

Explain performance in terms of four characteristics, n=209 CPUs

```
# preliminary transformations -  
# original variables all have long right tails
```

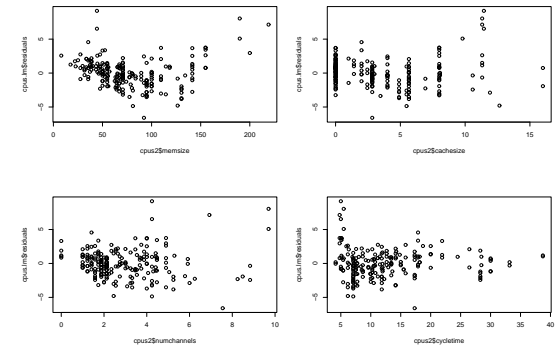
```
library("MASS")  
cpus2 <- data.frame(  
  perf=sqrt(cpus$perf),  
  memsize=sqrt((cpus$mmin+cpus$mmax)/2),  
  cachesize=sqrt(cpus$cach),  
  numchannels=sqrt((cpus$chmin+cpus$chmax)/2),  
  cycletime=sqrt(cpus$syct))
```

1

Try linear model

```
cpus.lm <- lm(perf ~ memsize + cachesize +  
             numchannels + cycletime, data=cpus2)
```

Residuals versus predictors



2

Now try additive model

```
> library("mgcv")  
> cpus.am <- gam(perf~s(memsize)+s(cachesize)+  
                s(numchannels)+s(cycletime),  
                data=cpus2)
```

```
> summary(cpus.am)  
Family: gaussian  
Link function: identity  
Formula:  
perf ~ s(memsize) + s(cachesize) + s(numchannels) +  
      s(cycletime)
```

Parametric coefficients:

	Estimate	std. err.	t ratio	Pr(> t )
(Intercept)	8.7308	0.0954	91.52	< 2.22e-16

3

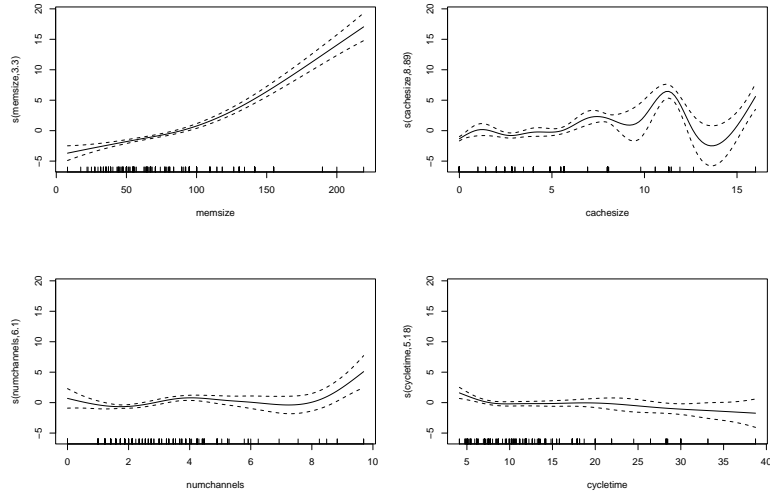
Approximate significance of smooth terms:

	edf	chi.sq	p-value
s(memsize)	3.299	376.5	< 2.22e-16
s(cachesize)	8.891	199.74	< 2.22e-16
s(numchannels)	6.096	45.903	3.6473e-06
s(cycletime)	5.177	24.626	0.0050775

R-sq.(adj) = 0.936    Deviance explained = 94.3%  
GCV score = 2.1544    Scale est. = 1.9022    n = 209

4

```
> plot(cpus.am)
```



5

Want something smoother?

```
> cpus.am2 <- gam(perf ~ s(memsize,k=4,fx=T) +
  s(cachesize,k=4,fx=T) +
  numchannels +
  s(cycletime,k=4,fx=T),
  data=cpus2)
```

```
> summary(cpus.am2)
```

Family: gaussian

Link function: identity

Parametric coefficients:

	Estimate	std. err.	t ratio	Pr(> t )
(Intercept)	7.7403	0.2864	27.02	< 2.22e-16
numchannels	0.33988	0.09059	3.752	0.00023042

6

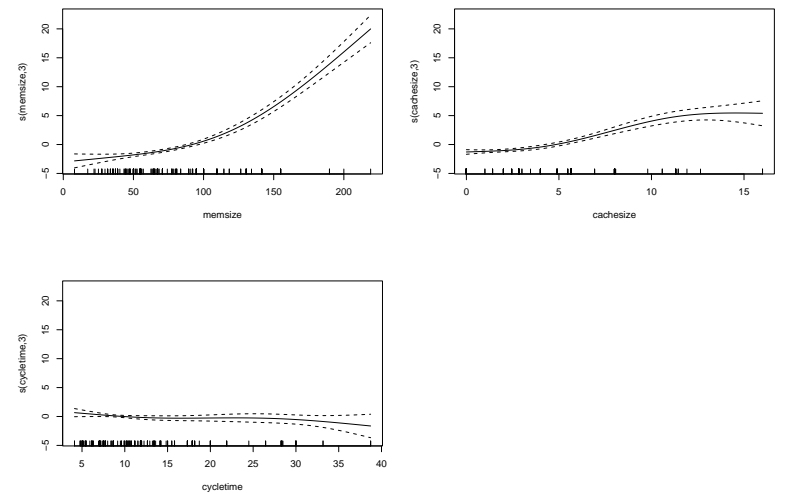
Approximate significance of smooth terms:

	edf	chi.sq	p-value
$s(\text{memsize})$	3	355.77	< 2.22e-16
$s(\text{cachesize})$	3	120.7	< 2.22e-16
$s(\text{cycletime})$	3	5.1098	0.16763

R-sq.(adj) = 0.913    Deviance explained = 91.7%  
 GCV score = 2.7259    Scale est. = 2.5824    n = 209

7

```
> plot(cpus.am2)
```



8