Time Series HW 3

Justin Gomez and Andrea Mack

**Disclaimer:** Justin and Andrea are new partners, we have never worked together in one of Dr. Greenwood's classes on an assignment until now.

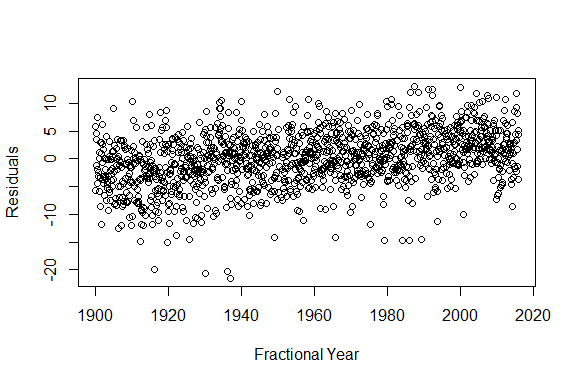
## HW 3

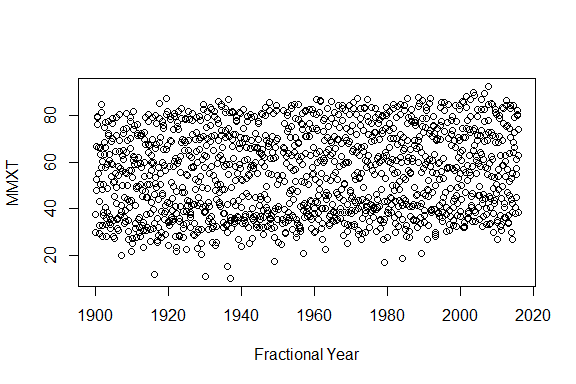
You can alone or in pairs that may or may not be people you worked with before. You can discuss it with old partners but should try work as much as possible with new collaborators. 5% bonus if you find someone completely new to work with - that you did not work with on first two assignments.

I mentioned de-seasonalizing of time series, where the seasonal variation is removed from the series to highlight variation at either higher or lower frequencies. There are a variety of techniques for doing this but the simplest is to just subtract the mean for each month from the observations. And the easiest way to find that is using lm(y~month,data=...).

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.

*Comparing the plot of the residuals and the fractional year to the plot of the max monthly temperatures and the fractional year, we can see that the noise has been reduced in the residuals plot, and a slight positive linear trend is observed. Unusual points are also more apparent in the plot of the residuals as they are slightly set apart from the bulk of the data, compared to the plot of the max monthly themperatures with so much noise that extreme points seem to fall in with the rest of the data.*





1. 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).

*The degrees of freedom change when we are estimating more parameters, and the F statistic changes as well. However, both models resulted in pvalues that were very, very small.*

|  |  |  |  |
| --- | --- | --- | --- |
| model | test stat | df | pvalue |
| resid year | Fstat=195.9 |  | <0.00001 |
| temp year+monthf | Fstat=1628 |  | <0.00001 |
|  |  |  |  |

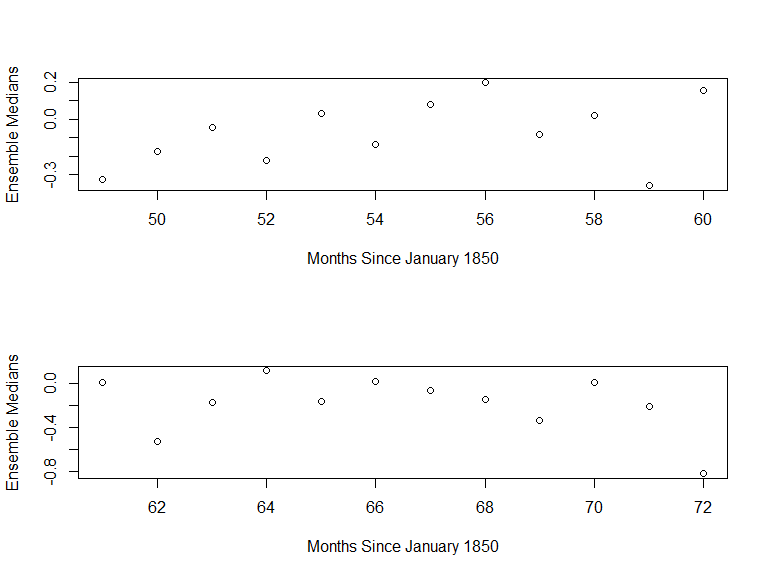
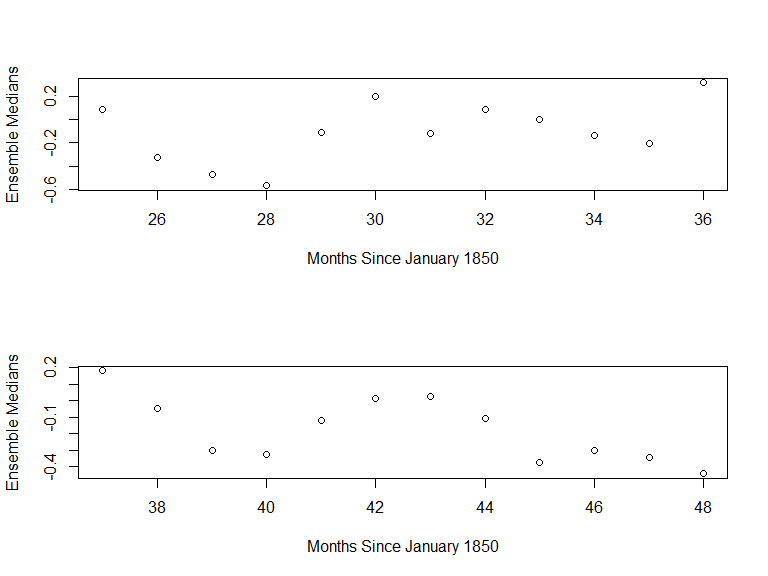
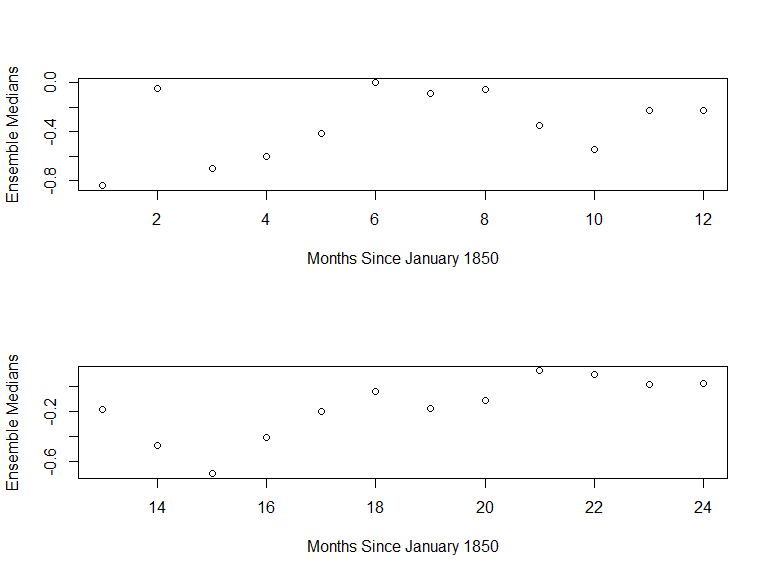
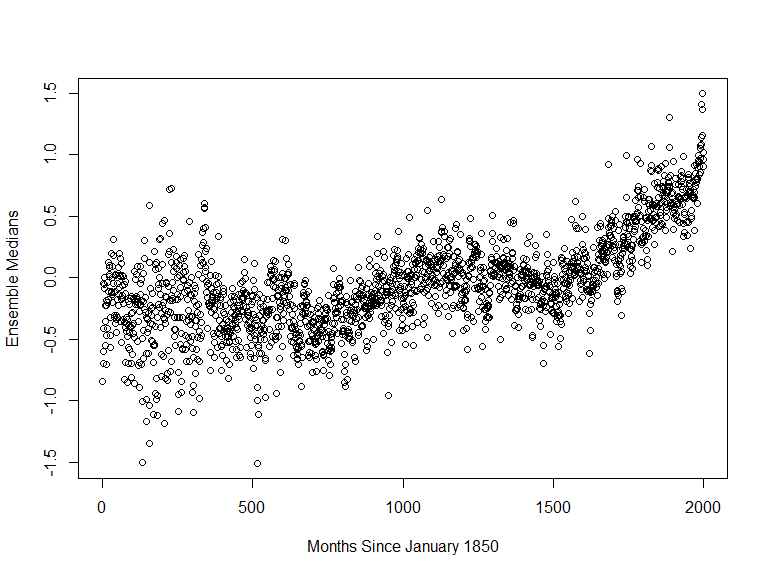
1. I briefly discussed the HADCRUT data set in class. We will consider the HADCRUT4 series of temperature anomalies for the Nothern Hemisphere. The fully up to date data set is available at: <http://www.metoffice.gov.uk/hadobs/hadcrut4/data/current/time_series/HadCRUT.4.4.0.0.monthly_nh.txt>

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.

*Assessing the plot of the ensemble medians versus time, we can see an overall positive trend since January of 1850. The trend has curvature, suggesting a cyclic trend in ensemble medians post 1900. It will be interesting to see if we are currently at a peak in ensemble medians. The resolution of the plot below does not allow us to visally assess seasonality trends. The variability appears to be larger in the first 50 years of the data set as we can see a range of observations between about -1.5 and 0.7. In the later years, this spread appears to have decreased. Across the entire 156 year interval, there are a few observations that stand out from the rest of the data as outliers, especially in the first 50 years.*

*The next six plots are zoomed in to the first six years. There appears to be seasonality that is visible in the first half of the first four years. Ensemble medians decrease in the first quarter, and then increase in the second quarter of these years. We cannot see seasonality in the third and fourth quarters of these first four years and years five and six do not appear to have a consistent seasonality in them. Other seasonality patterns may or may not emerge in later years*

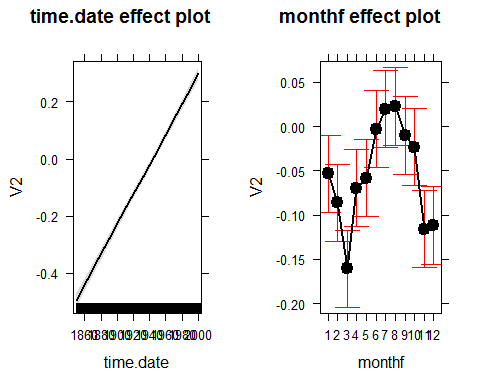


4) Our main focus with these data will be on estimating the long-term trend, starting with polynomial trend models. But first, check for seasonality in a model that accounts for a linear time trend. Use our previous fractional year for the trend. Report an effects plot and a test for the month model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

: all Bmonth\_j = 0

: at least one Bmonth\_j != 0 where != means "not equal to"

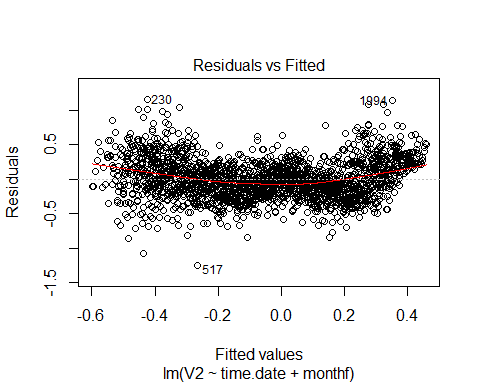
*Based on an F-stat of 6.74 on 11 and 1986 degrees of freedom, using type II sums of squares, with an associated pvalue of < 0.001, there is strong evidence that the true mean of the median ensemble changes by month after accounting for fractional date. Examining the effect plot for the monthly factor, we see seasonality in the ensemble medians with average ensemble medians decreasing in January to March, increasing until August, and then decreasing until November, where there is a slight increase again in December.*



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)).

1. 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.

*Viewing the residuals vs. fitted values plot below, there appears to be a quadratic relationship between the fitted values and residuals that was not captured by the model. For fitted values less than -0.3, there appears to be more variation in the residuals than for fitted values above -0.3, indicating the constant variance assumption may be violated. The model is less accurately predicting median ensembles when the fitted median ensemble was below -0.3, which means that we may have left out a variable or structure that explains when median ensemble was lower than expected.*



1. 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.

When I got to the fifth degree polynomial, the estimate could not be computed. WHY? My guess is that it has something to do with the seasonality of the time.data variable, or it be perfectly collinear with another variable.

*We can see that the estimate for the fifth degree term in the model cannot be computed, while all lower order terms could be. Examining the estimates themselves, we can see the intercept is a large number, as is the estimate for the first degree term. The estimate for the second degree term is relatively small compared to the first two estimates, with the third and fourth degree terms being even smaller. One reason the fifth degree term returns NAs could be that it is almost nearly collinear with one or a combination of the other predictors, making the coefficient for the fifth degree term not estimable.*

*Partial output from the model with a 5th degree polynomial term in it. You will notice that the output does not include a line for the 5th degree term. This is a formatting issue we were having and could not get resolved. The actual output for the 5th degree term returns NA's, which means that that coefficient could not be estimated.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr(>|t|) |
| (Intercept) | 2.146402e+05 | 2.102369e+04 | 10.2094424 | 0.0000000 |
| poly(time.date, i, raw = TRUE)1 | -4.458441e+02 | 4.352932e+01 | -10.2423869 | 0.0000000 |
| poly(time.date, i, raw = TRUE)2 | 3.472438e-01 | 3.378960e-02 | 10.2766371 | 0.0000000 |
| poly(time.date, i, raw = TRUE)3 | -1.202000e-04 | 1.170000e-05 | -10.3123040 | 0.0000000 |
| poly(time.date, i, raw = TRUE)4 | 0.000000e+00 | 0.000000e+00 | 10.3494916 | 0.0000000 |
| monthf2 | -3.290620e-02 | 2.703940e-02 | -1.2169716 | 0.2237598 |
| monthf3 | -1.073839e-01 | 2.703940e-02 | -3.9713798 | 0.0000740 |
| monthf4 | -1.649880e-02 | 2.703950e-02 | -0.6101762 | 0.5418150 |
| monthf5 | -5.424800e-03 | 2.703950e-02 | -0.2006249 | 0.8410124 |
| monthf6 | 5.012570e-02 | 2.703960e-02 | 1.8537911 | 0.0639175 |
| monthf7 | 7.245800e-02 | 2.703960e-02 | 2.6796976 | 0.0074298 |
| monthf8 | 7.885290e-02 | 2.708040e-02 | 2.9118085 | 0.0036336 |
| monthf9 | 4.594480e-02 | 2.708040e-02 | 1.6966074 | 0.0899278 |
| monthf10 | 3.252210e-02 | 2.708040e-02 | 1.2009454 | 0.2299158 |
| monthf11 | -6.024050e-02 | 2.708050e-02 | -2.2244985 | 0.0262271 |
| monthf12 | -5.592130e-02 | 2.708050e-02 | -2.0650010 | 0.0390526 |
| | | Estimate| Std. Error| t value| Pr(>|t|)| |
| |:-------------------------------|-------------:|------------:|-----------:|------------------:| |
| |(Intercept) | 2.146402e+05| 2.102369e+04| 10.2094424| 0.0000000| |
| |poly(time.date, i, raw = TRUE)1 | -4.458441e+02| 4.352932e+01| -10.2423869| 0.0000000| |
| |poly(time.date, i, raw = TRUE)2 | 3.472438e-01| 3.378960e-02| 10.2766371| 0.0000000| |
| |poly(time.date, i, raw = TRUE)3 | -1.202000e-04| 1.170000e-05| -10.3123040| 0.0000000| |
| |poly(time.date, i, raw = TRUE)4 | 0.000000e+00| 0.000000e+00| 10.3494916| 0.0000000| |
| |monthf2 | -3.290620e-02| 2.703940e-02| -1.2169716| 0.2237598| |
| |monthf3 | -1.073839e-01| 2.703940e-02| -3.9713798| 0.0000740| |
| |monthf4 | -1.649880e-02| 2.703950e-02| -0.6101762| 0.5418150| |
| |monthf5 | -5.424800e-03| 2.703950e-02| -0.2006249| 0.8410124| |
| |monthf6 | 5.012570e-02| 2.703960e-02| 1.8537911| 0.0639175| |
| |monthf7 | 7.245800e-02| 2.703960e-02| 2.6796976| 0.0074298| |
| |monthf8 | 7.885290e-02| 2.708040e-02| 2.9118085| 0.0036336| |
| |monthf9 | 4.594480e-02| 2.708040e-02| 1.6966074| 0.0899278| |
| |monthf10 | 3.252210e-02| 2.708040e-02| 1.2009454| 0.2299158| |
| |monthf11 | -6.024050e-02| 2.708050e-02| -2.2244985| 0.0262271| |
| |monthf12 | -5.592130e-02| 2.708050e-02| -2.0650010| 0.0390526| |
|  |

1. 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 notation we are using assumes the intercept is B0, the linear term is B1, the quadratic term is B2, etc. The results from testing the largest polynomial term in each model are given below. The model with a 10th degree polynomial has a pvalue of 0.17 that provides no evidence that the coefficient of the 10th degree polynomial is different from zero, after considering the information in first 9 polynomial terms and the month factor terms.*

Ho: B 0 = 0 Ha B 0 tstat = -2.402741 df = 1986 pvalue = 0.01636397

Ho: B 1 = 0 Ha B 1 tstat = 44.32188 df = 1985 pvalue = 7.438213e-299

Ho: B 2 = 0 Ha B 2 tstat = 22.69996 df = 1984 pvalue = 1.326403e-101

Ho: B 3 = 0 Ha B 3 tstat = 6.823243 df = 1983 pvalue = 1.178341e-11

Ho: B 4 = 0 Ha B 4 tstat = 10.44361 df = 1982 pvalue = 6.84603e-25

Ho: B 5 = 0 Ha B 5 tstat = 6.163959 df = 1981 pvalue = 8.573235e-10

Ho: B 6 = 0 Ha B 6 tstat = -6.607179 df = 1980 pvalue = 5.019343e-11

Ho: B 7 = 0 Ha B 7 tstat = -9.988379 df = 1979 pvalue = 5.928416e-23

Ho: B 8 = 0 Ha B 8 tstat = 5.415902 df = 1978 pvalue = 6.840845e-08

Ho: B 9 = 0 Ha B 9 tstat = 5.94207 df = 1977 pvalue = 3.317819e-09

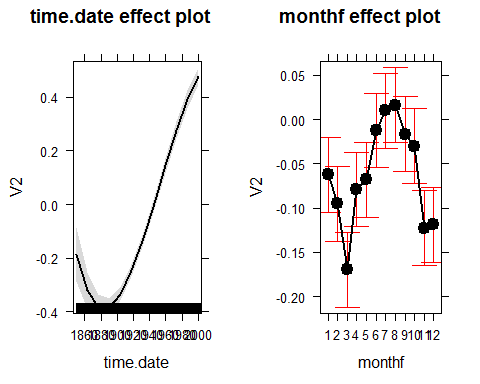
Ho: B 10 = 0 Ha B 10 tstat = 1.370143 df = 1976 pvalue = 0.1707979

Ho: B 11 = 0 Ha B 11 tstat = 2.828209 df = 1975 pvalue = 0.00472814

Ho: B 12 = 0 Ha B 12 tstat = 0.7055275 df = 1974 pvalue = 0.4805653

Ho: B 13 = 0 Ha B 13 tstat = 6.056338 df = 1973 pvalue = 1.663016e-09

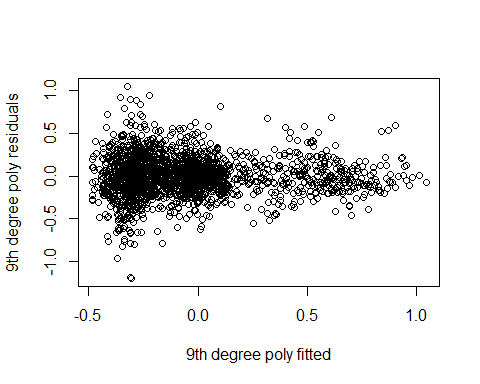
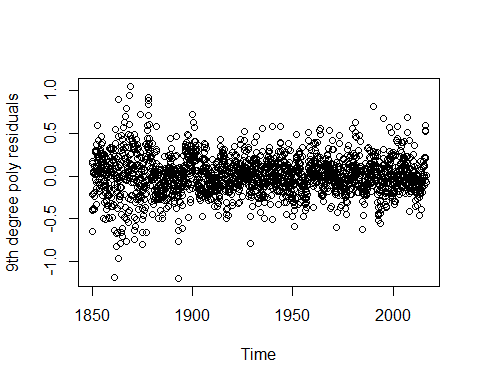
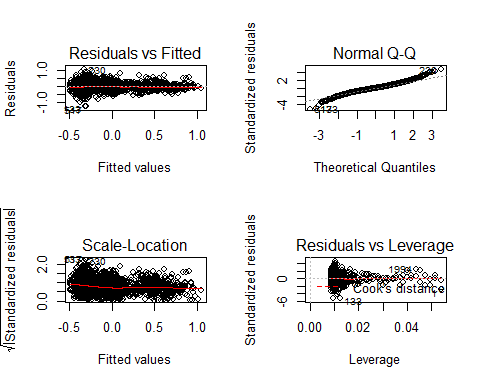
*In the first quarter of the years included in the dataset, or about the first 40 years, after accounting for month, the mean of the median ensembles appears to decrease with time, and then appears to increase through 2016. The relationship has curvature resembling a quadric relationship between time and mean of the median ensembles after accounting for the month effects.*



1. 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.

*The residuals versus fitted plot improved in terms of linearity. To assume that the residuals are linearly related to the response is now reasonable. However, other assumptions are now largely violated. In the residuals vs. fitted plot we see the variation much larger for smaller fitted values than for larger fitted values. While the sample size is large enough to not be too concerned with violations to normality, the residuals are farther from the normal QQ line with the 9th degree polynomial than with the linear model. We very clearly see long tails in the nomral QQ plot with the 9th degree polynomial. Observations 517 and 133 appear to be outliers.*

*Comparing the plots of the residuals from the 9th degree polynomial vs. time to the 9th degree polynomial residuals vs. fitted, we see that there is more variation in the residuals for smaller fitted values. In the residuals vs. time plot, there is more variation in earlier years. This may suggest that it was cooler in earlier years, and may be why the model is more variable in the fitted values for these years.*



1. Run the following code so I can see what version of R you are now using:

### Documenting R version

[1] '3.3.1'

**R Code**

require(xtable)  
require(effects)  
require(car)#Anova  
require(pander)#tables  
require(knitr)  
  
knitr::opts\_chunk$set(echo = FALSE, comment = NA, warning = FALSE, message = FALSE)

setwd("C:/Users/Andrea Mack/Desktop/mack\_hub/course\_work/Time Series/Homework/HW3/hw3\_stat537")  
rawbozemandata <- read.csv("rawbozemandata.csv", header = T)  
  
rawd <- rawbozemandata  
head(rawd)  
rawd$year <- as.numeric(substr(rawd$DATE, 1,4))  
rawd$monthf <- as.factor(month.abb[as.numeric(substr(rawd$DATE, 5,6))])  
rawd$month <- as.numeric(substr(rawd$DATE, 5,6))  
rawd$year.frac <- rawd$year + (rawd$month)/12  
rawd$temp <- rawd$MMXT  
  
lm1 <- lm(temp ~ monthf, data = rawd)  
  
lm1.resid <- lm1$residuals  
  
options(show.sigif.stars = FALSE)

plot(lm1.resid ~ rawd$year.frac, xlab = "Fractional Year", ylab = "Residuals")

plot(MMXT ~ year.frac, xlab = "Fractional Year", ylab ="MMXT", data = rawd )

require(pander)  
lm2 <- lm(lm1.resid ~ rawd$year)  
options(show.signif.stars = FALSE)  
lm2.sum <- summary(lm2)  
  
xtable((lm2.sum))

lm3 <- lm(temp ~ year + monthf, data = rawd)  
lm3.sum <- summary(lm3)  
  
xtable(lm3.sum)

setwd("~/Desktop/mack\_hub/course\_work/Time Series/Homework/HW3/hw3\_stat537")  
hadcrut <- read.table("hadcrut.txt", header = FALSE)  
head(hadcrut)  
  
  
hadcrut$date <- ts(hadcrut[,1], start = c(1850,01), end = c(2016,07), frequency = 12)  
#only use ts if no months/years missing

#col2 are the ensemble medians  
par(mfrow=c(1,1))  
plot(hadcrut$V2 ~ hadcrut$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
  
set.seed(12)  
#plot(shuffle(hadcrut$V2) ~ hadcrut$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians", main = "Shuffled")  
  
  
  
  
par(mfrow=c(2,1))  
plot(hadcrut[1:12,]$V2 ~ hadcrut[1:12,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
plot(hadcrut[13:24,]$V2 ~ hadcrut[13:24,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
par(mfrow = c(2,1))  
plot(hadcrut[25:36,]$V2 ~ hadcrut[25:36,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
plot(hadcrut[37:48,]$V2 ~ hadcrut[37:48,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
plot(hadcrut[49:60,]$V2 ~ hadcrut[49:60,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
plot(hadcrut[61:72,]$V2 ~ hadcrut[61:72,]$date, xlab = "Months Since January 1850", ylab = "Ensemble Medians")  
  
  
#an ensemble data set in which the 100 constituent  
#ensemble members sample the distribution of likely surface temperature anomalies given our  
#current understanding of these uncertainties  
#Sea Surface Temperature anomalies in degrees Celsius, or "SST anomalies" for short, are how much #temperatures depart from what is normal for that time of year.

#%%\_\_we can see there is very little overlap in the confidence intervals for the median ensemble #%%estimate for many pairs of months, indicating time has an affect on the monthly median ensemble #changes. WE SHOULD DISCUSS THIS INTERPRETATION\_\_  
  
hadcrut$monthf <- as.factor(as.numeric(substr(hadcrut$V1,6,7)))  
hadcrut$time.date <- as.vector(time(hadcrut$date))  
lm4 <- lm(V2 ~ time.date + monthf, data = hadcrut)#don't need monthf as explanatory bc month already induced in time??  
  
plot(allEffects(lm4))  
  
lm4.anova <- Anova(lm4, type = "II")  
(xtable(lm4.anova))

par(mfrow = c(1,1))  
plot(lm4,which=1)

lm.d1 <- lm(V2~poly(time.date,1,raw=TRUE) + monthf, data = hadcrut)  
summary(lm.d1)  
test<-lm(V2~monthf,data=hadcrut)  
resids1<-test$residuals  
par(mfrow=c(1,1))  
  
plot(resids1^5~hadcrut$time.date,xlim = c(1850,1900), ylim = c(-0.5,0.5))  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time)), add = TRUE, col = "yellow")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^2, add = TRUE, col = "purple")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^3, add = TRUE, col = "red")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^4, add = TRUE, col = "blue")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^5, add = TRUE, col = "green")  
  
plot.window(xlim = c(1900,1950), ylim = c(-3,3))  
plot(resids1^5~hadcrut$time.date,xlim = c(1900,1950), ylim = c(-0.075,0.075))  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^3, add = TRUE, col = "red")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^4, add = TRUE, col = "blue")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^5, add = TRUE, col = "green")  
  
plot(resids1^5~hadcrut$time.date,xlim = c(1950,2016), ylim = c(-0.5,0.5))  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time)), add = TRUE, col = "yellow")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^2, add = TRUE, col = "purple")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^3, add = TRUE, col = "red")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^4, add = TRUE, col = "blue")  
curve((1/sd(hadcrut$time))\*((x-mean(hadcrut$time))/sd(hadcrut$time))^5, add = TRUE, col = "green")  
  
lm.d <- NULL  
for(i in 1:25){  
lm.d[i] <- lm(V2~poly(time.date,i,raw=TRUE) + monthf, data = hadcrut)  
}  
  
#s4 <- summary(lm.d[[4]])  
#lm.d[5]  
#

other

#another way, I was getting errors in 7 with the for loop  
polyfunc<-function(i) {  
 poly.lm<-lm(V2~poly(time.date,i,raw=TRUE) + monthf, data = hadcrut)  
 return (summary(poly.lm))  
}  
#xtable(polyfunc(4))  
require(knitr)  
kable(polyfunc(5)$coeff)

#orthofunc<-function(i) {  
 #ortho.lm<-lm(V2~poly(time.date,i,raw=FALSE) + monthf,data=hadcrut)  
 #return (summary(ortho.lm))  
#}  
###orthofunc(10) #"large p value"  
  
final.lm<-lm(V2~poly(time.date,9,raw=FALSE) + monthf,data=hadcrut)  
plot(allEffects(final.lm))

par(mfrow=c(2,2))  
plot(final.lm)  
par(mfrow=c(1,1))  
plot(final.lm$residuals~hadcrut$time.date, xlab = "Time", ylab = "9th degree poly residuals")  
  
plot(final.lm$residuals ~ final.lm$fitted.values, xlab = "9th degree poly fitted