Applied Soccer Analytics with R

Devin Pleuler, translated by: Derrick Yam

March 12, 2019

This tutorial is a translated version of Devin Pleuler's, "Applied Soccer Analytics with Python" which was given as an introduction to football analytics course at the Rotman Sports Tech & Analytics Conference.

Using Statsbomb Open Data from World Cup 2018

Tutorial will cover:

1. Downloading Statsbomb Data
2. Basic Pass Map Visualization
3. Pass Classification using KMeans Clustering
4. Pass Sequence Prediction using LSTM
5. Evaluate Predictability of Team Ball Movement

R Dependances Include:

- StatsBombR
  - Written by StatsBomb to easily read in StatsBomb Data https://github.com/statsbomb/StatsBombR.
  - also loads numerous dependencies for easy data cleaning.
  - Installation Instructions:
    1. If devtools is not yet installed into R, run: `install.packages("devtools")`
    2. Then, install this R package as: `devtools::install_github("statsbomb/StatsBombR")`
    3. Finally, `library(StatsBombR)`
- clue
  - For cluster predictions.
- ggplot2
  - For visualizations.
- keras
  - For Neural Network Model.
  - Installation Instructions:
    1. Please see: https://tensorflow.rstudio.com/keras/reference/install_keras.html#windows-installation
- fastDummies
  - For quick representation of outcome format required by neural network.

Load Libraries.

```
library(StatsBombR)
library(clue)
library(ggplot2)
library(keras)
library(fastDummies)
```

1. Downloading StatsBomb Data.

There are two ways to read in free StatsBomb Data through StatsBombR. 1. Load All Free Data Available using: `StatsBombData <- StatsBombFreeEvents()` 2. Load Free Data By Individual Matches.

We are going to show you the second way.

```
Comp <- FreeCompetitions() # Get all competitions available for free
Comp <- Comp %>% #Filter for the world cup
  filter(competition_name == "FIFA World Cup")

Matches <- FreeMatches(Comp$competition_id) #Get all matches in the world cup

#Now loop through each match to get the events and then bind them together.
WC <- tibble()
for(i in 1:nrow(Matches[,1])){
  WC <- bind_rows(WC, get.matchFree(Matches[i,1]))
}
```

```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

```
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
## [1] "Whilst we are keen to share data and facilitate research, we also urge you to be responsible wi
```

Clean Data

Unlike spreadsheet-able data frames, more complex data files like JSON often use nested arrays where each variable is not simply a number or a character, but can be its own data frame, or array nested inside of that variable. For example, locations are an x, y pair and are stored in the data as an array of length 2. See below:

```r
WC$location[10] #select the location value for the tenth event, first few events do not have a location
dim(WC) #See the number of rows and columns
```

```
## [[1]]
## [1] 38 72
##
```

```
## [1] 177355    119
```

We wrote a function for StatsBomb Data in R that cleans all locations.

```
WC <- cleanlocations(WC)
dim(WC) #check dimensions again
```

```
## [1] 177355    126
```

Note: the number of columns has expanded. We have now added a new variable for the location.x, location.y, pass.end_location.x, pass.end_location.y, shot.end_location.x, shot.end_location.y and shot.end_location.z.

2. Basic Pass Map Visualization

Reduce Data

The data structure is very large, with a lot of variables that are not relevant for each analysis. It is good practice to reduce the dimensions of your data whenever possible.

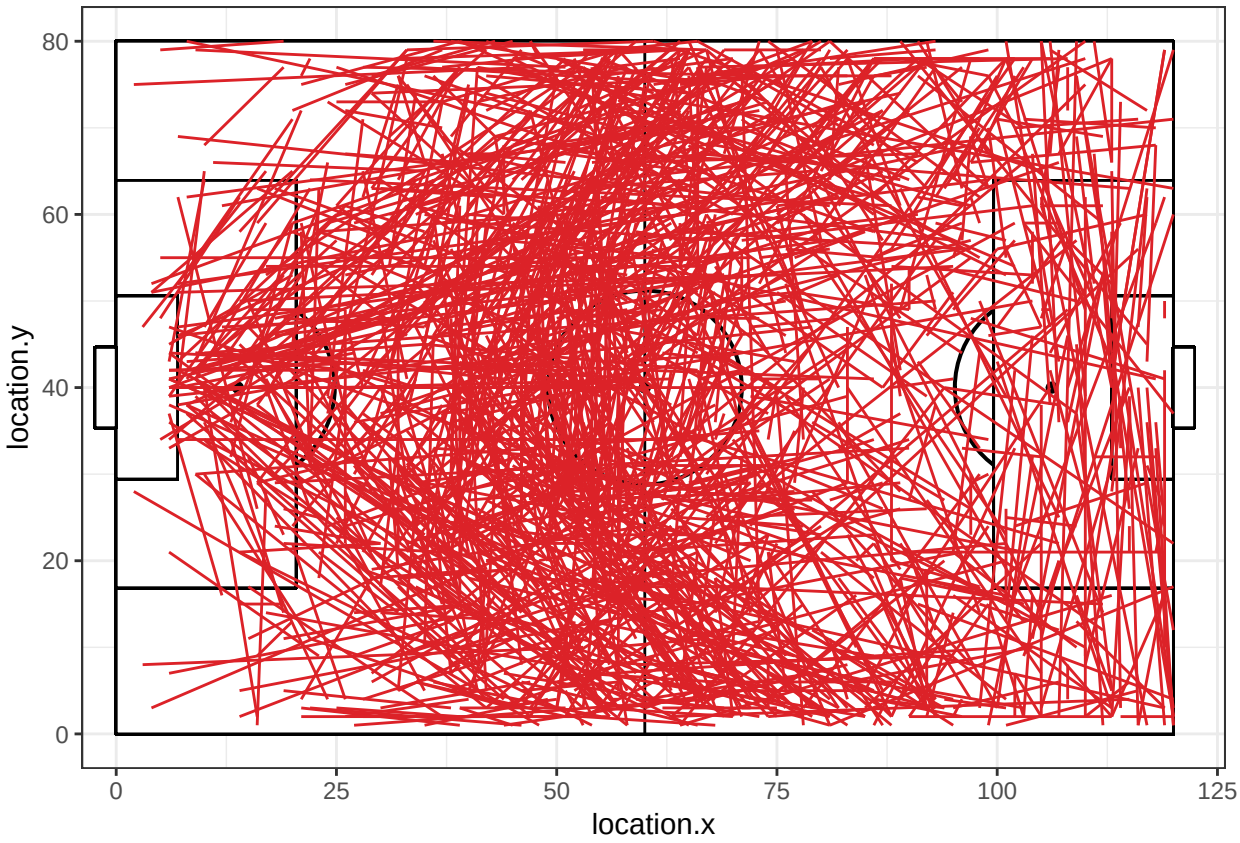
```
Passes <- WC %>%
  filter(type.name == "Pass") %>% #filter for only pass events
  select(match_id,
         team.name,
         possession,
         location.x,
         location.y,
         pass.end_location.x,
         pass.end_location.y) #Select only the variables that we want

dim(Passes)
```

```
## [1] 62871     7
```

The dimensions of the Passes, data frame is 62,871 passes with 7 variables for each pass. If we were to plot 62,871 passes it would be almost impossible to discern individual passes (if your graphics processor even allowed it to run).

Plot One Game

```
ggplot(Passes %>% filter(match_id == 7581), # Denmark vs. Croatia,
       aes(location.x, location.y, xend = pass.end_location.x, yend = pass.end_location.y)) +
  annotate_pitchSB() + #This function is adapted from the ggsoccer package.
  geom_segment(color = "#DC2228") + #Create line "segment"-s
  theme_bw()
```



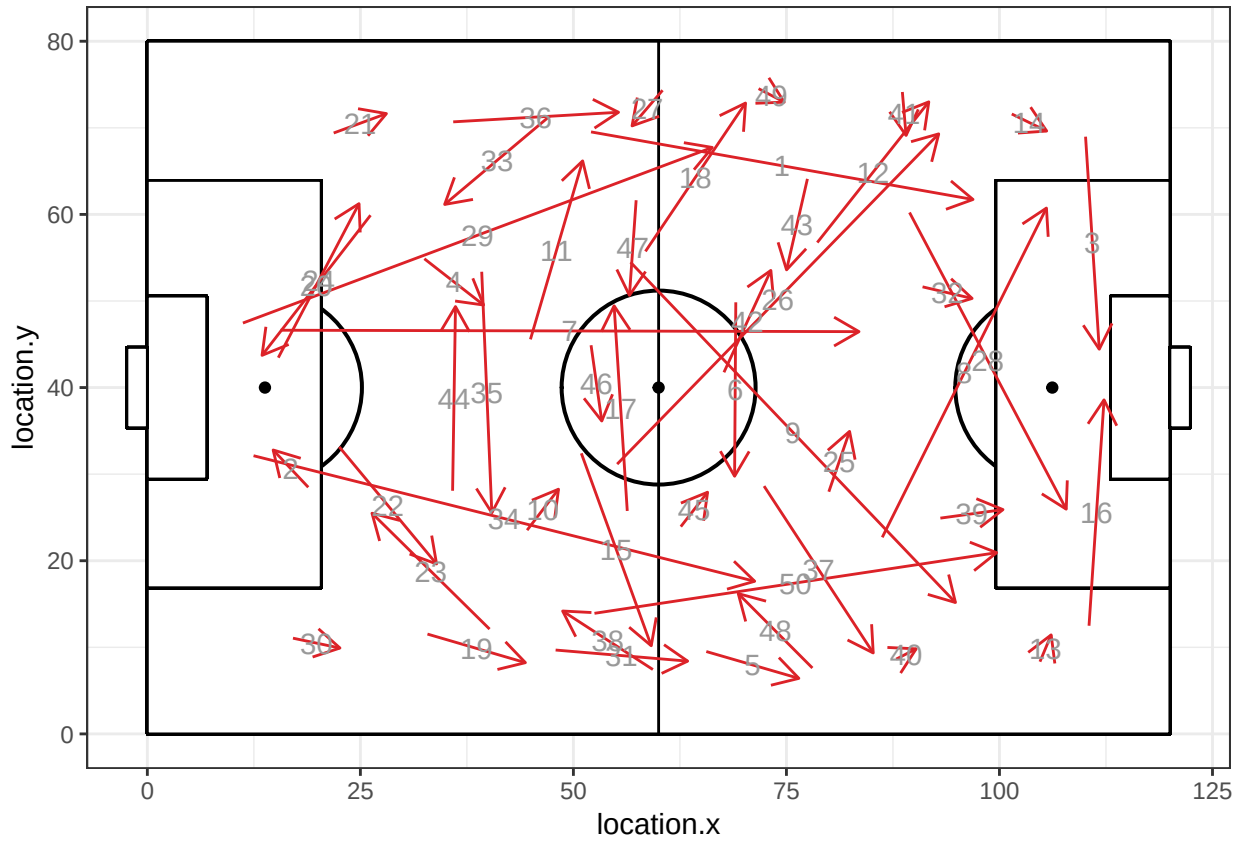
This is still a lot of passes, many of which are very similar. We may be able to analyze passing better if we clustered the passes to further reduce the dimensions.

3. Pass Classification using KMeans Clustering

Using the stats package, we are going to cluster passes into 50 clusters based on the pass x, y location and the pass end x, y location.

```
set.seed(1) #set seed for reproducibility

#The function kmeans works with matrices so select the variables we want and convert to a matrix.
training_data <- Passes %>%
  select(location.x, location.y, pass.end_location.x, pass.end_location.y)
training_data <- as.matrix(training_data)

# Perform the clustering on the training data
cluster_model <- kmeans(training_data, centers = 50, iter.max = 50)

# Get the clusters!
#The cluster number is saved in the row name of the matrix returned from "fitted"
clusters <- fitted(cluster_model)
clusters <- as_tibble(clusters) %>%
  mutate(clusternumber = rownames(fitted.values(cluster_model))) %>%
  group_by(clusternumber) %>%
  slice(1) #we only need one pass from each cluster as they all have the same center value.
```

```
ggplot(clusters, aes(location.x, location.y, xend = pass.end_location.x, yend = pass.end_location.y)) +
  annotate_pitchSB() + #This function is adapted from the ggsoccer package.
  geom_segment(color = "#DC2228", arrow = arrow(length = unit(0.15, "inches"))) + #Create line "segment
  theme_bw() +
  xlim(c(-1, 121)) +
  geom_text(data = clusters, aes(x = (location.x + pass.end_location.x)/2,
                                 y = (location.y + pass.end_location.y)/2,
                                 label = clusternumber),
            color = "#999999") #add text of the cluster number at the center of the pass.
```



4. Pass Sequence Prediction using LSTM

We are going to generate a pass sequence prediction model. The objective of this model is to predict the probability of your next pass being each of the pass clusters. LSTM models utilize a “Long-Short Term Memory” layer of neural networks which allows the model to recall information from earlier in the possession. More-standard models will only recall information up to the current state.

Clean Data For LSTM

We are going to work on the possession level generating pass sequences with a maximum length of 5.

```
#Predict cluster for all passes
Passes.reduced <- Passes %>%
```

```
  select(location.x, location.y, pass.end_location.x, pass.end_location.y)
Passes.reduced <- as.matrix(Passes.reduced) #Must be in matrix form to predict

clusterids <- cl_predict(cluster_model, Passes.reduced)
Passes <- as_tibble(Passes) %>%
  mutate(clusterids = clusterids)

#Create Pass Sequences From Previous Passes
Pass.Sequences <- Passes %>%
    group_by(match_id, possession) %>%
    mutate(cluster.1 = lag(clusterids, 1),
           cluster.2 = lag(clusterids, 2),
           cluster.3 = lag(clusterids, 3),
           cluster.4 = lag(clusterids, 4),
           Outcomes = lead(clusterids, 1))

#Pad sequences shorter than 5 passes with 0s this is common for LSTMs
Pass.Sequences <- Pass.Sequences %>%
  ungroup() %>%
  mutate_at(vars(contains("cluster")), funs(ifelse(is.na(.), 0, .)))

#Filter for only passes that stay in possession.
OriginalPasses <- Pass.Sequences %>%
  filter(!is.na(Outcomes))

#LSTM needs matrix form.
x_train = as.matrix(OriginalPasses %>% select(contains("cluster"))) #Select the variables we need
y_train = as.matrix(OriginalPasses$Outcomes)

#LSTM needs "one-hot" representation which is essentially a binary 0 or 1 for the pass cluster
#with a column for each cluster.
#There is a r package called fast Dummies which is very helpful.
y_train <- dummy_cols(y_train)
y_train <- y_train[,-1] #Remove the original outcomes matrix
colnames(y_train) <- gsub("V1_", "", names(y_train)) #Re order the outcomes from 1 to 50
col.order <- as.character(sort(as.integer(colnames(y_train))))
y_train <- y_train[ , col.order]
y_train <- as.matrix(y_train)

#Data must be in an array form for neural networks.
x_train <- array(x_train, dim = c(nrow(x_train), ncol(x_train), 1))
y_train <- as.array(y_train)
```

Quick View of what the data should look like before training.

This is something that really helps me out when working through other people's code and model's. More researchers adamant on sharing information and reproducibility should always include real samples of their data structure before training.

```
dim(x_train) #dimensions.
x_train[1:6, ,] #first six rows, all columns and all slices
dim(y_train)
y_train[1:6, ] #first six rows and all columns
```

```
## [1] 52028     5     1
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   46    0    0    0    0
## [2,]    9   46    0    0    0
## [3,]   36    9   46    0    0
## [4,]   40   36    9   46    0
## [5,]   40   40   36    9   46
## [6,]   13   40   40   36    9
## [1] 52028    50
##      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## [1,] 0 0 0 0 0 0 0 0 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [2,] 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [3,] 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [4,] 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [5,] 0 0 0 0 0 0 0 0 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## [6,] 0 0 0 0 0 0 0 0 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
##      27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## [1,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [2,]  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## [3,]  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
## [4,]  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
## [5,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [6,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##      50
## [1,]  0
## [2,]  0
## [3,]  0
## [4,]  0
## [5,]  0
## [6,]  0
```

Construct The Model

```
# Build Sequential Neural Network
model <- keras_model_sequential()

#Build layers
model %>%
  layer_lstm(units = 200, input_shape = c(5,1)) %>%
  layer_dense(units = 50, activation = 'softmax') #Final layer is 50, the number of pass clusters.

#Compile using loss of Categorical Crossentropy
model %>% compile(loss = "categorical_crossentropy", optimizer = "adam")

model ##See the model structure
```

```
## Model
## _________________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =========================================================================
## lstm_1 (LSTM)                (None, 200)               161600
## _________________________________________________________________________
```

```
## dense_1 (Dense)                 (None, 50)                    10050
## ================================================================================
## Total params: 171,650
## Trainable params: 171,650
## Non-trainable params: 0
## _________________________________________________________________________
```

Train the model.

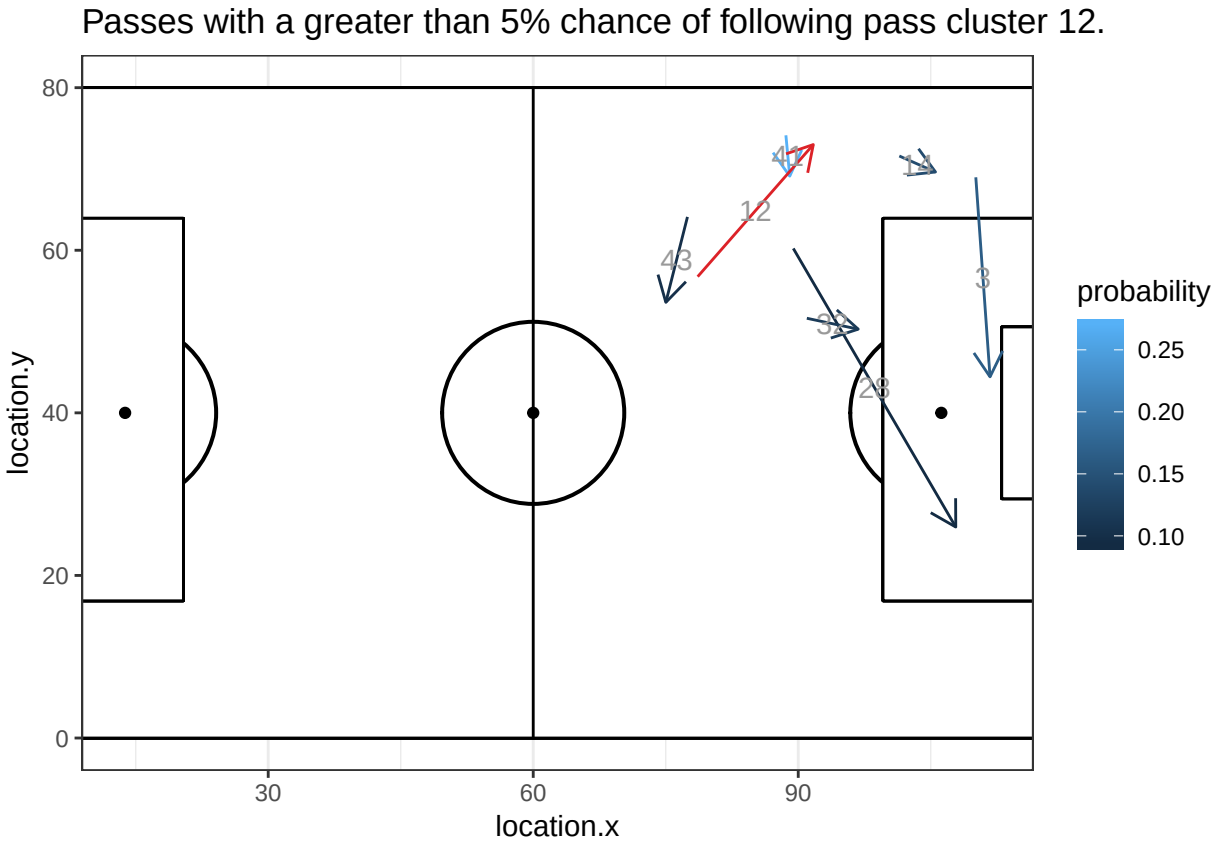
The neural network takes approximately 10 minutes to train on a standard PC laptop.

```
strt <- Sys.time() #record starting time before training

model %>% fit(x_train, y_train, epochs=50, verbose=2,
              batch_size = 50, batch_input_shape = c(50, 5))

Sys.time() - strt #Calculate time to train.
```

```
## Time difference of 9.602246 mins
```

Create a Sequence and Predict The Next Passes

```
sequence <- array(c(12,0,0,0,0), dim = c(1, 5, 1)) #Again data must be in array form.

probability <- predict_on_batch(model, sequence)

prediction.data <- tibble(clusternumber = as.character(seq(1:50)),
                          probability = as.numeric(probability)) %>%
  left_join(clusters) #Join in the locations from the first plot.

ggplot(prediction.data %>% filter(probability > 0.05), aes(location.x, location.y,
                           xend = pass.end_location.x, yend = pass.end_location.y)) +
  #This function is adapted from the ggsoccer package.
  annotate_pitchSB() +
  #Create line "segment"-s
  geom_segment(aes(color = probability), arrow = arrow(length = unit(0.15, "inches"))) +
  geom_segment(data = prediction.data %>% filter(clusternumber == "12"),
               aes(location.x, location.y,
                   xend = pass.end_location.x, yend = pass.end_location.y),
               arrow = arrow(length = unit(0.15, "inches")), color = "#DC2228") +
  theme_bw() +
  #add text of the cluster number at the center of the pass.
  geom_text(data = prediction.data %>% filter(probability > 0.05 | clusternumber == "12"),
            aes(x = (location.x + pass.end_location.x)/2, y = (location.y + pass.end_location.y)/2, labe
            color = "#999999")  +
  labs(title = "Passes with a greater than 5% chance of following pass cluster 12.")
```



5. Evaluate Predictability of Team Ball Movement

Now that we have sense checked a reasonable pass sequence model, we can extend this work to rank teams based on their "predictability". With every pass sequence, we predict the probability of each pass cluster being the next pass in the sequence. We then calculate the proportion of times that each team chooses the most probable pass. The higher proportion of times the model predicts the actual next pass the more predictable a team is.

```
allpredictions <- predict_on_batch(model, x_train)
mostlikely <- apply(allpredictions, 1, which.max) #which.max returns the index of the greatest value. a
OriginalPasses$Predicted <- mostlikely #Bind the predictions to the original data frame.

#OriginalPasses includes the team names
TeamSummary <- OriginalPasses %>%
  group_by(team.name) %>%
  summarise(Passes = n(),
            PredictedPasses = sum(Outcomes == Predicted),
            Predictability = mean(Outcomes == Predicted)) %>%
  arrange(Predictability)


TeamSummary %>% print(n = Inf)
```

```
## # A tibble: 32 x 4
##    team.name    Passes PredictedPasses Predictability
##    <chr>         <int>           <int>          <dbl>
```

```
##  1 Iceland         678             159          0.235
##  2 Iran            541             128          0.237
##  3 Serbia          975             233          0.239
##  4 Mexico         1426             351          0.246
##  5 Uruguay        1883             477          0.253
##  6 South Korea     780             199          0.255
##  7 Denmark        1476             383          0.259
##  8 Costa Rica      895             233          0.260
##  9 Senegal         869             229          0.264
## 10 Sweden         1277             340          0.266
## 11 Japan          1835             492          0.268
## 12 Panama          850             228          0.268
## 13 Switzerland    1895             509          0.269
## 14 Poland         1269             342          0.270
## 15 England        3448             933          0.271
## 16 Russia         1510             409          0.271
## 17 Croatia        3356             925          0.276
## 18 Colombia       1646             454          0.276
## 19 Nigeria        1026             283          0.276
## 20 Portugal       1662             464          0.279
## 21 Tunisia        1224             342          0.279
## 22 Argentina      2108             590          0.280
## 23 Belgium        3204             907          0.283
## 24 France         2681             778          0.290
## 25 Egypt          1088             317          0.291
## 26 Morocco        1008             296          0.294
## 27 Peru           1147             338          0.295
## 28 Germany        1788             527          0.295
## 29 Saudi Arabia   1572             465          0.296
## 30 Australia      1295             392          0.303
## 31 Brazil         2457             746          0.304
## 32 Spain          3159            1054          0.334
```