Rates Série A 2014-2019

From Dixon & Robinson, 1998

In a match picked a random, let T_{xy} be the time to the next goal while the current score is (x, y) for $x,\ y = 0,\ 1,\ 2,\ ...,$ and let δ_{xy} be a censoring indicator that is 0 if the match ends before the next goal is scored and 1 if a goal is observed. Then, assuming $T_{xy} \sim \exp(\nu_{xy})$, standart survival analysis give the maximum likelihood estimate of ν_{xy} as

$$\hat{\nu}_{xy} = \frac{\sum_{i=1}^{N} \delta_{xy,i}}{\sum_{i=1}^{N} t_{xy,i}}$$

where N is the number of matches, $t_{xy,i}$ and $\delta_{xy,i}$ are the observed times to the next goal and censoring indicators respectively, at score (x, y) in match i.

```
options(knitr.kable.NA = "-",
        scipen = 999)

library(dplyr)
library(knitr)
library(reshape2)
library(ggplot2)

load("input_2014_2019.RData")

x = list(); y = list(); xy = list()
for(i in 1:N) {
  x[[i]] = c(x1[[i]], x2[[i]])
  y[[i]] = c(y1[[i]], y2[[i]])
  xy[[i]] = paste(x[[i]], y[[i]], sep = "-")
}

tables = lapply(xy, table)

scores = NULL
c = 0
for(i in 0:3) {
  for(j in 0:3) {
    c = c + 1
    scores[c] = paste(i, j, sep = "-")
  }
}

delta_home = list(); delta_away = list(); t = list()

for(i in 1:length(scores)) {
  tmp_delta_home = NULL; tmp_delta_away = NULL; tmp_t = NULL
  for(k in 1:N) {
```

```r
    if(scores[i] %in% names(tables[[k]])) {
      next_score = names(tables[[k]])[which(names(tables[[k]]) ==
                                            names(tables[[k]][scores[i]])) + 1]
      if(is.na(next_score)) {
        tmp_delta_home[k] = 0
        tmp_delta_away[k] = 0
      } else {
        if(as.integer(substr(next_score, 1, 1)) > as.integer(substr(scores[i], 1, 1))) {
          tmp_delta_home[k] = 1
          tmp_delta_away[k] = 0
        } else {
          tmp_delta_home[k] = 0
          tmp_delta_away[k] = 1
        }
      }
      tmp_t[k] = tables[[k]][scores[i]]
    } else {
      tmp_delta_home[k] = 0
      tmp_delta_away[k] = 0
      tmp_t[k] = 0
    }
  }
  delta_home[[i]] = tmp_delta_home
  delta_away[[i]] = tmp_delta_away
  t[[i]] = tmp_t
}

rates = NULL; rates_home = NULL; rates_away = NULL
for(i in 1:length(scores)) {
  rates[i] = (sum(delta_home[[i]]) + sum(delta_away[[i]])) / sum(t[[i]])
  rates_home[i] = sum(delta_home[[i]]) / sum(t[[i]])
  rates_away[i] = sum(delta_away[[i]]) / sum(t[[i]])
}

# Crowder pag 66
sd_home = NULL; sd_away = NULL; sd = NULL
for(i in 1:length(scores)) {
  sd[i] = rates[i] / sqrt(sum(delta_home[[i]]) + sum(delta_away[[i]]))
  sd_home[i] = rates_home[i] / sqrt(sum(delta_home[[i]]))
  sd_away[i] = rates_away[i] / sqrt(sum(delta_away[[i]]))
}

tib = tibble(Rate = paste0("$\\nu_{", stringr::str_replace(scores, "-", ""), "}$"),
             `Est.(both)` = rates, `Est.(home)` = rates_home, `Est.(away)` = rates_away,
             `S.e.(both)` = sd, `S.e.(home)` = sd_home, `S.e.(away)` = sd_away)
```

```r
kable(tib, digits = 4, caption = "Estimates and standard errors of the rate of the time
      to the next goal")
```

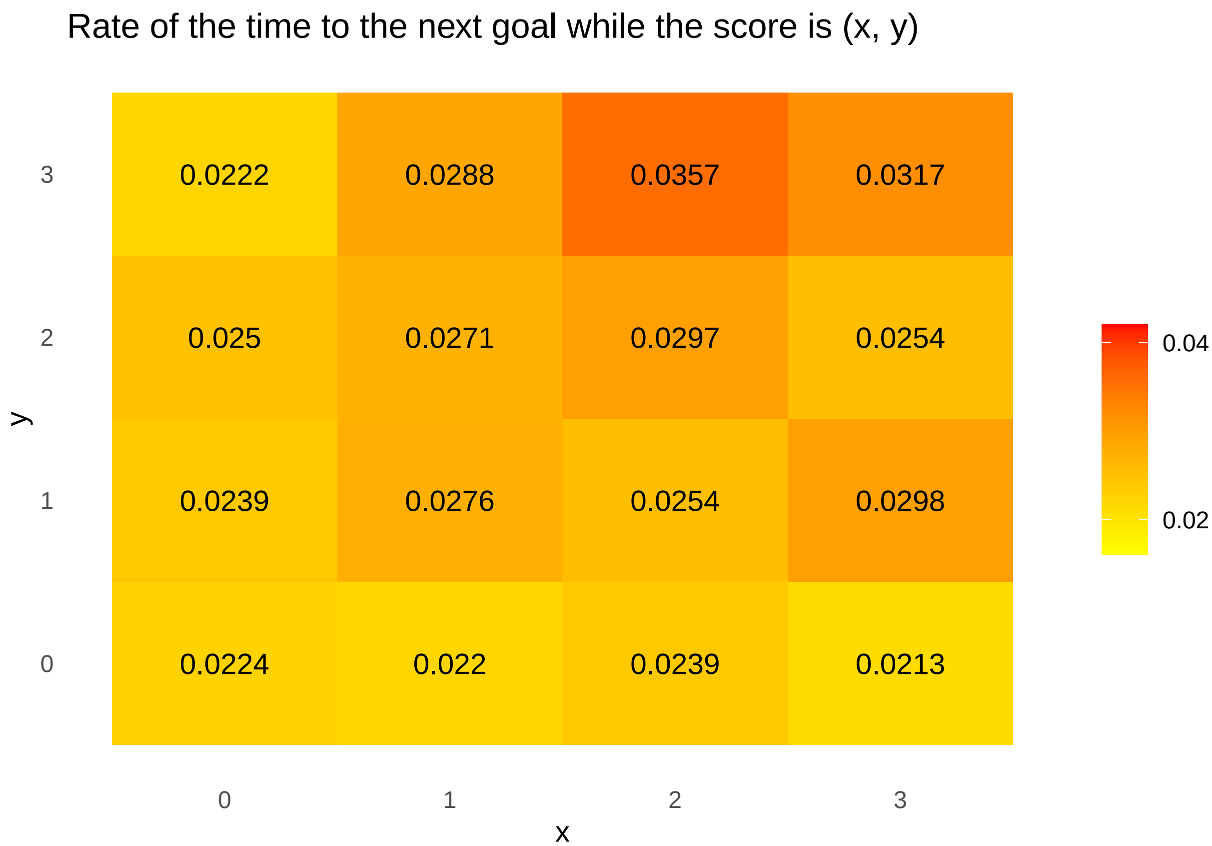
Table 1: Estimates and standard errors of the rate of the time to the next goal

Rate	Est.(both)	Est.(home)	Est.(away)	S.e.(both)	S.e.(home)	S.e.(away)
ν_{00}	0.0224	0.0137	0.0087	0.0005	0.0004	0.0003
ν_{01}	0.0239	0.0151	0.0088	0.0010	0.0008	0.0006
ν_{02}	0.0250	0.0158	0.0092	0.0021	0.0017	0.0013
ν_{03}	0.0222	0.0141	0.0081	0.0047	0.0038	0.0028
ν_{10}	0.0220	0.0128	0.0091	0.0007	0.0006	0.0005
ν_{11}	0.0276	0.0174	0.0102	0.0012	0.0010	0.0008
ν_{12}	0.0271	0.0183	0.0088	0.0023	0.0019	0.0013
ν_{13}	0.0288	0.0144	0.0144	0.0061	0.0043	0.0043
ν_{20}	0.0239	0.0145	0.0094	0.0014	0.0011	0.0008
ν_{21}	0.0254	0.0151	0.0103	0.0017	0.0013	0.0011
ν_{22}	0.0297	0.0182	0.0115	0.0033	0.0026	0.0021
ν_{23}	0.0357	0.0198	0.0159	0.0084	0.0063	0.0056
ν_{30}	0.0213	0.0129	0.0085	0.0023	0.0018	0.0015
ν_{31}	0.0298	0.0168	0.0130	0.0035	0.0027	0.0023
ν_{32}	0.0254	0.0154	0.0100	0.0048	0.0037	0.0030
ν_{33}	0.0317	0.0159	0.0159	0.0112	0.0079	0.0079

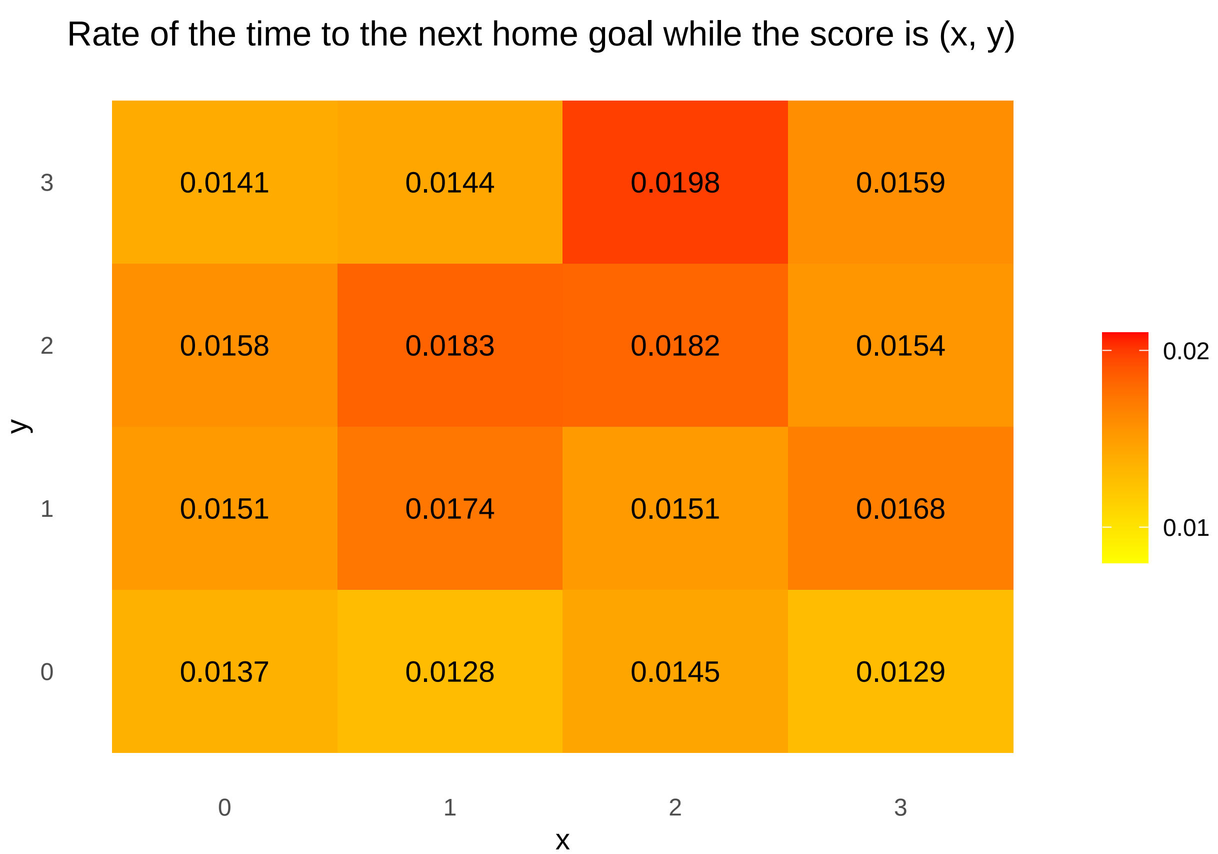
```r
names(rates) = scores
names(rates_home) = scores
names(rates_away) = scores

mat = matrix(NA, nrow = 4, ncol = 4)
rownames(mat) = paste0(0:3)
colnames(mat) = paste0(0:3)
mat_home = mat
mat_away = mat
for(i in 1:4) {
  for(j in 1:4) {
    mat[i,j] = rates[paste(i-1, j-1, sep = "-")]
    mat_home[i,j] = rates_home[paste(i-1, j-1, sep = "-")]
    mat_away[i,j] = rates_away[paste(i-1, j-1, sep = "-")]
  }
}

melted_mat = melt(mat) %>%
  rename(x = Var1, y = Var2)
melted_mat_home = melt(mat_home) %>%
  rename(x = Var1, y = Var2)
melted_mat_away = melt(mat_away) %>%
  rename(x = Var1, y = Var2)
```

```r
ggplot(data = melted_mat, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.029, limit = c(0.016, 0.042),
                       name = "",
                       breaks = c(0.02, 0.04)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
```

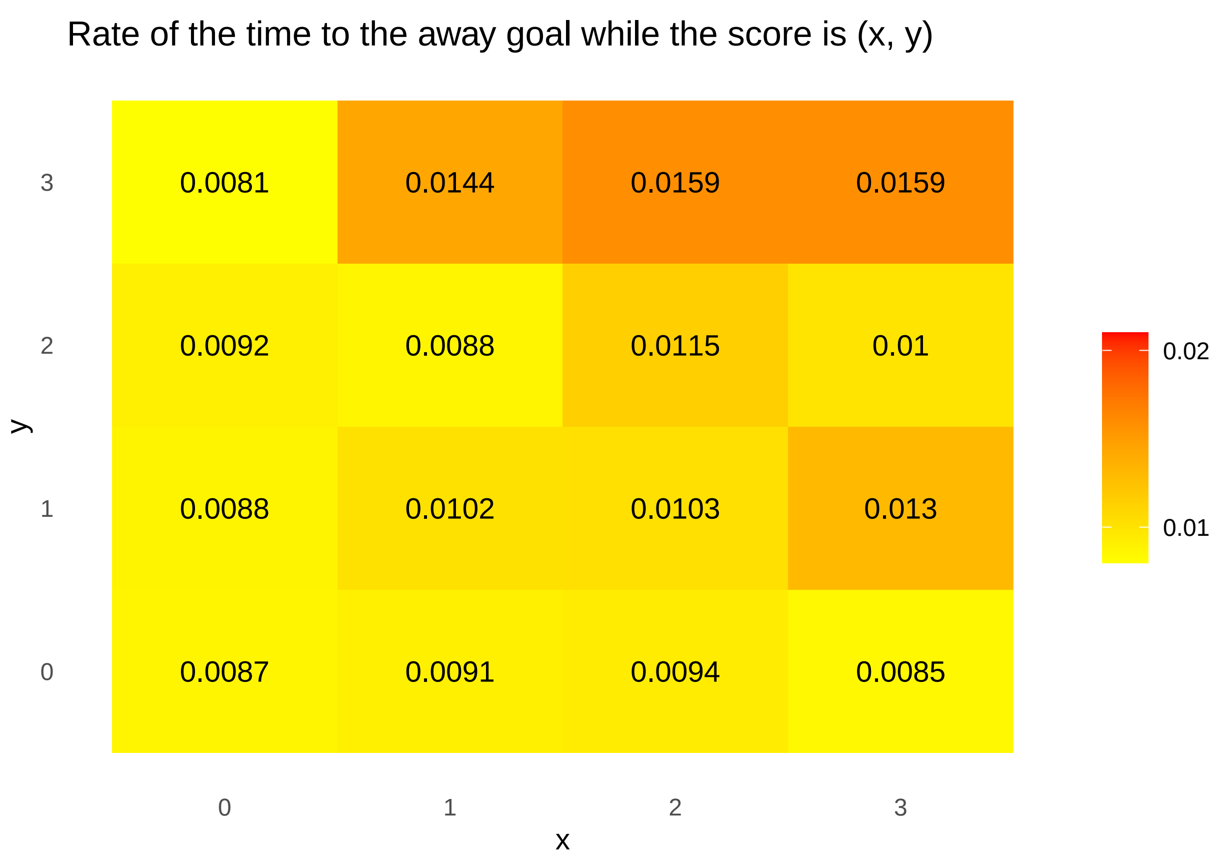
```
theme(panel.grid.major = element_blank(),
      panel.border = element_blank(),
      panel.background = element_blank(),
      axis.ticks = element_blank()) +
ggtitle("Rate of the time to the next goal while the score is (x, y)")
```



```r
ggplot(data = melted_mat_home, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.0145, limit = c(0.008, 0.021),
                       name = "",
                       breaks = c(0.01, 0.02)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
  theme(panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Rate of the time to the next home goal while the score is (x, y)")
```



```r
ggplot(data = melted_mat_away, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.0145, limit = c(0.008, 0.021),
                       name = "",
                       breaks = c(0.01, 0.02)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
  theme(panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Rate of the time to the away goal while the score is (x, y)")
```



```r
ts = NULL; Js = NULL;
for(k in 1:N) {
  ts[[k]] = c(t1s[[k]], t2s[[k]]+45+U1[[k]])
  Js[[k]] = c(J1s[[k]], J2s[[k]])
}

delta_home = list(); delta_away = list(); t = list()

for(i in 1:length(scores)) {
  tmp_delta_home = NULL; tmp_delta_away = NULL; tmp_t = NULL
  for(k in 1:N) {
    if(scores[i] %in% names(tables[[k]])) {
      w = which(xy[[k]] == scores[i])
      start = w[1]
      end = w[length(w)]
      tmp_t[k] = end - start
      tmp_delta_home[k] = sum((ts[[k]] %in% start:end) * !Js[[k]])
      tmp_delta_away[k] = sum((ts[[k]] %in% start:end) * Js[[k]])
    } else {
      tmp_delta_home[k] = 0
      tmp_delta_away[k] = 0
      tmp_t[k] = 0
    }
  }
  delta_home[[i]] = tmp_delta_home
  delta_away[[i]] = tmp_delta_away
  t[[i]] = tmp_t
}

rates = NULL; rates_home = NULL; rates_away = NULL
for(i in 1:length(scores)) {
  rates[i] = (sum(delta_home[[i]]) + sum(delta_away[[i]])) / sum(t[[i]])
  rates_home[i] = sum(delta_home[[i]]) / sum(t[[i]])
  rates_away[i] = sum(delta_away[[i]]) / sum(t[[i]])
}

sd_home = NULL; sd_away = NULL; sd = NULL
for(i in 1:length(scores)) {
  sd[i] = rates[i] / sqrt(sum(delta_home[[i]]) + sum(delta_away[[i]]))
  sd_home[i] = rates_home[i] / sqrt(sum(delta_home[[i]]))
  sd_away[i] = rates_away[i] / sqrt(sum(delta_away[[i]]))
}

rates_away[which(rates_away == 0)] = NA

tib = tibble(Rate = paste0("$\\rho_{", stringr::str_replace(scores, "-", ""), "}$"),
             `Est.(both)` = rates, `Est.(home)` = rates_home, `Est.(away)` = rates_away,
             `S.e.(both)` = sd, `S.e.(home)` = sd_home, `S.e.(away)` = sd_away)
```

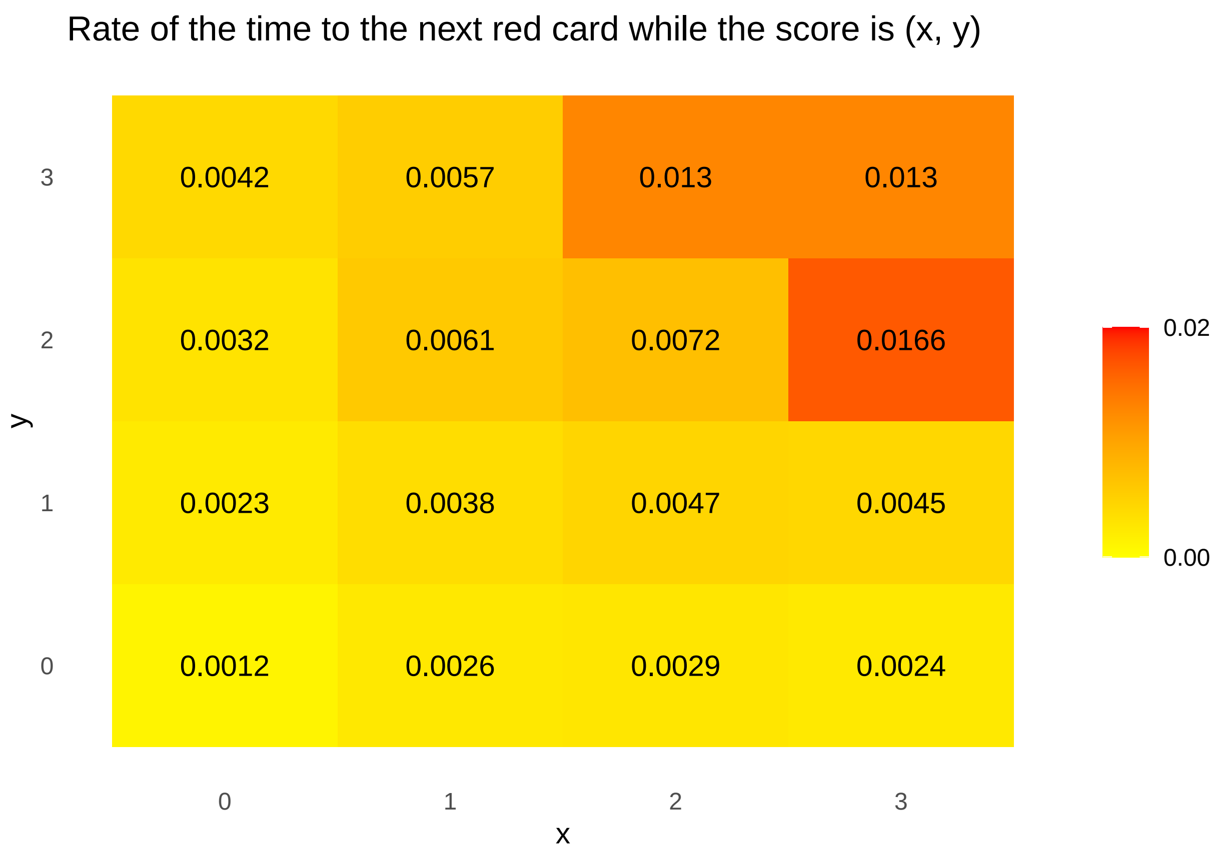
```
kable(tib, digits = 4, caption = "Estimates and standard errors of the rate of the time
to the next goal")
```

Table 2: Estimates and standard errors of the rate of the time to the next goal

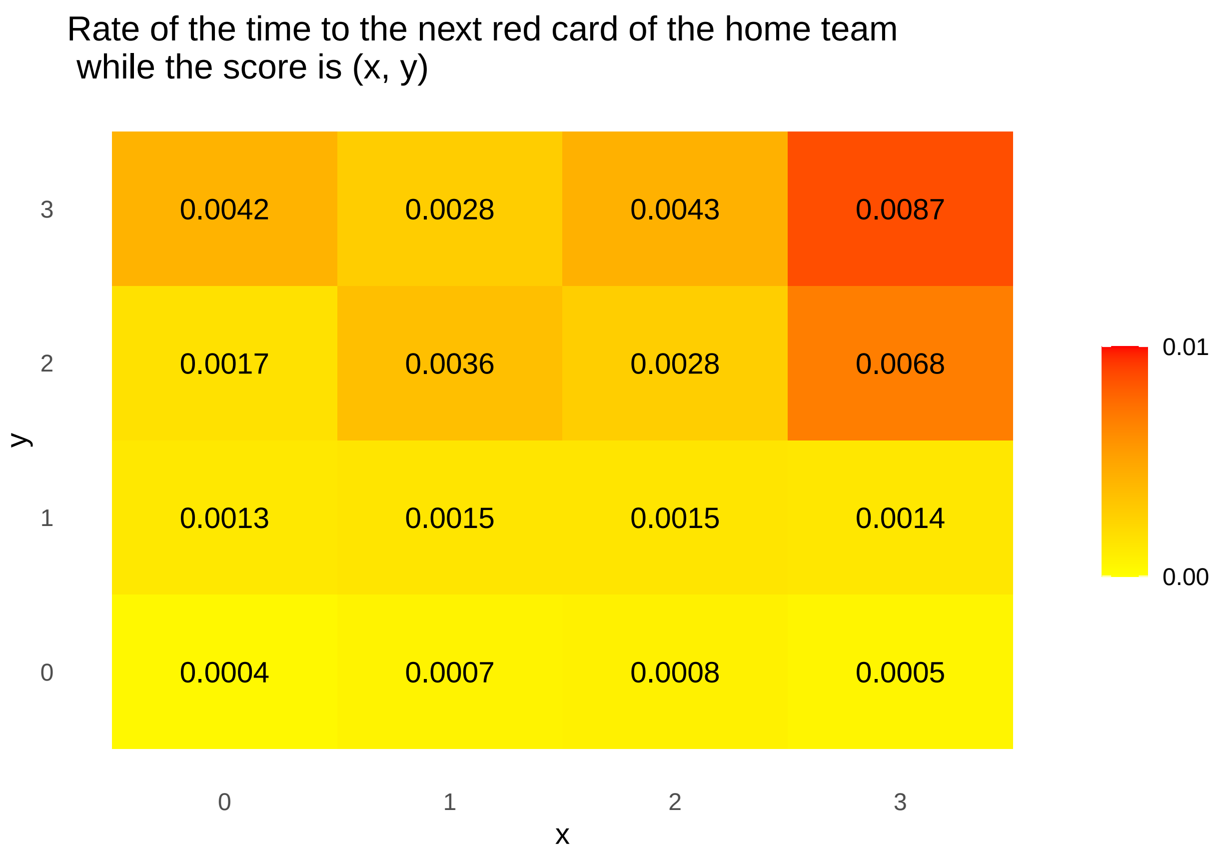
Rate	Est.(both)	Est.(home)	Est.(away)	S.e.(both)	S.e.(home)	S.e.(away)
ρ_{00}	0.0012	0.0004	0.0008	0.0001	0.0001	0.0001
ρ_{01}	0.0023	0.0013	0.0010	0.0003	0.0002	0.0002
ρ_{02}	0.0032	0.0017	0.0015	0.0008	0.0006	0.0005
ρ_{03}	0.0042	0.0042	–	0.0021	0.0021	–
ρ_{10}	0.0026	0.0007	0.0019	0.0003	0.0001	0.0002
ρ_{11}	0.0038	0.0015	0.0023	0.0005	0.0003	0.0004
ρ_{12}	0.0061	0.0036	0.0025	0.0011	0.0009	0.0007
ρ_{13}	0.0057	0.0028	0.0028	0.0028	0.0020	0.0020
ρ_{20}	0.0029	0.0008	0.0021	0.0005	0.0003	0.0004
ρ_{21}	0.0047	0.0015	0.0032	0.0008	0.0004	0.0006
ρ_{22}	0.0072	0.0028	0.0044	0.0017	0.0011	0.0013
ρ_{23}	0.0130	0.0043	0.0087	0.0053	0.0031	0.0043
ρ_{30}	0.0024	0.0005	0.0019	0.0008	0.0004	0.0007
ρ_{31}	0.0045	0.0014	0.0032	0.0014	0.0008	0.0012
ρ_{32}	0.0166	0.0068	0.0098	0.0040	0.0026	0.0031
ρ_{33}	0.0130	0.0087	0.0043	0.0075	0.0061	0.0043

```
names(rates) = scores
names(rates_home) = scores
names(rates_away) = scores

mat = matrix(NA, nrow = 4, ncol = 4)
rownames(mat) = paste0(0:3)
colnames(mat) = paste0(0:3)
mat_home = mat
mat_away = mat
for(i in 1:4) {
  for(j in 1:4) {
    mat[i,j] = rates[paste(i-1, j-1, sep = "-")]
    mat_home[i,j] = rates_home[paste(i-1, j-1, sep = "-")]
    mat_away[i,j] = rates_away[paste(i-1, j-1, sep = "-")]
  }
}

melted_mat = melt(mat) %>%
  rename(x = Var1, y = Var2)
melted_mat_home = melt(mat_home) %>%
  rename(x = Var1, y = Var2)
melted_mat_away = melt(mat_away) %>%
  rename(x = Var1, y = Var2)
```

```r
ggplot(data = melted_mat, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.01, limit = c(0, 0.02),
                       name = "",
                       breaks = c(0, 0.02)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
  theme(panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Rate of the time to the next red card while the score is (x, y)")
```



```r
ggplot(data = melted_mat_home, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.005, limit = c(0, 0.01),
                       name = "",
                       breaks = c(0, 0.01)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
  theme(panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Rate of the time to the next red card of the home team \n while the score is (x, y)")
```



```r
ggplot(data = melted_mat_away, aes(x, y, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "yellow", mid = "orange", high = "red",
                       midpoint = 0.005, limit = c(0, 0.01),
                       name = "",
                       breaks = c(0, 0.01)) +
  geom_text(aes(x, y, label = round(value, 4)), color = "black") +
  theme(panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Rate of the time to the next red card of the away team \n while the score is (x, y)")
```