Linear Regression

Naomi Zilber

18 February 2023

Overview

Linear regression is a supervised regression technique in which the target is a real number variable and the predictors could be any combination of quantitative or qualitative variables. In regression, the target variable is quantitative. Some strengths of linear regression are:

- It is a relatively simple and intuitive algorithm
- It works well if the data has a linear pattern
- It has low variance

Some weaknesses are the fact that linear regression has a high bias because it assumes that there is a linear relationship between the target and the predictors (that the data has a linear shape), and thereby tend to underfit the data.

The data set used in this notebook is from this link. (https://www.kaggle.com/datasets/muthuj7/weather-dataset)

Load data

Read in the data of weatherHistory.

```
df <- read.csv("weatherHistory.csv", header=TRUE)
str(df)
```

```
## 'data.frame':    96453 obs. of  12 variables:
##  $ Formatted.Date          : chr  "2006-04-01 00:00:00.000 +0200" "2006-04-0
1 01:00:00.000 +0200" "2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000
+0200" ...
##  $ Summary                 : chr  "Partly Cloudy" "Partly Cloudy" "Mostly Cl
oudy" "Partly Cloudy" ...
##  $ Precip.Type             : chr  "rain" "rain" "rain" "rain" ...
##  $ Temperature..C.         : num  9.47 9.36 9.38 8.29 8.76 ...
##  $ Apparent.Temperature..C.: num  7.39 7.23 9.38 5.94 6.98 ...
##  $ Humidity                : num  0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.
82 0.72 ...
##  $ Wind.Speed..km.h.       : num  14.12 14.26 3.93 14.1 11.04 ...
##  $ Wind.Bearing..degrees.  : num  251 259 204 269 259 258 259 260 259 27
9 ...
##  $ Visibility..km.         : num  15.8 15.8 15 15.8 15.8 ...
##  $ Loud.Cover              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pressure..millibars.    : num  1015 1016 1016 1016 1017 ...
##  $ Daily.Summary           : chr  "Partly cloudy throughout the day." "Partl
y cloudy throughout the day." "Partly cloudy throughout the day." "Partly cloud
y throughout the day." ...
```

Data cleaning

The only data I need is the quantitative data, so I reduce the data frame df to only the date, temperature, apparent temperature, humidity, and wind speed. I left the date for the data exploration and made it a Date object.

```
df <- df[c(1,4,5,6,7)]
names(df)[1] <- "formatted_date"
names(df)[2] <- "temperature"
names(df)[3] <- "apparent_temperature"
names(df)[5] <- "wind_speed_kmh"

df$formatted_date <- as.Date(df$formatted_date)
str(df)
```

```
## 'data.frame':    96453 obs. of  5 variables:
##  $ formatted_date      : Date, format: "2006-04-01" "2006-04-01" ...
##  $ temperature         : num  9.47 9.36 9.38 8.29 8.76 ...
##  $ apparent_temperature: num  7.39 7.23 9.38 5.94 6.98 ...
##  $ Humidity            : num  0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82
0.72 ...
##  $ wind_speed_kmh      : num  14.12 14.26 3.93 14.1 11.04 ...
```

Handle missing values

There are no NAs to handle in this data set so there was nothing to modify

```
sapply(df, function(x) sum(is.na(x)==TRUE))
```

```
##       formatted_date          temperature apparent_temperature 
##                    0                    0                    0 
##             Humidity       wind_speed_kmh 
##                    0                    0 
```

Divide into train and test data

Divide the data to 80% train data and 20% test data

```
set.seed(1234)
i <- sample(1:nrow(df), 0.8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Data exploration

I looked at the first 4 rows using head() to get a peek into what my data looks like, and then used summary() to get an even better idea of the distribution of the data and get more detailed statistics about it. I also used mean() to get the averages of the temperature and humidity data features just to see what my data looks like.

```
head(train, n=4)
```

```
summary(train)
```

```
##  formatted_date        temperature      apparent_temperature    Humidit
y
##  Min.   :2006-01-01   Min.   :-21.822   Min.   :-27.717      Min.   :0.000
0
##  1st Qu.:2008-10-01   1st Qu.:  4.614   1st Qu.:  2.283      1st Qu.:0.600
0
##  Median :2011-07-04   Median : 11.978   Median : 11.978      Median :0.780
0
##  Mean   :2011-07-01   Mean   : 11.933   Mean   : 10.855      Mean   :0.734
8
##  3rd Qu.:2014-03-28   3rd Qu.: 18.839   3rd Qu.: 18.839      3rd Qu.:0.890
0
##  Max.   :2016-12-31   Max.   : 39.589   Max.   : 38.661      Max.   :1.000
0
##  wind_speed_kmh
##  Min.   : 0.000
##  1st Qu.: 5.796
##  Median : 9.918
##  Mean   :10.799
##  3rd Qu.:14.120
##  Max.   :63.853
```

```
mean(train$temperature)
```

```
## [1] 11.93288
```

```
mean(train$Humidity)
```

```
## [1] 0.7348262
```

Find the covariance and correlations in the data

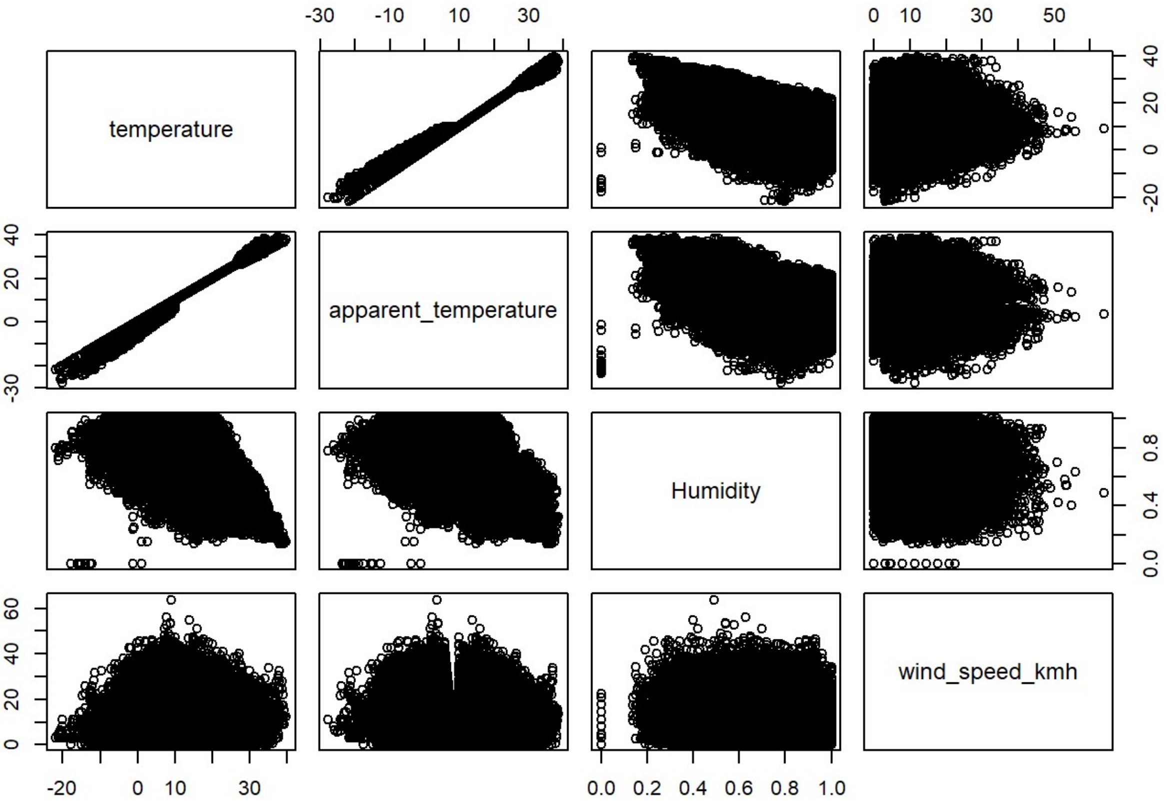
```
cov(train[,-1])
```

```
##                      temperature apparent_temperature    Humidity
## temperature           91.4646427           101.657875 -1.18328107
## apparent_temperature 101.6578752           114.669769 -1.26246592
## Humidity              -1.1832811            -1.262466  0.03831426
## wind_speed_kmh         0.5945195            -4.185549 -0.30279824
##                      wind_speed_kmh
## temperature               0.5945195
## apparent_temperature     -4.1855495
## Humidity                 -0.3027982
## wind_speed_kmh           47.7546022
```

```
cor(train[,-1])
```

```
##                        temperature apparent_temperature   Humidity
## temperature            1.000000000           0.99263581 -0.6320932
## apparent_temperature   0.992635811           1.00000000 -0.6023030
## Humidity              -0.632093170          -0.60230296  1.0000000
## wind_speed_kmh         0.008995636          -0.05656143 -0.2238543
##                       wind_speed_kmh
## temperature              0.008995636
## apparent_temperature    -0.056561432
## Humidity                -0.223854346
## wind_speed_kmh           1.000000000
```

```
pairs(train[,-1])
```



From the cor(), I found that the correlation between:

- Temperature and apparent temperature is very strong, which is expected
- Temperature and humidity is relatively strong
- Temperature and wind speed is very weak
- Humidity and apparent temperature is relatively strong

- Humidity and wind speed is weak
- Apparent temperature and wind speed is very weak

The pairs() plots all of these relationships which helps visualize and see the correlations

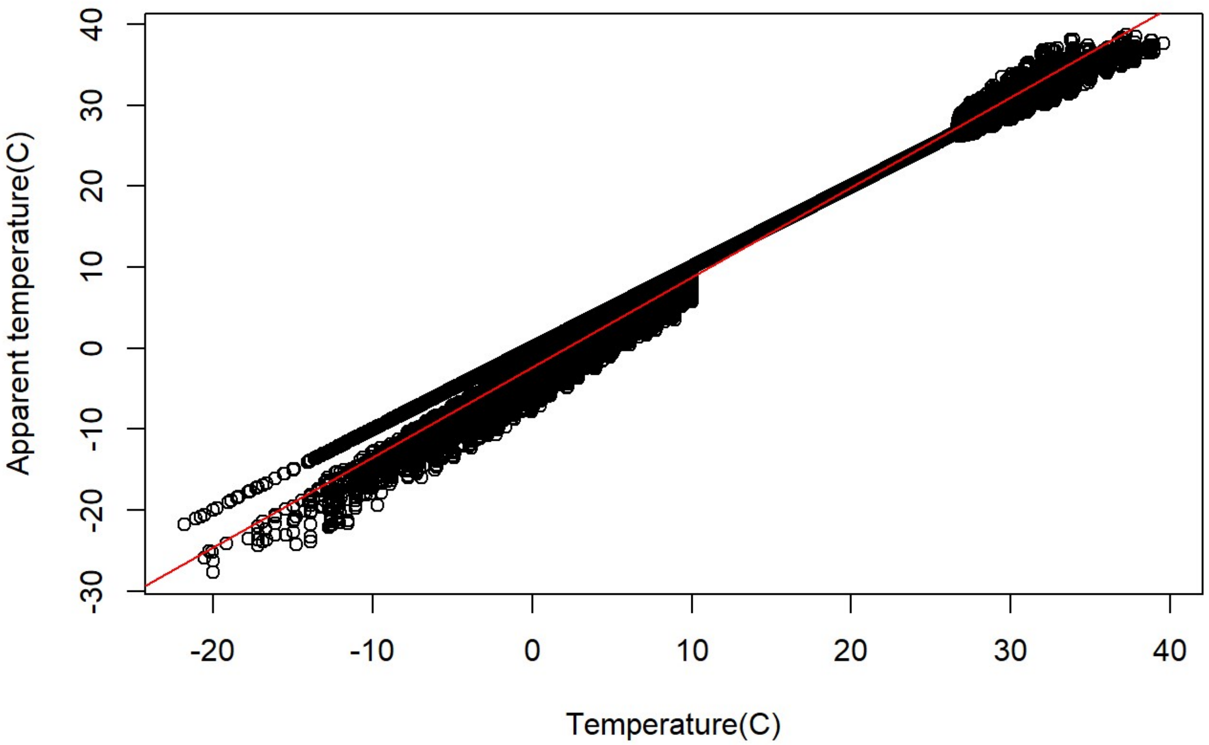
From the cov(), the covariance values tell me that:

- Temperature and apparent temperature are strongly and positively related
- Temperature and humidity are weakly and negatively related
- Temperature and wind speed are very weakly and positively related
- Humidity and apparent temperature are weakly and negatively related
- Humidity and wind speed very weakly and negatively related
- Apparent temperature and wind speed are weakly and negatively related

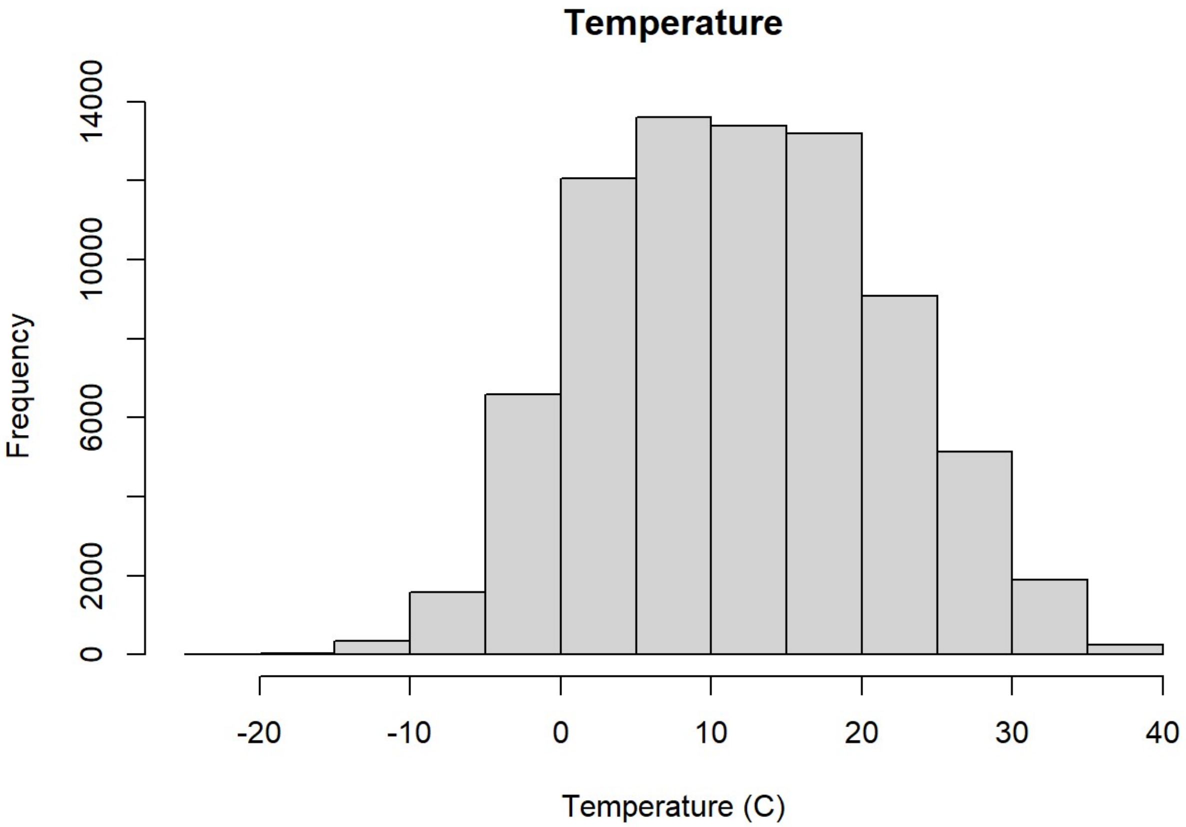
Plots and graphs

Plots the temperature vs apparent temperature using plot() and draws a red line for the linear model using abline(). Shows that temperature and apparent temperature are closely related to one another and follow a linear trend. The hist() draws a histogram of the temperature and its frequency, which shows a fairly normal distribution of the temperature data.

```
plot(train$temperature, train$apparent_temperature, xlab="Temperature(C)", ylab="Apparent temperature(C)")
abline(lm(train$apparent_temperature~train$temperature), col="red")
```

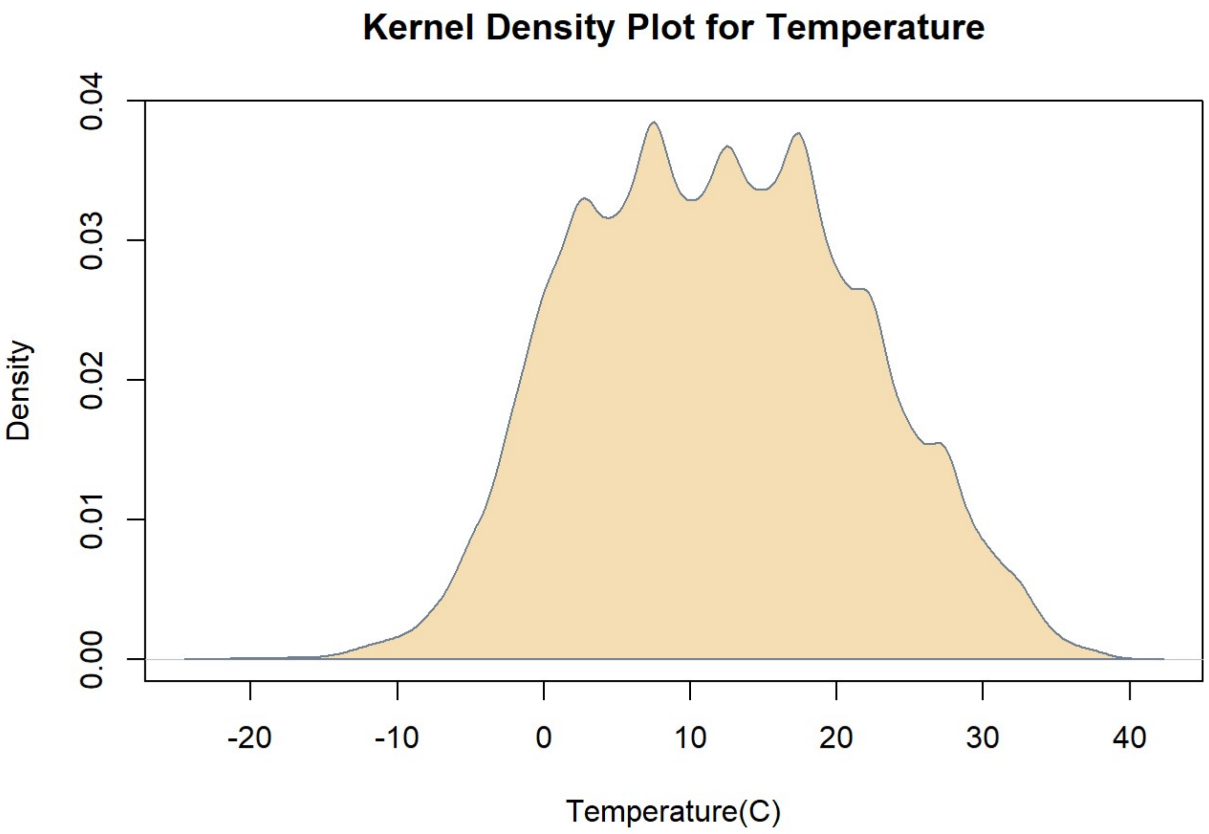


```
hist(train$temperature, main="Temperature", xlab="Temperature (C)")
```



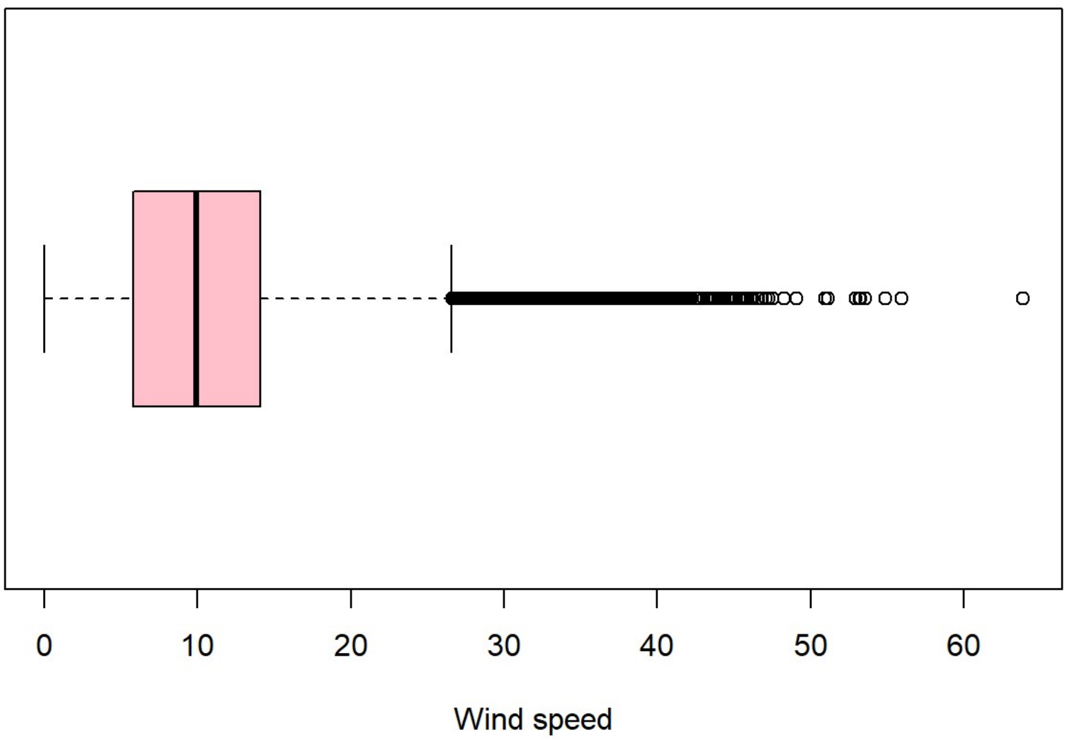
Draw the density plot for the temperature

```
d <- density(train$temperature, na.rm=TRUE)
plot(d, main="Kernel Density Plot for Temperature", xlab="Temperature(C)")
polygon(d, col="wheat", border="slategray")
```

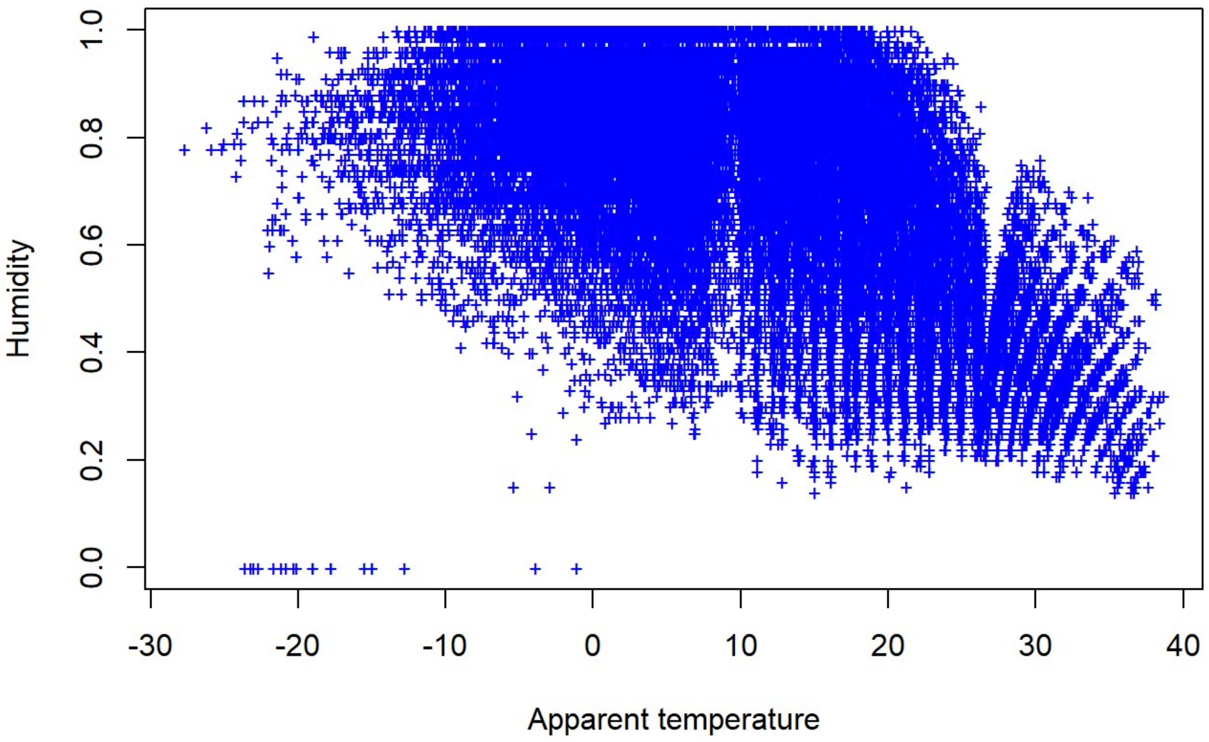


A boxplot is drawn of the wind speed, which shows that wind speed ranges from 0 to roughly 27 km/h, but there seem to be many data points above that range that R believes are outliers. Further investigation would be needed to determine if they are outliers or not. The first plot shows apparent temperature vs humidity, the second plot shows temperature vs humidity, and the last plot shows the temperature vs the date. This plots give a general idea of what the data looks like and the relationships between certain attributes.

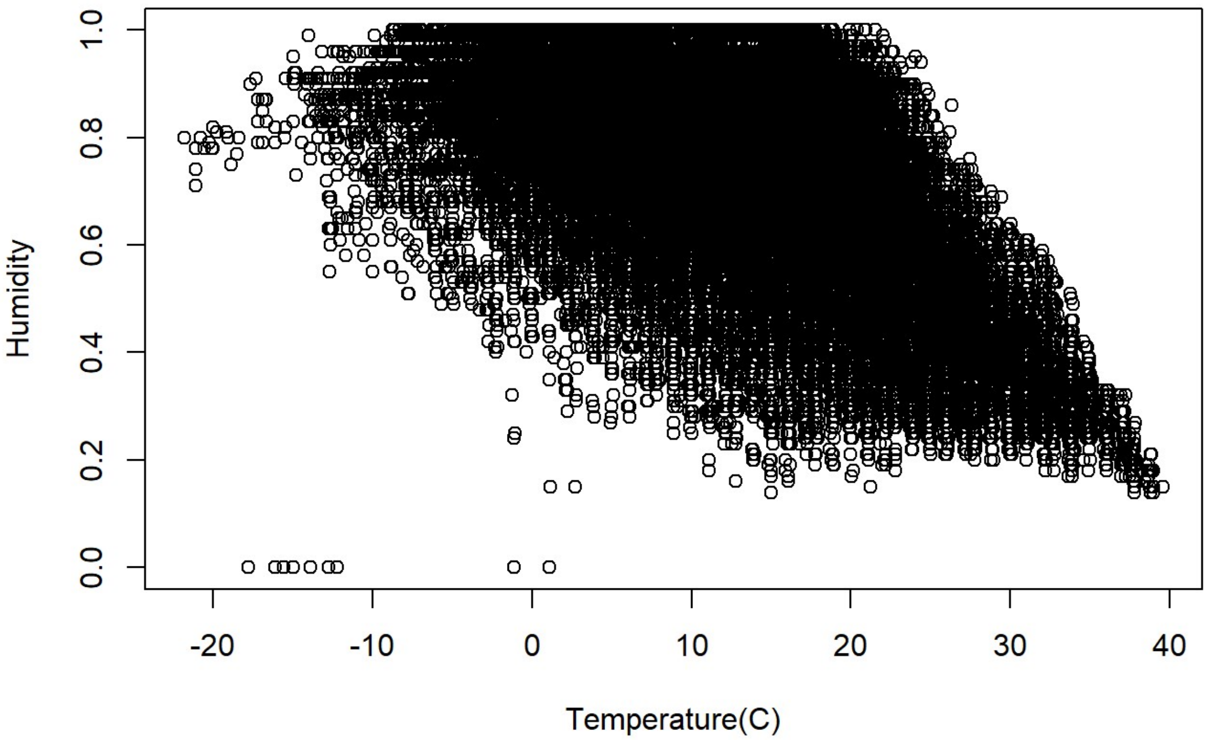
```
boxplot(train$wind_speed_kmh, col="pink", horizontal=TRUE, xlab="Wind speed")
```



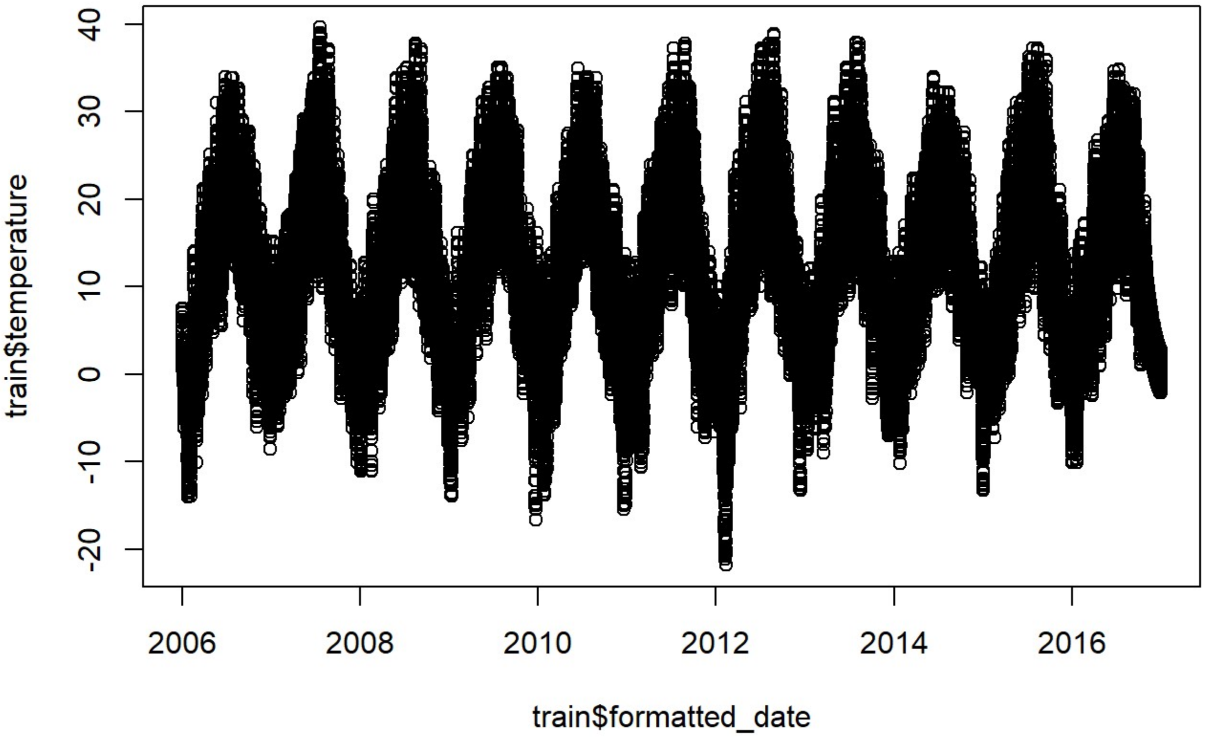
```
plot(train$apparent_temperature, train$Humidity, pch='+', cex=0.75, col="blue",
xlab="Apparent temperature", ylab="Humidity")
```



```
plot(train$temperature, train$Humidity, xlab="Temperature(C)", ylab="Humidity")
```



```
plot(train$formatted_date, train$temperature)
```



Make the linear regression model

Build a linear regression model to predict humidity from temperature, and print a summary of the model.

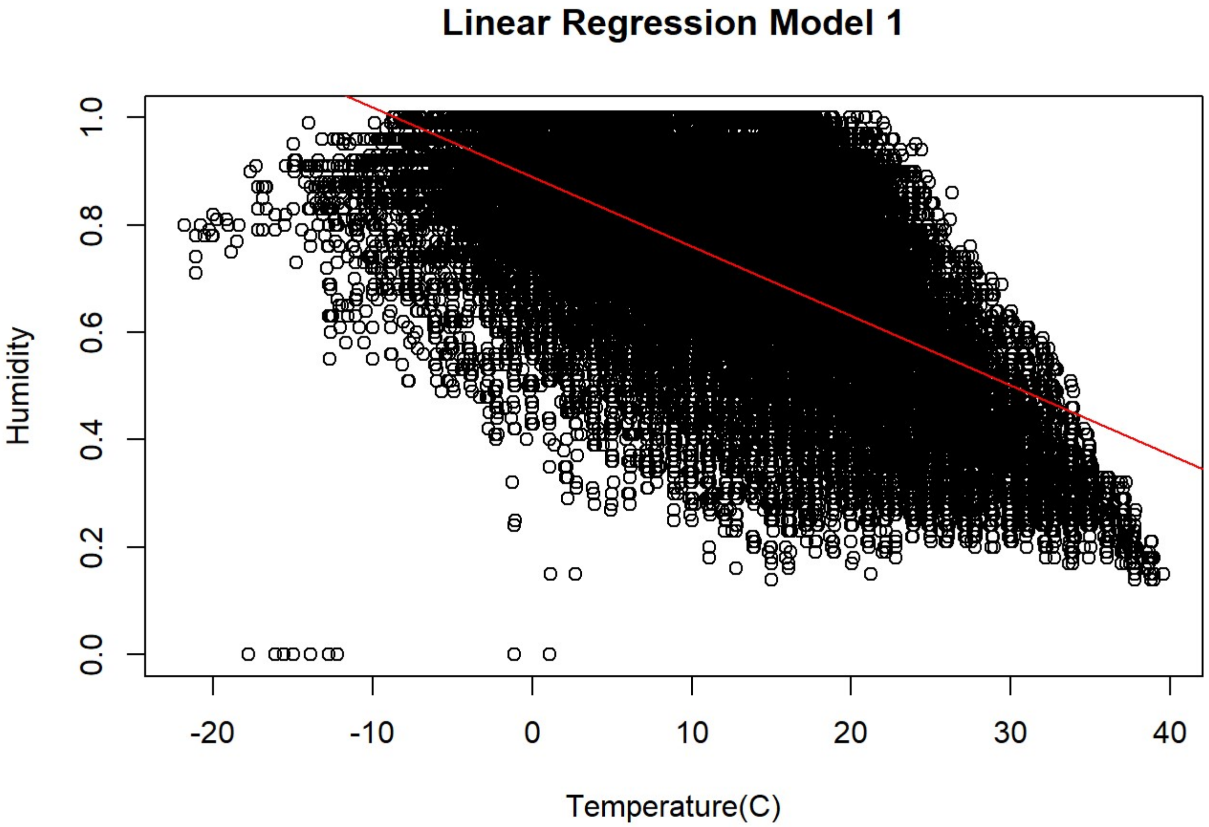
The summary shows some statistics about the residuals, and it then shows the coefficients of the model it build which has the structure y=wx+b. Here you can see that this model determined that w=-0.0129370 and b=0.8892022. I also see that R-squared is not too great at 0.3995, meaning that the variance in the model is not explained too well by the predictors; however, the RSE is low, which means that the model isn't far off from the data (fits the model relatively well), in this case the model is off by only 0.1517 in units of y. Lastly, I see that the F-statistics is much larger than 1 with a small associated p-value, which indicates that R is relatively confident in the model.

```
lm1 <- lm(Humidity~temperature, data=train)
summary(lm1)
```

```
## 
## Call:
## lm(formula = Humidity ~ temperature, data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.11919 -0.10147  0.00923  0.10943  0.38966 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.8892022  0.0008731  1018.4   <2e-16 ***
## temperature -0.0129370  0.0000571  -226.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1517 on 77160 degrees of freedom
## Multiple R-squared:  0.3995, Adjusted R-squared:  0.3995 
## F-statistic: 5.134e+04 on 1 and 77160 DF,  p-value: < 2.2e-16
```

Plot model 1

```
plot(train$Humidity~train$temperature, xlab="Temperature(C)", ylab="Humidity",
main="Linear Regression Model 1")
abline(lm1, col="red")
```

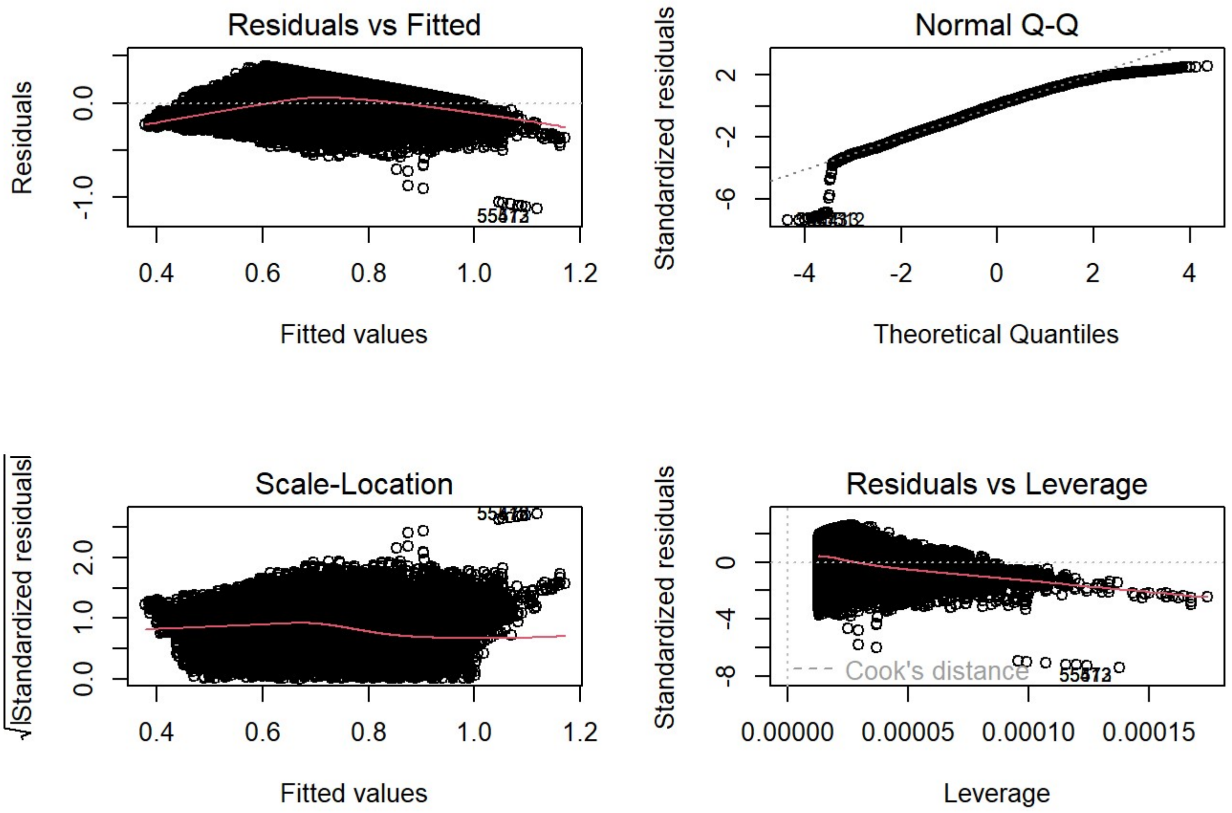


Plot residuals for model 1

The 4 residual plots tell us the following:

- Residuals vs Fitted - the residuals have a non-linear pattern of a concave parabola which the linear model lm1 didn’t capture
- Normal Q-Q - the residuals seem to mostly follow the dashed line except at the edges of the residuals range, so the residuals are not perfectly normally distributed
- Scale-Location - the line is fairly horizontal but it’s a bit difficult to judge whether the residuals are equally distributed around the line. It appears that the residuals are somewhat equally distributed, which leads to believe that the data is homoscedastic
- Residuals vs Leverage - it seems that there are no leverage points that are influencing the regression line

```
par(mfrow=c(2,2))
plot(lm1)
```



Build a multiple linear regression model

Build a multiple linear regression model to predict humidity from temperature and wind speed, and print a summary of the model.

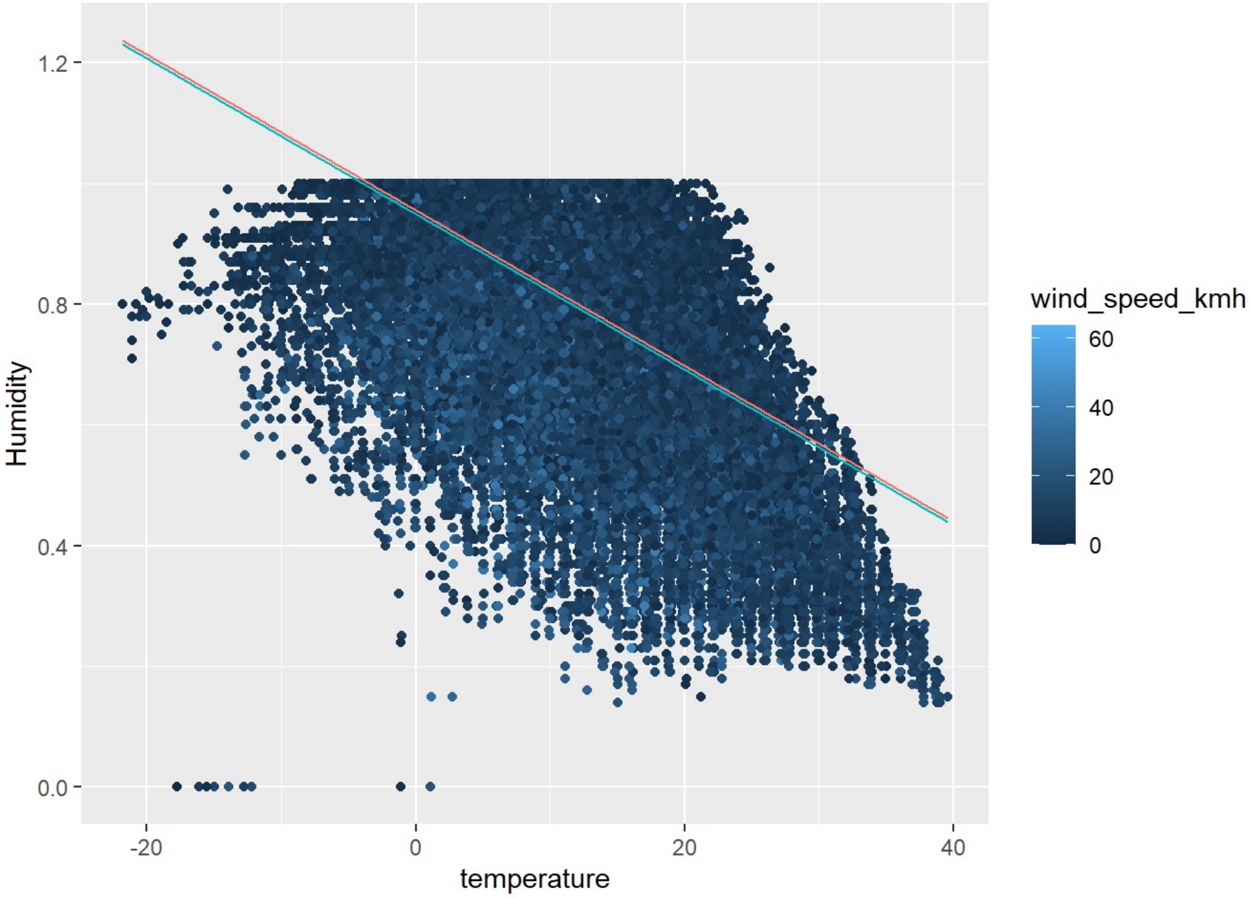
The summary shows that R-squared is 0.4471, which is not too great but still shows that the variation in humidity is predicted by the temperature and wind speed. Like in model 1, in this model the RSE is low, which means that the model isn’t far off from the data (the model is off by only 0.1455 in units of y). Lastly, I see that the F-statistics is much larger than 1 with a small associated p-value, which indicates that R is confident in the model.

```
lm2 <- lm(Humidity~temperature+wind_speed_kmh, data=train)
summary(lm2)
```

```
## 
## Call:
## lm(formula = Humidity ~ temperature + wind_speed_kmh, data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.18474 -0.09674  0.01130  0.10434  0.45726 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     9.555e-01  1.167e-03  818.46   <2e-16 ***
## temperature    -1.290e-02  5.479e-05 -235.40   <2e-16 ***
## wind_speed_kmh -6.180e-03  7.582e-05  -81.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1455 on 77159 degrees of freedom
## Multiple R-squared:  0.4471, Adjusted R-squared:  0.4471 
## F-statistic: 3.12e+04 on 2 and 77159 DF,  p-value: < 2.2e-16
```

Plot model 2

```
library(ggplot2)
ggplot(train, aes(y=Humidity, x=temperature, color=wind_speed_kmh))+geom_point
()+
  stat_function(fun=function(x){coef(lm2)[2]*x+coef(lm2)[1]}, geom="line", colo
r=scales::hue_pal()(2)[1])+
  stat_function(fun=function(x){coef(lm2)[2]*x+coef(lm2)[1]+coef(lm2)[3]}, geom
="line", color=scales::hue_pal()(2)[2])
```

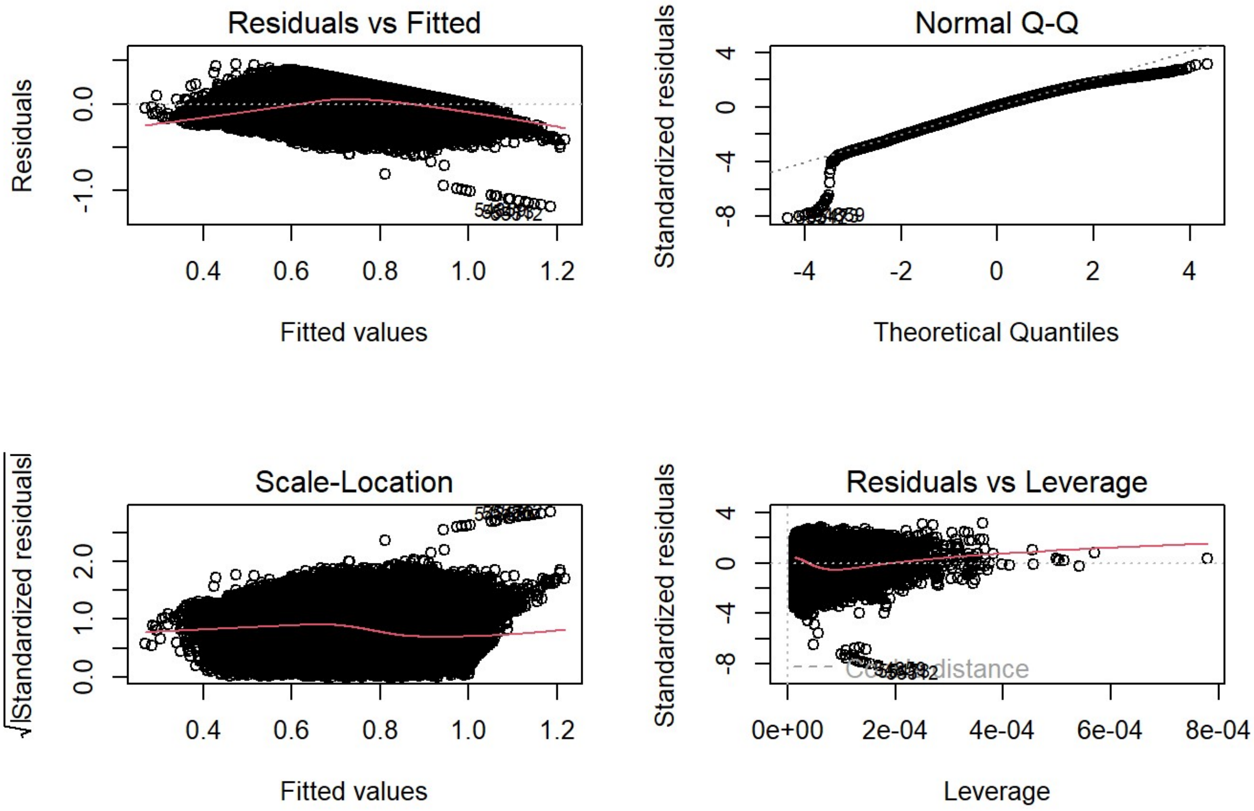


Plot residuals for model 2

The 4 residual plots tell us the following:

- Residuals vs Fitted - the residuals have a non-linear pattern of a concave parabola which the linear model lm2 didn’t capture
- Normal Q-Q - the residuals seem to mostly follow the dashed line except at the edges of the residuals range, so the residuals are not perfectly normally distributed (similarly to model 1)
- Scale-Location - the line is fairly horizontal and it seems like the residuals are equally distributed around the line, though it is a bit difficult to determine. Assuming the residuals are equally distributed, it can be inferred that the data is homoscedastic
- Residuals vs Leverage - it seems that there are no leverage points that are influencing the regression line

```
par(mfrow=c(2,2))
plot(lm2)
```



Build a third linear regression model

Build a polynomial regression model to predict humidity from temperature with degree=2, and print a summary of the model.

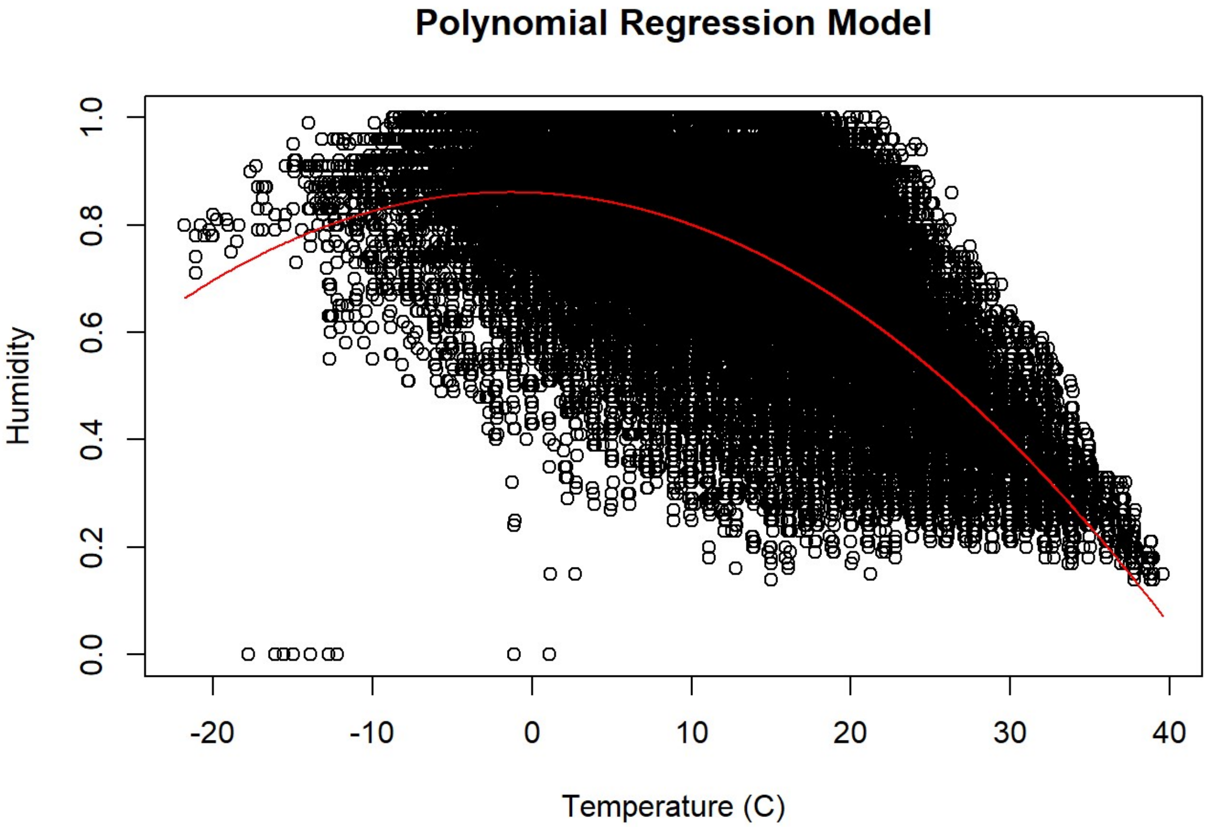
The summary shows that R-squared is not too great (R^2=0.4684) but still relatively good compared to the other models. Additionally, the RSE is low, which means that the model isn’t far off from the data (the model is off by only 0.1427 units of y). Lastly, I see that the F-statistics is much larger than 1 with a small associated p-value, which indicates that R is confident in the model.

```
lm3 <- lm(Humidity~poly(temperature, 2), data=train)
summary(lm3)
```

```
## 
## Call:
## lm(formula = Humidity ~ poly(temperature, 2), data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.86100 -0.08331  0.01900  0.09940  0.39270 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            7.348e-01  5.138e-04  1430.3   <2e-16 ***
## poly(temperature, 2)1 -3.437e+01  1.427e-01  -240.8   <2e-16 ***
## poly(temperature, 2)2 -1.427e+01  1.427e-01  -100.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1427 on 77159 degrees of freedom
## Multiple R-squared:  0.4684, Adjusted R-squared:  0.4684 
## F-statistic: 3.4e+04 on 2 and 77159 DF,  p-value: < 2.2e-16
```

Plot model 3

```
plot(train$temperature, train$Humidity, xlab="Temperature (C)", ylab="Humidit
y", main="Polynomial Regression Model")
x <- sort(train$temperature)
y <- lm3$fitted.values[order(train$temperature)]
lines(x, y, col="red")
```

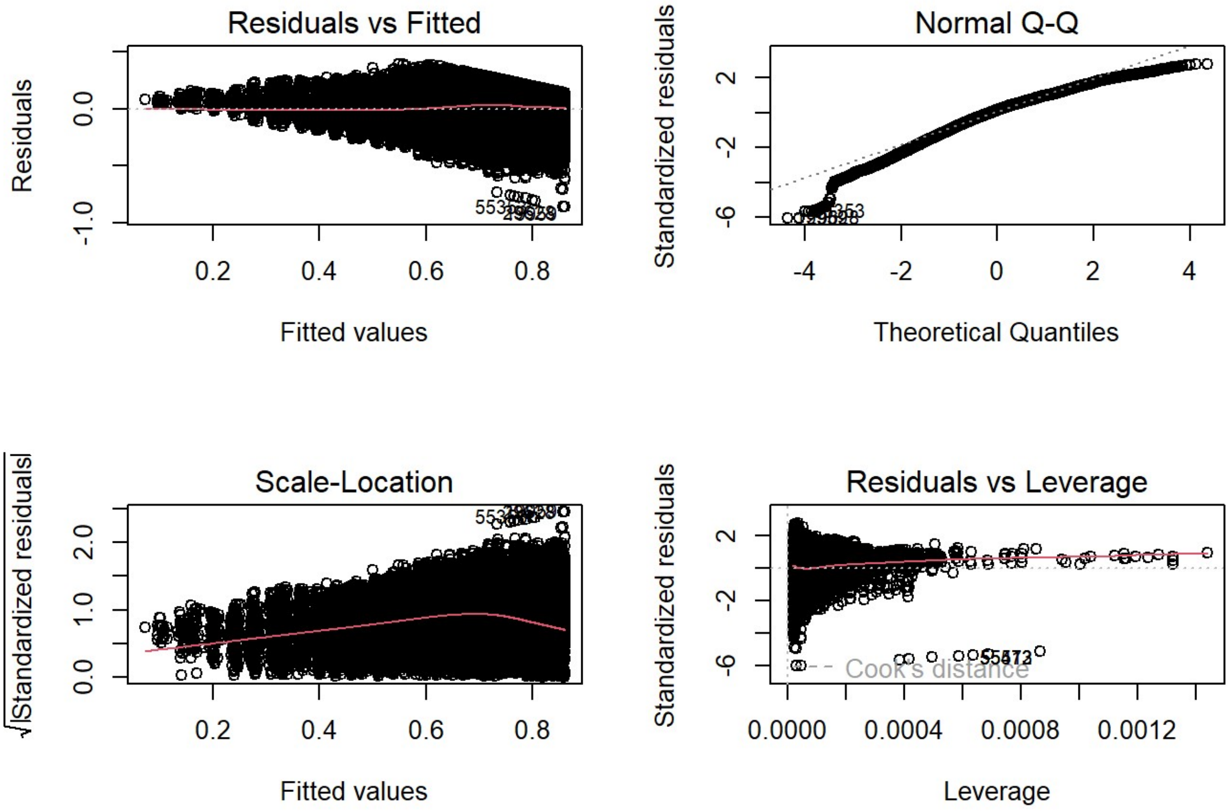


Plot residuals for model 3

The 4 residual plots tell us the following:

- Residuals vs Fitted - the residuals have a linear pattern which means that the linear model lm3 captured the variations in the data
- Normal Q-Q - the residuals seem to mostly follow the dashed line except at the edges of the residuals range, so the residuals are not perfectly normally distributed (though probably not as much as in models 1 and 2)
- Scale-Location - the line is not very horizontal though the residuals are fairly equally distributed around the line. This means that the data may not be homoscedastic
- Residuals vs Leverage - it seems that there are no leverage points that are influencing the regression line

```
par(mfrow=c(2,2))
plot(lm3)
```



Compare models

When comparing the models using anova(), I expected model 1 to be the least favorable since it had the smallest R-squared, and I was surprised to see that R choose model 2 as the model that best fits the data. This seems a bit counter intuitive since the RSS of model 3 is smaller than that of model 2, which means that model 3 fitted the data better.

```
anova(lm1, lm2, lm3)
```

When looking at the statistics from the summaries of the models, it looks like model 3 performed the best out of all of the models. The F-statistic of model 1 was the highest out of all the models, followed by model 3 and model 2, which means that R is most confident in model 1. However, the R-squared of model 3 was the closest to 1 out of all the models, though it was very close to that of model 2, which could mean that it didn’t outperform it by much. The RSE value of model 3 was the smallest, meaning that the model was the least off from the data (the lack of fit of the model was the smallest), which further indicates that model 3 fits the data better than the other models, though model 2 has a very close RSE value to that of model 3. Based on these metrics, it seems like model 3 performed the best and therefore is the model that best fits the data. However, based on the results of anova(), it seems that R believes that model 2 performed the best out of all the models. Since the R-squared, RSE, and F-statistic of models 2 and 3 were all very close to one another, I think it is understandable why R might choose model 2 as the best model.

Predictions for the models

For the first model

```
pred <- predict(lm1, newdata=test)

correlation <- cor(pred, test$Humidity)
print(paste("correlation: ", correlation))
```

```
## [1] "correlation:  0.632910093630736"
```

```
mse <- mean((pred-test$Humidity)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  0.0226529029704227"
```

```
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  0.150508813597154"
```

For the second model

```
pred <- predict(lm2, newdata=test)

correlation <- cor(pred, test$Humidity)
print(paste("correlation: ", correlation))
```

```
## [1] "correlation:  0.671313199014223"
```

```
mse <- mean((pred-test$Humidity)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  0.0207605687657763"
```

```
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  0.144085282960392"
```

For the third model

```
pred <- predict(lm3, newdata=test)

correlation <- cor(pred, test$Humidity)
print(paste("correlation: ", correlation))
```

```
## [1] "correlation:  0.681986486903022"
```

```
mse <- mean((pred-test$Humidity)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  0.0202142315004611"
```

```
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  0.142176761464246"
```

When evaluating the models using the test data, the conclusion seems to be the same, where model 3 outperformed the other models. The correlation, MSE, and RMSE values all support this conclusion since the R-squared of model 3 is the largest, and both its MSE and RMSE are the smallest, which shows that it is the best fit for the data out of all the other models. I think these results happened because the data had a non-linear relationship between the predictor (temperature) and target (humidity), which model 3 was able to capture best out of all the models, though it seems like model 2 also fit the data well when compared to model 3 using the above metrics.

References:

Mazidi, Karen. *Machine Learning Handbook Using R and Python*. 2nd ed., 2020.