Homework 8

Advanced Methods for Data Analysis (36-402)

Due Wednesday April 8, 2022 at 6:00 pm EDT

Solutions – not to be posted online or shared, even after the end of the semester.

You should **always show all your work** and submit both a writeup and *R* code.

- Assignments must be submitted through Gradescope as a PDF. Follow the instructions here: `https://www.cmu.edu/teaching/gradescope/`
- Gradescope will ask you to mark which parts of your submission correspond to each homework problem. This is mandatory; if you do not, grading will be slowed down, and your assignment will be penalized.
- Make sure your work is legible in Gradescope. You may not receive credit for work the TAs cannot read. **Note:** If you submit a PDF with pages much larger than 8.5 × 11", they will be blurry and unreadable in Gradescope.
- For questions involving R code, we strongly recommend using R Markdown. The relevant code should be included with each question, rather than in an appendix. A template Rmd file is provided on Canvas.

1. **Gross Metropolitan Product.** In Data Exam 1, you used data on the per-capita gross metropolitan product of cities to explore the relationship between population size and income. The power-law scaling hypothesis says that

$$Y \approx \beta N^{\gamma}, \tag{1}$$

where Y is the per-capita GMP and N is the population.

In this problem, you will compare the power-law and urban-hierarchy models of `pcgmp`, but using additive models, which provide a more direct way to do the analysis you did in the Data Exam.

As a reminder, the data file `gmp.csv` contains 133 observations with the following variables:

MSA the name of each metropolitan area;

pcgmp its per-capita gross metropolitan product, in dollars;

pop its population;

finance the share of its economy derived from finance;

prof.tech the share of its economy derived from “professional and technical services”;

ict the share of its economy derived from “information, communication and technology” (ICT); and

management the share of its economy derived from “management of firms and enterprises.”

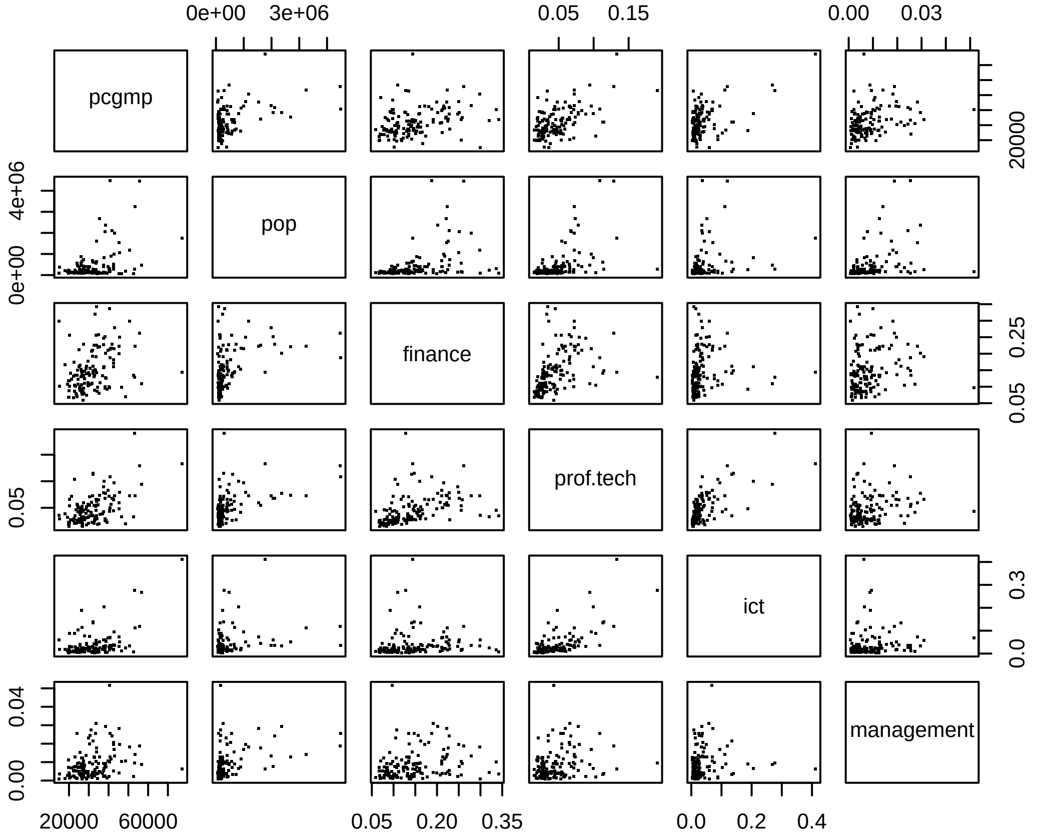
To fit models in this problem, use the package `mgcv`, which fits additive models and allows the degrees of freedom to be chosen by generalized cross-validation (GCV). See Demo 9.1 for examples.

A reminder on the syntax of `mgcv`’s `gam` function and the model formulas it accepts: If you specify a component of the additive model as `s(ict)`, for example, the effective degrees of freedom (edf) for this spline will be chosen by GCV. If you want to fix the edf at `d`, specify `s(ict, k = d + 1, fx = TRUE)`. By default, the `k` parameter sets an upper bound for the edf, and it counts an intercept for each spline, which gets removed before the fitting is finished. The `fx = TRUE` argument forces the specified edf to be the value of `k` (minus 1) rather than using `k` as an upper bound and doing GCV to choose the best edf.

Look at the `summary` for your fits. The chosen edf for each (non-linear) predictor is reported in `summary`.

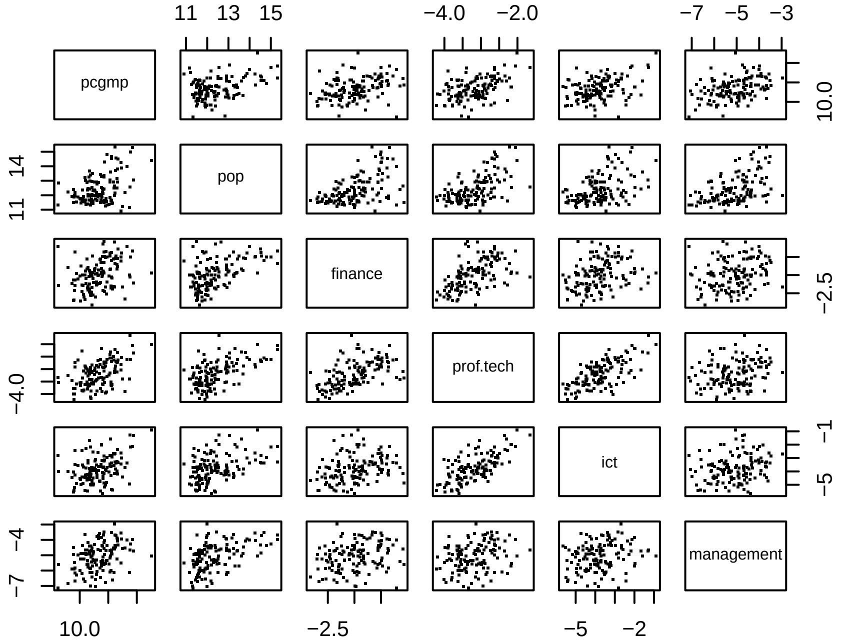
Solution:

```r
suppressPackageStartupMessages(library(mgcv))

# Read the data
gmp <- read.csv("gmp.csv", header = TRUE)

# Here is a pairs plot of the variables we will use as part of an
# exploratory analysis:
pairs(gmp[, 3:8], pch=".")
```



Before getting to deeply into the analysis of this data set, we should note that several, if not all, of the variables have long upper tails. Taking logarithms of all of the variables could make it easier to see the relationships between them.

```
pairs(log(gmp[, 3:8]), pch=".")
```



As we see below, a linear relationship in the logs corresponds to a power-law relationship in the original variables.

(a) As you did in the Data Exam, fit the linear version of the power law (1):

$$\log(Y) = \alpha + \gamma \log(N) + \varepsilon.$$

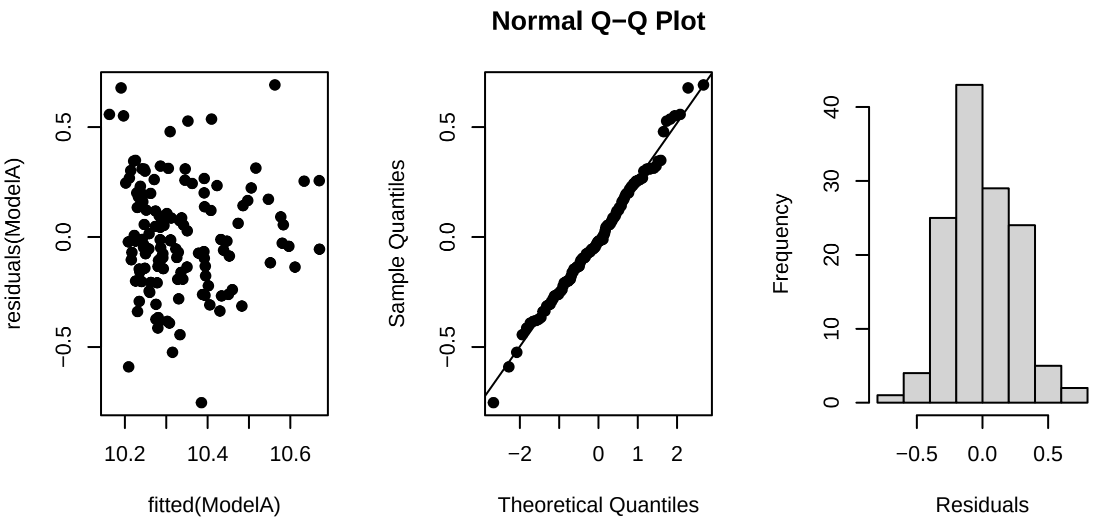
Call this Model A. Look at residuals to see if there are any assumptions of a linear model that appear to be violated.

Solution: The linear model is fit below, however we use `gam` to do the fit so that we have access to the same model features that we will use for the other models.

```
ModelA <- gam(log(pcgmp) ~ log(pop), data=gmp)
summary(ModelA)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcgmp) ~ log(pop)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.90287    0.28802  30.910  < 2e-16 ***
## log(pop)     0.11541    0.02316   4.984 1.93e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.153   Deviance explained = 15.9%
## GCV =  0.067  Scale est. = 0.065992  n = 133
```

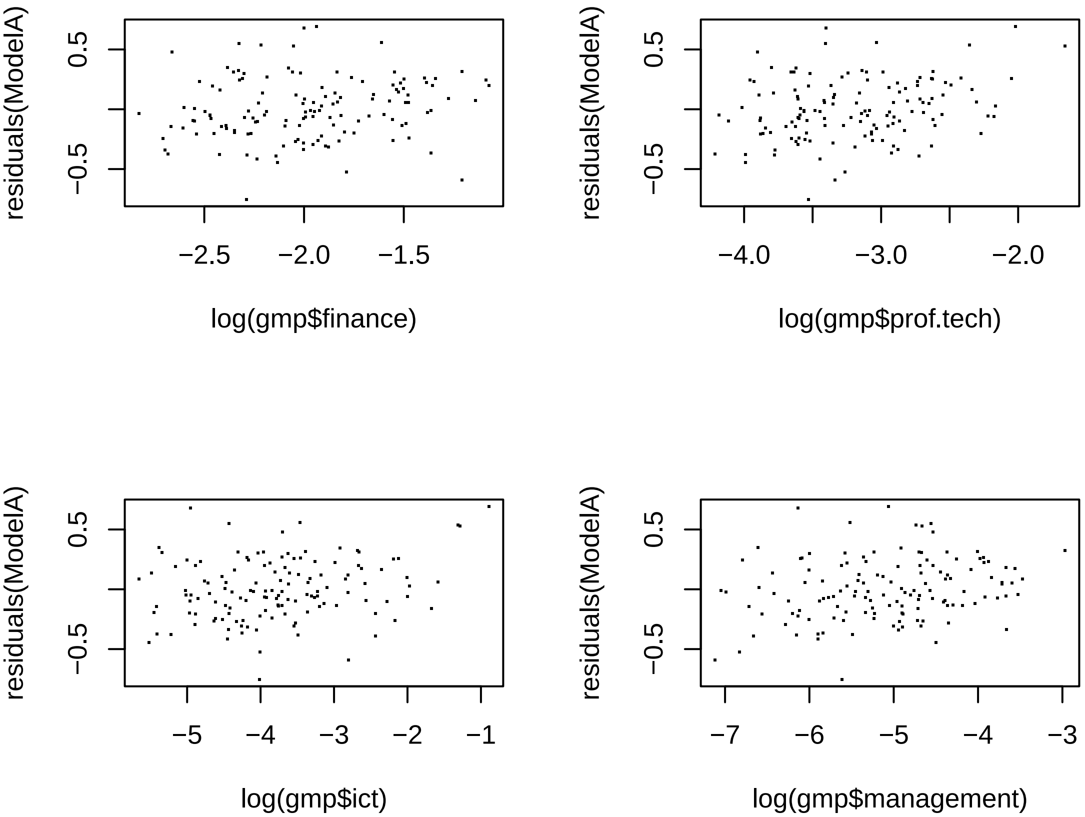
Now let's consider the residual diagnostics.

```
par(mfrow = c(1, 3))
plot(fitted(ModelA), residuals(ModelA), pch=19)
qqnorm(residuals(ModelA), pch=19)
qqline(residuals(ModelA))
hist(residuals(ModelA), xlab = "Residuals", main = "")
```



The plot against fitted values is not so bad. Considering some of the data sets we've seen recently, the normal plot is remarkably straight. Plotting residuals against the covariates *not* included in the model, we find:

```
par(mfrow=c(2,2))
plot(log(gmp$finance), residuals(ModelA), pch=".")
plot(log(gmp$prof.tech), residuals(ModelA), pch=".")
plot(log(gmp$ict), residuals(ModelA), pch=".")
plot(log(gmp$management), residuals(ModelA), pch=".")
```



There appear to be some positive relationships between the residuals and the predictors that were not included, although it is difficult to assess by eye how strong these relationships are.

(b) Fit an additive model for $\log(Y)$ in which $\log(N)$ enters linearly as in Model A, and the logarithms of the four "share" variables each enter non-linearly as splines with 4 degrees of freedom. For example, the term for `finance` would be `s(log(finance), k = 5, fx = TRUE)`. Call this Model B.

Look at the residuals and model summary. Comment on what they tell you.

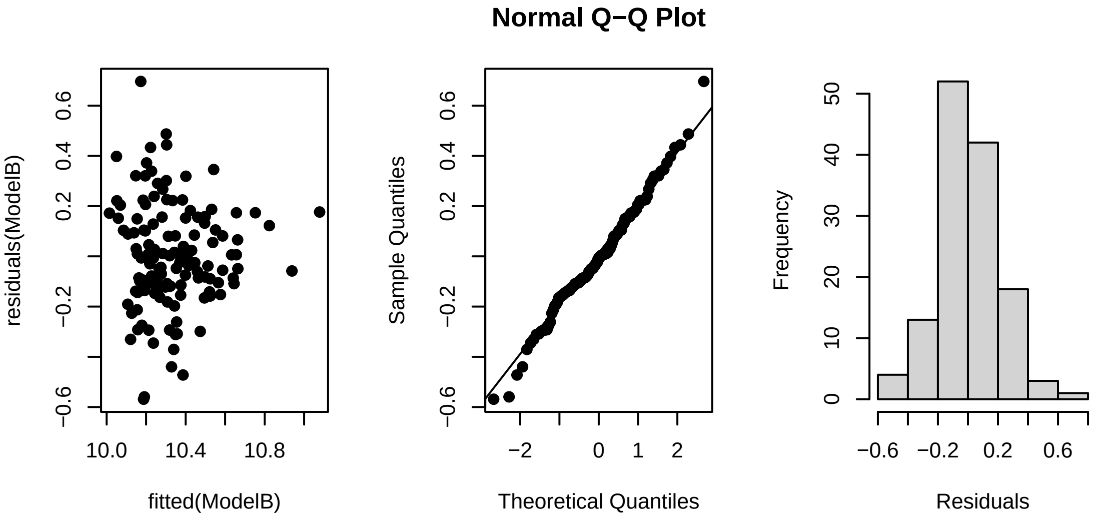
Solution: We start by fitting Model B:

```
ModelB <- gam(log(pcgmp) ~ log(pop) +
                  s(log(finance), k=5, fx=TRUE) +
                  s(log(prof.tech), k = 5, fx=TRUE)+
                  s(log(ict), k=5, fx = TRUE) +
                  s(log(management), k=5, fx=TRUE),
              data=gmp)
summary(ModelB)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(pcgmp) ~ log(pop) + s(log(finance), k = 5, fx = TRUE) + s(log(prof.tech),
##     k = 5, fx = TRUE) + s(log(ict), k = 5, fx = TRUE) + s(log(management),
##     k = 5, fx = TRUE)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.75356    0.39448  27.260   <2e-16 ***
## log(pop)    -0.03383    0.03177  -1.065    0.289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                    edf Ref.df     F p-value
## s(log(finance))      4      4 2.356 0.05780 .
## s(log(prof.tech))    4      4 1.255 0.29183
## s(log(ict))          4      4 3.030 0.02041 *
## s(log(management))   4      4 4.395 0.00242 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.346   Deviance explained =   43%
## GCV = 0.058945  Scale est. = 0.050968  n = 133
```

The model summary suggests that `log(pop)` isn't contributing much to the fit, in the presence of the other predictors; its coefficient is much smaller in size than it was in Model A. It also appears that `log(finance)` and `log(prof.tech)` might not contribute much to the fit either, compared to the other two covariates.

Now let's examine the residuals:

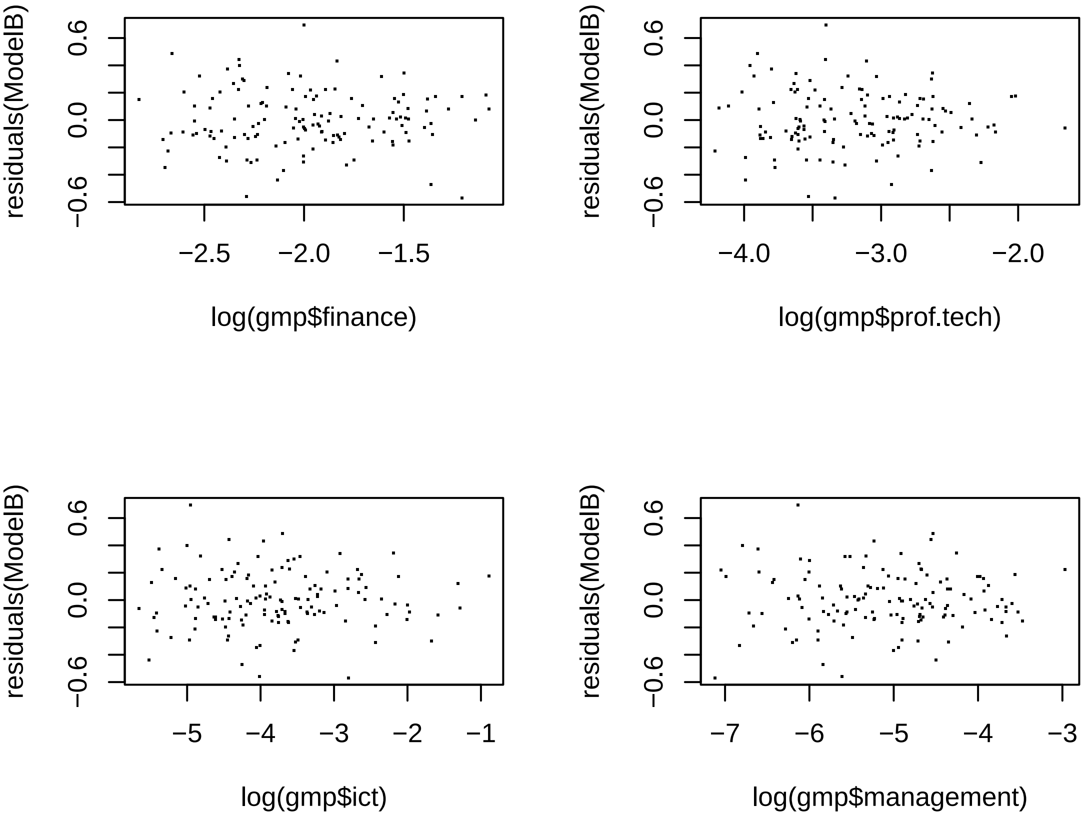
```
par(mfrow = c(1, 3))
plot(fitted(ModelB), residuals(ModelB), pch=19)
qqnorm(residuals(ModelB), pch=19)
qqline(residuals(ModelB))
hist(residuals(ModelB), xlab = "Residuals", main = "")
```



The residuals, although smaller, appear to be less normal, in that they seem to have heavier tails than a normal distribution. This effect is quite small, though.

Plotting residuals against the four smoothed covariates, we get

```
par(mfrow=c(2,2))
plot(log(gmp$finance), residuals(ModelB), pch=".")
plot(log(gmp$prof.tech), residuals(ModelB), pch=".")
plot(log(gmp$ict), residuals(ModelB), pch=".")
plot(log(gmp$management), residuals(ModelB), pch=".")
```



The positive relationships between residuals and the predictors have been removed, as one would expect.

(c) The F test allows us to compare two nested models and test the null hypothesis that the simpler model is correct. (By "nested," we mean that one model includes the other as a special case. Here, Model B includes Model A, because if the "share" variables have zero association with the outcome, the model is the same as Model A.) The `anova` function in R can conduct an F test:

```
anova(model_a, model_b, test = "F")
```

Review your 36-401 notes for details on the F statistic, the assumptions required for it to be F-distributed, and the interpretation of the results.

Use an F test to test the null hypothesis that Model A is correct against the alternative that the larger Model B is correct. Is there evidence that the additional terms in Model B are really needed?

For each of the four "share" variables, comment on the extent to which a non-linear fit is really needed. Use the `plot` function to draw the partial response functions, and see if these might contain some information.

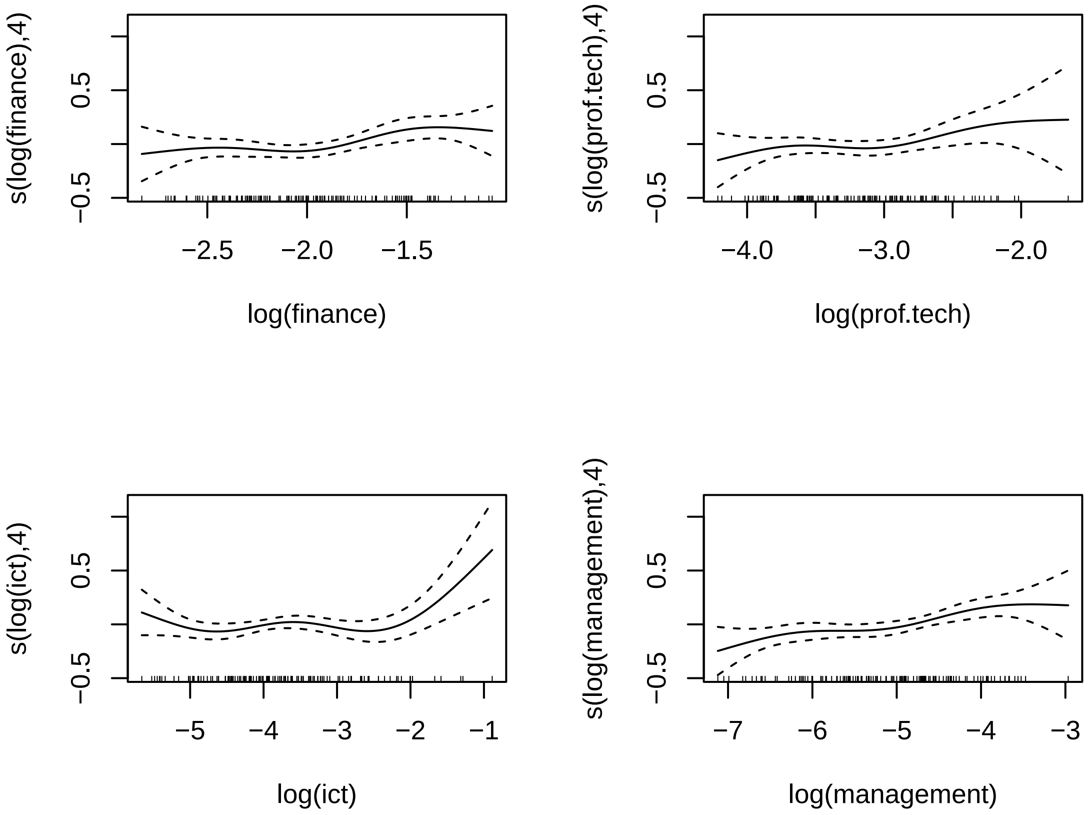
Solution: We can use the `anova` command to compare the two models with the F test:

```
anova(ModelA, ModelB, test = "F")
## Analysis of Deviance Table
##
## Model 1: log(pcgmp) ~ log(pop)
## Model 2: log(pcgmp) ~ log(pop) + s(log(finance), k = 5, fx = TRUE) + s(log(prof.tech)
##     k = 5, fx = TRUE) + s(log(ict), k = 5, fx = TRUE) + s(log(management),
##     k = 5, fx = TRUE)
##   Resid. Df Resid. Dev Df Deviance      F    Pr(>F)
## 1       131     8.6450
## 2       115     5.8613 16   2.7837 3.4136 6.137e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p value is fairly small, suggesting that Model B is a better fit to the data than Model A. (Specifically, we reject the null hypothesis that the additional terms in Model B have zero effect.) This suggests the additional terms are indeed useful.

The partial response functions look like this:

```
par(mfrow = c(2, 2))
plot(ModelB)
```



Two of the “share” predictors have rather flat partial response functions. Perhaps a third model without those terms would be a useful comparison.

(d) Compute 90% confidence intervals for the response function from Model B at the predictor vectors corresponding to the observations numbered 16, 25, and 81 in the data set. The `predict` function, when applied to an additive model object, will compute the standard errors that are the square-roots of what we called $\hat{s}^2(\hat{Y}_i)$ in the lecture notes. (Use `help(predict.gam)` in R to learn the options for the `predict` function that works with `gam` objects.)

Compare these confidence intervals to pivotal bootstrap confidence intervals based on resampling cases. Use 1000 bootstrap samples. Use the same bootstrap samples to estimate the bias and standard deviation of the three $\hat{r}(x_i)$ values.

Do the biases appear large? Are the `se.fit` values produced by the `predict` function close to the standard deviations estimated by the bootstrap? Do the three sets of bootstrap values of $\hat{r}(x_i)$ appear to have approximately the t distributions that correspond to confidence intervals that the were computed without the bootstrap?

Solution: We can use the `predict` function to get confidence intervals based on the assumption that

$$T_i = \frac{\hat{r}(x_i) - r(x_i)}{\hat{s}(\hat{Y}_i)}, \tag{2}$$

has approximately a $t_{n-\nu}$ distribution where ν is the effective degrees of freedom for the fitted model ($\nu = 18$ for Model B, so $n - \nu = 115$).

```
newdat <- gmp[c(16, 25, 81),]
t_df <- nrow(gmp) - 18

ciparts <- predict(ModelB, newdat, se.fit=TRUE)

## The rows of this matrix are the three intervals
cbind(ciparts$fit - qt(0.95, t_df) * ciparts$se.fit,
      ciparts$fit - qt(0.05, t_df) * ciparts$se.fit)
##        [,1]     [,2]
## 16 10.33479 10.55798
## 25 10.36786 10.62261
## 81 10.17115 10.43351
```

Since the F statistic did not have a distribution very close to the nominal F distribution, perhaps the T_i in (2) does not have the nominal t_{115} distribution. This time, we will use the "resample-cases" bootstrap since we want the distribution of $\hat{r}(x_i)$ based on the distribution from which the data were drawn.

```
B <- 1000
n <- nrow(gmp)
centers <- matrix(NA, nrow = 3, ncol = B)

for (b in 1:B) {
    cases <- sample(n, replace=TRUE)
    thedata <- gmp[cases,]
    thefit <- gam(log(pcgmp) ~ log(pop) +
                      s(log(finance), k=5, fx=TRUE) +
                      s(log(prof.tech), k=5, fx=TRUE) +
                      s(log(ict), k = 5, fx = TRUE) +
                      s(log(management), k = 5, fx = TRUE),
                  data = thedata)
    centers[, b] <- predict(thefit, newdat)
}
```

We get the quantiles of the bootstrap distributions with

```
quants <- apply(centers, 1, quantile, prob = c(0.05, 0.95))
```

We can then construct the three pivotal confidence intervals (as the rows of this matrix):

```
cbind(2 * ciparts$fit - quants[2,], 2 * ciparts$fit - quants[1,])
##        [,1]     [,2]
## 16 10.32528 10.55945
## 25 10.37335 10.60753
## 81 10.17264 10.46700
```

These are remarkably similar to the confidence intervals based on the t distribution.

The biases are

```
apply(centers, 1, mean) - ciparts$fit
##           16           25           81
##  0.004597376 -0.008206256 -0.006471600
```

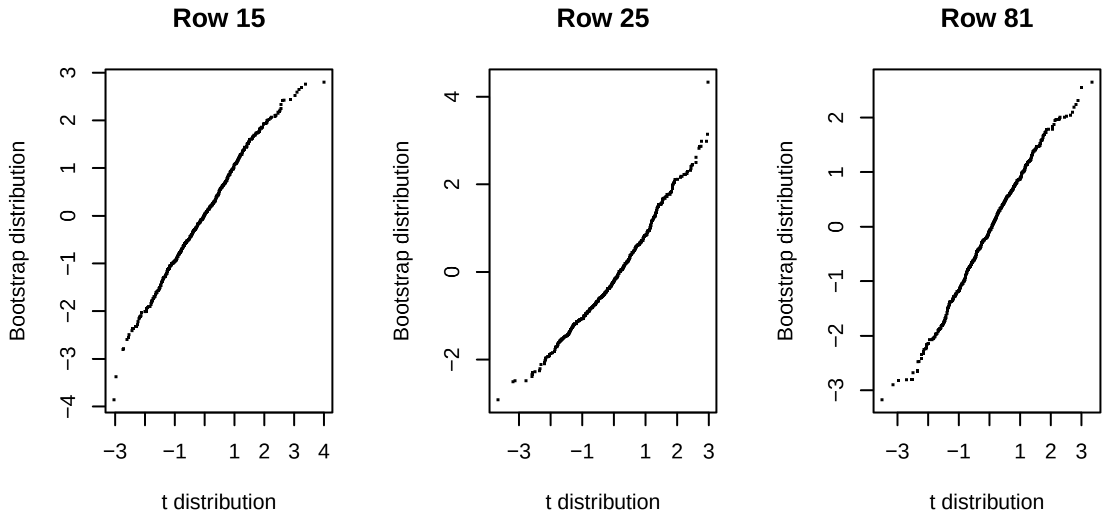
The biases are only small fractions of the sizes of the fitted values.

The standard deviations $\hat{s}(\hat{Y}_i)$ estimated by the bootstrap are simply

```
sds <- apply(centers, 1, sd)
sds
## [1] 0.06957967 0.07177132 0.08817202
```

We conclude by drawing quantile plots for the three sets of bootstrap pivotal $\tilde{r}(x_i) - \hat{r}(x_i)$ values, to see if they match the expected t distributions. You're probably familiar with using `qqnorm` to check normality of samples (such as of residuals), but one can make a Q-Q plot to check against just about any distribution. The `qqplot` function can compare the distributions of two vectors against each other, so we can provide a vector of values with the right t distribution and produce the comparisons:

```r
par(mfrow = c(1, 3))
qqplot(rt(B, t_df), (centers[1, ] - ciparts$fit[1]) / sds[1],
       xlab = "t distribution",
       ylab = "Bootstrap distribution",
       main = "Row 15", pch = ".")
qqplot(rt(B, t_df), (centers[2, ] - ciparts$fit[2]) / sds[2],
       xlab = "t distribution",
       ylab = "Bootstrap distribution",
       main = "Row 25", pch = ".")
qqplot(rt(B, t_df), (centers[3, ] - ciparts$fit[3]) / sds[3],
       xlab = "t distribution",
       ylab = "Bootstrap distribution",
       main = "Row 81", pch = ".")
```



The apparent distributions of the bootstrap $\tilde{r}(x_i)$ values are close to the nominal $t_{n-\nu}$ distribution for row 15. Rows 25 and 81 have something happening with their right tails.

You were not expected to know about the following, but it is highly suggested by the presentation above. It is an alternative (even more pivotal) form of pivotal confidence interval. It is based on the distribution of the T_i in (2). While running the same bootstrap as above, we could also obtain bootstrap samples of

$$\tilde{T}_{b,i} = \frac{\tilde{r}(x_i)_b - \hat{r}(x_i)}{\tilde{s}(\hat{Y}_i)_b},$$

for $i = 15, 24, 80$ and $b = 1, \ldots, B$, where $\tilde{r}(x_i)_b$ is the ith fitted value in the bth bootstrap sample and $\tilde{s}(\hat{Y}_i)_b$ is the estimated standard error of the ith fitted value in the bth bootsrap sample. If $\tilde{q}_1$ and $\tilde{q}_2$ are respectively the 0.05 and 0.95 quantiles of the empirical distribution of $\tilde{T}_{b,i}$, then a pivotal confidence interval for $r(x_i)$ is

$$\left[\hat{r}(x_i) - \tilde{q}_2\hat{s}(\hat{Y}_i), \hat{r}(x_i) - \tilde{q}_1\hat{s}(\hat{Y}_i)\right].$$

The advantage of such intervals is that the distribution of $\tilde{T}_{b,i}$ is less dependent on specific features of the empirical distribution of the data than is the distribution of $\tilde{r}(x_i)_b - \hat{r}(x_i)$.