Statistical Computing - Exercises 04 - Ithaca Weather

The US government, via its agencies NOAA and NCEI, maintains the US Climate Reference Network, a collection of 139 high quality weather stations whose purpose is to monitor atmospheric and soil conditions. One of them is located right here outside of Ithaca, along Route 38.

We will be formatting and analyzing the hourly Ithaca data from 2021 and 2022, with the goal of building a model for predicting afternoon temperatures.

1. Read the dataset `datasets/CRNH0203-2021-NY_Ithaca_13_E.txt` into R as a data frame and add column names according to the **documentation**. Make sure you inspect your data frame and do any necessary cleaning.

```
# Putting read.table & strsplit on separate lines for readability creates a
#knitting error. All other lines formatted.

dat <- read.table("C:/Users/Nick/Documents/GitHub/statcomp2023/datasets/CRNH0203-2021-NY_Ithaca_13_E.tx
columns = strsplit("WBANNO UTC_DATE UTC_TIME LST_DATE LST_TIME CRX_VN LONGITUDE LATITUDE T_CALC T_HR_AV

colnames(dat) <- columns[[1]]



for (i in columns){

   holder = dat[i]
   holder = replace(holder,(dat[i] == -999.0), NA )
   holder = replace(holder,(dat[i] == -9999.0), NA )
   holder = replace(holder,(dat[i] == -99999.0), NA )

   dat[i] = holder
}
```

2. Save the data frame to your computer using both `save` and `saveRDS`. What are the two file sizes, and how do the sizes compare to the size of the original .txt file? (You can check this in the terminal with the command `ls -lah`)

```
saveRDS(dat, file = "C:/Users/Nick/Documents/weatherdat.rds")

save(dat, file = "C:/Users/Nick/Documents/weatherdat.RData")
```

The saved files are 204 KB, which is much smaller than the 2096 KB text document.

3. Create a data frame containing only the longitude and latitude. Save the result as both a .csv and a .RData file. How do the sizes compare? Is this surprising to you? Can you explain the results?

```
dat1 = dat[, c("LONGITUDE", "LATITUDE")]

save(dat1, file = "C:/Users/Nick/Documents/weatherdatlonglat.RData")
save(dat1, file = "C:/Users/Nick/Documents/weatherdatlonglat2.csv")
write.csv(dat1, file = "C:/Users/Nick/Documents/weatherdatlonglat.csv")
```

Both save files are only 1 KB, this is surprising because the Rdata file should be smaller due to compression. This is likely happening because the stored data is still being compressed, and then sent to csv when using the save function. When you directly write to a csv file, the size is much larger (179 kb). This is because write.csv does not undergo compression.

The Rdata file is only 1 KB while the csv file is 179 KB. This is because RData files undergo compression while csv files only separate the values with a comma.

4. Create a new data frame that has one row for each day. Its columns should be the date, 3pm air temperature (use local standard time for all times and T_CALC for all air temperatures), 6am temperature, 7am temperature, 8am temperature, and 9am temperature.

```
dat2 <- dat[(dat["LST_TIME"] == 600) | (dat["LST_TIME"] == 700) |
               (dat["LST_TIME"] == 800)
            | (dat["LST_TIME"] == 900) | (dat["LST_TIME"] == 1500) ,
            c("T_CALC","LST_TIME","LST_DATE")]

date <- unique(dat2["LST_DATE"])
sixAM <- dat2[dat2["LST_TIME"]==600,"T_CALC"]
sevenAM <- dat2[dat2["LST_TIME"]==700,"T_CALC"]
eightAM <- dat2[dat2["LST_TIME"]==800,"T_CALC"]
nineAM <- dat2[dat2["LST_TIME"]==900,"T_CALC"]
threePM <- dat2[dat2["LST_TIME"]==1500,"T_CALC"]

newdat<- data.frame(date,sixAM,sevenAM,eightAM,nineAM,threePM,
                    row.names = 1:365)
head(newdat)
```

```
##   LST_DATE sixAM sevenAM eightAM nineAM threePM
## 1 20210101  -1.9    -3.2    -2.6   -2.9    -0.2
## 2 20210102   2.6     4.0     4.4    4.3    -0.3
## 3 20210103  -2.8    -3.2    -2.1   -1.7    -0.3
## 4 20210104  -0.1    -0.2    -0.2    0.0    -0.2
## 5 20210105  -1.4    -1.7    -1.4   -1.4     0.5
## 6 20210106  -2.5    -2.2    -1.9   -1.6    -1.0
```

5. Do a multiple regression of 3pm temperature on 6am, 7am, 8am, and 9am temperature and print out the summary. How accurate does the model say your predictions should be?

```
m1 <- lm(newdat$threePM ~ newdat$sixAM + newdat$sevenAM + newdat$eightAM +
            newdat$nineAM)
summary(m1)
```

```
##
## Call:
## lm(formula = newdat$threePM ~ newdat$sixAM + newdat$sevenAM +
##     newdat$eightAM + newdat$nineAM)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5107  -1.6942   0.0618   1.8453   8.6913
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      2.36547    0.24206   9.772  < 2e-16 ***
## newdat$sixAM     0.01469    0.15322   0.096    0.924    
## newdat$sevenAM   0.01311    0.27291   0.048    0.962    
## newdat$eightAM  -1.18764    0.27042  -4.392 1.48e-05 ***
## newdat$nineAM    2.08588    0.14494  14.392  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.027 on 357 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.9237, Adjusted R-squared:  0.9229 
## F-statistic:  1081 on 4 and 357 DF,  p-value: < 2.2e-16
```

With an F-statistic of 1081 and a p-value less than 2.2e-16, we can reject a null hypothesis of there being no relation between 6-9 am temps and 3 pm temp. Since the residual standard error is 3.027, the model claims that predictions will be within 3.027 degrees of observed ~68% (1 sd) of the time, and within 6.054 degrees of observed ~95% (2 sd) of the time.

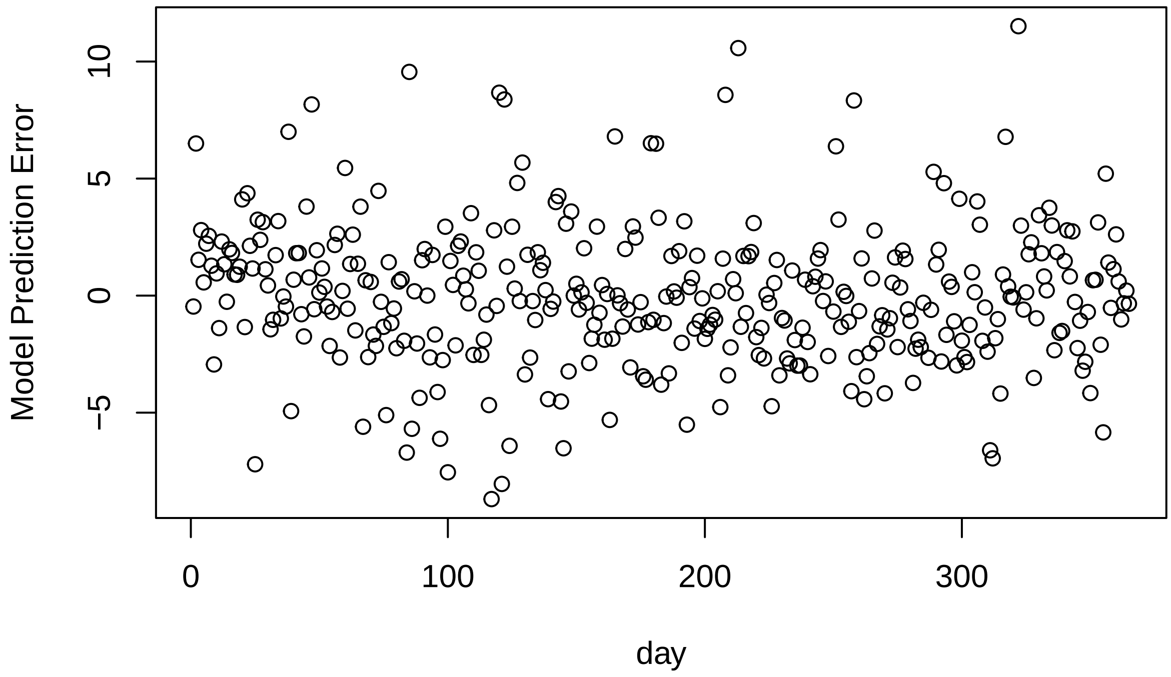
6. Use your model to predict 2022 3pm temperatures from morning temperatures. You'll have to read in the 2022 data and organize it the same way you organized the 2021 data. Report the root mean squared prediction error and make a plot of the model's prediction errors against day of year.

```
dat_ <- read.table("C:/Users/Nick/Documents/GitHub/statcomp2023/datasets/CRNH0203-2022-NY_Ithaca_13_E.t
columns = strsplit("WBANNO UTC_DATE UTC_TIME LST_DATE LST_TIME CRX_VN LONGITUDE LATITUDE T_CALC T_HR_AV

colnames(dat_) <- columns[[1]]

holder = dat$T_CALC
holder = replace(holder,(dat$T_CALC == -9999.0), NA )

dat_$T_CALC = holder

dat2_ <- dat_[(dat_["LST_TIME"] == 600) | (dat_["LST_TIME"] == 700) |
                 (dat_["LST_TIME"] == 800)
            | (dat_["LST_TIME"] == 900) | (dat_["LST_TIME"] == 1500) ,
            c("T_CALC","LST_TIME","LST_DATE")]
#dat2

date <- unique(dat2_["LST_DATE"])
sixAM <- dat2_[dat2_["LST_TIME"]==600,"T_CALC"]
sevenAM <- dat2_[dat2_["LST_TIME"]==700,"T_CALC"]
eightAM <- dat2_[dat2_["LST_TIME"]==800,"T_CALC"]
nineAM <- dat2_[dat2_["LST_TIME"]==900,"T_CALC"]
threePM <- dat2_[dat2_["LST_TIME"]==1500,"T_CALC"]

newdat_<- data.frame(date,sixAM,sevenAM,eightAM,nineAM,threePM,
                     row.names = 1:365)

predictions <- predict(m1, newdata = newdat_)

perrors = predictions-threePM
predictionerrors = abs(predictions-threePM)
```

```
rmse = sqrt(sum((predictionerrors[(is.na(predictionerrors)==F)])^2)/365)
rmse
```

```
## [1] 2.99401
```

```
plot(1:365, perrors, xlab = "day", ylab = "Model Prediction Error")
```



The model rmse is 2.9940099

7. Build your own model for 3pm temperatures based on the 2021 data. You can use any information in the data, and any kind of model you like, but you must use only information that's available at the 9am hour or earlier (e.g. you cannot use today's 2pm temperature to predict today's 3pm temperature).

```
dat2 <- dat[ (dat["LST_TIME"] == 500) |(dat["LST_TIME"] == 700) |
                (dat["LST_TIME"] == 800)
            | (dat["LST_TIME"] == 900) | (dat["LST_TIME"] == 1500) , ]

date <- unique(dat2["LST_DATE"])
minsunrise = dat2[dat2["LST_TIME"]==500,"T_MIN"]
max9am = dat2[dat2["LST_TIME"]==900,"T_MAX"]
fourhourtempchange = max9am-minsunrise
sevenAM <- dat2[dat2["LST_TIME"]==700,"T_CALC"]
eightAM <- dat2[dat2["LST_TIME"]==800,"T_CALC"]
nineAM <- dat2[dat2["LST_TIME"]==900,"T_CALC"]
threePM <- dat2[dat2["LST_TIME"]==1500,"T_CALC"]
```

```
sevenAMrain <- dat2[dat2["LST_TIME"]==700,"P_CALC"]
eightAMrain <- dat2[dat2["LST_TIME"]==800,"P_CALC"]
nineAMrain <- dat2[dat2["LST_TIME"]==900,"P_CALC"]

nineAMSOLARAD <- dat2[dat2["LST_TIME"]==900,"SOLARAD"]
nineAMSoil <- dat2[dat2["LST_TIME"]==900,"SOIL_TEMP_5"]

rain <- sevenAMrain + eightAMrain + nineAMrain

newdat<- data.frame(date,sevenAM,eightAM,nineAM,fourhourtempchange, rain ,
                    nineAMSOLARAD, nineAMSoil ,threePM, row.names = 1:365)


m1 <- lm(newdat$threePM ~ newdat$fourhourtempchange + newdat$sevenAM +
            newdat$eightAM + newdat$nineAM + newdat$sevenAM*newdat$nineAM
         + newdat$rain + newdat$nineAMSOLARAD + newdat$nineAMSoil)
summary(m1)
```

```
##
## Call:
## lm(formula = newdat$threePM ~ newdat$fourhourtempchange + newdat$sevenAM +
##     newdat$eightAM + newdat$nineAM + newdat$sevenAM * newdat$nineAM +
##     newdat$rain + newdat$nineAMSOLARAD + newdat$nineAMSoil)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9768  -1.6592   0.0205   1.7962   8.7325
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.4309579  0.3492626   4.097 5.20e-05 ***
## newdat$fourhourtempchange     0.0832973  0.1020906   0.816  0.41510
## newdat$sevenAM                0.0997916  0.2191197   0.455  0.64909
## newdat$eightAM               -1.1951445  0.2717730  -4.398 1.45e-05 ***
## newdat$nineAM                 1.8887454  0.1749438  10.796  < 2e-16 ***
## newdat$rain                   0.0493084  0.0858912   0.574  0.56628
## newdat$nineAMSOLARAD          0.0009909  0.0014481   0.684  0.49425
## newdat$nineAMSoil             0.2386615  0.0643045   3.711  0.00024 ***
## newdat$sevenAM:newdat$nineAM -0.0026881  0.0016723  -1.607  0.10887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.992 on 350 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.9264, Adjusted R-squared:  0.9247
## F-statistic: 550.5 on 8 and 350 DF,  p-value: < 2.2e-16
```

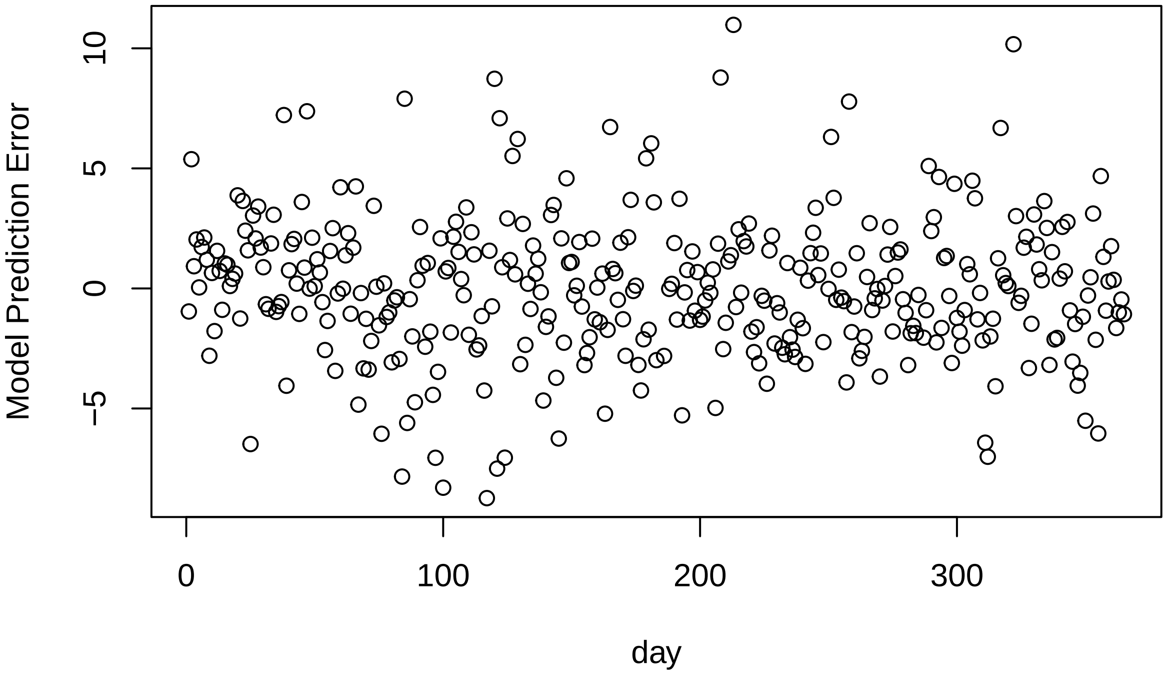
8. Test out your model on the 2022 data. Report the root mean squared prediction error and make a plot of the model's prediction errors against day of year.

```
dat2_ <- dat_[(dat_["LST_TIME"] == 500) | (dat_["LST_TIME"] == 700) |
                 (dat_["LST_TIME"] == 800)
```

```
            | (dat_["LST_TIME"] == 900) | (dat_["LST_TIME"] == 1500) ,]
#dat2

date <- unique(dat2_["LST_DATE"])
minsunrise = dat2_[dat2_["LST_TIME"]==500,"T_MIN"]
max9am = dat2_[dat2_["LST_TIME"]==900,"T_MAX"]
fourhourtempchange = max9am-minsunrise
sevenAM <- dat2_[dat2_["LST_TIME"]==700,"T_CALC"]
eightAM <- dat2_[dat2_["LST_TIME"]==800,"T_CALC"]
nineAM <- dat2_[dat2_["LST_TIME"]==900,"T_CALC"]
threePM <- dat2_[dat2_["LST_TIME"]==1500,"T_CALC"]

sevenAMrain <- dat2_[dat2_["LST_TIME"]==700,"P_CALC"]
eightAMrain <- dat2_[dat2_["LST_TIME"]==800,"P_CALC"]
nineAMrain <- dat2_[dat2_["LST_TIME"]==900,"P_CALC"]

nineAMSOLARAD <- dat2_[dat2_["LST_TIME"]==900,"SOLARAD"]
nineAMSoil <- dat2_[dat2_["LST_TIME"]==900,"SOIL_TEMP_5"]

newdat_<- data.frame(date,sevenAM,eightAM,nineAM,fourhourtempchange, rain ,
                     nineAMSOLARAD, nineAMSoil ,threePM, row.names = 1:365)

predictions <- predict(m1, newdata = newdat_)

perrors = predictions-threePM
predictionerrors = abs(predictions-threePM)
rmse = sqrt(sum((predictionerrors[(is.na(predictionerrors)==F)])^2)/365)
rmse
```

```
## [1] 2.929946
```

```
plot(1:365, perrors, xlab = "day", ylab = "Model Prediction Error")
```



The model rmse is 2.9299455