431 Class 25

Thomas E. Love

2017-11-30

Today's Agenda

- Stepwise Regression with the `nyfs2` data
- Ginzberg's Depression Data from the `car` package
    - Should we be modeling a transformed outcome?
    - Comparing Models with R^2, adjusted R^2, AIC, BIC
    - Comparing Model Predictions Out of Sample
        - Partitioning the Data Set
        - Building the Model in a Training Sample
        - Using a Test Sample, MAPE and MSPE for Validation
- Getting Better Calibrated on Residual Plots

Today's R Setup and Data Set

```r
library(car); library(magrittr)
library(broom); library(tidyverse)

nyfs2 <- read.csv("data/nyfs2.csv") %>% tbl_df

ginz0 <- tbl_df(car::Ginzberg)
ginz0$id <- 1:82
ginz <- select(ginz0, id, fatalism, simplicity, depression)

source("Love-boost.R")
```

Stepwise Regression and the `nyfs2` data

The Kitchen Sink Model (m4)

```r
m4 <- lm(bmi ~ calf.circ + age.exam + race.eth + sex,
         data = nyfs2)

glance(m4)
```

```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8300404     0.8291954 1.686596  982.3316       0
  df    logLik      AIC      BIC deviance df.residual
1  8 -2745.366 5508.732 5556.033 4005.205        1408
```

Stepwise (Backwards Elimination) Variable Selection

The rest of the output follows on the next slide...

```
step(m4)
```

```
Start:  AIC=1488.3
bmi ~ calf.circ + age.exam + race.eth + sex

            Df Sum of Sq     RSS    AIC
- sex        1       0.2  4005.4 1486.4
<none>                    4005.2 1488.3
- race.eth   4     191.6  4196.8 1546.5
- age.exam   1    2408.5  6413.7 2153.0
- calf.circ  1   13086.9 17092.1 3540.9

Step:  AIC=1486.36
bmi ~ calf.circ + age.exam + race.eth
```

Second Part of the `step(m4)` output

```
Step:  AIC=1486.36
bmi ~ calf.circ + age.exam + race.eth

            Df Sum of Sq     RSS    AIC
<none>                    4005.4 1486.4
- race.eth   4     191.8  4197.2 1544.6
- age.exam   1    2410.3  6415.6 2151.4
- calf.circ  1   13098.2 17103.6 3539.9

Call:
lm(formula = bmi ~ calf.circ + age.exam + race.eth,
     data = nyfs2)
```

Model `m5`: Leaving out `sex` from `m4`

```r
m4 <- lm(bmi ~ calf.circ + age.exam + race.eth + sex,
         data = nyfs2)

m5 <- lm(bmi ~ calf.circ + age.exam + race.eth,
         data = nyfs2)

select(glance(m4), r.squared, adj.r.squared, AIC)
```

```
  r.squared adj.r.squared      AIC
1 0.8300404     0.8291954 5508.732
```

```r
select(glance(m5), r.squared, adj.r.squared, AIC)
```

```
  r.squared adj.r.squared      AIC
1 0.8300336     0.8293098 5506.789
```

Model m5 coefficients (output edited)

```
Coefficients:                     Estimate    SE      t        p
(Intercept)                         -3.996 0.288 -13.87 < 2e-16
calf.circ                            0.972 0.014  67.88 < 2e-16
age.exam                            -0.625 0.021 -29.12 < 2e-16
race.eth2 Non-Hispanic Black        -0.027 0.119  -0.23   0.822
race.eth3 Mexican American           0.997 0.136   7.31 4.3e-13
race.eth4 Other Hispanic             0.375 0.135   2.78   0.006
race.eth5 Other or Multi-Race       -0.132 0.172  -0.77   0.443
```

anova(m5)

```
Analysis of Variance Table

Response: bmi
            Df  Sum Sq Mean Sq  F value    Pr(>F)
calf.circ    1 16917.2 16917.2 5951.115 < 2.2e-16 ***
age.exam     1  2451.2  2451.2  862.282 < 2.2e-16 ***
race.eth     4   191.8    48.0   16.869 1.618e-13 ***
Residuals 1409  4005.4     2.8
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Signs of Meaningful Collinearity in model m5?

```r
vif(m5)
```

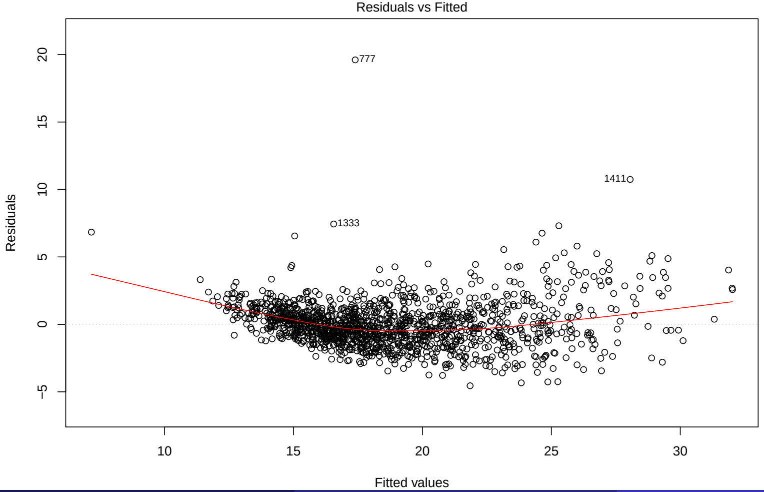
```
              GVIF Df GVIF^(1/(2*Df))
calf.circ 3.114603  1        1.764824
age.exam  3.105121  1        1.762135
race.eth  1.013989  4        1.001738
```

Note the use of a generalized variance inflation factor here. This will be used if any of the regression inputs are associated with more than one degree of freedom, usually because of indicator variables representing a multi-categorical variable.

As none of these values exceed 5 (let alone 10), again, we don't have any serious concerns.

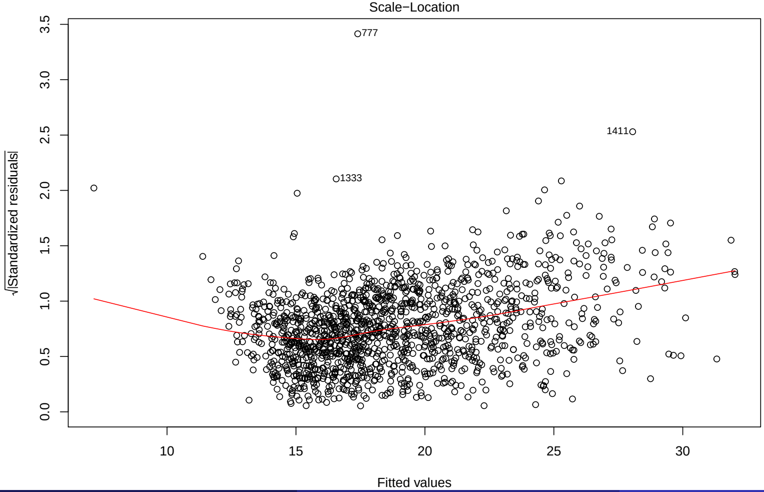
Check Assumptions via Residuals for `m5`

```
plot(m5, which = 1)
```



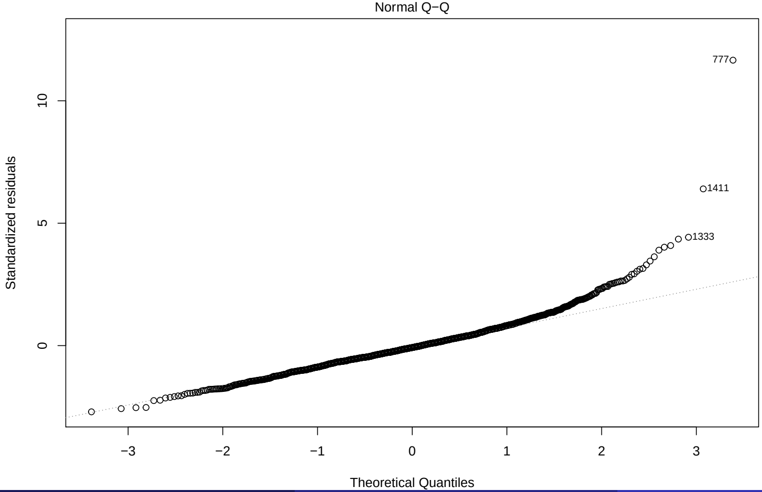
Problems with the "constant variance" assumption?

```r
plot(m5, which = 3)
```



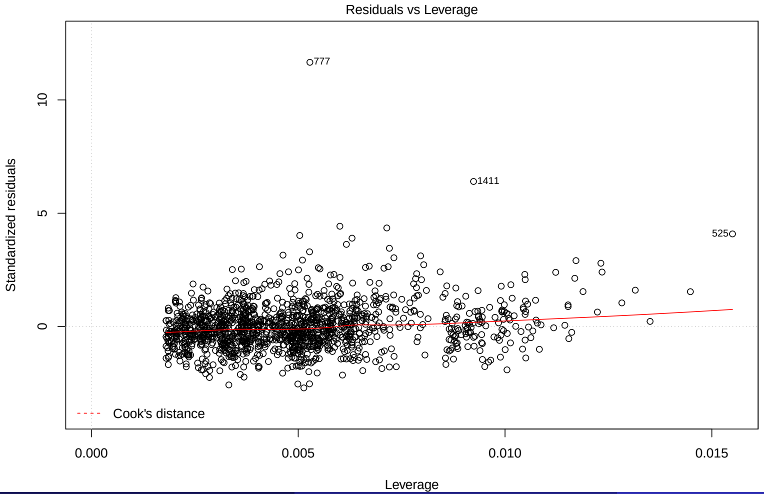
Are the m5 residuals Normally distributed?

```
plot(m5, which = 2)
```



Any influential points in m5?

```
plot(m5, which = 5)
```



Conclusions from m5?

Model m5 includes three inputs to predict bmi:

```
bmi = - 4.00 + 0.97 calf.circ - 0.62 age.exam
      - 0.03 if race.eth = Non-Hispanic Black
      + 1.00 if race.eth = Mexican American
      + 0.37 if race.eth = Other Hispanic
      - 0.13 if race.eth = Other or Multi-Race
```

- $R^2 = 0.83$ (adjusted $R^2 = 0.829$)
- global F test is highly significant ($p < 0.0001$)
- still some issues with our residual plots
- no signs of meaningful collinearity

Ginzberg's Depression Data

Ginzberg's Depression Data

The Ginzberg data are part of the car package. The data describe psychiatric patients hospitalized for depression. We'll look at three variables, each of which is scaled to have mean 1 and standard deviation 0.5 in this sample…

- our outcome, fatalism, which measures the subject's fatalism (the belief that all events are inevitable)
- simplicity, which measures the need to see the world in black and white
- depression, which is the Beck self-report depression scale

Subjects with values exceeding 1 on these measures are reporting greater than average fatalism, simplicity or depression, respectively.

Standardized key variables in the ginz tibble

Remember that the values for each variable have been standardized to mean 1 and standard deviation 0.5

```r
summary(select(ginz, -id))
```

```
    fatalism           simplicity         depression    
 Min.   :-0.05837   Min.   :0.2507   Min.   :0.4695  
 1st Qu.: 0.56301   1st Qu.:0.6563   1st Qu.:0.5664  
 Median : 0.97727   Median :0.8827   Median :0.8247  
 Mean   : 1.00000   Mean   :1.0000   Mean   :1.0000  
 3rd Qu.: 1.39152   3rd Qu.:1.2694   3rd Qu.:1.3737  
 Max.   : 2.22003   Max.   :2.8541   Max.   :2.2456  
```

- What does a value of zero mean on these scales?
- A change of one unit on these scales is how large?

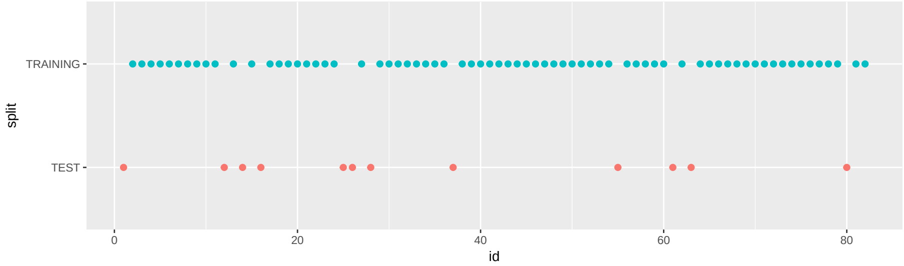
Partitioning into Training and Test Samples

We'll build a training sample (ginz.train) for building models with 70 patients, and hold back a test sample (ginz.test) of the remaining 12 patients for evaluating the model after it's been built.

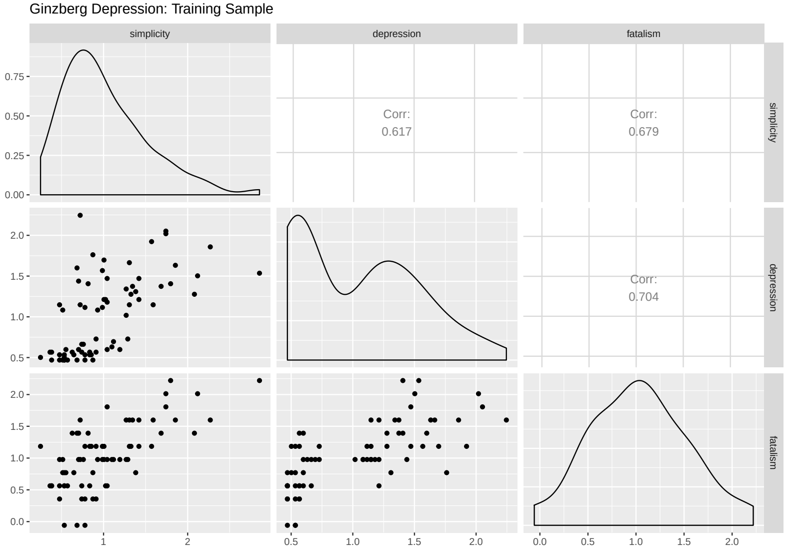
```
set.seed(43111)
ginz.train <- sample_n(ginz, 70, replace = FALSE)
ginz.test <- anti_join(ginz, ginz.train, by = "id")
```

Showing the Partition

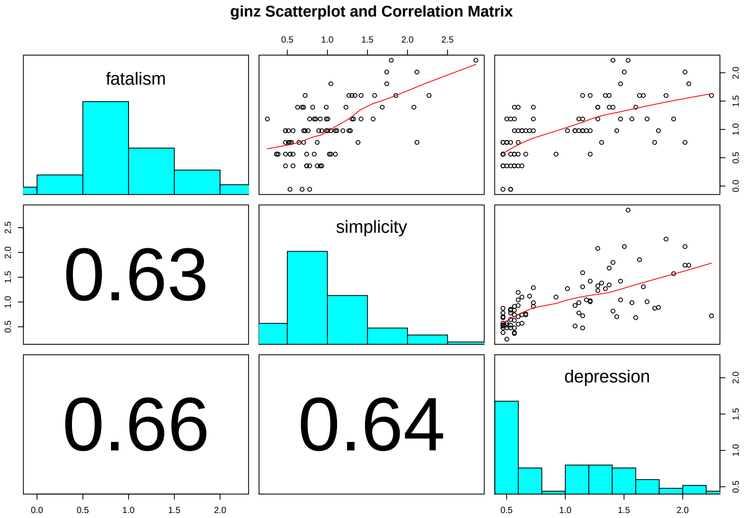
```
ginz$split <- ifelse(ginz$id %in% ginz.train$id,
                     "TRAINING", "TEST")

ggplot(ginz, aes(x = id, y = split, col = split)) +
  geom_point(cex = 2) + guides(col = FALSE)
```



Scatterplot Matrix for Ginzberg's Depression Data



Alternate Scatterplot Matrix for Ginzberg's data



Does Box-Cox suggest a transformation?

```r
m1 <- lm(fatalism ~ simplicity + depression,
         data = ginz.train)
boxCox(m1)
```

This throws an error message:

```
Error in bc1(out, lambda) :
  First argument must be strictly positive.
```

Oops, we have some non-positive values of our outcome

```
summary(ginz.train$fatalism)
```

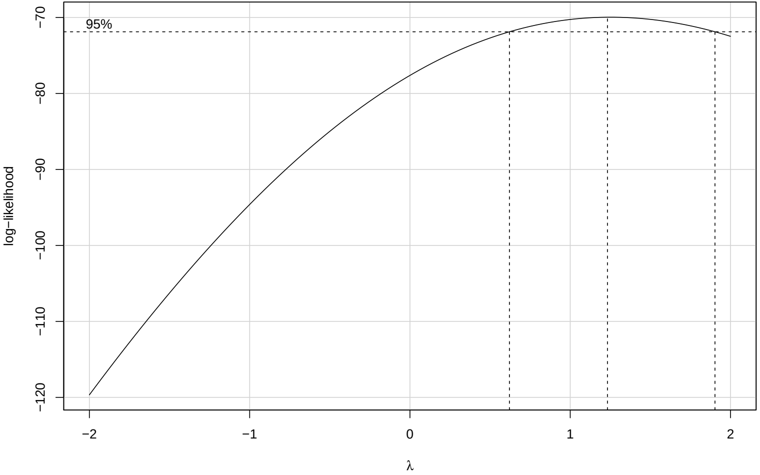
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
-0.05837  0.56301  0.97727  1.03349  1.39152  2.22003 
```

Could just add 1 to every value for Box-Cox check…

```
ginz.train$fat <- ginz.train$fatalism + 1

m1a <- lm(fat ~ simplicity + depression, data = ginz.train)
```

```
boxCox(m1a)
```

Here's the new plot (on 1 + fatalism)



And, if we need backup for our eyes...

```
powerTransform(m1a)
```

```
Estimated transformation parameters
      Y1
1.251195
```

Take advantage of the `roundlam` object contained within `powerTransform`.

```
powerTransform(m1a)$roundlam
```

```
Y1
 1
```

Regression Model with Simplicity and Depression

```r
m1 <- lm(fatalism ~ simplicity + depression,
         data = ginz.train)

arm::display(m1)
```

```
lm(formula = fatalism ~ simplicity + depression, data = ginz.t
            coef.est coef.se
(Intercept) 0.14     0.10
simplicity  0.40     0.10
depression  0.48     0.10
---
n = 70, k = 3
residual sd = 0.33, R-Squared = 0.59
```

```r
summary(m1) # edited output on next page
```

Complete `m1` output, edited lightly

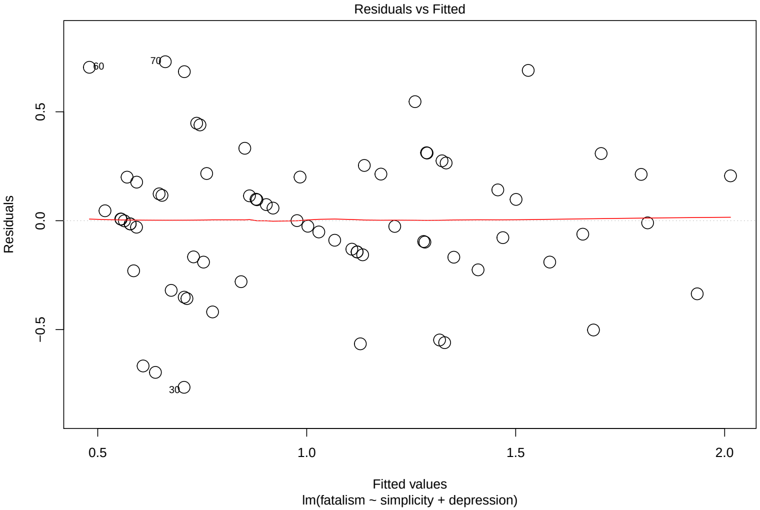
```
lm(fatalism ~ simplicity + depression, data = ginz.train)

Multiple R-squared: 0.593, Adjusted R-squared: 0.580
F-statistic: 48.71 on 2 and 67 DF,  p-value: 8.699e-14

Coefficients: Estimate      SE     t         p
(Intercept)      0.140  0.099  1.42     0.161
simplicity       0.400  0.100  3.98     0.0002 ***
depression       0.477  0.103  4.64   1.64e-05 ***

Residuals:  Min    Q1    Med    Q3   Max     SE
          -0.76 -0.17 -0.005  0.20  0.73   0.33
```

Is collinearity a big issue here?

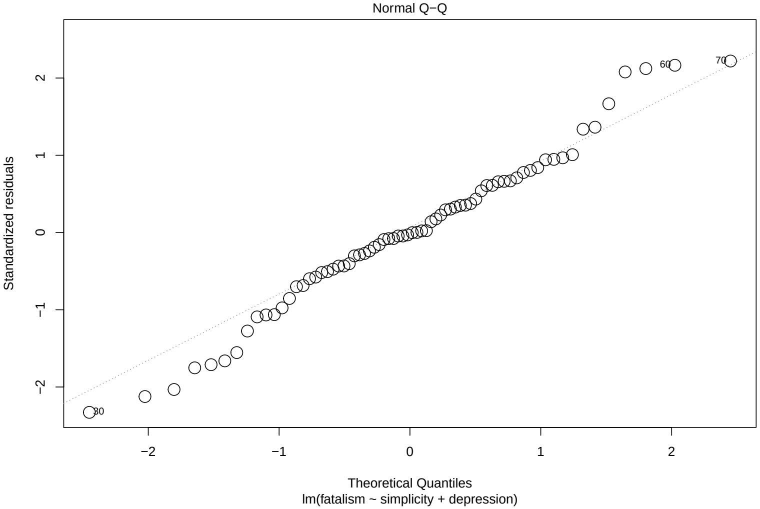
```
vif(m1)
```

```
simplicity depression 
  1.616002   1.616002 
```

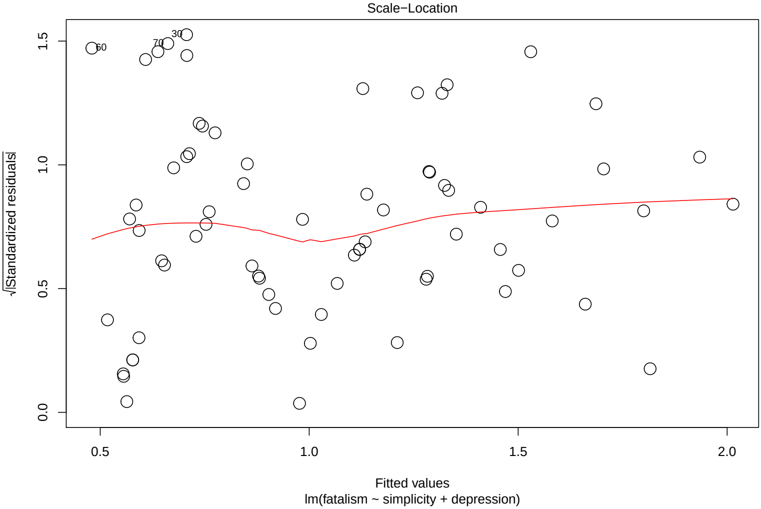
Residuals vs. Fitted Values



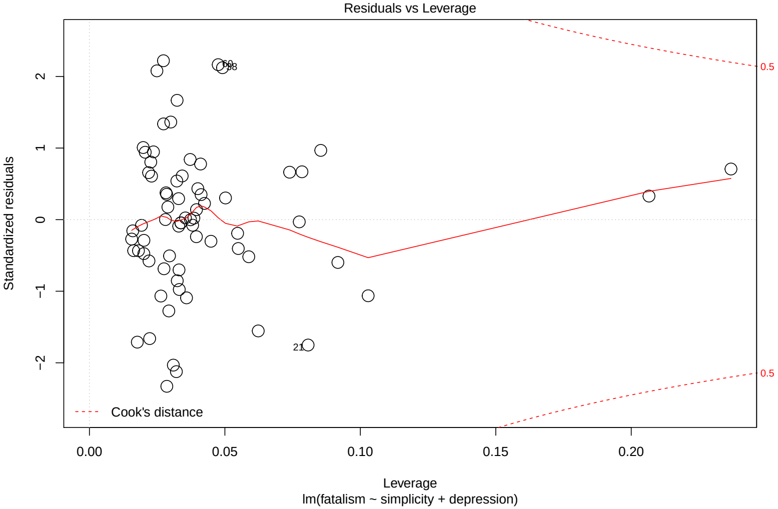
Residuals in a Normal Q-Q Plot



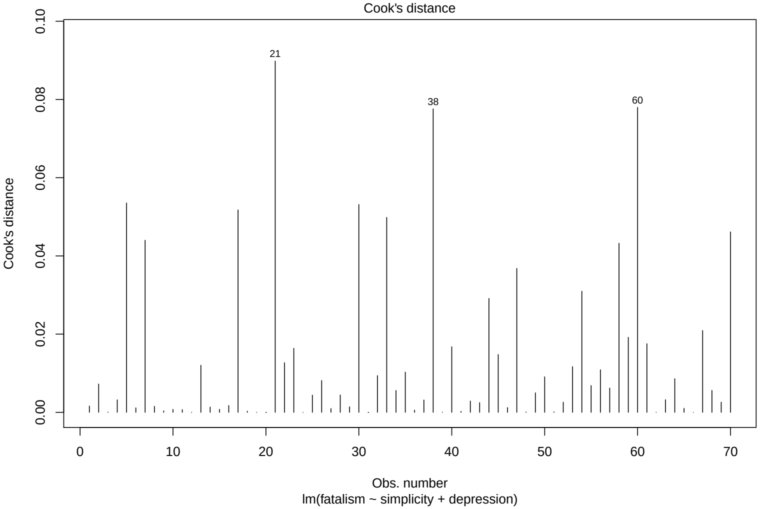
Scale-Location Plot



Plot 5: Residuals, Leverage, Influence?



Plot 4: Index plot of Cook's distance



Consider a second model

- Model `m1` included both depression *and* simplicity.
- Let's fit Model `m2` which only includes depression.

```
m2 <- lm(fatalism ~ depression, data = ginz.train)

arm::display(m2)
```

```
lm(formula = fatalism ~ depression, data = ginz.train)
            coef.est coef.se
(Intercept) 0.29     0.10
depression  0.73     0.09
---
n = 70, k = 2
residual sd = 0.37, R-Squared = 0.50
```

Model `m2` summary

```
Call:
lm(formula = fatalism ~ depression, data = ginz.train)

Multiple R-squared:  0.496, Adjusted R-squared:  0.489
F-statistic: 66.92 on 1 and 68 DF,  p-value: 1.031e-11

Coefficients: Estimate    SE     t         p
(Intercept)      0.289 0.101 2.86     0.006 **
depression       0.730 0.089 8.18  1.03e-11 ***

Residuals:  Min    Q1   Med    Q3   Max     SE
          -0.80 -0.19 -0.06  0.20  0.90   0.37
```

Hypothesis Test comparing m1 to m2

```r
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: fatalism ~ simplicity + depression
Model 2: fatalism ~ depression
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     67 7.4448                                  
2     68 9.2081 -1   -1.7632 15.868 0.0001699 ***
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Does the order in which we list the models matter?

Hypothesis Test comparing `m2` to `m1`

```r
anova(m2, m1)
```

```
Analysis of Variance Table

Model 1: fatalism ~ depression
Model 2: fatalism ~ simplicity + depression
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     68 9.2081
2     67 7.4448  1    1.7632 15.868 0.0001699 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How does order matter here?

Which Model Looks Best in the training sample?

```
round(glance(m1),3) # depression and simplicity
```

```
  r.squared adj.r.squared sigma statistic p.value df
1     0.592          0.58 0.333    48.708       0  3
   logLik    AIC    BIC deviance df.residual
1 -20.892 49.783 58.777    7.445          67
```

```
round(glance(m2),3) # depression alone
```

```
  r.squared adj.r.squared sigma statistic p.value df
1     0.496         0.489 0.368    66.916       0  2
   logLik    AIC    BIC deviance df.residual
1 -28.331 62.662 69.408    9.208          68
```

Making Predictions in the Test Sample with m1

Let's use model m1 to predict fatalism scores for our test sample group.

```
fatalism = 0.14 + 0.40 simplicity + 0.48 depression
```

```
head(ginz.test,1)
```

```
# A tibble: 1 x 4
     id fatalism simplicity depression
  <int>    <dbl>      <dbl>      <dbl>
1     1  0.35589    0.92983     0.5987
```

So, predicted fatalism for subject 1 here is. . .

```
fatalism = 0.14 + 0.40 (0.92983) + 0.48 (0.5987), or 0.80
```

Observed error is 0.36 - 0.80 = -0.44

There must be an easier way

And there is…

```
predict(m1, newdata = ginz.test)
```

```
        1         2         3         4         5
0.7977475 0.7377445 0.6013353 0.8820320 0.6698679
        6         7         8         9        10
0.6455903 0.6082102 0.5705040 1.2421200 1.0197707
       11        12
1.3530064 1.9511248
```

Making Predictions in the Test Sample with `m1`

Let's use our model `m1` to predict fatalism scores for the test sample group of 12 patients on the basis of their simplicity and depression scores.

```r
m1.preds <- predict(m1, newdata = ginz.test)
# make predictions

m1.error <- ginz.test$fatalism - m1.preds
# calculate errors

m1.abserror <- abs(m1.error)
# absolute value of errors

m1.sqerror <- m1.error^2
# squared errors
```

Back to the first member of our Test Sample

```
head(ginz.test, 1)
```

```
# A tibble: 1 x 4
     id fatalism simplicity depression
  <int>    <dbl>      <dbl>      <dbl>
1     1  0.35589    0.92983     0.5987
```

```
m1.preds[1] # predicted fatalism from m1
```

```
        1
0.7977475
```

```
m1.error[1] # error (observed - predicted)
```

```
         1
-0.4418575
```

Making Predictions in the Test Sample with m2

Using model m2, we have:

```r
m2.preds <- predict(m2, newdata = ginz.test) # predictions
m2.error <- ginz.test$fatalism - m2.preds # errors
m2.abserror <- abs(m2.error) # absolute value of errors
m2.sqerror <- m2.error^2 # squared errors
```

Mean Absolute Prediction Error (MAPE) across the Models

```
summary(m1.abserror)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1245  0.2391  0.3758  0.3963  0.4456  1.1810 
```

```
summary(m2.abserror)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1146  0.2131  0.3350  0.3811  0.4420  0.9936 
```

Mean Squared Prediction Error (MSPE) across the Model

```
summary(m1.sqerror)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.01551 0.05771 0.14125 0.22686 0.19859 1.39473 
```

```
summary(m2.sqerror)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.01313 0.04601 0.11351 0.20242 0.20093 0.98721 
```

Which Model Looks Best in the test sample?

Model	MAPE	MSPE	Max Abs Err
`m1` (depression + simplicity)	0.396	0.227	1.18
`m2` (depression only)	0.381	0.202	0.99

What we see here in the 12(**!**) people in our test group doesn't entirely match what we saw in the training sample of 70 people.

- But should it?
- In 432, we'll learn some better ways to validate our models.

Calibrating Yourself on Residual Plots

Multivariate Regression: Checking Assumptions

Assumptions (see Course Notes, Section 42)

- Linearity
- Normality
- Homoscedasticity
- Independence

Available Residual Plots

```
plot(model, which = c(1:3,5))
```

1. Residuals vs. Fitted Values
2. Normal Q-Q Plot of Standardized Residuals
3. Scale-Location Plot
4. Index Plot of Cook's Distance
5. Residuals, Leverage and Influence

An Idealized Model (by Simulation)

```r
set.seed(431122)

x1 <- rnorm(200, 20, 5)
x2 <- rnorm(200, 20, 12)
x3 <- rnorm(200, 20, 10)

er <- rnorm(200, 0, 1)

y <- .3*x1 - .2*x2 + .4*x3 + er

sim0 <- data.frame(y, x1, x2, x3) %>% tbl_df

mod0 <- lm(y ~ x1 + x2 + x3, data = sim0)

summary(mod0) # appears on next slide
```

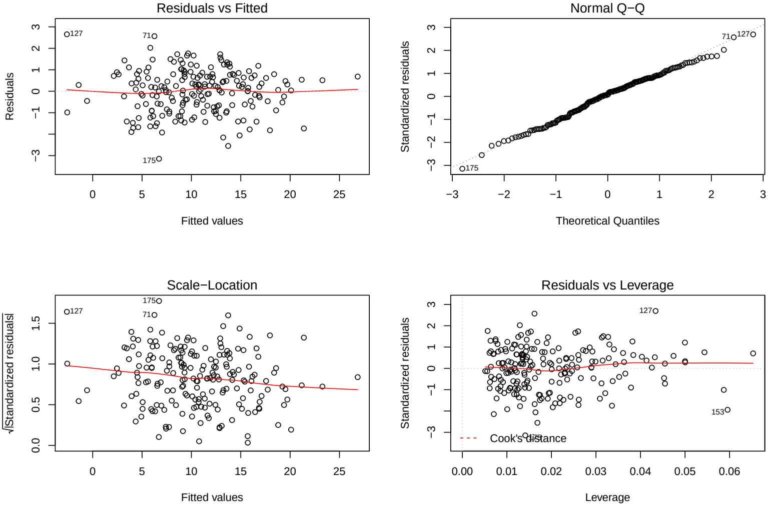
An Idealized Model (by Simulation)

```
Call: lm(formula = y ~ x1 + x2 + x3, data = sim0)

Residuals:     Min       1Q   Median       3Q      Max
          -3.14553 -0.68079  0.08096  0.69216  2.65265

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.122852   0.348584   0.352    0.725
x1            0.285539   0.014211  20.093   <2e-16 ***
x2           -0.204908   0.005828 -35.159   <2e-16 ***
x3            0.413308   0.007172  57.631   <2e-16 ***
---
Signif codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.007 on 196 degrees of freedom
Multiple R-squared:  0.9589,    Adjusted R-squared:  0.9583
F-statistic:  1524 on 3 and 196 DF,  p-value: < 2.2e-16
```

Building Residual Plots for Idealized Model

```
par(mfrow=c(2,2))
plot(mod0)
par(mfrow=c(1,1))
```

Residual Analysis (Idealized Model: n = 200)

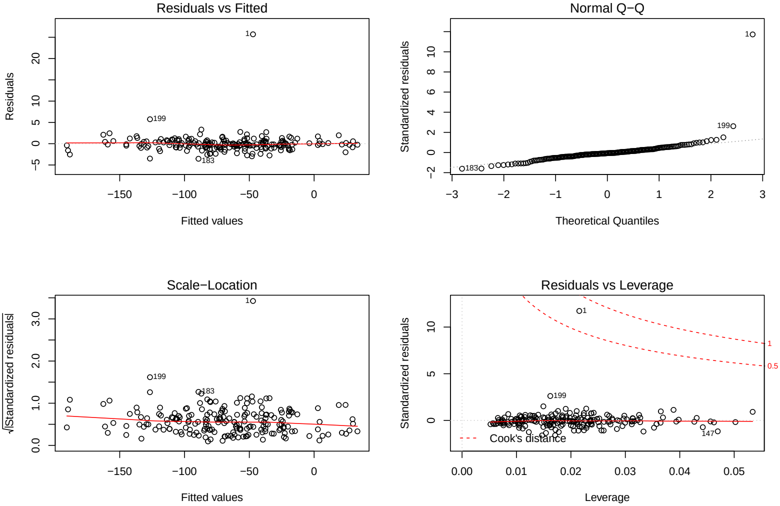


What's the Goal Here?

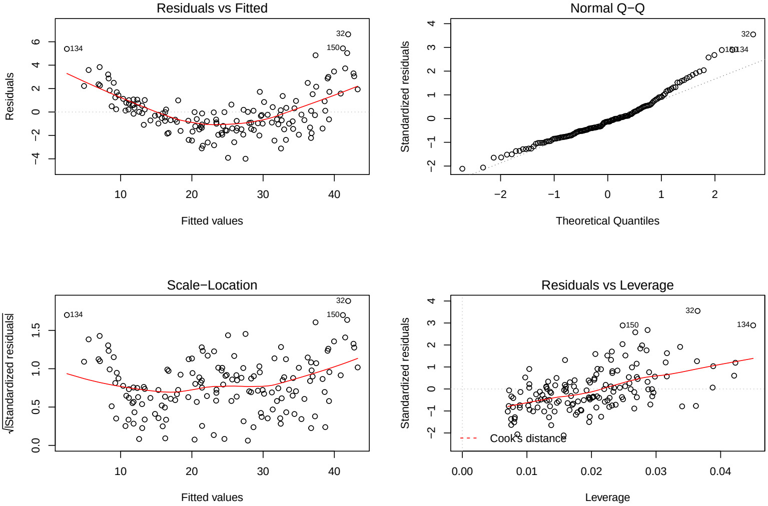
Develop an effective model. (?) (!)

- Models can do many different things. What you're using the model for matters, a lot.
- Don't fall into the trap of making binary decisions (this model isn't perfect, no matter what you do, and so your assessment of residuals will also have shades of gray).
- The tools we have provided (scatterplots, mostly) are well designed for rather modest sample sizes. When you have truly large samples, they don't scale very well.
- Just because R chooses four plots for you to study doesn't mean they provide the only relevant information.
- Embrace the uncertainty. Look at it as an opportunity to study your data more effectively.

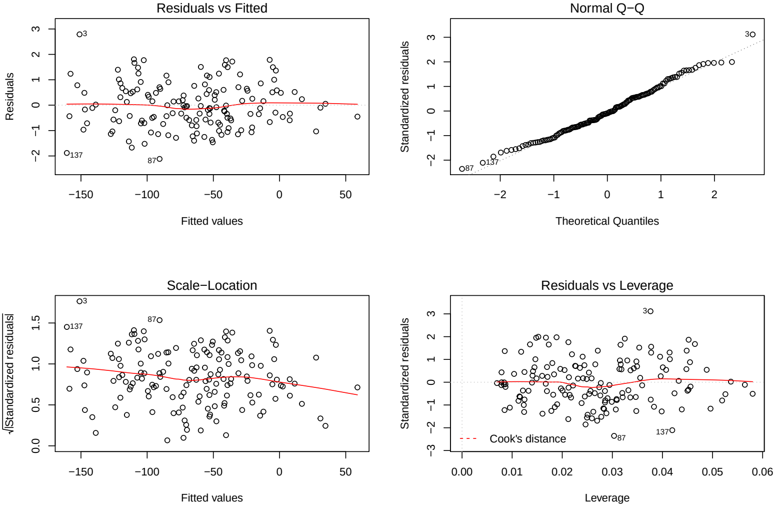
Simulation 1 (n = 200 subjects)



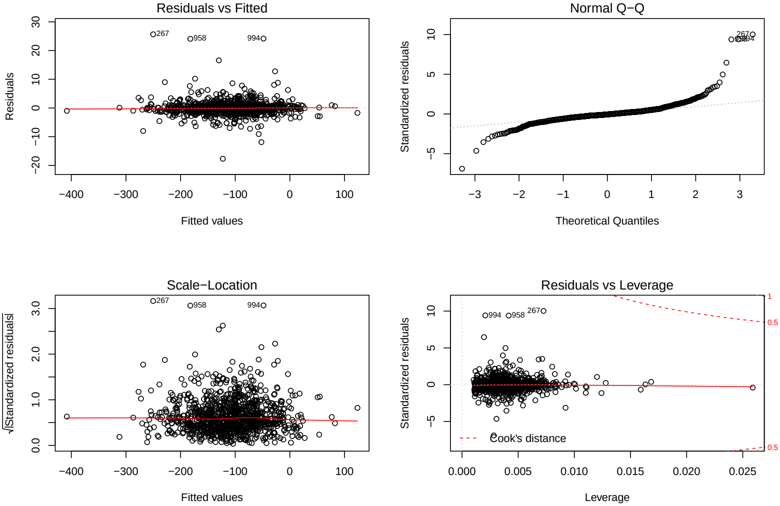
Simulation 2 (n = 150)



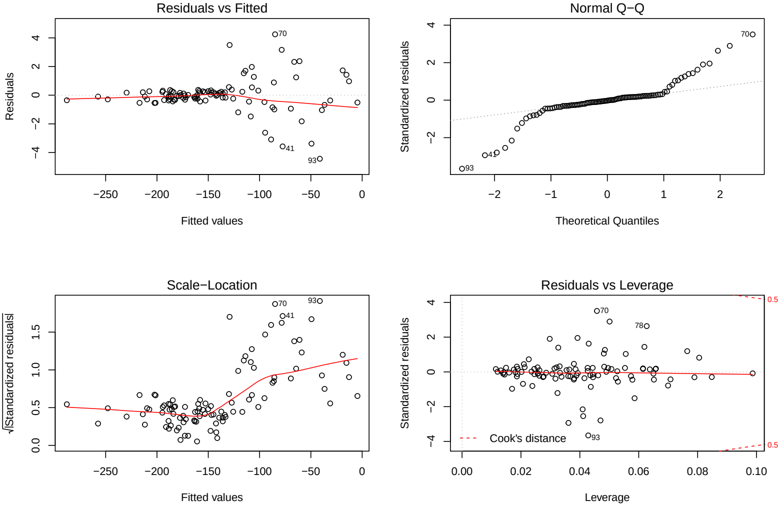
Simulation 3 (n = 150)



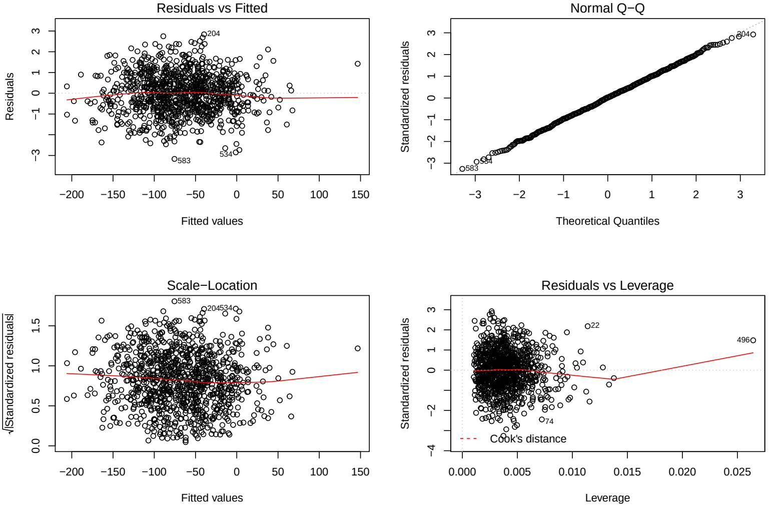
Simulation 4 (n = 1000)



Simulation 5 (n = 100)



Simulation 6 (n = 1000)



Some Reactions to the Six Simulations

For those of you playing along at home...

1. Observation 1 has an impossibly large standardized residual, and some influence.
2. Curve in residuals vs. fitted values plot suggests potential non-linearity.
3. No substantial problems, although there's a little bit of heteroscedasticity.
4. Normality issues - outlier-prone even with 1000 observations.
5. Serious heteroscedasticity - residuals much more varied for larger fitted values.
6. No serious violations - point 496 has very substantial leverage, though.