Predicting The Nenana Ice Classic

Authors:

Julian Benitez Mages, Aidan Johansson, Garrin Shieh, Michael Lembck

Summary:

This project was inspired by Julian's trip to Alaska in the summer of 2015. On a tour, he passed through the small town of Nenana, located on the Tanana River. Something that makes Nenana unique is its status as the home of the Nenana Ice Classic, a contest in which betters guess the exact time that the ice on a nearby river will break. Guesses are open all winter and must be submitted by early April each year. The ice usually breaks between mid-April and early May, with the latest recorded instance occurring on May 20, 2013. Tickets cost $2.50 each, and in 2021, the twelve winners who guessed correctly split the jackpot, with each taking home more than $19,000.

For our project, we used methods learned in DS 5110 to build a model that would predict the ice breakage time, with high hopes of one day applying it to a real winter season and winning big. We found three datasets supplied by the National Oceanic and Atmospheric Administration (NOAA), which had information on temperature, precipitation (including both rain and snow), and wind. Additionally, we used data supplied by the contest organizers, which contained information about the ice breakage of each year on record and went to work.

Methods:

Two of the NOAA datasets, one consisting of daily values and the other of monthly values, came from the Nenana Municipal Airport, located in Nenana. Geographically, these datasets are ideal because they come from a very close location to the Ice Classic itself. The third dataset came from neighboring Fairbanks, a much larger town 55 miles to the northeast. We used data from Fairbanks despite its distance from Nenana because of the high amounts of missing data from the local Nenana datasets. For example, the Nenana Airport weather station had recorded data on minimum and maximum temperature dating back to the beginning of the ice classic (1917), but columns on metrics such as snowfall, rainfall, and wind speed were largely missing in years prior to the 1980s. We assumed that this was because there wasn't an obvious need to upgrade the weather station in such a remote location in the Alaskan Interior until the technology became much cheaper and more widely accessible, especially if nearby towns were able to generally fill in the gap for wider usage.

The initial format of the data wasn't ideal for our purposes of prediction and modeling, so a considerable amount of time and effort was put into tidying and organizing our datasets. Of the three datasets we found from the NOAA, one was in a monthly format and the other two were in

a daily format. Our plan was to look at all the weather history in a monthly form, so we knew we had to modify the daily data in order to integrate it with our plans.

Once we were able to obtain the data that we wanted to use, we went to work organizing and tidying it. Our plan was to finish with a final dataset with as many rows as there are years on record of the ice classic, and have all the candidate predictors be the columns. We knew that if we wanted to use monthly data, we would need an individual column for each combination of month and metric.

In order to do this, we had to aggregate each metric into a monthly format. For average metrics, such as temperature, we calculated the mean for each month, and for others such as snowfall and rainfall, we took the sum for each month of the year. We then had to pivot the data so that months were measured by columns instead of rows, leaving just the number of years as the height of the dataframe.

In addition to creating this format within our final joined dataframe, we also needed to stagger the monthly values in such a way that every value would be appropriate as a predictor for the time of the ice melting. This issue arose due to the fact that the ice breaks early in the year, so only the months preceding the melting of the ice are appropriate as predictors. To account for this issue, we decided to make it such that for any given year, the monthly values from January to May are from that year, and the values from June to December are from the year prior. This change results in the case where, for each row, representing a year, the monthly values present are those from the approximate year leading up to the ice breaking.

At the end of our data tidying, we have a dataset with just over 100 rows and columns, representing roughly a century of data and several predictor values spread over all months of the year. By joining several different datasets and improvising with values from a nearby town, we were able to develop a fairly complete picture of the weather data and ice breakage within Nenana.

It's important to note that our dataset isn't perfect for our ends, however, nor could it be. Due to the nature of the data we're using and the relatively short timeframe in which the competition has occurred, there are far fewer rows present than one would ideally have for purposes of modelling and prediction. Especially if the data is to be partitioned, this means that for a 60/20/20 split, there are literally only 60, 20, and 20 rows in each partition respectively. However, given this issue, we did the best we could to try to accumulate as much data as possible.

Results:

First, we modelled the three best individual predictors with linear regression models. Those three predictors were the average temperature in April, the maximum temperature in April, and the lowest minimum temperature in July (Shown in Figures 1 to 3). The fact that April's temperature was a top indicator for the ice breaking is intuitive as the temperature would

affect the melting rate and affect when in April or May the ice breaks. July's lowest minimum temperature being a top indicator though is less intuitive. Possibly, because July is the last month of melting in the prior summer, so it counteracts how much melting needs to happen in the winter.

To improve the linear regression model, we next ran a stepwise model selection. It would find the top individual predictors through a greedy algorithm. After 8 steps and removing an attribute with an especially high p-value, the best predictors found were: the average maximum temperature of April, average temperature in October, absolute minimum temperature of July, absolute maximum temperature of February, total rain of October, and average daily temperature of August. It is interesting how many of the values in August, July, and October are good predictors of an event in April/May. A visualization of the residuals can be found in figure 4. An exploration of the outliers will follow.

To verify our model, we ran k-fold validation next, with 5 partitions. The mean RMSE after validation was 4.714 in comparison to the stepwise's mean value of 3.992. The difference in these values can be explained by the low row count that the k-fold validation was using of 20 rows in a test partition and also by the random seed used.

Next we wanted to explore how related our attributes were through dimension reduction. We ran two iterations of dimension reduction using the classic approach of principal component analysis. One iteration was with the wind data included and the other without. It is interesting how in comparison, the dataset with wind data had two principal components while the dataset without the wind data had one principal component that stood out. (See figures 5 and 6).

Lastly we ran k-means clustering with 4 clusters (See figure 7). The data here got scaled down to only 40 rows of complete data which could mean lower quality of results as 60% of our data was removed. Diving deeper into the dates of each cluster gave no further insight unfortunately.

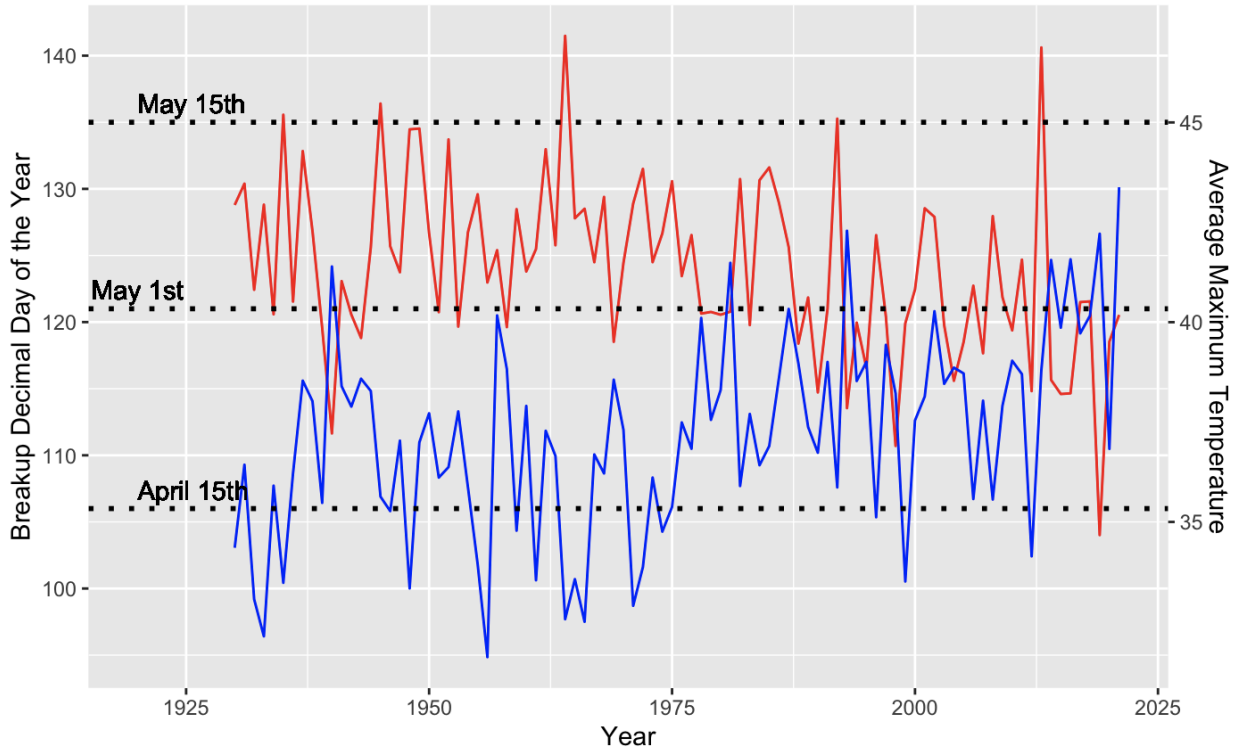
The residual plot shows that the three data points with the greatest difference between predictor and reality are marked by 104, 20, and 49, corresponding to the years 2020, 1936, and 1965. The 1936 spike is most likely explained by the North American heat and cold wave of 1936. Both extreme temperatures can be attributed to the dust bowl and were devastating to the public in addition to the Great Depression. 1965's anomaly could be explained by its March being record breakingly contrasting, with extreme lows in the beginning of the month and extreme highs at the end.

Discussion:

This project is beneficial to many people across multiple sectors. For starters, if it is able to provide a clear prediction of ice breakage, it could be used by bettors who are looking to increase their odds of winning it big. It could be published and used amongst any fans of the Nenana Ice Classic who are interested in increasing their chances. Additionally, through our data

analysis, it is clear that ice breakup times are getting progressively earlier in the year, on average. This is a trend that began around the 1980s, and will only begin to accelerate as climate change worsens.

We can see the trend in the below graph showing the average maximum temperature and the ice breakup day by year. As you can see, the red line which represents the ice breakup day is trending downwards why the blue line which is the average max temperature is trending upwards. These trends, although both slight, are a worrying sign that global warming is causing ice to break earlier in the year, potentially causing massive ramifications in the environment.



There are many aspects of the project that can be improved in future work. For example, the data we use can be taken from a different source. The data we used, while free from the NOAA, was extremely incomplete and we had to combine multiple datasets in order to make our data anywhere close to complete. We did find a dataset that has extremely complete data with multiple variables, but it cost $75 which is unreasonable for our project. Additionally, we could have used more complex regressions and methods in order to find a better predictive model. We are all undergrad students in the PlusOne program so it is understandable that we only were able to do introductory analysis discussed in class on our data. However, as we take more DS classes and go on co-op, we will certainly learn more methods that will definitely help improve on our initial analysis in this project.

Statement of Contribution:

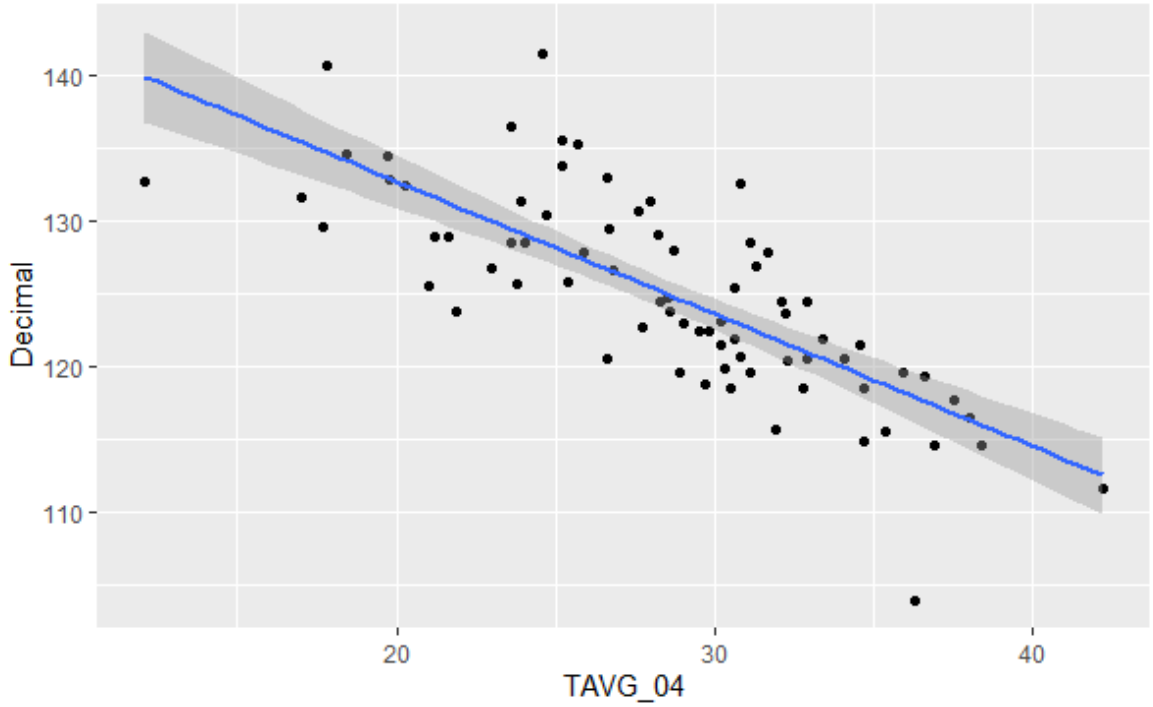
	Proposal	Presentation	Report
Julian	Summary / Proposed Plan	Background Info.	Summary, Discussion
Aidan	Proposed Plan	Issues and methods	Methods, Code Formatting
Garrin	Preliminary Results	Results pt. 1	Discussion
Michael	Preliminary Results	Results pt. 2 and Predictions	Results
All	References	Q&A	References and Appendix

References:

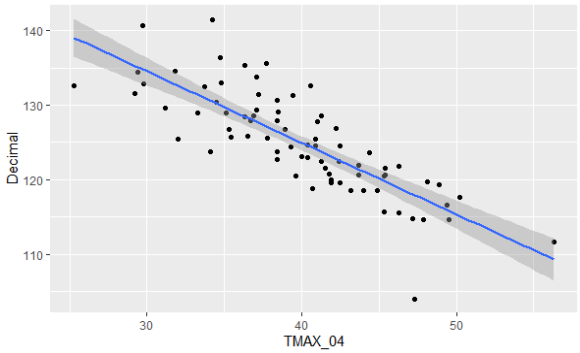
https://www.nenanaakiceclassic.com/
https://nsidc.org/data/nsidc-0064/versions/2
https://www.ncdc.noaa.gov/cdo-web/datasets/GSOM/stations/GHCND:USW00026435/detail
https://www1.ncdc.noaa.gov/pub/data/cdo/documentation/GSOM_documentation.pdf
https://weatherspark.com/h/y/145082/2014/Historical-Weather-during-2014-at-Nenana-Municipal-Airport-Alaska-United-States#Figures-Temperature

Appendix:

Figure 1:



Figures 2 and 3:



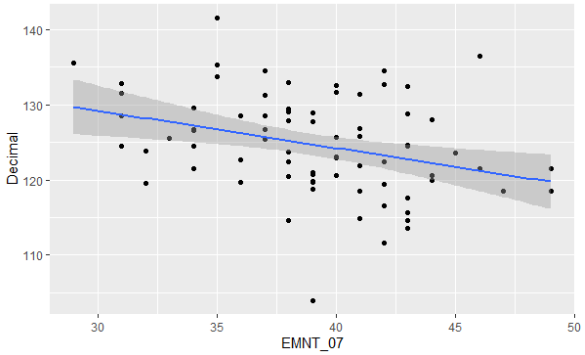
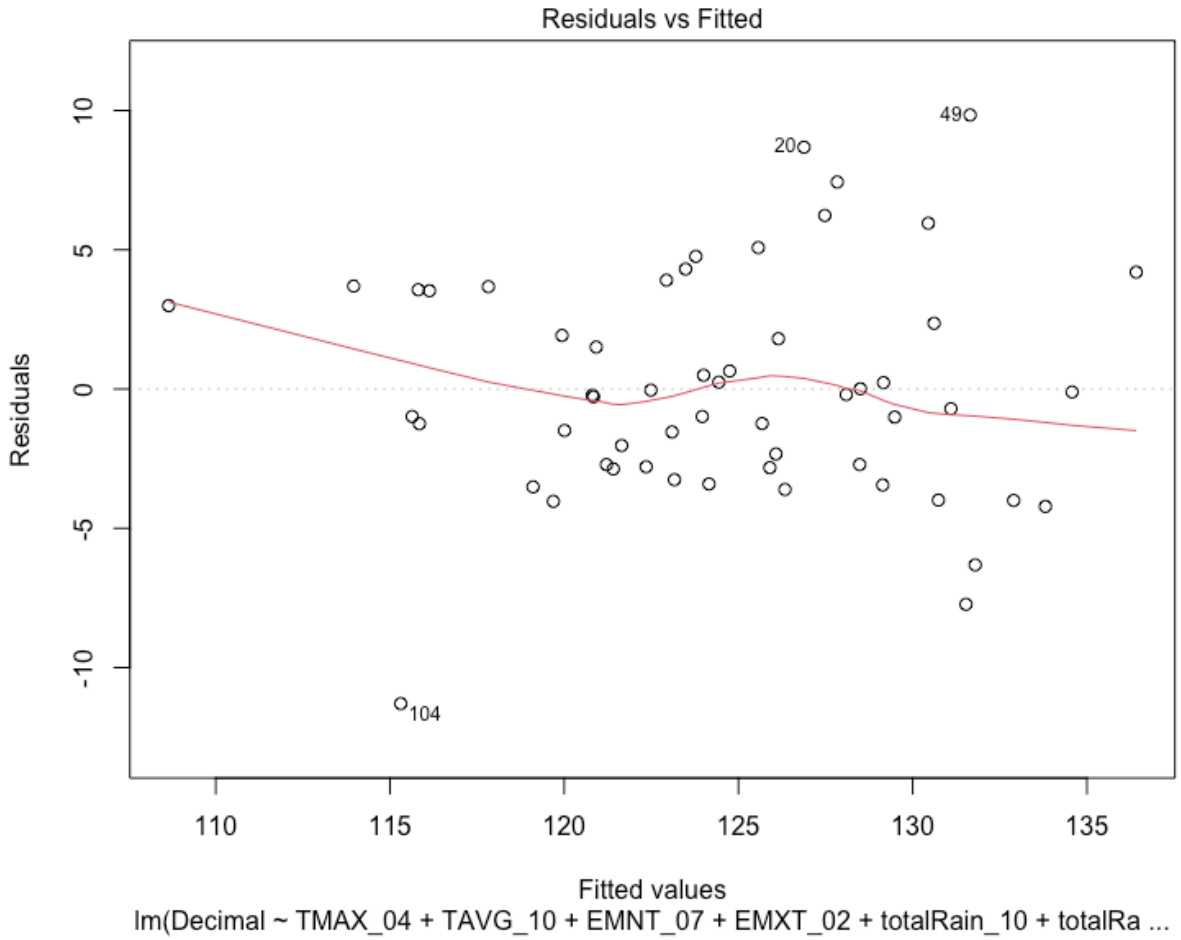


Figure 4:



Figures 5 and 6:

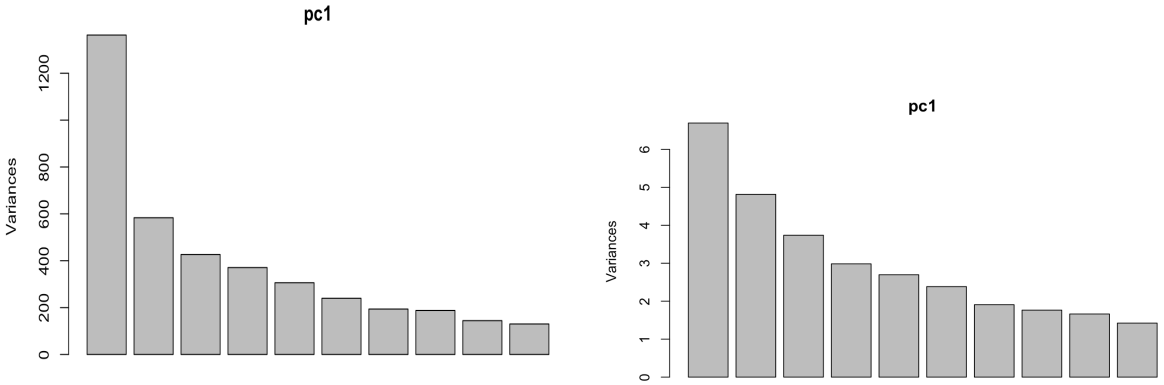
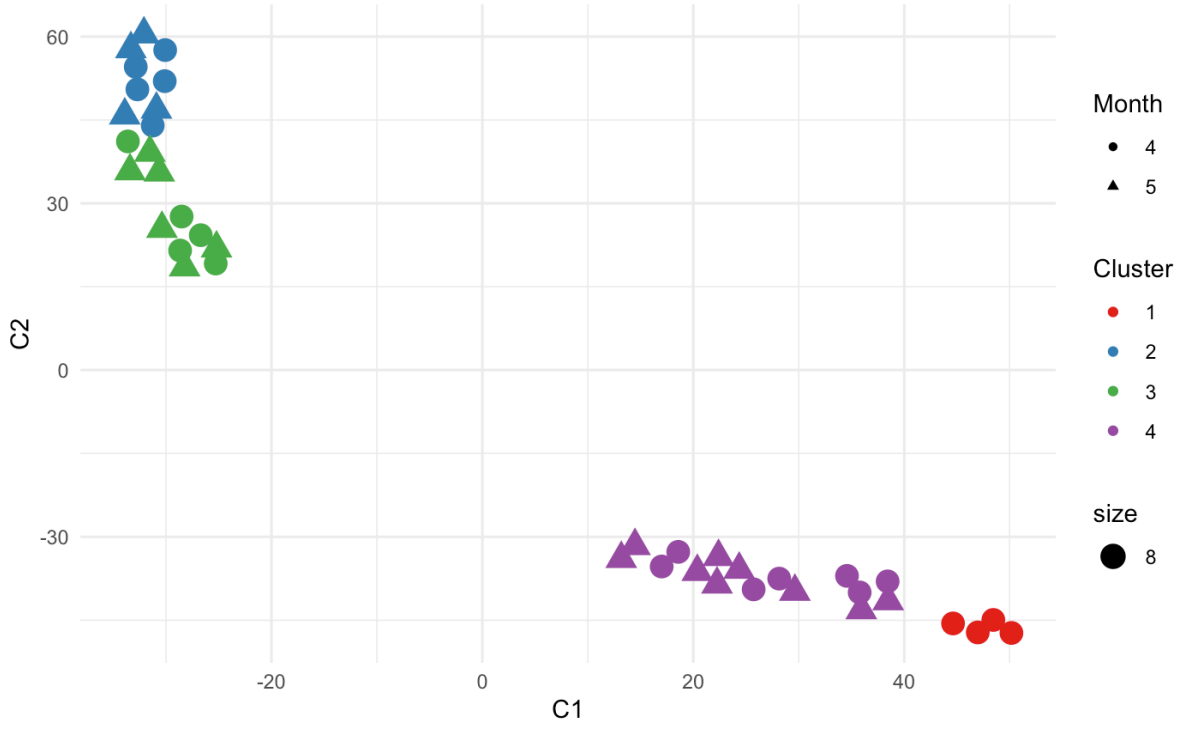


Figure 7:



For runnable code, see Rmd file.

DATA LOADING:

```
winds <- read_csv("alaska_airports_hourly_winds_PANN.csv")

winds$ts = as.Date(winds$ts, "%m/%d/%y %H:%M")

winds$month <- month(as.POSIXlt(winds$ts, format="%m/%d/%y %H:%M"))
winds$year <- year(as.POSIXlt(winds$ts, format="%m/%d/%y %H:%M"))

winds <- winds %>% group_by(month, year) %>% summarize(ws_sum = sum(ws), wd_avg =
mean(wd))

nenanaData <- read_csv("DS5110-Proposal/NenanaIceClassic_1917-2021.csv")

nenanaData <- nenanaData %>% rename("Decimal" = `Decimal Day of Year`)

nenanaData

fairbanksDailyOG <- read_csv("DS5110-Proposal/fairbanksdaily.csv")
fairbanksDaily <- fairbanksDailyOG %>%
  select(DATE, PRCP, SNOW, SNWD)
```

```
fairbanksDaily <- fairbanksDaily %>%
  separate(DATE, c("Year", "Month", "Day"), "-")
Fairbanks_Info <- fairbanksDaily %>%
  group_by(Year, Month) %>%
  summarise(totalSnow = sum(SNOW), totalRain = sum(PRCP), avgSdepth = mean(SNWD))
Fairbanks_Info <- Fairbanks_Info %>%
  pivot_wider(names_from = Month, values_from = c(totalSnow, totalRain, avgSdepth))

nenanaMonthlyOG<- read_csv("DS5110-Proposal/nenanamonthly.csv")
nenanaMonthly <- nenanaMonthlyOG %>%
  select(DATE, EMNT, EMXT, TAVG, TMAX, TMIN)
nenanaMonthly <- nenanaMonthly %>%
  separate(DATE, c("Year", "Month"), "-")
nenanaMonthly <- nenanaMonthly %>%
  pivot_wider(names_from = Month, values_from = c(EMNT, EMXT, TAVG, TMAX, TMIN))

ourData <- merge(Fairbanks_Info, nenanaMonthly,by="Year", all = TRUE)
ourData <- merge(ourData, nenanaData, by = "Year", all = TRUE)
ourData
```

Linear Regression:

```
TAVG_04_model <- lm(`Decimal` ~ TAVG_04, ourData)
summary(TAVG_04_model)

ggplot(ourData, aes(x=TAVG_04, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()

TAVG_10_model <- lm(`Decimal` ~ TAVG_10, ourData)
summary(TAVG_10_model)

ggplot(ourData, aes(x=TAVG_10, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()

TMAX_04_model <- lm(`Decimal` ~ TMAX_04, ourData)
summary(TMAX_04_model)

ggplot(ourData, aes(x=TMAX_04, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()

EMNT_07_model <- lm(`Decimal` ~ EMNT_07, ourData)
```

```
summary(EMNT_07_model)

ggplot(ourData, aes(x=EMNT_07, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()

EMXT_02_model <- lm(`Decimal` ~ EMXT_02, ourData)
summary(EMXT_02_model)

ggplot(ourData, aes(x=EMXT_02, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()

totalRain_10_model <- lm(`Decimal` ~ totalRain_10, ourData)
summary(totalRain_10_model)

ggplot(ourData, aes(x=totalRain_10, y=`Decimal`)) +
  geom_point() +
  geom_smooth(method="lm") +
  labs()
```

Stepwise:

```
model <- NULL

# Step 1
preds <- "1"
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_04", "TMAX_05", "TMAX_06",
"TMAX_07", "TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01",
"TAVG_02", "TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08",
"TAVG_09", "TAVG_10", "TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03",
"TMIN_04", "TMIN_05", "TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10",
"TMIN_11", "TMIN_12", "EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05",
"EMNT_06", "EMNT_07", "EMNT_08", "EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12",
"EMXT_01", "EMXT_02", "EMXT_03", "EMXT_04", "EMXT_05", "EMXT_06", "EMXT_07",
"EMXT_08", "EMXT_09", "EMXT_10", "EMXT_11", "EMXT_12", "avgSdepth_01",
"avgSdepth_02", "avgSdepth_03", "avgSdepth_04", "avgSdepth_05", "avgSdepth_06",
"avgSdepth_07", "avgSdepth_08", "avgSdepth_09", "avgSdepth_10", "avgSdepth_11",
"avgSdepth_12", "totalRain_01", "totalRain_02", "totalRain_03", "totalRain_04",
"totalRain_05", "totalRain_06", "totalRain_07", "totalRain_08", "totalRain_09",
"totalRain_10", "totalRain_11", "totalRain_12", "totalSnow_01", "totalSnow_02",
"totalSnow_03", "totalSnow_04", "totalSnow_05", "totalSnow_06", "totalSnow_07",
"totalSnow_08", "totalSnow_09", "totalSnow_10", "totalSnow_11", "totalSnow_12")

s1 <- step1("Decimal", preds, cands, ourData_part)
```

```
model <- c(model, attr(s1, "best"))
s1

# Step 2
preds <- "TMAX_04"
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_10", "TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04",
"TMIN_05", "TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11",
"TMIN_12", "EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06",
"EMNT_07", "EMNT_08", "EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01",
"EMXT_02", "EMXT_03", "EMXT_04", "EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08",
"EMXT_09", "EMXT_10", "EMXT_11", "EMXT_12", "avgSdepth_01", "avgSdepth_02",
"avgSdepth_03", "avgSdepth_04", "avgSdepth_05", "avgSdepth_06", "avgSdepth_07",
"avgSdepth_08", "avgSdepth_09", "avgSdepth_10", "avgSdepth_11", "avgSdepth_12",
"totalRain_01", "totalRain_02", "totalRain_03", "totalRain_04", "totalRain_05",
"totalRain_06", "totalRain_07", "totalRain_08", "totalRain_09", "totalRain_10",
"totalRain_11", "totalRain_12", "totalSnow_01", "totalSnow_02", "totalSnow_03",
"totalSnow_04", "totalSnow_05", "totalSnow_06", "totalSnow_07", "totalSnow_08",
"totalSnow_09", "totalSnow_10", "totalSnow_11", "totalSnow_12")

s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 3
preds <- c("TMAX_04", "TAVG_10")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05",
"TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12",
"EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_07",
"EMNT_08", "EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_02",
"EMXT_03", "EMXT_04", "EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09",
"EMXT_10", "EMXT_11", "avgSdepth_01", "avgSdepth_02", "avgSdepth_03",
"avgSdepth_04", "avgSdepth_05", "avgSdepth_06", "avgSdepth_07", "avgSdepth_08",
"avgSdepth_09", "avgSdepth_10", "avgSdepth_11", "avgSdepth_12", "totalRain_01",
"totalRain_02", "totalRain_03", "totalRain_04", "totalRain_05", "totalRain_06",
"totalRain_07", "totalRain_08", "totalRain_09", "totalRain_10", "totalRain_11",
"totalRain_12", "totalSnow_01", "totalSnow_02", "totalSnow_03", "totalSnow_04",
"totalSnow_05", "totalSnow_06", "totalSnow_07", "totalSnow_08", "totalSnow_09",
"totalSnow_10", "totalSnow_11", "totalSnow_12")
```

```
s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 4
preds <- c("TMAX_04", "TAVG_10", "EMNT_07")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05",
"TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12",
"EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_08",
"EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_02", "EMXT_03",
"EMXT_04", "EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09", "EMXT_10",
"EMXT_11", "avgSdepth_01", "avgSdepth_02", "avgSdepth_03", "avgSdepth_04",
"avgSdepth_05", "avgSdepth_06", "avgSdepth_07", "avgSdepth_08", "avgSdepth_09",
"avgSdepth_10", "avgSdepth_11", "avgSdepth_12", "totalRain_01", "totalRain_02",
"totalRain_03", "totalRain_04", "totalRain_05", "totalRain_06", "totalRain_07",
"totalRain_08", "totalRain_09", "totalRain_10", "totalRain_11", "totalRain_12",
"totalSnow_01", "totalSnow_02", "totalSnow_03", "totalSnow_04", "totalSnow_05",
"totalSnow_06", "totalSnow_07", "totalSnow_08", "totalSnow_09", "totalSnow_10",
"totalSnow_11", "totalSnow_12")

s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 5
preds <- c("TMAX_04", "TAVG_10", "EMNT_07", "EMXT_02")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05",
"TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12",
"EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_08",
"EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_03", "EMXT_04",
"EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09", "EMXT_10", "EMXT_11",
"avgSdepth_01", "avgSdepth_02", "avgSdepth_03", "avgSdepth_04", "avgSdepth_05",
"avgSdepth_06", "avgSdepth_07", "avgSdepth_08", "avgSdepth_09", "avgSdepth_10",
"avgSdepth_11", "avgSdepth_12", "totalRain_01", "totalRain_02", "totalRain_03",
"totalRain_04", "totalRain_05", "totalRain_06", "totalRain_07", "totalRain_08",
"totalRain_09", "totalRain_10", "totalRain_11", "totalRain_12", "totalSnow_01",
"totalSnow_02", "totalSnow_03", "totalSnow_04", "totalSnow_05", "totalSnow_06",
"totalSnow_07", "totalSnow_08", "totalSnow_09", "totalSnow_10", "totalSnow_11",
"totalSnow_12")
```

```
s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 6
preds <- c("TMAX_04", "TAVG_10", "EMNT_07", "EMXT_02", "totalRain_10")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05",
"TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12",
"EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_08",
"EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_03", "EMXT_04",
"EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09", "EMXT_10", "EMXT_11",
"avgSdepth_01", "avgSdepth_02", "avgSdepth_03", "avgSdepth_04", "avgSdepth_05",
"avgSdepth_06", "avgSdepth_07", "avgSdepth_08", "avgSdepth_09", "avgSdepth_10",
"avgSdepth_11", "avgSdepth_12", "totalRain_01", "totalRain_02", "totalRain_03",
"totalRain_04", "totalRain_05", "totalRain_06", "totalRain_07", "totalRain_08",
"totalRain_09", "totalRain_11", "totalRain_12", "totalSnow_01", "totalSnow_02",
"totalSnow_03", "totalSnow_04", "totalSnow_05", "totalSnow_06", "totalSnow_07",
"totalSnow_08", "totalSnow_09", "totalSnow_10", "totalSnow_11", "totalSnow_12")

s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 7
preds <- c("TMAX_04", "TAVG_10", "EMNT_07", "EMXT_02", "totalRain_10",
"totalRain_12")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_08", "TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02",
"TAVG_03", "TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09",
"TAVG_11", "TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05",
"TMIN_06", "TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12",
"EMNT_01", "EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_08",
"EMNT_09", "EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_03", "EMXT_04",
"EMXT_05", "EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09", "EMXT_10", "EMXT_11",
"avgSdepth_01", "avgSdepth_02", "avgSdepth_03", "avgSdepth_04", "avgSdepth_05",
"avgSdepth_06", "avgSdepth_07", "avgSdepth_08", "avgSdepth_09", "avgSdepth_10",
"avgSdepth_11", "avgSdepth_12", "totalRain_01", "totalRain_02", "totalRain_03",
"totalRain_04", "totalRain_05", "totalRain_06", "totalRain_07", "totalRain_08",
"totalRain_09", "totalRain_11", "totalSnow_01", "totalSnow_02", "totalSnow_03",
"totalSnow_04", "totalSnow_05", "totalSnow_06", "totalSnow_07", "totalSnow_08",
"totalSnow_09", "totalSnow_10", "totalSnow_11", "totalSnow_12")
```

```
s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

# Step 8
preds <- c("TMAX_04", "TAVG_10", "EMNT_07", "EMXT_02", "totalRain_10",
"totalRain_12", "TMAX_08")
cands <- c("TMAX_01", "TMAX_02", "TMAX_03", "TMAX_05", "TMAX_06", "TMAX_07",
"TMAX_09", "TMAX_10", "TMAX_11", "TMAX_12", "TAVG_01", "TAVG_02", "TAVG_03",
"TAVG_04", "TAVG_05", "TAVG_06", "TAVG_07", "TAVG_08", "TAVG_09", "TAVG_11",
"TAVG_12", "TMIN_01", "TMIN_02", "TMIN_03", "TMIN_04", "TMIN_05", "TMIN_06",
"TMIN_07", "TMIN_08", "TMIN_09", "TMIN_10", "TMIN_11", "TMIN_12", "EMNT_01",
"EMNT_02", "EMNT_03", "EMNT_04", "EMNT_05", "EMNT_06", "EMNT_08", "EMNT_09",
"EMNT_10", "EMNT_11", "EMNT_12", "EMXT_01", "EMXT_03", "EMXT_04", "EMXT_05",
"EMXT_06", "EMXT_07", "EMXT_08", "EMXT_09", "EMXT_10", "EMXT_11", "avgSdepth_01",
"avgSdepth_02", "avgSdepth_03", "avgSdepth_04", "avgSdepth_05", "avgSdepth_06",
"avgSdepth_07", "avgSdepth_08", "avgSdepth_09", "avgSdepth_10", "avgSdepth_11",
"avgSdepth_12", "totalRain_01", "totalRain_02", "totalRain_03", "totalRain_04",
"totalRain_05", "totalRain_06", "totalRain_07", "totalRain_08", "totalRain_09",
"totalRain_11", "totalSnow_01", "totalSnow_02", "totalSnow_03", "totalSnow_04",
"totalSnow_05", "totalSnow_06", "totalSnow_07", "totalSnow_08", "totalSnow_09",
"totalSnow_10", "totalSnow_11", "totalSnow_12")

s1 <- step1("Decimal", preds, cands, ourData_part)

model <- c(model, attr(s1, "best"))
s1

step_model <- tibble(index=seq_along(model),
                     variable=factor(names(model), levels=names(model)),
                     RMSE=model)

ggplot(step_model, aes(y=RMSE)) +
  geom_point(aes(x=variable)) +
  geom_line(aes(x=index)) +
  labs(title="Stepwise model selection") +
  theme_minimal() +
  coord_flip()

best_fit <- lm(Decimal
               ~ TMAX_04 +TAVG_10 + EMNT_07 + EMXT_02 + totalRain_10 + TMAX_08,
data = ourData_part$test)
rmse(best_fit, ourData_part$test)

summary(best_fit)
```

```
plot(best_fit)
```

K-fold:

```
set.seed(2)
model_cv <- crossv_kfold(ourData, 5)

cv_t1 <- model_cv %>%
  mutate(fit = purrr::map(train,
                          ~ lm(Decimal
               ~ TMAX_04 +TAVG_10 + EMNT_07 + EMXT_02 + totalRain_10 + TMAX_08,
                               data = .)),
         rmse = purrr::map2_dbl(fit, test, ~ rmse(.x, .y)))

mean(cv_t1$rmse)
summary(cv_t1)


PCA:
d <- subset(dataWithWind, select = -c(Time))

d <- d[ , colSums(is.na(d)) == 0]

pc1 <- prcomp(na.omit(d[ , which(apply(na.omit(d), 2, var) != 0)]), scale.=TRUE)

summary(pc1)

plot(pc1)

as_tibble(pc1$x) %>%
  ggplot(aes(x=PC1, y=PC2)) +
  geom_point() +
  scale_color_brewer(palette="Dark2") +
  labs(color="Landscape") +
  theme_minimal()
```

KMEANS

```
set.seed(1)

library(Rtsne)

simplified <- na.omit(d[ , which(apply(na.omit(d), 2, var) != 0)])

copy <- simplified
```

```
km <- kmeans(copy, centers=4)

km_cl <- factor(km$cluster)

set.seed(3)
tsne2 <- Rtsne(copy, perplexity=7)

colnames(tsne2$Y) <- c("C1", "C2")

tc2 <- as_tibble(tsne2$Y)

simplified$cluster <- km$cluster
simplified$C1 <- tc2$C1
simplified$C2 <- tc2$C2

simplified$Month <- as.factor(simplified$Month)

ggplot(simplified, aes(x=C1, y=C2, color=km_cl, shape=Month, size=8)) +
  geom_point() +
  scale_color_brewer(palette="Set1") +
  labs(shape="Month", color="Cluster") +
  theme_minimal()

ks <- 2:10

tot_within_ss <- sapply(ks, function(k) {
    cl <- kmeans(na.omit(d[ , which(apply(na.omit(d), 2, var) != 0)]), k, nstart =
10)
    cl$tot.withinss
})
plot(ks, tot_within_ss, type = "b",
     main = "Selection of # of clusters for satellite data",
     ylab = "Total within squared distances",
     xlab = "Values of k tested")
abline(v=4, col="green", lty=2)
```