Time Series HW 3

Kenny Flagg and Paul Harmon
who have not yet worked together and want the bonus!

September 16, 2016

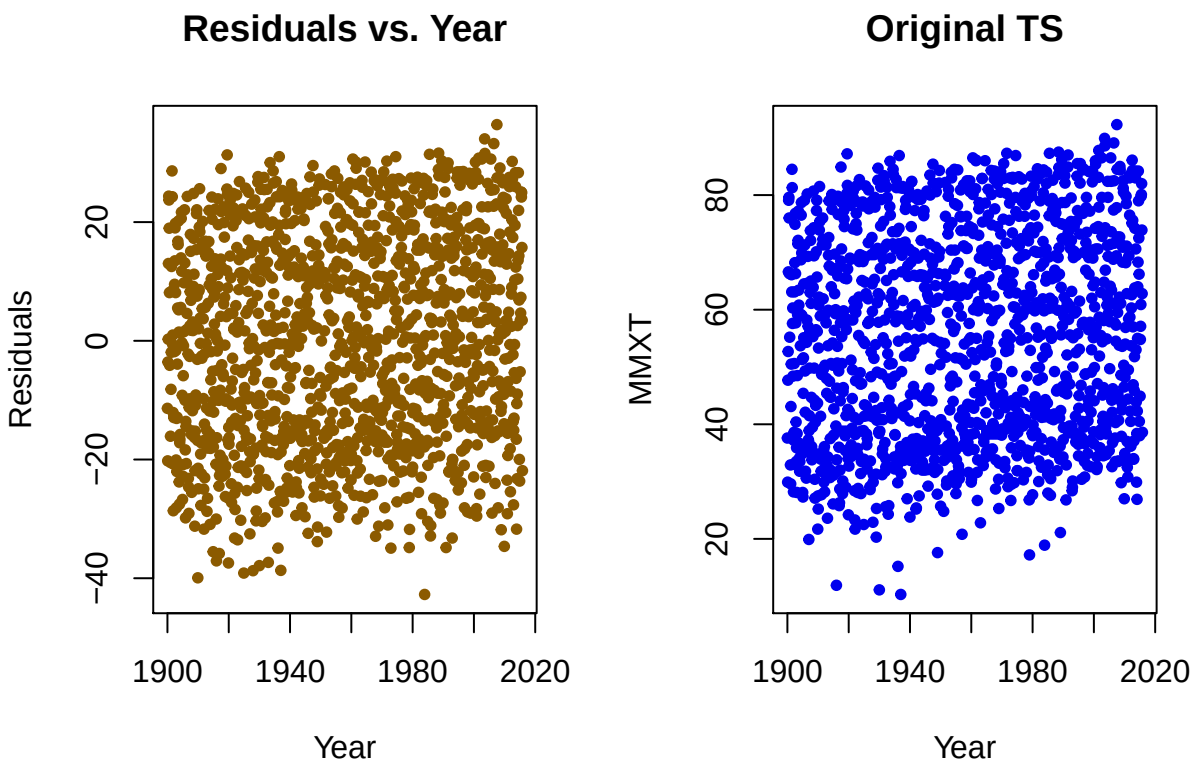
HW 3

You can alone or in pairs that may or may not be people you worked with before. You can discuss it with old partners but should try work as much as possible with new collaborators. 5% bonus if you find someone completely new to work with - that you did not work with on first two assignments.

1) *For the Bozeman temperature data from HW 1 and 2, estimate a model with month only, subtract its fitted values from the responses (or just extract residuals). Plot the residuals vs the fractional* `Year` *variable and compare the plot of this result to the plot of the original time series.* The plot of the residuals and the plot of the original time series are shown below. Note that because the residuals have mean 0, the plot fo the residuals is centered around 0 whereas the plot of the original TS is still on the temperature scale. However, the same increasing linear trend appears evident in both plots.

```
weather = read.csv("rawbozemandata.csv", header = TRUE)

weather$year = (floor(weather$DATE/100))
weather$Month<-round((weather$DATE/100-weather$year)*100,1)
weather$Yearfrac<-weather$year+(weather$Month-1)/12 #makes it an integer variable
#fit the linear model
lm.weather = lm(MMXT ~ Month, data = weather)
#lm.weather$residuals
par(mfrow = c(1,2))
plot(weather$Yearfrac, lm.weather$residuals, pch = 20, col = "orange4",
 main = "Residuals vs. Year", xlab = "Year", ylab = "Residuals")
plot(weather$Yearfrac, weather$MMXT, pch = 20, col = "blue2",main = "Original TS", xlab = "Year", y
```



2) *In the de-seasonalized Bozeman temperature data set, re-assess evidence for the linear trend. Compare the result (test statistic, degrees of freedom and size of p-value) of just fitting a linear time trend in these de-seasonalized responses to the result from our original model with a linear year component and a month adjustment (not the quadratic trend model).*

In the tables below, we list the tests for the linear year-based trend. The test for the original model is a t_{1371} as is the test for the deseasonalized model. In both models we find strong evidence in favor of the linear time trend; the seasonalized model yields a p-value of 0.00014 and the deseasonalized model yields a p-value of 0.00014. Further, the raw slopes are very similar - this indicates that the deseasonalization of the data preserved the long-term time trend.

```
library(pander) #because Paul thinks "library" is cooler than "require"
panderOptions('missing', '-')
lm.seasoned <- lm(MMXT ~ Yearfrac + Month, data = weather)
lm.deseasoned <- lm(lm.weather$residuals ~ Yearfrac + Month, data = weather)
pander(summary(lm.seasoned))
```

	Estimate	Std. Error	t value	Pr(>|t|)
Yearfrac	0.05347	0.01404	3.809	0.0001455
Month	1.144	0.1361	8.408	1.029e-16
(Intercept)	-56.82	27.49	-2.067	0.03892

Table 2: Fitting linear model: MMXT ~ Yearfrac + Month

Observations	Residual Std. Error	R^2	Adjusted R^2
1374	17.41	0.05904	0.05767

```
pander(summary(lm.deseasoned))
```

	Estimate	Std. Error	t value	Pr(>|t|)
Yearfrac	0.05347	0.01404	3.809	0.0001455
Month	-0.006431	0.1361	-0.04726	0.9623
(Intercept)	-104.6	27.49	-3.807	0.000147

Table 4: Fitting linear model: lm.weather$residuals ~ Yearfrac + Month

Observations	Residual Std. Error	R^2	Adjusted R^2
1374	17.41	0.01047	0.00903

3) *I briefly discussed the HADCRUT data set in class. We will consider the HADCRUT4 series of temperature anomalies for the Nothern Hemisphere. The fully up to date data set is available at:* ***http://www.metoffice.gov.uk/hadobs/hadcrut4/data/current/time_series/HadCRUT.4.4.0.0.monthly_nh.tx***

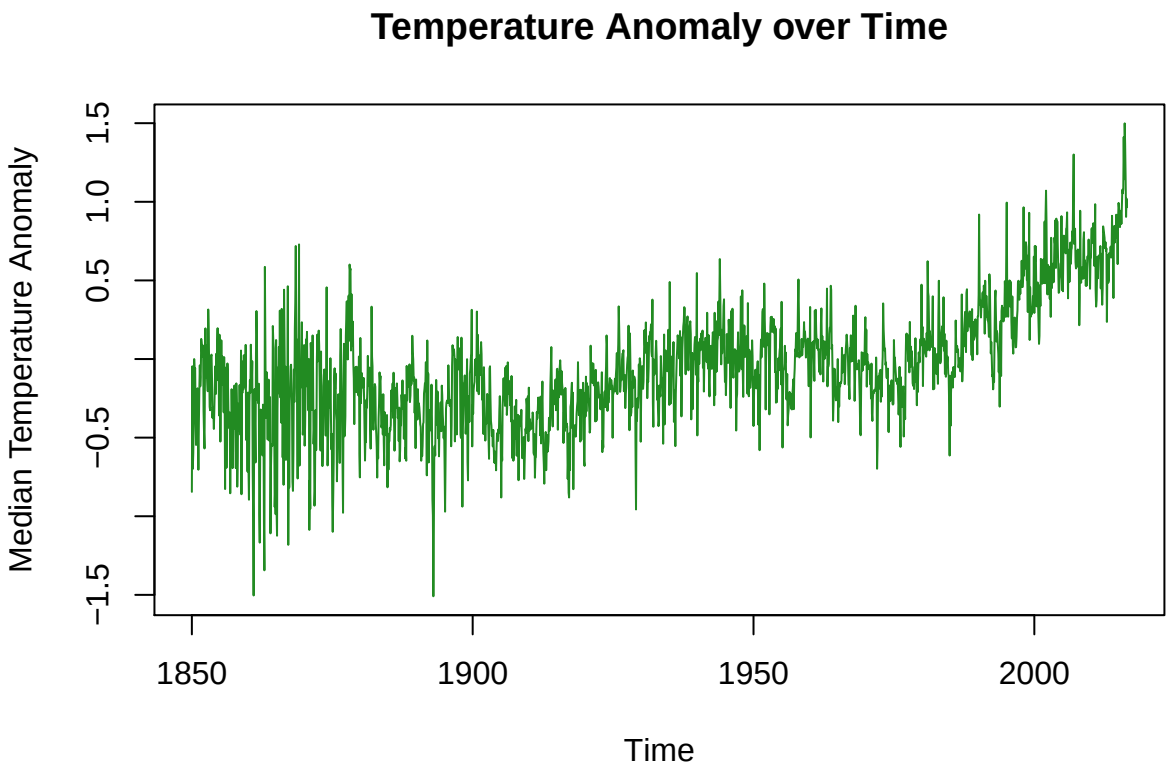
Download the ensemble median monthly northern hemisphere temperature data. We will use the entire time series that is currently available (January 1850 to July 2016). You might want to read ***http://www.metoffice.gov.uk/hadobs/hadcrut4/data/current/series_format.html*** *for more information on the columns in the data set.*

Because the time series is complete over the time frame under consideration, you can use ***ts()*** *to create a time variable instead of messing around with their* ***Year/Month*** *variable.*

Make a plot versus time of the ensemble medians and use that as your response variable in the following questions. Discuss trend, seasonality, outliers, and variability.

The plot of median temperature anomaly over time appears to show some increase in temperatures over time. Further, the variability of temperatures was wider during early times, likely due to the fact that people measuring temperatures in the late 1800s/early 1900s were using less precise methods to measure temperature. There are a couple of outliers in the negative direction (there was an anomaly of -1.509 in 1893) but not any temperatures that appear to be outlying in the positive direction, especially given the increasing trend. We definitely see seasonal oscilations in the data; this is to be expected.

```
par(mfrow = c(1,1))
    hadcrut <- read.table('HadCRUT.4.4.0.0.monthly_nh.txt',
                      col.names = c('Date', 'median.temp.anom',
                                    'bias.l95', 'bias.u95',
                                    'meas.l95', 'meas.u95',
                                    'coverage.l95', 'coverage.u95',
                                    'error.l95', 'error.u95',
                                    'l95', 'u95'))
hadcrut.ts <- ts(hadcrut$median.temp.anom, start = 1850, frequency = 12)
plot(hadcrut.ts, ylab = "Median Temperature Anomaly", main = "Temperature Anomaly over Time", col = "fo
```



4) *Our main focus with these data will be on estimating the long-term trend, starting with polynomial trend models. But first, check for seasonality in a model that accounts for a linear time trend. Use our previous fractional year for the trend. Report an* `effects` *plot and a test for the month model component.*

Note: when you use `time()` *to generate the* `Year` *variable from a time series object it retains some time series object information that can cause conflicts later on. Create a new variable in your data.frame that uses something like* `as.vector(time(tsdataname))`*.*

```
hadcrut$yearfrac <- as.vector(time(hadcrut.ts))
hadcrut$month <- factor(round(1 + 12 * (hadcrut$yearfrac %% 1), 0))
had.lm1 <- lm(median.temp.anom ~ month + yearfrac, data = hadcrut)
summary(had.lm1)
```

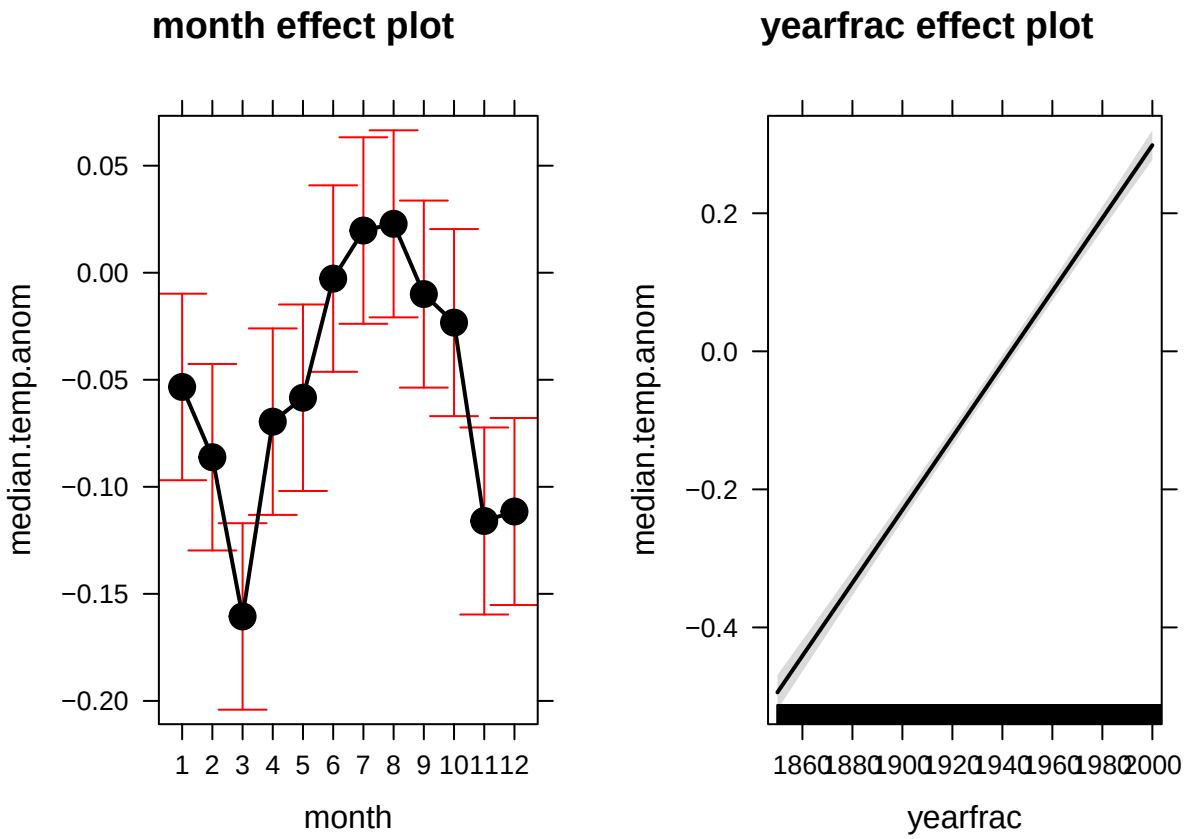
```
##
## Call:
## lm(formula = median.temp.anom ~ month + yearfrac, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24294 -0.19385 -0.02191  0.18118  1.15426
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.027e+01  2.590e-01 -39.654  < 2e-16 ***
## month2      -3.281e-02  3.141e-02  -1.045 0.296288
## month3      -1.072e-01  3.141e-02  -3.413 0.000656 ***
## month4      -1.621e-02  3.141e-02  -0.516 0.605890
## month5      -5.031e-03  3.141e-02  -0.160 0.872754
```

```
## month6       5.062e-02  3.141e-02   1.612 0.107154
## month7       7.306e-02  3.141e-02   2.326 0.020102 *
## month8       7.620e-02  3.145e-02   2.423 0.015502 *
## month9       4.339e-02  3.145e-02   1.379 0.167953
## month10      3.006e-02  3.145e-02   0.956 0.339359
## month11     -6.260e-02  3.145e-02  -1.990 0.046701 *
## month12     -5.818e-02  3.145e-02  -1.850 0.064507 .
## yearfrac     5.285e-03  1.335e-04  39.588  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.287 on 1986 degrees of freedom
## Multiple R-squared:  0.4525, Adjusted R-squared:  0.4492
## F-statistic: 136.8 on 12 and 1986 DF,  p-value: < 2.2e-16
```

```
library(car)
Anova(had.lm1, test = "F", type = "III") #test for the month component
```

```
## Anova Table (Type III tests)
##
## Response: median.temp.anom
##              Sum Sq   Df  F value    Pr(>F)
## (Intercept) 129.517    1 1572.417 < 2.2e-16 ***
## month         6.104   11    6.737 3.389e-11 ***
## yearfrac    129.090    1 1567.241 < 2.2e-16 ***
## Residuals   163.583 1986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

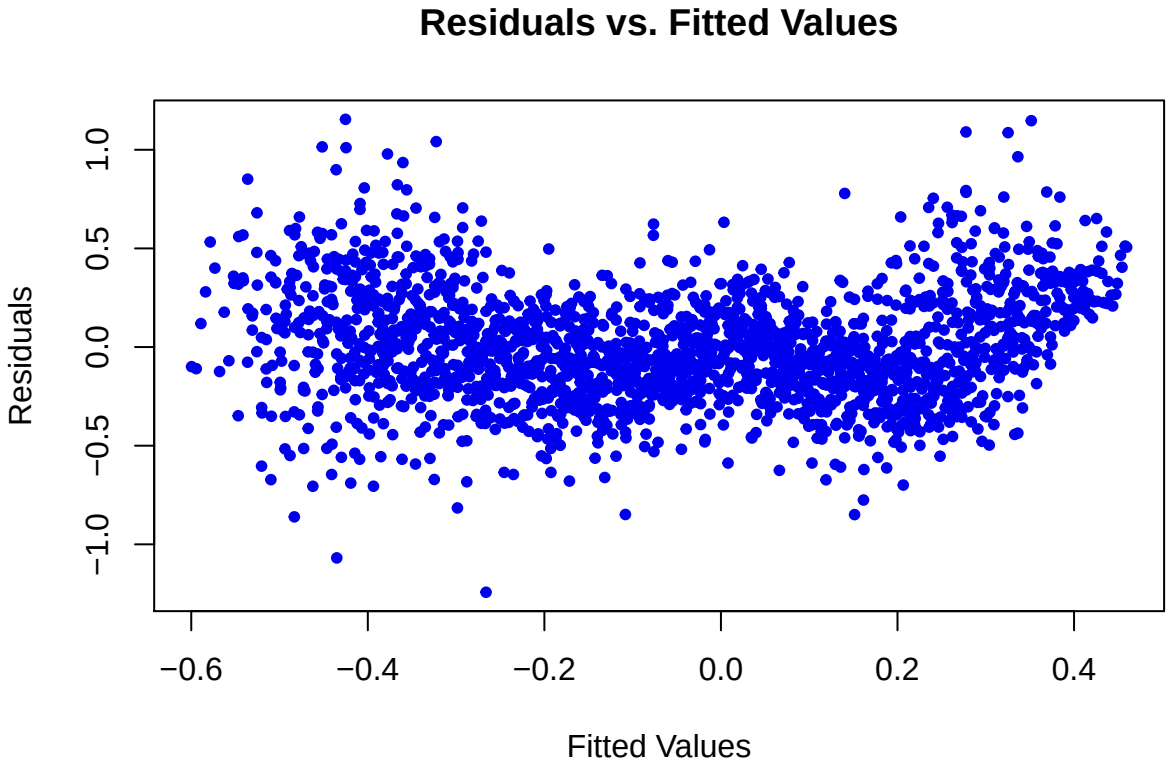
```
require(effects)
plot(allEffects(had.lm1))
```



5) *Check the residuals versus fitted values for any evidence of nonlinearity in the residuals vs fitted that was missed by the model with a linear trend and month component. Also note any potential issues with the constant variance assumption.*

Based on the plot below, there appears to be a wavy pattern occuring in the residuals over time that is not being accounted for in the model with a linear trend and month component. Furthermore, the spread of observations is not constant; it appears that the spread for smaller fitted values is wider near fitted values of -0.5 and much narrower for larger fitted values.

```
plot(had.lm1$fitted.values, had.lm1$residuals, pch = 20, col = "blue2", main = "Residuals vs. Fitte
```



6) *You can add higher order polynomial terms to models using* `x1+I(x1^2)+I(x1^3)...` *or using the* `poly` *function, such as* `poly(x1,3,raw=T)` *for a cubic polynomial that includes the linear and quadratic components (we want this!). The* `raw=T` *keeps the variables in their raw or input format. Estimate the same model but now using polynomial trends that steps up from linear (poly(time,1,raw=T)) and stop when you get a failure to estimate a part of the model. Briefly discuss what happened.*

The following code estimates the ith-degree polynomial, where i ranges from 1 to 10. Thus, we see that from the returned summary output we get estimates for the coefficients up to the 5th degree polynomial. (Note that had we successfully estimated every part of the model for all 10 coefficients, we would have simply expanded our for-loop to contain more iterations.) It appears that the 5th degree

```
for (i in 1:10){
  had.lm.poly = lm(median.temp.anom ~ poly(yearfrac,i,raw = TRUE) + month, data = hadcrut)
  print(summary(had.lm.poly))
}
```

```
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24294 -0.19385 -0.02191  0.18118  1.15426
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.027e+01  2.590e-01 -39.654  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)  5.285e-03  1.335e-04  39.588  < 2e-16 ***
```

```
## month2                        -3.281e-02  3.141e-02  -1.045 0.296288
## month3                        -1.072e-01  3.141e-02  -3.413 0.000656 ***
## month4                        -1.621e-02  3.141e-02  -0.516 0.605890
## month5                        -5.031e-03  3.141e-02  -0.160 0.872754
## month6                         5.062e-02  3.141e-02   1.612 0.107154
## month7                         7.306e-02  3.141e-02   2.326 0.020102 *
## month8                         7.620e-02  3.145e-02   2.423 0.015502 *
## month9                         4.339e-02  3.145e-02   1.379 0.167953
## month10                        3.006e-02  3.145e-02   0.956 0.339359
## month11                       -6.260e-02  3.145e-02  -1.990 0.046701 *
## month12                       -5.818e-02  3.145e-02  -1.850 0.064507 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.287 on 1986 degrees of freedom
## Multiple R-squared:  0.4525, Adjusted R-squared:  0.4492
## F-statistic: 136.8 on 12 and 1986 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24912 -0.15751  0.00625  0.15731  1.04262
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.223e+02  1.036e+01  21.460  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)1 -2.354e-01  1.072e-02 -21.963  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)2  6.225e-05  2.772e-06  22.457  < 2e-16 ***
## month2                         -3.281e-02  2.805e-02  -1.170 0.242328
## month3                         -1.072e-01  2.805e-02  -3.821 0.000137 ***
## month4                         -1.620e-02  2.805e-02  -0.578 0.563604
## month5                         -5.027e-03  2.805e-02  -0.179 0.857790
## month6                          5.063e-02  2.805e-02   1.805 0.071281 .
## month7                          7.306e-02  2.805e-02   2.604 0.009270 **
## month8                          7.793e-02  2.810e-02   2.774 0.005594 **
## month9                          4.512e-02  2.810e-02   1.606 0.108469
## month10                         3.179e-02  2.810e-02   1.132 0.257961
## month11                        -6.087e-02  2.810e-02  -2.167 0.030385 *
## month12                        -5.645e-02  2.810e-02  -2.009 0.044643 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2563 on 1985 degrees of freedom
## Multiple R-squared:  0.5634, Adjusted R-squared:  0.5606
## F-statistic: 197.1 on 13 and 1985 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24359 -0.15689  0.00191  0.15217  1.04117
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -2.893e+03  4.687e+02  -6.172 8.15e-10 ***
## poly(yearfrac, i, raw = TRUE)1  4.602e+00  7.278e-01   6.324 3.15e-10 ***
## poly(yearfrac, i, raw = TRUE)2 -2.441e-03  3.766e-04  -6.482 1.14e-10 ***
## poly(yearfrac, i, raw = TRUE)3  4.316e-07  6.493e-08   6.648 3.83e-11 ***
## month2                         -3.291e-02  2.775e-02  -1.186 0.235830
## month3                         -1.074e-01  2.775e-02  -3.870 0.000113 ***
## month4                         -1.651e-02  2.775e-02  -0.595 0.552067
## month5                         -5.432e-03  2.775e-02  -0.196 0.844853
## month6                          5.012e-02  2.775e-02   1.806 0.071075 .
## month7                          7.246e-02  2.775e-02   2.611 0.009101 **
## month8                          7.782e-02  2.779e-02   2.800 0.005162 **
## month9                          4.491e-02  2.779e-02   1.616 0.106289
## month10                         3.149e-02  2.779e-02   1.133 0.257400
## month11                        -6.127e-02  2.780e-02  -2.204 0.027606 *
## month12                        -5.695e-02  2.780e-02  -2.049 0.040596 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2536 on 1984 degrees of freedom
## Multiple R-squared:  0.573,  Adjusted R-squared:  0.5699
## F-statistic: 190.1 on 14 and 1984 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22122 -0.15262 -0.00671  0.14672  1.09471
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.146e+05  2.102e+04  10.209  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)1 -4.458e+02  4.353e+01 -10.242  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)2  3.472e-01  3.379e-02  10.277  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)3 -1.202e-04  1.165e-05 -10.312  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)4  1.560e-08  1.507e-09  10.349  < 2e-16 ***
## month2                         -3.291e-02  2.704e-02  -1.217  0.22376
## month3                         -1.074e-01  2.704e-02  -3.971  7.4e-05 ***
## month4                         -1.650e-02  2.704e-02  -0.610  0.54181
## month5                         -5.425e-03  2.704e-02  -0.201  0.84101
## month6                          5.013e-02  2.704e-02   1.854  0.06392 .
## month7                          7.246e-02  2.704e-02   2.680  0.00743 **
## month8                          7.885e-02  2.708e-02   2.912  0.00363 **
## month9                          4.594e-02  2.708e-02   1.697  0.08993 .
## month10                         3.252e-02  2.708e-02   1.201  0.22992
```

```
## month11                         -6.024e-02  2.708e-02  -2.224  0.02623 *
## month12                         -5.592e-02  2.708e-02  -2.065  0.03905 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2471 on 1983 degrees of freedom
## Multiple R-squared:  0.5948, Adjusted R-squared:  0.5918
## F-statistic: 194.1 on 15 and 1983 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22122 -0.15262 -0.00671  0.14672  1.09471
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.146e+05  2.102e+04  10.209  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)1 -4.458e+02  4.353e+01 -10.242  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)2  3.472e-01  3.379e-02  10.277  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)3 -1.202e-04  1.165e-05 -10.312  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)4  1.560e-08  1.507e-09  10.349  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## month2                          -3.291e-02  2.704e-02  -1.217  0.22376
## month3                          -1.074e-01  2.704e-02  -3.971  7.4e-05 ***
## month4                          -1.650e-02  2.704e-02  -0.610  0.54181
## month5                          -5.425e-03  2.704e-02  -0.201  0.84101
## month6                           5.013e-02  2.704e-02   1.854  0.06392 .
## month7                           7.246e-02  2.704e-02   2.680  0.00743 **
## month8                           7.885e-02  2.708e-02   2.912  0.00363 **
## month9                           4.594e-02  2.708e-02   1.697  0.08993 .
## month10                          3.252e-02  2.708e-02   1.201  0.22992
## month11                         -6.024e-02  2.708e-02  -2.224  0.02623 *
## month12                         -5.592e-02  2.708e-02  -2.065  0.03905 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2471 on 1983 degrees of freedom
## Multiple R-squared:  0.5948, Adjusted R-squared:  0.5918
## F-statistic: 194.1 on 15 and 1983 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.22122 -0.15262 -0.00671  0.14672  1.09471
##
## Coefficients: (2 not defined because of singularities)
```

```
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.146e+05  2.102e+04  10.209  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)1 -4.458e+02  4.353e+01 -10.242  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)2  3.472e-01  3.379e-02  10.277  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)3 -1.202e-04  1.165e-05 -10.312  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)4  1.560e-08  1.507e-09  10.349  < 2e-16 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)6         NA         NA      NA       NA
## month2                         -3.291e-02  2.704e-02  -1.217  0.22376
## month3                         -1.074e-01  2.704e-02  -3.971  7.4e-05 ***
## month4                         -1.650e-02  2.704e-02  -0.610  0.54181
## month5                         -5.425e-03  2.704e-02  -0.201  0.84101
## month6                          5.013e-02  2.704e-02   1.854  0.06392 .
## month7                          7.246e-02  2.704e-02   2.680  0.00743 **
## month8                          7.885e-02  2.708e-02   2.912  0.00363 **
## month9                          4.594e-02  2.708e-02   1.697  0.08993 .
## month10                         3.252e-02  2.708e-02   1.201  0.22992
## month11                        -6.024e-02  2.708e-02  -2.224  0.02623 *
## month12                        -5.592e-02  2.708e-02  -2.065  0.03905 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2471 on 1983 degrees of freedom
## Multiple R-squared:  0.5948, Adjusted R-squared:  0.5918
## F-statistic: 194.1 on 15 and 1983 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24512 -0.15285 -0.00182  0.14408  1.04866
##
## Coefficients: (2 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -3.945e+06  6.874e+05  -5.740 1.10e-08 ***
## poly(yearfrac, i, raw = TRUE)1  9.605e+03  1.661e+03   5.784 8.46e-09 ***
## poly(yearfrac, i, raw = TRUE)2 -9.017e+00  1.547e+00  -5.829 6.51e-09 ***
## poly(yearfrac, i, raw = TRUE)3  3.918e-03  6.671e-04   5.873 4.99e-09 ***
## poly(yearfrac, i, raw = TRUE)4 -6.808e-07  1.150e-07  -5.918 3.82e-09 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)6         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)7  2.753e-18  4.546e-19   6.055 1.68e-09 ***
## month2                         -3.302e-02  2.680e-02  -1.232  0.21805
## month3                         -1.076e-01  2.680e-02  -4.015 6.16e-05 ***
## month4                         -1.684e-02  2.680e-02  -0.628  0.52983
## month5                         -5.880e-03  2.680e-02  -0.219  0.82635
## month6                          4.956e-02  2.680e-02   1.849  0.06459 .
## month7                          7.177e-02  2.680e-02   2.678  0.00746 **
## month8                          7.873e-02  2.684e-02   2.933  0.00339 **
## month9                          4.571e-02  2.684e-02   1.703  0.08870 .
## month10                         3.219e-02  2.684e-02   1.199  0.23061
```

```
## month11                         -6.068e-02  2.684e-02  -2.261  0.02388 *
## month12                         -5.647e-02  2.684e-02  -2.104  0.03552 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2449 on 1982 degrees of freedom
## Multiple R-squared:  0.6022, Adjusted R-squared:  0.599
## F-statistic: 187.5 on 16 and 1982 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24512 -0.15285 -0.00182  0.14408  1.04866
##
## Coefficients: (3 not defined because of singularities)
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -3.945e+06  6.874e+05  -5.740 1.10e-08 ***
## poly(yearfrac, i, raw = TRUE)1  9.605e+03  1.661e+03   5.784 8.46e-09 ***
## poly(yearfrac, i, raw = TRUE)2 -9.017e+00  1.547e+00  -5.829 6.51e-09 ***
## poly(yearfrac, i, raw = TRUE)3  3.918e-03  6.671e-04   5.873 4.99e-09 ***
## poly(yearfrac, i, raw = TRUE)4 -6.808e-07  1.150e-07  -5.918 3.82e-09 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)6         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)7  2.753e-18  4.546e-19   6.055 1.68e-09 ***
## poly(yearfrac, i, raw = TRUE)8         NA         NA      NA       NA
## month2                          -3.302e-02  2.680e-02  -1.232  0.21805
## month3                          -1.076e-01  2.680e-02  -4.015 6.16e-05 ***
## month4                          -1.684e-02  2.680e-02  -0.628  0.52983
## month5                          -5.880e-03  2.680e-02  -0.219  0.82635
## month6                           4.956e-02  2.680e-02   1.849  0.06459 .
## month7                           7.177e-02  2.680e-02   2.678  0.00746 **
## month8                           7.873e-02  2.684e-02   2.933  0.00339 **
## month9                           4.571e-02  2.684e-02   1.703  0.08870 .
## month10                          3.219e-02  2.684e-02   1.199  0.23061
## month11                         -6.068e-02  2.684e-02  -2.261  0.02388 *
## month12                         -5.647e-02  2.684e-02  -2.104  0.03552 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2449 on 1982 degrees of freedom
## Multiple R-squared:  0.6022, Adjusted R-squared:  0.599
## F-statistic: 187.5 on 16 and 1982 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.24512 -0.15285 -0.00182  0.14408  1.04866
##
## Coefficients: (4 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -3.945e+06  6.874e+05  -5.740 1.10e-08 ***
## poly(yearfrac, i, raw = TRUE)1  9.605e+03  1.661e+03   5.784 8.46e-09 ***
## poly(yearfrac, i, raw = TRUE)2 -9.017e+00  1.547e+00  -5.829 6.51e-09 ***
## poly(yearfrac, i, raw = TRUE)3  3.918e-03  6.671e-04   5.873 4.99e-09 ***
## poly(yearfrac, i, raw = TRUE)4 -6.808e-07  1.150e-07  -5.918 3.82e-09 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)6         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)7  2.753e-18  4.546e-19   6.055 1.68e-09 ***
## poly(yearfrac, i, raw = TRUE)8         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)9         NA         NA      NA       NA
## month2                        -3.302e-02  2.680e-02  -1.232  0.21805
## month3                        -1.076e-01  2.680e-02  -4.015 6.16e-05 ***
## month4                        -1.684e-02  2.680e-02  -0.628  0.52983
## month5                        -5.880e-03  2.680e-02  -0.219  0.82635
## month6                         4.956e-02  2.680e-02   1.849  0.06459 .
## month7                         7.177e-02  2.680e-02   2.678  0.00746 **
## month8                         7.873e-02  2.684e-02   2.933  0.00339 **
## month9                         4.571e-02  2.684e-02   1.703  0.08870 .
## month10                        3.219e-02  2.684e-02   1.199  0.23061
## month11                       -6.068e-02  2.684e-02  -2.261  0.02388 *
## month12                       -5.647e-02  2.684e-02  -2.104  0.03552 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2449 on 1982 degrees of freedom
## Multiple R-squared:  0.6022, Adjusted R-squared:  0.599
## F-statistic: 187.5 on 16 and 1982 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = median.temp.anom ~ poly(yearfrac, i, raw = TRUE) +
##     month, data = hadcrut)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24512 -0.15285 -0.00182  0.14408  1.04866
##
## Coefficients: (5 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -3.945e+06  6.874e+05  -5.740 1.10e-08 ***
## poly(yearfrac, i, raw = TRUE)1  9.605e+03  1.661e+03   5.784 8.46e-09 ***
## poly(yearfrac, i, raw = TRUE)2 -9.017e+00  1.547e+00  -5.829 6.51e-09 ***
## poly(yearfrac, i, raw = TRUE)3  3.918e-03  6.671e-04   5.873 4.99e-09 ***
## poly(yearfrac, i, raw = TRUE)4 -6.808e-07  1.150e-07  -5.918 3.82e-09 ***
## poly(yearfrac, i, raw = TRUE)5         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)6         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)7  2.753e-18  4.546e-19   6.055 1.68e-09 ***
## poly(yearfrac, i, raw = TRUE)8         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)9         NA         NA      NA       NA
## poly(yearfrac, i, raw = TRUE)10        NA         NA      NA       NA
```

```
## month2                          -3.302e-02  2.680e-02  -1.232  0.21805
## month3                          -1.076e-01  2.680e-02  -4.015 6.16e-05 ***
## month4                          -1.684e-02  2.680e-02  -0.628  0.52983
## month5                          -5.880e-03  2.680e-02  -0.219  0.82635
## month6                           4.956e-02  2.680e-02   1.849  0.06459 .
## month7                           7.177e-02  2.680e-02   2.678  0.00746 **
## month8                           7.873e-02  2.684e-02   2.933  0.00339 **
## month9                           4.571e-02  2.684e-02   1.703  0.08870 .
## month10                          3.219e-02  2.684e-02   1.199  0.23061
## month11                         -6.068e-02  2.684e-02  -2.261  0.02388 *
## month12                         -5.647e-02  2.684e-02  -2.104  0.03552 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2449 on 1982 degrees of freedom
## Multiple R-squared:  0.6022, Adjusted R-squared:  0.599
## F-statistic: 187.5 on 16 and 1982 DF,  p-value: < 2.2e-16
```

7) *If we center or, even better, make the polynomial functions orthogonal to one another, we can avoid the issue in the previous question. Use* `poly(x1,?,raw=F)` *and step up the polynomial order for time until the p-value for the last coefficient (use* `summary()`*) is "large", reporting the single test result for each step in the building process. Then drop back one order and re-fit the model. Report the* `effects` *plot of the resulting model and describe the estimated trend. Note: aside from access to orthogonal polynomials the* `poly` *function interfaces with* `Anova` *and the* `effects` *package really nicely.*

The first coefficient with a "large" p-value was the 11th-order coefficient. Thus, we re-fit the model with a 10th order polynomial. The effects plot of the resulting model is also included.

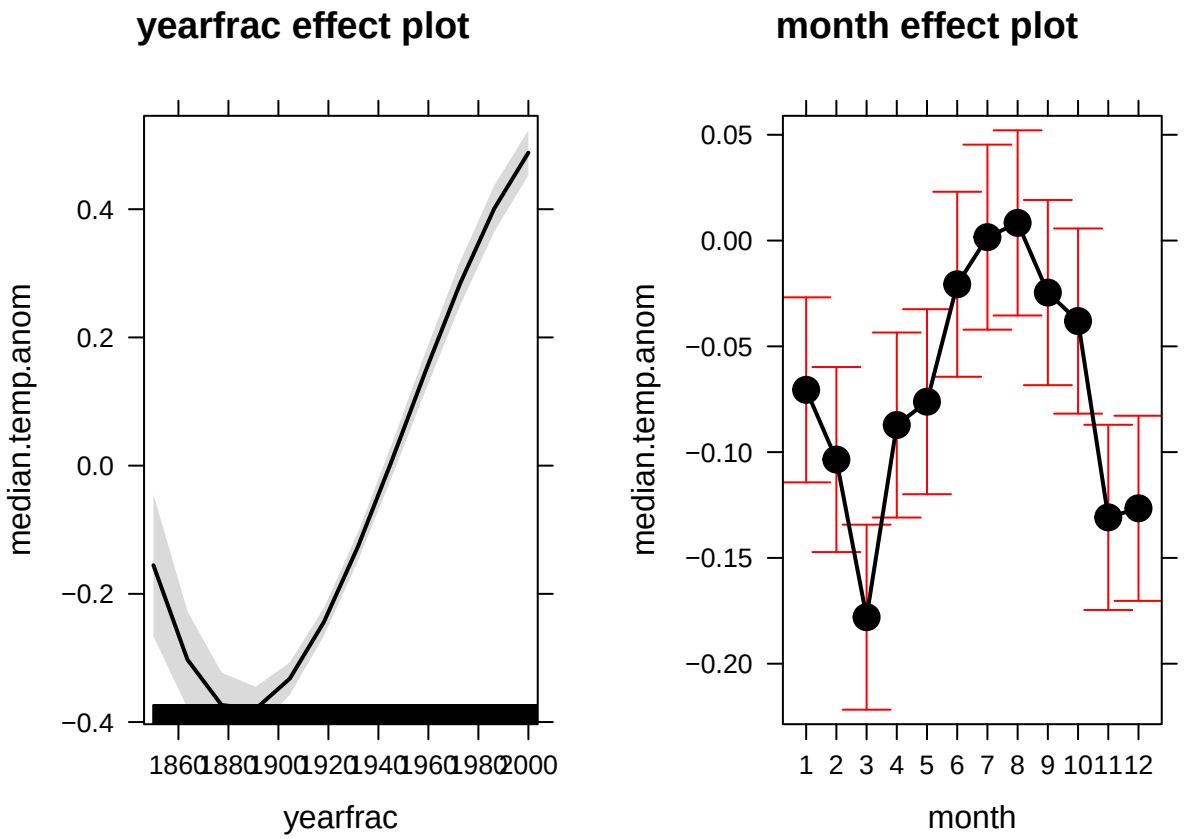
```
pvals <- numeric(20)
for (i in 1:20){
  lm.poly.1 = lm(median.temp.anom ~ poly(yearfrac,i,raw = FALSE)+ month, data = hadcrut)
  pvals[i] = (summary(lm.poly.1)$coefficients[i,4])
}
pvals
```

```
##  [1]  1.636397e-02 7.438213e-299 1.326403e-101  1.178341e-11  6.846030e-25
##  [6]  8.573235e-10  5.019343e-11  5.928416e-23  6.840845e-08  3.317819e-09
## [11]  1.707979e-01  4.728140e-03  4.805653e-01  1.663016e-09  3.600367e-03
## [16]  1.215671e-04  3.413115e-02  2.194461e-01  5.972341e-03  6.220013e-01
```

```
which(pvals > 0.05)
```

```
## [1] 11 13 18 20
```

```
lm.poly.final = lm(median.temp.anom ~ poly(yearfrac,10,raw = FALSE) + month , data = hadcrut)
plot(allEffects(lm.poly.final))
```

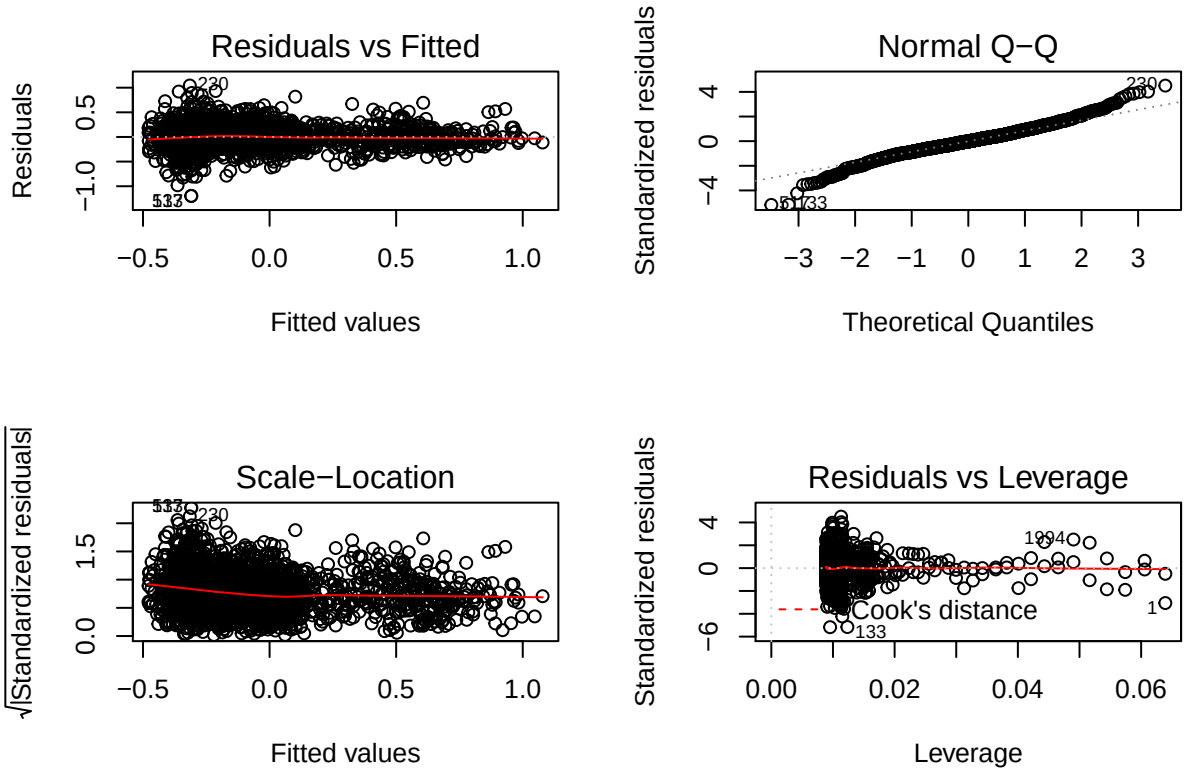


8) *Check the diagnostic plots from your final model. Does anything improve from the first version. Also plot the residuals vs time and compare that plot to residuals vs fitted.*

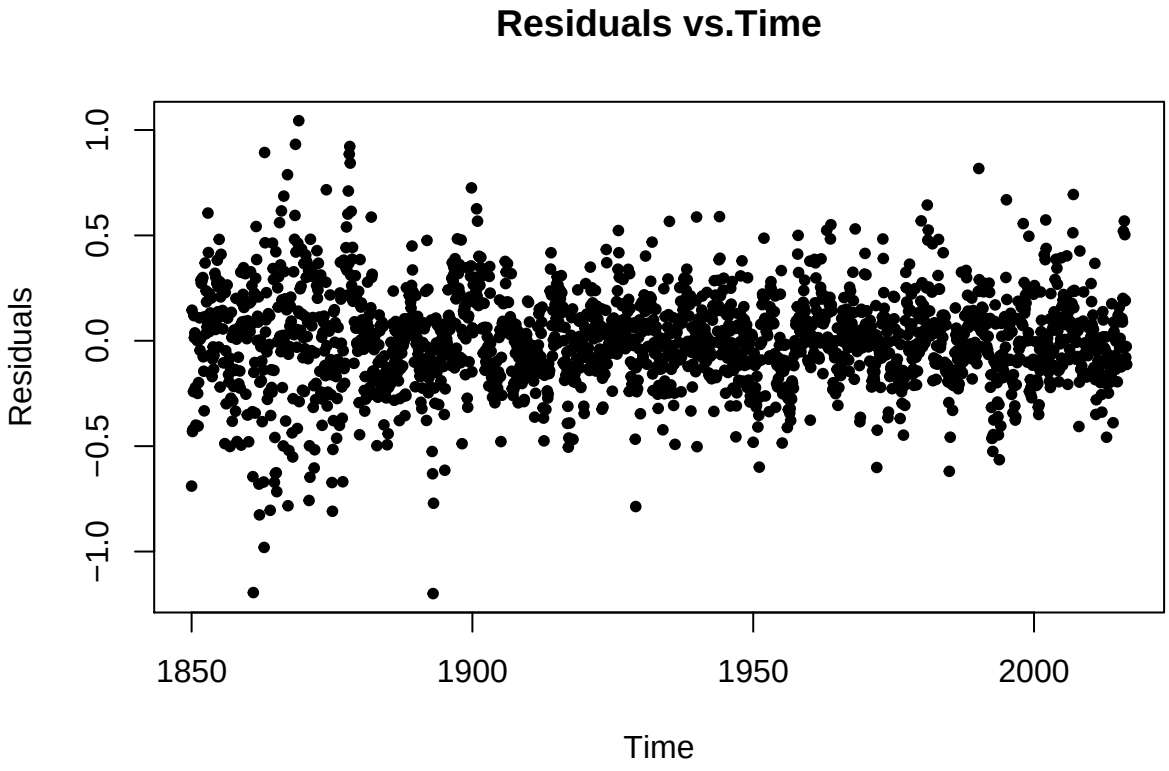
Based on the plots below, we can see that...

Further, the plot of residuals vs. time shows...

```
par(mfrow = c(2,2))
plot(lm.poly.final)
```




```
par(mfrow = c(1,1))
plot(hadcrut$yearfrac,resid(lm.poly.final), main = "Residuals vs.Time",xlab = "Time", ylab = "Resid
```



9) *Run the following code so I can see what version of R you are now using:*

Documenting R version

```
sessionInfo()$R.version$nickname
```

```
## [1] "Bug in Your Hair"
```

```
getRversion()
```

```
## [1] '3.3.1'
```

```
sessionInfo()$platform
```

```
## [1] "x86_64-pc-linux-gnu (64-bit)"
```