stat420 final proj

Contents

- Wenhao Tan - wenhaot2
- Ray Pan - yulinp3
- Yulan Ma - yulanma2

Introduction

Australia is a continental island country located in the southern hemisphere. It attracts billions of travelers each year. This data set contains the daily weather observations from numerous Australian weather stations from 2008. There are a total of 1685 observations and 19 columns of variables, including locations of the weather stations, minimum temperature, maximum temperature, wind gust speed, humidity, pressure, etc.

We took this data set from https://www.kaggle.com/jsphyg/weather-dataset-rattle-package. The original data set contains about 10 years of daily weather observations from many locations across Australia, and we took only the data of the year 2008 from Canberra.

We hope to understand the general weather in Australia, for example the minimum temperature and chance of rain, so that we know when we should travel to Australia during the year for vacation.

Methods

```
#import packages
library(readr)
library(faraway)
library(ggplot2)
library(lmtest)
library(caret)
library(ROCR)
library(knitr)
library(kableExtra)
```

Data

We first do some data cleaning work, including cleaning up the NAs contained in the dataset.

```
#import the original dataset
weather_raw <- read_csv("weatherAUS.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Date = col_date(format = ""),
##   Location = col_character(),
##   Evaporation = col_logical(),
##   Sunshine = col_logical(),
##   WindGustDir = col_character(),
##   WindDir9am = col_character(),
##   WindDir3pm = col_character(),
##   RainToday = col_character(),
##   RainTomorrow = col_character()
## )
## i<U+00A0>Use `spec()` for the full column specifications.
```

```
# function to determine proportion of NAs in a vector
na_prop = function(x) {
  mean(is.na(x))
}

# check proportion of NAs in each column
sapply(weather_raw, na_prop)
```

```
##          Date      Location       MinTemp       MaxTemp      Rainfall
##    0.00000000    0.00000000    0.01020899    0.00866905    0.02241853
##   Evaporation      Sunshine   WindGustDir WindGustSpeed    WindDir9am
##    0.98746047    0.98151382    0.07098859    0.07055548    0.07263853
##    WindDir3pm  WindSpeed9am  WindSpeed3pm   Humidity9am   Humidity3pm
##    0.02906641    0.01214767    0.02105046    0.01824557    0.03098446
##   Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am
##    0.10356799    0.10331363    0.38421559    0.40807095    0.01214767
##       Temp3pm     RainToday  RainTomorrow
##    0.02481094    0.02241853    0.02245978
```

```
# create dataset without columns containing more than 33% NAs
weather_clean = na.omit(weather_raw[, !sapply(weather_raw, na_prop) > 0.33])
# proportion of cleaned dataset to full dataset
nrow(weather_clean)/nrow(weather_raw)
```

```
## [1] 0.7763303
```

```
weather_clean$Date=as.character(weather_clean$Date)
weather <- subset(weather_clean,startsWith(weather_clean$Date, "2008"))
#select the capital city to predict
```

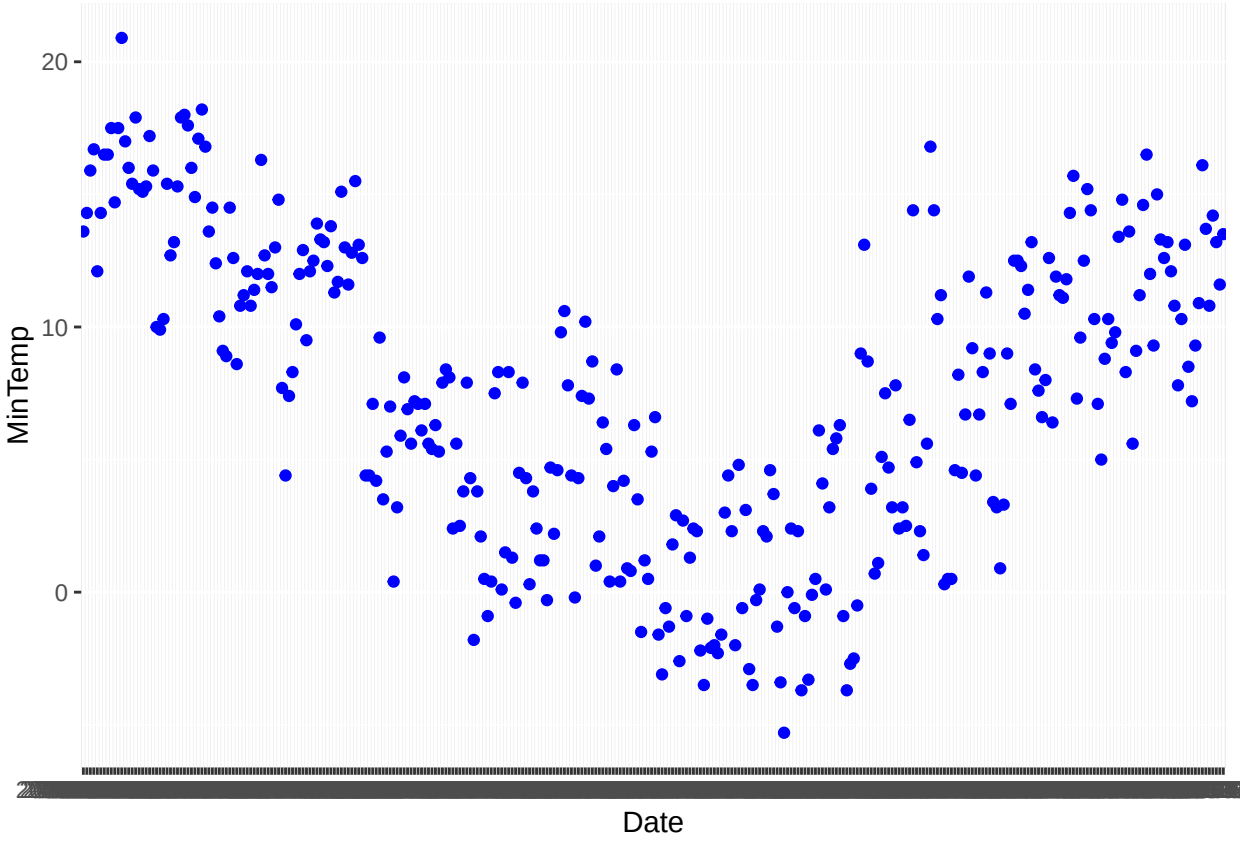
```
weather_can<-weather[which(weather$Location=='Canberra'),]
head(weather_can, 5)
```

```
## # A tibble: 5 x 19
##   Date    Location MinTemp MaxTemp Rainfall WindGustDir WindGustSpeed WindDir9am
##   <chr>   <chr>      <dbl>   <dbl>    <dbl> <chr>               <dbl> <chr>
## 1 2008-0~ Canberra    13.6    34.2        0 NNE                    35 ESE
## 2 2008-0~ Canberra    14.3    35          0 ESE                    41 ESE
## 3 2008-0~ Canberra    15.9    23.4        0 ESE                    50 ESE
## 4 2008-0~ Canberra    16.7    25.3        0 ESE                    46 ESE
## 5 2008-0~ Canberra    12.1    27.5        0 NE                     35 SSE
## # ... with 11 more variables: WindDir3pm <chr>, WindSpeed9am <dbl>,
## #   WindSpeed3pm <dbl>, Humidity9am <dbl>, Humidity3pm <dbl>,
## #   Pressure9am <dbl>, Pressure3pm <dbl>, Temp9am <dbl>, Temp3pm <dbl>,
## #   RainToday <chr>, RainTomorrow <chr>
```

We cleaned up the data with NA omitted and took only the data of the year 2008 from the city Canberra.

Predicting Minimum Temperature

```
#scatter plot of date vs. minimum temperature
ggplot(data = weather_can, aes(x = Date, y = MinTemp))+geom_point(color="blue")
```



```r
#remove the column of Location to avoid potential problems
weather_can=weather_can[,-2]
#fit the model with possible predictors
raw_mod_1<-lm(MinTemp ~ (RainToday + MaxTemp + Pressure3pm + Humidity3pm )^2,
              data=weather_can)
raw_mod_2<-lm(MinTemp ~ (RainToday + MaxTemp +
                         Pressure3pm + Humidity3pm + WindGustSpeed)^2,
              data=weather_can)
raw_mod_3<-lm(MinTemp ~ (RainToday + MaxTemp + Pressure3pm +
                         Humidity3pm + WindGustSpeed + Temp3pm)^2,
              data=weather_can)
#p-values for ANOVA test
anova(raw_mod_1, raw_mod_2)["Pr(>F)"][2,]
```

```
## [1] 0.0001745825
```

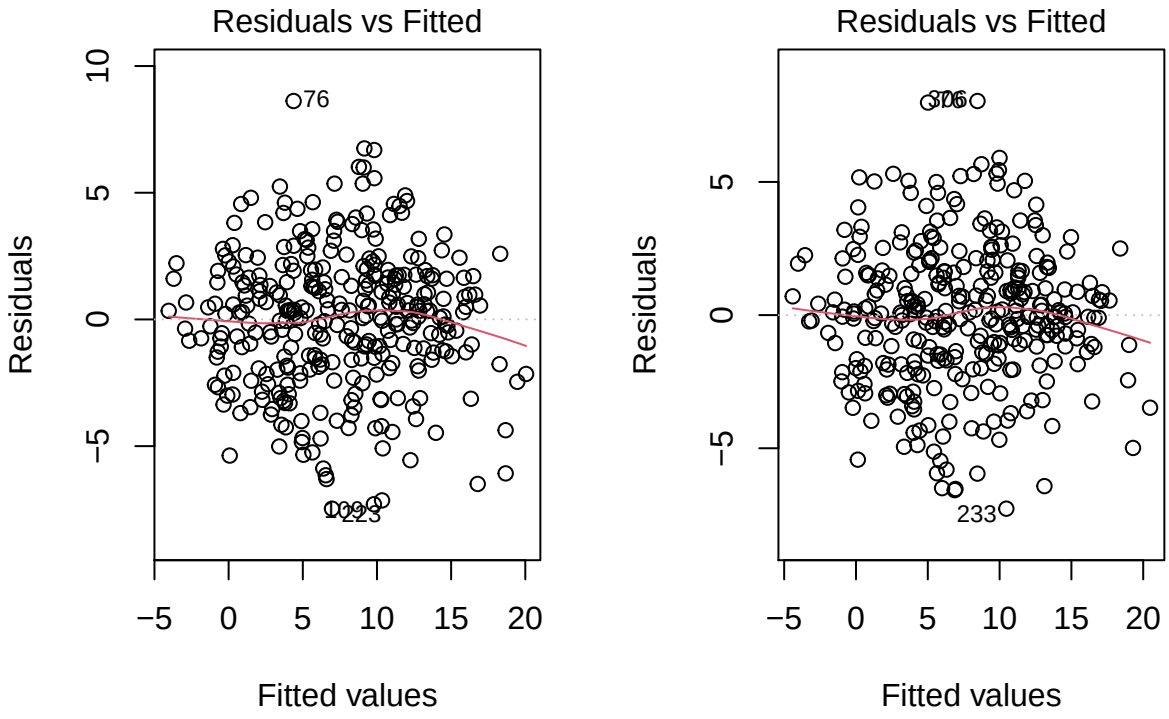
```r
anova(raw_mod_2, raw_mod_3)["Pr(>F)"][2,]
```

```
## [1] 0.001611429
```

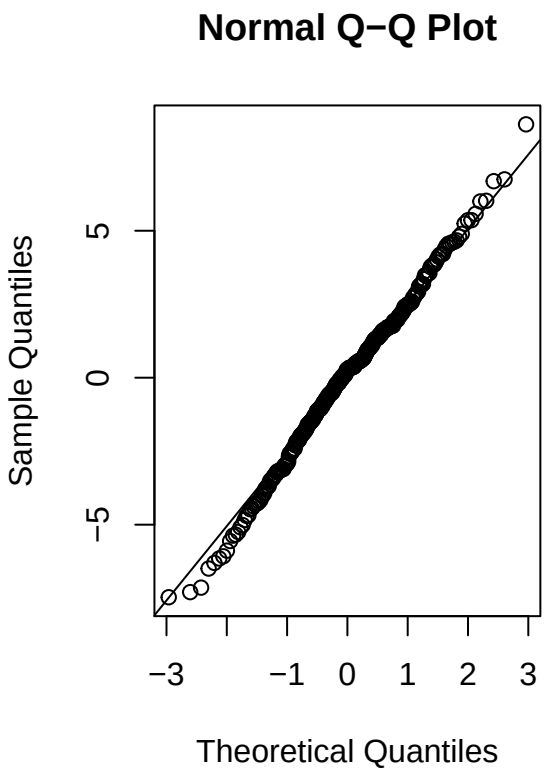
```r
#backward selection on the model
select_2=step(raw_mod_2, direction="backward", trace=0)
select_3=step(raw_mod_3, direction="backward", trace=0)
#adjusted R-squared values
adj_2=summary(select_2)$adj
adj_3=summary(select_3)$adj

#Breusch-Pagan Test and Shapiro-Wilk test test values for model diagnostics
bp_2=bptest(select_2)$p.value
sha_2=shapiro.test(resid(select_2))$p.value
bp_3=bptest(select_3)$p.value
sha_3=shapiro.test(resid(select_3))$p.value

#calculate RMSE
calc_loocv_rmse<-function(model){
  sqrt(mean((resid(model) / (1 - hatvalues(model)))^2))
}
rmse_2=calc_loocv_rmse(select_2)
rmse_3=calc_loocv_rmse(select_3)

data=data.frame(
  c(adj_2, adj_3),
  c(bp_2, bp_3),
  c(sha_2, sha_3),
  c(rmse_2, rmse_3))

#put all values together
colnames(data)=c("adj r.squared", "Breusch-Pagan", "Shapiro-Wilk", "LOOCV RMSE")
rownames(data)=c("Mod2", "Mod3")
kable(data)%>%
  kable_styling(full_width = T)
```

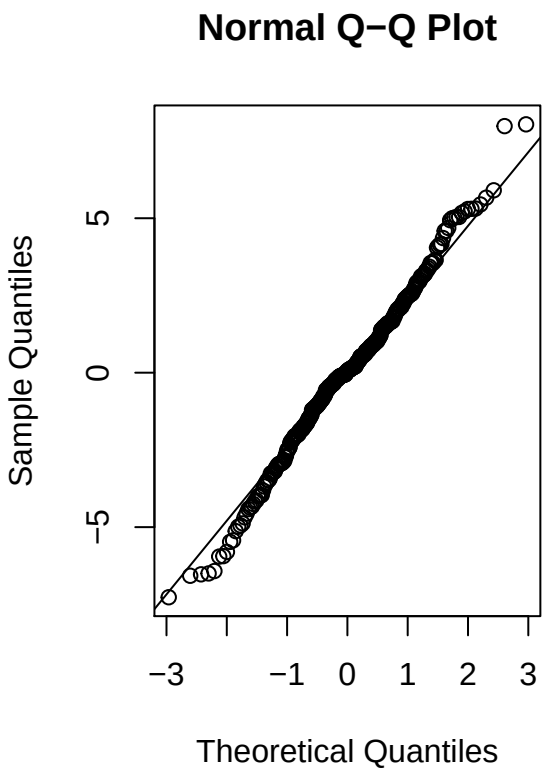
	adj r.squared	Breusch-Pagan	Shapiro–Wilk	LOOCV RMSE
Mod2	0.7733356	0.0013179	0.4204914	2.790363
Mod3	0.7856335	0.0756129	0.1261813	2.719191

```
#Residual vs Fitted plot
par(mfrow=c(1,2))
plot(select_2, which=1)
plot(select_3, which=1)
```




```
#Normal Q-Q plot
par(mfrow=c(1,2))
qqnorm(resid(select_2))
qqline(resid(select_2))
qqnorm(resid(select_3))
qqline(resid(select_3))
```





The scatter plot makes sense because Australia is in the southern hemisphere and it has a relatively low temperature in the middle of the year.

We thought the predictors in raw_mod_1 would be relative to the response variable "MinTemp", but we were unsure whether the variables "WindGustSpeed" and "Temp3pm" have effects on "MinTemp". We did two ANOVA tests and found out that both the p-values are lower than 0.05, which means the two variables somewhat contribute to "MinTemp", so we added these two variables in our model.

We run several diagnostics, including Breusch-Pagan Test, Shapiro–Wilk test, and the LOOCV RMSE calculation on our model and print a table for those values. Even though the value for Shapiro–Wilk test decreases, it's still in an acceptable range.

Also, the adjusted R-squared value of the third model increases, which indicates an increase in the power of the regression model. So we conclude that the third model would be the best here.

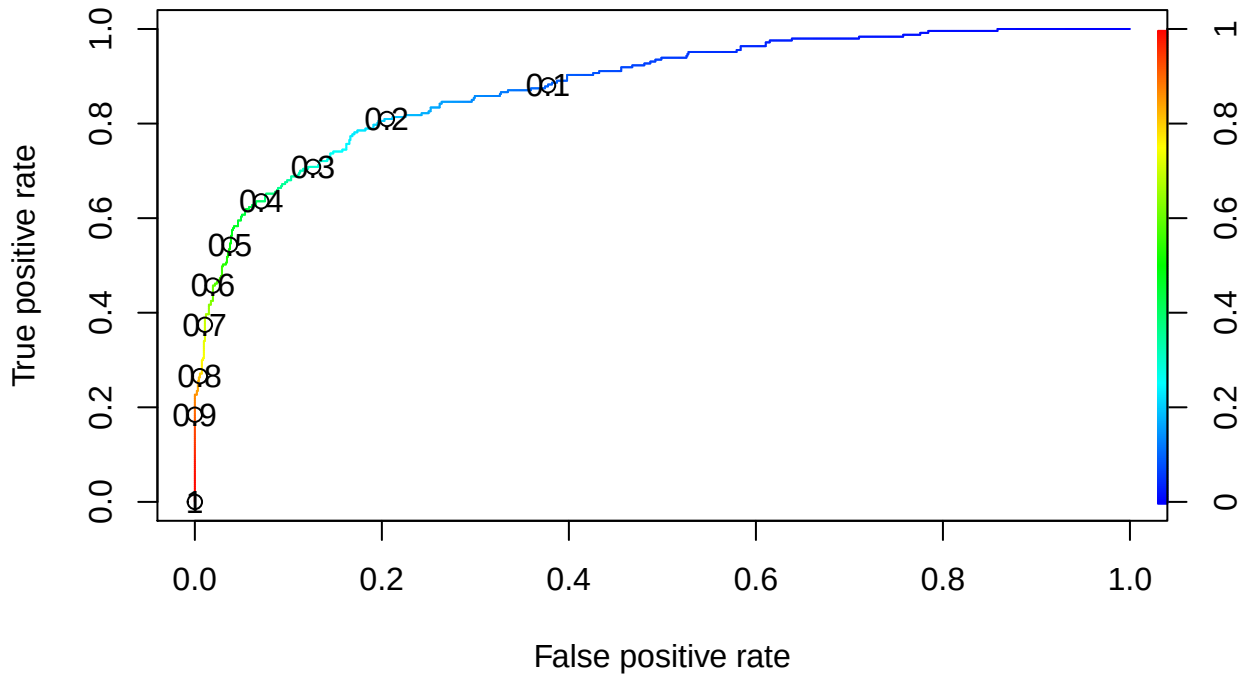
We also plot the fitted vs. residual and normal Q-Q plots as part of the diagnostic process. Some outlier like the value of 76, 233, 306 were detected.

Predicting whether there will be rain tomorrow

Here we selected another response variable "RainTomorrow".

```
set.seed(100)
#remove the column of Date to avoid potential problems
weather_new<-weather[,-1]
#change the binary factors into numeric values
weather_new$RainTomorrow[weather_new$RainTomorrow=="Yes"]<-1
```

```
weather_new$RainTomorrow[weather_new$RainTomorrow=="No"]<-0
weather_new$RainTomorrow<-as.numeric(weather_new$RainTomorrow)
#dividing the dataset into training(70%) and testing(30%)
we_trn_idx = sample(nrow(weather_new), size = 0.7 * nrow(weather_new))
we_trn = weather_new[we_trn_idx, ]
we_tst = weather_new[-we_trn_idx, ]
#fit glm model
fit<-glm(RainTomorrow~., data=we_trn, family="binomial")
prob=predict(fit, we_tst, type="response")

we_trn$pred<-fitted(fit)
pred<-prediction(we_trn$pred,we_trn$RainTomorrow)
perf<-performance(pred,"tpr","fpr")
plot(perf,colorize = T,print.cutoffs.at = seq(0.1,by = 0.1))
```




```
#use 0.5 as the cutoff according to the ROC curve
pred = factor(ifelse(prob>0.5, "1", "0"))
confusionMatrix(pred, factor(we_tst$RainTomorrow))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 366  65
##          1  23  52
```

```
##
##                Accuracy : 0.8261
##                  95% CI : (0.7902, 0.8581)
##     No Information Rate : 0.7688
##     P-Value [Acc > NIR] : 0.0009934
##
##                   Kappa : 0.4406
##
##  Mcnemar's Test P-Value : 1.239e-05
##
##             Sensitivity : 0.9409
##             Specificity : 0.4444
##          Pos Pred Value : 0.8492
##          Neg Pred Value : 0.6933
##              Prevalence : 0.7688
##          Detection Rate : 0.7233
##    Detection Prevalence : 0.8518
##       Balanced Accuracy : 0.6927
##
##        'Positive' Class : 0
##
```

From the confusion matrix, we have an accuracy of 0.8261 when predicting whether there will be rain tomorrow with a sensitivity of 0.9409 and specificity of 0.4444, which are fairly reasonable.

Results

Mod3 would be our best model here for predicting minimum temperature. It has a adjusted R.squared value of 0.7856 with reasonable diagnostic values. It has a relative lower LOOCV RMSE value comparing to Mod2.

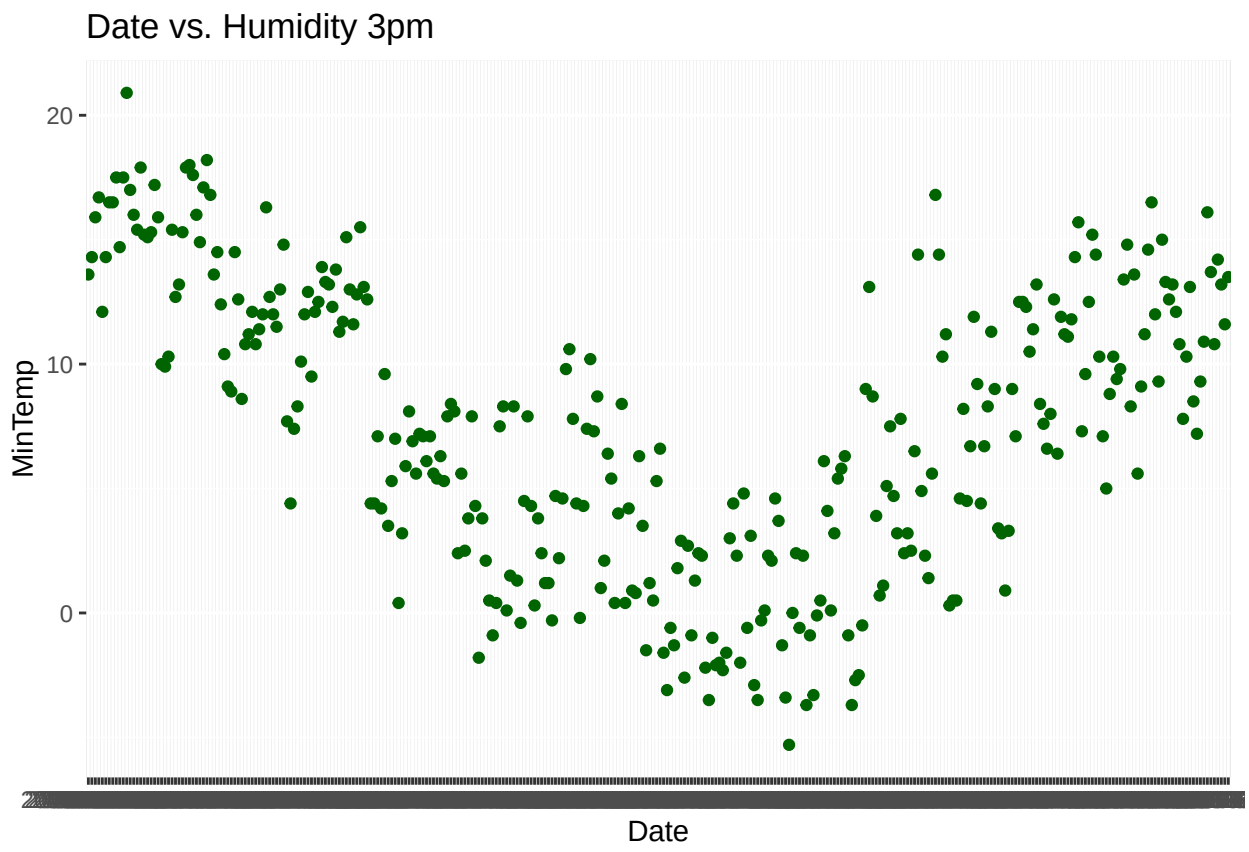
Final model: $\hat{y}_{MinTemp} = 143.8 + 3.507x_{RainTodayYes} + 37.95x_{MaxTemp} - 0.1705x_{Pressure3pm} + 25.45x_{Humidity3pm} - 3.155x_{WindGustSpeed} - 36.79x_{Temp3pm} - 0.04286x_{RainTodayYes}x_{Humidity3pm} - 0.03517x_{MaxTemp}x_{Pressure3pm} - 0.02637x_{MaxTemp}x_{Humidity3pm} - 0.007057x_{MaxTemp}x_{WindGustSpeed} + 0.003436x_{Pressure3pm}x_{WindGustSpeed} + 0.03499x_{Pressure3pm}x_{Temp3pm} - 0.003212x_{Humidity3pm}x_{WindGustSpeed} + 0.03091x_{Humidity3pm}x_{Temp3pm}$

```
predict(select_3, data.frame(RainToday='No',
                             MaxTemp=32.3,
                             Pressure3pm=1006,
                             Humidity3pm=33,
                             WindGustSpeed=41,
                             Temp3pm=29.7))
```

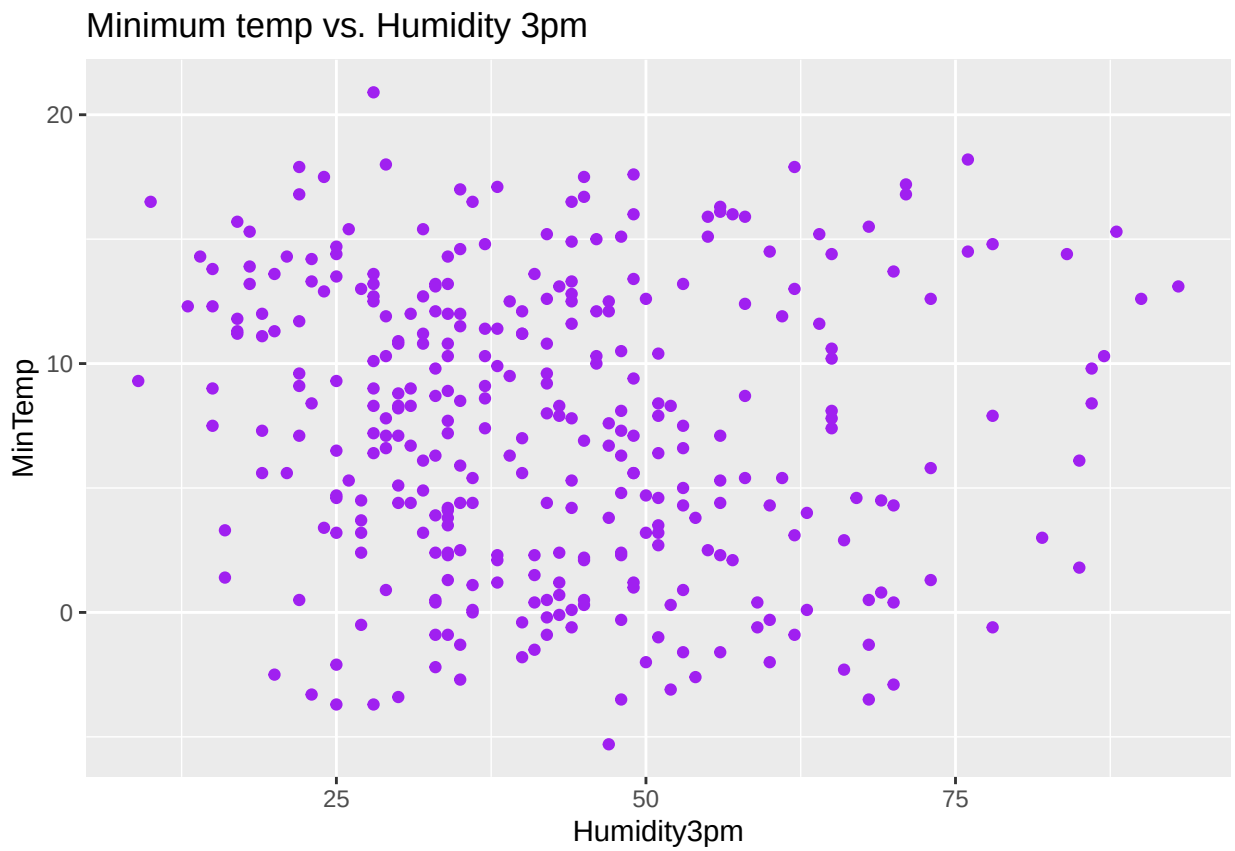
```
##        1
## 17.37102
```

We selected a random day from 2019 to see if our model is accurate enough. The predicted minimum temperature is 17.37, and the recorded minimum temperature is 17.5, which is pretty close.

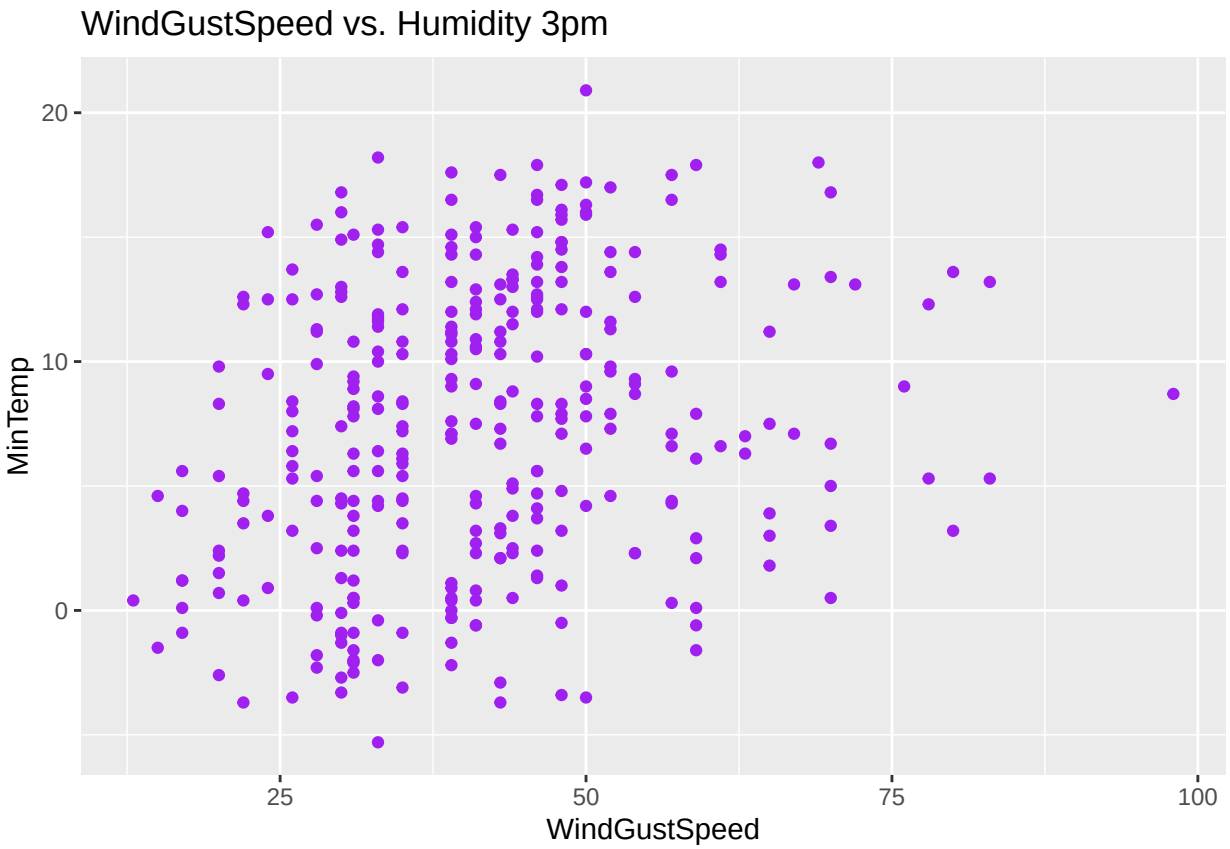
```
ggplot(data = weather_can,aes(x=Date,y=MinTemp))+
  geom_point(color="dark green")+
  ggtitle('Date vs. Humidity 3pm')
```




```
ggplot(data = weather_can,aes(x=Humidity3pm,y=MinTemp))+
  geom_point(color="purple")+
  ggtitle('Minimum temp vs. Humidity 3pm')
```




```
ggplot(data = weather_can,aes(x=WindGustSpeed,y=MinTemp))+
  geom_point(color="purple")+
  ggtitle('WindGustSpeed vs. Humidity 3pm')
```



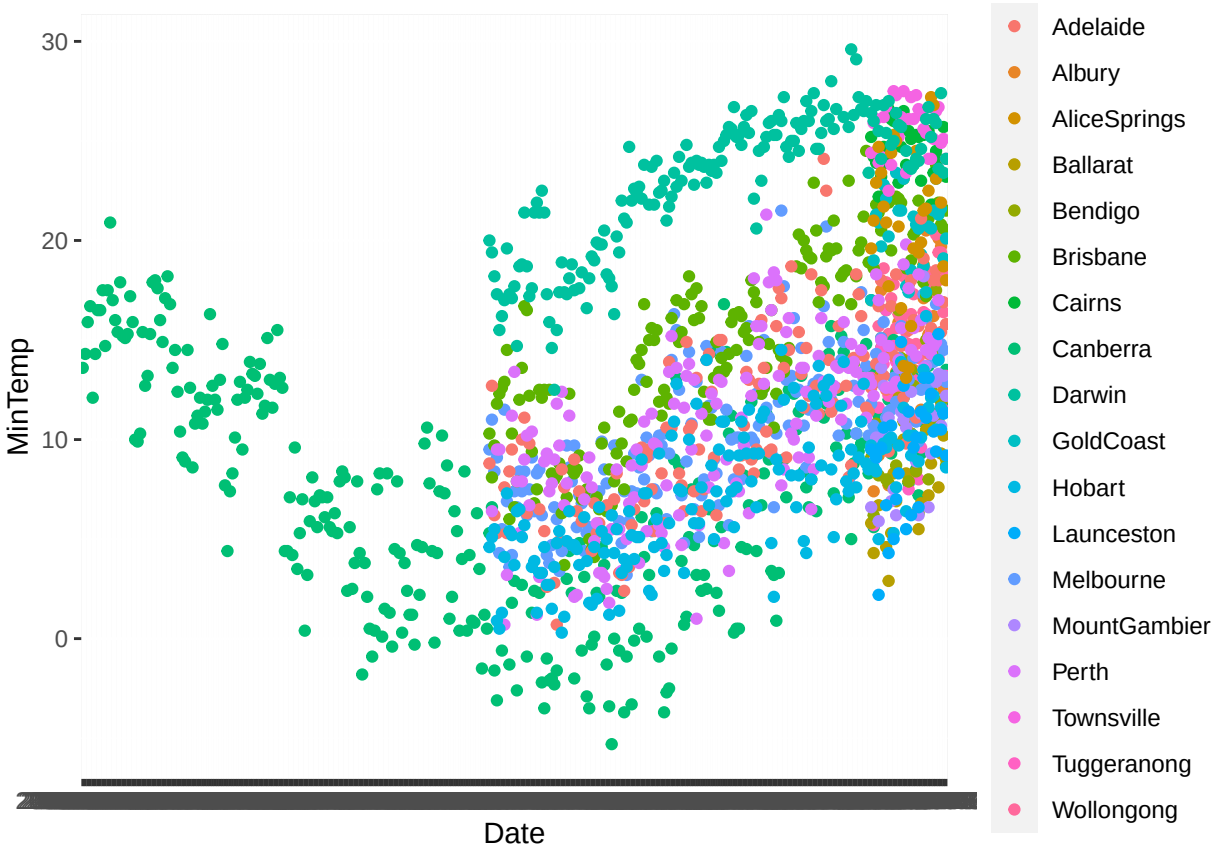
On the other hand, if one wants to see whether there will be rain tomorrow, we would use the glm model provided above with an accuracy of 0.8261. The sensitivity (the ability to correctly predict a rainy day) is 0.9409, which means it could basically predict correctly with a few exceptions.

Discussion

We were able to improve our model based on adding more relative predictors to the existing model. We evaluated the prediction of minimum temperature in Canberra, which would be helpful for our vacation decisions in Canberra, Australia. Also, we believe agriculture and animal husbandry could benefit from these predictions. There are many missing values in the dataset which may cause the result to be inaccurate. There are only 328 values in a total of 365 days, which means there are possibilities that not all possible conditions are considered in the model. Even though the models we chose have fairly high reliability, people should use caution when making decisions based on the prediction due to the chance of possible margin of error or false prediction.

Appendix

```
ggplot(data = weather, aes(x = Date, y = MinTemp,      color=Location))+geom_point()
```



Summary of model 2

```
summary(select_2)
```

```
##
## Call:
## lm(formula = MinTemp ~ RainToday + MaxTemp + Pressure3pm + Humidity3pm +
##     WindGustSpeed + MaxTemp:Humidity3pm + MaxTemp:WindGustSpeed +
##     Pressure3pm:Humidity3pm + Humidity3pm:WindGustSpeed, data = weather_can)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4685 -1.6984  0.2806  1.7123  8.6195
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.067e+02  7.741e+01   2.670  0.00796 **
## RainTodayYes                1.517e+00  4.591e-01   3.305  0.00106 **
## MaxTemp                     1.051e+00  1.101e-01   9.544  < 2e-16 ***
## Pressure3pm                -2.307e-01  7.517e-02  -3.070  0.00233 **
## Humidity3pm                -3.656e+00  1.585e+00  -2.307  0.02171 *
## WindGustSpeed               3.615e-01  7.337e-02   4.927 1.35e-06 ***
## MaxTemp:Humidity3pm         4.845e-03  1.529e-03   3.169  0.00168 **
## MaxTemp:WindGustSpeed      -9.425e-03  2.221e-03  -4.244 2.89e-05 ***
```

```
## Pressure3pm:Humidity3pm    3.772e-03  1.539e-03   2.452  0.01476 *
## Humidity3pm:WindGustSpeed -2.979e-03  8.722e-04  -3.415  0.00072 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.741 on 318 degrees of freedom
## Multiple R-squared:  0.7796, Adjusted R-squared:  0.7733
## F-statistic:   125 on 9 and 318 DF,  p-value: < 2.2e-16
```

Summary of model 3

```
summary(select_3)
```

```
##
## Call:
## lm(formula = MinTemp ~ RainToday + MaxTemp + Pressure3pm + Humidity3pm +
##     WindGustSpeed + Temp3pm + RainToday:Humidity3pm + MaxTemp:Pressure3pm +
##     MaxTemp:Humidity3pm + MaxTemp:WindGustSpeed + Pressure3pm:WindGustSpeed +
##     Pressure3pm:Temp3pm + Humidity3pm:WindGustSpeed + Humidity3pm:Temp3pm,
##     data = weather_can)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2691 -1.6294  0.0556  1.5893  8.0389
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.438e+02  1.122e+02   1.281 0.201070
## RainTodayYes                3.507e+00  1.341e+00   2.616 0.009340 **
## MaxTemp                     3.795e+01  2.531e+01   1.499 0.134797
## Pressure3pm                -1.705e-01  1.090e-01  -1.564 0.118827
## Humidity3pm                 2.545e-01  4.470e-02   5.694 2.86e-08 ***
## WindGustSpeed              -3.155e+00  1.567e+00  -2.014 0.044909 *
## Temp3pm                    -3.679e+01  2.561e+01  -1.436 0.151859
## RainTodayYes:Humidity3pm   -4.286e-02  2.451e-02  -1.749 0.081337 .
## MaxTemp:Pressure3pm        -3.517e-02  2.486e-02  -1.415 0.158191
## MaxTemp:Humidity3pm        -2.637e-02  7.707e-03  -3.421 0.000706 ***
## MaxTemp:WindGustSpeed      -7.057e-03  2.559e-03  -2.757 0.006169 **
## Pressure3pm:WindGustSpeed   3.436e-03  1.519e-03   2.262 0.024372 *
## Pressure3pm:Temp3pm         3.499e-02  2.517e-02   1.390 0.165533
## Humidity3pm:WindGustSpeed -3.212e-03  8.011e-04  -4.010 7.61e-05 ***
## Humidity3pm:Temp3pm         3.091e-02  7.666e-03   4.032 6.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.666 on 313 degrees of freedom
## Multiple R-squared:  0.7948, Adjusted R-squared:  0.7856
## F-statistic:  86.6 on 14 and 313 DF,  p-value: < 2.2e-16
```