NBA Shots Analysis

#Step 1: Logistic Model for Expected Value of a shot being made

```
#loading required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(RCurl)
```

```
##
## Attaching package: 'RCurl'
##
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
library(ggplot2)
```

```
#the data
library(RCurl)
link <- getURL("https://raw.githubusercontent.com/M-ttM/Basketball/master/class.csv")
basket <- read.csv(text = link)
head(basket)
```

```
##   X period time team           player points type  x  y timenum
## 1 1      3 6:03  BOS Kendrick Perkins      0  3pt -5 44     363
## 2 2      3 4:44  CHI     Kirk Hinrich      0  3pt -3 43     284
## 3 3      4 4:40  NYK Danilo Gallinari      0  3pt  0  3     280
## 4 4      1 7:09  MIL     Michael Redd      0  3pt  0  3     429
## 5 5      4 0:13  CHA    Ronald Murray      3  3pt  0  3      13
## 6 6      4 2:13  LAL    Jordan Farmar      0  3pt  0  4     133
```

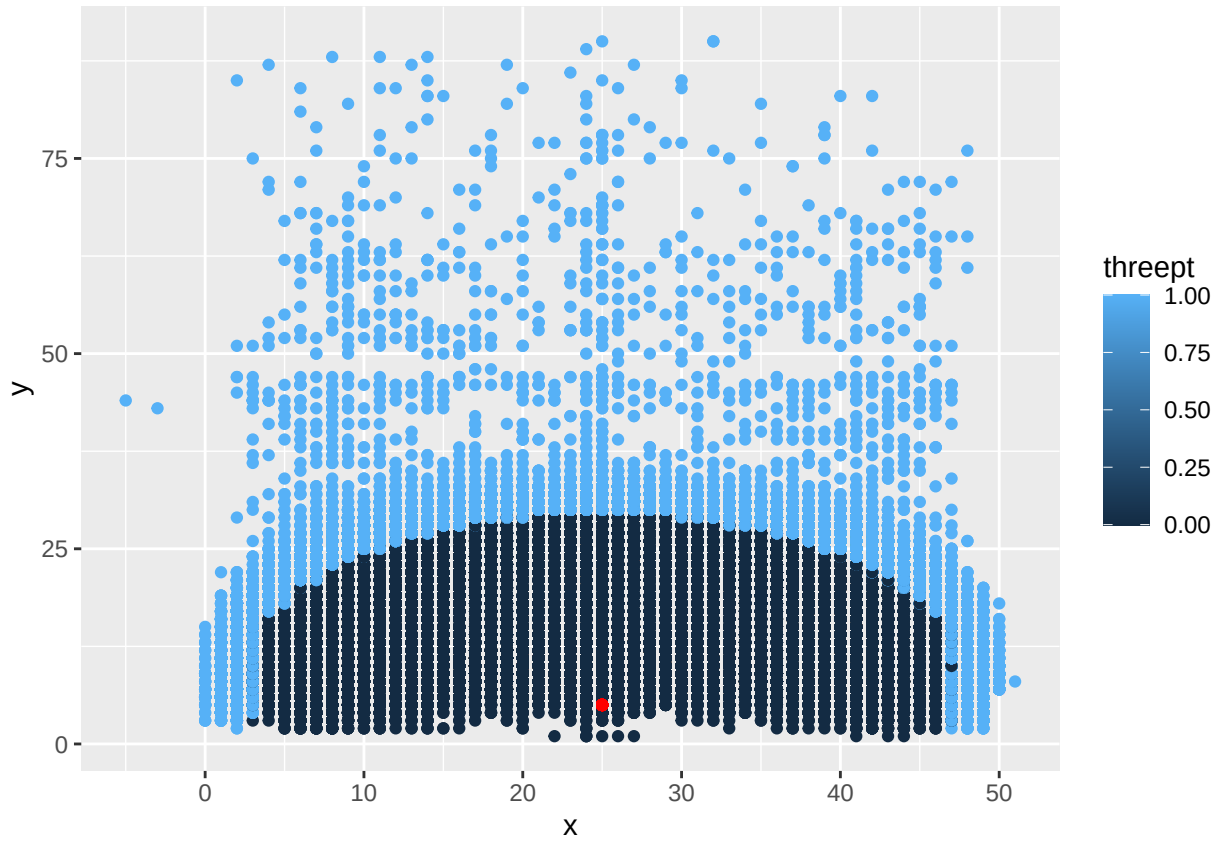
```
dim(basket)
```

```
## [1] 197015     10
```

```
#creating variables

#shot type (2 or 3 pointer)
basket <- basket %>%
  mutate(shot=as.integer(ifelse(str_detect(type, '3pt'), 3, 2)))
```

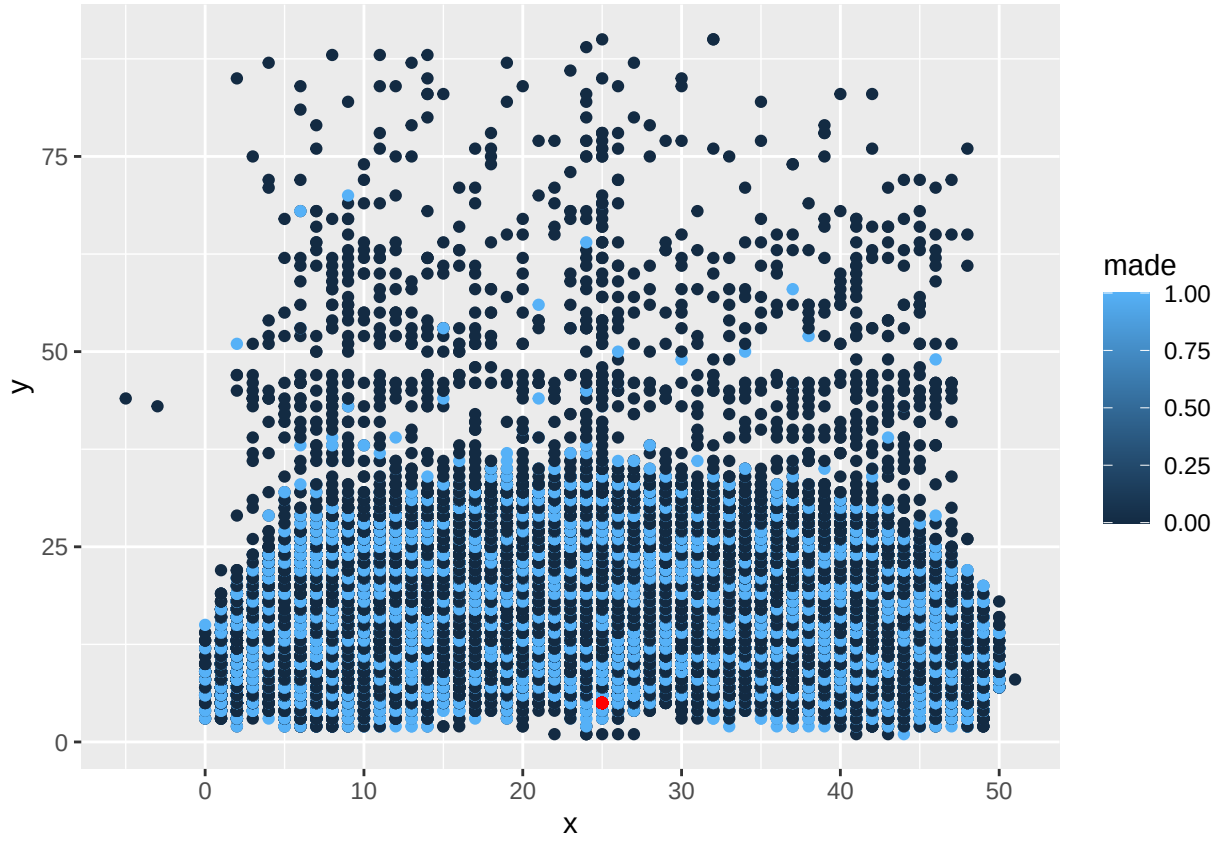
```r
#dummy variable for threepoint shot
basket <- basket %>%
  mutate(threept=as.integer(ifelse(str_detect(type, '3pt'), 1, 0)))

#minutes remaining (as an integer)
basket <- basket %>%
  mutate(timemin = as.integer(timenum/60))

#horizontal distance from basket
basket <- basket %>%
  mutate(x_1 = as.integer(abs(25 - x)))

#vertical distance from basket
basket <- basket %>%
  mutate(y_1 = as.integer(abs(5 - y)))

#dummy variable indicating overtime
basket <- basket %>%
  mutate(overtime=as.integer(ifelse(period==5, 1, 0)))

#dummy varaiable indicating whether a basket was made
basket <- basket %>%
  mutate(made=as.integer(ifelse(points==0, 0, 1)))
```

```r
#mapping coordinates of the basketball court
ggplot(data = basket, aes(x = x, y = y, colour=threept)) +
geom_point() + geom_point(x = 25, y = 5, colour='red')
```



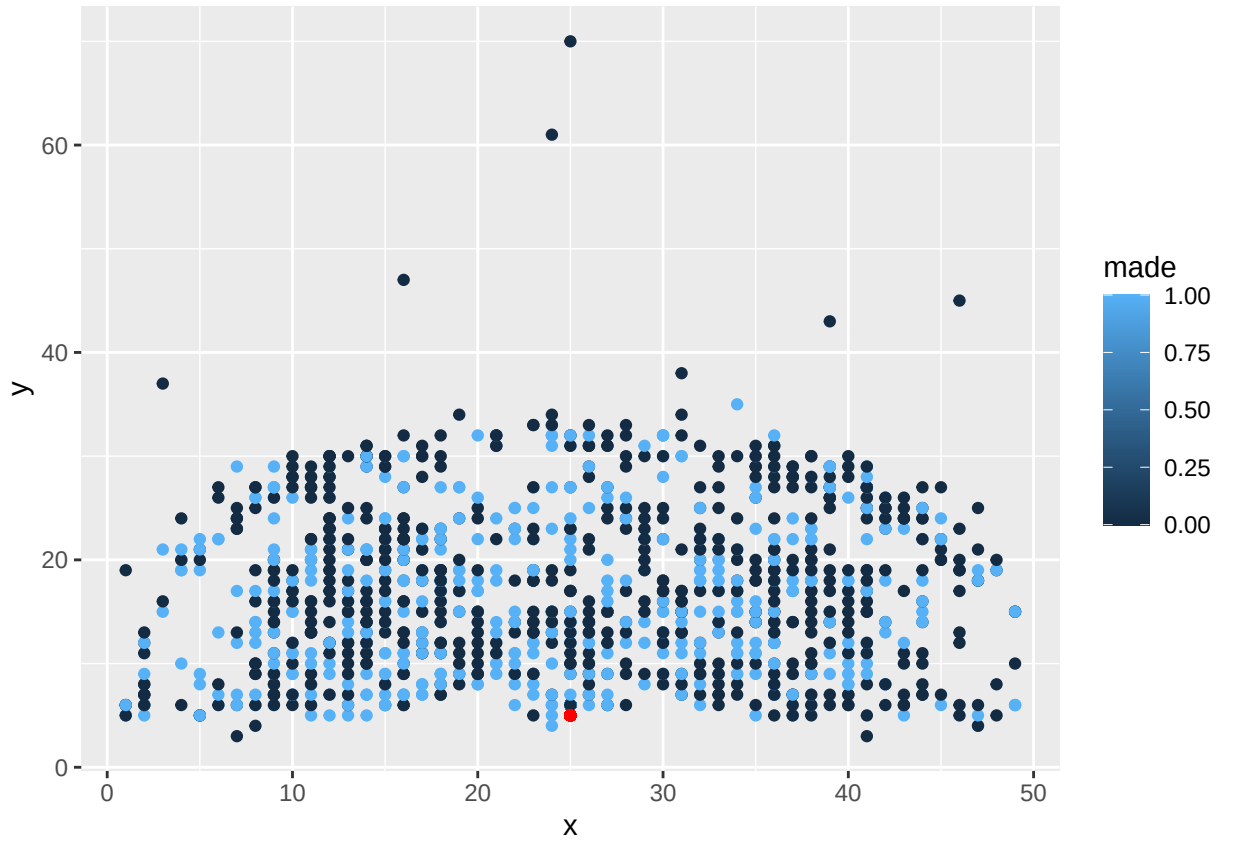
The output above shows the basketball court on a x-y plane, with the basket being at (25,5), marked as red. The black dots are shots that were two pointers, and blue are three pointers. The visualisation helps in understanding the mapping of the court in this data.

```
#mapping whether a shot was made
ggplot(data = basket, aes(x = x, y = y, colour=made)) +
geom_point() + geom_point(x = 25, y = 5, colour='red')
```



This visualization shows the points on the court-map of shots that were made (blue) or not (black).

```
#shot map of the goat
kobe <- basket %>%
  filter(player=='Kobe Bryant')

ggplot(data = kobe, aes(x = x, y = y, colour=made)) +
geom_point() + geom_point(x = 25, y = 5, colour='red')
```




```
half_court <- filter(basket, y>45)
filter(basket, (y_1>65 & made==1))
```

```
##  [1] X        period   time     team     player   points   type     x
##  [9] y        timenum  shot     threept  timemin  x_1      y_1      overtime
## [17] made
## <0 rows> (or 0-length row.names)
```

```
filter(basket, (x_1>25 & made==1))
```

```
##  [1] X        period   time     team     player   points   type     x
##  [9] y        timenum  shot     threept  timemin  x_1      y_1      overtime
## [17] made
## <0 rows> (or 0-length row.names)
```

From the code above, we can see that out of close to 200,000 observations, no shot was made 65 units away vertically from the basket; and 25 units away horizontally. These shots can be considered with large amount of data as shots that have extremely low probability of being made. Its not observed in the data, and hence, the model as well will learn accordingly.

```
filter(half_court, timenum>2)
```

```
##        X period time team          player points type  x  y timenum shot
## 1  13576      2 2:30  ATL  Jamal Crawford      0  3pt  7 52     150    3
## 2  29856      1 7:13  IND   Mike Dunleavy      0  3pt 12 60     433    3
## 3  32750      3 6:07  ORL   Jameer Nelson      0  3pt 13 46     367    3
## 4  35830      1 1:16  POR     Brandon Roy      0  3pt 14 52      76    3
## 5 128792      3 9:37  LAL      Ron Artest      0  3pt 25 51     577    3
```

```
## 6 128798      3 6:41  MIA Michael Beasley      0  3pt 25 54     401    3
## 7 171785      2 6:08  PHI    Jrue Holiday      0  3pt 39 55     368    3
## 8 189558      4 4:06  CLE     Mo Williams      3  3pt 46 49     246    3
## 9 195064      4 4:54  BOS     Rajon Rondo      0  3pt 48 65     294    3
##   threept timemin x_1 y_1 overtime made
## 1       1       2  18  47        0    0
## 2       1       7  13  55        0    0
## 3       1       6  12  41        0    0
## 4       1       1  11  47        0    0
## 5       1       9   0  46        0    0
## 6       1       6   0  49        0    0
## 7       1       6  14  50        0    0
## 8       1       4  21  44        0    1
## 9       1       4  23  60        0    0
```

```
made_half <- half_court %>%
  filter(made==1)

print(t(made_half['time']))
```

```
##      [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]   [,10]
## time "0:00" "0:00" "0:00" "0:01" "0:00" "0:00" "0:00" "0:00" "0:00" "0:02"
##      [,11]  [,12]  [,13]  [,14]  [,15]  [,16]  [,17]
## time "0:00" "0:00" "0:00" "0:00" "0:00" "0:00" "4:06"
```

The first output shows that only 9 shots (0.004% of shots) were taken behind the half line, when it wasn't the last couple seconds of the quarter. The second output shows that, only 17 shots were made behind the half line and one unique observation made the shot when it was not the last seconds of the quarter.

This implies that shots behind the half line are taken simply because there is no additional time left to travel further and is extremely unlikely to be made.

For the purposes of our model, we will assess shots that were taken inside the half of the opponent.

```
basket_new <- basket %>%
  filter(y<=45 & x_1<=25)
```

```
dunk_dt <- basket_new %>%
  filter(str_detect(type, 'dunk'))

summary(dunk_dt['made'])
```

```
##       made
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.9129
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
dim(dunk_dt)
```

```
## [1] 9321   17
```

It also seems like dunks are usually made, with 91.29% of dunks attempted being made.

```
basket_new <- basket_new %>%
  mutate(dunk=ifelse(str_detect(type, 'dunk'), 1, 0))
```

```
a <- basket_new %>%
  select(X, period, x, y, timenum, timemin, x_1, y_1, threept, overtime, dunk, made)

summary(a)
```

```
##        X              period            x               y
##  Min.   :     3   Min.   :1.000   Min.   : 0.00   Min.   : 1.00
##  1st Qu.: 49332   1st Qu.:1.000   1st Qu.:19.00   1st Qu.: 6.00
##  Median : 98512   Median :2.000   Median :25.00   Median :10.00
##  Mean   : 98515   Mean   :2.472   Mean   :25.08   Mean   :13.42
##  3rd Qu.:147728   3rd Qu.:3.000   3rd Qu.:31.00   3rd Qu.:20.00
##  Max.   :197014   Max.   :5.000   Max.   :50.00   Max.   :45.00
##     timenum         timemin            x_1             y_1
##  Min.   :  0.0   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:169.0   1st Qu.: 2.000   1st Qu.: 0.000   1st Qu.: 1.000
##  Median :351.0   Median : 5.000   Median : 6.000   Median : 5.000
##  Mean   :349.3   Mean   : 5.342   Mean   : 7.828   Mean   : 8.449
##  3rd Qu.:530.0   3rd Qu.: 8.000   3rd Qu.:14.000   3rd Qu.:15.000
##  Max.   :720.0   Max.   :12.000   Max.   :25.000   Max.   :40.000
##     threept          overtime            dunk              made
##  Min.   :0.0000   Min.   :0.000000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.000000   Median :0.00000   Median :0.0000
##  Mean   :0.2202   Mean   :0.005819   Mean   :0.04745   Mean   :0.4625
##  3rd Qu.:0.0000   3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.000000   Max.   :1.00000   Max.   :1.0000
```

The distributions of the variables in the data are shown above. It appears that slightly more shots are taken towards the second half of the game. 25% of shots are taken very close to the basket; 50% are taken with the coordinate (6,5); and 75% of shots are taken within (14, 15). It is tough to determine which coordinates signal to three pointers, but since 22% of shots are three pointers, the (14, 15) coordinate could signal to a potential three point line estimate.

```
b <- a %>%
  arrange(made) %>%
  group_by(made) %>%
  summarise(mean_period=mean(period), mean_timemin=mean(timemin),
            mean_x_1=mean(x_1), mean_y_1=mean(y_1),
            mean_three=mean(threept), mean_ot=mean(overtime),
            mean_dunk=mean(dunk))
```

```
b
```

```
## # A tibble: 2 x 8
##    made mean_period mean_timemin mean_x_1 mean_y_1 mean_three mean_ot mean_dunk
##   <int>       <dbl>        <dbl>    <dbl>    <dbl>      <dbl>   <dbl>     <dbl>
## 1     0        2.50         5.29     8.91     9.65      0.263 0.00641   0.00769
## 2     1        2.44         5.40     6.57     7.06      0.171 0.00513   0.0937
```

The output above shows the average of the variables. The average shot missed was shot further by at least 2 units on average than an average shot that was made. A three pointer is a difficult shot with 17% of shots made being three pointers and 26% of shots that were not made were three pointers. 10% of shots made were dunks and only 0.7% of shots that were not made were dunks. This implies that dunk is an easier shot.

The variables we select are: - Quarter given by the variable *period*. - The Minute of the game in the quarter. The variable show how many minutes are left in the quarter, given as an integer by *timemin*. - The horizontal

distance from the basket, given by the transformed *X_1* coordinate. - The vertical distance from the basket, measured by the transformed *y_1* coordinate. - Whether a shot taken was a three-pointer, given by the variable *threept*. - Whether a shot taken was a dunk, given by the *dunk* variable - A dummy variable to indicate whether a shot was taken in *overtime* (Quarter = 5).

The logistic regression model is below:

$$\log(\frac{P(make=1)}{1-P(make=1)}) = \beta_0+\beta_1*period+\beta_2*timemin+\beta_3*x_1+\beta_4*y_1+\beta5*threept+\beta_6*dunk+\beta_7*overtime+e$$

#fitting the model using logistic regression

```
fit.1 <- glm(made ~ period + timemin + x_1 + y_1 + threept + dunk + overtime, data = basket_new, family
summary(fit.1)
```

```
##
## Call:
## glm(formula = made ~ period + timemin + x_1 + y_1 + threept +
##     dunk + overtime, family = "binomial", data = basket_new)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2505  -1.0530  -0.9007   1.1873   1.7361
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.2084389  0.0149660  13.928  < 2e-16 ***
## period      -0.0318975  0.0041963  -7.601 2.93e-14 ***
## timemin      0.0083915  0.0013576   6.181 6.37e-10 ***
## x_1         -0.0251683  0.0007271 -34.616  < 2e-16 ***
## y_1         -0.0268525  0.0006664 -40.297  < 2e-16 ***
## threept      0.0975035  0.0153732   6.342 2.26e-10 ***
## dunk         2.2076191  0.0375196  58.839  < 2e-16 ***
## overtime    -0.0331737  0.0629564  -0.527    0.598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 271203  on 196429  degrees of freedom
## Residual deviance: 257901  on 196422  degrees of freedom
## AIC: 257917
##
## Number of Fisher Scoring iterations: 4
```

All variables except overtime are statistically significant. This could be due to lack of data for overtime and the fact that overtime lasts for only 5 minutes.

The following model is produced:

$$\log(\frac{P(made=1)}{1-P(made=1)}) = 0.2-0.032*period+0.008*timemin-0.025*x_1-0.026*y_1+0.09*threept+2.21*dunk-0.033*overtime$$

From the coefficients, we can understand the effect of a variable of the log-odds of a basket. Positive coefficients indicate a positive relationship with the log-odds of a basket being made (success), and vica versa.

```
exp(fit.1$coeff)
```

```
## (Intercept)      period     timemin         x_1         y_1     threept
##   1.2317536   0.9686059   1.0084268   0.9751458   0.9735048   1.1024152
##        dunk    overtime
##   9.0940387   0.9673705
```

```
exp(confint(fit.1))
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) 1.1961516 1.2684243
## period      0.9606718 0.9766048
## timemin     1.0057471 1.0111138
## x_1         0.9737569 0.9765361
## y_1         0.9722339 0.9747767
## threept     1.0696938 1.1361375
## dunk        8.4548143 9.7949903
## overtime    0.8546643 1.0939587
```

Since the model is expressed in log-odds, we exponentiate the coefficients to better understand the model.

$$\frac{P(success = 1)}{1 - P(success = 1)} = 1.23 + 0.97 * period + 1.008 * timemin + 0.975 * x_1 + 0.973 * y_1 + 1.1 * threept + 9.09 * dunk + 0.967 * overtime + e$$

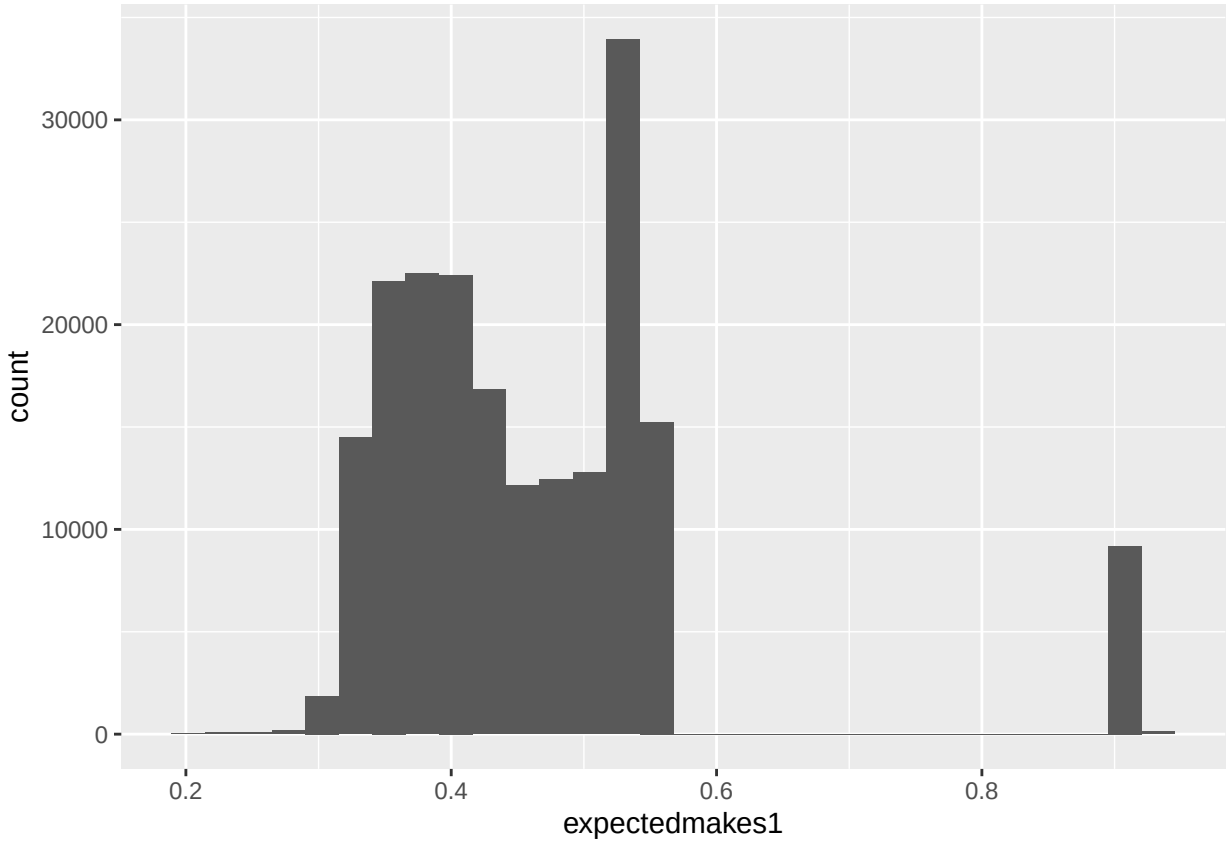
When expressed in exponent terms, coefficients greater than 1 increase the odds of a basket being made and inversely a coefficient below 1 reduces the odds of a basket being made. From the coefficients of coordinates, it can be interpreted that a shot taken a unit further vertically or horizontally affects the odds of making a basket negatively (by approximately 3% on average). Fatigue plays a role in shot being made. This is indicated by the time variables. A shot has 3% lower odds of being made as the quarters go on; and the impact of minutes remaining in the quarter is weak, but significant, on odds, with a lower odds as the minutes tick down.

Odds are negatively impacted in overtime as predicted by the model. This relationship is not statistically significant. These finding indicate that fatigue reduces the odds of a shot being made.

Interestingly, three point shots have 10% higher odds of being made. This relationship is statistically significant. The prediction by the model could be due to three point shots being practiced more, and are usually taken by 'three-point specialists' in the team. The model predicts that a dunk increases the odds 9 times, which indicates that dunk shots will have a higher predicted value of being made.

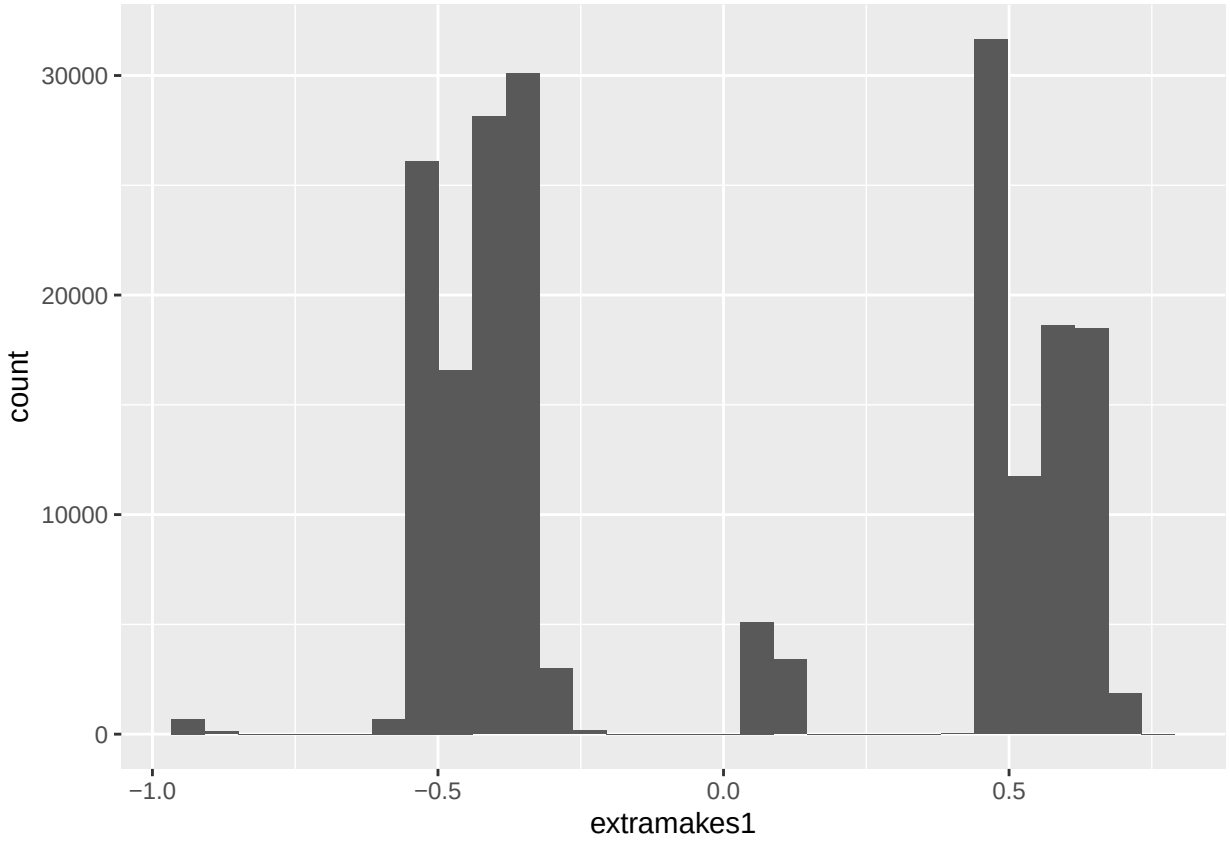
```
basket_new <- basket_new %>%
  mutate(expectedmakes1 = fitted(fit.1))
basket_new <- basket_new %>%
  mutate(extramakes1 = made-expectedmakes1)

ggplot(basket_new, aes(expectedmakes1)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

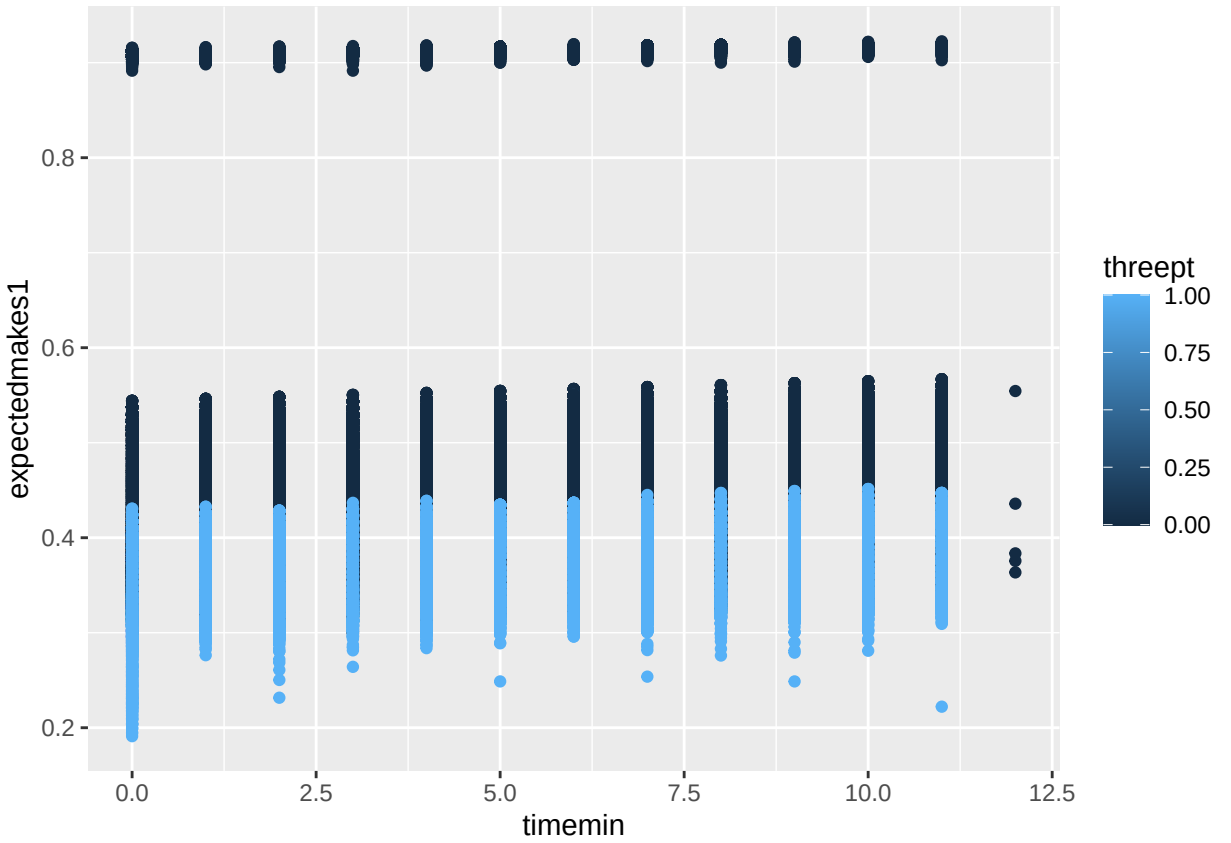



```
ggplot(basket_new, aes(extramakes1)) +
  geom_histogram()
```

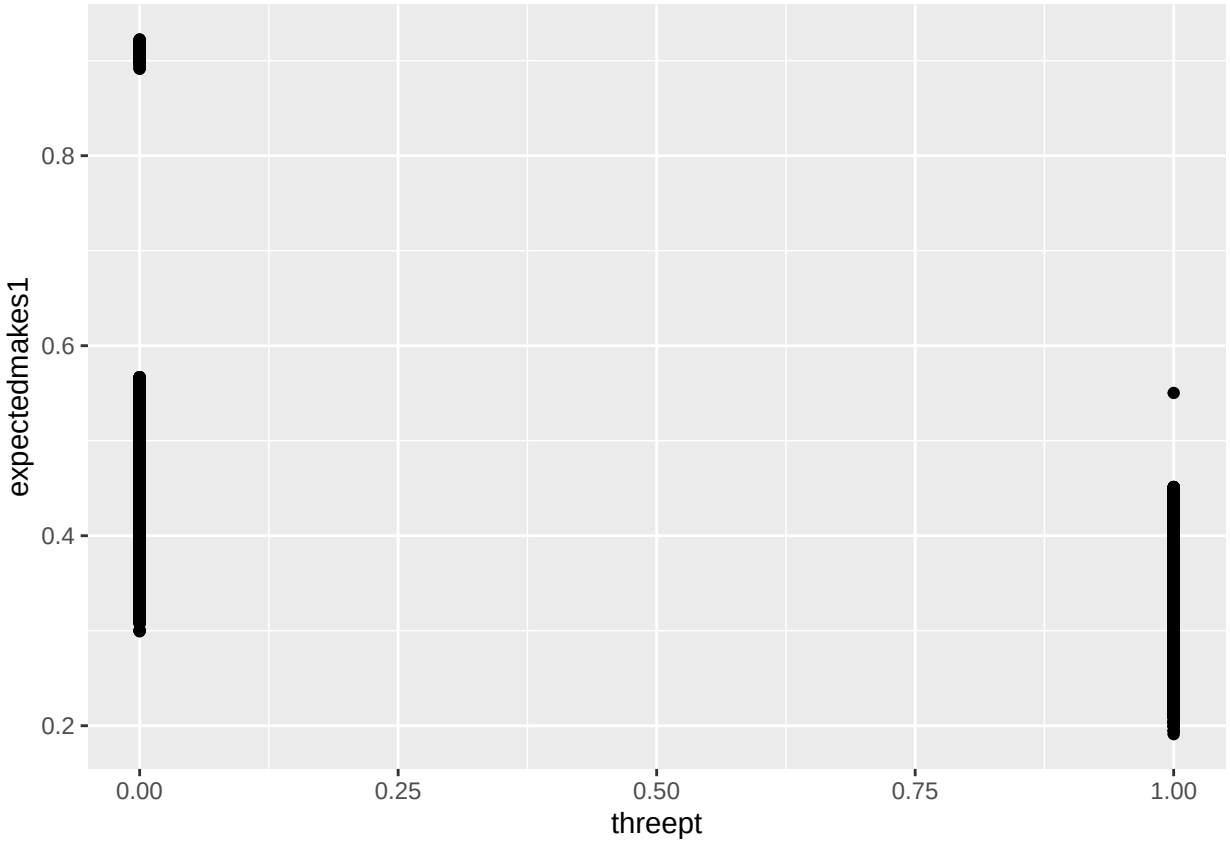
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



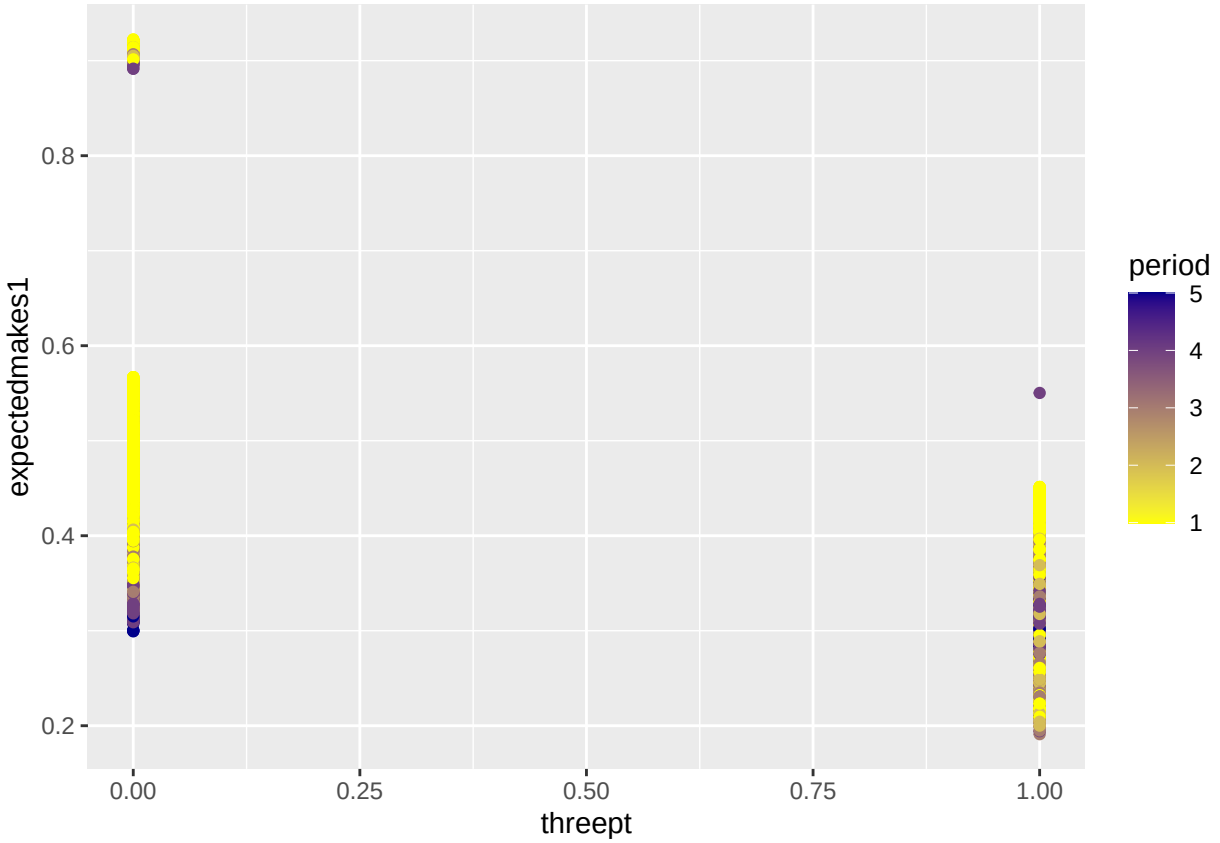

```
ggplot(basket_new,aes(timemin, expectedmakes1, colour=threept)) + geom_point()
```



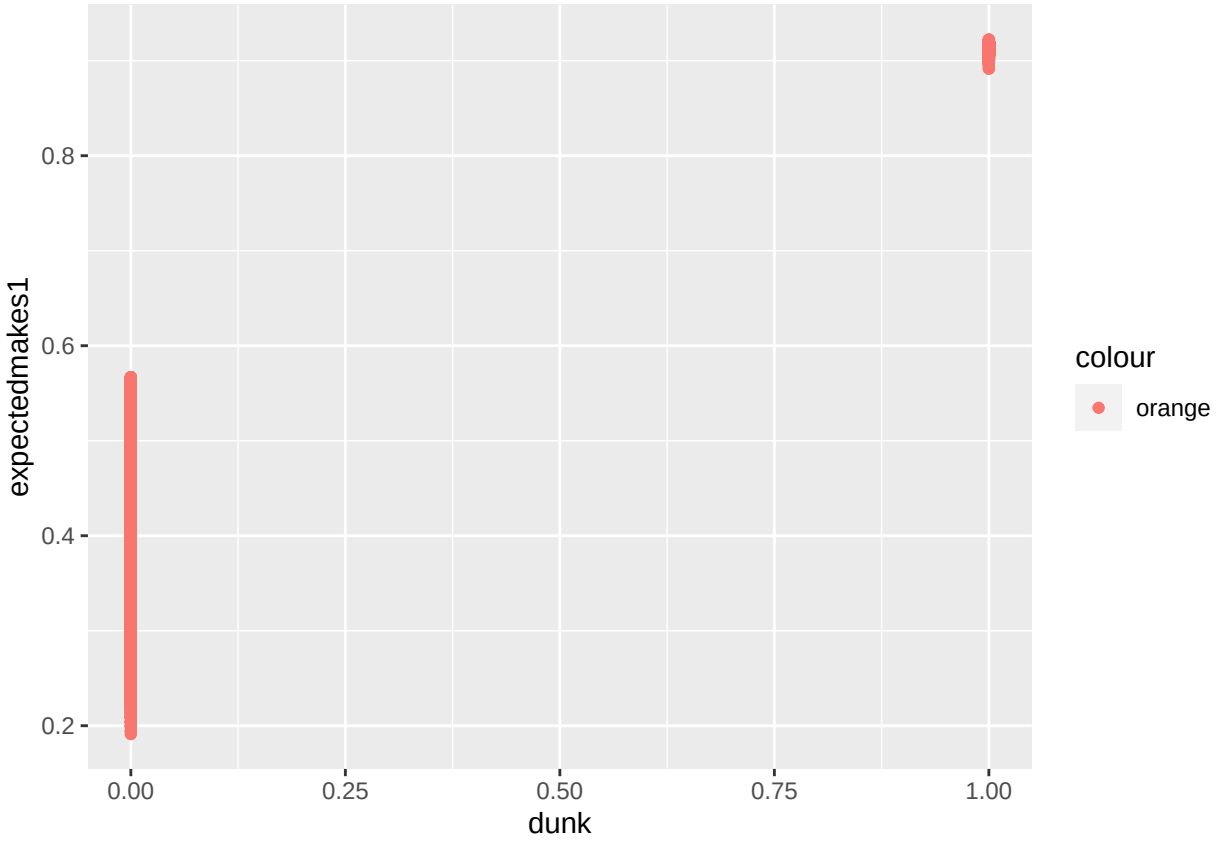

```
ggplot(basket_new,aes(threept, expectedmakes1)) + geom_point()
```



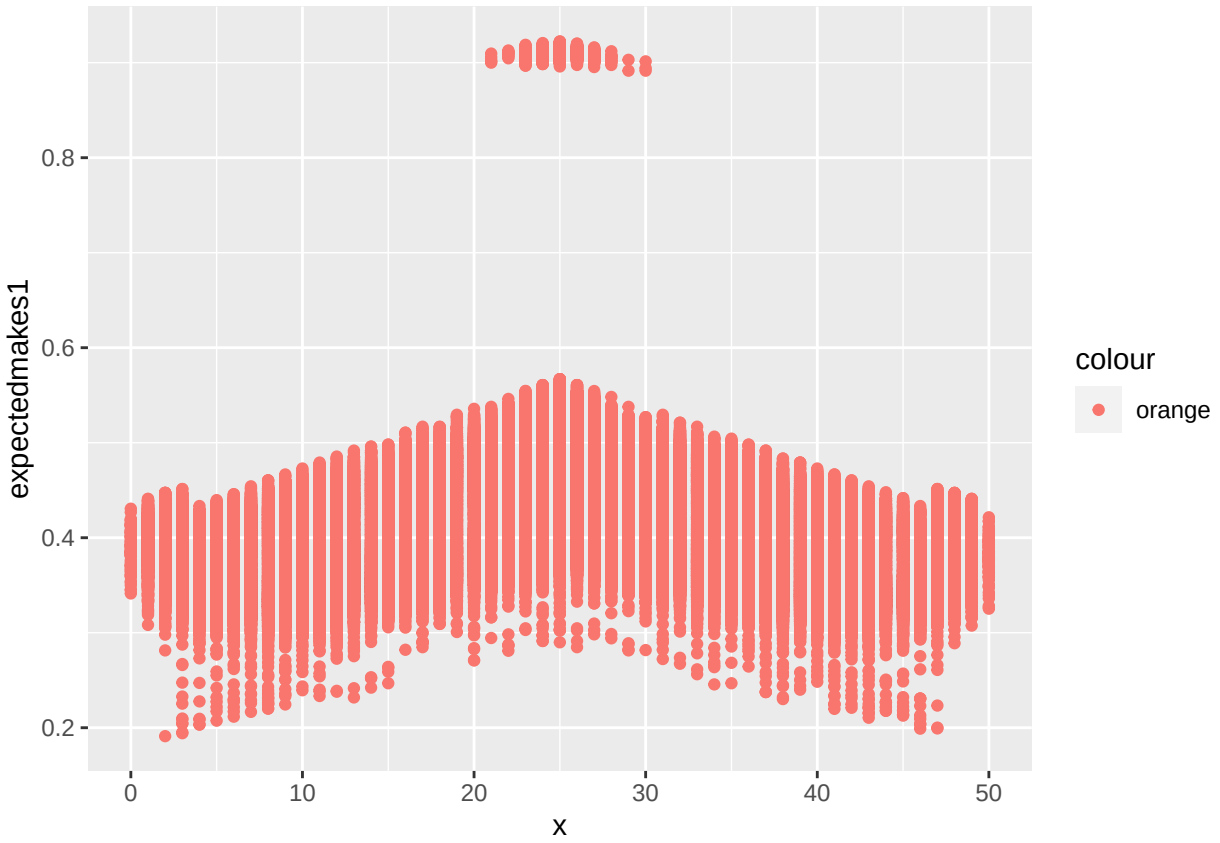

```
ggplot(basket_new,aes(threept, expectedmakes1 ,color=period)) + geom_point() + scale_color_gradient(low
```



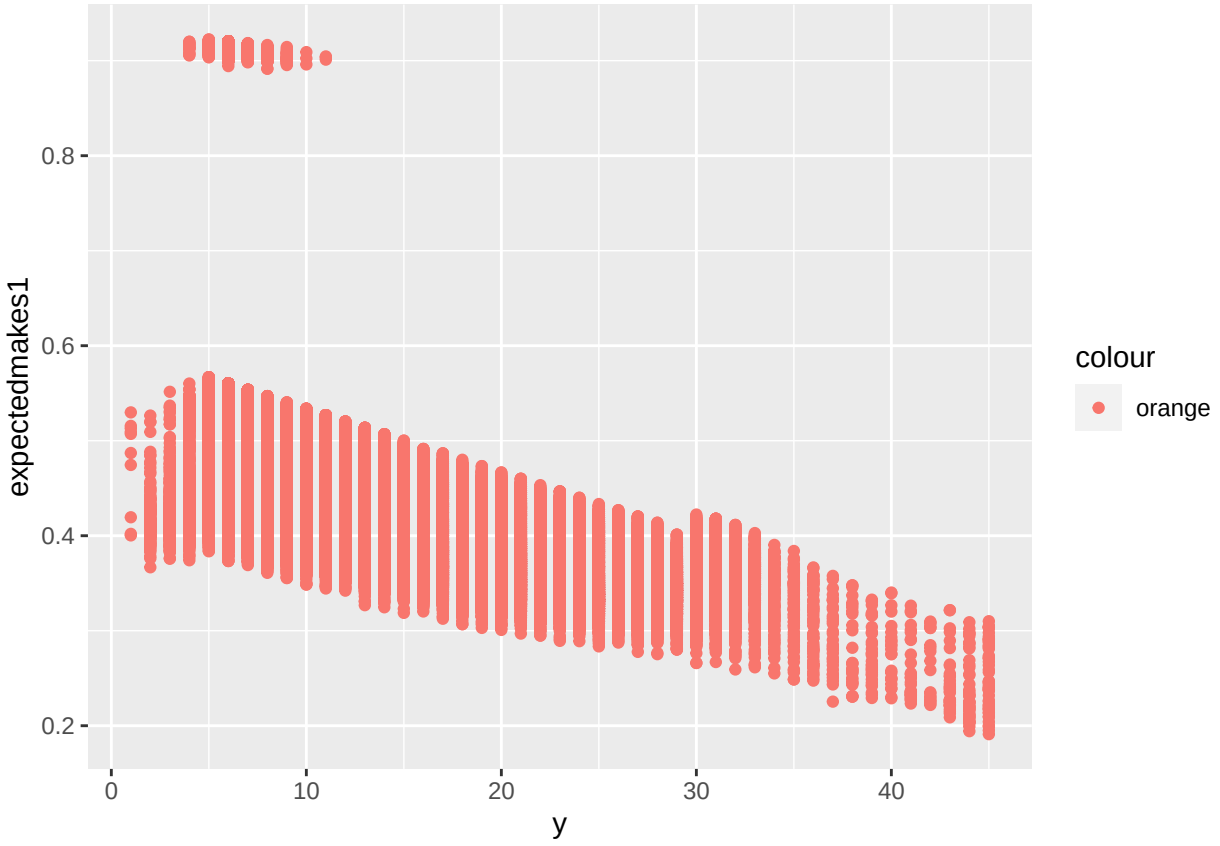

```
ggplot(basket_new,aes(dunk, expectedmakes1, colour='orange')) + geom_point()
```



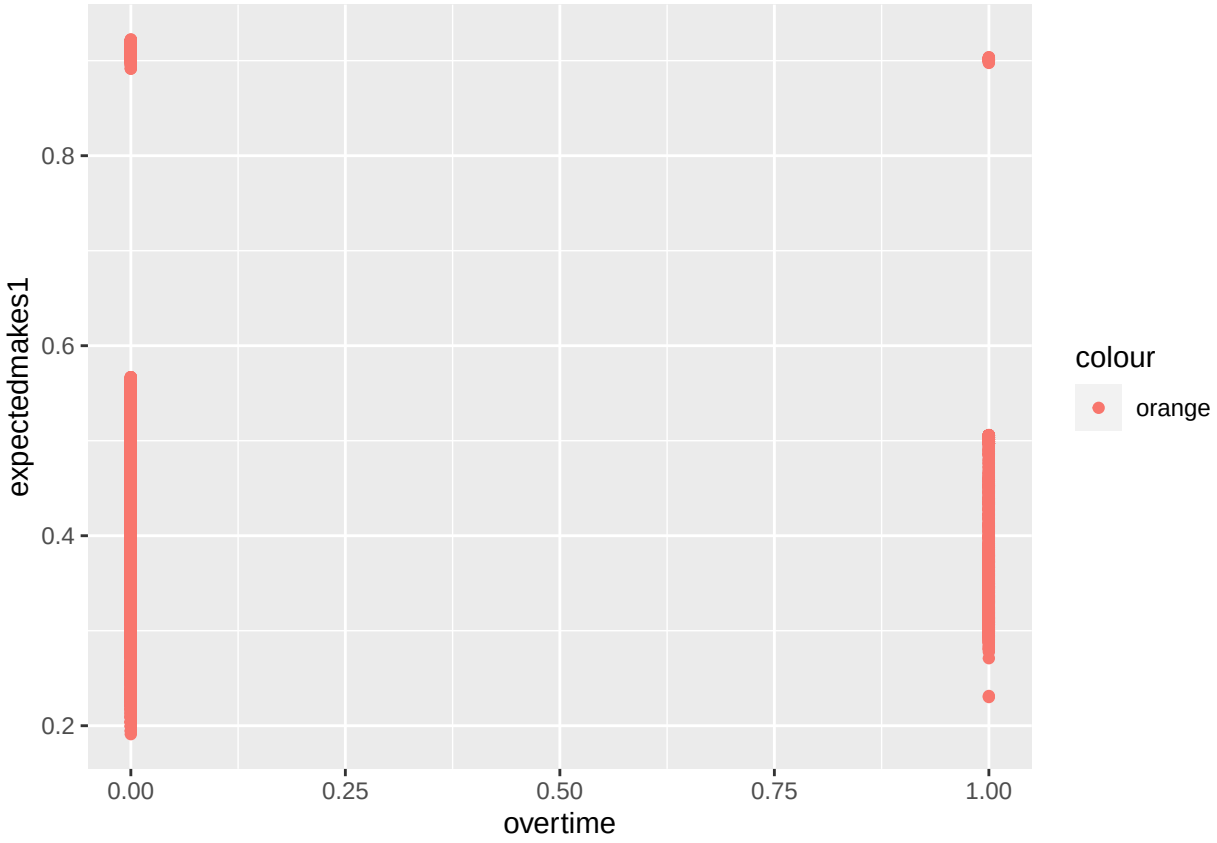

```
ggplot(basket_new,aes(x, expectedmakes1, colour='orange')) + geom_point()
```




```
ggplot(basket_new,aes(y, expectedmakes1, colour='orange')) + geom_point()
```



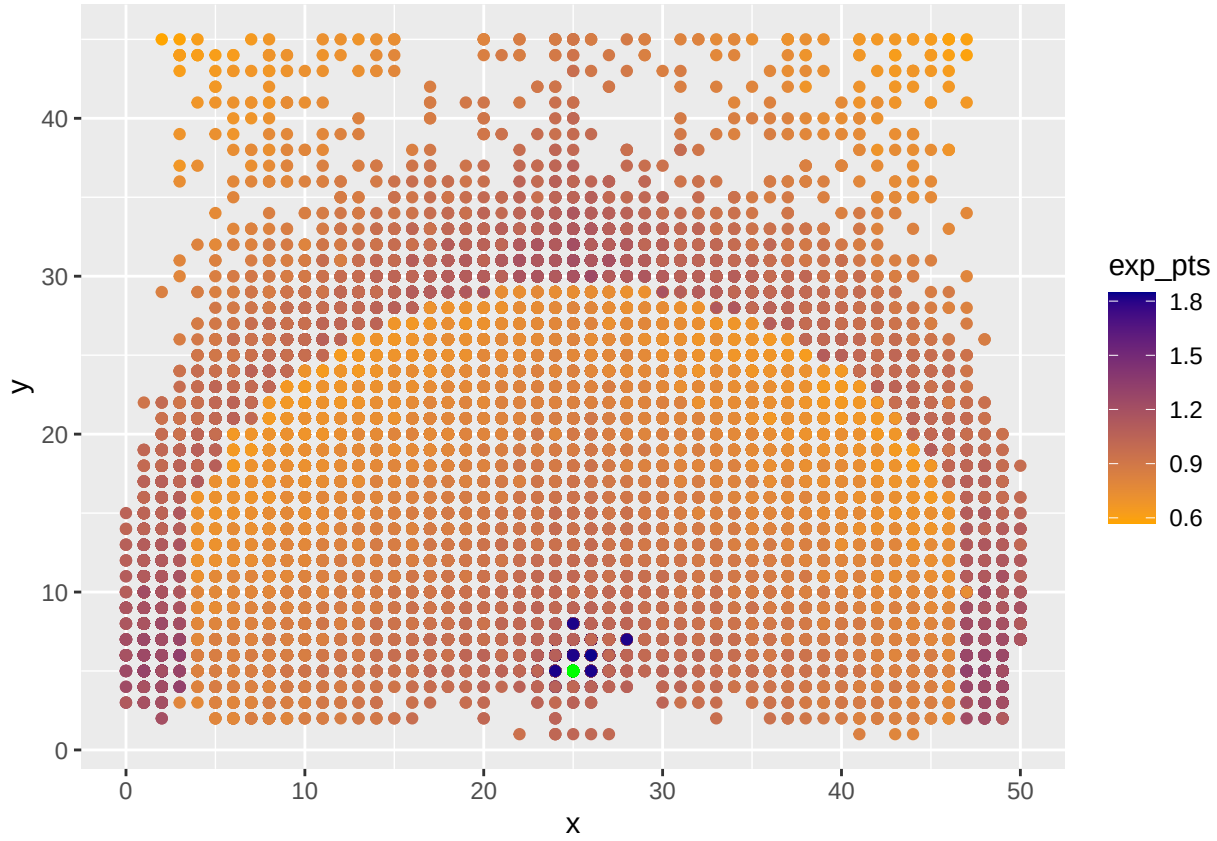

```
ggplot(basket_new,aes(overtime, expectedmakes1, colour='orange')) + geom_point()
```



#Step 2: Heat maps

```
basket_new <- basket_new %>%
  mutate(exp_pts = ifelse(threept==1, (expectedmakes1*3), (expectedmakes1*2)))
```

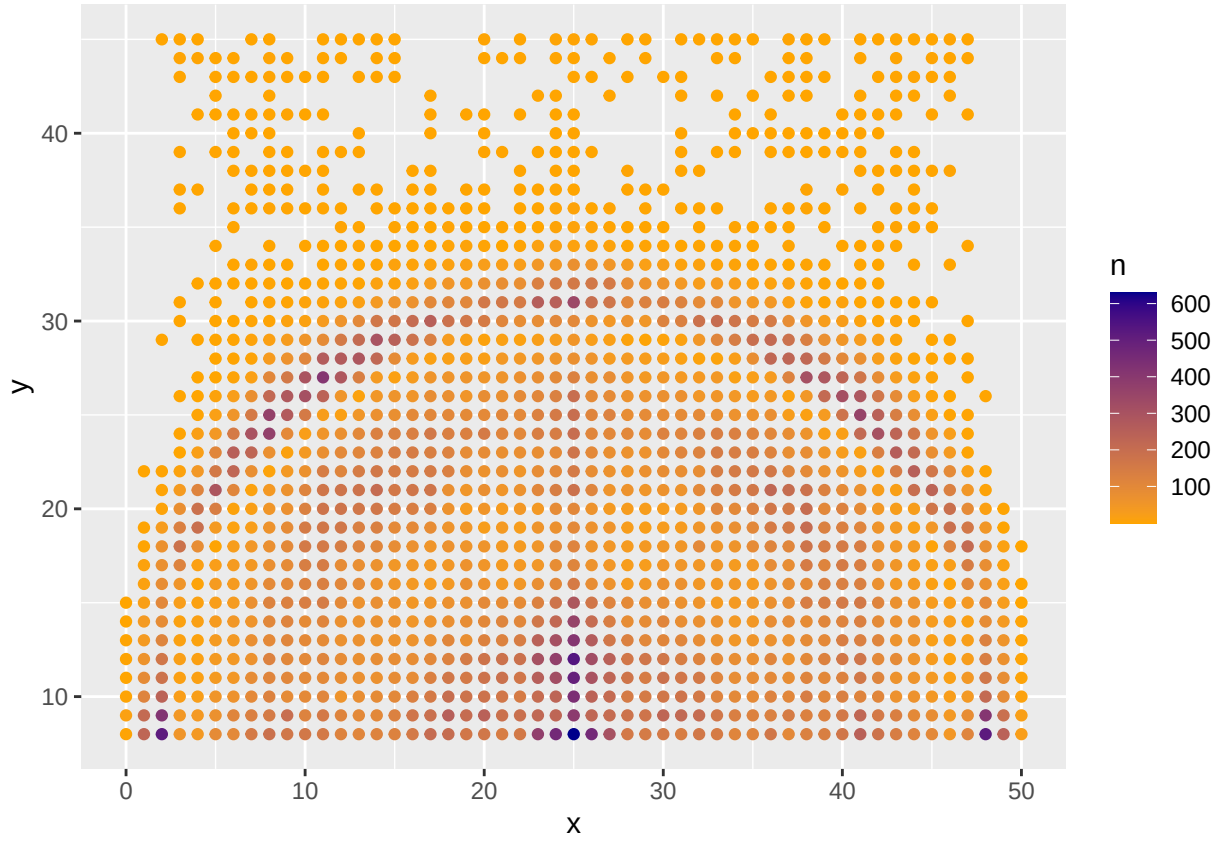
```
ggplot(data = basket_new, aes(x = x, y = y, colour=exp_pts)) +
geom_point() + geom_point(x = 25, y = 5, colour='green') + scale_color_gradient(low = "orange", high =
```




```
coord_shot <- basket_new %>%
  group_by(x, y) %>%
  summarize(n=n())
```

```
## `summarise()` has grouped output by 'x'. You can override using the `.groups`
## argument.
```

```
yscale <- coord_shot %>%
  filter(y>7)
ggplot(data = yscale, aes(x = x, y = y, colour=n)) +
geom_point() + scale_color_gradient(low = "orange", high = "darkblue", n.breaks=10)
```



From the two visualizations, we can see that shots that were made right around the basket (including dunks) have high expected point values. For two pointers, we do see the gradient going towards 'orange' as we approach the three point line. This indicates that two point shots taken away from the basket have low expected points. Three pointers are best when shot from the edge of the line.

This is also shown in the second output which shows the number of shots by location. Most shots are taken as dunks/layups and three pointers (at the edge of the line).

The implication from this is that shots have high value when they are dunks or three pointers. Layups can also be included in this as those shots are made near the basket. Shots that are made inside the three point line have lower expected points. The ratio, for eg, a 0.6 expected shot, inside the three point line, has a ratio of 0.3; and a shot made from the edge (a three pointer) with an expected value of 1.5 has a ratio of 0.5. It is also worth mentioning that dots that have a shade of purple inside the two point line have a ratio or expected value of at least 50% of the points available.

Three point shots have more value than two point shots taken close to the three point line. Dunks and layups have maximum value.

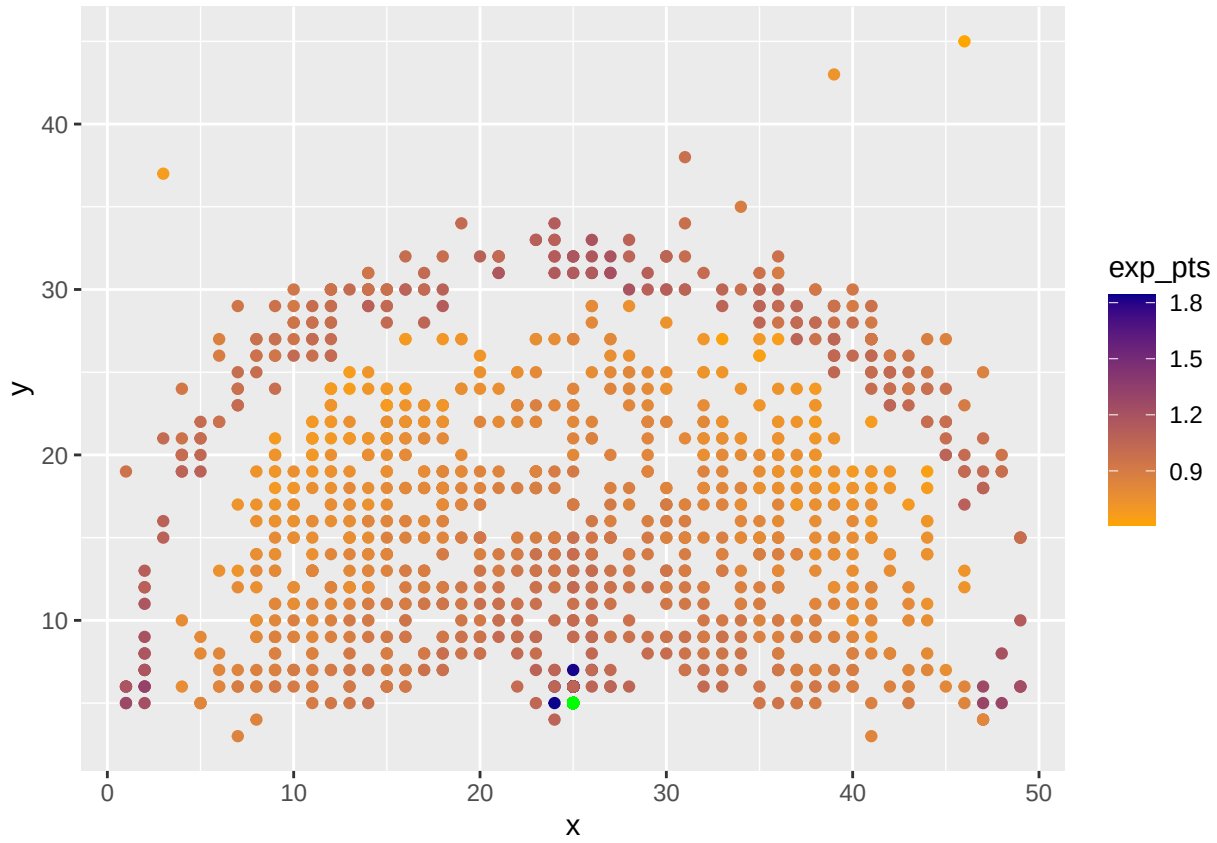
#Step 3: Shot charts

```
basket_new <- basket_new %>%
  mutate(resd=(shot - exp_pts))
```

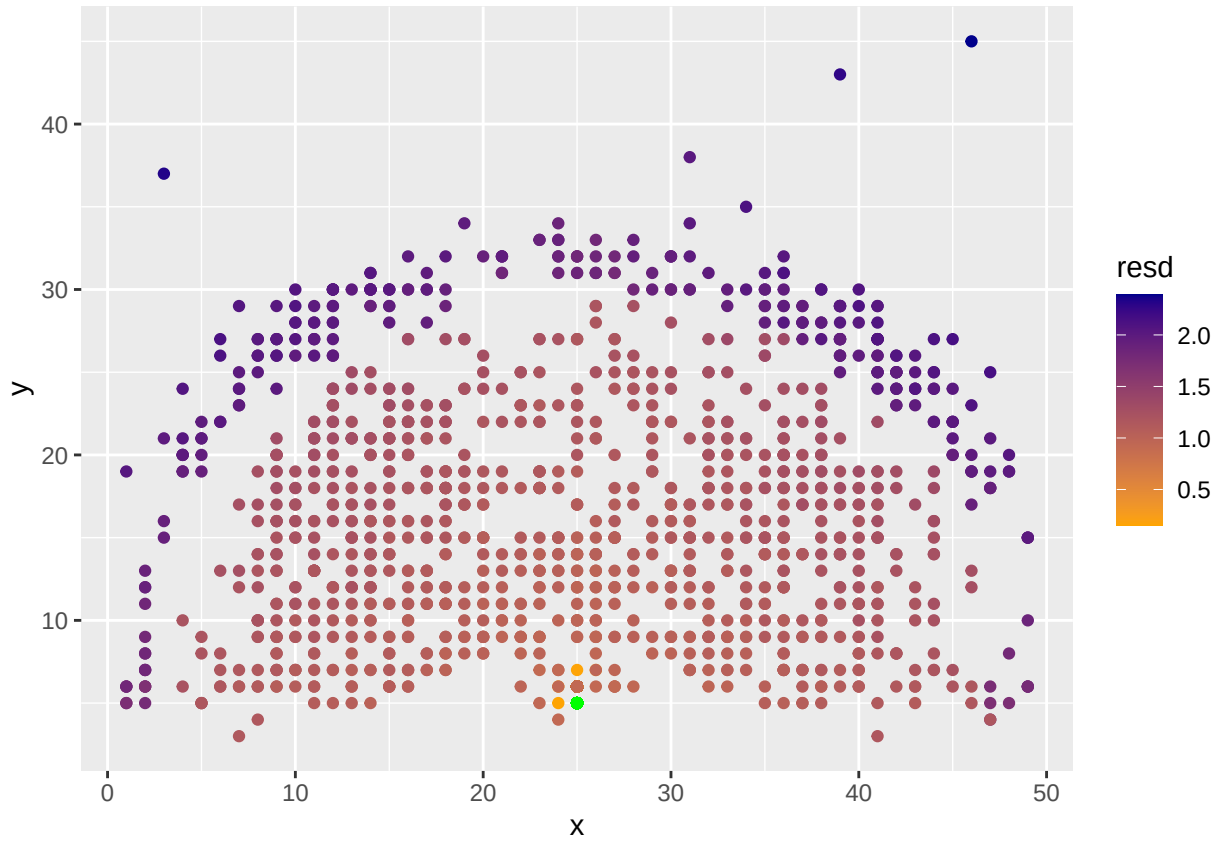
#PLAYER : KOBE BRYANT

```
kobe <- basket_new %>%
  filter(player=='Kobe Bryant')
```

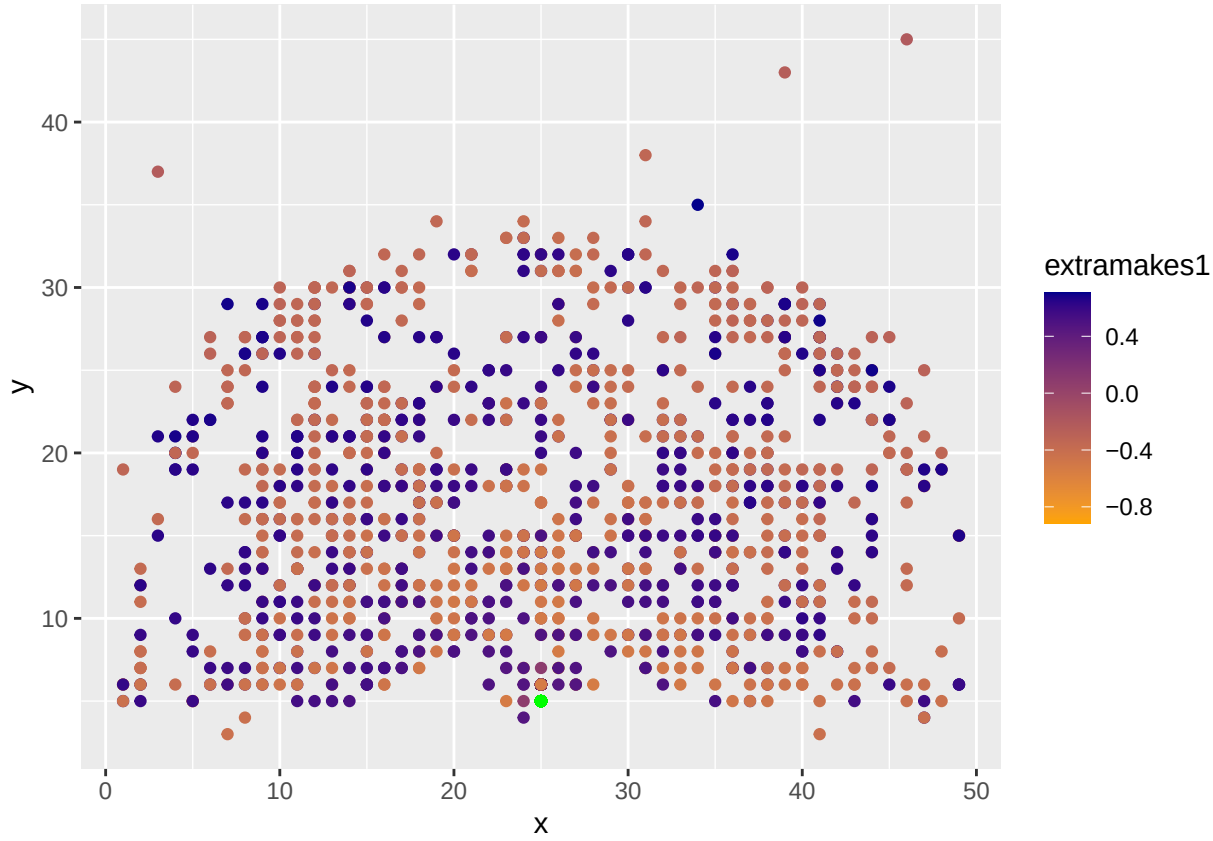
```
ggplot(data = kobe, aes(x = x, y = y, colour=exp_pts)) +
geom_point() + geom_point(x = 25, y = 5, colour='green') + scale_color_gradient(low = "orange", high =
```




```
ggplot(data = kobe, aes(x = x, y = y, colour=resd)) +
geom_point() + geom_point(x = 25, y = 5, colour='green') + scale_color_gradient(low = "orange", high =
```




```
ggplot(data = kobe, aes(x = x, y = y, colour=extramakes1)) +
geom_point() + geom_point(x = 25, y = 5, colour='green') + scale_color_gradient(low = "orange", high =
```



#Step 4: Fixed effects regression

```
player_800 <- basket_new %>%
  group_by(player) %>%
  summarize(n=n())

player_800 <- player_800 %>%
  filter(n>800)

player_800
```

```
## # A tibble: 78 x 2
##    player              n
##    <chr>           <int>
##  1 Aaron Brooks     1320
##  2 Al Harrington    1010
##  3 Al Horford        831
##  4 Al Jefferson     1114
##  5 Amare Stoudemire 1246
##  6 Andray Blatche    998
##  7 Andre Iguodala   1083
##  8 Andre Miller      877
##  9 Andrea Bargnani  1134
## 10 Andrew Bogut      845
## # i 68 more rows
```

```r
pl_800 <- list(player_800$player)

basket_800 <- merge(basket_new, player_800, by.y = 'player')
```

```r
length(unique(basket_800$player))
```

```
## [1] 78
```

```r
fit.2 <- glm(made ~ period + timemin + x_1 + y_1 + threept + dunk + overtime + factor(player), data = ba
```

```r
sigball <- summary(fit.2)$coefficients[summary(fit.2)$coefficients[,4] <.05,]
sigball
```

```
##                                    Estimate  Std. Error    z value
## (Intercept)                      0.324340421 0.060924206   5.323671
## period                          -0.041034130 0.006501013  -6.311959
## timemin                          0.008908038 0.002152335   4.138778
## x_1                             -0.028698269 0.001154368 -24.860587
## y_1                             -0.028670859 0.001077927 -26.598134
## threept                          0.127660543 0.025366878   5.032568
## dunk                             2.321995859 0.065634876  35.377470
## factor(player)Baron Davis       -0.194421605 0.086383221  -2.250687
## factor(player)Beno Udrih         0.248908141 0.090817739   2.740744
## factor(player)Brandon Jennings  -0.280465093 0.083987263  -3.339377
## factor(player)Brook Lopez       -0.204158378 0.086260080  -2.366777
## factor(player)Corey Brewer      -0.188658362 0.089071788  -2.118048
## factor(player)Devin Harris      -0.253142149 0.092370982  -2.740494
## factor(player)Gerald Wallace    -0.175460523 0.089001646  -1.971430
## factor(player)Jonny Flynn       -0.189112949 0.087736042  -2.155476
## factor(player)Josh Smith        -0.242751495 0.089035356  -2.726462
## factor(player)Rodney Stuckey    -0.289638231 0.084374191  -3.432782
## factor(player)Russell Westbrook -0.287413835 0.083438324  -3.444626
## factor(player)Stephen Curry      0.162666454 0.082559870   1.970285
## factor(player)Steve Nash         0.358002347 0.086339165   4.146465
## factor(player)Trevor Ariza      -0.358301451 0.089183198  -4.017589
##                                     Pr(>|z|)
## (Intercept)                     1.016938e-07
## period                          2.755247e-10
## timemin                         3.491609e-05
## x_1                            1.986749e-136
## y_1                            7.133895e-156
## threept                         4.839526e-07
## dunk                           3.792264e-274
## factor(player)Baron Davis       2.440536e-02
## factor(player)Beno Udrih        6.130031e-03
## factor(player)Brandon Jennings  8.396667e-04
## factor(player)Brook Lopez       1.794374e-02
## factor(player)Corey Brewer      3.417097e-02
## factor(player)Devin Harris      6.134685e-03
## factor(player)Gerald Wallace    4.867470e-02
## factor(player)Jonny Flynn       3.112459e-02
## factor(player)Josh Smith        6.401737e-03
## factor(player)Rodney Stuckey    5.974216e-04
## factor(player)Russell Westbrook 5.718497e-04
## factor(player)Stephen Curry     4.880574e-02
```

```
## factor(player)Steve Nash         3.376474e-05
## factor(player)Trevor Ariza       5.879659e-05
```

```
sigball <- data.frame(sigball)
sigball$Estimate <- exp(sigball$Estimate)
```

```
sigball %>% select(Estimate) %>%
  arrange(-Estimate)
```

```
##                                  Estimate
## dunk                            10.1960038
## factor(player)Steve Nash         1.4304690
## (Intercept)                      1.3831181
## factor(player)Beno Udrih         1.2826242
## factor(player)Stephen Curry      1.1766442
## threept                          1.1361673
## timemin                          1.0089478
## y_1                              0.9717362
## x_1                              0.9717096
## period                           0.9597964
## factor(player)Gerald Wallace     0.8390705
## factor(player)Corey Brewer       0.8280694
## factor(player)Jonny Flynn        0.8276930
## factor(player)Baron Davis        0.8233107
## factor(player)Brook Lopez        0.8153332
## factor(player)Josh Smith         0.7844664
## factor(player)Devin Harris       0.7763575
## factor(player)Brandon Jennings   0.7554323
## factor(player)Russell Westbrook  0.7502012
## factor(player)Rodney Stuckey     0.7485343
## factor(player)Trevor Ariza       0.6988624
```

Conditional on the shot being taken by a player who had taken more than 800 shots in 2009, if that specific player was Steve Nash, Beno Udrih, or Steph Curry, it would positively affect the odds of the shot being made. They increase the odds by 43%, 28% and 17% respectively. ALl other players who would be taking that shot (in the 800 shot category) have a negative effect on odds. This goes to show the quality that Steve Nash, Beno Udrih and Steph Curry possess and effectiveness of their shot making in 2009.

#Step 5: Correlations How correlated, by location, is shot frequency with value? With a regression, predict shot frequency by location with shot value by location. How do you interpret the coefficient? Do the same for the player you chose in step 3.

```
coord_shot <- basket_new %>%
  group_by(x, y) %>%
  summarize(n=n(), value=(mean(exp_pts)))
```

```
## `summarise()` has grouped output by 'x'. You can override using the `.groups`
## argument.
```

```
coord_shot
```

```
## # A tibble: 1,798 x 4
## # Groups:   x [51]
##        x     y     n value
##    <int> <int> <int> <dbl>
##  1     0     3     3  1.17
##  2     0     4     2  1.20
```

```
##  3     0     5     8  1.24
##  4     0     6    15  1.21
##  5     0     7    10  1.20
##  6     0     8     5  1.13
##  7     0     9    16  1.16
##  8     0    10     5  1.12
##  9     0    11     2  1.11
## 10     0    12     3  1.07
## # i 1,788 more rows
```

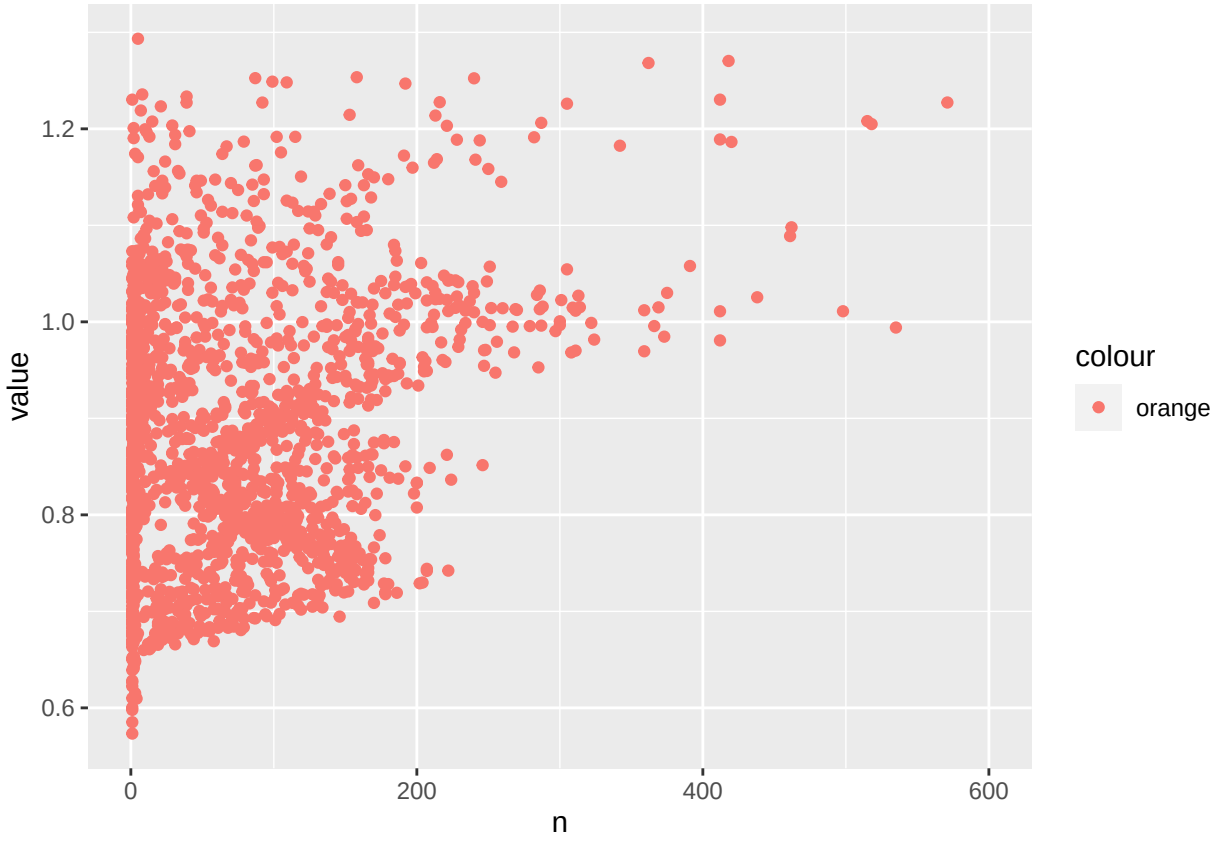
```
coord_shot <- data.frame(coord_shot)
coord_shot <- coord_shot %>% filter(n<1500)
```

```
cor(coord_shot$n, coord_shot$value)
```

```
## [1] 0.2944158
```

```
ggplot(coord_shot, aes(x=n, value, colour='orange')) + geom_point() + xlim(0, 600)
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```
linearDist <- lm(n ~ value, data = coord_shot)
summary(linearDist)
```

```
##
## Call:
## lm(formula = n ~ value, data = coord_shot)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -169.23  -68.35  -13.13   43.26 1207.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -122.55      15.80  -7.755 1.47e-14 ***
## value         229.46      17.58  13.052  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100.4 on 1795 degrees of freedom
## Multiple R-squared:  0.08668,    Adjusted R-squared:  0.08617
## F-statistic: 170.4 on 1 and 1795 DF,  p-value: < 2.2e-16
```

```
coord_shot <- coord_shot %>%
  mutate(LinearPrediction = fitted(linearDist))
```

The regression shows a positive relation of number of shots by location with value by location. FOr every 0.1 increase in value, the number of shots taken from that location increases by 15 on average.

```
kobe_shot <- kobe %>%
  group_by(x, y) %>%
  summarize(n=n(), value=(mean(exp_pts)))
```

```
## `summarise()` has grouped output by 'x'. You can override using the `.groups`
## argument.
```

```
kobe_shot
```

```
## # A tibble: 695 x 4
## # Groups:   x [49]
##        x     y     n value
##    <int> <int> <int> <dbl>
##  1     1     5     2 1.28
##  2     1     6     3 1.22
##  3     1    19     1 0.962
##  4     2     5     1 1.23
##  5     2     6     3 1.27
##  6     2     7     3 1.20
##  7     2     8     2 1.21
##  8     2     9     1 1.20
##  9     2    11     1 1.18
## 10     2    12     2 1.10
## # i 685 more rows
```

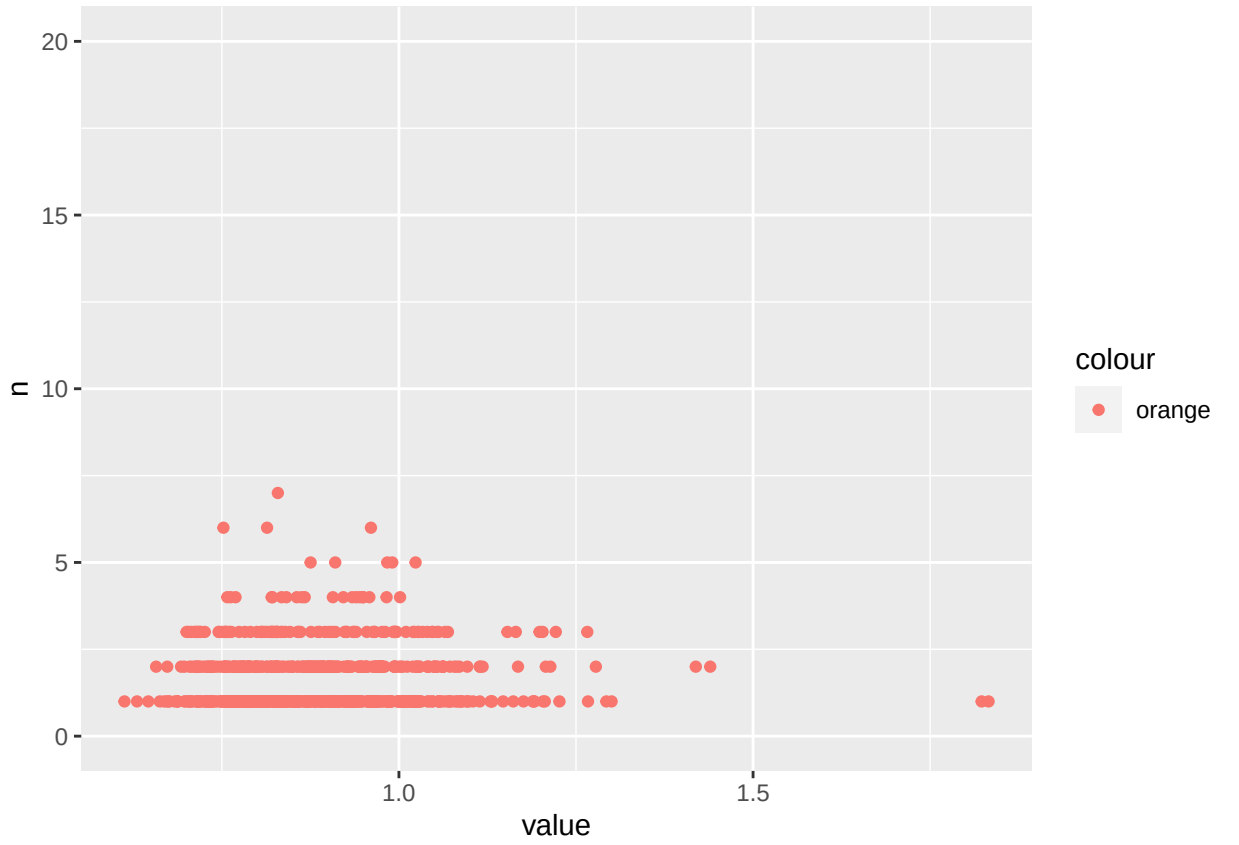
```
kobe_shot <- data.frame(kobe_shot)
```

```
cor(kobe_shot$n, kobe_shot$value)
```

```
## [1] 0.07424705
```

```
ggplot(kobe_shot, aes(x=value, n, colour='orange')) + geom_point() + ylim(0, 20)
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```




```
linearDist <- lm(n ~ value, data = kobe_shot)
summary(linearDist)
```

```
##
## Call:
## lm(formula = n ~ value, data = kobe_shot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -6.529  -1.143  -0.490   0.283 275.382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.063      2.661  -1.151   0.2501
## value          5.780      2.949   1.960   0.0504 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.54 on 693 degrees of freedom
## Multiple R-squared:  0.005513,   Adjusted R-squared:  0.004078
## F-statistic: 3.841 on 1 and 693 DF,  p-value: 0.0504
```

```
coord_shot <- kobe_shot %>%
  mutate(LinearPrediction = fitted(linearDist))
```

The lowest value of a shot that has been taken from a spot where the player has previously shot from, is greater than the lowest value for a shot from a spot where they haven't taken a shot.