Time Series HW 1

Andrea Mack and Kenny Flagg

September 2, 2016

HW 1

Disclaimer: We originally did the assignment separately, then combined the best parts of our answers. As a result, Kenny's code is mostly used. If you would like Andrea to provide her separately, she will.

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)

head(rawbozemandata)

      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

dim(rawbozemandata)

[1] 1374    4

# Make a new data frame for tinkering.
rawt <- rawbozemandata

# The date is stored as YYYYMM. To get the year and month, we treat it as a character
# string, get characters 1-4 for the year and characters 5-6 as the month.
rawt$year <- as.numeric(substr(as.character(rawt$DATE), 1, 4))
rawt$month <- as.numeric(substr(as.character(rawt$DATE), 5, 6))
```

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 8 for code.)

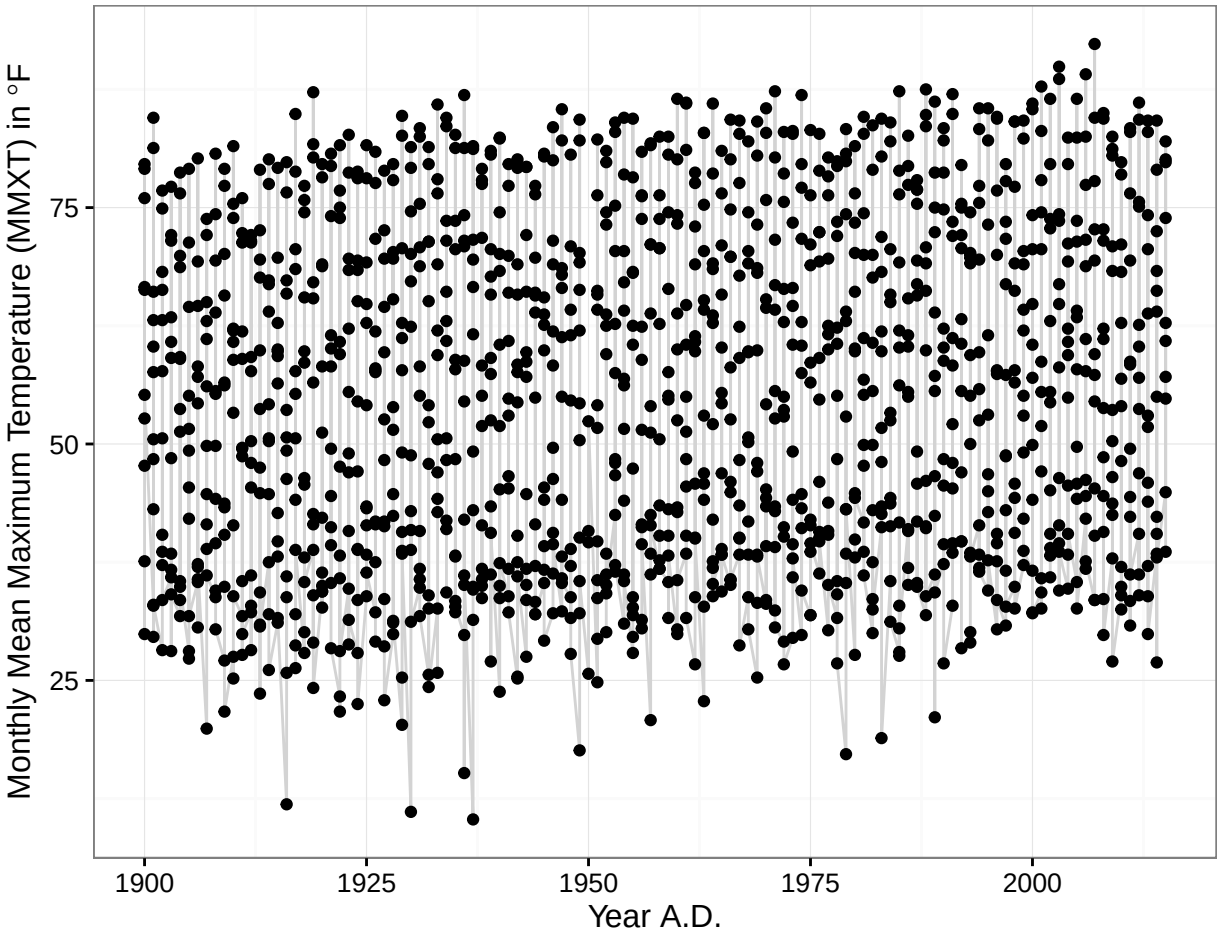

Figure 1: Monthly mean maximum temperatures (MMXT) at the MSU weather station plotted over time. The lines connect the monthly points to illustrate the periodic annual trend.

Below is plotted the yearly mean maximum temperature by year. This plot is interesting because it more visibly shows and increasing linear trend in yearly mean temperatures over time. Notice the low average temperature in 1950. In 1950, only months January, Februrary, March, and April had observations, making the average MMXT for that year quite low.

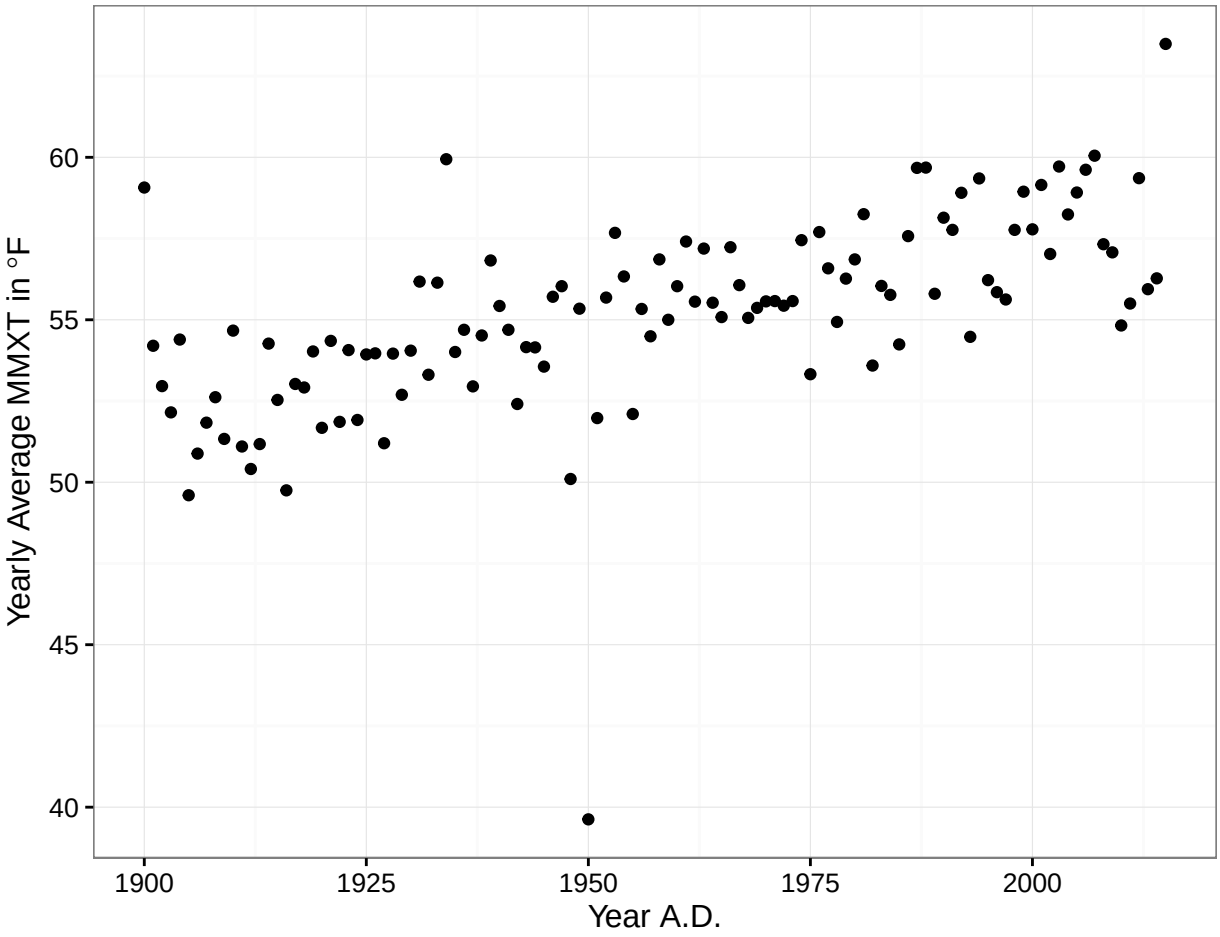

Figure 2: Yearly average MMXT at the MSU weather station plotted over time.

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 is best to treat year as continuous here because there are many years and it would take up a lot of degrees of freedom to fit a model with a different parameter for each year. Treating year as continuous also allows us to capture long-term linear (in this case) trends in average MMXT (Figure 1 shows an increasing linear trend). The relationship between average MMXT and month is not linear or simple, it is periodic, so month is treated as categorical.

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

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

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

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
month2        3.705752   0.604983   6.125 1.18e-09
month3       11.197994   0.604983  18.510  < 2e-16
month4       21.990235   0.604983  36.349  < 2e-16
month5       31.241228   0.606291  51.528  < 2e-16
month6       39.729923   0.606291  65.529  < 2e-16
month7       49.590800   0.608979  81.433  < 2e-16
month8       48.495978   0.607628  79.812  < 2e-16
month9       37.663815   0.608966  61.849  < 2e-16
month10      25.860881   0.607617  42.561  < 2e-16
month11      10.358170   0.607629  17.047  < 2e-16
month12       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 3, mean MMXT is higher in the summer months and lower in the winter months, with fall and spring months having mild temperatures. The yearly cyclic pattern seen in the plot seems consistent with what we learned in elementary school about the seasons in Montana.

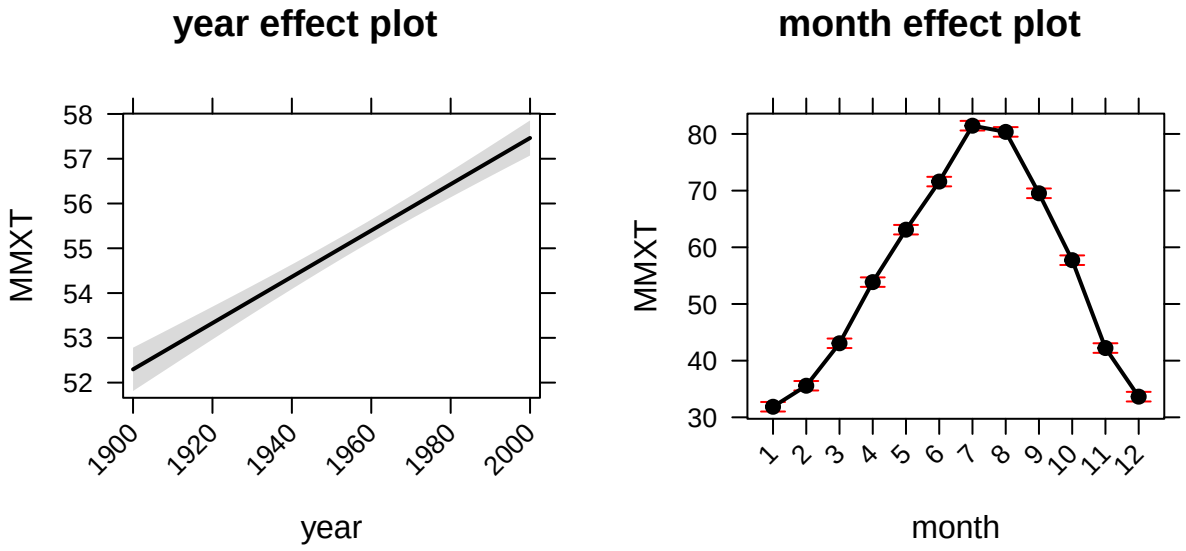

Figure 3: Plots showing the estimated effect of year on MMXT (averaged over months) and the estimated effect of month on MMXT (averaged over years).

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 is expected to increase by an estimated 5.17 °F every 100 years. We are 95% confident that the true mean increase every 100 years is between 4.44 °F and 5.89 °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 8 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.*

   We used both type I and type III sums of squares, and since they were not equal that means at least one month within a year doesn't have an observation, and the data are unbalanced. Therefore, we went with the type III sums of squares.

```
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
```

$$H_0\text{: all month coefficients} = 0$$
$$H_a\text{: at least one month coefficient} \neq 0$$

   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 9 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 shows the standardized residuals plotted against the standard normal quantiles, so if the residuals follow a normal distribution the plot will show a linear relationship. The normal Q-Q plot helps us assess whether it is reasonable to assume the errors are normally distributed, which we need to make inference, not to ensure the parameter estimates are BLUE. Though there are some major deviations from normality in the right tail, the sample size is large enough that it is reasonable to use the normal distribution.

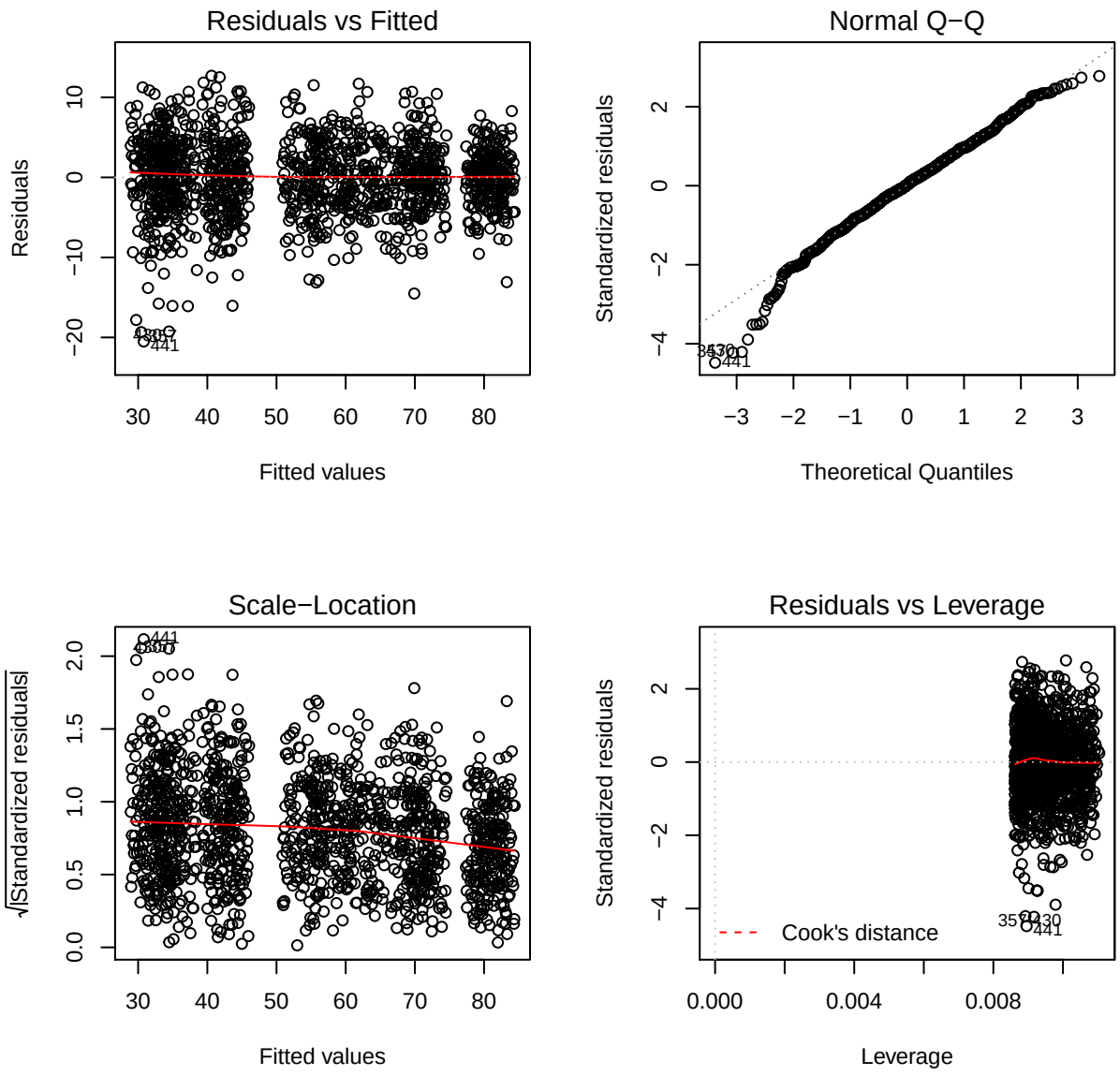

Figure 4: The four standard linear model diagnostic plots.

R Code

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

head(rawbozemandata)
dim(rawbozemandata)

# Make a new data frame for tinkering.
rawt <- rawbozemandata

# The date is stored as YYYYMM. To get the year and month, we treat it as a character
# string, get characters 1-4 for the year and characters 5-6 as the month.
rawt$year <- as.numeric(substr(as.character(rawt$DATE), 1, 4))
rawt$month <- as.numeric(substr(as.character(rawt$DATE), 5, 6))
```

2.
```
ggplot(data = rawt, aes(x=year, y=MMXT)) + geom_line(col = "lightgrey") + geom_point() +
  labs(x = "Year A.D.", y = expression(paste("Monthly Mean Maximum Temperature (MMXT) in ", degree, "F")))


# Get the average for each year.
yearly <- data.frame(meant = tapply(rawt$MMXT, rawt$year, mean), year = unique(rawt$year))

ggplot(data = yearly, aes(x=year, y=meant)) + geom_point() +
  labs(x = "Year A.D.", y = expression(paste("Yearly Average MMXT in ", degree, "F")))
```

3.
```
# Make month into a factor with levels ordered by first appearance.
rawt$month <- factor(rawt$month, levels = unique(rawt$month))

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

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

5.
```
# Make a CI for the slope.
slope <- coef(model1)["year"]
se <- summary(model1)$coefficients["year", "Std. Error"]
confintyear <- slope + qt(c(0.025, 0.975), model1$df) * se

# Get the number of years then make a CI for that many years by multiplying.
nyears <- range(rawt$year) %*% c(-1, 1) # max(year) - min(year)
confintrange <- nyears * slope + qt(c(0.025, 0.975), model1$df) * nyears * se

cat("The mean MMXT is expected to increase by an estimated", signif(slope, 3)*100,
    "$^\\circ$F every 100 years. We are 95\\% confident that the true mean",
    "increase every 100 years is between", signif(confintyear[1], 3)*100, "$^\\circ$F and",
    signif(confintyear[2], 3)*100, "$^\\circ$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)


# Get the pertinent stuff out of the anova object and write a conclusion.
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)
```