project-583

Lili Tang, Zhijia Ju

```
data <- read.csv("data_cleaned.csv", header = TRUE, check.names = FALSE)
dim(data) # 189, 31
data_no_date <- data[, -1]
head(data)
```

We aim to predict the price of ethylene glycol using the following 29 explanatory variables.
MEG refers to ethylene glycol.
y MEG spot price

Up-stream price of MEG
x1 WTI: West Texas Intermediate(Crude Oil)Price
x2 Brent: Brent Crude oil price
x3 Coal price

Up-stream(Ethylene)Profit
x4 domestic; x5 foreign

MEG Profit
x6 made of coal; x7 made of Ethylene

MEG operating rate
x8 domestic; x9 foreign

Downstream profits
x10 Recycled Bottle Chips; x11 polyester chips; x12 polyester bottle chip; x13 POY; x14 FDY; x15 DTY; x16 Polyester
Downstream operating rate
x17 Polyester; x18 filament; x19 Direct spinning; x20 chip spun filament; x21 texturing machine; x22 weaving machine

Downstream Inventory
x24 Polyester; x25 FDY; x26 DTY; x27 POY

MEG Inventory
x28 foreign Port; x29 domestic Port; x30 factory

1. A statistically descriptive analysis of the dataset.

The data structure is shown below. We can see that all the variables are continuous values.

```
str(data)
```

```
## 'data.frame':    189 obs. of  31 variables:
##  $ date: chr  "2019/1/4" "2019/1/11" "2019/1/18" "2019/1/25" ...
##  $ y   : int  5160 5115 4985 5070 5110 5010 4980 5125 5230 5130 ...
##  $ x1  : num  48 51.6 53.8 53.7 56.3 ...
```

```
##  $ x2  : num  57.1 60.5 62.7 61.6 62.8 ...
##  $ x3  : num  570 565 570 565 570 570 575 600 640 628 ...
##  $ x4  : num  345.2 28.3 85.5 246.8 364 ...
##  $ x5  : num  29.14 4.46 3.8 4.84 26.26 ...
##  $ x6  : num  1300 1227 1078 1166 1198 ...
##  $ x7  : num  241 234 -270 -573 -683 ...
##  $ x8  : num  0.714 0.729 0.761 0.77 0.801 ...
##  $ x9  : num  0.998 0.982 0.941 0.963 0.957 ...
##  $ x10 : num  -219.8 -155.2 -155.2 60.3 60.3 ...
##  $ x11 : num  213.8 175.6 172.2 218.4 98.1 ...
##  $ x12 : num  68.8 275.6 322.1 378.4 323.1 ...
##  $ x13 : num  -81.2 -154.4 -137.8 108.4 -11.9 ...
##  $ x14 : num  479 396 442 628 508 ...
##  $ x15 : int  490 480 470 460 460 470 450 590 215 250 ...
##  $ x16 : num  689 681 677 698 578 ...
##  $ x17 : num  84.8 83.9 81.9 78.3 74 78.6 85.4 86.6 88.5 90.5 ...
##  $ x18 : num  75.7 75.1 71.7 62.9 62.1 62.1 71.7 75.7 80.1 83.6 ...
##  $ x19 : num  82.3 81.8 78 71.7 67.5 73 82.7 84.5 86.7 88.9 ...
##  $ x20 : num  48.8 48.8 47 29 29 29 29 84 54 60 ...
##  $ x21 : int  76 72 66 17 10 20 50 81 87 89 ...
##  $ x22 : int  65 57 50 9 5 34 45 75 84 85 ...
##  $ x24 : num  4.7 0.2 1.4 0 2 5.9 7.6 6 5.2 6.4 ...
##  $ x25 : num  15.6 12.5 11.1 9.2 11.1 14.2 16.8 18 11.5 14.2 ...
##  $ x26 : num  16.7 13.5 13.8 13.8 18 27.2 27.8 28.6 22.9 24.2 ...
##  $ x27 : num  13.1 6.7 6.5 3.8 5.5 11.6 14.8 15.4 9.8 10.3 ...
##  $ x28 : int  620 624 606 618 627 608 610 629 650 641 ...
##  $ x29 : num  78.5 83.4 90.2 89.3 92.4 ...
##  $ x30 : num  15.1 14 13.5 14.8 17.8 16.5 15.4 15.4 14.8 14.6 ...
```

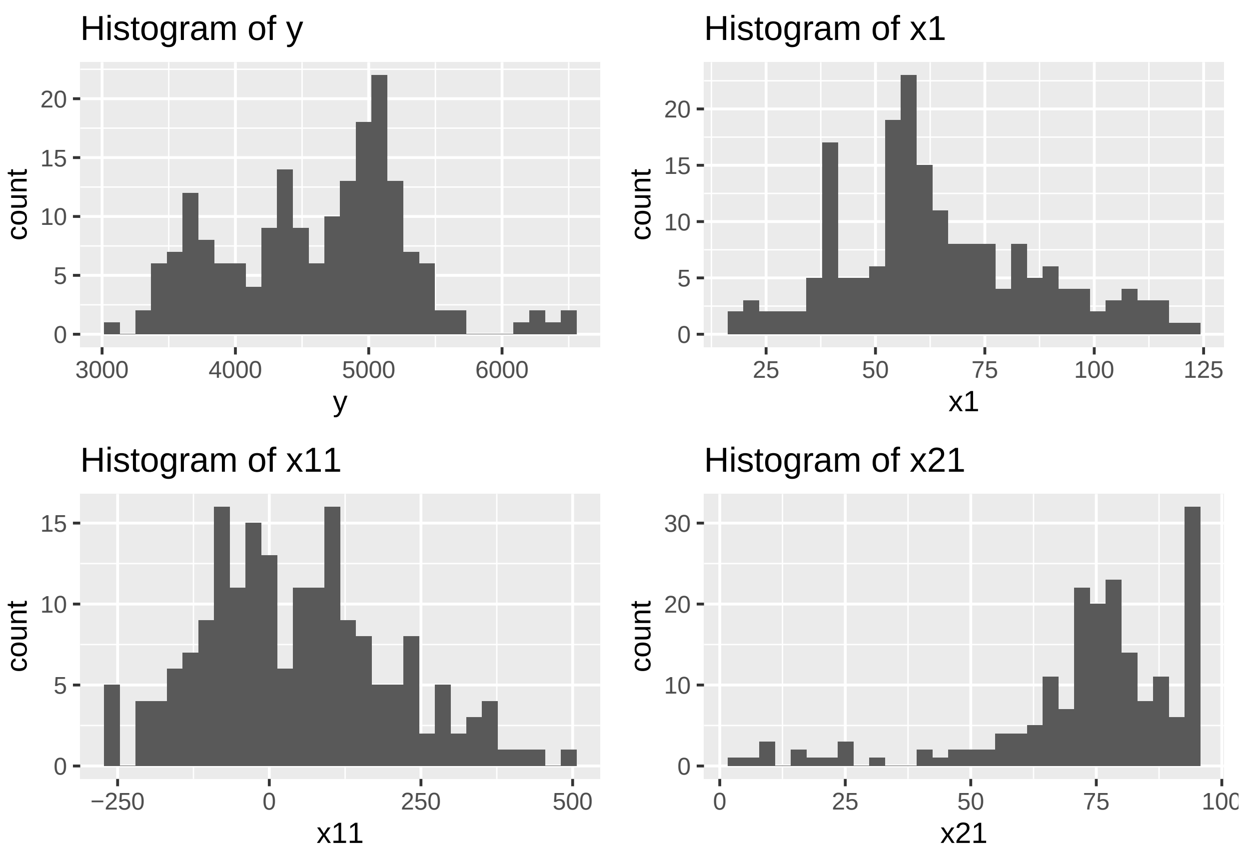
Here is the summary statistics of the response variable.

```r
summary(data$y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3130    4065    4705    4612    5110    6557
```

Let's explore the histogram of the response variable y, and some explanatory variables.

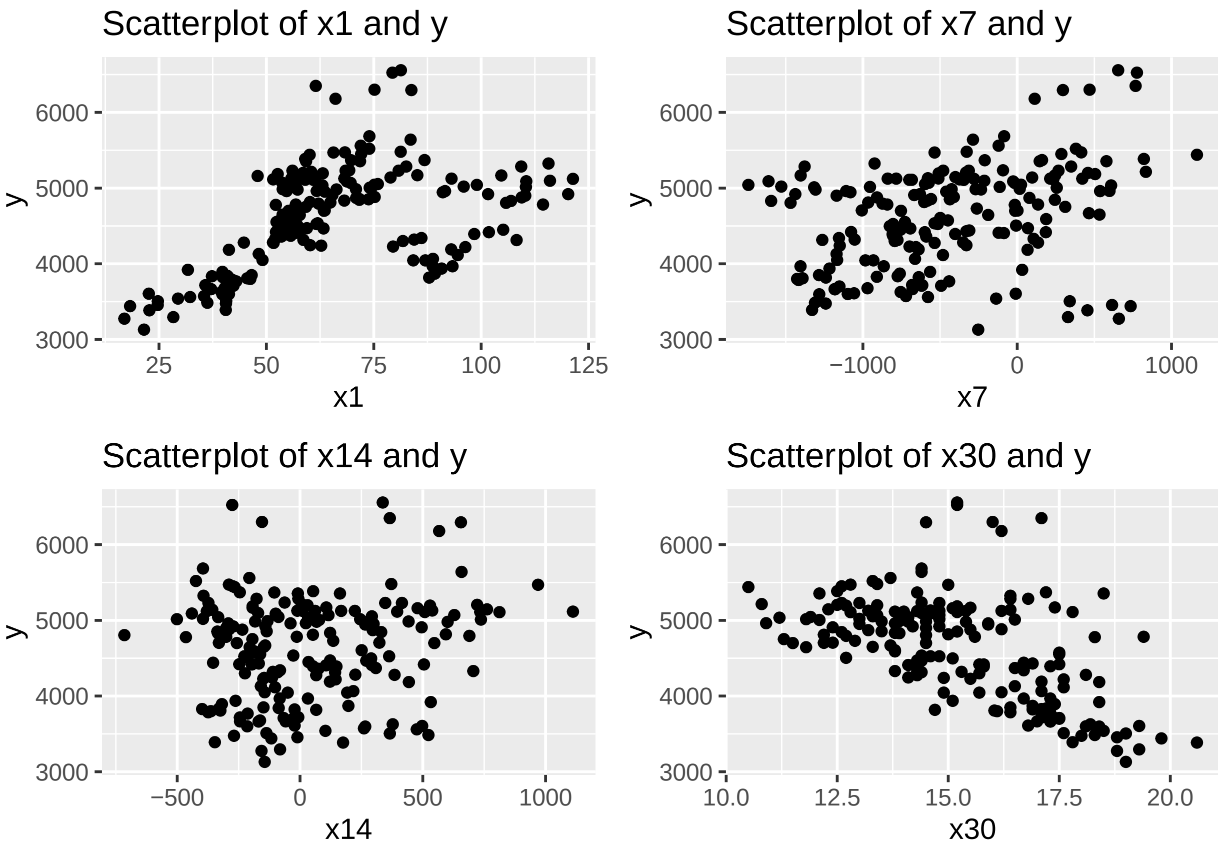
```r
hist_y <- ggplot(data, aes(x = y)) + geom_histogram() + labs(title = "Histogram of y")
hist_x1 <- ggplot(data, aes(x = x1)) + geom_histogram() + labs(title = "Histogram of x1")
hist_x11 <- ggplot(data, aes(x = x11)) + geom_histogram() + labs(title = "Histogram of x11")
hist_x21 <- ggplot(data, aes(x = x21)) + geom_histogram() + labs(title = "Histogram of x21")
grid.arrange(hist_y, hist_x1, hist_x11, hist_x21, nrow = 2, ncol = 2)
```



The histogram of the response variable y seems to have 3 peaks. The histogram of the explanatory variable x1 and x11 are right skewed. The histogram of the explanatory variable x21 is left skewed.

Let's explore the scatter plots of the response variable y and some explanatory variables.

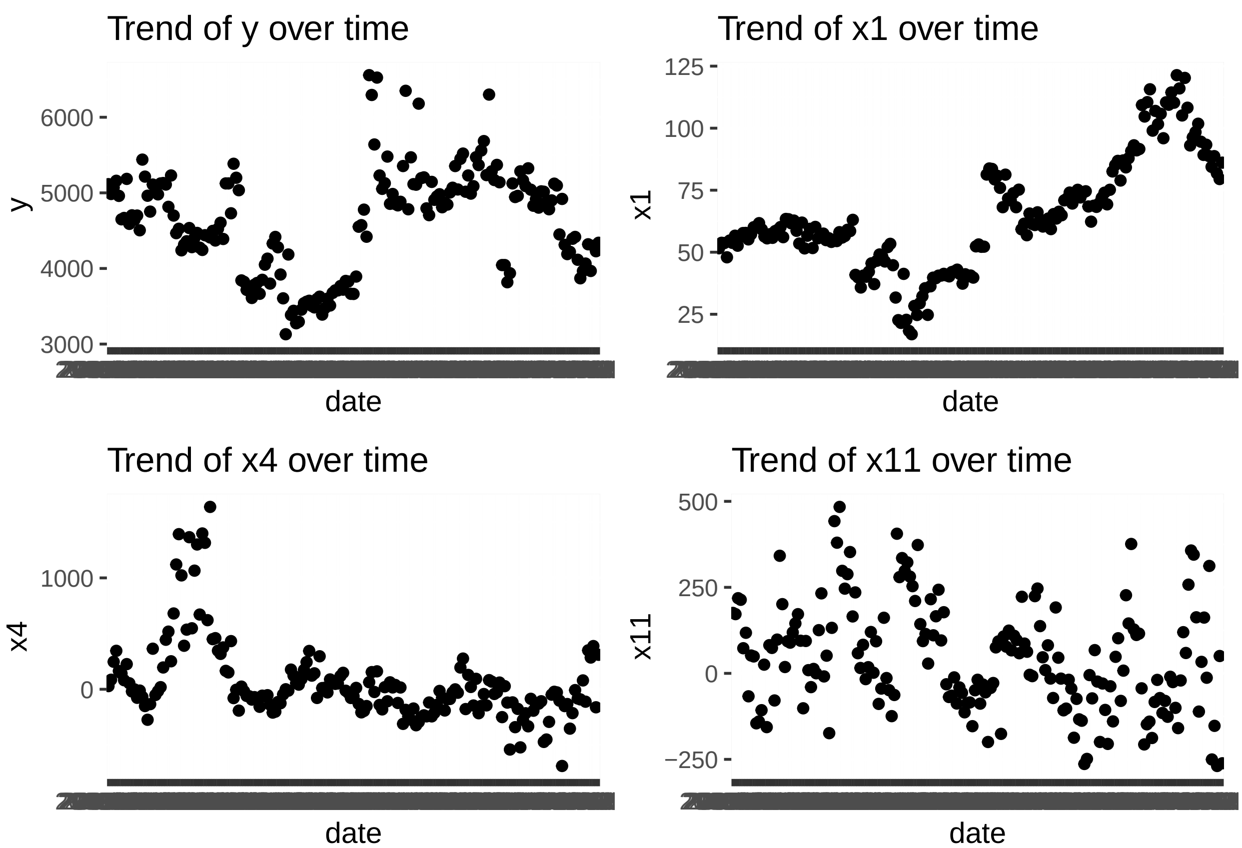
```
# create scatterplots of the variables x1, x2, x3, x4, x5, x7, x10, x11, x12
scatter_x1 <- ggplot(data = data, aes(x = x1, y = y)) + geom_point() + labs(title = "Scatterplot of x1 a
scatter_x7 <- ggplot(data = data, aes(x = x7, y = y)) + geom_point() + labs(title = "Scatterplot of x7 a
scatter_x14 <- ggplot(data = data, aes(x = x14, y = y)) + geom_point() + labs(title = "Scatterplot of x1
scatter_x30 <- ggplot(data = data, aes(x = x30, y = y)) + geom_point() + labs(title = "Scatterplot of x3
grid.arrange(scatter_x1, scatter_x7, scatter_x14, scatter_x30, ncol = 2, nrow = 2)
```



There seems to have a positive linear relationship between the response variable y and the explanatory variables x1, x7. And a negative relationship between y and x30. It seems reasonable because x1 represent Crude Oil Price, which is positively related to the price of y(MEG), while x30 represents the factory inventory, the higher the inventory, the lower the price. There is no obvious relationship between y and x14.

Let's explore the trend of the response variable y and some explanatory variables over time.

```
# plot the variables versus date
trend_y <- ggplot(data = data, aes(x = date, y = y)) + geom_point() + labs(title = "Trend of y over time
trend_x1 <- ggplot(data = data, aes(x = date, y = x1)) + geom_point() + labs(title = "Trend of x1 over t
trend_x4 <- ggplot(data = data, aes(x = date, y = x4)) + geom_point() + labs(title = "Trend of x4 over t
trend_x11 <- ggplot(data = data, aes(x = date, y = x11)) + geom_point() + labs(title = "Trend of x11 ove
grid.arrange(trend_y, trend_x1, trend_x4, trend_x11, ncol = 2, nrow = 2)
```



The trend plots of y and x1 seem to have some similarities. The trend plots of x4 and x11 have no obvious trend.

2. Applications of statistical analysis techniques

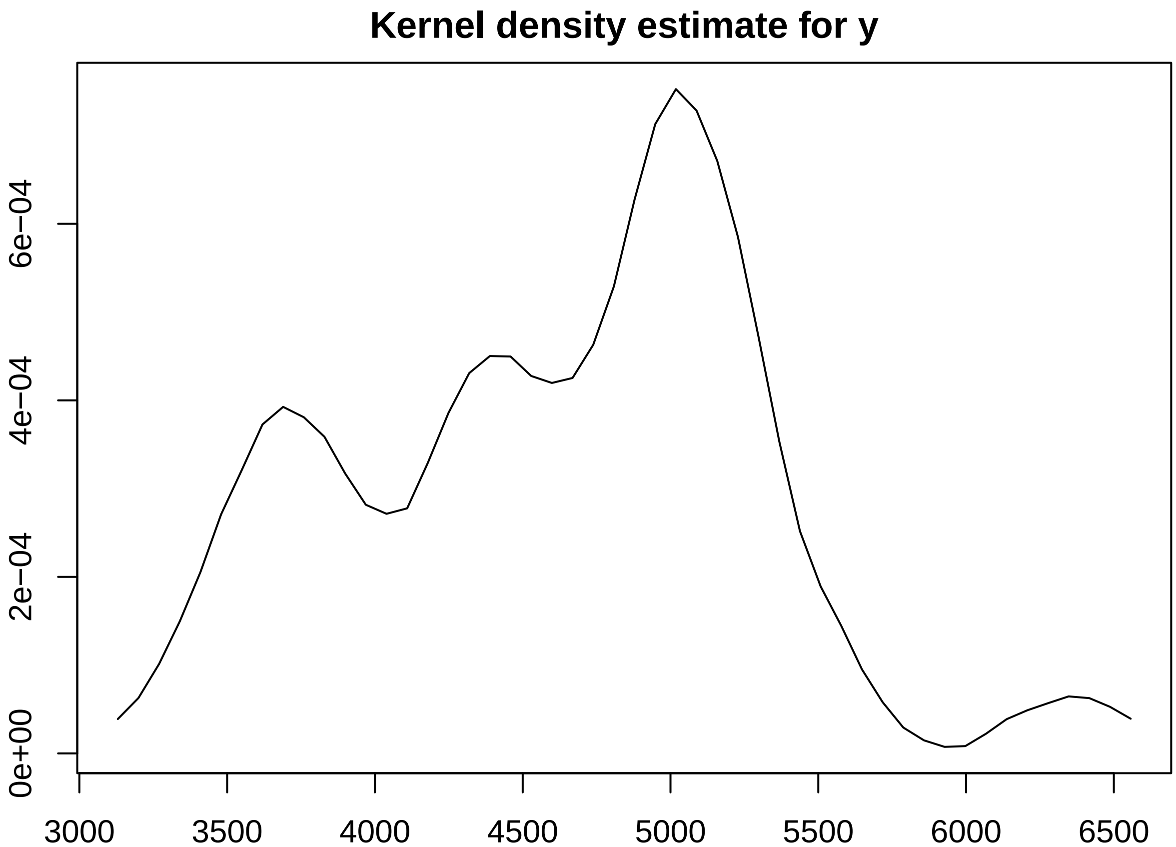
kernel density

Let's try to plot a kernel density estimate for y using an Epanechnikov kernel.

```
par(mar = c(2, 2, 2, 2))
bw <- npudensbw( ~ y, data = data, ckertype = "epanechnikov", bwmethod = "cv.ml")
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multistart 1 of 1 |Multi
```

```
fhat <- npudens(bws = bw)
plot(fhat, main = "Kernel density estimate for y")
```



As stated in the histogram part, the distribution of y is not normal, and it seems to have 3 peaks.

Let's conduct Pearson tests for normality.

```
pearson.test(data$y)
```

```
##
##  Pearson chi-square normality test
##
## data:  data$y
## P = 44.772, p-value = 4.441e-05
```

```
pearson.test(data$x11)
```

```
##
##  Pearson chi-square normality test
##
## data:  data$x11
## P = 16.349, p-value = 0.2925
```

From the pearson chi-square normality tests, we can see that the p-value is very small for y, and large for x11. So we have evidence that y is not normally distributed, but x11 is normally distributed.

Linear regression

Let's try a linear regression model to do diagnostics.

```r
model_ls <- lm(y ~ ., data = data_no_date)
summary(model_ls)
```

```
##
## Call:
## lm(formula = y ~ ., data = data_no_date)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -165.392  -52.794   -1.755   44.694  171.121
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.981e+03  3.006e+02   6.590 6.14e-10 ***
## x1          -7.209e+00  4.750e+00  -1.518 0.131105
## x2           9.303e+00  4.693e+00   1.982 0.049149 *
## x3           1.185e+00  1.305e-01   9.076 4.13e-16 ***
## x4           3.944e-02  5.211e-02   0.757 0.450279
## x5           7.938e-02  5.962e-01   0.133 0.894244
## x6           4.855e-01  5.288e-02   9.181  < 2e-16 ***
## x7           9.900e-02  1.654e-02   5.985 1.38e-08 ***
## x8          -1.227e+02  1.491e+02  -0.823 0.411734
## x9           2.112e+02  1.421e+02   1.486 0.139360
## x10         -9.404e-02  4.265e-02  -2.205 0.028886 *
## x11          1.536e-01  8.178e-02   1.878 0.062259 .
## x12          8.932e-03  2.857e-02   0.313 0.754965
## x13         -1.344e-01  4.317e-02  -3.115 0.002185 **
## x14          1.451e-01  4.062e-02   3.571 0.000471 ***
## x15         -6.469e-02  5.319e-02  -1.216 0.225689
## x16         -1.069e-01  5.485e-02  -1.950 0.052972 .
## x17         -3.266e+00  4.925e+00  -0.663 0.508218
## x18          1.187e-01  1.250e-01   0.950 0.343681
## x19          6.022e+00  4.419e+00   1.363 0.174922
## x20         -3.370e+00  1.397e+00  -2.412 0.016986 *
## x21         -2.405e+00  1.211e+00  -1.986 0.048775 *
## x22          3.065e+00  1.164e+00   2.634 0.009275 **
## x24          5.010e+00  3.501e+00   1.431 0.154350
## x25         -1.115e+01  4.879e+00  -2.285 0.023624 *
## x26         -3.468e+00  3.101e+00  -1.118 0.265201
## x27          1.080e+01  3.528e+00   3.060 0.002597 **
## x28          3.228e+00  4.014e-01   8.043 1.90e-13 ***
## x29         -5.339e-01  6.139e-01  -0.870 0.385811
## x30         -2.251e+01  5.826e+00  -3.864 0.000162 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.85 on 159 degrees of freedom
## Multiple R-squared:   0.99,  Adjusted R-squared:  0.9882
## F-statistic: 542.8 on 29 and 159 DF,  p-value: < 2.2e-16
```

From the summary output, the can see that only variables x3, x6, x7, x13, x14, x20, x22, x25, x27, x28 and x30 are significant. The R-squared and adjusted R-square are high, which is expected as we have lots of variables. Variable selection is needed to reduce the number of variables.

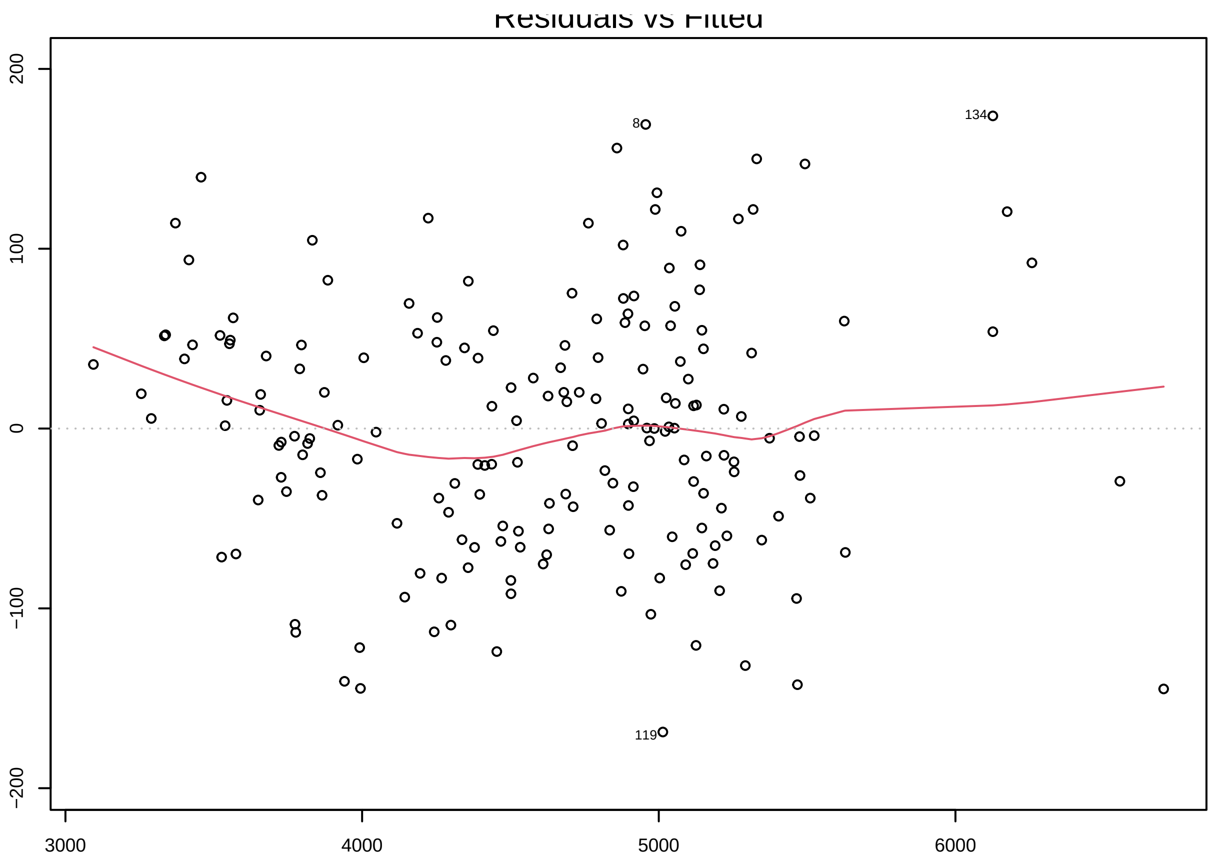
Let's try to do variable selection using stepwise regression.

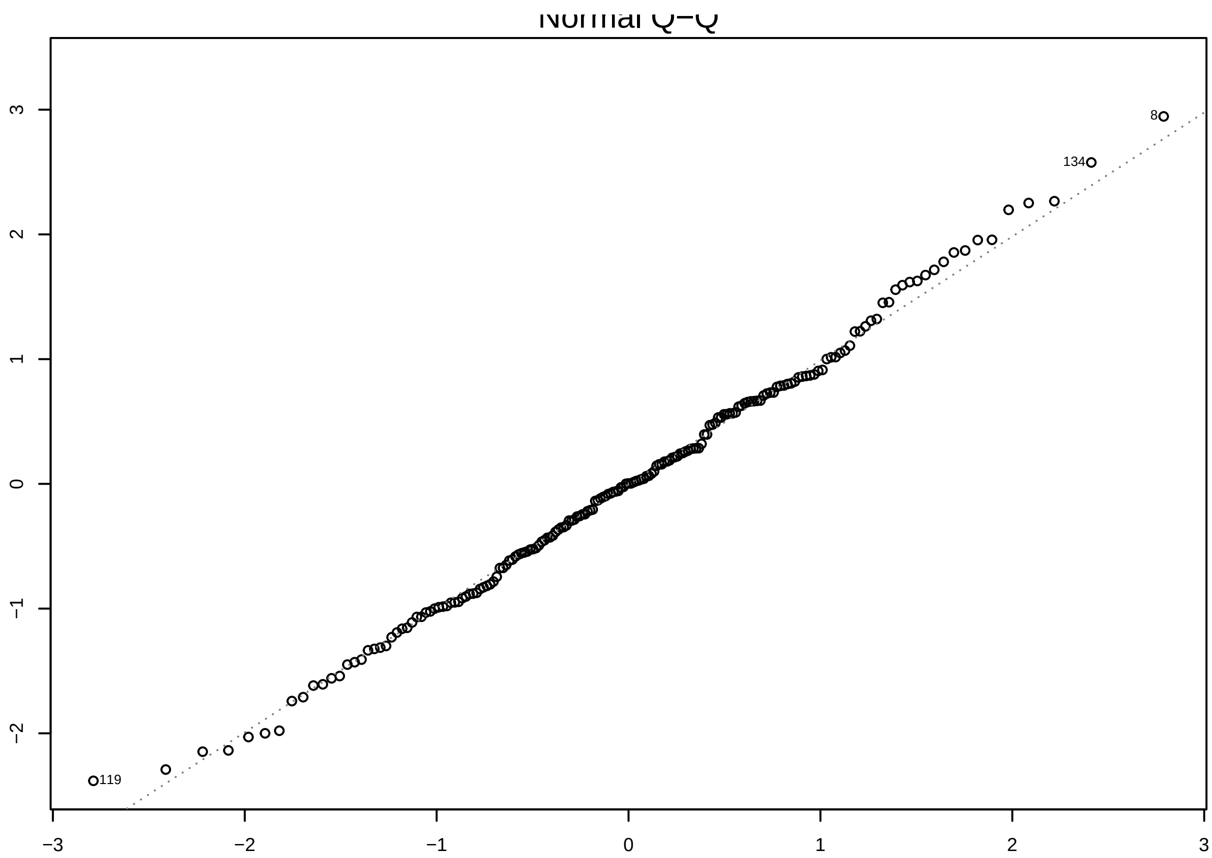
```
model_step <- stepAIC(model_ls, direction = "both", criterion = "bic", trace = FALSE)
model_step
```

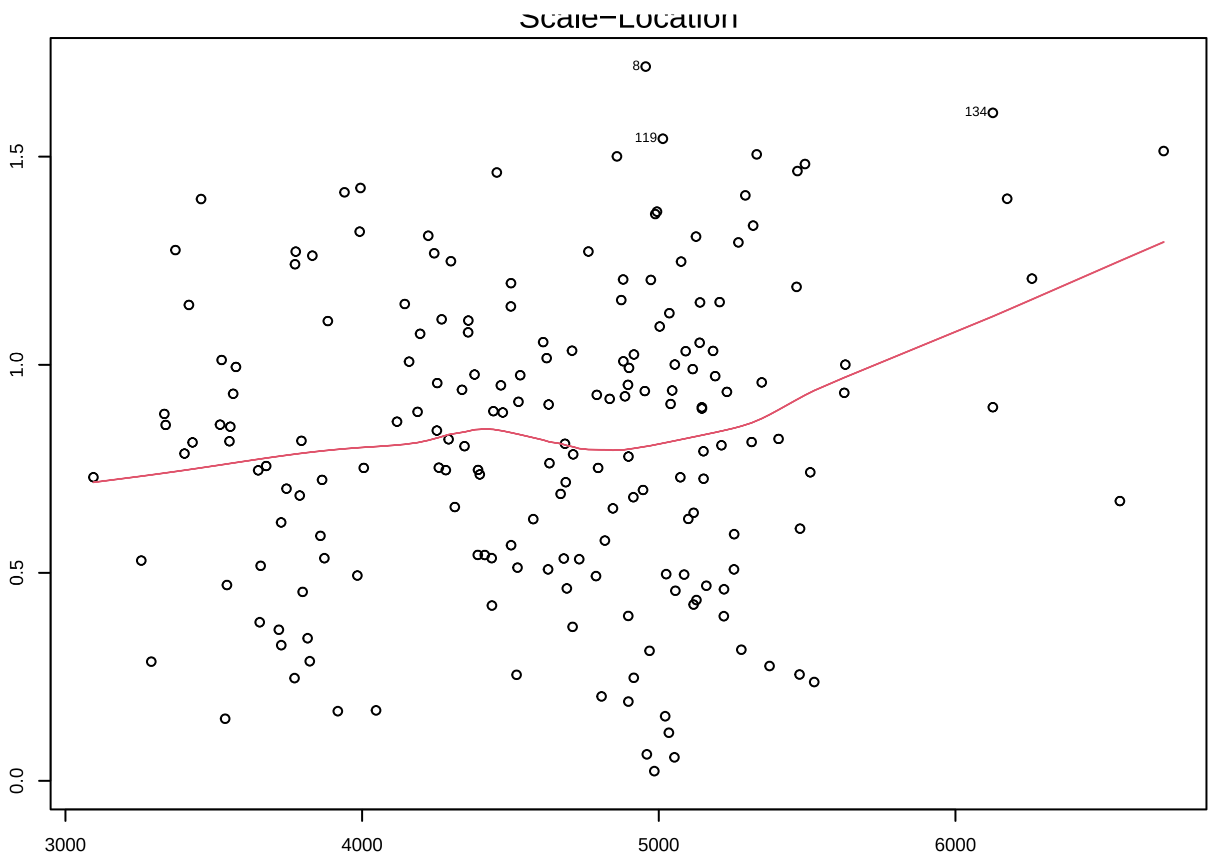
```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x6 + x7 + x9 + x10 + x11 +
##     x13 + x14 + x15 + x16 + x19 + x20 + x21 + x22 + x24 + x25 +
##     x26 + x27 + x28 + x30, data = data_no_date)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x6
##  1858.80540     -9.16272     11.29471      1.22914      0.04681      0.49736
##          x7           x9          x10          x11          x13          x14
##     0.10436    217.80130     -0.11073      0.18016     -0.11633      0.12868
##         x15          x16          x19          x20          x21          x22
##    -0.06754     -0.14431      3.00727     -3.38212     -1.94951      2.42204
##         x24          x25          x26          x27          x28          x30
##     4.01386     -8.11560     -5.01568     10.32603      3.15784    -24.67567
```

From the stepwise model selection, 6 variables are removed from the model.

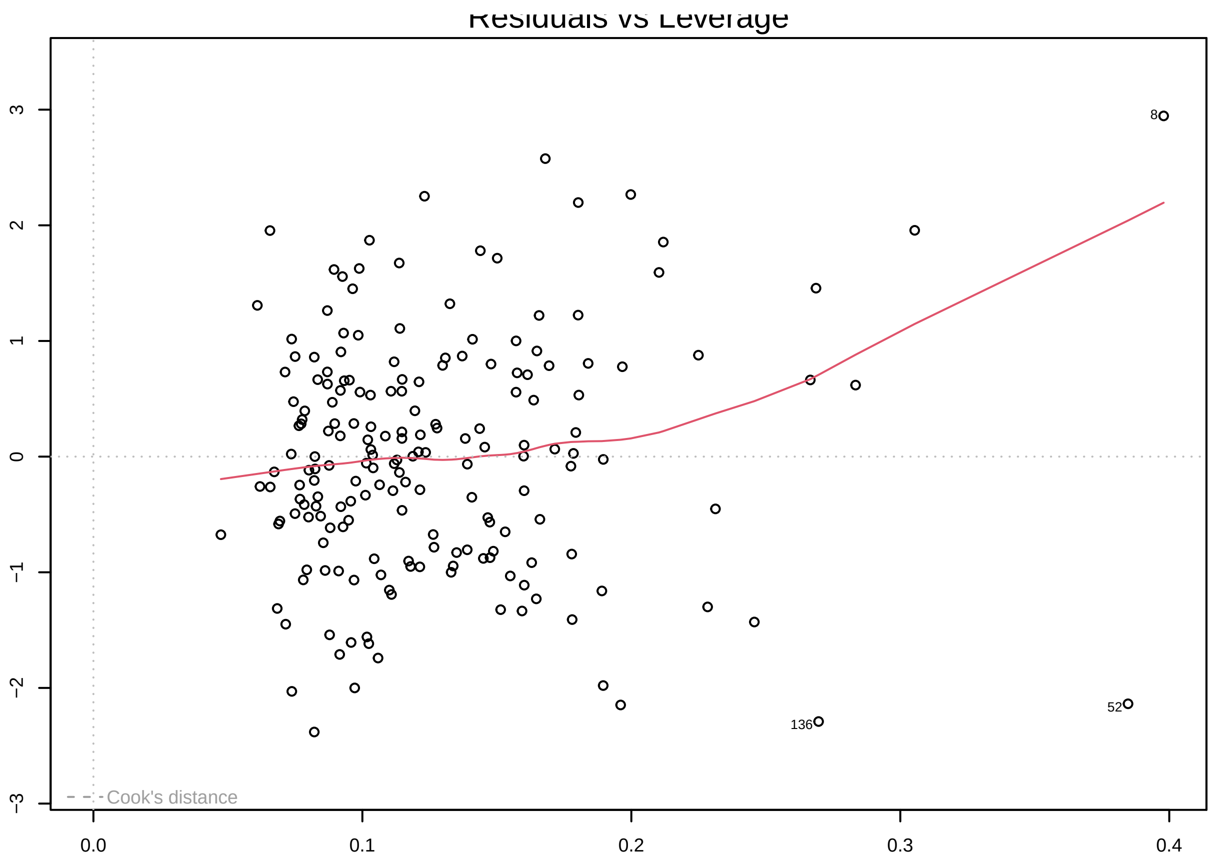
```
par(mar = c(3, 3, 1, 1), cex = 0.6)
plot(model_step)
```


Residuals vs Fitted
200
100
0
−100
−200
3000
4000
5000
6000
8
134
119


Normal Q−Q
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
8
134
119



Scale−Location
8
119
134
0.0
0.5
1.0
1.5
3000
4000
5000
6000

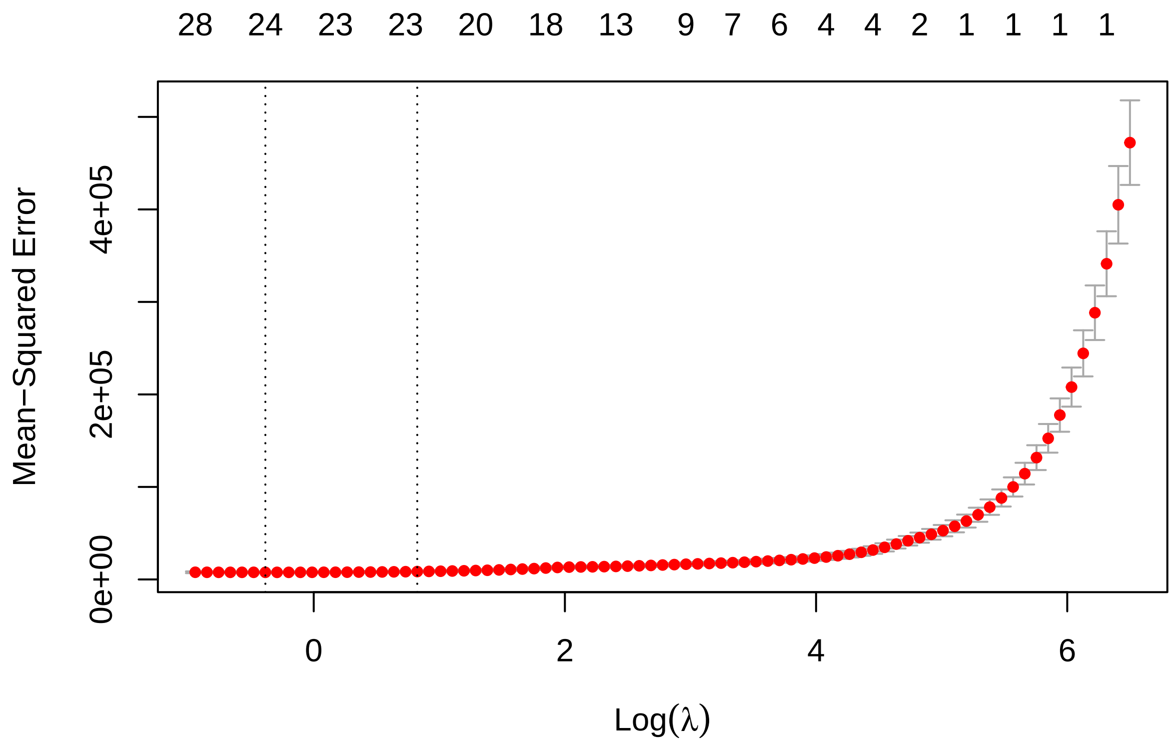


The residual plot shows no obvious trend. The scale-location plot shows a slightly increasing trend. The normal Q-Q plot shows that the residuals are roughly normally distributed. The leverage plot shows that there are couple potential outliers.

Lasso

Let's try to do variable selection using lasso regression.

```
knitr::opts_chunk$set(fig.width=5, fig.height=4)
model_lasso <- cv.glmnet(x = as.matrix(data_no_date[, -1]), y = data_no_date[, 1], alpha = 1)
plot(model_lasso)
```



Both minimum and 1se lines suggest to keep 23 variables, which is the same as the stepwise model selection.

3. Scientific questions.

4. Statistical analysis techniques I will use to answer those questions (with justification).

Q1. Based on our dataset, which models can be used to forecast the price of ethylene glycol?

From the above analysis, linear regression model seems to be good in interpretation. R-square is 0.99 and many variables are significant. However, there is a potential over-fitting problem with so many variables included in the model. Therefore, we can use other methods such as **nonparametric local linear regression model or regression trees or random forest**.

We also can consider using the **logistic regression**, but we need to convert the continuous response variable into a binary outcome by applying a threshold value. For example, if the response variable increases compared with last day or last week, then we assign it to 1, otherwise, we assign it to 0.

Nonparametric local linear regression model

```
library(np)
#bw <- npregbw(y ~ x1 + x2 + x3 + x4 + x6 + x7 + x9 + x10 + x11 +
#   x13 + x14 + x15 + x16 + x19 + x20 + x21 + x22 + x24 + x25 +
#   x26 + x27 + x28 + x30,data=data_no_date,regtype="ll")
```

```
#model.ll <- npreg(bws=bw)
#summary(model.ll)
```

In nonparametric local linear regression model, R square is quite close to 1, which is expected with so many variables. We should use some variable selection methods to reduce the number of variables.

Logistic

```
data_logistic <- data_no_date[-1,]
data_logistic$newy <- ifelse(diff(data_no_date$y) > 0, 1, 0)
data_logistic$newy <- as.factor(data_logistic$newy)
model_Log<- glm(newy ~., data=data_logistic[,-1],family =binomial)
model_Log
```

```
##
## Call:  glm(formula = newy ~ ., family = binomial, data = data_logistic[,
##     -1])
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##  -9.740e+00   -4.922e-01    4.557e-01   -4.805e-03   -2.858e-03    1.070e-02
##          x6           x7           x8           x9          x10          x11
##  -1.511e-03   -1.358e-04   -1.403e+01    1.912e+01    2.153e-03   -4.981e-03
##         x12          x13          x14          x15          x16          x17
##  -1.557e-03   -4.663e-04   -1.151e-03   -2.624e-03   -9.184e-04    2.506e-01
##         x18          x19          x20          x21          x22          x24
##   2.042e-01   -5.879e-01    4.699e-02    1.139e-02   -6.274e-03    1.132e-02
##         x25          x26          x27          x28          x29          x30
##  -1.111e-01    3.385e-02   -1.722e-01    2.020e-02   -1.899e-02    6.198e-01
##
## Degrees of Freedom: 187 Total (i.e. Null);  158 Residual
## Null Deviance:       260.4
## Residual Deviance: 169.3     AIC: 229.3
```

```
model_Log_step <- stepAIC(model_Log, direction = "both", criterion = "bic", trace = FALSE)
model_Log_step
```

```
##
## Call:  glm(formula = newy ~ x1 + x2 + x4 + x8 + x9 + x10 + x11 + x14 +
##     x15 + x19 + x20 + x27 + x28 + x30, family = binomial, data = data_logistic[,
##     -1])
##
## Coefficients:
## (Intercept)           x1           x2           x4           x8           x9
##    2.189873    -0.410252     0.368373    -0.002149   -11.226781    14.832846
##         x10          x11          x14          x15          x19          x20
##    0.001883    -0.006644    -0.001311    -0.003754    -0.221076     0.083814
##         x27          x28          x30
##   -0.220353     0.007029     0.447557
##
```

```
## Degrees of Freedom: 187 Total (i.e. Null);  173 Residual
## Null Deviance:       260.4
## Residual Deviance: 176.2     AIC: 206.2
```

The variable selection helped us to reduce the number of variables, and we have a lower BIC value.

Q2. There are 29 variables in our dataset, how can we reduce the dimension and avoid multicollinearity?

There are 2 methods we can consider, the first method is lasso, as used above, it can be used to reduce high-dimensional data in a model by shrinking the coefficients of irrelevant variables to zero. The other method is PCA, which aims to reduce the dimensionality of the data while retaining as much of its variance as possible.

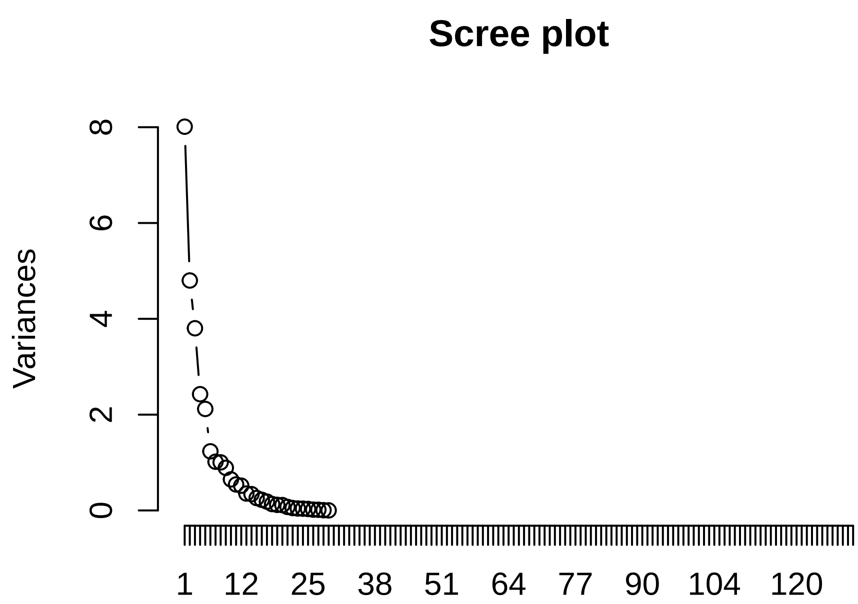
PCA

```r
pca <- prcomp(data_no_date[,-c(1)], scale.=TRUE)
summary(pca)
```

```r
knitr::opts_chunk$set(fig.width=5, fig.height=4)
var_explained <- cumsum(pca $sdev^2 / sum(pca $sdev^2))
# Determine the number of components needed to retain at least 90% of the variance
which.max(var_explained >= 0.9)
```

```
## [1] 11
```

```r
plot(pca, type="lines", npcs=131, main = "Scree plot")
```



By using PCA, we can reduce the 29 variables to 11 principal components (those can explain 90% of the variance).

Q3. which model is the most appropriate and accurate?

First of all, it depends on our purpose. If our priority is to interpret, linear regression, logistic regression or trees will be better. This involves examining the coefficients of the model to determine the direction and strength of the relationships between the variables. This information can be used to identify the most important variables and to generate hypotheses about the underlying mechanisms driving the relationship between the variables.

If we focus more on predication, we can divide our data into training and testing datasets, using cross validation to test the model with the smallest test MSE. Usually, random forest or boosting would be better in this case. As the data is time series, we can also use ARIMA model to do prediction.