Coursework_Week3

HT

6/3/2019

```r
library(ggplot2)
library(forecast)
library(reshape2)
library(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
```

```r
library(stringr)
library(lubridate)
```

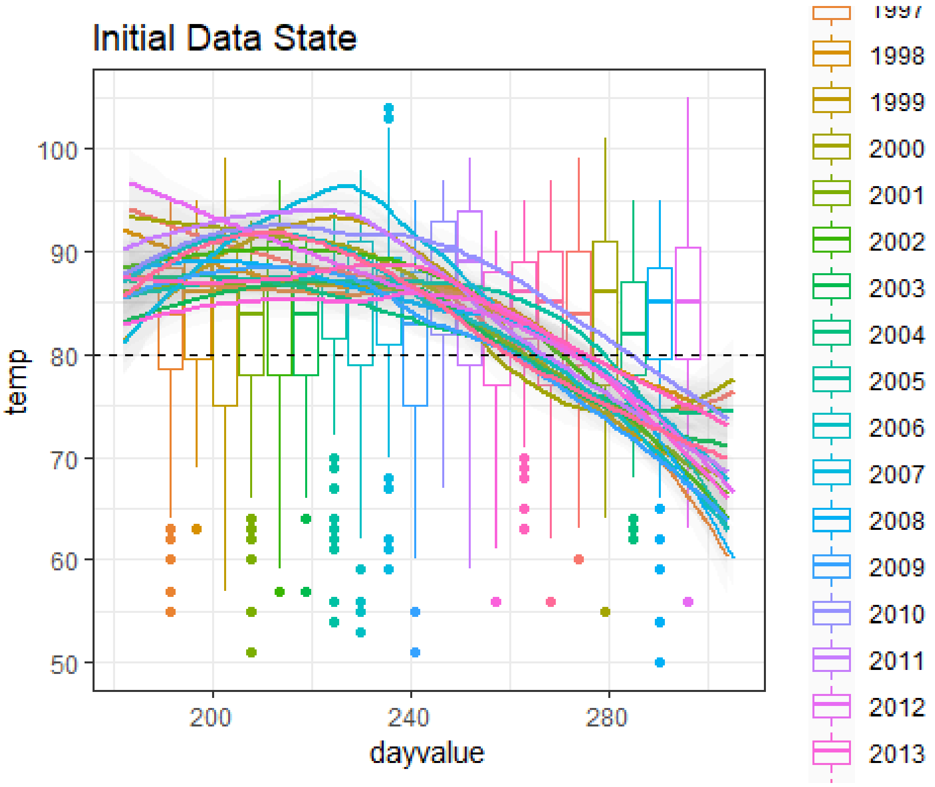
```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

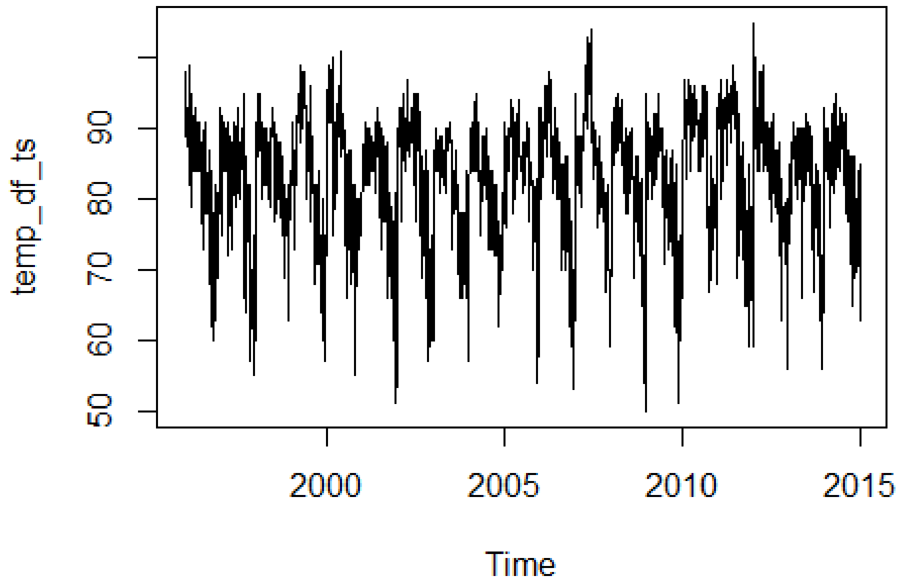
```r
# Reading the temperature source file
temp<- read.table("temps.txt", header = TRUE, sep = "\t")

# Reshaping the data and making it in a tabular form to use later in analysis
temp_df <- melt(temp[,c("DAY","X1996","X1997","X1998","X1999","X2000","X2001"
,"X2002","X2003","X2004","X2005","X2006","X2007","X2008","X2009","X2010","X20
11","X2012","X2013","X2014","X2015")])
```

```
## Using DAY as id variables
```

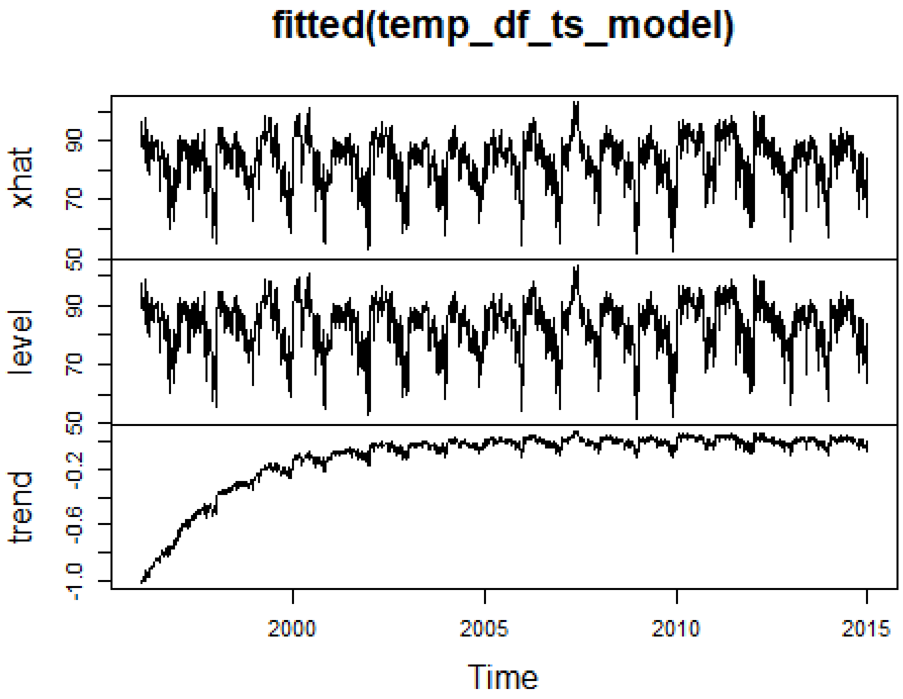
```r
# Formatting the data: Changing the year to clean value, changing date to a p
roper date format, using that to calculate the day of year, as otherwise time
series was converting date to a very big continous number
temp_df <- temp_df%>%
  mutate(variable = str_replace(variable,"X",""))%>%
  mutate(Dt = dmy(paste(DAY, variable, sep = "-")))%>%
  select(Dt,variable,value)%>%
  mutate(dayvalue = yday(as.Date(Dt)))%>%
  select(dayvalue, variable, value)

# Renaming columns
colnames(temp_df)[colnames(temp_df)=='variable'] <- 'year'
colnames(temp_df)[colnames(temp_df)=='value'] <- 'temp'

#plotting data to see the general trend. Threshold for summer end is indicate
d by y intercept set to 80F.
ggplot(temp_df,aes(x= dayvalue, y= temp, color = year))+
  geom_boxplot() +
  theme_bw()+
  geom_smooth(alpha = .05)+
  labs(title = "Initial Data State")+
  geom_hline( yintercept = 80, linetype = "dashed", color = "black")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



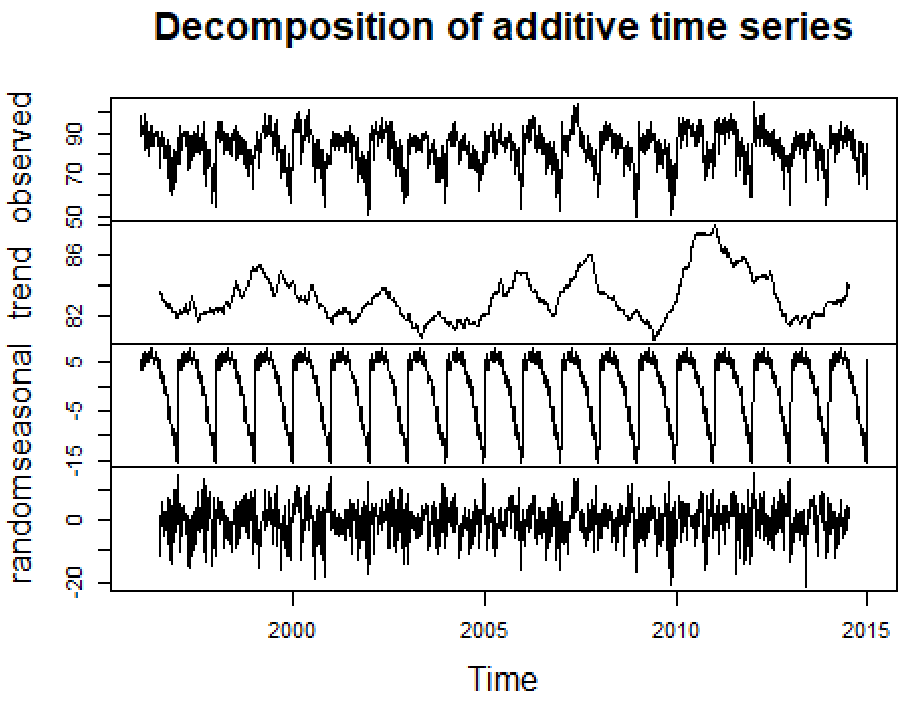
```
# From initial potting it is not very clear if summer end is changing or not


# Converting to time series
temp_df_ts <-ts(temp_df$temp, start = 1996, end = 2015, frequency = 123)
plot(temp_df_ts)
```



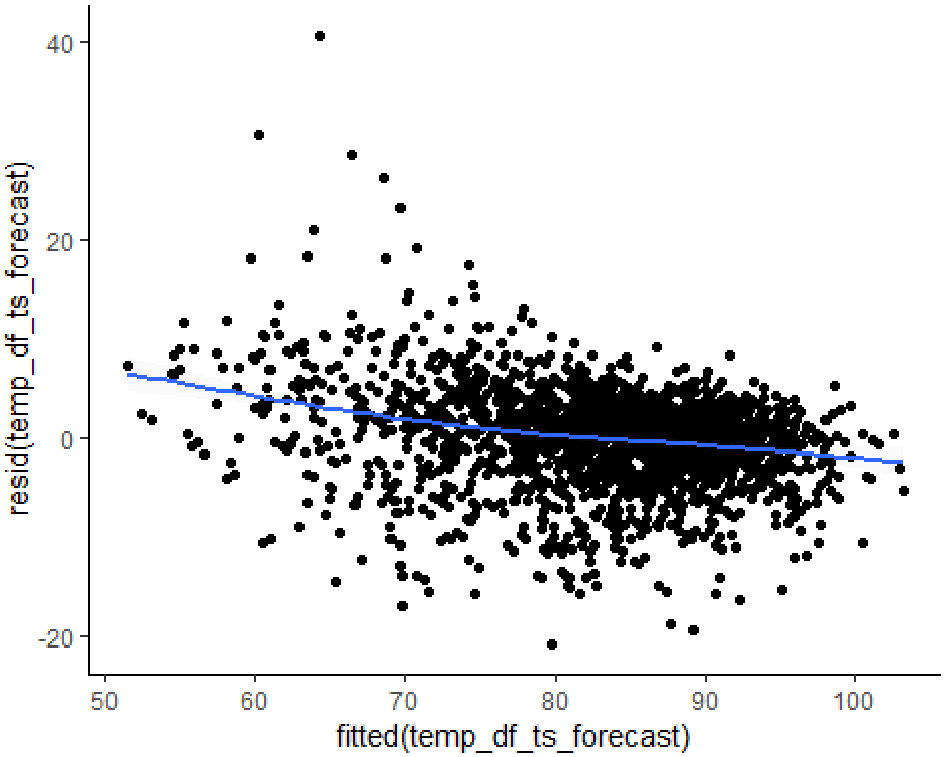
```
temp_df_ts

## Time Series:
## Start = c(1996, 1)
## End = c(2015, 1)
## Frequency = 123
##    [1]  98  97  97  90  89  93  93  91  93  93  90  91  93  93  82  91  96
##   [18]  95  96  99  91  95  91  93  84  84  82  79  90  91  87  86  90  84
##   [35]  91  93  88  91  84  90  89  88  86  84  86  89  90  91  91  90  89
##   [52]  90  91  91  91  84  88  84  86  88  84  82  80  73  87  84  87  89
##   [69]  89  89  91  84  86  88  78  79  86  82  82  78  79  79  78  81  84
##   [86]  84  87  84  79  75  72  64  66  72  84  70  66  64  60  78  70  72
##  [103]  69  69  73  79  81  80  82  66  63  68  79  81  69  73  73  75  75
##  [120]  81  82  82  81  86  90  93  91  84  84  75  87  84  87  84  88  86
##  [137]  90  91  91  89  89  89  90  89  84  87  88  89  89  91  91  89  88
##  [154]  72  80  84  88  89  88  84  84  80  73  80  86  88  88  87  88  91
##  [171]  91  89  89  88  82  79  81  82  84  87  90  90  91  91  88  88  91
##  [188]  93  81  81  82  86  88  84  80  82  86  87  87  88  88  90  88  91
##  [205]  95  89  70  80  82  66  70  64  68  77  86  75  73  75  78  81  82
##  [222]  82  82  80  82  82  79  80  68  63  57  66  64  69  70  70  62  63
##  [239]  62  75  71  57  55  64  66  60  91  88  91  91  91  89  93  95  95
##  [256]  91  91  86  88  87  91  87  90  91  95  91  91  89  91  91  86  88
##  [273]  80  88  89  90  86  86  82  84  86  90  89  89  86  82  87  88  84
##  [290]  86  80  82  86  84  87  90  79  84  87  87  88  90  91  89  90  93
##  [307]  93  91  87  84  77  90  91  89  90  89  79  78  81  84  89  87  87
##  [324]  88  87  82  80  82  82  88  84  81  82  84  87  80  75  75  86  78
##  [341]  77  82  82  73  82  69  72  73  78  78  78  75  79  78  77  78  82
##  [358]  75  73  63  63  72  75  79  79  79  78  82  79  84  82  87  88  90
##  [375]  91  82  86  87  87  82  77  73  81  81  86  82  87  88  90  90  91
##  [392]  93  93  91  93  93  93  93  97  99  96  93  88  89  91  93  93  93
##  [409]  91  90  96  98  97  98  93  93  96  98  98  89  91  91  90  80  82
##  [426]  89  88  90  91  91  84  88  91  84  93  96  96  91  91  77  87  87
##  [443]  87  86  87  89  81  81  82  79  68  79  72  75  78  81  82  78  80
##  [460]  77  71  73  75  84  71  73  71  73  73  72  72  73  70  64  75  73
##  [477]  77  80  71  66  60  64  73  57  59  64  69  75  73  72  75  75  89
##  [494]  91  93  95  96  96  96  91  96  99  96  93  91  93  93  93  91  97
##  [511] 100  99  93  96  87  82  75  82  88  91  89  87  86  86  81  84  88
##  [528]  91  91  91  91  96  95  89  89  89  89  94  97  99 101 101  97  87
##  [545]  86  88  92  92  90  90  92  92  88  87  79  81  82  87  81  66  66
##  [562]  75  80  82  84  86  87  86  80  75  73  73  84  87  77  73  81  84
##  [579]  82  68  71  75  73  75  77  79  82  81  82  73  66  55  55  64  71
##  [596]  73  75  75  77  80  80  80  73  73  75  79  75  75  78  75  78  80
##  [613]  75  77  78  84  87  87  84  86  87  87  89  91  87  90  90  86  82
##  [630]  82  84  87  88  90  87  84  87  90  84  82  88  90  84  89  89  87
##  [647]  84  84  84  86  88  84  86  88  87  88  86  86  81  87  84  90  91
##  [664]  91  87  86  88  90  88  93  90  91  91  81  86  81  82  80  75  73
##  [681]  81  90  88  87  86  86  89  87  84  84  86  77  77  81  81  82  84
##  [698]  86  87  88  69  66  72  75  78  71  71  75  80  81  80  79  70  68
##  [715]  79  66  73  75  78  78  75  75  62  60  64  71  75  79  80  81  79
##  [732]  73  64  51  55  63  72  71  90  90  87  89  93  93  89  89  90  91
##  [749]  84  77  82  88  91  93  93  93  93  91  95  91  89  87  84  86  89
##  [766]  91  91  88  90  93  91  91  91  93  97  87  87  86  88  89  91  91
##  [783]  89  88  90  91  93  91  93  93  91  95  93  91  88  84  82  82  78
##  [800]  77  84  84  89  95  93  91  88  87  91  95  95  90  75  78  91  88
##  [817]  86  81  80  86  84  77  82  73  69  75  75  79  73  79  82  84  84
##  [834]  82  87  86  80  71  66  70  78  84  79  68  57  66  64  68  71  73
##  [851]  71  64  59  68  60  68  69  75  75  68  60  73  81  87  86  80  84
##  [868]  87  90  89  84  84  86  87  84  86  88  88  88  88  88  89  86  81
##  [885]  82  84  87  87  89  88  84  88  84  84  84  82  84  82  84  84  86
##  [902]  87  84  81  87  89  90  86  89  90  90  87  88  88  90  89  88  89
##  [919]  90  91  89  88  89  88  86  87  87  84  73  75  81  82  79  80  81
##  [936]  84  82  82  81  81  81  84  87  82  75  81  80  82  82  82  73  66
##  [953]  71  72  68  66  77  78  75  73  73  73  73  66  78  78  78  69  72
##  [970]  68  70  75  78  84  78  78  73  73  68  64  57  70  77  75  82  81
##  [987]  86  88  90  90  89  87  88  89  90  89  91  91  84  84  84  87  84
## [1004]  88  89  89  93  95  89  87  84  89  87  89  90  91  90  91  91  90
## [1021]  84  81  82  84  75  82  80  77  82  82  84  86  86  89  88  82  84
## [1038]  84  87  82  86  88  90  87  88  87  82  80  81  82  84  81  86  73
## [1055]  84  84  84  81  79  79  73  75  80  79  78  73  75  80  84  82  81
## [1072]  79  72  78  78  80  82  82  80  81  80  75  75  73  71  71  77  73
## [1089]  64  63  62  71  75  73  68  71  73  73  70  73  78  79  81  78  75
## [1106]  78  82  91  89  86  86  89  82  76  88  89  78  83  86  84  87  84
## [1123]  85  89  90  89  89  90  91  91  90  92  94  92  90  83  78  84  82
## [1140]  86  88  91  88  86  80  82  85  83  87  88  86  90  92  89  90  90
## [1157]  89  92  94  93  87  85  84  84  86  86  85  85  85  85  88  87  85
## [1174]  81  81  83  85  86  84  84  86  88  88  91  88  86  88  90  90  90
## [1191]  86  87  88  85  77  86  85  85  82  83  85  83  85  81  72  72  73
## [1208]  70  77  82  74  77  78  79  76  75  81  83  83  80  67  70  56  54
## [1225]  61  63  62  64  69  70  93  93  93  91  90  81  80  82  84  84  90
## [1242]  91  91  91  91  91  93  93  96  93  93  91  86  87  88  93  95  96
## [1259]  91  91  94  95  95  97  98  96  89  97  96  95  96  88  84  81  87
## [1276]  86  89  86  88  88  93  91  88  87  83  85  88  88  90  90  88  80
## [1293]  85  86  85  88  83  85  80  83  83  85  84  82  70  80  82  83  85
## [1310]  85  79  73  75  82  86  84  75  78  79  81  70  75  83  81  82  84
## [1327]  86  76  72  72  79  80  80  71  62  69  70  59  71  77  76  69  69
## [1344]  70  53  56  55  62  66  63  72  73  68  95  85  82  86  88  87  82
## [1361]  82  89  86  85  87  86  84  81  86  89  89  88  86  86  79  82  87
## [1378]  87  87  90  89  87  92  90  92  92  94  97  96  98  98 100 103 103
## [1395] 100  90 100  99 102 101 101  97  95  96  99 104  98  95  94  92  88
## [1412]  88  89  89  86  84  83  88  91  89  85  86  88  89  89  89  86  85
## [1429]  81  82  76  78  79  82  81  78  86  83  89  87  84  85  85  81  79
## [1446]  80  82  77  80  81  82  83  83  81  81  67  72  74  78  78  76  82
## [1463]  77  76  75  78  72  81  59  61  68  67  70  62  67  71  85  87  91
## [1480]  90  88  82  88  90  89  87  89  93  85  88  89  89  88  90  91  94
## [1497]  95  92  87  88  89  87  90  93  92  90  88  89  92  91  91  92  94
## [1514]  90  86  85  85  88  81  81  84  87  86  85  86  90  90  85  82  78
## [1531]  83  78  83  80  86  89  89  88  81  85  83  85  88  87  89  90  88
## [1548]  87  83  87  86  88  79  80  69  82  81  79  75  84  82  78  82  80
## [1565]  77  86  86  86  74  74  80  83  83  82  82  72  75  77  78  77  77
## [1582]  80  81  83  69  67  65  66  72  68  62  54  67  70  59  50  59  65
## [1599]  67  95  90  89  91  80  87  86  82  84  84  86  90  84  89  89  90
## [1616]  88  82  80  82  86  84  87  88  90  92  90  89  85  82  85  89  83
## [1633]  90  92  92  89  91  92  93  93  95  86  90  90  90  88  87  88  90
## [1650]  88  88  85  81  86  87  90  83  75  86  79  79  71  78  79  83  83
## [1667]  85  84  87  84  80  75  81  80  82  79  82  73  80  74  81  79  84
## [1684]  83  85  87  85  80  83  72  74  76  75  76  74  62  71  79  80  85
## [1701]  74  77  66  73  66  61  61  51  55  61  68  71  74  72  69  65  65
## [1718]  60  71  75  66  69  87  84  83  85  88  89  94  97  96  90  93  90
## [1735]  91  91  94  89  87  83  90  91  94  95  97  94  95  95  93  90  94
## [1752]  95  95  96  84  92  95  93  93  91  93  94  94  95  95  96  89  90
## [1769]  90  91  93  92  93  93  94  93  90  89  90  89  87  84  85  89  90
## [1786]  91  92  84  85  90  91  93  92  94  96  89  86  91  91  89  95  93
## [1803]  92  96  95  92  91  88  93  76  81  76  79  76  79  78  68  67  70
## [1820]  73  81  82  85  86  86  80  80  73  78  76  80  78  82  77  80  78
## [1837]  76  81  76  85  76  74  68  71  75  92  94  95  92  90  90  94  94
## [1854]  91  92  95  95  97  90  80  85  87  89  94  91  92  94  92  92  90
## [1871]  94  94  90  93  96  96  91  96  97  85  96  93  93  94  91  95  94
## [1888]  95  95  94  88  90  92  94  96  93  94  98  92  93  95  99  95  95
## [1905]  93  90  92  95  96  95  80  78  75  69  73  81  84  86  87  89  92
## [1922]  86  72  79  77  77  82  86  80  83  82  88  86  84  79  84  78  65
## [1939]  68  75  80  83  81  79  78  72  68  65  73  74  77  80  84  85  80
## [1956]  67  59  63  68  70  73  76  77  79  74  59  61  65 105  93  99  98
## [1973] 100  98  93  95  97  95  90  84  90  90  90  92  93  93  91  84  90
## [1990]  95  97  97  98  98  97  97  94  96  88  94  99  94  87  90  86  84
## [2007]  92  88  87  85  88  91  88  85  91  87  87  84  84  88  84  88  86
## [2024]  85  90  90  80  86  80  89  91  89  85  77  85  85  92  88  83  84
## [2041]  83  81  81  83  87  86  83  79  81  79  85  87  81  78  82  86  88
## [2058]  86  84  72  75  72  74  82  82  83  68  63  70  73  75  79  75  77
## [2075]  77  74  75  74  73  71  76  79  78  79  80  80  70  56  56  56  65
## [2092]  82  85  76  77  83  83  79  88  88  87  80  87  78  85  86  87  91
## [2109]  87  90  86  87  85  84  86  89  86  82  86  86  90  80  87  89  88
## [2126]  90  88  88  86  83  89  90  90  90  89  83  73  67  66  77  82  84
## [2143]  84  88  90  84  82  82  86  90  92  87  90  90  84  90  89  89  88
## [2160]  88  91  90  89  89  90  87  82  84  89  79  78  84  86  73  82  82
## [2177]  71  67  78  79  77  76  77  82  82  82  85  84  84  74  72  76  80
## [2194]  79  81  82  77  68  74  72  73  63  70  72  69  63  66  56  61  69
## [2211]  64  75  78  74  90  93  87  84  86  87  89  90  90  87  85  90  89
## [2228]  90  86  83  86  82  85  76  82  83  88  87  88  89  92  90  82  84
## [2245]  85  81  84  88  90  89  92  95  90  89  86  83  88  84  85  87  88
## [2262]  89  89  86  89  92  93  93  88  84  86  88  91  92  88  89  90  90
## [2279]  92  82  89  91  90  84  84  86  90  92  86  78  80  86  86  85  84
## [2296]  83  87  82  77  78  77  74  78  74  71  84  86  85  78  65  71  78
## [2313]  82  86  86  86  86  85  85  75  69  70  80  76  73  73  77  70  72
## [2330]  74  77  84  84  77  73  68  63  85

# Using holtwinters method for making model
temp_df_ts_model <- HoltWinters(temp_df_ts, gamma = FALSE,start.periods = 199
6)

# Checking the parameters of Model
temp_df_ts_model

## Holt-Winters exponential smoothing with trend and without seasonal compone
nt.
##
## Call:
## HoltWinters(x = temp_df_ts, gamma = FALSE, start.periods = 1996)
##
## Smoothing parameters:
##  alpha: 0.8455303
##  beta : 0.003777803
##  gamma: FALSE
##
## Coefficients:
##           [,1]
## a 81.729657393
## b -0.004838906

# Plotting model to see the trends, levels and Xhats
plot(fitted(temp_df_ts_model))
```



```
# Decomposing data to check for seasonality and trends

temp_df_ts_components <- decompose(temp_df_ts)

head(temp_df_ts_components$seasonal)

## Time Series:
## Start = c(1996, 1)
## End = c(1996, 6)
## Frequency = 123
## [1] 5.335357 4.719730 5.212051 5.209793 5.210696 4.543126

plot(temp_df_ts_components)
```



```
# Forecasting the trend to 1 year ahead
temp_df_ts_forecast <- forecast(temp_df_ts_model, h = 122)


# Generating a chart to visualize how much is residual compared to forecasted
values for given temperature
qplot(fitted(temp_df_ts_forecast),resid(temp_df_ts_forecast))+geom_smooth(met
hod = "auto", alpha = .05) + theme_classic()

## Don't know how to automatically pick scale for object of type ts. Defaulti
ng to continuous.

## Don't know how to automatically pick scale for object of type ts. Defaulti
ng to continuous.

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).
```


40
20
0
-20
resid(temp_df_ts_forecast)
50
60
70
80
90
100
fitted(temp_df_ts_forecast)