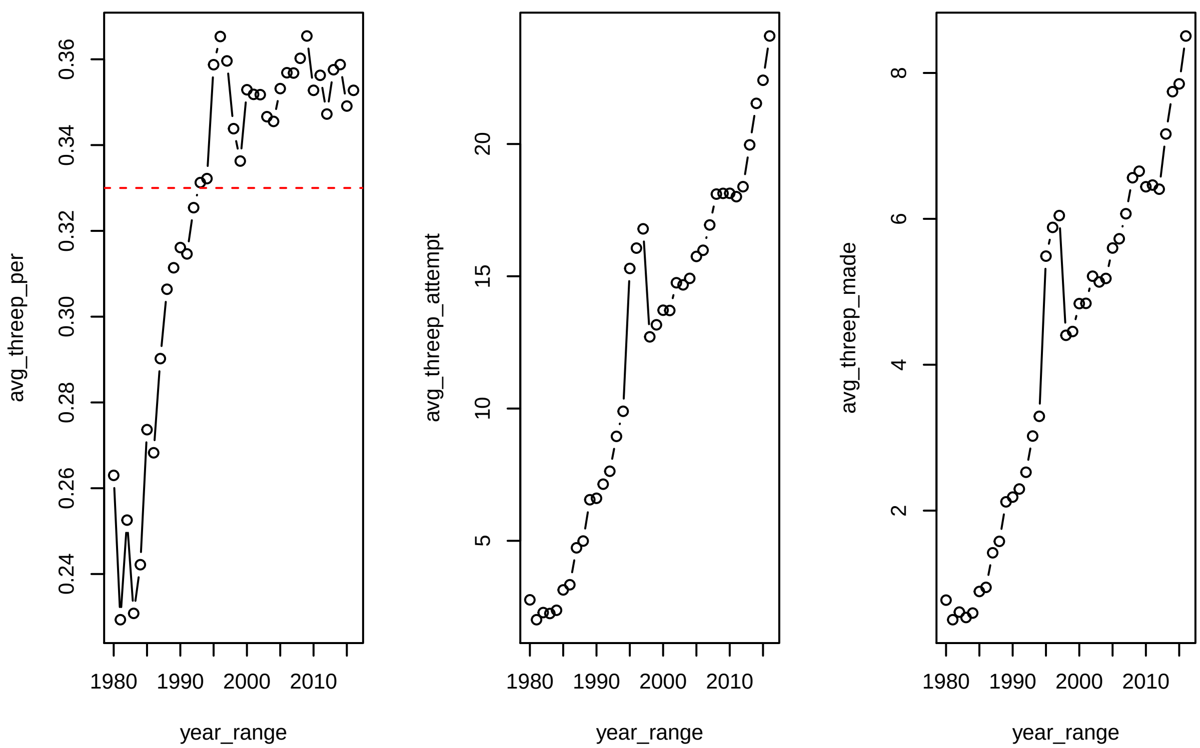
Three pointer time-series analysis

Loading data

```
threep_per_org <- read.csv('../../results/team_data/comb_3pper.csv')
year_range <- threep_per_org$X
threep_per <- subset(threep_per_org, select=-c(X))
threep_attempt_org <- read.csv('../../results/team_data/comb_3pattempt.csv')
year_range <- threep_attempt_org$X
threep_attempt <- subset(threep_attempt_org, select=-c(X))
threep_made_org <- read.csv('../../results/team_data/comb_3pmade.csv')
year_range <- threep_made_org$X
threep_made <- subset(threep_made_org, select=-c(X))
```

Plotting the mean Long term trend or fad?

```
avg_threep_per <- rowMeans(threep_per, na.rm=TRUE)
par(mfrow=c(1,3))
plot(year_range, avg_threep_per, type='b', lty=1)
abline(0.33,0, col="red", lty=2)
avg_threep_attempt <- rowMeans(threep_attempt, na.rm=TRUE)
plot(year_range, avg_threep_attempt, type='b', lty=1)
avg_threep_made <- rowMeans(threep_made, na.rm=TRUE)
plot(year_range, avg_threep_made, type='b', lty=1)
```

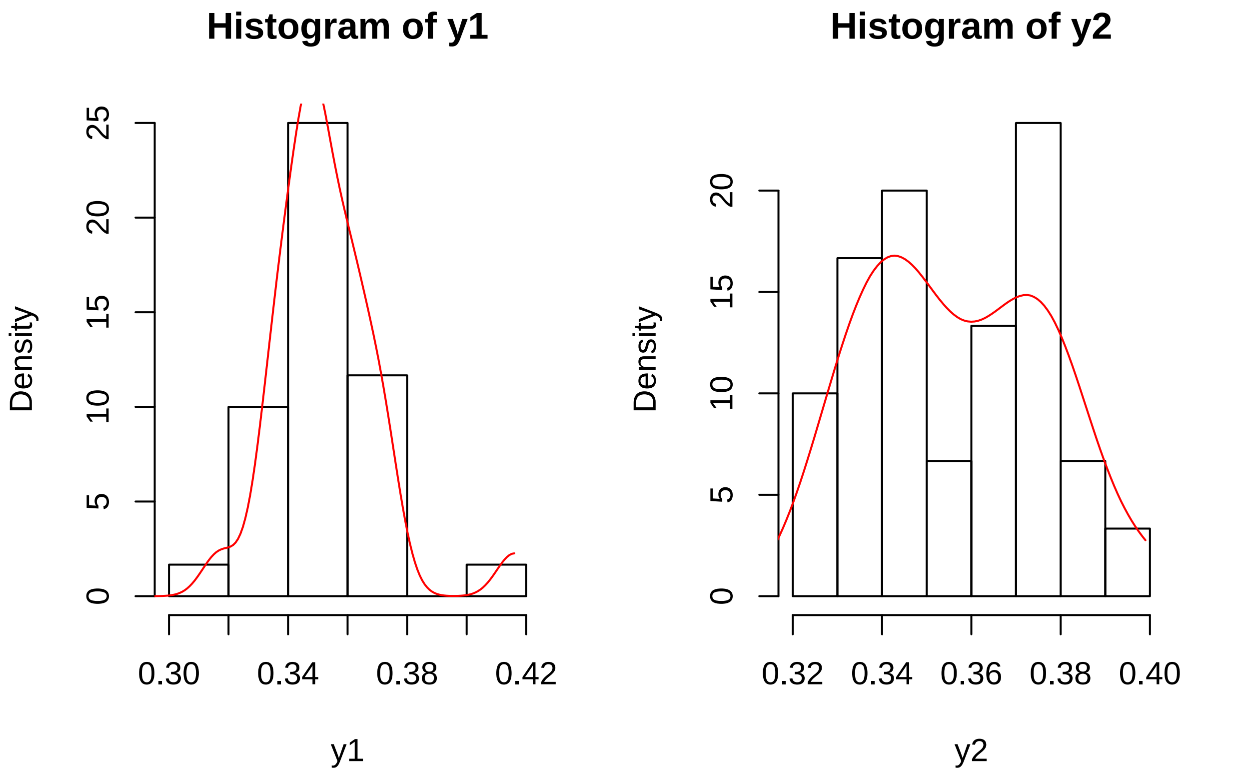


After crossing 0.33 percent it has largely saturated. Plot 2-pointers too

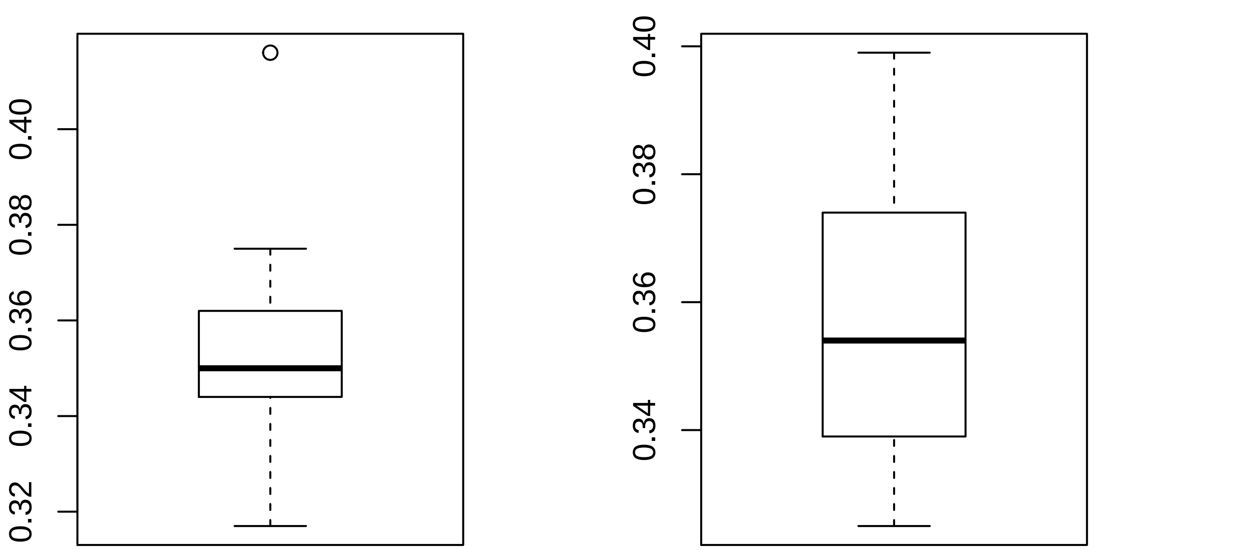
The golden state warrior outlier

```
par(mfrow=c(1,2))
y1 = as.numeric(threep_per[37,])
```

```
y2 = as.numeric(threep_per[27,])
hist(y1, freq=FALSE)
lines(density(y1, from = 0, to = max(y1)), col="red")
hist(y2, freq = FALSE)
lines(density(y2, from = 0, to = max(y2)), col="red")
```

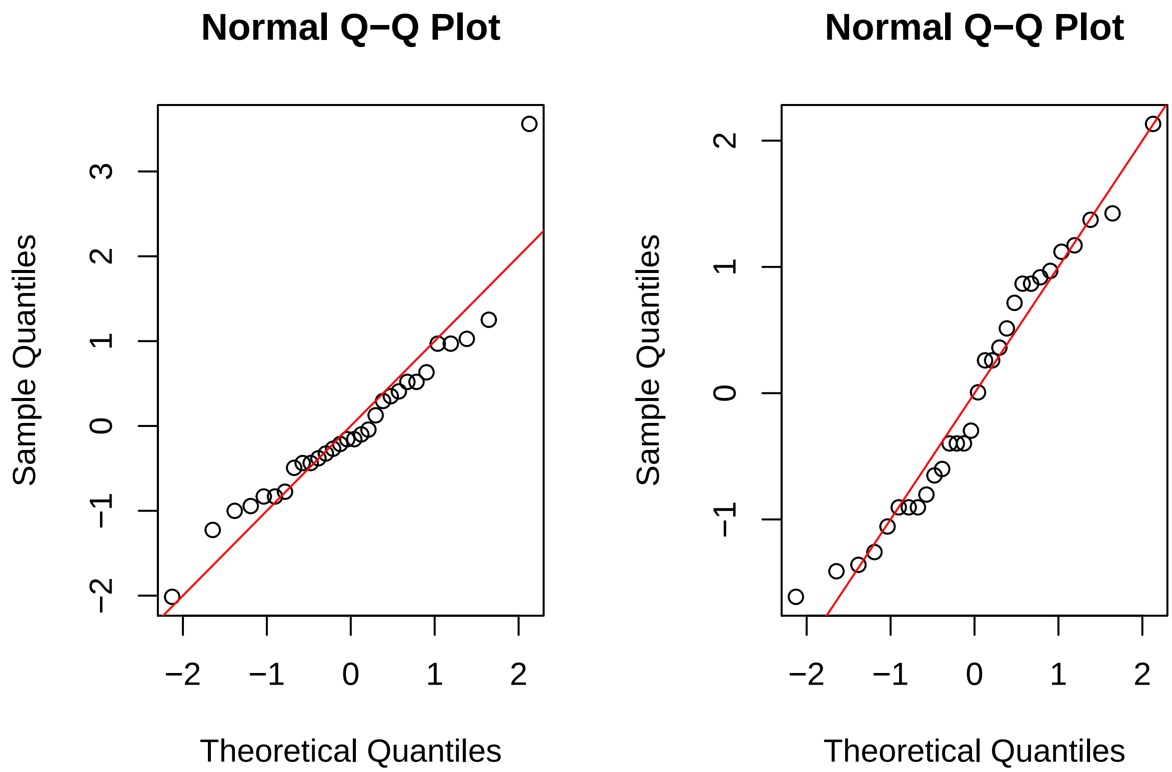



```
par(mfrow=c(1,2))
boxplot(y1)
boxplot(y2)
```



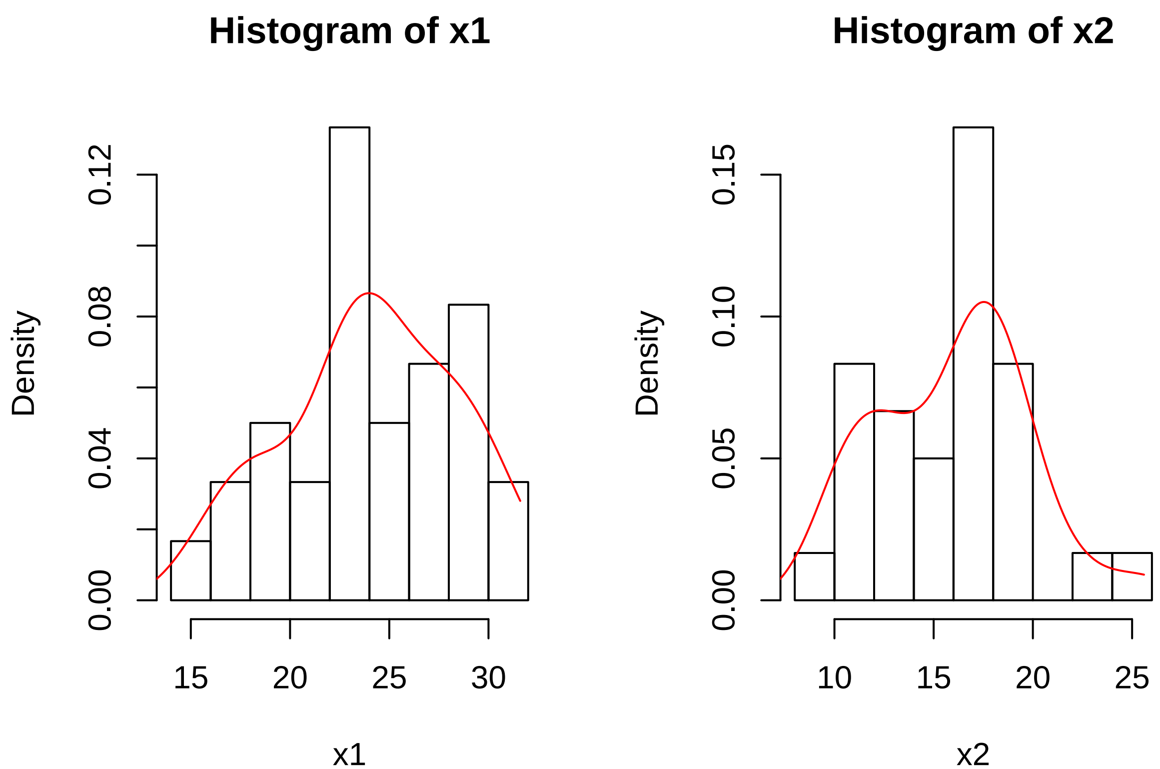
```
par(mfrow=c(1,2))
qqnorm((y1-mean(y1))/sd(y1))
```

```
abline(0,1, col="red")
qqnorm((y2-mean(y2))/sd(y2))
abline(0,1, col="red")
```

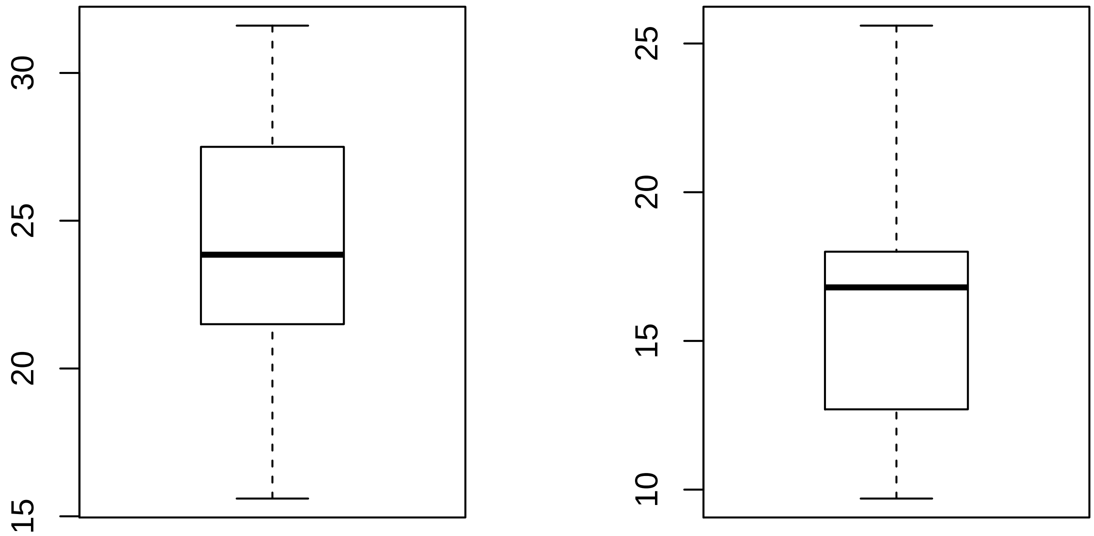


But they don't have as much as big difference in attempts

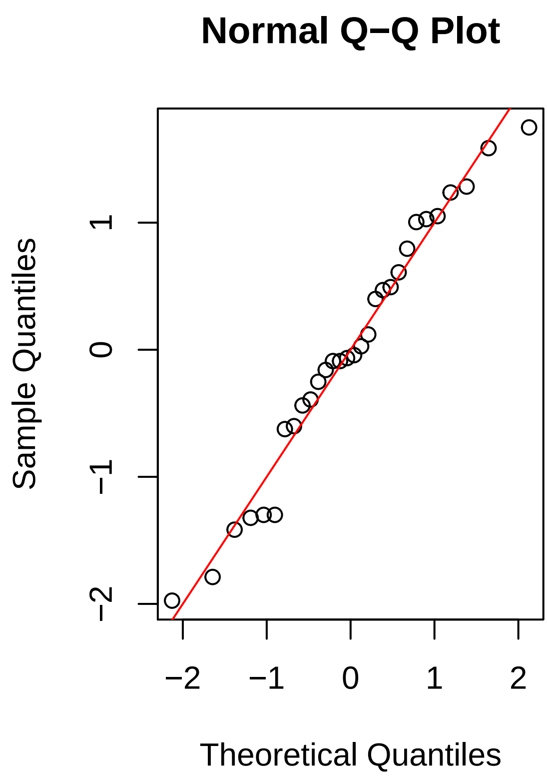
```
par(mfrow=c(1,2))
x1 = as.numeric(threep_attempt[37,])
x2 = as.numeric(threep_attempt[27,])
hist(x1, freq=FALSE)
lines(density(x1, from = 0, to = max(x1)), col="red")
hist(x2, freq = FALSE)
lines(density(x2, from = 0, to = max(x2)), col="red")
```

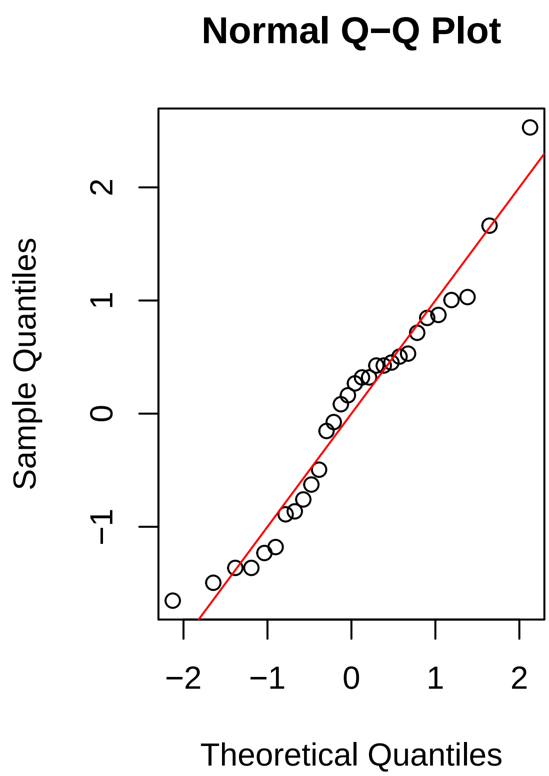



```r
par(mfrow=c(1,2))
boxplot(x1)
boxplot(x2)
```




```r
par(mfrow=c(1,2))
qqnorm((x1-mean(x1))/sd(x1))
abline(0,1, col="red")
qqnorm((x2-mean(x2))/sd(x2))
abline(0,1, col="red")
```





Plotting per team per year data

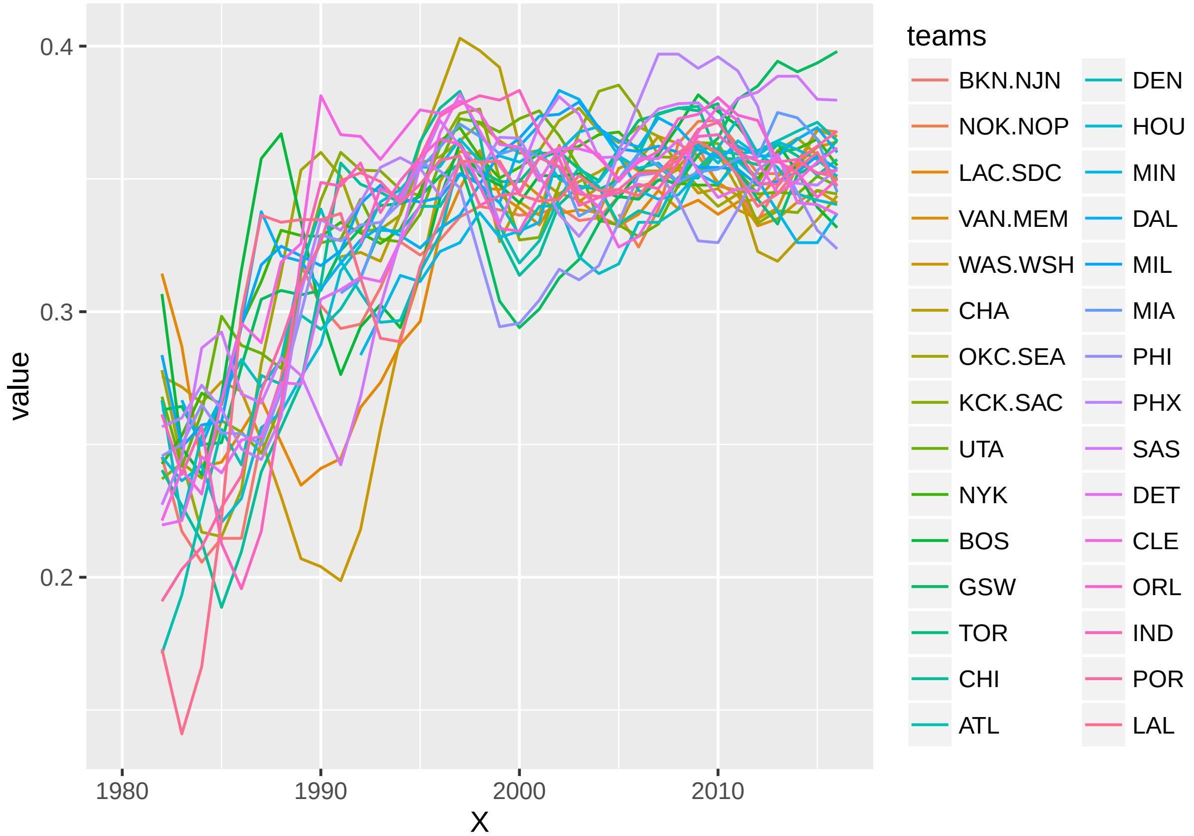
```
threep_anal <- function(fname, ind)
{
    data <- read.csv(fname)
    df <- melt(data, id.vars='X', variable_name = 'teams')
    ggplot(df, aes(X,value)) + geom_line(aes(colour = teams))
    # Smoothing moving average
    col_names <- names(data)
    # FIXME: Fudging data for these two years (Interpolation)
    if (ind==1)
    {
      data[24, 'CHA'] <- 0.353
      data[25, 'CHA'] <- 0.357
    }
    if (ind==2)
    {
      data[24, 'CHA'] <- 11.4
      data[25, 'CHA'] <- 11.4
    }
    smooth_data <- data.frame(time = data["X"])
    for (col_name in col_names[2:ncol(data)])
    {
        smooth_data[col_name] <- SMA(data[col_name], n=3)
    }
    df2 <- melt(smooth_data, id.vars='X', variable_name = 'teams')

}

threep_per_smooth <- threep_anal('../../results/team_data/comb_3pper.csv', 1)
```

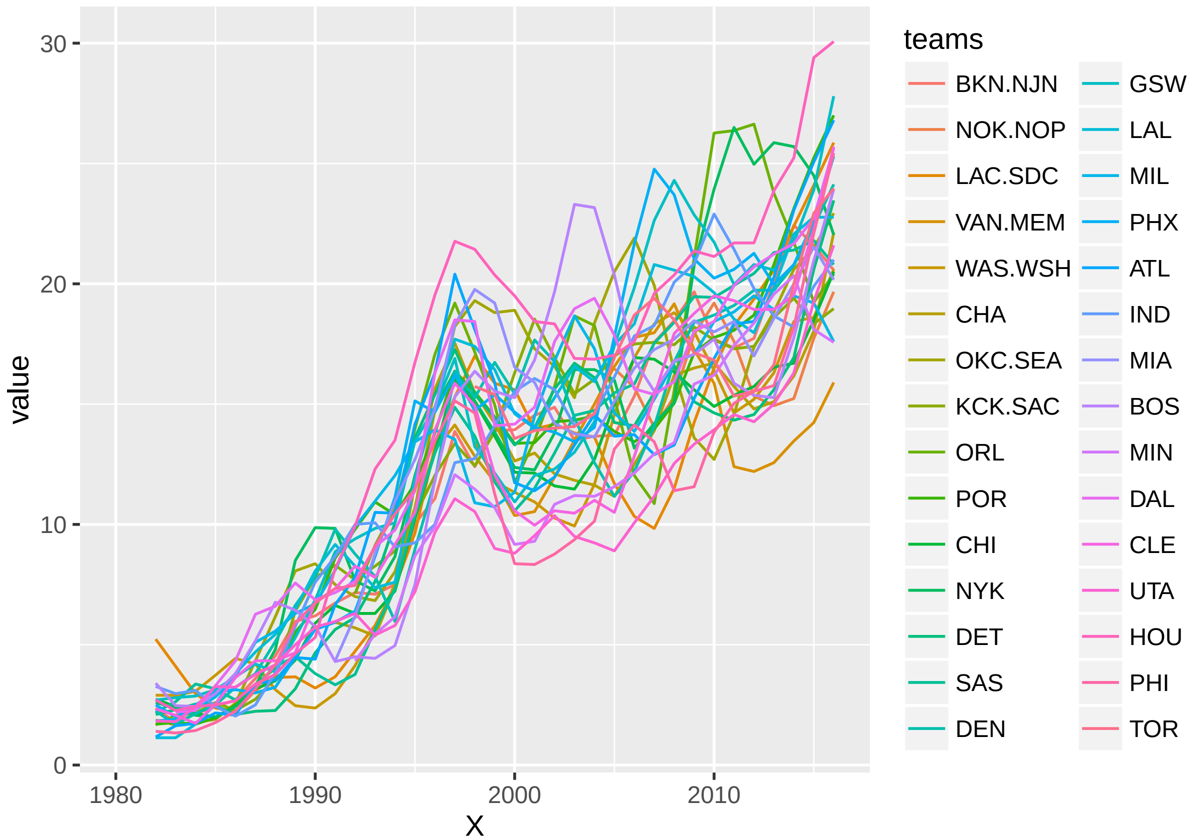
```
threep_attempt_smooth <- threep_anal('../../results/team_data/comb_3pattempt.csv', 2)
ggplot(threep_per_smooth, aes(X,value)) + geom_line(aes(colour = teams))
```

```
## Warning: Removed 154 rows containing missing values (geom_path).
```




```
ggplot(threep_attempt_smooth, aes(X,value)) + geom_line(aes(colour = teams))
```

```
## Warning: Removed 154 rows containing missing values (geom_path).
```



Looking at statistical significance

```
t.test(y1,y2)
```

```
##
##  Welch Two Sample t-test
##
## data:  y1 and y2
## t = -0.84531, df = 57.354, p-value = 0.4014
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.013811194  0.005611194
## sample estimates:
## mean of x mean of y
## 0.3527667 0.3568667
```

```
y3 <- as.numeric(threep_per[17,])
t.test(y1,y3)
```

```
##
##  Welch Two Sample t-test
##
## data:  y1 and y3
## t = -2.3698, df = 53.206, p-value = 0.02145
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02315934 -0.00192802
## sample estimates:
## mean of x mean of y
```

```
## 0.3527667 0.3653103
```

Hence threepointer percent looks more or less the same

```r
x1 <- as.numeric(threep_attempt[37,])
x2 <- as.numeric(threep_attempt[27,])
x3 <- as.numeric(threep_attempt[17,])
t.test(x1,x2)
```

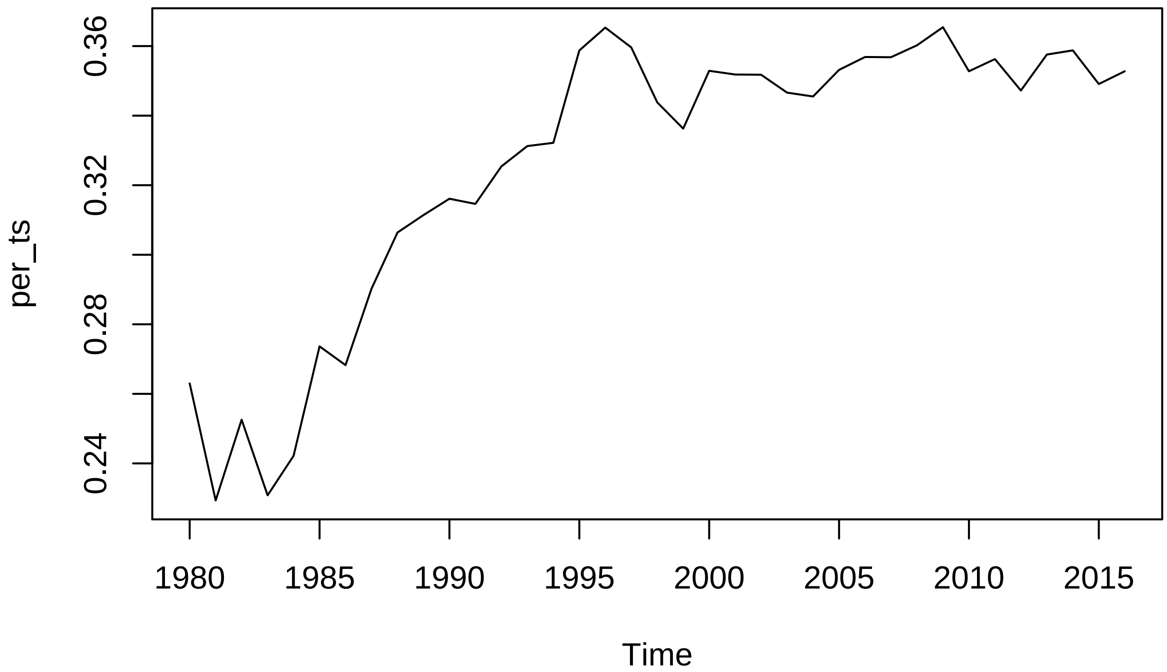
```
##
##  Welch Two Sample t-test
##
## data:  x1 and x2
## t = 7.7327, df = 57.151, p-value = 1.881e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   6.002548 10.197452
## sample estimates:
## mean of x mean of y
##  24.08333  15.98333
```

```r
t.test(x1,x3)
```

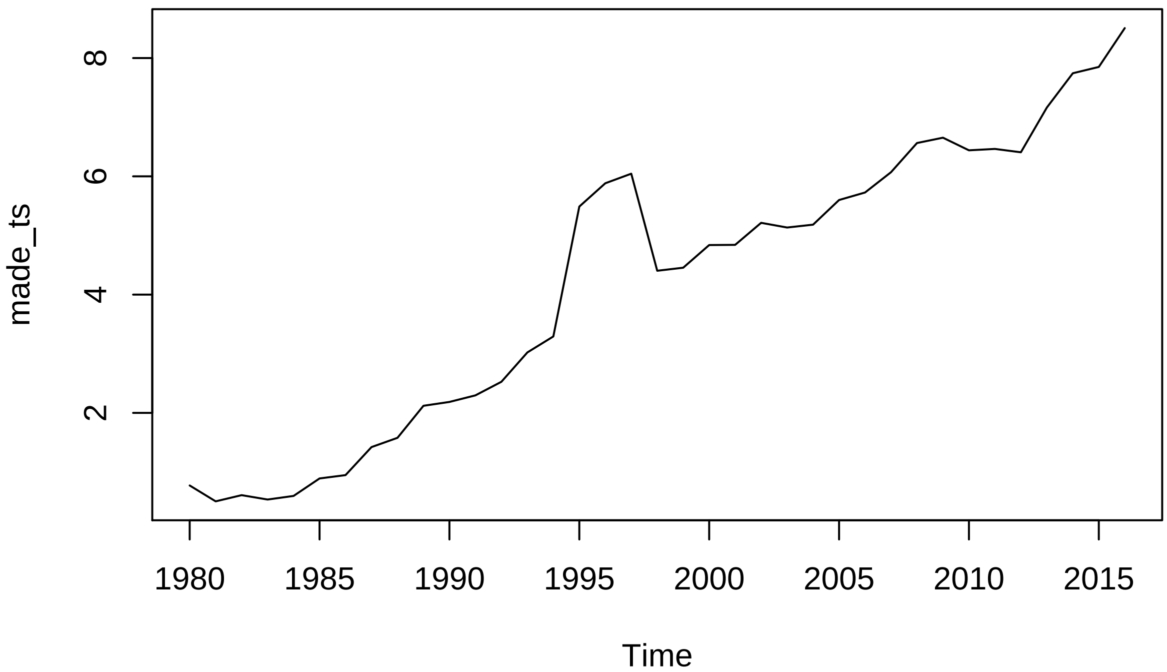
```
##
##  Welch Two Sample t-test
##
## data:  x1 and x3
## t = 7.9372, df = 55.038, p-value = 1.099e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   5.993459 10.042174
## sample estimates:
## mean of x mean of y
##  24.08333  16.06552
```

Time series analysis HoltWinters predictions

```r
per_ts <- ts(avg_threep_per, start=head(year_range,n=1))
plot.ts(per_ts)
```




```r
made_ts <- ts(avg_threep_made, start=head(year_range,n=1))
plot.ts(made_ts)
```

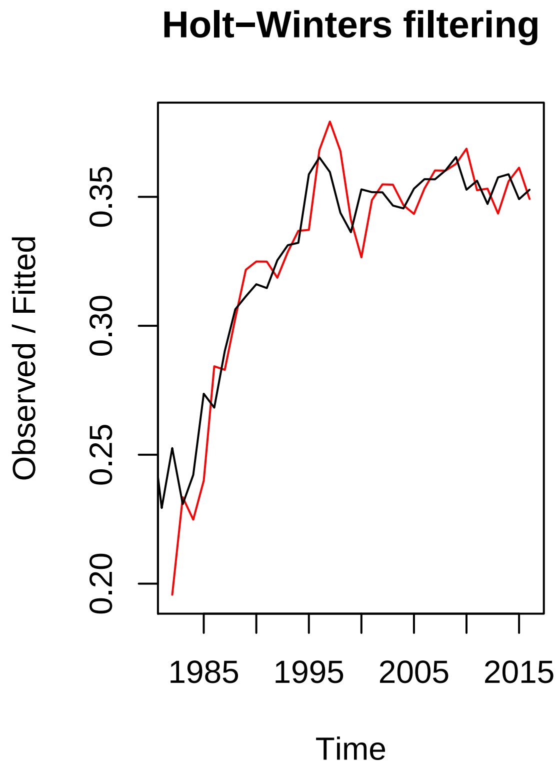


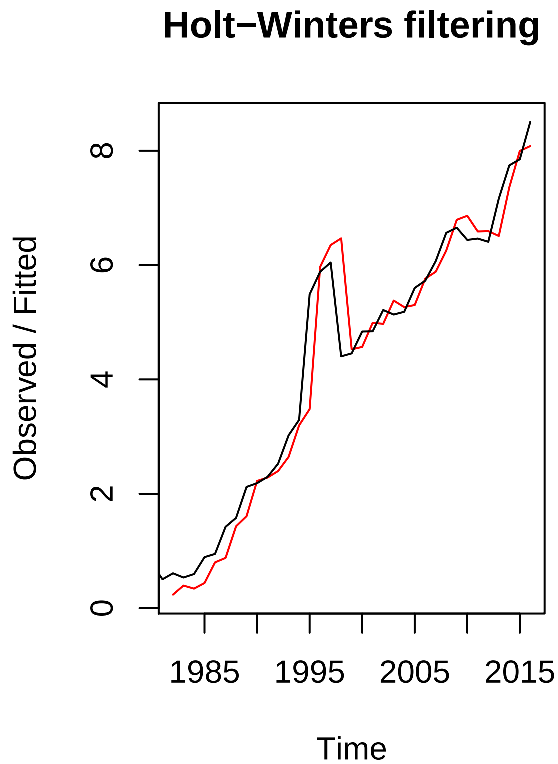
gamma=False means non-seasonal

```r
par(mfrow=c(1,2))
per_forecasts <- HoltWinters(per_ts, gamma=FALSE)
plot(per_forecasts)
per_forecasts$SSE
```

```
## [1] 0.008394918
```

```
per_future <- forecast.HoltWinters(per_forecasts, h=5)
made_forecasts <- HoltWinters(made_ts, gamma=FALSE)
plot(made_forecasts)
```

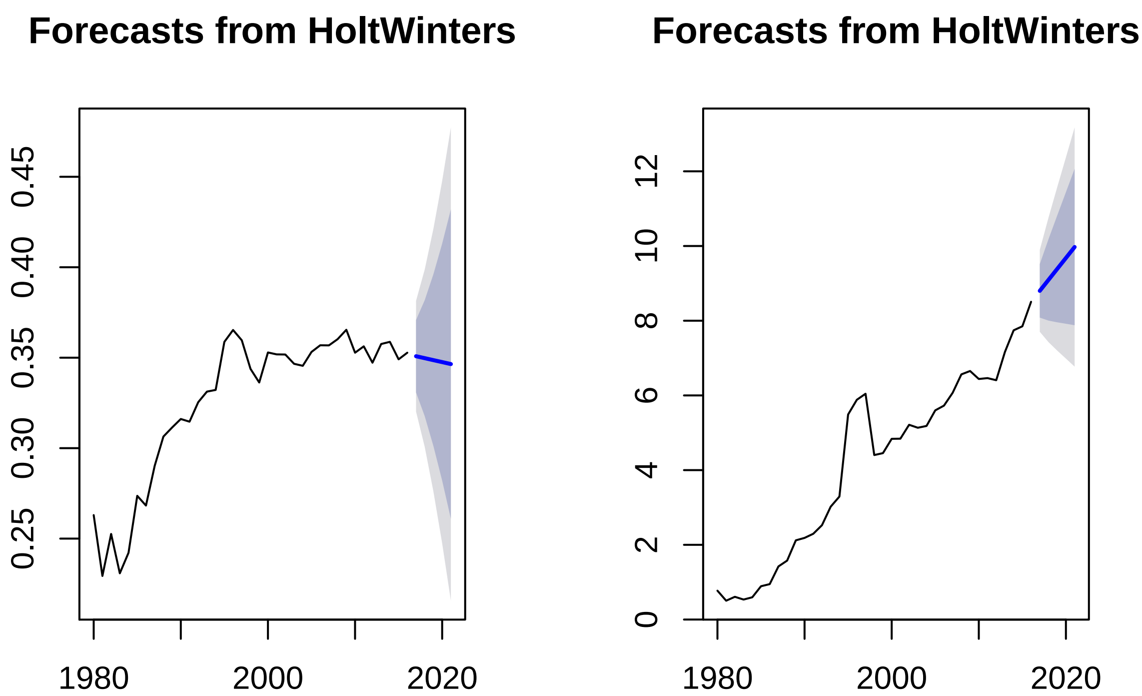





```
made_forecasts$SSE
```

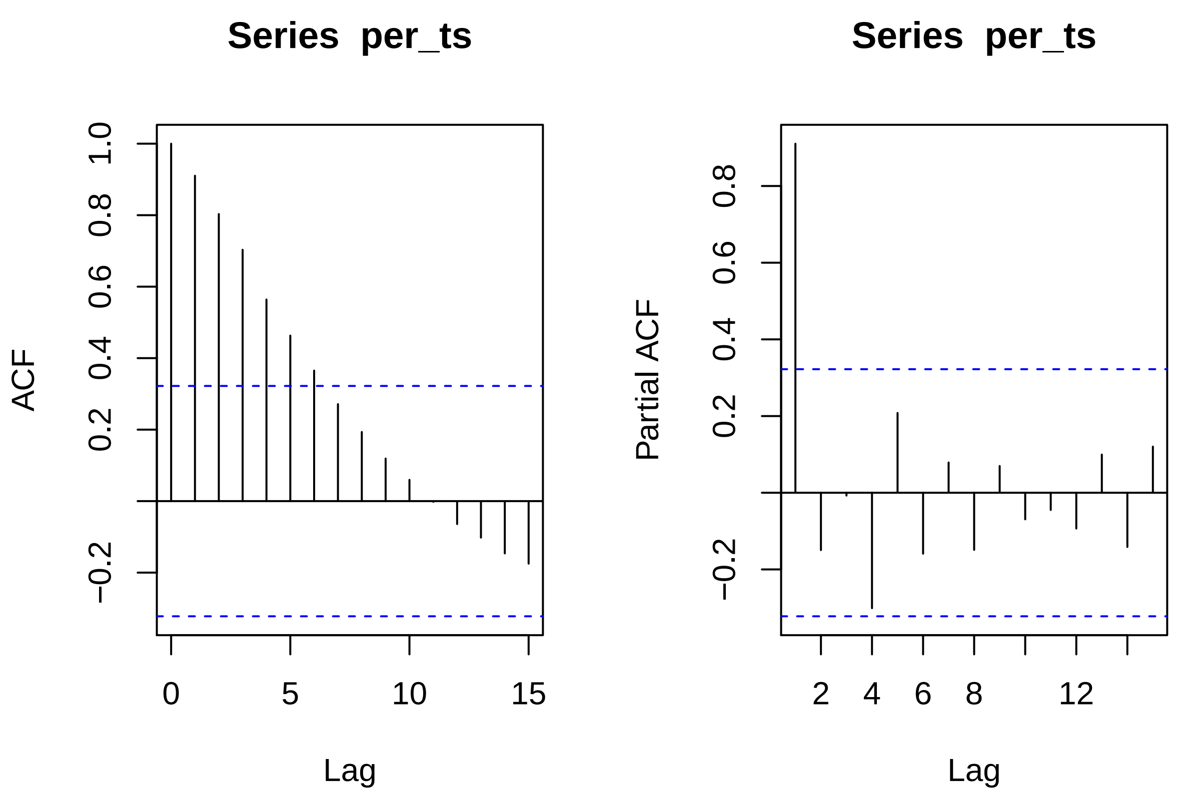
```
## [1] 11.06333
```

```
made_future <- forecast.HoltWinters(made_forecasts, h=5)
plot.forecast(per_future)
plot.forecast(made_future)
```



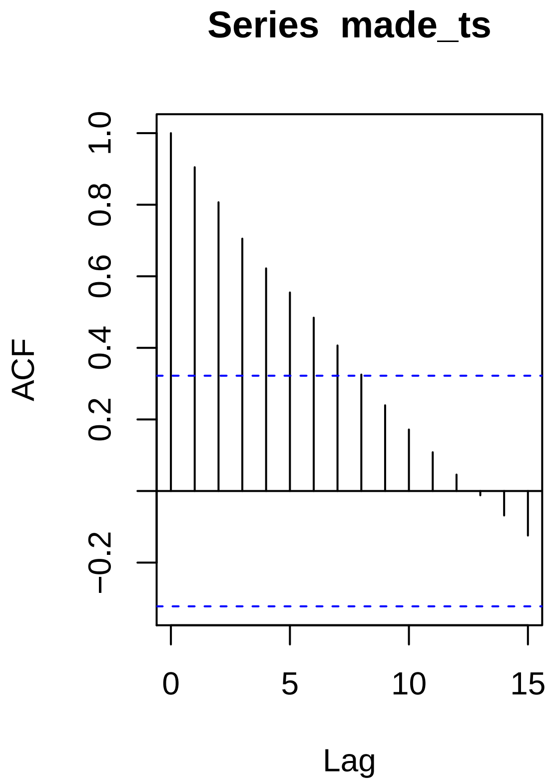
Arima models

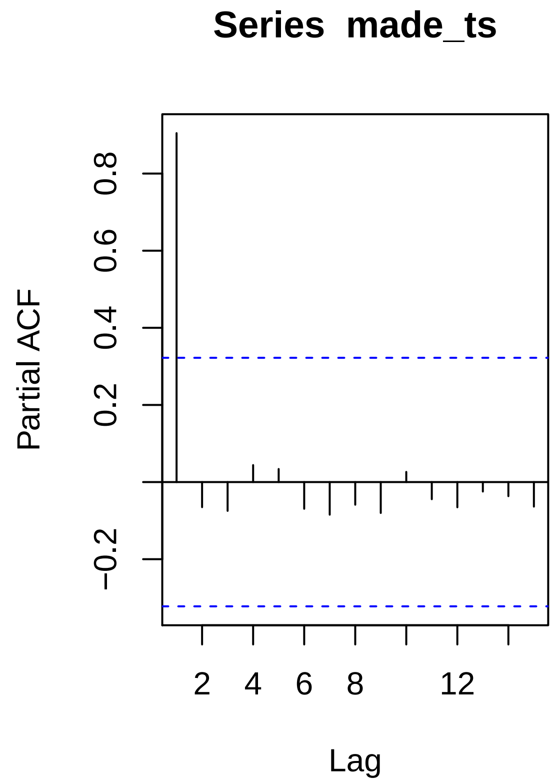
```
par(mfrow=c(1,2))
acf(per_ts)
pacf(per_ts)
```



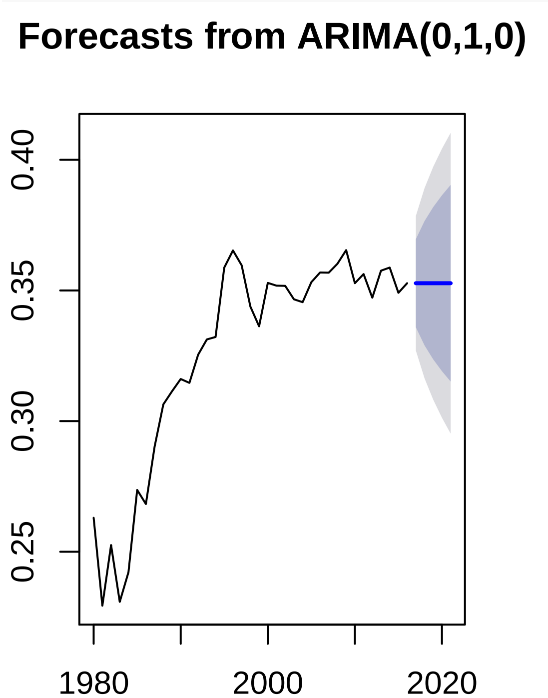

```
per_fit <- auto.arima(per_ts, seasonal = FALSE)
acf(made_ts)
```

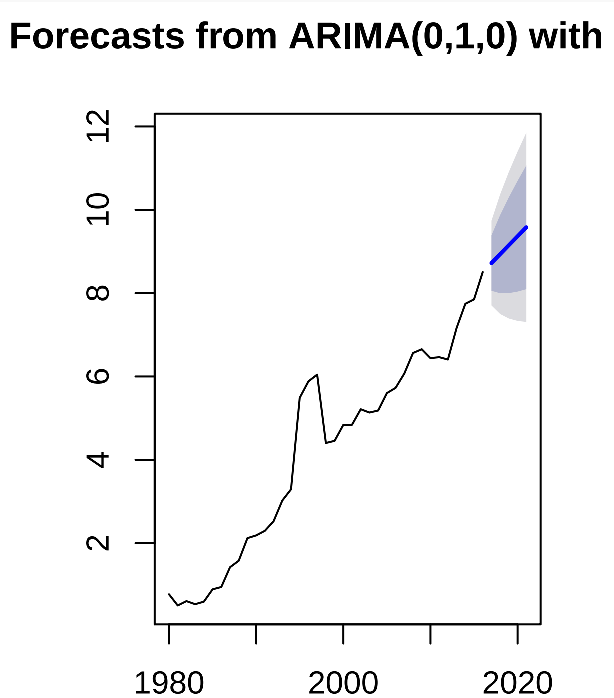
```r
pacf(made_ts)
```






```r
made_fit <- auto.arima(made_ts, seasonal = FALSE)
plot(forecast(per_fit, h=5))
plot(forecast(made_fit, h=5))
```

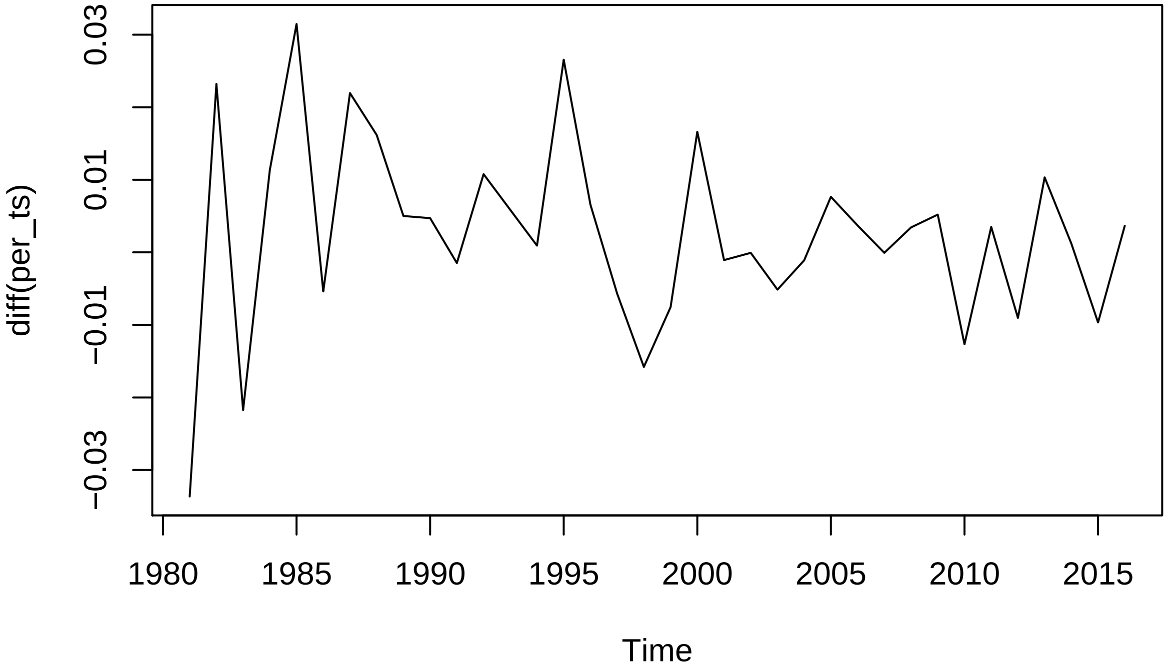




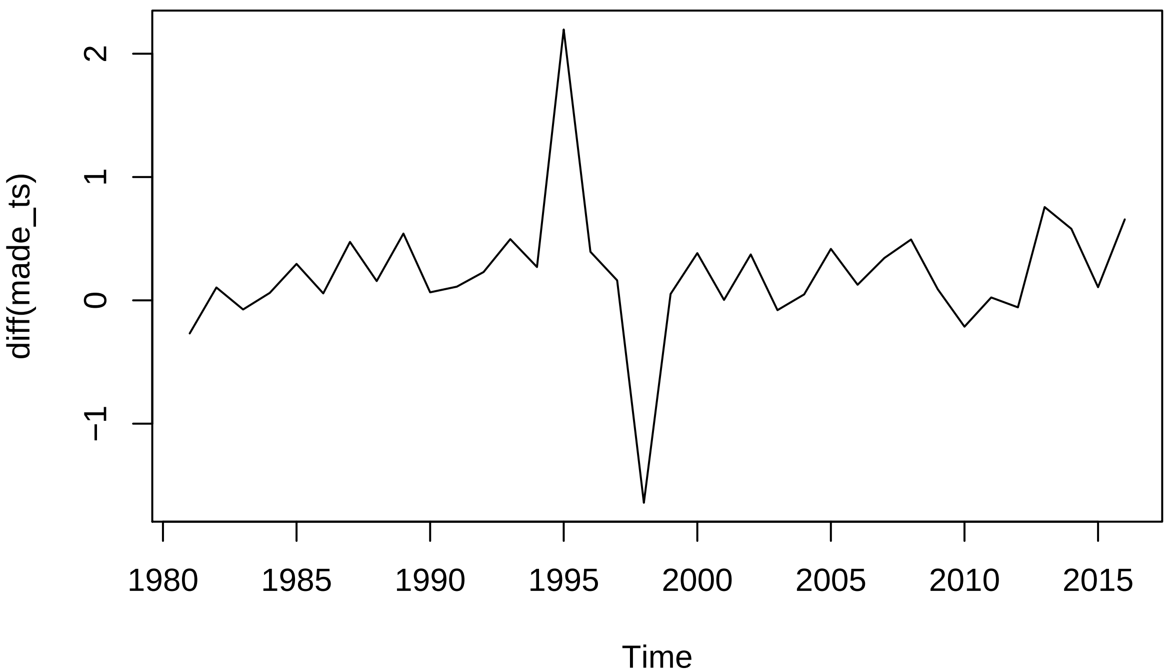
ARIMA stands for Autoregressive Integrated Moving Average models. Univariate (single vector) ARIMA is a forecasting technique that projects the future values of a series based entirely on its own inertia.

Its main application is in the area of short term forecasting requiring at least 40 historical data points. It works best when your data exhibits a stable or consistent pattern over time with a minimum amount of outliers. Sometimes called Box-Jenkins (after the original authors), ARIMA is usually superior to exponential smoothing techniques when the data is reasonably long and the correlation between past observations is stable. If the data is short or highly volatile, then some smoothing method may perform better. If you do not have at least 38 data points, you should consider some other method than ARIMA.

```r
plot(diff(per_ts))
```




```r
plot(diff(made_ts))
```



The first order differences look largely stationary. This and the auto.arima giving 0,1,0 implies that the trend is mostly a random walk.

From all these analyses we can say that 3-pointer shooting is not a fad due to one team but rather a long term trend that one team just capitalized on because they excelled at it.