

431 Class 26

Thomas E. Love

2017-12-05

R Setup for Today

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

dm192 <- read_csv("data/dm192.csv")
```

Today's Agenda

- Discussion of Assignment 6
- Reasoning About Data
 - A data analysis allow you to understand how the *data*, as opposed to other aspects of an analysis like assumptions or models, played a role in producing the outputs. (Roger Peng)
- Anscombe's Data: A Quartet of Simple Regressions
- Calibrating Yourself on Residual Plots
- The dm192 data
 - We have 7 regression inputs. How well can we predict today's systolic BP?

Assignment 6 Discussion

What was the hardest part of the assignment?

- Question 2
 - [a] finding the visualization [b] describing it [c] improving it
- Question 3
 - [a] transformation check [b] predictor selection [c] assess collinearity
- Question 4
 - [a] detailed description of model in English [b] model utility (R^2 , sig tests) [c] check assumptions
- Question 5
 - turn the model into a single sentence

The history of Rome in 10 words

A small city grew gradually to become a vast empire.

The history of Rome in 25 words

Under kings and emperors and as a republic, Rome grew constantly for almost a thousand years, and became the greatest power of the ancient world.

The history of Rome in 50 words.

Founded 753 BC, Rome was ruled by kings until 509 BC, and was a republic then until Julius Caesar took power in 46 BC. After Caesar Rome became an empire, with an emperor. Rome's growth from a small city to a great empire was sustained by their success in wars.

The history of Rome in 100 words.

Founded about 753 BC, Rome was a belligerent little town that consistently prevailed in conflicts with rivals, and thus grew gradually in size and influence. Ruled by kings until 509 BC, Rome transitioned into a disciplined, well-organized republic which sustained excellence in military preparation and capacity, and thus continued to grow for hundreds of years. Ruled by an emperor from about 27 BC, Rome reached the zenith of its power in the years 100 to 180 AD, under the rulers known as the Five Good Emperors. A series of corrupt and incompetent rulers allowed Rome to slip gradually into history.

- Bill James *Fools Rush Inn: More Detours on the Way to Conventional Wisdom*

<https://xkcd.com/1349/>

WHAT I'M TRYING TO
DO IS REALLY SIMPLE.
IT SHOULDN'T BE HARD.



ALL COMPUTERS ARE JUST
CAREFULLY ORGANIZED SAND.
**EVERYTHING IS HARD UNTIL
SOMEONE MAKES IT EASY.**



MAYBE I SHOULD TURN
THIS ONE BACK INTO SAND.



Every detail matters. Computers aren't smart enough to guess our intent.

Reasoning About Data (Roger Peng)

“...I think it's tempting to think of the goal of methods development as removing the need to think about data and assumptions. The "ultimate" method is one where you don't have to worry about distributions or nonlinearities or interactions or anything like that. But I don't see that as the goal. Good methods, and good analyses, help us think about all those things much more efficiently. So what I might say is that...

When doing large-scale data analyses, the data analyst always has to think about the data and assumptions, and as such, some approaches can actually make that harder to do than others. The goal of the good data analysis is to make it easier to reason about how the data are related to the result, relative to the assumptions you make about the data and the models."

Roger Peng: <https://simplystatistics.org/2017/11/20/follow-up-on-reasoning-about-data/>

Just show me the data.



**“Let’s shrink Big Data into Small Data ...
and hope it magically becomes Great Data.”**

The history of Rome in 1000 words



Anscombe's Data: A Famous Example

A Quartet of Simple Linear Regressions

anscombe

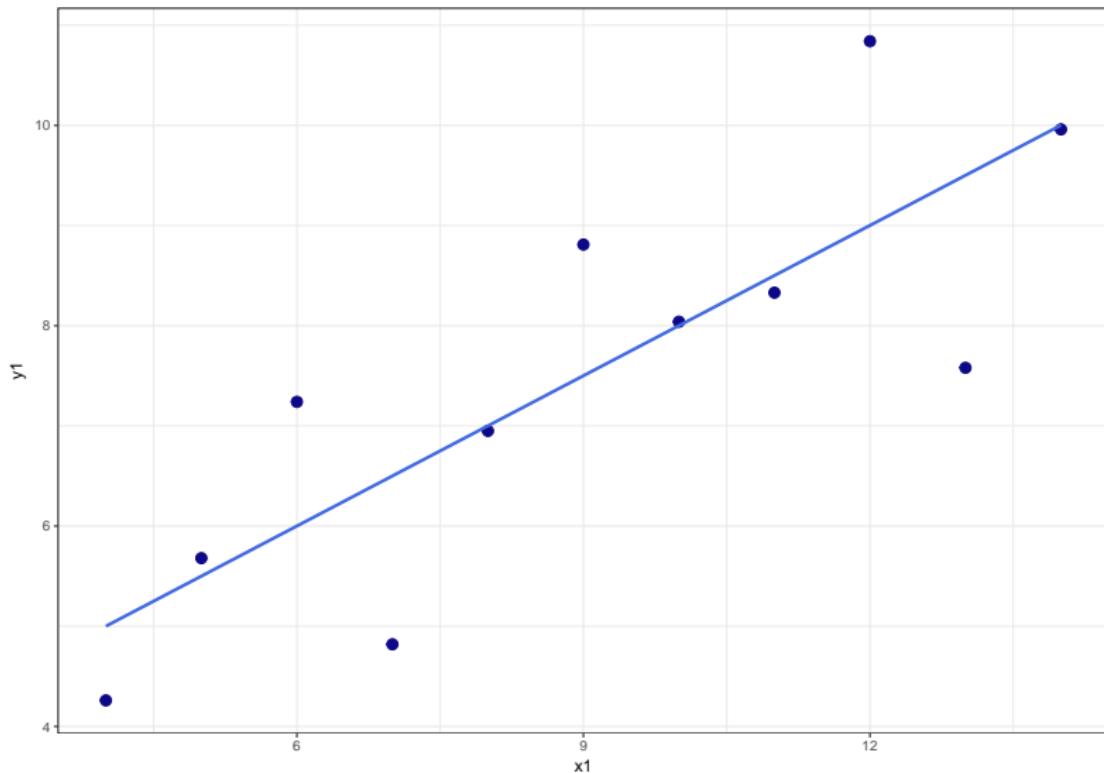
	x1	x2	x3	x4	y1	y2	y3	y4
1	10	10	10	8	8.04	9.14	7.46	6.58
2	8	8	8	8	6.95	8.14	6.77	5.76
3	13	13	13	8	7.58	8.74	12.74	7.71
4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04
7	6	6	6	8	7.24	6.13	6.08	5.25
8	4	4	4	19	4.26	3.10	5.39	12.50
9	12	12	12	8	10.84	9.13	8.15	5.56
10	7	7	7	8	4.82	7.26	6.42	7.91
11	5	5	5	8	5.68	4.74	5.73	6.89

Anscombe Model 1

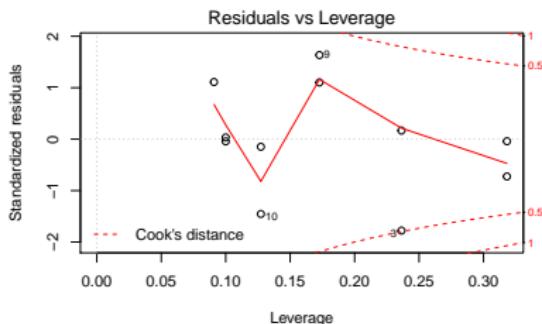
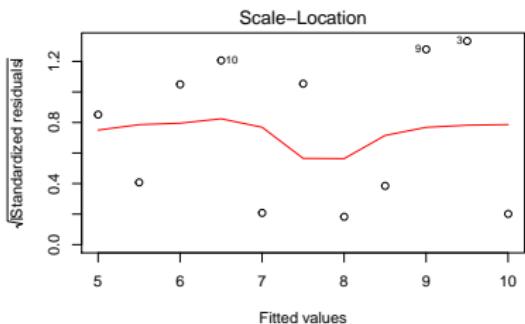
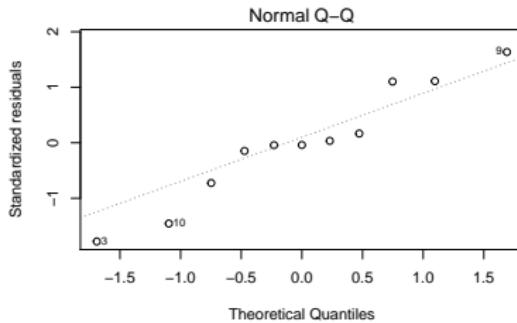
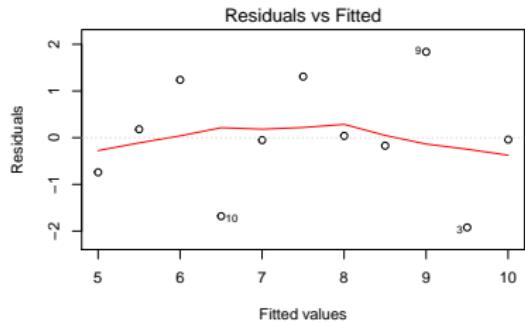
```
arm::display(lm(y1 ~ x1, data = anscombe))
```

```
lm(formula = y1 ~ x1, data = anscombe)
      coef.est  coef.se
(Intercept) 3.00     1.12
x1          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

Plot of y_1 vs. x_1



Residual Plots for Anscombe Model 1



Anscombe Model 2

```
arm::display(lm(y2 ~ x2, data = anscombe))
```

```
lm(formula = y2 ~ x2, data = anscombe)
      coef.est  coef.se
(Intercept) 3.00     1.13
x2          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

Anscombe Model 3

```
arm::display(lm(y3 ~ x3, data = anscombe))
```

```
lm(formula = y3 ~ x3, data = anscombe)
      coef.est  coef.se
(Intercept) 3.00     1.12
x3          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

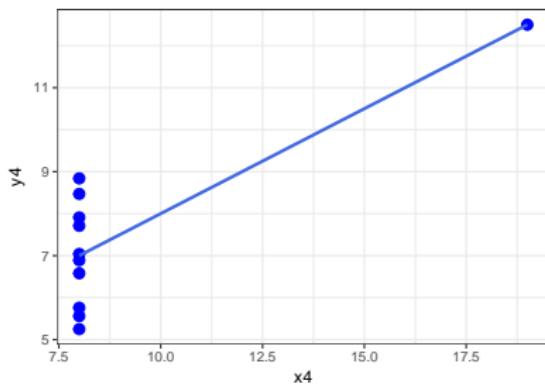
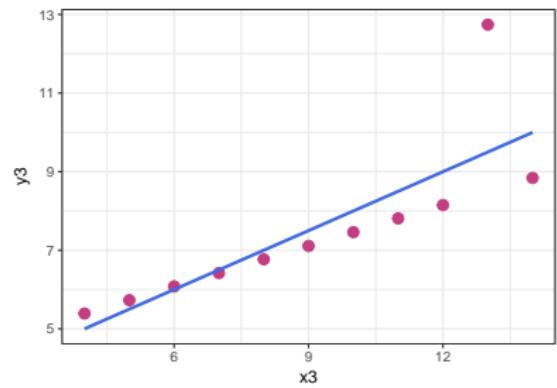
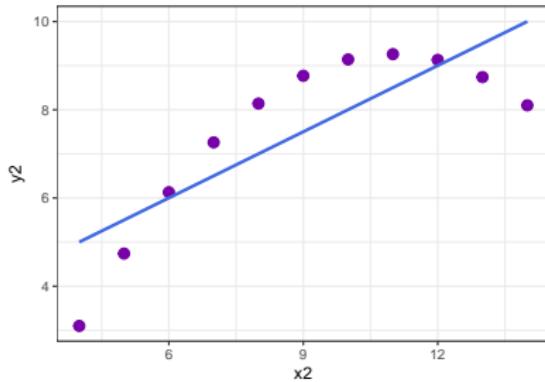
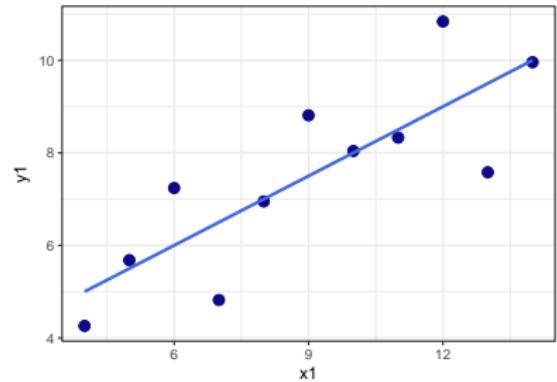
Anscombe Model 4

```
arm::display(lm(y4 ~ x4, data = anscombe))
```

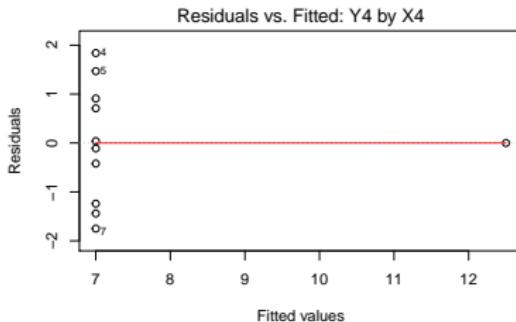
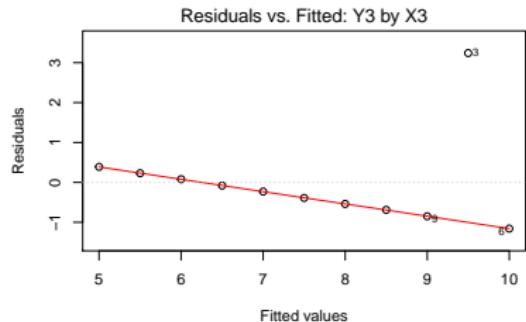
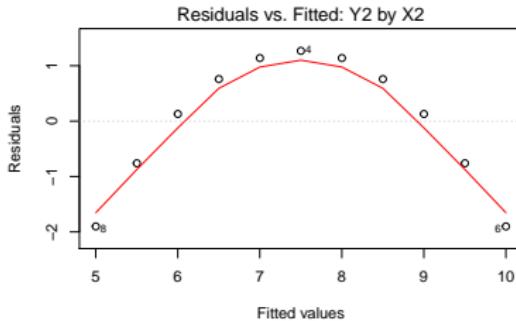
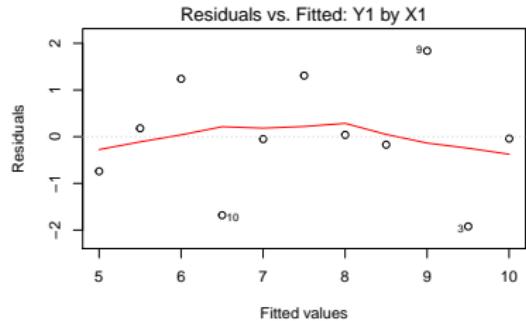
```
lm(formula = y4 ~ x4, data = anscombe)
      coef.est  coef.se
(Intercept) 3.00     1.12
x4          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

Models 1-4 all look about the same, but what happens if we plot the data?

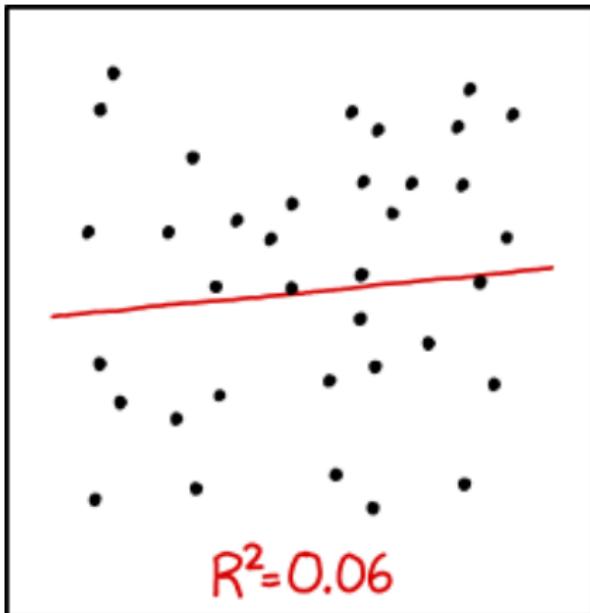
Plot the Data (and the regression lines)



Do Residuals vs. Fitted Plots reveal the problems?



<https://xkcd.com/1725/>



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER
TO GUESS THE DIRECTION OF THE CORRELATION FROM THE
SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

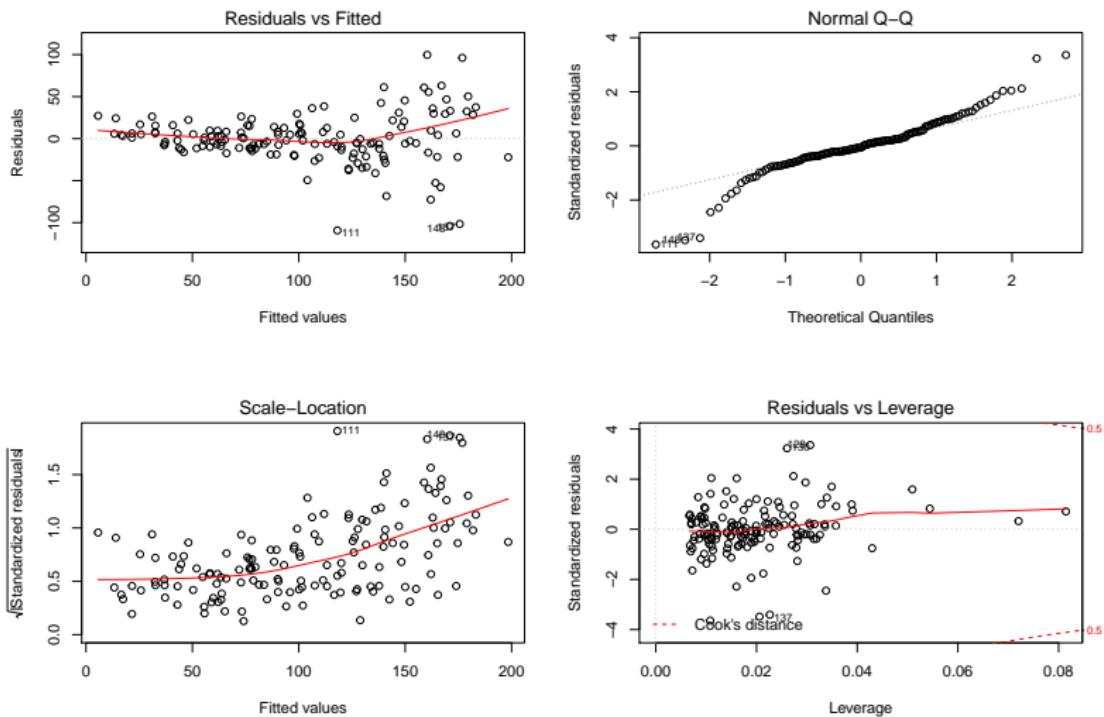
Calibrating Yourself on Residual Plots: Five New Examples

Your Response Options are

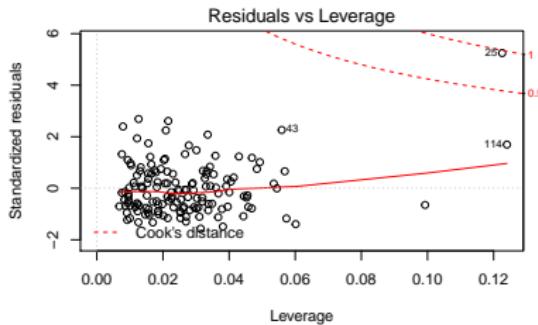
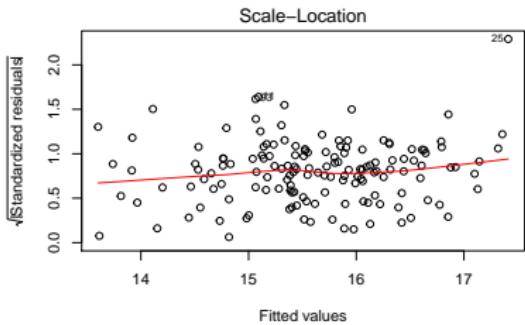
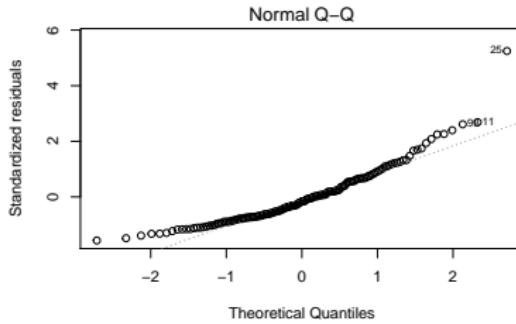
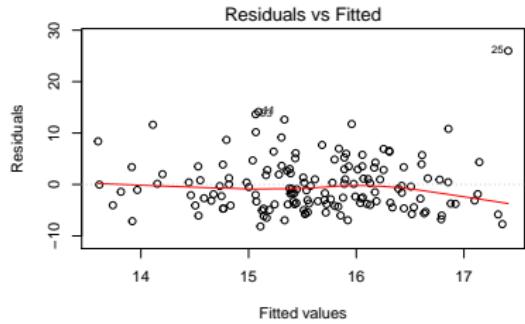
- ① Assumptions of regression look reasonable.
- ② Biggest problem is Linearity.
- ③ Biggest problem is non-constant variance.
- ④ Biggest problem is Normality
- ⑤ Some other problem is the biggest issue.

All of these models describe cross-sectional data, and so you can assume there's no possible issue regarding independence of residuals.

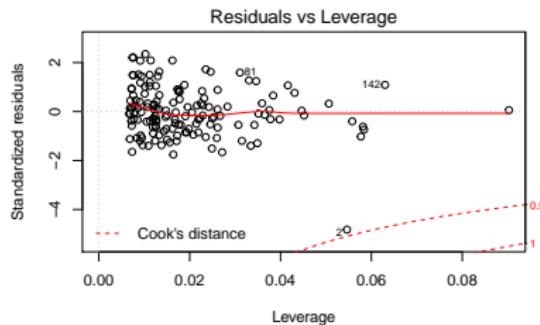
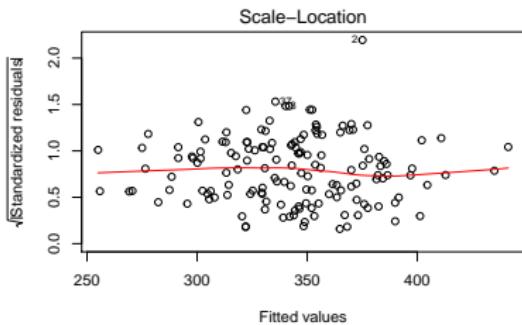
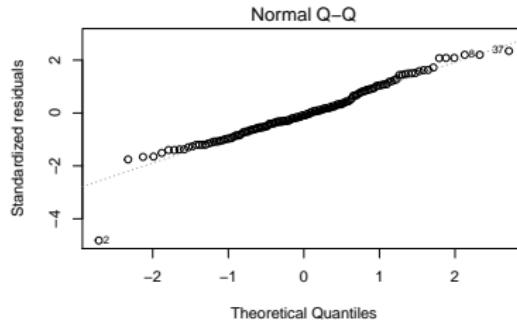
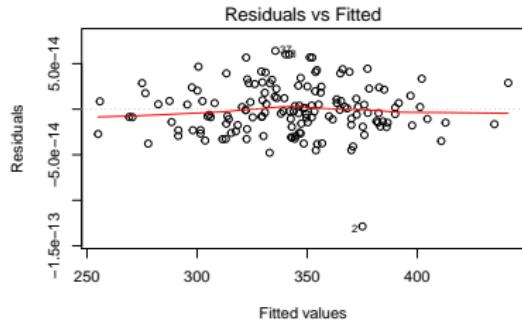
Example A



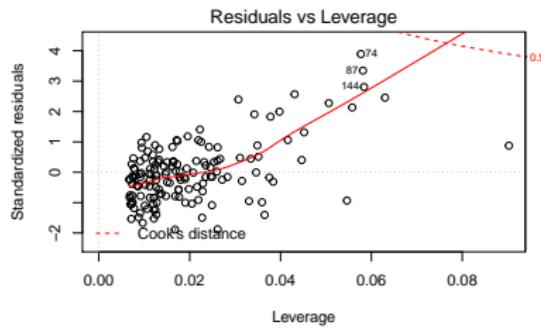
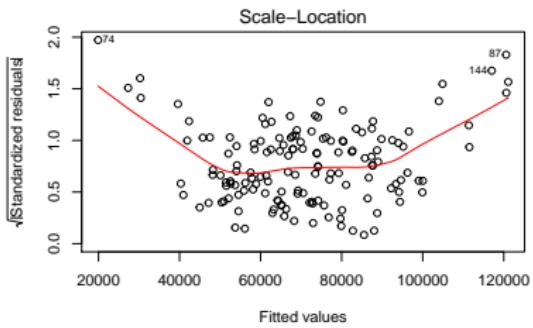
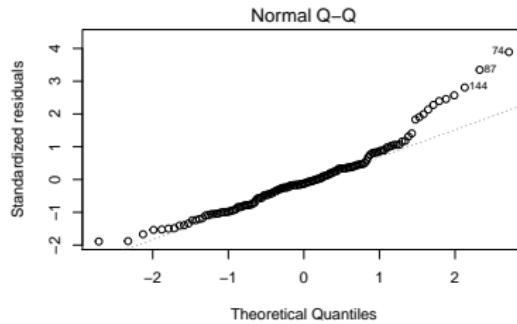
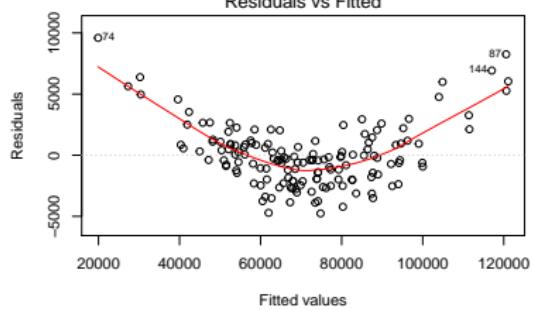
Example B



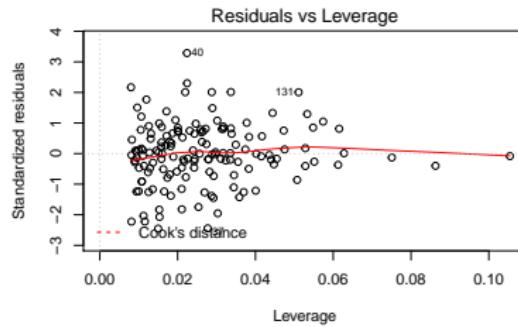
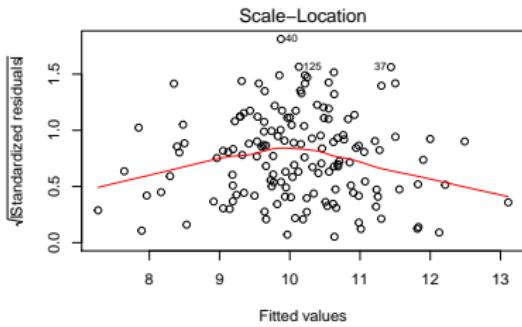
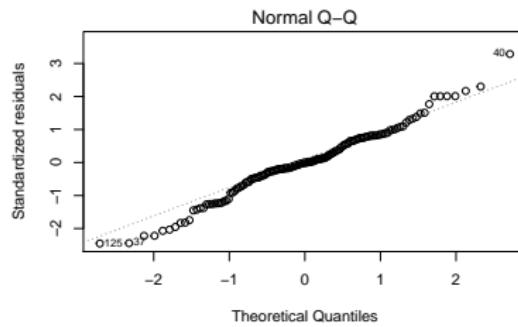
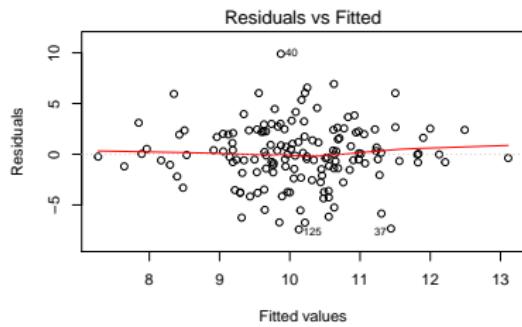
Example C



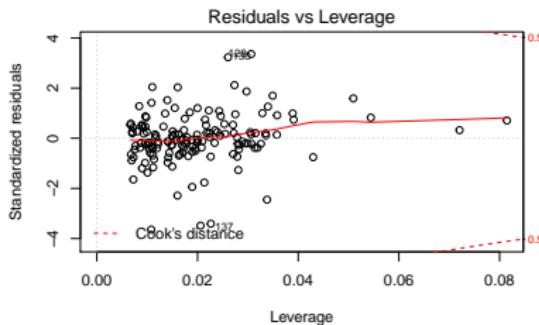
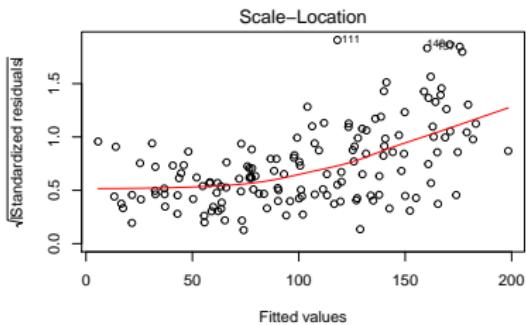
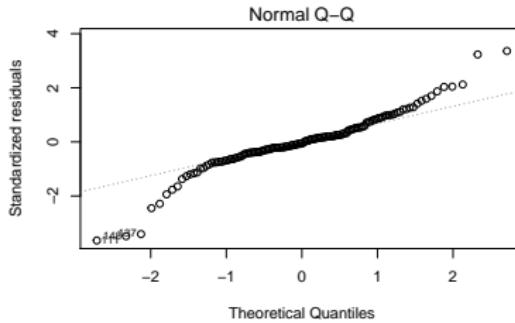
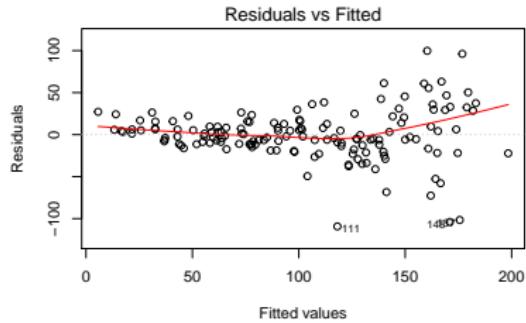
Example D



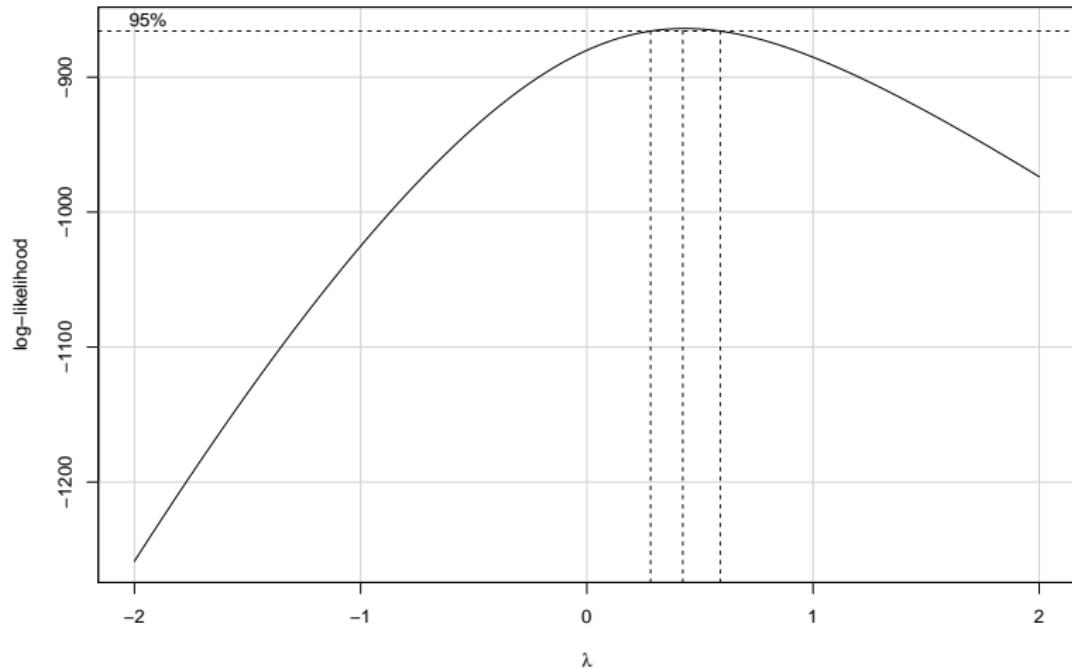
Example E



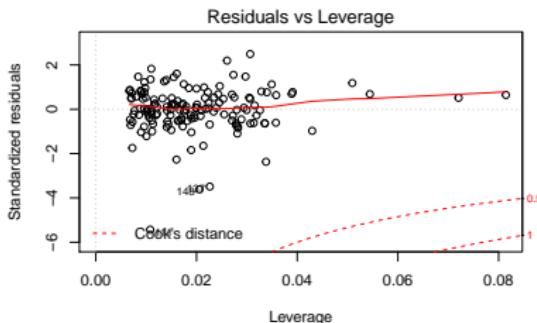
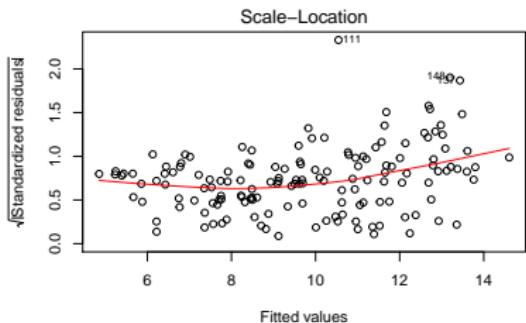
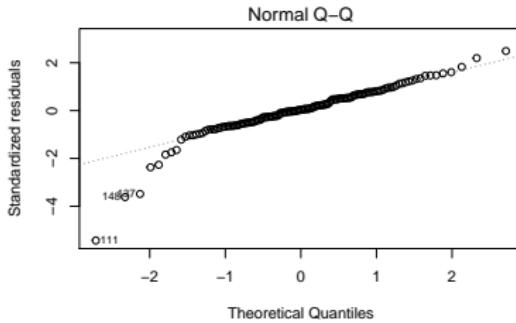
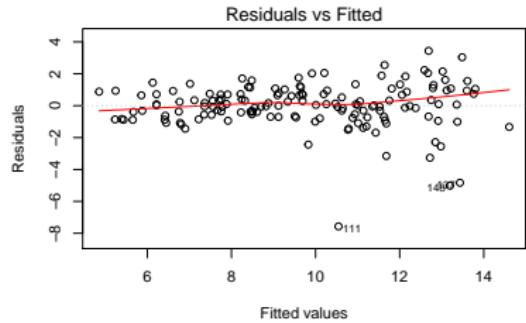
OK, back to Example A



Box Cox for Model A



Model A, but with square root of Y



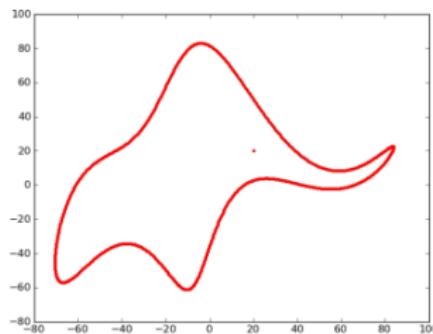
Why I limit transformations in 431 a bit

John von Neumann famously said

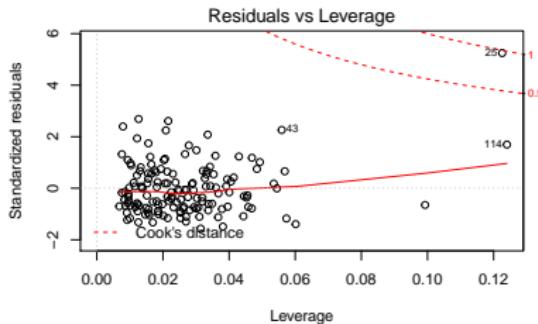
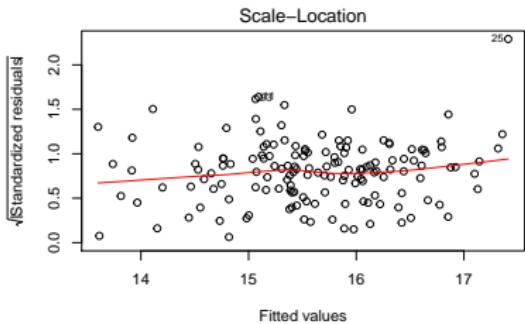
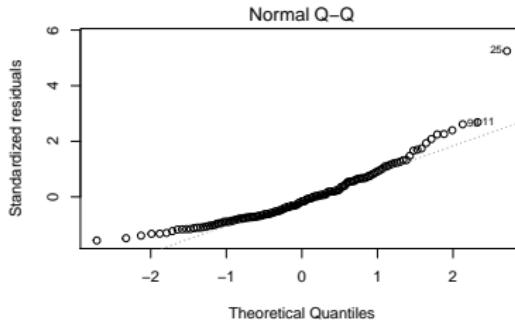
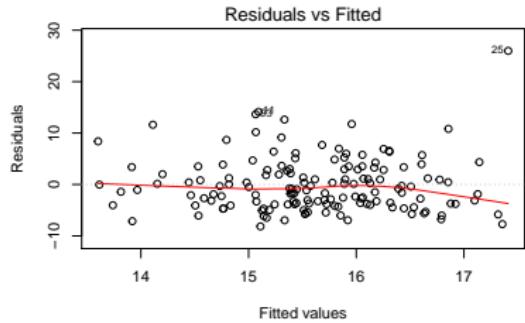
With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

By this he meant that one should not be impressed when a complex model fits a data set well. With enough parameters, you can fit any data set.

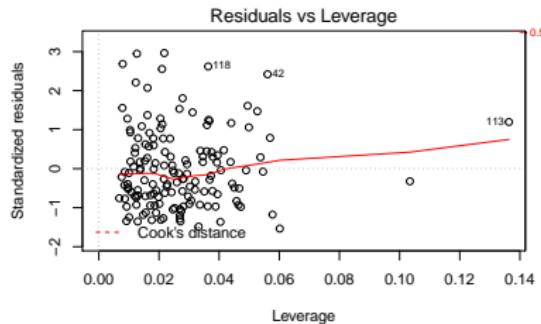
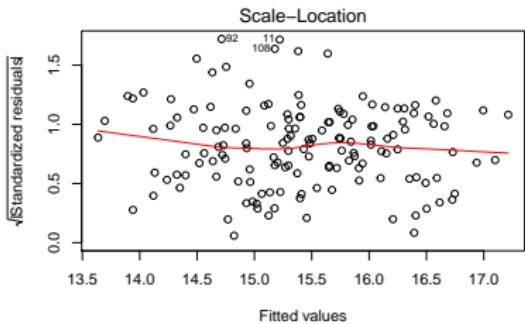
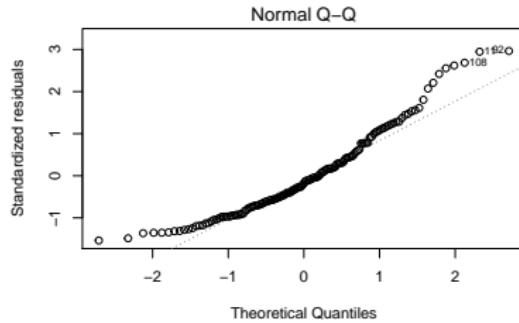
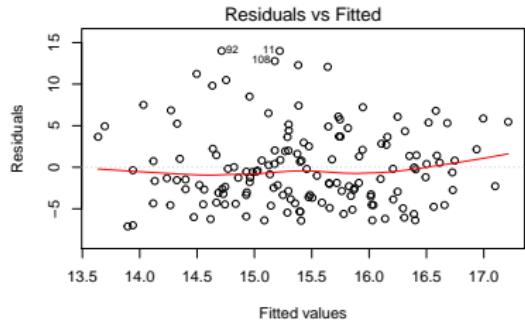
It turns out you can literally fit an elephant with four parameters if you allow the parameters to be complex numbers.



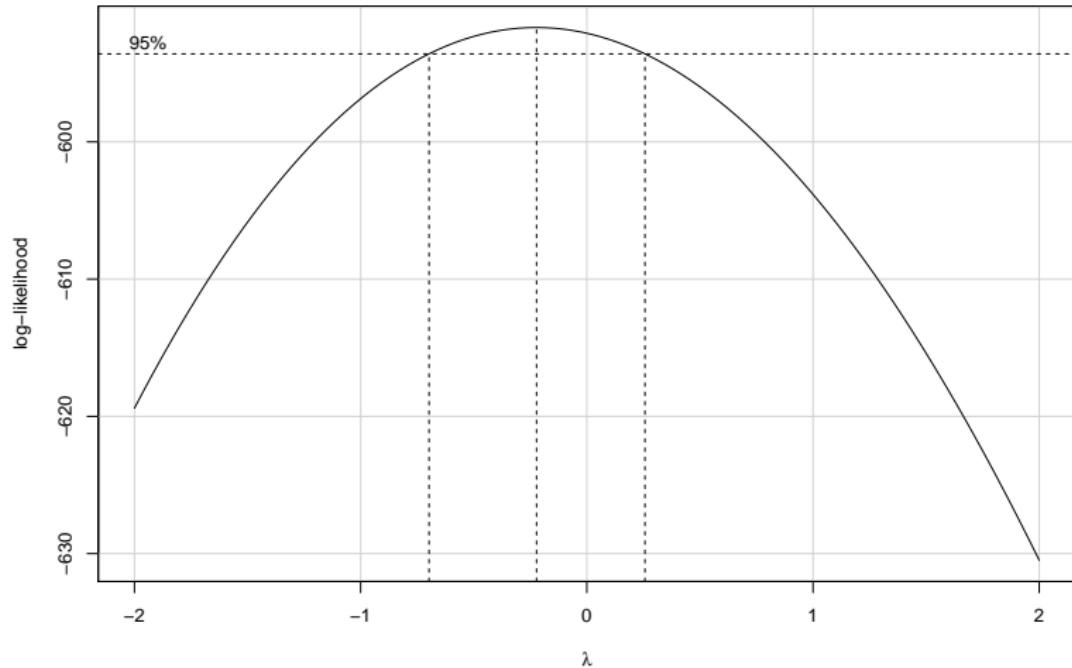
Example B



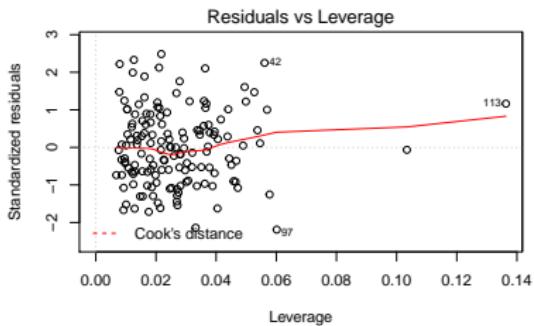
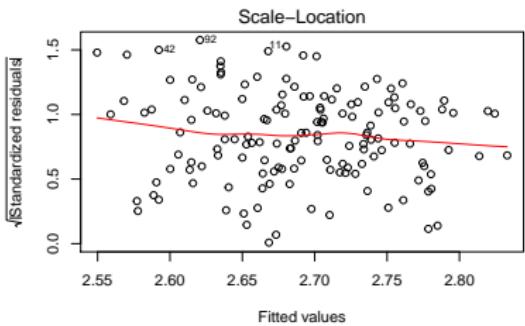
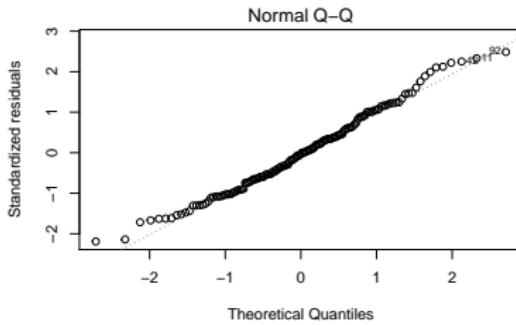
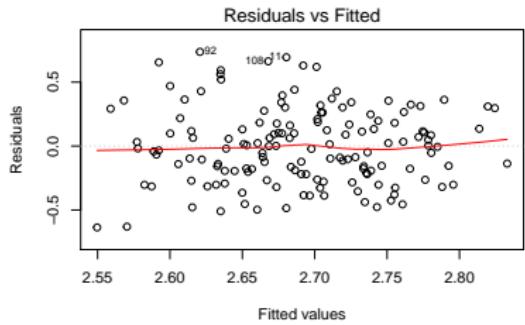
Example B without point 25



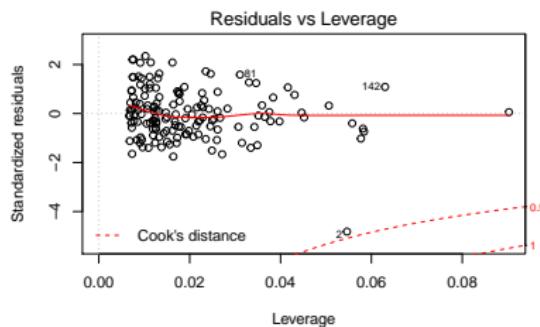
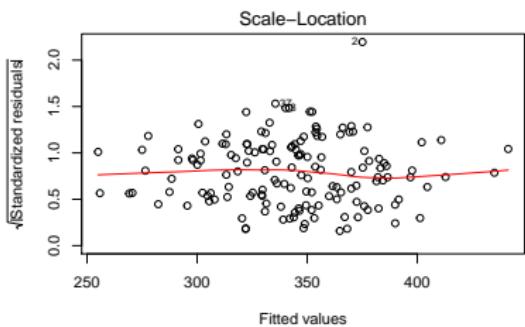
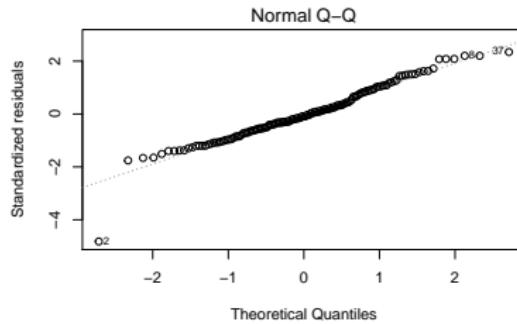
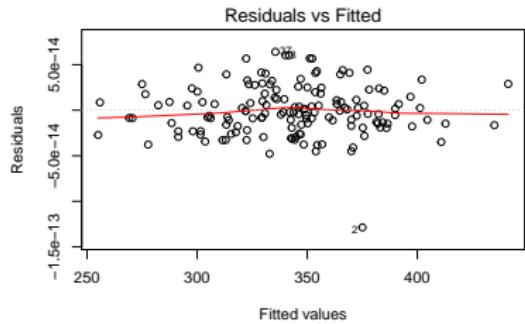
Box-Cox for new Model B (-25)



$\log(Y)$ model B (-25)



Example C



summary(mC)

Warning message:

In summary.lm(mC) : essentially perfect fit:
summary may be unreliable

Coefficients:

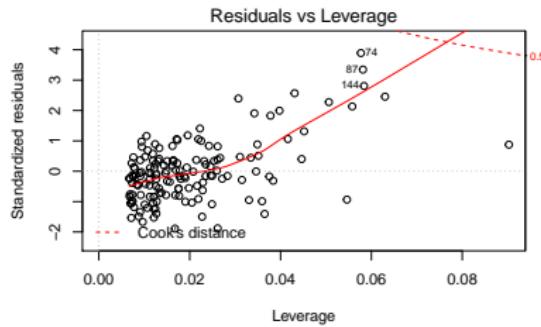
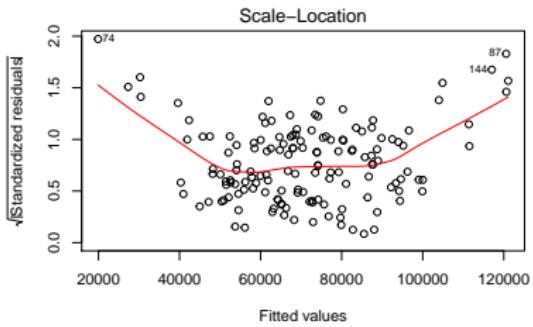
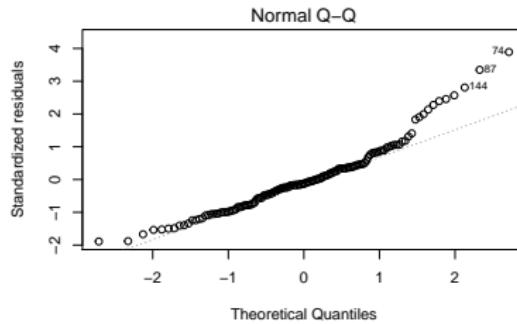
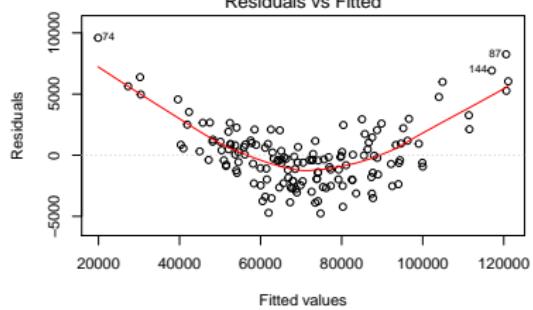
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	7.426e-14	2.275e-14	3.264e+00	0.00137	**
x1	3.000e+00	2.020e-16	1.485e+16	< 2e-16	***
x2	2.000e+00	4.526e-16	4.419e+15	< 2e-16	***

Residual standard error: 2.745e-14 on 147 df

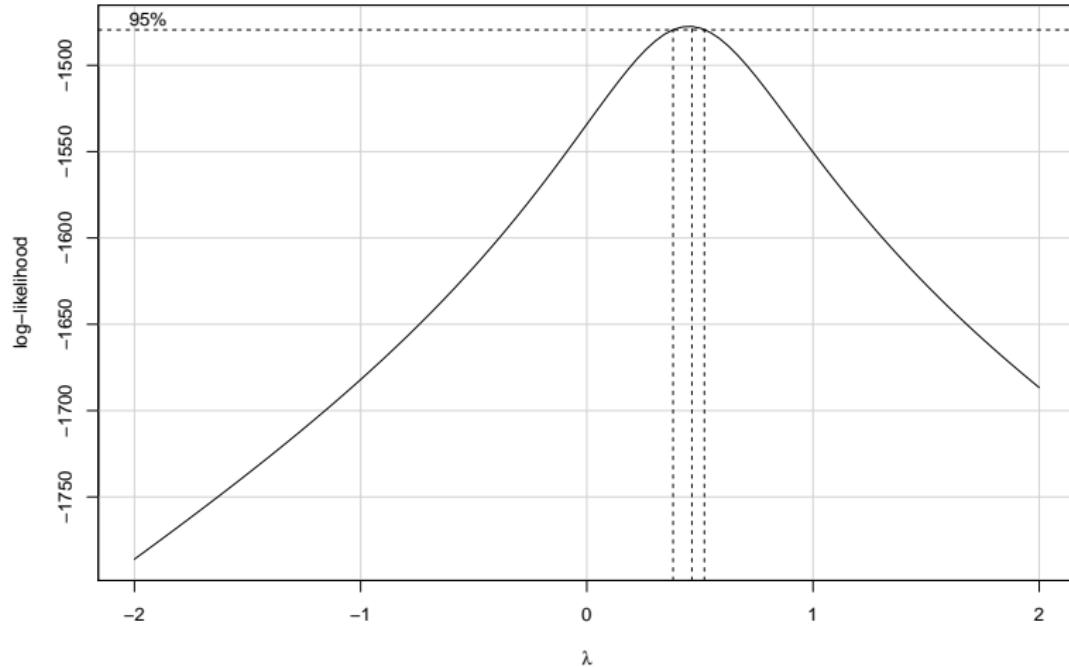
Multiple R-squared: 1, Adjusted R-squared: 1

F-stat: 1.176e+32 on 2 and 147 DF, p-value: < 2.2e-16

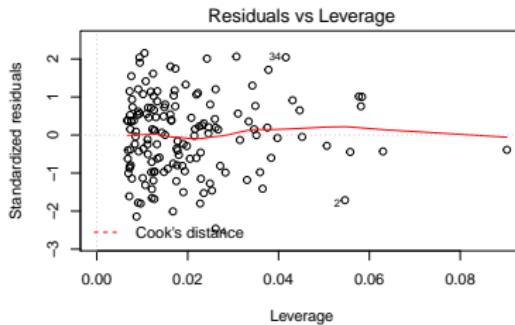
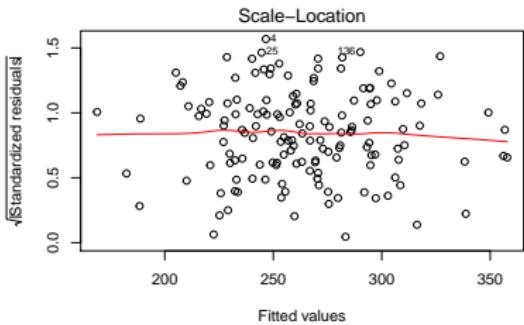
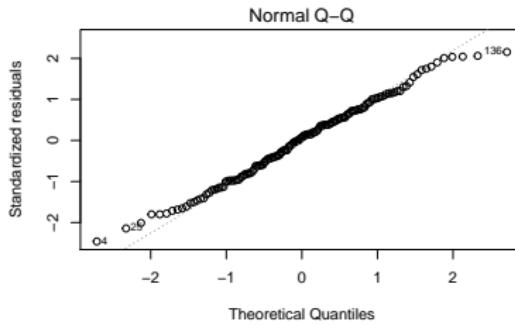
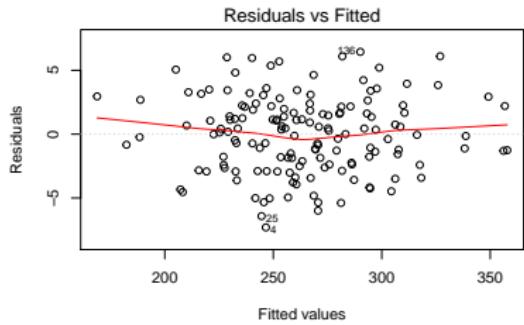
Example D



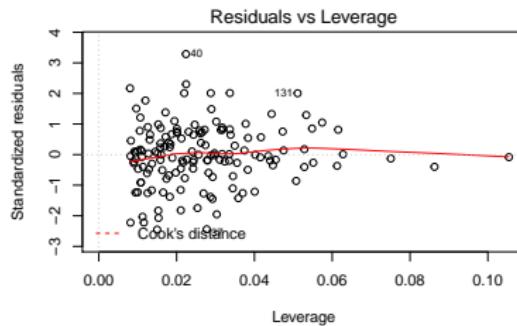
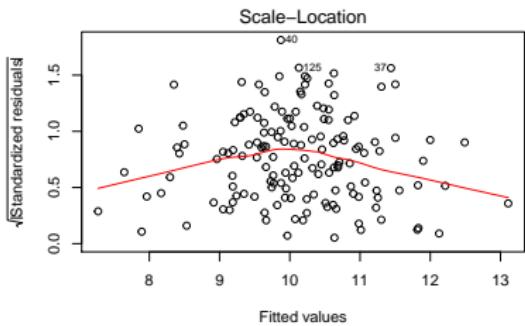
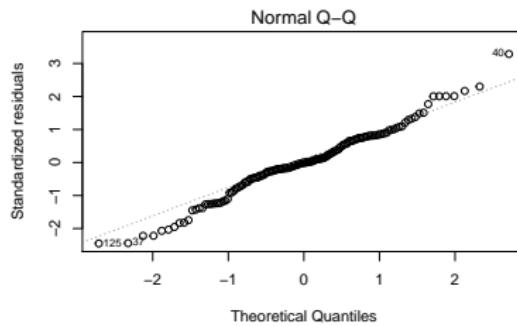
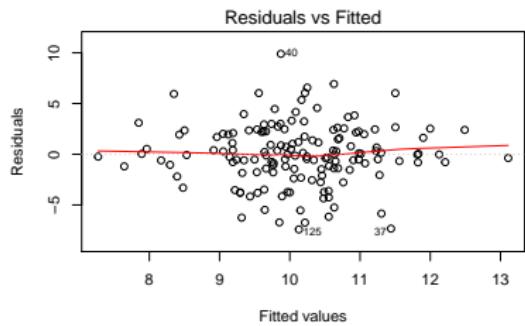
Box-Cox for Model D



New Example D (sqrt of Y)



Example E



Regression and the dm192 data

Our Research Question

Can we predict a patient's sbp level today, if the seven features we can use to predict that are:

- their sbp level one year ago
- their a1c level now
- their age, race, sex and insurance type
- and the practice where they are seen

We want to use some or all of these seven regression inputs to do the best possible job of predicting today's sbp, regardless of which predictors fall in or out of the model.

The dm192 data

```
head(dm192, 4)
```

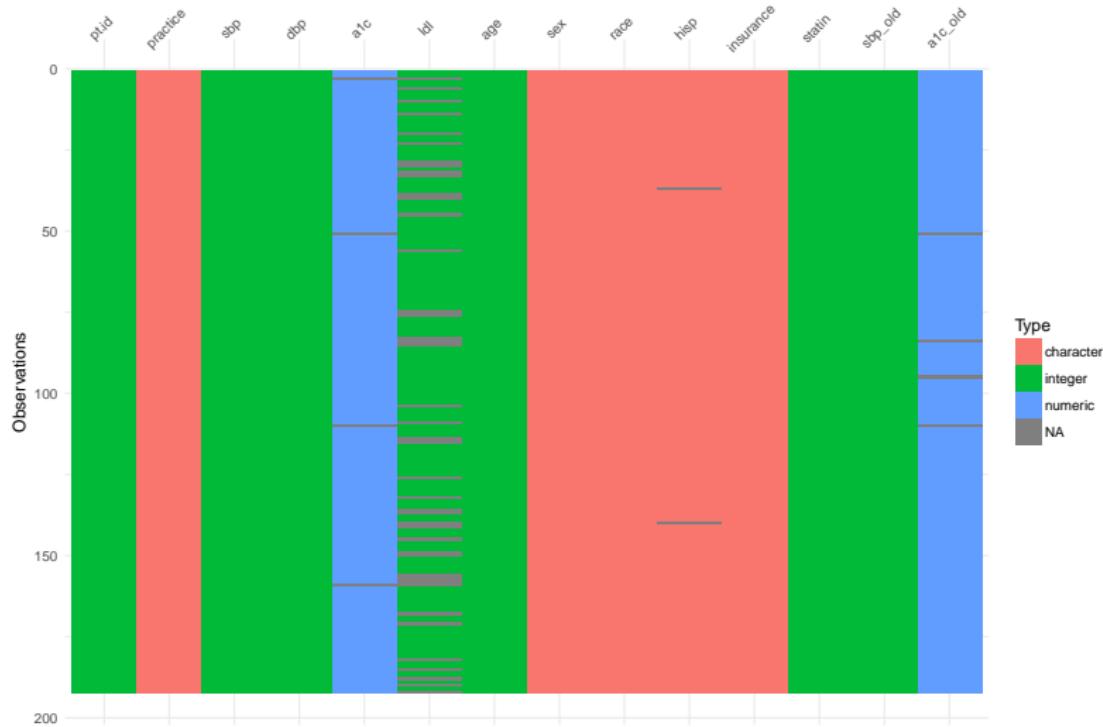
```
# A tibble: 4 x 14
  pt.id practice sbp dbp a1c ldl age sex
  <int>    <chr> <int> <int> <dbl> <int> <int> <chr>
1     1         A   108   71   5.8    58   44 male
2     2         A   162   92  11.6    54   28 female
3     3         B   135   84    NA    NA   58 female
4     4         C   133   87  12.7   112   56 male
# ... with 6 more variables: race <chr>, hisp <chr>,
# insurance <chr>, statin <int>, sbp_old <int>,
# a1c_old <dbl>
```

- We may want to change some of those chr variables to factors.
- We probably want to address the missingness, too.

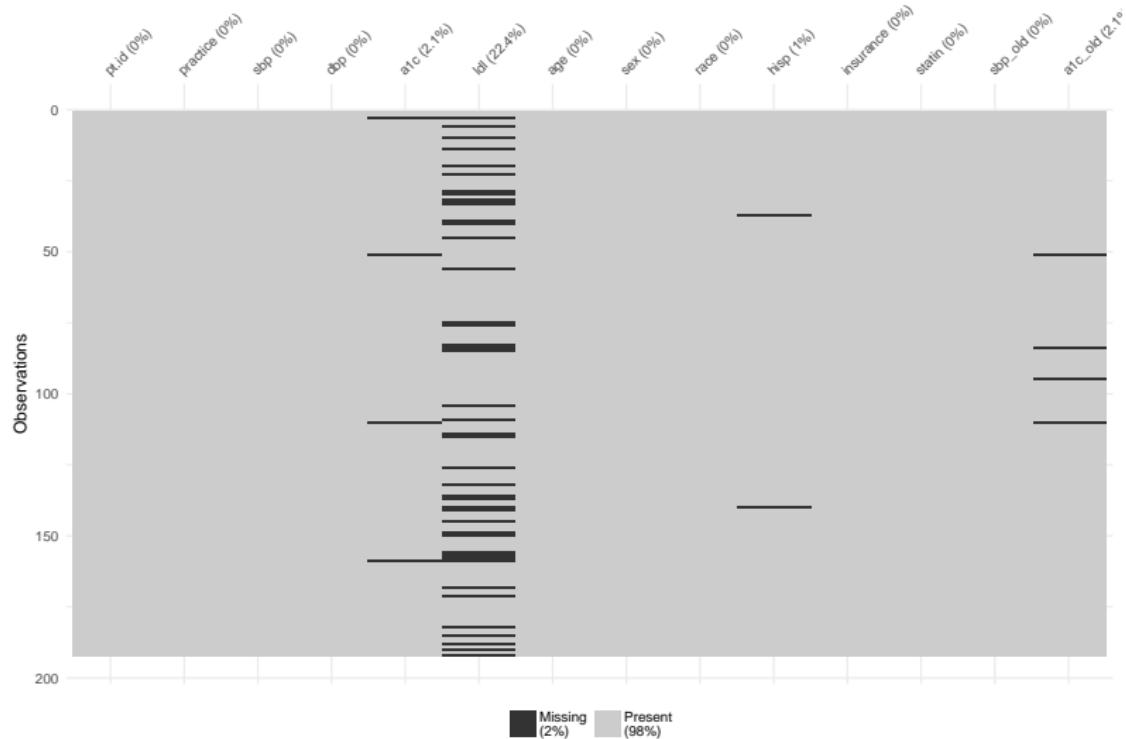
visdat to get a first look?

```
## assumes visdat package is installed from CRAN
library(visdat)
vis_dat(dm192, sort_type = FALSE)
```

vis_dat(dm192, sort_type = FALSE)



vis_miss(dm192)



Forget the algorithms, start cleaning your data!



For 431, we're working with complete cases only

```
dm192_work <- dm192 %>%  
  select(pt.id, sbp, sbp_old, a1c, age, race,  
         sex, insurance, practice) %>%  
  filter(complete.cases(.))  
  
head(dm192_work, 3)
```

```
# A tibble: 3 x 9  
  pt.id    sbp sbp_old    a1c    age   race     sex  
  <int> <int>    <int> <dbl> <int> <chr>    <chr>  
1      1    108      110    5.8     44 black    male  
2      2    162      158   11.6     28 black female  
3      4    133      145   12.7     56 black    male  
# ... with 2 more variables: insurance <chr>,  
#   practice <chr>
```

Change several character variables to factors

```
cols_temp <- c("race", "sex", "insurance", "practice")  
  
dm192_work[cols_temp] <- lapply(dm192_work[cols_temp], factor)  
  
head(dm192_work, 3)  
  
# A tibble: 3 x 9  
  pt.id    sbp sbp_old    a1c    age    race     sex  
  <int> <int>   <int> <dbl> <int> <fctr> <fctr>  
1      1    108      110    5.8     44 black    male  
2      2    162      158   11.6     28 black female  
3      4    133      145   12.7     56 black    male  
# ... with 2 more variables: insurance <fctr>,  
#       practice <fctr>
```

Are the factor levels sensible and sensibly ordered?

(1)

```
dm192_work %>% count(race)
```

```
# A tibble: 4 x 2
  race     n
  <fctr> <int>
1 asian      5
2 black    119
3 other     16
4 white     48
```

Auto-collapse to most common 2 levels, plus “Others”

```
dm192_work$race <- dm192_work$race %>%  
  fct_lump(n = 2, other_level = "Others")  
  
table(dm192_work$race)
```

	black	white	Others
119	119	48	21

Are the factor levels sensible and sensibly ordered? (2)

```
dm192_work %>% count(sex)
```

```
# A tibble: 2 x 2
  sex     n
  <fctr> <int>
1 female   96
2 male     92
```

Are the factor levels sensible and sensibly ordered? (3)

```
dm192_work %>% count(insurance)
```

```
# A tibble: 4 x 2
  insurance     n
  <fctr> <int>
1 commercial    39
2 medicaid      67
3 medicare      76
4 uninsured      6
```

Collapse Medicaid and Uninsured together

```
dm192_work$insurance <-
  fctcollapse(dm192_work$insurance,
              Medicare = "medicare",
              Commercial = "commercial",
              Medicaid_Unins = c("medicaid", "uninsured"))

table(dm192_work$insurance)
```

Commercial	Medicaid_Unins	Medicare
39	73	76

Reorder Factor Levels by Hand

```
dm192_work$insurance <-
  fct_relevel(dm192_work$insurance,
              "Medicare", "Commercial")

table(dm192_work$insurance)
```

	Medicare	Commercial	Medicaid_Unins
	76	39	73

Are the factor levels sensible and sensibly ordered? (4)

```
dm192_work %>% count(practice)
```

```
# A tibble: 4 x 2
  practice     n
  <fctr> <int>
1       A     48
2       B     45
3       C     47
4       D     48
```

The tidyverse can do just about everything.

I, FOR ONE, WELCOME



OUR NEW TIDY OVERLORDS!

themergetteratwork.net

Except think.

Predict sbp as well as you can, in new data

Stage 1. Partition the Data

```
set.seed(43123)
dm192_train <-
  sample_frac(dm192_work, 0.8, replace = FALSE)
dm192_test <-
  anti_join(dm192_work, dm192_train)
```

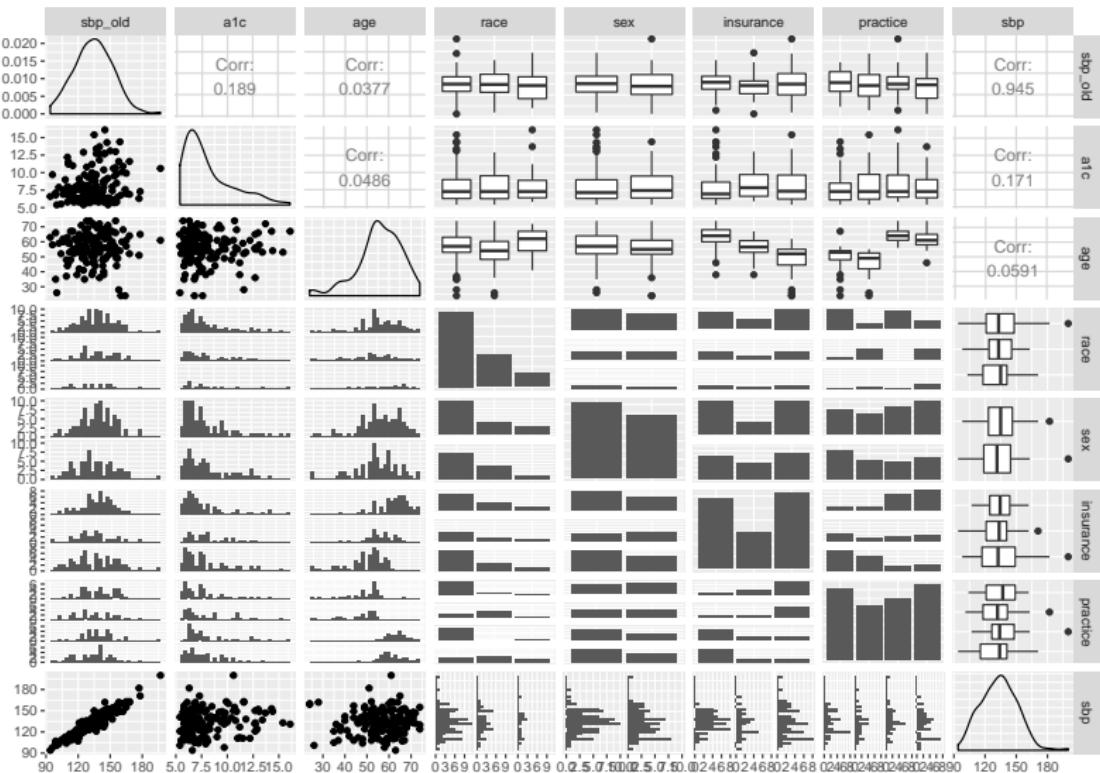
Joining, by = c("pt.id", "sbp", "sbp_old", "a1c", "age", "race")

```
dim(dm192_train); dim(dm192_test)
```

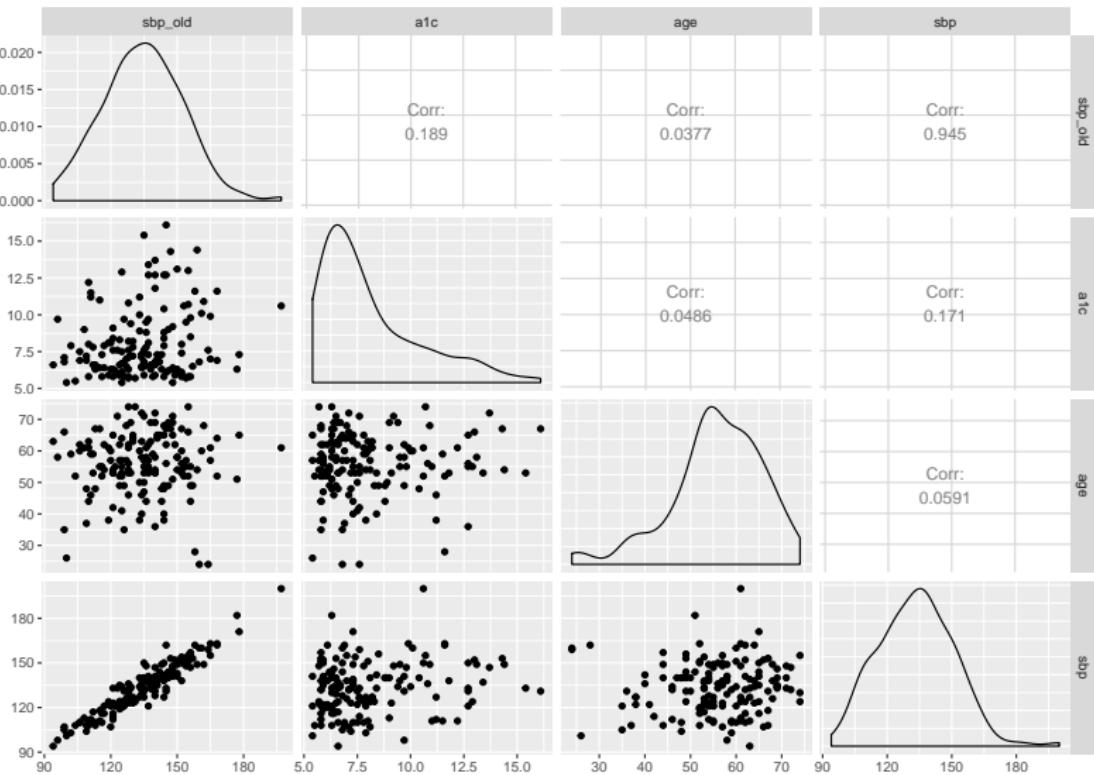
```
[1] 150    9
```

```
[1] 38    9
```

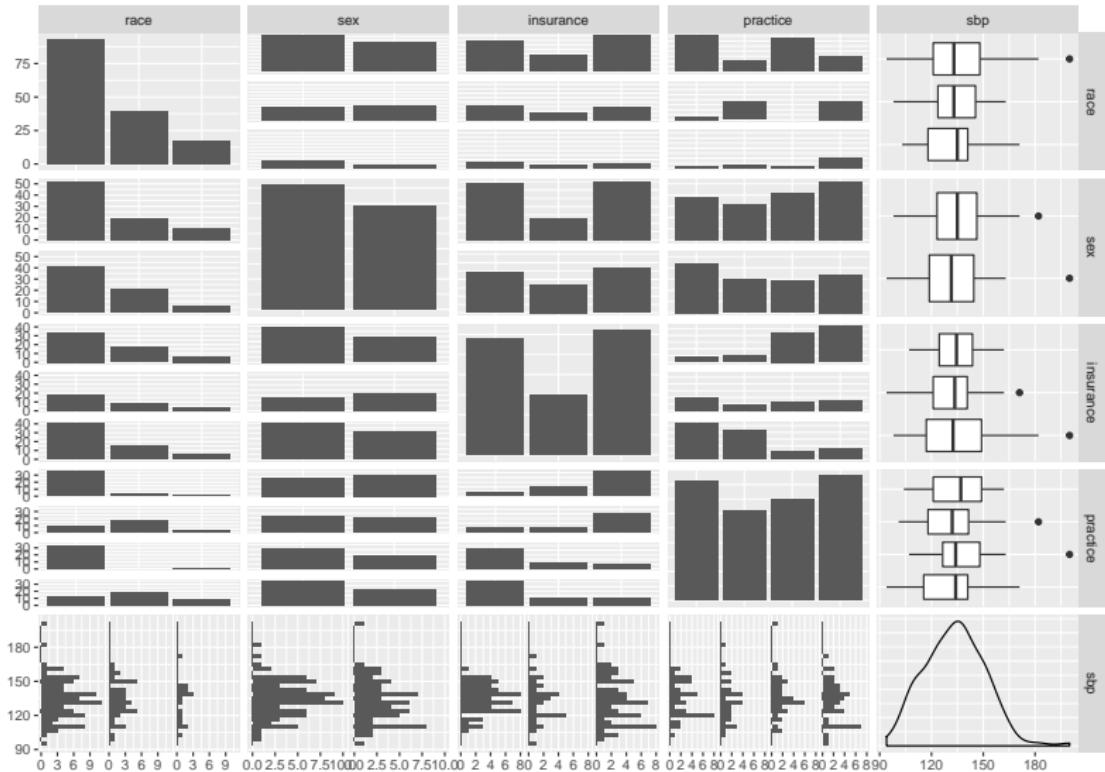
Stage 2. DTDP (everything in training set)



Stage 2. DTDP (quantitative predictors)



Stage 2. DTDP (categorical predictors)



Stage 3. Exploratory Data Analysis

```
mosaic::favstats(dm192_train$sbp)
```

	min	Q1	median	Q3	max	mean	sd	n
94	121.25		133	145.5	200	133.42	17.48605	150
missing								
0								

```
mosaic::favstats(dm192_train$sbp ~ dm192_train$sex)
```

	dm192_train\$sex	min	Q1	median	Q3	max
1	female	98	123.25	135.0	146.25	182
2	male	94	118.75	131.5	144.50	200
	mean	sd	n	missing		
1	134.6463	16.19966	82	0		
2	131.9412	18.93814	68	0		

Usually, I stop myself from doing this.

BRACE YOURSELVES



Time to fit a Kitchen Sink Model



Stage 4. Fit Kitchen Sink Model in Training Sample

```
mod_ks1 <- lm(sbp ~ sbp_old + a1c + age + race +  
                 sex + insurance + practice,  
                 data = dm192_train)  
  
round(glance(mod_ks1), 3)
```

	r.squared	adj.r.squared	sigma	statistic	p.value	df
1	0.898	0.89	5.791	110.954	0	12

	logLik	AIC	BIC	deviance	df.residual
1	-470.034	966.067	1005.206	4627.97	138

arm::display(mod_ks1) (n = 150, r-sq = 0.90)

```
lm(formula = sbp ~ sbp_old + a1c + age + race + sex + insurance  
    practice, data = dm192_train)
```

	coef.est	coef.se
(Intercept)	6.70	5.70
sbp_old	0.93	0.03
a1c	-0.09	0.21
age	0.02	0.08
racewhite	-1.16	1.39
raceOthers	-1.16	1.66
sexmale	-0.67	0.97
insuranceCommercial	1.59	1.40
insuranceMedicaid_Unins	1.85	1.38
practiceB	1.29	1.59
practiceC	2.27	1.73
practiceD	2.91	1.75

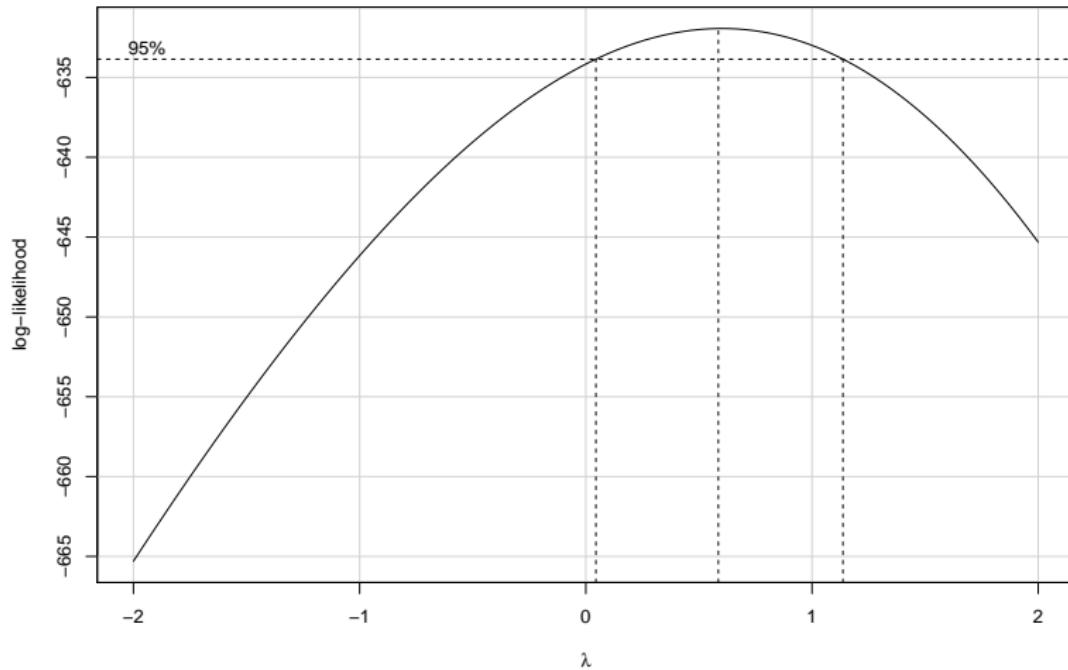
n = 150	12	

Stage 5. Consider collinearity, residual plots, potential transformations of the outcome

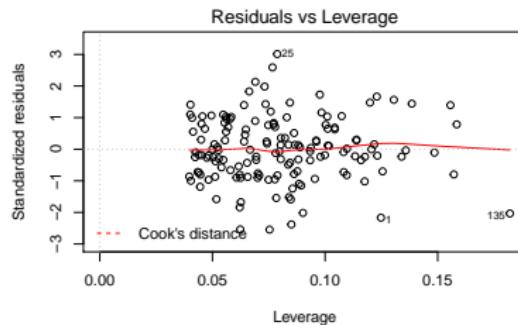
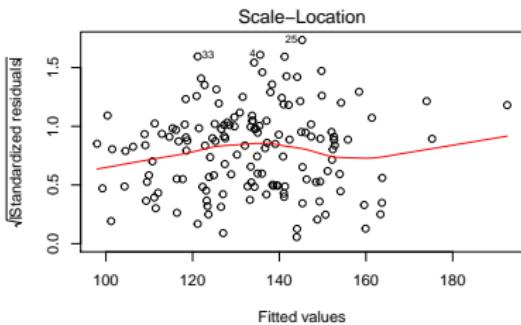
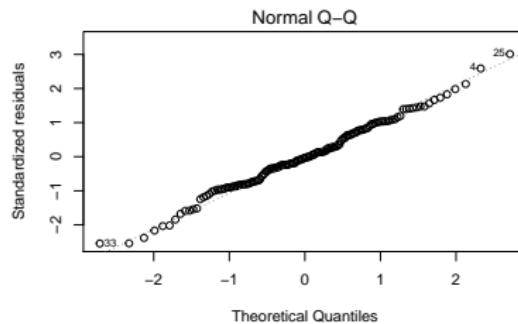
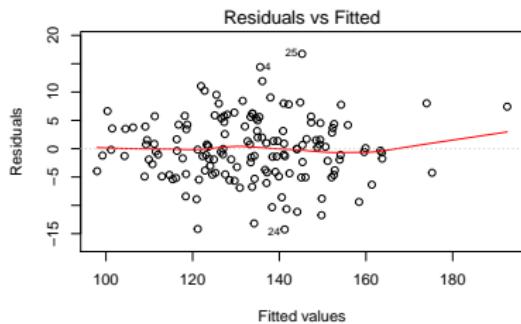
```
vif(mod_ks1)
```

	GVIF	Df	GVIF^(1/(2*Df))
sbp_old	1.082669	1	1.040514
a1c	1.075402	1	1.037016
age	2.645307	1	1.626440
race	1.745582	2	1.149437
sex	1.052235	1	1.025785
insurance	1.757957	2	1.151468
practice	4.032427	3	1.261618

boxCox(mod_ks1) ($\lambda = 0.6$, round to 1)



plot(mod_ks1)



Stage 6. Consider stepwise regression to prune the model

```
step(mod_ks1)
```

Start: AIC=538.39

sbp ~ sbp_old + a1c + age + race + sex + insurance + practice

	Df	Sum of Sq	RSS	AIC
- race	2	30	4658	535.35
- practice	3	98	4726	535.53
- age	1	1	4629	536.43
- a1c	1	7	4635	536.61
- insurance	2	69	4697	536.61
- sex	1	16	4644	536.89
<none>			4628	538.39
- sbp_old	1	38268	42896	870.39

Suggested model from step is

Step: AIC=524.98

sbp ~ sbp_old

	Df	Sum of Sq	RSS	AIC
<none>		4836	524.98	
- sbp_old	1	40722	45559	859.42

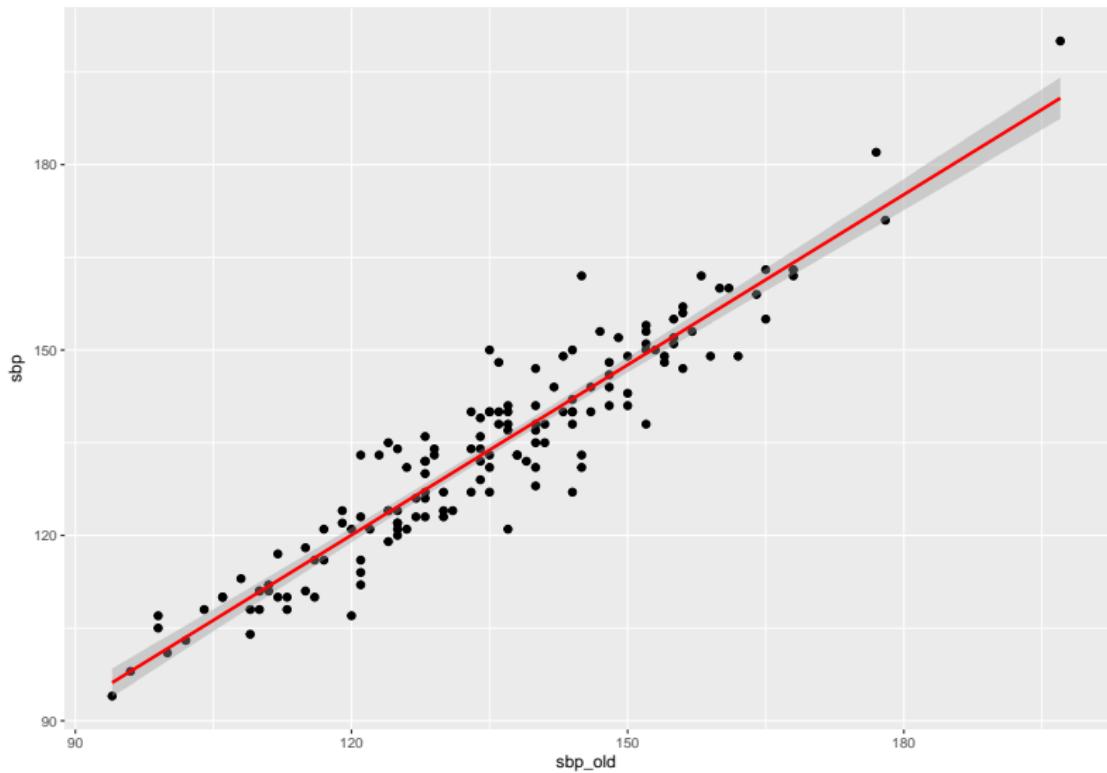
Call:

```
lm(formula = sbp ~ sbp_old, data = dm192_train)
```

Coefficients:

(Intercept)	sbp_old
9.8485	0.9183

So that's just . . .



Stage 7. Compare potential models in-sample

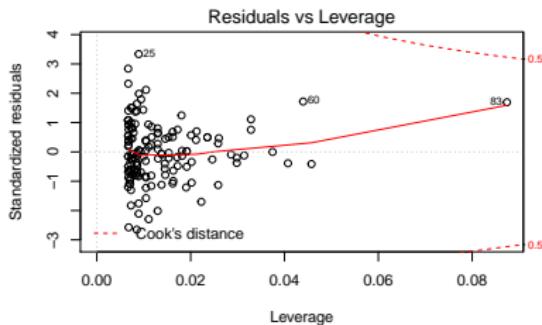
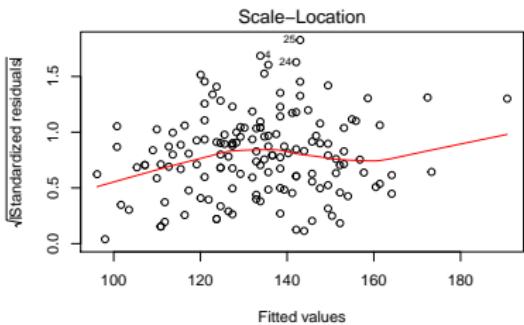
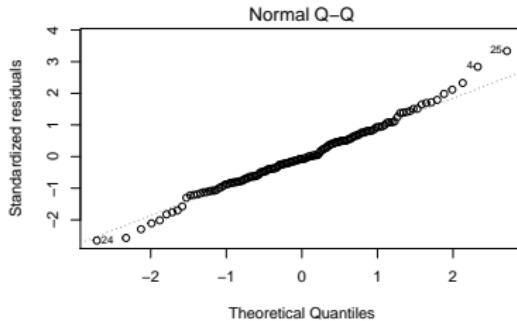
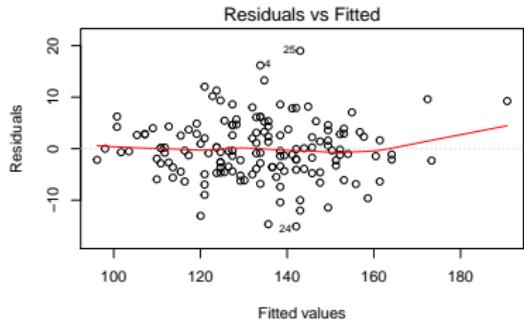
```
mod_simple <- lm(sbp ~ sbp_old, data = dm192_train)
glance(mod_simple) %>% select(r.squared, adj.r.squared, AIC, BIC)
```

	r.squared	adj.r.squared	AIC	BIC
1	0.8938499	0.8931326	952.6641	961.696

```
glance(mod_ks1) %>% select(r.squared, adj.r.squared, AIC, BIC)
```

	r.squared	adj.r.squared	AIC	BIC
1	0.8984171	0.8903199	966.0673	1005.206

Residual Plots for Simple One-Predictor Model



Stage 8. Compare potential models on test data

```
pred_ks <- predict(mod_ks1, newdata = dm192_test)
err_ks <- dm192_test$sbp - pred_ks
round(summary(abs(err_ks)),3)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.578	2.876	4.287	5.130	6.848	14.480

```
round(summary(err_ks^2),3)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.334	8.271	18.396	37.102	46.906	209.672

```
round(cor(pred_ks, dm192_test$sbp)^2,4)
```

[1] 0.9163

Simple Model

```
pred_simple <- predict(mod_simple, newdata = dm192_test)
err_simple <- dm192_test$sbp - pred_simple
round(summary(abs(err_simple)),3)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.151	2.334	4.298	4.994	6.298	14.278

```
round(summary(err_simple^2),3)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.023	5.468	18.478	37.965	39.677	203.851

```
round(cor(pred_simple, dm192_test$sbp)^2,4)
```

[1] 0.9149

MAPE and MSPE results

Model	MAPE	MSPE	Max Abs. Error	Out of Sample R ²
Kitchen Sink	5.13	37.1	14.48	0.9163
Simple	4.99	38	14.28	0.9149

Remember that the training sample here has only 38 observations.

Stage 9. Re-combine sample and fit final model

```
model_all <- lm(sbp ~ sbp_old, data = dm192_work)  
  
glance(model_all)
```

```
r.squared adj.r.squared      sigma statistic  
1 0.8962414     0.8956835 5.785442  1606.622  
    p.value df      logLik      AIC      BIC deviance  
1 1.907078e-93  2 -595.7599 1197.52 1207.229 6225.669  
df.residual  
1          186
```

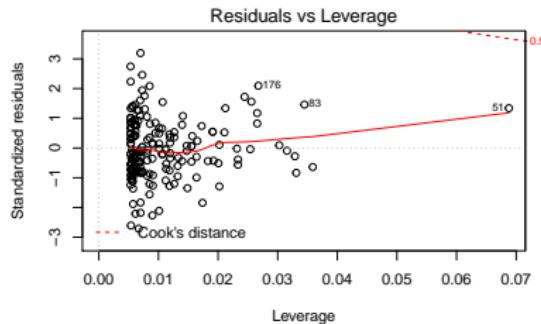
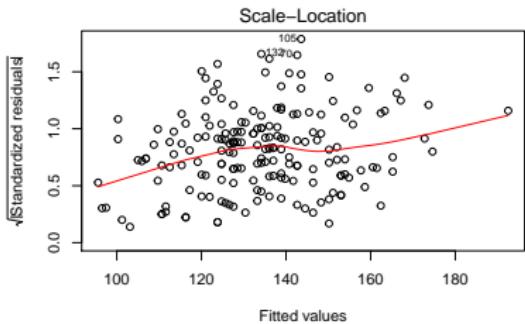
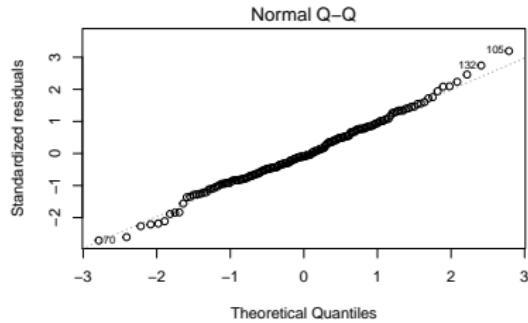
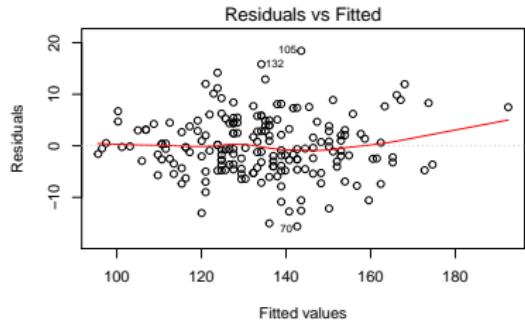
Tidied model_all Coefficients

```
tidy(model_all)
```

	term	estimate	std.error	statistic
1	(Intercept)	7.1213074	3.19565828	2.228432
2	sbp_old	0.9410682	0.02347817	40.082692

	p.value
1	2.704951e-02
2	1.907078e-93

Residual Plot for model_all



That's all, folks!

Please complete the Task after Class 26 form.



That's all Folks!