Sacramento Kings All Star Break Analysis (2022-23)

The Sacramento Kings are my team favourite in the NBA and are on the cusp of breaking a 16-year playoff drought. As we enter the All-Star break, I wanted to find out where the Kings were performing well/poorly, and how this impacts the outcome of games.

## Aim: Which stats are important to the Sacramento Kings in wins and losses?

nba = read\_csv("./clean data/Regular\_Season\_H2H\_Boxscore\_2022-23.csv") %>%  
 mutate(`GAME DATE`=as.Date(`GAME DATE`, "%m/%d/%Y"))  
sac = nba %>%   
 filter(`GAME DATE` < "2023-02-17") %>% # games before all star break  
 filter(`TEAM` == "SAC")  
head(sac)

## # A tibble: 6 × 91  
## TEAM MATCH…¹ GAME DAT…² `W/L` MIN PTS FGM FGA `FG%` `3PM` `3PA` `3P%`  
## <chr> <chr> <date> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 SAC SAC @ … 2023-02-14 L 48 109 37 83 44.6 14 38 36.8  
## 2 SAC SAC vs… 2023-02-11 W 53 133 51 100 51 8 32 25   
## 3 SAC SAC vs… 2023-02-10 L 48 114 39 72 54.2 9 27 33.3  
## 4 SAC SAC @ … 2023-02-08 W 48 130 46 92 50 10 30 33.3  
## 5 SAC SAC @ … 2023-02-06 W 48 140 52 89 58.4 21 41 51.2  
## 6 SAC SAC @ … 2023-02-05 L 48 104 35 82 42.7 11 42 26.2  
## # … with 79 more variables: FTM <dbl>, FTA <dbl>, `FT%` <dbl>, OREB <dbl>,  
## # DREB <dbl>, REB <dbl>, AST <dbl>, TOV <dbl>, STL <dbl>, BLK <dbl>,  
## # PF <dbl>, `+/-` <dbl>, OFFRTG <dbl>, DEFRTG <dbl>, NETRTG <dbl>,  
## # `AST%` <dbl>, `AST/TO` <dbl>, ASTRATIO <dbl>, `OREB%` <dbl>, `DREB%` <dbl>,  
## # `REB%` <dbl>, `TOV%` <dbl>, `EFG%` <dbl>, `TS%` <dbl>, PACE <dbl>,  
## # PIE <dbl>, FTARATE <dbl>, `PTSOFF TO` <dbl>, `2NDPTS` <dbl>, FBPS <dbl>,  
## # PITP <dbl>, `2PM` <dbl>, `2PA` <dbl>, `2P%` <dbl>, `TEAM OPP` <chr>, …

This dataset was personally scraped from NBA.com (using Python Selenium) and is a collection of advanced boxscores from each NBA game played so far in the 2022-23 season. Since we’re interested in the Kings, we have filtered for such boxscores only.

## Data Cleaning

We have 91 columns to work with. Many of the variables are redundant or not necessary.

# Remove unnecessary columns  
sac = sac %>%  
 select(-c("TEAM","MIN","MATCH UP","GAME DATE","W/L","MIN OPP","MATCH UP OPP","W/L OPP","MIN OPP"))

# Remove redundant columns  
remove1 = c("FGM","FGA","FG%","3PM","3PA","FTM","FTA","REB","AST%","ASTRATIO","OREB%","DREB%","REB%","TOV%","TS%","FTARATE","2PM","2PA")  
remove2 = paste(remove1, "OPP")  
sac = sac %>%  
 select(-all\_of(c(remove1, remove2)))

Removing redundant columns helps reduce multi-collinearity, important for model building. I have decided to keep EFG% instead of FG% since it accounts for 3PT being worth one more point than 2PT (Formula: EFG% = (FGM+0.5\*3PM)/FGA).

OFFRTG/DEFRTG are heavily influenced by points. This statistic would be better suited for player analysis; For teams, looking at points will suffice.

I want to keep statistics from actual aspects of the game, and not derivatives. Doing so, I can look at areas of improvement WITHIN the context of the game; derived statistics (e.g. ORTG, PIE, etc.) is not something players can track/improve upon in-game.

# Keep simple, interpretative stats  
remove3 = c("OFFRTG","DEFRTG","NETRTG","AST/TO","PIE","PACE")  
remove4 = paste(remove3, "OPP")  
sac = sac %>%  
 select(-all\_of(c(remove3, remove4)))

# Change TEAM OPP to factor  
sac = sac %>%  
 mutate(`TEAM OPP`=as.factor(`TEAM OPP`), `+/- OPP`=NULL) %>%  
 rename(`NET PTS`=`+/-`)

## Data Exploration

There are two sides to the game: Offense and Defense. Offense aims to increase the number of points scored, and defense aims to reduce the opponent’s points scored. The aim of the game is to ultimately score more points than your opponent by strategically utilising these two aspects.

#### Response variable: NET PTS.

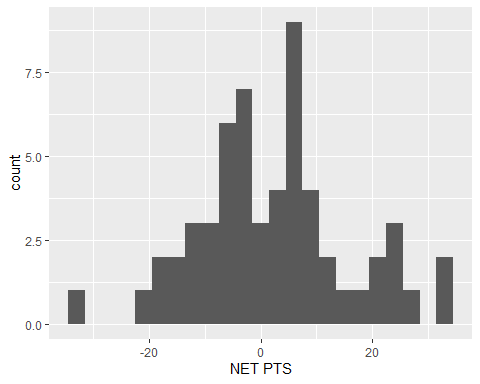
NET PTS (PTS FOR - PTS AGAINST) captures two things:

1. Outcome of a game (+ve := win, -ve := loss)
2. How close of a win/loss the game was.

This is useful since we can understand which stats were the biggest contributors to such outcomes, and by what magnitude.

#### NET PTS Distribution:

# Distribution of NET PTS  
ggplot(aes(x=`NET PTS`), data=sac) +  
 geom\_histogram(binwidth=3)



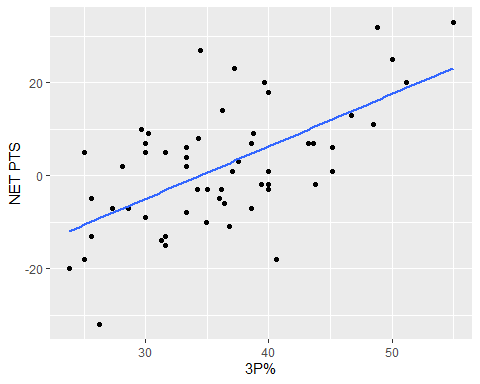
NET PTS is reasonably normally distributed.

Given that there are many explanatory variables, we will explore a few interesting ASSOCIATIONS first. Then, find a combination of explanatory variables which help explain the NET PTS (outcome of games) observed.

### NET PTS and 3P%

sac %>%  
 ggplot(aes(x=`3P%`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'

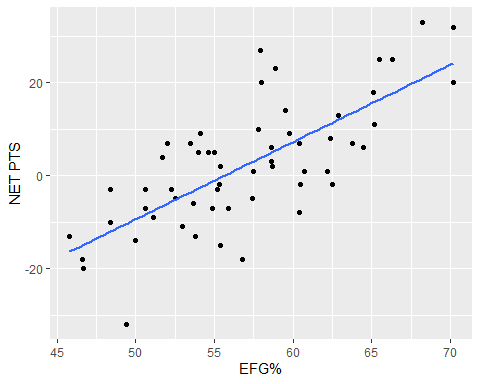


A moderate linear relationship between 3P% and NET PTS. When the Kings shoot <30% from three, they more often lose in blow-out fashion. When the Kings shoot >45%, they more often than not win the game in blow-out fashion. In between the 30-40% range, the outcome of the game is inconclusive (there are other factors likely at play here).

### EFG% and NET PTS

sac %>%  
 ggplot(aes(x=`EFG%`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'

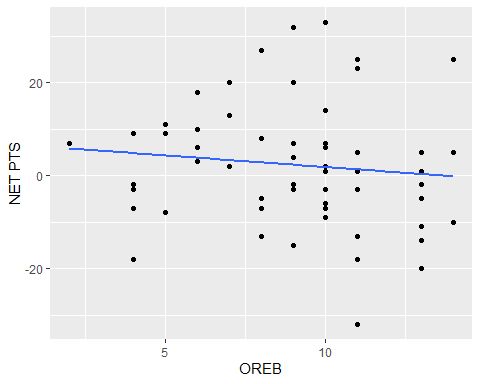


A moderately strong linear relationship between EFG% and NET PTS. How well the Kings shooting can dictate the outcome of games. This isn’t unique to the Kings since this is true for all teams. More factors at play dictate how well the Kings can shoot in a game (such as defensive stats).

### OREB and NET PTS

sac %>%  
 ggplot(aes(x=`OREB`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'

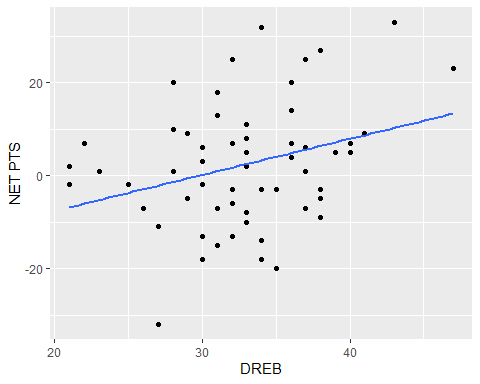


At first glance, it seems like offensive rebounds (OREB) have little impact on the margin of the outcome. As OREB increases, NET PTS slightly decreases. This makes sense, as more OREB indicates more missed shots (but also more 2nd chance opportunities). This relationship is weak (highly variable) and so by itself, cannot explain NET PTS.

### DREB and NET PTS

sac %>%  
 ggplot(aes(x=`DREB`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'

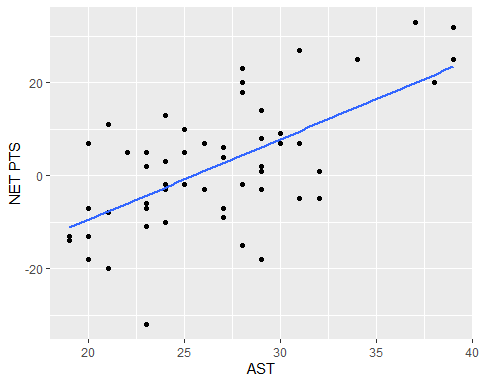


Defensive rebounds are important for closing out a defensive possession (minimises the opportunity for opponents to score). Hence why as DREB increases, NET PTS slightly increases. This relationship is weak (highly variable) and so by itself, cannot explain NET PTS.

### AST and NET PTS

sac %>%  
 ggplot(aes(x=`AST`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'

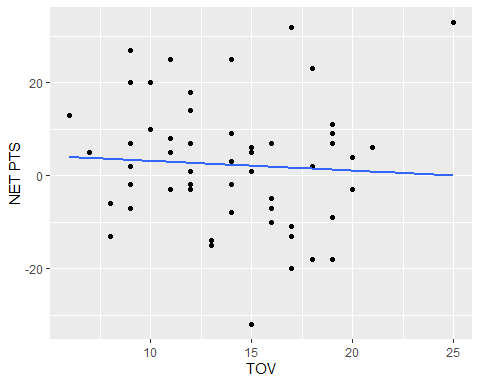


A moderately strong correlation between AST and NET PTS. As AST increases, so too does NET PTS. Assists seem to be a good contributor to the success of a Kings win or loss.

### TOV and NET PTS

sac %>%  
 ggplot(aes(x=`TOV`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

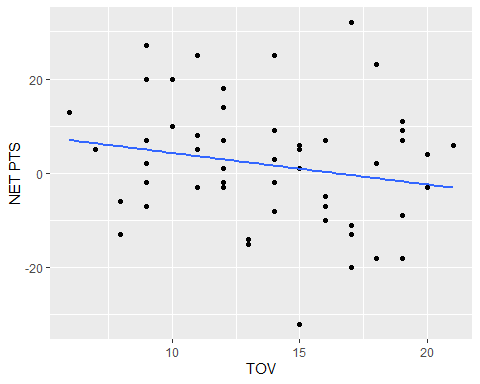
## `geom\_smooth()` using formula = 'y ~ x'



There appears to be an outlier where TOV > 25 and NET PTS > 20 (in actuality, this doesn’t happen often).

# Remove outlier  
sac %>%  
 filter(`TOV` < 25) %>%  
 ggplot(aes(x=`TOV`, y=`NET PTS`)) +  
 geom\_point() + geom\_smooth(method="lm", se=F)

## `geom\_smooth()` using formula = 'y ~ x'



Weak decreasing relationship between TOV and NET PTS. As TOV increases, NET PTS slightly decreases. This makes sense, as less TOV means more opportunities to increase PTS FOR and decreases opportunities to increase PTS AGAINST.

There are many more variables we can explore with NET PTS. However, we will utilise multiple linear regression to find the most important associations for us!

## Data Modelling

First, fit a full linear model. Our objective is to find all explanatory variables with a statistically significant (strong evidence result is not due to chance) impact on NET PTS.

full.fit = lm(`NET PTS` ~ ., data=sac)  
summary(full.fit)

NaN values are produced due to multi-collinearity; Some explanatory variables are almost perfectly correlated with each other, introducing redundancy to the model which doesn’t help. We want variables that further increase our ability to explain the variation seen in NET PTS.

### Handling Multi-Collinearity

We look at variance inflation factors (VIFs) to remove serious cases (>5) of multi-collinearity.

corr\_matrix = sac %>%  
 select(-"NET PTS") %>%  
 select(where(is.numeric)) %>%  
 cor()  
vifs = corr\_matrix %>% solve() %>% diag()  
vifs

## PTS 3P% FT% OREB DREB   
## 41.380075 96.422597 5.954078 7.405339 9.353682   
## AST TOV STL BLK PF   
## 5.290019 17.031341 4.786711 2.302654 4.766412   
## EFG% PTSOFF TO 2NDPTS FBPS PITP   
## 191.103221 4.452911 2.608209 3.857204 3.955581   
## 2P% PTS OPP 3P% OPP FT% OPP OREB OPP   
## 52.005835 45.907507 32.151474 4.048212 15.097716   
## DREB OPP AST OPP TOV OPP STL OPP BLK OPP   
## 11.503624 2.328609 15.064862 6.447461 2.028859   
## PF OPP EFG% OPP PTSOFF TO OPP 2NDPTS OPP FBPS OPP   
## 2.681595 89.506158 6.574455 2.882378 3.097360   
## PITP OPP 2P% OPP   
## 5.352199 45.686499

EFG% is the most serious case of multi-collinearity. This makes sense because 2P% and 3P% are accounted for in this variable.

# Remove EFG% and check VIF  
sac %>%   
 select(-c("NET PTS","EFG%","EFG% OPP")) %>%  
 select(where(is.numeric)) %>%  
 cor() %>%   
 solve() %>%   
 diag()

## PTS 3P% FT% OREB DREB   
## 28.032995 13.592066 4.673042 6.057833 9.110155   
## AST TOV STL BLK PF   
## 4.646800 15.132818 3.943722 2.162243 4.347680   
## PTSOFF TO 2NDPTS FBPS PITP 2P%   
## 4.408639 2.569578 3.492242 3.879730 9.885788   
## PTS OPP 3P% OPP FT% OPP OREB OPP DREB OPP   
## 32.791173 17.427696 3.539683 9.328440 10.252422   
## AST OPP TOV OPP STL OPP BLK OPP PF OPP   
## 2.108205 11.344213 5.968947 1.967655 2.446477   
## PTSOFF TO OPP 2NDPTS OPP FBPS OPP PITP OPP 2P% OPP   
## 6.390745 2.627781 2.827964 4.020193 26.458935

PTS is the next serious case of multi-collinearity.

# Remove PTS and check VIF  
sac %>%   
 select(-c("NET PTS","EFG%","EFG% OPP","PTS","PTS OPP")) %>%  
 select(where(is.numeric)) %>%  
 cor() %>%   
 solve() %>%   
 diag()

## 3P% FT% OREB DREB AST   
## 5.738074 2.680651 3.291834 8.236197 4.121910   
## TOV STL BLK PF PTSOFF TO   
## 9.656480 3.846077 1.957499 2.678300 3.347391   
## 2NDPTS FBPS PITP 2P% 3P% OPP   
## 2.455659 3.038540 3.665697 5.849904 3.877239   
## FT% OPP OREB OPP DREB OPP AST OPP TOV OPP   
## 2.261435 4.299851 6.569370 1.993304 5.406792   
## STL OPP BLK OPP PF OPP PTSOFF TO OPP 2NDPTS OPP   
## 5.915526 1.962295 2.251218 6.136047 2.614114   
## FBPS OPP PITP OPP 2P% OPP   
## 2.633898 3.358047 6.450186

While we still have some variables with VIF > 5, we can automate this process using stepwise (variable) selection. We should find that redundant columns are not selected in the final model.

For stepwise selection, we use both forwards and backward step-selection to find the most significant variables.

(Note: TEAM OPP is removed for both multi-collinear and runtime purposes)

BIC is the penalty term chosen to select to avoid over-fitting and reduce model complexity (handle multi-collinearity). This is because we want to find the model which is most likely to be the ‘true’ model.

sac = sac %>% select(-c("EFG%","EFG% OPP","PTS","PTS OPP","TEAM OPP"))  
full.fit2 = lm(`NET PTS` ~ ., data=sac)  
sac.fit1 = step(full.fit2, direction="both", k=log(nrow(sac)), trace=0) # log(n) is BIC (n := # observations)  
summary(sac.fit1)

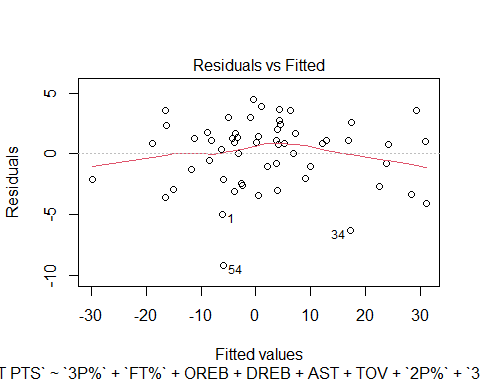
##   
## Call:  
## lm(formula = `NET PTS` ~ `3P%` + `FT%` + OREB + DREB + AST +   
## TOV + `2P%` + `3P% OPP` + `FT% OPP` + `OREB OPP` + `DREB OPP` +   
## `TOV OPP` + `FBPS OPP` + `2P% OPP`, data = sac)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1697 -2.1029 0.8644 1.6859 4.4734   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.72123 20.61983 0.568 0.572760   
## `3P%` 0.74509 0.10837 6.876 2.19e-08 \*\*\*  
## `FT%` 0.21538 0.05787 3.722 0.000582 \*\*\*  
## OREB 0.98918 0.18826 5.254 4.64e-06 \*\*\*  
## DREB 0.42516 0.16487 2.579 0.013511 \*   
## AST 0.33103 0.13288 2.491 0.016765 \*   
## TOV -1.43335 0.14817 -9.674 2.99e-12 \*\*\*  
## `2P%` 0.67117 0.13116 5.117 7.27e-06 \*\*\*  
## `3P% OPP` -0.79207 0.09411 -8.416 1.48e-10 \*\*\*  
## `FT% OPP` -0.14478 0.05308 -2.728 0.009273 \*\*   
## `OREB OPP` -1.11247 0.17983 -6.186 2.14e-07 \*\*\*  
## `DREB OPP` -0.38545 0.17501 -2.202 0.033176 \*   
## `TOV OPP` 1.42750 0.14613 9.769 2.25e-12 \*\*\*  
## `FBPS OPP` -0.22593 0.09324 -2.423 0.019779 \*   
## `2P% OPP` -1.04290 0.12524 -8.327 1.96e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.2 on 42 degrees of freedom  
## Multiple R-squared: 0.9575, Adjusted R-squared: 0.9433   
## F-statistic: 67.57 on 14 and 42 DF, p-value: < 2.2e-16

So far, variables from this model make sense. But before we can make inferences, we check model assumptions.

### Check Model Assumptions

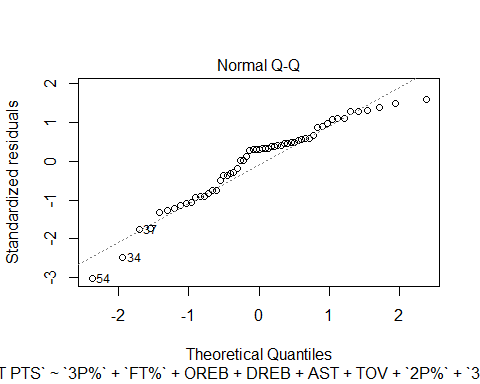
Since our response (NET PTS) is normally distributed, we need to validate the following assumptions…

# (1) Equality of Variance (EOV) Check  
plot(sac.fit1, which=1)

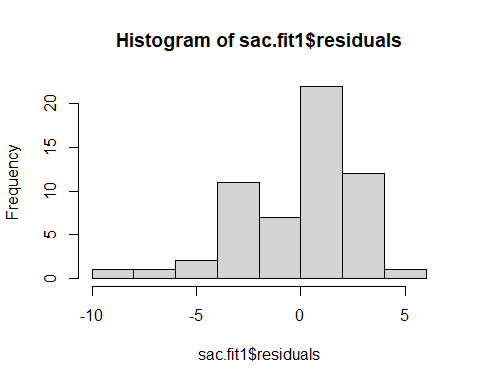


Looks good! Constant scatter with a mean of zero.

# (2) Check residuals normally distributed  
plot(sac.fit1, which=2) # Q-Q Plot

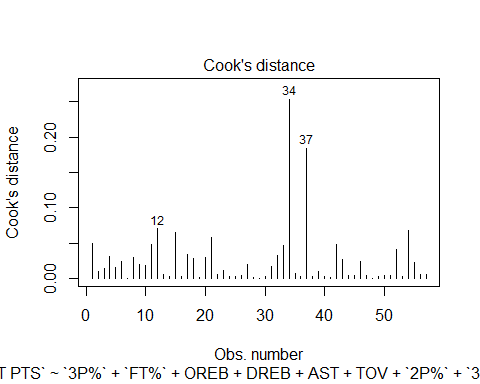


hist(sac.fit1$residuals)



Passable. Histogram may be a little left-skewed. However, the quantiles show no problem.

# (3) Independence - all good! Every game is independent of one another.  
  
# (4) Possible Outliers  
plot(sac.fit1, which=4)



No observation of too much concern.

## Model Inference

summary(sac.fit1)

##   
## Call:  
## lm(formula = `NET PTS` ~ `3P%` + `FT%` + OREB + DREB + AST +   
## TOV + `2P%` + `3P% OPP` + `FT% OPP` + `OREB OPP` + `DREB OPP` +   
## `TOV OPP` + `FBPS OPP` + `2P% OPP`, data = sac)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1697 -2.1029 0.8644 1.6859 4.4734   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.72123 20.61983 0.568 0.572760   
## `3P%` 0.74509 0.10837 6.876 2.19e-08 \*\*\*  
## `FT%` 0.21538 0.05787 3.722 0.000582 \*\*\*  
## OREB 0.98918 0.18826 5.254 4.64e-06 \*\*\*  
## DREB 0.42516 0.16487 2.579 0.013511 \*   
## AST 0.33103 0.13288 2.491 0.016765 \*   
## TOV -1.43335 0.14817 -9.674 2.99e-12 \*\*\*  
## `2P%` 0.67117 0.13116 5.117 7.27e-06 \*\*\*  
## `3P% OPP` -0.79207 0.09411 -8.416 1.48e-10 \*\*\*  
## `FT% OPP` -0.14478 0.05308 -2.728 0.009273 \*\*   
## `OREB OPP` -1.11247 0.17983 -6.186 2.14e-07 \*\*\*  
## `DREB OPP` -0.38545 0.17501 -2.202 0.033176 \*   
## `TOV OPP` 1.42750 0.14613 9.769 2.25e-12 \*\*\*  
## `FBPS OPP` -0.22593 0.09324 -2.423 0.019779 \*   
## `2P% OPP` -1.04290 0.12524 -8.327 1.96e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.2 on 42 degrees of freedom  
## Multiple R-squared: 0.9575, Adjusted R-squared: 0.9433   
## F-statistic: 67.57 on 14 and 42 DF, p-value: < 2.2e-16

Our final model is:

where

Overall, we found that 3P%, FT%, OREB, DREB, AST, 2P%, and TOV OPP have positive impacts on NET PTS (i.e. these stats increase the margin of win/loss).

TOV, 3P% OPP, FT% OPP, OREB OPP, DREB OPP, FBPS OPP, and 2P% OPP have negative impacts on NET PTS (i.e. these stats decrease the margin of win/loss).

Together, these statistics explain 95.8% of the variation observed in NET PTS (i.e. why Kings won/lost by such margins).

Keep in mind, looking at each variable individually is not appropriate as ALL these stats work together to produce the final margin of win/loss.

#### Main Effect Interpretations

round(confint(sac.fit1), 4) # 95% confidence interval

## 2.5 % 97.5 %  
## (Intercept) -29.8913 53.3337  
## `3P%` 0.5264 0.9638  
## `FT%` 0.0986 0.3322  
## OREB 0.6093 1.3691  
## DREB 0.0924 0.7579  
## AST 0.0629 0.5992  
## TOV -1.7324 -1.1343  
## `2P%` 0.4065 0.9359  
## `3P% OPP` -0.9820 -0.6021  
## `FT% OPP` -0.2519 -0.0377  
## `OREB OPP` -1.4754 -0.7496  
## `DREB OPP` -0.7386 -0.0323  
## `TOV OPP` 1.1326 1.7224  
## `FBPS OPP` -0.4141 -0.0378  
## `2P% OPP` -1.2957 -0.7902

* For every one percent increase in 3P%, the Kings increase the margin of the outcome by 0.5264 to 0.9638 points on average.
* For every one percent increase in FT%, the Kings increase the margin of the outcome by 0.0986 to 0.3322 points on average.
* For every one unit increase in OREB, the Kings increase the margin of the outcome by 0.6093 to 1.3691 points on average.
* For every one unit increase in DREB, the Kings increase the margin of the outcome by 0.0924 to 0.7579 points on average.
* For every one unit increase in AST, the Kings increase the margin of the outcome by 0.0629 to 0.5999 points on average.
* For every one unit increase in TOV, the Kings decrease the margin of the outcome by 1.1343 to 1.7324 points on average.
* For every one percent increase in 2P%, the Kings increase the margin of the outcome by 0.4065 to 0.9359 points on average.
* For every one percent increase in 3P% OPP, the Kings decrease the margin of the outcome by 0.6021 to 0.9820 points on average.
* For every one percent increase in FT% OPP, the Kings decrease the margin of the outcome by 0.0377 to 0.2519 points on average.
* For every one unit increase in OREB OPP, the Kings decrease the margin of the outcome by 0.7496 to 1.4754 points on average.
* For every one unit increase in DREB OPP, the Kings decrease the margin of the outcome by 0.0323 to 0.7386 points on average.
* For every one unit increase in TOV OPP, the Kings increase the margin of the outcome by 1.1326 to 1.7224 points on average.
* For every one unit increase in FBPS OPP, the Kings decrease the margin of the outcome by 0.0378 to 0.4141 points on average.
* For every one percent increase in 2P% OPP, the Kings decrease the margin of the outcome by 0.7902 to 1.2957 points on average.

# Prediction for current league averages  
league\_avgs = nba %>%  
 filter(`GAME DATE` < "2023-02-17") %>% # before all-star break  
 select(where(is.numeric)) %>%  
 summarise\_all(mean) %>%  
 select(c(`3P%`,`FT%`,OREB,DREB,AST,TOV,`2P%`,`3P% OPP`,`FT% OPP`,`OREB OPP`,`DREB OPP`,`TOV OPP`,`FBPS OPP`,`2P% OPP`))  
  
predict(sac.fit1, league\_avgs, interval="confidence")

## fit lwr upr  
## 1 0.2424465 -0.9953677 1.480261

Against the current league averages (i.e. Instances where the Kings and the opponent put up league average numbers in a game); We expect 95% of the time, the final margin of outcome to be between -0.9925 and 1.4794 points on average. That is to say, the outcome of the game can go either way for the Kings in this situation.

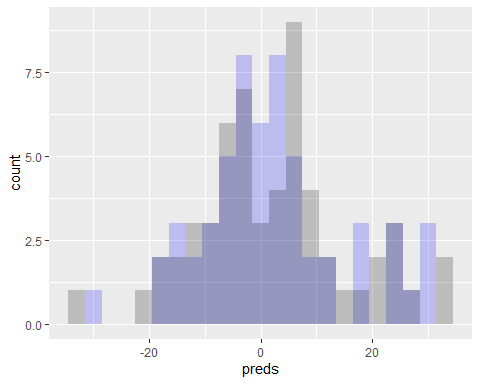
## Conclusion

We found that 3P%, FT%, OREB, DREB, AST, 2P%, and TOV OPP have positive impacts on the outcome for the Kings. And that TOV, 3P% OPP, FT% OPP, OREB OPP, DREB OPP, FBPS OPP, and 2P% OPP have negative impacts on the outcome for the Kings.

Offensively, when the Kings are shooting, rebounding, and assisting each other, they increase their chances of winning.

Defensively, when they turn the opponent over and limit the opponents shooting and rebounding (especially in transition - limit fast break points), then they give themselves a better chance at winning.

# Predicted vs. Actual Values for NET PTS  
sac %>%  
 select(c(`3P%`,`FT%`,OREB,DREB,AST,TOV,`2P%`,`3P% OPP`,`FT% OPP`,`OREB OPP`,`DREB OPP`,`TOV OPP`,`FBPS OPP`,`2P% OPP`)) %>%  
 predict(sac.fit1, data=.) %>%  
 data.frame("preds"=.) %>%  
 ggplot(aes(x=`preds`)) +  
 geom\_histogram(alpha=0.2, fill="blue", binwidth=3) +  
 geom\_histogram(aes(x=`NET PTS`), alpha=0.2, fill="black", binwidth=3, data=sac)



Our fitted linear model (blue) does a reasonably good job of explaining the NET PTS observed with the Kings.

For more extreme values of NET PTS, it is expected that we see some differences - blowout wins/losses are more difficult to exactly predict. For games within 15 points, the linear model does a great job of capturing such outcomes.

## Future Improvements

* More in-depth statistics:

Broad statistics like 2PT, and 3PT don’t tell us a lot. It is possible to break these down further and get statistics about the shot depth from the rim. Eventually, I would like to get data like this to see where the Kings perform well/poorly on the court.

* Add in interactions:

Hidden from the document output were investigations about interactions between some variables. Some interactions would and would not make sense. So if I had the time, I’d like to explore adding the most possible interactions to the model, then running this analysis again.

* Factor in TEAM OPP:

TEAM OPP was eliminated from the model due to redundancy. That is, the Kings most likely performed similarly against certain teams. I still believe certain teams are a ‘bad match-up’ for the Kings. So I’d eventually like to find a way to incorporate this efficiently.

## Predictions for New Games:

Now that we’re a couple of games after the All-Star break. We can see how the model predicts NET PTS (the outcome of games, and by how much).

new\_sac\_games = nba %>%  
 filter(`TEAM` == "SAC") %>%  
 filter(`GAME DATE` > "2023-02-17")  
  
new\_sac\_games %>%  
 select(c(`3P%`,`FT%`,OREB,DREB,AST,TOV,`2P%`,`3P% OPP`,`FT% OPP`,`OREB OPP`,`DREB OPP`,`TOV OPP`,`FBPS OPP`,`2P% OPP`)) %>%  
 predict(sac.fit1, newdata=., interval="confidence") %>%  
 as.tibble() %>%  
 mutate(  
 `actual`=new\_sac\_games$`+/-`,  
 `game date`=new\_sac\_games$`GAME DATE`,  
 `opp`=new\_sac\_games$`MATCH UP`  
 )

## Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
## ℹ Please use `as\_tibble()` instead.  
## ℹ The signature and semantics have changed, see `?as\_tibble`.

## # A tibble: 6 × 6  
## fit lwr upr actual `game date` opp   
## <dbl> <dbl> <dbl> <dbl> <date> <chr>   
## 1 -2.98 -8.39 2.44 -4 2023-03-04 SAC vs. MIN  
## 2 -4.87 -9.24 -0.503 1 2023-03-03 SAC vs. LAC  
## 3 7.74 3.80 11.7 6 2023-02-28 SAC @ OKC   
## 4 6.91 3.25 10.6 9 2023-02-26 SAC @ OKC   
## 5 8.52 1.90 15.1 1 2023-02-24 SAC @ LAC   
## 6 15.4 12.4 18.4 17 2023-02-23 SAC vs. POR

Almost all predicted NET PTS fall within expectation for the Kings. The games against the LAC were anomalies in my opinion (having watched the games), hence why they fall outside our 95% confidence interval bounds. Other than that, the model does a great job of picking up the outcome of Sacramento Kings games based on their performance.