Soccer Penalty Kicks Analysis

Colin Yip, Dhruv Krishnan, Eunice Yang, Mudita Sai

04-30-2024

Introduction

In the 2014 World Cup, Netherlands manager Louis Van Gaal substituted goalkeeper Jasper Cilessen for Tim Krul, believing that Krul was the better penalty saver. Krul saved two out of five penalties, and the Netherlands won the penalty shootout. Deciding which goalkeeper to see for a penalty shootout can make the difference between a team winning or losing a knockout match. Additionally, even in normal league, play managers and front office personnel want to know how good their goalkeeper is at saving penalties as part of evaluation of their goalkeeper. In this report, we attempt to determine the penalty-saving ability of goalkeepers in the German Bundesliga, from 1963 to 2017.

Data

We obtained this dataset from Github. To access it, we downloaded the "footballpenalitiesBL" package. This dataset conducts basic analysis of all penalties taken in the German men's Bundesliga, which is a professional association football league in Germany, between the start of its inaugural season in 1963 and May 2017.

We will be using the following variables in the dataset as predictors:

goalkeeper: Character vector containing the name of the goalkeeper the penalty was taken against

penaltytaker: Character vector containing the name of the player who took the penalty

homegame: Numeric vector that specifies where the match was played, 1 indicates an away match for the goalkeeper, 2 a home match.

minute: Numeric vector specifying the minute of the match the penalty was taken in.

goaldiff: Numeric vector that gives the goal difference before the penalty was taken. A positive number indicates that the goalkeeper's club is in the lead.

gkage: Numeric vector giving the goalkeeper's age in years at the time the penalty was given

gkexp: Numeric vector giving the goalkeeper's experience, measured in number of seasons. 0 stands for the debut season, from then on, 1 is added for every following season, regardless of whether the player played in the Bundesliga or not.

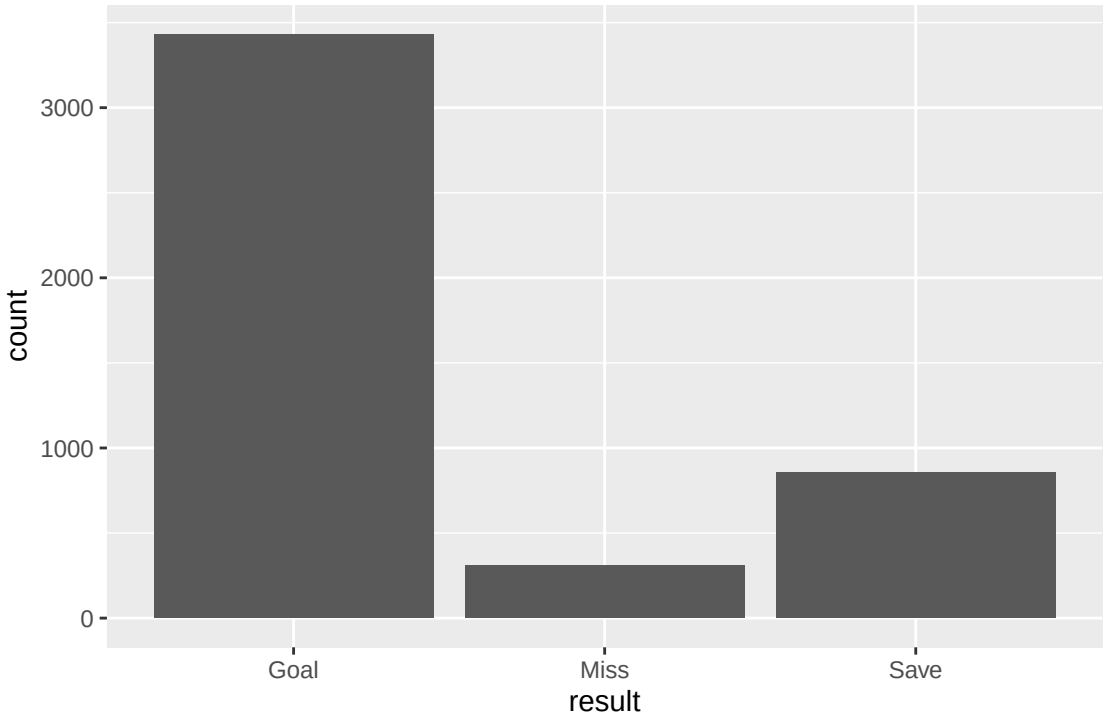
The response variable of interest is:

result: Character vector that gives the result of the penalty in German. Possible values are: - Tor: Goal - gehalten: Save - vorbei: Miss - drüber: Miss, too high - Latte: Miss, hit the crossbar - Pfosten: Miss, hit the post

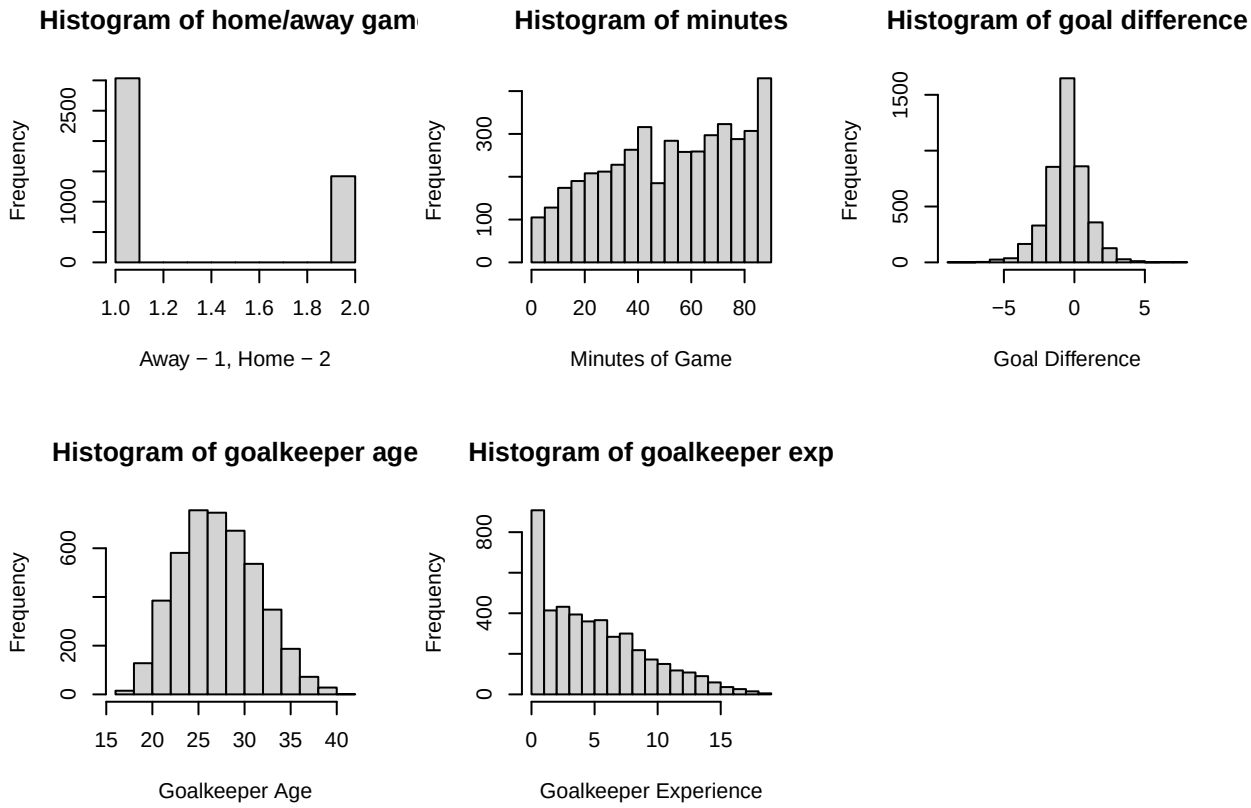
The data was pre-processed to transform the German values for `result`, such as tor, to English to make it more comprehensible and only included penalty shots that were either saved or resulted in a goal (`result` is a binary variable of 1 indicating a goal and 0 indicating a save). Apart than that, the dataset was very

straightforward to use. The dataset is also of good quality since it does not have any missing values, and has a fairly reasonable sample size to work with. It is also relevant to our problem since it includes variables we can use for predictors and response variable.

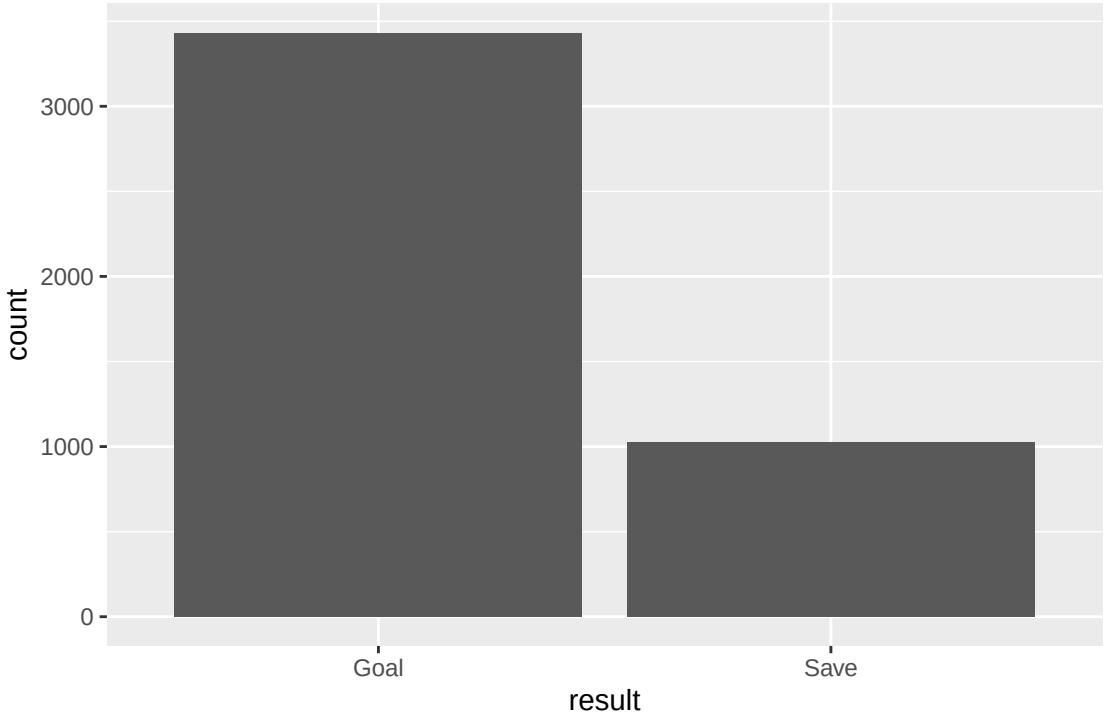
EDA



Before the pre-processing step, the most common outcome of a penalty is a goal, followed by save, and finally miss. There are roughly 3400 goals, 800 saves, and 250 misses. There are 4599 penalties in the dataset.



From these visualizations, we can see that most games in the data set are away. Penalty kicks happens most frequently at the 45th minute and 90th minute, which are near the end of either the half game or whole game. Penalty kicks occur most frequently when the penalty kicker team is losing. As for the goalkeeper attributes, the mean goalkeeper age is 27.7 years with the minimum being 17 years and the maximum being 42 years. The goalkeeper experience is skewed right with the median experience being 5.4 (this means the median goalkeeper has played for at least 5 seasons) and most goalkeepers experience usually being their debut season. Only a handful of goalkeepers have an experience of more than 6 seasons.



Much of the post-processed data has more goals than saves. The EDA section is useful to the problem we are focusing on, which examines the players' internal and external factors that affect a penalty kick's success rate. Each graph shows insightful trends, and there are no outliers nor clusters. This provides us valuable context for analyzing. For instance, the histogram of minutes shows that penalty kicks occur near the end of half game and whole game most frequently. Also, since the most common outcome of a penalty is a goal in this data set, we can assume that as the minute of game increases, the success of penalty is higher. We can then test this theory in the modeling part.

Methods

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: factor(result) ~ 1 + (1 | goalkeeper) + (1 | penaltytaker)
##    Data: penalties
##
##      AIC      BIC   logLik deviance df.resid
##   4792.4   4811.6  -2393.2   4786.4     4452
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3597  0.4412  0.5083  0.5461  0.7519
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
```

```
##  penaltytaker (Intercept) 0.10816  0.3289
##  goalkeeper   (Intercept) 0.07515  0.2741
## Number of obs: 4455, groups:  penaltytaker, 894; goalkeeper, 352
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.21697    0.04682   25.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We build a basic multi-level model that makes use of a fixed intercept effect which is the baseline probability of our result (save vs goal) when goalkeeper and penalty-taker are at their reference levels. We also assume random non-nested intercepts for both goalkeepers and penalty-takers. The output summary indicates that the fixed effect intercept estimate of 1.21697 is highly significant (p < 2e-16), suggesting an overall non-zero baseline probability for the response. The random effects variances indicate that there is considerable variability in the baseline probabilities across different goalkeepers and penalty-takers.

Now we build a more comprehensive multi-level model that like the earlier model assumes that the two groups of interest: goalkeeper and penalty-taker are not nested within each other hence the (1|goalkeeper) + (1|penaltytaker). Additionally, we use the variables homegame, minute, goaldiff, gkage and gkexp in a non-interactive manner (simply additive). That is, these predictors don't interact with the groups nor with each other. Since we only used a dataset that is filtered down to two response values, the family of models we use is binomial.

Response: $Y \in \{Goal(1)\ ,\ Save(0)\}$ which is the variable result where the distribution is as follows:

$$Y_{gpi} \sim Bernoulli(p_{gpi}) \implies \log\left(\frac{p_{gpi}}{1-p_{gpi}}\right)$$

$$\text{where } g = \{1, \ldots, G\}\ G \text{ is the number of goalkeepers}$$

$$\text{and } p = \{1, \ldots, P\}\ P \text{ is the number of penalty-takers}$$

Level 1:

$$\log\left(\frac{p_{gpi}}{1-p_{gpi}}\right) = a_{gp} + \beta_0 \text{homegame}_i + \beta_1 \text{minute}_i + \beta_2 \text{goaldiff}_i + \beta_3 \text{gkage}_i + \beta_2 \text{gkexp}_i$$

Level 2:

$$a_{gp} = \alpha_0 + u_g + v_p$$

$$\text{where } u_g \sim \mathcal{N}(0, \sigma_u^2),\ v_p \sim \mathcal{N}(0, \sigma_v^2)$$

Composite Model:

$$\log\left(\frac{p_{gpi}}{1-p_{gpi}}\right) = (u_g + v_p) + \alpha_0 + \beta_0 \text{homegame}_i + \beta_1 \text{minute}_i + \beta_2 \text{goaldiff}_i + \beta_3 \text{gkage}_i + \beta_2 \text{gkexp}_i$$

Fixed effects are given by: $(\alpha_0 + \beta_0 \text{homegame}_i + \beta_1 \text{minute}_i + \beta_2 \text{goaldiff}_i + \beta_3 \text{gkage}_i + \beta_2 \text{gkexp}_i)$

Random effects are given by: $(u_g + v_p)$

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: factor(result) ~ (1 | goalkeeper) + (1 | penaltytaker) + homegame +
```

```
##     minute + goaldiff + gkage + gkexp
##    Data: penalties
##
##      AIC      BIC   logLik deviance df.resid
##   4790.8   4842.1  -2387.4   4774.8     4447
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4472  0.4313  0.5045  0.5506  0.7387
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  penaltytaker (Intercept) 0.10322  0.3213
##  goalkeeper   (Intercept) 0.06206  0.2491
## Number of obs: 4455, groups:  penaltytaker, 894; goalkeeper, 352
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.7509359  0.3451135   5.074 3.91e-07 ***
## homegame     0.1200141  0.0822916   1.458   0.1447
## minute      -0.0009909  0.0014776  -0.671   0.5025
## goaldiff    -0.0369426  0.0259908  -1.421   0.1552
## gkage       -0.0303070  0.0130178  -2.328   0.0199 *
## gkexp        0.0381754  0.0133690   2.856   0.0043 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) homegm minute goldff gkage
## homegame -0.315
## minute   -0.210 -0.054
## goaldiff  0.033 -0.262  0.064
## gkage    -0.906  0.016 -0.004  0.043
## gkexp     0.564 -0.016  0.021 -0.044 -0.726
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0178835 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

From the model output the fixed effects estimates show that homegame, minute, and goaldiff are not statistically significant (p-values > 0.05), while gkage and gkexp are marginally significant (p-values < 0.05 and < 0.01, respectively). Also the correlation between minute and goaldiff is 0.064, indicating a relatively weak positive correlation. However, the correlation between gkage and gkexp is -0.726, which is a strong negative correlation implying some multicollinearity taking effect. This could be a problem as multicollinearity can inflate the standard errors of our predictors, leading to unreliable statistical inferences. For future modeling we can use either gkage or gkexp but not both.

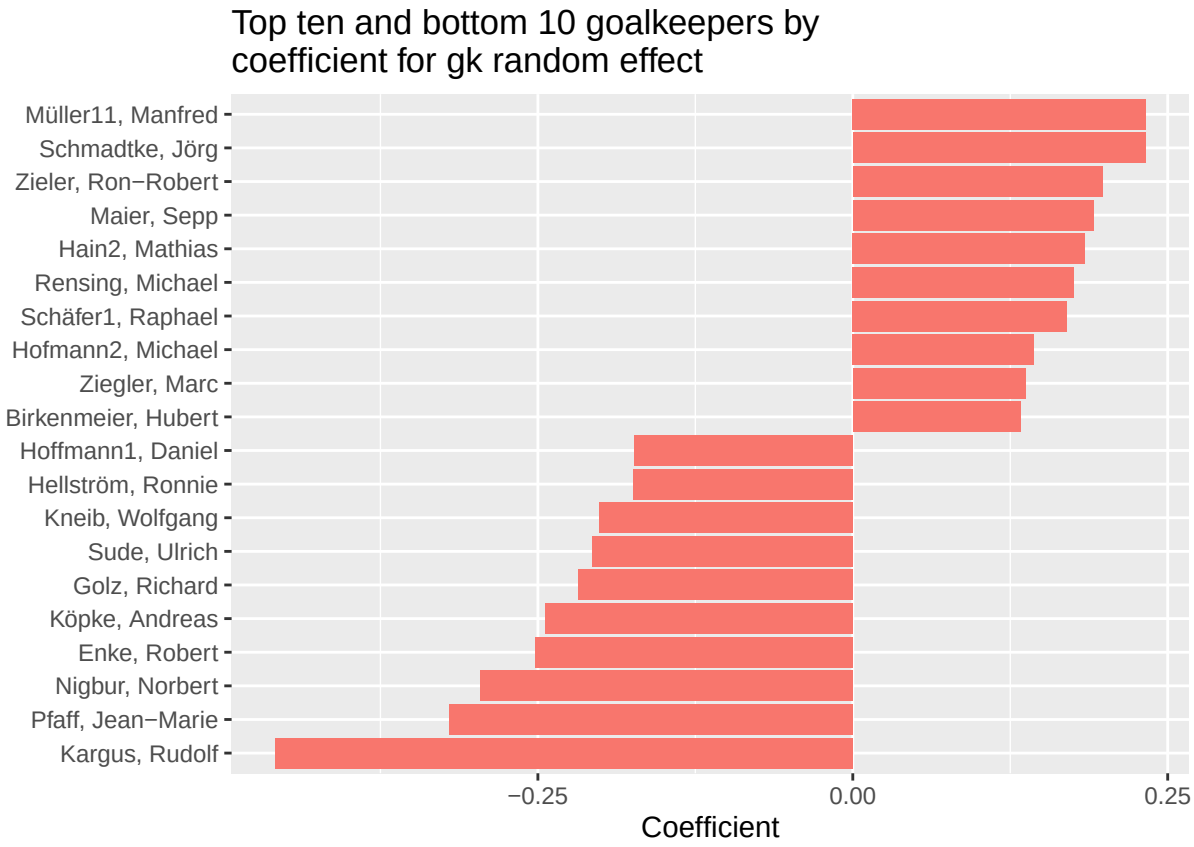
To quantify the uncertainity of our estimates, we use a 5-fold cross validation and compare the Brier scores of both the baseline and the main model. The data is grouped in folds based on the seasons. Ideally, the data would be grouped by some sort of play-by-play metric like match id but since we don't have access to this information, we opted for a season-by-season approach. Additionally, we divide the training data to include penalties before 2017 and reserve the testing data for penalties occurring in 2017.

After performing 5-fold CV there really isn't evidence that either of these models outperforms the other. The average Brier score is 0.1772012 +/- 0.003761519 for the simple baseline model and the average Brier

score is 0.1771643 +/- 0.003997966 for the multi-level crossed-effects model. There isn't much of a difference in the avg. Brier score for these two models and the standard error is the nearly the same as well.

Also, for evaluating on the 2017 penalty kicks, the Brier score is similar for both the baseline and the main model with additional goalkeeper attributes, with the later performing just slightly better which is given by the lower brier score (0.145432). Essentially, both models are able to predict the 2017 penalty kicks with similar accuracy. For the following sections, we will use the main model that includes additional goalkeeper attributes to assess goalkeeper performance because it performs slightly better and has a lower standard error.

Results



These estimates of coefficients align with the Bill James 80-20 guideline (when creating a new statistic to evaluate players, roughly eighty percent of the players with the highest metric should be expected, and roughly 20% should be a surprise). The goalkeeper with the highest coefficient is Rudolf Kargus, who earned the nickname Elfmetertöter (penalty killer) for his penalty-saving ability. Eight of the top ten goalkeepers by coefficient played for their national team, suggesting that they were generally high quality goalkeepers. One of the two who did not (Ulrich Sude) had a highly successful club career, winning the German Bundesliga and being runner up in the European Cup. By the Bill James guideline, our metric seems reasonable.

However, we cannot be too certain that our rankings reflect a goalkeeper's true penalty-saving ability due to large standard errors. The confidence intervals for coefficients of the top ten goalkeepers and bottom ten goalkeepers overlap, indicating that these coefficients are not statistically significantly different. Thus, based on our current model we should use caution when claiming that our top ten are among the best penalty savers in the Bundesliga.

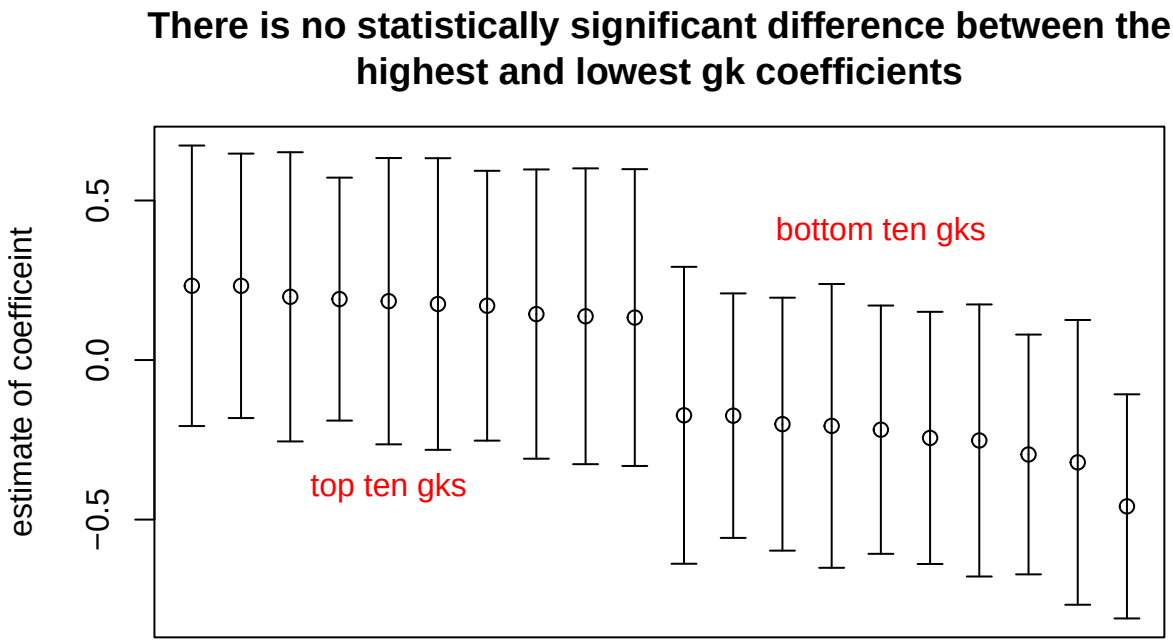


Figure 1: 95% confidence intervals for the coefficients of the top ten goalkeepers and bottom ten goalkeepers

Discusssion

We claim that the following goalkeepers are the ten best penalty savers in the German Bundesliga from 1963 to 2017

1. Rudolf Kargus
2. Norbert Nigbur
3. Pfaff Jean-Marie
4. Richard Golz
5. Oka Nikolov
6. Andreas Kopke
7. Robert Enke
8. Peter Radenkovic
9. Ulrich Sude
10. Diego Benaglio

However, our estimates are highly uncertain so we are not confident that these are actually the ten best penalty savers in the Bundesliga. The high uncertainty of our estimates is one limitation of the analysis. Another limitation is small sample sizes (some goalkeepers face only a few penalties in their career).

Possible future work includes comparing more models. For example, a Bayesian model may mitigate the small sample size limitation. We could also attempt to estimate the ability of penalty takers. To do this, we could combine two models: one to predict whether a penalty is on target and another to predict the chance a penalty is saved, given it is on target.

Code Appendix

```
sum(is.na(penalties))
nrow(penalties)
```

```
penalties %>%
  mutate(result = case_when(
    result == "Tor" ~ "Goal",
    result == "gehalten" ~ "Save",
    TRUE ~ "Miss"
  )) %>%
  ggplot(aes(result)) +
  geom_bar() +
  theme(plot.margin = margin(1, 1, 1, 1, "cm"))
```

```
#pre-processing steps
penalties = penalties %>%
  mutate(result = case_when(
    result == "Tor" ~ "Goal",
    result == "gehalten" ~ "Save",
    result == "vorbei" ~ "Miss",
    result == "drüber" ~ "Miss, too high",
    result == "Latte" ~ "Miss, hit the crossbar",
    result == "Pfosten" ~ "Miss, hit the post",
    TRUE ~ result
```

```
  ))

penalties = penalties %>%
  filter(result != "Miss") %>%
  mutate(result = ifelse(result=="Goal",1,0))
```

```
par(mfrow = c(2,3))
hist(penalties$homegame,
     main = "Histogram of home/away game",
     xlab = "Away - 1, Home - 2")
hist(penalties$minute,
     main = "Histogram of minutes",
     xlab = "Minutes of Game")
hist(penalties$goaldiff,
     main = "Histogram of goal difference",
     xlab = "Goal Difference")
hist(penalties$gkage,
     main = "Histogram of goalkeeper age",
     xlab = "Goalkeeper Age")
hist(penalties$gkexp,
     main = "Histogram of goalkeeper exp",
     xlab = "Goalkeeper Experience")

# summary(penalties$gkage)
# summary(penalties$gkexp)
```

```
penalties %>%
  mutate(result = case_when(result==1~"Goal",
                            result==0~"Save")) %>%
  ggplot(aes(result)) +
  geom_bar() +
  theme(plot.margin = margin(1, 1, 1, 1, "cm"))
```

```
#baseline model
baseline_model <- glmer(factor(result) ~ 1 + (1|goalkeeper) + (1|penaltytaker),
                family="binomial",data=penalties)
summary(baseline_model)
```

```
#fit a multilevel model
model <- suppressWarnings(glmer(factor(result) ~ (1|goalkeeper) +
                (1|penaltytaker) + homegame + minute + goaldiff + gkage + gkexp,
                family="binomial",data=penalties))
summary(model)
```

```
penalties_past = penalties %>% filter(!str_detect(date, "2017"))
penalties_2017 = penalties %>% filter(str_detect(date, "2017"))

set.seed(2022)
season_fold_table <- tibble(season = unique(penalties_past$season)) %>%
  mutate(season_fold = sample(rep(1:5, length.out = n()), n()))

# See how many games are in each fold:
```

```r
# season_fold_table
# table(season_fold_table$season_fold)

penalties_past <- penalties_past %>%
  left_join(season_fold_table, by = "season")
# table(penalties_past$season_fold)
```

```r
Brier = c()
for (test_fold in 1:5) {
      # Separate test and training data:
      test_data <- penalties_past %>% filter(season_fold == test_fold)
      train_data <- penalties_past %>% filter(season_fold != test_fold)

      # Train model:
      baseline_model <-
        glmer(factor(result) ~ 1 + (1|goalkeeper) + (1|penaltytaker),
              family="binomial",data=train_data)

      # Predict on test data:
      pred_y = predict(baseline_model, newdata = test_data, type = "response",
                       allow.new.levels = TRUE)
      test_y = test_data$result

      Brier = append(Brier, mean((test_y - pred_y)**2))
}

avg_Brier = mean(Brier)
se_Brier = sd(Brier) / sqrt(5)

cat("Average Brier score: ", avg_Brier, "+/-", se_Brier)
```

```r
Brier = c()
for (test_fold in 1:5) {
      # Separate test and training data:
      test_data <- penalties_past %>% filter(season_fold == test_fold)
      train_data <- penalties_past %>% filter(season_fold != test_fold)

      complex_model <- suppressWarnings(glmer(factor(result) ~
                  (1|goalkeeper) + (1|penaltytaker) + homegame +
                minute + goaldiff + gkage + gkexp,
              family="binomial",data=train_data))

      # Predict on test data:
      pred_y = predict(complex_model, newdata = test_data, type = "response",
                       allow.new.levels = TRUE)
      test_y = test_data$result
      Brier = append(Brier, mean((test_y - pred_y)**2))
}

avg_Brier = mean(Brier)
se_Brier = sd(Brier) / sqrt(5)

cat("Average Brier score: ", avg_Brier, "+/-", se_Brier)
```

```
baseline_model <- glmer(factor(result) ~ 1 + (1|goalkeeper) + (1|penaltytaker),
               family="binomial", data=penalties_past)

complex_model <- suppressWarnings(glmer(factor(result) ~ (1|goalkeeper) +
                (1|penaltytaker) + homegame + minute + goaldiff + gkage + gkexp,
               family="binomial", data=penalties_past))


pred_y_base = predict(baseline_model, newdata = penalties_2017,
                      type = "response", allow.new.levels = TRUE)
pred_y_main = predict(complex_model, newdata = penalties_2017,
                      type = "response", allow.new.levels = TRUE)
actual_y = penalties_2017$result

brier_base = mean((actual_y - pred_y_base)**2)
brier_main = mean((actual_y - pred_y_main)**2)

cat("Brier score using basic model: ", brier_base, "\n")
cat("Brier score using main model: ", brier_main)
```

```
coefs <- tidy(model, effects="ran_vals", conf.int=TRUE)
# coefs %>% filter(group == "goalkeeper") %>% arrange(-estimate)

top_gks <- coefs %>%
  filter(group == "goalkeeper") %>% arrange(-estimate) %>%
  select(level, estimate, "std.error") %>%
  head(10)

pdf("data_output.pdf", height=11, width=8.5)
grid.table(top_gks)
dev.off()
```

```
# colnames(coefs)
top_10 <- coefs %>% filter(group=="goalkeeper") %>% arrange(-estimate) %>% head(10)
bottom_10 <- coefs %>% filter(group=="goalkeeper") %>% arrange(-estimate) %>% tail(10)
top_and_bottom <- rbind(top_10, bottom_10)

top_and_bottom %>%
  ggplot(aes(x=reorder(level,estimate),y=estimate)) +
  geom_col(aes(fill="orange")) + coord_flip() +
  ggtitle("Top ten and bottom 10 goalkeepers by\ncoefficient for gk random effect") +
  ylab("Coefficient") + xlab("") +theme(legend.position="none")
```

```
plotCI(x = 1:20,y = top_and_bottom$estimate,
       li = top_and_bottom$conf.low,ui = top_and_bottom$conf.high,
       main="There is no statistically significant difference
       between the\nhighest and lowest gk coefficients",
       ylab="estimate of coefficeint",xlab="",xaxt="n")
text(x=5,y=-0.4,"top ten gks",col="red")
text(x=15,y=0.4,"bottom ten gks",col="red")
```