Week 3 homework

```r
#Loading packages
library('GGally')
library('stargazer')
library('tidyverse')
library('usdm')
```

```r
#Loading data

#temperature data file
tempData <- read.table("https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/592f3be3e90d2bdfe6a69

#Crime data file
crimeData <- read.table("http://www.statsci.org/data/general/uscrime.txt", header = TRUE)

#Set seed for reproducibility
set.seed(156)

# stargazer(tempData)
# stargazer(crimeData)
```

Question 1

Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of a (the first smoothing parameter) to be closer to 0 or 1, and why?

Answer

A situation where exponential smoothing might be used is to predict the value of a stock in the stock market. This works because it is time series data. The data that would be needed is a date and a stock price for each date. I would expect the value of a to be closer to 1 as it has less smoothing and gives more value to recent data. This is important as I believe a stocks price is tied to the most recent history.

Question 2

Using the 20 years of daily high temperature data for Atlanta (July through October) build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years.

Table 1: Summary of Temperature Data

Statistic	N	Mean	St. Dev.	Min	Max
M	47	13.857	1.257	11.900	17.700
So	47	0.340	0.479	0	1
Ed	47	10.564	1.119	8.700	12.200
Po1	47	8.500	2.972	4.500	16.600
Po2	47	8.023	2.796	4.100	15.700
LF	47	0.561	0.040	0.480	0.641
M.F	47	98.302	2.947	93.400	107.100
Pop	47	36.617	38.071	3	168
NW	47	10.113	10.283	0.200	42.300
U1	47	0.095	0.018	0.070	0.142
U2	47	3.398	0.845	2.000	5.800
Wealth	47	5,253.830	964.909	2,880	6,890
Ineq	47	19.400	3.990	12.600	27.600
Prob	47	0.047	0.023	0.007	0.120
Time	47	26.598	7.087	12.200	44.000
Crime	47	905.085	386.763	342	1,993

Summary of Temperature Data

Question 3

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

Answer

A situation that a linear regression model would be useful is to predict the value of a house. Good predictors for this would be square_footage, NumberofRooms, NumberofBathrooms, Plotsize and garagesize.

Question 4

Predict the observed crime rate in a city. Show your model (factors used and their coefficients), the software output, and the quality of fit.

Table 2: Summary of Crime Data

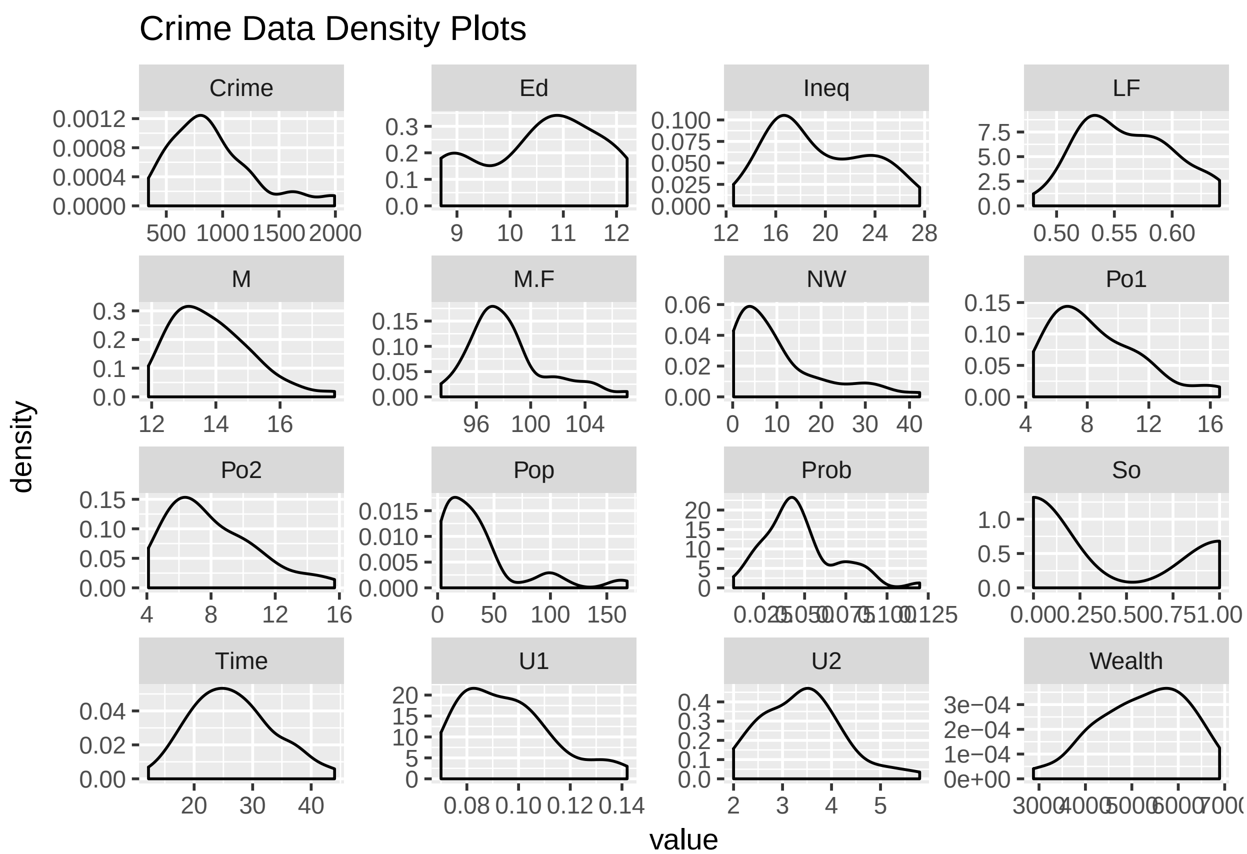
Statistic	N	Mean	St. Dev.	Min	Max
X1996	123	83.715	8.548	60	99
X1997	123	81.675	9.319	55	95
X1998	123	84.260	6.409	63	95
X1999	123	83.358	9.723	57	99
X2000	123	84.033	9.519	55	101
X2001	123	81.553	8.225	51	93
X2002	123	83.585	9.426	57	97
X2003	123	81.480	7.018	57	91
X2004	123	81.764	6.663	62	95
X2005	123	83.358	7.733	54	94
X2006	123	83.049	9.794	53	98
X2007	123	85.398	9.033	59	104
X2008	123	82.512	8.733	50	95
X2009	123	80.992	9.013	51	95
X2010	123	87.211	7.445	67	97
X2011	123	85.276	9.931	59	99
X2012	123	84.650	9.252	56	105
X2013	123	81.667	7.727	56	92
X2014	123	83.943	6.591	63	95
X2015	123	83.301	8.709	56	97

Summary of Crime Data

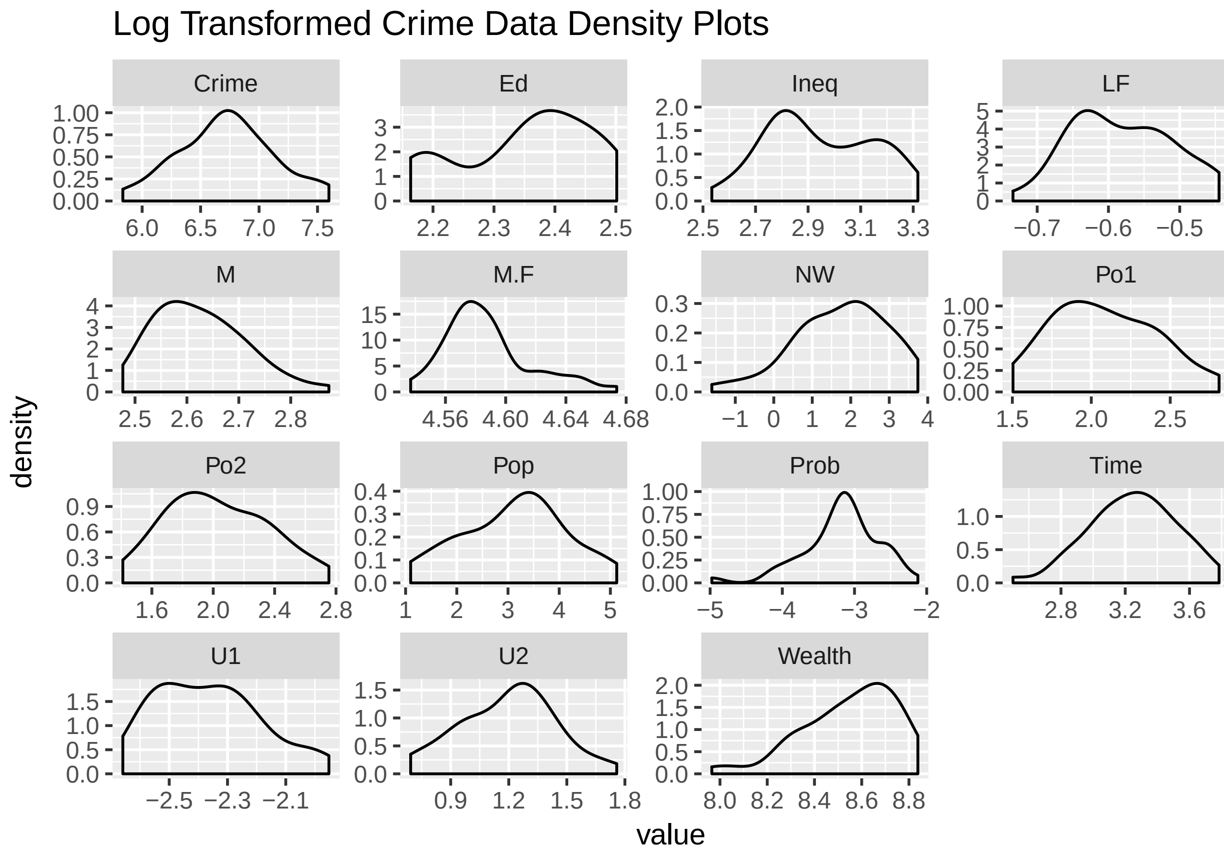
Crime Data Exploration and Transformation

linear models work best with gaussian distributions. Many of the predictors are skewed so I log transformed them to better fit a gaussian distribution.

```
crimeData %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density() +
    labs(title = "Crime Data Density Plots")
```




```r
#log transformation to better fit a gaussian distrubution - did not transform column 'SO' as it is logi
log(crimeData[,c(1,3:16)]) %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density() +
    labs(title = "Log Transformed Crime Data Density Plots")
```




```
#Building New Log dataset
logCrimeData <- log(crimeData[,c(1,3:4,6:16)])
logCrimeData$So <- crimeData$So

#testing for collinear variables - this shows me that po1 and po2 are collinear and there are no need f
#vif(crimeData)
```

Test for collinearity

This shows me that po1 and po2 are collinear and there are no need for both in the model. A VIF greater than 10 is a signal that the model has a collinearity problem. Because Wealth and Ineq are near 10 and I believe they are important factors in predicting a crime rate I am leaving them in.

% VIF test M 3.411365
So 5.342925
Ed 6.967803
Po1 118.641813
Po2 117.546092
LF 3.743340
M.F 3.897988
Pop 2.569876
NW 4.753696
U1 6.533978 U2 5.944206
Wealth 10.897084
Ineq 12.030316

Prob 3.328492
Time 2.739674

Crime Data Linear Model

Building linear models for both log and normal datasets to compare - chose log-transformed model for best results

```
crimeModelLog <- lm(Crime ~ ., data = logCrimeData)
crimeModel <- lm(Crime ~ ., data = crimeData)

# Obtain predicted and residual values
logCrimeData$predicted <- predict(crimeModelLog)
logCrimeData$residuals <- residuals(crimeModelLog)

crimeData$predicted <- predict(crimeModel)
crimeData$residuals <- residuals(crimeModel)

#Creating residual df for plotitng
modelResiduals <- data.frame(data=(cbind(residuals(crimeModel),residuals(crimeModelLog))))
colnames(modelResiduals) <- c('Normal', 'Log')

#summary to see accuracy
summary(crimeModelLog)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = logCrimeData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37220 -0.12001  0.00339  0.10944  0.36213
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.05111    8.53640  -0.475 0.638317
## M            1.56933    0.48985   3.204 0.003067 **
## Ed           2.15094    0.54736   3.930 0.000427 ***
## Po1          0.77794    0.19543   3.981 0.000370 ***
## LF           0.61067    0.67478   0.905 0.372230
## M.F         -2.38907    1.82213  -1.311 0.199143
## Pop         -0.07614    0.04817  -1.581 0.123813
## NW           0.10791    0.04445   2.428 0.021002 *
## U1          -0.12685    0.31390  -0.404 0.688826
## U2           0.44557    0.22457   1.984 0.055883 .
## Wealth       0.66766    0.39219   1.702 0.098374 .
## Ineq         1.59115    0.35365   4.499 8.46e-05 ***
## Prob        -0.30382    0.09638  -3.152 0.003506 **
## Time        -0.26841    0.16755  -1.602 0.118997
## So           0.06321    0.13263   0.477 0.636896
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

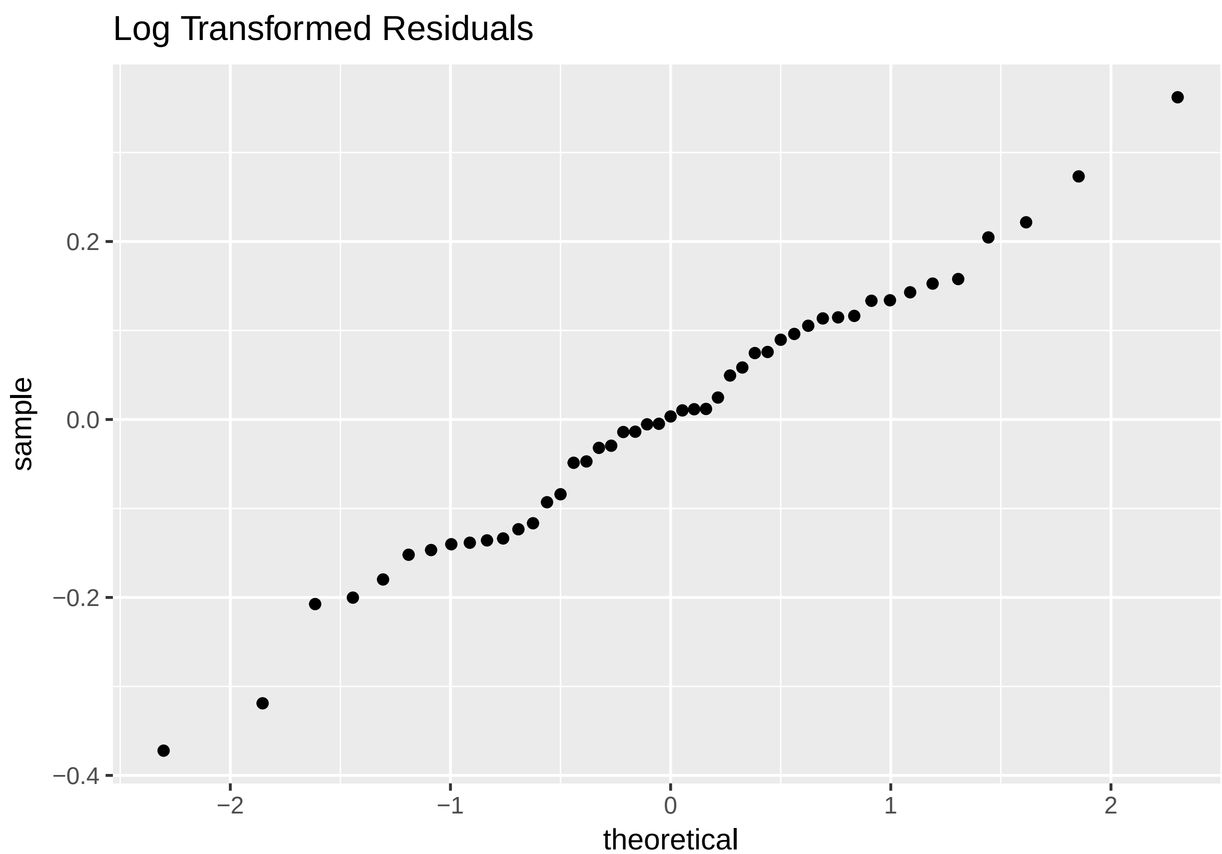
```
## Residual standard error: 0.178 on 32 degrees of freedom
## Multiple R-squared:  0.8695, Adjusted R-squared:  0.8124
## F-statistic: 15.23 on 14 and 32 DF,  p-value: 2.404e-10
```

```
#coefficients and formula
crimeModelLog
```

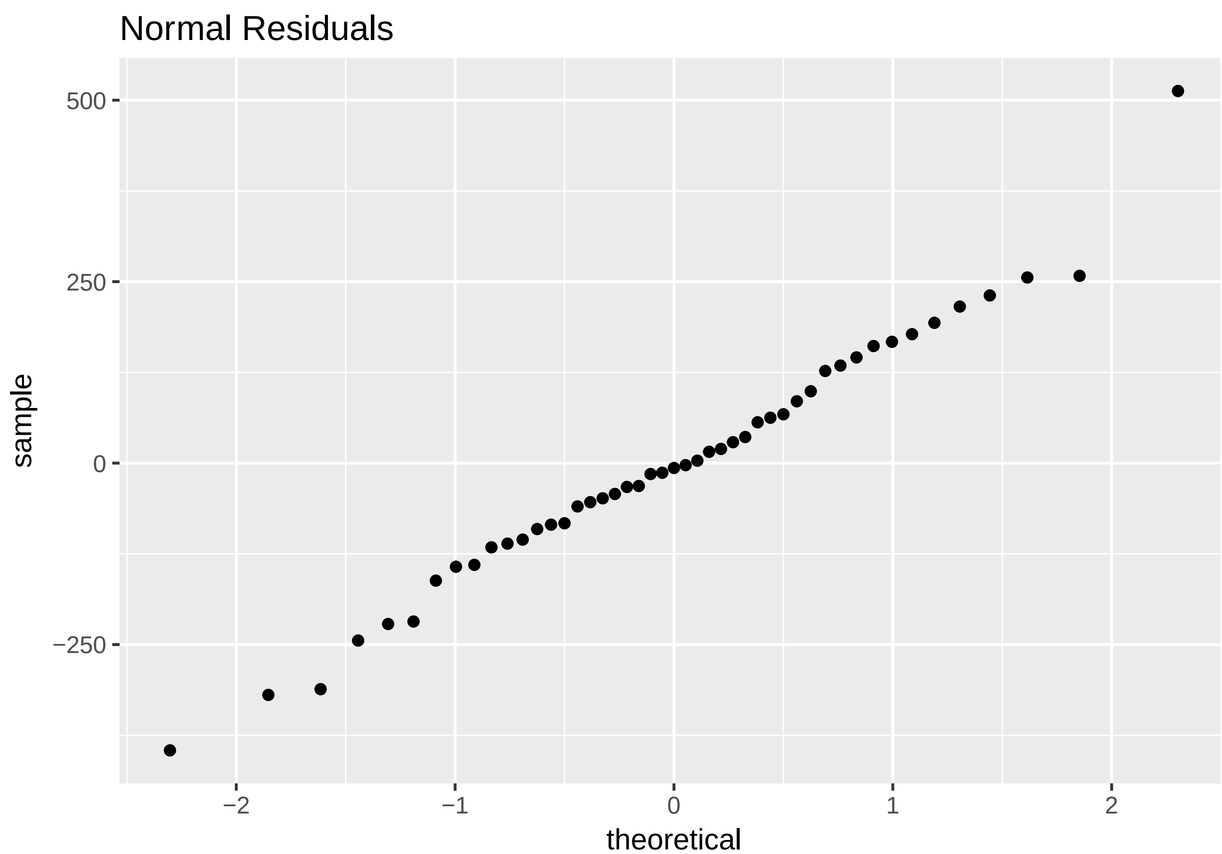
```
##
## Call:
## lm(formula = Crime ~ ., data = logCrimeData)
##
## Coefficients:
## (Intercept)            M           Ed          Po1           LF
##    -4.05111      1.56933      2.15094      0.77794      0.61067
##         M.F          Pop           NW           U1           U2
##    -2.38907     -0.07614      0.10791     -0.12685      0.44557
##      Wealth         Ineq         Prob         Time           So
##     0.66766      1.59115     -0.30382     -0.26841      0.06321
```

plotting the residuals

```
#qqplots to determine if residuals are normally distributed. Log transformed model has a better looking
modelResiduals %>%
  ggplot(aes(sample=modelResiduals$Log)) +
  stat_qq() +
  labs(title = "Log Transformed Residuals")
```




```
modelResiduals %>%
  ggplot(aes(sample=modelResiduals$Normal)) +
  stat_qq() +
  labs(title = "Normal Residuals")
```


Normal Residuals
500
250
0
−250
sample
−2
−1
0
1
2
theoretical