Regression Analysis

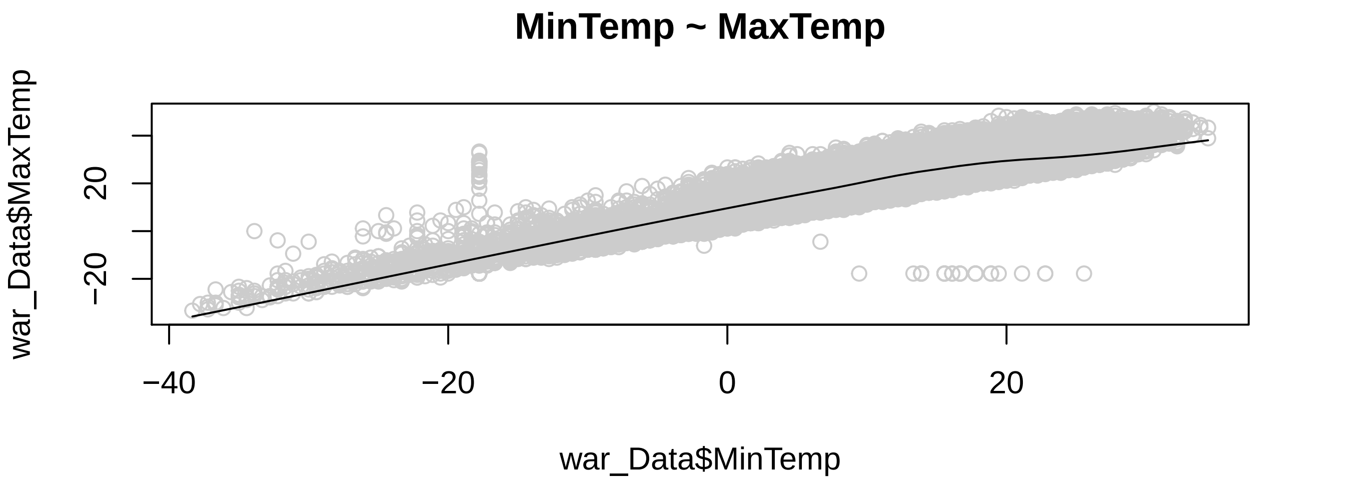
Sarah Bashir

PROJECT

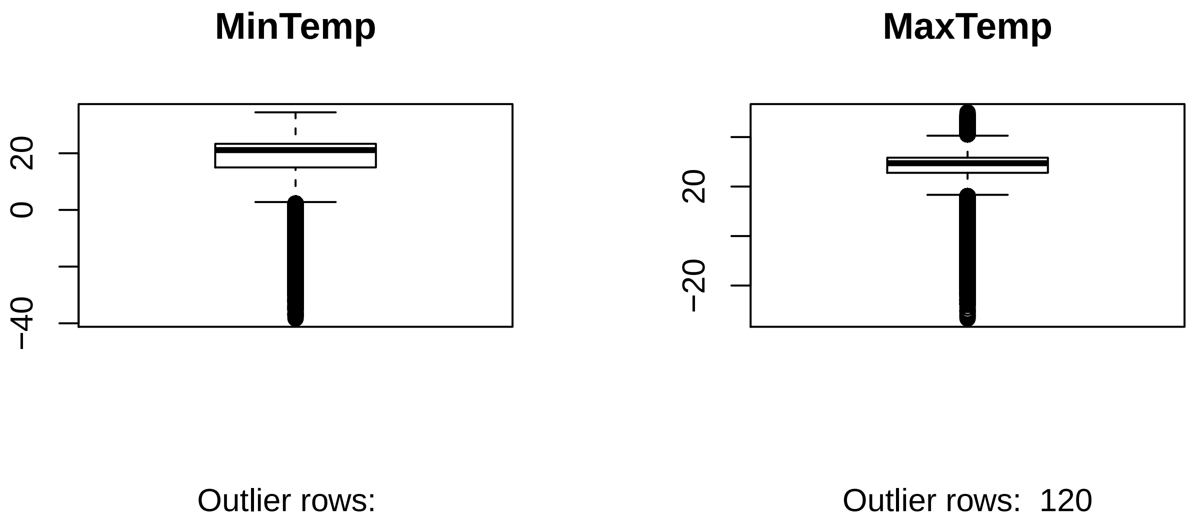
This project looks at multiple data sets taken from kaggle and compares the accuracies of different regression models.

Logistic Regression:

```
war_Data <- readr::read_csv("weather.csv")
#using scatter plot to visualize relationship between minTemp and maxTemp
scatter.smooth(x=war_Data$MinTemp, y=war_Data$MaxTemp, main="MinTemp ~ MaxTemp", col = '#CCCCCC')
```

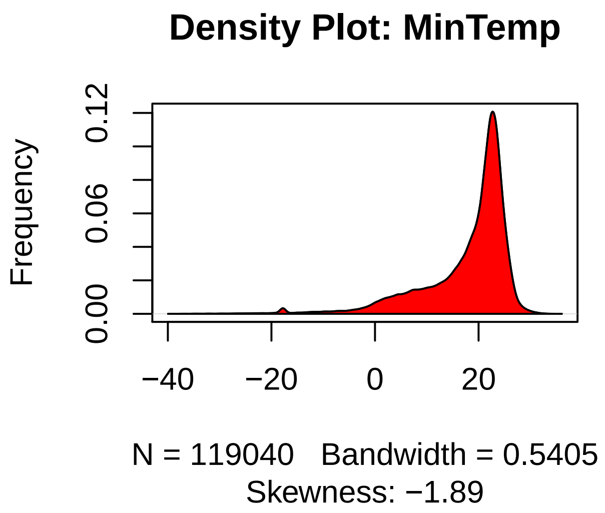


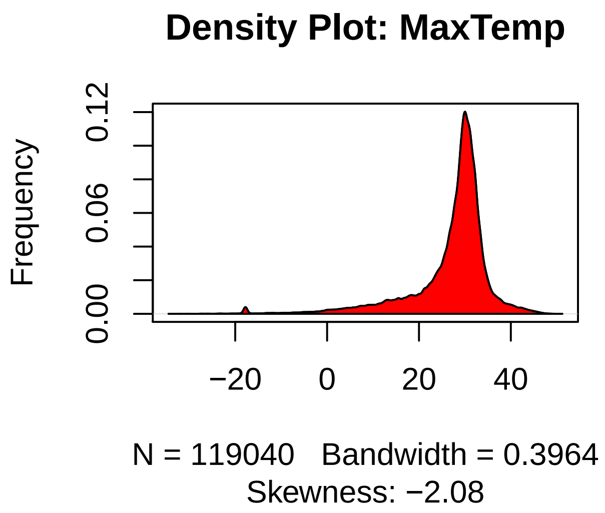
```
#checking for outliers using a box plot
par(mfrow=c(1, 2))  # divide graph area in 2 columns
boxplot(war_Data$MinTemp, main="MinTemp", sub=paste("Outlier rows: ", boxplot.stats(cars$speed)$out))  #
boxplot(war_Data$MaxTemp, main="MaxTemp", sub=paste("Outlier rows: ", boxplot.stats(cars$dist)$out))  #
```



```
#checking if response variable is close to normal using density plot
par(mfrow=c(1, 2))  # divide graph area in 2 columns
```

```r
plot(density(war_Data$MinTemp), main="Density Plot: MinTemp", ylab="Frequency", sub=paste("Skewness:", 
polygon(density(war_Data$MinTemp), col="red")
plot(density(war_Data$MaxTemp), main="Density Plot: MaxTemp", ylab="Frequency", sub=paste("Skewness:", 
polygon(density(war_Data$MaxTemp), col="red")
```





```r
#correlation
cor(war_Data$MinTemp, war_Data$MaxTemp)
```

```
## [1] 0.878
```

We see that there's a strong positive relationship between MinTemp and MaxTemp (close to 1).

```r
#create training and test data
set.seed(47)
trainingIndex <- sample(1:nrow(war_Data), 0.8*nrow(war_Data)) #row indices for training
trainData <- war_Data[trainingIndex, ] #training data
testData <- war_Data[-trainingIndex, ] #test data
#build model on training data
linMod <- lm(MaxTemp ~ MinTemp, data=trainData)
prediction <- predict(linMod, testData) #predict temp
actual<- testData$MaxTemp
#test data RMSE
sqrt(mean(prediction-actual)^2)
```

```
## [1] 0.0104
```

```r
#fit a model to the training data set
linMod2 <- lm(MaxTemp ~ MinTemp, data=trainData)
#predict in-sample
prediction2 <- predict(linMod2, trainData)
#get RMSE for train data
actual2 <- trainData$MaxTemp
sqrt(mean(prediction2-actual2)^2)
```

```
## [1] 6.47e-12
```

```r
summary(linMod)
```

```
##
```

```
## Call:
## lm(formula = MaxTemp ~ MinTemp, data = trainData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -51.96  -2.78  -0.52   2.18  38.40
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.70740    0.03184     336   <2e-16 ***
## MinTemp      0.91851    0.00162     567   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.17 on 95230 degrees of freedom
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.771
## F-statistic: 3.21e+05 on 1 and 95230 DF,  p-value: <2e-16
```

We see model and predictor p value are less than 0.05 so the model is statistically significant.

Now calculate prediction accuracy and error rates:

Look at correlation between actual and predicted values.

```
actual_pred <- data.frame(cbind(actual=testData$MaxTemp, predict=prediction)) #make precited and actual
correlation_Acc <- cor(actual_pred)
#strong positive relationship between actual and predicted values
correlation_Acc
```

```
##         actual predict
## actual   1.000   0.879
## predict  0.879   1.000
```

```
DMwR::regr.eval(actual_pred$actual,actual_pred$predict)
```

```
##   mae   mse  rmse  mape
##  3.17 17.34  4.16   Inf
```

Poisson Regression:

Looking at relationship between temperature and number of bikes.

```
bikes_Data <- readr::read_csv("bikes.csv")
bikes_Data<-as.data.frame(bikes_Data)
```

Will use mean temperature (high - low/2) to predict volume of bikes per day. First find class of each variable and if there are any missing variables. Then will plot to see the relationship between number of bikes and mean temperature.

```
print("Missing values?")
```

```
## [1] "Missing values?"
```

```
sum(is.na(bikes_Data[,c('High Temp (°F)', 'Low Temp (°F)','Total')]))
```

```
## [1] 0
```

```
print("Numeric variables?")
```

```
## [1] "Numeric variables?"
```

```
is.numeric(bikes_Data[,'High Temp (°F)'])
```

```
## [1] TRUE
```

```
is.numeric(bikes_Data[,'Low Temp (°F)'])
```

```
## [1] TRUE
```

```
is.numeric(bikes_Data[,'Total'])
```

```
## [1] TRUE
```

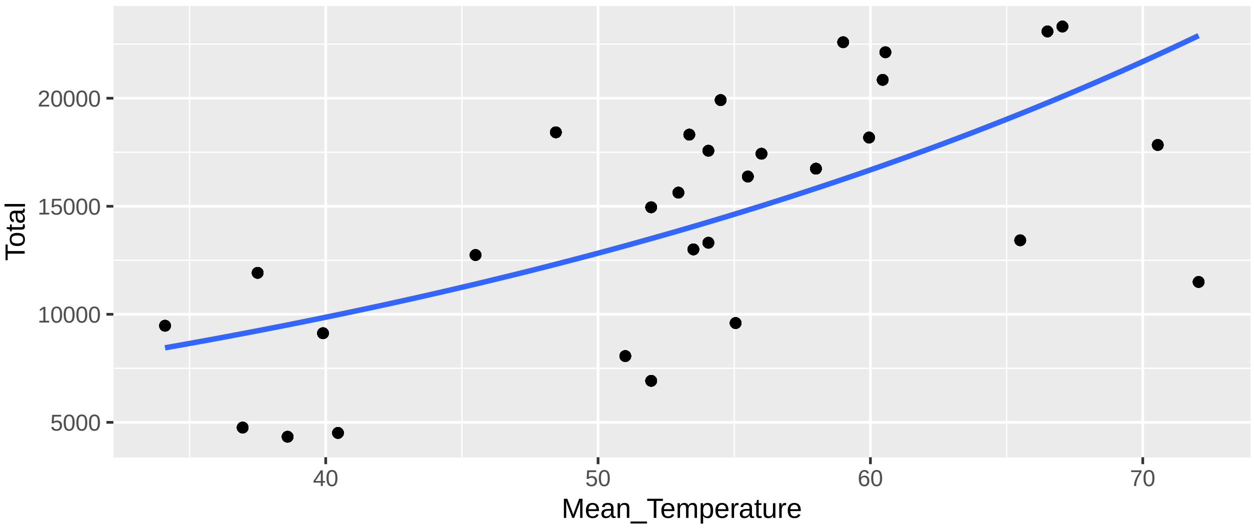
```
print("Integers?")
```

```
## [1] "Integers?"
```

```
all.equal(bikes_Data[,'Total'], as.integer(bikes_Data[,'Total'])) == T
```

```
## [1] TRUE
```

```
#create "Mean temperature".
bikes_Data$Mean_Temperature<-(bikes_Data[,'High Temp (°F)']+bikes_Data[,'Low Temp (°F)'])/2
#Poisson plot
ggplot(bikes_Data, aes(x = Mean_Temperature, y = Total)) +
    geom_point() +
    geom_smooth(method = "glm", #plot poisson regression
    method.args = list(family = "poisson"))
```



```
bikesModel <- glm(bikes_Data$Mean_Temperature ~ bikes_Data$Total, poisson)
cov.model <- vcovHC(bikesModel, type = "HC0")
std.error <- sqrt(diag(cov.model))
r.est <- cbind(Estimate = coef(bikesModel), "Robust SE" = std.error, "Pr(>|z|)" = 2 * pnorm(abs(coef(bi
LL = coef(bikesModel) - 1.96 * std.error,
UL = coef(bikesModel) + 1.96 * std.error)
r.est
summary(bikesModel)
```

Median is 0, deviance residuals are normally distributed. Coeffeicient for Total is close to 0. Residual deviance provides a goodness of fit test for the overall model. Residual deviance is the difference beween the deviance of the current model and the max deviance of the ideal model where the predicted values are the same as those observed. If the residual difference is small enough, the goodness of fit test will not be significant, indicating that the model fits the data. We conclude that the model fits well because the goodness of fit chi-squared test is not statistically significant. If it was, it would indicate that the data did not fit the model well.

The positive coefficient for Mean_Temperature means that as Mean Temeperature increases, the total number of bikes also increases.

This coeffecient is highly significant (p < 2e-16).

Residual deviance is slightly higher than the degrees of freedom, which means we have slight over dispersion. Means that there is extra variance not accounted for by the model.

Aim to refit the model using quasi Poisson errors (see if this will help). Over dispersion is an issue if the residual variance is larger than the conditional mean. The quasi-poisson model will fit an extra dispersion paramater to account for the extra variance.

```r
quasi_Model <- bikesModel <- glm(bikes_Data$Mean_Temperature ~ bikes_Data$Total, quasipoisson)
pchisq(quasi_Model$deviance, df=quasi_Model$df.residual, lower.tail = FALSE)
```

```
## [1] 0.193
```

```r
summary(quasi_Model)
```

```
##
## Call:
## glm(formula = bikes_Data$Mean_Temperature ~ bikes_Data$Total,
##     family = quasipoisson)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.042  -0.586  -0.141   0.312   2.978
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.65e+00   2.95e-02   123.7   <2e-16 ***
## bikes_Data$Total 2.22e-05   1.83e-06    12.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.11)
##
##     Null deviance: 392.36  on 209  degrees of freedom
## Residual deviance: 225.47  on 208  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

We see the dispersion paramater is slightly higher. This paramater tells us how many times larger the variance is than the mean. In this case, the difference between the residual and null deviance is the same as the original model and the dispersion paramater is higher, so the original Poisson model was better.

```r
set.seed(35)
inTrain <- caret::createDataPartition(y=bikes_Data$Total, p=.8, list=FALSE)
BR.train <- bikes_Data[inTrain,]
BR.test <- bikes_Data[-inTrain,]
#build model on training data
#in sample accuracy
```

```r
#need to convert log odds to a probability
poisMod <- glm(Mean_Temperature ~ Total, data = BR.train, family = poisson)
poisPrediction <- predict(poisMod, BR.train) #predict total number of bikes
poisPred <- gtools::inv.logit(poisPrediction)
predictionProbs <- data.frame(prob.poisson = poisPred)
mean(predictionProbs[,1])
```

```
## [1] 0.981
```

```r
#out of sample performance (on test data)
poisModOOS <- glm(Mean_Temperature ~ Total, data = BR.train, family = poisson)
poisPredictionOOS <- predict(poisModOOS, BR.test) #predict total number of bikes
poisPredOOS <- gtools::inv.logit(poisPredictionOOS)
predictionProbsOOS <- data.frame(prob.poissonOOS = poisPredOOS)
mean(predictionProbsOOS[,1])
```

```
## [1] 0.982
```

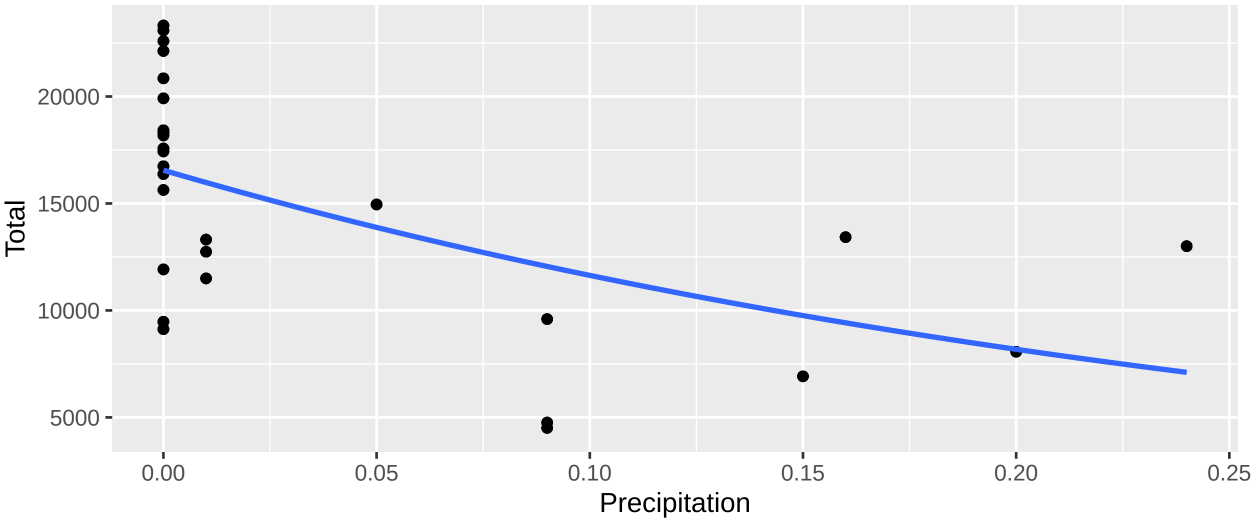
```r
pchisq(poisModOOS$deviance, df=poisModOOS$df.residual, lower.tail = FALSE)
```

```
## [1] 0.0899
```

P-value of 0.0899 suggests that a residual deviance as large or larger than what we observed under the model in poisson model is highly likely. Suggests that the model is of adequate fit, since the chi-squared test was not statistically significant.

Looking at Precipitation:

```r
set.seed(33)
#Look at precipatation
bikes_Data[,"Precipitation"]<-as.numeric(bikes_Data[,"Precipitation"]) #get rid of n/a's in dataset
ggplot(bikes_Data, aes(x = Precipitation, y = Total)) +
    geom_point() +
    geom_smooth(method = "glm", #plot regression
    method.args = list(family = "poisson"))
```



```r
#build model on training data
#in sample accuracy
#need to convert log odds to a probability
```

```
#poisModel <- glm(Precipitation ~ Total, data = BR.train, family = poisson)
#poisPrediction <- predict(poisModel, BR.train) #predict total number of bikes
#poisPred <- gtools::inv.logit(poisPrediction)
#predictionProbs <- data.frame(prob.poisson = poisPred)
#mean(predictionProbs[,1])
#out of sample performance (on test data)
#poisModOOS <- glm(Precipitation ~ Total, data = BR.train, family = poisson)
#poisPredictionOOS <- predict(poisModOOS, BR.test) #predict total number of bikes
#poisPredOOS <- gtools::inv.logit(poisPredictionOOS)
#predictionProbsOOS <- data.frame(prob.poissonOOS = poisPredOOS)
#mean(predictionProbsOOS[,1])
```

We see that a higher mean temperature means more bikes while a higher level of preciptation means a lower amount of bikes (not a very good model, however as the data doesn't fit a pattern well).