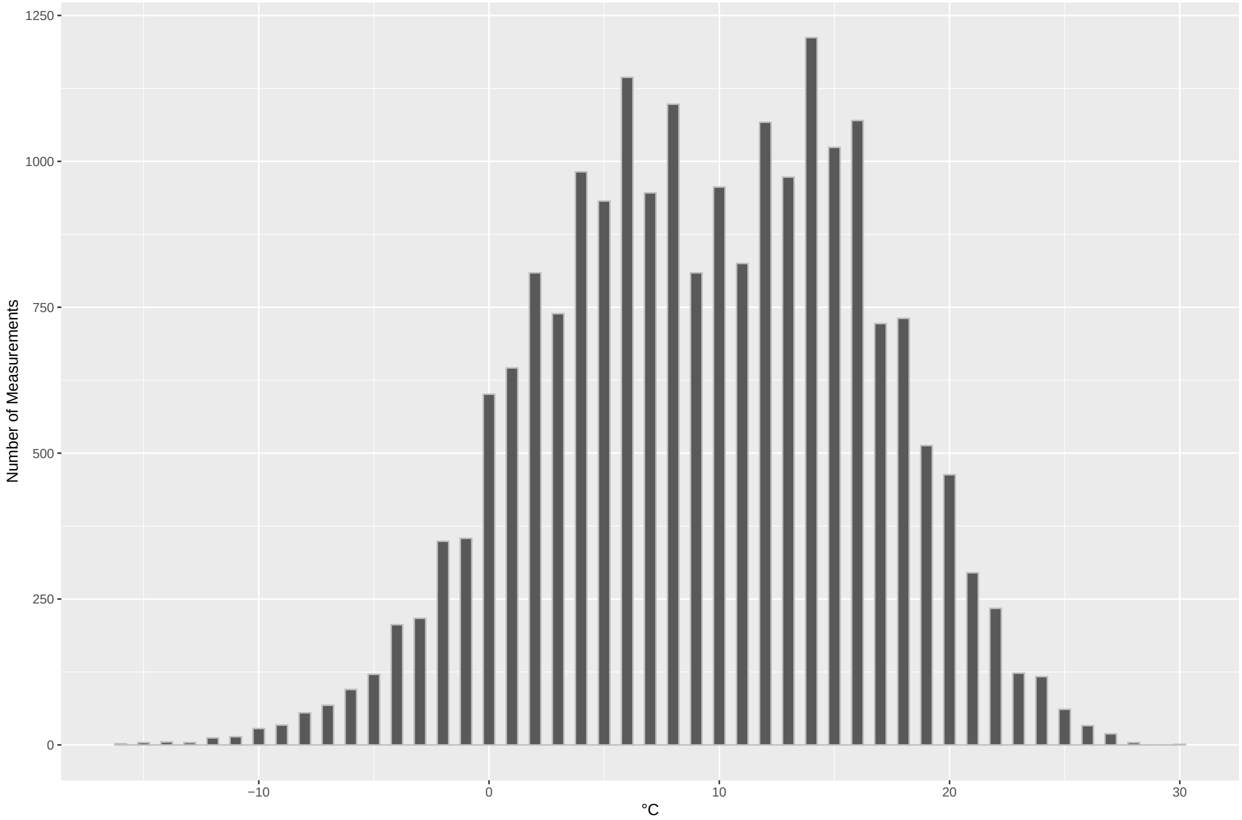
3.2.2 Seasonal trend: soil temperature

```
# Date of measurement versus soil temperature - excerpt
qplot(MESS_DATUM, V_TE005M, data = edx[1:1000, ])
```



As expected, the soil temperature measurements follow a seasonal trend.

3.2.3 Distribution: air temperature - mean



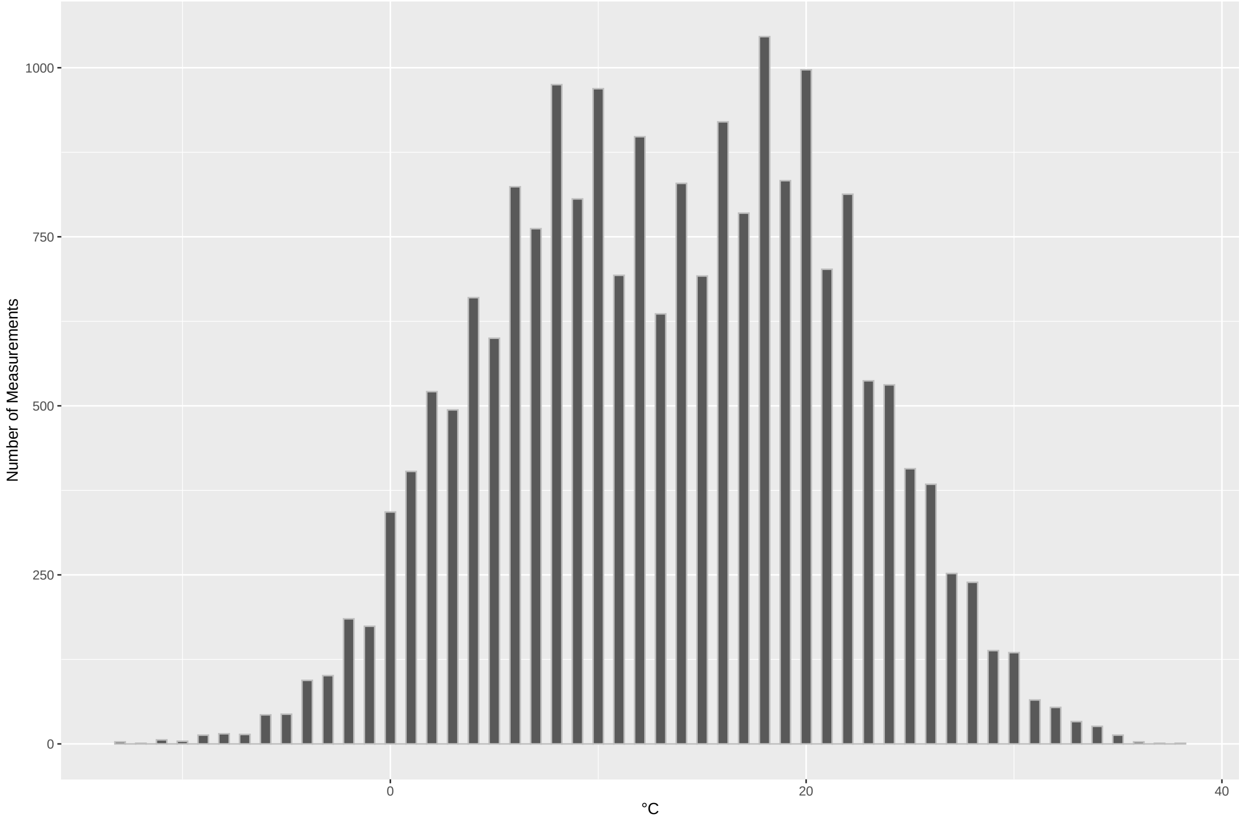
Just like soil temperature, the mean air temperature centers around a median of 10°C.

```
median(edx$TMK)
```

```
## [1] 10
```

However, 14°C measurements are most frequent.

3.2.4 Distribution: air temperature - max

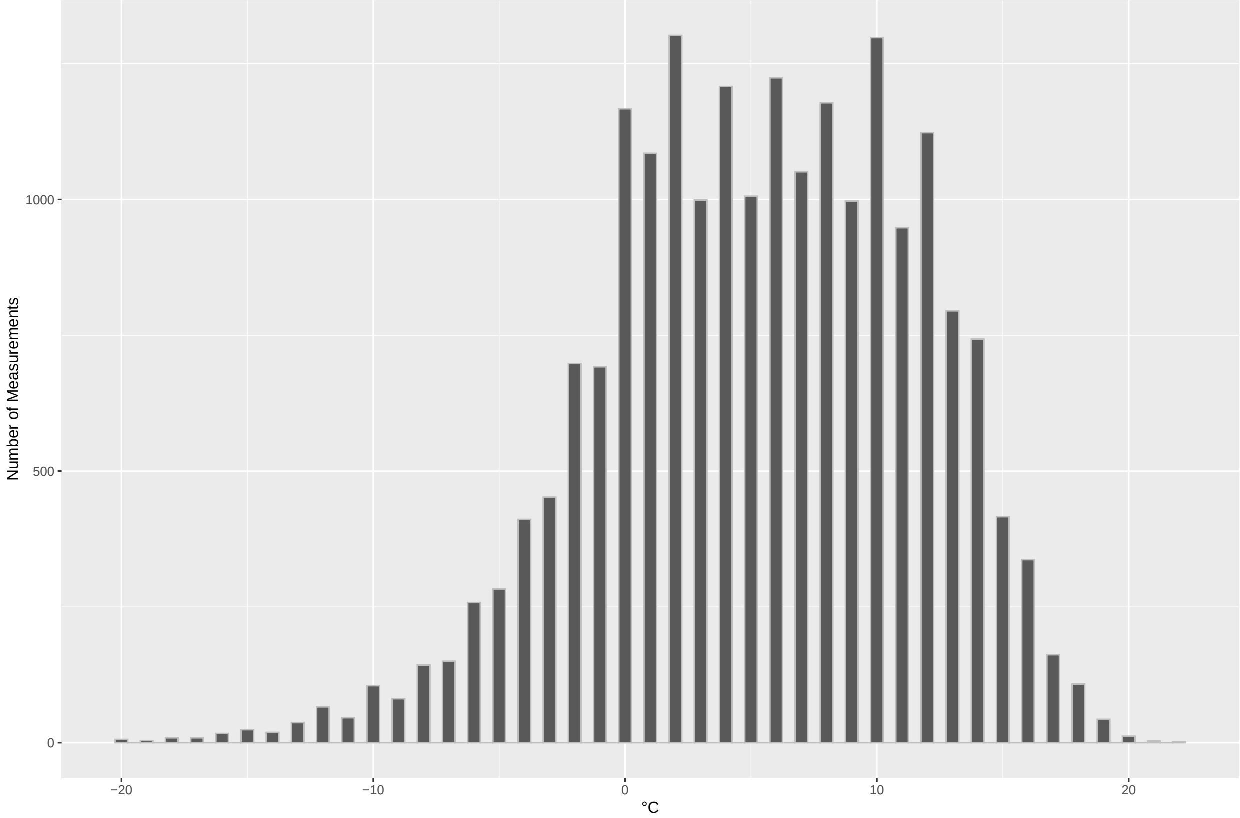


The distribution of the maximum air temperature centers around a median of 14°C.

```
median(edx$TXK)
```

```
## [1] 14
```

3.2.5 Distribution: air temperature - min

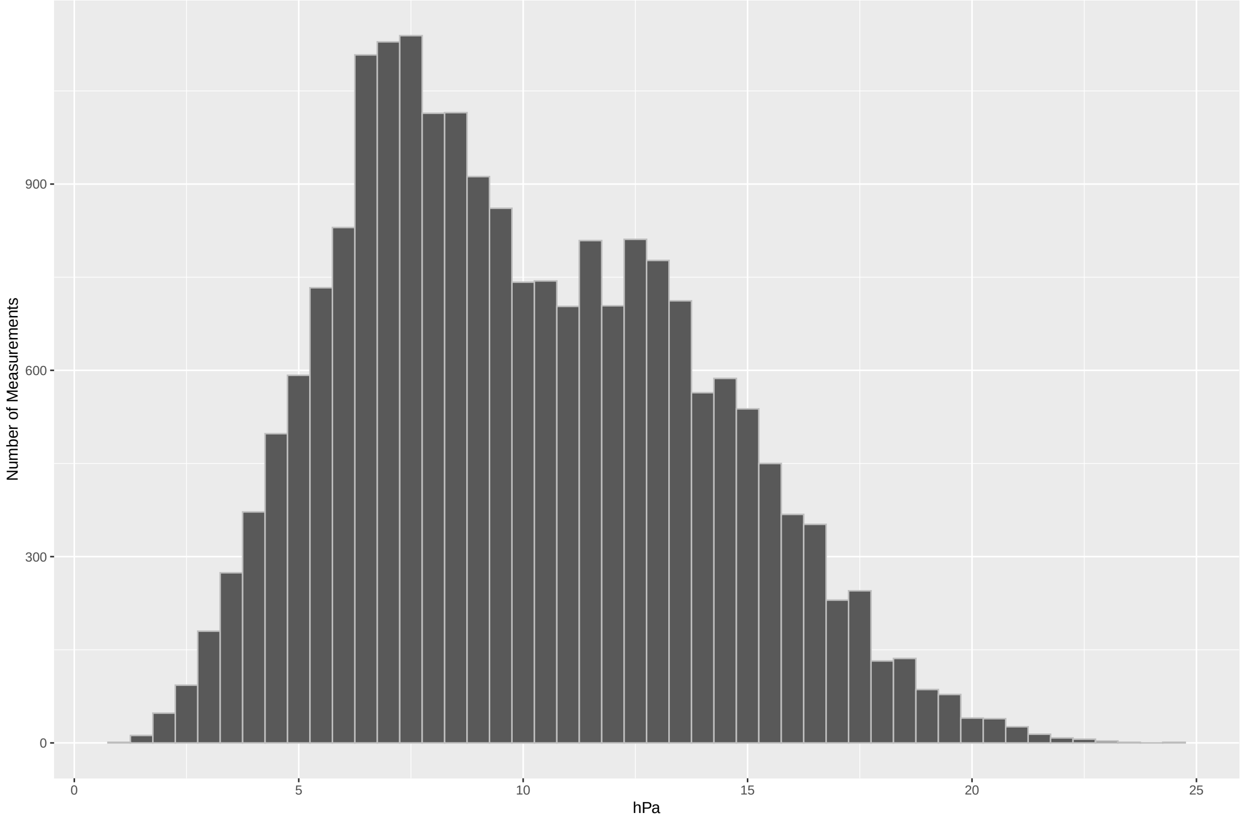


The distribution of the minimum air temperature centers around a median of 6°C.

```
median(edx$TNK)
```

```
## [1] 6
```

3.2.6 Distribution: vapor pressure

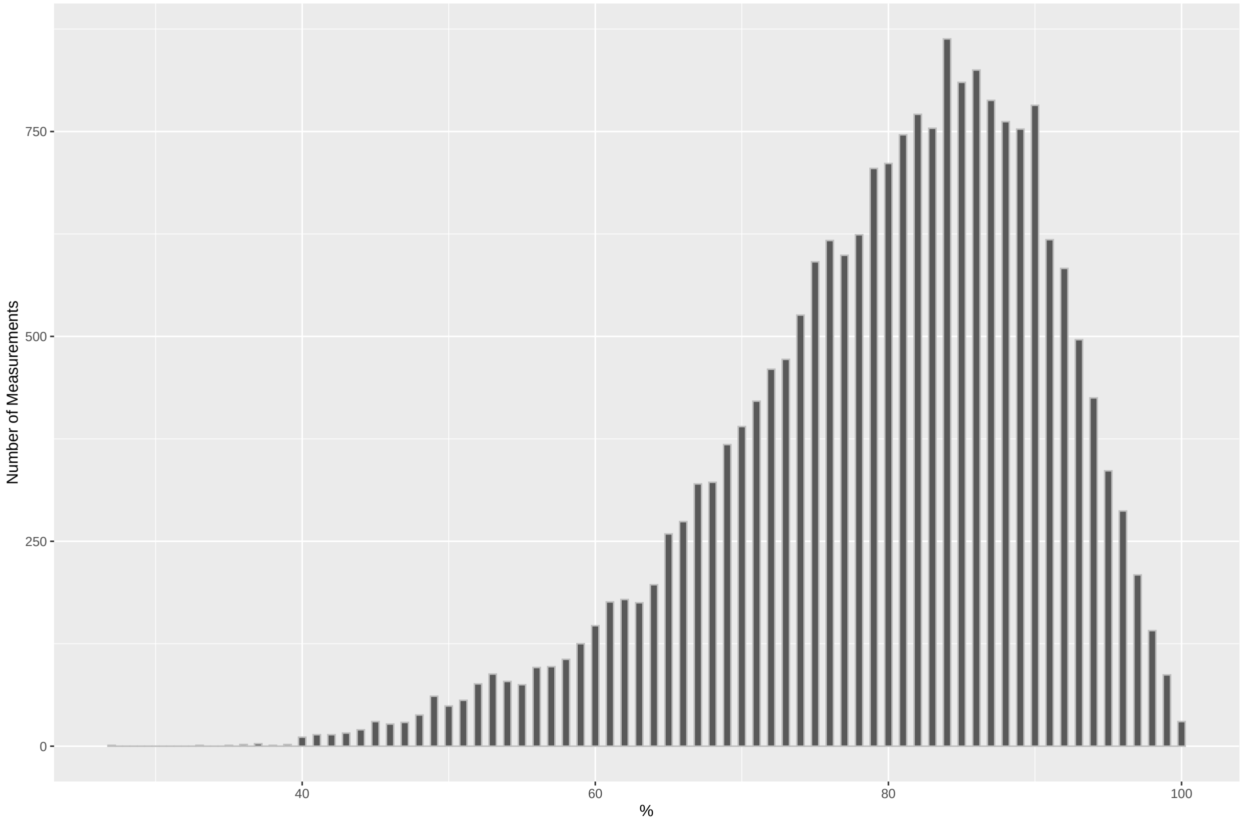


The distribution of vapor pressure appears to be shifted to below a median of 9.5%.

```
median(edx$VPM)
```

```
## [1] 9.5
```

3.2.7 Distribution: relative humidity



The distribution of the relative humidity, however, is shifted to values above the median of 81%.

```r
median(edx$UPM)
```

```
## [1] 81
```

3.3 Models: building & evaluation

Three models will be fitted: linear (with and without regularization), random forest, and k nearest neighbors.

First, a partition of the edx set for training/testing purposes needs to be created. I have opted for an 80/20 split inspired by the Pareto principle.

```
# Creating data partition of edx for cross-validation
# Validation set will be 20% of edx data
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = edx$V_TE005M, times = 1,
    p = 0.2, list = FALSE)
train_edx <- edx[-test_index, ]
temp <- edx[test_index, ]

# Make sure TMK in validation set are also in edx set
test_edx <- temp %>%
    semi_join(train_edx, by = "TMK") %>%
    semi_join(train_edx, by = "TMK") %>%
    semi_join(train_edx, by = "SDK") %>%
    semi_join(train_edx, by = "VPM") %>%
    semi_join(train_edx, by = "UPM") %>%
    semi_join(train_edx, by = "RSK") %>%
    semi_join(train_edx, by = "TXK")

# Add rows removed from validation set back into edx set
removed <- anti_join(temp, test_edx)
```

```
## Joining, by = c("MESS_DATUM", "V_TE005M", "RSK", "SDK", "VPM", "TMK", "UPM",
## "TXK", "TNK", "TGK", "month_m")
```

```
train_edx <- rbind(train_edx, removed)
rm(test_index, temp)
```

3.4 Linear model without regularization

3.4.1 Background

Given that I am curious about the performance of a linear model on technical measurements/climate data, I have opted to include/develop this approach in addition to two more advanced models (i.e., k nearest neighbor, random forest).

3.4.2 Predicting the mean temperature

Always predicting the mean soil temperature is perhaps the simplest model. In the training data, the mean is ~10.2°C.

```
mean(train_edx$V_TE005M)
```

```
## [1] 10.16686
```

```r
# Mean value of all soil temperatures
mu_hat <- mean(train_edx$V_TE005M)
# Predict the RMSE on the test_edx set
mean_model_result <- RMSE(test_edx$V_TE005M, mu_hat)
# Gathering RMSE results in a dataframe
results <- data.frame(model = "_Mean_ **test**", RMSE = mean_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744

An RMSE of >7 underscores a very poor model performance.

3.4.3 Considering air temperature

The goal is to improve the 'predicting the mean' model by considering the air temperature effect.

The following equation depicts this approach:

$$\boxed{Y_i = \hat{\mu} + b_i + \epsilon_i}$$

In short, $\hat{\mu}$ represents the mean soil temperature and ε_i the independent errors. The b_i represents the magnitude of the air temperature effect i.

```r
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Computing predicted soil temperatures on test_edx dataset
TMK_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    mutate(pred = mu_hat + b_i) %>%
    pull(pred)
mean_TMK_result <- RMSE(test_edx$V_TE005M, TMK_model)
# Expanding the results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_ **test**", RMSE = mean_TMK_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966

As expected, when considering the air temperature, the RMSE is very much improved (decreased), but still rather poor with a value >1.5.

3.4.4 Considering mean air temperature & vapor pressure

The RMSE is still too high. Hence, further considerations are warranted. Here, a vapor pressure effect is being accounted for.

The following equation depicts this more complex approach:

$$\boxed{Y_{v,i} = \hat{\mu} + b_v + b_i + \epsilon_{v,i}}$$

In addition to the mean air temperature equation above, b_v represents the magnitude of a potential vapor pressure effect v.

```
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
VPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(VPM) %>%
    summarize(b_v = mean(V_TE005M - mu_hat - b_i))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_VPM_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    mutate(pred = mu_hat + b_v + b_i) %>%
    pull(pred)
mean_TMK_VPM_model_result <- RMSE(mean_TMK_VPM_model, test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_VPM_ **test**", RMSE = mean_TMK_VPM_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593

Considering pressure increased the RMSE, which is why this parameter will not be considered.

3.4.5 Considering mean & minimum air temperature

The following equation represents this model:

$$\boxed{Y_{m,i} = \hat{\mu} + b_i + b_m + \epsilon_{m,i}}$$

In addition to the previous model, b_m mirrors the magnitude of a given minimum air temperature effect m on the soil temperature.

```r
# Mean soil temperatures of all soil temperature
# measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temperature
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_TNK_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    mutate(pred = mu_hat + b_m + b_i) %>%
    pull(pred)
mean_TMK_TNK_model_result <- RMSE(mean_TMK_TNK_model, test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_ **test**", RMSE = mean_TMK_TNK_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590

The minimum air temperature slightly decreases the RMSE. Perhaps including the relative humidity into our model may lead to further improvement.

3.4.6 Considering mean & minimum air temperature, & relative humidity

The following equation represents this model (b_h represents mean relative humidity):

$$\boxed{Y_{m,h,i} = \hat{\mu} + b_i + b_m + b_h + \epsilon_{m,h,i}}$$

```r
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Calculating mean by relative humidity
UPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    group_by(UPM) %>%
    summarize(b_h = mean(V_TE005M - mu_hat - b_m - b_i))
# Compute the predicted soil temps on test_edx dataset
mean_TMK_TNK_UPM_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h) %>%
    pull(pred)
mean_TMK_TNK_UPM_model_result <- RMSE(mean_TMK_TNK_UPM_model,
    test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_+_UPM_ **test**", RMSE =
    mean_TMK_TNK_UPM_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945

Adding relative humidity to the linear model improved (decreased) the RMSE, which is why this variable will be included.

3.4.7 Considering mean & minimum air temperature, relative humidity & precipitation height

The following equation represents this expanded model(b_p stands for mean precipitation height):

$$\boxed{Y_{m,h,i,p} = \hat{\mu} + b_i + b_m + b_h + b_p + \epsilon_{m,h,i,p}}$$

```
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Calculating mean by relative humidity
UPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    group_by(UPM) %>%
    summarize(b_h = mean(V_TE005M - mu_hat - b_i - b_m))
# Calculating mean by precipitation height
RSK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    group_by(RSK) %>%
    summarize(b_p = mean(V_TE005M - mu_hat - b_m - b_i - b_h))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_TNK_UPM_RSK_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(RSK_avgs, by = "RSK") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h + b_p) %>%
    pull(pred)
mean_TMK_TNK_UPM_RSK_model_result <- RMSE(mean_TMK_TNK_UPM_RSK_model,
    test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_+_UPM_+_RSK_ **test**",
        RMSE = mean_TMK_TNK_UPM_RSK_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258

Considering precipitation height increased the RMSE, which is why this parameter will not be considered.

3.4.8 Considering mean & minimum air temperature, relative humidity & sunshine duration

The following equation represents this expanded model (b_s represents mean daily sunshine duration):

$$\boxed{Y_{m,h,i,s} = \hat{\mu} + b_i + b_m + b_h + b_s + \epsilon_{m,h,i,s}}$$

```r
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Calculating mean by relative humidity
UPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    group_by(UPM) %>%
    summarize(b_h = mean(V_TE005M - mu_hat - b_i - b_m))
# Calculating mean by sunshine duration
SDK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    group_by(SDK) %>%
    summarize(b_s = mean(V_TE005M - mu_hat - b_m - b_i - b_h))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_TNK_UPM_SDK_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(SDK_avgs, by = "SDK") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h + b_s) %>%
    pull(pred)
mean_TMK_TNK_UPM_SDK_model_result <- RMSE(mean_TMK_TNK_UPM_SDK_model,
    test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_+_UPM_+_SDK_ **test**",
        RMSE = mean_TMK_TNK_UPM_SDK_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152

Considering daily sunshine duration increased the RMSE, which is why this parameter will not be considered.

3.4.9 Considering mean, maximum & minimum air temperature, & relative humidity

The following equation represents this expanded model:

$$\boxed{Y_{m,h,i,x} = \hat{\mu} + b_i + b_m + b_h + b_x + \epsilon_{m,h,i,x}}$$

```r
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
TMK_avgs <- train_edx %>% # Calculating mean by mean air temperature
   group_by(TMK) %>%
   summarize(b_i = mean(V_TE005M - mu_hat))
TNK_avgs <- train_edx %>% # Calculating mean by min air temp
   left_join(TMK_avgs, by="TMK") %>%
   group_by(TNK) %>%
   summarize(b_m = mean(V_TE005M - mu_hat - b_i))
UPM_avgs <- train_edx %>% # Calculating mean by relative humidity
   left_join(TMK_avgs, by="TMK") %>%
   left_join(TNK_avgs, by="TNK") %>%
   group_by(UPM) %>%
   summarize(b_h = mean(V_TE005M - mu_hat - b_i - b_m))
TXK_avgs <- train_edx %>% # Calculating mean by max air temp
    left_join(TMK_avgs, by="TMK") %>%
    left_join(VPM_avgs, by="VPM") %>%
    left_join(TNK_avgs, by="TNK") %>%
    left_join(UPM_avgs, by="UPM") %>%
    group_by(TXK) %>%
    summarize(b_x = mean(V_TE005M - mu_hat - b_m - b_i - b_h))
mean_TMK_TNK_UPM_TXK_model <- test_edx %>% # Predicting soil temperatures on test_edx
dataset
    left_join(TMK_avgs, by="TMK") %>%
    left_join(TNK_avgs, by="TNK") %>%
    left_join(UPM_avgs, by="UPM") %>%
    left_join(TXK_avgs, by="TXK") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h + b_x) %>%
    pull(pred)
mean_TMK_TNK_UPM_TXK_model_result <- RMSE(mean_TMK_TNK_UPM_TXK_model, test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>% add_row(model="_Mean_+_TMK_+_TNK_+_UPM_+_TXK_ **test**",
RMSE=mean_TMK_TNK_UPM_TXK_model_result)
results %>% kable(format = "simple", align = 'c')
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812

Adding the maximum air temperature to the model improved (decreased) the RMSE, which is why this variable will be included.

3.4.10 Considering mean, maximum & minimum air temperature, relative humidity & minimum air temperature 5cm above ground

The following equation represents this expanded model (b_g represents the minimum air temperature 5cm above ground):

$$\boxed{Y_{m,h,i,x,g} = \hat{\mu} + b_i + b_m + b_h + b_x + b_g + \epsilon_{m,h,i,x,g}}$$

```r
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Calculating mean by relative humidity
UPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    group_by(UPM) %>%
    summarize(b_h = mean(V_TE005M - mu_hat - b_i - b_m))
# Calculating mean by max air temp
TXK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    group_by(TXK) %>%
    summarize(b_x = mean(V_TE005M - mu_hat - b_m - b_i - b_h))
# Calculating mean by min air temp 2cm above ground
TGK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(TXK_avgs, by = "TXK") %>%
    group_by(TGK) %>%
    summarize(b_g = mean(V_TE005M - mu_hat - b_m - b_i - b_h -
        b_x))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_TNK_UPM_TXK_TGK_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(TXK_avgs, by = "TXK") %>%
    left_join(TGK_avgs, by = "TGK") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h + b_x + b_g) %>%
    pull(pred)
mean_TMK_TNK_UPM_TXK_TGK_model_result <- RMSE(mean_TMK_TNK_UPM_TXK_TGK_model,
    test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_+_UPM_+_TXK_+_TGK_ **test**",
        RMSE = mean_TMK_TNK_UPM_TXK_TGK_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094

Adding minimum temperature 2cm above ground to the model increased the RMSE, which is why this parameter will not be considered.

3.4.11 Considering mean, maximum & minimum air temperature, relative humidity & month of measurement

The following equation represents this expanded model (b_t mirrors the month of measurement):

$$\boxed{Y_{m,h,i,x,g,t} = \hat{\mu} + b_i + b_m + b_h + b_x + b_t + \epsilon_{m,h,i,x,g,t}}$$

```
# Mean value of all soil temperature measurements
mu_hat <- mean(train_edx$V_TE005M)
# Calculating the mean by mean air temperature
TMK_avgs <- train_edx %>%
    group_by(TMK) %>%
    summarize(b_i = mean(V_TE005M - mu_hat))
# Calculating mean by min air temp
TNK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    group_by(TNK) %>%
    summarize(b_m = mean(V_TE005M - mu_hat - b_i))
# Calculating mean by relative humidity
UPM_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    group_by(UPM) %>%
    summarize(b_h = mean(V_TE005M - mu_hat - b_i - b_m))
# Calculating mean by max air temp
TXK_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    group_by(TXK) %>%
    summarize(b_x = mean(V_TE005M - mu_hat - b_m - b_i - b_h))
# Calculating mean by min air temp 2cm above ground
month_avgs <- train_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(VPM_avgs, by = "VPM") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(TXK_avgs, by = "TXK") %>%
    group_by(month_m) %>%
    summarize(b_t = mean(V_TE005M - mu_hat - b_m - b_i - b_h -
        b_x))
# Computing predicted soil temperatures on test_edx dataset
mean_TMK_TNK_UPM_TXK_month_model <- test_edx %>%
    left_join(TMK_avgs, by = "TMK") %>%
    left_join(TNK_avgs, by = "TNK") %>%
    left_join(UPM_avgs, by = "UPM") %>%
    left_join(TXK_avgs, by = "TXK") %>%
    left_join(month_avgs, by = "month_m") %>%
    mutate(pred = mu_hat + b_m + b_i + b_h + b_x + b_t) %>%
    pull(pred)
mean_TMK_TNK_UPM_TXK_month_model_result <- RMSE(mean_TMK_TNK_UPM_TXK_month_model,
    test_edx$V_TE005M)
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_TNK_+_UPM_+_TXK_+_month_ **test**",
        RMSE = mean_TMK_TNK_UPM_TXK_month_model_result)
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826

3.5 Linear model with regularization

3.5.1 Background

The regularization approach allows us to adjust for variables with large estimates which were formed from small sample sizes by adding the term λ (lambda). However, given the nature of the technical measurements, I do not expect that regularization will have an important impact on the RMSE.

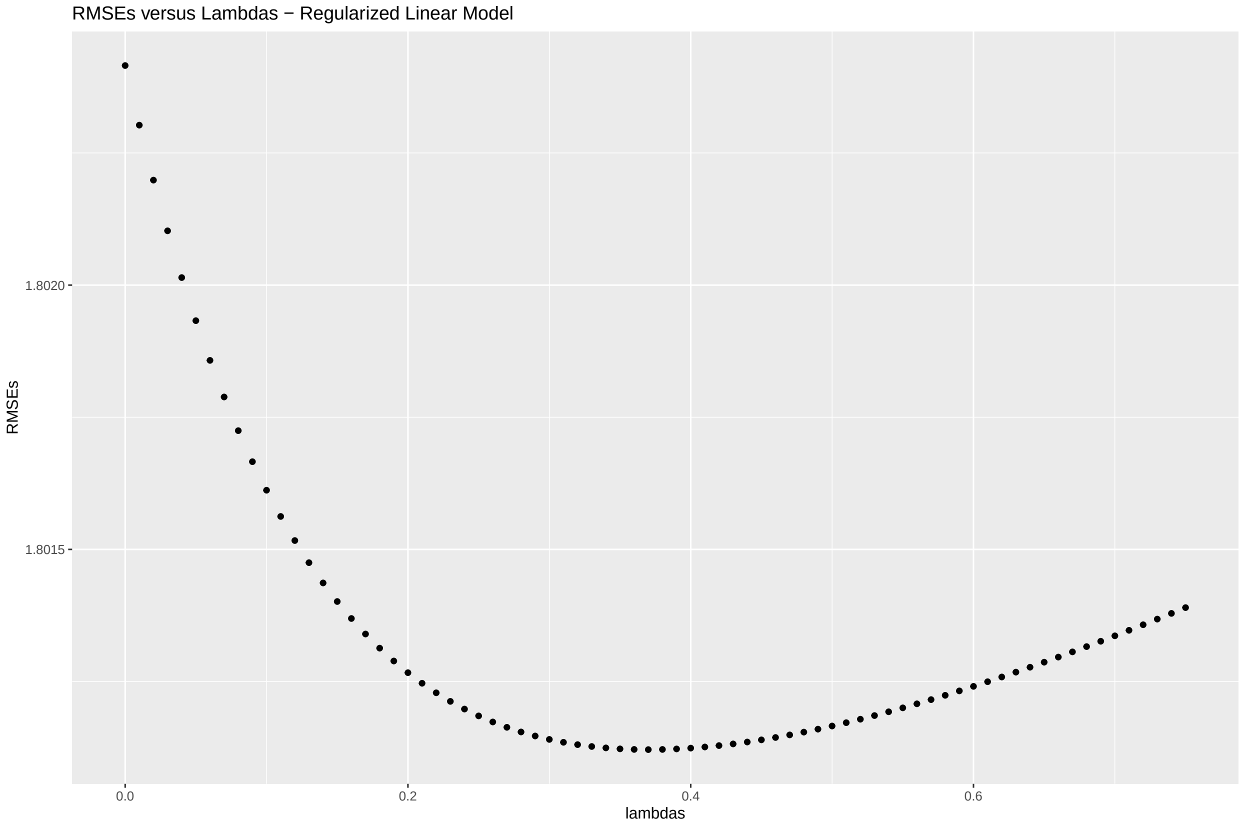
The following equation will be applied:

$$\boxed{\hat{b}_i(\lambda) = \frac{1}{\lambda + n_i}\sum_{u=1}^{n_i}(Y_i - \hat{\mu})}$$

3.5.2 Considering mean, maximum & minimum air temperature, relative humidity & month of measurement with regularization

```
# Mean of all soil temperatures
mu_hat <- mean(train_edx$V_TE005M)
# Defining string of lambdas
lambdas <- seq(0, 0.75, 0.01)
# Cross-validation of lambdas
rmses <- sapply(lambdas, function(lambda) {
    # Including mean air temperature
    b_i <- train_edx %>%
        group_by(TMK) %>%
        summarize(b_i = sum(V_TE005M - mu_hat)/(n() + lambda))
    # Including minimum temp at 2m
    b_m <- train_edx %>%
        left_join(b_i, by = "TMK") %>%
        group_by(TNK) %>%
        summarize(b_m = sum(V_TE005M - b_i - mu_hat)/(n() + lambda))
    # Including humidity
    b_h <- train_edx %>%
        left_join(b_i, by = "TMK") %>%
        left_join(b_m, by = "TNK") %>%
        group_by(VPM) %>%
        summarize(b_h = sum(V_TE005M - b_m - b_i - mu_hat)/(n() +
            lambda))
    # Including max temp at 2m height
    b_x <- train_edx %>%
        left_join(b_i, by = "TMK") %>%
        left_join(b_h, by = "VPM") %>%
        left_join(b_m, by = "TNK") %>%
        group_by(TXK) %>%
        summarize(b_x = sum(V_TE005M - b_m - b_i - b_h - mu_hat)/(n() +
            lambda))
    # Including month
    b_t <- train_edx %>%
        left_join(b_i, by = "TMK") %>%
        left_join(b_h, by = "VPM") %>%
        left_join(b_m, by = "TNK") %>%
        group_by(month_m) %>%
        summarize(b_t = sum(V_TE005M - b_m - b_i - b_h - b_x -
            mu_hat)/(n() + lambda))
    # Predicting soil temp
    predicted_temps <- test_edx %>%
        left_join(b_i, by = "TMK") %>%
        left_join(b_h, by = "VPM") %>%
        left_join(b_m, by = "TNK") %>%
        left_join(b_x, by = "TXK") %>%
        left_join(b_t, by = "month_m") %>%
        mutate(pred = mu_hat + b_m + b_i + b_h + b_x + b_t) %>%
        pull(pred)
    return(RMSE(predicted_temps, test_edx$V_TE005M))
})

# Plot: RMSEs versus lambdas
dataframe <- data.frame(RMSE = rmses, lambdas = lambdas)
ggplot(dataframe, aes(lambdas, rmses)) + geom_point() + labs(title = "RMSEs versus
Lambdas - Regularized Linear Model") +
    labs(x = "lambdas", y = "RMSEs")
```



```
# Identifying lambda value associated with lowest RMSE
lambda_min <- lambdas[which.min(rmses)]

# Predicting RMSE on the test_edx set
reg_model <- min(rmses)
```

```r
# Expanding results dataframe
results <- results %>%
    add_row(model = "_Mean_+_TMK_+_VPM_+_TNK_+_TXK_+_month_+_Reg_ **test**",
        RMSE = reg_model)

results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121

The optimal lambda generated using the training dataset only is:

```
## [1] 0.37
```

Adding regularization to the linear model did not improve (decrease) the RMSE. Perhaps a more complex model may be more accurate.

3.6 k nearest neighbor (knn)

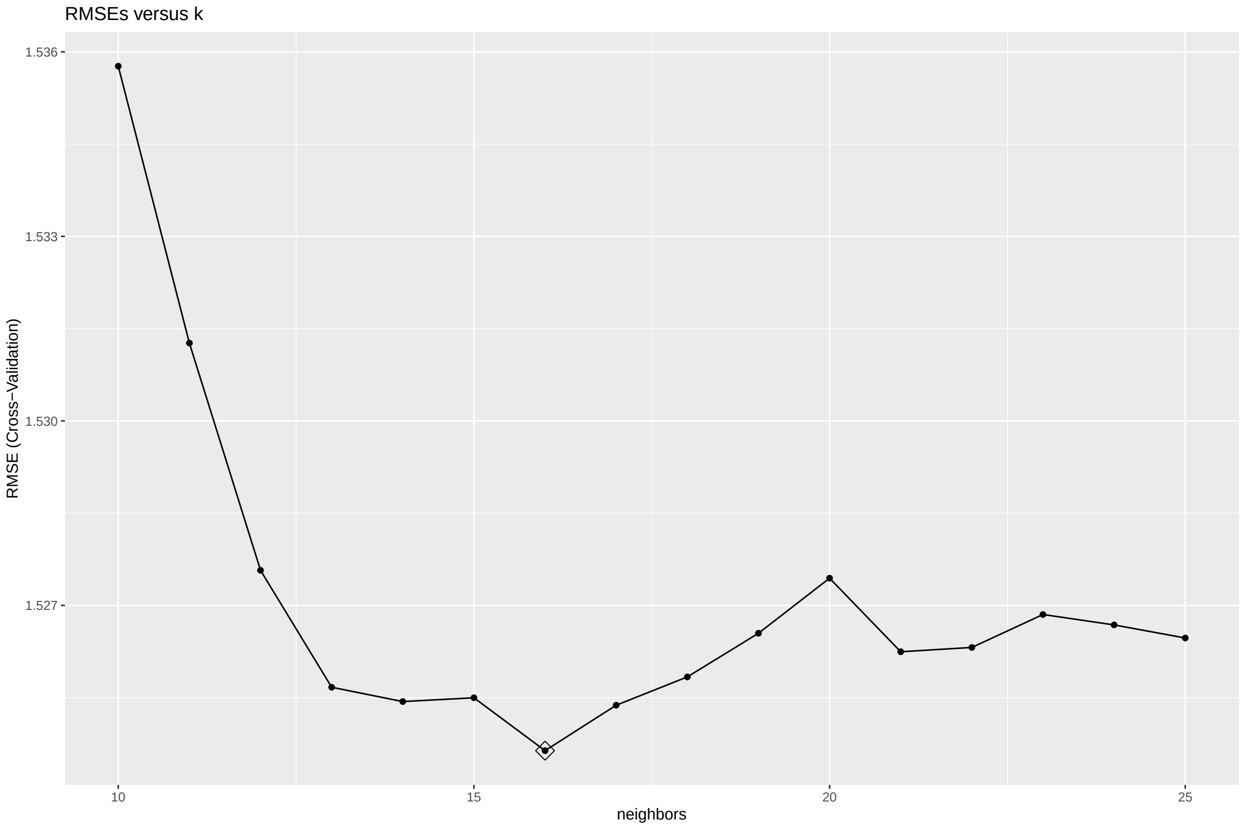
The knn concept assumes that similar observations (data points) are in close proximity to each other. First, the distance between all observations based on features will be defined. Next, for any data point for which an estimate is warranted, the k nearest data points will be assessed. In short, the knn model calculates distances between data points. The existing data points closest to a given data point by the calculated distance will become the "k-neighbors"; k stands for the number of neighbors considered. The optimal k can be determined using cross-validation.

```
model_knn <- train(V_TE005M ~ RSK + SDK + VPM + TMK + UPM + TXK +
    TNK + TGK + month_m, method = "knn", data = edx, trControl = trainControl(method =
"cv",
    number = 5, verboseIter = FALSE), tuneGrid = expand.grid(k = seq(10,
    25, 1)))

model_knn
```

```
## k-Nearest Neighbors
##
## 20717 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 16574, 16573, 16575, 16572, 16574
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##   10  1.535767  0.9579201  1.179377
##   11  1.531265  0.9581660  1.176003
##   12  1.527570  0.9583684  1.172766
##   13  1.525670  0.9584734  1.171284
##   14  1.525437  0.9584892  1.170180
##   15  1.525498  0.9584884  1.170165
##   16  1.524639  0.9585376  1.170099
##   17  1.525377  0.9585013  1.170497
##   18  1.525838  0.9584781  1.170968
##   19  1.526547  0.9584428  1.171315
##   20  1.527443  0.9583968  1.171978
##   21  1.526247  0.9584640  1.171149
##   22  1.526317  0.9584658  1.170993
##   23  1.526852  0.9584408  1.171797
##   24  1.526683  0.9584545  1.170779
##   25  1.526470  0.9584682  1.170569
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 16.
```

```
ggplot(model_knn, highlight = TRUE) + labs(title = "RMSEs versus k") +
    labs(x = "neighbors")
```



```
results <- results %>%
    add_row(model = "_knn_ **test**", RMSE =
    model_knn$results$RMSE[which.min(model_knn$results$RMSE)])
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639

The k nearest neighbor model further improved (decreased) the RMSE, as compared to the linear model. However, growing a random forest will perhaps result in a better prediction of soil temperatures.

3.7 Random forest

Next, a random forest model will be fitted using the ranger method with a 5-fold cross-validation (‘cv’; i.e., 80/20 split). The idea behind random forests is to improve shortcomings of decision trees. By averaging multiple decision trees, instability may be reduced. In contrast to R RandomForest function, the R ranger function runs in finite time (on my personal computer). Unfortunately, the number of trees cannot be determined in tuneGrid, which is why I opted for three training rounds with 250, 500, and 1,000 trees, respectively.

3.7.1 250 trees

```
model_rf_250 <- train(V_TE005M ~ RSK + SDK + VPM + TMK + UPM +
    TXK + TNK + TGK + month_m, tuneLength = 1, data = edx, method = "ranger",
    num.trees = 250, tuneGrid = expand.grid(splitrule = c("variance",
        "extratrees"), mtry = c(5:7), min.node.size = c(8:13)),
    trControl = trainControl(method = "cv", number = 5, verboseIter = FALSE))

model_rf_250
```

```
## Random Forest
##
## 20717 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 16572, 16574, 16573, 16574, 16575
## Resampling results across tuning parameters:
##
##   splitrule   mtry  min.node.size  RMSE      Rsquared   MAE
##   variance    5      8             1.246890  0.9722756  0.9535431
##   variance    5      9             1.245910  0.9723193  0.9527925
##   variance    5     10             1.246536  0.9722927  0.9535613
##   variance    5     11             1.245153  0.9723548  0.9524242
##   variance    5     12             1.246725  0.9722868  0.9533149
##   variance    5     13             1.246282  0.9723066  0.9531693
##   variance    6      8             1.248864  0.9721810  0.9549589
##   variance    6      9             1.247841  0.9722268  0.9538669
##   variance    6     10             1.248030  0.9722207  0.9543087
##   variance    6     11             1.247001  0.9722666  0.9534390
##   variance    6     12             1.247346  0.9722508  0.9541346
##   variance    6     13             1.248105  0.9722161  0.9543092
##   variance    7      8             1.250045  0.9721263  0.9564156
##   variance    7      9             1.249994  0.9721292  0.9563023
##   variance    7     10             1.249883  0.9721347  0.9549388
##   variance    7     11             1.247838  0.9722268  0.9542689
##   variance    7     12             1.249173  0.9721665  0.9555149
##   variance    7     13             1.248085  0.9722161  0.9538462
##   extratrees  5      8             1.228474  0.9731385  0.9435293
##   extratrees  5      9             1.227994  0.9731674  0.9435152
##   extratrees  5     10             1.228449  0.9731519  0.9438366
##   extratrees  5     11             1.229244  0.9731204  0.9440836
##   extratrees  5     12             1.229766  0.9730954  0.9451213
##   extratrees  5     13             1.229681  0.9731090  0.9450134
##   extratrees  6      8             1.227904  0.9731256  0.9415012
##   extratrees  6      9             1.227782  0.9731325  0.9411156
##   extratrees  6     10             1.226161  0.9732022  0.9411209
##   extratrees  6     11             1.227560  0.9731446  0.9415193
##   extratrees  6     12             1.226154  0.9732062  0.9400420
##   extratrees  6     13             1.226161  0.9732074  0.9403521
##   extratrees  7      8             1.229783  0.9730304  0.9421865
##   extratrees  7      9             1.228930  0.9730683  0.9415354
##   extratrees  7     10             1.228005  0.9731117  0.9408372
##   extratrees  7     11             1.227565  0.9731308  0.9408400
##   extratrees  7     12             1.227024  0.9731569  0.9402708
##   extratrees  7     13             1.226769  0.9731683  0.9399741
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 6, splitrule = extratrees
##  and min.node.size = 12.
```

```
results <- results %>%
    add_row(model = "_RandomForest_ *250* **test**", RMSE =
    model_rf_250$results$RMSE[which.min(model_rf_250$results$RMSE)])
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639
RandomForest 250 **test**	1.226154

3.7.2 500 trees

```
model_rf_500 <- train(V_TE005M ~ RSK + SDK + VPM + TMK + UPM +
    TXK + TNK + TGK + month_m, tuneLength = 1, data = edx, method = "ranger",
    num.trees = 500, tuneGrid = expand.grid(splitrule = c("variance",
        "extratrees"), mtry = c(5:7), min.node.size = c(8:13)),
    trControl = trainControl(method = "cv", number = 5, verboseIter = FALSE))

model_rf_500
```

```
## Random Forest
##
## 20717 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 16574, 16572, 16575, 16573, 16574
## Resampling results across tuning parameters:
##
##   splitrule   mtry  min.node.size  RMSE      Rsquared   MAE
##   variance    5      8             1.248204  0.9722245  0.9539282
##   variance    5      9             1.247374  0.9722611  0.9531022
##   variance    5     10             1.247713  0.9722484  0.9537764
##   variance    5     11             1.247927  0.9722380  0.9531837
##   variance    5     12             1.247350  0.9722664  0.9532142
##   variance    5     13             1.247844  0.9722449  0.9533106
##   variance    6      8             1.250993  0.9720953  0.9559330
##   variance    6      9             1.251099  0.9720916  0.9557359
##   variance    6     10             1.250322  0.9721260  0.9550421
##   variance    6     11             1.249583  0.9721589  0.9546924
##   variance    6     12             1.249457  0.9721654  0.9547782
##   variance    6     13             1.249393  0.9721674  0.9542270
##   variance    7      8             1.253325  0.9719913  0.9577217
##   variance    7      9             1.252473  0.9720294  0.9564827
##   variance    7     10             1.252371  0.9720344  0.9566550
##   variance    7     11             1.251651  0.9720676  0.9560211
##   variance    7     12             1.250318  0.9721276  0.9554578
##   variance    7     13             1.251389  0.9720784  0.9559861
##   extratrees  5      8             1.228487  0.9731484  0.9434024
##   extratrees  5      9             1.228422  0.9731559  0.9439691
##   extratrees  5     10             1.228100  0.9731692  0.9433897
##   extratrees  5     11             1.228014  0.9731749  0.9431200
##   extratrees  5     12             1.228956  0.9731383  0.9439104
##   extratrees  5     13             1.228956  0.9731453  0.9439344
##   extratrees  6      8             1.226261  0.9732052  0.9402407
##   extratrees  6      9             1.227281  0.9731617  0.9405606
##   extratrees  6     10             1.226738  0.9731867  0.9405965
##   extratrees  6     11             1.226152  0.9732155  0.9399677
##   extratrees  6     12             1.225301  0.9732541  0.9394129
##   extratrees  6     13             1.226303  0.9732117  0.9401251
##   extratrees  7      8             1.230062  0.9730292  0.9422086
##   extratrees  7      9             1.229379  0.9730586  0.9415880
##   extratrees  7     10             1.228433  0.9731001  0.9408153
##   extratrees  7     11             1.228098  0.9731173  0.9407945
##   extratrees  7     12             1.227713  0.9731351  0.9404466
##   extratrees  7     13             1.228301  0.9731109  0.9411348
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 6, splitrule = extratrees
##  and min.node.size = 12.
```

```
results <- results %>%
    add_row(model = "_RandomForest_ *500* **test**", RMSE =
    model_rf_500$results$RMSE[which.min(model_rf_500$results$RMSE)])
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639
RandomForest 250 **test**	1.226154
RandomForest 500 **test**	1.225301

3.7.3 1,000 trees

```
model_rf_1000 <- train(V_TE005M ~ RSK + SDK + VPM + TMK + UPM +
    TXK + TNK + TGK + month_m, tuneLength = 1, data = edx, method = "ranger",
    num.trees = 1000, tuneGrid = expand.grid(splitrule = c("variance",
        "extratrees"), mtry = c(5:7), min.node.size = c(8:13)),
    trControl = trainControl(method = "cv", number = 5, verboseIter = FALSE))

model_rf_1000
```

```
## Random Forest
##
## 20717 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 16574, 16573, 16575, 16573, 16573
## Resampling results across tuning parameters:
##
##   splitrule   mtry  min.node.size  RMSE      Rsquared   MAE
##   variance    5      8             1.246695  0.9722636  0.9535819
##   variance    5      9             1.246310  0.9722808  0.9534052
##   variance    5     10             1.245990  0.9722968  0.9530139
##   variance    5     11             1.245933  0.9722999  0.9530585
##   variance    5     12             1.246026  0.9722965  0.9530925
##   variance    5     13             1.245647  0.9723147  0.9526133
##   variance    6      8             1.248818  0.9721623  0.9550483
##   variance    6      9             1.248653  0.9721702  0.9548531
##   variance    6     10             1.248223  0.9721909  0.9543790
##   variance    6     11             1.247399  0.9722260  0.9537671
##   variance    6     12             1.247656  0.9722160  0.9539222
##   variance    6     13             1.247891  0.9722063  0.9541097
##   variance    7      8             1.250716  0.9720767  0.9563700
##   variance    7      9             1.250162  0.9721022  0.9559847
##   variance    7     10             1.249779  0.9721183  0.9553877
##   variance    7     11             1.249549  0.9721292  0.9551568
##   variance    7     12             1.249260  0.9721421  0.9548224
##   variance    7     13             1.248927  0.9721567  0.9546175
##   extratrees  5      8             1.227203  0.9731780  0.9431209
##   extratrees  5      9             1.227655  0.9731589  0.9435325
##   extratrees  5     10             1.226813  0.9731991  0.9429009
##   extratrees  5     11             1.227070  0.9731935  0.9432965
##   extratrees  5     12             1.227883  0.9731580  0.9437109
##   extratrees  5     13             1.227273  0.9731846  0.9432867
##   extratrees  6      8             1.225753  0.9731974  0.9406456
##   extratrees  6      9             1.225398  0.9732137  0.9402187
##   extratrees  6     10             1.225052  0.9732309  0.9402517
##   extratrees  6     11             1.225336  0.9732204  0.9399864
##   extratrees  6     12             1.225185  0.9732290  0.9400720
##   extratrees  6     13             1.225318  0.9732244  0.9402039
##   extratrees  7      8             1.228240  0.9730770  0.9416761
##   extratrees  7      9            1.227731  0.9730991  0.9414593
##   extratrees  7     10            1.227404  0.9731157  0.9408665
##   extratrees  7     11            1.226857  0.9731402  0.9407177
##   extratrees  7     12            1.226494  0.9731576  0.9406537
##   extratrees  7     13            1.226963  0.9731376  0.9408035
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 6, splitrule = extratrees
##  and min.node.size = 10.
```

```
results <- results %>%
    add_row(model = "_RandomForest_ *1000* **test**", RMSE =
    model_rf_1000$results$RMSE[which.min(model_rf_1000$results$RMSE)])
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639
RandomForest 250 **test**	1.226154
RandomForest 500 **test**	1.225301
RandomForest 1000 **test**	1.225053

The random forest model including 1,000 trees performed best. However, I do not expect that the RMSE decreases relevantly further by increasing the number of trees even more. Hence, I opt to apply this model to the validation dataset. However, it needs to be kept in mind that the interpretability of random forest is limited, as compared to e.g. linear models.

3.8 Final model applied to validation dataset

```
predvalues <- predict(model_rf_1000, newdata = validation)

final_rmse <- RMSE(predvalues, validation$V_TE005M)

results <- results %>%
    add_row(model = "_RandomForest_ *1000* **validation**", RMSE = RMSE(predvalues,
        validation$V_TE005M))
results %>%
    kable(format = "simple", align = "c")
```

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639
RandomForest 250 **test**	1.226154
RandomForest 500 **test**	1.225301
RandomForest 1000 **test**	1.225053
RandomForest 1000 **validation**	1.203768

4 Results

4.1 Table summary

The table below summarizes the results of this project.

model	RMSE
Mean **test**	7.493744
Mean+TMK **test**	1.741966
Mean+TMK+VPM **test**	1.743593
Mean+TMK+TNK **test**	1.743590
Mean+TMK+TNK+UPM **test**	1.722945
Mean+TMK+TNK+UPM+RSK **test**	1.724258
Mean+TMK+TNK+UPM+SDK **test**	1.733152
Mean+TMK+TNK+UPM+TXK **test**	1.716812
Mean+TMK+TNK+UPM+TXK+TGK **test**	1.718094
Mean+TMK+TNK+UPM+TXK+month **test**	1.658826
Mean+TMK+VPM+TNK+TXK+month+Reg **test**	1.801121
knn **test**	1.524639
RandomForest 250 **test**	1.226154
RandomForest 500 **test**	1.225301
RandomForest 1000 **test**	1.225053
RandomForest 1000 **validation**	1.203768

4.2 Final RMSE value

```
final_rmse
```

```
## [1] 1.203768
```

5 Conclusion

When predicting soil temperatures at 5cm depth, a random forest model provided the best result (lowest RMSE). However, this model was not capable of producing an RMSE below 1.0. Expanding the dataset with observations from other weather stations by pooling measurements, or perhaps stratifying soil temperatures may improve model performance.

6 References

https://rafalab.github.io/dsbook/
Behrendt, J., et al.: Beschreibung der Datenbasis des NKDZ. Version 3.5, Offenbach, 15.02.2011. DWD Vorschriften und Betriebsunterlagen Nr. 2 (VuB 2), Wetterschlüsselhandbuch Band D, Nov 2013. DWD Vorschriften und Betriebsunterlagen Nr. 3 (VuB 3), Beobachterhandbuch (BHB) für Wettermeldestellen des synoptisch-klimatologischen Mess- und Beobachtungsnetzes, März 2014a. DWD Vorschriften und Betriebsunterlagen Nr. 3 (VuB 3), Technikerhandbuch (THB) für Wettermeldestellen des synoptisch-klimatologischen Mess- und Beobachtungsnetzes, März 2014b.ml
https://upload.wikimedia.org/wikipedia/commons/6/6f/Locator_map_HB_%28Bremen%29_in_Germany.svg; TUBS, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons
https://extensionpublications.unl.edu/assets/pdf/g2122.pdf https://en.wikipedia.org/wiki/Pareto_principle
https://en.wikipedia.org/wiki/Enthalpy_of_fusion
https://stat-ata-asu.github.io/MachineLearningToolbox/tuning-model-parameters-to-improve-performance.html#fit-a-random-forest
https://cran.r-project.org/web/packages/ranger/ranger.pdf

7 Source

Deutscher Wetterdienst
Data basis: Deutscher Wetterdienst, averaged over individual values
https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/soil_temperature/historical/tageswerte_EB_00691_19510101_20211231_hist.zip
https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/tageswerte_KL_00691_18900101_20211231_hist.zip

8 Appendix

8.1 Transforming RMarkdown to RScript

```
# Example code knitr::purl('grafj_wetter_r_markdown.Rmd',
# documentation = 2)
```

8.2 R Version

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## crt            ucrt
## system         x86_64, mingw32
## status
## major          4
## minor          2.1
## year           2022
## month          06
## day            23
## svn rev        82513
## language       R
## version.string R version 4.2.1 (2022-06-23 ucrt)
## nickname       Funny-Looking Kid
```