Introduction to Machine Learning – Linear Regression and KNN

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

- Finish more considerations for regression modeling
    - Potential problems
- Comparing LR and KNN Regression

Warm Up: Breaking Linear Regression

- Potential issues
  1. Correlated error terms
  2. Non-constant variance of error terms
  3. Outliers
  4. High leverage points
  5. Collinearity

How do we check for correlated error terms? Where do we see them most often?

How do we check for non-constant variance of error terms? How can we ”fix” them?

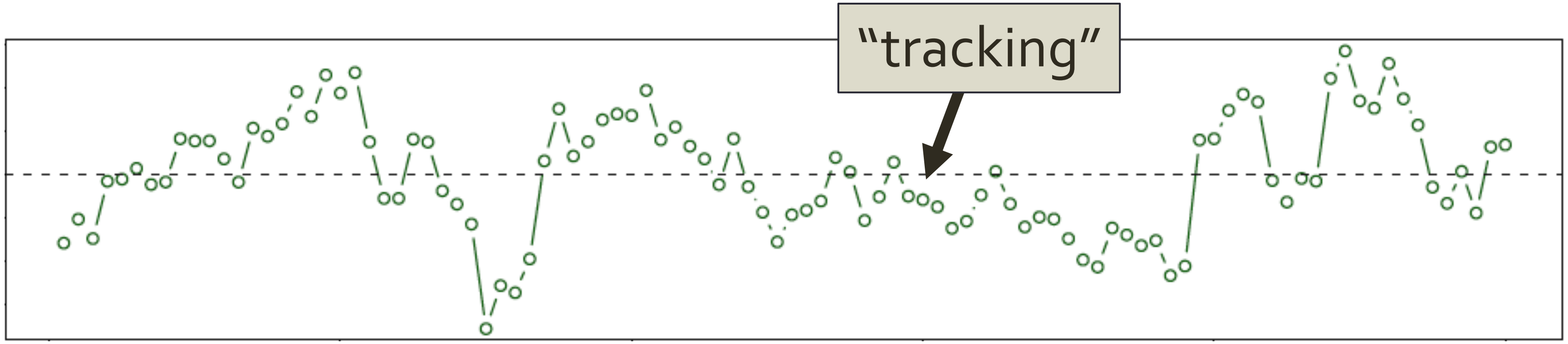
Correlated Error Terms

- LR assumes that the error terms are **uncorrelated**
- If these terms *are* correlated, the estimated standard error will tend to **underestimate** the true standard error. As a result,
  - CI's will be narrower than they should be and
  - p-values will be lower than they should be

Correlated Error Terms

Checking for correlated error terms in time-series data

- Plot residuals as a function of time
- If error are uncorrelated, there will be no discernable pattern
- If errors are correlated, we will see *tracking* (adjacent residuals with similar values)



Non-constant variance of error terms

- LR assumes that error terms have constant variance:

$$Var(e_i) = \sigma^2$$

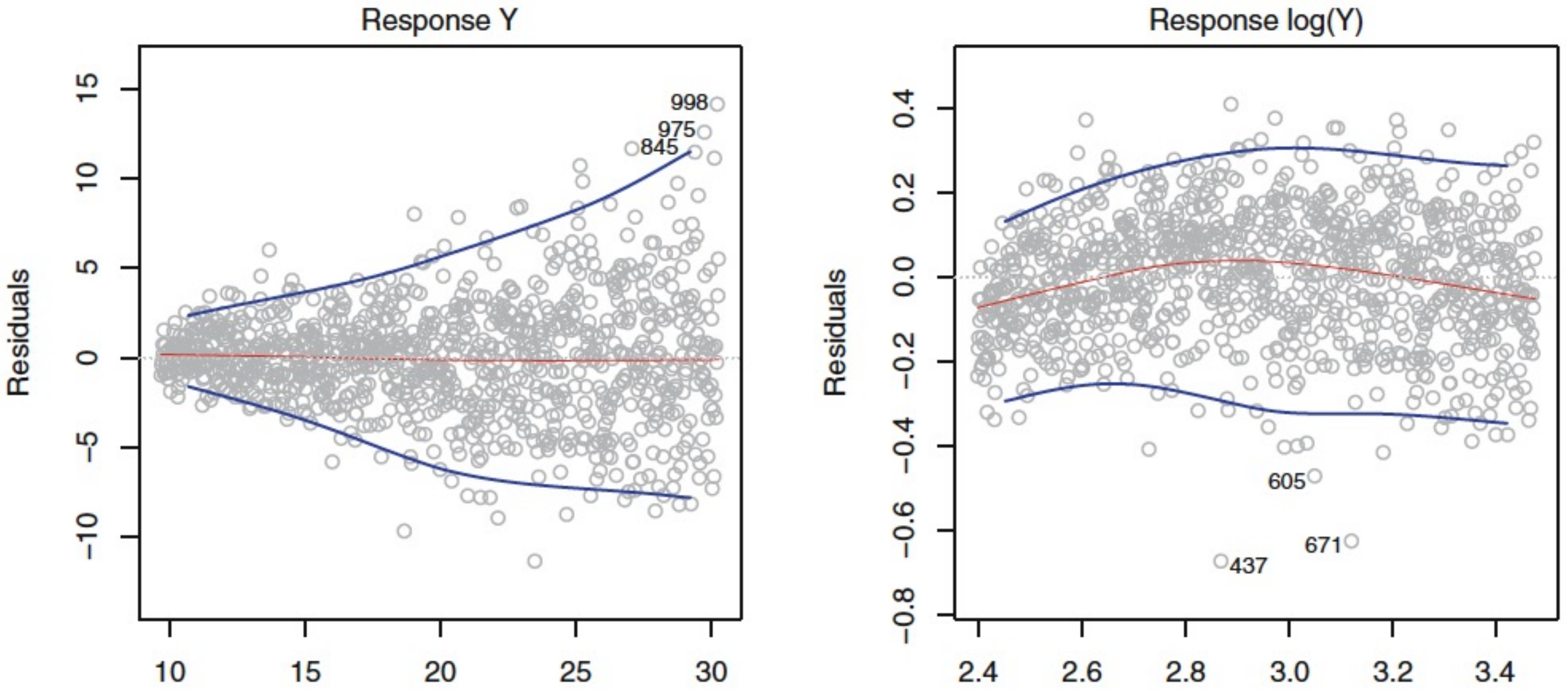
- SE's, CI's, and hypothesis tests rely on this assumption

- Often not the case (e.g. error terms might increase with the value of the response)

- Non-constant variance in errors is called ***heteroscedasticity***

Identify and Fix Heteroscedasticy

- Identifying: The residuals plot will show a funnel shape



- Fixing:
  - transform the response using a **concave function** (like *log* or *sqrt*)
  - **weight** the observations proportional to the inverse variance

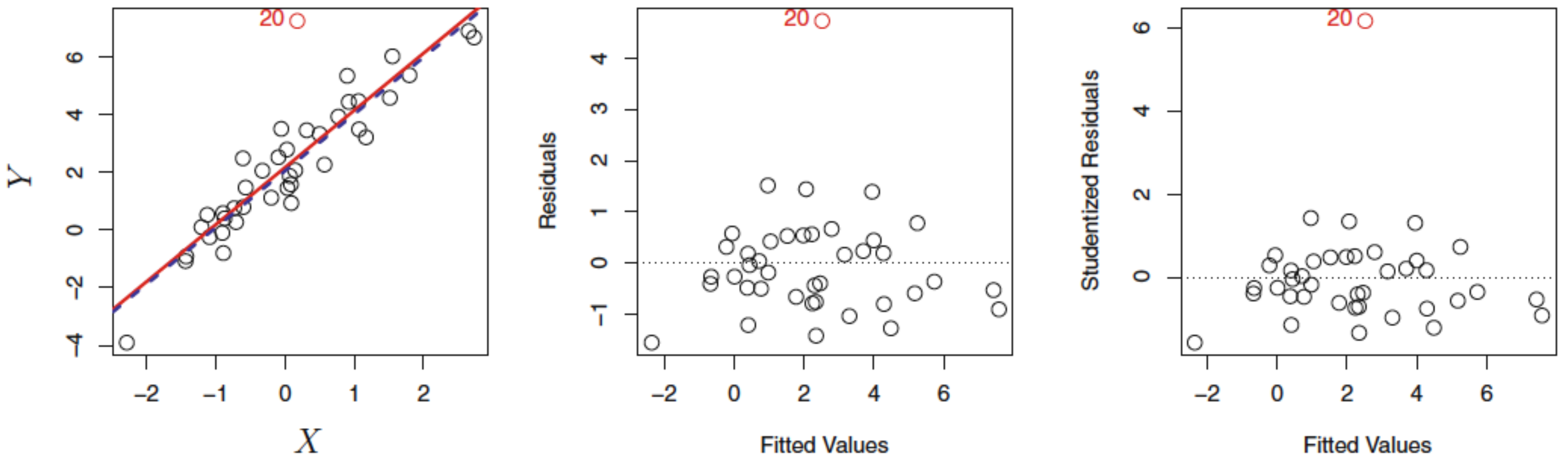
Outliers

An ***outlier*** is an observation whose true response is REALLY FAR from the one predicted by the model

- Sometimes indicate a problem with the model (i.e. a missing predictor), or might just be a data collection error
- Can mess with RSE and R^2, which can lead us to misinterpret the model's fit

Identify and Fix Outliers

Identify: Residual plots can help identify outliers, but sometimes it’s hard to pick a cutoff point (how far is “too far”?)



Fix:

- Divide each residual by dividing by its estimated standard error (*studentized residuals*), and flag anything larger than 3 in absolute value

High Leverage Points

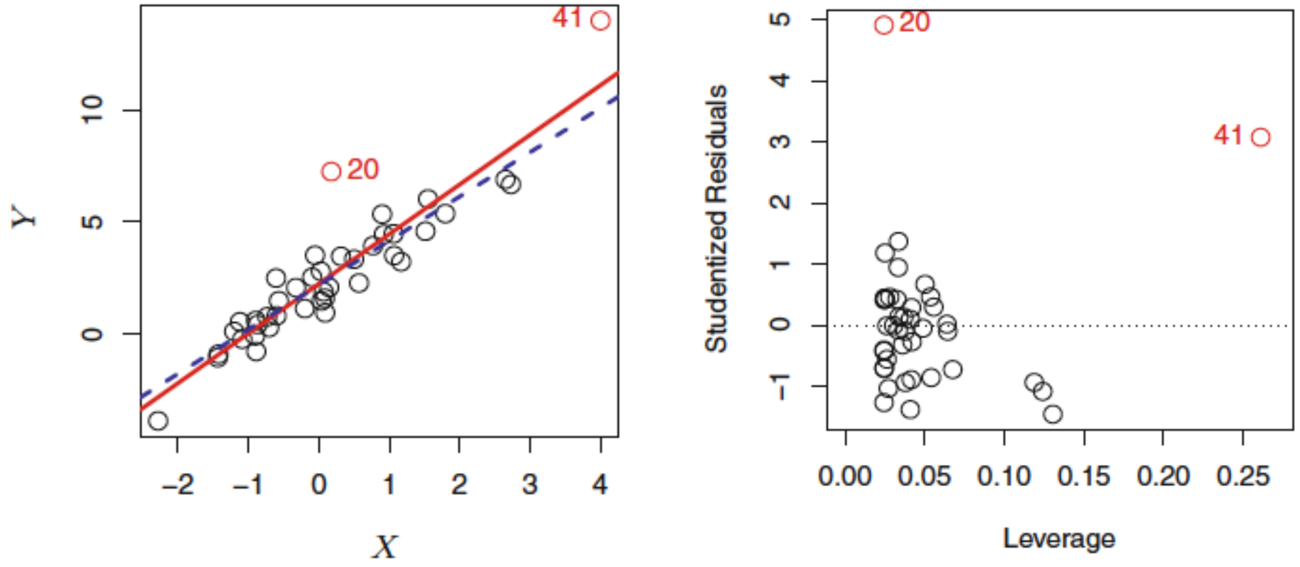
- Outliers are unusual values in the response
- ***High leverage points are*** unusual values in the predictor(s)
    - The more predictors you have, the harder they can be to spot
    - These points can have a major impact on the least squares line, which could invalidate the entire fit
        - We don't want only one or a few inputs to cause large changes in the entire model

Identify High Leverage Points

Compute the leverage statistic. For SLR:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- The leverage statistic is always a value between $\frac{1}{n}$ and n
- The average for all observations is $\frac{p+1}{n}$, if a statistic is much greater than the average, the point is probably a high leverage point



Collinearity

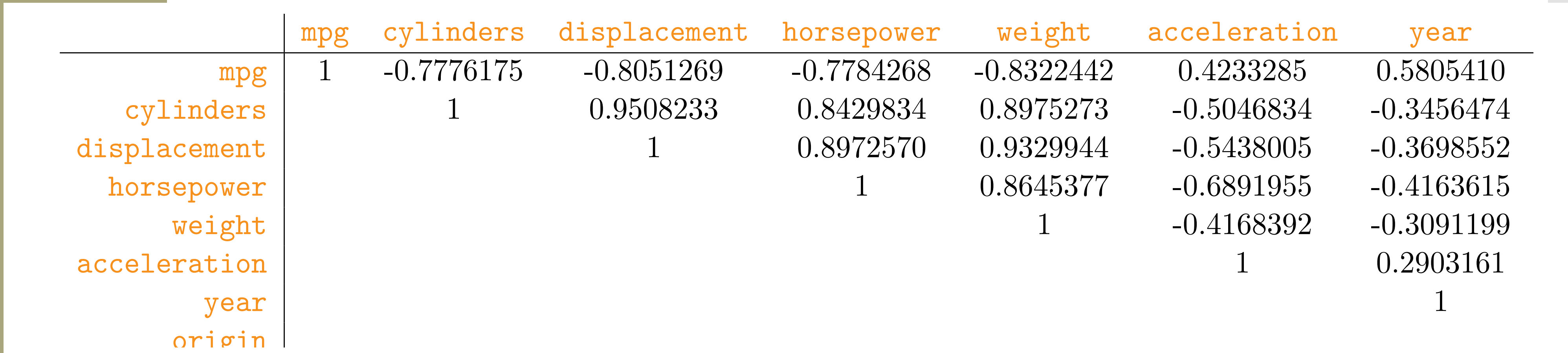
Collinearity is when two or more predictor variables are closely related to one another

- This makes it hard to isolate the individual effects of each predictor, which increases uncertainty in coefficient estimates

- As a result, it is harder to detect whether or not an effect is actually present (because SE has increased)

Identify Collinearity

- Look at the correlation matrix of the predictors
- **`Auto`** dataset: just about everything is highly correlated



	mpg	cylinders	displacement	horsepower	weight	acceleration	year
mpg	1	-0.7776175	-0.8051269	-0.7784268	-0.8322442	0.4233285	0.5805410
cylinders		1	0.9508233	0.8429834	0.8975273	-0.5046834	-0.3456474
displacement			1	0.8972570	0.9329944	-0.5438005	-0.3698552
horsepower				1	0.8645377	-0.6891955	-0.4163615
weight					1	-0.4168392	-0.3091199
acceleration						1	0.2903161
year							1
origin							

- **Note**: ***multicollinearity*** is when more than two variables are correlated; this will not show in this correlation matrix.

Dealing with Collinearity

Options include:

- Drop one of the problematic variables from the regression (collinearity implies they're redundant)
- Combine collinear variables into a single predictor (ex. take the average)

Linear Regression and KNN Regression

Parametric vs. Non Parametric

- Linear Regression is a ***parametric approach*** to modeling
    - It assumes a linear function, $f(X)$

- Parametric models
    - Are easy to fit (there are few coefficients to estimate)
    - (For LR) coefficients have simple interpretations and tests of statistical significance are easy to perform

Parametric vs. Non Parametric

- Linear Regression is a ***parametric approach*** to modeling
  - It assumes a linear function, $f(X)$

- Parametric models
  - Are easy to fit (there are few coefficients to estimate)
  - (For LR) coefficients have simple interpretations and tests of statistical significance are easy to perform

  - However…
    - They make strong assumptions about the form of $f(X)$, and if reality is far from this form predictions will be very poor

Parametric vs. Non Parametric

- Non-parametric models
  - No not explicitly assume a parametric form for $f(X)$, allowing for more flexibility in regression

- A common approach is K-nearest neighbors regression (KNN regression)

KNN Regression

- Given a value for K and a prediction point, x_0, KNN regression will
  1. identify the K training observations that are closest to x_0 (we will call these N_0)
  2. estimate $f(x_0)$ using the average of all training responses in N_0

KNN Regression

- Given a value for K and a prediction point, x_0, KNN regression will
  1. identify the K training observations that are closest to x_0 (we will call these N_0)
  2. estimate $f(x_0)$ using the average of all training responses in N_0

In other words…

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

KNN Regression

- Given a value for K and a prediction point, x_0, KNN regression will
  1. identify the K training observations that are closest to x_0 (we will call these N_0)
  2. estimate $f(x_0)$ using the average of all training responses in N_0

In other words…

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

KNN Regression

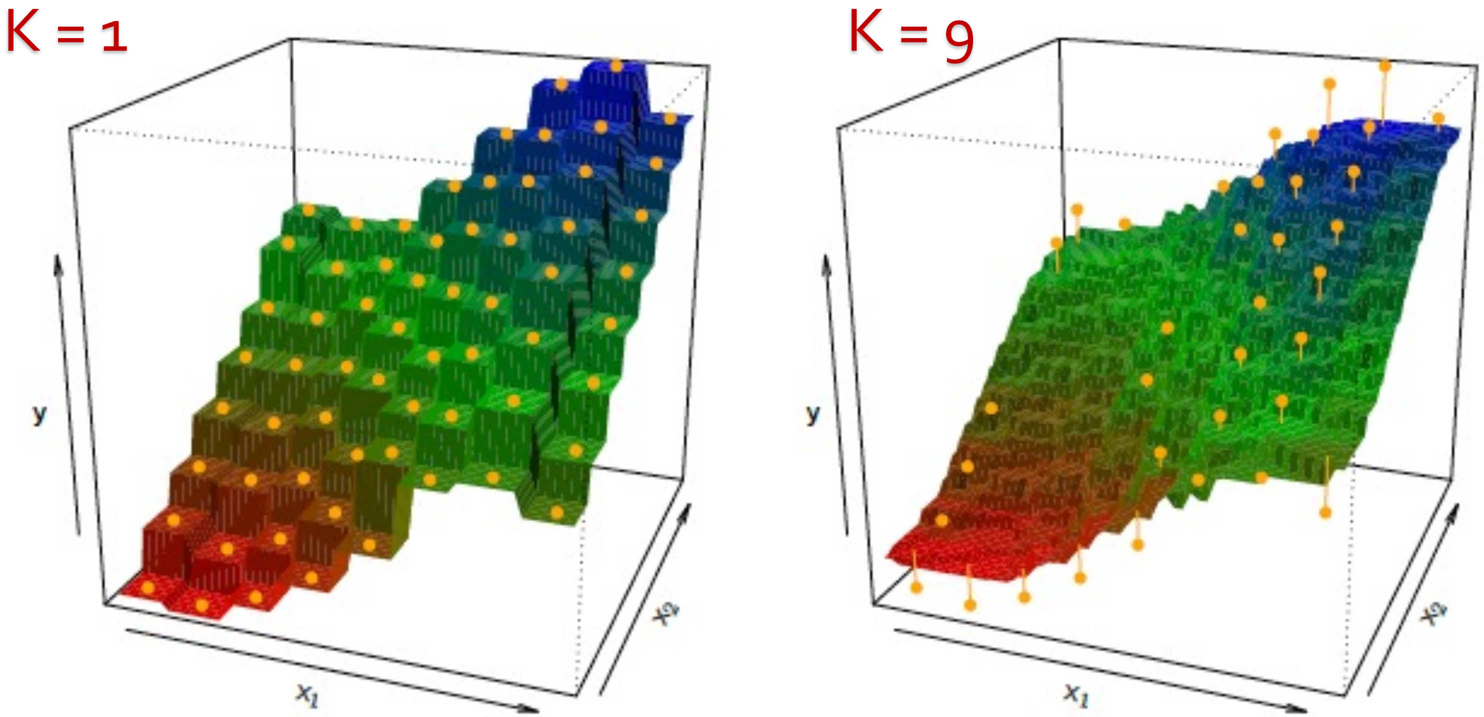
$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

So, how do we choose K?

KNN Regression

What can you say about bias and variance for the two plots below? Do they seem to relate to K? If so, how?

So, how do we choose K?



K = 1

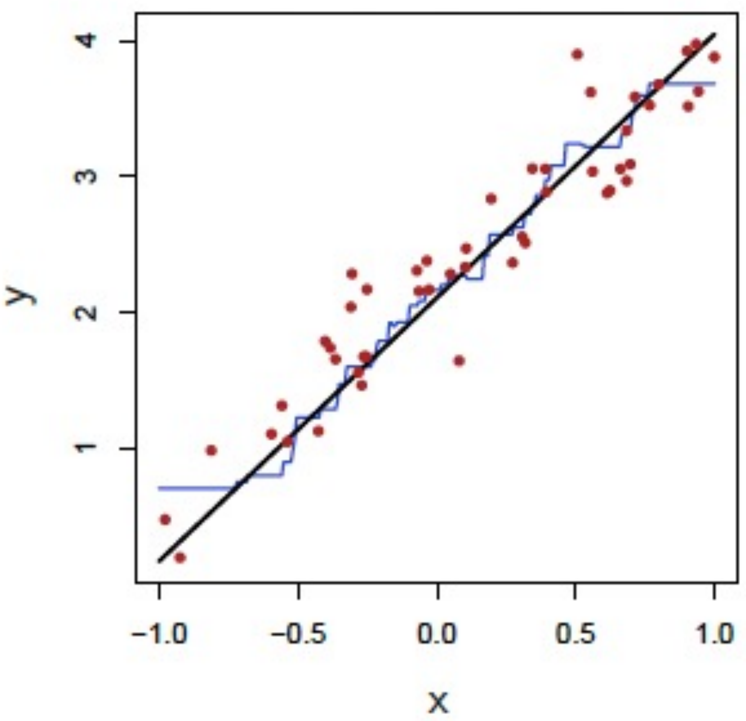
K = 9

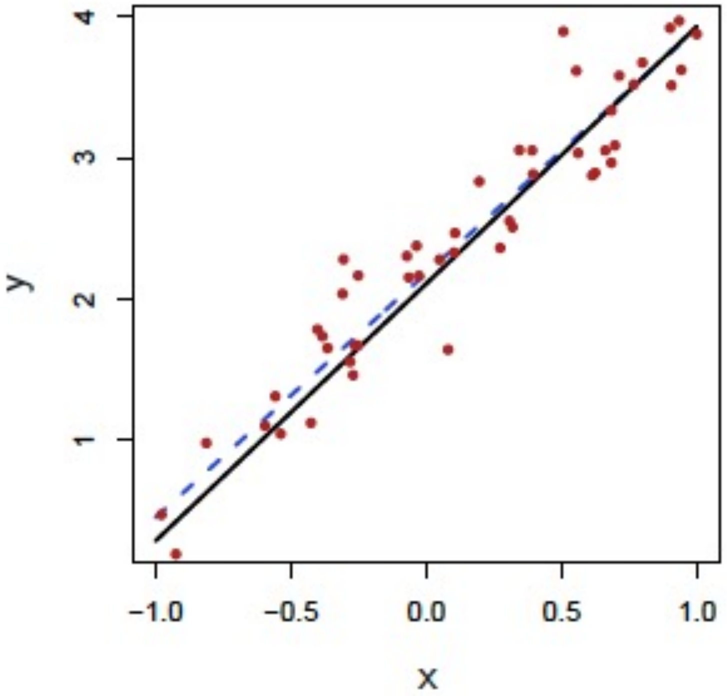
LR vs. KNN Regression

When is parametric better than non-parametric? And vice versa?

LR vs. KNN Regression

When is parametric better than non-parametric? And vice versa?



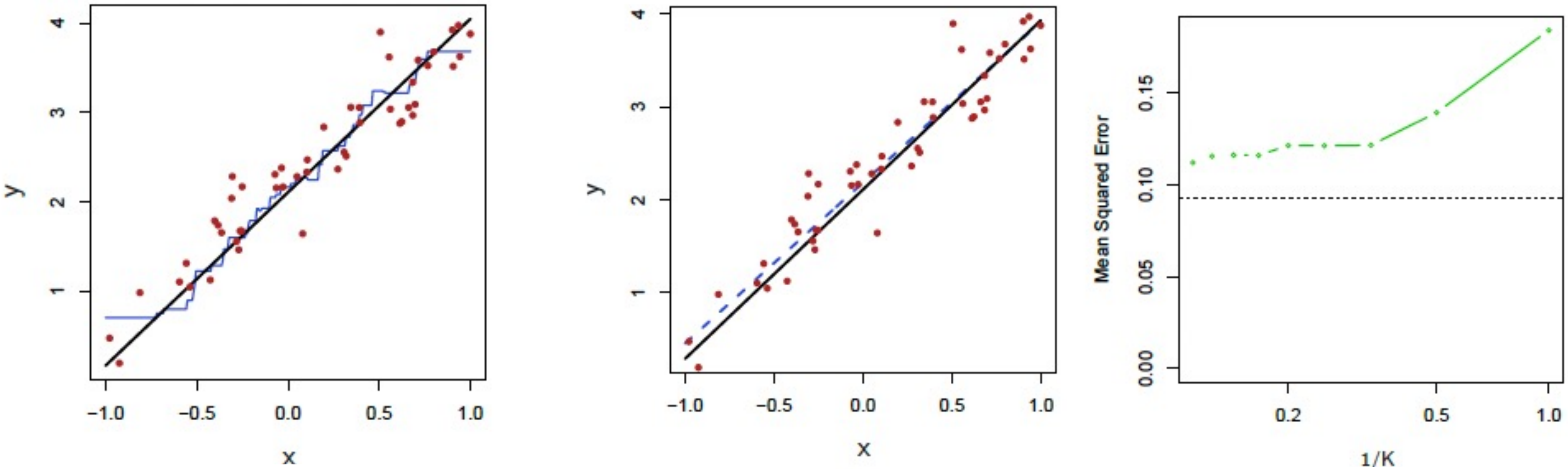


True relationship = black solid line

Left: blue curve is KNN regression with K = 9

Right: blue curve is LR regression

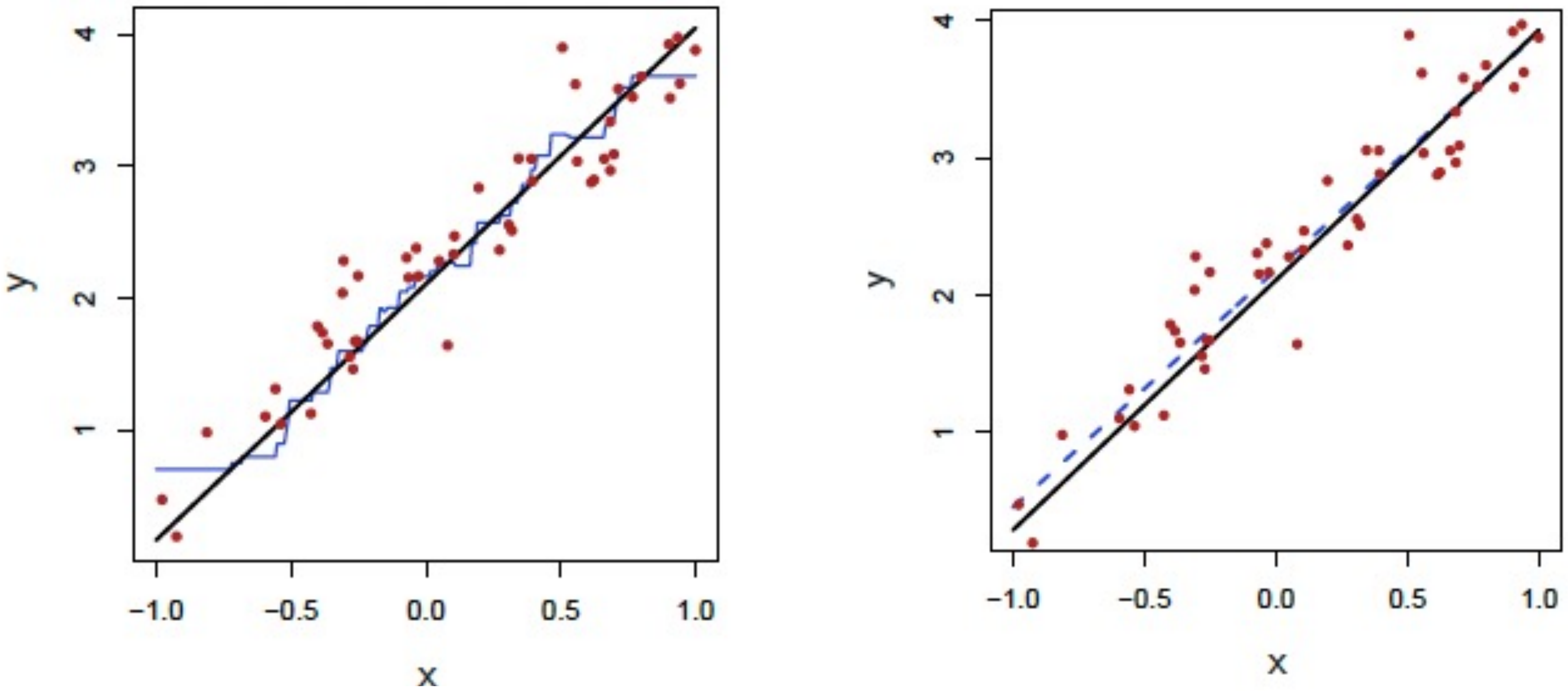
When is parametric better than non-parametric? And vice versa?



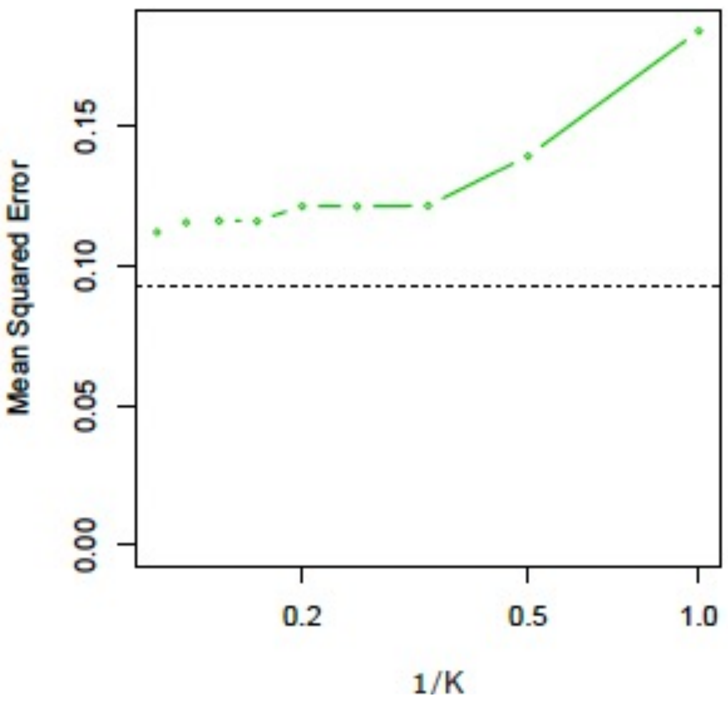
True relationship = black solid line
Left: blue curve is KNN regression with K = 9
Right: blue curve is LR regression

Dashed black = MSE for LR
Dashed green = MSE for KNN as a function of 1/K

What happens to MSE as K increases?
Which fit is better?



True relationship = black solid line
Left: blue curve is KNN regression with K = 9
Right: blue curve is LR regression

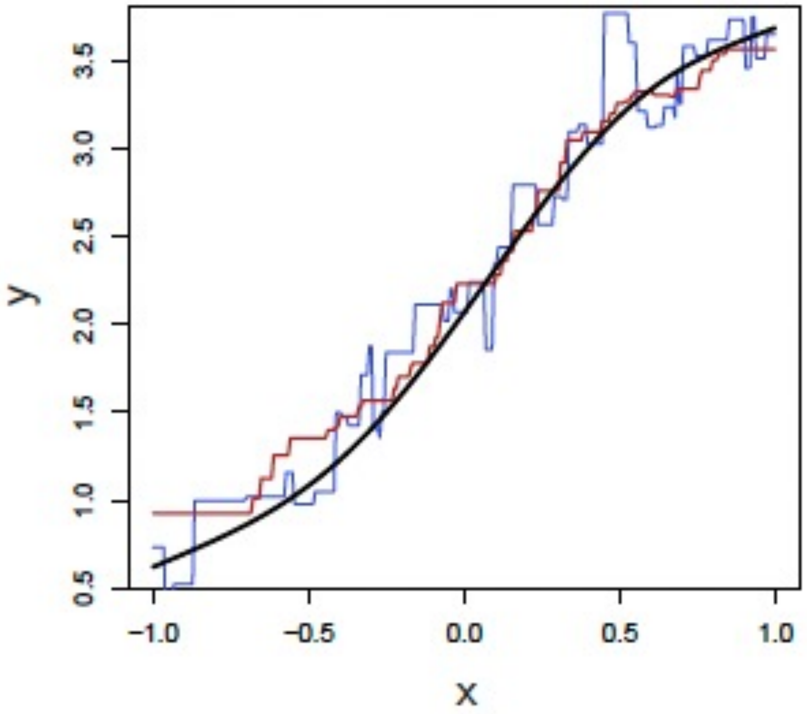


Dashed black = MSE for LR
Dashed green = MSE for
KNN as a function of 1/K

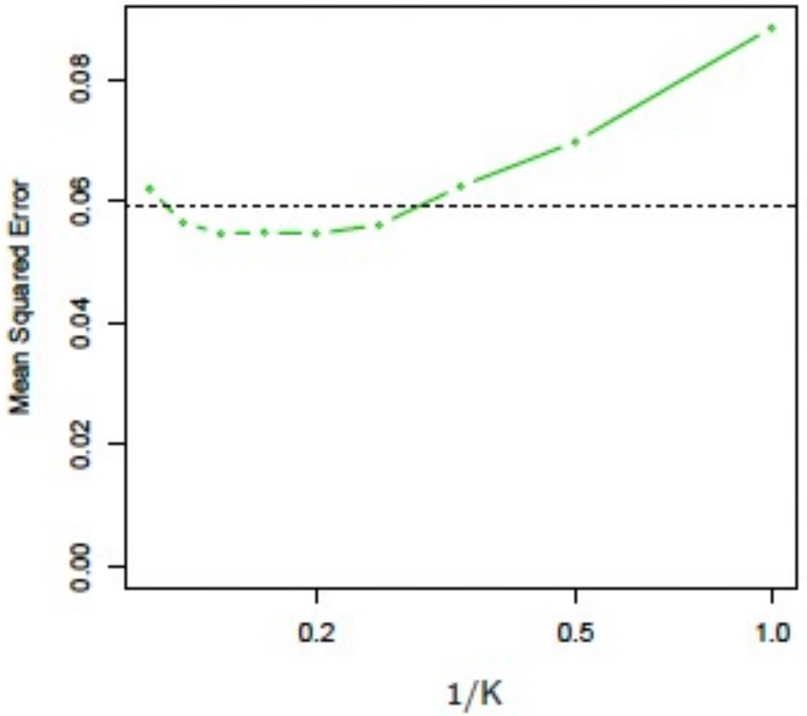
LR vs. KNN Regression

What happens to MSE as K increases?
Which fit is better?

versa:



True relationship = black solid line
Blue curve is KNN regression with K = 1
Red curse is KNN regression with K = 9

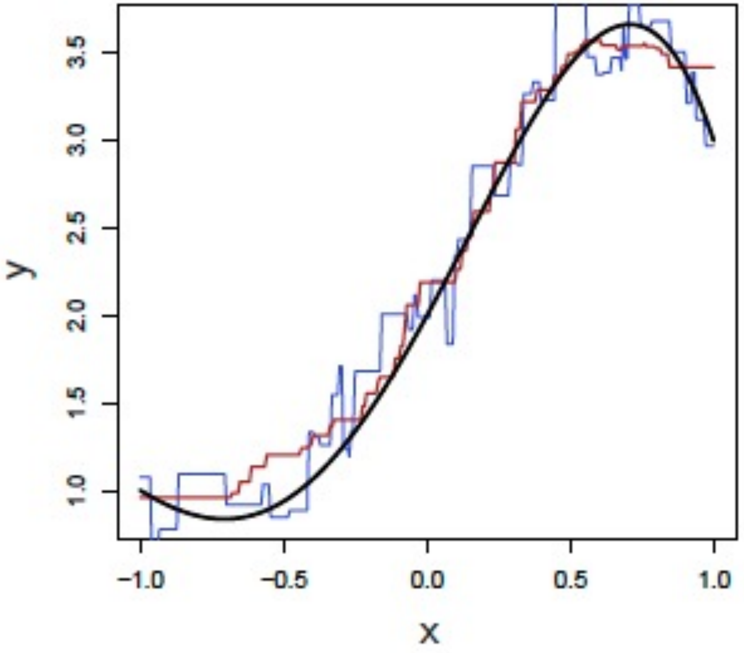


Dashed black = MSE for LR
Dashed green = MSE for
KNN as a function of 1/K

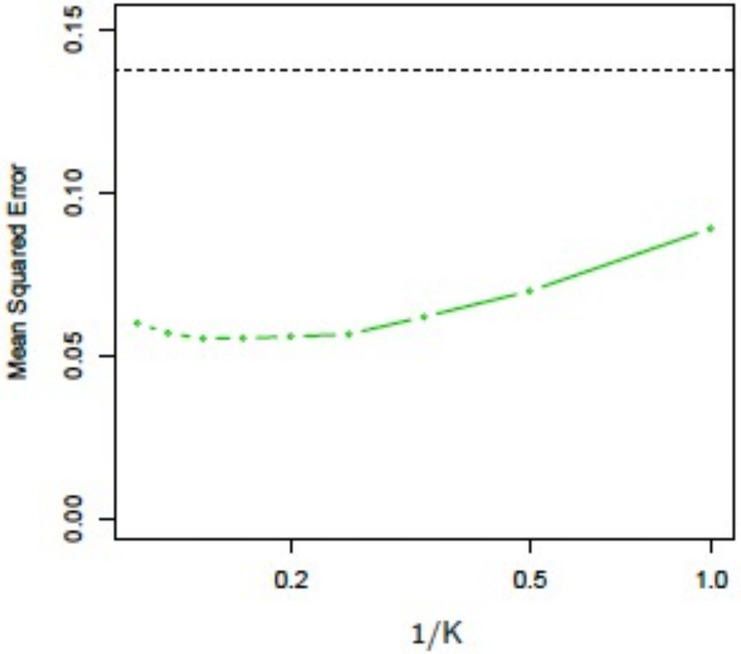
LR vs. KNN Regression

What happens to MSE as K increases?
Which fit is better?

versa.



True relationship = black solid line
Blue curve is KNN regression with K = 1
Red curse is KNN regression with K = 9



Dashed black = MSE for LR
Dashed green = MSE for KNN as a function of 1/K

LR vs. KNN Regression

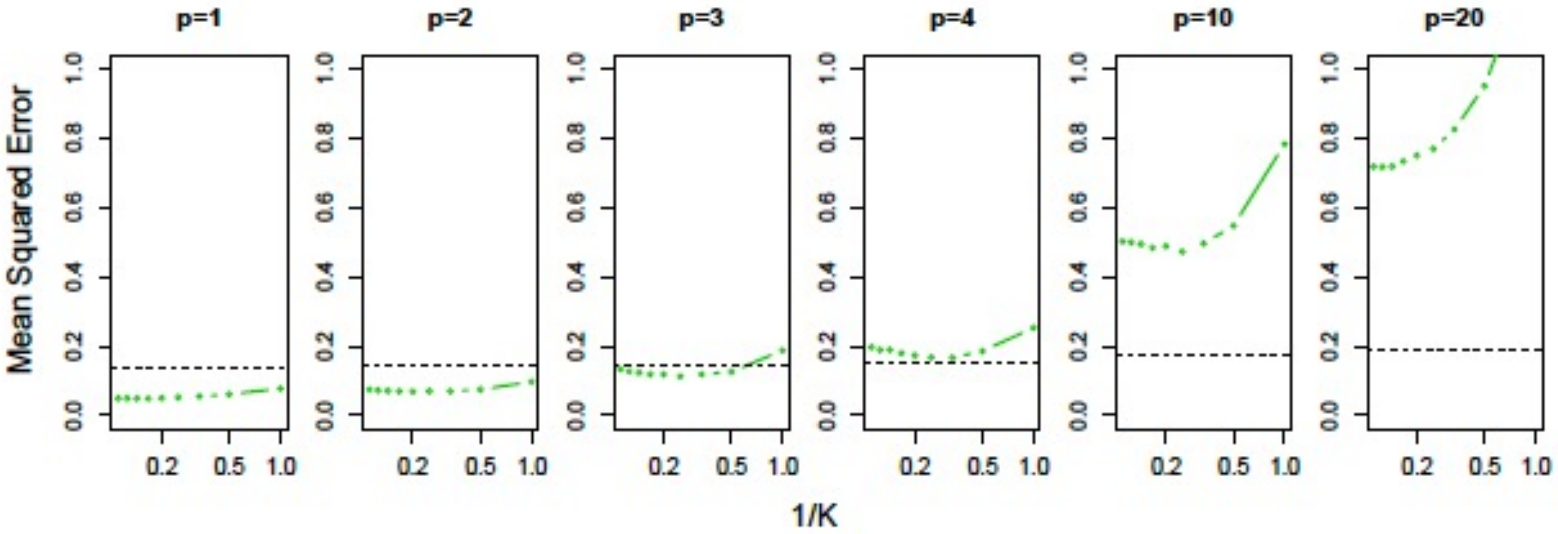
When is parametric better than non-parametric? And vice versa?

- Parametric will always win for linear relationships
- Non-parametric will often win for. non-linear relationships

LR vs. KNN Regression

When is parametric better than non-parametric? And vice versa?

- Parametric will always win for linear relationships
- Non-parametric will often win for. non-linear relationships
- KNN performance will degrade more quickly as noise (number of predictors) increases


p=1
p=2
p=3
p=4
p=10
p=20
Mean Squared Error
1/K