Data Analytics

Yong Zheng

Illinois Institute of Technology

Chicago, IL, 60616, USA

School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

Multiple Linear Regression

- General Workflow
- Advanced Topics
  - Multicollinearity Problems
  - Dummy Variables (When X is a qualitative variable)
  - Higher-Order Multiple Linear Regressions
  - Interaction Terms
  - Influential Points
- Final Note: Predictions

Multicollinearity using SAS/R

SAS users

The "tolerance" and "vif" multi-collinearity statistics are computed using the option "vif" or "tol" in the model statement.

```
PROC REG;
MODEL yvar = xvar_1 xvar_2 ... xvar_k / vif tol;
RUN;
```

R users

```
fit = lm(y~xvar1+xvar2)
# Evaluate Collinearity
vif(fit) # variance inflation factors
sqrt(vif(fit)) > 2 # problem?
```

How do we include qualitative variables in the regression model?

Dummy Variable == Binary Variable

What if a qualitative that has more than 2 values?

Season
Spring
Summer
Fall
Winter
Fall

Spring	Summer	Fall
1	0	0
0	1	0
0	0	1
0	0	0
0	0	1

You can convert qualitative variable to multiple dummy variables
Usually N-1 new variables is enough. Not necessary to have N ones

Creating dummy variables in R

METHOD 1

Create dummy variables:

```
numstar= (star == "Star")*1;
numsum= (release == "Summer")*1;
```

METHOD 2

Using the `as.factor()` function to automatically transform the categorical variable in factors or dummy variables to be used in LM() regression model.

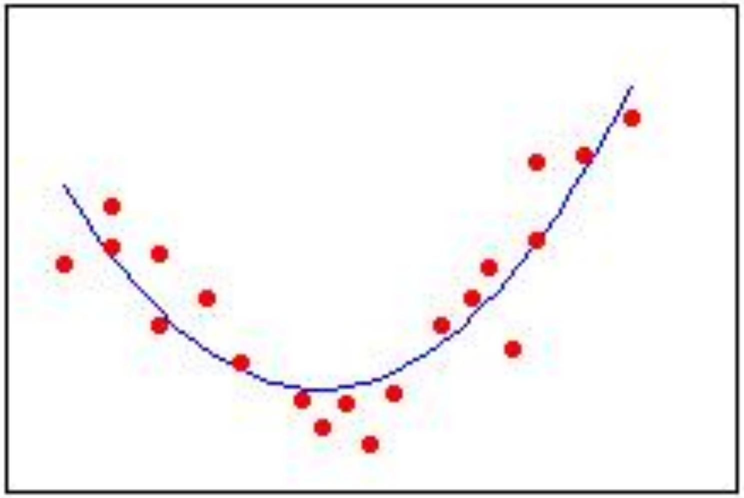
```
fit = lm(y~ xvar1 + xvar2 + as.factor(star)
         +as.factor(release))
summary(fit)
```

Polynomial models

Quadratic

$Y = b_0 + b_1X + b_{11}X^2$

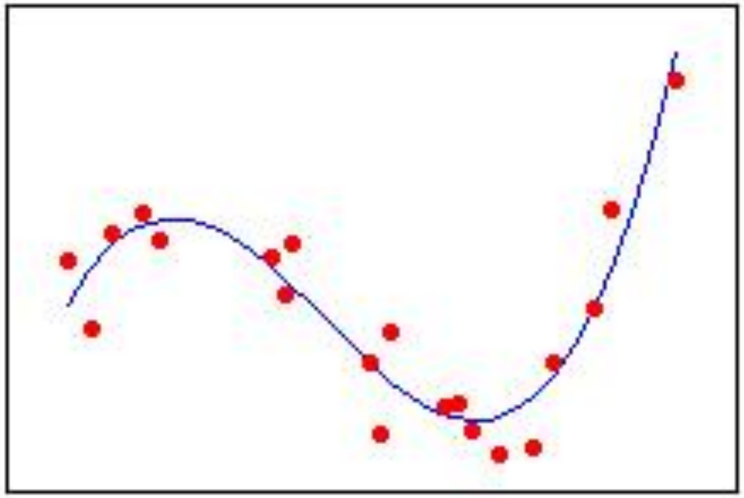
(second order)



Cubic

$Y = b_0 + b_1X + b_{11}X^2 +$
$b_{111}X^3$

(third order)

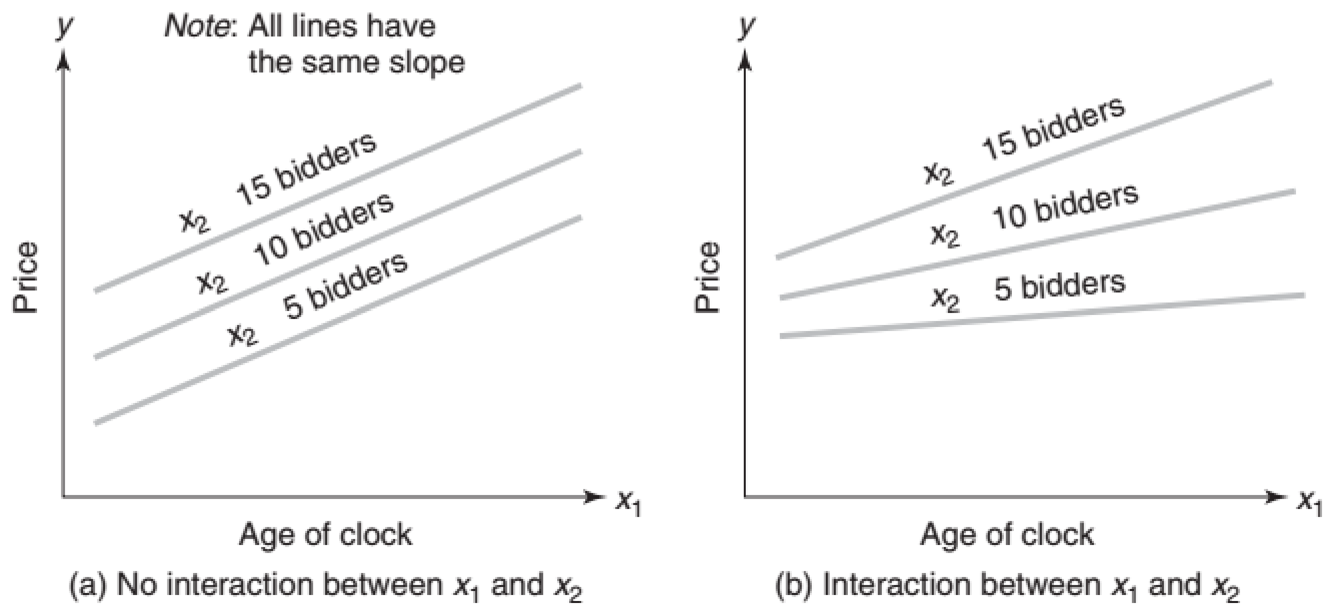


Interaction models

However, if you can observe straight lines with different slopes, like fig b).

It implies that there should be an interaction term x_1x_2 in your model

This is a special case in higher-order regression models.



Interaction models

- Modeling changes in response variable Y with quantitative and qualitative variables

Interaction term

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \boldsymbol{\beta_3 x_1 x_2} + e$$

- Interaction models are useful when associations between Y and X-variables vary with the values of some other variable (slopes are not constant)

- Often used with dummy variables – as association between the response variable Y and a predictor X varies for different levels of the dummy variable

Influential Points

- Influential points are the outliers that affect the fitted model
- Note: not all of the outliers are influential points
- Influential points are observations (typically outliers) that have a strong influence on the fitted model. If removed, the parameter estimates change.

Metrics to Identify Influential Points

Function	Description	Rough Cut-off
dffits()	the change in the fitted values (with appropriately scaled)	$\|DFFITS\|>2\sqrt{(k+1)/n}$
dfbetas()	the changes in the**coefficients**(with appropriately scaled)	> 2/sqrt(n)
covratio()	the change in the estimate of OLS covariance matrix	|covratio-1|≥3*(k+1)/n
hatvalues()	standardized distance to mean of predictors used to measure the leverage of observation	> 2*(k+1)/n
cooks.distance()	standardized distance change for how far the estimate**vector**	> 4/n

k = Number of x variables

n = Number of records to build the model = the size of your data to build the model

Influential points by R

fit = lm(y~x1+x2+x3)

- Print all of the measures and influential points
  - influence.measure (fit); //influential point measures
  - summary (influence.measure (fit)); //print out only influential observations
- Print measures one by one
  - dfbeta (fit)
  - covratio (fit)
  - dffits (fit)
  - cooks.distance (fit)

Multiple Linear Regression

- General Workflow
- Advanced Topics
  - Multicollinearity Problems
  - Dummy Variables (When X is a qualitative variable)
  - Higher-Order Multiple Linear Regressions
  - Interaction Terms
  - Influential Points
- Final Note: Predictions

A confidence interval for predictions

- Suppose we want to predict a **specific response value Y** at a particular value of the X-variables.

- The **predicted value of Y** for values x^*_1, x^*_2, x^*_3 is computed as

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1^* + \hat{\beta}_2 x_2^* + \hat{\beta}_3 x_3^*$$

- **Prediction Interval at 95% confidence level:**

$$\hat{y} \pm t_{0.95,n-2} S.E.(\hat{y})$$

$$S.E.(\hat{Y}) = s_e \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

Additional term that makes standard error of predictions larger

Prediction and estimations in R

# Example of prediction for one data point.
# create **new** data frame containing xvalues for prediction
new = data.frame(linet=c(7), step=c(6), device=c(3))
# use predict() to compute predicted value and standard error
# predict(model_name, new_dataframe, ….)
se.fit=T to compute predicted value
predict(fit, new, se.fit = T)
compute predicted value and prediction interval
predict(fit, new, interval="prediction", level=0.95)

# Example of prediction for many data points.
linet = c(6, 4, 8)
step = c(6, 3, 1)
device=c(3, 2, 1)
new <- data.frame(linet, step, device)

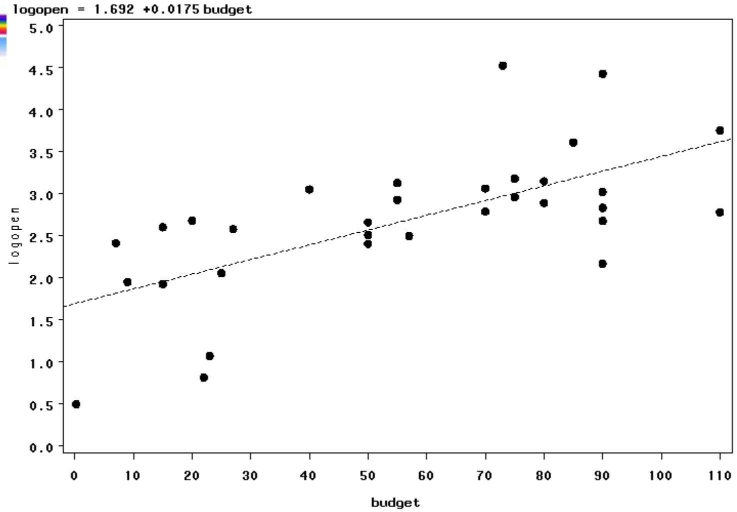
# compute predicted value and standard error
predict(fit, new, se.fit = T)
# compute predicted value and prediction interval
predict(fit, new, se.fit = T, interval="prediction", level=0.95)
# compute average response value and confidence interval
predict(fit, new, se.fit = T, interval="confidence",level=0.95)

Predictions for transformed variables

Data on OPEN = opening revenue for new movies, and BUDGET= cost of the movie. Fitted regression line is

$$\log(open) = 1.692 + 0.0175\,budget$$

Movies with higher budget costs, typically gain more money at their first weekend opening.



Suppose you want to estiamate the average opening revenue for a new movie whose budget was equal to 65 million dollars.

```
The REG Procedure
              Dependent Variable: logopen
   Dep Var     Predicted    Std Error
Obs    logopen        Value      Mean Predict            95% CL Mean
                        2.8314      0.1203        2.5856           3.0771
```

Predictions for Original variables

Thus a movie that costs 65 million dollars can expect to gain on **average**

Average Log(Y)= 2.8314 - with 95% C.I. Equal to (2.5856, 3.0771)

Need to transform the dependent variable back to the original value!

Estimated average opening revenue= exp(2.8314)

=16.969 million dollars.

Apply the **same inverse transformation** to the 95% C.I.to obtain an approximate 95% C.I. for the estimated average response.

Thus, the approximate 95% C.I. for the estimated average gross revenues for movies with a budget cost of 65 million dollars is

(exp(2.5856), exp(3.0771))=(13.27, 21.69) million dollars.

Predictions in Linear Regression

- Important Notes
  - Output: predicted value + confidence interval
  - If you applied transformation on the y variable, the predicted value you produce is the predictions based on the transformed y variable. You should convert it back to the original unit
  - For example, $\log(y) = 6 + 2x_1 + 3x_2$
    To get predicted y values, you should use exp() function to be applied on the predicted log(y)

In-Class Practice

- In-Class Practice
  - N-fold Cross validation
  - Advanced Techniques to improve the models
    - Using categorical/dummy variables
    - Examination of multi-collinearity problems
    - Try higher-order terms or interaction terms
    - Improve models by removing influential points
  - By using Case Study 2

In-Class Practice

- Load Data

```
> mydata=read.table("case2_clerical.txt", header=T, sep='\t')
> head(mydata)
  day hours mail cert  acc change check misc tickets
1   M 128.5 7781  100  886    235   644   56     737
2   T 113.6 7004  110  962    388   589   57    1029
3   W 146.6 7267   61 1342    398  1081   59     830
4  Th 124.3 2129  102 1153    457   891   57    1468
5   F 100.4 4878   45  803    577   537   49     335
6   S 119.2 3999  144 1127    345   563   64     918
```

In-Class Practice

- Categorical variable “day”
- In linear regression, you need to convert it to binary variables. Or, simply use as.factor() function
- Notes: you may need to merge some categories if there are too many values/categories in the nominal variable
- In our case, we simply convert it to weekend and weekday. Or, as 1 or 0

In-Class Practice

```
> library(plyr)
> mydata$day=revalue(mydata$day, c("S"="Weekend"))
> mydata$day=revalue(mydata$day, c("M"="Weekday"))
> mydata$day=revalue(mydata$day, c("T"="Weekday"))
> mydata$day=revalue(mydata$day, c("W"="Weekday"))
> mydata$day=revalue(mydata$day, c("Th"="Weekday"))
> mydata$day=revalue(mydata$day, c("F"="Weekday"))
> head(mydata)
      day hours mail cert  acc change check misc tickets
1 Weekday 128.5 7781  100  886    235   644   56     737
2 Weekday 113.6 7004  110  962    388   589   57    1029
3 Weekday 146.6 7267   61 1342    398  1081   59     830
4 Weekday 124.3 2129  102 1153    457   891   57    1468
5 Weekday 100.4 4878   45  803    577   537   49     335
6 Weekend 119.2 3999  144 1127    345   563   64     918
```

In-Class Practice

- The data is small, we decide to use 5-fold cross validation.
- Workflow
  - Use the whole piece of the data or a sample of the data to build models /with feature selections
  - Validate models by using F-test and residual analysis
  - Evaluate models based on N-folds cross validation
  - Note: use glm() to build models, cv.glm() for evaluations

In-Class Practice

- The data is small, we decide to use 5-fold cross validation.
- Workflow
  – Use the whole piece of the data or a sample of the data to build models /with feature selections
  – Validate models by using F-test and residual analysis
  – Evaluate models based on N-folds cross validation
  – Note: use glm() to build models, cv.glm() for evaluations

In-Class Practice

- Examine linear relationship between y and x

```
> day=mydata$day
> hours=mydata$hours
> mail=mydata$mail
> cert=mydata$cert
> acc=mydata$acc
> change=mydata$change
> check=mydata$check
> misc=mydata$misc
> tickets=mydata$tickets
> fs=cbind(hours,mail,cert,acc, change, check, misc, tickets)
> cor(fs)
              hours         mail        cert        acc      change       check        misc    tickets
hours    1.000000000 -0.007650103  0.29281923 0.46151908  0.08479822  0.58731901  0.49901266  0.4495941
mail    -0.007650103  1.000000000  0.01128202 0.05480359 -0.04311752 -0.27658574 -0.01594041 -0.3117669
cert     0.292819235  0.011282017  1.00000000 0.24521511  0.03686148 -0.01588972  0.33892441  0.1222646
acc      0.461519082  0.054803588  0.24521511 1.00000000  0.47780716  0.50899367  0.34892016  0.5087885
change   0.084798217 -0.043117518  0.03686148 0.47780716  1.00000000  0.44280516  0.16735176  0.2750750
check    0.587319010 -0.276585736 -0.01588972 0.50899367  0.44280516  1.00000000  0.38227195  0.5660733
misc     0.499012658 -0.015940412  0.33892441 0.34892016  0.16735176  0.38227195  1.00000000  0.2971547
tickets  0.449594128 -0.311766861  0.12226462 0.50878854  0.27507497  0.56607326  0.29715473  1.0000000
```

- Note that we exclude the binary variable, since it is difficult to interpret the correlations

In-Class Practice

- We tried transformations
- We decide to ignore variable ‘mail’
- And use new variable 1/change

```
> change2=1/change
> mydata[,"change2"]=change2
> head(mydata)
      day hours mail cert  acc change check misc tickets     change2
1 Weekday 128.5 7781  100  886    235   644   56     737 0.004255319
2 Weekday 113.6 7004  110  962    388   589   57    1029 0.002577320
3 Weekday 146.6 7267   61 1342    398  1081   59     830 0.002512563
4 Weekday 124.3 2129  102 1153    457   891   57    1468 0.002188184
5 Weekday 100.4 4878   45  803    577   537   49     335 0.001733102
6 Weekend 119.2 3999  144 1127    345   563   64     918 0.002898551
```

In-Class Practice

- Next, build models by using feature selection
- For demo purpose, we just try the stepwise method in the class. In real practice, you should try multiple feature selection methods

In-Class Practice

```
> full=glm(hours~cert+acc+change2+check+misc+tickets+as.factor(day))
> base=glm(hours~check)
> full2=glm(hours~cert+acc+change2+check+misc+tickets)
> step(Base, scope=list(upper=Full, lower=~1), direction="forward", trace=F)

> step(base, scope=list(upper=full2, lower=~1), direction="both", trace=F)

Call:  glm(formula = hours ~ check + cert + misc + change2)

Coefficients:
(Intercept)        check         cert         misc      change2
  5.120e+01    5.203e-02    1.159e-01    2.695e-01    1.574e+03

Degrees of Freedom: 51 Total (i.e. Null);  47 Residual
Null Deviance:      12310
Residual Deviance: 5976        AIC: 406.3
> step(base, scope=list(upper=full, lower=~1), direction="both", trace=F)

Call:  glm(formula = hours ~ check + cert + misc + change2)

Coefficients:
(Intercept)        check         cert         misc      change2
  5.120e+01    5.203e-02    1.159e-01    2.695e-01    1.574e+03

Degrees of Freedom: 51 Total (i.e. Null);  47 Residual
Null Deviance:      12310
Residual Deviance: 5976        AIC: 406.3
> m1=glm(hours~check+cert+misc+change2)
```

With binary variable

Without binary variable

Same model

In-Class Practice

- We are going to compare the following models
  - full → the model using all variables
  - full2 → the models using numerical variables only
  - base → only use one numerical variable
  - m1 → the model by using stepwise feature selection
- Currently you build these models, next you should examine whether they are qualified or not

In-Class Practice

- First of all, we'd like to examine multi-collinearity problems

install.packages("car",dependencies=TRUE)
library(car)

Install all necessary dependent libraries for the package "car"

```
> vif(ml)
   check     cert     misc  change2 
1.571975 1.211148 1.363520 1.445765 
> vif(base)
Error in vif.default(base) : model contains fewer than 2 terms
> vif(full)
           cert             acc         change2           check            misc         tickets as.factor(day) 
       1.311209        2.183786        1.521196        2.270196        1.397397        1.687087        1.486069 
> vif(full2)
    cert      acc  change2    check     misc  tickets 
1.272675 1.663163 1.489835 2.111563 1.375282 1.640427 
```

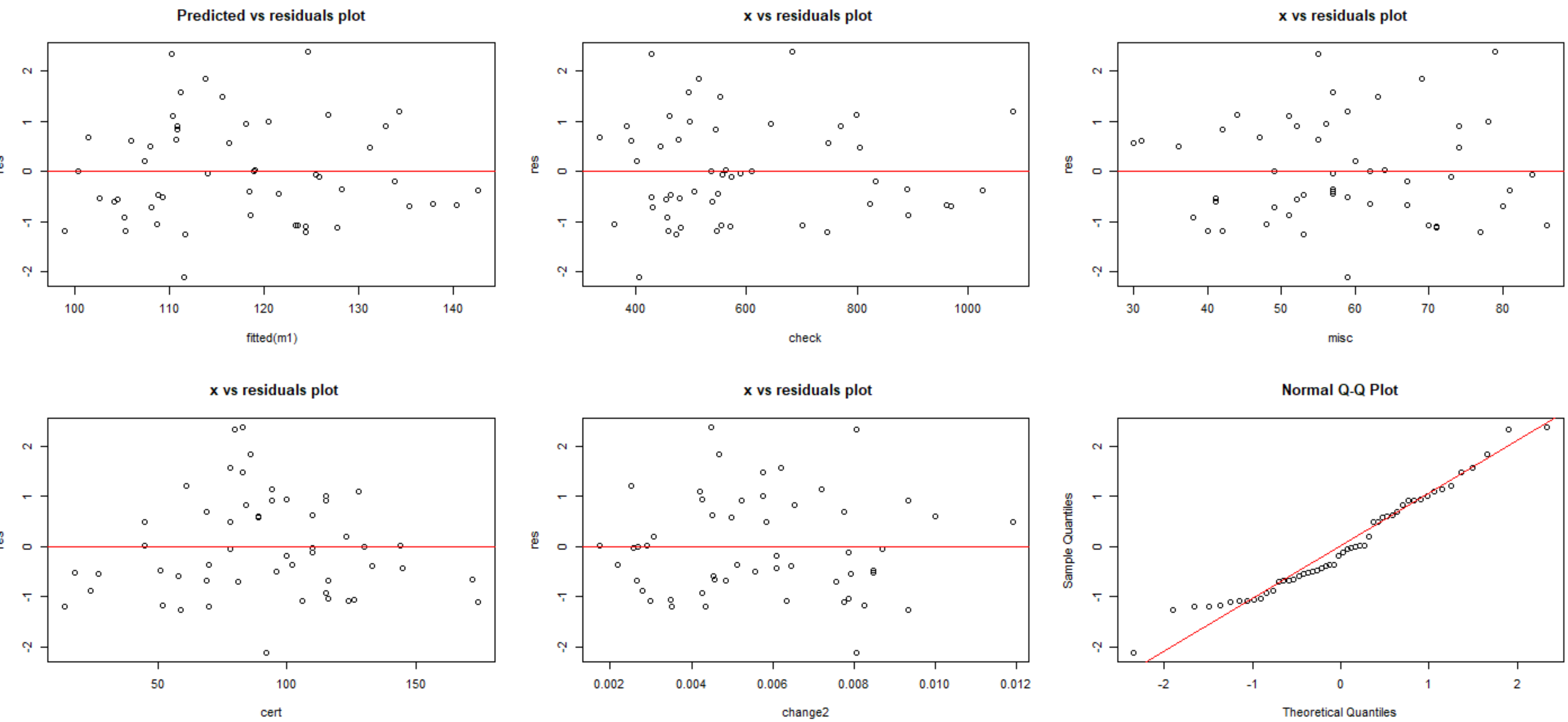
In-Class Practice

Residual analysis, take model m1 for example

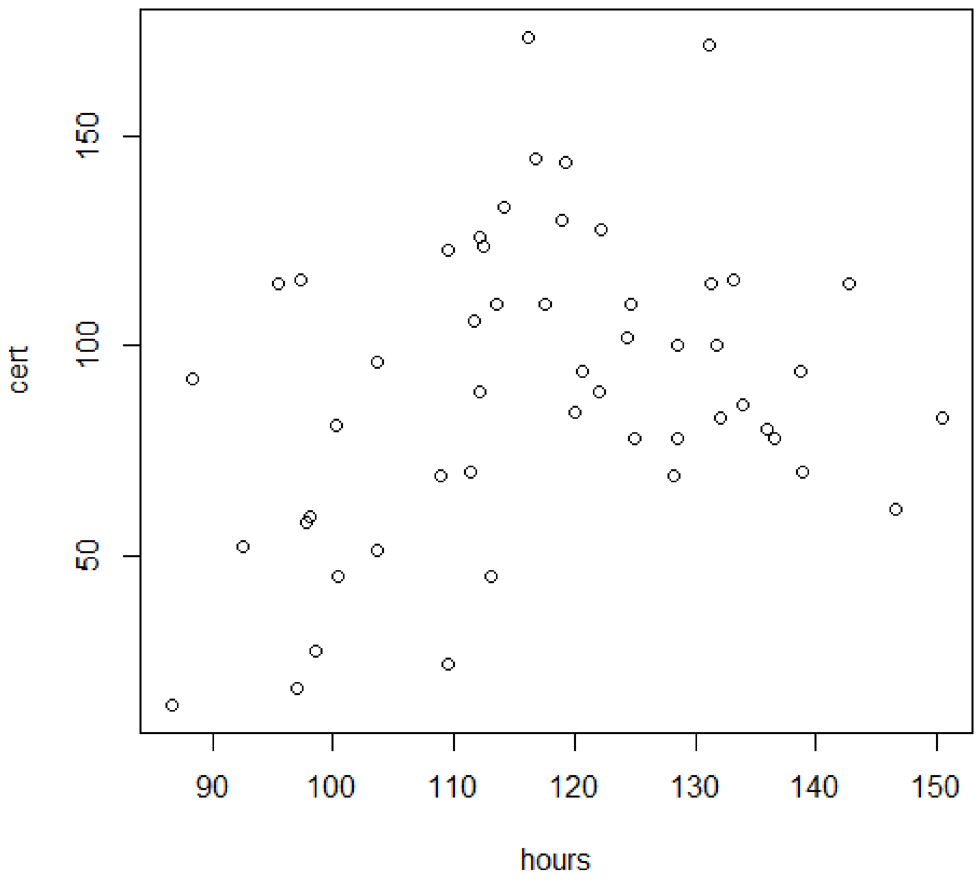
```
res=rstandard(m1)
attach(mtcars)
par(mfrow=c(2,3))
plot( fitted(m1), res, main="Predicted vs
residuals plot")
abline(a=0, b=0, col='red')
plot(check, res, main=" x vs residuals plot")
abline(a=0, b=0,col='red')
plot(misc, res, main=" x vs residuals plot")
abline(a=0, b=0,col='red')
plot(cert, res, main=" x vs residuals plot")
abline(a=0, b=0,col='red')
plot(change2, res, main=" x vs residuals plot")
abline(a=0, b=0,col='red')
qqnorm(res)
qqline(res,col=2)
```

It is used to produce plots in matrix,
Put them in 2 rows and 3 columns

In-Class Practice



In-Class Practice



In-Class Practice

```
> cert2=cert*cert
> mydata[,"cert2"]=cert2
> m2=glm(hours~check+misc+change2+cert+cert2)
> summary(m2)

Call:
glm(formula = hours ~ check + misc + change2 + cert + cert2)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-25.8116   -7.7111    0.5933    5.9529   23.9056

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.768e+01  1.221e+01   3.086  0.00343 **
check        5.251e-02  1.004e-02   5.232 4.02e-06 ***
misc         2.227e-01  1.257e-01   1.771  0.08316 .
change2      1.523e+03  7.631e+02   1.996  0.05191 .
cert         5.319e-01  1.632e-01   3.259  0.00211 **
cert2       -2.268e-03  8.539e-04  -2.656  0.01084 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 112.642)

    Null deviance: 12312.5  on 51  degrees of freedom
Residual deviance:  5181.5  on 46  degrees of freedom
AIC: 400.85

Number of Fisher Scoring iterations: 2
```

Add 2^{nd} order term into model

I also tried stepwise
Both cert and cert2 are included in the selected model

In-Class Practice

```
> summary(ml)

Call:
glm(formula = hours ~ check + cert + misc + change2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-23.228   -7.366   -1.634    7.882   25.695

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.120e+01  1.180e+01   4.341 7.50e-05 ***
check       5.203e-02  1.066e-02   4.881 1.26e-05 ***
cert        1.159e-01  4.876e-02   2.378   0.0215 *
misc        2.695e-01  1.323e-01   2.038   0.0472 *
change2     1.574e+03  8.104e+02   1.942   0.0581 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 127.1483)

    Null deviance: 12313  on 51  degrees of freedom
Residual deviance:  5976  on 47  degrees of freedom
AIC: 406.27

Number of Fisher Scoring iterations: 2
```

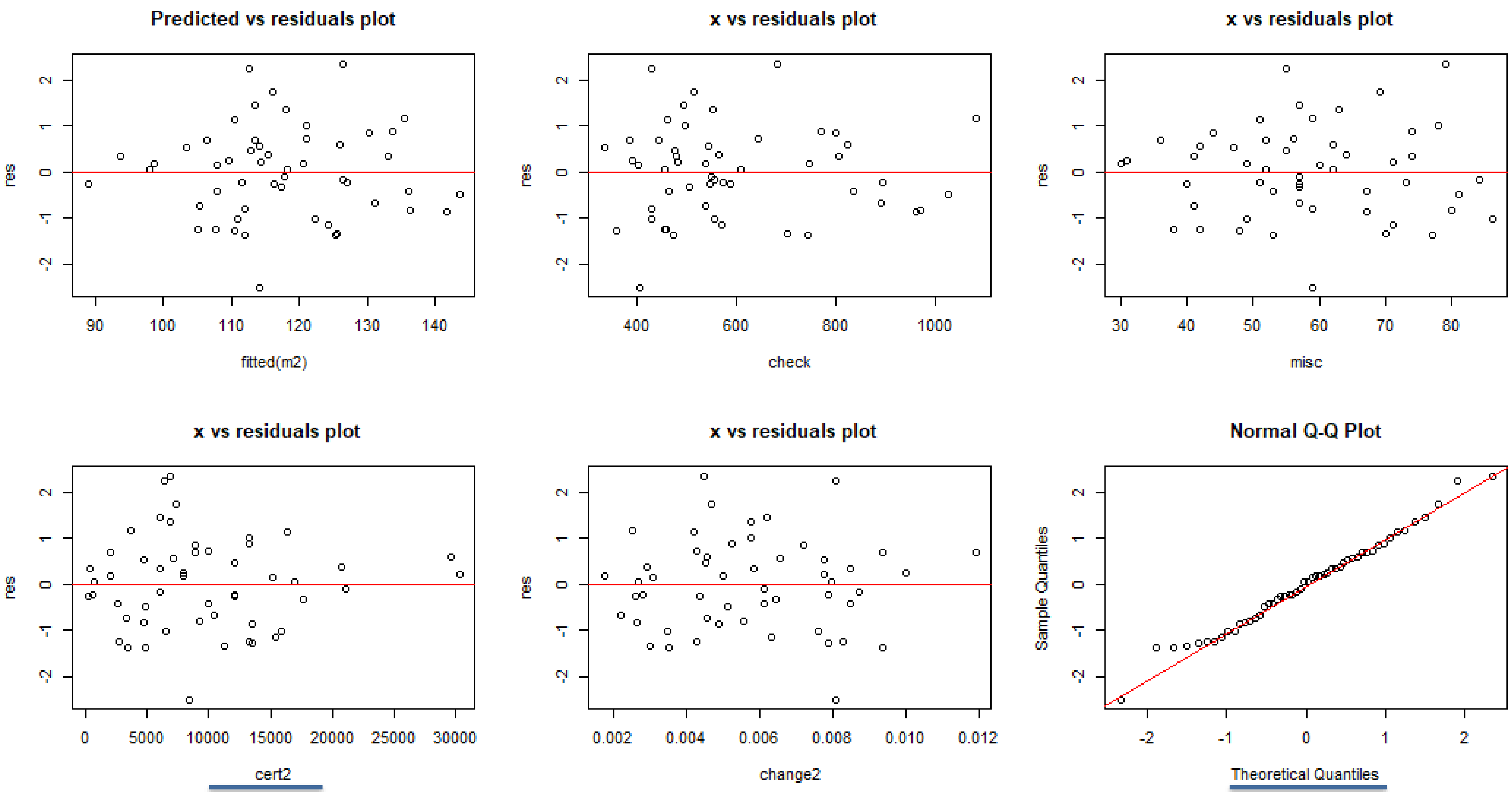
Note that we do not have adjR2 and F-test results

It is because we use glm to build the linear regression models

F-test ➔ as long as one x variable has small p-value in t-test, it is satisfied

AdjR2 ➔ you need to use lm() function to build models, if you need adjR2

In-Class Practice



In-Class Practice

- We are going to compare the following models
  - full → the model using all variables
  - full2 → the models using numerical variables only
  - base → only use one numerical variable
  - m1 → the model by using stepwise feature selection
  - m2→ add cert2 (2nd order term) into m1

In-Class Practice

```
> library(boot)
> mse_full = cv.glm(mydata, full, K=5)$delta
> mse_full2 = cv.glm(mydata, full2, K=5)$delta
> mse_base = cv.glm(mydata, base, K=5)$delta
> mse_m1 = cv.glm(mydata, m1, K=5)$delta
> mse_m2 = cv.glm(mydata, m2, K=5)$delta
> errs = cbind(mse_full, mse_full2, mse_base, mse_m1, mse_m2)
> errs
     mse_full mse_full2 mse_base   mse_m1   mse_m2
[1,] 161.8878  137.4882 176.3617 137.1369 110.8290
[2,] 155.6756  134.2253 173.9649 134.3850 109.4892
```

→ Raw MSE

→ Adjusted MSE

In-Class Practice

- Model m2 seems to be the best
- Are there any other ways to further improve m2?

In-Class Practice

- Model m2 seems to be the best
- Are there any other ways to further improve m2?
  - You can try interaction terms → a binary variable and a numerical variable. If you believe there could be an effect based on the binary variable
  - You can also identify influential points

In-Class Practice

- Influential points

```
> library(stats)
> influence.measures(m2)
Influence measures of
	 glm(formula = hours ~ check + misc + change2 + cert + cert2) :

     dfb.1_  dfb.chck  dfb.misc  dfb.chn2  dfb.cert dfb.crt2   dffit cov.r   cook.d    hat inf
1   0.013258  0.01191 -0.055265 -0.058569  0.080161 -0.07521  0.1504 1.111 3.81e-03 0.0416
2  -0.032913  0.02408  0.017360  0.058907 -0.024432  0.02219 -0.0768 1.227 1.00e-03 0.0788
3  -0.064652  0.40719 -0.149526 -0.046216  0.021182 -0.04799  0.5505 1.165 5.01e-02 0.1814
4   0.010202 -0.12740  0.109853  0.081796 -0.087248  0.07622 -0.2431 1.209 9.96e-03 0.1128
5   0.067894 -0.03808 -0.004535 -0.066404 -0.021653  0.00999  0.0801 1.396 1.09e-03 0.1856   *
6   0.050495 -0.03791  0.001548 -0.064428 -0.029216  0.04834  0.1271 1.255 2.74e-03 0.1072
7   0.031002 -0.04079  0.003455 -0.041444  0.005488 -0.00335  0.0571 1.292 5.56e-04 0.1200
8   0.046439 -0.12447  0.038678 -0.037277  0.092836 -0.12561  0.2844 0.899 1.32e-02 0.0368
9   0.004254  0.17939 -0.073994  0.077906 -0.224348  0.31550  0.4539 1.705 3.48e-02 0.3623   *
10 -0.011418  0.20952 -0.308514  0.148133  0.040972 -0.04078 -0.4114 1.153 2.82e-02 0.1385
11 -0.222430  0.09968  0.328452  0.215669 -0.167484  0.12128 -0.4715 1.061 3.66e-02 0.1244
12  0.045056 -0.00414 -0.037527 -0.042035 -0.014107  0.01125 -0.0702 1.227 8.38e-04 0.0777
13 -0.050050  0.12981 -0.032304  0.070544 -0.069512  0.07762 -0.1906 1.110 6.11e-03 0.0546
14 -0.015824  0.02160 -0.023581 -0.043492  0.030335 -0.01369 -0.1124 1.195 2.14e-03 0.0672
15  0.195408 -0.28112  0.057046 -0.117843 -0.076822  0.04203 -0.3279 1.191 1.80e-02 0.1295
16  0.012575  0.04369 -0.278113  0.146413  0.015635  0.05543 -0.4415 0.971 3.18e-02 0.0904
17 -0.152722  0.04473  0.109187  0.114750 -0.006917  0.02975 -0.2158 1.152 7.84e-03 0.0787
18 -0.007463 -0.08864  0.094557 -0.034108  0.089218 -0.11876  0.2624 0.928 1.13e-02 0.0357
19 -0.054451 -0.17582  0.063881  0.188488  0.177323 -0.21161  0.5523 0.601 4.63e-02 0.0523   *
20 -0.054135 -0.15007  0.230070 -0.013189  0.022282 -0.02332  0.3048 1.092 1.55e-02 0.0848
21 -0.157015 -0.12639  0.547709 -0.121879  0.070976 -0.16413  0.7380 0.577 8.16e-02 0.0815   *
22  0.037477  0.00883 -0.061246 -0.035918  0.000282  0.00666 -0.0766 1.378 9.98e-04 0.1752
23  0.093062 -0.01031  0.000035  0.031330 -0.143914  0.12356  0.1824 1.433 5.65e-03 0.2166   *
24  0.003460  0.00357  0.013886 -0.015279 -0.004445 -0.01205 -0.0766 1.194 9.98e-04 0.0568
25  0.121534  0.26962 -0.168870 -0.220088 -0.236005  0.25238 -0.6732 0.502 6.67e-02 0.0598   *
```

In-Class Practice

- Cook.dist, cut off = 4/n
- We have 52 records in the data
  cut off = 4/52 = 0.0769
- Note
  - You may find many influential points
  - But your data is small, you cannot remove all of them
  - Remove the observations with larger cook.dist
  - For example, we have 52 records, I decide to remove the top-3 observations with largest cook.dist. They have the row index: 9, 41, 52

In-Class Practice

```
> # create new data by removing influential points
> newdata=mydata[-c(9,41,52),]
> hours=newdata$hours
> check=newdata$check
> misc=newdata$misc
> cert=newdata$cert
> cert2=newdata$cert2
> change2=newdata$change2
> # build models based on the newdata
> m3 = glm(hours~check+misc+change2+cert+cert2)
> # N-fold cross validation
> mse_m3 = cv.glm(newdata, m3, K=5)$delta
> mse_m3
[1] 114.9111 113.6145
```

Error was increased in comparison with m2
It is because you may identify different influential points
by using different criterion.
Here we use cook.dist which may not find good influential points

In-Class Practice

- Next, you can practice by yourself
  - Retry my codes by using Case Study 2
  - Build your own models by using data in Case Study 1
    - You should use all the useful variables, including nominal variables
    - You should try the advanced techniques to improve the models