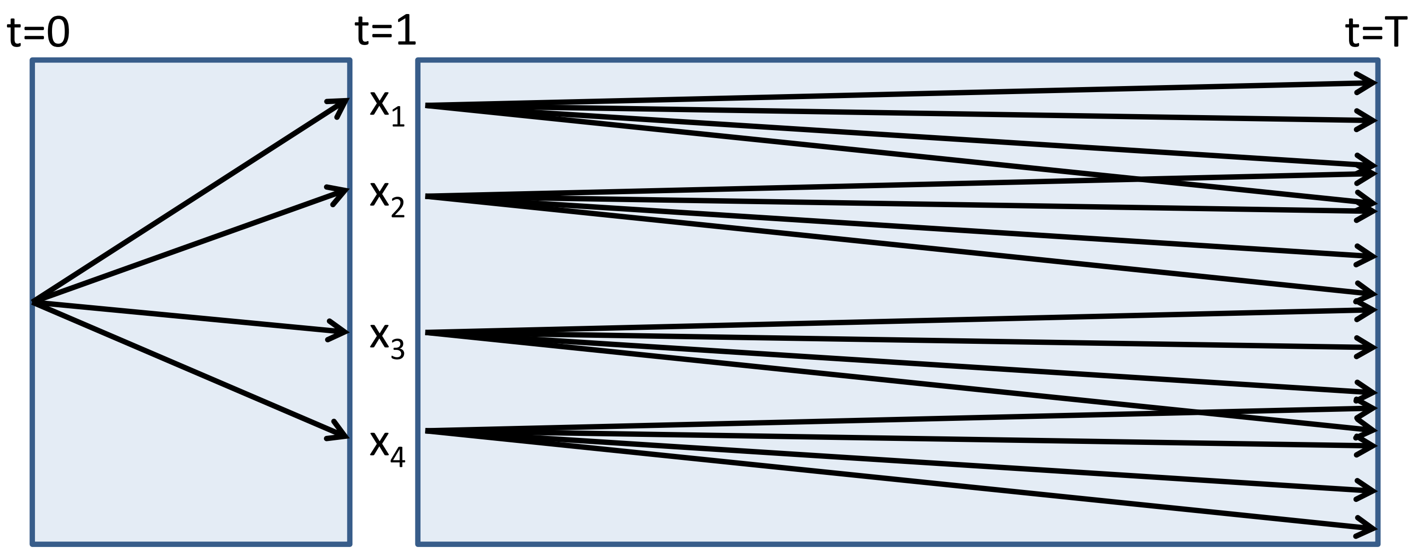
Kleynen Consultants
Actuarissen & Risk Managers

Difference between LSMC and Replicating Portfolios

Based on joint work with: Eric Beutner and Janina Schweizer at Maastricht University

Risk Calculations under Solvency II

- Price at t is calculated as conditional expectation under **Q**-measure for a specific scenario x at t
  - A scenario is a specific value for the relevant risk-drivers
- Mathematical notation: price(t,x) = $\mathbf{E}^{Q}[\, f(S_T) \mid S_t = x \,]$
- How to compute this value?
  - "Brute force": simulation-in-simulation
  - Alternative : fit a function at t=1 or t=T



Outline

- Approximation of Functions
- Approximation in Higher Dimensions
- Regress Now vs Regress Later

Approximation of Functions

Approximation - Distance

- Consider a random variable S_T with a probability density function $p(S_T)$.
  - The variable S is a risk-driver, e.g. stock-price or interest rate.
- Consider a (payoff) function $f(S_T)$
  - For example: $f(S_T) = \max\{ S_T - K , 0 \}$ or $f(S_T) = \ln S_T$
- Consider another function $g(S_T)$.
- What is the "distance" between f and g?
  - Distance = 0 $\Leftrightarrow$ $f \equiv g$
  - Distance >0 for any $f \neq g$
  - Symmetry: $d(f,g) = d(g,f)$
  - Triangle inequality: $d(f,g) \leq d(f,h) + d(h,g)$ for all f,g,h

Approximation - Distance

- Use "root mean square error" as distance:

$$d(f,g) = \left(\int \left(f(S) - g(S) \right)^2 p(S) dS \right)^{\frac{1}{2}}$$

$$= E\left[\left(f(S) - g(S) \right)^2 \right]^{\frac{1}{2}}$$

- Satisfies all properties
  - Only for $f \equiv g$ for all S do we get $d(f,g)=0$, otherwise $d(f,g)>0$
  - Makes intuitive sense: give more weight to errors with high probability
- This choice is not unique. Other distance functions are also possible.
  - For example: use different probability *q(S)* or error-power.
  - "Norm equivalence": convergence for one distance-function implies convergence in other norms as well.

Approximation – Polynomials

- Approximate complicated payoff function $f(S)$ with "simple" functions.
  - Easy to compute market-price for the simple functions
- Example: choose polynomials S^k

- Approximate $f(S)$ with $\sum a_k (S_T)^k$ for k=0…K
  - Make smart choice for coefficients a_k
- Best choice: min $d(f, \sum a_k S^k) = E[(f - \sum a_k S^k)^2]$
  - Solve system of K+1 equations:

$$\begin{pmatrix} E[f] \\ E[Sf] \\ E[S^2 f] \\ \vdots \\ E[S^K f] \end{pmatrix} = \begin{pmatrix} 1 & E[S] & E[S^2] & \cdots & E[S^K] \\ E[S] & E[S^2] & E[S^3] & & \\ E[S^2] & E[S^3] & E[S^4] & & \vdots \\ \vdots & & & \ddots & \\ E[S^K] & & \cdots & & E[S^{2K}] \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_K \end{pmatrix}$$

Approximation – Polynomials

- Optimal solution:

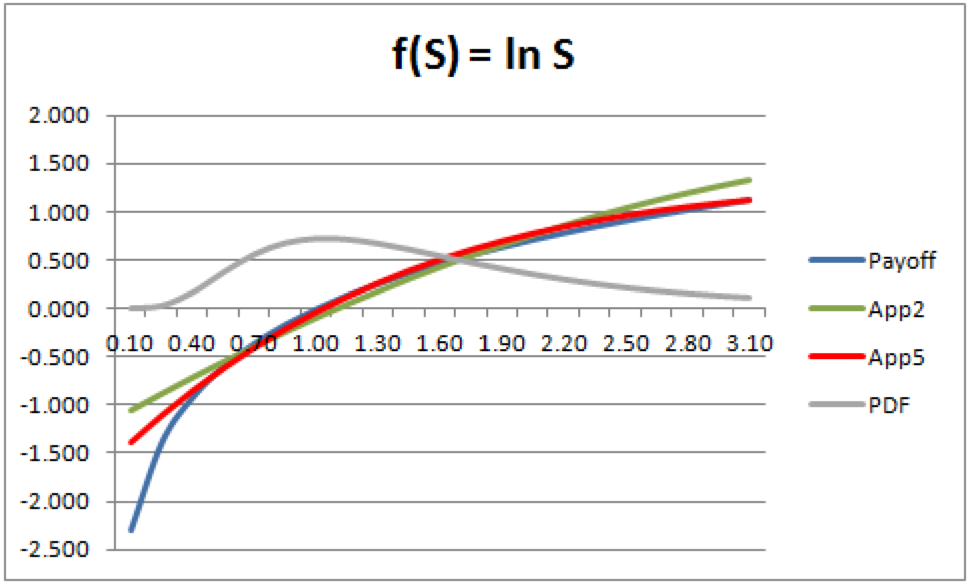
$$\begin{pmatrix} a_0^* \\ a_1^* \\ a_2^* \\ \vdots \\ a_K^* \end{pmatrix} = \begin{pmatrix} 1 & E[S] & E[S^2] & \cdots & E[S^K] \\ E[S] & E[S^2] & E[S^3] & & \\ E[S^2] & E[S^3] & E[S^4] & & \vdots \\ \vdots & & & \ddots & \\ E[S^K] & & \cdots & & E[S^{2K}] \end{pmatrix}^{-1} \cdot \begin{pmatrix} E[f] \\ E[Sf] \\ E[S^2 f] \\ \vdots \\ E[S^K f] \end{pmatrix}$$

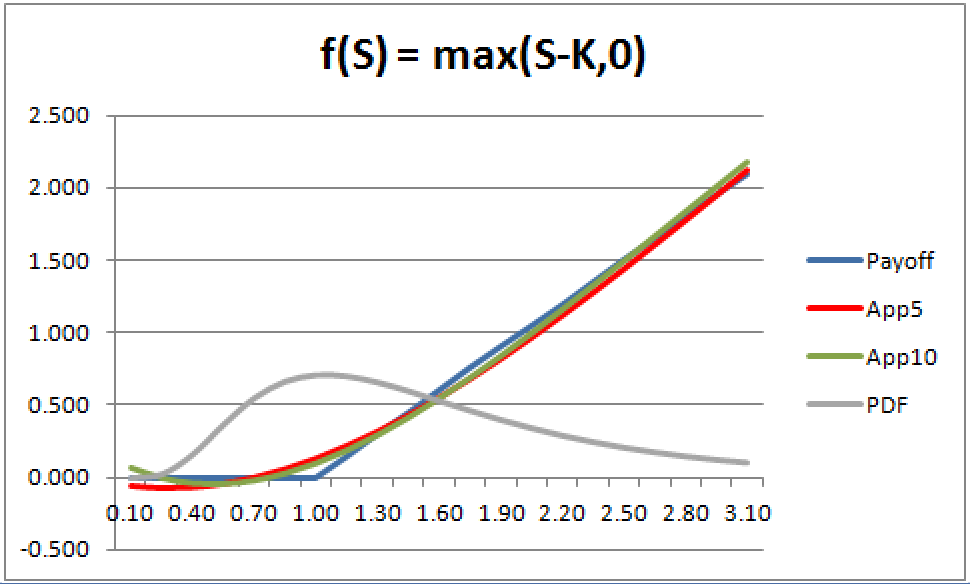
- This is a least squares solution: $a^* = (X'X)^{-1}(X'f)$
  - Implement this estimator for a finite sample
  - Each column in X is S^k
  - Each row in X and f is a draw from the random variable S

Approximation - Example

- Examples of approximation of payoffs with polynomials
  - Works very well for smooth functions
  - Payoff with kink is difficult for polynomials

- Lesson: choose appropriate basis for payoff





Approximation - Theory

- The collection of polynomials 1, S, S^2, ... forms a *basis* for the space of payoff functions
- Every function (with E[f^2] < ∞) can be perfectly replicated with polynomials for $K \longrightarrow \infty$
- Every function f has a unique representation: $f(S) = \sum_{k=0}^{\infty} a_k S^k$
- The coefficients a_k are deterministic (do not depend on S)
- Therefore we can compute (any measure **Q** and time t):

$$\mathbf{E}^{\mathbf{Q}}\left[f(S) \mid F_t\right] = \mathbf{E}^{\mathbf{Q}}\left[\sum_{k=0}^{\infty} a_k S^k \mid F_t\right] = \sum_{k=0}^{\infty} a_k \mathbf{E}^{\mathbf{Q}}\left[S^k \mid F_t\right]$$

- Express price of complicated payoff as sum of simple payoffs.

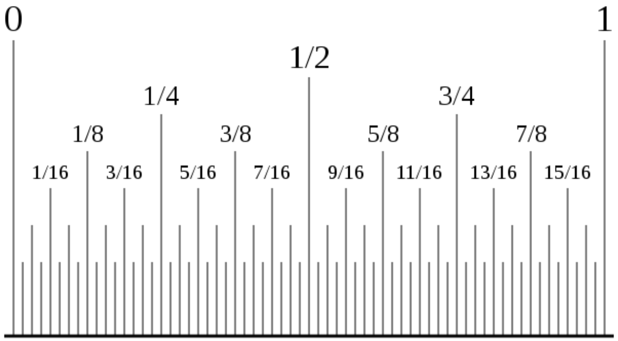
Approximation - Practice

- In practice we can only approximate a complicated *f(S)* with a finite number of terms: $f(S) \approx \sum_{k=0}^{K} a_k S^k$
- We can only use a finite sample to estimate the a_k coefficients: $f(S) \approx \sum_{k=0}^{K} \hat{a}_k S^k$
- Two sources of error:
  - *Truncation error* due to finite *K*, e.g. converge as $O(K^{-g})$
  - *Estimation error* due to estimate for a_k on sample of size *N*
  - Study of converge $(K,N) \rightarrow \infty$ by Beutner-Pelsser-Schweizer (2015)

- Choice of different basis will determine convergence rate *g* for a class of payoff functions
  - Polynomials work very well for smooth functions
  - Polynomials converge slow for kinked payoffs

Approximation – Choice of Basis

- There are many possible choices for basis-functions
  - Polynomials
  - Sin(), Cos() functions (Fourier basis)
  - Piecewise linear: max(S – K_k , 0) with K_k= $P^{-1}(d_k)$
    - With d_k are dyadic rationals



  - Other, see “machine learning” literature
- Find “good” basis to approximate payoff f(S) with a few basis functions
  - Also compute analytical price for each basis function
  - Piecewise linear ⇔ call/put options.

Approximation in Higher Dimensions

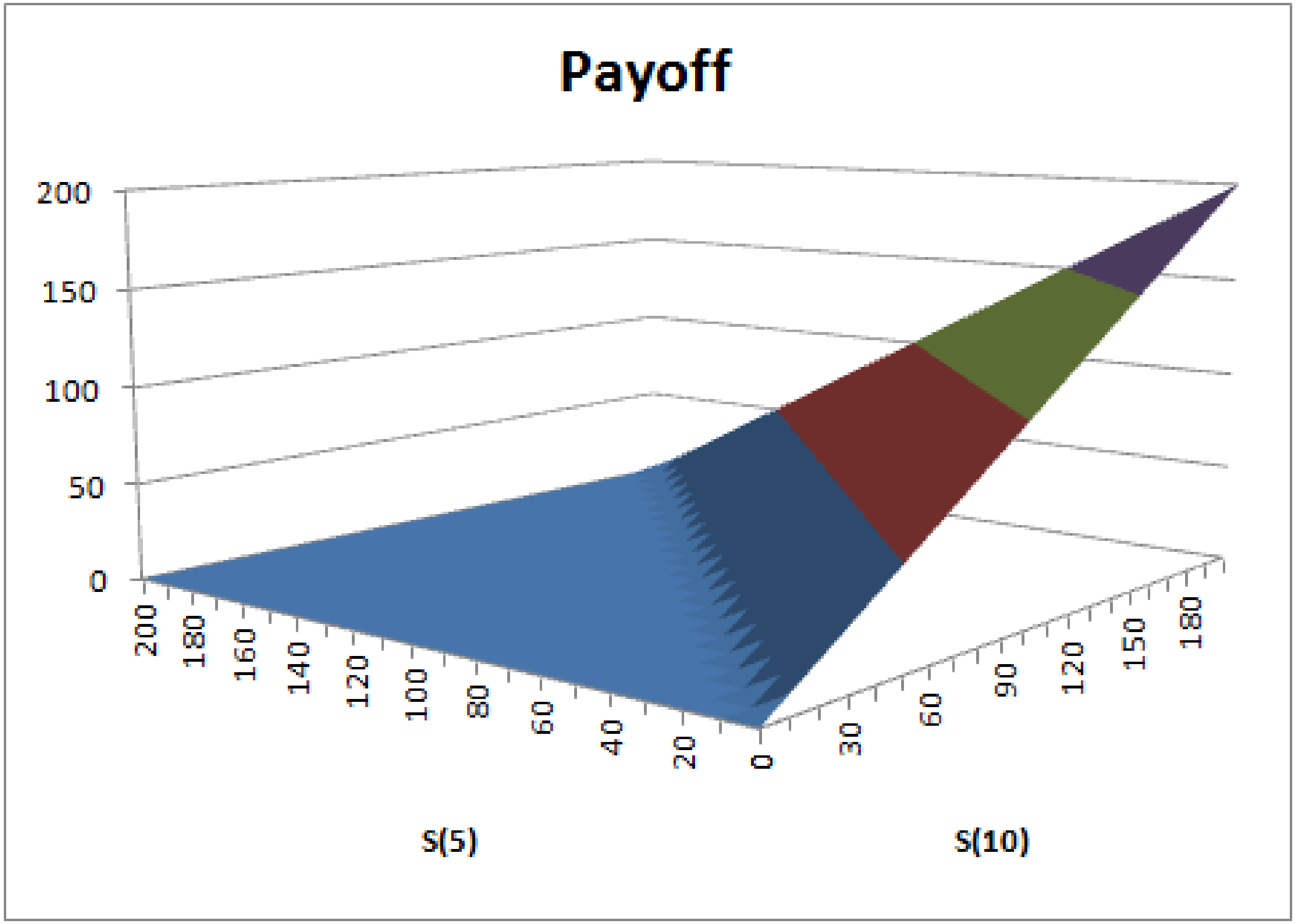
Higher Dimensions

- Realistic insurance products have a payoff that depends on multiple risk drivers
- Same risk driver at different points in time
  - Path dependent payoff, such as profit-sharing
- Different risk drivers
  - Unit-linked: mortality and financial
  - Interest rates and inflation

- General theory outlined before still works
- Use more elaborate basis to encompass all relevant risks
- Choice of good basis is even more important

Higher dimension – 2d example

- Consider a path-dependent payoff $\max(S_T - S_t , 0)$ with $t<T$
  - Only pay out positive return of S between t and T.



Higher dimension – 2d basis

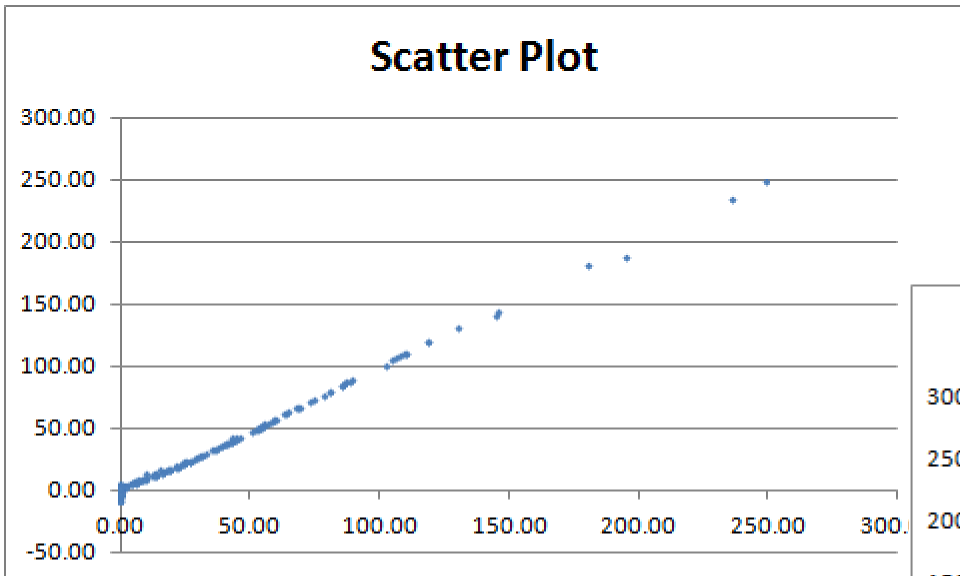
- Consider the following basis
- Poly's up to degree 4
- 15 terms in total
- Need cross-terms
  - Uni terms do not form basis!
  - Eur options do not form basis!

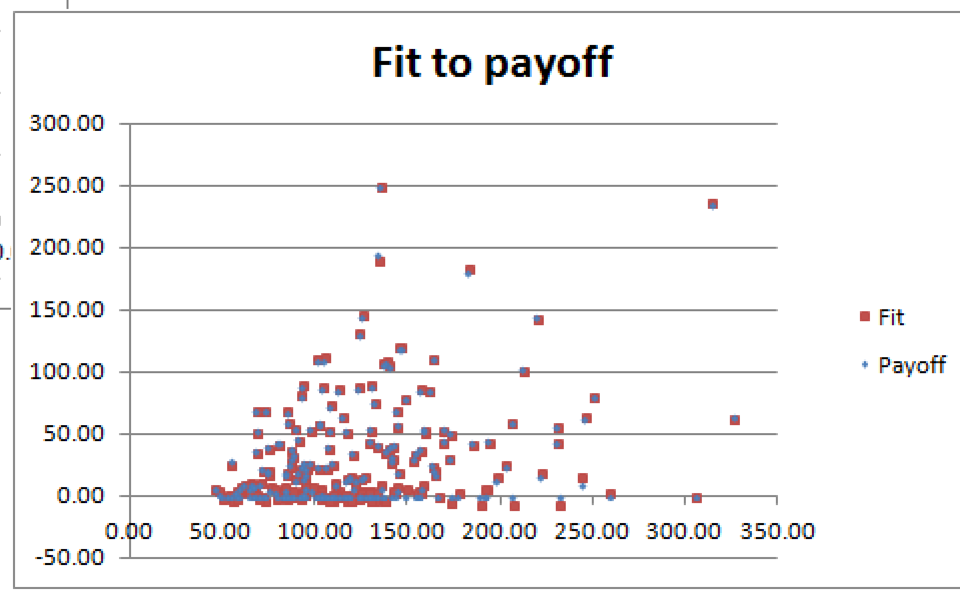
1	S_t	S_t^2	S_t^3	S_t^4
S_T	$S_t S_T$	$S_t^2 S_T$	$S_t^3 S_T$	
S_T^2	$S_t S_T^2$	$S_t^2 S_T^2$		
S_T^3	$S_t S_T^3$			
S_T^4				

- Curse of dimensionality for dim d: truncation error $O(K^{-g/d})$
  - General result for product basis
  - Really important to find "optimal" basis

Higher dimension – 2d example

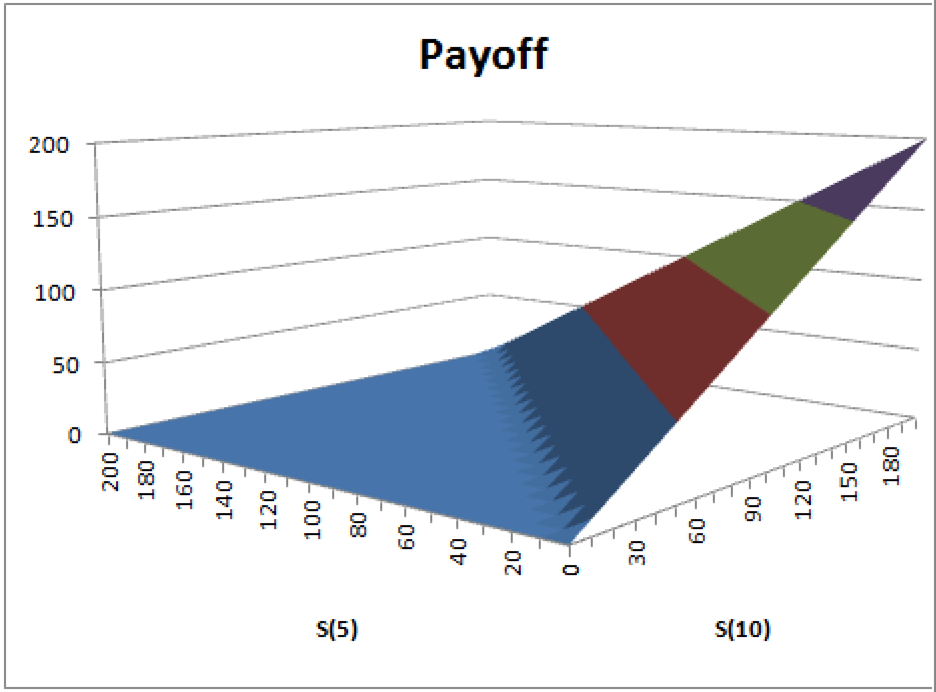
- Draw 200 random values from lognormal process
  - *dS=(4%)Sdt + (16%)SdW*
  - Fit payoff max(S_{10} - S_5 ,0) on the 15-term basis

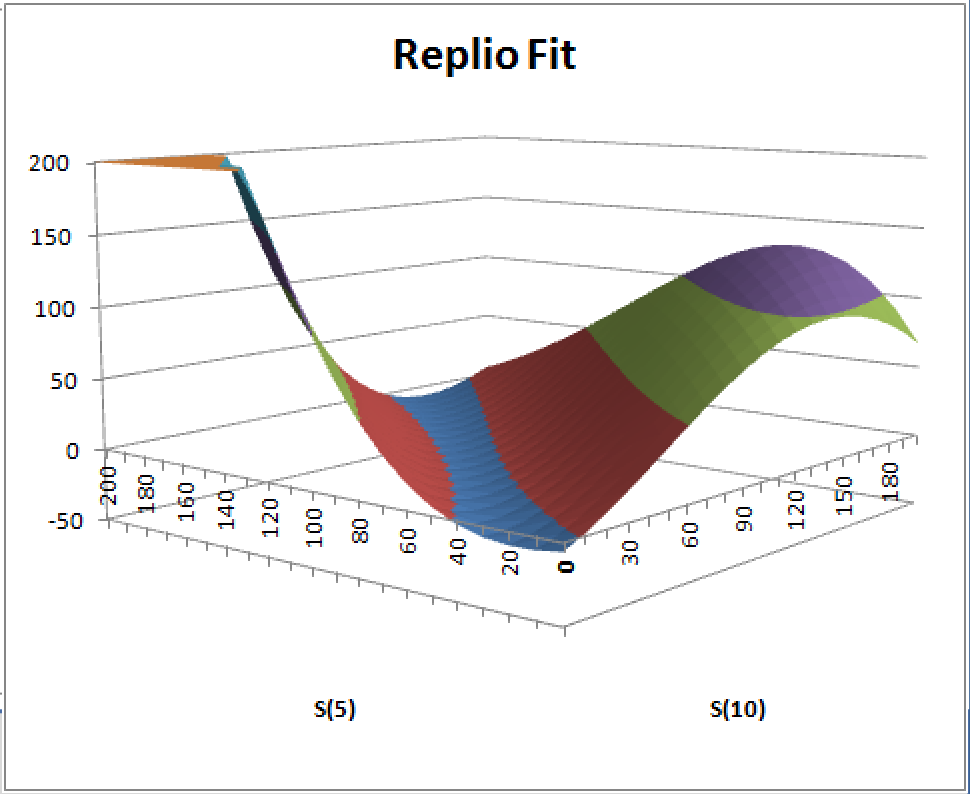




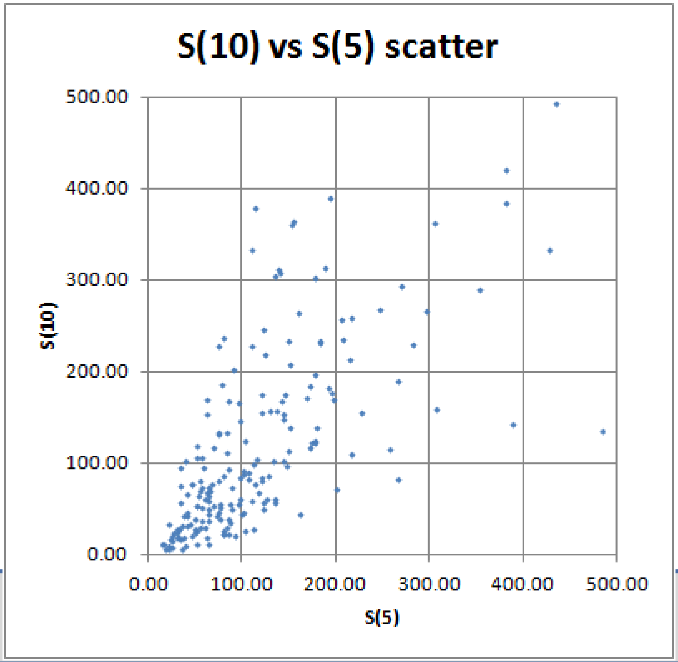
Higher dimension – 2d example

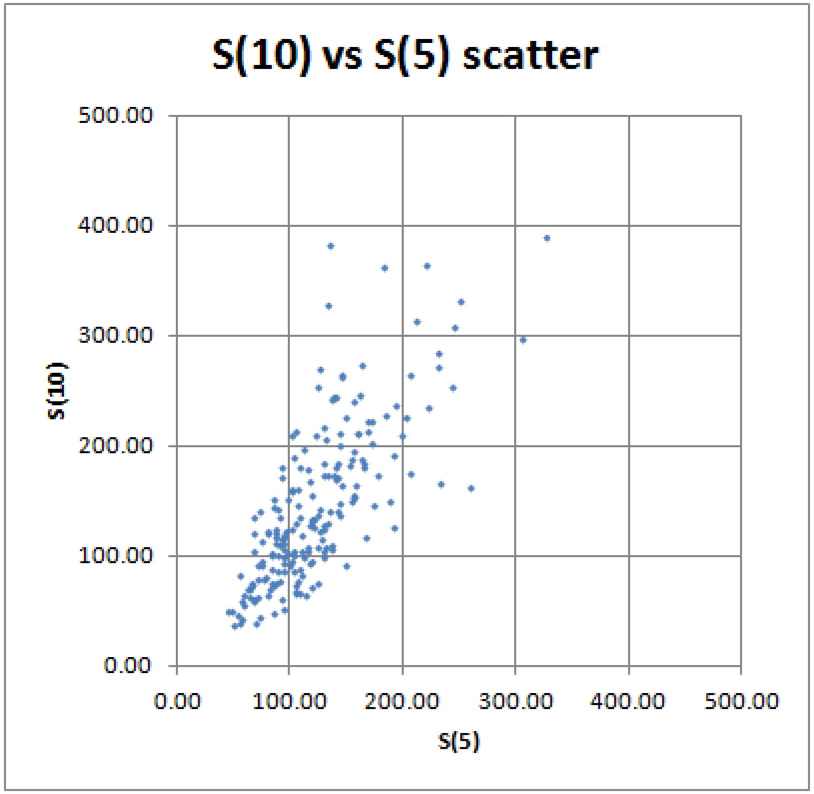
- Target vs Fitted function
  - Huge errors for S_5 high and S_{10} low...
  - But nearly perfect scatter plot!
- What went wrong?





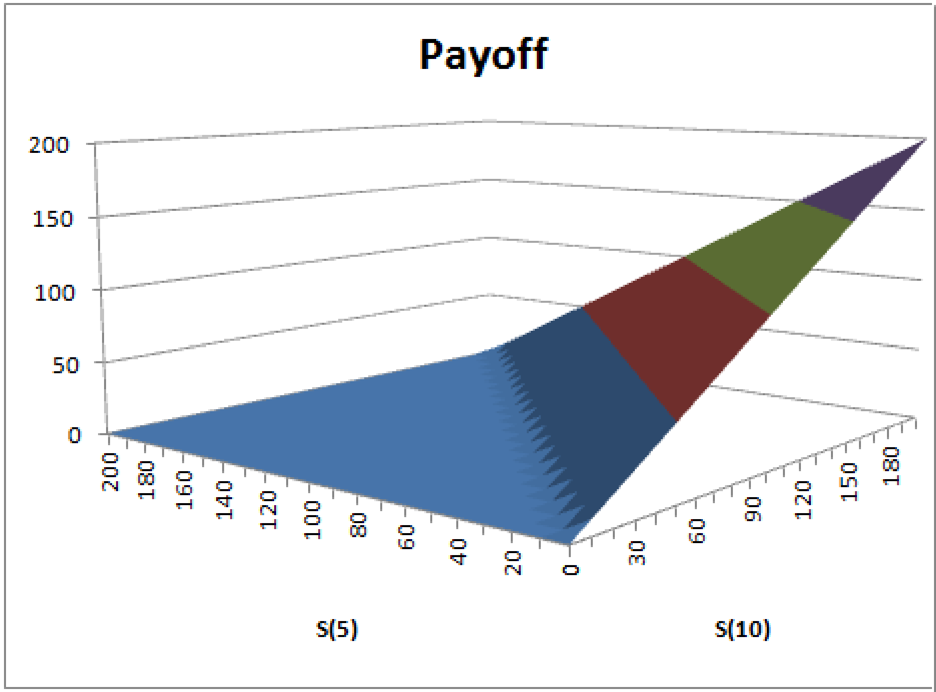
- What went wrong?
- Realistic training scenarios do not cover the whole space
  - They only cover "realistic" outcomes
  - Out-of-sample simulation from same model will cover same region
- Need to cover whole space
  - Increase volatility in model
  - At 32%

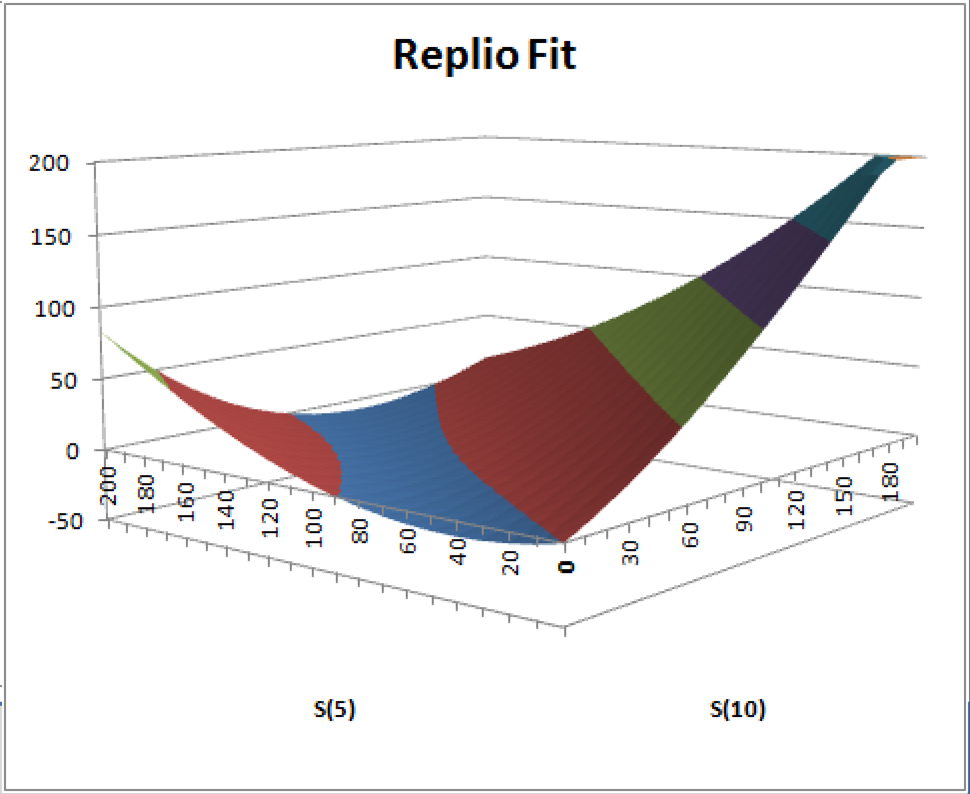




Higher dimension – 2d example

- Target vs Fitted function
  - Training sample with sig = 32%
- Much improved fit
  - Still errors for S_5 high and S_{10} low





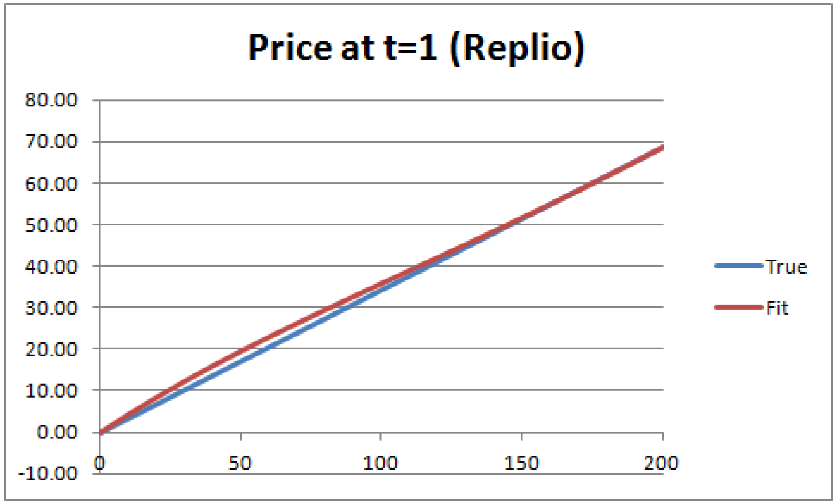
Higher dimension – Price at t=1

- Calculate price at t=1 of payoff under **Q**
  - Using realistic training sample

$$\mathbf{E}^{\mathbf{Q}}\left[\max\left\{S_{10}-S_5,0\right\}\mid S_1\right]=$$

$$\sum_{k,l} a_{k,l}\mathbf{E}^{\mathbf{Q}}\left[S_{10}^{k}S_{5}^{l}\mid S_1\right]$$

  - Using sig=32% training sample
  - Same blue line in both graphs!



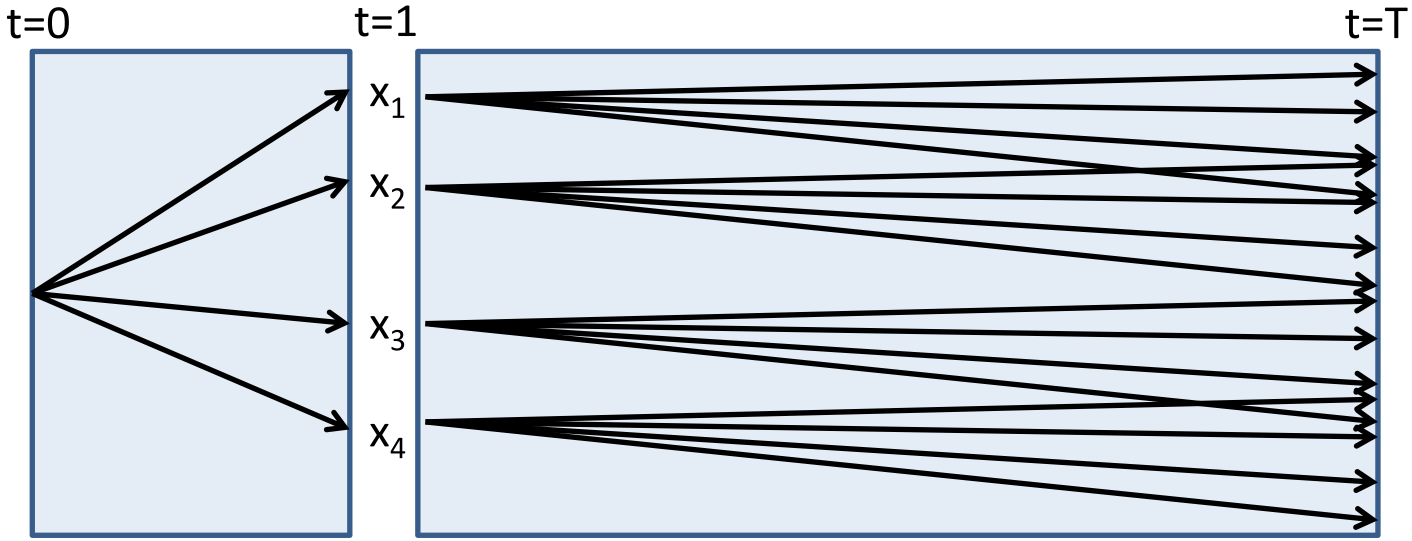
- Note: decoupling of training and pricing measure

Regress Now vs Regress Later

Calculate prices at t

- Price at t is calculated as conditional expectation under **Q**-measure for a specific scenario x at t
  - A scenario is a specific value for the relevant risk-drivers
- Mathematical notation: price(t,x) = $\mathbf{E}^{Q}[\ f(S_T)\ |\ S_t = x\ ]$

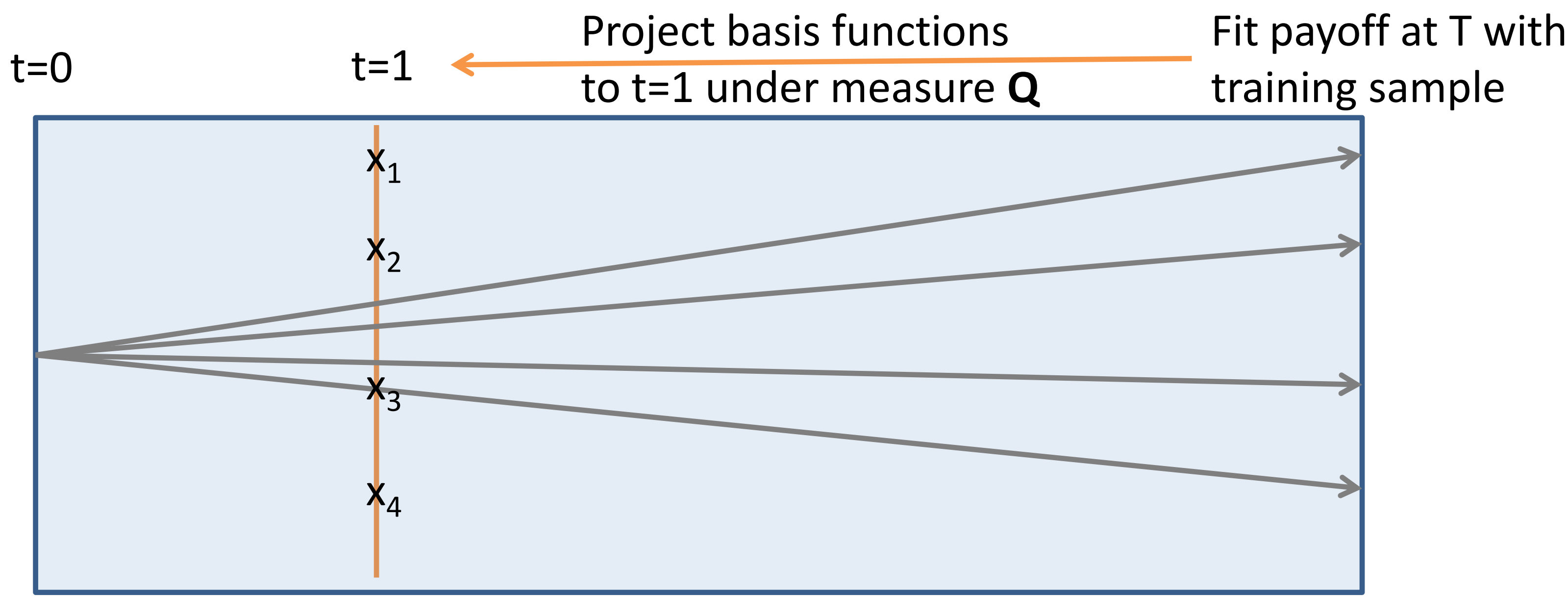
- How to compute this value?
- "Brute force": simulation-in-simulation



Calculate price at t

- Alternative methods to calculate prices at *t*
- Replicating portfolio:
  - First fit payoff on basis at *T*, then calculate expectation at *t*

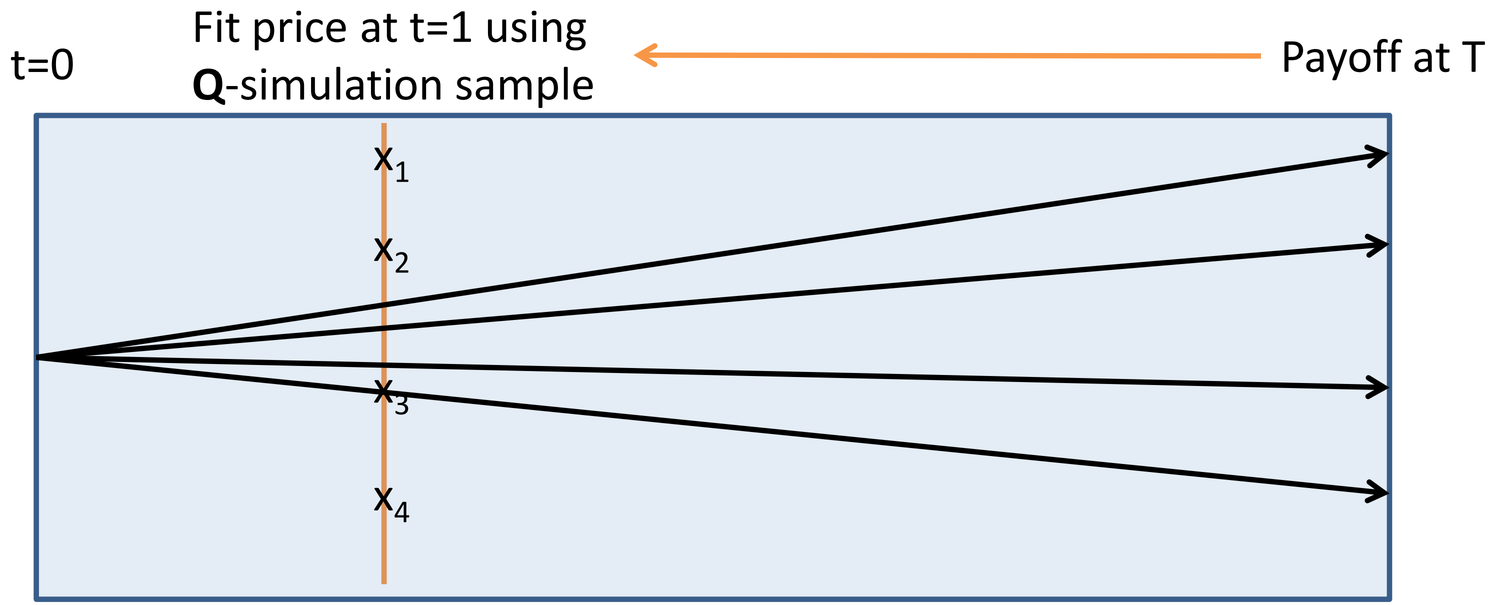
$$f(t,S_t) \approx \sum_{k=0}^{K} \hat{a}_k \mathbf{E}^{\mathbf{Q}}\left[(S_T)^k \mid F_t\right] \quad \text{with} \quad \hat{a} = \left(\mathbf{E}^{\mathbf{R}}[S_T^k S_T^l]\right)^{-1} \cdot \mathbf{E}^{\mathbf{R}}[S_T^k f(T,S_T)]$$



Calculate price at t

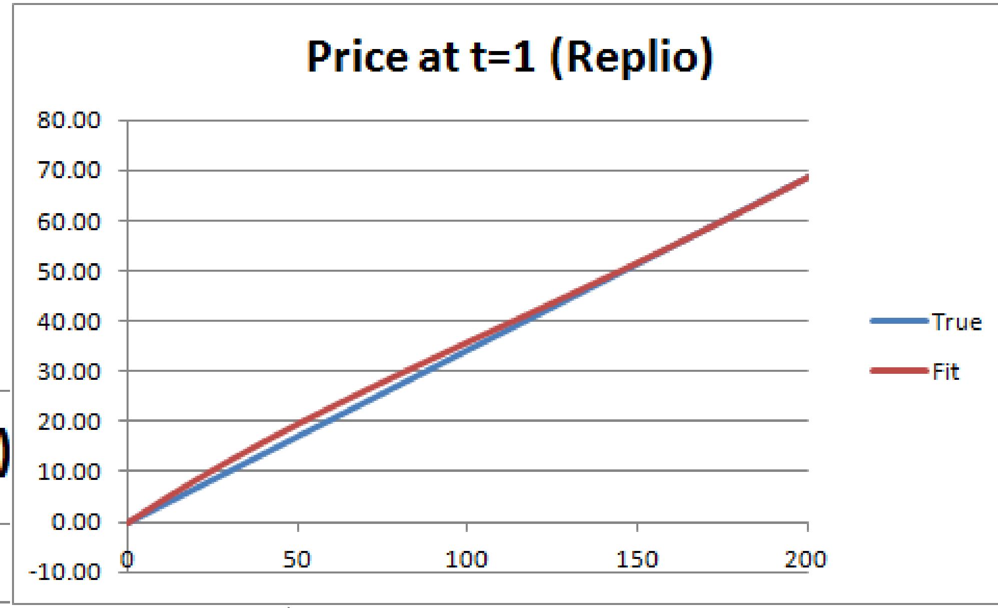
- Alternative methods to calculate prices at *t*
- Function fitting:
  - Calculate price at *t* by regressing payoff at *T* on basis at *t*

$$f(t,S_t) \approx \sum_{k=0}^{K} \hat{b}_k^{\mathbf{Q}} (S_t)^k \quad \text{with} \quad \hat{b}^{\mathbf{Q}} = \left(\mathbf{E}^{\mathbf{Q}}[S_t^k S_t^l]\right)^{-1} \cdot \mathbf{E}^{\mathbf{Q}}[S_t^k f(T,S_T)]$$

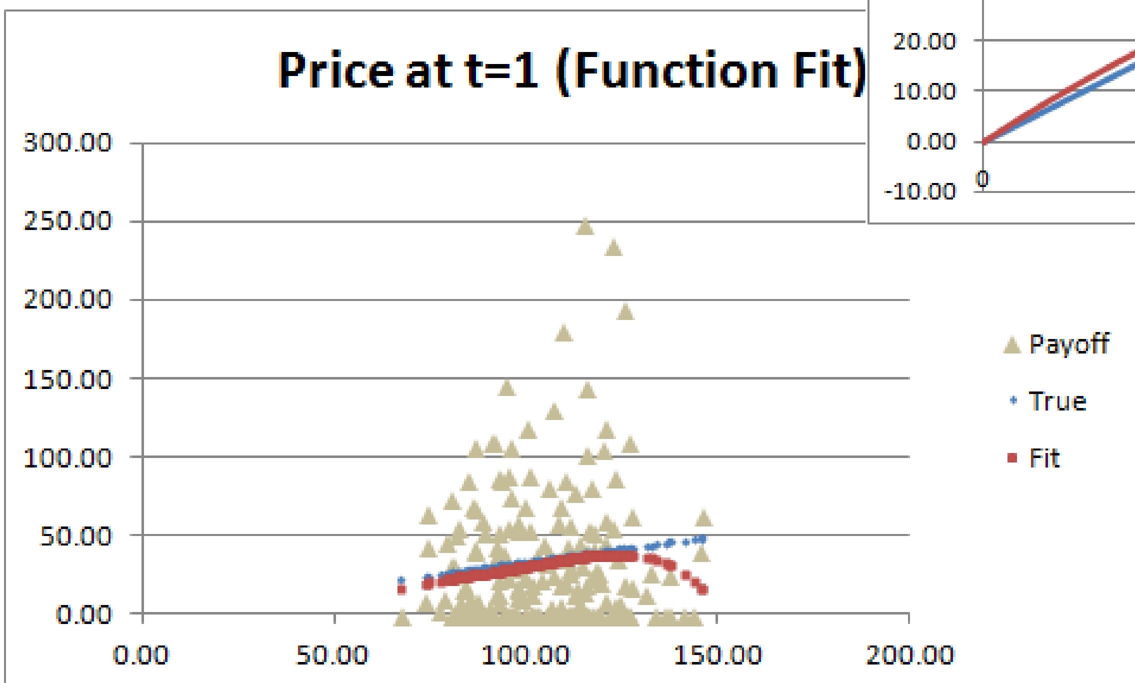


Example for 2d payoff

- Replio fit (training sig=32%, Q-sig=16%)



- Function fit (Q-sig=16%)



Function Fit:

- Regress payoff at T=10 on basis functions at t=1
- Low R^2 by construction

Function Fitting vs Replio

- Replicating portfolio / *Regress Later*
- First fits the payoff function
- Compute cond.expectation of basis analytically
- Harder for path-dep payoff
- Test quality of fit
- Is model-independent: changing the pricing **Q**-measure does not affect the coefficients a_k

- Function Fitting / LSMC / *Regress Now*
- Directly fits the pricing function
- Applies a smoothing during estimation
- Easy for path-dep payoff
- Cannot test quality of fit
- Is model-dependent: calculated price depends on simulated sample under **Q**-measure

Contact

Kleynen Consultants BV

St. Franciscusweg 21
6411 GH HEERLEN

T 045 – 571 47 83
E r.kleynen@kleynen-consultants.nl
I www.kleynen-consultants.nl

DISCLAIMER