Time Series HW 1

Andrea Mack and Kenny Flagg

September 2, 2016

HW 1

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.*

```
rawbozemandata <- read.csv('rawbozemandata.csv', header = TRUE)

# A date without a day is NA, so make all these dates the first of the month.
rawbozemandata$date2 <- as.Date(paste0(rawbozemandata$DATE, '01'), '%Y%m%d')

# Fill the rest of the page because the plot for #2 doesn't fit here.
head(rawbozemandata, n = 24)
```

```
       STATION                          STATION_NAME   DATE MMXT      date2
1  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190001 37.6 1900-01-01
2  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190002 29.9 1900-02-01
3  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190003 47.7 1900-03-01
4  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190004 52.7 1900-04-01
5  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190005 66.6 1900-05-01
6  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190006 79.1 1900-06-01
7  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190007 79.6 1900-07-01
8  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190008 76.0 1900-08-01
9  COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190009 66.3 1900-09-01
10 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190010 55.2 1900-10-01
11 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190101 32.9 1901-01-01
12 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190102 29.6 1901-02-01
13 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190103 43.1 1901-03-01
14 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190104 50.5 1901-04-01
15 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190105 66.1 1901-05-01
16 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190106 63.1 1901-06-01
17 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190107 84.5 1901-07-01
18 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190108 81.3 1901-08-01
19 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190109 57.6 1901-09-01
20 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190110 60.3 1901-10-01
21 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190111 48.4 1901-11-01
22 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190112 33.0 1901-12-01
23 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190201 28.2 1902-01-01
24 COOP:241044 BOZEMAN MONTANA STATE UNIVERSITY MT US 190202 37.2 1902-02-01
```

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

(See page 7 for code.)

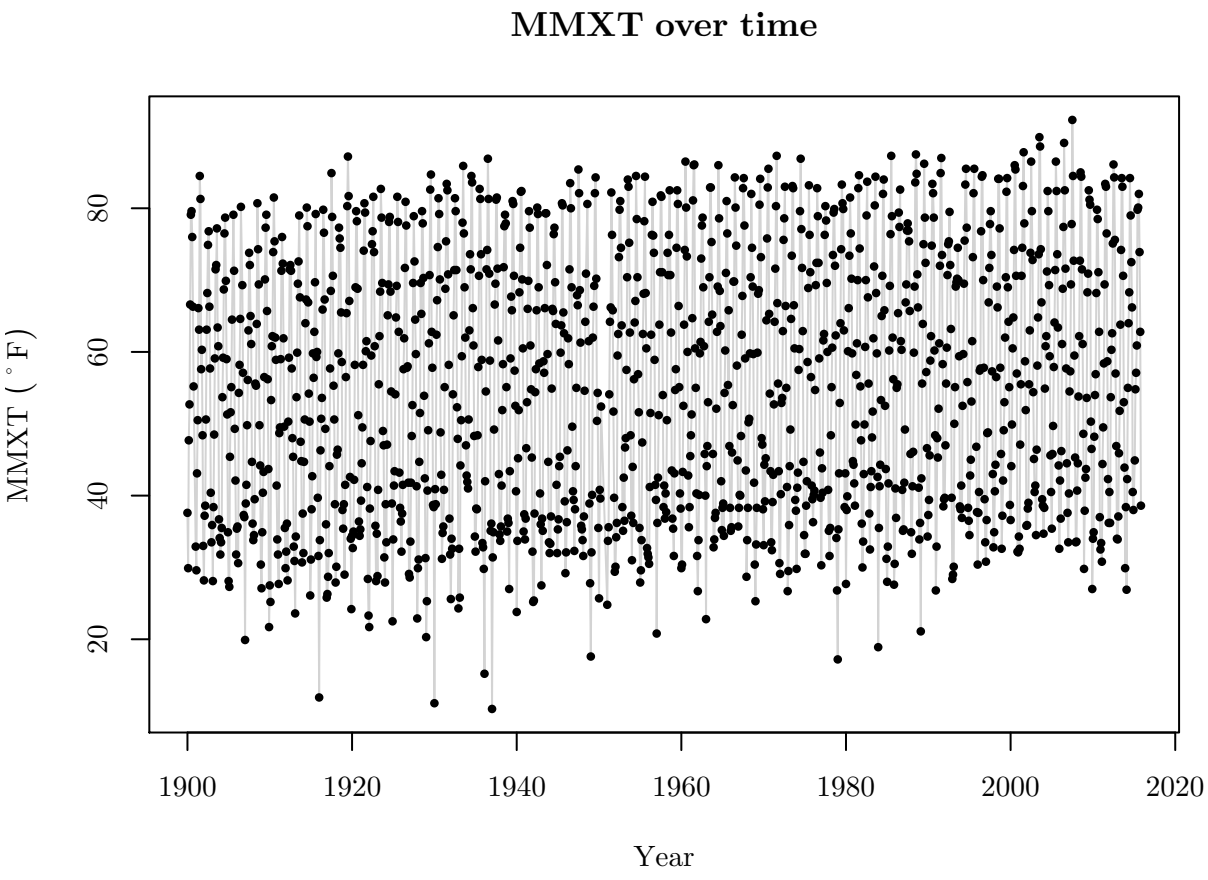

Figure 1: Monthly mean maximum temperatures (MMXT) at the MSU weather station plotted over time. The faint grey line connects the monthly points to illustrate the periodic annual trend.

3. *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)`.

   Time is naturally quantitative and it makes sense to treat year as quantitative to capture long-term trends in MMXT (Figure 1 shows an increasing linear trend). The relationship between MMXT and month is not linear or simple—it is periodic—so I treat month as categorical.

   (See page 7 for code.)

```
Call:
lm(formula = MMXT ~ year + month, data = rawbozemandata)

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)    -69.290162   7.266158  -9.536  < 2e-16
year             0.051674   0.003706  13.942  < 2e-16
monthFebruary    3.705752   0.604983   6.125 1.18e-09
monthMarch      11.197994   0.604983  18.510  < 2e-16
monthApril      21.990235   0.604983  36.349  < 2e-16
monthMay        31.241228   0.606291  51.528  < 2e-16
monthJune       39.729923   0.606291  65.529  < 2e-16
monthJuly       49.590800   0.608979  81.433  < 2e-16
monthAugust     48.495978   0.607628  79.812  < 2e-16
monthSeptember  37.663815   0.608966  61.849  < 2e-16
monthOctober    25.860881   0.607617  42.561  < 2e-16
monthNovember   10.358170   0.607629  17.047  < 2e-16
monthDecember    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: < 2.2e-16
```

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.*

   As seen in Figure 2, mean MMXT is higher in the summer months than in the winter months. Months that are close togther in the year have mean MMXTs that are more similar than the MMXT values of months that are far apart. For example, the difference in mean MMXT between July and January (six months apart) is larger than the difference in mean MMXT between July and August. Additionally, December and January have relatively similar mean MMXT values so that the monthly trend cycles from one year to then next. None of this is surprising given what I learned about the seasons in elementary school.

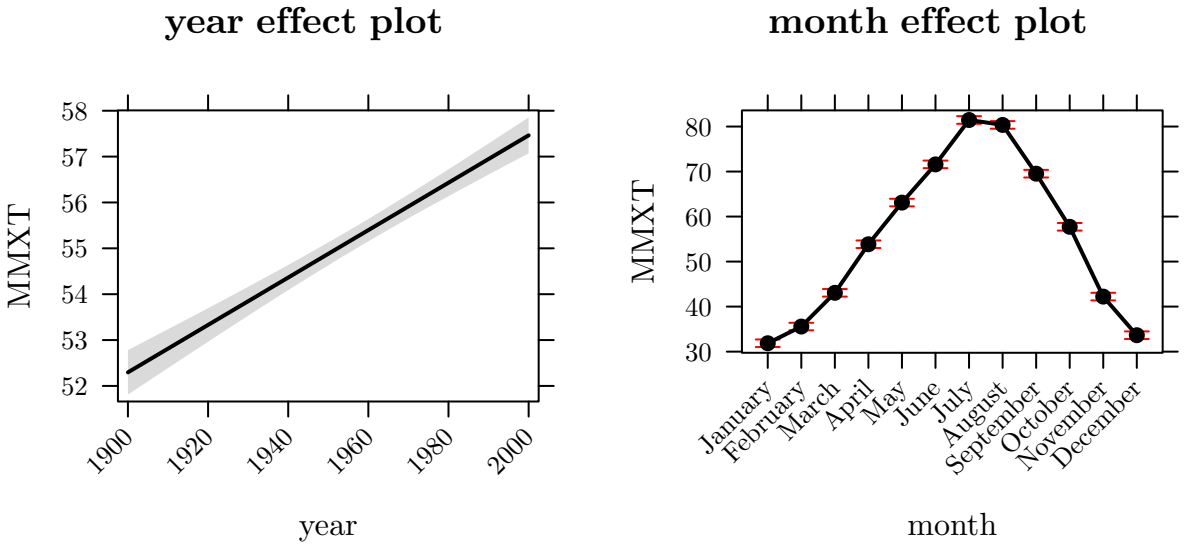

Figure 2: Plots showing the estimated effect of year on MMXT (averaged over months) and the estimated effect of month on MMXT (averaged over years).

5. *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 mean MMXT increases by an estimated 0.0517 °F each year. We are 95% confident that the true yearly increase is between 0.0444 °F and 0.0589 °F. Over the 115 years for which we have data, this amounts to an expected 5.94 °F increase, with a 95% confidence interval of 5.11 °F to 6.78 °F.

   (See page 7 for the code that generated that paragraph!)

6. *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.*

```
Anova Table (Type III tests)

Response: MMXT
            Sum Sq   Df  F value    Pr(>F)
(Intercept)   1922    1   90.936 < 2.2e-16
year          4108    1  194.370 < 2.2e-16
month       408393   11 1756.549 < 2.2e-16
Residuals    28766 1361
```

H_0: all month coefficients $= 0$

H_a: some month coefficient $\neq 0$

With $F_{11,\,1361} = 1756.55$ (p-value < 0.0001) there is very strong evidence that, within a year, the true mean MMXT differs by month.

(See page 8 for code.)

7. *Run the following code:*

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

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

The normal Q-Q plot helps us assess whether the residuals can be approximated by a normal distribution. It shows the standardized residuals plotted against the standard normal quantiles, so if the residuals follow a normal distribution the plot would show a linear relationship. Most of the points are along the line, but the downward elbow in the lower left indicates that the distribution of residuals has a long left tail compared to a normal distribution. From this plot, it appears that the residual distribution is approximately normal but that there are some very cold observations that are poorly described by the model.

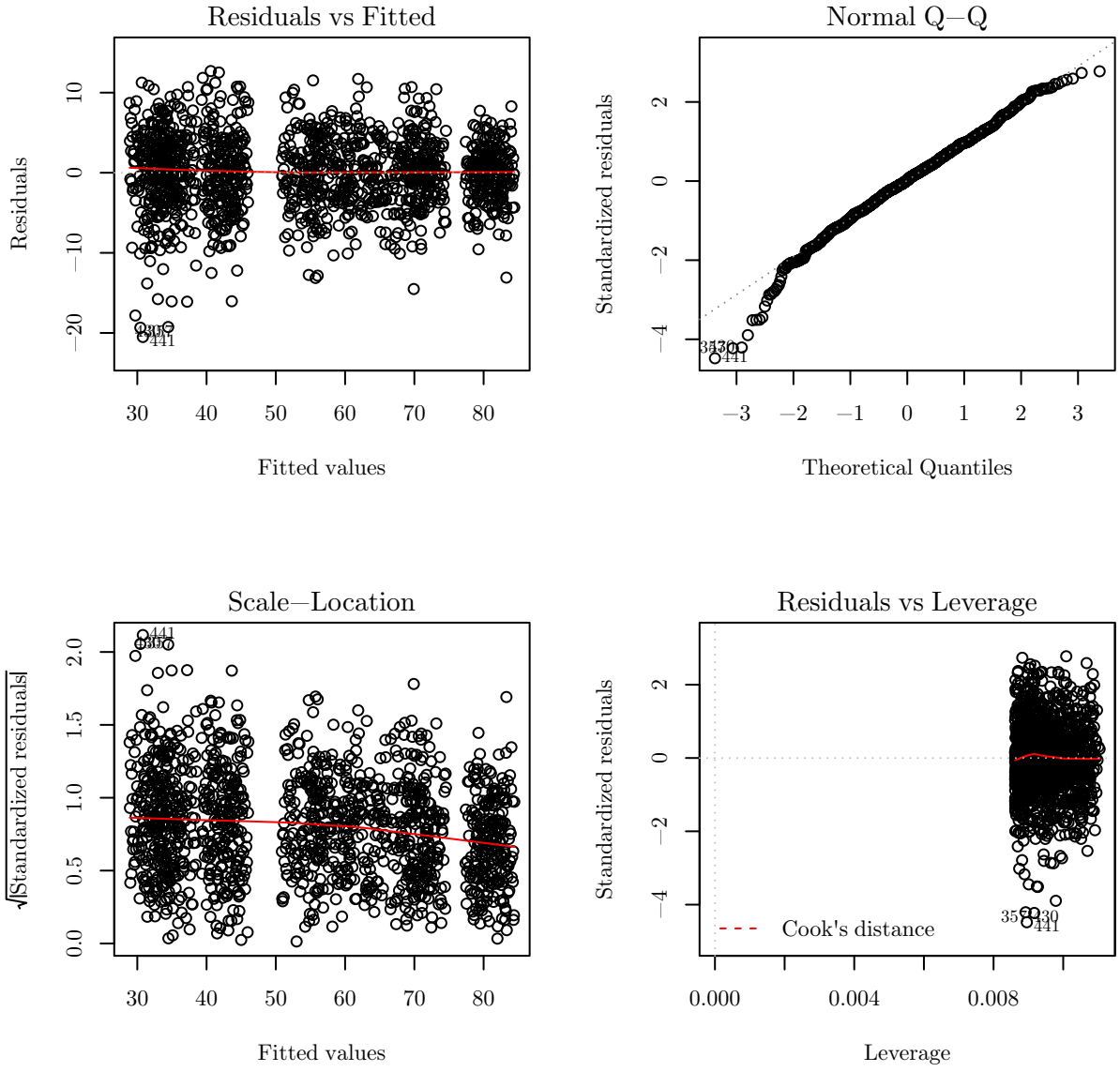

Figure 3: The four standard linear model diagnostic plots.

R Code

1.
```
rawbozemandata <- read.csv('rawbozemandata.csv', header = TRUE)

# A date without a day is NA, so make all these dates the first of the month.
rawbozemandata$date2 <- as.Date(paste0(rawbozemandata$DATE, '01'), '%Y%m%d')

# Fill the rest of the page because the plot for #2 doesn't fit here.
head(rawbozemandata, n = 24)
```

2.
```
# Start with a light grey line to connect points through years.
plot(MMXT ~ date2, data = rawbozemandata, type = 'l', col = 'lightgrey',
     main = 'MMXT over time', xlab = 'Year', ylab = expression(MMXT~(degree*F)))

# Now points for the monthly observations.
points(MMXT ~ date2, data = rawbozemandata, pch = 19, cex = 0.5, col = 'black')
```

3.
```
# Get the year and month.
rawbozemandata$year <- as.numeric(format(rawbozemandata$date2, '%Y'))
rawbozemandata$month <- format(rawbozemandata$date2, '%B')

# Make month into a factor with levels ordered by first appearance.
rawbozemandata$month <- factor(rawbozemandata$month,
                               levels = unique(rawbozemandata$month))

model1 <- lm(MMXT ~ year + month, data = rawbozemandata)
summary(model1)
```

4.
```
require(effects)
plot(allEffects(model1), rug = FALSE, cex = 0.75, rotx = 45)
```

5.
```
slope <- coef(model1)['year']
se <- summary(model1)$coefficients['year', 'Std. Error']
confintyear <- slope + qt(c(0.025, 0.975), model1$df) * se

nyears <- range(rawbozemandata$year) %*% c(-1, 1) # max(year) - min(year)
confintrange <- nyears * slope + qt(c(0.025, 0.975), model1$df) * nyears * se

cat('The mean MMXT increases by an estimated', signif(slope, 3),
    '\\(^\\circ\\)F each year. We are 95\\% confident that the true yearly',
    'increase is between', signif(confintyear[1], 3), '\\(^\\circ\\)F and',
    signif(confintyear[2], 3), '\\(^\\circ\\)F. Over the', nyears,
    'years for which we have data, this amounts to an expected',
    signif(nyears * slope, 3), '\\(^\\circ\\)F increase, with a 95\\%',
    'confidence interval of', signif(confintrange[1], 3), '\\(^\\circ\\)F to',
    signif(confintrange[2], 3), '\\(^\\circ\\)F.\n\n')
```

6.
```
anova1 <- Anova(model1, type = 3)
print(anova1)
```

```
cat('With \\(F_{', anova1['month','Df'], ',\\: ', anova1['Residuals','Df'],
    '} = ', signif(anova1['month','F value', 6]), '\\) (\\(\\text{p-value} ',
    ifelse(anova1['month','Pr(>F)'] < 0.0001, '< 0.0001',
           sprintf('= %.4f', anova1['month','Pr(>F)'])),
    '\\)) there is very strong evidence that, within a year,
the true mean MMXT differs by month.')
```

7.
```
par(mfrow=c(2,2))
plot(model1)
```