**KANSAS CITY ROYALS WINS BY SEASON**

For this project, I tried to model the number of wins each year by the Kansas City Royals (of Major League Baseball). I got this data from the respected baseball statistics website Baseball Reference (<http://www.baseball-reference.com/teams/KCR/>). The data goes back to 1969, when the Royals were an expansion team, and continues through the last completed season in 2012. This gives a total of 44 data points each uniformly indexed by 1 year, which I feel is a large enough data set to model. The goal of this project is to find an accurate portrayal of the Royals’ performance through time, as well as attempting to crudely predict the number of wins in future seasons without having data like roster salaries, individual player statistics, rival teams rosters, etc, that would drastically affect this model. However, I will attempt to do this only having the time series of number of wins in a given year.

> Wins<-as.matrix(read.table("wins.txt",header=FALSE))

> Years<-matrix(seq(1969,2012),ncol=1)

> model<-lm(Wins~Years)

> model

Call:

lm(formula = Wins ~ Years)

Coefficients:

(Intercept) Years

1005.2448 -0.4667

> plot(Years,Wins)

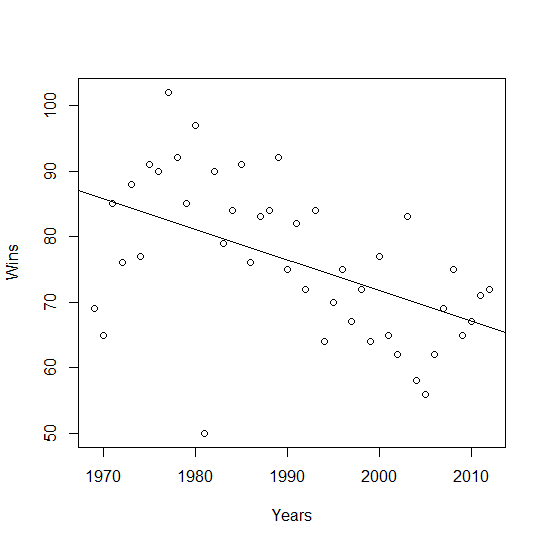
> abline(model)

Modeling this data with R, I get

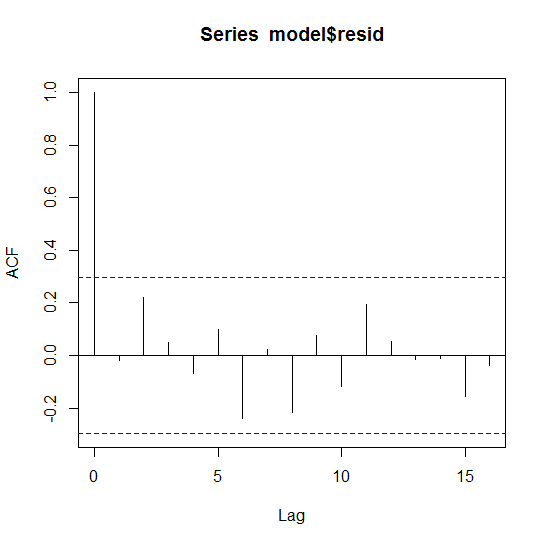
Number of Wins = -0.4667x + 1005.2448

as a linear model, with x being the year. For example, the approximation for the year 2012 with this model is -0.4667\*2012 + 1005.2448 = 66.2444, which compares to the observed value of 72 rather well.

**Graph of linear fit model**



**Residual Graph**



> acf(model$resid,type="covariance")

> acf(model$resid,type="correlation")

**Finding an ARMA process**

Because the ACF and ACVF both bounce between positive and negative values, it seems unlikely to me that this is an MA process. Due to this, I attempted to model this as an AR process.

**As white noise:**

> Y<-arima(model$resid,order=c(0,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(0, 0, 0), include.mean = FALSE, method = "ML")

sigma^2 estimated as 99.58: log likelihood = -163.65, aic = 329.31

**As AR(1):**

> Y<-arima(model$resid,order=c(1,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(1, 0, 0), include.mean = FALSE, method = "ML")

Coefficients:

ar1

-0.0180

s.e. 0.1549

sigma^2 estimated as 99.55: log likelihood = -163.65, aic = 331.3

**As AR(2):**

> Y<-arima(model$resid,order=c(2,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(2, 0, 0), include.mean = FALSE, method = "ML")

Coefficients:

ar1 ar2

0.0102 0.2563

s.e. 0.1499 0.1574

sigma^2 estimated as 93.65: log likelihood = -162.37, aic = 330.74

**As AR(3):**

> Y<-arima(model$resid,order=c(3,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(3, 0, 0), include.mean = FALSE, method = "ML")

Coefficients:

ar1 ar2 ar3

-0.0114 0.2644 0.0888

s.e. 0.1541 0.1571 0.1600

sigma^2 estimated as 92.93: log likelihood = -162.22, aic = 332.44

**As AR(4):**

> Y<-arima(model$resid,order=c(4,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(4, 0, 0), include.mean = FALSE, method = "ML")

Coefficients:

ar1 ar2 ar3 ar4

-0.0008 0.2952 0.0742 -0.1323

s.e. 0.1531 0.1612 0.1595 0.1577

sigma^2 estimated as 91.35: log likelihood = -161.87, aic = 333.74

**As AR(5):**

> Y<-arima(model$resid,order=c(5,0,0),method="ML",include.mean=FALSE)

> Y

Call:

arima(x = model$resid, order = c(5, 0, 0), include.mean = FALSE, method = "ML")

Coefficients:

ar1 ar2 ar3 ar4 ar5

0.0089 0.2893 0.0541 -0.1258 0.0745

s.e. 0.1540 0.1610 0.1645 0.1579 0.1579

sigma^2 estimated as 90.84: log likelihood = -161.76, aic = 335.52

**Conclusion**

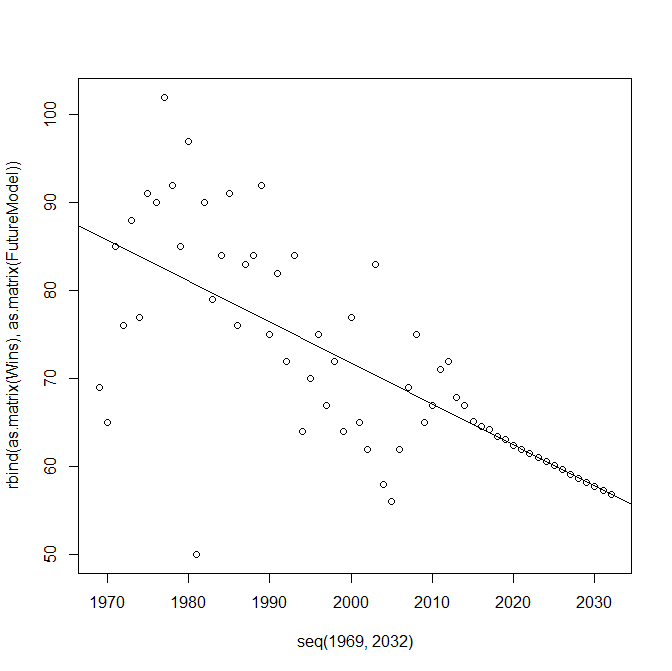
Given the 6 AR models from R, it seems at this point that the best model for the residuals of my data is actually White Noise, based on the increasing AIC values.

FutureWins<-predict(Y,n.ahead=20)

FutureModel<-model$coef[1]+model$coef[2]\*seq(2013,2032)+FutureWins$pred

plot(seq(1969,2032),rbind(as.matrix(Wins),as.matrix(FutureModel)))

abline(model)



Prediction for future number of wins:

> FutureModel

Time Series:

Start = 45

End = 64

Frequency = 1

[1] 67.85368 66.99966 65.16611 64.52429 64.21189 63.39651 63.10498 62.46749 61.99859

[10] 61.54729 61.02280 60.59444 60.09972 59.63861 59.17432 58.69859 58.23916 57.76723

[19] 57.30218 56.83567

Given this model, it appears that a White Noise model is perhaps not the best, given the drastic change in predicted values. The observed values clearly show an oscillating trend, given the plethora of variables involved in a game as complicated as professional baseball. Instead, this prediction model shows a neat, strictly decreasing linear trend, similar to the line of best fit for the observed number of wins on page 2. In the future, it would probably be better to attempt to fit an MA component to the model, despite my initial skepticism from the ACF graph.