regression analysis for the paper

Tianxiang Zhou

2019/7/13

```
setwd('C:/Users/coolt/Desktop')
rm(list=ls())
library(knitr)
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.5.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.5.3
```

```
data <- read.csv('10000samples.csv')
dim(data)
```

```
## [1] 10000    30
```

```
attach(data)
```

PART I: Mutiple Linear Regression for DepDelay, ArrDelay

DepDelay Analysis

Predictors: 'Month','DayofMonth','DayOfWeek','DepTime',
'CRSDepTime','ArrTime','CRSArrTime','CRSElapsedTime'
'AirTime','Distance','ActualElapsedTime'

Response:
DepDelay

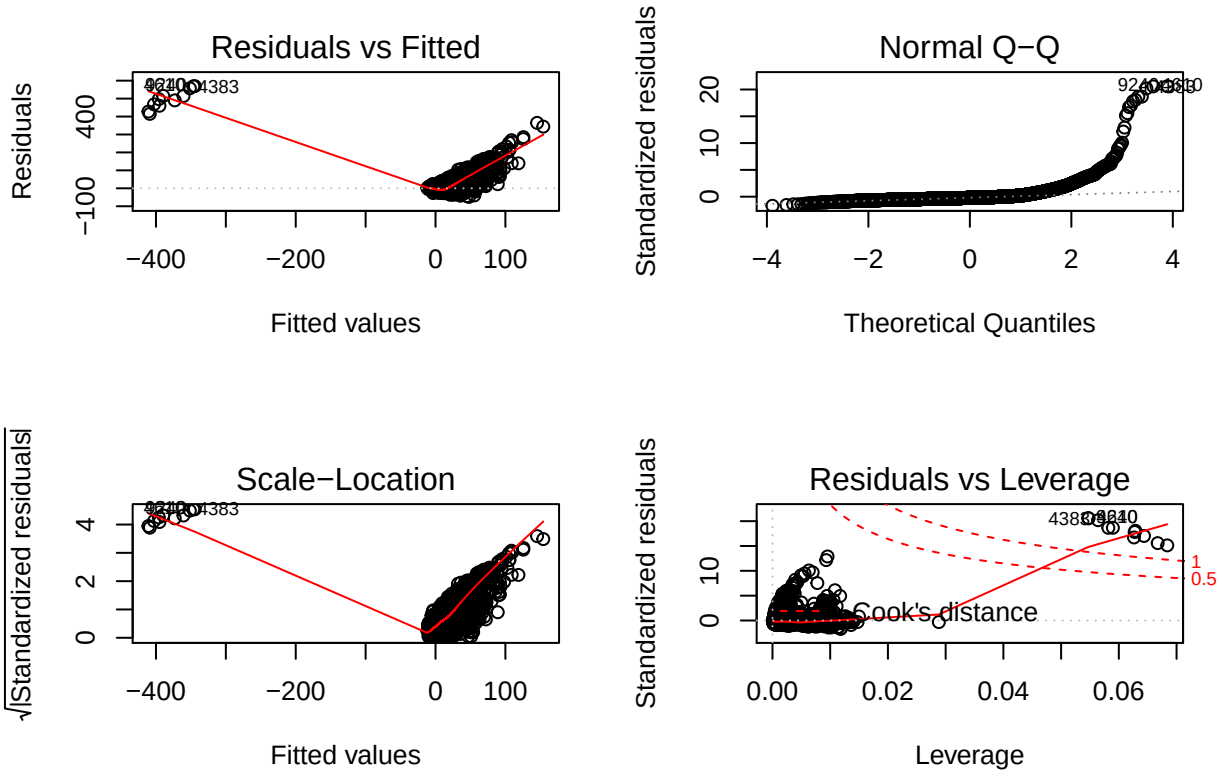
```
model <-lm(DepDelay~Month+DayofMonth+DayOfWeek+DepTime+CRSArrTime+CRSDepTime
           +CRSElapsedTime+ArrTime+Distance+ActualElapsedTime+ArrTime)

kable(summary(model)$coef,
      booktabs = TRUE,
      caption = "Coefficients DepDelay model.")
```

Table 1: Coefficients DepDelay model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.0338307	1.6366363	-3.0757174	0.0021056
Month	-0.4003120	0.0839750	-4.7670377	0.0000019
DayofMonth	-0.0036749	0.0324290	-0.1133206	0.9097787
DayOfWeek	0.2002388	0.1429527	1.4007350	0.1613244
DepTime	0.1848587	0.0030997	59.6379024	0.0000000
CRSArrTime	0.0048298	0.0013795	3.5010521	0.0004654
CRSDepTime	-0.1737201	0.0031729	-54.7519516	0.0000000
CRSElapsedTime	-0.0053580	0.0298690	-0.1793826	0.8576410
ArrTime	-0.0091001	0.0011317	-8.0409695	0.0000000
Distance	-0.0139629	0.0027591	-5.0607305	0.0000004
ActualElapsedTime	0.1329548	0.0199155	6.6759432	0.0000000

We can see that: 28.42% of the variance in DepDelay can be explained by the 11 predictors, but some of the predictors have really large p-values.

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



As we made the following plot, it seems like there are some outliers.
Let's do something to discard them(discard points who has standarized residual>4)

```
std_residuals <- rstandard(model)
indx <- std_residuals<4
newdata <- data[indx,]
```

```
dim(newdata)
```

```
## [1] 9918   30
```

```
detach(data)
attach(newdata)
```

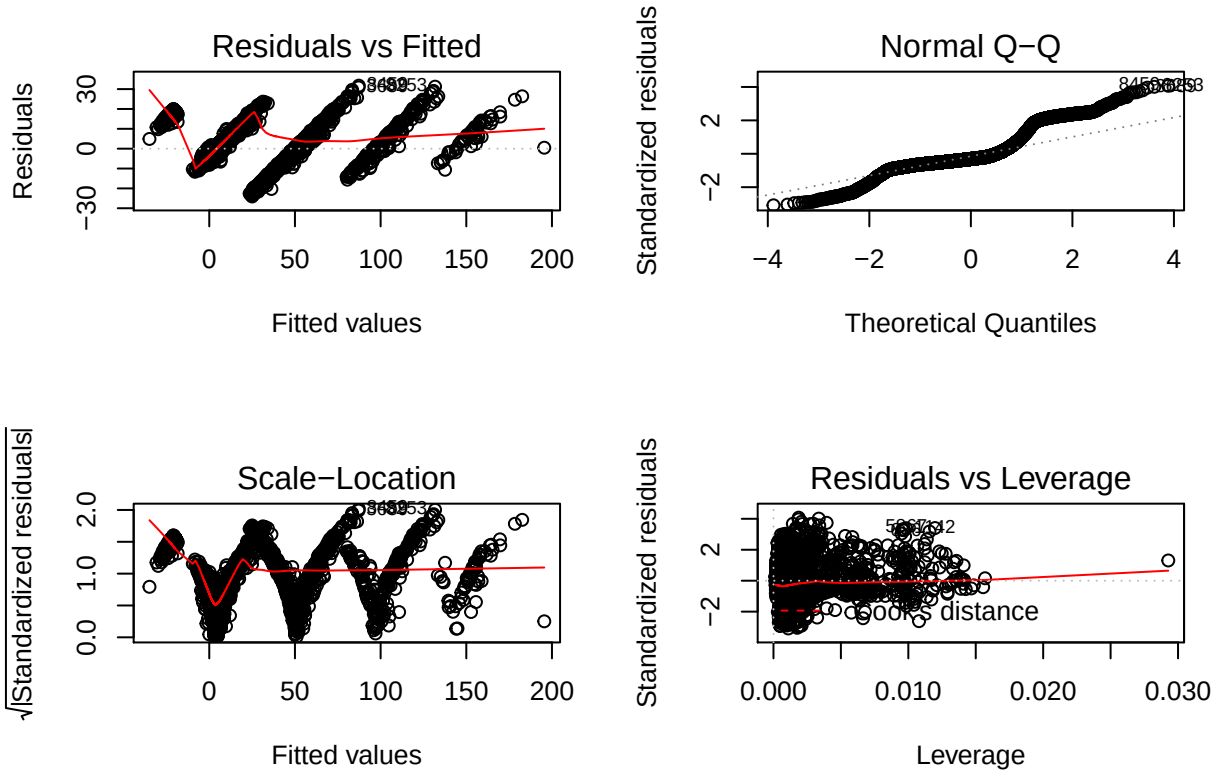
Now let's do some new analysis to the new data

```
model_new <-lm(DepDelay~Month+DayofMonth+DayOfWeek+DepTime+CRSArrTime+CRSDepTime
           +CRSElapsedTime+ArrTime+Distance+ActualElapsedTime+ArrTime)

kable(summary(model_new)$coef,
      booktabs = TRUE,
      caption = "Coefficients DepDelay model with new data.")
```

Table 2: Coefficients DepDelay model with new data.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.4423956	0.4523419	5.3994461	0.0000001
Month	-0.0651894	0.0231850	-2.8117082	0.0049376
DayofMonth	-0.0030015	0.0089342	-0.3359610	0.7369074
DayOfWeek	-0.0233292	0.0394190	-0.5918255	0.5539810
DepTime	0.5376599	0.0017375	309.4438587	0.0000000
CRSArrTime	0.0007455	0.0003916	1.9034354	0.0570127
CRSDepTime	-0.5376686	0.0017810	-301.8936022	0.0000000
CRSElapsedTime	-0.0009229	0.0082418	-0.1119826	0.9108395
ArrTime	-0.0011980	0.0003227	-3.7126729	0.0002062
Distance	-0.0020377	0.0007619	-2.6745822	0.0074945
ActualElapsedTime	0.0193953	0.0055193	3.5141049	0.0004432

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



As we see, it fits so much better and it follows all the assumptions.

Right now, let's proceed with subset selections.

```
##discard predictors with extremely high p-values
newdata2 <- data.frame(DepDelay,Month,DepTime,CRSArrTime,CRSDepTime
                       ,ArrTime,Distance,ActualElapsedTime, AirTime)
regfit <- regsubsets(DepDelay~.,data=newdata2)
summary_regfit <- summary(regfit)
names(regfit)
```

```
##  [1] "np"         "nrbar"      "d"          "rbar"       "thetab"
##  [6] "first"      "last"       "vorder"     "tol"        "rss"
## [11] "bound"      "nvmax"      "ress"       "ir"         "nbest"
## [16] "lopt"       "il"         "ier"        "xnames"     "method"
## [21] "force.in"   "force.out"  "sserr"      "intercept"  "lindep"
## [26] "nullrss"    "nn"         "call"
```

```
summary_regfit$rsq
```

```
## [1] 0.06849989 0.91102633 0.91110885 0.91131810 0.91141365 0.91148634
## [7] 0.91151958 0.91152116
```

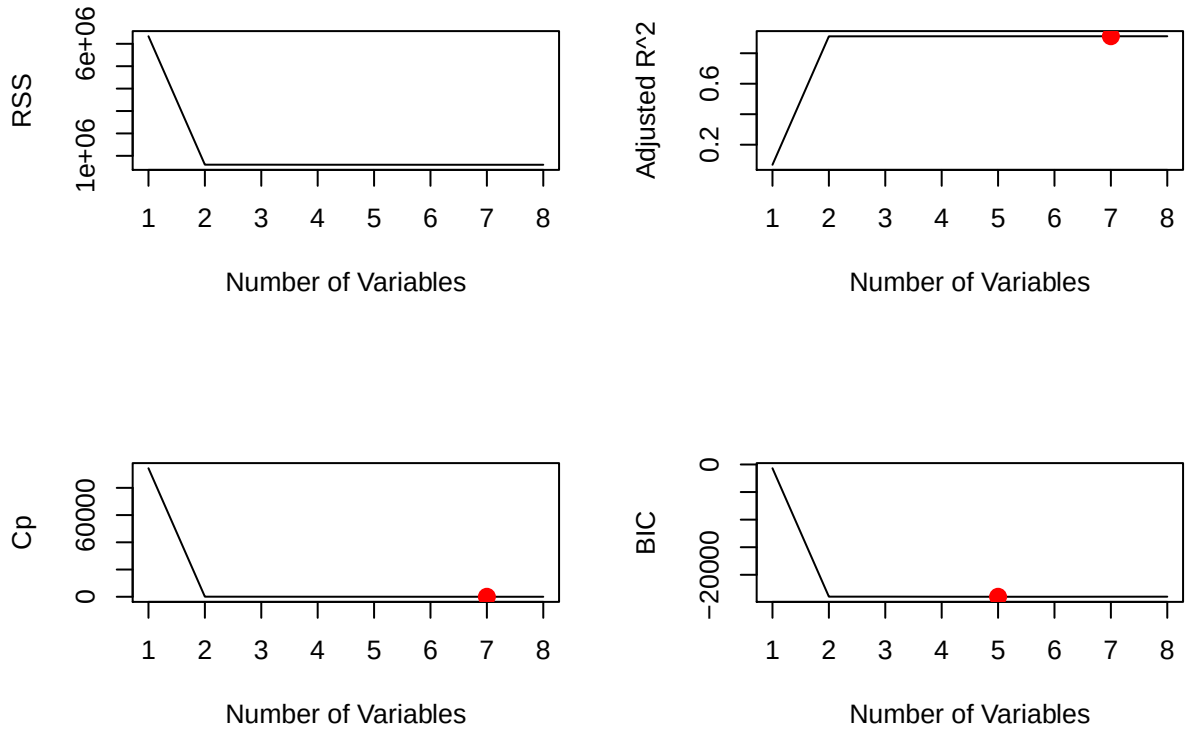
We see that the R2 statistic increases from 6% when only one variable is included inthe model to almost 91.1% when two variables are included. As expected, the R2 statistic increases monotonically as more variables are included.

Let's create a function that generates plots of RSS, cp, BIC *and adjusted* R^2

```
subsetplot <- function(summary_regfit){
  par(mfrow=c(2,2))
  plot(summary_regfit$rss, xlab='Number of Variables', ylab='RSS', type='l')
  plot(summary_regfit$adjr2, xlab = "Number of Variables", ylab = "Adjusted R^2", type = "l")
  adj_r2_max = which.max(summary_regfit$adjr2)
  points(adj_r2_max, summary_regfit$adjr2[adj_r2_max], col ="red", cex = 2, pch = 20)
  plot(summary_regfit$cp,xlab = "Number of Variables", ylab = "Cp", type = "l")
  cp_min = which.min(summary_regfit$cp) # 10
  points(cp_min, summary_regfit$cp[cp_min], col = "red", cex = 2, pch = 20)
  plot(summary_regfit$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
  bic_min = which.min(summary_regfit$bic) # 6
  points(bic_min, summary_regfit$bic[bic_min], col = "red", cex = 2, pch = 20)
}
```

Let's generate some plots

```
subsetplot(summary_regfit)
```



As we can see from the plots here, 2 predictors should work just fine, more predictors did not offer any improvement.

let's output the coefficent

```
coef(regfit, 2)
```

```
## (Intercept)     DepTime  CRSDepTime
##   2.5621895   0.5390955  -0.5394724
```

Create a new linear model with just 2 predictors: $DepTime$ $CRSDepTime$

```r
model_2v <- lm(DepDelay~DepTime+CRSDepTime)
summary(model_2v)
```

```
##
## Call:
## lm(formula = DepDelay ~ DepTime + CRSDepTime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.341  -4.173  -2.381   1.879  31.212
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.562190   0.237998   10.77   <2e-16 ***
## DepTime      0.539095   0.001720  313.39   <2e-16 ***
## CRSDepTime  -0.539472   0.001761 -306.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.813 on 9915 degrees of freedom
## Multiple R-squared:  0.911,  Adjusted R-squared:  0.911
## F-statistic: 5.076e+04 on 2 and 9915 DF,  p-value: < 2.2e-16
```

From summary, we have a .911 adjusted R^2, meaning 91.1% of the variance in DepDelay can be explained by DepTime and CRSDeptime