CWRU DSCI351-351m-451: Lab Exercise LE7 NAME

Inference, Linear Regression, Timeseries Analysis

Prof.:Roger French, Paul W. Leu, TA: Raymond Wieser, Sameera Nalin Venkat

04 December, 2022

Contents

7.0.1 LE7, 10 points, questions.

Coding style: 1 point

- Q1 - OIS: Numerical inference, 1 pt.
- Q2 - OIS: Linear regression, 1 pt.
- Q3 - OIS: Logistic regression, 1 pt.
- Q4 - Logistic regression: Palmer's penguins, 3 pts.
- Q5 - Houston crime data, 3 pts.

```
library(tidyverse)
```

7.0.1.1 Lab Exercise (LE) 7

7.1 Q1. OIS: Numerical inference (1 point)

OIS v3 5.44: Teaching descriptive statistics.

A study compared five different methods for teaching descriptive statistics.

- The five methods were
  - traditional lecture and discussion,
  - programmed textbook instruction,
  - programmed text with lectures,
  - computer instruction,
  - and computer instruction with lectures.
- 45 students were randomly assigned,
  - 9 to each method.
- After completing the course,
  - students took a 1-hour exam.

What are the hypotheses for evaluating

- if the average test scores are different
  - for the different teaching methods?

What are the degrees of freedom associated with the F -test

- for evaluating these hypotheses?

Suppose the p-value for this test is 0.0168.

- What is the conclusion?

ANSWER:

The hypotheses for evaluating if the average test scores are different for the different teaching methods:

Null Hypothesis (H_0): all means across all 5 methods are the same ($\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$)

Alternative Hypothesis (H_a): at least one pair of the means is the same

The degrees of freedom associated with the F-test for evaluating these hypotheses:

Degrees of freedom for treatment: 5-4 = 1

Degrees of freedom for error: 45-5 = 40

The conclusion for this test where the p-value = 0.0168:

Assuming standard significance level of $\alpha = 0.05$, reject the null hypothesis becuase p-value $< \alpha$ => 0.0168 < 0.05

7.2 Q2. OIS: Linear regression (1 point)

OIS v3 7.12: Trees.

This dataset

- shows the relationship between
  - height,
  - diameter (girth),
  - and volume of timber
- in 31 felled black cherry trees.

The diameter of the tree is measured

- 4.5 feet above the ground.

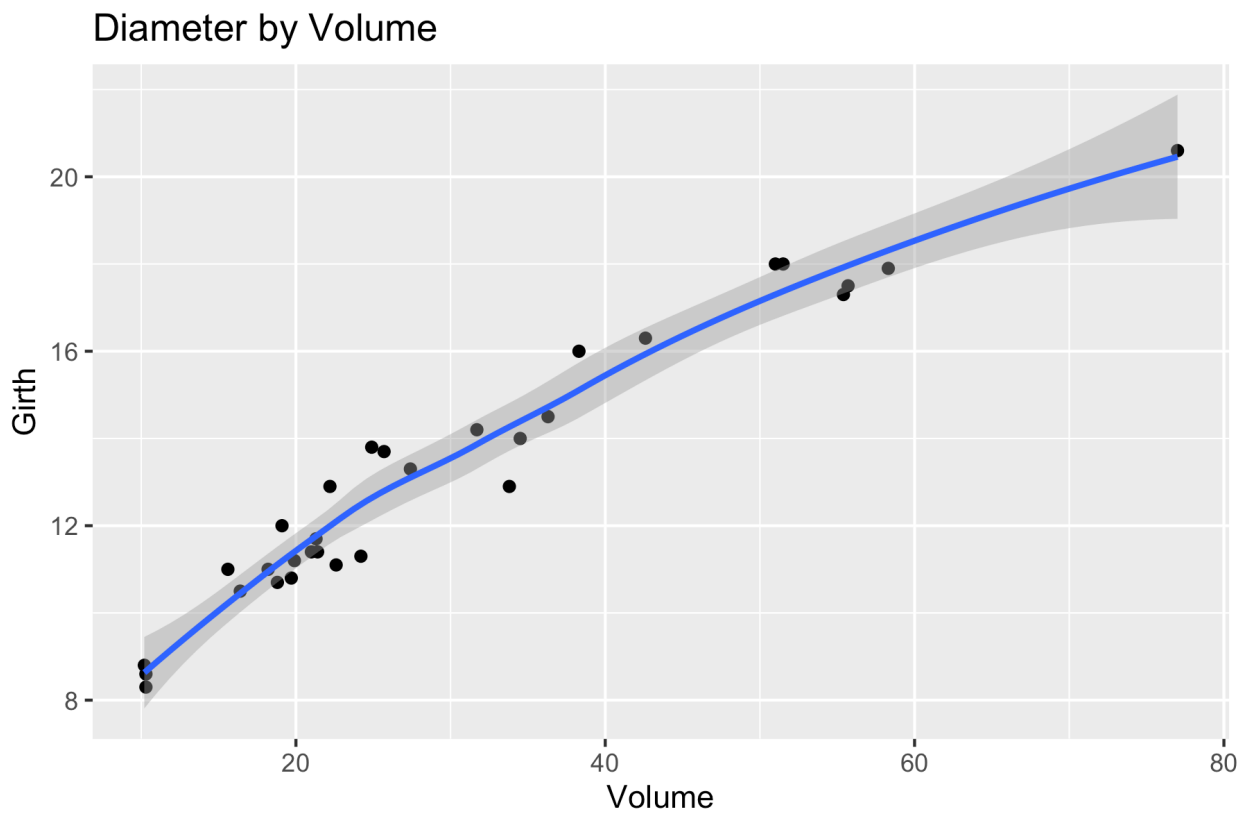
```
library(tidyverse)
library(ggplot2)

data(trees)
summary(trees)
```

```
##      Girth           Height       Volume
##  Min.   : 8.30   Min.   :63   Min.   :10.20
##  1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
##  Median :12.90   Median :76   Median :24.20
##  Mean   :13.25   Mean   :76   Mean   :30.17
##  3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
##  Max.   :20.60   Max.   :87   Max.   :77.00
```

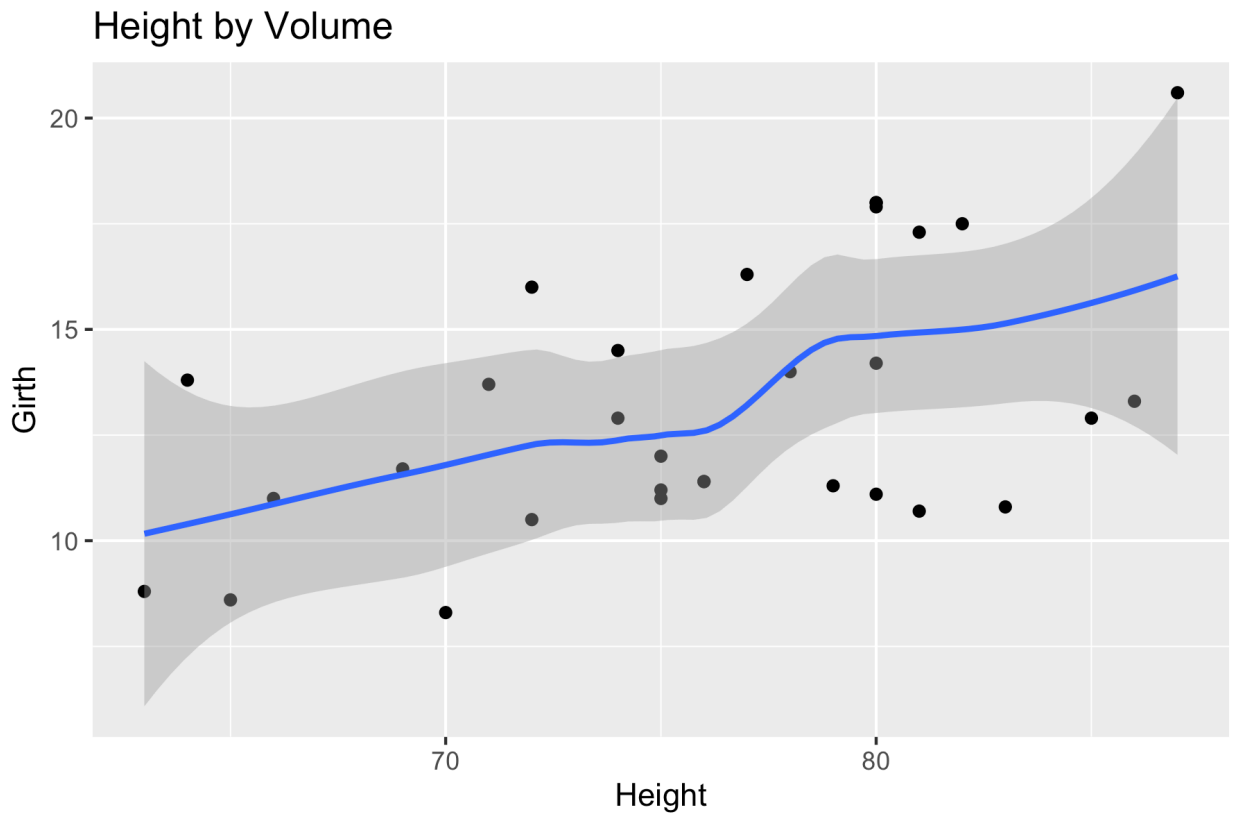
Visualize the relationships in the data

- height vs. volume
- diameter vs. volume
- your visualizations should contain all important info (labels, etc.)
- (hint: scatterplots)

```
ggplot(trees, aes(x = Volume, y = Girth)) + ggtitle("Diameter by Volume") + geom_point() + geom_smooth()
```




```
ggplot(trees, aes(x = Height, y = Girth)) + ggtitle("Height by Volume") + geom_point() + geom_smooth()
```



Let's answer questions using linear regression

- Describe the relationship between volume and height of these trees.
- Describe the relationship between volume and diameter of these trees.

Volume and Diameter (Girth): Using the geom_smooth function, there is a positive linear association between volume and diameter.

Volume and Height: Using the geom_smooth function, there is a positive linear association between volume and height

Summarizing the model results from the `lm()` function

- will provide valuable numerical insights.

```
girth_lm <- lm(Volume ~ Girth, trees)
height_lm <- lm(Volume ~ Height, trees)

girth_lm
```

```
##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## Coefficients:
## (Intercept)        Girth
##     -36.943        5.066
```

```
height_lm
```

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Coefficients:
## (Intercept)       Height
##     -87.124        1.543
```

Volume and Diameter (Girth): using the lm function, we can see that the girth slope coefficient is positive (+5.066), meaning positive association between girth and volume

Volume and Height: using the lm function, we can see that the height slope coefficient is positive (+1.543), meaning positive association between height and volume

The diameter seems to have a stronger positive correlation with volume than height

Suppose you have height and diameter measurements

- for another black cherry tree.

Which of these variables would be preferable to use

- to predict the volume of timber in this tree
- using a simple linear regression model?

Explain your reasoning

As mentioned before, when looking at the line of association and lower spread within the graph, and comparing with the outputs between the lm function, it's more preferable to use diameter measurements to predict the volume of timber for the tree using a linear regression model.

7.3 Q3. OIS: Logistic regression (1 point)

OIS v3 8.16 Challenger disaster, Part I.

On January 28, 1986, a routine launch was anticipated for the Challenger space shuttle.

Seventy-three seconds into the flight, disaster happened:

- the shuttle broke apart,
- killing all seven crew members on board.

An investigation into the cause of the disaster focused on a critical seal called an O-ring, and it is believed that damage to these O-rings during a shuttle launch may be related to the ambient temperature during the launch.

The orings.txt file in the `data` subfolder

- contains data on

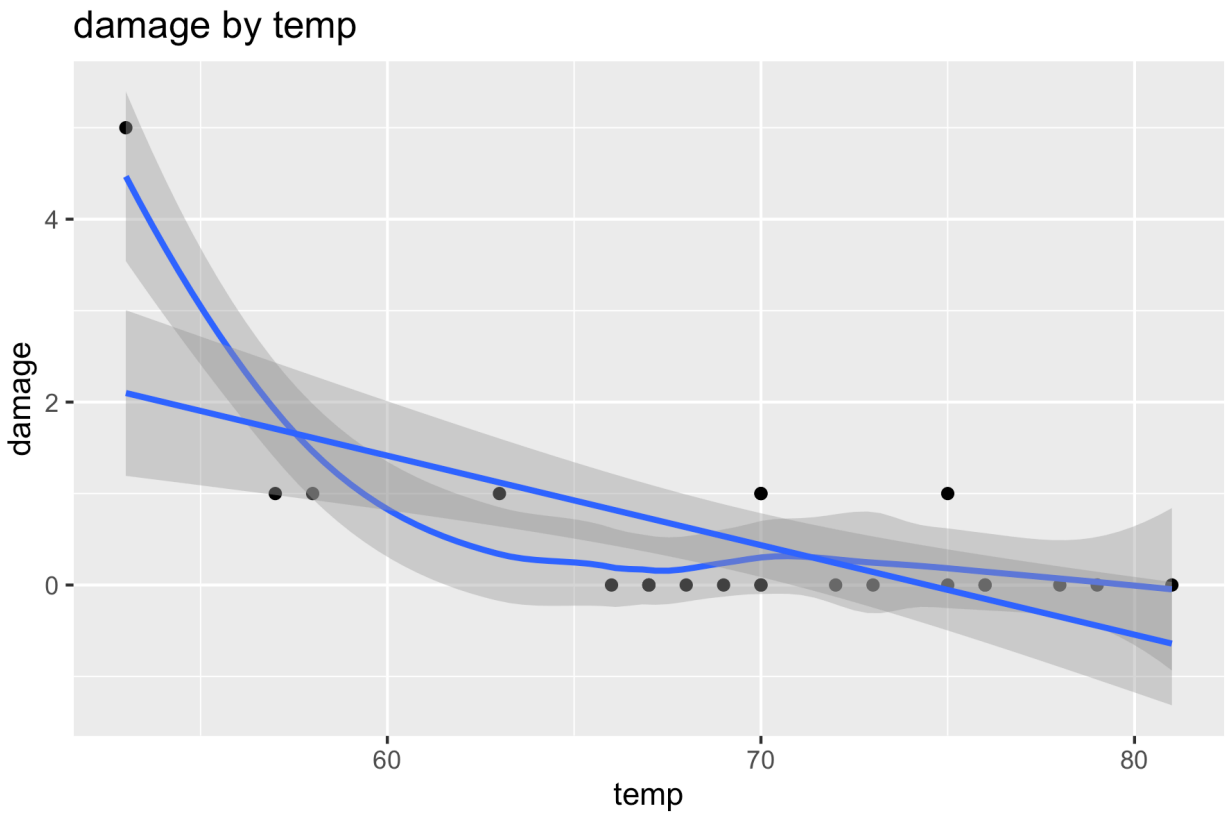
  - the temperature and
  - number of damaged O-rings

- for 23 shuttle missions,
- where the mission order is based on
  - the temperature at the time of the launch.
- `Temp` gives the temperature in Fahrenheit,
- `Damaged` represents the number of damaged O-rings.
- There are 6 O-rings total,
  - so the number of undamaged O-rings can be calculated.

```
o_rings <- read.table('data/orings.txt', header = TRUE)
```

Visualize the data. what relationships do you observe between temperature and failure?

```
ggplot(o_rings, aes(x = temp, y = damage)) + ggtitle("damage by temp") + geom_point() + geom_smooth() +
```



it can be seen from the plot that there is a converging (exponentially decaying/logarithmic) relationship between temperature and damage (failure)

Create a logistic regression model.

- classify each case as having either damaged or undamaged O-rings (1 or 0)
  - a binary “failure” variable will help us

  - determine probability of failure as a result

- use temperature as a predictor
- use the `glm()` function
- display the summary statistics of your model

```
logistic_challenger <- glm(temp~damage, data = o_rings)
summary(logistic_challenger)
```

```
##
## Call:
## glm(formula = temp ~ damage, data = o_rings)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.3919   -4.4747    0.4426    3.9426    9.4426
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   71.557      1.272  56.248  < 2e-16 ***
## damage        -4.166      1.096  -3.801  0.00104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 30.90637)
##
##     Null deviance: 1095.65  on 22  degrees of freedom
## Residual deviance:  649.03  on 21  degrees of freedom
## AIC: 148.09
##
## Number of Fisher Scoring iterations: 2
```

Based on the model, do you think concerns regarding O-rings are justified? Explain. what does the p-value tell you?

Damage estimate is -4.166, negative exponential or logarithmic relationship. The logistic regression indicates a lower amount of damaged O-rings for higher temperatures. From the p-value ($0.00104 < \alpha = 0.05$), meaning there is an logistic association between damaged O-rings and temperature.

What assumption has to be made for logistic regression to be valid in this case?

The assumption that the relationship with the logarithmic value of the relationship between the damaged O-rings and temperature being linear and independent variable outcomes has to be made for logistic regression to be valid. From what we can see in the graph and the glm function, this assumption has been satisfied and makes the logistic regression being performed valid.

7.4 Q4. Logistic regression: Palmer's penguins (3 points)

Let's make some logistic models using Palmer's penguins.

- we've looked at regression with a single predictor
- let's make a logistic model with multiple predictors

  - we're increasing the dimensions of the model
  - in order to get more information out of the data
- we want to create a model that can predict penguin species

We will used the package `nnet`

- This package is used for machine learning
- But it also has a function
  - `nnet::multinom()`
    - Which works like `stats::lm()`
    - see `?nnet::multinom()` for more help
- This is because logistic models
  - Are used to baseline more complex Machine Learning Models
    - For performance
- `nnet::multinom()` is used to build multiple logistic models
  - Which can be used to classify multiple outputs
  - Instead of a binary classification model like a logistic model

```
library(nnet)
library(caret)
glimpse(palmerpenguins::penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

```
df_penguins <- palmerpenguins::penguins
```

7.4.1 Q4.1 Setup Training & Testing Dataframes

Processing the data

- divide the data into training and testing datasets
  - Using `caret::createDataPartiton()`
  - Which will divide the data into groups
    - So we can "train" the model on one subset
    - And use the other subset to "test" the models accuracy

```r
# perform and 80/20 split on penguins data
# goal: determine species of penguins
df_penguins$species <- as.factor(df_penguins$species)
partition <- createDataPartition(df_penguins$species,
                                 p = 0.8,
                                 list = FALSE)

penguins_train <- df_penguins[partition, ]
penguins_test <- df_penguins[-partition, ]
test_species <- penguins_test$species
penguins_test <- penguins_test[ , -1]

glimpse(penguins_train)
```

```
## Rows: 277
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.2, 34.1, 42.0, 37.8, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 19.6, 18.1, 20.2, 17.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 195, 193, 190, 186, 180, 182~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 4675, 3475, 4250, 3300, ~
## $ sex               <fct> male, female, female, NA, female, male, NA, NA, NA, ~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

```r
glimpse(penguins_test)
```

```
## Rows: 67
## Columns: 7
## $ island            <fct> Torgersen, Torgersen, Torgersen, Biscoe, Biscoe, Dre~
## $ bill_length_mm    <dbl> 39.3, 38.9, 46.0, 35.3, 37.9, 37.2, 37.5, 36.0, 42.3~
## $ bill_depth_mm     <dbl> 20.6, 17.8, 21.5, 18.9, 18.6, 18.1, 18.9, 17.9, 21.2~
## $ flipper_length_mm <int> 190, 181, 194, 187, 172, 178, 179, 190, 191, 186, 19~
## $ body_mass_g       <int> 3650, 3625, 4200, 3800, 3150, 3900, 2975, 3450, 4150~
## $ sex               <fct> male, female, male, female, female, male, NA, female~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

7.4.2 Q4.2 Build a logistic regression model

Build a logistic model

- To predict species
- Based on all the predictors in the dataset
  - The short hand way of doing this is to use
  - `model_function_call(Response_Variable ~ .)`
  - Where the `.` will automatically use the rest of the columns
    - As predictors

Once you have obtained your model object

- It's time to use `stats::predict()`

  - Which takes in a model object
  - Then applies it to new data (your testing subset)

- Different model types can have different prediction classes
  - Here we are trying to predict class

```
set.seed(69)
# Build your model
penguins_model <- multinom(species ~ ., penguins_train, model = FALSE)
```

```
## # weights:  30 (18 variable)
## initial  value 293.329481
## iter  10 value 35.848635
## iter  20 value 1.020309
## iter  30 value 0.012341
## final  value 0.000000
## converged
```

```
cat("\nmodel summary: \n")
```

```
##
## model summary:
```

```
summary(penguins_model)
```

```
## Call:
## multinom(formula = species ~ ., data = penguins_train, model = FALSE)
##
## Coefficients:
##           (Intercept) islandDream islandTorgersen bill_length_mm bill_depth_mm
## Chinstrap  -0.4382546   152.97104        42.00299       37.51762     -49.44851
## Gentoo      0.5235722   -86.42704       -63.43538       29.57834     -55.73298
##           flipper_length_mm body_mass_g   sexmale        year
## Chinstrap          3.784241 -0.13527829 -45.28759 -0.55042511
## Gentoo            -4.179658  0.09571021 -61.15792  0.05984267
##
## Std. Errors:
##            (Intercept)  islandDream islandTorgersen bill_length_mm
## Chinstrap 3.319707e-02 3.319707e-02    6.012541e-36   2.082527e-01
## Gentoo    1.406919e-16 6.234319e-24    1.859263e-42   6.518544e-15
##           bill_depth_mm flipper_length_mm  body_mass_g      sexmale
## Chinstrap  2.181273e-01      6.018649e+00 3.277977e+01 6.234917e-24
## Gentoo     1.985879e-15      3.057792e-14 6.177830e-13 6.234918e-24
##                   year
## Chinstrap 6.708131e+01
## Gentoo    2.826554e-13
##
## Residual Deviance: 3.439335e-07
## AIC: 36
```

```
# Predict classes
predictions <- predict(penguins_model, penguins_test)

# predictions vs actual
cat("\npredicted: \n")
```

```
##
## predicted:
```

```
predictions
```

```
##  [1] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    <NA>
##  [8] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [15] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [22] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [29] Adelie    Adelie    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [36] Adelie    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [43] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [50] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Chinstrap Chinstrap
## [57] Adelie    Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap
## [64] Chinstrap Chinstrap Chinstrap Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

```
cat("\nactual: \n")
```

```
##
## actual:
```

```
test_species
```

```
##  [1] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
##  [8] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [15] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [22] Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
## [29] Adelie    Adelie    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [36] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [43] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Gentoo
## [50] Gentoo    Gentoo    Gentoo    Gentoo    Gentoo    Chinstrap Chinstrap
## [57] Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap Chinstrap
## [64] Chinstrap Chinstrap Chinstrap Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

ANSWER: Looking at the output of the neural net model, we can see all the metadata for training, what general activation function (softmax function) is used, and other inputs that can be modified (weights as well). Using a seed here to ensure same responses after multiple reruns

7.4.3 Q4.3 Evaluate accuracy on your test data

Evaluate the accuracy of your model against test data

- create a confusion matrix to evaluate your results
- using `caret::confusionMatrix()`
  - This compares the predicted class against the actual class
  - Which shows how the model classified the data

```
# comparison of predictions vs actual
confusionMatrix(predictions, reference = test_species)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Adelie Chinstrap Gentoo
##   Adelie        29         1      1
##   Chinstrap      0        12      0
##   Gentoo         0         0     23
##
## Overall Statistics
##
##                Accuracy : 0.9697
##                  95% CI : (0.8948, 0.9963)
##     No Information Rate : 0.4394
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.952
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Adelie Class: Chinstrap Class: Gentoo
## Sensitivity                 1.0000           0.9231        0.9583
## Specificity                 0.9459           1.0000        1.0000
## Pos Pred Value              0.9355           1.0000        1.0000
## Neg Pred Value              1.0000           0.9815        0.9767
## Prevalence                  0.4394           0.1970        0.3636
## Detection Rate              0.4394           0.1818        0.3485
## Detection Prevalence        0.4697           0.1818        0.3485
## Balanced Accuracy           0.9730           0.9615        0.9792
```

Are there any things that were commonly misclassified?

- Why do you think the model had trouble with these?
- What can be done to improve this model?

ANSWER:

The model was able to predict the species with 96.97% (close to 100%) accuracy. There wasn't really anything commonly misclassified, and looking at the statistics from the confusion matrix, only 2 of the 13 variables were improperly classified (2 Chinstraps were classified as Adelie), making chinstrap have a sensitivity at 84%. Probably one of the best ways to improve the model is setting custom initial weights when training to ensure quick convergence, thus eliminating the weights to be set at random, which may or may not be good.

7.5 Q5 Houston Crime Reports

We will be working with the Huston crime data file provided by the ggmap package, `ggmap::crimes`.

This CSV file contains the location (latitude and longitude) for crimes reported from January 2010 - August 2010

```
library(ggmap)
rgdal_show_exportToProj4_warnings = "none"
library(sp)
library(rgdal)
library(leaflet)
library(lubridate)
library(RColorBrewer)
library(classInt)
library(tidyverse)
library(Rfast)
```

7.5.1 Q5.1 EDA to identify trends

Exploratory Data Analysis (EDA)

Let's do some exploratory data analysis on the data we'll start with the temporal aspect of the crime data. What can we say about **when** people commit crimes?

What trends do you see looking at different time frames?

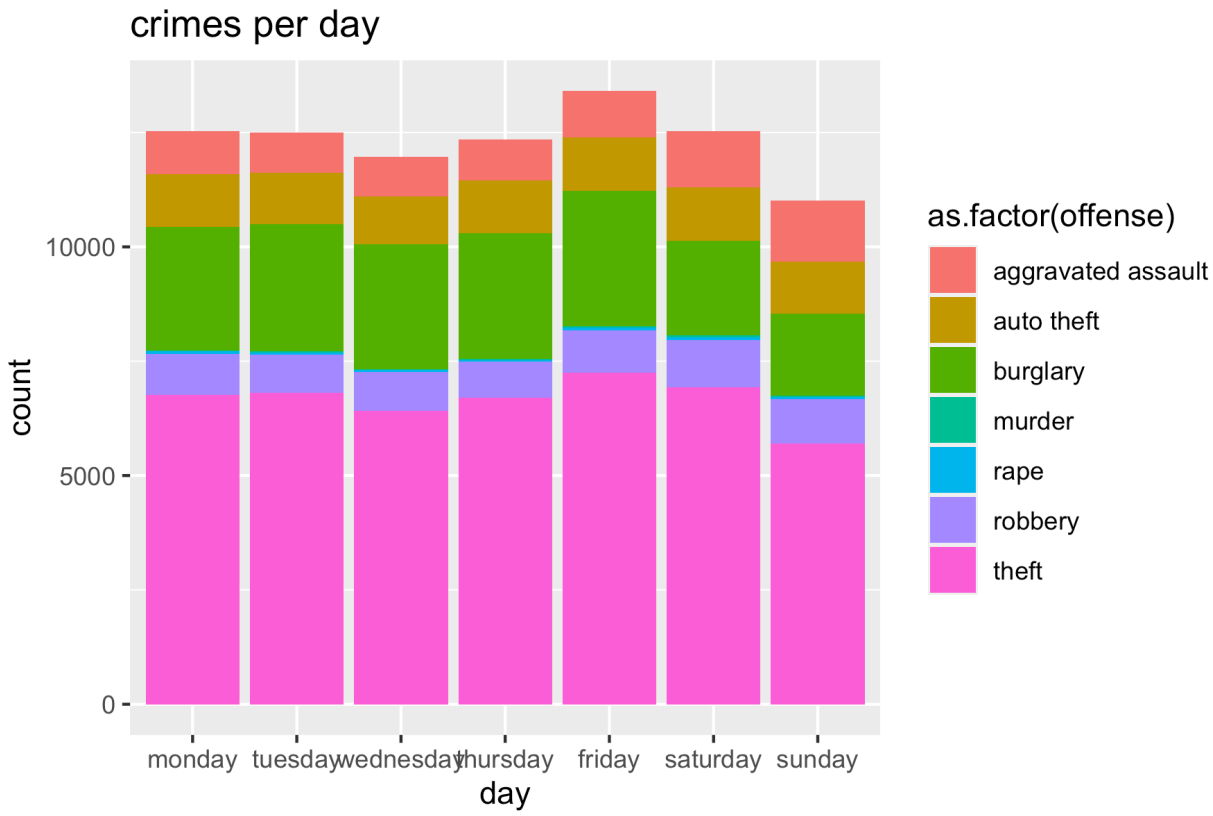
- what months have particularly high crime rates?
- what times of day have increased crime rates?
- what days of the week have higher crime rates?
- produce three different visuals that represent each of these trends.

(hint: histograms are helpful for showing distributions)

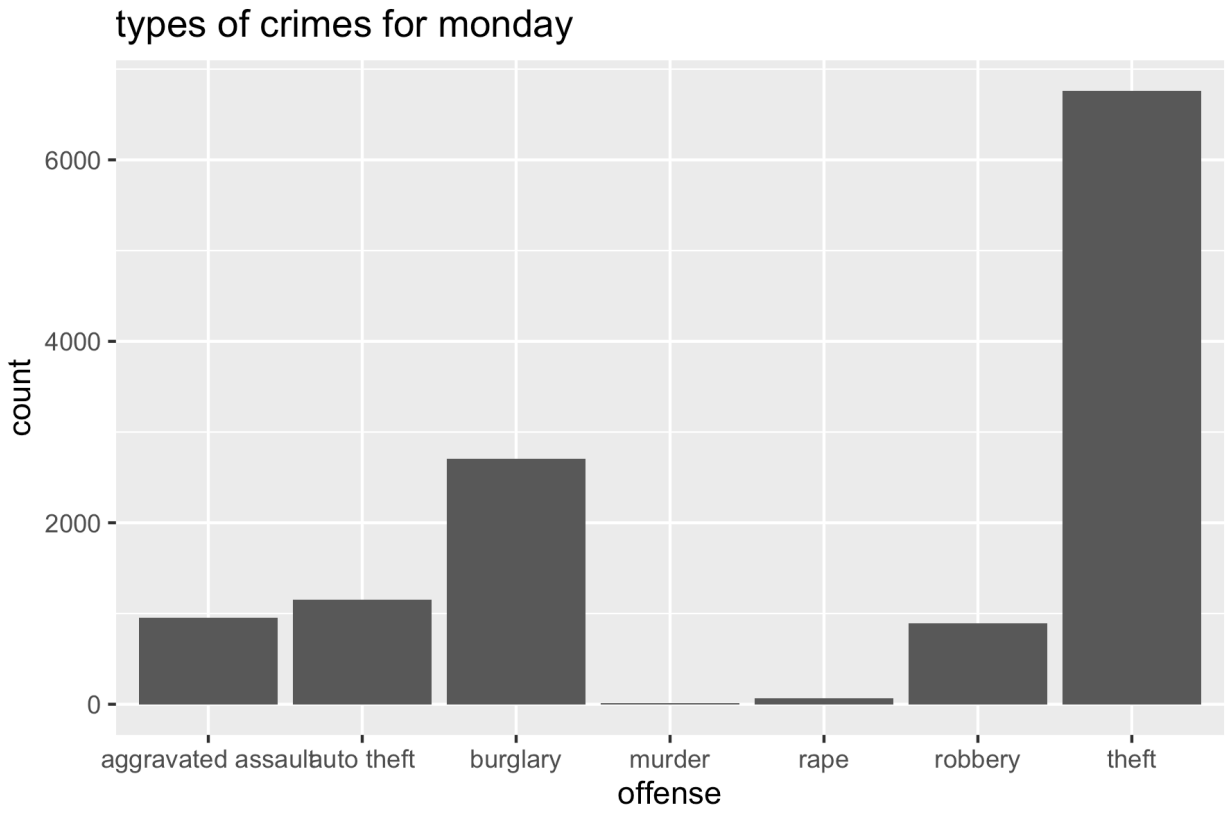
Which of these trends could you have predicted? Does anything surprise you?

Are there any relationships between types of crime and time of day? Produce a stacked histogram and comment on the results

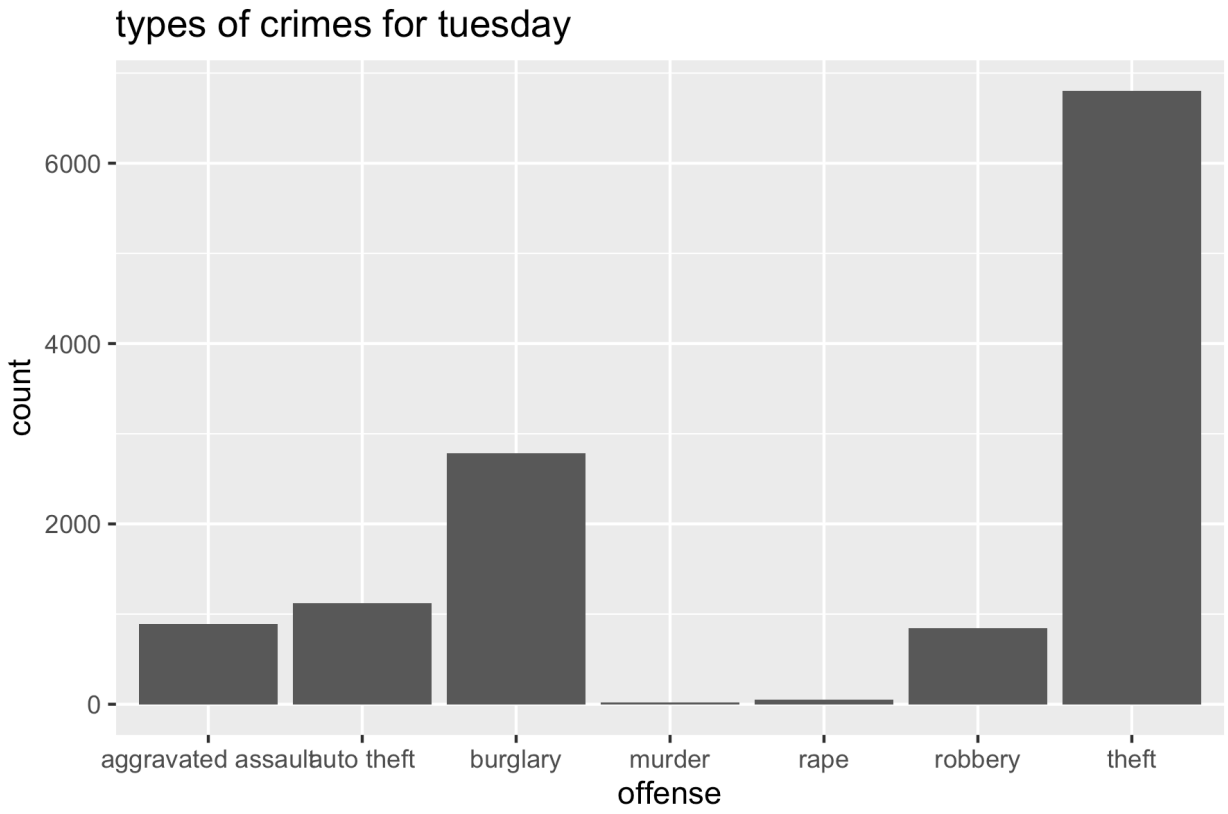
```
# Load Data
houston_crime_report <- ggmap::crime
# fields:
# time, date, hour, month, day
# premise, offense, beat, number
# block, street, type, suffix, location, address
# lon, lat
# copy:
crime_report <- houston_crime_report

# Crime per Day
ggplot(crime_report, aes(x = day, fill = as.factor(offense))
       ) + geom_histogram(stat = "count"
       ) + ggtitle("crimes per day"
       ) + xlab("day"
       )
```



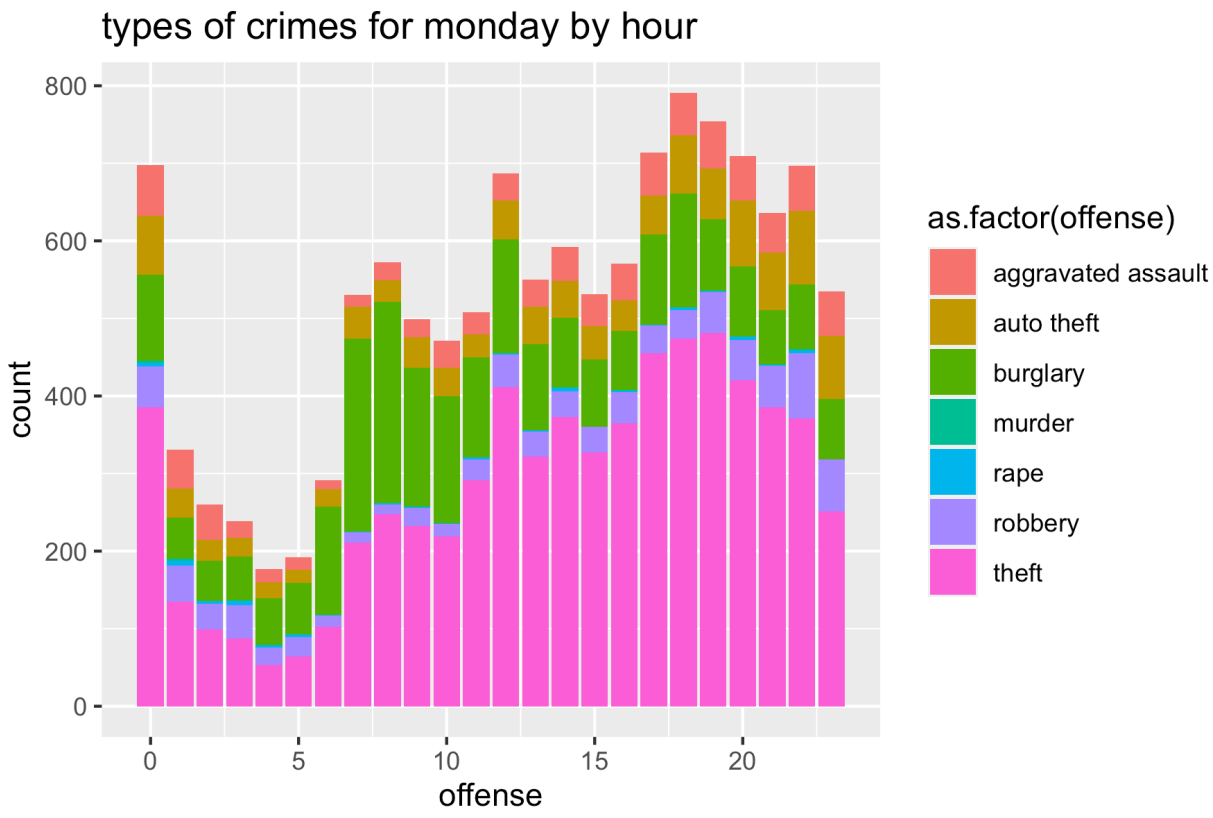

```
# Lets Do some EDA
# Looking at the types of Crime on a given day
plot_desired_crimes_for_day <- function(desired_day) {
  crimes_for_day <- subset(crime_report, day == desired_day)
  crimes_tit <- paste("types of crimes for", desired_day, sep = " ")
  ggplot(crimes_for_day, aes(x = offense)
         ) + geom_histogram(stat = "count"
         ) + ggtitle(crimes_tit
         ) + xlab("offense")
}

plot_desired_crimes_for_day("monday")
```



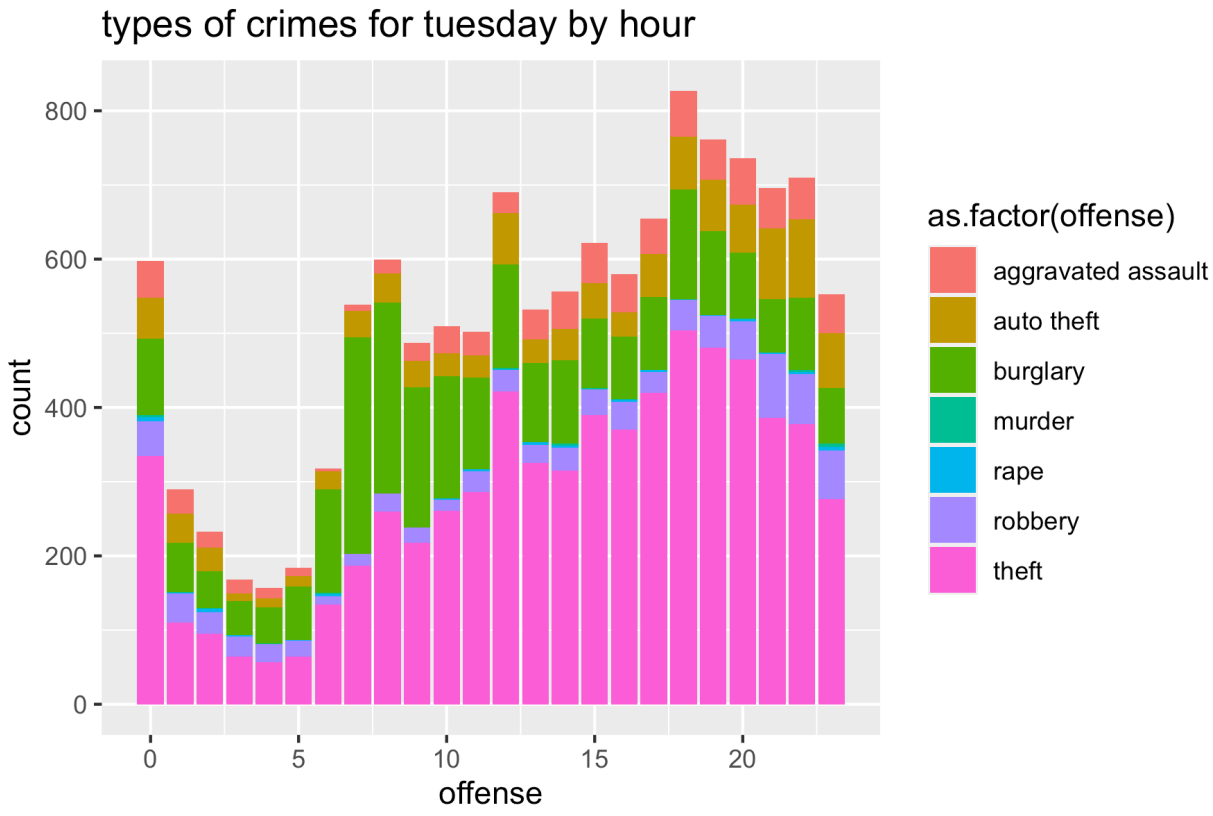

```
plot_desired_crimes_for_day("tuesday")
```



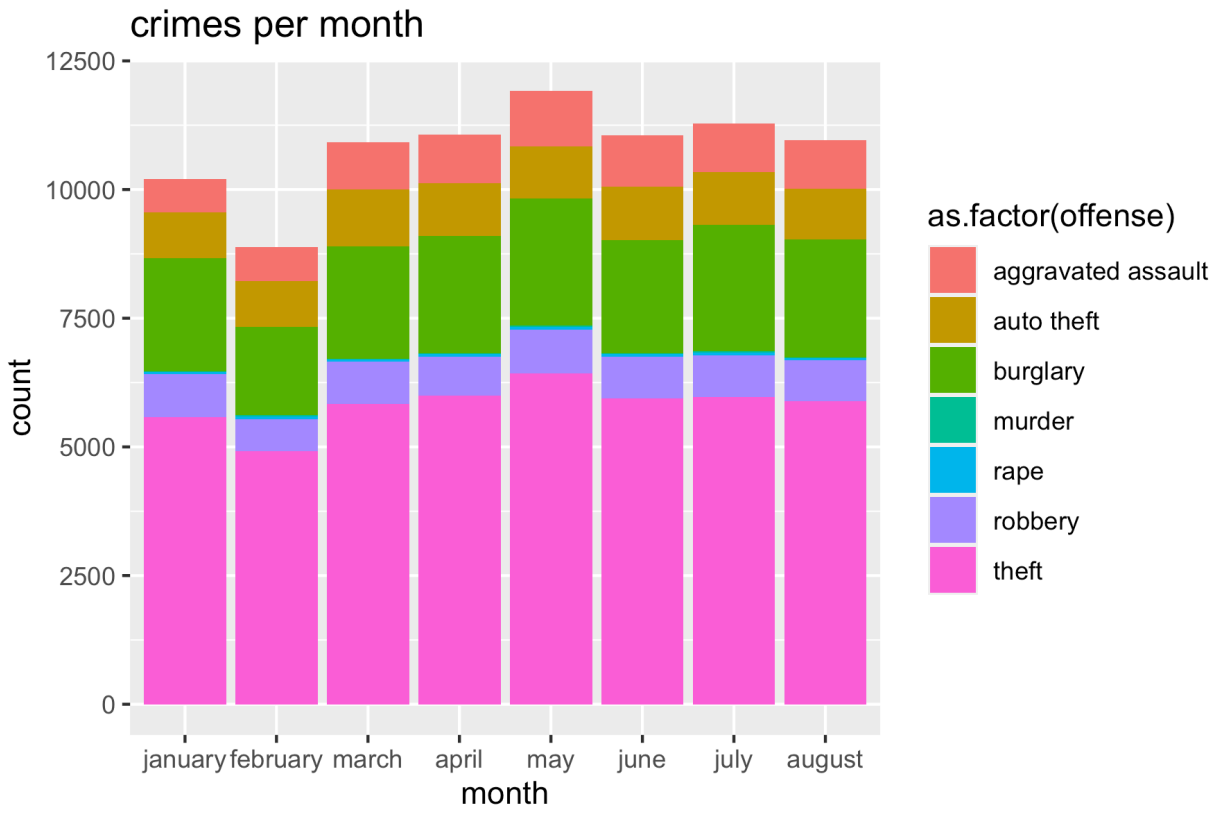

```
plot_correlation_for_day <- function(desired_day) {
  crimes_for_day <- subset(crime_report, day == desired_day)
  crimes_tit <- paste("types of crimes for",
                      desired_day,
                      "by hour",
                      sep = " ")
  ggplot(crimes_for_day, aes(x = hour, fill = as.factor(offense))
         ) + geom_histogram(stat = "count", position = "stack"
         ) + ggtitle(crimes_tit
         ) + xlab("offense")
}

plot_correlation_for_day("monday")
```



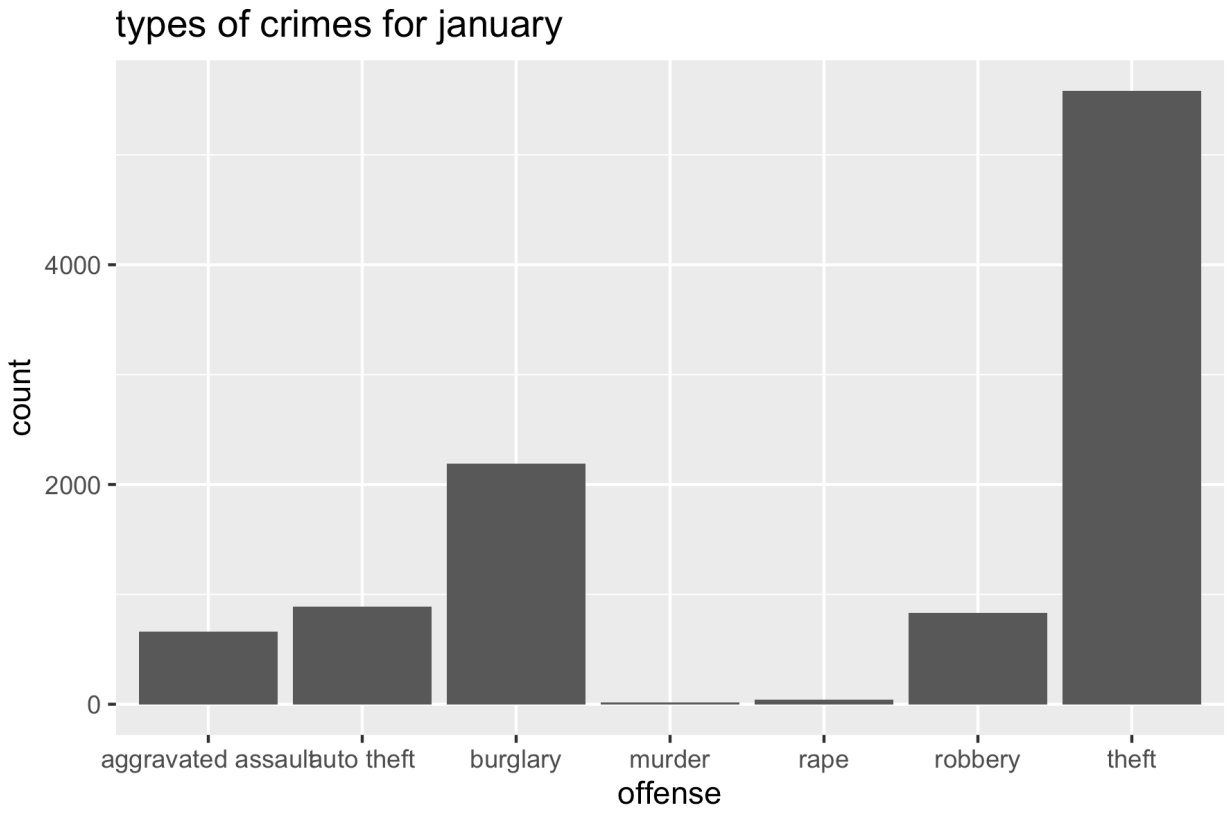

```
plot_correlation_for_day("tuesday")
```



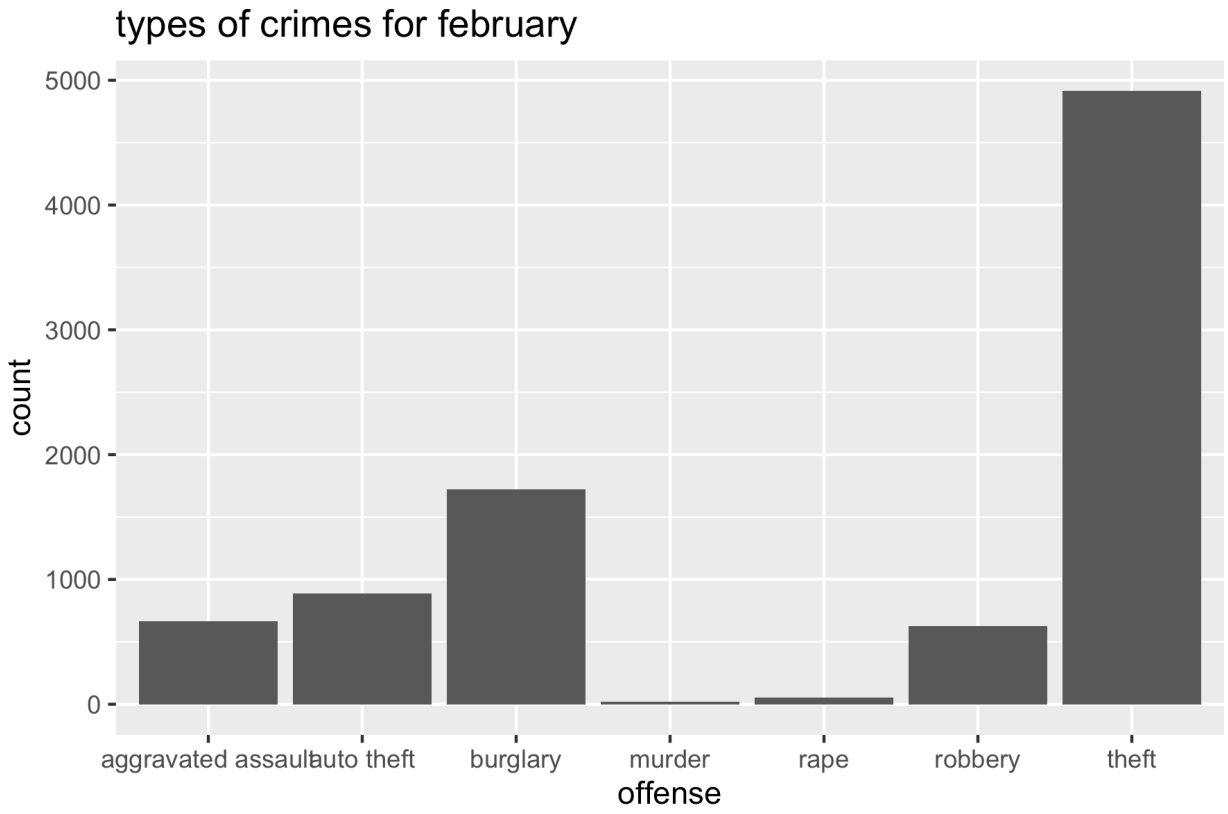

```
# Now lets look at the Month
ggplot(crime_report, aes(x = month, fill = as.factor(offense))
       ) + geom_histogram(stat = "count"
       ) + ggtitle("crimes per month"
       ) + xlab("month")
```



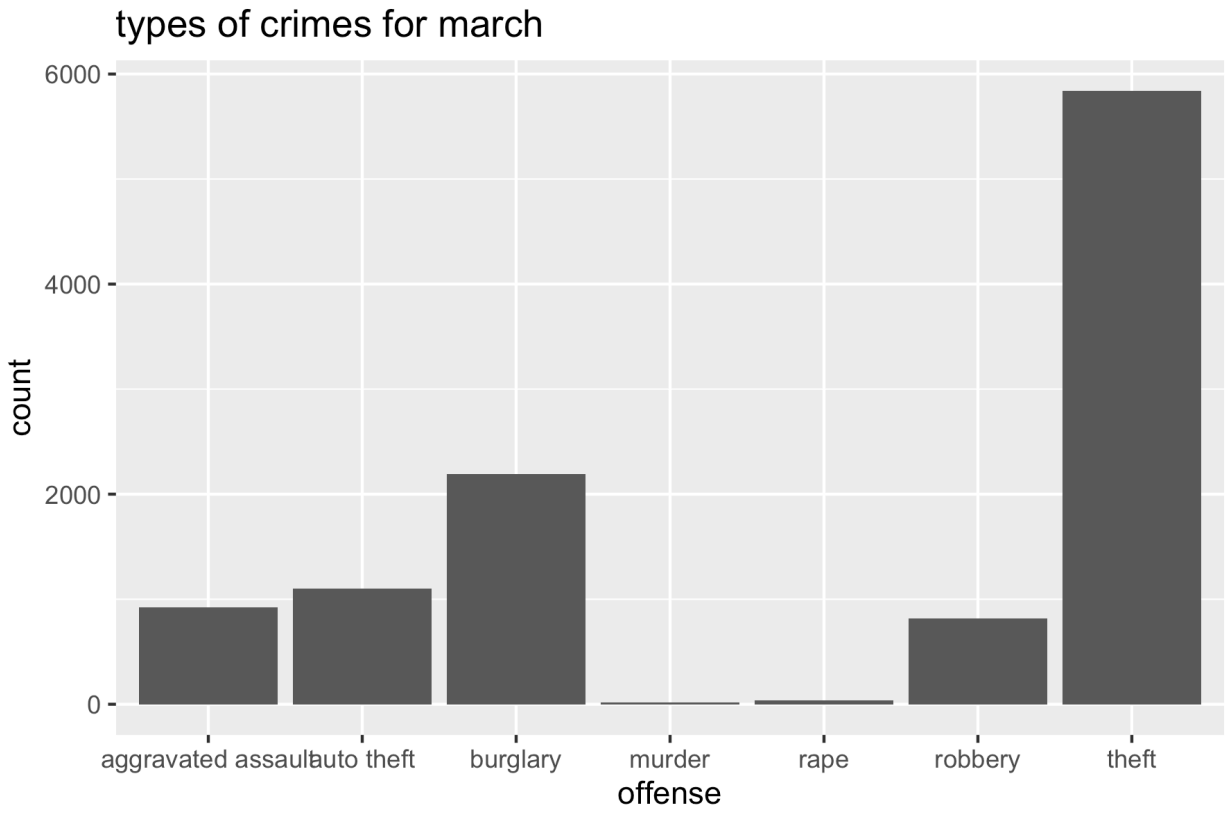

```
plot_desired_crimes_for_month <- function(desired_month) {
  crimes_for_month <- subset(crime_report, month == desired_month)
  crimes_tit <- paste("types of crimes for", desired_month, sep = " ")
  ggplot(crimes_for_month, aes(x = offense)
         ) + geom_histogram(stat = "count"
         ) + ggtitle(crimes_tit
         ) + xlab("offense")
}

plot_desired_crimes_for_month("january")
```




```
plot_desired_crimes_for_month("february")
```




```
plot_desired_crimes_for_month("march")
```



ANSWER:

What we can say about **when** people commit crimes:

Most months and days follow a similar pattern.

Theft seems to be the most common offense, with burglary coming in after

The month May seems to have the highest crime rates

Towards the afternoon into the evening, there are increased crime rates

Thursday, Friday, Saturday have generally higher crime rates than other days

I honestly couldn't have predicted any of this, and none surprise me

There is a normal relationship between types of crime and time of day

7.5.2 Q5.2 Geospatial Analysis

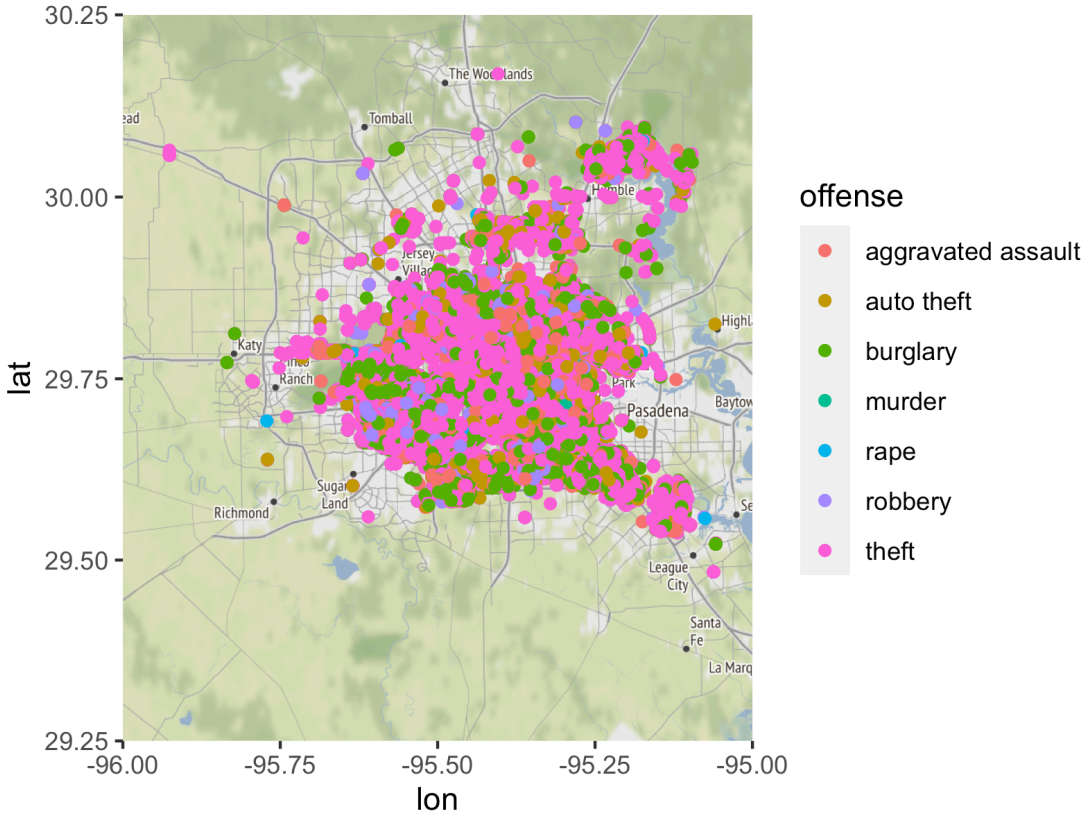
```
library(sp)
library(rgdal)
library(maptools)
```

Geospatial Analysis

Next we'll look at the spatial distribution of this data.

- Plot the data on an OpenStreetMap (**Error: 'get_openstreetmap' is defunct. Use 'OSM is at least temporarily not supported, see https://github.com/dkahle/ggmap/issues/117.' instead. See help("Defunct")**)
  - Using `source = "stamen"`
- You will have to specify the location in the function call
  - This is because `ggmap::get_map()`
  - Defaults to Google Maps
    - when a bounding box is not specified
  - A bounding box
    - Gives the boundaries of the map that is downloaded
    - Specified with a list
    - `c(left = '' , right = '' , top = '', bottom = '')`
- Color the map based on
  - Type of Crime Reported

```
#Get the bbox

bbox <- c(left = -96,
          right = -95,
          top = 30.25,
          bottom = 29.25)

#Retrieve the map

houstonmap <- get_map(location = bbox,
                      source = 'stamen')

#Plot

crime_report$offense <- as.factor(crime_report$offense)

ggmap(houstonmap) + geom_point(aes(x = lon, y = lat, color = offense), data = crime_report)
```



ANSWER: plotted

7.5.3 Q5.3a Modify the incident occurrence layer, to better see whats happening

In the last map, it was a bit tricky

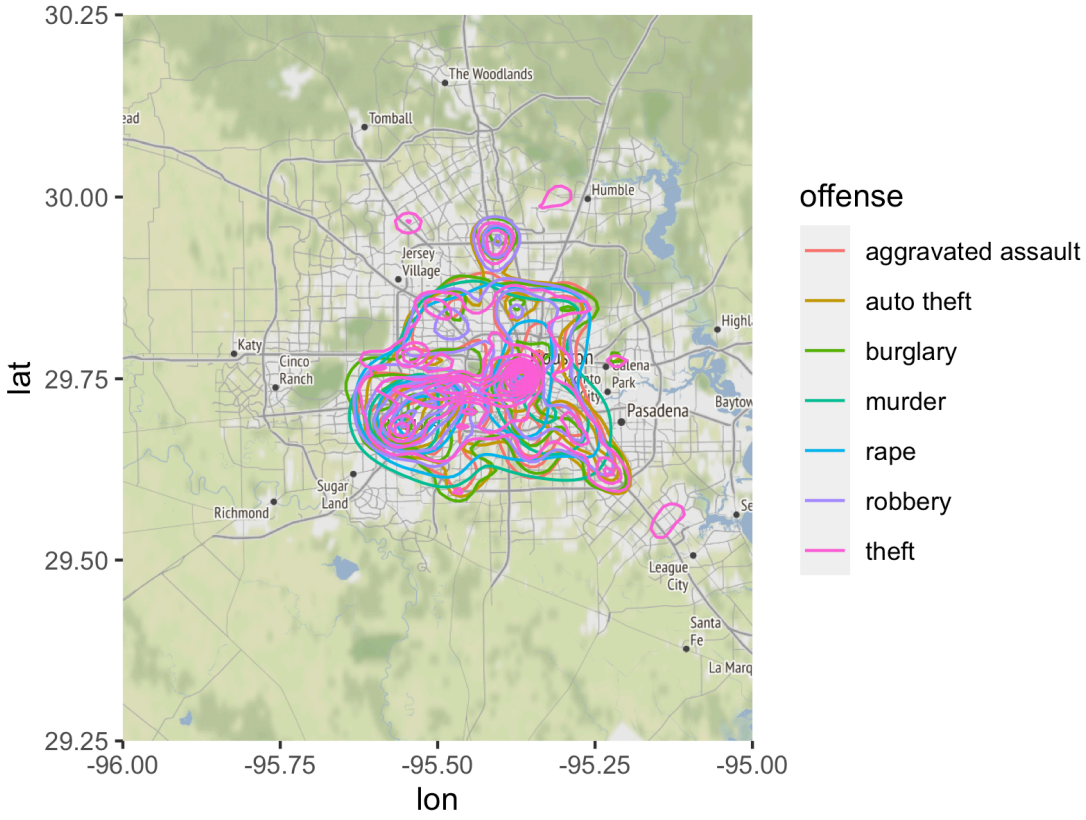
- to see the density of the incidents
  - because all the graphed points
  - were sitting on top of each other.

We're going to now modify the incident occurrence layer

- to plot the density of points
  - vs plotting each incident individually.

- We accomplish this with
  - the `ggplot2::stat_density2d()` function
  - vs using `ggplot2::geom_point()`.

```
#Plot using stat_density2d()
ggmap(houstonmap) + stat_density2d(aes(x = lon,
                                       y = lat,
                                       color = offense),
                                   data = crime_report)
```

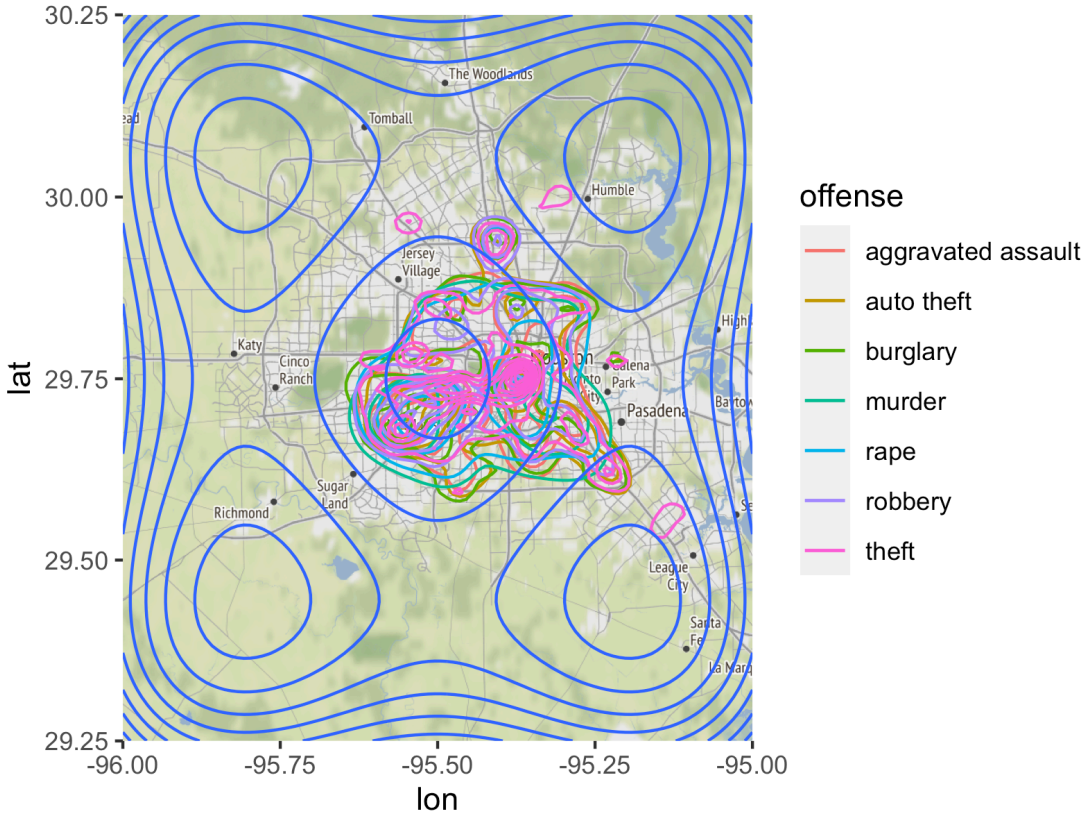


ANSWER:

7.5.4 Q5.3b What does `..level..` do

- What does `..level..` do
  - in the `ggplot2::stat_density2d()` function call?
- Hint: Look at the help topics for this function

```
ggmap(houstonmap) + stat_density2d(aes(x = lon,
                                       y = lat,
                                       color = offense,
                                       fill = ..level..),
                                   data = crime_report
                                   ) + stat_density2d(aes(
                                       fill = ..level..)
                                       )
```



ANSWER: ..level.. tells ggmap to reference the column in a newly built data frame, and sketches contour lines

7.5.5 Q5.4 What is the safest and most dangerous neighborhoods

```
#Filter
library(broom)
library(ggplot2)
```

Finally

- Filter out a specific crime of your choosing (auto-theft)
- Plot the crime density

We will use a new package that assists in geospatial analysis

- `rdgal`
- This package is used to transform and project geospatial objects
- It also has some nice functions for working with `.shp` files
  - `.shp` files contain information about regions on a map
  - i.e `.shp` files can contain the information
    - the size, shape, and location of countries or states

Add Polygons for the specific Neighborhoods

- Using NeighborShapefile
  - and `rdgal::readOGR()`
  - or `mapdata::readShapeSpatial()`

```
nbhd_path <- 'data/shp/COH_SUPER_NEIGHBORHOODS.shp'
nbhd_file <- readOGR(nbhd_path)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/momo/Documents/GitRem/22f-dsci351-451-mxd601/1-assignments/lab-exercise/LE7/data/shp
## with 88 features
## It has 14 fields
```

```
nbhds <- tidy(nbhd_file)
names(nbhd_file)
```

```
##  [1] "OBJECTID"   "PERIMETER"  "POLYID"     "SNBNAME"    "COUNCIL_AC"
##  [6] "RECOGNITIO" "SnbrInfoUR" "WeCan"      "Top10"      "CEA_FLAG"
## [11] "cohgis_COH" "cohgis_C_1" "ShapeSTAre" "ShapeSTLen"
```

```
auto_thefts <- subset(crime_report, offense == 'auto theft')
filtered_auto_thefts <- as.data.frame(as.tibble(auto_thefts) %>% select(number, lon, lat, block, premis


id_name_map <- data.frame(
  id = nbhd_file$POLYID,
  name = nbhd_file$SNBNAME
)

nbhds <- merge(nbhds, id_name_map, by.x = 'id')
```

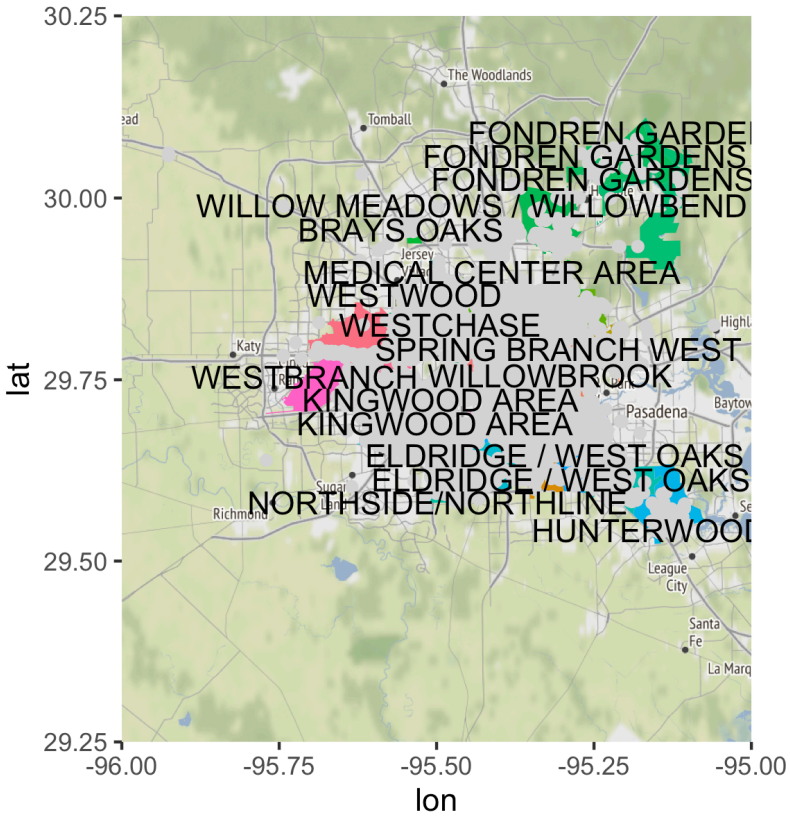
What is the most dangerous Neighborhood for your crime? Where is the safest neighborhood?

Label the map with the neighborhoods

- Hint: It's OK to remove some of the labels, there's a lot
- `ggplot2::geom_text()` has a built in function for this
  - check `overlap = TRUE`

```
ggmap(houstonmap
) + geom_polygon(data = nbhds,
                 mapping = aes(x = long,
                               y = lat,
                               group = group,
                               fill = id),
                 show.legend = FALSE
)  + geom_point(data = auto_thefts,
                mapping = aes(x = lon,
                              y = lat,
```

```
                                ),
                  color = "lightgrey",
                  show.legend = FALSE
) + geom_text(data = nbhds,
                mapping = aes(x = long,
                              y = lat,
                              label = name),
                show.legend = FALSE,
                check_overlap = TRUE,
                # vjust = "inward",
                # hjust = "inward"
)
```



ANSWER:

Looks like midtown-willowbrook, and kingwood area are some of the most dangerous neighborhoods for auto-theft, whereas fondren gardens, willowbend area, westbranch, and central southwest are some of the safest.

7.5.6 Links

https://blog.dominodatalab.com/applied-spatial-data-science-with-r/

http://www.r-project.org

http://rmarkdown.rstudio.com/

https://www.openintro.org/stat/textbook.php?stat_book=os

https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation