Simulation Results: Kernel Selection

Contents

We review results from our simulations for kernel selection.

Prepare Working Environment

We begin by loading the relevant R packages:

```
# Load relevant packages
pkgs_to_load <- c('dplyr', 'ggplot2', 'tidyr', 'viridis', 'cowplot')
lapply(X = pkgs_to_load, FUN = library, character.only = TRUE)

# Define directories for scripts and data
dir_main <- dirname(dirname(rstudioapi::getActiveDocumentContext()$path))
dir_src <- file.path(dir_main, 'source_scripts')
source(file.path(dir_src, 'define_directories.R'))

# Define custom functions used for plotting
```

```
source(file.path(dir_src, "define_plot_functions.R"))
source(file.path(dir_src, "define_plot_settings.R"))
```

All scenarios in this simulation setting share the following parameter values:

```
num_replicates  <- 1000
error_distribution <- 'normal'
signal_strength <- 1
values_for_signal_density <- c('sparse', 'dense')
values_for_sample_size <- c(50, 100, 150, 200)
values_for_error_corr_strength <- c(0, 0.5)
```

Continuous Features

This section includes scenarios where $\boldsymbol{X}$ is simulated as a continuous random vector.

```
x_type <- 'cts'
source(file.path(dir_src, 'initialize_adaptive_across.R'))

# Create filenames and load plot data
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
```

Kernel Selection Rates for Y_1

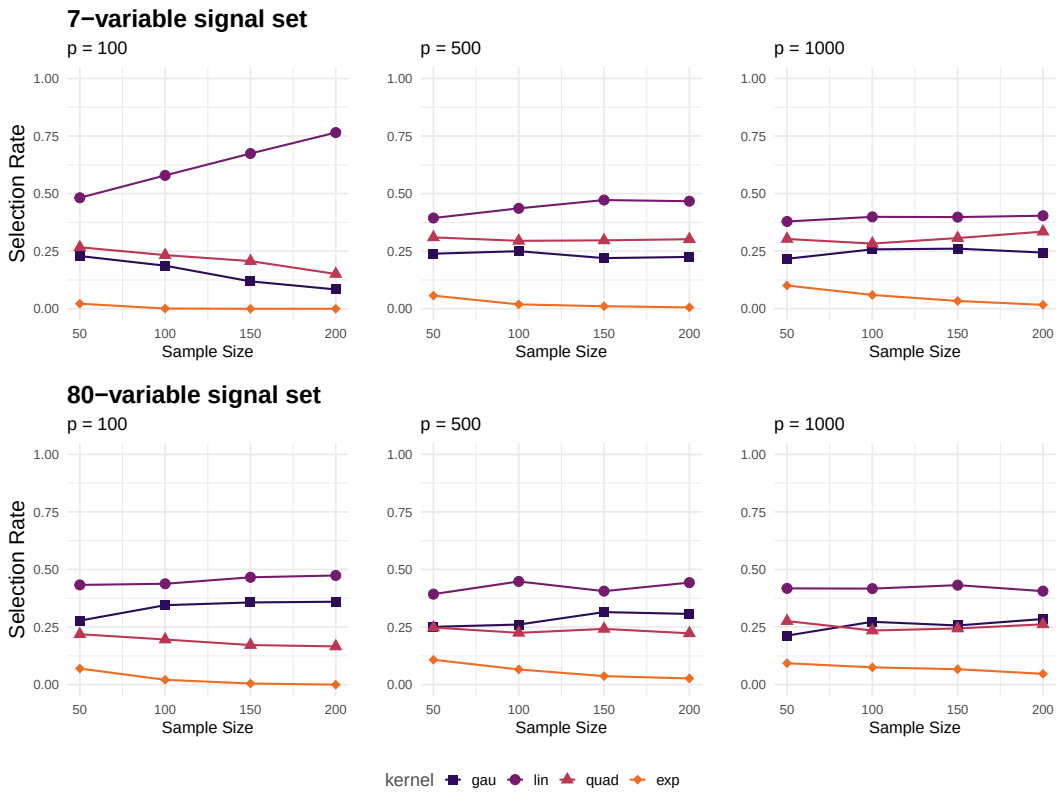
We consider the selection rate of each candidate kernel function for the 1st response component Y_1. The effect function for this response component had a linear functional form.

```
# Set index of current response component
y_ind <- 1
```

Uncorrelated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly uncorrelated components, i.e., the Pearson correlation coefficient is zero between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



We see that the linear kernel was selected more frequently than the other kernel functions.

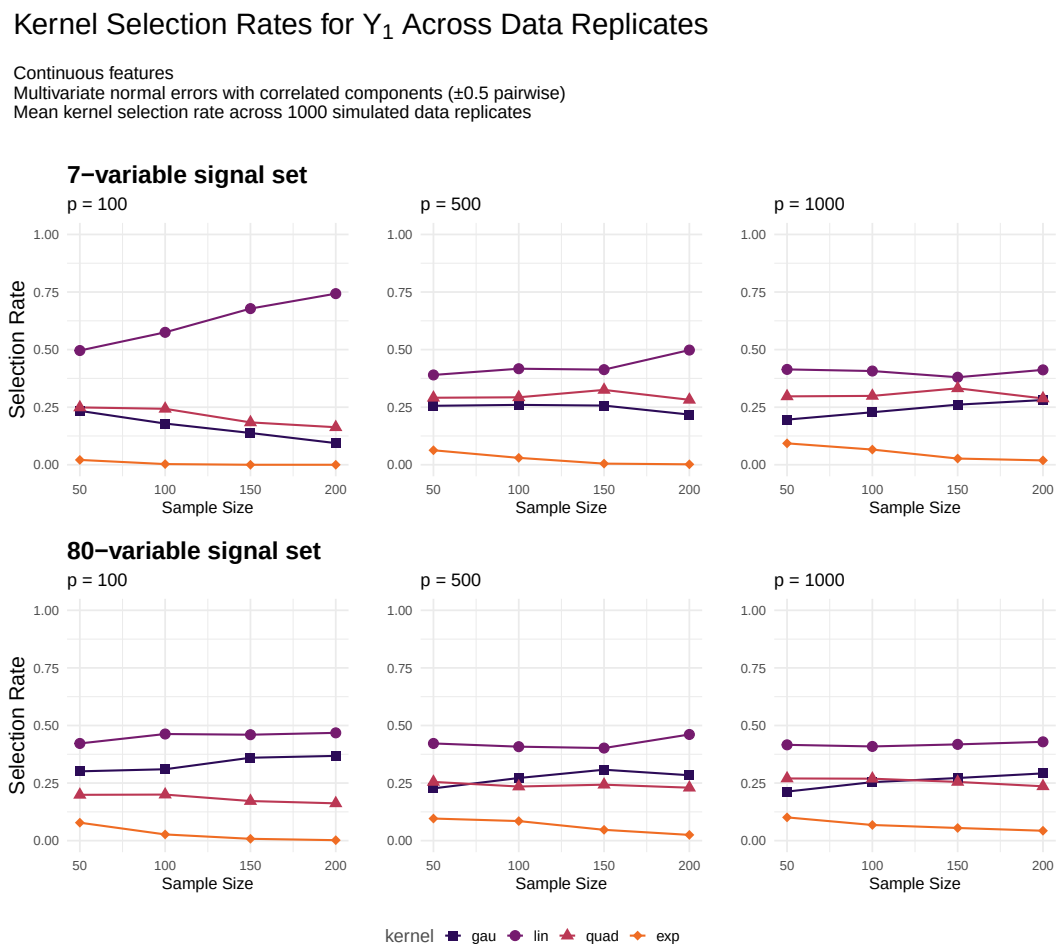
For the 7-variable signal set at $p = 100$ (top left panel), the sample size had a clear effect on the frequency with which the linear kernel was selected, while in other scenarios the effect of sample size was minimal over the range of values considered.

The feature dimension p typically had little effect on the selection rates, with the 7-variable signal at $p = 100$ (top left panel) yielding the only evidence of a substantial effect when compared to other scenarios.

Correlated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly correlated components, with the Pearson correlation coefficient equal to $(-1)^{j+j'}0.5$ between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0.5
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



The results here resemble those for the case with uncorrelated error components.

Kernel Selection Rates for Y_2

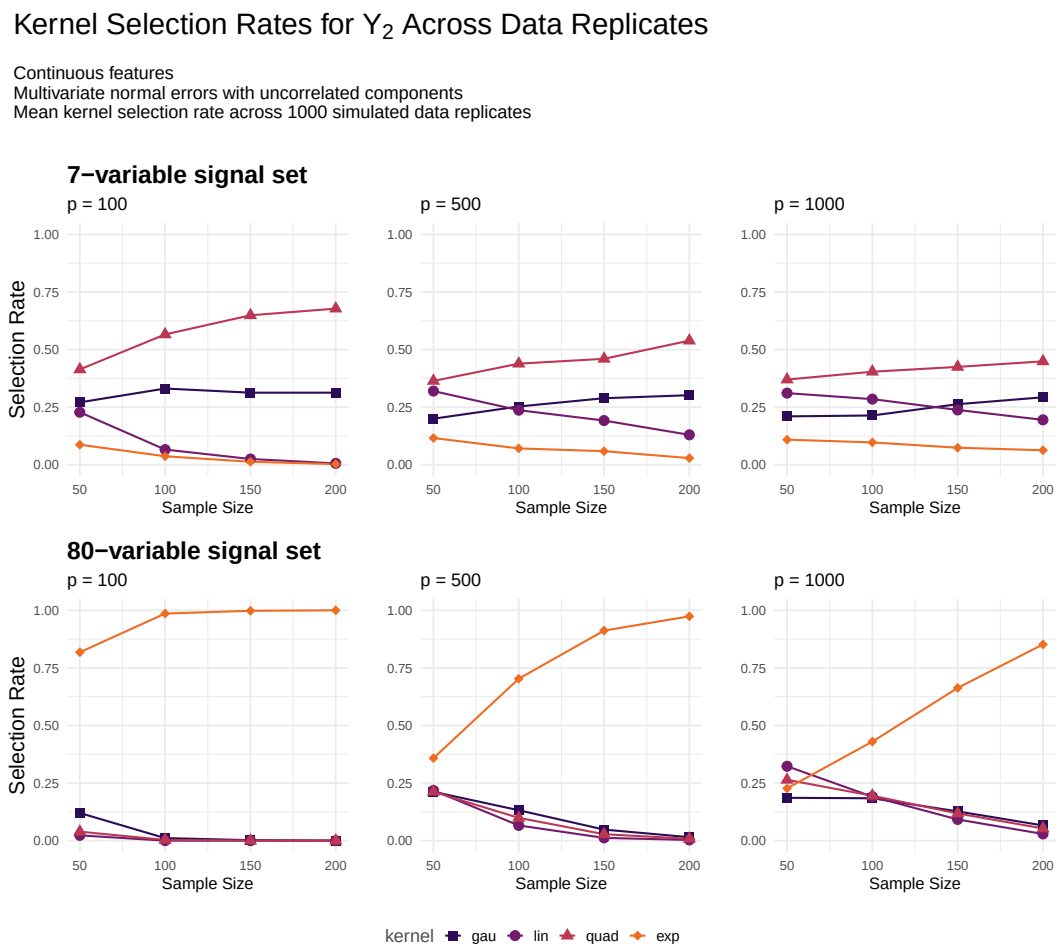
We consider the selection rate of each candidate kernel function for the 2nd response component Y_2. The effect function for this response component had a quadratic functional form.

```
# Set index of current response component
y_ind <- 2
```

Uncorrelated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly uncorrelated components, i.e., the Pearson correlation coefficient is zero between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



For the 7-variable signal, the quadratic kernel was selected most frequently, while for the 80-variable signal the exponential kernel was strongly favored. One difference in the effect function between the two signal sets is that for the 7-variable signal, the effect function contains squared and linear terms, while for the 80-variable signal set, the effect function consists solely of pairwise interaction terms.

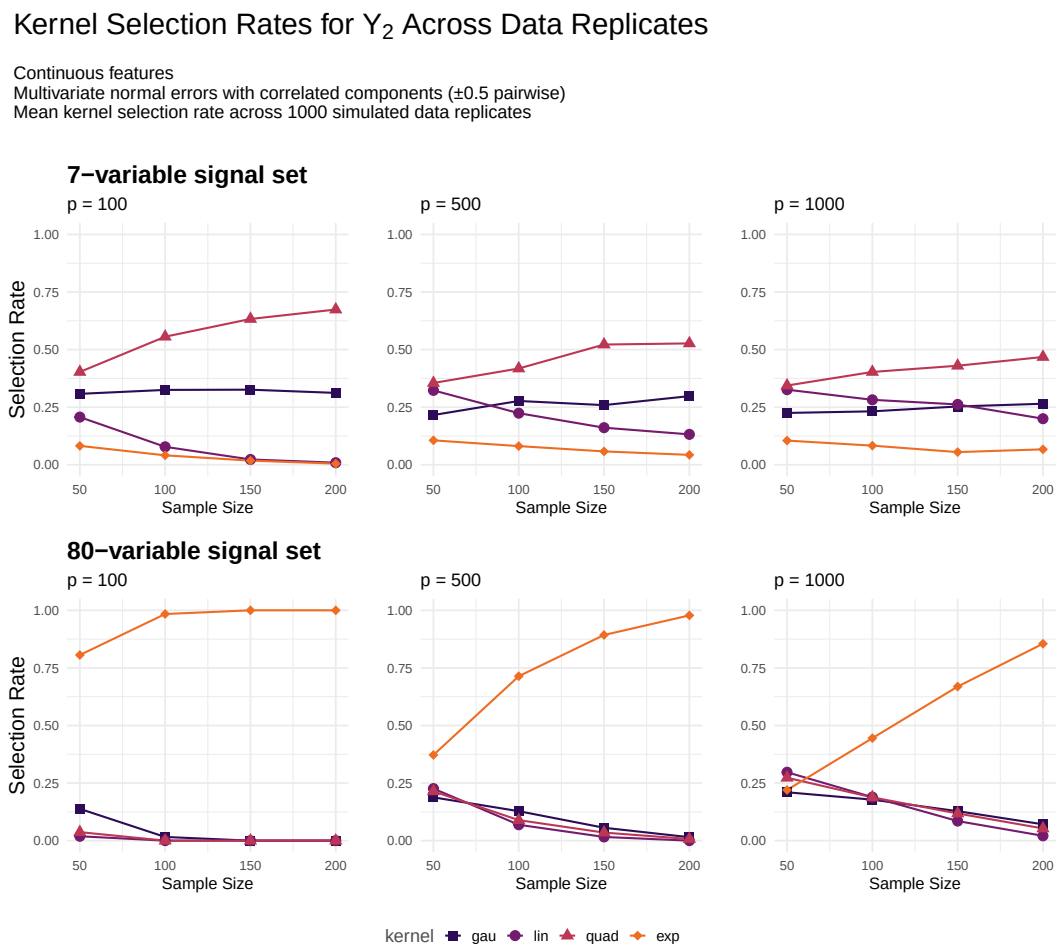
The selection frequency of the favored kernel increased with sample size and decreased with feature dimension.

For the 80-variable signal, the three non-favored kernels were selected at similar frequencies; for the 7-variable signal, we see a ranking in preference among the kernels, with the exponential kernel favored least. For this signal set, as sample size increased, the linear and exponential kernels were selected less frequently, while the Gaussian and quadratic kernels were selected more frequently.

Correlated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly correlated components, with the Pearson correlation coefficient equal to $(-1)^{j+j'}0.5$ between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0.5
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



The results here resemble those for the case with uncorrelated error components.

Kernel Selection Rates for Y_3

We consider the selection rate of each candidate kernel function for the 3rd response component Y_3. The effect function for this response component was nonlinear, consisting of terms of the form

$$a_1 H_d(a_2 X_k) e^{-\frac{(a_2 X_k)^2}{a_3}}$$

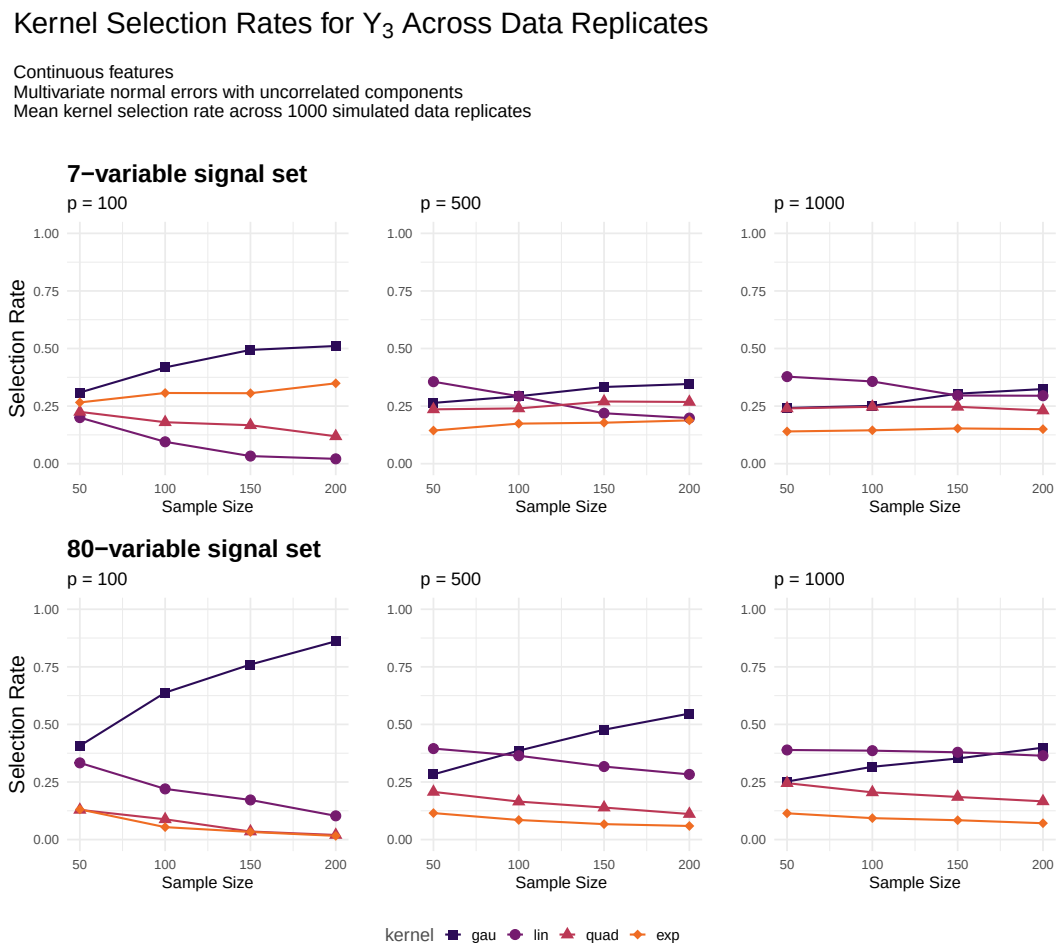
for some set of constant values of $a_1 \in \mathbb{R}$, $a_2 \in (0, 1]$, $a_3 \geq 1$, $d \in \{2, 3, 4\}$ and $k \in \{1, \ldots, p\}$, and where H_d is the dth degree physicist's Hermite polynomial.

```
# Set index of current response component
y_ind <- 3
```

Uncorrelated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly uncorrelated components, i.e., the Pearson correlation coefficient is zero between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



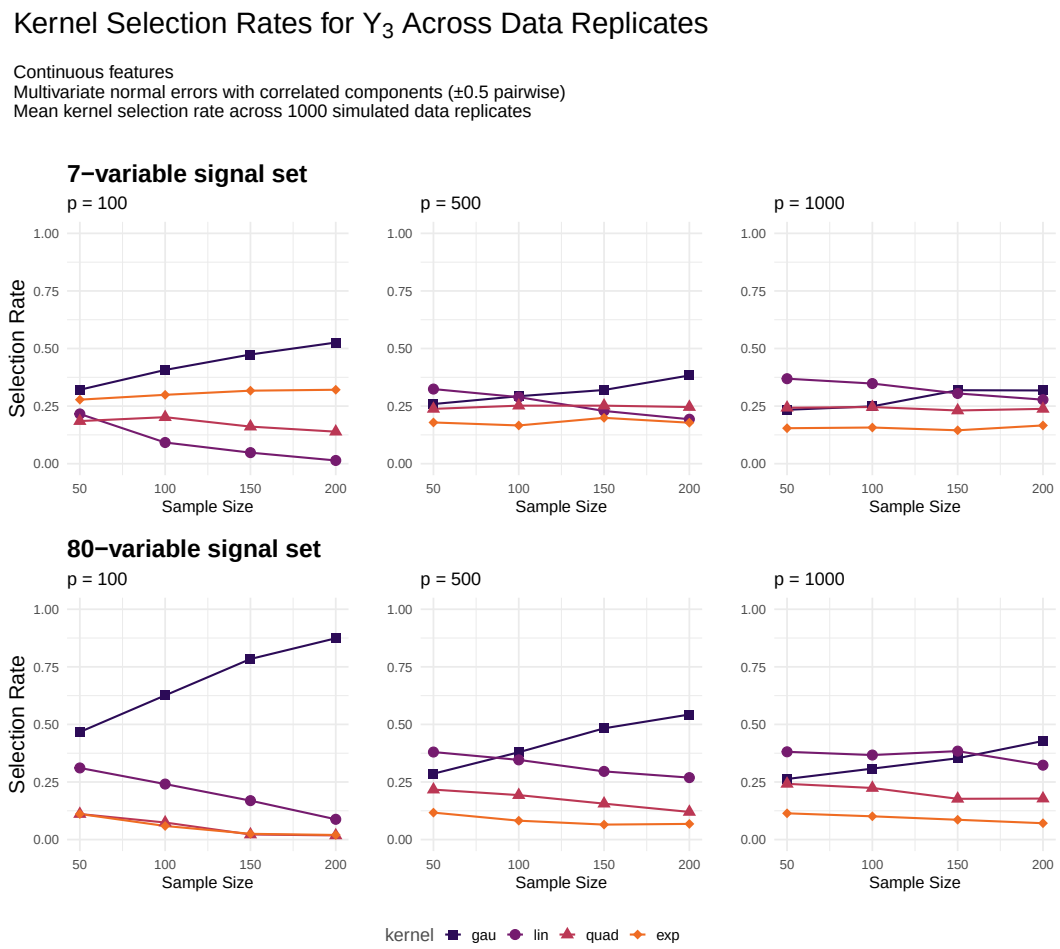
The Gaussian kernel was favored over other kernel functions, though for very high-dimensional data the linear kernel was favored; as the dimensionality of the data decreased (either through an increase in sample size or reduction in feature dimension), the Gaussian kernel's selection rate increased.

The Gaussian kernel was more strongly favored for the 80-variable signal than for the 7-variable signal. A difference between the two signal sets is that for the 7-variable signal, the effect function includes 2nd, 3rd and 4th degree Hermite polynomials, while for the 80-variable signal only 3rd degree Hermite polynomials are used.

Correlated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly correlated components, with the Pearson correlation coefficient equal to $(-1)^{j+j'}0.5$ between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0.5
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



The results here resemble those for the case with uncorrelated error components.

Kernel Selection Rates for Y_4

We consider the selection rate of each candidate kernel function for the 4th response component Y_4. The effect function for this response component was nonlinear, consisting of terms of the form

$$a\cos(X_k)e^{-\frac{X_k^2}{10}}$$

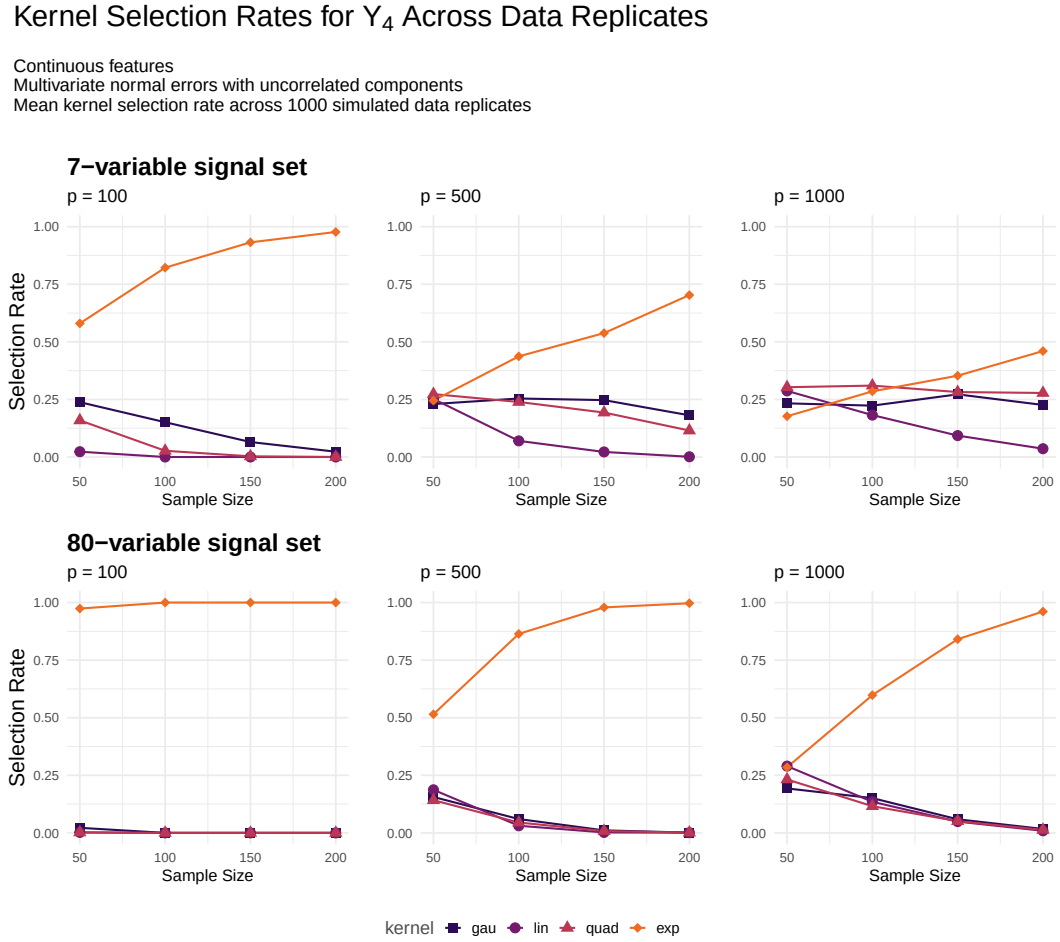
for some $a > 0$ and $k \in \{1, \ldots, p\}$.

```
# Set index of current response component
y_ind <- 4
```

Uncorrelated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly uncorrelated components, i.e., the Pearson correlation coefficient is zero between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```

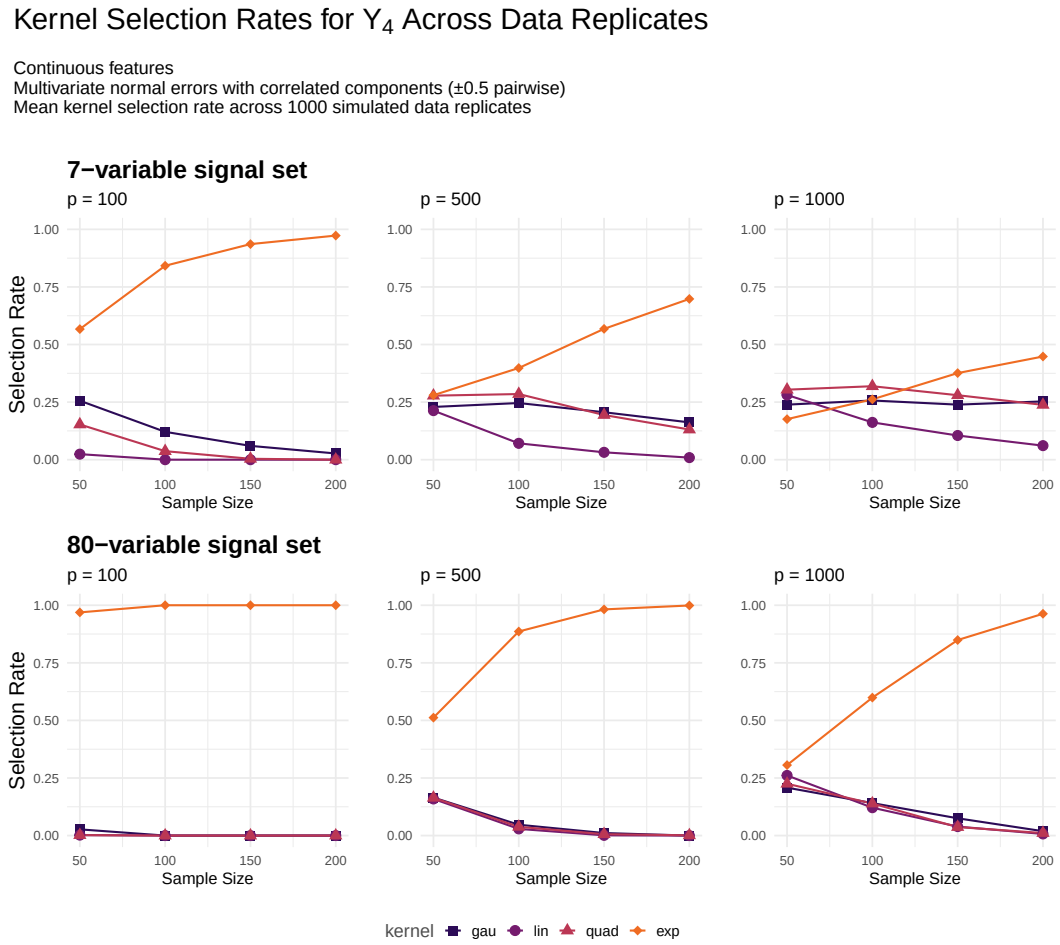


The exponential kernel was favored over the other kernel functions, more strongly so for the 80-variable signal than for the 7-variable signal. Its rate of selection increased with sample size and decreased with feature dimension. For the 7-variable signal with very high-dimensional data ($n/p \leq 0.1$), all kernel functions were selected at roughly equal frequencies.

Correlated Error Components We plot the results for scenarios where the random error vector $\boldsymbol{\epsilon}$ is simulated as having linearly correlated components, with the Pearson correlation coefficient equal to $(-1)^{j+j'}0.5$ between each pair of components ϵ_j and $\epsilon_{j'}$.

```
# Absolute value of Pearson corr coef. between pairs of error components
rho <- 0.5
```

```
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type, rho,
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle =
                   makeKerSelSubtitle(x_type, rho)))
```



The results here resemble those for the case with uncorrelated error components.

Discrete Features

This section includes scenarios where $\boldsymbol{X}$ is simulated as a discrete random vector representing additive-encoded SNP-set data.

```
x_type <- 'snp'
source(file.path(dir_src, 'initialize_adaptive_across.R'))
```

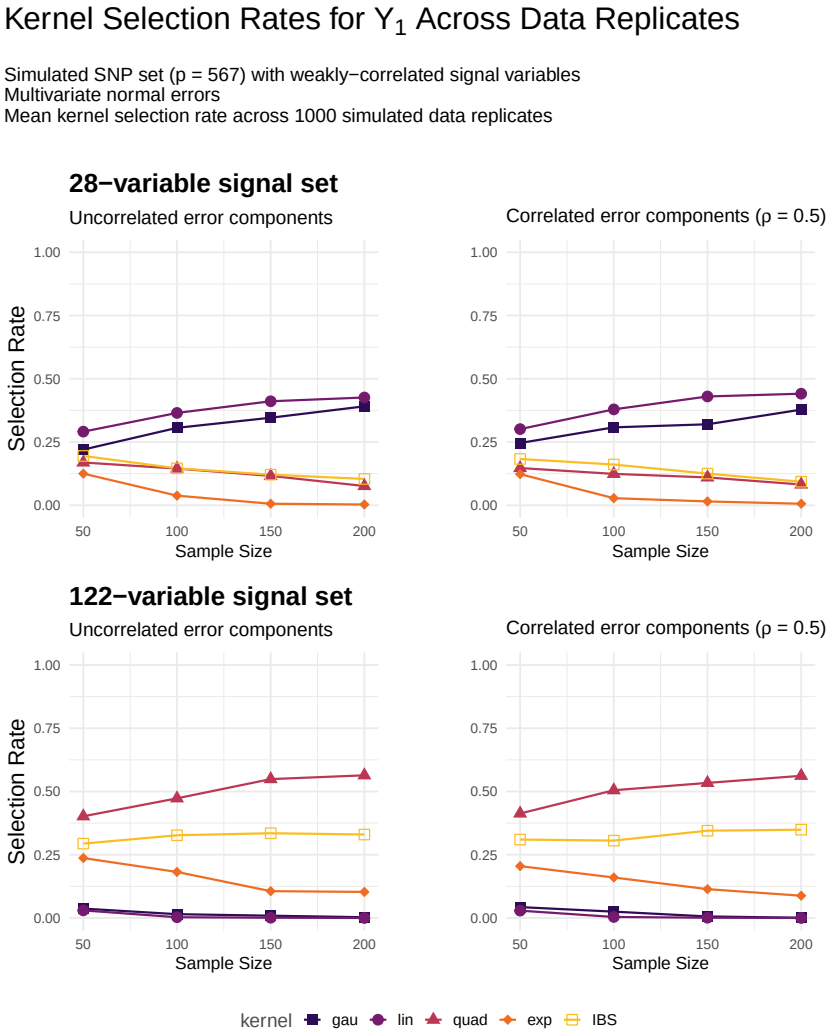
Kernel Selection Rates for Y_1

We consider the selection rate of each candidate kernel function for the 1st response component Y_1. The effect function for this response component had a linear functional form.

```
# Set index of current response component
y_ind <- 1
```

Weakly Correlated Signal Variables We consider the case where the signal variables consist of weakly-correlated SNPs.

```
signal_correlation <- "low"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



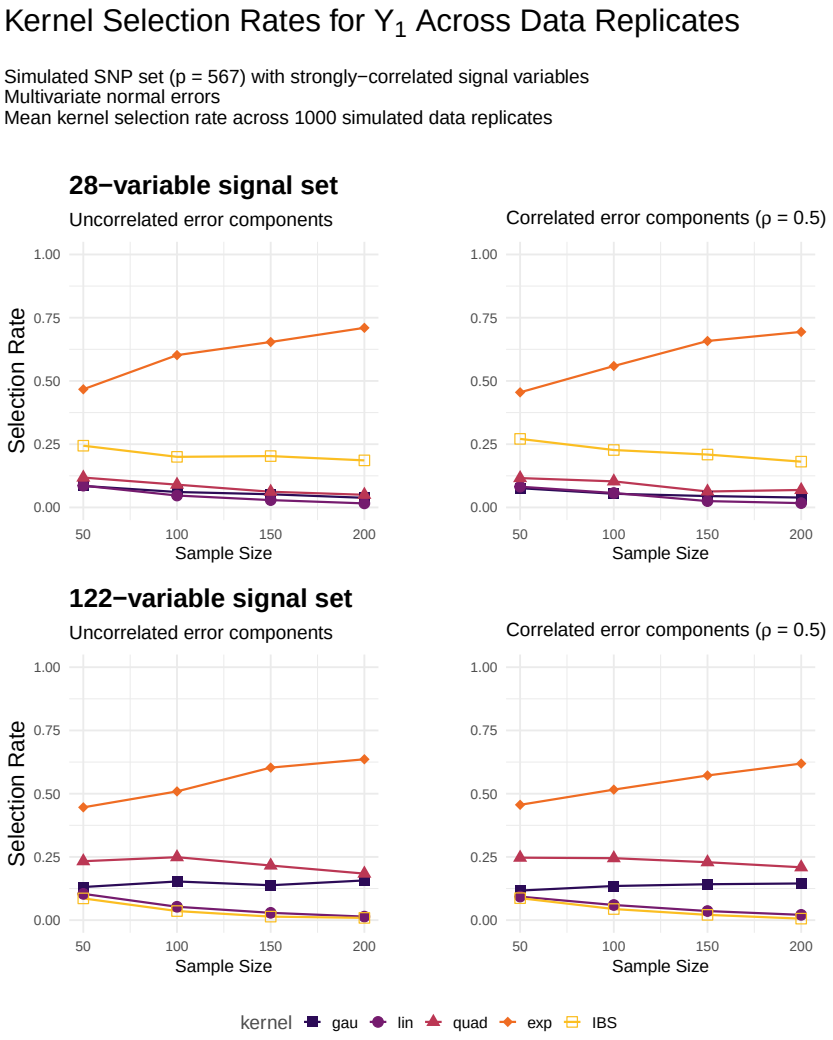
For the 28-variable signal set, the linear kernel was favored, though not much more frequently than the Gaussian kernel.

For the 122-variable signal set, the quadratic kernel was most-favored, followed by the IBS kernel and then by

the expeonential kernel; the linear and Gaussian kernels were very rarely chosen here, which is the opposite of the case observed for the 28-variable signal.

Strongly Correlated Signal Variables We consider the case where the signal variables consist of strongly-correlated SNPs.

```
signal_correlation <- "high"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



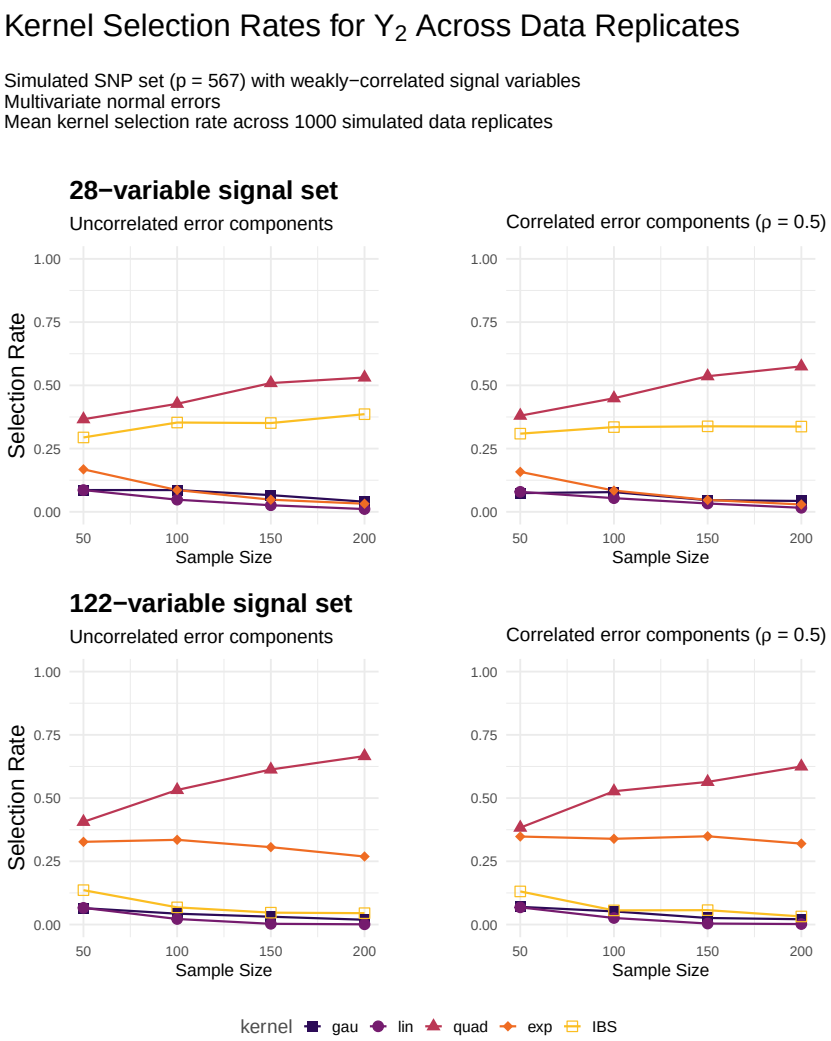
Interestingly, when the signals were strongly correlated, we see that the exponential kernel was most-favored for both signal sets, with the IBS kernel and quadratic kernels being the second preference for the 28-variable signal and 80-variable signal, respectively. In both signal sets, the linear kernel was rarely selected, despite the effect function being linear.

Kernel Selection Rates for Y_2

We consider the selection rate of each candidate kernel function for the 2nd response component Y_2. The effect function for this response component had a quadratic functional form.

```
# Set index of current response component
y_ind <- 2
```

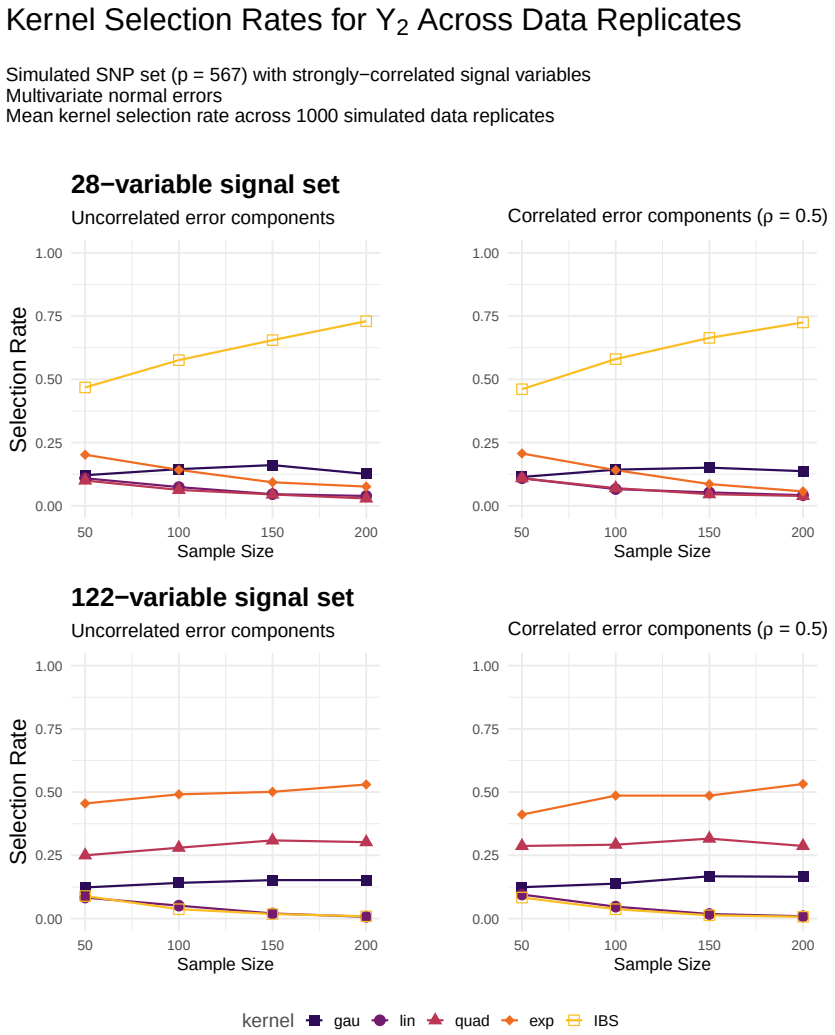
```
signal_correlation <- "low"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Weakly Correlated Signal Variables

Here the results match our expectations, with the quadratic kernel being the most frequently selected kernel in all scenarios. The second most favored kernel for the 28-variable signal was the IBS kernel, while for the 122-variable signal it was the exponential kernel. In each signal set, kernels other than the two most favored were rarely selected.

```
signal_correlation <- "high"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Strongly Correlated Signal Variables

When the signals were strongly correlated, the IBS kernel was strongly favored for the 28-variable signal set, with the other three kernels being rarely selected.

For the 122-variable signal set we observe a distinct ranking in preference for the kernels, with the exponential kernel most favored, followed by the quadratic kernel, then the Gaussian kernel, with the linear and IBS kernels rarely selected; the distinct preference for the Gaussian kernel over the linear and IBS kernels required larger sample sizes than the preference for the top two kernels.

Kernel Selection Rates for Y_3

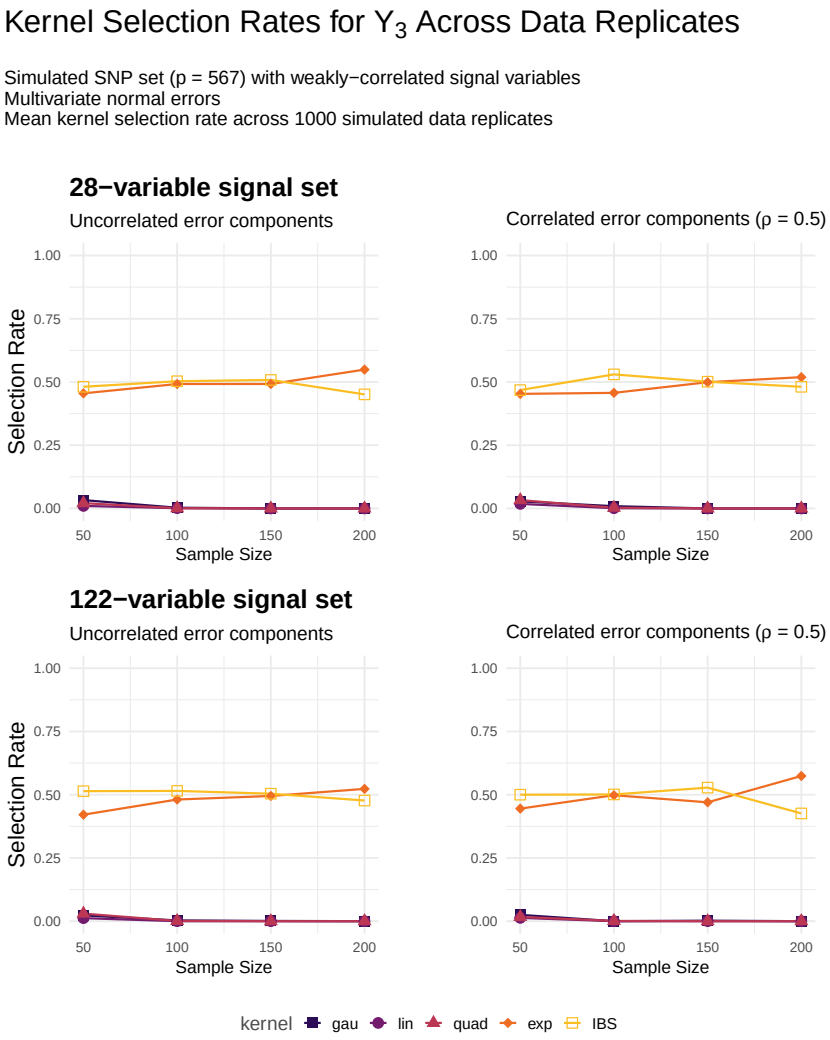
We consider the selection rate of each candidate kernel function for the 3rd response component Y_3. The effect function for this response component was nonlinear, consisting of terms of the form

$$a_1 H_d(a_2 X_k) e^{-\frac{(a_2 X_k)^2}{a_3}}$$

for some set of constant values of $a_1 \in \mathbb{R}$, $a_2 \in (0, 1]$, $a_3 \geq 1$, $d \in \{2, 3, 4\}$ and $k \in \{1, \ldots, p\}$, and where H_d is the dth degree physicist's Hermite polynomial.

```
# Set index of current response component
y_ind <- 3
```

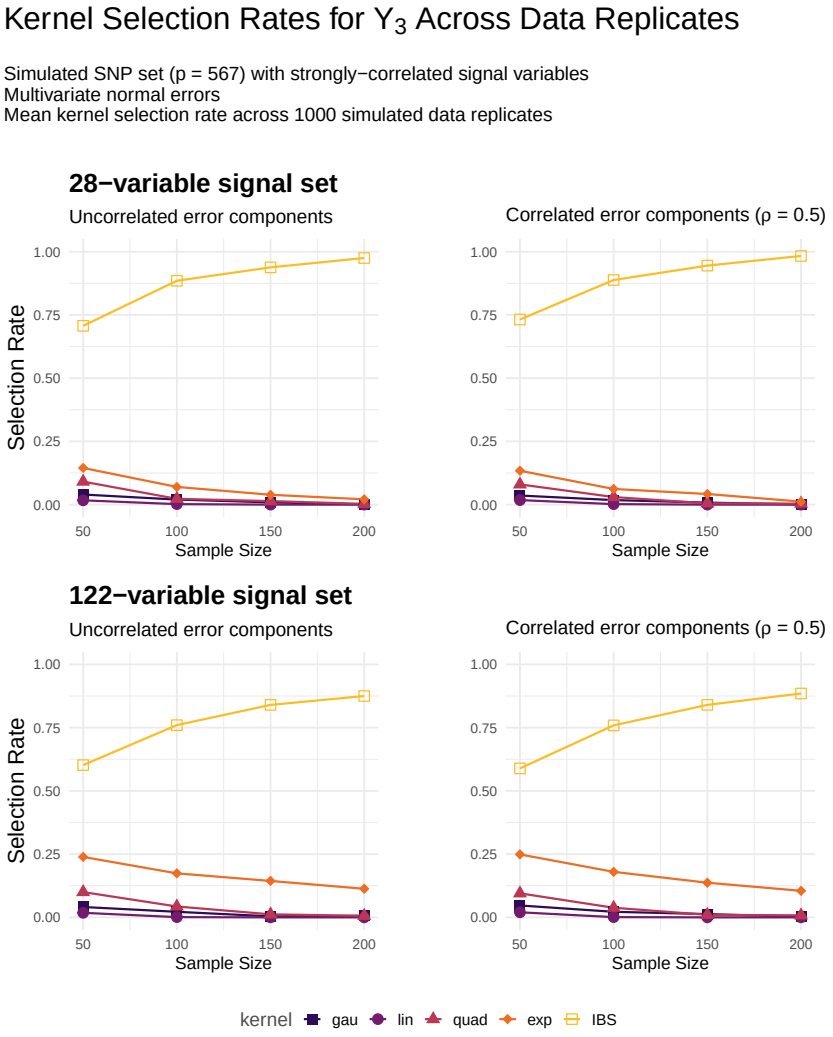
```
signal_correlation <- "low"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Weakly Correlated Signal Variables

Here the exponential and IBS kernels were both equally-favored in all scenarios, with the other three kernels almost never being selected.

```
signal_correlation <- "high"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Strongly Correlated Signal Variables

When the signals were strongly correlated, the IBS kernel became the most-favored kernel, with the exponential kernel selected much less frequently than when the signals were weakly correlated.

Kernel Selection Rates for Y_4

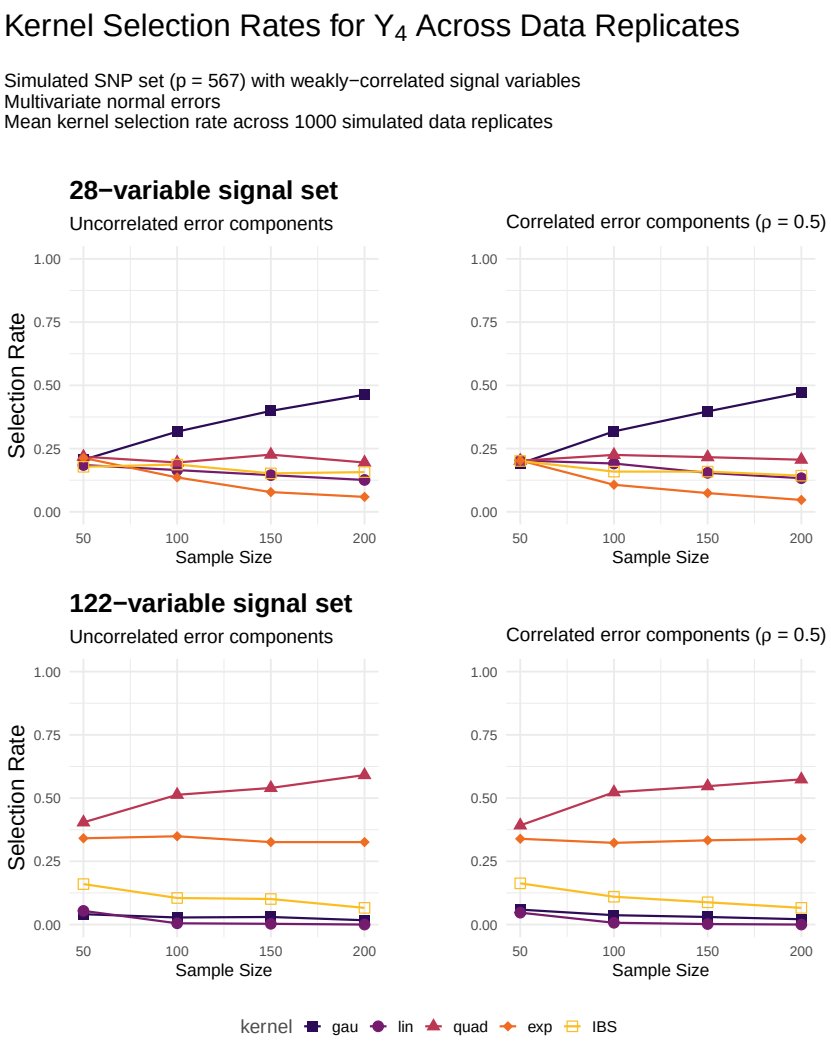
We consider the selection rate of each candidate kernel function for the 4th response component Y_4. The effect function for this response component was nonlinear, consisting of terms of the form

$$a\cos(X_k)e^{-\frac{x_k^2}{10}}$$

for some $a > 0$ and $k \in \{1, \ldots, p\}$.

```
# Set index of current response component
y_ind <- 4
```

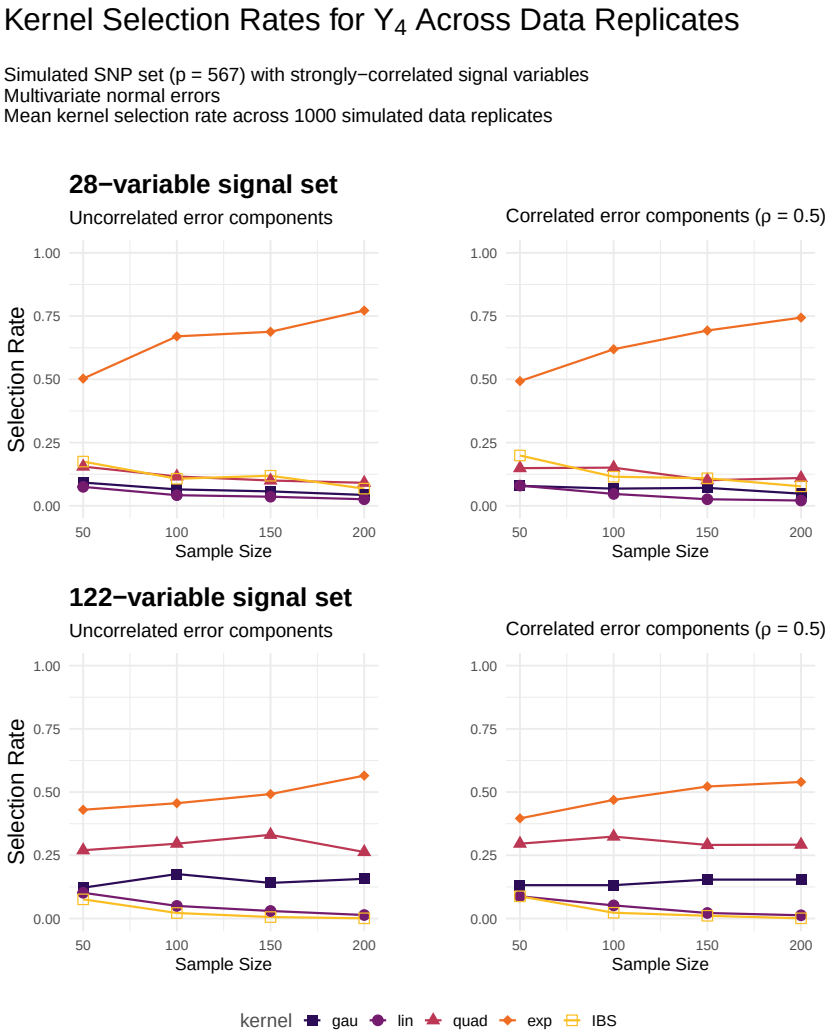
```
signal_correlation <- "low"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Weakly Correlated Signal Variables

For the 28-variable signal, the Gaussian kernel became increasingly favored as sample size grew, while for the 122-variable signal, the quadratic kernel was most favored and the exponential kernel was second most favored.

```
signal_correlation <- "high"
files <- adaptiveAcrossPlotFiles()
load(files$plotdata)
makeKerSelPlots(
  plotdata_kersel[[y_ind]], x_type,
  theme_settings =
    theme_kersel(plot_margin = margin(l = -20, b = 15)),
  title_settings =
    title_kersel(joint_title = makeKerSelTitle(y_ind),
                 joint_subtitle = makeKerSelSubtitle(x_type)))
```



Strongly Correlated Signal Variables

Here the exponential kernel was the most favored kernel for both signal sets, with its selection rate increasing with sample size.

For the 28-variable signal set, the other kernels were selected rarely with similar frequency, while for the 122-variable signal set the quadratic kernel was preferred to the Gaussian kernel, which was preferred to the linear and IBS kernels.