Homework 5 Solution

Due: 4/10/2019 before 11pm. Submit in Canvas (file upload). Rmd file and the html output file (submit both files) are strongly recommended, but not required.

1. [**70 points**]

Consider a simple version of functional data regression on the `fbiwide` data in `data and code` folder in Canvas. Drop the variable `Rape` due to too many missing values.

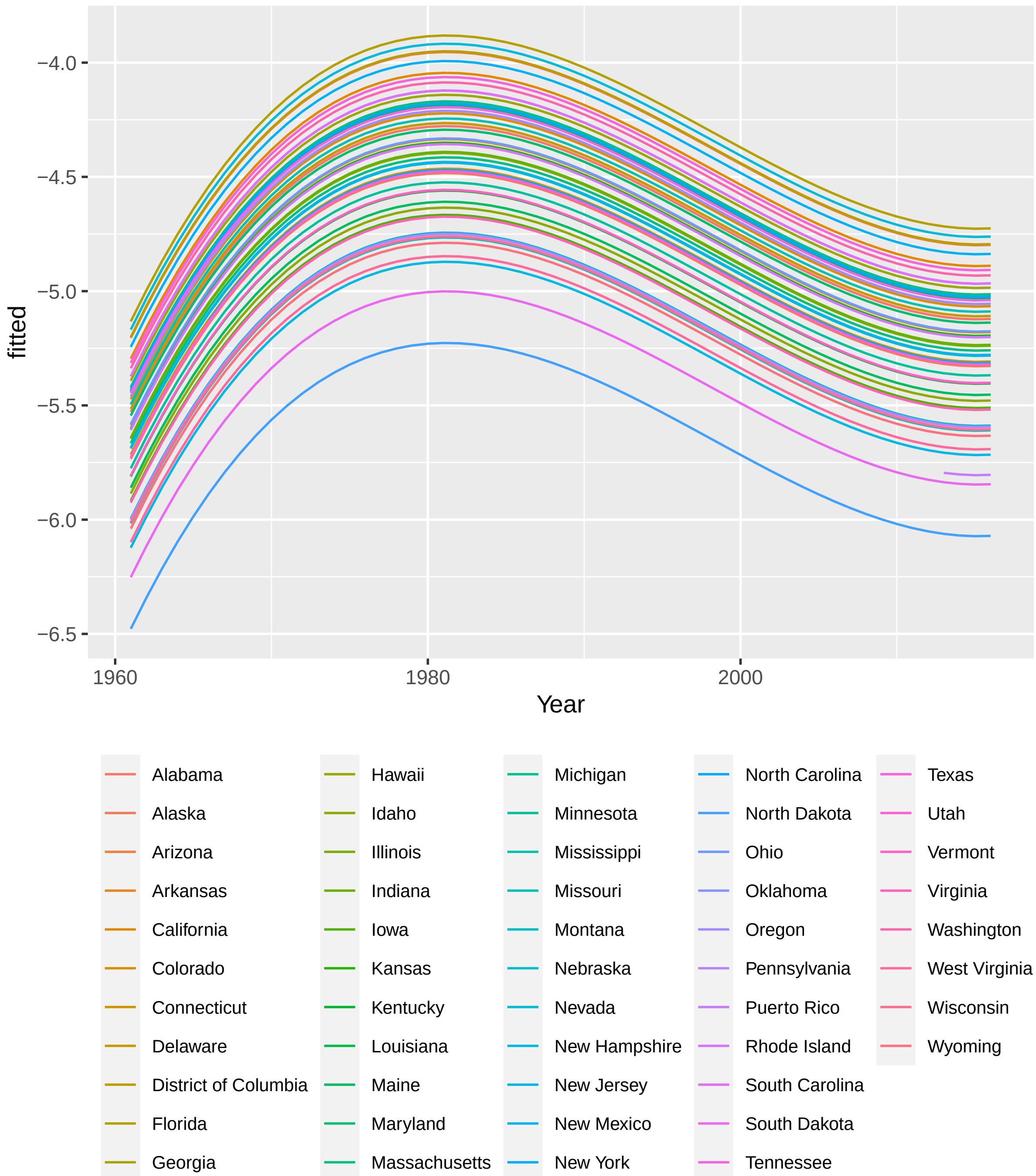
a. Use the log transformation of `crime counts over Population` (expect `Rape`) as response. Use $State$, $Year - 1961$, $(Year - 1961)^2$, $(Year - 1961)^3$ as covariates. Fit the regression model and output the anova analysis results for each covariate. (Delete the obervations with missing values in the data, after dropping the variable `Rape`.) [10 points]

```
library(tidyverse)

fbi <- classdata::fbiwide %>%
  select(-Abb, -Rape) %>%
  mutate_at(vars(-State, -Year, -Population), ~ log(. / Population)) %>%
  mutate(Time = Year - 1961) %>%
  select(Time, everything()) %>%
  na.omit()

fit_lm <- lm(as.matrix(fbi[, -(1:4)]) ~ State + Time + I(Time ^ 2) + I(Time ^ 3), data = fbi)
car::Manova(fit_lm)
```

```
##
## Type II MANOVA Tests: Pillai test statistic
##             Df test stat approx F num Df den Df    Pr(>F)
## State       51    3.9055    69.29    357  19600 < 2.2e-16 ***
## Time         1    0.6721   818.20      7   2794 < 2.2e-16 ***
## I(Time^2)    1    0.5255   442.02      7   2794 < 2.2e-16 ***
## I(Time^3)    1    0.4177   286.34      7   2794 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

b. Use ggplot to plot the regression curve of `Burglary` over time for all the states. [10 points]

```
mutate(fbi, fitted = fit_lm$fitted.values[, "Burglary"]) %>%
  qplot(x = Year, y = fitted, color = State, data = ., geom = "line") + labs(color = NULL) +
  theme(legend.position = "bottom", legend.text = element_text(size = 8))
```



c. Construct simultaneous 95% prediction intervals for all the responses in the model at Iowa in 2019. [10 points]

```
predict.mlm <- function(object, newdata, level = 0.95, interval = c("confidence", "prediction")) {
  interval <- match.arg(interval)
  n <- nrow(object$model)
  r <- object$rank - 1
  p <- ncol(object$coef)

  Z <- model.matrix(object)
  terms <- delete.response(terms(object))
```

```r
  z0 <- model.matrix(terms, newdata, contrasts.arg = object$contrasts, xlev = object$xlevels)

  pred <- z0 %*% object$coef
  flag <- switch(interval, confidence = 0, prediction = 1)
  se <- sqrt(diag(z0 %*% solve(crossprod(Z)) %*% t(z0)) + flag) %o% sigma(object)
  # sigma(object) ^ 2 == SSE / (n - r - 1)

  pred %o% c(1, 1) + sqrt(p * (n - r - 1) / (n - r - p) * qf(level, p, n - r - p)) * se %o% c(-1, 1
}

predict(fit_lm, newdata = data.frame(State = "Iowa", Time = 2019 - 1961), interval = "predict")[1,
```

```
##                            [,1]       [,2]
## Aggravated.assault   -7.752787  -5.562681
## Burglary             -6.305276  -4.649747
## Larceny.theft        -4.882904  -3.550346
## Legacy.rape          -9.775235  -7.650511
## Motor.vehicle.theft  -8.282940  -6.032247
## Murder              -12.153580 -10.061437
## Robbery              -9.190620  -6.898705
```

d. Use `linearHypothesis` function and `anova` function in R to test for the signifcance of the 3 polynomial terms of `Year`. Do the two tests have the same results? [10 points]

```r
fit_lm2 <- lm(as.matrix(fbi[, -(1:4)]) ~ State, data = fbi)
anova(fit_lm, fit_lm2)
```

```
## Analysis of Variance Table
##
## Model 1: as.matrix(fbi[, -(1:4)]) ~ State + Time + I(Time^2) + I(Time^3)
## Model 2: as.matrix(fbi[, -(1:4)]) ~ State
##   Res.Df Df Gen.var. Pillai approx F num Df den Df    Pr(>F)
## 1   2800    0.049210
## 2   2803  3 0.082737 1.8137   610.67     21   8388 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
C <- cbind(matrix(0, 3, 52), diag(3))
car::linearHypothesis(fit_lm, hypothesis.matrix = C) %>% print(SSP = F)
```

```
##
## Multivariate Tests:
##                   Df test stat  approx F num Df  den Df     Pr(>F)
## Pillai             3  1.813697  610.6727     21 8388.00 < 2.22e-16 ***
## Wilks              3  0.026136  977.3155     21 8023.41 < 2.22e-16 ***
## Hotelling-Lawley   3 10.259345 1364.3301     21 8378.00 < 2.22e-16 ***
## Roy                3  7.396180 2954.2455      7 2796.00 < 2.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

They have the same results.

e. Use principal components analysis to reduce the dimensionality of the crimes into fewer dimensions. How many principal components should be chosen? Explain the meaning of the leading principal components. Notice that the data need to be centered separately for each state first. [10 points]

```r
source("PCs.proportion.variation.enuff.R")
options(digits = 3)

fbi_centered <- fbi %>%
  select(-Time, -Year, -Population) %>%
  group_by(State) %>%
  mutate_at(vars(-group_cols()), ~ . - mean(.)) %>%
  ungroup() %>%
  select(-State)
fbi_pca <- prcomp(fbi_centered)
fbi_pca$rotation
```

```
##                       PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Aggravated.assault  0.503  0.410  0.329 -0.547  0.3929  0.0639 -0.1152
## Burglary            0.284 -0.512 -0.113  0.213  0.4765 -0.3645 -0.4896
## Larceny.theft       0.329 -0.110 -0.247  0.088  0.2972 -0.1865  0.8292
## Legacy.rape         0.492  0.459 -0.073  0.417 -0.4408 -0.3946 -0.1312
## Motor.vehicle.theft 0.282 -0.420 -0.267 -0.610 -0.5324 -0.1307 -0.0212
## Murder              0.163 -0.384  0.844  0.166 -0.2211 -0.0405  0.1887
## Robbery             0.458 -0.152 -0.170  0.273 -0.0599  0.8086 -0.0784
```

```r
pvals <- sapply(1:ncol(fbi_centered), function(i)
  PCs.proportion.variation.enuff(fbi_pca$sdev ^ 2, i, .9, nrow(fbi_centered)))
rbind(summary(fbi_pca)$importance, "P-value" = pvals)
```

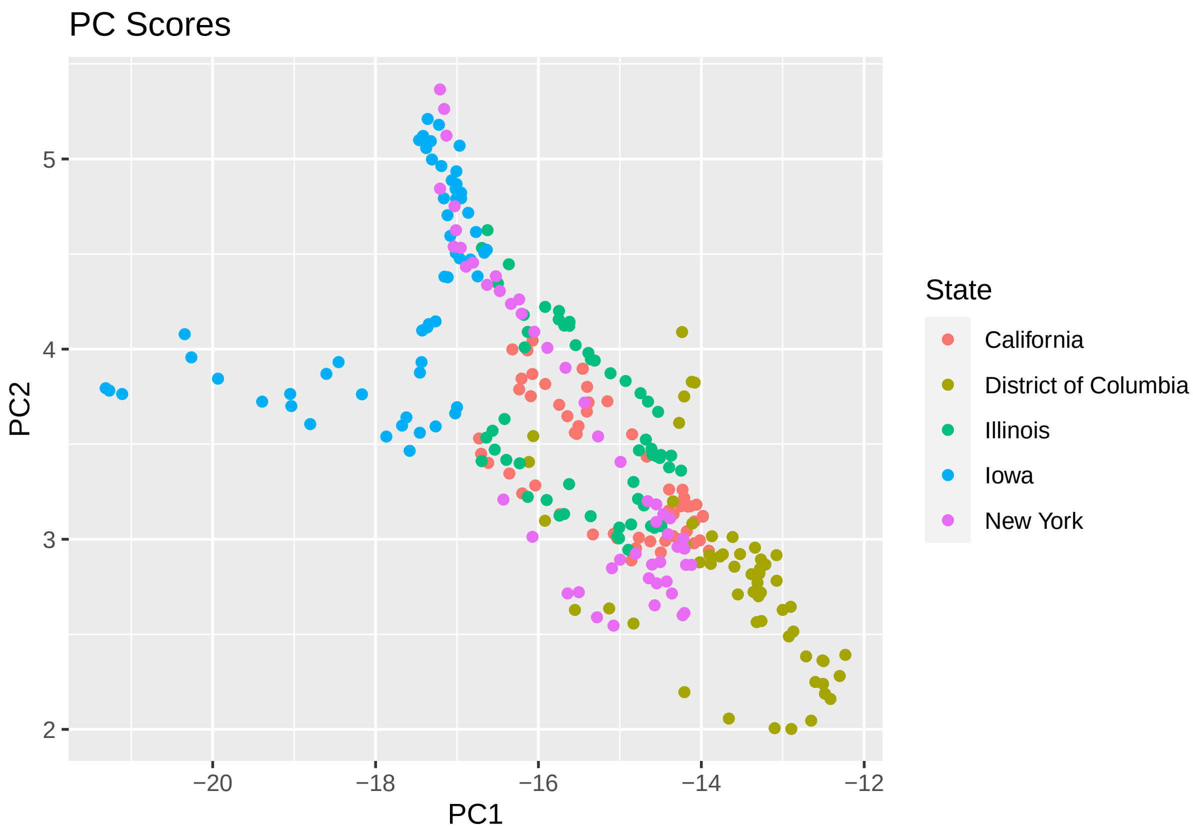
```
##                              PC1      PC2      PC3    PC4    PC5    PC6    PC7
## Standard deviation      9.84e-01 4.69e-01 2.73e-01 0.2454 0.2126 0.1895 0.1226
## Proportion of Variance  6.82e-01 1.55e-01 5.27e-02 0.0424 0.0318 0.0253 0.0106
## Cumulative Proportion   6.82e-01 8.37e-01 8.90e-01 0.9323 0.9641 0.9894 1.0000
## P-value                8.51e-243 1.72e-72 2.31e-05 1.0000 1.0000 1.0000 1.0000
```

Thus, at least 4 principal components are necessary to explain 90% of the variarion with significance level 0.05.

- The first PC is the average of the crimes, with more emphasis on violent crimes (`Aggravated.assault`, `Legacy.rape` and `Robbery`).
- The second PC is a contrast of the common violent crimes versus the non-violent crimes.
- The third PC is a contrast of the severe violent crimes (`Murder` and `Aggravated.assault`) versus the other.

f. Is there any distinctiveness of the states `California`, `Iowa`, `Illinois`, `District of Columbia` and `New York` in the first two principal components? (Transformed versions of sample means of each state need to be added back on the PC scores.) [10 points]

```r
fbi_score <- as.matrix(fbi[, -(1:4)]) %*% fbi_pca$rotation
data.frame(fbi_score[, 1:2], State = fbi$State) %>%
  filter(State %in% c("California", "Iowa", "Illinois", "District of Columbia", "New York")) %>%
  qplot(x = PC1, y = PC2, color = State, data = ., main = "PC Scores")
```



It can be seen that D.C. and Iowa can be distinguished out but the remaining 3 states are overlapped.

2. [30 points]

The United States Postal Service has had a long-term project to automating the recognition of handwritten digits for zip codes. The data on different numbers of specimens for each digit are available in Canvas. Each observation is in the form of a 256-dimensional vector of pixel intensities. These form a 16×16 image of pixel intensities for each letter. The objective is to distinguish one digit from another.

a. We will see whether the digits are distinguishable. To do so, we will first prepare the dataset by rooting out those pixels (coordinates) which do not contribute to categorization. Do so, using univariate anova test for each coordinate. Choose the 100 most significant coordinates (in terms of the p-value for the above test). [10 points]

```
zip_x <- read.table("ziptrain.dat")
zip_y <- read.table("zipdigit.dat", col.names = "class", colClasses = "factor")
apply(zip_x, 2, function(x) anova(lm(x ~ zip_y$class))$`Pr`[1]) %>% order() %>% head(100)
```

```
##   [1] 104 105 120 121 136 152 168 213 219 230 204 169 214 184 196 185 220 189
##  [19] 197 180 137 153 229 205 235  89 212 164 173  88 234  72 116 231 154 170
##  [37] 179 163  56 101 132  85 148  28  27  76 181 188 147 131 123  73 100  38
##  [55] 200  22  60 107  23 115 122 139 174 138  57 195  69 117 247 158 157  86
##  [73]  44  91  92  54 142 126  43 201 167 165 102  29 106 182 203  70  59  75
##  [91] 190 110 198 233  40  12  84  53  26 211
```

b. We will now use principal components to reduce dimensionality of the original dataset. Note that the images for the different digits have different means and chanracteristics, therefore, it would be preferred to remove the effect of the digit-specific means before performing the principal components analysis. (Transformed versions of these means need to be added back on the PC scores.) Use the principal components and determine the number of components needed to explain at least 80% of the total variation in the data, at the 5% level of significance. [10 points]

```
zip <- cbind(zip_y, zip_x)
zip_centered <- zip %>%
  group_by(class) %>%
  mutate_at(vars(-group_cols()), ~ . - mean(.)) %>%
  ungroup() %>%
  select(-class)
zip_pca <- prcomp(zip_centered)
pvals <- sapply(1:ncol(zip_centered), function(i)
  PCs.proportion.variation.enuff(zip_pca$sdev ^ 2, i, .8, nrow(zip_centered)))
# Number of PCs
min(which(pvals >= 0.05))
```

```
## [1] 39
```

c. Use `ggplot` to display the leading components (using color or characters for each digit). [10 points]

```
zip_score <- as.matrix(zip[, -1]) %*% zip_pca$rotation
data.frame(zip_score[, 1:3], digit = zip$class) %>%
  GGally::ggpairs(aes(colour = digit, size = I(.2), alpha = I(.5)), columns = 1:3)
```