Goalie Analysis

Ryan Dajani

2024-08-23

#Loading Libraries

```
# Ensure all required libraries are loaded
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ---------------------- tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.5.1     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- 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(hockeyR)
library(rvest)
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.3.3
```

```
library(readxl)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.3.2
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(anytime)
```

```
## Warning: package 'anytime' was built under R version 4.3.3
```

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.3.3
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

##
## Attaching package: 'car'
##
## The following object is masked from 'package:purrr':
##
##     some
##
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(lattice)
library(ggplot2)
library(gt)
```

```
## Warning: package 'gt' was built under R version 4.3.3
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.3.3

##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
```

```
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

#Load data

```
data <- load_pbp('2023-2024')
```

#Filtering Data

```
all_games <- unique(data$game_id)


# Filter data for shots and goals that exclude
# empty net goals and empty net shot attempts
shot_goal_data <- data %>%
  filter(event_type %in% c("SHOT", "GOAL")) %>%
  filter(extra_attacker %in% c("FALSE"))

# Get unique goalie names
unique_goalie_names <- data %>%
  distinct(event_goalie_name) %>%
  filter(!is.na(event_goalie_name))
```

#General Statistics

```
#function to get the number of games that a goalie played
goalie_games_played <- function(goalie_name){
  goalie_data <- data %>%
    filter(event_goalie_name == goalie_name)
  goalie_data <- unique(goalie_data$game_id)

  return(length(goalie_data))
}
#function that gets the distance of each shot that a goalie has faced
goalie_shot_distance <- function(goalie_name) {
  shot_distance_vector <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name) %>%
    pull(shot_distance)
  return(shot_distance_vector)
}

#function to get the number of shots that a goaltender faced
goalie_shots_against <- function(goalie_name) {
  shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name) %>%
    nrow()
```

```
  return(shots)
}

#function to get the number of goals that a goaltender allowed
#in the 2023-2024 season
goalie_goals_allowed <- function(goalie_name) {
  goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, event_type == "GOAL") %>%
    nrow()
  return(goals)
}

#function to calculate the save percentage of a goaltender
goalie_save_percent <- function(goalie_name) {
  shots <- goalie_shots_against(goalie_name)
  goals <- goalie_goals_allowed(goalie_name)

  if (shots == 0) return(NA)

  save_percentage <- (shots - goals) / shots
  return(round(save_percentage, 3))
}
```

#Penalty Kill Stats

```
#function to calculate the number of shots that a goaltender who's
#team is on the penalty kill faced
goalie_pk_shots_against <- function(goalie_name) {
  pk_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, strength_code == "PP", event_type == "SHOT") %>%
    nrow()
  pk_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, strength_code == "PP", event_type == "GOAL") %>%
    nrow()

  return(pk_shots + pk_goals)
}

#function that returns the number of goals that a goaltender allowed
#while their team was on the penalty kill
goalie_pk_goals_against <- function(goalie_name) {
  pk_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, strength_code == "PP", event_type == "GOAL") %>%
    nrow()
  return(pk_goals)
}

#function to get the save percentage of a goaltender who's team is on the penalty kill
goalie_pk_save_percent <- function(goalie_name) {
  pk_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, strength_code == "PP", event_type == "SHOT") %>%
    nrow()
  pk_goals <- shot_goal_data %>%
```

```
    filter(event_goalie_name == goalie_name, strength_code == "PP", event_type == "GOAL") %>%
    nrow()

  if (pk_shots == 0) return(NA)

  pk_save_percentage <- (pk_shots - pk_goals) / pk_shots
  return(round(pk_save_percentage, digits = 3))
}
```

#High Danger Statistics (shots being <= 29ft away from opposing teams' net)

```
#function that calculates the number of high danger shots that a goalie has faced
goalie_high_danger_shots_against <- function(goalie_name) {
  high_danger_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance <= 29, event_type == "SHOT") %>%
    nrow()
  high_danger_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance <= 29, event_type == "GOAL") %>%
    nrow()

  return(high_danger_shots + high_danger_goals)
}
#function that calculates the number of high danger shots that a goalie has allowed
goalie_high_danger_goals_against <- function(goalie_name) {
  high_danger_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance <= 29, event_type == "GOAL") %>%
    nrow()
  return(high_danger_goals)
}

#function that calculates the high danger save percentage for a goalie.
goalie_high_danger_save_pct <- function(goalie_name) {
  high_danger_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance <= 29, event_type == "SHOT") %>%
    nrow()
  high_danger_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance <= 29, event_type == "GOAL") %>%
    nrow()

  if (high_danger_shots == 0) return(NA)

  high_danger_save_pct <- (high_danger_shots - high_danger_goals) / high_danger_shots
  return(round(high_danger_save_pct, digits = 3))
}
```

#Mid Range Statistics (shot being >= 29ft and <= 43 ft away from the opposing teams' net)

```
#function that calculates the number of shots that a goaltender has faced from mid range shots
goalie_mid_range_shots_against <- function(goalie_name) {
  mid_range_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 29 & shot_distance <= 43, event_type == "
    nrow()
  mid_range_goals <- shot_goal_data %>%
```

```
    filter(event_goalie_name == goalie_name, shot_distance >= 29 & shot_distance <= 43, event_type == "G
    nrow()

  return(mid_range_shots + mid_range_goals)
}
#function that calculates the number of goals that a goaltender has allowed from mid range shots
goalie_mid_range_goals_against <- function(goalie_name) {
  mid_range_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 29 & shot_distance <= 43, event_type == "G
    nrow()
  return(mid_range_goals)
}
#function that calculates the save percentage for a goaltender from mid range shots and goals
goalie_mid_range_save_pct <- function(goalie_name) {
  mid_range_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 29 & shot_distance <= 43, event_type == "S
    nrow()
  mid_range_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 29 & shot_distance <= 43, event_type == "G
    nrow()

  if (mid_range_shots == 0) return(NA)

  mid_range_save_pct <- (mid_range_shots - mid_range_goals) / mid_range_shots

  return(round(mid_range_save_pct, digits = 3))
}
```

#Long Range Shots (shot being >= 43 ft away while shot is still located within the offensive zone)

```
#function that calculates the number of shots that a goaltender has faced from long range shots
goalie_long_range_shots_against <- function(goalie_name) {
  long_range_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 43 & shot_distance <= 64, event_type == "S
    nrow()
  long_range_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 43, event_type == "GOAL") %>%
    nrow()

  return(long_range_shots + long_range_goals)
}
#function that calculates the number of goals that a goaltender has allowed from long range shots
goalie_long_range_goals_against <- function(goalie_name) {
  long_range_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 43, event_type == "GOAL") %>%
    nrow()
  return(long_range_goals)
}
#function that calculates the save percentage of goaltender from long range shots and goals
goalie_long_range_save_pct <- function(goalie_name) {
  long_range_shots <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 43, event_type == "SHOT") %>%
    nrow()
```

```
  long_range_goals <- shot_goal_data %>%
    filter(event_goalie_name == goalie_name, shot_distance >= 43, event_type == "GOAL") %>%
    nrow()

  if (long_range_shots == 0) return(NA)

  long_range_save_pct <- (long_range_shots - long_range_goals) / long_range_shots
  return(round(long_range_save_pct, digits = 3))
}
```

#For loop and Dataframe creation

```
# Initialize vectors for storing data
goalie_name_vector <- c()
goalie_games_played_vector <- c()
goalie_shot_distance_vector <- c()
goalie_sa_vector <- c()
goalie_goals_allowed_vector <- c()
goalie_save_pct_vector <- c()

goalie_pk_shots_against_vector <- c()
goalie_pk_goals_allowed_vector <- c()
goalie_pk_save_pct_vector <- c()

goalie_high_danger_shots_against_vector <- c()
goalie_high_danger_goals_allowed_vector <- c()
goalie_high_danger_save_pct_vector <- c()

goalie_mid_range_shots_against_vector <- c()
goalie_mid_range_goals_allowed_vector <- c()
goalie_mid_range_save_pct_vector <- c()

goalie_long_range_shots_against_vector <- c()
goalie_long_range_goals_allowed_vector <- c()
goalie_long_range_save_pct_vector <- c()

# Process each goalie
for (i in 1:length(unique_goalie_names$event_goalie_name)) {
  tendy_name <- unique_goalie_names$event_goalie_name[i]

  shots_against <- goalie_shots_against(tendy_name)
  games_played <- goalie_games_played(tendy_name)
  shot_distance <- goalie_shot_distance(tendy_name)
  goals_allowed <- goalie_goals_allowed(tendy_name)
  save_pct <- goalie_save_percent(tendy_name)

  pk_shots_against <- goalie_pk_shots_against(tendy_name)
  pk_save_pct <- goalie_pk_save_percent(tendy_name)
  pk_goals_allowed <- goalie_pk_goals_against(tendy_name)

  high_danger_shots_against <- goalie_high_danger_shots_against(tendy_name)
  high_danger_save_pct <- goalie_high_danger_save_pct(tendy_name)
  high_danger_goals_allowed <- goalie_high_danger_goals_against(tendy_name)
```

```
  mid_range_shots_against <- goalie_mid_range_shots_against(tendy_name)
  mid_range_save_pct <- goalie_mid_range_save_pct(tendy_name)
  mid_range_goals_allowed <- goalie_mid_range_goals_against(tendy_name)

  long_range_shots_against <- goalie_long_range_shots_against(tendy_name)
  long_range_save_pct <- goalie_long_range_save_pct(tendy_name)
  long_range_goals_allowed <- goalie_long_range_goals_against(tendy_name)

  goalie_name_vector <- c(goalie_name_vector, tendy_name)
  goalie_games_played_vector <- c(goalie_games_played_vector, games_played)
  goalie_sa_vector <- c(goalie_sa_vector, shots_against)
  goalie_shot_distance_vector <- c(goalie_shot_distance_vector, shot_distance)
  goalie_goals_allowed_vector <- c(goalie_goals_allowed_vector, goals_allowed)
  goalie_save_pct_vector <- c(goalie_save_pct_vector, save_pct)

  goalie_pk_shots_against_vector <- c(goalie_pk_shots_against_vector, pk_shots_against)
  goalie_pk_goals_allowed_vector <- c(goalie_pk_goals_allowed_vector, pk_goals_allowed)
  goalie_pk_save_pct_vector <- c(goalie_pk_save_pct_vector, pk_save_pct)

  goalie_high_danger_shots_against_vector <- c(goalie_high_danger_shots_against_vector, high_danger_sho
  goalie_high_danger_goals_allowed_vector <- c(goalie_high_danger_goals_allowed_vector, high_danger_goa
  goalie_high_danger_save_pct_vector <- c(goalie_high_danger_save_pct_vector, high_danger_save_pct)

  goalie_mid_range_shots_against_vector <- c(goalie_mid_range_shots_against_vector, mid_range_shots_aga
  goalie_mid_range_goals_allowed_vector <- c(goalie_mid_range_goals_allowed_vector, mid_range_goals_all
  goalie_mid_range_save_pct_vector <- c(goalie_mid_range_save_pct_vector, mid_range_save_pct)

  goalie_long_range_shots_against_vector <- c(goalie_long_range_shots_against_vector, long_range_shots_
  goalie_long_range_goals_allowed_vector <- c(goalie_long_range_goals_allowed_vector, long_range_goals_
  goalie_long_range_save_pct_vector <- c(goalie_long_range_save_pct_vector, long_range_save_pct)
}

# Create a data frame with the statistics
goalie_stats_df <- data.frame(
  "Goalie Name" = goalie_name_vector,
  "Goalie Games Played" = goalie_games_played_vector,
  "Goalie Shots Against" = goalie_sa_vector,
  "Goalie Goals Allowed" = goalie_goals_allowed_vector,
  "Goalie Save %" = goalie_save_pct_vector,
  "Goalie PK Shots Against" = goalie_pk_shots_against_vector,
  "Goalie PK Goals Allowed" = goalie_pk_goals_allowed_vector,
  "Goalie PK Save %" = goalie_pk_save_pct_vector,
  "Goalie High Danger Shots Against" = goalie_high_danger_shots_against_vector,
  "Goalie High Danger Goals Allowed" = goalie_high_danger_goals_allowed_vector,
  "Goalie High Danger Save %" = goalie_high_danger_save_pct_vector,
  "Goalie Mid Range Shots Against" = goalie_mid_range_shots_against_vector,
  "Goalie Mid Range Goals Allowed" = goalie_mid_range_goals_allowed_vector,
  "Goalie Mid Range Save %" = goalie_mid_range_save_pct_vector,
  "Goalie Long Range Shots Against" = goalie_long_range_shots_against_vector,
  "Goalie Long Range Goals Allowed" = goalie_long_range_goals_allowed_vector,
  "Goalie Long Range Save %" = goalie_long_range_save_pct_vector
)
```

```r
# View the results
View(goalie_stats_df)
```

#CSV Writing

```r
goalie_stats_csv <- write.csv(goalie_stats_df, "goalie_stats_csv", row.names = FALSE)
```

#Table and Graphs code

```r
#Data used to create graphs based on best overall save %
best_save_pct <- goalie_stats_df %>%
  group_by(Goalie.Name) %>%
  filter(Goalie.Shots.Against > 855) %>%
  summarize(
    games_played = max(Goalie.Games.Played),
    save_pct = max(Goalie.Save..),
    shots_against = max(Goalie.Shots.Against),
    goals_allowed = max(Goalie.Goals.Allowed)
  ) %>%
  ungroup() %>%
  arrange(desc(save_pct)) %>%
  mutate(Rank = row_number()) %>%
  slice_head(n = 10)

#Data used to create graphs based on best PK Save %
best_pk_save_pct <- goalie_stats_df %>%
  group_by(Goalie.Name) %>%
  filter(Goalie.Shots.Against > 855) %>%
  summarize(
    games_played = max(Goalie.Games.Played),
    pk_save_pct = max(Goalie.PK.Save..),
    pk_goals_allowed = max(Goalie.PK.Goals.Allowed),
    pk_shots_against = max(Goalie.PK.Shots.Against)
  ) %>%
  ungroup() %>%
  arrange(desc(pk_save_pct)) %>%
  mutate(Rank = row_number()) %>%
  slice_head(n = 10)

#Data filtered to create graphs based on High Danger Save %
 best_hd_save_pct <- goalie_stats_df %>%
   group_by(Goalie.Name) %>%
   filter(Goalie.Shots.Against > 855) %>%
   summarize(
     high_danger_shots_against = max(Goalie.High.Danger.Shots.Against),
     high_danger_goals_allowed = max(Goalie.High.Danger.Goals.Allowed),
     high_danger_save_pct = max(Goalie.High.Danger.Save..)
   ) %>%
   ungroup() %>%
   arrange(desc(high_danger_save_pct)) %>%
   mutate(Rank = row_number()) %>%
   slice_head(n = 10)
```

```r
#Filtered data that is being used to create graphs based on
#mid danger/range save %
best_mid_range_save_pct <- goalie_stats_df %>%
  group_by(Goalie.Name) %>%
  filter(Goalie.Shots.Against > 855) %>%
  summarize(
    mid_danger_shots_against = max(Goalie.Mid.Range.Shots.Against),
    mid_danger_goals_allowed = max(Goalie.Mid.Range.Goals.Allowed),
    mid_range_save_pct = max(Goalie.Mid.Range.Save..)
  ) %>%
  ungroup() %>%
  arrange(desc(mid_range_save_pct)) %>%
  mutate(Rank = row_number()) %>%
  slice_head(n = 10)

#Filtered data used to create graphs based on long range save %
best_long_range_save_pct <- goalie_stats_df %>%
  group_by(Goalie.Name) %>%
  filter(Goalie.Shots.Against > 855) %>%
  summarize(
    long_range_shots_against = max(Goalie.Long.Range.Shots.Against),
    long_range_goals_allowed = max(Goalie.Long.Range.Goals.Allowed),
    long_range_save_pct = max(Goalie.Long.Range.Save..)
  ) %>%
  ungroup() %>%
  arrange(desc(long_range_save_pct)) %>%
  mutate(Rank = row_number()) %>%
  slice_head(n = 10)

#Table used to rank top 10 goalies by overall save percentage
best_save_pct_table <- best_save_pct %>%
  gt() %>%
  tab_header(
    title = "Top 10 Goalies by Save %"
  ) %>%
  cols_label(
    Goalie.Name = "Goalie Name",
    shots_against = "Shots Faced",
    goals_allowed = "Goals Allowed",
    save_pct = "Save Percentage"
  ) %>%
  tab_options(
    table.font.size = 12
  )

#Table used to rank top 10 goalies by best PK save %
best_pk_save_pct_table <- best_pk_save_pct %>%
  gt() %>%
  tab_header(
    title = "Top 10 Goalies by PK Save %"
  ) %>%
  cols_label(
    Goalie.Name = "Goalie Name",
```

```
    pk_shots_against = "Shots Faced",
    pk_goals_allowed = "Goals Allowed",
    pk_save_pct = "Save Percentage"
  ) %>%
  tab_options(
    table.font.size = 12
  )
#Table used to rank top 10 goalies based on High Danger save %
best_hd_save_pct_table <- best_hd_save_pct %>%
  gt() %>%
  tab_header(
    title = "Top 10 Goalies by High Danger Save %"
  ) %>%
  cols_label(
    Goalie.Name = "Goalie Name",
    high_danger_shots_against = "Shots Faced",
    high_danger_goals_allowed = "Goals Allowed",
    high_danger_save_pct = "Save Percentage"
  ) %>%
  tab_options(
    table.font.size = 12
  )
#Table used to rank top 10 goalies based on Mid Range/Danger save %
best_mid_range_save_pct_table <- best_mid_range_save_pct %>%
  gt() %>%
  tab_header(
    title = "Top 10 Goalies by Mid Range Save %"
  ) %>%
  cols_label(
    Goalie.Name = "Goalie Name",
    mid_danger_shots_against = "Shots Faced",
    mid_danger_goals_allowed = "Goals Allowed",
    mid_range_save_pct = "Save Percentage"
  ) %>%
  tab_options(
    table.font.size = 12
  )
#Table used to rank top 10 goalies based on Long Range save %

best_long_range_save_pct_table <- best_long_range_save_pct %>%
  gt() %>%
  tab_header(
    title = "Top 10 Goalies by Long Range Save %"
  ) %>%
  cols_label(
    Goalie.Name = "Goalie Name",
    long_range_shots_against = "Shots Faced",
    long_range_goals_allowed = "Goals Allowed",
    long_range_save_pct = "Save Percentage"
  ) %>%
  tab_options(
    table.font.size = 12
  )
```

```
# Print the tables
best_pk_save_pct_table
```

Top 10 Goalies by PK Save %

Goalie Name	games_played	Save Percentage	Goals Allowed	Shots Faced	Rank
Adin Hill	38	0.899	13	142	1
Pyotr Kochetkov	43	0.893	14	145	2
Cam Talbot	57	0.888	28	277	3
Thatcher Demko	52	0.886	25	244	4
Jeremy Swayman	56	0.882	29	275	5
Ukko-Pekka Luukkonen	54	0.882	24	227	6
Andrei Vasilevskiy	57	0.874	25	223	7
Joey Daccord	50	0.873	19	169	8
Sergei Bobrovsky	82	0.870	41	357	9
Sam Montembeault	41	0.867	26	222	10

```
best_save_pct_table
```

Top 10 Goalies by Save %

Goalie Name	games_played	Save Percentage	Shots Faced	Goals Allowed	Rank
Jeremy Swayman	56	0.925	1601	120	1
Connor Hellebuyck	65	0.924	1903	145	2
Joey Daccord	50	0.924	1326	101	3
Linus Ullmark	42	0.923	1212	93	4
Semyon Varlamov	33	0.922	953	74	5
Igor Shesterkin	71	0.921	2070	163	6
Adin Hill	38	0.919	1028	83	7
Thatcher Demko	52	0.919	1448	117	8
Jordan Binnington	57	0.917	1750	146	9
Sergei Bobrovsky	82	0.917	2109	175	10

```
best_hd_save_pct_table
```

Top 10 Goalies by High Danger Save %

Goalie Name	Shots Faced	Goals Allowed	Save Percentage	Rank
Jeremy Swayman	694	87	0.857	1
Thatcher Demko	648	84	0.851	2
Connor Hellebuyck	841	111	0.848	3
Jacob Markstrom	652	86	0.848	4
Linus Ullmark	523	71	0.843	5
Jordan Binnington	828	115	0.839	6
Ukko-Pekka Luukkonen	621	86	0.839	7
Pyotr Kochetkov	481	67	0.838	8
Semyon Varlamov	393	55	0.837	9

Connor Ingram	636	91	0.833	10

```
best_mid_range_save_pct_table
```

Top 10 Goalies by Mid Range Save %

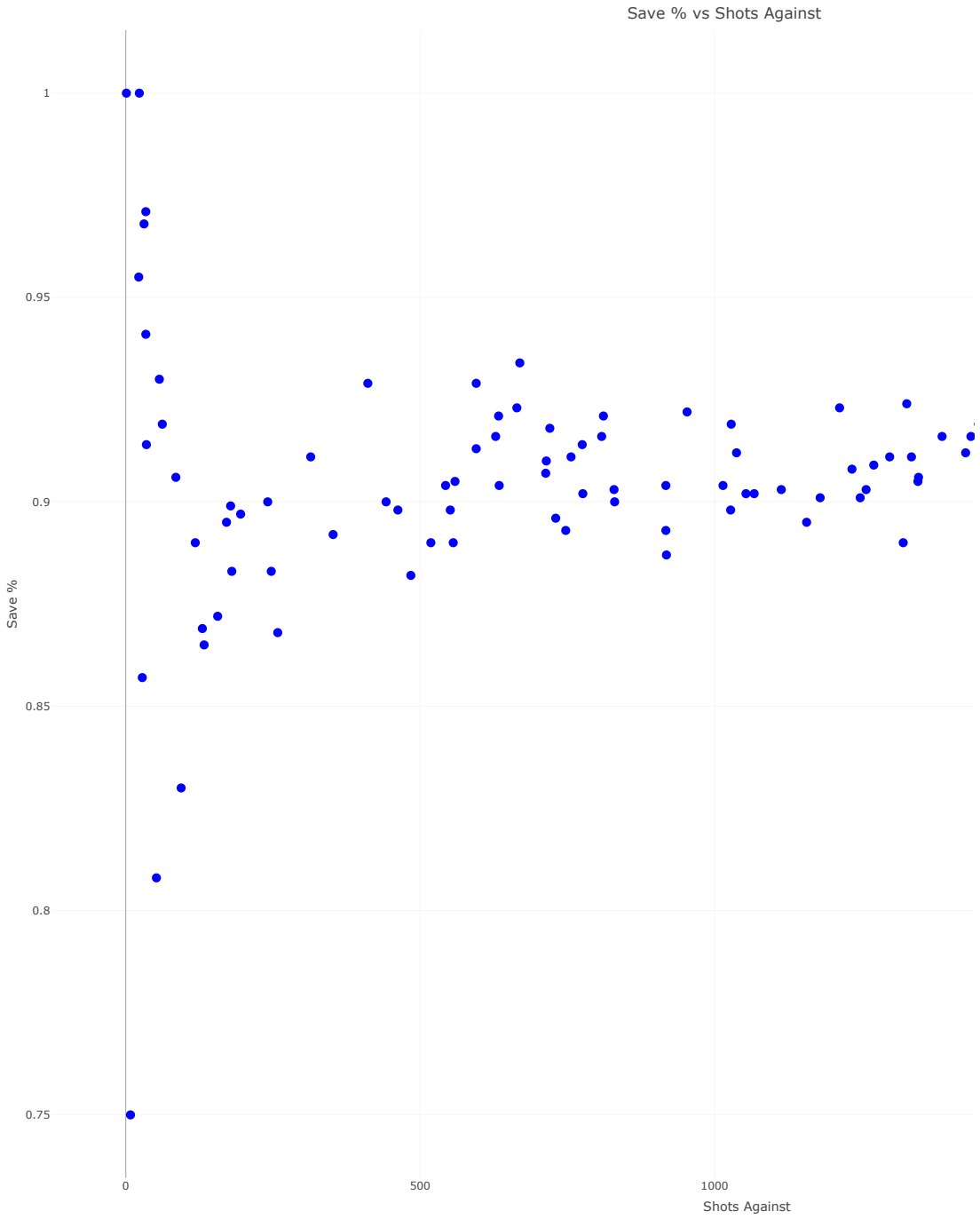
Goalie Name	Shots Faced	Goals Allowed	Save Percentage	Rank
Igor Shesterkin	522	18	0.964	1
Philipp Grubauer	243	9	0.962	2
Alex Nedeljkovic	255	11	0.955	3
Tristan Jarry	339	15	0.954	4
Jordan Binnington	410	19	0.951	5
Elvis Merzlikins	314	15	0.950	6
Juuse Saros	545	27	0.948	7
Lukas Dostal	332	17	0.946	8
Logan Thompson	351	19	0.943	9
Sam Montembeault	288	16	0.941	10

```
best_long_range_save_pct_table
```

Top 10 Goalies by Long Range Save %

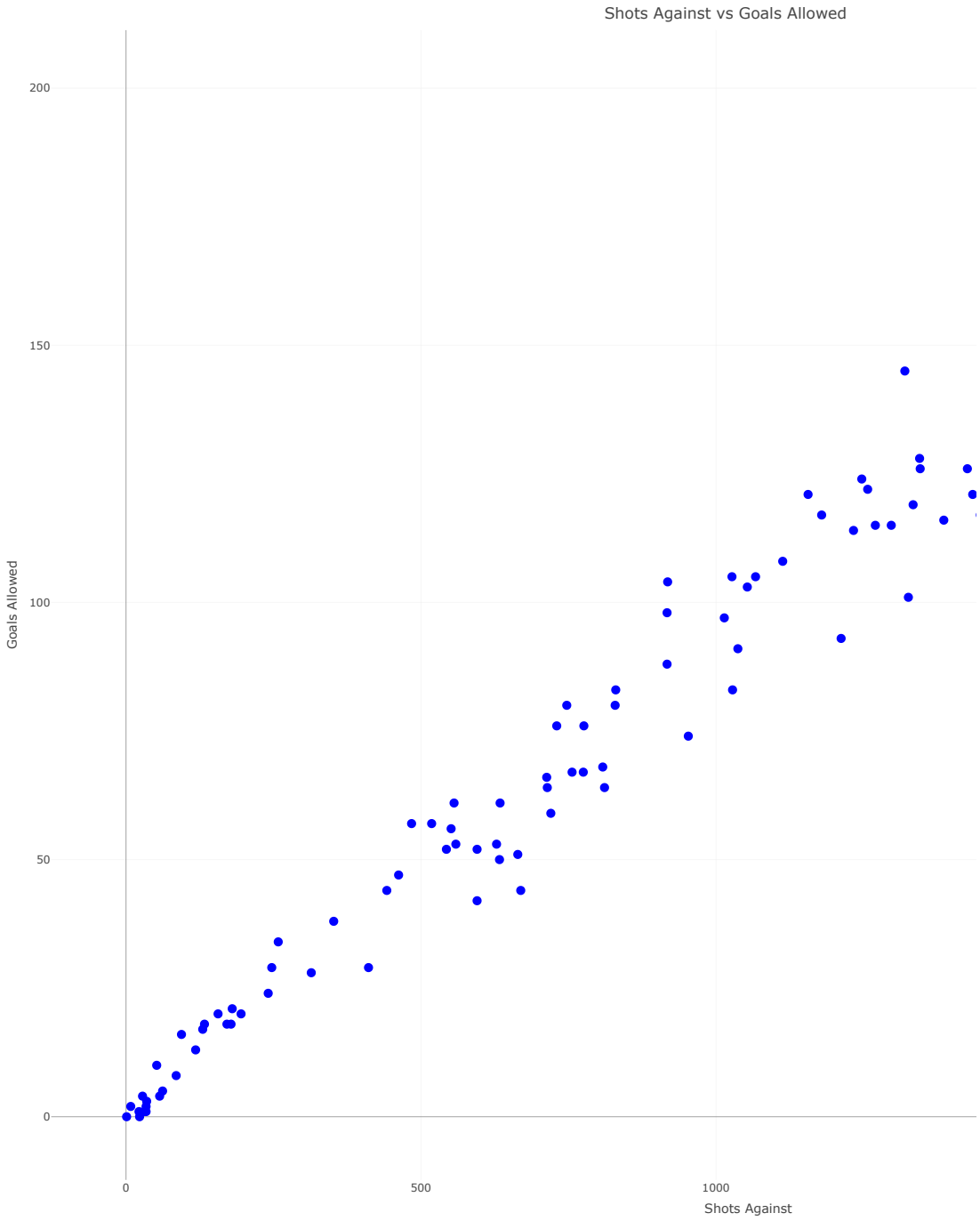
Goalie Name	Shots Faced	Goals Allowed	Save Percentage	Rank
Adin Hill	238	1	0.997	1
Connor Hellebuyck	461	5	0.992	2
Linus Ullmark	297	4	0.990	3
Petr Mrazek	373	5	0.990	4
Joey Daccord	340	5	0.989	5
Semyon Varlamov	222	4	0.987	6
Jake Oettinger	466	9	0.986	7
John Gibson	299	6	0.985	8
Sergei Bobrovsky	509	11	0.985	9
Igor Shesterkin	465	10	0.984	10

```
#Plotting a graph showcasing shots against vs save percentage for
#different goalies
plot_ly(x = goalie_stats_df$Goalie.Shots.Against,
        y = goalie_stats_df$Goalie.Save..,
        type = "scatter",
        mode = "markers",
        marker = list(size = 10, color = "blue"),
        text = goalie_stats_df$Goalie.Name,
        hoverinfo = 'text') %>%
  layout(
    title = "Save % vs Shots Against",
    xaxis = list(title = "Shots Against"),
    yaxis = list(title = "Save %")
  )
```



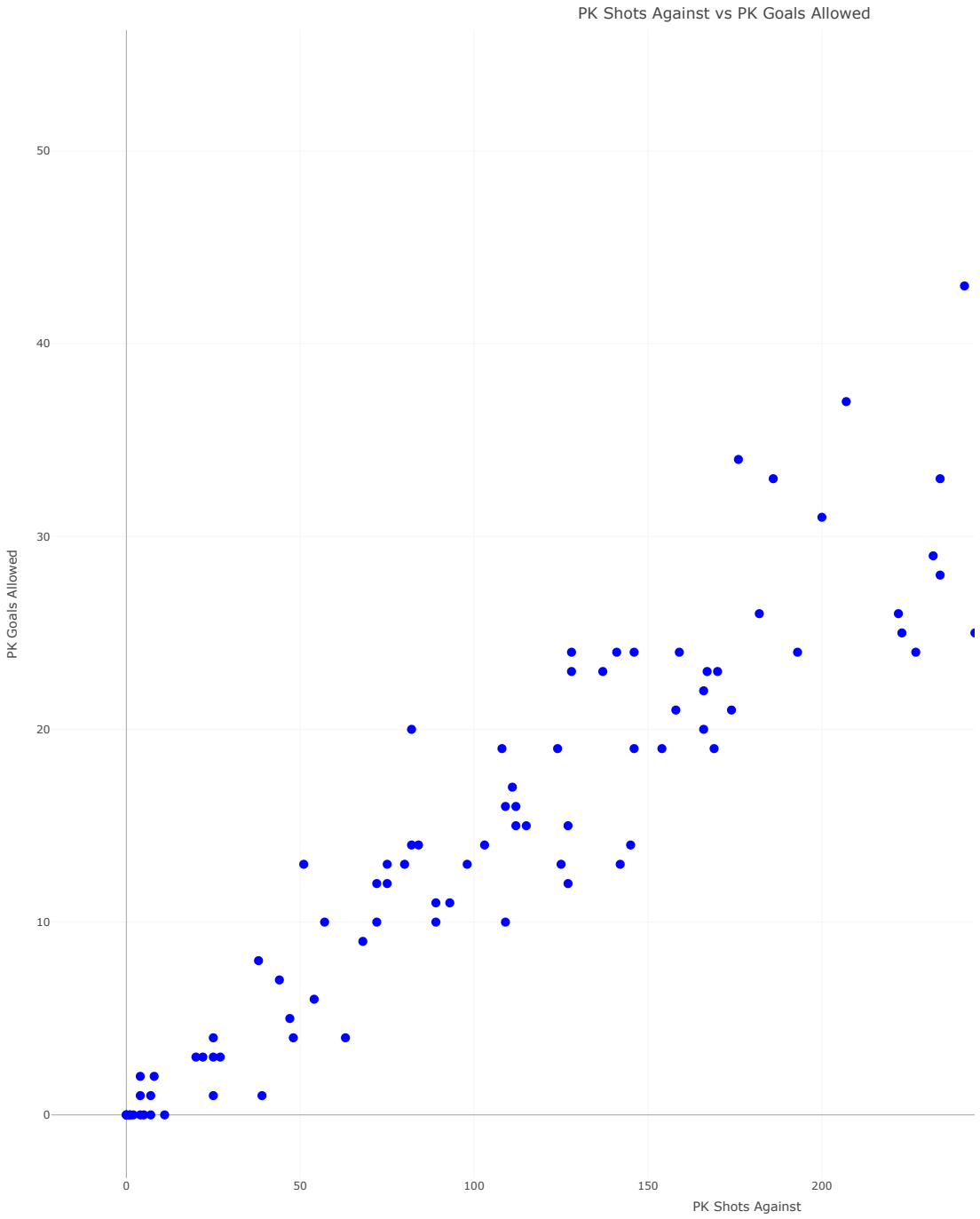

```
#creating a graph showcasing goals against vs goals allowed
#for every goalie
```

```r
plot_ly(x = goalie_stats_df$Goalie.Shots.Against,
        y = goalie_stats_df$Goalie.Goals.Allowed,
        type = "scatter",
        mode = "markers",
        marker = list(size = 10, color = "blue"),
        text = goalie_stats_df$Goalie.Name,
        hoverinfo = 'text') %>%
  layout(
    title = "Shots Against vs Goals Allowed",
    xaxis = list(title = "Shots Against"),
    yaxis = list(title = "Goals Allowed")
  )
```




```
#Creating a graph analysing PK Shots Against vs PK Goals Allowed
#for every goalie
```

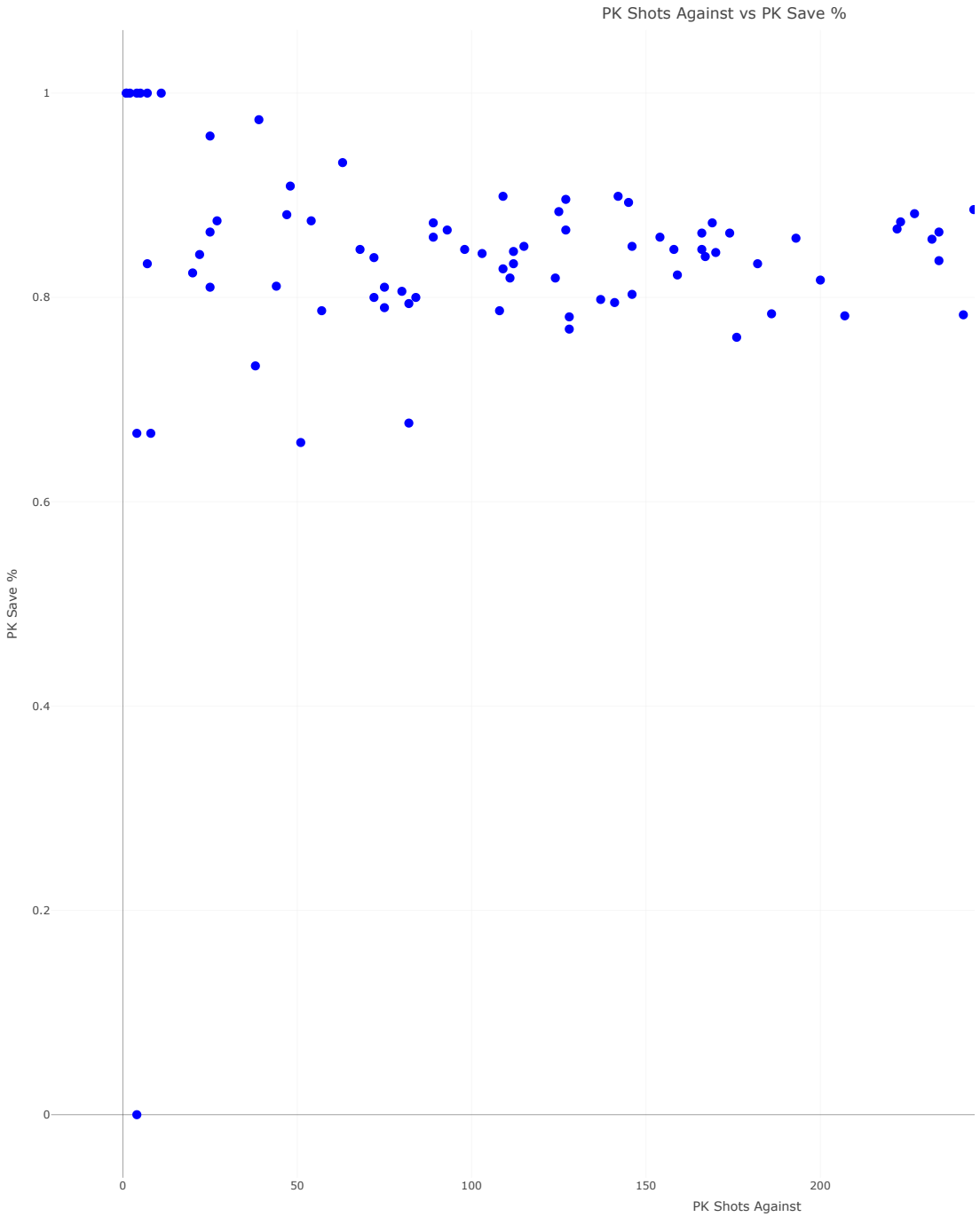
```r
plot_ly(x = goalie_stats_df$Goalie.PK.Shots.Against,
        y = goalie_stats_df$Goalie.PK.Goals.Allowed,
        type = "scatter",
        mode = "markers",
        marker = list(size = 10, color = "blue"),
        text = goalie_stats_df$Goalie.Name,
        hoverinfo = 'text') %>%
  layout(
    title = "PK Shots Against vs PK Goals Allowed",
    xaxis = list(title = "PK Shots Against"),
    yaxis = list(title = "PK Goals Allowed")
  )
```




```
# Creating a graph showcasing PK Shots Against vs PK Save % for
# every goalie
```

```r
plot_ly(x = goalie_stats_df$Goalie.PK.Shots.Against,
        y =  goalie_stats_df$Goalie.PK.Save..,
        type = "scatter",
        mode = "markers",
        marker = list(size = 10, color = "blue"),
        text = goalie_stats_df$Goalie.Name,
        hoverinfo = 'text') %>%
  layout(
    title = "PK Shots Against vs PK Save %",
    xaxis = list(title = "PK Shots Against"),
    yaxis = list(title = "PK Save %")
  )
```

```
## Warning: Ignoring 3 observations
```


PK Shots Against vs PK Save %
PK Save %
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
PK Shots Against