Lab 4: Multiple Linear Regression

Generalize simple linear regression to:

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

<u>Steps</u>

1. Visualize the data
2. Identify potential predictors
3. Apply *backward elimination* process
4. Perform residual analysis to check quality of the model

Visualize the data

- pairs(int00.dat, gap=0.5)
- Note nperf is normalized version of perf
  - 0 ≤ nperf < 100

Identify potential predictors

- Use smallest number of predictors necessary to make good predictions
- Too many or redundant predictors □ over-fits the model
  - Builds random noise into model
  - Perfect fit for that data set, but does not generalize
- More predictors always improves R^2
  - But not necessarily a better model
  - May simply be better at modeling the random noise
- *Must find balance between too many and too few predictors*

Adjusted R^2

$$R^2_{adjusted} = 1 - \frac{n-1}{n-m}(1 - R^2)$$

- n = # observations
- m = # estimated parameters

  = number of predictors in the model + 1
- Adjusted R^2 increases if
  - Adding a new predictor increases previous model's R^2 by more than we would expect from random fluctuations
  - If adjusted R^2 increases, new predictor improved the model
  - Adding new predictors is not always helpful
    - Adjusted R^2 will still the same or decrease

Identify potential predictors

- Start with all available predictors (columns)
- Use knowledge of the system to:
  - Eliminate predictors that are not meaningful:
    - E.g. Thermal design power (TDP), index number, …
    - Predictors with only a few entries – e.g. L3 cache
  - Add functions of existing predictors that could be useful
    - E.g. Previous research suggests CPU performance increases as the sqrt(cache size)
      - Add $a_m x_m^{1/2}$ terms
    - Still include first-degree terms

Backward elimination

1. Generate model with all potential predictors

   int00.lm <- lm(nperf ~ clock + threads + cores + transistors + …)

Backward elimination

1. Generate model with all potential predictors

   ```
   int00.lm <- lm(nperf ~ clock + threads + cores + transistors + …)
   ```

2. Use summary(int00.lm) to find each predictor’s p-value

Backward elimination

1. Generate model with all potential predictors

   int00.lm <- lm(nperf ~ clock + threads + cores + transistors + …)

2. Use summary(int00.lm) to find each predictor's p-value
3. Drop predictor_i with largest p-value that is > our threshold (typically p≤0.05 or 0.1) and recompute the model

   int00.lm <- update(int00.lm, .~. - predictor_i)

Backward elimination

1. Generate model with all potential predictors
   ```
   int00.lm <- lm(nperf ~ clock + threads + cores + transistors + ...)
   ```
2. Use summary(int00.lm) to find each predictor's p-value
3. Drop predictor_i with largest p-value that is > our threshold (typically p≤0.05 or 0.1) and recompute the model
   ```
   int00.lm.j <- update(int00.lm, .~. - predictor_i)
   ```
4. Repeat #2-3 until all p-values ≤ threshold

Backward Elimination Example

```
>int00.lm.full <- lm(nperf
~ clock + threads +
cores + transistors +
dieSize + voltage +
featureSize + channel +
FO4delay + L1icache +
sqrt(L1icache) +
L1dcache +
sqrt(L1dcache) +
L2cache +
sqrt(L2cache),
data=int00.dat)
```

Backward Elimination Example

```
>int00.lm.full <- lm(nperf ~ clock + threads + cores + transistors + dieSize + voltage + featureSize + channel + FO4delay + L1icache + sqrt(L1icache) + L1dcache + sqrt(L1dcache) + L2cache + sqrt(L2cache), data=int00.dat)

>summary(int00.lm.full)
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.108e+01  7.852e+01  -0.268  0.78927
clock           2.605e-02  1.671e-03  15.594  < 2e-16 ***
threads        -2.346e+00  2.089e+00  -1.123  0.26596
cores           2.246e+00  1.782e+00   1.260  0.21235
transistors    -5.580e-03  1.388e-02  -0.402  0.68897
dieSize         1.021e-02  1.746e-02   0.585  0.56084
voltage        -2.623e+01  7.698e+00  -3.408  0.00117 **
featureSize     3.101e+01  1.122e+02   0.276  0.78324
channel         9.496e+01  5.945e+02   0.160  0.87361
FO4delay       -1.765e-02  1.600e+00  -0.011  0.99123
L1icache        1.102e+02  4.206e+01   2.619  0.01111 *
sqrt(L1icache) -7.390e+02  2.980e+02  -2.480  0.01593 *
L1dcache       -1.114e+02  4.019e+01  -2.771  0.00739 **
sqrt(L1dcache)  7.492e+02  2.739e+02   2.735  0.00815 **
L2cache        -9.684e-03  1.745e-03  -5.550 6.57e-07 ***
sqrt(L2cache)   1.221e+00  2.425e-01   5.034 4.54e-06 ***
---
Residual standard error: 4.632 on 61 degrees of freedom
  (179 observations deleted due to missingness)
Multiple R-squared:  0.9652,     Adjusted R-squared:  0.9566
F-statistic: 112.8 on 15 and 61 DF,  p-value: < 2.2e-16
```

Backward Elimination Example

```
>int00.lm.full <- lm(nperf ~ clock + threads + cores + transistors + dieSize + voltage + featureSize + channel + FO4delay + L1icache + sqrt(L1icache) + L1dcache + sqrt(L1dcache) + L2cache + sqrt(L2cache), data=int00.dat)

>summary(int00.lm.full)
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.108e+01  7.852e+01  -0.268  0.78927
clock           2.605e-02  1.671e-03  15.594  < 2e-16 ***
threads        -2.346e+00  2.089e+00  -1.123  0.26596
cores           2.246e+00  1.782e+00   1.260  0.21235
transistors    -5.580e-03  1.388e-02  -0.402  0.68897
dieSize         1.021e-02  1.746e-02   0.585  0.56084
voltage        -2.623e+01  7.698e+00  -3.408  0.00117 **
featureSize     3.101e+01  1.122e+02   0.276  0.78324
channel         9.496e+01  5.945e+02   0.160  0.87361
FO4delay       -1.765e-02  1.600e+00  -0.011  0.99123
L1icache        1.102e+02  4.206e+01   2.619  0.01111 *
sqrt(L1icache) -7.390e+02  2.980e+02  -2.480  0.01593 *
L1dcache       -1.114e+02  4.019e+01  -2.771  0.00739 **
sqrt(L1dcache)  7.492e+02  2.739e+02   2.735  0.00815 **
L2cache        -9.684e-03  1.745e-03  -5.550 6.57e-07 ***
sqrt(L2cache)   1.221e+00  2.425e-01   5.034 4.54e-06 ***
---
Residual standard error: 4.632 on 61 degrees of freedom
  (179 observations deleted due to missingness)
Multiple R-squared:  0.9652,     Adjusted R-squared:  0.9566
F-statistic: 112.8 on 15 and 61 DF,  p-value: < 2.2e-16
```

Backward Elimination Example

```
>int00.lm.2 <- update(int00.lm.full, .~. - FO4delay, data = int00.dat)

>summary(int00.lm.2)
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.088e+01  7.584e+01  -0.275 0.783983
clock           2.604e-02  1.563e-03  16.662  < 2e-16 ***
threads        -2.345e+00  2.070e+00  -1.133 0.261641
cores           2.248e+00  1.759e+00   1.278 0.206080
transistors    -5.556e-03  1.359e-02  -0.409 0.684020
dieSize         1.013e-02  1.571e-02   0.645 0.521488
voltage        -2.626e+01  7.302e+00  -3.596 0.000642 ***
featureSize     3.104e+01  1.113e+02   0.279 0.781232
channel         8.855e+01  1.218e+02   0.727 0.469815
L1icache        1.103e+02  4.041e+01   2.729 0.008257 **
sqrt(L1icache) -7.398e+02  2.866e+02  -2.581 0.012230 *
L1dcache       -1.115e+02  3.859e+01  -2.889 0.005311 **
sqrt(L1dcache)  7.500e+02  2.632e+02   2.849 0.005937 **
L2cache        -9.693e-03  1.494e-03  -6.488 1.64e-08 ***
sqrt(L2cache)   1.222e+00  1.975e-01   6.189 5.33e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.594 on 62 degrees of freedom
  (179 observations deleted due to missingness)
Multiple R-squared:  0.9652,     Adjusted R-squared:  0.9573
F-statistic: 122.8 on 14 and 62 DF,  p-value: < 2.2e-16
```

Backward Elimination Example

>int00.lm.3 <- update(int00.lm.2, .~. - featureSize, data=int00.dat)

>summary(int00.lm.3)

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -3.129e+01  6.554e+01  -0.477 0.634666
clock            2.591e-02  1.471e-03  17.609  < 2e-16 ***
threads         -2.447e+00  2.022e+00  -1.210 0.230755
cores            1.901e+00  1.233e+00   1.541 0.128305
transistors     -5.366e-03  1.347e-02  -0.398 0.691700
dieSize          1.325e-02  1.097e-02   1.208 0.231608
voltage         -2.519e+01  6.182e+00  -4.075 0.000131 ***
channel          1.188e+02  5.504e+01   2.158 0.034735 *
L1icache         1.037e+02  3.255e+01   3.186 0.002246 **
sqrt(L1icache)  -6.930e+02  2.307e+02  -3.004 0.003818 **
L1dcache        -1.052e+02  3.106e+01  -3.387 0.001223 **
sqrt(L1dcache)   7.069e+02  2.116e+02   3.341 0.001406 **
L2cache         -9.548e-03  1.390e-03  -6.870 3.37e-09 ***
sqrt(L2cache)    1.202e+00  1.821e-01   6.598 9.96e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.56 on 63 degrees of freedom
  (179 observations deleted due to missingness)
Multiple R-squared:  0.9651,     Adjusted R-squared:  0.958
F-statistic: 134.2 on 13 and 63 DF,  p-value: < 2.2e-16
```

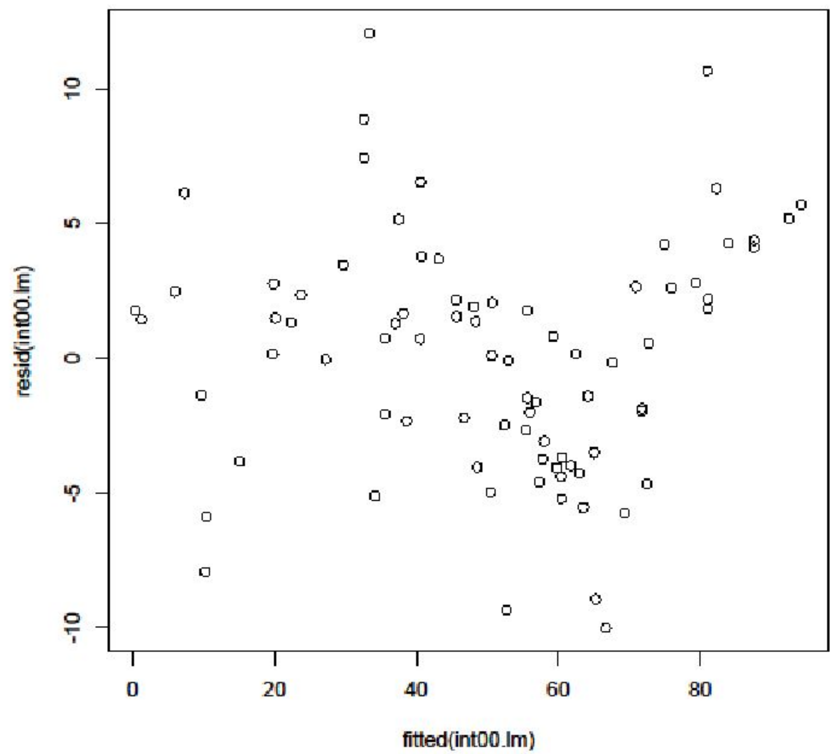
Backward Elimination Example

- Continue to iterate on models by removing predictors until all predictors have p-values <= your threshold (or pretty close to the threshold)
- Final model for int00:

$$
\begin{aligned}
\text{nperf} = & -58.22 + 0.02482*\text{clock} \\
& + 2.397*\text{cores} \\
& - 23.58*\text{voltage} \\
& + 139.9*\text{channel} \\
& + 87.03*\text{L1icache} \\
& - 576.8*\mathrm{sqrt}(\text{L1icache}) \\
& - 89.03*\text{L1dcache} \\
& + 598*\mathrm{sqrt}(\text{L1dcache}) \\
& - 0.008621*\text{L2cache} \\
& + 1.085*\mathrm{sqrt}(\text{L2cache})
\end{aligned}
$$

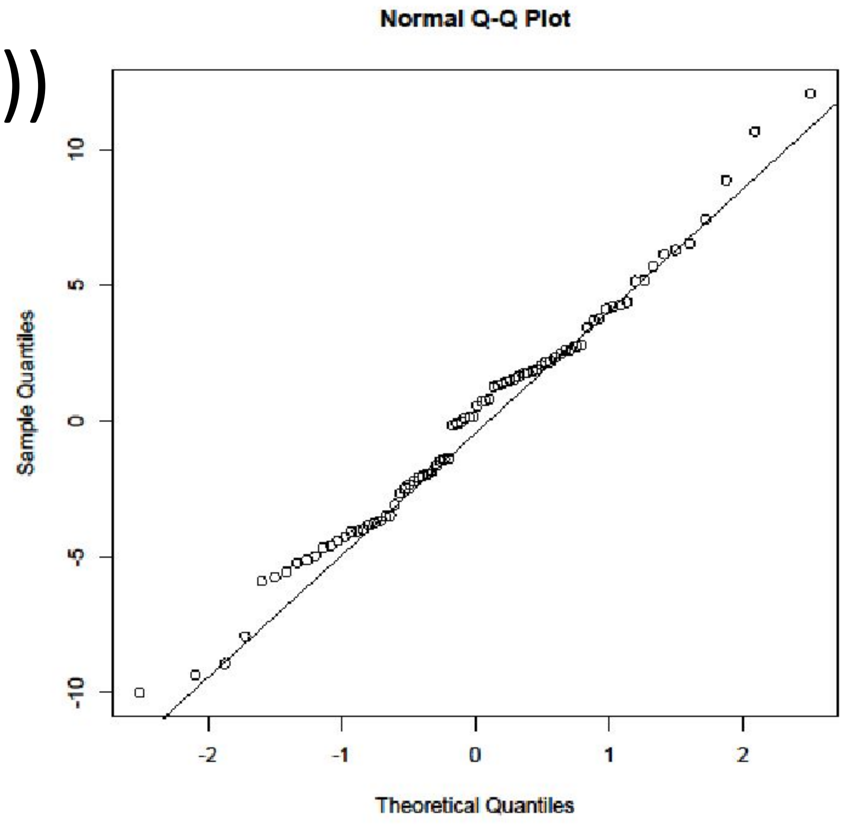
Residual analysis

- The same as for a simple linear model
- plot(fitted(int00.lm), resid(int00.lm))
- Expect to see values uniformly distributed around 0



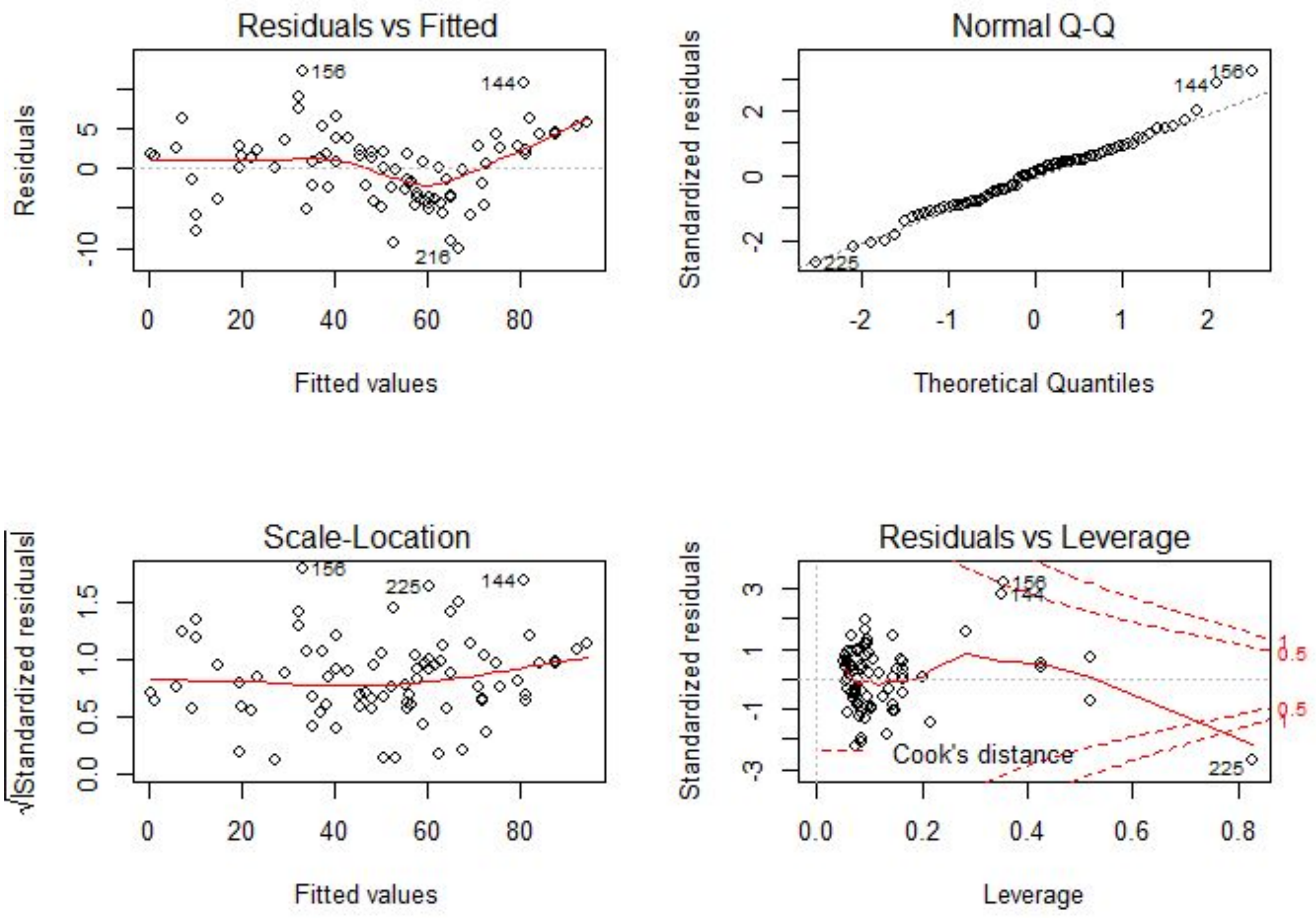
Residual analysis – QQ plot

- qqnorm(resid(int00.lm))
- qqline(resid(int00.lm))
- Expect to see values follow the line.



Diagnostic plots

```
par(mfrow=c(2,2))
plot(int00.lm.6)
```



A Little Deeper on p-values

- H_0 = Null hypothesis: that the given coefficient is equal to 0
  - i.e., it is not significant to the model
- Type I Error: rejecting the null hypothesis when it is, in fact, true.
  - i.e., reject a coefficient when it is actually significant
  - Want this probability to be small
- If $p \le \alpha$ □ reject null hypothesis
  - Small p value □ predictor is likely to be meaningful
  - α is called the *significance level* – you choose this value

When Things Go Wrong

- As before, minimize SSE
  - Set partial derivatives to zero

All input factors that were measured.

$$X = \begin{matrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{matrix}$$

$$A = X^T X$$

Must be invertible; else see Sec. 4.6.

$$b = \begin{matrix} b_0 \\ b_1 \\ \cdots \\ b_k \end{matrix}$$

$$d = \begin{matrix} \sum y_i \\ \sum x_{1i} y_i \\ \cdots \\ \sum x_{ki} y_i \end{matrix}$$

$$Ab = d$$

$$b = A^{-1} d$$

Regression coefficients.

Example: int92 (Sec. 4.6)

- All predictors □ mostly NA
  - “14 [coefficients] not defined because of singularities”
  - “72 observations deleted due to missingness”
  - Only 4 degrees of freedom available!
- Use table() to find anomalous data
  - table(int92.dat$threads) – all the same = 1 thread
  - table(int92.dat$cores) – all the same = 1 core
  - table(int92.dat$L2cache) – only 3 unique values
- Not enough unique values (observations) in these columns to compute all of the coefficients
  - *Variables must vary!*
- Drop threads, cores, L2cache, sqrt(L2cache)
  - Then start backward elimination process

To do

- Read Chapter 4
- Download and complete Lab 4
- Follow the example in Section 4.4
- Note how the R^2, adjusted R^2, and degrees of freedom change as predictors are dropped from the model
- Especially note Section 4.6
  - When things go wrong