Lesson 9 Boardsheet

Kevin Cummiskey

2/2/2020

Review

During the last two lessons, we discussed three models for predicting the number of wins a team should expect to have in a season. Here are three models:

$$Wpct = \beta_0 + \beta_1 RD + \epsilon \tag{1}$$

$$Wpct = \frac{R^2}{R^2 + RA^2} + \epsilon \tag{2}$$

$$Wpct = \frac{R^k}{R^k + RA^k} + \epsilon \tag{3}$$

where $Wpct$ is Win Percentage, R is Runs Scored, RA is Runs Allowed, and ϵ is the random error. Recall that we fit Model 1 on the 1997-2001 seasons.

Briefly discuss the strengths/limitations of these models.

How would you assess which model is the best? Please be specific.

Model 2 Pythagorean Formula (Bill James)

First, let's create a function to calculate the expected wins under a Pythagorean Formula>

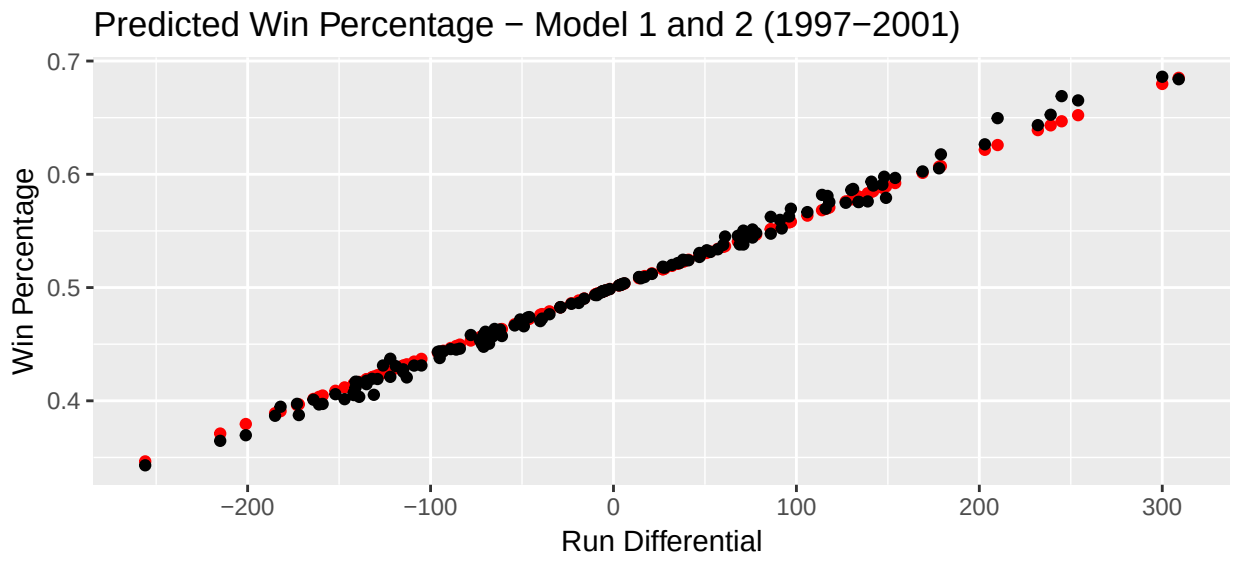
```
#function to calculate expected wins
#using pythagorean formula
#arguments:
# R - runs scored
# RA - runs allowed
# k - exponent
# values:
# expected win percentage
pyt_wins <- function(R, RA, k = 2){
  return(R^k/(R^k + RA^k))
}
```

Using Model 2, let's calculate the expected number of wins for each team (1997-2001).

```
library(tidyverse)
library(Lahman)

my_teams = Teams %>%
  filter(yearID >= 1997, yearID <= 2001) %>%
  mutate(RD = R - RA,
         Wpct = W/(W+L),
         Wpct.pyt2 = pyt_wins(R,RA,2))
```

Next, let's compare graphically the predictions from Model 1 and 2.

```
library(broom)

#get Model 1 predicted win percentages
lin.fit <- lm(Wpct ~ RD, data = my_teams)
my_teams <- augment(lin.fit, data = my_teams)

#plot Model 1 and Model 2 predictions
my_teams %>%
  ggplot(aes(x = RD, y = .fitted)) +
  geom_point(col = "red") +
  geom_point(aes(x = RD, y = Wpct.pyt2)) +
  labs(x = "Run Differential",
       y = "Win Percentage",
       title = "Predicted Win Percentage - Model 1 and 2 (1997-2001)")
```



Briefly discuss interesting how the models differ.

Next, let's compare the root mean square error (RMSE). Write an equation for the RMSE.

```
# Model 1 RMSE
sqrt(mean(my_teams$.resid^2))
```

```
## [1] 0.0228099
```

```
# Note this is very close to the Residual Standard Error
# which you could also use
summary(lin.fit)$sigma
```

```
## [1] 0.02296561
```

```
# Model 2 RMSE
# calculate residuals
my_teams = my_teams %>%
  mutate(.resid.pyt2 = Wpct - Wpct.pyt2)
# calculate RMSE for Model 2
sqrt(mean(my_teams$.resid.pyt2^2))
```

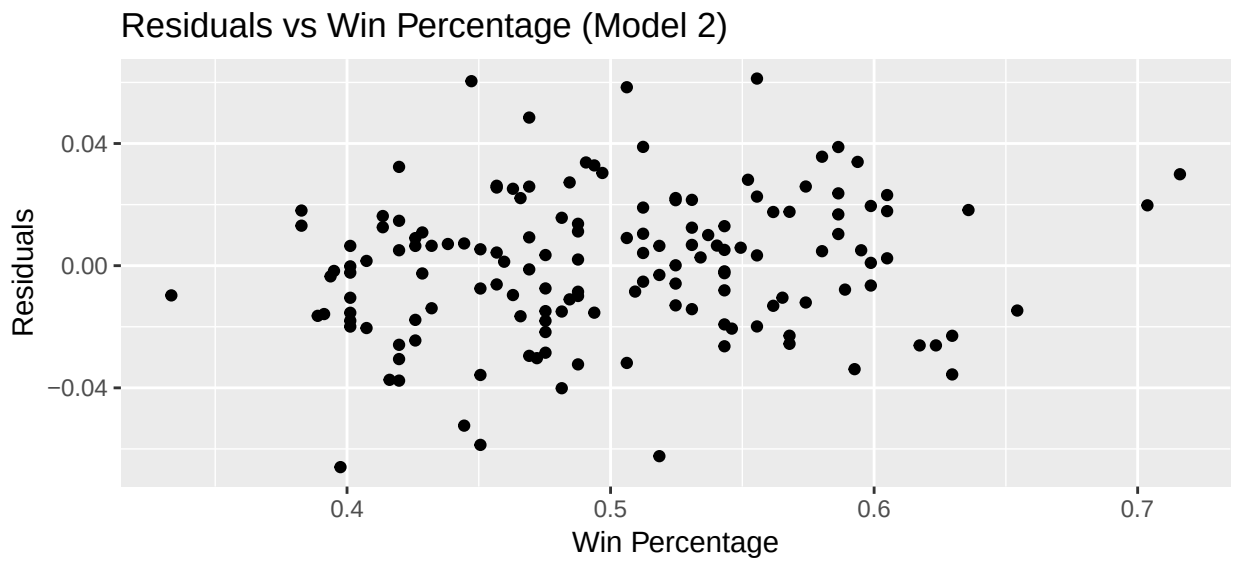
```
## [1] 0.023058
```

Next, let's take a look at the residuals from Model 2.

```
# let's look at min, max, 1Q, 3Q, median residual
my_teams %>%
  summarise(min(.resid.pyt2),
            quantile(.resid.pyt2, 0.25),
            median(.resid.pyt2),
            quantile(.resid.pyt2, 0.75),
            max(.resid.pyt2))
```

```
## # A tibble: 1 x 5
##   `min(.resid.pyt~ `quantile(.resi~ `median(.resid.~ `quantile(.resi~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1          -0.0660          -0.0154          0.00143           0.0158
## # ... with 1 more variable: `max(.resid.pyt2)` <dbl>
```

```
my_teams %>%
  ggplot(aes(x = Wpct,
             y = .resid.pyt2)) +
  geom_point() +
  labs(x = "Win Percentage", y = "Residuals",
       title = "Residuals vs Win Percentage (Model 2)")
```



Next, let's try Model 3.

How do we find an estimate of k in Model 3?

```
my_teams = my_teams %>%
  mutate(logWratio = log(W/L),
         logRratio = log(R/RA))

#0 in formula means we don't include an intercept
pytFit <- lm(logWratio ~ 0 + logRratio, data = my_teams)
pytFit
```

```
##
## Call:
## lm(formula = logWratio ~ 0 + logRratio, data = my_teams)
##
## Coefficients:
## logRratio
##     1.904
```

Why don't we want an intercept in this model? (Warning: this is very unusual!!)

```
k = pytFit$coefficients[1]

#get predictions
my_teams = my_teams %>%
  mutate(Wpct.pyt_k = pyt_wins(R,RA,k = k))

#get residuals
my_teams = my_teams %>%
```

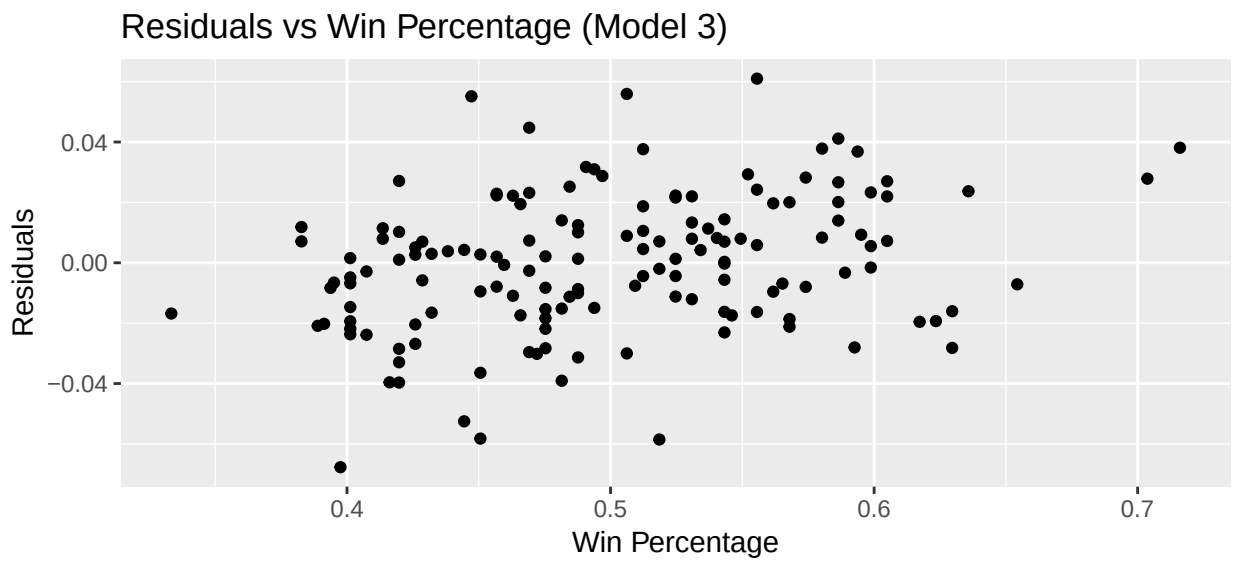
```
  mutate(.resid.pyt_k = Wpct - Wpct.pyt_k)

#calculate RMSE
sqrt(mean(my_teams$.resid.pyt_k^2))
```

```
## [1] 0.02279093
```

```
my_teams %>%
  summarise(min(.resid.pyt_k),
            quantile(.resid.pyt_k, 0.25),
            median(.resid.pyt_k),
            quantile(.resid.pyt_k, 0.75),
            max(.resid.pyt_k))
```

```
## # A tibble: 1 x 5
##   `min(.resid.pyt~ `quantile(.resi~ `median(.resid.~ `quantile(.resi~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1          -0.0677          -0.0163          0.00116           0.0140
## # ... with 1 more variable: `max(.resid.pyt_k)` <dbl>
```

```
my_teams %>%
  ggplot(aes(x = Wpct,
             y = .resid.pyt_k)) +
  geom_point() +
  labs(x = "Win Percentage", y = "Residuals",
       title = "Residuals vs Win Percentage (Model 3)")
```



#add expand.grid code here