

Spatial NBA

Riley Leonard

5/12/2021

```
pacman::p_load(spdep, maps, spData, spatstat, maptools)
```

Load Data

```
data1 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots1.csv")

## 
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_character()
## )
## i Use `spec()` for the full column specifications.

data2 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots2.csv")

## 
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_character()
## )
## i Use `spec()` for the full column specifications.

data3 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots3.csv")

## 
## -- Column specification -----
## cols(
```

```

##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_character()
## )
## i Use `spec()` for the full column specifications.
data4 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots4.csv")

## 
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_logical()
## )
## i Use `spec()` for the full column specifications.
data5 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots5.csv")

## 
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_logical()
## )
## i Use `spec()` for the full column specifications.
data6 <- read_csv("/home/leonardr/Spatial Data NBA/Data/shots6.csv")

## 
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),

```

```

##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_logical()
## )
## i Use `spec()` for the full column specifications.
data <- rbind(data1, data2, data3, data4, data5, data6)

data <- data %>%
  dplyr::select(x, y, name, team_name, period, minutes_remaining,
                seconds_remaining, shot_made_flag, shot_distance,
                dribbles, touch_time, defender_distance, shot_clock)

lillard <- read_csv("/home/leonardr/Spatial Data NBA/Data/lillard.csv")

##
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_character()
## )
## i Use `spec()` for the full column specifications.
gobert <- read_csv("/home/leonardr/Spatial Data NBA/Data/gobert.csv")

##
## -- Column specification -----
## cols(
##   .default = col_double(),
##   name = col_character(),
##   team_name = col_character(),
##   game_date = col_date(format = ""),
##   seconds_remaining = col_character(),
##   action_type = col_character(),
##   shot_type = col_character(),
##   opponent = col_character(),
##   defender_name = col_character()
## )
## i Use `spec()` for the full column specifications.

```

Wrangling

```

data <- data %>%
  filter(shot_distance < 47) %>%
  mutate(shot_outcome = as_factor(shot_made_flag))

lillard <- lillard %>%

```

```

filter(shot_distance < 47) %>%
mutate(shot_outcome = as_factor(shot_made_flag)) %>%
drop_na()

gobert <- gobert %>%
filter(shot_distance < 47) %>%
mutate(shot_outcome = as_factor(shot_made_flag)) %>%
drop_na()

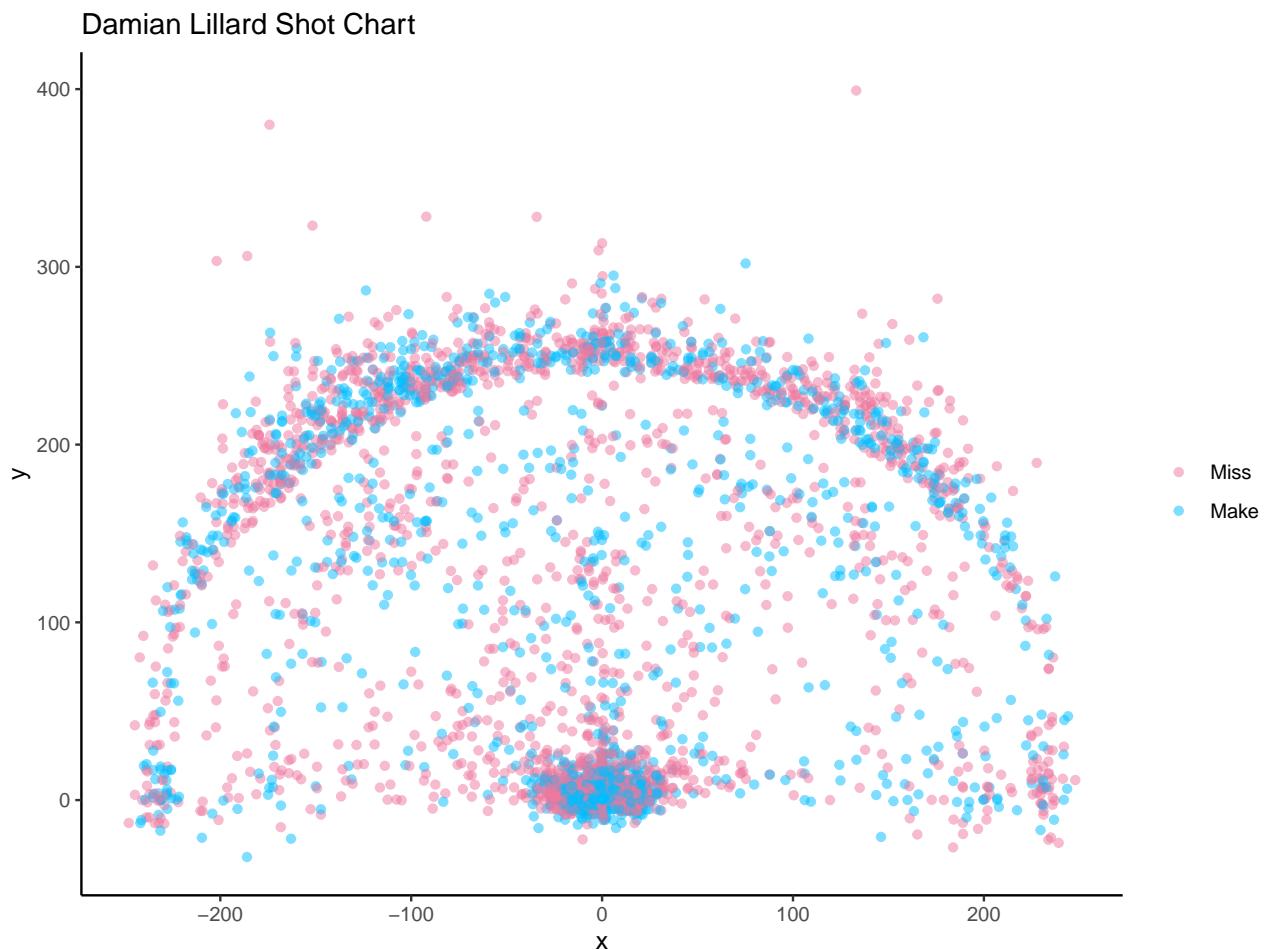
```

Lillard

```

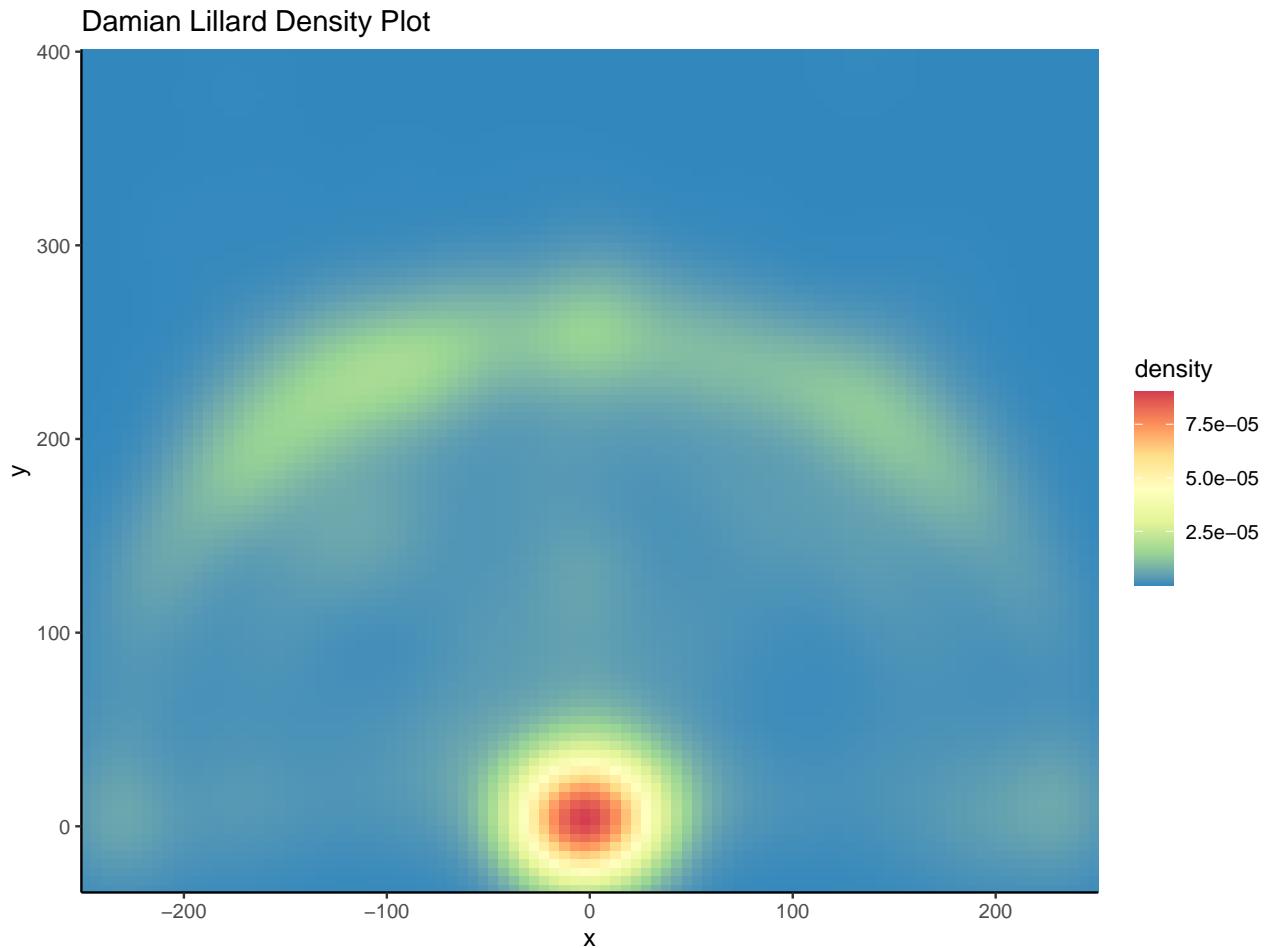
ggplot(lillard,
       aes(x = x,
            y = y)) +
  geom_jitter(alpha = 0.5, aes(color = shot_outcome)) +
  scale_color_manual(name = " ",
                     labels = c("Miss", "Make"),
                     values = c("0" = "palevioletred2", "1" = "deepskyblue")) +
  ggtitle("Damian Lillard Shot Chart") +
  theme_classic()

```



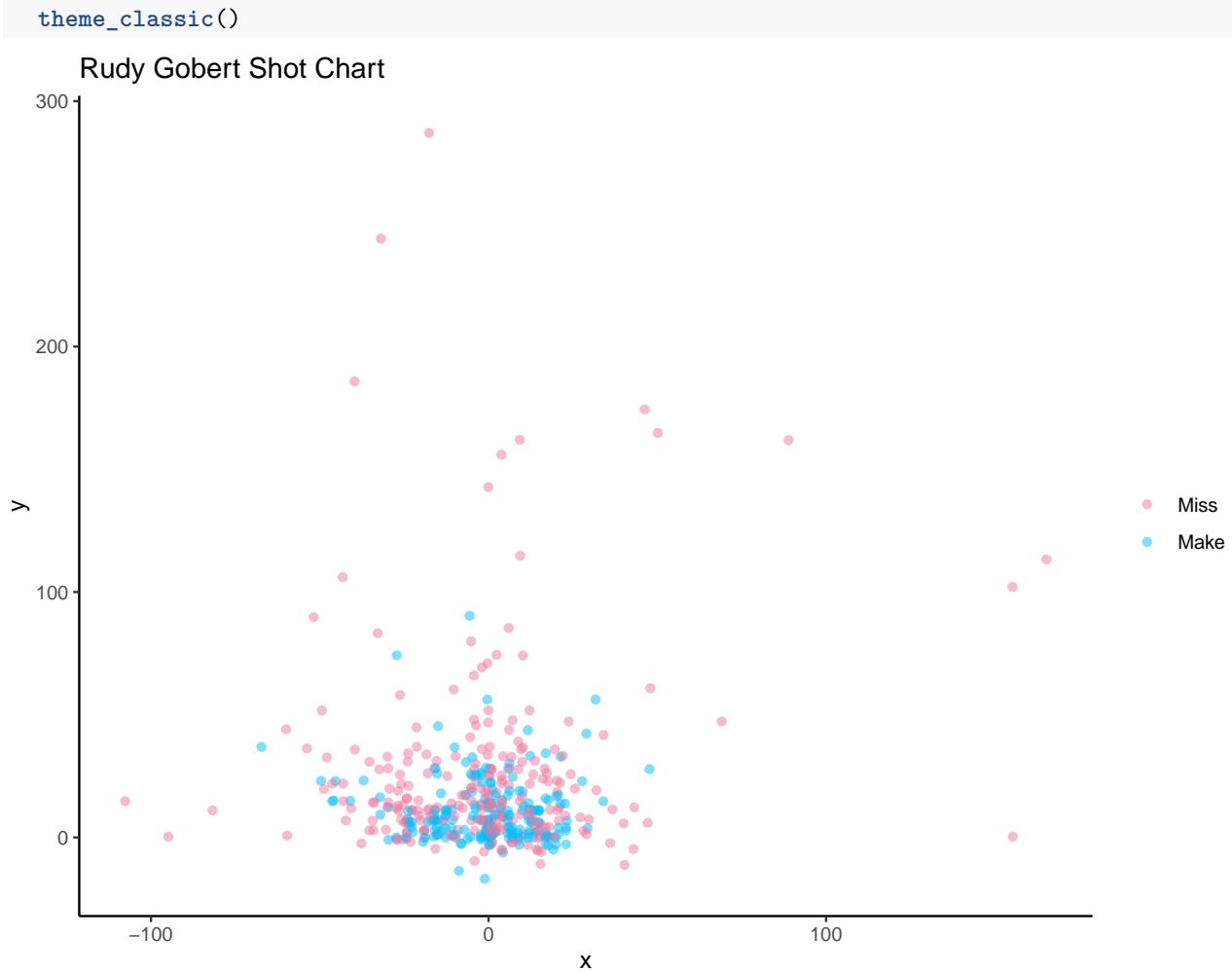
Density

```
ggplot(lillard,
       aes(x = x,
            y = y)) +
  stat_density_2d(aes(fill = ..density..), geom = "raster", contour = FALSE) +
  scale_fill_distiller(palette = "Spectral", direction = -1) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position='none') +
  ggtitle("Damian Lillard Density Plot") +
  theme_classic()
```



Gobert

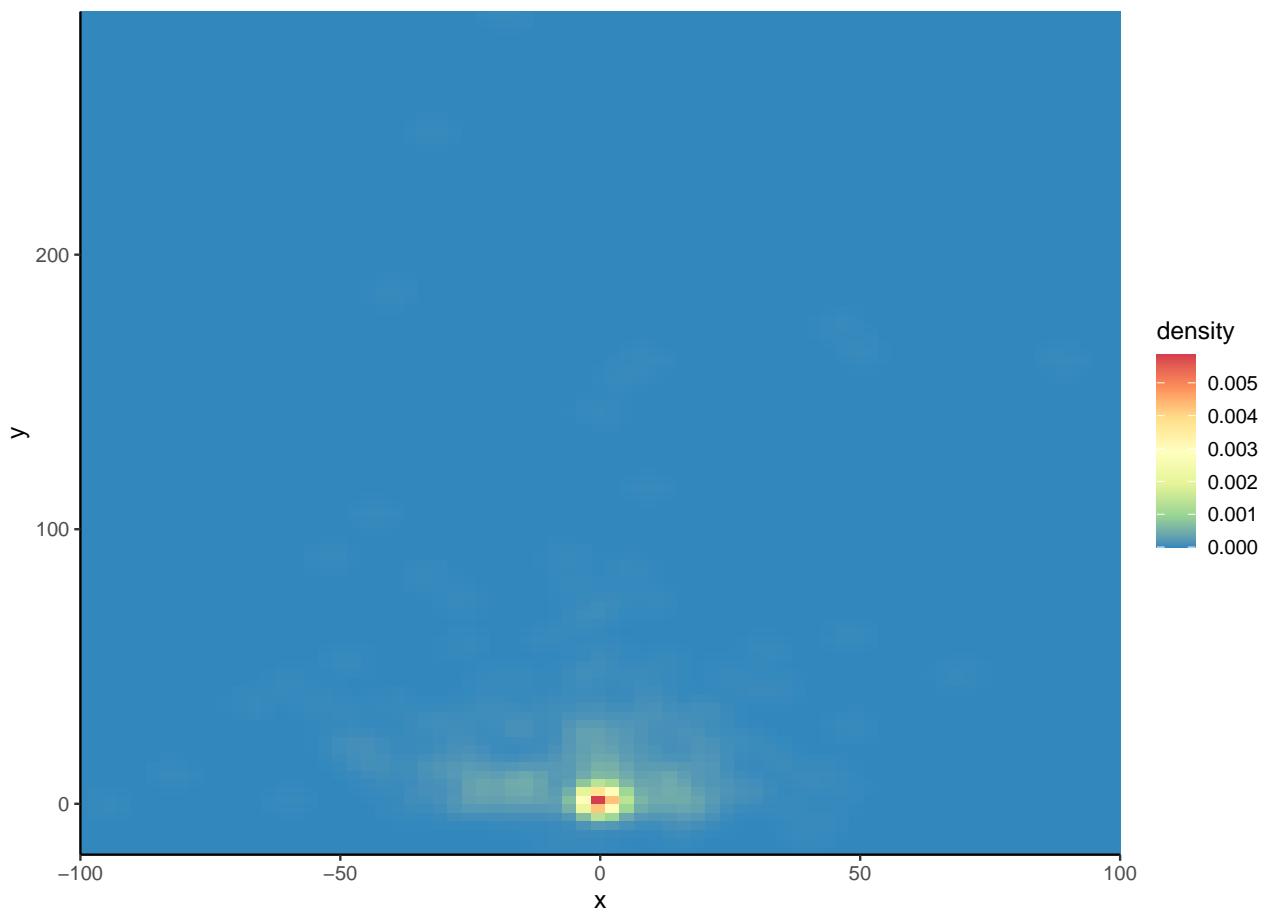
```
ggplot(gobert,
       aes(x = x,
            y = y)) +
  geom_jitter(alpha = 0.5, aes(color = shot_outcome)) +
  scale_color_manual(name = " ",
                     labels = c("Miss", "Make"),
                     values = c("0" = "palevioletred2", "1" = "deepskyblue")) +
  ggtitle("Rudy Gobert Shot Chart") +
```



Density

```
ggplot(gobert,
       aes(x = x,
           y = y)) +
  stat_density_2d(aes(fill = ..density..), geom = "raster", contour = FALSE) +
  scale_fill_distiller(palette = "Spectral", direction = -1) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position='none') +
  coord_cartesian(xlim =c(-100, 100)) +
  ggtitle("Rudy Gobert Density Plot") +
  theme_classic()
```

Rudy Gobert Density Plot



Summary Data

```
summary_data <- data %>%
  group_by(name) %>%
  summarize(avg_dist = mean(shot_distance),
            avg_dribbles = mean(dribbles),
            avg_touch_time = mean(touch_time),
            avg_defender = mean(defender_distance),
            avg_clock = mean(shot_clock),
            fg = sum(shot_made_flag == 1),
            fga = (sum(shot_made_flag == 0) + sum(shot_made_flag == 1)),
            pct = fg/fga)

summary_data <- summary_data %>%
  filter(fga >= 100)
```

Scatterplots

```
# Average Distance vs. FG %

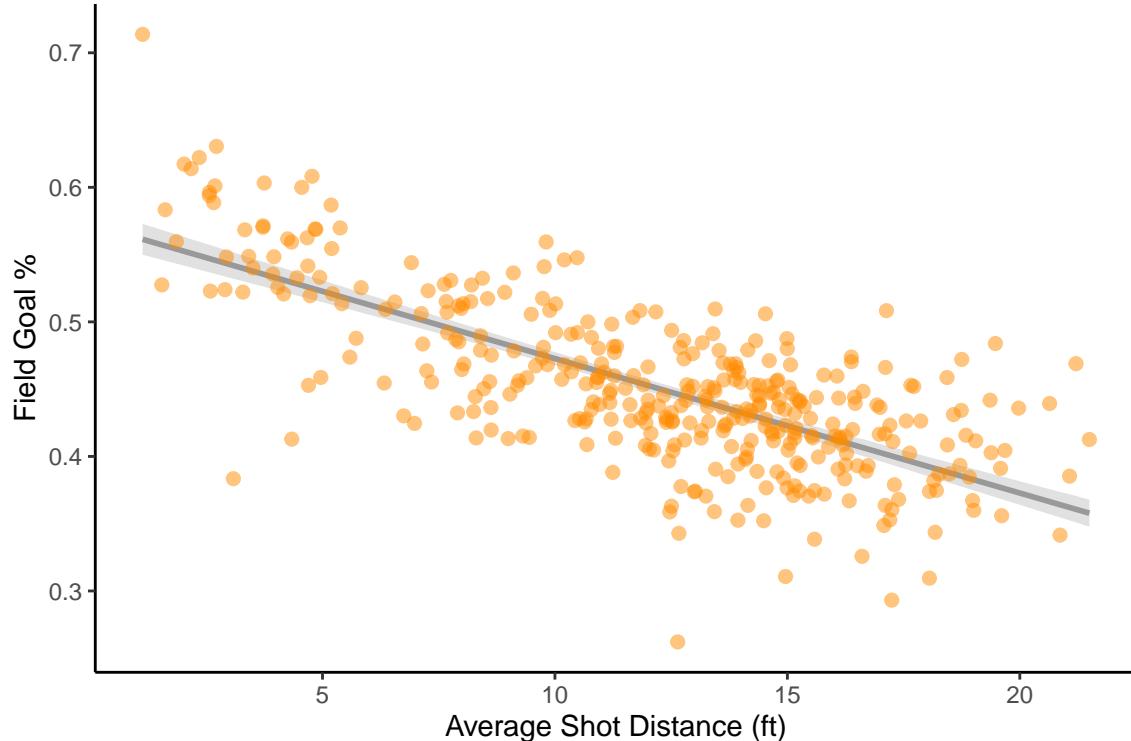
ggplot(summary_data,
       aes(x = avg_dist,
```

```

        y = pct)) +
  geom_smooth(method = lm,
              color = "grey60",
              alpha = 0.3) +
  geom_point(size = 2, alpha = 0.5, color = "darkorange") +
  labs(x = "Average Shot Distance (ft)",
       y = "Field Goal %") +
  theme_classic()

```

`geom_smooth()` using formula 'y ~ x'



```

cor(summary_data$avg_dist,
    summary_data$pct)

```

[1] -0.7265144

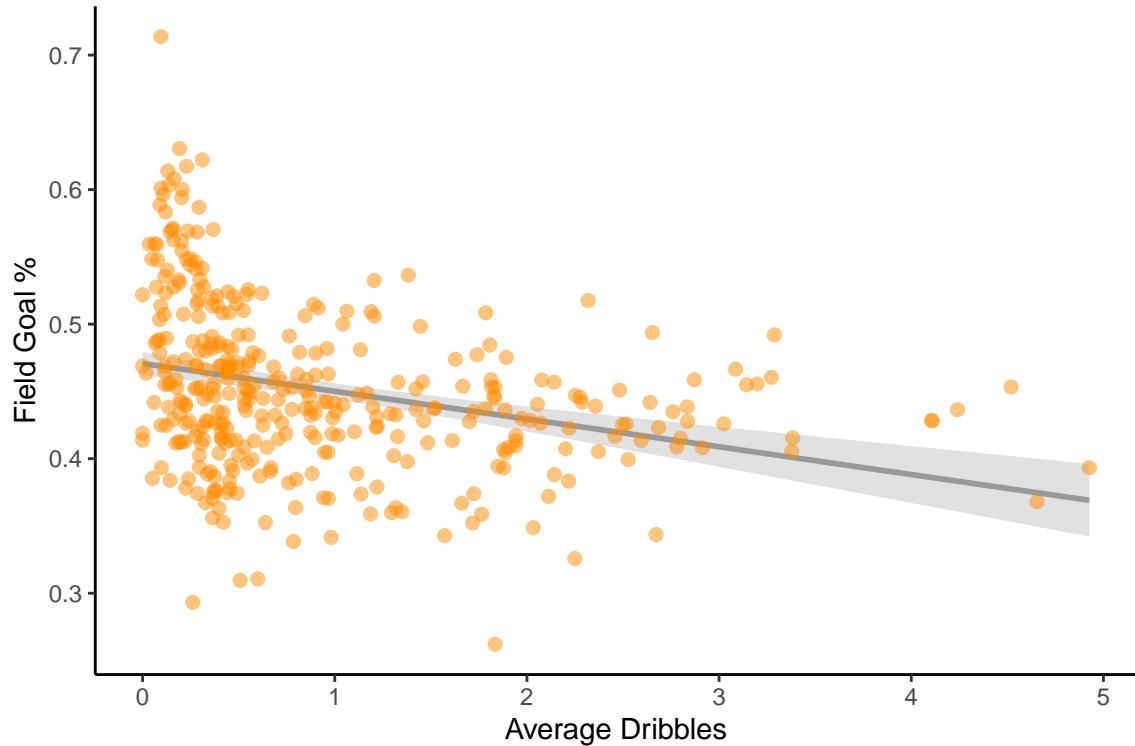
Dribbles vs. FG %

```

ggplot(summary_data,
       aes(x = avg_dribbles,
            y = pct)) +
  geom_smooth(method = lm,
              color = "grey60",
              alpha = 0.3) +
  geom_point(size = 2, alpha = 0.5, color = "darkorange") +
  labs(x = "Average Dribbles",
       y = "Field Goal %") +
  theme_classic()

```

`geom_smooth()` using formula 'y ~ x'



```

cor(summary_data$avg_dribbles,
    summary_data$pct)

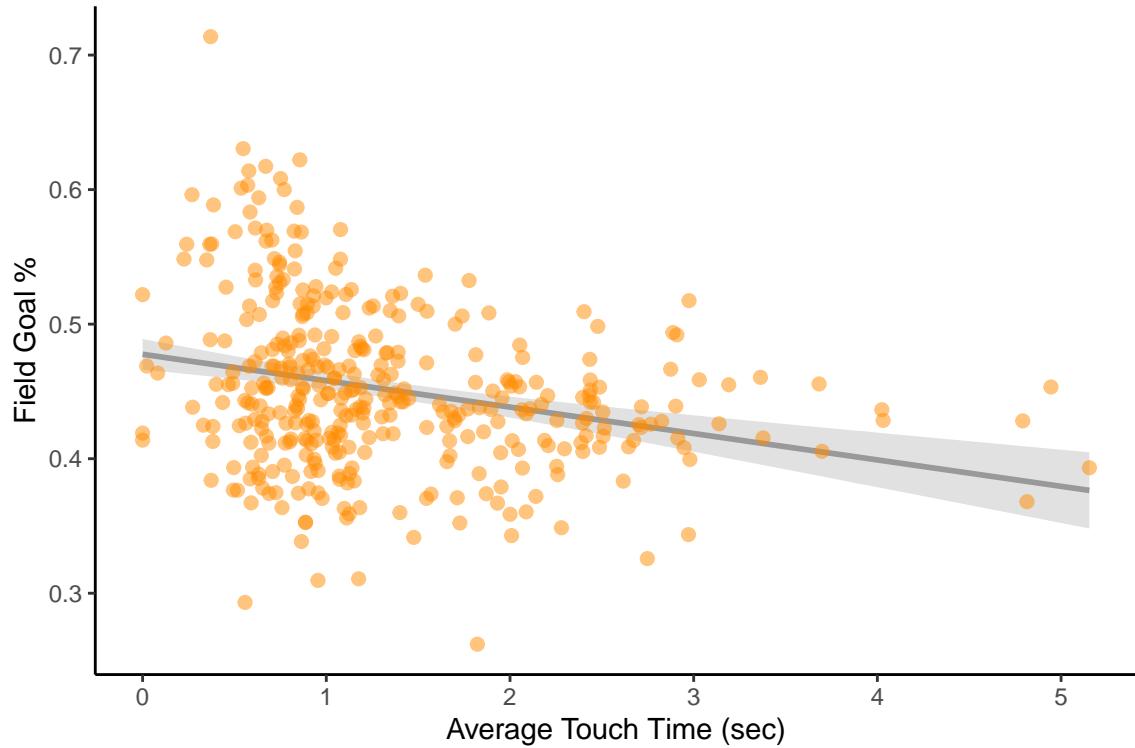
## [1] -0.3046269

# Touch Time vs. FG %

ggplot(summary_data,
       aes(x = avg_touch_time,
            y = pct)) +
  geom_smooth(method = lm,
              color = "grey60",
              alpha = 0.3) +
  geom_point(size = 2, alpha = 0.5, color = "darkorange") +
  labs(x = "Average Touch Time (sec)",
       y = "Field Goal %") +
  theme_classic()

## `geom_smooth()` using formula 'y ~ x'

```



```

cor(summary_data$avg_touch_time,
    summary_data$pct)

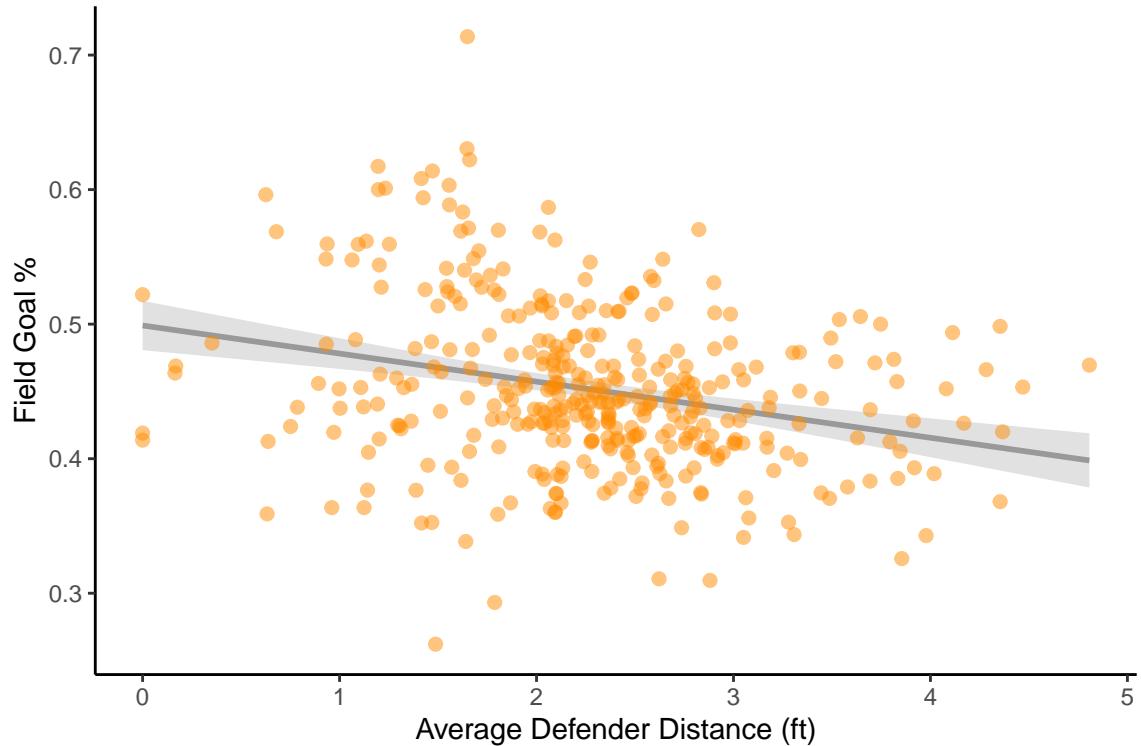
## [1] -0.2673416

# Touch Time vs. FG %

ggplot(summary_data,
       aes(x = avg_defender,
            y = pct)) +
  geom_smooth(method = lm,
              color = "grey60",
              alpha = 0.3) +
  geom_point(size = 2, alpha = 0.5, color = "darkorange") +
  labs(x = "Average Defender Distance (ft)",
       y = "Field Goal %") +
  theme_classic()

## `geom_smooth()` using formula 'y ~ x'

```



```

cor(summary_data$avg_defender,
    summary_data$pct)

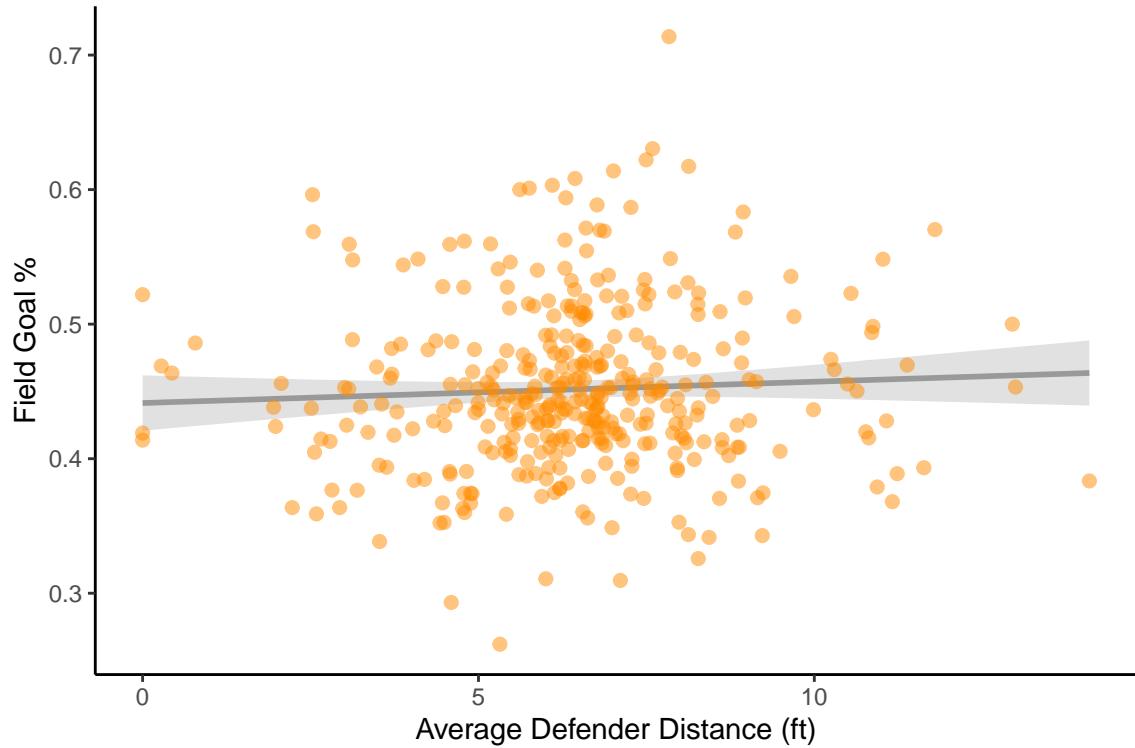
## [1] -0.2705009

# Shot Clock vs. FG %

ggplot(summary_data,
       aes(x = avg_clock,
            y = pct)) +
  geom_smooth(method = lm,
              color = "grey60",
              alpha = 0.3) +
  geom_point(size = 2, alpha = 0.5, color = "darkorange") +
  labs(x = "Average Defender Distance (ft)",
       y = "Field Goal %") +
  theme_classic()

## `geom_smooth()` using formula 'y ~ x'

```

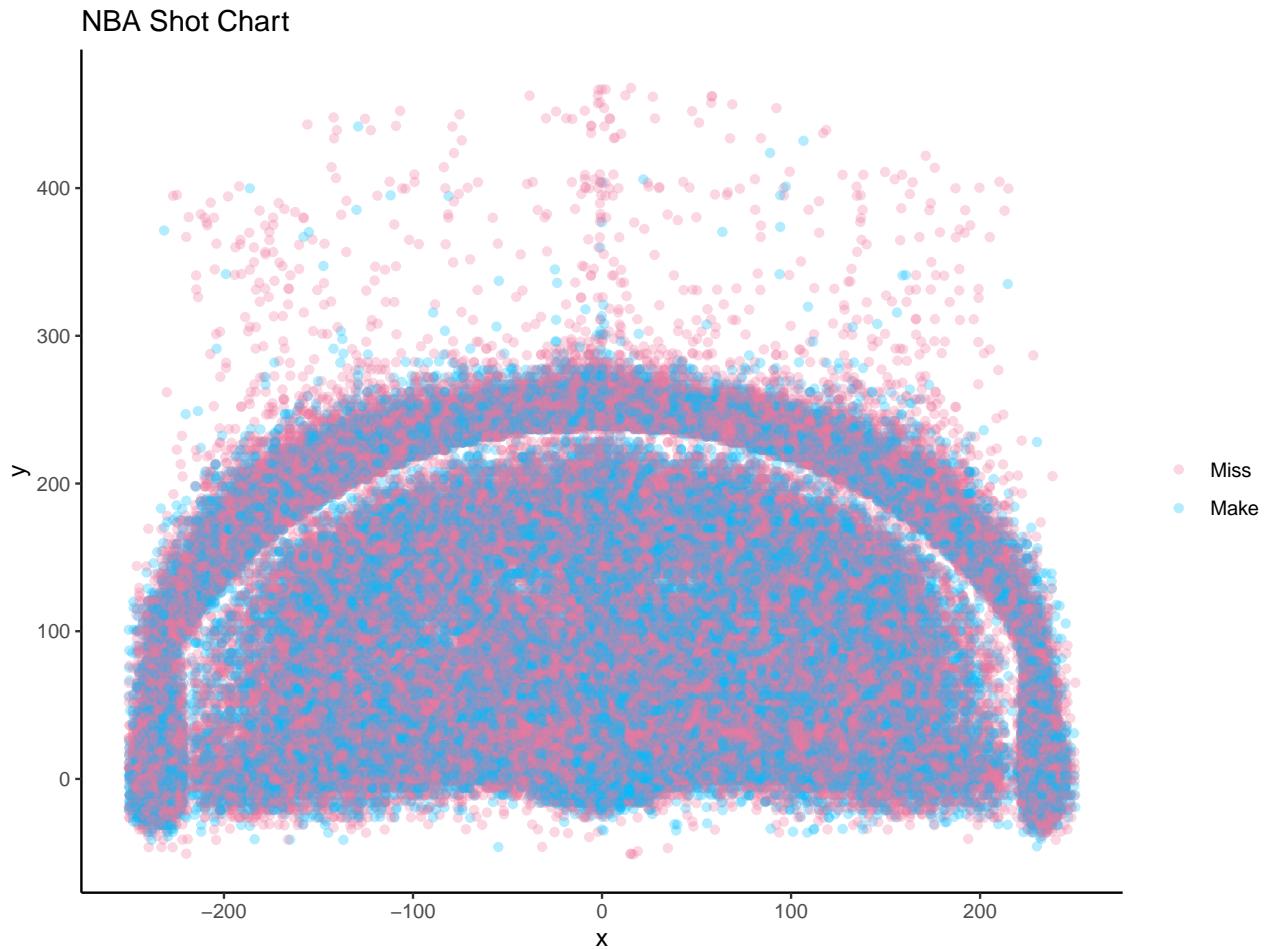


```
cor(summary_data$avg_clock,
    summary_data$pct)
```

```
## [1] 0.05299949
```

League-wide Shot Chart

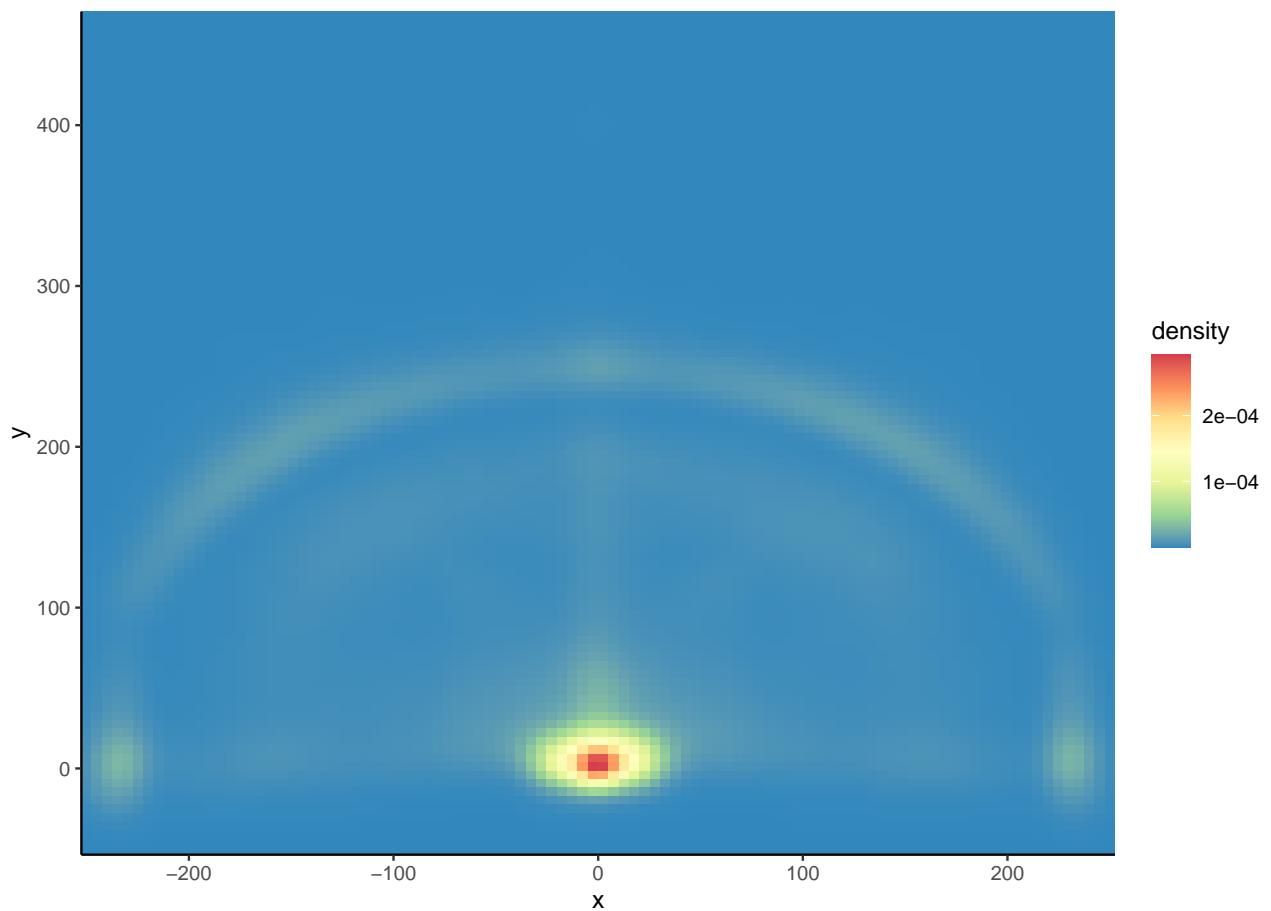
```
ggplot(data,
       aes(x = x,
           y = y)) +
  geom_jitter(alpha = 0.3, aes(color = shot_outcome)) +
  scale_color_manual(name = " ",
                     labels = c("Miss", "Make"),
                     values = c("0" = "palevioletred2", "1" = "deepskyblue")) +
  ggtitle("NBA Shot Chart") +
  theme_classic()
```



League-wide Density

```
ggplot(data,
  aes(x = x,
      y = y)) +
  stat_density_2d(aes(fill = ..density..), geom = "raster", contour = FALSE) +
  scale_fill_distiller(palette = "Spectral", direction = -1) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  theme(legend.position='none') +
  ggtitle("NBA Density Plot") +
  theme_classic()
```

NBA Density Plot



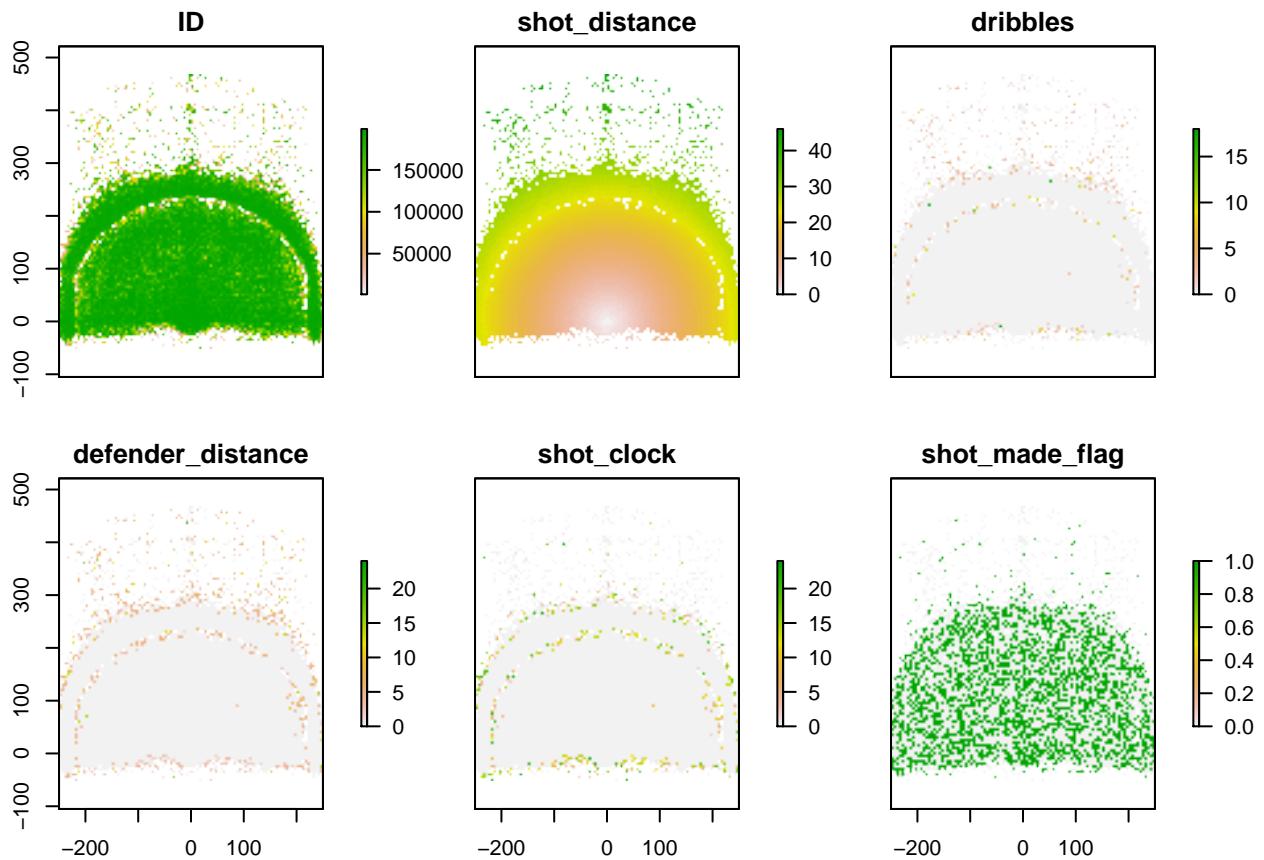
Covariate Raster

```
data_short <- data %>%
  dplyr::select(shot_distance, dribbles,
                defender_distance, shot_clock, shot_made_flag)

pts <- SpatialPoints(c(data[,1], data[,2]))

spData <- SpatialPointsDataFrame(pts, data = data_short)

r <- raster(spData)
res(r) <- 5
f <- rasterize(spData, r)
plot(f)
```



Logistic Model

```

model <- glm(shot_made_flag ~ shot_distance + dribbles + defender_distance + shot_clock,
              data = data, family = "binomial")

summary(model)

##
## Call:
## glm(formula = shot_made_flag ~ shot_distance + dribbles + defender_distance +
##       shot_clock, family = "binomial", data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.3369   -1.0681   -0.8601    1.1415    1.9625
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.3064537  0.0086491  35.432  <2e-16 ***
## shot_distance -0.0444108  0.0005180 -85.728  <2e-16 ***
## dribbles     -0.0424913  0.0018655 -22.777  <2e-16 ***
## defender_distance 0.0384715  0.0020848  18.454  <2e-16 ***
## shot_clock     0.0006611  0.0007793   0.848   0.396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 273991 on 198922 degrees of freedom
## Residual deviance: 265569 on 198918 degrees of freedom
## AIC: 265579
##
## Number of Fisher Scoring iterations: 4
kable(summary(model)$coef, digits = c(3, 3, 3, 4),
"latex", booktabs=T)

```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.306	0.009	35.432	0.0000
shot_distance	-0.044	0.001	-85.728	0.0000
dribbles	-0.042	0.002	-22.777	0.0000
defender_distance	0.038	0.002	18.454	0.0000
shot_clock	0.001	0.001	0.848	0.3963

```

stargazer(model, type = "latex", header=FALSE,
title = "Logistic Regression: Explaining Field Goal Probability")

```

Table 1: Logistic Regression: Explaining Field Goal Probability

<i>Dependent variable:</i>	
	shot_made_flag
shot_distance	-0.044*** (0.001)
dribbles	-0.042*** (0.002)
defender_distance	0.038*** (0.002)
shot_clock	0.001 (0.001)
Constant	0.306*** (0.009)
Observations	198,923
Log Likelihood	-132,784.600
Akaike Inf. Crit.	265,579.200

Note: *p<0.1; **p<0.05; ***p<0.01