models_pdf.rmd

Billiau/Eng/Li

12/6/2019

Set up

```
source('packages.R')
```

```
## Loading required package: optimx
## Loading required package: parallel
## Loading required package: minqa
## Loading required package: lme4
## Loading required package: Matrix
## Loading required package: segmented
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Loading required package: ggplot2
## Loading required package: hrbrthemes
## Loading required package: ggcorrplot
```

```
source('styleguide.R')
source('helpers.R')
source('cleaner.R')

# Read in Clean DF
df.clean <- add_time("complete_data_clean.csv")
df.clean <- add_coach_change(df.clean)
df.tourney <- add_time("tourney_data_clean.csv")
df.tourney <- add_coach_change(df.tourney)

# Check dimensions - len(unique schools) * len(unique years) must equal # of rows
dim_checker(df.clean)
```

```
## [1] "Dim Check Successful"
```

```
dim_checker(df.tourney)
```

```
## [1] "Dim Check Successful"
```

Models

Model 1: pool all teams together, OLS model for 3PAr change over time

```r
### Model 1: pool all teams together, OLS model for 3PAr change over time
lm1 <- lm(X3PAr ~ time, df.tourney)
summary(lm1)
```

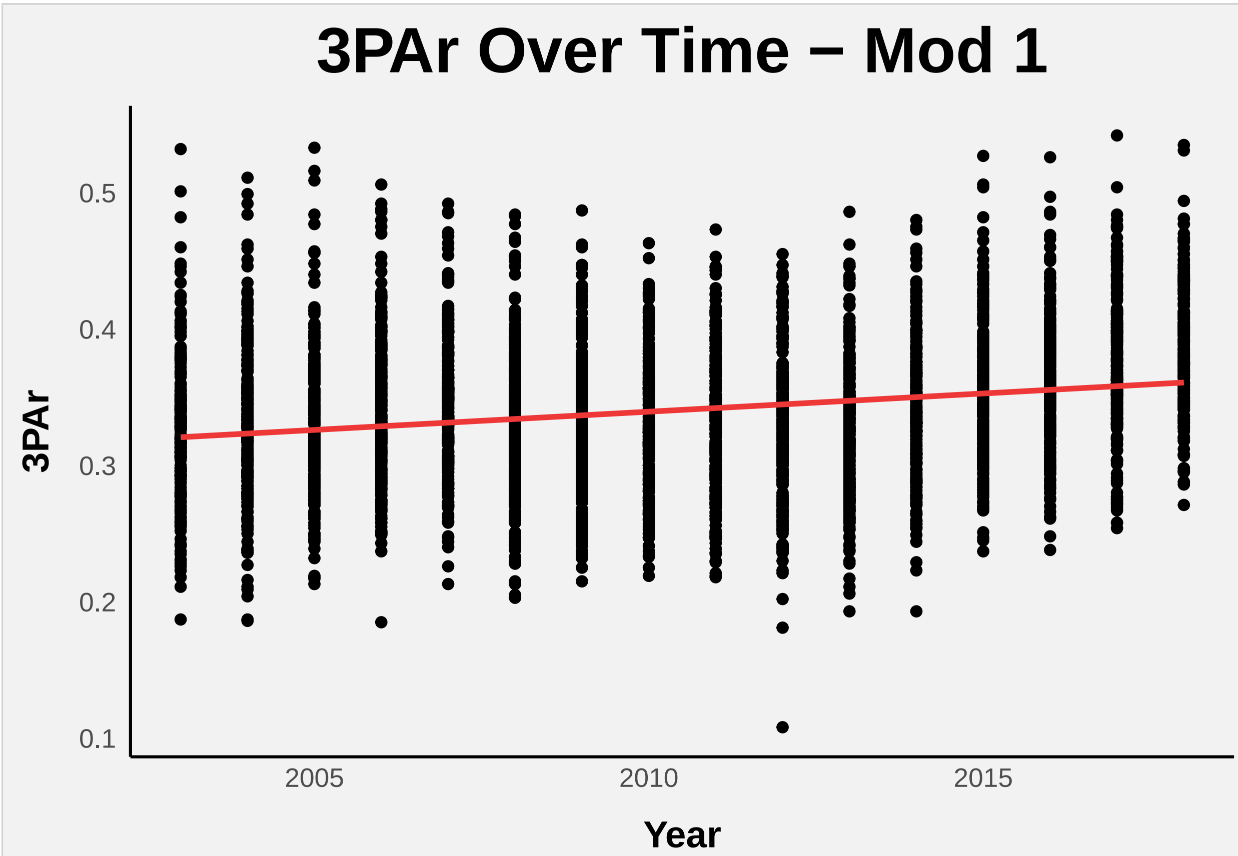
```
##
## Call:
## lm(formula = X3PAr ~ time, data = df.tourney)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.236756 -0.035506 -0.002073  0.032615  0.211303
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3206972  0.0016671  192.37   <2e-16 ***
## time        0.0026732  0.0001894   14.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05319 on 3710 degrees of freedom
## Multiple R-squared:  0.05098,    Adjusted R-squared:  0.05072
## F-statistic: 199.3 on 1 and 3710 DF,  p-value: < 2.2e-16
```

```r
# Model 1: Control for team no pooling, OLS model for 3PAr change over time
lm1a <- lm(X3PAr ~ time, df.tourney)
summary(lm1a)
```

```
##
## Call:
## lm(formula = X3PAr ~ time, data = df.tourney)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.236756 -0.035506 -0.002073  0.032615  0.211303
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3206972  0.0016671  192.37   <2e-16 ***
## time        0.0026732  0.0001894   14.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05319 on 3710 degrees of freedom
## Multiple R-squared:  0.05098,    Adjusted R-squared:  0.05072
## F-statistic: 199.3 on 1 and 3710 DF,  p-value: < 2.2e-16
```

```r
p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
  geom_point() +
  stat_smooth(method = "lm", col = '#EE3838', se = F) +
  labs(title="3PAr Over Time - Mod 1") +
  xlab("Year") +
```

```r
  ylab("3PAr") +
  #ylim(c(0,0.6)) +
  theme_hodp()
p
```



Model 2 & 3

Model 2 - Mixed Model, fixed effect of time, random intercept stratified on School Model 3 - Mixed Model, fixed effect of time, random slopes and intercept stratified on School

Includes model comparison

```r
lmer2 <- lmer(X3PAr ~ time + (1 | School), data=df.tourney)
summary(lmer2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: X3PAr ~ time + (1 | School)
##    Data: df.tourney
##
## REML criterion at convergence: -11988
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5054 -0.6539 -0.0153  0.6318  3.6385
##
## Random effects:
```

```
##  Groups   Name        Variance  Std.Dev.
##  School   (Intercept) 0.0007982 0.02825
##  Residual             0.0020335 0.04509
## Number of obs: 3712, groups:  School, 232
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 0.3206972  0.0023321  137.52
## time        0.0026732  0.0001606   16.65
##
## Correlation of Fixed Effects:
##      (Intr)
## time -0.516
```

```r
head(coef(summary(lmer2)))
```

```
##                Estimate   Std. Error   t value
## (Intercept) 0.320697167 0.0023320635 137.51648
## time        0.002673244 0.0001605606  16.64944
```

```r
head(coef(lmer2)$School)
```

```
##                     (Intercept)        time
## Air Force             0.3935939 0.002673244
## Akron                 0.3714346 0.002673244
## Alabama               0.3106719 0.002673244
## Alabama A&M           0.3017759 0.002673244
## Alabama State         0.3134216 0.002673244
## Alabama-Birmingham    0.3160095 0.002673244
```

```r
# split into categories -- > see if there are
# binary indicator as for if they were a winning team or losing team for most of the seasons

#Fitting a random slopes, random intercepts model may fail to converge
lmer3 <- lmer(X3PAr ~ time + (1  + time|School), data=df.tourney) # fails to converge

### Nelder_Mead converges successfully!!! - but only for the df.tourney
lmer3d <- update(lmer3, control=lmerControl(optimizer="Nelder_Mead"))

# Summary
summary(lmer3d)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: X3PAr ~ time + (1 + time | School)
##    Data: df.tourney
## Control: lmerControl(optimizer = "Nelder_Mead")
##
## REML criterion at convergence: -12175.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7320 -0.6343  0.0068  0.6308  3.4423
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  School   (Intercept) 1.641e-03 0.040505
```

```
##            time        1.072e-05 0.003274 -0.72
##  Residual              1.792e-03 0.042327
## Number of obs: 3712, groups:  School, 232
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 0.3206972  0.0029719  107.91
## time        0.0026732  0.0002625   10.18
##
## Correlation of Fixed Effects:
##      (Intr)
## time -0.745
```

```r
### COMPARE
# Fixed coefs
coef(summary(lmer2))
```

```
##                Estimate   Std. Error   t value
## (Intercept) 0.320697167 0.0023320635 137.51648
## time        0.002673244 0.0001605606  16.64944
```

```r
coef(summary(lmer3d))
```

```
##                Estimate   Std. Error   t value
## (Intercept) 0.320697167 0.0029718513 107.91158
## time        0.002673244 0.0002624929  10.18406
```

```r
# Look at differences b/t individual schools coefs
head(coef(lmer2)$School)
```

```
##                      (Intercept)        time
## Air Force              0.3935939 0.002673244
## Akron                  0.3714346 0.002673244
## Alabama                0.3106719 0.002673244
## Alabama A&M            0.3017759 0.002673244
## Alabama State          0.3134216 0.002673244
## Alabama-Birmingham     0.3160095 0.002673244
```

```r
head(coef(lmer3d)$School)
```

```
##                      (Intercept)         time
## Air Force              0.4520123 -0.004791179
## Akron                  0.3402685  0.006838891
## Alabama                0.2994755  0.004111946
## Alabama A&M            0.3073300  0.001910670
## Alabama State          0.3168367  0.002213276
## Alabama-Birmingham     0.3354296  0.000132666
```

```r
# Unsurprisingly, our random slopes and intercepts model is significantly better than
# our simple random intercepts model. It may be even more overfit though.
anova(lmer2,lmer3d)
```

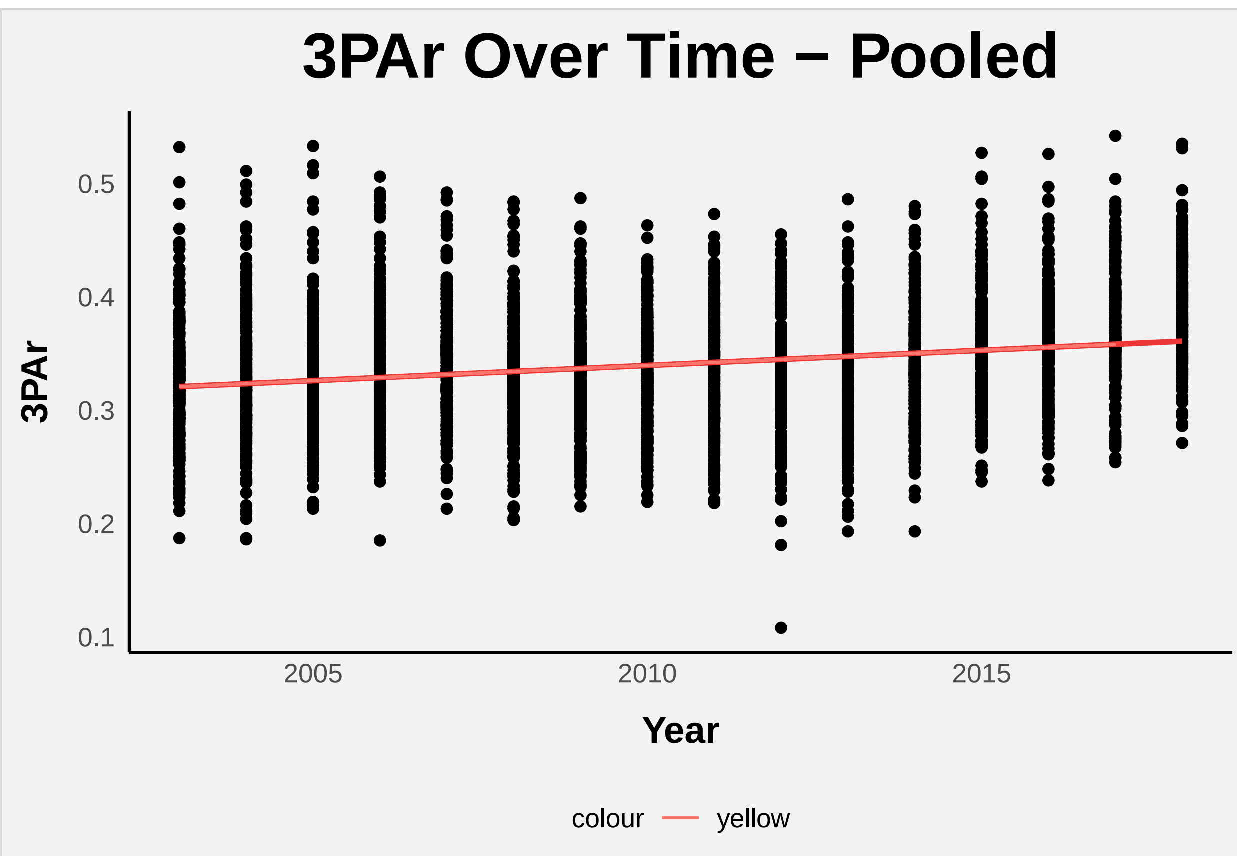
```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df.tourney
## Models:
## lmer2: X3PAr ~ time + (1 | School)
## lmer3d: X3PAr ~ time + (1 + time | School)
##        Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
```

```
## lmer2   4 -12006 -11981 6007.1   -12014
## lmer3d  6 -12189 -12152 6100.5   -12201 186.85      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Let's do some plots
# Get coefficients
year <- 2003:2017
intercept.mm <- summary(lmer3d)$coef[1,1]
slope.mm <- summary(lmer3d)$coef[2,1]
lmer3fn <- function(year) {
  return(intercept.mm + (year - 2003) * slope.mm)
}

p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
  geom_point() +
  stat_smooth(method = "lm", col = '#EE3838', se = F) +
  geom_segment(aes(x = 2003, y = lmer3fn(2003), xend = 2017, yend = lmer3fn(2017), colour = "yellow"),
               data = df.tourney) +
  labs(title="3PAr Over Time - Pooled") +
  xlab("Year") +
  ylab("3PAr") +
  #ylim(c(0,0.6)) +
  theme_hodp()
p
```



```r
# Very similar fixed effects vs. Completely pooled OLS
```

SEGMENTED REGRESSION with OLS

Using the segmented package to explore segmenting the OLS model

```r
# Using the segmented package
# have to provide estimates for breakpoints.
# apriori guess of 3 based on when the rule change was announced,
seg4 <- segmented(lm1,
                  seg.Z = ~ time,
                  psi =c(3,10))
# display the summary
summary(seg4)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = lm1, seg.Z = ~time, psi = c(3, 10))
##
## Estimated Break-Point(s):
##             Est. St.Err
## psi1.time 3.353  0.685
## psi2.time 9.708  0.276
##
## Meaningful coefficients of the linear terms:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.326258   0.002842 114.808   <2e-16 ***
## time         0.003859   0.001519   2.541   0.0111 *
## U1.time     -0.006405   0.001722  -3.719       NA
## U2.time      0.014392   0.001148  12.534       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05174 on 3706 degrees of freedom
## Multiple R-Squared: 0.103,  Adjusted R-squared: 0.1018
##
## Convergence attained in 2 iter. (rel. change 0)
```

```r
# get breakpoints
seg4$psi
```

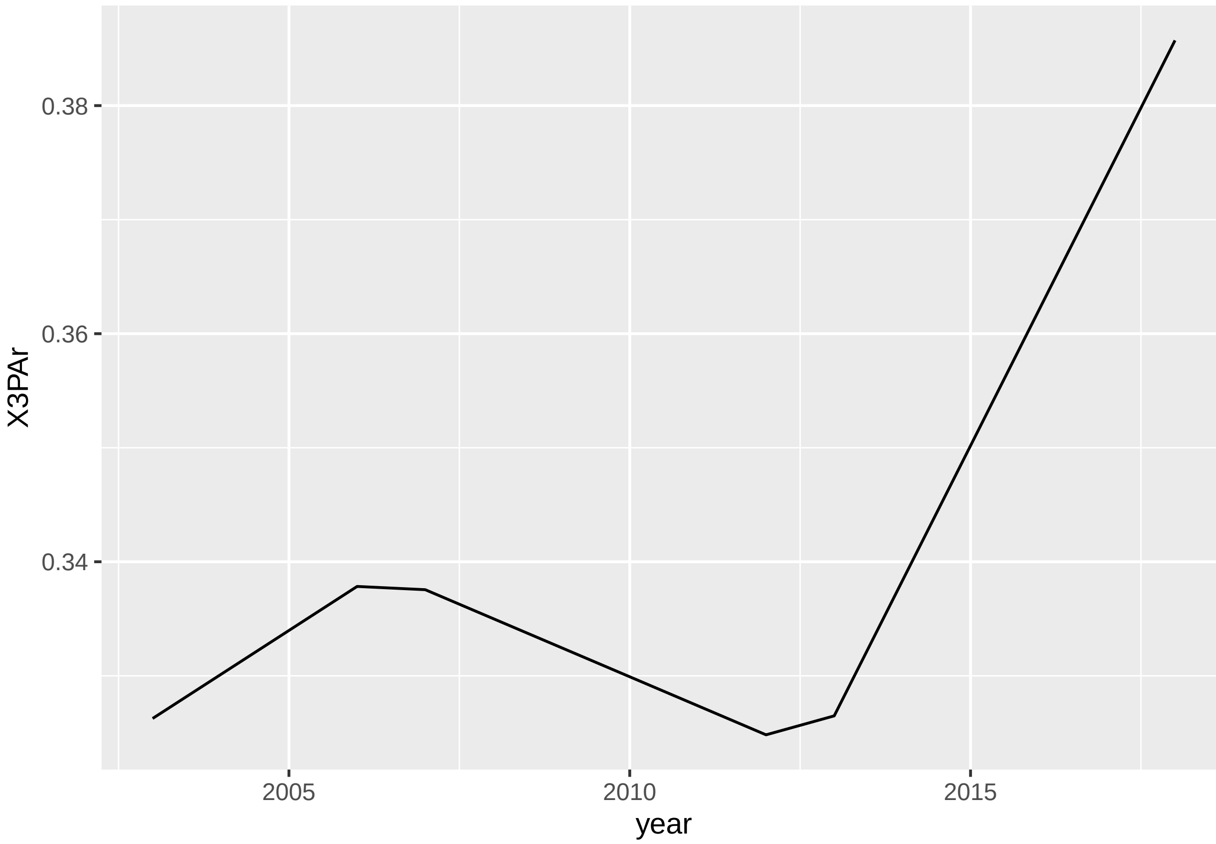
```
##           Initial     Est.    St.Err
## psi1.time       3 3.352943 0.6851873
## psi2.time      10 9.707727 0.2759217
```

```r
# get the slopes
head(slope(seg4))
```

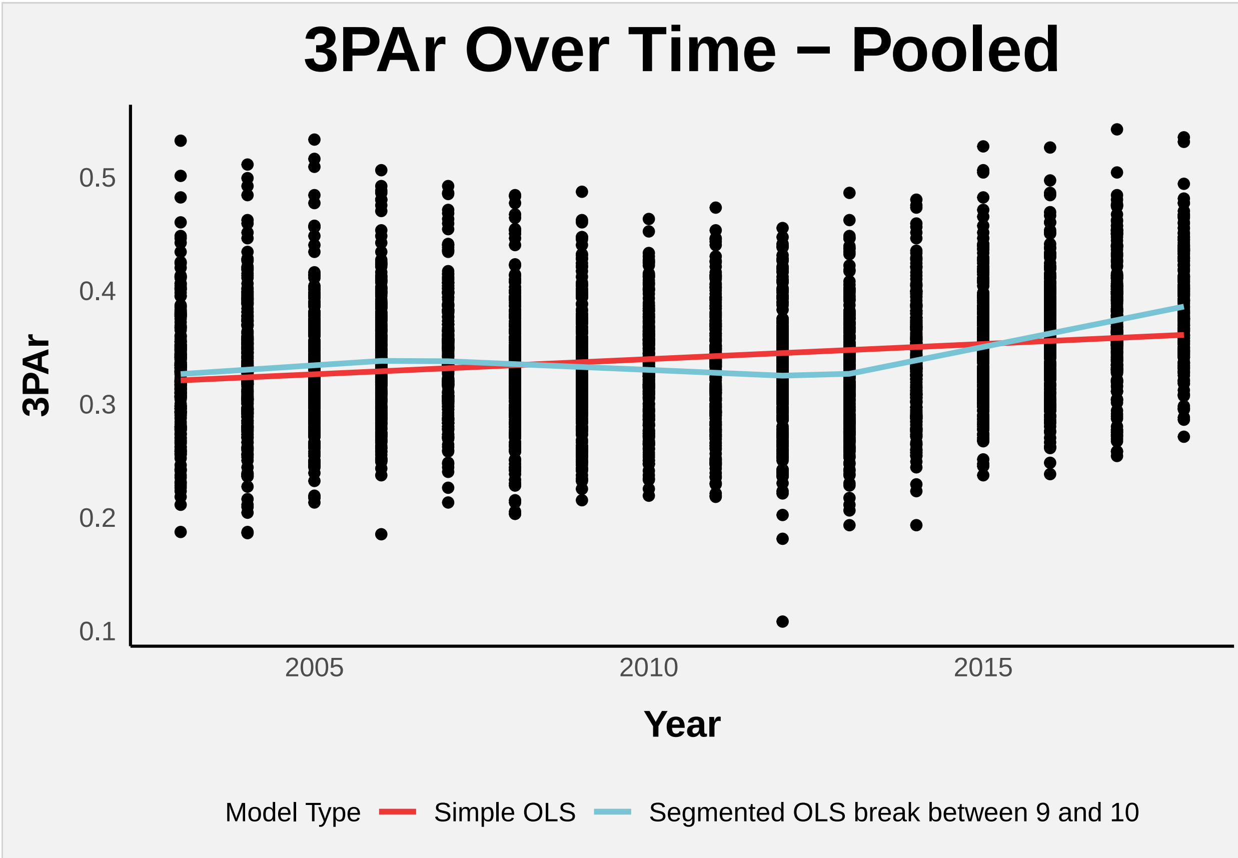
```
## $time
##              Est.    St.Err. t value   CI(95%).l   CI(95%).u
## slope1  0.0038595 0.00151900  2.5408  0.00088134  0.00683760
## slope2 -0.0025452 0.00081194 -3.1347 -0.00413710 -0.00095331
## slope3  0.0118470 0.00081194 14.5900  0.01025500  0.01343800
```

```r
# get the fitted data
my.fitted <- fitted(seg4)
my.model <- data.frame(year = df.tourney$year, X3PAr = my.fitted)

# plot the fitted model
ggplot(my.model, aes(x = year, y = X3PAr)) + geom_line()
```




```r
# Replot things
cols <- c("Simple OLS"='#EE3838',"Segmented OLS"='#78C4D4')
p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
  geom_point() +
  stat_smooth(method = "lm", aes(col = '#EE3838'), se = F,size=1) +
  geom_line(data = my.model, aes(x = year, y = X3PAr, color = '#78C4D4'),
            linetype = "solid", size=1) +
  scale_colour_identity(name="Model Type",
                        breaks = c('#EE3838','#78C4D4'),
                        labels = c("Simple OLS", "Segmented OLS break between 9 and 10"),
                        guide = "legend")  +
  labs(title="3PAr Over Time - Pooled") +
  xlab("Year") +
  ylab("3PAr") +
  theme_hodp()
p
```




```r
# Method Test if breakpoints are significant
davies.test(lm1, ~time)
```

```
##
##  Davies' test for a change in the slope
##
## data:  formula = X3PAr ~ time ,   method = lm
## model = gaussian , link = identity
## segmented variable = time
## 'best' at = 10, n.points = 8, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```r
seg6 <- segmented(lm1,
                  seg.Z = ~ time,
                  psi = list(time = c(9.3)))
# Check for existence of one breakpoint using the pscore.test command

davies.test(seg6, ~time)
```

```
##
##  Davies' test for a change in the slope
##
## data:  formula = X3PAr ~ time + U1.time ,   method = segmented.lm
## model = gaussian , link = identity
## segmented variable = time
## 'best' at = 3.3333, n.points = 8, p-value = 0.001722
## alternative hypothesis: two.sided
```

```r
seg7 <- segmented(lm1,
                  seg.Z = ~ time,
                  psi = list(time = c(3.1, 9.3)))
summary(seg7)
```

```
##
##  ***Regression Model with Segmented Relationship(s)***
##
## Call:
## segmented.lm(obj = lm1, seg.Z = ~time, psi = list(time = c(3.1,
##     9.3)))
##
## Estimated Break-Point(s):
##             Est. St.Err
## psi1.time 3.353  0.685
## psi2.time 9.708  0.276
##
## Meaningful coefficients of the linear terms:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.326258   0.002842 114.808   <2e-16 ***
## time         0.003859   0.001519   2.541   0.0111 *
## U1.time     -0.006405   0.001722  -3.719       NA
## U2.time      0.014392   0.001148  12.534       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05174 on 3706 degrees of freedom
## Multiple R-Squared: 0.103,  Adjusted R-squared: 0.1018
##
## Convergence attained in 2 iter. (rel. change 1.773e-16)
```

```r
davies.test(seg7, ~time)
```

```
##
##  Davies' test for a change in the slope
##
## data:  formula = X3PAr ~ time + U1.time + U2.time ,   method = segmented.lm
## model = gaussian , link = identity
## segmented variable = time
## 'best' at = 5, n.points = 8, p-value = 0.4164
## alternative hypothesis: two.sided
```

```r
# 2 Breakpoints
# between the 2006-2007 and 2007-08 seasons - Rule change was announced in May 2007
# https://www.espn.com/mens-college-basketball/news/story?id=2859065

# Then another, more significant breakpoint between 2012-2013 and 2013-2014 seasons
# Curry sets record for NBA 3's in 2012-13
# get breakpoints
seg7$psi
```

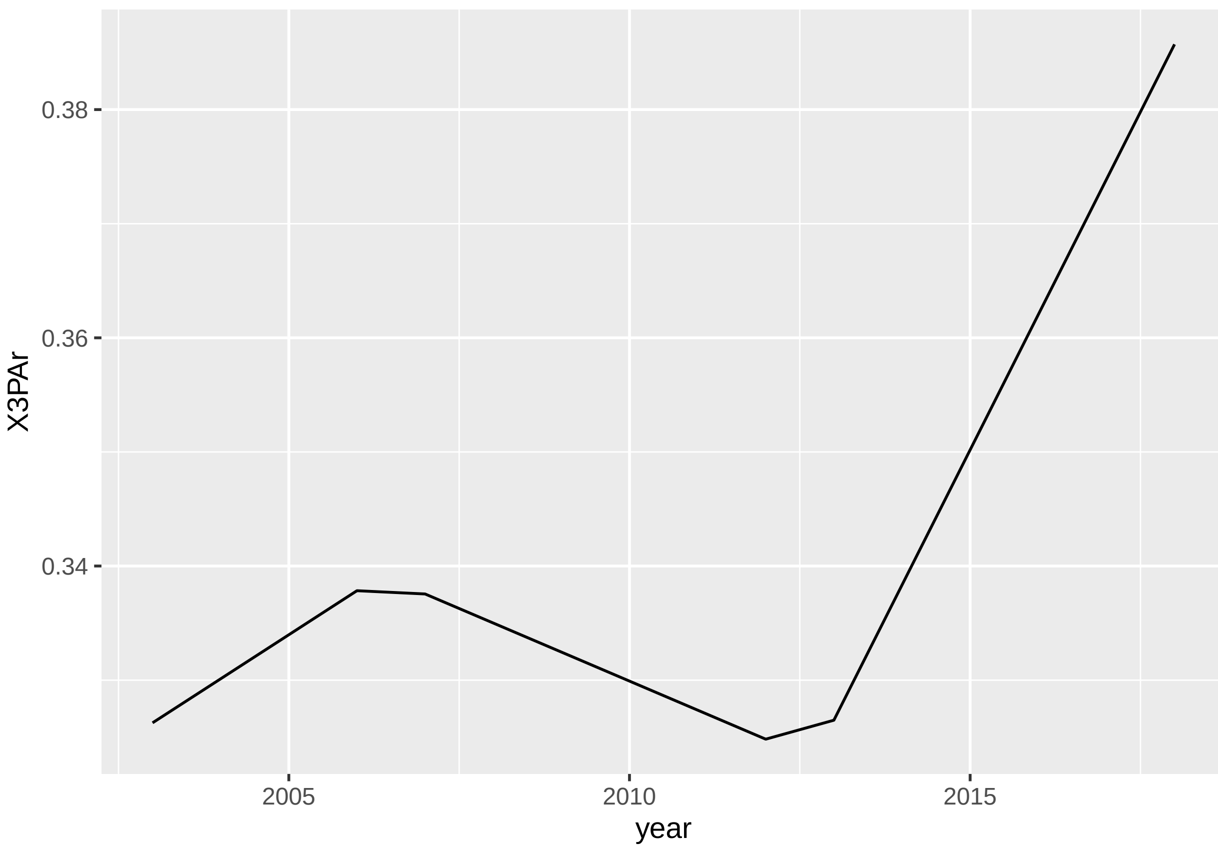
```
##           Initial     Est.    St.Err
## psi1.time     3.1 3.352943 0.6851873
## psi2.time     9.3 9.707727 0.2759217
```

```r
# get the slopes
slope(seg7)
```

```
## $time
##              Est.    St.Err. t value   CI(95%).l   CI(95%).u
## slope1  0.0038595 0.00151900  2.5408  0.00088134  0.00683760
## slope2 -0.0025452 0.00081194 -3.1347 -0.00413710 -0.00095331
## slope3  0.0118470 0.00081194 14.5900  0.01025500  0.01343800
```

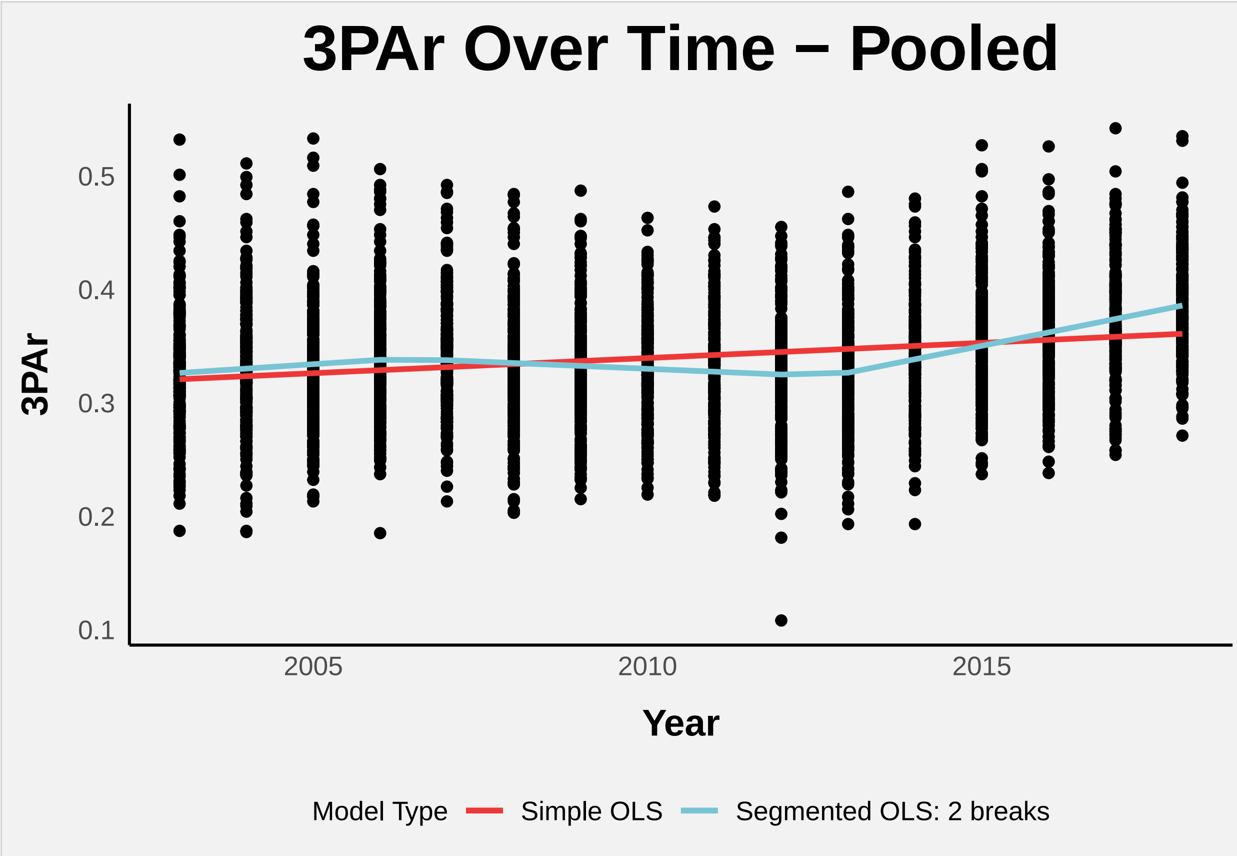
```r
# get the fitted data
seg7.fitted <- fitted(seg7)
seg7.fitted.df <- data.frame(year = df.tourney$year, X3PAr = seg7.fitted)

# plot the fitted model
ggplot(seg7.fitted.df, aes(x = year, y = X3PAr)) + geom_line()
```




```r
p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
  geom_point() +
  stat_smooth(method = "lm", aes(col = '#EE3838'), se = F,size=1) +
  geom_line(data = seg7.fitted.df, aes(x = year, y = X3PAr, color = '#78C4D4'),
          linetype = "solid", size=1) +
  scale_colour_identity(name="Model Type",
                      breaks = c('#EE3838','#78C4D4'),
                      labels = c("Simple OLS", "Segmented OLS: 2 breaks"),
                      guide = "legend")  +
  labs(title="3PAr Over Time - Pooled") +
  xlab("Year") +
  ylab("3PAr") +
```

```r
  theme_hodp()
p
```



Mixed model - segmented, unsuccessful aside exploring teams

Unsuccessful Aside playing around with same coach

```r
# Mixed Model Segmented - coaching change
lmer8 <- lmer(X3PAr ~ time*same.coach + (1  + time|School) , data=df.tourney) # fails to converge
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00304692
## (tol = 0.002, component 1)
```

```r
# Use all fit to find a model that converges
# Source: https://joshua-nugent.github.io/allFit/
ncores <- detectCores()
diff_optims <- allFit(lmer8, maxfun = 1e6, parallel = 'multicore', ncpus = ncores)
```

```
## Loading required namespace: dfoptim
```

```r
is.OK <- sapply(diff_optims, is, "merMod")
diff_optims.OK <- diff_optims[is.OK]
lapply(diff_optims.OK,function(x) x@optinfo$conv$lme4$messages)
```

```
## $bobyqa
## NULL
```

```
##
## $Nelder_Mead
## NULL
##
## $nlminbwrap
## NULL
##
## $nmkbw
## NULL
##
## $`optimx.L-BFGS-B`
## [1] "Model failed to converge with max|grad| = 0.00337925 (tol = 0.002, component 1)"
##
## $nloptwrap.NLOPT_LN_NELDERMEAD
## [1] "Model failed to converge with max|grad| = 0.00304692 (tol = 0.002, component 1)"
##
## $nloptwrap.NLOPT_LN_BOBYQA
## [1] "Model failed to converge with max|grad| = 0.00304692 (tol = 0.002, component 1)"
```

```
# Nelder_Mead for convergence
lmer8a <- update(lmer8, control=lmerControl(optimizer="Nelder_Mead"))

# Summary
summary(lmer8a)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: X3PAr ~ time * same.coach + (1 + time | School)
##    Data: df.tourney
## Control: lmerControl(optimizer = "Nelder_Mead")
##
## REML criterion at convergence: -12159.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7300 -0.6386  0.0047  0.6307  3.4451
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  School   (Intercept) 1.637e-03 0.040458
##           time        1.078e-05 0.003284 -0.72
##  Residual             1.792e-03 0.042327
## Number of obs: 3712, groups:  School, 232
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)          0.3196693  0.0030915 103.404
## time                 0.0026603  0.0002739   9.713
## same.coachTRUE       0.0132478  0.0110987   1.194
## time:same.coachTRUE  0.0001665  0.0009833   0.169
##
## Correlation of Fixed Effects:
##             (Intr) time   s.TRUE
## time        -0.748
## sam.cchTRUE -0.279  0.208
## tm:sm.cTRUE  0.208 -0.279 -0.748
```

```r
### COMPARE
# Fixed coefs
coef(summary(lmer8a))
```

```
##                        Estimate   Std. Error     t value
## (Intercept)         0.3196693238 0.0030914586 103.4040457
## time                0.0026603285 0.0002738848   9.7133119
## same.coachTRUE      0.0132477514 0.0110986759   1.1936335
## time:same.coachTRUE 0.0001664689 0.0009832765   0.1693002
```

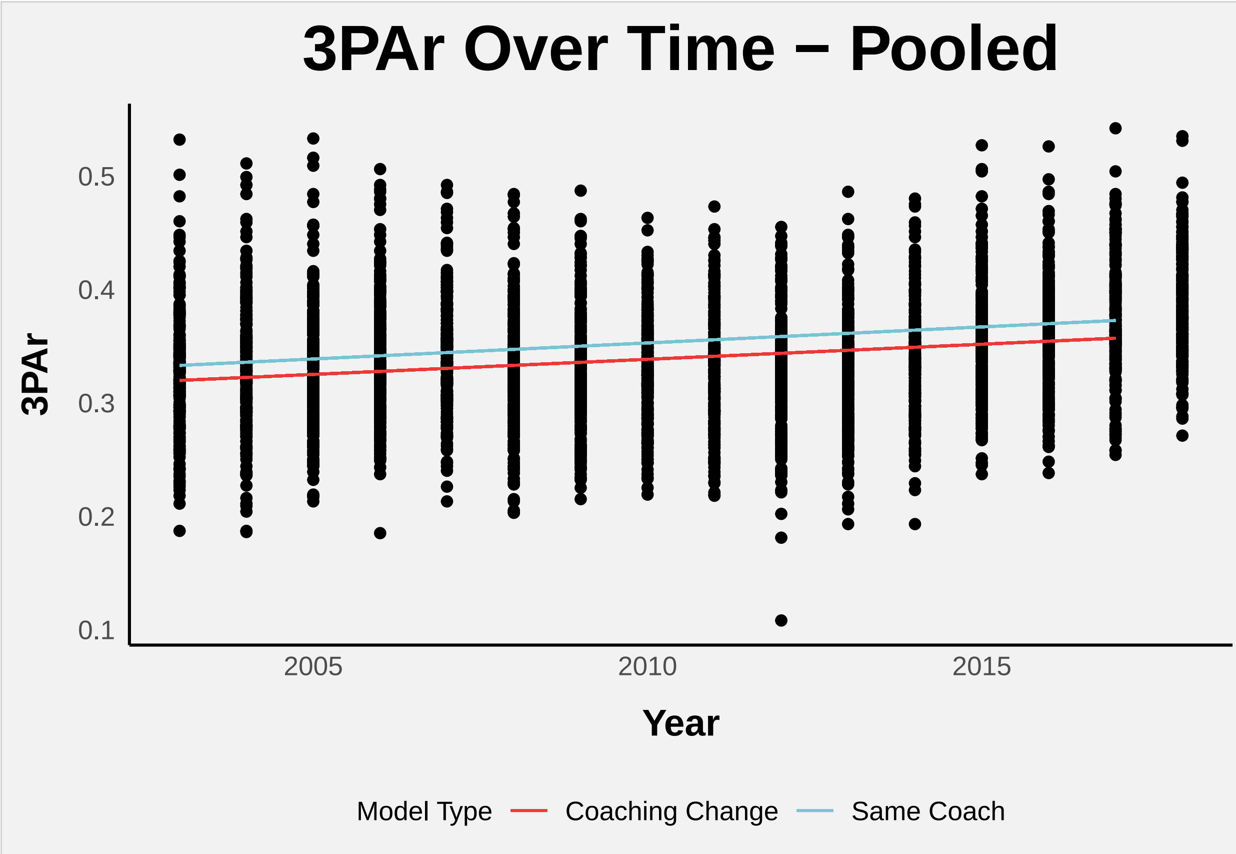
```r
# Look at differences b/w individual schools coefs
head(coef(lmer8a)$School)
```

```
##                     (Intercept)          time same.coachTRUE
## Air Force             0.4520259 -0.0048256252     0.01324775
## Akron                 0.3400927  0.0068285569     0.01324775
## Alabama               0.2993953  0.0041054275     0.01324775
## Alabama A&M           0.3072958  0.0019014669     0.01324775
## Alabama State         0.3167881  0.0022029461     0.01324775
## Alabama-Birmingham    0.3354160  0.0001182198     0.01324775
##                     time:same.coachTRUE
## Air Force                  0.0001664689
## Akron                      0.0001664689
## Alabama                    0.0001664689
## Alabama A&M                0.0001664689
## Alabama State              0.0001664689
## Alabama-Birmingham         0.0001664689
```

```r
### Let's do some plots
# Get coefficients
year <- 2003:2017
intercept.false <- summary(lmer8a)$coef[1,1]
slope.false <- summary(lmer8a)$coef[2,1]
intercept.true <- summary(lmer8a)$coef[3,1]
slope.true <- summary(lmer8a)$coef[4,1]
lmer8fn <- function(year, true_flag) {
  return(intercept.false + (year - 2003) * slope.false +
           intercept.true*true_flag + slope.true * (year - 2003) * true_flag)
}

p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
  geom_point() +
  geom_segment(aes(x = 2003, y = lmer8fn(2003,0), xend = 2017, yend = lmer8fn(2017,0),
                   colour = '#EE3838'),
               data = df.tourney) +
  geom_segment(aes(x = 2003, y = lmer8fn(2003,1), xend = 2017, yend = lmer8fn(2017,1),
                   colour = '#78C4D4'),
               data = df.tourney) +
  scale_colour_identity(name="Model Type",
                        breaks = c('#EE3838','#78C4D4'),
                        labels = c("Coaching Change", "Same Coach"),
                        guide = "legend") +
  labs(title="3PAr Over Time - Pooled") +
  xlab("Year") +
  ylab("3PAr") +
```

```
  theme_hodp()
p
```



Model 9 - Final mode

Mixed model by era

```
df.tourney$era <- as.factor((df.tourney$year > 2006) + (df.tourney$year > 2012))

# Mixed Model Segmented - coaching change
lmer9 <- lmer(X3PAr ~ time*era + (1  + time|School), data=df.tourney) # fails to converge
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00938467
## (tol = 0.002, component 1)
```

```
# Use all fit to find a model that converges
# Source: https://joshua-nugent.github.io/allFit/
ncores <- detectCores()
diff_optims <- allFit(lmer9, maxfun = 1e6, parallel = 'multicore', ncpus = ncores)
is.OK <- sapply(diff_optims, is, "merMod")
diff_optims.OK <- diff_optims[is.OK]
lapply(diff_optims.OK,function(x) x@optinfo$conv$lme4$messages)
```

```
## $bobyqa
## NULL
```

```
##
## $Nelder_Mead
## NULL
##
## $nlminbwrap
## NULL
##
## $nmkbw
## [1] "Model failed to converge with max|grad| = 0.00829571 (tol = 0.002, component 1)"
##
## $`optimx.L-BFGS-B`
## [1] "Model failed to converge with max|grad| = 0.00335849 (tol = 0.002, component 1)"
##
## $nloptwrap.NLOPT_LN_NELDERMEAD
## [1] "Model failed to converge with max|grad| = 0.00938467 (tol = 0.002, component 1)"
##
## $nloptwrap.NLOPT_LN_BOBYQA
## [1] "Model failed to converge with max|grad| = 0.00938467 (tol = 0.002, component 1)"
```

```
### T tests to determine whether or not slopes are significantly different
# https://influentialpoints.com/Training/simple_linear_regression-principles-properties-assumptions.htm

# Nelder_Mead for convergence
lmer9a <- update(lmer9, control=lmerControl(optimizer="Nelder_Mead"))
summary(lmer9a)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: X3PAr ~ time * era + (1 + time | School)
##    Data: df.tourney
## Control: lmerControl(optimizer = "Nelder_Mead")
##
## REML criterion at convergence: -12470.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4691 -0.6182  0.0001  0.6032  3.8625
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  School   (Intercept) 1.681e-03 0.040994
##           time        1.123e-05 0.003351 -0.72
##  Residual             1.616e-03 0.040205
## Number of obs: 3712, groups:  School, 232
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.326258   0.003482  93.712
## time         0.003859   0.001201   3.214
## era1         0.021475   0.004781   4.491
## era2        -0.118237   0.008261 -14.312
## time:era1   -0.006405   0.001339  -4.785
## time:era2    0.007987   0.001339   5.967
##
## Correlation of Fixed Effects:
##           (Intr) time   era1   era2   tim:r1
```

```
## time      -0.602
## era1      -0.293  0.364
## era2      -0.170  0.211  0.123
## time:era1  0.449 -0.867 -0.731 -0.189
## time:era2  0.449 -0.867 -0.327 -0.639  0.778
```

```r
# draws the mean number of threes point attempts per year across years

### Compare

### COMPARE
# Fixed coefs
coef(summary(lmer9a))
```

```
##                 Estimate  Std. Error    t value
## (Intercept)  0.326258190 0.003481512  93.711649
## time         0.003859483 0.001200790   3.214121
## era1         0.021474528 0.004781192   4.491459
## era2        -0.118236638 0.008261211 -14.312264
## time:era1   -0.006404680 0.001338516  -4.784912
## time:era2    0.007987069 0.001338516   5.967109
```

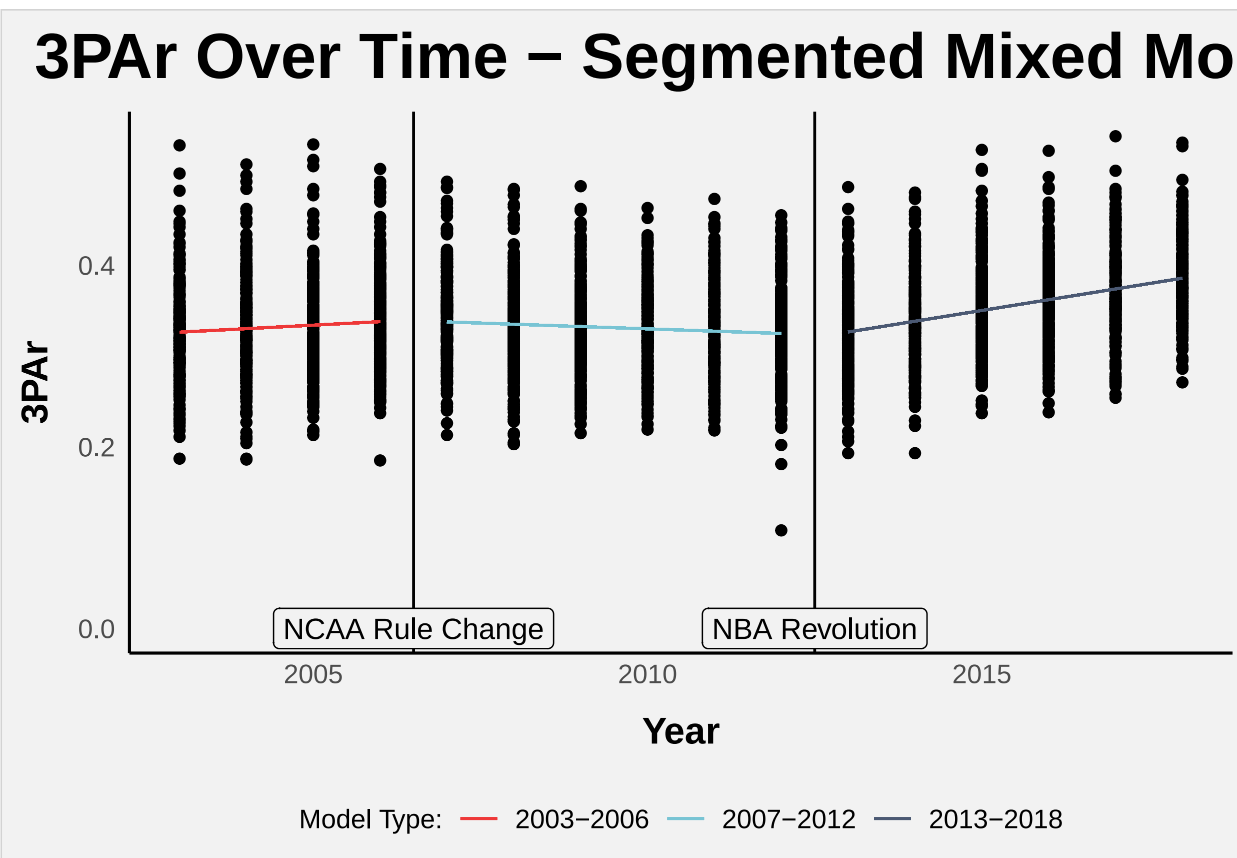
```r
# Look at differences b/w individual schools coefs
head(coef(lmer9a)$School)
```

```
##                     (Intercept)         time       era1       era2
## Air Force             0.4607191 -0.003904832 0.02147453 -0.1182366
## Akron                 0.3447244  0.008276783 0.02147453 -0.1182366
## Alabama               0.3044588  0.005359697 0.02147453 -0.1182366
## Alabama A&M           0.3130248  0.003041958 0.02147453 -0.1182366
## Alabama State         0.3225078  0.003370185 0.02147453 -0.1182366
## Alabama-Birmingham    0.3418515  0.001189635 0.02147453 -0.1182366
##                       time:era1   time:era2
## Air Force           -0.00640468 0.007987069
## Akron               -0.00640468 0.007987069
## Alabama             -0.00640468 0.007987069
## Alabama A&M         -0.00640468 0.007987069
## Alabama State       -0.00640468 0.007987069
## Alabama-Birmingham -0.00640468 0.007987069
```

```r
### Let's do some plots
# Get coefficients
year <- 2003:2017
intercept.0 <- summary(lmer9a)$coef[1,1]
slope.0 <- summary(lmer9a)$coef[2,1]
intercept.1 <- summary(lmer9a)$coef[3,1]
intercept.2 <- summary(lmer9a)$coef[4,1]
slope.1 <- summary(lmer9a)$coef[5,1]
slope.2 <- summary(lmer9a)$coef[6,1]
lmer9fn <- function(year, era1, era2) {
  return(intercept.0 + (year - 2003) * slope.0 +
           intercept.1*era1 + slope.1 * (year - 2003) * era1 +
           intercept.2*era2 + slope.2 * (year - 2003) * era2)
}

p <- ggplot(df.tourney, aes(x = time + 2003, y = X3PAr)) +
```

```
  geom_point() +
  geom_segment(aes(x = 2003, y = lmer9fn(2003,0,0), xend = 2006, yend = lmer9fn(2006,0,0),
                   colour = '#EE3838'),
               data = df.tourney) +
  geom_segment(aes(x = 2007, y = lmer9fn(2007,1,0), xend = 2012, yend = lmer9fn(2012,1,0),
                   colour = '#78C4D4'),
               data = df.tourney) +
  geom_segment(aes(x = 2013, y = lmer9fn(2013,0,1), xend = 2018, yend = lmer9fn(2018,0,1),
                   colour = '#4B5973'),
               data = df.tourney) +
  geom_vline(xintercept = 2012.5) +
  geom_vline(xintercept = 2006.5) +
  scale_colour_identity(name="Model Type:",
                        breaks = c('#EE3838','#78C4D4','#4B5973'),
                        labels = c("2003-2006", "2007-2012", "2013-2018"),
                        guide = "legend") +
  annotate(geom="label", x = 2012.5, y = 0, label = "NBA Revolution", fill ="#F2F2F2", color = "black")
  annotate(geom="label", x = 2006.5, y = 0, label = "NCAA Rule Change", fill ="#F2F2F2", color = "black
  labs(title="3PAr Over Time - Segmented Mixed Model") +
  xlab("Year") +
  ylab("3PAr") +
  theme_hodp()
p
```




```
# lmer9a vs. lmer3d anova test
anova(lmer3d, lmer9a)
```

```
## refitting model(s) with ML (instead of REML)

## Data: df.tourney
## Models:
## lmer3d: X3PAr ~ time + (1 + time | School)
## lmer9a: X3PAr ~ time * era + (1 + time | School)
##        Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer3d  6 -12189 -12152 6100.5   -12201
## lmer9a 10 -12519 -12457 6269.6   -12539 338.14      4  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Contrast t-tests

```r
# Test if slopes are significantly different
# Create a vector of coefficients to test if difference between slope for
# era0*time and era1*time is 0 or not
coefs <- summary(lmer9a)$coef[,1]

# Construct our vector of differences C
C = c(0,0,0,0,-1,0)
contrast_test_lmer9a(C,coefs)
```

```
## $tstat
## 1 x 1 Matrix of class "dgeMatrix"
##          [,1]
## [1,] 4.784912
##
## $pval
## [1] 1.777425e-06
```

```r
# slopes are different

# Test if difference between slope for era0*time and era2*time is 0 or not
# Construct our vector of differences C
C = c(0,0,0,0,0,-1)
contrast_test_lmer9a(C,coefs)
```

```
## $tstat
## 1 x 1 Matrix of class "dgeMatrix"
##           [,1]
## [1,] -5.967109
##
## $pval
## [1] 2.641975e-09
```

```r
# slopes are different

# Test if difference between slope for era0*time and era2*time is 0 or not
# Construct our vector of differences C
C = c(0,0,0,0,0,-1)
contrast_test_lmer9a(C,coefs)
```

```
## $tstat
## 1 x 1 Matrix of class "dgeMatrix"
```

```
##            [,1]
## [1,] -5.967109
##
## $pval
## [1] 2.641975e-09
```

```r
# slopes are different

# Test if difference between slope for era1*time and era2*time is 0 or not
# Construct our vector of differences C
C = c(0,0,0,0,1,-1)
contrast_test_lmer9a(C,coefs)
```

```
## $tstat
## 1 x 1 Matrix of class "dgeMatrix"
##            [,1]
## [1,] -16.12803
##
## $pval
## [1] 0
```

```r
# slopes are different

# Pairwise tests were performed. The problem of multiple comparisons is ignored here
# (a) because our pvalues are <<0.001 and (b) because we are only doing 3 tests to
# compare all of the slopes.



### AIC TABLE ###
# Make AIC Table
simple_OLS_AIC = AIC(lm1)
lmer2_AIC = AIC(lmer2)
lmer3d_AIC = AIC(lmer3d)
lin_seg_AIC = AIC(seg7)
mm_final_AIC = AIC(lmer9a)

AICs = c(simple_OLS_AIC,lmer2_AIC, lmer3d_AIC, lin_seg_AIC,mm_final_AIC)
titles = c("Simple OLS",
           "Simple Random Intercepts",
           "Simple Random Slopes and Intercepts",
           "Segmented OLS",
           "Segmented Random Slopes and Intercepts")
data.frame(titles, AICs)
```

```
##                                   titles      AICs
## 1                             Simple OLS -11243.57
## 2               Simple Random Intercepts -11979.96
## 3    Simple Random Slopes and Intercepts -12163.78
## 4                          Segmented OLS -11444.87
## 5 Segmented Random Slopes and Intercepts -12450.06
```

```r
########## EOF ##########
```