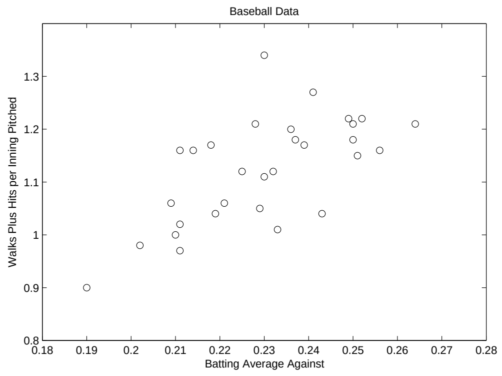
1. Bardsley 1.1. *This problem was written in the fall of 2011 by former University of Montana Math PhD students (and now PhD's) John Hossler, Marylesa Howard, and Jordan Purdy:* In baseball two commonly collected statistics for pitchers are batting average against (AVG) and walks plus hits per inning pitched (WHIP). Suppose we believe that a linear relationship exists between AVG and WHIP, and, in particular, we believe we can predict a pitcher's WHIP from their AVG. AVG and WHIP data for 30 Major League Baseball (MLB) pitchers from the 2011 regular season (through September 2) are provided in `BaseballData.m`. Use this data to answer the following questions.

(a) Plot AVG against WHIP such that AVG predicts WHIP.



(b) Write out by hand the first four rows of the design matrix. Construct the entire design matrix in MATLAB. Include an intercept in your model.

Solution:

Let Y_j be the j-th observed response (WHIP), x_j the explanatory variable (AVG), and $\eta_i \sim N(0, \sigma^2)$ the stochastic error with some constant variance σ^2. The simple linear model is

$$Y_j = \beta_0 + \beta_1 x_j + \eta_j \quad \text{for } j = 1 \dots 30.$$

If we let $\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\eta}$ be vectors with the given entries and $\boldsymbol{\beta}$ a vector with the linear coefficients, we may write the model in matrix-vector form as

$$\boldsymbol{Y} = A\boldsymbol{\beta} + \boldsymbol{\eta},$$

where

$$A = \begin{bmatrix} 1 & 0.209 \\ 1 & 0.210 \\ 1 & 0.190 \\ 1 & 0.211 \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow \\ \mathbf{1} & \boldsymbol{x} \\ \downarrow & \downarrow \end{bmatrix}.$$

The following MATLAB code constructs the design matrix from the given data:

```
>> A = [ones(size(AVG)) AVG];
```

□

(c) Compute the least squares estimate. Note that $\boldsymbol{\beta}$ is a 2×1 vector.

Solution:

The corresponding normal equation $A^T A\boldsymbol{\beta} = A^T\boldsymbol{Y}$ can be solved numerically with the following MATLAB command:

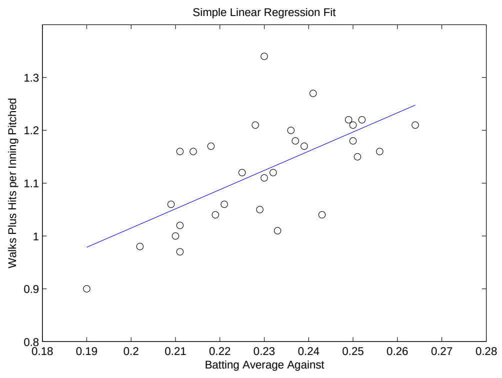
```
>> beta = (A'*A)\(A'*WHIP)

beta =

    0.2872
    3.6386
```

$\square$

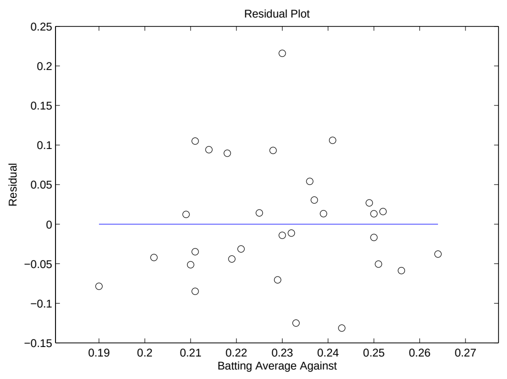
(d) Using the model created in part (c), plot the model on the data plot from part (a). This model predicts a pitcher's WHIP given AVG.

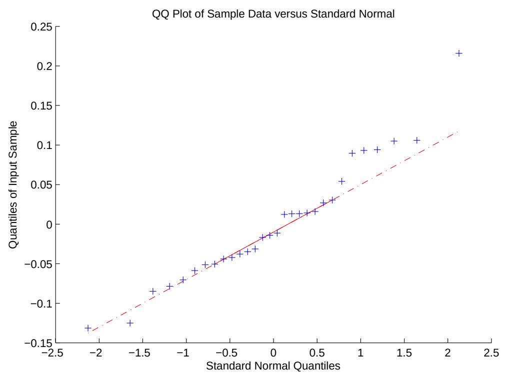


(e) For each observed data point, the residual is that observed value (AVG) minus the corresponding predicted value (predicted WHIP) from the model. Calculate the thirty residuals and plot them against their corresponding AVG value. On the same graph plot $y = 0$. Comment on whether a pattern is distinguishable or whether there is random scatter in the residual plot. A pattern in the residuals may suggest a different model or a linear model with different error structure is more appropriate for this data set. While random scatter does not verify our choice of this linear model is correct, it does suggest this model may be useful for analysis.

Solution:

The apparent absense of pattern in the plot on the left suggests that the model may be useful. Additionally, the figure on the right plots the quantiles of the residuals versus the same quantiles of the standard normal distribution. The "linearity" of these points suggests that the error structure of this model (normal and addtive).





□

2. Bardsley 1.2. The midpoint quadrature rule for a function f defined on $[a, b]$ is defined by

$$\int_a^b f(x)\,dx = (b-a)f\left(\frac{a+b}{2}\right) + \frac{f''(\eta)}{24}(b-a)^3, \tag{1.26}$$

where $a \le \eta \le b$.

(a) Write your own MATLAB code applying midpoint quadrature with $n = 100$ grid points to approximate $\int_0^{\pi/2} \cos x\,dx = 0$.

Solution:

We assume a uniform grid on $[0, \pi/2]$, and estimate the integral by applying (1.26) on each subinterval. The following MATLAB codes implement this:

```
n = 100;
a = 0;
b = pi/2;

s = (1/(2*n)):1/n:(1-1/(2*n)); % quadrature midpoints on unit interval
s = s*(b-a) + a; % scale to given interval
sum( cos(s)/n )
```

□

(b) Determine an upper bound for the quadrature error in part (a) using (1.26) (n times) and then verify that the inequality holds.

Solution:

Note that the second derivative of $\cos(\eta)$ is $-\cos(\eta)$, so applying (1.26) to each subinterval,

$$\left|\int_0^{\pi/2} \cos(x)\,dx - \sum_{n=1}^{100} \cos(s_i)\frac{1}{100}\right| = \left|\sum_{n=1}^{100} \frac{-\cos(\eta_i)}{24}(\pi/2)^3 \times 10^{-6}\right| = \sum_{n=1}^{100} \frac{\cos(\eta_i)}{24}(\pi/2)^3 \times 10^{-6},$$

where the last equality holds since $\cos(x) \ge 0$ for $0 \le x \le \pi/2$. Note cos is decreasing on $[0, \pi/2)$, so its maximum is $\cos(a_i)$ on each subinterval where $a_i = i\pi/(2n), i = 0 \ldots (n-1)$. An upper bound is given by summing with $\eta_i = a_i$. The following codes compare the actual error to the upper bound.

```
>> err = abs((sin(b)-sin(a)) - approx)  % Calculate error
err =

   1.0281e-05

>> err_bound = sum( cos(s-h/2) )/24e6*(pi/2)^3     % Calculate upper bound for error

err_bound =

   1.0361e-05
```

□

(c) Modify `Deblur1d.m` so that it implements midpoint quadrature on the convolution model (1.7) with kernel

$$a(x) = \begin{cases} 100s + 10, & -\frac{1}{10} \le s \le 0, \\ -100s + 10, & 0 \le s \le \frac{1}{10}, \\ 0, & \text{otherwise.} \end{cases}$$

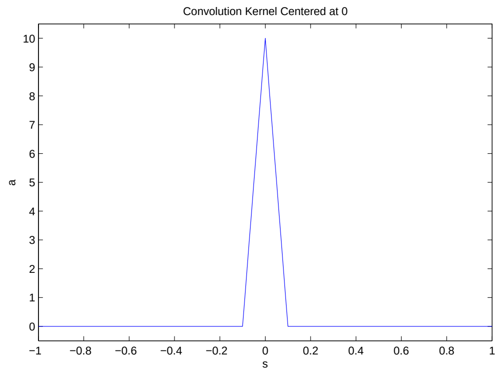
to obtain $\boldsymbol{b} = \boldsymbol{Ax}$. Plot a on $[-1, 1]$. Is the deblurring problem ill-posed with this kernel? Support your answer.

Solution:

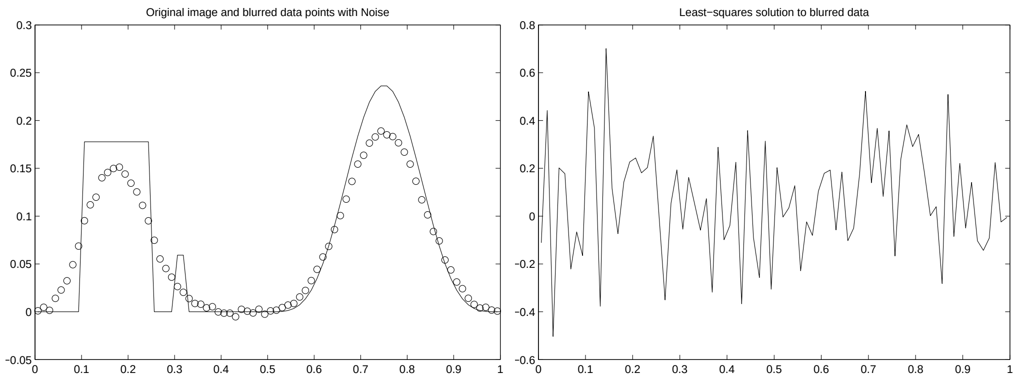
One can define a as an anonymous function in MATLAB with the following code

```
a = @(s)( ((100*s + 10).*(s<=0) + (-100*s +10).*(s>0)).*(abs(s) < 1/10) );
```

The figure below is a graph of a for $-1 < s < 1$.



The deblurring problem with this kernel is also ill-posed, but less so. The original function and blurred data points (using the kernel $a(s)$) with added Gaussian noise ($\sigma = .02$) are given in the figure on the left and the least-squares reconstruction is given in the figure to the right.



Note that the reconstructed solution is quite poor in terms of capturing any of the characteristics of the original. However, the magnitude of change is much less than the deblurring problem with a Gaussian kernel. The following codes indicate the magnification of the change in the solution upon deblurring.

```
delta_b =

    0.0188

>> delta_x

delta_x =

    1.9031
```

□

3. Bardsley 1.3. **(a)** Show that

$$\boldsymbol{A} = \frac{1}{n}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{n\times n}$$

has inverse

$$\boldsymbol{A}^{-1} = n\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & \ddots & \ddots & 0 \\ 0 & -1 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{n\times n}.$$

Use induction on n.

Solution:

Note that when $n = 1$, both A and A^{-1} are the 1×1 identity matrix. The coefficients on the matricies are reciprocals, hence they are omitted in the following argument. Now, assuming the nth case, let us denote the $n \times n$ matrices as $\boldsymbol{A}_n$

and $\boldsymbol{A}_n^{-1}$ (without the coefficients $n, \frac{1}{n}$). Note that

$$\boldsymbol{A}_{n+1} = \begin{bmatrix} \boldsymbol{A}_n & \boldsymbol{0}_{n\times 1} \\ \boldsymbol{1}_{1\times n} & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{A}_{n+1}^{-1} = \begin{bmatrix} \boldsymbol{A}_n^{-1} & \boldsymbol{0}_{n\times 1} \\ \boldsymbol{x}_{1\times n} & 1 \end{bmatrix},$$

where $\boldsymbol{x}_{1\times n} = [0 \ \cdots \ 0 \ -1]$. Since each matrix has compatible block decompositions, they can multiply in blocks as follows

$$\boldsymbol{A}_{n+1}\boldsymbol{A}_{n+1}^{-1} = \begin{bmatrix} \boldsymbol{A}_n\boldsymbol{A}_n^{-1} + \boldsymbol{0}_{n\times 1}\boldsymbol{x}_{1\times n} & \boldsymbol{A}_n\boldsymbol{0}_{n\times 1} + \boldsymbol{0}_{n\times 1}1 \\ \boldsymbol{1}_{1\times n}\boldsymbol{A}_n^{-1} + 1\boldsymbol{x}_{1\times n} & \boldsymbol{1}_{1\times n}\boldsymbol{0}_{n\times 1} + 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 1} \\ -\boldsymbol{x}_{1\times n} + \boldsymbol{x}_{1\times n} & 1 \end{bmatrix},$$

and

$$\boldsymbol{A}_{n+1}^{-1}\boldsymbol{A}_{n+1} = \begin{bmatrix} \boldsymbol{A}_n^{-1}\boldsymbol{A}_n + \boldsymbol{0}_{n\times 1}\boldsymbol{1}_{1\times n} & \boldsymbol{A}_n\boldsymbol{0}_{n\times 1} + \boldsymbol{0}_{n\times 1}1 \\ \boldsymbol{x}_{1\times n}\boldsymbol{A}_n + 1\boldsymbol{1}_{1\times n} & \boldsymbol{x}_{1\times n}\boldsymbol{0}_{n\times 1} + 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 1} \\ -\boldsymbol{1}_{1\times n} + \boldsymbol{1}_{1\times n} & 1 \end{bmatrix}.$$

□

(b) Show that $\boldsymbol{A}^{-1}$ corresponds to the discritization of $\frac{d}{ds}$ on the grid $\{s'_j\}_{j=1}^n$, where $s'_j = (j - \frac{1}{2})/n$, with a zero Dirichlet boundary condition on the left (i.e. $x(s'_0) = 0$) and a zero Neumann boundary condition on the right (i.e. $s'_{n+1} = s'_n$). Use the forward difference approximation of the derivative:

$$\frac{d}{ds}x(s'_j) \approx \frac{x(s'_j) - x(s'_{j-1})}{h}.$$

Solution:

The columns of the linear operator defined above are given by how it operates on the standard basis vectors. I.e. $\boldsymbol{a}_j = \frac{d}{ds}\boldsymbol{e}_j$ where $\boldsymbol{a}_j$ is the jth column of the matrix representation and $\boldsymbol{e}_j$ is the vector with 1 in the jth coordinate and 0 elsewhere. For the case of $0 \le j \le n$, note that $a_{jj} = (1-0)/h$, $a_{j+1,j} = (0-1)/h$ and $(0-0)/h = 0$ otherwise.

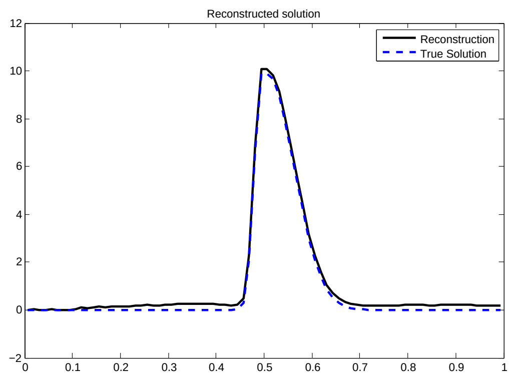
Note that with the Neumann condition, we can extend the operator to $\widetilde{A} : \mathbb{R} \to \mathbb{R}^{n+1}$ where $[\widetilde{A}x](s_{n+1}) = \frac{1}{h}\cdot(x(s_n) - x(s_{n+1})) \equiv 0$. Hence the final row must have each coordinate be 0.

□

(c) Replace $\boldsymbol{A}$ by $\boldsymbol{A}^{-1}$ in `PSFrecon.m`. How do the results change? Is $b = \boldsymbol{A}^{-1}\boldsymbol{x}$ an illposed problem, i.e. does it fail any of the three conditions of Definition 1.1?

Solution:

In this case, we are inverting the discrete derivative by backward differencing which results multiplication by $\boldsymbol{A}$. Because this inverse exists, both existence and uniqueness are guaranteed. However, there appears to be little sensitivity to changes in the data, as evidenced by the following reconstruction.



Note that multiplication by $\boldsymbol{A}$ is numerically integrating the diffential equation $x' = d$ using something similar to Euler's method, which is known to be reasonably stable. Moreover, note that the columns of $\boldsymbol{A}$ are orthogonal.

$\square$

4. Bardsley 1.5. Define

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

(a) Compute the singular value decomposition of $\boldsymbol{A}$ by hand: $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{3\times 3}$ is orthogonal and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 2}$ is diagonal, and $\boldsymbol{V} \in \mathbb{R}^{2\times 2}$ is orthogonal.

(b) Using the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ as basis elements write down bases for the four subspaces: the column space $C(\boldsymbol{A})$, the row space $C(\boldsymbol{A}^T)$, the nullspace $N(\boldsymbol{A})$, and the left null space $N(\boldsymbol{A}^T)$.

Solution:

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ have a singular value decomposition $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$ and k be the index of the first *positive* singular value. For $\boldsymbol{x} \in \mathbb{R}^m$,

$$\boldsymbol{Ax} = \sum_{i=k}^{\min\{n,m\}} (\sigma_i[\boldsymbol{V}^*\boldsymbol{x}]_i)\ \boldsymbol{u}_i,$$

and since each u_i are mutually orthogonal non-zero vectors, the subset $\{\boldsymbol{u}_k,\dots,\boldsymbol{u}_{\min\{m,n\}}\}$ forms a basis for the range $C(\boldsymbol{A})$. Since $\boldsymbol{A}^T = \boldsymbol{V\Sigma}^T\boldsymbol{U}^T$, A similar argument shows the vectors $\{\boldsymbol{v}_k,\dots,\boldsymbol{v}_{\min\{m,n\}}\}$ form a basis for $C(\boldsymbol{A}^T)$.

Note that $\boldsymbol{Ax} = 0$ if $\boldsymbol{x} = \{\boldsymbol{v}_1,\dots,\boldsymbol{v}_{k-1},\boldsymbol{v}_{\min\{n,m\}+1},\dots,\boldsymbol{v}_{\max\{n,m\}}\}$. By the Rank-Nullity theorem, these vectors suffice for making up a basis for all of $N(\boldsymbol{A})$. Similarly, $\boldsymbol{x} = \{\boldsymbol{u}_1,\dots,\boldsymbol{u}_{k-1},\boldsymbol{u}_{\min\{n,m\}+1},\dots,\boldsymbol{u}_{\max\{n,m\}}\}$ makes up a basis of $N(\boldsymbol{A}^T)$.

□

(c) Using the SVD computed in (a), compute the psuedo-inverse $\boldsymbol{A}^\dagger$ of $\boldsymbol{A}$, and then use it to compute the least squares solution $\boldsymbol{x}_{\text{LS}} = \boldsymbol{A}^\dagger b$ of

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Solution:

Note that $\boldsymbol{\Sigma}^\dagger$ is given by the $n \times m$ matrix with σ_i^{-1} on the main diagonal. So, from the above calculations

$$\begin{aligned}
\boldsymbol{A}^\dagger &= \boldsymbol{V\Sigma}^\dagger\boldsymbol{U}^T \\
&= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \\
&= \frac{1}{3} \begin{bmatrix} 1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}
\end{aligned}$$

Hence, $\boldsymbol{x}_{\text{LS}}$ is given by

$$\boldsymbol{A}^{\dagger}\begin{bmatrix}1\\1\\1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1 & 2 & -1\\1 & -1 & 2\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}\frac{2}{3}\\\frac{2}{3}\end{bmatrix}$$

□

(d) Compare the solutions in (c) with the solution to the normal equations, $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$. The two solutions should agree.

Solution:

Note the MATLAB output numerically solves the normal equation which agrees with the analysis above:

```
>> A = [1 1; 1 0; 0 1]; b = [1 1 1]';
>> A\b

ans =

    0.6667
    0.6667

>>
```

□