Time Series HW 1 Key

Mark Greenwood

## HW 1

* Graded out of 48.5 points but scored out of 60 since some of you are new to working with me and what I look for in these models. There also was a potential 5% bonus. You will see points taken off around the row of the offending part of your response. I usually do not bother to include points possible as I am pretty generous with denominators as happened here but:

Number 1) 5 points

Number 2) 5 ponts

Number 3) 6 points

Number 4) 6 points

Number 5) 9 points

Number 6) 11.5 points

Number 7) 6 points

1. Read in the data set and use R to make a correct date code that separates year and month. There are many ways to do this. If you can't figure out how to do this using functions in R, you can do this outside R (say in Excel) or by some sort of hand coding of the date information but will get a small deduction in points for bypassing the challenge of doing this in an efficient way in R.

rawd1<-read.csv("https://dl.dropboxusercontent.com/u/77307195/rawbozemandata.csv",header=T)  
#View(rawd1)  
head(rawd1)

## STATION STATION\_NAME DATE MMXT  
## 1 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190001 37.6  
## 2 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190002 29.9  
## 3 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190003 47.7  
## 4 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190004 52.7  
## 5 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190005 66.6  
## 6 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190006 79.1

# My first simple idea - generate floor truncated years and then subtract  
rawd1$Year<-(floor(rawd1$DATE/100))  
rawd1$Month<-round((rawd1$DATE/100-rawd1$Year)\*100,1)  
rawd1$Yearfrac<-rawd1$Year+(rawd1$Month-1)/12  
rawd1$Monthf<-factor(rawd1$Month)  
  
head(rawd1)

## STATION STATION\_NAME DATE MMXT Year  
## 1 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190001 37.6 1900  
## 2 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190002 29.9 1900  
## 3 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190003 47.7 1900  
## 4 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190004 52.7 1900  
## 5 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190005 66.6 1900  
## 6 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190006 79.1 1900  
## Month Yearfrac Monthf  
## 1 1 1900.000 1  
## 2 2 1900.083 2  
## 3 3 1900.167 3  
## 4 4 1900.250 4  
## 5 5 1900.333 5  
## 6 6 1900.417 6

#Alternate version using substring:  
rawd2<-read.csv("https://dl.dropboxusercontent.com/u/77307195/rawbozemandata.csv",header=T)  
  
rawd3 <- rawd2[,c("DATE","MMXT")] #Just extracting only interesting columns  
rawd3$Year<-as.numeric(substring(rawd3$DATE,1,4)) #Pick out digits 1 to 4 from the fixed width string of 6 digits  
rawd3$Month<-as.factor(substring(rawd3$DATE,5,6)) #Pick out digits 5 and 6 from the fixed width string of 6 digits   
  
head(rawd3)

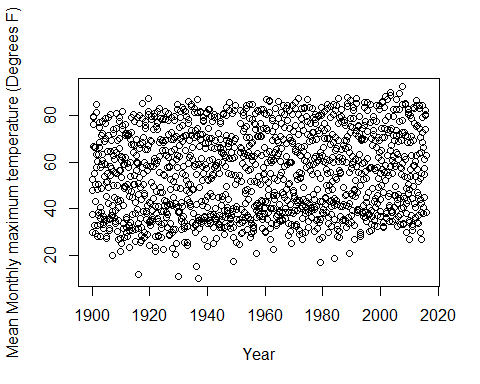
## DATE MMXT Year Month  
## 1 190001 37.6 1900 01  
## 2 190002 29.9 1900 02  
## 3 190003 47.7 1900 03  
## 4 190004 52.7 1900 04  
## 5 190005 66.6 1900 05  
## 6 190006 79.1 1900 06

#Alternate dplyr and zoo version:  
require(dplyr)  
require(zoo)  
rawd4<-tbl\_df(rawd2) %>% mutate(DATE=as.yearmon(as.character(DATE),"%Y%m"))  
  
rawd4<-mutate(rawd4,YEAR=as.numeric(format(DATE,"%Y"))) %>% mutate(MONTH=as.factor(format(DATE,"%m")))  
head(rawd4)

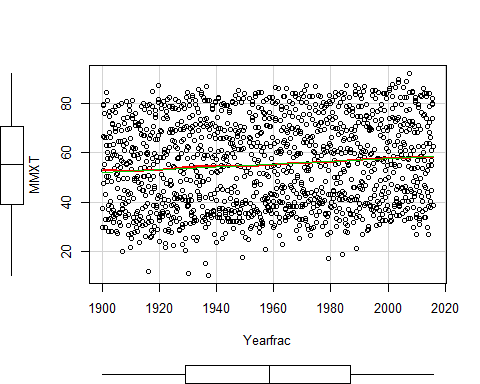
## # A tibble: 6 x 6  
## STATION STATION\_NAME DATE MMXT  
## <fctr> <fctr> <S3: yearmon> <dbl>  
## 1 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US Jan 1900 37.6  
## 2 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US Feb 1900 29.9  
## 3 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US Mar 1900 47.7  
## 4 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US Apr 1900 52.7  
## 5 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US May 1900 66.6  
## 6 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US Jun 1900 79.1  
## # ... with 2 more variables: YEAR <dbl>, MONTH <fctr>

1. Plot the monthly mean maximum temperatures (y-axis) vs year (x-axis), labelling the axes with the name and units of each variable.

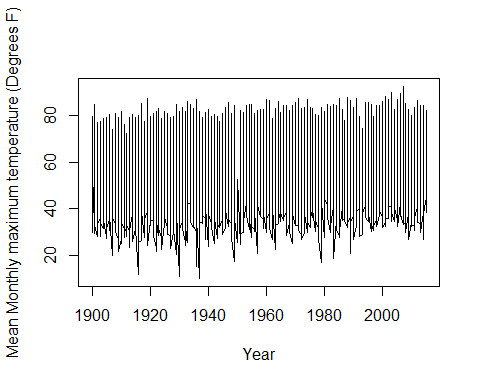
plot(MMXT~Yearfrac,data=rawd1,type="p",xlab="Year", ylab="Mean Monthly maximum temperature (Degrees F)")



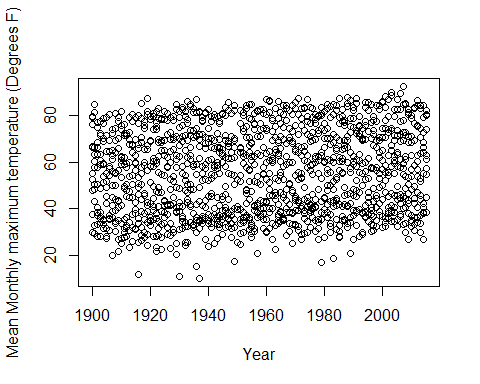
require(car)  
scatterplot(MMXT~Yearfrac,data=rawd1,spread=F) #Adds linear regression line and marginal boxplots



#Or versus year with no fractional information:  
plot(MMXT~Year,data=rawd1,xlab="Year", ylab="Mean Monthly maximum temperature (Degrees F)",type="l") #Better to just plot points if not separating points by time within year in some way...

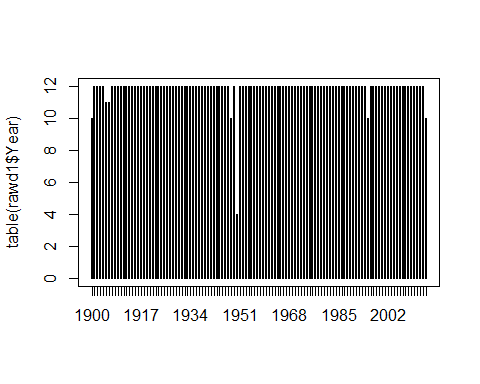


plot(MMXT~Year,data=rawd1,xlab="Year", ylab="Mean Monthly maximum temperature (Degrees F)",type="p")



1. Create a variable that is just the year of each observation and another for the month. Then fit a linear model with temperature as the response and year and month as explanatory variables treated correctly as either quantitative or categorical predictors. Do not consider any higher order model terms such as polynomials or interactions. For many reasons but especially for the following question, do any variable manipulations prior to fitting the model and use the general code format for your lm of: model1<-lm(y~x1+x2,data=mydatasetname).

plot(table(rawd1$Year)) #Not all months had observations recorded - usually because of too few days available in given month to trust monthly mean



options(show.signif.stars=FALSE) #Turns off stars  
model1<-lm(MMXT~Year+Monthf,data=rawd1)  
source("https://dl.dropboxusercontent.com/u/77307195/concise\_lm.R") #Function to clean up lm model summaries  
print(summary(model1), concise = T)

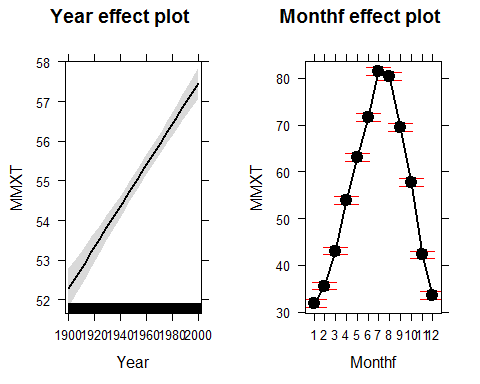
##   
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -69.290162 7.266158 -9.536 < 1e-04  
## Year 0.051674 0.003706 13.942 < 1e-04  
## Monthf2 3.705752 0.604983 6.125 < 1e-04  
## Monthf3 11.197994 0.604983 18.510 < 1e-04  
## Monthf4 21.990235 0.604983 36.349 < 1e-04  
## Monthf5 31.241228 0.606291 51.528 < 1e-04  
## Monthf6 39.729923 0.606291 65.529 < 1e-04  
## Monthf7 49.590800 0.608979 81.433 < 1e-04  
## Monthf8 48.495978 0.607628 79.812 < 1e-04  
## Monthf9 37.663815 0.608966 61.849 < 1e-04  
## Monthf10 25.860881 0.607617 42.561 < 1e-04  
## Monthf11 10.358170 0.607629 17.047 < 1e-04  
## Monthf12 1.780214 0.608968 2.923 0.00352  
##   
## Residual standard error: 4.597 on 1361 degrees of freedom  
## Multiple R-squared: 0.9349, Adjusted R-squared: 0.9343   
## F-statistic: 1628 on 12 and 1361 DF, p-value: < 1e-04

#You can also use summary(model1)

1. Install and load the effects package and run the following code to get effects (also better called termplots) of the model that you fit: plot(allEffects(model1)). Discuss the month term-plot in general.

* The month term-plot suggests that the mean temperature is highest in July and next highest in August with the coldest month of January and December colder than February, on average and after controlling for the linear trend over time.
  + You should not use the 95% CIs for comparing levels but it is reassuring to see that the mean monthly mean maximum temperatures are pretty well estimated. If you want to compare months, you should do some sort of pairwise comparison of all 12 choose 2 levels (like using Tukey's HSD) or maybe contrast winter vs summer if you want some higher level comparisons.

require(effects)  
plot(allEffects(model1))

 4) Install and load the effects package and run the following code to get effects (also better called termplots) of the model that you fit: plot(allEffects(model1)). Discuss the month effect plot in general.

1. For the "year" model component, interpret the estimated slope coefficient and report a 95% confidence interval. Also note the size of the estimated change in the mean temperature over the entire length of the data set and report and confidence interval for that result.

* The year component has an estimated slope of 0.0517 (95% CI from 0.0444 to 0.0589).
* For a 1 year increase in year, we expect the mean of the mean monthly maximum temperatures to increase by 0.0517 degrees F, after controlling for variation in the months (95% CI from 0.0444 to 0.0589).

confint(model1)

## 2.5 % 97.5 %  
## (Intercept) -83.54424523 -55.03607826  
## Year 0.04440294 0.05894485  
## Monthf2 2.51895278 4.89255173  
## Monthf3 10.01119416 12.38479311  
## Monthf4 20.80343554 23.17703449  
## Monthf5 30.05186075 32.43059485  
## Monthf6 38.54055640 40.91929050  
## Monthf7 48.39616142 50.78543847  
## Monthf8 47.30398959 49.68796620  
## Monthf9 36.46920129 38.85842781  
## Monthf10 24.66891461 27.05284779  
## Monthf11 9.16617976 11.55016124  
## Monthf12 0.58559583 2.97483205

* The estimated change in the mean over the 115 years in the data set is 5.94 degrees F (95% CI from 5.1 to 6.8), after controlling for the month to month differences.

#Observations from 1900.0 to 2015 so 115 years from min to max (could use 116 years or 115 if you don't count the first year)  
  
#Raw slope was 0.05167  
  
#Change in mean over range of data set is nearly 6 degrees F  
  
model1$coefficients[2]\*115

## Year   
## 5.942498

#95% CI for total change:  
confint(model1)[2,]\*115

## 2.5 % 97.5 %   
## 5.106339 6.778657

#Or could refit model with year transformed to go from 0 to 1:  
rawd1$YearScale<-(rawd1$Year-min(rawd1$Year))/(max(rawd1$Year)-min(rawd1$Year))  
  
model1\_altscale<-lm(MMXT~YearScale+Monthf,data=rawd1)  
summary(model1\_altscale)

##   
## Call:  
## lm(formula = MMXT ~ YearScale + Monthf, data = rawd1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.5022 -2.9005 0.1112 3.0412 12.6950   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 28.8902 0.4779 60.447 < 2e-16  
## YearScale 5.9425 0.4262 13.942 < 2e-16  
## Monthf2 3.7058 0.6050 6.125 1.18e-09  
## Monthf3 11.1980 0.6050 18.510 < 2e-16  
## Monthf4 21.9902 0.6050 36.349 < 2e-16  
## Monthf5 31.2412 0.6063 51.528 < 2e-16  
## Monthf6 39.7299 0.6063 65.529 < 2e-16  
## Monthf7 49.5908 0.6090 81.433 < 2e-16  
## Monthf8 48.4960 0.6076 79.812 < 2e-16  
## Monthf9 37.6638 0.6090 61.849 < 2e-16  
## Monthf10 25.8609 0.6076 42.561 < 2e-16  
## Monthf11 10.3582 0.6076 17.047 < 2e-16  
## Monthf12 1.7802 0.6090 2.923 0.00352  
##   
## Residual standard error: 4.597 on 1361 degrees of freedom  
## Multiple R-squared: 0.9349, Adjusted R-squared: 0.9343   
## F-statistic: 1628 on 12 and 1361 DF, p-value: < 2.2e-16

confint(model1\_altscale)

## 2.5 % 97.5 %  
## (Intercept) 27.9526621 29.827818  
## YearScale 5.1063386 6.778657  
## Monthf2 2.5189528 4.892552  
## Monthf3 10.0111942 12.384793  
## Monthf4 20.8034355 23.177034  
## Monthf5 30.0518607 32.430595  
## Monthf6 38.5405564 40.919291  
## Monthf7 48.3961614 50.785438  
## Monthf8 47.3039896 49.687966  
## Monthf9 36.4692013 38.858428  
## Monthf10 24.6689146 27.052848  
## Monthf11 9.1661798 11.550161  
## Monthf12 0.5855958 2.974832

1. Generate a test for the month model component, write out the hypotheses, report the results (extract any pertinent numerical results from output), and write a conclusion based on these results.

* This should be some sort of F-test to explore evidence that the twelve monthly means are the same after controlling for the linear trend over time. I like Type II tests as they are the same as Type I (conditional all anything higher up in the table) except that the model is refit to have each component assessed conditional on all terms at the same level. Type II SS are only conditional on things higher up in the ANOVA table.
* Here the result is F(11,1361)=1756.55, p-value<0.0001
* Hypotheses:
  + Null: True mean of the mean monthly max temperatures is the same for all months, controlling for linear trend.
  + Alternative: At least one true mean of the mean monthly max temperatures is different across the months, controlling for linear trend.
* Conclusion: There is strong evidence (F(11,1361)=1756.55, p-value<0.0001) of some difference in the true mean of the monthly maximum temperatures in Bozeman between 1900 and 2015 after controlling for the linear time trend.
* Scope of inference (not graded but should be basic instinct if you completed pre-requisites here at MSU): Since these are not a random sample of sites or years, we can only make inferences to these years in Bozeman. Because there is no random assignment of year or month to responses, we can't make causal inferences, only associations or differences in the responses based on these variables.
* With no causal inferences, try to avoid using "effects" and save that terminology for when random assignment is present (so never in this class).

require(car)  
Anova(model1) #Type II tests that are conditional on all components at the same level in the model, not just ones in the table above it as in Type I tests

## Anova Table (Type II tests)  
##   
## Response: MMXT  
## Sum Sq Df F value Pr(>F)  
## Year 4108 1 194.37 < 2.2e-16  
## Monthf 408393 11 1756.55 < 2.2e-16  
## Residuals 28766 1361

1. Run the following code:

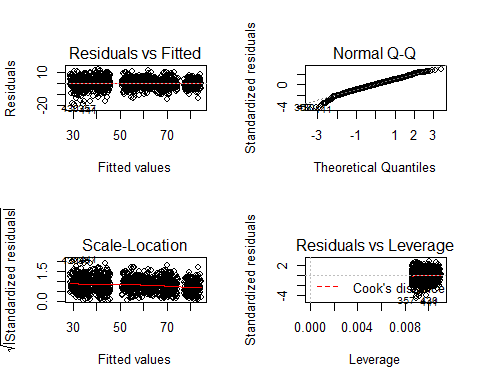
par(mfrow=c(2,2))

plot(model1)

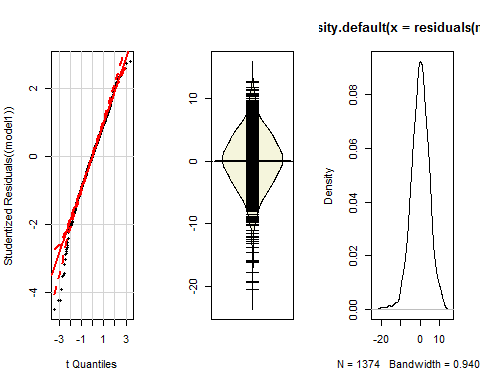
It should produce four panels with residuals vs fitted, normall QQ, scale-location, and residuals vs leverage plots. Only discuss the normall QQ plot. What model assumptions does this help us assess and what does it suggest here?

* QQ-plot only assesses normality of residuals, so we are interested in detecting outliers and shape issues that deviate from normality using this plot. Note that some plots are added below to help re-inforce my conclusion:
* There is a small deviation in the left tail that suggests slightly more variability than expected from a normal distribution and so this may suggest a slight left skew in the residuals from this model. We might want to explore this further but there is not a huge concern here in this plot.

par(mfrow=c(2,2))  
plot(model1)



par(mfrow=c(1,3))  
#Or using car to obtain studentized residuals  
qqPlot((model1),pch=16)  
require(beanplot)  
beanplot(residuals(model1),method="jitter", col="beige") # A little bit of left skew  
plot(density(residuals(model1)))

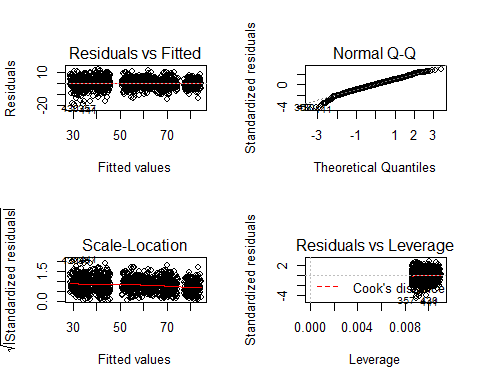


You might also be curious about the difference in using the truncated year (1900 for all observations in 1900) as opposed to the fractional year. It ends up that it is very little except for minor impacts on the within year predictions each year:

model2<-lm(MMXT~Yearfrac+Monthf,data=rawd1)  
print(summary(model2), concise = T)

##   
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -69.290162 7.266158 -9.536 <1e-04  
## Yearfrac 0.051674 0.003706 13.942 <1e-04  
## Monthf2 3.701446 0.604984 6.118 <1e-04  
## Monthf3 11.189381 0.604985 18.495 <1e-04  
## Monthf4 21.977317 0.604986 36.327 <1e-04  
## Monthf5 31.224003 0.606297 51.500 <1e-04  
## Monthf6 39.708393 0.606299 65.493 <1e-04  
## Monthf7 49.564963 0.608994 81.388 <1e-04  
## Monthf8 48.465835 0.607645 79.760 <1e-04  
## Monthf9 37.629365 0.608976 61.791 <1e-04  
## Monthf10 25.822126 0.607627 42.497 <1e-04  
## Monthf11 10.315109 0.607657 16.975 <1e-04  
## Monthf12 1.732846 0.608989 2.845 0.0045  
##   
## Residual standard error: 4.597 on 1361 degrees of freedom  
## Multiple R-squared: 0.9349, Adjusted R-squared: 0.9343   
## F-statistic: 1628 on 12 and 1361 DF, p-value: < 1e-04

par(mfrow=c(2,2))  
plot(model2)



Anova(model2)

## Anova Table (Type II tests)  
##   
## Response: MMXT  
## Sum Sq Df F value Pr(>F)  
## Yearfrac 4108 1 194.37 < 2.2e-16  
## Monthf 408223 11 1755.82 < 2.2e-16  
## Residuals 28766 1361